

## Pourquoi Java s'appelle Java

Je ne sais pas si vous vous êtes déjà posé la question suivante: "Mais pourquoi Java s'appelle Java ?". Moi si... :-). La version que j'avais entendue, c'est que les développeurs de SUN cherchaient un nom pour le nouveau langage, James Gosling et ses copains avaient plus ou moins trouvé "Oak" un nom un peu space, mais pas franchement sexy. Et puis à force de réunion autour de leur machine à café, l'un d'eux eu l'idée de prendre Java, ce qui correspond à notre Kawa en français. Et oui, c'est simplement le mot "coffee" en slang, l'argot américain. Et puis cela tombait bien, les gens du marketing allaient pouvoir se défouler sur la communication et la publicité. C'est pas beau tout cela ? A moins que tout ceci ne soit qu'un énorme buzz et que je sois dans l'erreur depuis tant d'années... Bref moi j'ai trouvé que "Touilleur Express" c'est un très beau nom marketing pour mon blog.

## Le paradoxe des spécifications et des rapports de TP

En informatique, lorsque l'on est étudiant, on passe des heures avec son binôme à rédiger des rapports de TP une fois vos travaux pratiques terminés, votre distributeur de café en Java écrit ou votre simulateur de lavage de voiture programmé en C. Et un jour vous voilà dans le monde du travail. Là votre manager se fiche de savoir après qu'un projet s'est achevé si celui s'est bien déroulé. Par contre avant de commencer, vous aurez à rédiger des spécifications fonctionnelles (qu'est-ce que votre programme va faire, quel est le contexte ? qu'est-ce qu'il saura faire ou pas ?) et des spécifications techniques (voilà comment vous compter le faire, ici les scénarii, là le moyen de le tester...).<BR> C'était la pensée du jour.

## James Gosling a aussi son blog apres tout

En cherchant environ 7.43 ms via Google j'ai trouvé le blog de James Gosling. Le look&feel est sympa et on peut suivre un peu l'actualité de James. Son blog est dispoà cette url: <http://weblogs.java.net/jag/> (autre fenetre).

## Sous windows, comment déterminer quel process écoute sur quel port ?

Je suis entrain de configurer JBoss pour faire du clustering et lorsque je le démarre, j'ai une exception "Port already in use". Après avoir fait un "netstat -an | findstr :1098" (on est sous windows, n'oubliez pas) effectivement, il semble qu'un autre process soit déjà à l'écoute. Enfin cela ne m'avance pas plus. En cherchant sur le net j'ai trouvé un freeware très sympa qui liste sous Windows les processus ainsi que les ports TCP ou UDP utilisés. La merveille s'appelle TCPView, développé par Mark Russinovich. Le site web : [est ici](#). A télécharger si vous bossez sous windows.

## ByteArrayOutputStream: vive les buffers de taille variable

Le cas classique: vous êtes entrain de lire le contenu d'un fichier, ou vous téléchargez une page HTML dans votre programme. Simplement vous ne connaissez pas à l'avance la quantité de données, en conséquence de quoi vous ne pouvez pas de manière sûre allouer un buffer. Et bien grâce à la class `java.io.ByteArrayOutputStream` il est assez simple de lire des données sur un `InputStream` et d'écrire à la volée dans ce buffer vos données. La réallocation est effectuée en interne par le `ByteArrayOutputStream`. J'ai écrit un petit exemple d'une class qui se connecte sur un site web, télécharge le contenu et retourne une `String`.

```
/* TestBytes * Created by : nicmarti * Date: 27-Nov-2003 - Time: 13:44:28 * * Copyright (c) 2003 Nicolas Martignole * All rights reserved.
```

Enfin si vous voulez relire le contenu de ce `ByteArrayOutputStream` vous pouvez aussi réutiliser un `ByteArrayInputStream`. Cela permet de rejouer par exemple avec les retour-chariots, une suite de commande sur un serveur telnet ou ftp. Enfin plus clairement, si vous avez un buffer de bytes (ou une `String`) vous pouvez relire le tout facilement via un `ByteArrayInputStream`.

## Exemple d'utilisation de java.util.BitSet

La class `java.util.BitSet` permet de positionner facilement des bits dans un tableau à vrai ou faux et de faire

ensuite des opérations booléennes directement sur l'objet `BitSet`.

Autre avantage, ce tableau est un vecteur dont nous n'avons pas à gérer la taille,

l'allocation est masquée et permet de s'affranchir de tableaux statiques.

L'intérêt d'utiliser la class `java.util.BitSet`?

La class `BitSet` a des fonctions simples, rapides et puissante. Les opérateurs booléens sont des opérations Java qui sont appelées pour modifier le `BitSet`.

Si nous prenons un exemple concret, cela va plus vous parler.

Dans une application vous devez lire une `Map` et positionner des drapeaux si un champ de la `Map` existe. On ne vérifie pas si pour une clé donnée il existe une valeur, les valeurs

vides étant acceptées dans mon exemple. D'autre part afin d'éviter de parcourir

à chaque fois la `Map`, nous voulons garder un `BitSet` qui va représenter

les champs renseignés dans la `Map`.

Pour créer un `BitSet` c'est simple:

```

BitSet bitSet=new BitSet();
bit.set(2); // 100
bit.set(5); // 100100
bit.set(0); // 100101

```

Pour voir si au moins un des bytes est positionné à true, la fonction `intersects(BitSet ensemble)` est très pratique:

```

BitSet first = new BitSet();
first.set(2); // 100
System.out.println("first=" + first);

BitSet second = new BitSet();
second.set(4); // 10000
second.set(3); // 11000
System.out.println("second=" + second);

BitSet third = new BitSet();
third.set(0, 3); // positionne de 0 à 2 --> 111

System.out.println(first.intersects(second));
System.out.println(first.intersects(third));
System.out.println(second.intersects(third));
System.out.println("third="+third);

```

L'exécution de ce petit exemple nous donne:

```

first={2}
second={3, 4}
false
true
false
third={0, 1, 2}

```

Reprenons l'exemple de parsing d'une Map. J'ai écrit un petit programme d'exemple qui parcourt une `java.util.Map` et en suivant le pattern *Visitor* va positionner des bits à true dans un object

`java.util.BitSet`.

La fonction retourne donc un `BitSet` qui est en quelques sortes un cliché des champs qui existent dans la Map spécifiée. Ensuite il devient facile de faire des opérations booléennes sur ce `BitSet`.

Voici l'exemple complet:

```

/* testBitSet
 * Created by : nicmarti
 * Date: 04-Dec-2003 - Time: 16:10:05
 */
import java.util.*;
import java.util.logging.Logger;

/** testBitSet is a sample for BitSet.
 * @version 04-Dec-2003 16:10:05
 * @author Nicolas Martignole
 */
public class testBitSet {
    public static final String TO = "TO";
    public static final String SUBJECT = "SUBJECT";
    public static final String BODY = "BODY";
    public static final String FROM_ADDRESS = "FROM_ADDRESS";
    public static final String FROM_NAME = "FROM_NAME";
    public static final String DEVICE_CHARSET = "DEVICE_CHARSET";

    public static final int HAS_TO=1;
    public static final int HAS_SUBJECT = 2;
    public static final int HAS_BODY = 3;
    public static final int HAS_FROM_ADDRESS = 4;
    public static final int HAS_FROM_NAME = 5;
    public static final int HAS_DEVICE_CHARSET = 6;

    /** Check the Map for field, return a BitSet that is a snapshot
     * of the current Map configuration.
     */
    private static BitSet parseHeaders(Map msg) {
        BitSet mask = new BitSet();

        for (Iterator i = msg.keySet().iterator(); i.hasNext();) {
            String curs = i.next().toString();
            if (curs.toUpperCase().equals(TO)) {
                mask.set(HAS_TO);
            }
            if (curs.toUpperCase().equals(SUBJECT)) {
                mask.set(HAS_SUBJECT);
            }
            if (curs.toUpperCase().equals(BODY)) {
                mask.set(HAS_BODY);
            }
            if (curs.toUpperCase().equals(FROM_ADDRESS)) {
                mask.set(HAS_FROM_ADDRESS);
            }
            if (curs.toUpperCase().equals(FROM_NAME)) {

```



catastrophiques dont certains se servent dans leur code...

## CVS pour remplacer VSS

J'ai écrit un article ("white-paper") pour mon manager : "Why using CVS instead of VSS ?" avec beaucoup de plaisir. Historiquement ma société utilise VSS de Microsoft pour la gestion des sources. VSS n'est pas un mauvais produit, mais il y a quelques limitations qui nous empêchent d'être plus efficace. Je peux faire ici de l'évangélisation pour CVS face à VSS, mais j'ai l'impression que ces 2 produits ne sont pas comparables. Tout simplement.

### Les bonnes raisons pour passer de VSS à CVS:

1. Il existe des clients CVS sous Solaris et Linux.
2. CVS est un produit robuste, open-source, libre et sûr alors que VSS est connu pour sa capacité à corrompre une base
3. CVS est plus rapide pour les opérations de mise à jour que VSS. Je télécharge notre arborescence VSS en 14 mn alors qu'il me faut 2mn43 pour faire un "cvs checkout"
4. CVS propose différent moyen d'accès réseau: ssh, local, rsh, kerberos alors que VSS utilise des montages SMB (disque réseau) et ne marche pas à distance. Il vous faut un VPN avec des disques réseaux montés sous Windows.
5. Microsoft n'a pas mis à jour VSS depuis fin 2002 alors que les nouvelles versions de CVS pour fixer des bugs ou des problèmes de sécurité sont mise à jour régulièrement.
6. Lorsque CVS est utilisé avec l'éditeur Java [IntelliJ IDEA](#) un nouveau panel est ajouté. Il permet de voir en 1 clique les fichiers modifiés localement. Plus globalement, CVS est reconnu par casiment tous les éditeurs Java: Netbeans, JBuilder, Forte ou SUN One studio (pareil que netbeans) et Eclipse. Mais pas par VisualAge qui est une antiquité de toute façon.
7. CVS est bien intégré avec [Ant](#) dans les makefiles et permet de faire des scripts d'auto-compilation nocturnes (nightly build) ou des scripts de packaging de sources facilement
8. CVS est capable de faire des branches et de merger des sources. VSS est une horreur avec les Labels. Si vous réappliquez le même label VSS sur une arbo, que vous faites ensuite une opération de Checkout en utilisant ce Label, vous ne récupérez pas les fichiers les plus récents mais les premiers qui avaient eu ce Label...
9. CVS et [InstallAnywhere de Zerog.com](#) travaillent ensemble à partir de Ant: vous pouvez sortir vos sources de CVS, les compiler et enfin les packager avec InstallAnywhere. Et ce, à partir de Solaris, Linux et Windows.
10. CVS ne coûte rien à l'achat, il faut par contre prendre le temps de bien l'apprendre du côté administrateur, il ne faut pas le jeter en pâture aux développeurs qui n'ont jamais utilisé un gestionnaire de sources ou pire qui ont déjà travaillé sur VSS avant.

CVS est un logiciel qui vous fera gagner de l'argent ou CVS is really a money-maker tool

## Quelle est la différence entre un Iterator et une Enumeration ?

La class java.util.Vector est sans doute une des classes Java les plus pratiques. Depuis la sortie de Java 1.2 lorsque vous devez énumérer (lister) le contenu d'un Vector, vous pouvez utiliser la méthode "à l'ancienne" comme dans cet exemple:

```
Vector v=new Vector();
String prenom=new String("Arnold");
String nom=new String("Governator of California");
Integer numb=new Integer(3);

v.addElement(prenom);
v.addElement(nom);
v.addElement(numb);

for(Enumeration e=v.elements() ; e.hasMoreElements() ; ) {
    System.out.println(e.nextElement());
}
```

Lorsque ce programme est exécuté nous obtenons:

```
Arnold Governator of California 3
```

Maintenant si je rajoute à ce programme la version java 1.2 utilisant un iterator nous avons:

```
Vector v = new Vector();
String prenom = new String("Arnold");
String nom = new String("Governator of California");
Integer numb = new Integer(3);

v.addElement(prenom);
v.addElement(nom);
v.addElement(numb);
System.out.println("Version java 1.1:");
for (Enumeration e = v.elements(); e.hasMoreElements(); ) {
    System.out.print(e.nextElement());
    System.out.print(' ');
}
System.out.println("\nVersion java 1.2:");
for(Iterator i=v.iterator() ; i.hasNext();){
    System.out.print(i.next());
```

```

        System.out.print(' ');
    }

```

L'exécution donnera:

```

Version java 1.1:
Arnold Governor of California 3
Version java 1.2:
Arnold Governor of California 3

```

La question est la suivante: **quelle est la différence entre un Iterator et une Enumeration ?** la réponse se trouve dans [la javadoc de la classe java.util.Vector](#) dans ce paragraphe: *[...]The Iterators returned by Vector's iterator and listIterator methods are fail-fast: if the Vector is structurally modified at any time after the Iterator is created, in any way except through the Iterator's own remove or add methods, the Iterator will throw a ConcurrentModificationException. Thus, in the face of concurrent modification, the Iterator fails quickly and cleanly, rather than risking arbitrary, non-deterministic behavior at an undetermined time in the future. The Enumerations returned by Vector's elements method are not fail-fast.*

En clair cela veut dire que si vous utilisez un Iterator lors de la lecture du Vector, vous êtes assuré que si quelqu'un venait à modifier un des éléments du vecteur, alors une exception `ConcurrentModificationException` sera levée, ce qui n'est pas le cas lorsque l'on utilise une Enumeration. Dans le cas d'un programme multi-threadé, il est donc important d'utiliser ce nouveau moyen pour s'assurer que les données utilisées sont consistantes.

## Et si on regardait ce qu'il y a dans cette HashMap ?

Aujourd'hui je n'ai pas trop d'inspiration alors j'ai regardé dans les Referers ce qui m'arrivait via Google. Une des query était "lister le contenu d'une HashMap". Je vais parler un petit peu de cette classe.

Tout d'abord rapidement, qu'est-ce qu'une [HashMap](#) en Java ? C'est une des nouvelles classes de l'API Java Collection qui permet de stocker dans un tableau une association de clé/valeur. Il s'agit du principe de l'interface `java.util.Map`. Les clés ne peuvent pas être null mais par contre, les valeurs "null" sont autorisées pour une HashMap.

Prenons l'exemple d'une HashMap qui contient en tant que clé le prénom d'une personne et en tant que valeur, son âge sous la forme d'un Integer. Cela nous donne:

Clé (String)	Valeur (Integer)
Zinedine Zidane	28
Michael Jackson	44
Georges W.Bush	12
Footix	4

Note: quand je disais que j'étais en manque d'inspiration, je le prouve encore ici :-)

Une HashMap est quasi identique à la class `java.util.Hashtable` que vous avez sans doute aussi croisé. Les différences sont que la HashMap n'est pas synchronisée et que vous pouvez avoir une valeur "null" alors qu'avec une Hashtable cela n'est pas possible. Cela complique un peu la chose parfois car si vous voyez une clé présente, vous devrez peut-être tester que la valeur associée n'est pas nulle...

Cette class n'est pas intéressante pour trier les données, et en plus lorsque vous allez l'itérer, si vous ajoutez ensuite d'autres éléments vous verrez peut-être que l'ordre des autres éléments a changé. Bref ne vous en servez pas pour lister par ordre des éléments. Les méthodes `get()` et `put(Object o)` par contre vous donneront un temps d'accès quasi-identique que vous accédez au premier ou au dernier élément de cette Map. Visualisez un arbre en mémoire sur lequel les éléments sont répartis.

Enfin comme d'autres objets Java du framework Collection, vous n'avez pas à vous soucier de l'allocation et de la taille de la HashMap. De toute façon vous savez comme moi que les problèmes d'allocation sont mis de côté à partir du moment où vous travaillez avec ces objets. Merci Java :-)

**Revenons à nos moutons** enfin à notre liste d'invités et voyons comment itérer sur les clés, comment lister une HashMap. J'ai écrit rapidement le petit bout de code suivant:

```

// Creation d'une HashMap
Map map=new HashMap();
map.put("Zinedine Zidane",new Integer(28));
map.put("Michael Jackson", new Integer(44));
map.put("Georges W.Bush", new Integer(12));
map.put("Footix",new Integer(4));

// Liste les clés
for(Iterator i=map.keySet().iterator();i.hasNext();){
    Object key=i.next();
    System.out.println(key + "=" + map.get(key));
}

```

Après compilation voici le résultat de l'exécution:

```

Michael Jackson=44
Footix=4
Georges W.Bush=12
zinedine zidane=28

```

Comme je le disais précédemment on voit que l'ordre d'affichage n'est pas l'ordre d'insertion.

Pour lister le contenu d'une HashMap, vous devez acquérir un Iterator sur la liste des clés, puis ensuite vous servir de cet Iterator pour lister le contenu. Pourquoi ? parce que la class Iterator va vous garantir que quelqu'un d'autre n'a pas modifié la HashMap alors que vous étiez entrain de l'afficher justement. Cela n'est pas garanti, car on ne maîtrise pas la synchronisation, mais si cela arrive vous aurez une exception ConcurrentModificationException. Voir la javadoc de la class HashMap pour plus d'informations.

Enfin si vous voulez lister uniquement les valeurs sans lister les clés, bien que je ne vois pas trop l'intérêt, vous utiliserez la méthode values qui retourne un objet Collection sur lequel vous allez ensuite pouvoir utiliser un Iterator:

```
public static void test3() {
    // Creation d'une HashMap
    Map map = new HashMap();
    map.put("Robert", new Integer(28));
    map.put("Roger", null);
    map.put("Ghislaine", new Integer(12));
    map.put("Bernado", null);

    // Liste les clés
    for(Iterator i=map.values().iterator();i.hasNext();){
        Object o= i.next() ;
        if(o!=null){
            System.out.println(o.toString());
        }else{
            System.out.println("Valeur null");
        }
    }
}
```

A noter que j'utilise souvent des déclarations du type:

```
InterfaceName monObject=new InterfaceImplementation();
```

ce qui donne dans le code:

```
Map map=new HashMap();
```

## Jakarta contre OpenSymphony

Autant je connais bien la communauté [Apache Jakarta](#) qui héberge des projets comme Apache, Ant, Tomcat, Log4J ou Maven, autant je n'avais jamais entendu parler d'[OpenSymphony](#) jusqu'à lire un article sur [TheServerSide.com](#) de Mike Cannons un des développeurs qui est à la base de OpenSymphony.

OpenSymphony propose **WebWork**, une plateforme J2EE pour le développement facile et rapide des applications Webs. C'est un concurrent direct de [Struts](#) destiné à simplifier le développement en Java des servlets, des pages JSP et des pages HTML représentées.

Je testerai WebWork prochainement afin de voir le gain de productivité par rapport à Struts.

## Adieu System.out.println, vive Log4J

Je vais faire de la pub pour Log4J, une API Java développée par le groupe Apache. Log4J permet d'ajouter facilement à une application Java la possibilité de loguer des messages d'info, de debug, d'erreur etc. Les messages peuvent être affichés directement sur la console, enregistré dans un fichier, envoyé à un serveur distant via Socket, ajouté aux syslogs unix ou au journal des événements de Windows. Bref vraiment très puissant. Le formatage des messages est facilement configurable, il est possible d'activer/désactiver l'affichage d'une trace en temps réel, de faire tourner les fichiers de logs lorsqu'ils sont pleins... .

Les 3 concepts à retenir de log4j: logger, appender et layout: un **logger** logue un message vers un **'appender'** en utilisant un **'layout'**. [Le site d'Apache Log4J](#) **Exemple avec un fichier de configuration XML** Voici un exemple d'une class Java utilisant un logger, dont la configuration est externe au programme et est effectuée dans un fichier XML:

```
/* TestLog4JXML * Created by : nicmarti * Date: 27-Nov-2003 - Time: 12:40:50 * Copyright(c) 2003 Nicolas Martignole. */package com.test.nicmarti;
```

Voici le fichier de configuration **log4j.xml** que j'ai défini:

```
<?xml version="1.0" encoding="UTF-8" ?><!DOCTYPE log4j:configuration SYSTEM "log4j.dtd"><!-- =====
```

En conclusion, nous voyons que Log4J est très pratique pour mettre en place rapidement et proprement une gestion des logs dans une application. Voili voilà.

## Ant et SQL

[Ant](#) est un excellent outil de compilation écrit en Java qui permet de définir des makefile en XML et d'exécuter ensuite un grand nombre d'opérations. Compilation, documentation, tests unitaires, packaging, ftp, envoi d'email, gestion de CVS ou VSS et même execution.

J'ai écrit des tests d'intégration avec Ant, en utilisant un build.xml dans lequel je déploie le logiciel sur lequel je travaille. Grâce à Ant j'exécute une suite de commandes SQL directement à partir de Ant, mais je peux aussi faire appel à des scripts SQL.

Cela donne un peu près cela:

```
<property name="oracle.jdbc" value="ojdbc14.jar"/>
<property name="login" value="nicmarti" description="DB Login"/>
<property name="pass" value="nicmarti" description="DB Password"/>
<property name="oracle.host" value="getix02" description="Oracle Database hostname"/>
<property name="oracle.port" value="1521" description="Oracle Port"/>
<property name="oracle.sid" value="DEV9i" description="Database instance"/>

...
...
<!-- =====//-->
<!-- Delete Web App Tables on a oracle database //-->
<!-- =====//-->
<target name="oracle.webapp.drop" description="Delete the Oracle db">
  <sql
    driver="oracle.jdbc.driver.OracleDriver"
    url="jdbc:oracle:thin:@{oracle.host}:{oracle.port}:{oracle.sid}"
    userid="{login}"
    password="{pass}"
    onerror="continue"
    delimiter=";"
    src="{oracle.dir}/drop_webapp_tables.sql"
  >
  <classpath>
    <pathelement location="{lib.dir}/{oracle.jdbc}"/>
  </classpath>
</sql>
</target>
...
...
```

## Astuce avec le ClassLoader

**Comment savoir à partir de quel fichier Jar une class Java est chargée ?** Pour connaître le nom du fichier JAR à partir duquel une class Java spécifique a été chargée, utilisez la commande -verbose:classpath. Pratique lorsque vous n'êtes pas certain de votre classpath ainsi que l'ordre des fichiers Jar. C:\Test>java -verbose:classpath com.reuters.test.MyClass

```
[Opened C:\j2sdk1.4.1_02\jre\lib\rt.jar][Opened C:\j2sdk1.4.1_02\jre\lib\sunrsasign.jar][Opened C:\j2sdk1.4.1_02\jre\lib\jsse.jar][Opened C:\j2
```

## Synchronisation avec Vector, Hashtable, HashMap et ArrayList

Tout d'abord un peu de théorie. Les class Vector et Hashtable sont des classes dont les accès sont synchronisés. Si 2 Threads manipulent un Vector, nous sommes sûrs que l'état de ce Vector est consistant et nous n'avons pas de problèmes d'accès concurrents. Ensuite Java 2 a introduit un nouvel ensemble de class "Collections" qui sont un formalisme de l'interface Collection et de l'interface Map. Ainsi la class Hashtable dans Java 2 implémente l'interface Map, et la class ArrayList est identique à la class Vector. Il y a cependant une différence de taille: ces nouvelles classes ne sont pas synchronisées. Un object ArrayList sans protections particulières de notre part, n'est pas synchronisé.

Quel est l'intérêt ?

A partir du moment où la méthode add(Object o) de la class ArrayList par exemple n'est pas synchronisé, vous pouvez maintenant effectuer plusieurs opérations au sein d'un bloc synchronized dans votre code.

Exemple:

Dans cet exemple simpliste nous voulons ajouter 1 objet à la fois dans un Vector. Une autre thread est susceptible d'utiliser la méthode getLastest ici. Comme la class Vector est synchronisé il n'est pas nécessaire et même inutile de rajouter 2 synchronized sur nos méthodes. Cela reviendrait à avoir 2 synchronisations imbriquées avec un risque d'interblocage.

```
private Vector list=new Vector();

public void storeId(Object first){
    list.add(first);
}

public Object getLastest() {
    return list.lastElement();
}
```

Cependant si nous voulons enregistrer 2 objets à la fois à l'aide d'une seule méthode et que notre getLastest() ne puisse retourner une valeur qu'une fois nos 2 ajouts effectués alors nous aurions le code suivant:

```
private Vector list=new Vector();

public synchronized void storeIds(Object first,Object second){
    list.add(first);
    list.add(second);
}

public synchronized Object getLastest() {
```

```

        return list.lastElement();
    }

```

et là nous avons un soucis. En entrant dans la méthode storeIds, Java acquiert un verrou et bloque la lecture sur ce vecteur. Ensuite lorsque nous effectuons nos 2 opérations add() nous avons à chaque fois un autre verrou qui est inutile...

La solution est d'utiliser un objet ArrayList non-synchronisé et de synchroniser nous-même les accès. A noter que l'instance est privée car nous ne voulons pas qu'une autre class référence directement notre liste. En règle générale le référencement externe des variables d'instance d'une class est quelque chose à éviter. Autant refaire du C sinon.

```

private ArrayList list = new ArrayList();

public void storeIds(Object first, Object second){
    synchronized(list){
        list.add(first);
        list.add(second);
    }
}

public synchronized Object getLastest() {
    synchronized(list){
        return list.lastElement();
    }
}

```

Voilà voilà...

## Lotus Notes sous Linux avec Wine

J'ai écrit il y a un moment un article sur Lotus Notes et Linux. Voici une page qui explique comment faire fonctionner Lotus Notes mais aussi Excel ou Word, sous Linux:

<http://www.winecentric.com/notes5.shtml>

## Comment lister le contenu d'un fichier jar ?

Un ami me demandait comment faire pour comparer 2 fichiers jar et en voir le contenu. La commande jar est similaire à la commande tar sous unix. Si en ligne de commande vous faites un "jar tvf toto.jar" vous allez lister le contenu du fichier jar (t=tape, v=verbose et f=file was specified). Sous windows, dans une fenêtre terminal Dos, vérifiez que jar est dans le path en tapant "jar" et entrée. Vous devez obtenir ceci:

```

Usage: jar {ctxu}[vfm0Mi] [jar-file] [manifest-file] [-C dir] files ...
Options:
  -c create new archive
  -t list table of contents for archive
  -x extract named (or all) files from archive
  -u update existing archive
  -v generate verbose output on standard output
  -f specify archive file name
  -m include manifest information from specified manifest file
  -0 store only; use no ZIP compression
  -M do not create a manifest file for the entries
  -i generate index information for the specified jar files
  -C change to the specified directory and include the following file
If any file is a directory then it is processed recursively.
The manifest file name and the archive file name needs to be specified
in the same order the 'm' and 'f' flags are specified.

```

Example 1: to archive two class files into an archive called classes.jar:

```
jar cvf classes.jar Foo.class Bar.class
```

Example 2: use an existing manifest file 'mymanifest' and archive all the files in the foo/ directory into 'classes.jar':

```
jar cvfm classes.jar mymanifest -C foo/ .
```

Pour visualiser votre fichier jar toto.jar vous ferrez donc un jar tvf toto.jar > liste\_fichier.txt par exemple. Et sous unix avec grep et compagnie, il devient facile de comparer 2 fichiers.

Autre astuce: avec Winzip vous pouvez ouvrir les fichiers jars sous Windows. Alors pourquoi s'en priver ?

## API pour les Threads, la Synchronisation, les ThreadPool

Article original en anglais sur [http://www.theserverside.com/articles/article.tss?t=JavaOneDayFour\\_03](http://www.theserverside.com/articles/article.tss?t=JavaOneDayFour_03)

Je me suis permis de traduire en français un article à propos des prochaines fonctionnalités pour J2SE 1.5.

JSR-166 représente une amélioration majeur des capacités concurrentielles de Java en introduisant le package **java.util.concurrent**, potentiellement inclus dans J2SE 1.5. Ce package se base sur le travail de [Doug Lea](#) dont le package a été adopté et utilisé de nombreuses fois, et inclus des utilitaires largement utilisés dans l'industrie.

Ce qui est fascinant à propos de ces bibliothèques c'est qu'elles n'utilisent pas ou peu la synchronisation de Java. Au lieu de cela, la bibliothèque inclue une partie de code natif afin d'utiliser les spécificités de chaque plateforme. Cela devrait conduire à une grande amélioration des capacités et des performances par rapport à l'utilisation de **synchronized**.

Exemples d'outils ou de class dans ce kit:

- Executors, un framework pour effectuer des invocations asynchrones. Essayez d'éliminer la nécessité de piloter nous-mêmes des Threads d'exécution. Cela permettrait d'utiliser des Thread Pools, des tâches annulables, des threads de shutdown...
- Différents nouveaux objets comme une PriorityQueue, une LinkedQueue rapide, thread-safe et non bloquante, une BlockingQueue (utilisé dans les patterns producteur/consomateur). Une ConcurrentHashMap est aussi incluse permettant de manière propre et sûre des lectures et écritures concurrentes, la majorité des lectures étant non-bloquantes.
- Une class TimeUnit pour des opérations dont la granularité serait les nanosecondes, en assumant que l'OS supporte lui-même ce niveau de timing
- Des objets ReadWrite locks permettant à plusieurs lecteurs d'accéder à une valeur mais n'autorisant qu'un seul écrivain (wa wa wa) et ce afin de remplacer les fameux synchronized block que nous voyons dans le code. C'est sans doute l'un des ajouts les plus sympathiques mais qui à terme rendra obsolète des outils comme JProfiler...

Bref c'est un projet prometteur et qui devrait introduire une plus grande facilité (le mot d'ordre de Java One en 2003) ainsi qu'une architecture plus propre dans Java 1.5.

[Concurrency JSR-166 Interest Site](#)

## Ant et les couleurs ANSI sous Unix

[Ant](#) est un outil de Makefile pour Java. Si vous utilisez Ant sous Unix, il est possible de modifier le logger utilisÃ© par Ant lorsque vous compilez un projet afin d'avoir un affichage dans votre xterm utilisant les couleurs ANSI. Pour cela, voici comment procÃ©der:

- CrÃ©er un fichier .ant\_color dans votre repertoire \$HOME contenant:

```
AnsiColorLogger.ERROR_COLOR=2;31 AnsiColorLogger.WARNING_COLOR=2;35 AnsiColorLogger.INFO_COLOR=2;36 AnsiColorLogger.VERBOSE_COLOR=2;32 Ans
```

- Dans votre fichier .bash\_profile ou .bashrc, ajoutez une variable d'environnement ANT\_OPTS. Voici le dÃ©tail de mon fichier .bash\_profile sous Linux:BR>

```
# .bash_profile # Get the aliases and functions if [ -f ~/.bashrc ]; then . ~/.bashrc fi# ANT export ANT_OPTS=-Dant.logger.defaults=
```

- Placez-vous dans un rÃ©pertoire contenant un fichier build.xml puis utilisez:

```
ant -logger org.apache.tools.ant.listener.AnsiColorLogger
```

Pour plus de dÃ©tails regardez dans la section **Logger and Listeners** de la documentation de Ant [Ã](#) [cette adresse](#)

## Log4J et log4j.xml

En reprenant un ancien exemple que j'avais Ã©crit sur Log4, je montrais comment charger facilement un fichier XML de configuration pour ouvrir/fermer des traces. Cependant mon appel systÃ©me pour charger le fichier en question n'est pas des plus pratique. Si le fichier en question n'est pas dans le rÃ©pertoire d'exÃ©cution de l'application, alors Log4J ne peut pas s'initialiser et l'application plante. Pour Ã©viter cela, le plus simple est de placer le fichier log4j.xml dans le classpath de votre application puis d'utiliser le ClassLoader du system pour trouver et charger ce fichier.

En reprenant le petit exemple sur log4j que j'avais Ã©crit il y a un moment, voici ce que j'ajouterai au dÃ©but du programme:

```
public class EJBSampleWithPDS {
    static Logger log = Logger.getLogger(EJBSampleWithPDS.class);

    /** My Main for testing.
     * @param args the command line
     */
    public static void main(String[] args) {

        // Use the system class loader.
        URL u=ClassLoader.getSystemClassLoader().getResource("log4j.xml");
        if(u==null){
            throw new RuntimeException("Please put a valid log4j.xml in the classpath");
        }
        DOMConfigurator.configure(u);

        log.info("Application starting...");

        // Use JNDI to lookup EJB's Home interface
        Properties props = new Properties();
        ...
        ...
    }
}
```

## Acide caustique de qualitÃ© sur Java en gÃ©nÃ©ral et

## n'importe quoi en particulier.

Envie d'un peu de lecture en anglais pour améliorer votre vocabulaire ? J'ai trouvé un blog sympa dans lequel vous pouvez piocher un sujet souvent assez caustique et décupant sur la communauté Java, les outils open-sources et les news autour de Java.

De ce que j'ai lu, Hani Suleiman (core developer chez [OpenSymphony.com](http://OpenSymphony.com)), l'auteur, a [une dent contre Maven \(Maven\)](#) ainsi que contre les grandes discussions sur "est-ce que Java doit être OpenSource ou pas ?". Quelques articles sont carrément violent pour les auteurs et les contributeurs des projets open-source (JBoss.org), d'autres sont justifiés et plein de bon sens. Mon favori: pourquoi Ant utilise le langage XML comme un langage de commandes et de scripting ?

L'adresse de **The BileBlog**: [BileBlog](#)

## JBoss clustering

JBoss 3.x supporte le load-balancing (répartition de charges) et le fail-over (reprise sur plantage) ce que l'on regroupe sous le terme 'clustering'.

Les caractéristiques de JBoss:

- Découverte automatique: les serveurs JBoss se découvrent mutuellement.
- Fail-over et load-balancing pour JNDI,RMI,Entity Beans,Stateful Session Beans avec recopie des états en mémoire, Stateless Session Beans.
- Support de la répartition de charges des sessions HTTP grace à Tomcat ou Jetty.
- Support des load-balanceurs matériels CISCO en frontal pour la partie HTTP
- Découverte automatique des serveurs JNDI
- Farming / Fermes de serveurs JBoss : cela permet d'auto déployer une webapp sur un ensemble de serveurs JBoss.
- etc...

JBoss utilise JavaGroups (<http://www.javagroups.com>) une API qui permet de créer des groupes de processus Java qui communiquent entre eux via multicast ou TCP. Cette API permet de définir des partitions qui dialoguent entre elles pour se tenir informées de leur état. C'est un peu similaire à Tibco Rendezvous, mais cette API est orientée gestion de groupes de JVM et n'est pas destinée à échanger des messages comme avec Tibco RV.

Si vous devez étudier sur un projet la mise en place du clustering pour JBoss, je vous conseille d'acheter la documentation "JBoss Clustering" rédigée par S.Labourney et B.Burke. Elle est disponible sur le site de Jboss.

## How to install SubVersion plugin for IDEA IntelliJ 4.5 under Win32 ?

I wrote a small document for my team that describes how to install the SubVersion plug-in for IDEA IntelliJ 4.5. SubVersion is a version control system that aims to replace CVS one day. IntelliJ 4.5 is my favorite Java IDE I work with.

The **svnup** plugin for IntelliJ 4.5 is still in beta mode. I spent 2 hours to make it work under Windows just because I was missing some DLL... The web site is a bit out of date, and the documentation also. I finally found some answers in the mailing list.

Here is my small how-to:

-----  
How to install the subversion plugin for IDEA IntelliJ 4.5 ?  
-----

Platform: Microsoft Windows 2000 Professional Edition.  
-----

Author: Nicolas Martignole  
Date : 19th oct 2004

Install TortoiseSVN for Win32. The zip file is available from <http://tortoisesvn.tigris.org/download.html>  
After a reboot, from the Start Menu select Settings->Control Panel  
Select "System". In the "Advanced" tab pane, select "Environment variables"  
then in the 2nd pane, add to PATH the TortoiseSVN/bin folder so that the various DLL which are located under this folder gets added to the PATH.  
Your PATH env should now contain "C:\Program Files\TortoiseSVN\bin"

The official plugin website is <http://svnup.tigris.org/>

1/ Download svnup-0.9.2.jar from <http://svnup.tigris.org/files/documents/646/16639/svnup-0.9.2.jar>

2/ Download svn4idea4-0.9.1.jar for IDEA IntelliJ 4.x. There's no 9.2 actually but that is not important. The URL is: <http://svnup.tigris.org/files/documents/646/16579/svn4idea45-0.9.1.jar>

3/ Download svn-win32-1.1.0\_javahl.zip from the link: [http://subversion.tigris.org/files/documents/15/17374/svn-win32-1.1.0\\_javahl.zip](http://subversion.tigris.org/files/documents/15/17374/svn-win32-1.1.0_javahl.zip)

4/ Unzip svn-win32-1.1.0\_javahl.zip to a temp folder, you should get 2 files:

```

- libsvnjavahl-1.dll
- svnjavahl.jar

5/ Creates a new folder named "svn4idea" under the IDEA IntelliJ 4.5 plugins folder
(C:\IntelliJ-IDEA-4.5\plugins)

6/ Creates a lib folder into this folder (e.g. C:\IntelliJ-IDEA-4.5\plugins\svn4idea)

7/ Copy svnjavahl.jar, svn4idea4-0.9.1.jar and svnup-0.9.2.jar to the lib folder
you just created (C:\IntelliJ-IDEA-4.5\plugins\svn4idea\lib)

9/ Copy libsvnjavahl-1.dll (file from svn-win32-1.1.0_javahl.zip )
to C:\IntelliJ-IDEA-4.5\bin so that IDEA IntelliJ can find it.

10/ Start idea 4.5

11/ In the File menu, select "Settings"

12/ In the Project tab, select "Version Control". A Subversion option should appear
in the Combo box.

```

It is really important that you install a SVN client such as TortoiseSVN before you intend to use the svnup plugin for IDEA IntelliJ. The JavaHL svnup DLL requires a bunch of DLL from any SVN client.

#### Update 28th oct 2004:

Thanks Michael. I corrected the file name and the how-to should be correct now.

## Comment trier un tableau en une ligne de code

Dans l'api java.util, la class Arrays est une class utilitaire qui permet de manipuler des tableaux d'objet. Vous pouvez trier un tableau de String[] par exemple. Il existe aussi des fabriques d'objets (factory) pour retourner sous forme de java.util.List un tableau.

Voici un exemple simple pour trier un tableau de chaîne qui utilise la fonction sort():

```

String[] arrays=new String[]{"uniforme", "alpha", "tango", "zoulou", "charlie", "papa", "hotel"};
System.out.println("Before:");
for(int i=0; i<arrays.length;i++){
    System.out.print(arrays[i]+" ");
}
System.out.println("After:");
Arrays.sort(arrays);

for(int i=0; i<arrays.length;i++){
    System.out.print(arrays[i]+" ");
}

System.out.println("Create a List from the array object");
List l=Arrays.asList(arrays);
for (int i = 0; i < l.size(); i++) {
    System.out.print(l.get(i)+" ");
}

```

L'exécution de ce code retourne:

```

Before:
uniforme, alpha, tango, zoulou, charlie, papa, hotel,
After:
alpha, charlie, hotel, papa, tango, uniforme, zoulou,
Create a List from the array object
alpha, charlie, hotel, papa, tango, uniforme, zoulou,

```

Pour plus de détails, voir l'API javadoc [java.util.Arrays](#)

## Rencontre avec Brian Behlendorf et présentation de CollabNet

Après un [message](#) que j'avais posté sur JBoss Inc. où je parlais de Brian Behlendorf le patron de [CollabNet Inc.](#), l'ancien webmaster de Wired, un des contributeurs du serveur web Apache et qui siège toujours au board de Apache.org... j'ai eu la chance de le rencontrer par hasard grâce à ma société.

Nous sommes actuellement sur un gros projet visant à mettre nos sources chez Collabnet. Brian est venu à Paris pour présenter sa société et ses équipes et surtout répondre aux questions des différentes équipes. Il a présenté ses produits et les solutions qu'ils proposent. c'est un excellent orateur et je retire ce que j'ai précédemment raconté... Mea culpa. Bref c'est quelqu'un avec du recul, qui a commencé à bosser en tant que WebMaster avec le serveur Web NCSA. Avec d'autres ils passaient leur temps à le patcher, à patch, d'où le nom Apache. On prononce "apachiiuu" en anglais à propos, et pas "apacheuu" si vous voulez que vos interlocuteurs comprennent.

CollabNet propose une solution d'hébergement de source code ultra-sécurisée et aussi un ensemble d'outils tels que: bug tracking, forums, mailing-list, project tracker et schedule managment. Via une interface Web, l'utilisateur dispose d'un ensemble d'outil pour travailler en réseau. Pour être plus clair, si vous connaissez [SourceForge](#) et bien CollabNet propose la même chose aux entreprises.

D'autre part CollabNet supporte activement SubVersion (SVN). En bref, SubVersion sera le remplaçant de CVS. CollabNet participe à l'effort de la communauté open-source et finance quelques développeurs du coeur du système.

Bref nous avons passé un bon moment, c'était sympa de le rencontrer en vrai.

## Parlez-vous anglais ?

**Comment prononcer SQL, Oracle, Hibernate et autres en anglais...** Pas plus tard que la semaine dernière lors d'une formation, les gens commencent par se présenter chacun à leur tour. Arrive le tour d'un gars qui explique qu'il est un ex employé de "Bee Hee"... Je reste un peu perplexe. Bien que travaillant dans un environnement anglophone, parfois certains acronymes ou termes m'échappent. Par "Bee Hee" il fallait comprendre **BEA** (béé-euh-Aaah comme disent les français) ! Je profite de ce post pour parler de quelques trucs et de la manière de les prononcer en anglais si vous voulez que vos interlocuteurs vous comprennent.

- SQL se prononce "ssi-quou-ailleu"
- SQLServer devient donc "ssi-quou-ailleu sirveur"
- Oracle - "Oreille-keule"
- POJO - "Pau-D-jo"
- Hibernate - "aie-beurre-net"

J'ai un peu exagéré les prononciations mais si quelqu'un trouve comment faire l'alphabet phonétique en HTML, je suis preneur.

## Astuce pour ClassLoader et Java

**Comment retrouver le nom complet du fichier JAR à partir duquel une Class a été chargée ?** Une Class est chargée par le ClassLoader à partir d'un fichier jar ou d'un répertoire. Si vous voulez retrouver le nom et le chemin complet d'un JAR pour une class donnée dans votre code, vous pouvez utiliser l'appel suivant:

```
// Pour retrouver d'où a été chargé la class Configuration : System.out.println("Configuration.class was loaded from: "
```

Cet appel permet normalement de retrouver le domaine d'exécution d'une Class ainsi que les Certificats ayant servi à signer le code.

## Log4J est capable de recharger un fichier de configuration XML automatiquement

On arrête pas le progrès: en fixant un bug sur notre logiciel, je me suis aperçu que Log4j est capable de recharger sa configuration régulièrement.

Log4j se configure via un fichier XML, un fichier de Properties ou directement dans votre application. Il est possible bien sûr de modifier la verbosité de votre application en effectuant des appels directement en Java. Cependant il n'est pas facile de configurer et d'ajouter des Appenders dynamiquement. La class `DOMConfigurator` de Log4J permet de charger un fichier de configuration pour Log4J au format XML. Cependant ce fichier une fois chargé, si vous veniez à modifier le fichier, vous devez redémarrer votre application. Et bien non ! La fonction `configureAndWatch(String file, long delay)` configure Log4J et rechargera le fichier automatiquement toutes les n millisecondes.

Pour plus de détails, voir la javadoc [de DOMConfigurator](#).

## Formation JBoss for Advanced J2EE developers à Berlin

### Journée 1

Depuis hier j'ai commencé la formation **JBoss for Advanced J2EE developers** à Berlin, Allemagne. La formation se déroule sur 4 jours. Je vais profiter de ces 4 jours pour noter ici mes impressions et faire partager "de l'intérieur" ce que j'aurai appris.\* La formation en anglais est faite par Kabir Khan, qui vient de Norvège, secondé par Thomas Heute, monsieur JBoss-WS en personne. Tous les deux sont des employés permanents de JBoss Inc. La formation se déroule en anglais, nous sommes 9 personnes. Il faut avoir un ordinateur portable assez puissant en prenant soin d'installer son IDE favori, un JDK, un éditeur de texte qui tienne la route. Vous pouvez ajouter éventuellement Ant, Cygwin et la javadoc pour être à l'aise. **Première journée:** La formation commence par une présentation de JBoss Inc., la société. Ensuite une "Architecture Overview" et enfin dans l'après-midi, un lab (un TP) sur le MicroKernel de JBoss. Ce lab consistera à écrire des MBeans et à les tester avec un adaptateur RMI. Kabir commence par expliquer l'histoire de JBoss, et sa place aujourd'hui sur le marché. Il aborde ensuite un argumentaire sur les avantages d'un développement Open-Source. Le principe du micro-kernel est expliqué, montrant le parallèle entre J2EE et le micro-kernel de JBoss avec J2SE. Le micro-kernel, qui ne fait que 44kb,

permet d'écrire des services. Ces services interagissent entre eux pour effectuer des fonctions métiers. Les invocations entre services ne sont pas directes. Tout est négocié par un médiateur (Pattern GOF) qui assure la résolution de nom. Ceci permet de redémarrer un des services sans devoir arrêter tout le serveur d'application. Tous les services de JBoss sont des JMX MBeans. JBoss utilise JMX en interne pour l'invocation mais aussi pour le monitoring et l'instrumentation. Pour instancier des services, JBoss dispose de **deployers**. Il s'agit de class capable de reconnaître un pattern à un endroit spécifié (\*.sar dans le répertoire deploy/nic) et de charger des class et des services. Java 1.4 sait charger des class grâce au ClassLoader. Cependant il n'existe pas de fonction pour décharger une class. Il faut donc utiliser des ClassLoader pour faire la gestion du byte code. Pour cela, JBoss a un système complexe et assez génial (je trouve) de **Repository**. Dur d'expliquer ici comment cela fonctionne, mais c'est vraiment intéressant. Pour ajouter des fonctions génériques aux services comme la **Security**, **Transaction** ou encore un **Cache** JBoss dispose d'un système de **"Server Interceptor and Invokers"**. Ceci permettent d'ajouter à une pile d'appel de manière transparente des fonctions. Les **invokers** sont les agents qui effectuent les appels de méthodes d'autres services. Si vous vous connectez via un navigateur, un **Invoker** adaptera votre appel (HTTP) en un appel JMX en interne. Il est possible d'utiliser pas mal d'invoker (RMI,IIOP,HTTP et HTTPS) par défaut. Les **Invokers** eux-même sont des MBeans (des services). Ensuite nous avons parlé des **Client proxies**. Ce système permet depuis Java 1.3 grâce au système de l'**InvocationHandler** de recevoir côté client un proxy. Ainsi au lieu de devoir envoyer et recevoir les appels de méthode avec JMX, vous pouvez télécharger un proxy local. En clair c'est une interface home (comme J2EE) et qui permet de valider le code et les méthodes à la compilation. Après un bon repas, nous avons travaillé sur JMX (Java Managment Extension). JMX est une API qui permet d'administrer des composants à distance. On appelle ces composants des **MBean's** et JBoss utilise les MBeans comme implémentation de base de tout les services. D'autre part (JSR 160) depuis JMX v1.2, l'API définie aussi la partie de distribution entre le composant managé et les clients. Les protocoles supportés sont HTTP, RMI, SOAP, SNMP, JMS et d'autres... JMX découple l'implémentation des services via un bus d'invocation. Il est donc possible de recharger un des composants MBeans de JBoss sans devoir l'arrêter. Côté client l'api est générique et il n'y a pas de prototypage fort. L'inconvénient est que les clients sont testés à l'exécution et il n'y a pas de validation à la compilation. L'avantage est que l'application managé peut évoluer en interne, être redémarrer sans que les clients ne soient impactés (ce qui est différent de RMI). Il existe cependant un mécanisme propre à JBoss qui permet de recevoir un proxy local, avec cette fois-ci un typage à la compilation. Le coeur de JBoss est le **MBean Server**. Les MBeans sont enregistrés dans ce serveur avec des **ObjectName** qui permettent aux différents MBeans (les services) de discuter entre eux. Ensuite nous avons eu une présentation des 3 types de MBeans. Les MBeans standard implémentent une interface dont le nom se termine par MBean. Les attributs et les opérations sont définis dans cette interface. Le MBean Server retrouve par introspection les méthodes disponibles sous la forme de service. L'avantage est qu'il est très simple d'écrire un MBean, l'inconvénient est que cette interface est figée, et qu'elle n'est pas très documenté car elle est générée à la volée. Pour écrire des composants plus souples, il est possible d'utiliser des **Dynamic MBeans**. L'interface de managment peut alors être déterminer à l'exécution. L'inconvénient est qu'il faut écrire un MBeanInfo assez complexe. JBoss propose une 3ème solution: **XMBean**. Cela permet de définir dans un fichier XML de JBoss pour une class donnée les attributs et les méthodes. Grâce à ce système il est possible de rendre "JMX-compliant" une class existante sans réécrire du code. D'autre part, il devient possible en implémentant des interfaces de JBoss de rajouter des fonctions telles que la persistance ou des event listeners. Ensuite on nous a présenté le **MicroKernel** de JBoss. Les services de JBoss sont packagés dans des fichier **.SAR** pour Service Application Archive. En effaçant un fichier .sar du répertoire deploy de JBoss, le service est retiré. Evidemment ce qui est plus intéressant c'est de rajouter des services à la volée sans redémarrer JBoss. Ce que j'ai retenu et qui diffère de notre **ComponentManager** c'est que JBoss ne force pas un service à définir des méthodes pour la gestion du cycle de vie. Dans le cas où un MBean désire être notifié lorsqu'il démarre (pour créer des Threads) il est cependant possible d'implémenter l'interface Service de JBoss (create, start, stop et destroy). D'autre part un MBean est démarré que lorsque les MBeans dont il a besoin sont aussi démarrés. A noter que si vous écrivez un MBean qui a besoin d'autres services de JBoss (JNDI, Transactions, Logging, Invokers...) alors il faut implémenter cette interface ou surcharger ServiceMBeanSupport. La séquence de JBoss a ensuite été détaillée. JBoss utilise son propre système pour instancier et initialiser les services de base. Ce que j'ai surtout retenu c'est le **Dependency Injection** qui permet à un MBean TOTO qui utilise un MBean DBTiti, de recevoir l'instance de DBTiti lorsque celle-ci aura été instanciée. Il faut écrire un setDBTiti dans la class TOTO, et JBoss se charge de la gestion de l'instance. Oui cela remplacerait notre système de **References** que nous avons mis en place dans le ComponentManager. Mais là où c'est plus complexe, c'est que la class locale n'est pas une instance de DBTiti ni une class qui implémente une interface. C'est un dynamic proxy, soit une class construite à la volée côté client (dans TOTO) et qui se fait passer pour un DBTiti. Oui c'est du charabia mais ça vaut le coup de chercher sur Google pour comprendre. Enfin le système de ClassLoading du Kernel a été présenté. JBoss a un système de Repository interne qui permet de gérer les ClassLoaders et les class chargées. Après tout cela nous avons bossé encore 2 bonnes heures sur un TP pour nous apprendre à écrire un MBean simple et un XMLMBean, beaucoup plus sympa à écrire. **Fin de la 1ère journée**

## Lire la suite

- [Journée 1](#) (Cette page)
- [Journée 2](#)
- [Journée 3](#)
- [Journée 4](#)

\*: Indicatif Futur Antérieur, vive le français... :-D

## Jour 2 Formation JBoss for Advanced J2EE developers in Berlin

## Journée 2

Deuxième journée de formation du training "JBoss for Advanced J2EE developers". Tout d'abord je tiens à présenter les 3 personnes de JBoss qui nous ont fait cette formation (et compléter ce que je disais hier). L'orateur principal est **Kabir Khan**, venu de Norvège. Après un temps passé en Angleterre et en Espagne, il a rejoint JBoss Europe en septembre 2004. Il est l'un des développeurs de JBoss AOP avec Bill Burke. Pour l'assister, **Thomas Diesler** qui vient d'Allemagne, était aussi présent. Il a solide background sur J2EE, les EJB, l'architecture CMP 2.0. Il travaille maintenant sur JBoss Web Services, c'est le monsieur JBoss-WS. C'est un gars très sympa et qui explique simplement des choses complexes. Enfin **Mladen Turk** était aussi présent. Venu de Croatie, Mladen nous a présenté un article qu'il écrit actuellement sur l'intégration entre Apache, Tomcat et mod\_jk. C'est un des contributeurs d'Apache, il est l'auteur de mod\_jk le module qui permet de brancher Tomcat avec Apache. Au sein de JBoss il travaille sur JBossWeb et l'intégration native de serveurs Webs comme Apache ou IIS avec Tomcat. La deuxième journée a commencé par le système des proxy dynamiques côté client (Dynamic Proxies et Client Interceptors). La génération de proxy côté client repose sur la class `java.lang.reflect.Proxy` introduite à partir de la version 1.3 du JDK. Cela permet au client au niveau du code d'utiliser un objet comme si celui-ci était présent dans la même JVM, et de transférer à la volée les appels locaux vers une JVM distante. D'autre part, cela permet à JBoss d'avoir ce qu'ils appellent des **DETYPED CHAIN** que je ne me risquerai pas à traduire. Dans le monde de RMI, vous implémentez une interface locale du côté du client. Votre client va donc appeler la méthode `userTest.getUserName()`. JBoss propose de détyper ces appels pour que l'implémentation change sans que le client soit obligé de réécrire le code. On a alors une méthode `invoke(...)` qui est appelée et qui sera exécutée avec `getUserName` en argument. La deuxième notion qui nous a été présentée est le principe des **Interceptors**. Il s'agit de class qui ajoutent des comportements à la volée sur un appel de méthode. Utilisant le pattern *Chain of Responsibility* ce principe permet côté client d'ajouter à la volée des fonctionnalités telles que le logging, la gestion des transactions, la sécurité, etc. Nous avons vu cela en détail. Enfin ce pattern de Client Proxy est poussé jusqu'au bout au niveau des EJB. Les `EJBHome` et les `EJBObjects` sont des instances de proxy qui implémente la partie EJB du côté client. Par client, il faut comprendre aussi qu'il peut s'agir d'un client distant via RMI mais aussi un autre MBean dans le conteneur JBoss ! Et aussi un `SessionStateful` bean ! Il faut se souvenir que EJB sont des `Remotes Interfaces`. Cela permet d'accélérer le fonctionnement du conteneur de manière transparente. La suite nous a présenté les **Invokers** qui sont des MBeans fonctionnant dans le micro-kernel de JBoss. Les Invokers permettent de fournir une interface de transport pour recevoir des appels distants de client qui ne peuvent pas être connectés directement avec des MBean. Ainsi si vous avez un MBean qui envoie un email, il est facile d'ajouter un Invoker pour recevoir des commandes via JMS, puis d'ajouter un autre Invoker utilisant RMI/JRMP pour un client Java distant, le tout utilisant le même MBean. Et ce, parce que le micro-kernel de JBoss est un `mediator` (Pattern) qui centralise les appels. Cerise sur le gâteau, j'ai compris qu'il était facile d'avoir un Invoker de type **Web Services** pour pouvoir ajouter une interface SOAP à une application J2EE existante. Ensuite nous avons abordé la partie serveur. Comment sont gérés les connexions vers les MBean dans JBoss ? Comment s'effectue l'invocation d'une méthode ? Nous avons eu aussi la présentation des Interceptors mais côté serveur qui permettent de fournir les services suivants: sécurité, transaction, locking, instance et synchronization. Nous avons vu comment écrire nos propres Interceptors si nécessaire. Après avoir parlé du JRMP Proxy, l'invoker principal de JBoss pour les MBeans, nous avons vu comment le serveur d'application J2EE est implémenté avec l'architecture qui nous a été présentée. Un container EJB c'est des Interceptors, un plugin par conteneur et des MBeans sur le serveur. Lorsqu'un fichier `.EAR` est placé dans le répertoire `deploy` de JBoss, le `deployer` EAR charge l'archive, retrouve l'interface Home et l'enregistre sur JNDI. Les **Interceptors** permettent de fournir les fonctions suivantes: logging, security, transaction (géré par le conteneur), le locking d'instance lorsque 2 threads accèdent au même objet, l'acquisition d'instance (ou comment retrouver l'objet XXX qui correspond à l'appel) et donc la gestion du cache. Les EJB entity disposent d'un cache dans JBoss dont la gestion (policy) est déterminée dans la configuration. Il existe 4 types de "Commit" pour le cache:

- Commit option A: le cache est toujours vrai par rapport à la DB. Cas où JBoss est le seul client de la DB
- Commit option B: le cache n'est jamais vrai, il faut tout le temps recharger les données à partir de la DB.
- Commit option C: pareil que B mais en plus le contexte utilisé dans les entity n'est jamais valide (encore + lent)
- Commit option D (JBoss only) pareil que commit option A mais avec un rechargement toutes les n secondes du cache. Cette option permet de décider que les données du cache peuvent être désynchronisées de la DB pendant n secondes par exemple.

En conclusion, nous avons appris lors de cette session le fonctionnement de JBoss et du conteneur EJB. Nous avons aussi vu comment écrire nos propres Interceptors. Changement de sujet l'après-midi: **Aspect-Oriented Middleware/AOP** par Kabir. Qu'est-ce que AOM/AOP? présentation de l'outil JBoss AOP et exemples de programmation par Aspects. J'avais entendu parler de AOP et j'avais quelques à-priori négatifs. Pour moi, encore une couche de "je mets des tags dans mon fichier Java et ça fait telle fonction" comme le fait XDoclet. Et bien ça n'a rien n'à voir... **Aspect Oriented Programming (AOP)** est une technologie Java qui part du constat suivant: lorsque l'on écrit du code Java, le code est pollué par du code pour effectuer des fonctions métiers comme le logging, la sécurité ou l'authentification. D'autre part ces fonctionnalités ne sont pas particulières au code que l'on écrit. C'est pour cela qu'il est recommandé d'utiliser des API comme Log4J pour tout ce qui est logging, JXTA pour les transactions ou JAAS pour l'authentification et la sécurité. Cependant on reste obligé d'ajouter dans le code des appels vers ces outils et cela rajoute du code inutile. Il n'est pas possible ensuite de retirer ce code sans réécrire une partie. Et ce, parce que l'assemblage de ces fonctions se fait à la compilation. AOP propose une autre approche. Il définit la notion d'**Aspect**. Un aspect est une class Java qui définit des **Advices**. Un advice est un event-handler appelé par le framework AOP lorsqu'un certain événement est rencontré. Les types d'événement sont: les accès à une instance, à une class, à un champ (Field), l'entrée dans une méthode et la sortie, etc. D'autre part, AOP étant ajouté à la compilation, il est possible de voir les accès aux membres private d'une classe. Pour mettre en place cela, AOP définit une autre notion: les **Pointcuts**. C'est un langage de requête qui permet de dire au moteur: *si une méthode du package org.jboss est*

appelée alors appelle tel advice ou "Si le constructeur est appelé avec en argument une instance de la class XXX alors execute tel advice". La syntaxe du langage Pointcuts est assez souple pour permettre des opérations booléennes et utiliser des jokers. Comment AOP s'intègre dans du code ? Si le code existe déjà, les Pointcuts qui sont définis dans un fichier XML sont compilés avec un outil livré avec AOP. Sinon il est aussi possible d'utiliser des mots clés dans le code Java, comme Java 5.0 le permet. Cela ressemble à XDoclet à la différence que ces marqueurs ne sont pas mis dans la javadoc mais directement dans le code. L'inconvénient est qu'il faut alors utiliser Java 5.0 et que tous les éditeurs java ne le supportent pas encore (IntelliJ IDEA le fait très bien cependant). Nous avons demandé quel est le coût en terme de vitesse d'exécution d'AOP ? Cela dépend de la quantité d'appels délégués à AOP. Mais il est assez minime puisque le code est compilé et le framework travaille sur la JVM directement. Il intercepte les appels en examinant la pile d'exécution et le bytecode. Cela peut paraître assez étrange mais c'est le moyen trouvé pour permettre d'ajouter des appels AOP sur du code existant. Quels sont les use-cases d'AOP ? Pour effectuer des mesures de performances par exemple. Il est possible ainsi d'ajouter des Metrics le temps de faire des benchmarks. Puis une fois les tests terminés, de retirer cela sans devoir modifier l'application... Je crois que les tests de performances sur P.D.S. (notre produit) seront fait avec AOP et très bientôt... Nous avons terminé cette deuxième journée par un labs sur AOP. **Fin de la journée 2**

## Lire la suite

- Revenir à [Journée 1](#)
- [Journée 2](#) (Cette page)
- [Journée 3](#)
- [Journée 4](#)

# Jour 4 Formation JBoss for Advanced J2EE developers in Berlin

## Journée 4

**Vendredi 28 janvier 2005** j'ai un peu de mal à me lever. La veille nous sommes allés au restaurant avec Thomas, Kabir, Mladen et un allemand. Soirée très sympa et qui m'a permis d'en savoir un peu plus sur JBoss Europe, comment les équipes travaillent, qui fait quoi, etc. Les gars sont très sympas et nous avons passés un bon moment.

Nous commençons la matinée sous la tutelle de Thomas Diesler. Il commence par **Advanced EJB** destiné à présenter les techniques d'optimisation pour les applications J2EE. Tout d'abord nous avons vu où se situent en général les points de ralentissement dans une application J2EE et ce, parce que les spécifications obligent à suivre certaines règles. La **Serialization** est l'ennemi numéro 1 du développeur J2EE. Sans que nous nous en rendions compte, au sein même d'un conteneur J2EE il est très facile de ralentir son application sans le savoir. Pour éviter les problèmes il ne faut pas simplement apprendre à écrire des EJB. Il faut aussi investir du temps et apprendre des Patterns tels que Session Facade, Value Object, Composite Entity ou Value Object ASsembler. Ainsi accéder un Entity Bean directement à partir de la couche client est une très mauvaise idée. Cela entraîne des appels à `ejbLoad` et `ejbStore` qui sont très lents ! Il vaut mieux utiliser un `SessionStatefulBean` pour accéder à un `EntityBean` et surtout utiliser les transactions pour effectuer plusieurs appels sur un entity bean. Un conteneur comme JBoss peut alors utiliser les transactions pour regrouper les appels en un seul appel. Cela accélère grandement l'application.

Nous avons ensuite vu aussi que les clusters n'accélèrent pas forcément une application. Si les données doivent être répliquées entre les différentes nodes du cluster, il n'y a pas de gain de vitesse. C'est un gain de sécurité et de redondance. Beaucoup de personnes pensent qu'un cluster accélère le traitement, ce n'est pas le cas.

Les **Transactions** permettent de s'assurer de l'intégrité d'une opération. Cependant il faut prendre le temps de bien examiner son architecture et faire attention à la manière où le système de transaction est implémenté. Concernant l'accès aux données et la réplication: il existe plusieurs moyens de s'assurer que des données sont répliquées sans impacter les performances. JBoss dispose d'un `MBean` pour invalider un cache. Avec Oracle il est possible d'exécuter du code Java qui invaliderait via JMX le cache du conteneur CMP de JBoss. Il est aussi possible dans un objet comme un `MDB` ou une `SessionStatefulBean` d'invalider le cache manuellement en utilisant JMS. La node principal publie une `Topic JMS` et les autres nodes rechargent alors leur contenu en retournant sur la base de données. Dans notre application nous avons un système similaire utilisant TIBCO Rendezvous (LoadManager + PDSA) mais c'est assez compliqué et une partie du travail est fait par la base de données...

Pour les **Entity Beans** JBoss permet au sein de la config de marquer certains attributs comme étant "read-only" ce qui accélère et optimise les appels. Un champ qui ne change jamais n'a pas besoin d'être rechargé de la base de données. Pour améliorer les performances, l'idéal est d'utiliser l'option A du cache de JBoss. Le cache est toujours vrai et la base de données n'est pas accédée. Ensuite, nous pouvons mettre en place un cache qui s'invalide toutes les n secondes (option D). Dans le cas d'une page Web qui affiche des news, un retard maximum de 5 mn entre ce que l'utilisateur voit et la base de données est parfois acceptable (stock de produits, fiche description d'un livre...). Il faut prendre en compte la granularité et la durée de vie d'une donnée dans le cache.

Dans le cas où l'on modifie une donnée, et que le cache type A ou D est utilisé, il est possible dans les SQL Statements d'effectuer quelque chose comme dans cet exemple:

```
UPDATE TOTO SET A="Nic" WHERE PKey="C" AND A="OldValue"
```

Ainsi si jamais la valeur a été changée par quelqu'un d'autre, ce statement va signaler qu'aucune ligne n'a été modifiée. Il faut alors effectuer côté client un appel pour recharger la valeur et éventuellement avertir l'utilisateur. Ce système permet alors d'optimiser le cache en lecture et accélère grandement les EntityBeans.

**Mladen Turk** a fait ensuite une présentation sur le connecteur mod\_jk qui permet de brancher Apache avec Tomcat. Il a expliqué tout d'abord comment établir une architecture réseau qui fonctionne en production avec un serveur web pour le contenu statique et Tomcat uniquement pour la partie dynamique. Tomcat est le conteneur de Servlets et de JSP qui assure la partie Vue dans le patron modèle-vue-controller. Sa présentation part de son expérience en tant que consultant où il est appelé souvent pour optimiser les performances d'un site internet. Comme il le dit lui-même, les clients espèrent souvent que leur serveur sera capable de gérer des milliers de connexion à la seconde, et cependant ne prennent pas en compte tous les paramètres. Le réseau lorsque l'on branche Apache et Tomcat ou Microsoft IIS et Tomcat est parfois négligé. Ainsi comme il l'explique, une carte réseau 100Mbits qui fait transiter des données codées sur 8 bits ne peut donc en théorie faire passer que 12.5 Mcoctets Mbps. Ensuite si vous prenez en compte le poids moyen de votre page internet, disons 20ko, vous ne pouvez faire passer que 625 clients différents à la seconde par ce même tuyau... L'article en ligne de Mladen est disponible [sur le site Apache.org](#).

Après un bon repas, pour terminer la formation nous avons parlé de sécurité et de JMS.

**JBoss Security Model** avec **JAAS**. Ce que j'ai retenu: le modèle de sécurité de JBoss est surtout destiné à authentifier les utilisateurs et à ajouter un rôle à un utilisateur dans le conteneur. La partie authentification est externalisée grâce à JAAS. Il est possible d'utiliser LDAP, une base de données, des fichiers plats, Kerberos ou des certificats SSL. Cependant cette partie est effectuée par l'api JAAS et n'est pas intéressante pour JBoss. Ce que JBoss ajoute par rapport à un autre conteneur c'est la gestion de rôle. Elle permet en interne, quelque soit le composant, de retrouver qui est l'utilisateur actif pour pouvoir effectuer des traitements personnalisés. Evidemment elle permet aussi de mettre en place de la sécurité au niveau de chaque MBean et de chaque composant d'une application J2EE. La sécurité est configurée dans les fichiers de JBoss. Les Server Interceptors permettent de gérer la sécurité par composant sans surcharger le code.

J'ai surtout retenu la facilité de mise en place et le fait que le développeur peut écrire son application et s'en soucier à la fin, une fois son système mis en production.

Enfin nous avons terminé par **JBossMQ** et **JMS**. Après une présentation rapide de JMS, la différence entre une Queue et un Topic, nous avons eu une présentation détaillée de l'architecture de JBossMQ. L'avantage numéro 1 de JMS est l'échange de message asynchrone entre 2 applications. D'autre part, comme il s'agit d'échanger des données, l'implémentation ou le traitement de ces données peut changer côté client ou côté serveur sans que cela n'impacte l'un ou l'autre. Il faut s'accorder sur le contenu et le contenant mais le traitement est particulier à chaque client. C'est le paradigme **Document-based** opposé au modèle **Remote Procedure Call (RPC)** inventé par SUN.

Nous avons vu l'architecture interne du serveur et les dispositifs pour tout ce qui a trait au clustering et fail-over. Grâce à JBoss et aux transactions, il est facile de s'assurer qu'un message a été délivré. Du côté implémentation, si les messages sont envoyés sur des Queues ou des Topics **persistents** alors JBoss est capable de sauver sur le disque des messages lorsque la Queue est pleine. Dans le cas où des messages importants sont marqués avec livraison garantie, les messages sont enregistrés sur un système de persistance avant d'être envoyé. Enfin JBoss dispose d'un système appelé DeadLetterQueue pour enregistrer les messages qui n'ont pas été délivrés correctement. Encore une fois, pour notre logiciel, cela correspond au Retransmission Agent et il y aurait certainement quelque chose à faire avec JMS... Mais c'est une autre histoire ;-) )

## Conclusion

Pour conclure sur cette formation: elle est effectuée par des développeurs de JBoss. Elle est très technique et je pense qu'elle conviendra à des développeurs, des consultants J2EE ou des architectes. Une bonne connaissance de J2EE permet de comprendre la partie sur le conteneur CMP 2.0. A l'issue de la formation je pense avoir un bon aperçu de l'architecture de JBoss AS. Il ne s'agit pas uniquement d'un serveur J2EE. C'est aussi un micro-kernel puissant basé sur JMX qui permet aux développeurs d'écrire du code qui se concentre sur la fonction **métier** sans perdre du temps à réécrire tout un assemblage de code.

La qualité de la formation est bonne. L'organisation et les moyens sont très bons. Les formateurs étant des développeurs, ils sont vraiment au fait et capable de répondre aux questions posées.

Alors y-a-t-il des points négatifs ?

... je n'ai même pas eu le temps de visiter Berlin !

## Fin de la journée 4

### Liste des articles apparentés si vous n'avez pas lu les anciens billets

- Revenir à [Journée 1](#)
- [Journée 2](#)
- [Journée 3](#)
- [Journée 4](#) (Cette page)

# Jour 3 Formation JBoss for Advanced J2EE developers in Berlin

## Journée 3

La journée commence par la présentation de **JBossCache**. JBossCache est un système virtuel en mémoire destiné à stocker soit des objets simples soit des objets complexes comme des class. Ce qui est amusant c'est que JBossCache me rappelle le **PropertySpace** inventé par Guillaume Pelletier mais avec une architecture et des fonctions beaucoup plus complexe. JBossCache est disponible soit comme une API à intégrer dans votre code, soit comme un MBean au sein de JBoss.

Il existe 2 implémentations: TreeCache et TreeCacheAOP. TreeCache permet de stocker et répliquer vers un autre cache des valeurs. Chaque objet est référencé par un path (/toto/titi/object) sous la forme d'un arbre en mémoire. Imaginez votre disque c:\ avec la structure de répertoire et de fichiers, vous y êtes. L'autre implémentation est TreeCacheAOP. Cette implémentation permet de gérer la réplication sur des objets Java complets. Grâce à AOP il est possible de savoir que seul le champ "Address" de la class "Customer" a été modifié et donc de ne répliquer que cette modification. C'est tout simple et très performant.

JBossCache utilise JavaGroups pour fonctionner en cluster. Il est possible d'effectuer des réplifications synchrones ou asynchrones. Les Transactions permettent d'assurer la contrainte d'intégrité. JBossCache enfin permet de mettre en place plusieurs polices d'évictions pour nettoyer la mémoire et effacer les objets trop vieux. Enfin il existe un cache qui permet de cacher les accès dans le cas où JBossCache utilise une base de données pour persister son contenu. Bref c'est assez complexe et puissant.

L'implémentation TreeCacheAOP est la plus intéressante. Des interceptors permettent de voir les modifications effectuées sur les champs d'un objet et donc, d'optimiser soit l'accès à la couche de persistance (la base de données) soit la propagation au sein d'un cluster des modifications. Efficace et très rapide car seul ce qui a changé est envoyé aux nodes.

JBossCache permet donc d'avoir en mémoire des données, en les classant comme dans un système de fichier. Grâce à JGroups il est possible de répliquer le contenu du cache vers une autre instance de JBossCache. Il est aussi possible d'enregistrer dans une base de données le contenu pour pouvoir le conserver.

Après JBossCache, nous avons vu **JBoss Clustering and Caching**. Comment assurez les fonctions de load-balancing pour la charge et de fail-over pour la sécurité des données avec JBoss ? Présentation de JBoss HA-JNDI (High Availability), de la notion de farming, de la réplication de session HTTP, de HA-JMS, du fonctionnement du côté client du clustering... Bref c'était très chargé comme présentation.

Une fois encore grâce au système de **Client Proxy** les fonctions de fail-over et clustering sont invisibles pour le client. Il faut simplement utiliser les class JBoss client pour bénéficier de ce support, mais il n'y a aucunes modifications côté client sur le code.

Nous avons terminé la journée par la présentation du conteneur CMP 2.0 et donc les modifications sur le code lorsque l'on écrit des EJB. J'avoue que c'était le moins intéressant. Tout d'abord tout ce qui nous a été expliqué sera remis en cause lorsque EJB 3.0 sortira. Mais c'était intéressant de voir comment écrire un Entity Bean et le gérer avec JBoss. JBoss est capable de mettre en place des relations entre Entity Bean et dispose d'un système qui permet de définir en XML dans la config la structure d'un entity. Le conteneur se chargera de charger les données venant de la base, en suivant des règles d'optimisation que nous pouvons définir.

Nous avons terminé par des labs sur le clustering et sur CMP avec pour exemple non pas le PetStore habituel mais la gestion d'une bande de Gangster rattaché à un Boss et disposant de différentes Capacity... Ca change.

### Fin de la journée 3

### Lire la suite

- Revenir à [Journée 1](#)
- [Journée 2](#)
- [Journée 3](#) (Cette page)
- [Journée 4](#)

## Ant best practices

Cet article sur [OnJava.com](http://OnJava.com) donne 15 bonnes pratiques pour ANT. Cela m'a donné l'idée de donner ici quelques astuces et trucs que nous faisons dans l'équipe sur notre projet.

**Un ou plusieurs buildfile ?** Pour simplifier la maintenance de votre makefile, il faut qu'un seul script situé à la racine de votre projet vous permette de tout faire. Si votre arborescence est très complexe, chaque sous-module aura son propre buildfile pour permettre au(x) développeur(s) de travailler indépendamment. Mais il est important que ces buildfiles soient appelés par des targets situés dans le script principal.

**Comment rendre configurable un script Ant ?** Prenons un exemple: dans une équipe, chaque développeur dispose d'un compte sur une base Oracle. Lors des tests, un script ANT exécute des requêtes SQL. Si vous avez spécifié les propriétés de connexion sous la forme de `property` le problème est que chaque développeur doit modifier ces valeurs et qu'il ne faut pas ensuite sauvegarder ces valeurs dans un script utilisé par tout le monde. Pour résoudre cela, ANT peut importer un fichier de `Properties` lorsqu'il démarre. Et vous pouvez même forcer chaque utilisateur à écrire un fichier personnel en utilisant un bout de code ANT comme celui-ci:

```

<!-- Author: Nicolas Martignole -->
<!-- Checks if a property file exists, then load its content -->
<target name="checkcustomproperty"
  description="Checks that the developer has defined a custom property file">
  <property name="buildproperties.file" value="{basedir}/{user.name}_ant.properties"/>
  <condition property="file.exist">
    <available file="{buildproperties.file}"/>
  </condition>
  <fail unless="file.exist">
    ERROR: {buildproperties.file} wasn't found.
    Please copy buildtemplate_ant.properties to {buildproperties.file}
    then customize project properties and restart the build.
  </fail>
</target>

<target name="test" depends="checkcustomproperty"
  description="Load and display one of the custom property">
  <property file="{buildproperties.file}" description="Loads the specified file"/>
  <echo message="The custom property file should contain this property: {property}"/>
</target>

```

Ce code vérifie qu'un fichier a été défini par le développeur, puis le charge en mémoire. Idéalement vous devez fournir un fichier template qui sera géré sous CVS ou SubVersion dans lequel les propriétés sont définies avec des valeurs d'exemple.

**ANT et les tests:** ANT est très pratique pour les tests unitaires. Vous pouvez facilement tester votre code et ensuite générer un rapport au format HTML. Reportez-vous au manuel de ANT pour voir comment définir une target junit

**ANT et des tests d'intégrations:** les équipes d'assurance qualité (Q.A) peuvent utiliser ANT pour leurs tests. En effet il est possible de démarrer une application, d'effectuer des requêtes SQL à partir de ANT et même d'envoyer des emails. Avec CruiseControl dont j'ai déjà parlé, il est donc possible de mettre en place un système complexe de tests automatiques. A voir si l'investissement est intéressant ou non.

## Gestion et surveillance de processus à distance

Aujourd'hui nous donnons la dernière touche à notre application de surveillance et d'administration. J'ai casi terminé la partie packaging avec InstallAnywhere. Du coup cela m'a aussi donné l'envie de raconter ici en quoi consiste ce boulot qui m'a pris presque 2 mois avec un collègue (yo stéphane ;-)).

Le problème à résoudre: notre logiciel est souvent installé sur un grand nombre de machine en salle de marché. Ainsi un de nos clients gère 80 000 portefeuilles boursiers. Autant dire que notre logiciel doit être en mesure de tenir la montée en charge. Jusqu'à aujourd'hui le souci est que lorsqu'une machine plante ou que le réseau est surchargé, l'équipe d'administration technique doit se connecter sur la machine pour relancer les processus. Nous avons donc développé une application de surveillance et d'administration distante. Elle est composée d'un petit agent qui tourne sur chaque machine et d'une ou plusieurs console d'administration centrale. L'interface utilisateur est une application Web classique avec Java Server Faces pour la partie GUI et une application java pour la partie serveur.

Pour la partie agent, un petit composant Java léger permet de:

- - démarrer un processus Unix ou Windows (directement via java.util.Process)
- - découvrir une instance qui fonctionne déjà (via TIBCO Rendezvous)
- - détecter les processus zombi (n'ayant plus d'activité mais étant présent en mémoire)
- - arrêter un processus (via TIBCO Rendezvous)

Pour la partie serveur, j'ai développé une application Web qui fonctionne avec Tomcat 5.0.28. La partie modèle est commune à ce que l'agent gère. Pour synchroniser la liste des agents, j'utilise l'api JGroups de JBoss. Celle-ci permet de découvrir sur le réseau mes agents, puis ensuite de recevoir la liste des applications avec pour chacune un status (démarré/arrêté/planté...). Ensuite pour la partie GUI, comme j'aime bien faire simple j'ai choisi JSF (Java Server Faces) plutôt que Struts. Pourquoi ? Et bien JSF permet contrairement à Struts de générer soit du HTML assez complexe pour un navigateur, soit du WML pour un PDA ou un BlackBerry. Et hop ! d'une pierre, deux coups. En plus de l'interface classique en DHTML, notre application est donc aussi accessible via un BlackBerry ou un SmartPhone. Cela permet donc à une personne de l'exploitation de suivre en temps réel si le logiciel fonctionne correctement, puis ensuite d'effectuer des actions pour éventuellement démarrer d'autres machines, relancer un processus suspect, etc.

De ce projet de 3 mois, j'en ai profité pour proposer JSF afin d'apprendre un peu quelque chose de neuf. Tout d'abord je dirai qu'une fois l'api maîtrisée, il est très facile de créer une page ou deux et de rajouter des fonctionnalités. Pour la partie où je devais représenter l'état du réseau, j'ai utilisé l'api [MyFaces](#) d'Apache. Il s'agit en fait de l'implémentation open-source de JavaServer(tm) Faces dit "JSF". Si vous regardez [le composant Tree2](#) vous aurez une idée de ce que j'ai fait pour la partie visuelle... Tiens il faudrait que je mette des screenshots, en faisant juste attention au problème de Copyright.

Quoi d'autre ? et bien si vous êtes intéressés et que vous voulez en savoir plus [vous pouvez me contacter](#).

## [Pourquoi le pattern lazy initialization ne marche pas](#)

[JavaPractices](#) est un site web qui propose des idées et des astuces pour Java. Chaque article présente un

point avec quelques lignes de code, une explication claire et concise. Cependant je ne suis pas d'accord avec tout les articles. Ainsi celui qui parle de [Lazy Initialization/Instantiation](#) m'a rappelé un autre article dont je vais parler. Avant cela, voyons déjà **qu'est-ce que le processus de Lazy Initialization**

L'instanciation retardée (baaah) ou **Lazy Initialization** est une petite astuce qui permet d'économiser de la mémoire. Lorsqu'une propriété d'un objet n'est pas systématiquement utilisée, vous pouvez l'initialiser que lorsque celle-ci sera appelée. Comment expliquer cela par le code ? Prenons une class **Monster** avec les attributs **color** et **size**. Nous décidons que la couleur du monstre a dépendre de sa taille. Notre accesseur, `getColor`, retournera une couleur rouge si le monstre fait plus de 50m, sinon le monstre sera bleu. Ok c'est pas très original mais je n'ai pas l'inspiration. Le code que je vous propose sera le suivant:

```
import java.awt.*;

/**
 * Monster is a dangerous animal characterized with a color and a size.
 * The color depends on the size attribute
 *
 * @author Nicolas Martignole
 * @version Jun 1, 2005 1:17:49 PM
 */
public class Monster {
    private int size;
    private Color color;

    /**
     * Creates a Monster.
     * @param size is the preferred monster size
     */
    public Monster(final int size){
        this.size=size;
    }

    public int getSize(){
        return size;
    }

    /**
     * Returns the color attribute.
     * @return a safe copy of the color attribute.
     */
    public Color getColor(){
        if(color==null){
            if(size<50) {
                color=Color.RED;
            }else{
                color=Color.BLUE;
            }
        }
        return color;
    }
}
```

Le "Lazy Initialization" est la technique qui permet ici de dire "calcule et stocke la *couleur* uniquement si la méthode `getColor` est appelée". Sinon, l'attribut `color` ne sera pas initialisé.

**Premier problème: 2 Threads appellent getColor en même temps** et donc la première peut être entrain d'initialiser l'attribut couleur et la deuxième va retourner null, ou refaire l'initialisation... Bref la solution semble être la synchronisation. Le code deviendra alors le suivant:

```
/**
 * Step 2
 * Returns the color attribute.
 * @return a safe copy of the color attribute.
 */
public synchronized Color getColor(){
    if(color==null){
        if(size<50) {
            color=Color.RED;
        }else{
            color=Color.BLUE;
        }
    }
    return color;
}
```

**Quel est alors le problème ?** Imaginez dans un jeu video que cette fonction `getColor` soit executée 500 fois par seconde.... Comme nous avons synchronisé toute la méthode, l'accès pour simplement lire la valeur de cet attribut est alors très très lent. Environ 105 fois plus lent. D'autre part comme l'object est déjà initialisé, la synchronisation ne sert plus à rien.

La solution qui peut résoudre cela serait donc de ne se synchroniser que lors de la création de l'attribut `color`. C'est ce que l'on a appelé [Double-checked locking](#) et qui semblait pendant un temps être une solution pour résoudre ce problème. Pour revenir à notre class `Monstre`, si j'applique tel quel ce système voici le code:

```
/**
 * Step 2
 * Returns the color attribute.
 * @return a safe copy of the color attribute.
 */
```

```

*/
public Color getColor(){
    if(color==null){
        synchronized(this) {
            if(color==null){
                if(size<50) {
                    color=Color.RED;
                }else{
                    color=Color.BLUE;
                }
            }
        }
    }
    return color;
}
}

```

Voici l'explication: soit 2 threads executant la fonction getColor en même temps. La première vérifie que color est null, ce qui est très rapide, puis ensuite tente de prendre un verrou sur l'instance de l'objet. Une autre thread s'exécute en même temps. Elle aussi voit que l'attribut color est null, elle va donc tenter de prendre le verrou. Comme la Thread 1 a déjà celui-ci, la Thread 2 attend. La Thread 1 reverifie que color est toujours null, c'est le deuxième if dans le bloc synchronisé. Ensuite, puisque pour l'instant c'est vrai, elle procède à l'initialisation et donc color ne sera plus null. Enfin elle relache le verrou et retourne color. La Thread 2 qui était en attente est alors réveillée. Elle entre dans le bloc synchronisé. Cet fois-ci, comme color n'est plus null, elle va directement ressortir et donc il n'y aura pas 2 fois la même instantiation.

Jusqu'ici tout va bien pensez-vous... Erreur ! Pendant quelques temps, cette solution du double verrouillage pour pouvoir retarder l'initialisation semblait être la solution... jusqu'à ce que l'on prenne en compte que l'affectation d'un objet en Java n'est pas atomique. D'autre part, si le compilateur optimise le byte code ou sinon ma class **Monster** est executée sur une machine à plusieurs processeurs utilisant de la mémoire partagée, cela ne marchera pas. Ainsi parfois la Thread 2 recommence à s'exécuter alors que l'affectation de color n'est pas encore effectuée par la Thread 1. La question est **comment Java peut se permettre de changer l'ordre d'exécution** ? Dans les spécifications du modèle de mémoire de Java, il est écrit clairement que les compilateurs et les machines virtuelles Java sont autorisées à optimiser l'ordre du bytecode, tant que le résultat final de l'exécution n'est pas faussé. Ce qui veut dire que la fonction getColor retournera toujours une valeur correcte, mais peut-être que l'initialisation de l'objet a été effectué 2 fois. C'est un peu complexe à expliquer ici en quelques mots, et je vous propose de [lire cet article](#) en anglais qui en dit plus et surtout cet article assez connu [the "Double-Checked Locking is Broken" Declaration](#) dont les auteurs sont David Bacon (IBM Research) Joshua Bloch (Javsoft), Jeff Bogda, Cliff Click (Hotspot JVM project), Paul Haahr, Doug Lea, Tom May, Jan-Willem Maessen, John D. Mitchell (jGuru) Kelvin Nilsen, Bill Pugh, Emin Gun Sirer.

Faut-il jeter ce système à la poubelle ? Bien sûr que non. Peut-être que vous n'avez pas plusieurs threads et que dans ce cas, ce modèle est valide. Il faut savoir aussi que dans le cas où vous effectuez des initialisations de type simple comme int ou float, le système de la double synchronisation fonctionne. Il ne marche pas pour les objets complexes, ni pour les types java codés sur 64 bits comme double. Enfin grâce à Java il existe une solution basée sur l'utilisation de **ThreadLocal**. Il faut alors utiliser une JVM 1.4 ou 5 car les versions antérieures sont trop lentes.

## Java Message Service JMS

Aaah les transactions et JMS. Cela fait longtemps que j'avais envie de remettre le nez dans les spécifications de JMS. 5 mn sur Google et je trouve un bon article sur JMS API (Java? Message Service Application Programming Interface) qui présente en quelques pages le principe de JMS. JMS est une API Java destinée à l'échange de messages asynchrones entre application. JMS propose ce que JDBC a réussi à faire pour les bases de données: établir un standard pour l'échange de messages en Java. Cette API repose sur la présence d'un serveur qui répond à la définition JMS et qui joue le tiers de confiance entre l'application qui émet un message et l'application qui consomme ce message. Ceci permet de fournir des services à valeur ajoutée comme la gestion des transactions, la qualité de service et la sécurité.

L'article complet [sur le site de Core Developers Network](#) vaut le coup d'être lu car si vous ne connaissez pas JMS, vous en aurez ensuite un bon aperçu.

Je connaissais déjà le système des Queues et des Topics et je me demandais quel serait l'intérêt des transactions avec JMS. Les Topics et les Queues permettent d'envoyer des messages en étant certain que le client en face l'a lu complètement. C'est le système d'acquiescement. Donc pourquoi des transactions ? Voici quelques éléments de réponses: les transactions JMS permettent à un client d'envoyer plusieurs messages JMS à différents destinataires et d'annuler (rollback) la transaction si l'un des destinataires provoque une exception. Prenons le cas suivant: une application doit envoyer un bon de commande à un entrepôt et doit envoyer un avis de facturation à la comptabilité. Cependant si le bon de commande n'arrive pas à l'entrepôt, nous ne voulons pas que la compta soit activée. Et inversement ! C'est là que JMS propose une solution simple pour résoudre facilement ce genre de situation: lorsque de la création de la session, il suffit de spécifier que celle-ci requiert un contexte transactionnel.

```

// pour une connexion point à point:
ses = connection.createQueueSession(true, Session.AUTO_ACKNOWLEDGE);

// pour une connexion de type Topic (publish-subscribe)
ses = connection.createTopicSession(true, Session.AUTO_ACKNOWLEDGE);

```

Le fait de forcer un contexte transactionnel va évidemment ralentir le traitement du message. Mais à mon avis le coup sur les performances est minime par rapport au service que JMS offre ici. Evidemment nous

pouvons recoder tout ce système mais sommes-nous certain de pouvoir faire mieux ? combien de temps pour le faire ? pour le maintenir ? Pour revenir aux transactions et JMS il faut juste noter quelques détails (traduction en français de l'article de CoreDevelopers):

- les messages envoyés à une destination ne sont pas délivrés à celle-ci tant que la session côté client n'est pas committée. Si la session est annulé, aucun message n'est envoyé aux clients.
- les messages reçus sur une destination ne seront pas acquittés tant que la session n'aura pas été committée.
- N'importe quel travail en cours sera annulé lorsque la session sera fermée. Si votre application plante, n'importe quelle transaction qui n'a pas été committée est alors annulé.

Si nous reprenons notre problème d'entrepôt et de service comptabilité, voici comment Je reprends un exemple de code qui est dans l'article de Core Developers Network pour expliquer le principe des transactions:

(code adapté de l'article sur Core Developer Networks)

```
InitialContext ctx;
QueueConnectionFactory cf;
QueueConnection connection;
QueueSession session;
Queue destination;
QueueSender sender;
TextMessage message1,message2;

ctx = new InitialContext();
cf = (QueueConnectionFactory)ctx.lookup("ConnectionFactory");
destination = (Queue)ctx.lookup("queue/testQueue");

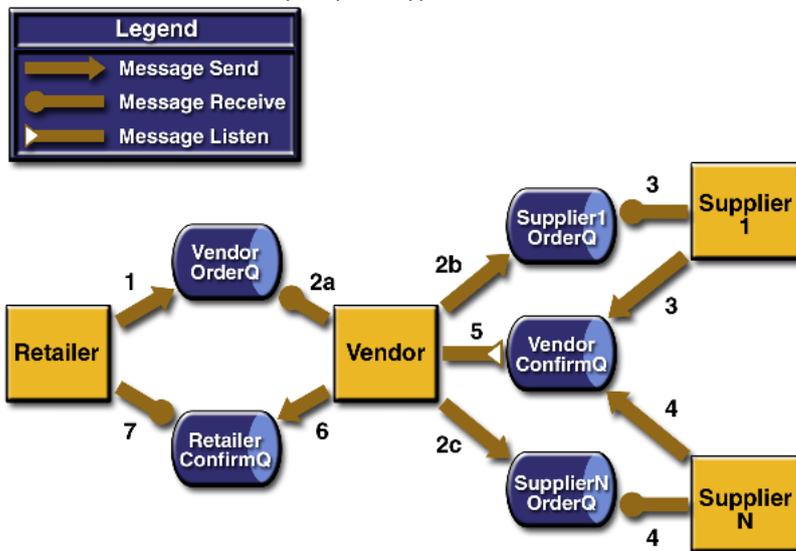
connection = cf.createQueueConnection();
// Creation d'une session avec transaction
session = connection.createQueueSession(true, Session.AUTO_ACKNOWLEDGE);
sender = session.createSender(destination);
message1 = session.createTextMessage();
message1.setText("Test customer order");

message2 = session.createTextMessage();
message2.setText("End of customer order");

System.out.println("Sending Message.");

try{
    sender.send(message1);
    sender.send(message2);
    session.commit();
}catch (Throwable e){
    session.rollback();
}
System.out.println("Done.");
connection.close();
```

Simple, efficace et souple. [Un autre exemple](#) sur le site de SUN permet aussi de voir l'intérêt des transactions. Voici le schéma de principe de l'application:



Bref voilà un système qui permet de se passer de moteur de règles ou tout du moins d'en limiter la complexité. A noter que JBoss avec JBossMQ offre un serveur JMS gratuit et open-source. Pour en savoir plus [c'est par là](#).

## Struts ou JSF ? ou les deux ?

Entre Struts et JSF mon coeur balance... En fait pas du tout. Après avoir mis le nez dans le code, je pense que les 2 ne sont pas antagonistes. Il est d'ailleurs possible d'intégrer Struts et JSF au sein d'une même application. Cependant ces 2 Frameworks se recoupent et il n'est pas évident de choisir l'un ou l'autre lorsque l'on commence à développer une application Web J2EE. Avant de vous donner mon avis, j'ai traduit et adapté un article de Roland Barcia [dispo en anglais ici](#)

### Quelles sont les différences entre JSF et Struts ?

Tout d'abord JSF est jeune. Par rapport à Struts, la première version de JSF est sortie en mai 2004. Struts au contraire est maintenant un framework mature. En cours de développement, il est plus facile de trouver de l'aide pour Struts que pour JSF.

Au niveau du design, je pense que JSF est plus puissant que Struts, surtout pour la partie *View*. Struts se base sur un [FrontController](#) et le pattern *Command*. JSF est basé sur le pattern *Page Controller*. Pour Struts, une servlet récupère la requête HTTP et transforme en ActionForm les paramètres avant de déléguer le tout à une Action class. Cela permet de coder simplement et rapidement des actions. L'inconvénient est qu'une seule Action est exécutée par appel. Le *Model* et les Actions (*Controller*) sont parfois mélangés.

JSF est basé sur le pattern Page Controller. Chaque requête passe par une seule servlet comme Struts à la différence que cette servlet récupère une Page complète avec ses composants (Input type, label, table...) et ensuite va déclencher des événements pour chacun d'eux. En plus du contexte de la page il existe donc un contexte de vue (FacesContext). La partie rendu est ensuite déléguée à une couche de présentation qui se charge de rendre du HTML, du WAP ou autre. Les objets de votre modèle peuvent être câblés avec des composants JSF (label, text, table) directement dans la page JSP en utilisant la syntaxe JSF EL (Expression Language). Lorsque l'utilisateur modifie et retourne la page vers le serveur, JSF peut mettre à jour le modèle de données automatiquement. C'est le principe du Backed Beans. JSF d'un point de vue événement peut gérer le changement d'état de plusieurs composants sur une seule page et donc, par la gestion d'événement inspirée de **Swing**, effectuer plusieurs tâches par requêtes. Côté Struts on ne gère qu'un seul événement par requête. Enfin les ActionForms doivent étendre les classes de Struts, ce qui force votre modèle à dépendre de Struts alors que JSF gère des POJO (Plain Old Java Object) ou Bean sans aucune dépendance. La partie modèle de votre application avec JSF n'a aucune dépendance sur JSF. D'ailleurs dans notre application de monitoring, nous nous contentons d'envoyer des simples POJO pour décrire une machine, une application et un processus.

Côté navigation les deux frameworks sont similaires. Cependant JSF ajoute en plus la possibilité de déclarer sur un élément de la page le nom de la vue suivante à afficher et à déléguer la partie "action" à une méthode. Pour la mise en page et l'intégration, JSF permet d'écrire ses propres composants. [MyFaces](#) propose des composants vraiment puissants comme Tree2 que nous utilisons dans notre application.

**Est-ce que JSF est mieux que Struts ?** Il est bien trop tôt pour y répondre. Tout d'abord historiquement, [Craig McClanahan](#) l'un des auteurs de Struts a participé aux spécifications de JSF. Il est donc clair que l'un n'est pas le concurrent de l'autre. Lisez [ce post de Craig](#) où il donne son point de vue sur JSF et Struts. Sa conclusion est que JSF est plus puissant que Struts sur la partie View, et que Struts dans le futur devra se concentrer sur la partie Model-Controller. Il y a d'ailleurs une librairie [Struts-Faces](#) qui permet de faciliter l'intégration de JSF sur une appli utilisant Struts.

**Le problème:** Struts est un framework 100% MVC, alors que JSF est un framework basé sur la page que voit l'utilisateur. Avec Struts vous devez coder l'action *doLogin* et la câbler pour qu'elle soit déclenchée lorsque l'utilisateur effectue une action précise.

Avec JSF vous pouvez:

- - déclarer que *doLogin* est appelé si l'utilisateur clique sur le bouton *Submit*
- - appeler la fonction *doLogin* parce que le champ Login et le champ Password ont été remplis
- - valider que le champ "Login" n'est pas vide et fait au moins 8 caractères grâce à un Validator
- - afficher en anglais, en japonais ou autre les labels "Login:" et "Password:"
- - ne pas afficher par exemple le bouton *Submit* si l'utilisateur est déjà authentifié
- - etc...

La différence est assez énorme: Struts est basé sur des Actions définies, JSF est basé sur un GUI représenté dans un navigateur sur lequel l'utilisateur a la possibilité de tout faire. Vous voyez pourquoi JSF gère plusieurs événements à la fois ? JSF est donc adapté lorsque ce que vous représentez à l'utilisateur est un GUI complexe composé de plusieurs actionneurs. Sinon pour la petite histoire, JSF est fortement inspiré des WebForms du framework Microsoft ASP.NET.

Voici donc les liens que j'ai le plus visités sur mon Bookmark JSF avec Firefox:

- [JSF Tutorial](#) très complet, avec des tutoriaux.
- [JSF QuickStart](#) est un tutorial qui explique les grands principes de JSF.
- [MyFaces](#) est une implémentation open-source de JavaServer(tm) Faces
- [La page de SUN sur JSF](#) donne accès à des articles, des FAQ et la possibilité de télécharger l'implémentation de référence.
- [JavaServerFaces sur Java.Net](#) est le site de la communauté open-source Java.Net sur JSF. C'est ici que l'on peut suivre le boulot des développeurs de l'implémentation de référence.

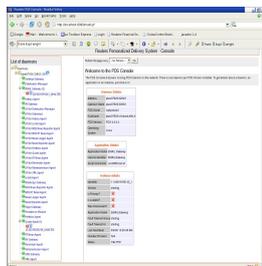
## AJAX lave plus blanc que... OK on l'a déjà fait

Un bon article intitulé "[Ajax: le retour des lessiviers ?](#)" sur le site Framasoft présente et résume mieux que moi ce qu'est AJAX. Grâce à cet article j'ai mis la main sur une librairie JavaScript très sympa: RICO. Le site de [démon](#) vous montre que vous pouvez ajouter des fonctionnalités de glisser-déposer, de mise à jour de page automatiquement et de gestion des layers facilement. Une librairie de plus à tester !

## Console d'admin: les captures d'écran

Voici quelques captures d'écran de l'application d'administration sur laquelle je travaille depuis un petit bout de temps. Le principe de cette application est de permettre à un opérateur système de visualiser dans un navigateur l'état de processus distant de notre application. L'administrateur peut aussi démarrer/arrêter une instance ou exécuter un scénario pour lancer plusieurs applications en même temps. La partie GUI a été codée avec Java Server Faces (JSF) et AJAX. La partie serveur communique avec des agents via JGroups, l'api réseau qu'utilise JBoss et JCache entre autre.

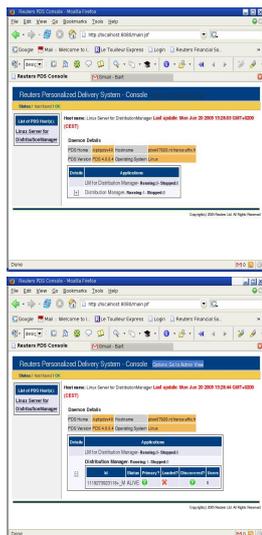
La première version utilise le composant tree2 de MyFaces pour lister les agents trouvés sur le réseau. J'avais découpé l'interface pour que les commandes se trouvent uniquement dans l'arbre à gauche. La vue centrale devait servir à visualiser les détails des éléments sélectionnés. Le souci fut qu'il était alors difficile de montrer en temps réel si une application fonctionnait ou non. Il aurait fallu recharger toute la page et changer par exemple les couleurs du texte de vert à rouge...



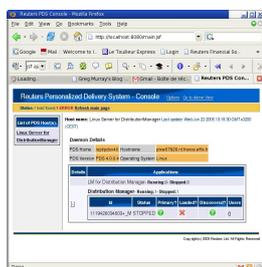
Cliquez sur l'image pour l'afficher en grand.

Il y a 2 semaines, j'ai commencé à refondre l'interface. Le bandeau supérieur permet de naviguer dans l'outil. La partie de gauche liste dynamiquement les machines découvertes sur le réseau. Le bandeau orange est une partie faite avec AJAX sur laquelle je reviendrai plus tard. La partie centrale mélange maintenant de l'information et des commandes.

Voici 2 captures d'écran avec la première version remaniée de l'outil:

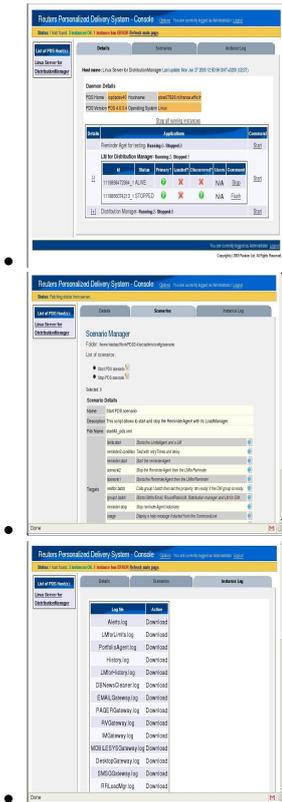


La version actuelle permet maintenant de voir en détail l'état d'une application.



Dans une partie réservée à l'administrateur système, un opérateur peut démarrer et arrêter des instances, ainsi qu'exécuter des scénarios. La mise en place des panels tabs a été très aisée avec JSF. Par contre j'avoue que pour mélanger AJAX et JSF, j'ai un peu triché: dans mes pages JSF je force les id de mes composants, ce qui me permet ensuite lorsque je code la partie AJAX (en Javascript) de retrouver mes lignes de tableau dans la page. Que permet AJAX ? Grâce à une librairie JavaScript ([Prototype 1.2](#)) je récupère le contenu d'une servlet qui me donne l'état de mon réseau au format XML. Si une instance est passée du status ALIVE à UNKNOW, le bandeau orange de status va alors se rafraichir et montrer un message d'alerte pour l'administrateur. D'autre part, il est possible de retrouver un composant dans une

page HTML pour en changer la valeur. Cela me permet de mettre à jour le tableau de status lorsque l'utilisateur a sélectionné l'instance en question. AJAX est très pratique pour mettre à jour une page dans le navigateur sans recharger celle-ci du serveur.



## Les conventions de codage Java: utilité et justification

### Pourquoi utiliser des conventions de codage ?

Tout d'abord le respect de règles de codage facilite la maintenance du code lorsque vous travaillez en équipe. Il faut savoir que le code passe 80% de son existence en phase de maintenance. Lors des phases de relecture ou de corrections de bugs, le respect de convention de nommage et de codage permettent de gagner du temps pour se concentrer uniquement sur le contenu. D'autre part, un problème existe si vous utilisez un gestionnaire de source comme CVS, VSS ou SVN et différents IDE Java. Eclipse ou IDEA d'IntelliJ ont tout les 2 une fonction de mise en page automatique. Lorsque vous ouvrez un fichier, celui-ci est réarrangé selon vos règles de codage. Si chaque développeur a des conventions différentes, il va créer des différences ligne par ligne qui n'en sont pas réellement à chaque fois qu'il va commiter son fichier. C'est ensuite le cauchemar lorsque vous voulez comparer 2 révisions. Impossible de comprendre ce que la personne a réellement ajouté ou modifié. D'où l'importance de fixer des conventions de codage: *la tabulation est de 4 caracteres avec des espaces, les accolades se mettent en bout de ligne, les classes sont nommés avec une majuscule comme première lettre (MyClass et pas maclasse\_qui\_fait\_qqchose) etc.*

### Ce que font les autres:

Si vous avez l'habitude de jeter un oeil sur les projets open-source Java sur Internet, vous verrez qu'à de rares exceptions, chaque classe est codée de la même manière. Les projets open-source forcent les contributeurs à suivre leurs règles de codage. Cela facilite le travail des personnes qui valident les patches, et permet à tout le monde de partager son code. Pourquoi partager ? Laisser un développeur coder à sa manière dans une équipe reviendrait à lui donner une propriété sur le code. Or le principe des méthodes de projet Agile comme l'eXtreme Programming est que le code n'appartient pas à un développeur précis. Il peut se référencer comme étant l'un des auteurs, c'est le rôle de la balise @author de la javadoc. Par contre, rien ne doit empêcher un autre développeur d'ouvrir le fichier et d'y apporter sa contribution. Cela facilite la relecture, les apports d'amélioration et favorise la communication. Pour communiquer efficacement, il faut simplement se mettre d'accord sur des conventions de codage. D'où la question suivante:

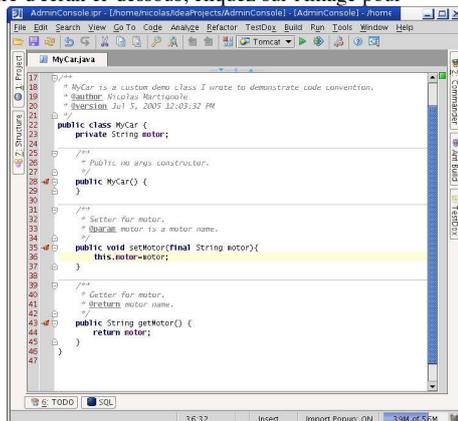
### Quelles conventions de codage utiliser ?

Vous pouvez partir d'un consensus au sein de l'équipe ou tenter d'utiliser les conventions proposées [par SUN](#). Pour notre équipe, la discussion fut vite réglée: nous utilisons tous [IDEA IntelliJ](#) qui par défaut se calque sur les conventions de SUN. Nous n'avons pas personnalisé les réglages. L'avantage d'IntelliJ est de proposer une fonction de [reformatage du code](#) qui permet de parser et arranger un fichier Java selon des règles établies à l'avance ([voir aussi cette page d'aide](#)). Certes il n'est pas gratuit par rapport à Eclipse, mais c'est tout simplement le meilleur IDE Java à ma connaissance. Et je dis cela alors que j'étais un

utilisateur accro à NetBeans auparavant.

### Quelques conventions de codage

Difficile de changer ses habitudes. Ceux qui sont habitués à placer les accolades à la ligne ont du mal à revenir aux conventions de SUN qui imposent de placer les accolades en bout de ligne. Avec un éditeur moderne comme Eclipse ou IntelliJ, les blocs d'instruction sont mis en valeur et il est possible d'ajouter des séparateurs virtuels avec IntelliJ. Voyez la capture d'écran ci-dessous, cliquez sur l'image pour



l'afficher dans une autre fenêtre de votre navigateur:

. Le fait de placer les accolades sur chaque nouvelle ligne ne vous apportera donc pas grand chose de plus, si ce n'est plus de lignes de codes pour rien.

### Should we code in english ?

L'utilisation du français ou de l'anglais. C'est un sujet assez vaste. Faut-il coder en anglais ou non ? Est-ce que l'on préfère *ParseurDynamiqueImplementation* ou *DynamicParserImpl* ? Est-ce que la javadoc doit être rédigée en anglais ou non ? Si vous codez votre projet dans votre coin, rien ne vous force à utiliser l'anglais. Si vous êtes plus à l'aise pour expliquer et commenter votre code en français, alors pourquoi se gêner ? Si vous contribuez à un projet open-source, même si celui-ci est francophone, vous verrez que toute la communauté des développeurs open-source java utilise l'anglais comme langue de travail. J'avoue que cela fait maintenant 8 ans que j'ai pris cette habitude, et que ce soit ma première applet à l'époque ou aujourd'hui une librairie de composant JSF, je n'utilise que l'anglais. C'est comme le vélo: on tombe au début mais une fois que l'on a compris, tout roule. Enfin j'exclu le grand débat des forcenés du français qui pensent qu'il faut utiliser uniquement le français. C'est vrai dans l'absolu et je respecte beaucoup le travail de [l'Office québécois de la langue française](#) par exemple. Pour ce qui est de programmer en Java, la question ne se pose pas. Ou alors il faut faire un autre boulot...

Pour conclure, les règles de codage sont importantes voir indispensables au sein d'une équipe. Vous pouvez vous aider grâce à des outils comme Jalopy ou JxBeauty si vos développeurs ne le font pas eux-même, mais je pense que cela serait dommage. [Cette page](#) vous donne les adresses de Jalopy ou autre si vous voulez les essayer. Sinon Eclipse ou IntelliJ à la place de vi (mon favori) ou emacs pourront améliorer la productivité et la qualité du code. Mais là je glisse sur un autre débat que je garderai pour un autre post...

## Log4J etait capable de recharger un fichier de conf XML automatiquement

Actuellement entrain de coder la partie client-serveur de notre logiciel, je suis plongé dans l'utilisation de la class `SocketHubAppender` de Log4J. Cette class démarre un serveur sur un port TCP donné et attend des connexions de client log4j comme chainsaw. Lorsqu'un client se connecte, le hub fait suivre les messages de logs de l'application vers le client distant. Pour vous représenter le principe, imaginez un serveur web (la partie `SocketHubAppender`). Un client se connecte (`SocketNode` avec log4j) vers ce serveur. Le serveur va alors démarrer une thread qui fera suivre les events Log4J vers ce client. C'est très pratique puisqu'en ajoutant quelques lignes utilisant un `SocketHubAppender` sur n'importe quelle application, vous pouvez alors la monitorer à distance. Le gros problème que j'ai rencontré est que l'implémentation de `SocketHubAppender` dans la version 1.2 de log4j est assez mauvaise. Je parle ici de la version officielle en juillet 05 au moment où j'écris cette note. Il faut savoir que dans votre code, si vous utilisez la class `SocketHubAppender`, celle-ci ne fonctionne que si vous appelez la fonction `activateOption()`... Par quel miracle le développeur qui a écrit cela n'a pas placé cet appel dans le constructeur ? Plus tard, lors de l'appel de la méthode `cleanUp()` nous aurons droit à une `NullPointerException` si la class `SocketHubAppender` n'avait pas été activé. Voula résoudre ces problèmes j'ai téléchargé la version 1.3.alpha-6 de Log4J. La première mauvaise surprise est que la fonction suivante a été retirée: `org.apache.log4j.xml.DOMConfigurator.configureAndWatch(String file)`. Rappelez-vous dans un ancien post j'avais parlé du fait que Log4J peut recharger automatiquement sa configuration lorsque celle-ci est dans un fichier XML. [Voir l'article ici](#). Cette méthode a été retiré du code sans être deprecated. Voir la version 1.68 de la class dans le repository CVS [ici](#) Je me demande ce que l'équipe de Log4J bricole... Bref tout cela pour vous dire de ne pas passer pour l'instant à la version 1.3 qui est loin d'être stable.

## Log4J and ANSI colors in a xterm

The product I am currently developing is a unix server for market datas. Log4J is being use for trace and debug. If you also use Log4J in your Java program, and your program generates traces in a term, this might interest you. I wrote a simple Log4J appender that generates ANSI colors for any Log4J trace. If your application outputs a DEBUG message, a green string message is outputed to the xterm. This appender doesn't work under Windows 2000 and NT4. It might be interesting for any console java application such as JBoss if your xterm has support for colors. Feel free to use it but don't forget to send me credits if you include-it in your apps.

```

/**
 * Copyright (c) 2004-2008 Nicolas Martignole
 * All rights reserved
 * http://www.jroller.com/page/Trecollo/Weblog
 * http://www.touilleur-express.fr
 * Created by : Nicolas Martignole
 * Date: 13 oct. 2004 Time: 15:12:23
 */
package com.reuters.pds.common.sdk.util;

import org.apache.log4j.Layout;
import org.apache.log4j.Level;
import org.apache.log4j.WriterAppender;
import org.apache.log4j.spi.LoggingEvent;
import org.apache.log4j.spi.ThrowableInformation;

/**
 * ANSIColorAppender is a Log4J Appender that output LoggingEvent messages using ANSI colors.
 *
 * Each level (DEBUG,INFO,WARN) has its own color that you can customize.
 * To use this file, in your log4j.properties or log4j.xml specify ANSIColorAppender
 * instead of a ConsoleAppender.
 *
 * @author Nicolas Martignole (nicolas_at_martignole_dot_net)
 * @version 13 oct. 2004
 */
public class ANSIColorAppender extends WriterAppender {
    /**
     * Method from Log4j AppenderSkeleton that gets call for any Log4J events.
     *
     * @param event
     * @see org.apache.log4j.AppenderSkeleton
     */
    public void append(LoggingEvent event) {
        System.out.print(colorizeToANSI(layout.format(event), event.getLevel(), event.getThrowableInformation()));
    }

    /**
     * Requires a layout
     *
     * @return true
     */
    public boolean requiresLayout() {
        return true;
    }

    /**
     * This method overrides the parent {@link WriterAppender#closeWriter}
     * implementation to do nothing because the console stream is not ours to close.
     */
    protected final void closeWriter() {
    }

    /**
     * @param layout
     */
    public void setLayout(Layout layout) {
        super.setLayout(layout);
    }

    /**
     * Colorizes the specified message for the specified log4j level.
     * To get a list of available colors see url http://pueblo.sourceforge.net/doc/manual/ansi\_color\_codes.html.
     */
    private String colorizeToANSI(Object message, Level level, ThrowableInformation ti) {
        StringBuffer oBuffer = new StringBuffer();
        switch (level.toInt()) {
            case Level.ALL_INT:
                oBuffer.append("\u001b[1m\u001b[30m");
                break;
            case Level.FATAL_INT:
                oBuffer.append("\u001b[1m\u001b[31m");
                break;
            case Level.ERROR_INT:
                oBuffer.append("\u001b[31m");
                break;
            case Level.WARN_INT:
                oBuffer.append("\u001b[35m");
                break;
            case Level.INFO_INT:
                oBuffer.append("\u001b[34m");
                break;
            case Level.DEBUG_INT:
                oBuffer.append("\u001b[32m");
                break; // dark green
        }
        oBuffer.append(message);
        oBuffer.append("\u001b[0m");
    }
}

```

```

// Print in red the whole exception stack trace
if (ti != null) {
    String s[] = ti.getThrowableStrRep();
    for (int i = 0; i < s.length; i++) {
        oBuffer.append("\n\u001b[31m");
        oBuffer.append(s[i]);
        oBuffer.append("\u001b[0m");
    }
}
oBuffer.append("\u001b[0m");
return oBuffer.toString();
}
}
}

```

## CruiseControl Monitor: un plugin pour Firefox

Mozilla Firefox est vraiment génial et il le prouve encore. Souvenez-vous de CruiseControl dont j'ai parlé il y a quelques temps. Et bien ce matin j'ai trouvé un plugin pour Firefox qui permet de surveiller CruiseControl à distance via Firefox. Pour cela, le plugin se base sur l'interface JMX de CruiseControl. Etant donné que, pour moi, Firefox fonctionne toute la journée, quel gain de temps ! Le plugin a été écrit par [Dmitri Maximovich](http://www.md.pp.ru) (home page <http://www.md.pp.ru>) et voici [la page du plugin](#)

## Bye bye Mandrake, je passe à Ubuntu

**Lundi matin, grand ménage sur ma machine au bureau. Hop un petit coup de qtparted histoire d'effacer mes partitions (clone de Partition Magic gratuit) et je boote avec un CD fraîchement gravé sur lequel il y a inscrit « Kubuntu 5.04 ». Mais que s'est-il passé ce week-end pour que moi, fan de Mandrake depuis presque 4 ans, j'installe une autre distribution ? Voici comment tout a commencé (fondu au noir, l'image se trouble et on passe à une image en noir et blanc de moi-même avec quelques années en moins) Voix d'un speaker dans le style ORTF, logo de l'INA en bas à droite de votre moniteur...** Le petit nicolas (c'est moi) a commencé à travailler sur Linux en décembre 2000. A l'époque les distributions Linux n'étaient pas encore monnaie courante. Il fit le choix de Linux Mandrake 8.2. Son parti pris fut dès le départ de choisir une distribution orientée facilité d'utilisation. Il ne voulait pas passer son temps à bidouiller (make && make install sans moi). Rappelons qu'à cette époque, au sein de la société DotVision (R.I.P) l'ambiance était plutôt aux « 35 heures / je dors / 35 heures / je dors » que « j'arrive vers 12h30 du matin pour repartir tard le soir vers 16h30 » et que mine de rien, on n'avait pas le temps de bidouiller. Exit la Debian aussi stable à l'époque qu'un rugbymen en 3ème mi-temps... Exit FreeBSD réservé aux gens un peu bizarre... et donc le voilà propulsé gentil utilisateur de Linux Mandrake. Les années passent, il change de société, pour arriver dans une grande société. Et là, c'est le drame... pas de linux. Exit mister-bidouille, vous utiliserez cette machine qui a un écran plat et toc. Quelques temps plus tard, après avoir pris livraison de son nouvel outil de travail, le voilà à nouveau développeur Java sur Windows 2000 Pro... Wow ! Donc nous disions « quelques temps plus tard » soit environ 3h après avoir pris cette machine, un Linux Mandrake en dual-boot fut installé sur la dite-machine. Rassurez-vous, avant de faire cela, il a prit le soin d'en avertir l'équipe système et d'autre part, cela était devenu indispensable car il devait porter vers Linux le logiciel serveur de son équipe. Bref voilà donc comment Linux est devenu ensuite son espace de travail au quotidien... Ok j'arrête le côté mégalo pour revenir aux explications du pourquoi du comment... Donc je disais que j'avais fait le choix de remplacer ma Mandrake Linux 10.1 par une Ubuntu 5.04. Qu'est-ce qu'Ubuntu ? Est-ce que c'est bien ? mieux ? bref voici mon avis sur la question Un tour sur DistroWatch.com vous montrera qu'Ubuntu est aujourd'hui la distribution Linux numéro 1 devant Mandriva (ex Mandrake). Je me suis dit qu'à priori cela valait donc le coup d'aller essayer « pour voir ». Un passage par la [page de Ubuntu](#) plus tard, me voilà avec un LiveCD qui m'a permis de tester sur le PC du bureau cette distro. Ubuntu est une distribution libre et gratuite basée sur Debian. La communauté d'Ubuntu vous certifie que cette distribution restera gratuite. Sur le site d'Ubuntu il est même possible de commander des CD gratuitement. Ubuntu vient d'un mot africain ancien qui signifie « humanité pour les autres ». D'un point de vue technique, elle est basée sur Debian et vous retrouverez donc un environnement très sympa. Le bureau est sous GNOME par défaut mais vous pouvez aussi installer Kubuntu, la soeur d'Ubuntu, packagée avec KDE 3.4.2 au moment où j'écris ce code. **Est-ce que c'est mieux qu'une Mandrake?** Oh là une question piège. Tout d'abord l'installation est plus simple si vous ne connaissez pas Linux. L'utilisation aussi. Sans grande surprise, mon écran plat Samsung SyncMaster 171s n'était pas reconnu. Un petit tour dans /etc/X11/xorg.conf plus tard, quelques Modeline ajouté [comme ce que j'avais fait pour Mandrake](#) et hop, l'écran a fonctionné. Avec Kubuntu j'ai eu quelques soucis avec mon clavier US. Je suis donc passé sous Ubuntu et tout va bien pour l'instant. Côté regret, Mandrake harddrake était bien pratique pour tout ce qui est matériel. Diskdrake aussi était pratique mais avec qtparted, il n'est pas indispensable. Enfin et je pense que j'en reparlerai: j'ai eu à effectuer quelques manipulations avec Ubuntu qu'il n'y a plus besoin de faire avec Mandrake depuis longtemps comme: "activer NumLock lors que GNOME démarre". Ubuntu a encore un peu de retard de ce côté-là. Le gros point positif est la facilité de mise à jour des packages. URPMI (mandrake) peut retourner en couper du bois en sibérie, Ubuntu est plus simple à maintenir. Le système d'Ubuntu (Synaptic) basé sur apt-get est vraiment performant. **Déjà l'heure du bilan ?** Euh non... j'ai eu quelques bugs avec le serveur X et j'attends encore avant de proclamer Ubuntu roi du pétrole. Restons prudent... Depuis 2 jours je travaille donc sur un PC relativement correct (2.4Ghz avec 512Mo de Ram, 40Go HD). J'ai réinstallé sans soucis TIBCO Rendezvous et TIBCO Repository pour le boulot. Pareil pour IDEA IntelliJ (mon éditeur Java). J'ai simplement installé à la main Java et ANT car je n'aime pas les packages de Blackdown. Je préfère configurer Java à ma manière. Côté perfs, j'ai fait quelques tests de compilation de notre projet (4360 fichiers, 360 000 lignes de codes). Avec la même machine sous Windows, la compilation complète prend environ 5mn34. Avec Mandrake Linux, 1mn20. Enfin avec Ubuntu je tourne à 1mn12. Légèrement plus rapide que sur ma station SUN d'intégration... Cool non ? Si vous êtes développeur et que vous ne voulez

pas perdre de temps en bossant, passez à linux. Quel gain de temps ! Et plus de système bloqué pendant de grosses compilations comme avec Windows 2000...Le 9 août 2005 j'ai donc effacé Mandrake Linux pour passer à un système simple, propre et efficace. RIP Mandrake mais bon, je garde ton numéro de portable au cas où...

## KDE 3.4 , ANT et les couleurs ANSI sous Unix

Ce matin j'ai compilé et installé KDE 3.4 grâce à konstrukt, l'utilitaire de KDE destiné à nous faciliter la vie. Le plus simple pour éviter de corrompre son utilisateur est de créer un nouvel utilisateur dédié pour tester. La compilation a commencé hier et s'est terminée dans la matinée. C'est long sur une Mandrake 10.1. Bref voici quelques impressions sur KDE 3.4.

Tout d'abord on note une net amélioration en terme de performances. Je n'utilise pas beaucoup d'application en général (IntelliJ IDEA, amaroK, evolution pour lire mes emails d'Outlook, un Firefox et quelques kconsole à droite à gauche). Konqueror dans la version de KDE 3.2, livré avec Mandrake/Mandriva 10.1 n'est pas franchement exploitable. Pour quelqu'un qui serait habitué à l'Explorateur de Windows, je trouve même que Konqueror est mal fichu. Bref à la poubelle. Si les utilisateurs de Windows arrivent à se passer de terminal (aka cmd.exe) sans problèmes, pourquoi avec Linux faut-il encore passer par un terminal pour certaines opérations ? KDE 3.4 rectifie le tir et la nouvelle version de l'explorateur de fichier est franchement plus pratique.

Après avoir installé IntelliJ et copié mes répertoires de dev, j'ai reconfiguré quelques petites astuces que j'aime bien avoir. L'une d'elle qui n'est pas très connue est pourtant très sympa. **ANT peut afficher en couleur ses messages**. Pour cela, lorsque la commande ant est exécutée, il suffit de spécifier un logger ANSI. Dans mon fichier **.bashrc** voici ce que j'ai ajouté:

```
# Fichier $HOME/.bashrc
# Alias pour Ant
alias ant="ant -logger org.apache.tools.ant.listener.AnsiColorLogger"
```

Ce qui permet de passer de:

```
quote.jar:
[jar] Building jar: /home/nic_kde34/Work/PDS3.4/build/lib/quote.jar

create.manifest:

rtnewsheadlines.jar.classes:
[javac] Compiling 12 source files to /home/nic_kde34/Work/PDS3.4/build/classes
[javac] Note: Some input files use unchecked or unsafe operations.
[javac] Note: Recompile with -Xlint:unchecked for details.

rtnewsheadlines.jar:
[jar] Building jar: /home/nic_kde34/Work/PDS3.4/build/lib/rtnewsheadlines.jar

create.manifest:

full:
[echo] build on 04 August 2005 13:19

BUILD SUCCESSFUL
Total time: 1 minute 51 seconds
[nic_kde34@ptxwXXXX] src>
```

à ceci

```
quote.jar:
[jar] Building jar: /home/nic_kde34/Work/PDS3.4/build/lib/quote.jar

create.manifest:

rtnewsheadlines.jar.classes:
[javac] Compiling 12 source files to /home/nic_kde34/Work/PDS3.4/build/classes
[javac] Note: Some input files use unchecked or unsafe operations.
[javac] Note: Recompile with -Xlint:unchecked for details.

rtnewsheadlines.jar:
[jar] Building jar: /home/nic_kde34/Work/PDS3.4/build/lib/rtnewsheadlines.jar

create.manifest:

full:
[echo] build on 04 August 2005 13:19

BUILD SUCCESSFUL
Total time: 1 minute 51 seconds
[nic_kde34@ptxwXXXX] src> echo $ANT
```

Si vous utilisez ANT dans un terminal xterm qui supporte les couleurs ANSI ou si votre shell Windows a chargé le fichier ANSI.SYS, le logger **AnsiColorLogger** de log4J est capable de vous fournir une sortie en couleur de votre build. Cela ne marche pas avec Windows NT, 2000. Par contre Windows XP semble être compatible.

J'en profite pour vous redonner un lien vers un article que j'ai écrit en octobre dernier. Il s'agit d'un appendix pour Log4J qui utilise les couleurs ANSI lors de l'affichage des traces dans un xterm. Très pratique avec un serveur comme JBoss ou Tomcat par exemple. [Voir l'article Log4J and ANSI colors in a](#)

[xterm](#)

## NTP, CVS et SVN: comment mettre sa machine à l'heure

NTP est un protocole réseau qui permet de mettre à jour l'heure de votre machine en utilisant l'heure donnée par un serveur de référence. Grâce à ce protocole, il est donc possible de mettre l'heure de sa machine automatiquement. Sous Windows 2000 et XP, vous disposez d'une option qui permet de synchroniser l'heure système avec un serveur de temps sur internet. [Voir cet article en anglais](#) et aussi [cet article de Microsoft](#) pour vous aider à configurer le client. Sous Solaris, il faut installer un daemon ntp. Sous Linux Mandrake, idem (urpmi ntp). Sous Linux Ubuntu, ce système est activé pour vous par défaut. Si vous êtes connectés à Internet, Ubuntu se synchronise tout seul comme un grand. Vous pouvez changer l'adresse du serveur ntp en éditant le fichier "/etc/default/ntpdate". **Pourquoi faut-il synchroniser l'heure machine ?** Lorsque vous travaillez en équipe, si vous utilisez un système de gestion de source comme CVS il est indispensable que toutes les machines utilisent la même heure. Si l'une des machines est en avance par rapport aux autres, elle risque de corrompre le repository. CVS se base sur l'heure de la machine du développeur. Pour CVS donc il semble indispensable de synchroniser l'heure [Voir ici un article en anglais](#).

## Linux Ubuntu: quelques sites

J'utilise [Ubuntu](#) maintenant quelques semaines. Voici quelques sites utiles pour configurer et utiliser correctement Ubuntu:

- Ubuntu Newbie Guide donne quelques astuces pour configurer et installer Ubuntu ou Kubuntu. <http://www.mrbass.org/linux/ubuntu/>
- Ubuntu Forums (en anglais) <http://www.ubuntuforums.org/>
- Ubuntu Forums en français <http://forum.ubuntu-fr.org>
- Ubuntu Backports permet de mettre à jour certains logiciels facilement <http://backports.ubuntuforums.org/url.php>

## Les "-ilities" d'un projet: usability, scalability...

J'avais lu un post à propos des "ilities" d'un projet. C'est une petite technique de gestion de projet qui permet de recadrer les attentes entre le client d'une part et l'équipe de développement d'autre part. La méthode consiste à noter de 1 à 10 une liste d'adjectif qui définit les caractéristiques d'un projet informatique. La liste d'origine est en anglais et j'ai traduit en français les différents adjectifs.

Prenez cette liste, demandez à un programmeur de la remplir. Faire de même avec le chef de projet et enfin avec l'équipe business ou avec votre client. Un score de 1 pour "pas du tout important" et 10 pour "critique". L'idéal est de le faire avant le début du projet, en cours et lors de la phase de recette.

- Adaptability (flexibilité) ? une application qui peut être facilement étendue pour ajouter de nouvelles fonctions (extensibilité marche aussi)
- Configurability (configurabilité) - permet sans recoder ou compiler d'élément de changer le comportement d'une application.
- Maintainability (maintenance) ? capacité à administrer l'application. Un doc d'exploitation, des procédures d'urgence améliorent la maintenance et le travail du support.
- Stability (stabilité) - capacité à rester en fonctionnement quelque soit le type d'erreur ou la charge. Cependant en cas d'erreur, l'application doit s'arrêter pour reporter le plus rapidement possible une erreur (fast error fail) sans bloquer de ressources.
- Deployability (déployabilité) le système est capable de déployer de nouvelles fonctions ou des corrections d'erreur rapidement, éventuellement sans s'arrêter. Ce critère est important si les mise-à-jour sont fréquentes. Par exemple le système "Windows Update" facilite le travail de l'administrateur.
- Scalability (capacité à monter en charge) un système est dit scalable lorsqu'il accepte de traiter une plus grande quantité d'information sans intervenir sur le matériel (CPU, mémoire...). C'est aussi valable pour des applications réparties sur plusieurs machines.
- Usability (ergonomie) ce critère caractérise la qualité ergonomique de l'interface. Une interface doit être rapide, robuste et éventuellement internationalisée (i18n). Valable aussi pour la documentation utilisateur.
- Testability (testabilité) capacité qu'a le système à s'auto-tester que ce soit en cours de développement (tests unitaires) ou en production (tests de bon fonctionnement). Se dit aussi d'un système ou d'une application qui donnerait un contexte d'exemple et de démonstration.
- Performability (réponse) autant la scalability définit l'accélération, autant la performability serait la vitesse maximum que l'application ou le système peut atteindre. Parfois la vitesse de traitement n'est pas une priorité (système asynchrone) et parfois le temps réel est un critère indispensable.
- Interopability (interopérabilité) capacité d'un système à s'interfacer avec d'autres applications externes. Ceci rajoute des efforts lors du développement pour supporter par exemple un autre type de plateforme, de base de données, de langage... Ce paramètre aura une influence sur la stabilité et la testabilité. Plus votre application utilise d'autres systèmes, plus celle-ci est complexe à tester et déboguer.

L'idéal est de voir quels sont les 3 critères les plus importants afin que chacun s'accorde dessus. Grâce à cette technique il sera peut-être facile de voir que le client n'est pas intéressé par un système maintenable

(Maintainability) puisqu'il n'envisage pas d'autres versions par la suite... D'autre part cela évitera de se lancer dans des développements complexes pour des critères qui ne seront pas discriminant aux yeux de l'équipe Business.

Sur ce, je retourne à mon code pour terminer la partie "Scalability" avec JGroups... j'ai encore pas mal de travail en perspective.

## GoogleMap Data et Ubuntu



Allez voir [cette page](#) sur le site Ubuntu-fr. Elle localise sur une carte les utilisateurs d'Ubuntu francophone dans le monde. Ce qui est intéressant c'est l'utilisation de la technologie de [GoogleMaps](#). Sur la page <http://www.google.com/apis/maps/> vous pouvez vous enregistrer pour utiliser une carte GoogleMaps sur votre site internet. Avec un peu de Javascript il est facile de créer une carte pour localiser ainsi où se trouve votre entreprise et où vous habitez.

Voici quelques exemples trouvés via [GoogleMapsMania](#):

- [Heritage Walk](#) montre un tour virtuel de la ville avec photos et descriptions des lieux remarquables.
- [JamBase](#) vous montre sur une carte les concerts programmés ce soir.
- Plus proche de l'actualité et moins amusant: [KatrinaShelter](#) est un site qui permet à des volontaires aux USA de proposer un hébergement pour les victimes de Katrina
- [Vue satellite de la Nouvelle Orleans](#) après le passage de Katrina

## Comment gérer les ressources JAR de vos projets J2EE

Je suis quasiment certain qu'au moins une de vos applications J2EE utilise des bibliothèques externes comme Log4j, Struts ou Hibernate. Cependant au moment de packager une application J2EE, la tentation est grande de ne pas s'encombrer de ces fichiers Jar. Souvent on préférera les placer une bonne fois pour toute dans \$JBOSSE\_HOME/server/default/lib alors qu'il existe plusieurs moyens de simplifier ce déploiement.

### Mécanisme 1: le répertoire WEB-INF/lib

Si vous placez les bibliothèques externes comme struts.jar dans le répertoire WEB-INF/lib d'une application Web, celle-ci sera en mesure de les charger. Les bibliothèques sont donc incluses dans le fichier **.war** de votre application. Cependant ce mécanisme ne fonctionne pas avec les EJB.

### Mécanisme 2: le référencement via le fichier Manifest

Ce principe est pratique si vous avez un fichier **.ear** contenant un ou plusieurs EJB et un WAR. Voici comment cela fonctionne: placez à la racine du fichier **.ear** tous les JAR à inclure en tant que bibliothèques. Ensuite, dans chacun des fichiers **.jar** ou **.war** de votre application, se trouve un fichier **META-INF/MANIFEST.MF**. Dans ce fichier la balise **Class-Path** permettra alors de référencer un des fichiers jar placés à la racine de votre **.ear**. Si tout cela vous semble complexe, voici un exemple:

```
META-INF/application.xml
struts.jar
mywebapp.war
  META-INF/MANIFEST.MF:
    Manifest-Version: 1.0
    Class-Path: struts.jar
WEB-INF/web.xml
```

Grâce à ce système simple votre application J2EE sera prête à être déployée sans aucune intervention sur le serveur d'application. L'article dont je me suis largement inspiré se trouve sur le site de SUN [Packaging Utility Classes or Library JAR Files in a Portable J2EE Application](#). Il existe d'autres méthodes de déploiement qui cependant me semblent plus complexes que celles-ci.

## De quoi s'envoler...

Histoire de se mettre à jour sur ce qu'il est possible aujourd'hui de faire avec un navigateur internet moderne, je vous invite à aller visiter le site [FlyakiteOSX](#). Après s'être logué, vous vous retrouverez avec le bureau de Mac OS-X dans votre navigateur. Ce site présente le produit FlyakiteOS-X qui permet de "skinner" ou réhabiliter Windows XP afin que celui-ci ressemble à Mac OS-X.

Au passage je fais un peu de publicité pour un ami graphiste qui a du talent et de l'expérience, Jean-Christophe Tabary dont le site [DesignMood.com](#) vous présente quelques unes des icônes qu'il a réalisées pour Mac OS-X. C'est un ancien de Coplanet et de DotVision.

## Je reprendrai de l'Ajax au petit matin

(Encore un titre de post qui ne veut rien dire)

AJAX est sans doute ce qui fait le plus de buzz dans la communauté des développeurs Web en ce moment... (si si je vous le dis). Cependant JavaScript et Debug étant 2 mots antagonistes, il est plus facile d'utiliser des bibliothèques open-sources que de se lancer dans un codage long et fastidieux pour obtenir un résultat qui ne fonctionnera qu'à moitié, et encore avec le vent dans le dos...

Tout cela pour vous parler de [ScriptAculous](#). C'est une bibliothèque JavaScript qui permet d'ajouter des effets visuels tels que glisser-déposer dans un navigateur web. Pour utiliser cette bibliothèque il vous faudra aussi télécharger la bibliothèque [Prototype](#)

## Drools rejoint JBoss

Drools, un moteur de règles open-source dont j'avais parlé dans un ancien poste, a rejoint la communauté JBoss le 11 octobre dernier. Ce qu'il faut retenir c'est que Drools offrira l'année prochaine à JBoss Enterprise Middleware System (JEMS) un moteur de règles capable de concurrencer JRules. A terme d'ici un an je pense qu'il sera possible d'écrire des applications d'entreprise complète mélangeant une architecture riche (JMS, JCA) et des règles métiers simples (Drools). En terme de coût de maintenance et de puissance, les moteurs de règles sont un énorme bénéfice pour les applications business complexes comme dans le monde de la finance par exemple.

## Aspect Oriented Programming (AOP): mais c'est quoi au fait ?

J'ai entendu parler de la programmation par Aspect pour la première fois en janvier dernier lors de la formation JBoss Advanced Developer à Berlin. L'un des présentateurs était Kabir Khan qui est l'un des contributeurs de [JBossAOP](#) avec Bill Burke. A l'issue de la formation/présentation ([voir l'article ici](#)) j'étais assez bluffé par les possibilités d'AOP. Ce qu'il faut retenir, c'est que les Aspects sont des fonctionnalités de code appelées dynamiquement et qui permettent d'ajouter des comportements à votre code sans impacter son contenu. De manière plus concrète vous pouvez créer un Aspect de "Log" qui sera appelé dès qu'une méthode d'une class ou dès que n'importe quelle méthode de votre application sera exécutée. Imaginez pouvoir ajouter ou retirer à la volée sur votre code des possibilités de logging, de gestion de transactions, de sécurité, de statistiques et d'optimisation de performances... Les Aspects sont les attributs communs d'une application qu'idéalement nous aimerions réutiliser d'application en application.

Le problème est qu'aujourd'hui ce nouveau mouvement dans la communauté Java n'est pas encore mature. C'est pourquoi des développeurs expérimentés se sont lancés dans l'écriture et l'implémentation de différents moteurs d'aspect. Les grands éditeurs comme IBM pour AspectJ ou BEA Systems pour AspectWerkz apportent leur support financier et technique pour aider à l'éclatement de ce nouveau courant. Il faut bien comprendre que ce principe qui en soit existe déjà depuis quelques années, a aujourd'hui un langage mature comme Java pour enfin pouvoir exister. D'ici 2 ans, si tout se passe bien, Java sera donc un langage Objet et programmable par Aspect. Bref je ne veux pas encore passer pour l'évangéliste Java de service, mais m'est avis qu'il faut s'y intéresser.

Aujourd'hui il serait très hasardeux de lancer un projet en production basé sur une implémentation d'AOP. Recemment AspectJ et AspectWerkz, deux des moteurs AOP les plus populaires, ont annoncé leur fusion. C'est bon signe.

Tout ceci au final pour vous parler d'un article sur IBM DeveloperWorks qui compare les 4 principaux moteurs AOP Java existant. A noter que des efforts de portage vers .Net et C++ sont en cours mais la communauté est plus active sur Java pour l'instant car la JVM et le bytecode permettent d'ajouter facilement à l'exécution des comportements. Pour une fois, on n'a trouvé personne pour râler. L'article: <http://www-128.ibm.com/developerworks/java/library/j-aopwork1/index.html>

## Client Riche: XUL, AJAX, Flash ?

J'ai assisté vendredi 25 novembre à une présentation de Valtech Training sur "les Clients Riche, périmètre et solutions techniques". Cette présentation visait à nous présenter les dernières technologies pour porter ou réaliser un client riche. Denis Peyrusaube de Valtech Training a tout d'abord présenté ce qu'est un client riche, pour expliquer ensuite XUL, AJAX, Flash, Java Web Start et enfin Eclipse SWT. Je vais essayer de résumer sa présentation, en donnant au passage mon avis.

### Qu'est-ce qu'un client riche ?

Un client riche est un système qui est hébergé sur un serveur pour faciliter les mises-à-jour et le déploiement, tout en offrant une interface riche à l'utilisateur. Il n'embarque pas toute la logique de l'application. La majorité des traitements sera effectuées côté serveur. Cependant l'ergonomie et la richesse fonctionnelle offertes à l'utilisateur sont très proches d'un client lourd classique.

La conception d'une application en mode page au sens Web du terme tend à disparaître. Aujourd'hui une application doit offrir une ergonomie et une navigation par composant plutôt que par page d'état comme ce fut le cas au début des applications webs.

Idéalement une techno doit proposer des composants graphiques de haut niveau tout en restant facilement déployable. D'autre part, quelque soit la techno retenue, la notion de conteneur sur le poste client apparaît. En règle générale le navigateur internet est suffisant.

### 1. XUL

XUL (prononcer [Zool]) est le précurseur des technologies de présentation web. Il s'agit d'un langage

XML décrivant des composants (Menu, ComboBox) destiné à remplacer le langage HTML. Dans les faits aujourd'hui, XUL est porté par la communauté open-source Mozilla, mais Microsoft s'y intéresse aussi.

Un exemple de site utilisant XUL: <http://www.georgenava.com/> Ce site ne fonctionne que si vous avez Firefox ou Mozilla. Il ne fonctionne pas avec Internet Explorer.

## 2. AJAX

AJAX (Asynchronous Javascript And XML) n'est pas réellement une nouveauté même si en ce moment c'est un sujet chaud dans la communauté Web. Au départ il ne s'agit que d'un article écrit par un consultant destiné à formaliser un mécanisme Asynchrone qui permet de rafraichir une page dans un navigateur sans devoir recharger toute la page. AJAX fonctionne avec Internet Explorer et Firefox car il s'agit ni plus ni moins que du? Javascript côté client. Premier avantage donc: la portabilité.

AJAX du côté serveur n'impose pas de modèle de programmation. Il n'y a donc aucunes normes XML, ni DTD. Le message envoyé par le serveur est au format XML et le client le reçoit de manière indirecte. Il peut alors afficher des informations en utilisant du Javascript, du Dynamic HTML et des feuilles de style.

Donc pour résumer: le serveur envoie un message XML, le navigateur le reçoit via un objet XMLHttpRequest de manière asynchrone. Une librairie Javascript modifie la page, change la couleur d'un combo, fait apparaître un menu déroulant ou masque un tableau par exemple.

Puisqu'une démo est mieux qu'un long discours voici quelques sites utilisant AJAX: [GoogleSuggest](#) Si vous tapez JAV puis vous attendez un instant, votre navigateur envoie au serveur ces 3 lettres. Le serveur retourne une réponse au format XML en utilisant un format propriétaire à Google. La partie Javascript d'AJAX dans le navigateur lit la réponse, construit alors un DIV en HTML (une boîte) et vous liste le nombre de site parlant de java, javascript...

AJAX permet d'ajouter des mécanismes côté client pour ne recharger qu'une partie d'une page internet. Il existe aussi des librairies Javascript destinées à faciliter les développements d'application AJAX comme ScriptAcuLous (<http://script.aculo.us/>) ou surtout ProTOTYPE (<http://prototype.conio.net/>). Elles permettent de coder des fonctions de Drag-and-drop par exemple (<http://wiki.script.aculo.us/scriptaculous/show/DragAndDrop>)

Les inconvénients ?

- Pas de DTD/XSD côté serveur concernant le XML. Chacun fait ce qui lui plait.
- Le Javascript est difficile à coder/débugger et n'est pas un langage très puissant.
- Difficile à tester en phase de développement
- Il existe des composants riches mais ils sont payant
- Pertinence et pérennité du JavaScript dans 2 ou 3 ans ?

Les avantages?

- Simple à mettre en ?uvre
- Puissant
- Effets intéressants offrant de nouvelles possibilités en terme d'ergonomie
- Compatible Internet Explorer et Firefox dans la mesure où le code JS l'est
- Communauté de développeur très active
- Bientôt intégré dans des composants Java Server Faces (JSF) pour faciliter le codage.

Outils AJAX:

- ATLAS (Microsoft) framework intégré dans Visual Studio et .NET destiné à simplifier le développement d'application Web. A noter que Microsoft a annoncé que d'ici 2 ans, Office 13 sera majoritairement en mode serveur lourd-client riche et que c'est l'une des raisons qui font que Microsoft pousse cette techno. Voir l'article d'un des gourou de Microsoft

<http://weblogs.asp.net/scottgu/archive/2005/06/28/416185.aspx> Est-ce un effet d'annonce de Microsoft ou vas-t-on voir un nouveau système pour créer une application riche ?

- SAJAX: Toolkit de dev pour AJAX (<http://www.modernmethod.com/sajax>)

- DoJoToolkit <http://dojotoolkit.org> API Javascript pour résoudre les problèmes de portabilité entre IE et les autres navigateurs.

## 3. Macromédia FLASH

Sans doute pour moi la révélation de cette présentation. Je pensais que Macromedia Flash servait soi à faire un site joli et animé, soit à afficher des bandeaux de publicité lorsque Firefox bloque les pop-ups. Et bien non? nous sommes en 2005 et ce que j'ai vu était vraiment intéressant.

Flash est une techno créée par Macromédia (qui appartient à ADOBE) utilisant un plug-in installé sur un navigateur Web. Conçu à l'origine pour l'animation Web, Flash est un outil pour les graphistes ou les animateurs. Aujourd'hui l'évolution est le fait que Flash peut aussi servir à faire des applications type Client Riche.

Exemple 1: [Le site internet du Club Med](#) utilise Flash pour vous permettre de trouver un séjour sur mesure.

L'évolution est du côté de la conception. La première monture du Studio Flash était destinée à des animateurs Web, avec une ligne de temps et des outils de déformation vectorielle. Ce qui est nouveau c'est la phase de développement et de rendu dynamique.

Il existe 2 solutions basées sur Flash: Macromedia Flex et Laszlo

**Macromedia Flex:** Solution propriétaire et payante de Macromédia. Il s'agit d'un studio qui vous permet de construire votre page en utilisant des composants (Label, Buttons, ComboBox?).

Flex se décompose d'un fichier MXML pour décrire l'interface et d'un fichier ActionScript pour décrire ce qui se passe et les interactions entre les composants. Ces fichiers sont compilés par un conteneur de Servlet (comme JSP) pour générer un flux binaire de type Flash qui sera envoyé au client. Etant donné qu'il ne s'agit que d'une servlet il est donc possible de rendre dynamique une partie du contenu XML par exemple.

Un plugin Eclipse pour FLEX est en cours de réalisation.

**OpenLaszlo** est basé sur le même système que Flex, à la différence prêt qu'il s'agit d'une initiative Open-source. Démo disponible à cette adresse : <http://www.laszlo.com/demos>. La société LaszloSystem propose des services comme le fait Jboss Inc. en terme de support si nécessaire.

Flash est finalement intéressant car les outils de conception de page existent déjà et permettent donc de bien séparer la partie GUI de la partie data. Cependant difficile de se rendre compte si les données dynamiques sont faciles à gérer ou non.

#### 4. Java Web Start

JWS ajoute à Java des fonctionnalités de déploiement, de gestion de cache et d'exécution distante. A ce titre c'est donc une technologie pour Client riche. JWS fonctionne via un navigateur internet ou non. Permet de mettre à jour rapidement les clients répartis, tout en ayant une richesse fonctionnelle complexe.

En gros inconvenient, il faudra recoder un container pour remplacer ce qu'offre le navigateur Web (partie réseau, chargement du contenu, gestion Proxy, CSS?) JWS ne convient que si le client riche doit vraiment avoir des fonctions qu'un navigateur web ne peut pas offrir.

En terme d'avantage, on revient au modèle client-serveur classique, ce qui est beaucoup plus facile à coder/débugger que via le Javascript par exemple.

#### 5. Eclipse Platform

Eclipse est un éditeur Java bien connu mais surtout un moteur de rendu appelé Platform. Ce système est composé d'un ensemble de service, de widgets, de gestion des ressources qui permettent d'utiliser le squelette d'eclipse comme base de développement de client lourd. D'autre part, il est possible d'intégrer d'autres plugins et donc d'enrichir son application facilement. A noter que l'on peut utiliser JWS pour installer une application basée sur Eclipse. <http://www.eclipse.org/platform/index.html>

#### Conclusion

Les clients riches seront fortement présent dans les années à venir car les navigateurs Web deviennent aussi de plus en plus puissant. La facilité de déploiement, l'absence de gestion de mise à jour sur le poste client et l'ergonomie riche sont les critères qui font un client riche. On parle beaucoup d'AJAX et je sais qu'il existe des composants JSF (côté serveur) qui permettent de s'interfacer via J2EE avec les applications métiers. Il sera donc certainement facile d'ici quelques mois de tester et voir si ce genre de solution peut nous intéresser. Flash via Flex ou Laszlo offre le meilleur résultat en terme de rendu côté client. N'ayant pas vu en détail le côté serveur, il est difficile d'estimer si cette technique est adaptée à des données très dynamique comme dans la finance. Java Web Start et Eclipse Platform sont deux solutions pour développer des clients riches sans retomber dans le modèle client-serveur.

## EJB3.0 Evolution ou révolution?

Vendredi 26 novembre 2005, Valtech proposait un séminaire sur les Java Enterprise Bean version 3 (EJB3). La présentation a été effectuée par Denis Peyrusaube de Valtech Training. Après avoir fait un état des lieux, nous avons vu les alternatives existantes aujourd'hui pour combler certaines lacunes des EJB 2.1. Dans la deuxième partie de la présentation nous avons jugés des différences avec différents exemples d'EJB3 utilisant le conteneur J2EE Jboss 4.0.3.

1) **Etat des lieux** Les spécifications EJB 2.1 publiées en novembre 2003 définissent 3 types d'EJB:

- Les EJB Session
- Les EJB Entité
- Les EJB messages (MDB)

Les EJB Session représentent souvent une façade au processus métier d'une application. Lourd à programmer (3 classes nécessaires) et à tester, les EJB sessions sont souvent utilisés en mode local par un client de type Servlet ou JSP. Par experience, il s'est averé que l'interface distribuée (remote interface) est peu utilisée alors que l'aspect distribution devait être l'un des points forts des EJB 2.1. Bien souvent les développeurs utilisent des patterns (Facade, DAO, Service Locator) pour contourner les limitations de l'architecture des EJB 2.1 ou pour simplifier leur application.

Les EJB Entités sont le type d'EJB le plus décrié. Au départ destiné à représenter des données, ils sont lourds à développer et à tester. Ils nécessitent un mapping objet/relationnel soit via le container (CMP Container Managed Persistence) soit via l'application (BMP Bean Managed Persistence). Or le mode BMP, rarement utilisé, laisse trop de travail de programmation sans garanties de performance. Le mode d'exécution distribué (local/remote) est à proscrire compte tenu des gros problèmes de performance engendrés. Aujourd'hui les entity sont remplacés par le pattern Value-Object en utilisant un DAO. Il y a très peu de projet J2EE viable utilisant les Entity distribués sur le réseau car la sérialisation/désérialisation est extrêmement coûteuse. A noter aussi qu'entre une application J2EE dans un EAR et une application Web packagée dans un WAR, la sérialisation est utilisée, ce qui est une catastrophe en terme de perf. Seul JBoss et le system de ProxyPattern est capable d'éviter cette sérialisation trop couteuse aujourd'hui.

Les EJB messages (MDB) sont des ejb asynchrones qui s'exécutent par exemple lorsqu'un message est publié sur une queue ou une topic JMS. Ils permettent d'effectuer des traitements de type batchm envoi d'email pour mot de passe perdus, validation vers un back-office... Rarement utilisé et complexe à tester, ils seront aussi améliorés par la spécification des EJB3.0. Parmi les freins du succès: modèle de programmation contraignant qui ne permet pas d'héritage ni de polymorphisme. Le modèle de persistance proposé par les entity n'est pas très performant. L'indépendance vis-à-vis des serveurs J2EE est relative, il faut configurer souvent des fichiers XML par serveur d'application.

2) **Les alternatives aujourd'hui** Face à ces lacunes, différentes initiatives souvent open-source proposent depuis 2 ans des compléments fonctionnels très puissant qui ont assuré le succès des EJB. Nous avons vu une présentation de JDO, Hibernate, Inversion Of Control, les annotations et la

programmation par aspect (AOP). Je vais rapidement donner quelques détails sur chacune de ces technos mais sinon le web est plein d'articles très intéressants à ce sujet.

**JDO** (Java Data Object) est une spécification proposée par SUN. C'est un mécanisme de persistance basé sur des POJO (class Java représentant un objet très simple sans méthodes métiers, comme un Account ou un User par exemple). Indépendant du lieu de stockage (SGBDO ou SGBDR) et fonctionnant sans conteneur d'exécution. JDO utilise un mécanisme d'enrichisseur de byte-code pour attraper les classes en cours d'exécution et les persister vers un lieu de stockage. JDO2 en cours de rédaction devrait combler les limitations de JDO qui sont la gestion des transactions et la sécurité.

Concurrent direct de JDO, **Hibernate** est un projet open-source intégré comme moteur de persistance dans Jboss. D'ailleurs les développeurs d'Hibernate ont rejoint Jboss Inc. et certains sont salariés par l'entreprise privée Jboss Inc. Très facile à mettre en œuvre dans un serveur J2EE ou dans une application Java classique, Hibernate est un moteur de mapping objet-relationnel. Ce système permet d'avoir en mémoire par exemple une class Person mappée à partir d'une base de données. Hibernate est à la base du principe de persistance des EJB3.

Le pattern **Inversion Of Control** a été mis en place par quelques frameworks comme Spring par exemple. Le principe de l'inversion de control est très simple et repose sur la réflexion en Java. Dans l'exemple que nous avons vu, voici le code que nous avons:

```
public class Product{
    private ProductDAO dao;
    public Product() {
        dao = (ProductDAO)DAOFactory.getProductDAO();
    }
    ..
}
```

La class Product récupère une class d'accès aux données sans en connaître l'implémentation. Imaginons que nous avons codé 2 classes implémentant l'interface ProductDAO, une pour Oracle et une autre pour Sybase. La factory retournerait alors la bonne implémentation selon un paramètre de configuration par exemple. En utilisant l'inversion de contrôle, le code de la class Product sera par exemple:

```
public class Product {
    private ProductDAO dao;

    public Product() {}

    public void setProductDAO(ProductDAO dao) {
        this.dao=dao;
    }
    ...
}
```

Il faut maintenant un moteur spécial (comme Spring) pour injecter à l'exécution la bonne class implémentant ProductDAO. Pour cela, nous allons configurer la class dans un fichier XML pour que Spring place la bonne classe.

```
<beans>
<bean id="OracleProductDAOBean" class="com.reuters.dao.OracleProductDAO"/>

<bean id="TestNic" class="com.reuters.product.Product">
<property name="productDAO">
<ref bean="OracleProductDAOBean"/>
</property>
</bean>
...
</beans>
```

Ce fonctionnement simplifie la programmation et permet aussi de définir des mock-object pour des tests unitaires très facilement.

**Les annotations** sont aussi un nouveau mécanisme inspiré au départ de Xdoclet. L'idée est que lors de la phase de développement, le programmeur spécifie à l'aide de tags spéciaux les fonctions de sa class ou de sa méthode. C'est l'une des grandes nouveautés de Java 5. Les annotations permettent alors de déclencher des actions génériques automatiquement sans trop surcharger le code.

**La programmation par aspect (AOP)** est encore relativement jeune. Il s'agit d'alléger le code en déclenchant des aspects automatiquement lorsque certains critères sont rencontrés. Imaginez que vous ayez un aspect LogInfo qui affiche un message d'information. A l'aide d'un moteur d'aspect vous pouvez par exemple demander d'appliquer cet aspect sur chaque constructeur de votre code. Dès qu'un objet est construit, un message est généré. Facilité de maintenance car il n'y aura pas de log.info("qqq") dans vos constructeurs. Souplesse de programmation car il est possible d'ajouter et retirer ces aspects sans recompiler le code. Centralisation aussi des séquences de traitement répétitives.... On verra avec les EJB3 des cas concrets d'aspect ainsi que leur puissance.

A ce stade nous avons donc vu qu'il existe des outils puissants pour répondre aux lacunes des EJB2.1. Cependant il n'y a pas de normalisation et une phase d'apprentissage puis d'intégration est nécessaire pour faire fonctionner toutes ces technologies Java ensemble.

**3) Les EJB3** Dans la deuxième partie nous verrons comment écrire un EJB3 Session, développer un entité, le fonctionnement de l'Entity Manager, la vue du côté client, les annotations avancées. Suite à la présentation j'ai essayé aussi de regarder où en était BEA, IBM et SUN par rapport à Jboss, qui, pour l'instant, est le seul serveur J2EE prêt pour les EJB3

Où en sommes-nous ?

Tout d'abord avant de continuer sur le résumé de la présentation Valtech, j'ai regardé sur le site Java Community Process l'historique des spécifications EJB3 pour voir où nous en sommes. Ce qui a été décidé: EJB3 sera partie intégrante de J2EE 5.0 actuellement en phase de Final Draft Version. La sortie officielle est donc prévue courant 2006.

2 objectifs pour la nouvelle spéc des EJB3:

- Simplification du développement
- Standardisation de l'API Java Persistence

Tout d'abord concernant la simplification du développement, les EJB3 retirent le besoin d'implémenter une interface spécifique pour les interfaces Home ainsi que les Beans eux-mêmes. Les SessionBeans, les Entity et les MDB sont donc maintenant de simples class Java auxquelles nous allons ajouter des fonctionnalités Business. Les EntityBeans sont maintenant des class java standards avec un design à respecter (Serializable, constructeur public sans args, annoté avec le tag @Entity, etc.)

EJB3 intègre le meilleur des alternatives que nous avons présenté dans la partie 1. Le modèle d'Hibernate a inspiré l'API Java Persistence (EJB3 n'est pas Hibernate). Les annotations permettent d'éviter les fichiers XML compliqués (à la Xdoclet). Le couple IoC/annotation permet au conteneur de binder à l'exécution la valeur d'une class. Ceci permet d'éviter d'avoir à faire des lookup JNDI. Il suffit de dire que vous avez besoin de la DataSource OracleTest pour que celle-ci soit branchée par le conteneur à l'exécution.

### Développer un Session Bean

Un Session Bean est une class Java simple qui peut implémenter une interface SessionBean sans que ceci soit nécessaire. Comment alors le conteneur sait qu'il s'agit d'un sessionBean ? La réponse est: les annotations. Et donc si nous avons des annotations cela signifie que nous utilisons Java 5. EJB3 utilise Java 5 et ne fonctionnera donc pas avec les anciennes versions. Sans se lancer dans le débat pour/contre, il faut voir que Java 5 est vraiment une version à part de Java bien plus riche en terme de fonctionnalités que Java 1.4. Cette migration est donc sans douleur, le code Java écrit en 1.3 et 1.4 étant évidemment compatible avec Java 5. Pour revenir à la class SessionBean, en utilisant une annotation nous aurons donc le code suivant:

```
/**
 * Demo Reuters EJB3.0 Nicolas Martignole
 */
@Stateless
public class CourseManagerBean implements CourseManager {

    public void create(Course c) {
        ...
    }

    public Course findCourseById(String courseId) {
        ...
    }

    public Collection findAll(){
        // return all courses.
    }
}
```

L'annotation Stateless (case-sensitive) signale au conteneur qu'il s'agit d'un Stateless Session Bean.

Voyons maintenant les interfaces

Interface remote:

```
@Remote interface CourseManagerRemote extends CourseManager {
}
```

Interface locale:

```
@Local interface CourseManagerLocal extends CourseManager {
}
```

L'utilisation de ces 2 annotations est optionnelle. Elle permet d'outrepasser le comportement par défaut du "compilateur" en mentionnant de façon explicite le nom des interfaces distante et locale de cet EJB. La version draft des spécifications EJB évoque aussi l'éventualité de n'avoir pas besoin d'écrire soi-même l'interface métier mais de laisser cette tâche au conteneur. Les méthodes publiques et non "injectées" seraient alors considérées comme accessibles via l'interface métier. Imaginez le temps que vous avez perdu dans vos ejb-truc.xml à y placer le nom des méthodes... Ca y est vous voyez ? Le code reprend l'ownership sur la logique. Si une méthode est publique, à priori c'est que nous voulons nous en servir dans notre EJB. Logique non ?

Comment est bindé ce SSB via JNDI ? Comment le retrouver ?

Ici un apport d'EJB3, pas évident à expliquer mais vraiment "tueur", est que les références se basent sur les class Java. Le client retrouvera une instance de ce CourseManagerBean de cette façon:

```
InitialContext ctx=new InitialContext();
CourseManager cm=(CourseManager)ctx.lookup(CourseManager.class.getName());
cm.create(myCourse);
cm.findCourseById("nicolasTest");
```

Simple non? (je commence à passer en mode "évangélisation", désolé) Mais cela sera encore plus simple, car si nous utilisons le pattern IoC il est possible de demander dans son code "donne moi une instance de CourseManager" de la façon suivante:

```
@EJB CourseManager myCm; // Sera bindé par le conteneur à la volée
myCm.create(testCourse);...
myCm.findCourseById("nicolasTest");
```

Voilà pour un petit aperçu de ce qu'il sera possible de faire via les EJB3 concernant les Sessions Bean. Il y a de nombreuses fonctionnalités (sécurité, transaction) ainsi que le support pour retrouver.

### Développer un Entity Bean

La gestion de la persistance est sans doute la plus grande révolution des EJB3. A elle seule elle peut couvrir plusieurs livres. Pour commencer un EJB Entity est un objet POJO (Plain Old Java Object) ou plus clairement, une class qui n'a rien de particulier. Pour définir notre entity Course utilisée ci-dessus voici comment nous devons la déclarer:

```
/**
 * EJB3 Entity Bean that represents a Course.
 * @author Nicolas
 */
@Entity
@Table(name="CourseTable", schema="Sybase_KiwiSample")
public class Course implements Serializable {
    private String description;
    private String id;

    /* Public no-arg constructor (required by spec) */
    public Course()

    @Id public String getId() {
        return id;
    }

    public void setId(String id){
        this.id=id;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String desc){
        this.description=desc;
    }
}
```

3 annotations permettent de transformer cette class simple en un entity bean qui sera persisté automatiquement sur une base de données. @Entity est une annotation marqueur pour le conteneur @Table est le nom de la table en base qui va mapper cet Entity. Si la table n'existe pas ? le conteneur va la créer automatiquement Si je change le type d'un attribut (Description devient un Integer) -> le conteneur se débrouille tout seul.

Comment mapper l'Entity Bean sur plusieurs tables de la base de données ?  
Voici comment procéder:

```
/**
 * EJB3 Entity Bean that represents a Course.
 * @author Nicolas
 */
@Entity
@Table(name="CourseTable", schema="Sybase_KiwiSample")
@SecondaryTable(name="CustId")
@JoinColumn(name="CustId")
public class Course implements Serializable {
    private String description;
    private String id;

    /* Public no-arg constructor (required by spec) */
    public Course()

    @Id public String getId() {
        return id;
    }

    public void setId(String id){
        this.id=id;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String desc){
        this.description=desc;
    }
}
```

Si un attribut de ma class ne porte pas le même nom que ma colonne en base ? Pas de soucis:

```
@Column(name="P_DESC")
public String getDescription() { return description; }
```

Il existe d'autres annotations pour résoudre les problèmes liés au mapping, spécialement si vous partez

d'une base de données existantes.

Qu'est-ce que le tag @Id ?

Directement inspiré d'Hibernate, il s'agit d'une annotation pour marquer le nom du champ qui sera la clé discriminante pour l'objet.

Concernant les Collections, je ne sais pas comment est fait le mapping. Aucune idée. Je sais qu'avec Hibernate c'est faisable via un fichier de configuration particulier.

Quoi d'autre ?

Il y a aussi la gestion de l'héritage et des relations en Java. Si un Entity Course a une relation de many-to-one avec un Entity Teacher, EJB3 est capable de gérer les mappings correctement. C'est vraiment puissant et simple car d'un point de vue développeur, tout est fait en Java directement. Votre class Course aura une instance de Teacher et la class Teacher aura une List de Course. Le mapping objet/relationnel est effectué par le conteneur.

Les performances ?

Difficile pour l'instant à évaluer. Cependant la gestion de la persistance par le conteneur permettra d'optimiser les appels en mémoire par rapport à la base. En terme de testabilité il est clair qu'un simple objet Java sera plus facile à tester qu'un objet héritant des interfaces J2EE.

L'EntityManager

L'API EntityManager a été introduite dans EJB3.0 et permet aux instances des objets Java d'être détaché et chargé localement pour être ensuite renvoyé au serveur et persisté dans la base de données. La "home" des EJB Entités disparaît pour laisser place à l'Entity Manager qui représente si vous voulez une "home" générique. Permet de créer un EJB, de rechercher par clé primaire, d'exécuter une requête EJBQL et vérifier l'existence d'une clé primaire. Pour essayer d'expliquer cela, voici un petit exemple:

```
@Stateless
public class CourseManagerBean implements CourseManager{
    @PersistenceContext private EntityManager mgr;

    // Save the Course to the persistence layer
    public void create(Course c){
        mgr.persist(c);
    }

    public Course findByPrimaryKey(String courseId) {
        return mgr.find(Course.class, courseId);
    }

    public Collection findAll(){
        return mgr.createQuery("select c from Course c").listResults();
    }
}
```

L'annotation @Stateless marque cette class comme étant un Stateless Session Bean. L'annotation @PersistenceContext déclare une instance d'EntityManager. Via le pattern IOC le conteneur injectera à l'exécution une référence vers une instance d'EntityManager.

La méthode create enregistre un nouveau objet Course (création d'une ligne dans la base de données, éventuellement création de la table si celle-ci n'existait pas)

La méthode findByPrimaryKey permet de rechercher un objet de type Course qui matche tel courseId.

La méthode findAll montre comment créer une requête EJBQL. Celle-ci est simplifiable mais je n'ai pas noté. Je crois que c'est "from Course c" pour faire un select \* par défaut.

Nous avons vu ensuite le langage de requête d'EJB ainsi que la vue Client des EJB3. C'est assez détaillé et je ne reprendrai pas ici toute la présentation.

A ce stade vous devez avoir déjà une idée de ce qu'est réellement la spécification des EJB3.

Quels sont les inconvénients visibles pour l'instant ?

- EJB3 se repose sur Java 5, ce qui peut être un facteur bloquant pour certains clients même si ce n'est pas justifié
- Les annotations ne doivent pas être plus nombreuses que le code lui-même
- Il apparaît que des conventions de code pour que le code soit homogène et propre dans une équipe. Sinon les annotations seront placées n'importe où. Pas forcément facile à mettre en place.

**Où en sont les principaux acteurs du marché ?**

**Jboss:** le serveur d'application Jboss est pour l'instant le seul serveur Java prêt pour les EJB3 qui fonctionne parfaitement depuis quelques mois. C'est un énorme avantage compétitif qui risque certainement de mettre en difficulté d'autres projets open-source

**GlassFish,** un projet open-source hébergé sur Java.net est une initiative pour créer le premier serveur compatible JEE5 (on ne dit plus J2EE mais JEE...) Home-page: <https://glassfish.dev.java.net/>

Abstract:

The GlassFish community is building a free, open source application server which implements the newest features in the Java EE 5 platform (the next version of the J2EE platform). The Java EE 5 platform includes the latest versions of technologies such as Enterprise JavaBeans (EJB) 3.0, JavaServer Faces (JSF) 1.2, Servlet 2.5, JavaServer Pages (JSP) 2.1, Java API for Web Services (JAX-WS) 2.0, Java Architecture for XML Binding (JAXB) 2.0, Java Persistence 1.0, Common Annotations 1.0, Streaming API for XML (StAX) 1.0, and many other new technologies.

**IBM** a annoncé le rachat le 10 mai 2005 dernier de GlueCode, l'équipe des développeurs à l'origine de Geronimo (serveur J2EE open-source). Une analyse Garnet ([http://www.gartner.com/DisplayDocument?doc\\_cd=127754](http://www.gartner.com/DisplayDocument?doc_cd=127754)) explique qu'IBM veut freiner la montée de Jboss et tester aussi ce nouveau business model "subscription-based software pricing" où le client vient acheter du support, de l'expertise technique et de l'aide. Parmi les risques identifiés, Geronimo n'est pas encore certifié J2EE comme l'est Jboss ou BEA Weblogic

**IBM Websphere 7.0 (Vela)** devrait sortir prochainement après la sortie de la 6.0 en décembre 2004. Aucune idée si les EJB3 seront intégrées dans cette version.

**BEA** a participé à la rédaction des spécifications EJB3.0 et a sorti mi-novembre 2005 Weblogic Server 9.0 beta pour tester. [Site de BEA](#) BEA a aussi annoncé le rachat de SolarMetrics Inc. ([Communiqué de presse](#) en français) pour conforter et certainement acquérir des compétences EJB3.0. SolarMetrics propose Kodo, un moteur de persistance compatible à la fois EJB3.0 et à la fois JDO. Donc c'est en préparation et BEA sortira une pré-version début 2006.

### Conclusion

Les EJB3 apportent de bonnes solutions sur les points décriés tels que le modèle de programmation, le modèle composant et la mauvaise gestion de la persistance.

Les EJB évoluent vers une version plus "mature" et cette nouvelle spécification prend en compte aussi les applications réellement réalisées depuis quelques années. Le modèle distribué est peu utilisé, la gestion des ressources restaient trop spécifiques aux containers.

Nous avons vu aussi au passage la puissance du serveur d'application Jboss. Ce n'est pas un simple conteneur J2EE mais un kernel serveur qui permet dès aujourd'hui de coder des EJB3.0 sans trop d'effort.

Voilà pour le résumé. Je parlerai prochainement de JBoss Seam, une application qui propose JBoss AS + EJB3 + JSF (MyFaces) le tout prêt à l'emploi.

## Premiers tests de JBoss Seam

JBoss Inc. a annoncé la sortie de [JBoss Seam](#) à la mi-septembre. Seam (couture, soudure en français) est un framework applicatif pour Java EE 5 qui regroupe les EJB 3.0 et Java Server Faces (JSF) pour permettre d'avoir un environnement prêt pour développer une application rapidement.

Seam se base sur les annotations et l'architecture EJB 3.0 pour intégrer la partie présentation ainsi que la gestion d'un workflow, le tout géré par un conteneur. Concernant la partie vue, Seam utilise JSF ainsi que les Facelets pour que l'écriture de la partie vue soit orientée composant. Basé sur du xhtml, la mise en page et la création des pages est donc simple. Côté persistance, grâce à EJB 3.0 il est facile de persister les données vers une base de données grâce à l'EntityManager des EJB 3.0.

J'ai téléchargé et installé JBoss Seam en 15mn. Facile à installer, j'ai ensuite configuré un nouveau projet sous [IDEA IntelliJ \(Best IDE for Java 2005\)](#) pour commencer à regarder le moteur de la machine... Tout d'abord voici comment l'installer rapidement pour tester (traduction du site de JBoss + mes commentaires):

- Installer Java 5 (obligatoire pour les EJB 3.0) via le site de SUN
- Installer Jakarta Apache ANT si nécessaire (1.6.2 minimum)
- Télécharger JBoss AS 4.0.3 en sélectionnant le profil EJB3.0 ([Installer JBoss AS 4.0.3 avec Java Web Start](#))
- Tester et démarrer JBoss via le script de démarrage run.bat dans %JBASS\_HOME%/bin. Vérifiez que JBoss utilise bien Java 5 si vous avez plusieurs JDK installés.
- [Téléchargez JBoss Seam](#) et le décompresser dans un répertoire de travail en dehors de l'arborescence du serveur JBoss AS.
- Editer et configurer le fichier **build.properties** de votre répertoire Seam en spécifiant le chemin d'installation de JBoss AS
- Dans le répertoire **jboss-seam/examples/booking** directory, tapez **ant deploy** pour compiler et déployer la démo Hotel Booking.
- Lancer un navigateur et essayer de vous connecter à l'adresse <http://localhost:8080/seam-booking/> (sinon aussi dispo ici [Demo sur le site de JBoss](#))
- Créez un compte en cliquant sur Register pour entrer ensuite sur l'interface principale.

JBoss Seam propose de gérer le contexte d'une conversation au sein d'une application J2EE. Pour comprendre la notion de Conversation et l'intérêt de Seam, je pense que le mieux est de prendre un exemple. Prenons le cas de l'application de réservation de chambres d'hôtel de Seam. L'achat doit se dérouler en 5 phases.

- Phase 1: l'utilisateur recherche un Hôtel dans une liste d'hôtels
- Phase 2: il examine un hôtel en particulier en le sélectionnant dans la liste. Il peut cliquer sur suivant/précédent pour passer d'un hôtel à l'autre.
- Phase 3: il clique sur "Réserver" ce qui l'amène à une page où il doit confirmer son achat, donner sa date d'arrivée et de départ. Il donne aussi son numéro de CB et d'autres détails pour la facturation. Il termine par un click sur "Continue".
- Phase 4: Le moteur après avoir vérifié les disponibilités pour les dates choisies, propose de conclure la transaction via une action "Confirm".
- Phase 5: le moteur enregistre la commande, il reçoit une facture détaillée qu'il peut imprimer,

l'achat est validé et le panier est vidé.

Si vous avez déjà codé ce genre d'application, je pense que vous savez que les problèmes de gestion de session, d'unicité des transactions, de la taille de l'environnement dans l'objet `HttpRequest` sont parfois voir souvent un problème. Sinon je vais expliquer ici les soucis que l'on rencontre souvent: **1) Le client ne termine pas la transaction** Pour une raison obscure (météorite, lassitude, plantage de Windows...) votre client ne termine pas sa transaction. Le souci est que pendant un delta T de temps sa session reste sur le serveur pour... rien ! **2) Le navigateur du client n'accepte pas les cookies** Dans ce cas, la gestion de la session s'effectue en envoyant et en recevant via la session HTTP un ensemble de paramètres pour redéfinir à chaque appel le contexte d'exécution, ce qui est lourd. D'autre part il faut alors que la partie Vue de l'application s'occupe de gérer ces sessions... ce qui complique et ralentit l'application. **3) Le client ouvre un autre navigateur et lance un autre achat** Il va alors falloir isoler les transactions du côté serveur ce qui est loin d'être le cas pour beaucoup de site Internet que je connais, spécialement ceux basés sur des moteurs PHP pour gérer un panier. Ils ne fonctionnent pas. **4) Le client clique sur le bouton Back de son navigateur** Combien d'application Web ne savent pas gérer le bouton Back et Forward ? Je me souviens d'une version du site `www.sncf.fr` qui m'a fait acheter 2 Paris-Bordeaux car j'avais utilisé le bouton Back et ensuite Forward de mon navigateur... Inimaginable sur un système grand public. JBoss Seam en s'attachant à gérer la notion de **Conversation** va nous permettre, à nous développeur Java, de conduire une transaction et un dialogue avec l'utilisateur très facilement. Je crois qu'il s'agit réellement d'une nouveauté, qui fait que Seam est intéressant.

Voyons comment cela se présente: Un des exemples de Seam est l'action "HotelBookingAction". Il s'agit d'un Session Stateful Bean comme le précise l'annotation `@Stateful` Le tag `@Name` est un tag Seam pour donner un nom à ce SSB. Il servira dans la partie JSF pour déclencher les actions et les brancher sur des boutons dans la page Web. Le tag `@Interceptor` est un tag Seam nécessaire pour réaliser la **Bijection**. J'en parlerai plus bas. Le tag `@Conversational` est un marqueur pour dire que ce SSB doit s'exécuter dans un contexte de conversation. Enfin le tag `@LoggedIn` est particulier à l'application HotelBooking. Il permet de dire que ce Bean ne doit s'exécuter que si l'utilisateur s'est préalablement authentifié. Pratique non?

```
@Stateful
@Name("hotelBooking")
@Interceptor(SeamInterceptor.class)
@Conversational(ifNotBeganOutcome = "main")
@LoggedIn
public class HotelBookingAction implements HotelBooking, Serializable {
    private static final Logger log = Logger.getLogger(HotelBooking.class);

    @PersistenceContext(type = EXTENDED)
    private EntityManager em;

    private String searchString;

    @DataModel
    private List<Hotel> hotels;
    @DataModelSelectionIndex
    private int hotelIndex;

    @Out(required = false)
    private Hotel hotel;

    @In(required = false)
    @Out(required = false)
    @Valid
    private Booking booking;

    @In
    private User user;

    @In
    private transient FacesContext facesContext;

    // Specify that this method starts a long running conversation
    @Begin
    public String find() {
        hotel = null;
        String searchPattern = searchString == null ? "% : '%'" + searchString.toLowerCase().replace('*', '%') + '%';
        hotels = em.createQuery("from Hotel where lower(name) like :search or lower(city) like :search or lower(zip) like :search or lower(address) like :search")
            .setParameter("search", searchPattern)
            .setMaxResults(50)
            .getResultList();

        log.info(hotels.size() + " hotels found");

        return "main";
    }

    public String getSearchString() {
```

```
    return searchString;
}
```

---

```
public void setSearchString(String searchString) {
    this.searchString = searchString;
}
```

---

```
public String selectHotel() {
    if (hotels == null) return "main";
    setHotel();
    return "selected";
}
```

---

```
public String nextHotel() {
    if (hotelIndex < hotels.size() - 1) {
        ++hotelIndex;
        setHotel();
    }
    return null;
}
```

---

```
public String lastHotel() {
    if (hotelIndex > 0) {
        --hotelIndex;
        setHotel();
    }
    return null;
}
```

---

```
private void setHotel() {
    hotel = hotels.get(hotelIndex);
    log.info(hotelIndex + "=>" + hotel);
}
```

---

```
public String bookHotel() {
    if (hotel == null) return "main";
    booking = new Booking(hotel, user);
    booking.setCheckinDate(new Date());
    booking.setCheckoutDate(new Date());
    return "book";
}
```

---

```
@IfInvalid(outcome = REDISPLAY)
public String setBookingDetails() {
    if (booking == null || hotel == null) return "main";
    if (!booking.getCheckinDate().before(booking.getCheckoutDate())) {
        log.info("invalid booking dates");
        FacesMessage facesMessage = new FacesMessage("Check out date must be later than check in date");
        facesContext.addMessage(null, facesMessage);
        return null;
    } else {
        log.info("valid booking");
        return "success";
    }
}
```

---

```
@End
public String confirm() {
    if (booking == null || hotel == null) return "main";
    em.persist(booking);
    log.info("booking confirmed");
    return "confirmed";
}
```

---

```

@Destroy
@Remove
public void destroy() {
    log.info("destroyed");
}
}

```

J'ai repris ici l'ensemble du code source de la démo Booking de Seam mais je vais expliquer le rôle des différentes annotations rencontrées pour vous présenter un des principes de Seam, **le contexte de conversation**.

```

@Begin
public String find() {

```

La méthode find est marquée avec l'annotation **@Begin** et la méthode confirm est marquée avec l'annotation **@End**. Ces 2 marqueurs permettent de commencer et de terminer la transaction. Lorsque l'utilisateur déclenche l'action find pour lister les hôtels, la conversation commence. Il va pouvoir ensuite visualiser un hôtel en détail, passer d'un hôtel à l'autre jusqu'à choisir celui qui l'intéresse.

```

@End
public String confirm() {

```

La method confirm() est appelé une fois qu'un hôtel a été sélectionné et qu'un objet Booking a été créé. La méthode setBookingDetails est marquée avec l'annotation **@IfInvalid(outcome = REDISPLAY)** qui permet de dire que si dans la méthode lors de la spécification de la réservation il y a une erreur, alors la page courant est réaffichée. **Qu'est-ce que la Bijection ?** Pour comprendre la bijection telle qu'elle est proposée par Seam, il faut connaître un moteur d'IoC (Inversion of Control) comme Spring. En quelques mots, l'inversion de contrôle est un pattern qui permet de déclarer des variables dans votre code, d'écrire une méthode setMyObject(Object newObject) mais de déléguer à un moteur l'assignation de cette variable. La Bijection c'est à la fois de l'inversion de contrôle mais aussi de la réinjection de résultat dans le reste de l'application.

```

@In(required = false)
@Out(required = false)
@Valid private Booking booking;

```

Le marqueur **@In** de Seam demande au moteur d'injecter la valeur de booking à l'exécution. Le marqueur **@Out** signifie que l'objet booking sera *outjected* vers une variable de contexte après que la méthode a été appelée. La variable booking qui est dans le contexte de l'utilisateur sera donc automatiquement remplacée par la valeur de cette variable après chaque appel. Cette notion d'outjection est spécifique à Seam et un peu complexe à expliquer ici. Le marqueur **@Valid** vient d'Hibernate. Il permet de valider que l'objet est valide, que les attributs de cet objet obligatoire sont bien renseignés pour pouvoir le sauver. **Conclusion** JBoss Seam est un moteur basé sur les EJB3 et JSF qui vise à faciliter l'écriture d'une applicaiton. Je pense cependant qu'en attendant de bons tutoriaux ou livres sur le sujet, se lancer sur JBoss Seam pour un projet en production est encore trop tôt au moment où j'écris ces lignes (janvier 2006). Réellement novateur, Seam permettra dans quelques mois d'écrire des applications J2EE de nouvelles générations efficacement, facilement et rapidement. A suivre donc.

## Salon Solutions Linux 2006, PJBUG et JBoss Seam par T.Heute

Je vais passer au [Salon Solutions Linux 2006](#) mercredi 1er février. Il y a toujours des choses intéressantes à voir, surtout du côté des associations et du libre. Je vais aller aussi pour la première fois à la réunion [Paris JBUG](#) (Paris JBoss User Group) le groupe des utilisateurs JBoss à Paris. Sacha Labourey, directeur général de JBoss Europe, participera à la réunion et devrait faire une présentation sur la haute disponibilité. Cela me rappelle il y a 1 an lorsque j'avais mis le nez dans JGroups et la partie HA de JBoss. C'est un très bon souvenir, je crois que j'avais blogué à l'époque là dessus ([un peu ici lorsque j'étais à Berlin](#) et aussi [un peu là pour le projet PDS Console en 2005](#)).

Suite à mon post sur JBoss Seam, j'ai vu que Thomas Heute est passé poster un commentaire ci-dessous (Chef de projet sur JBoss Seam) . Dans un article qu'il a écrit sur [Jailb.org](#) il explique que JBoss Seam est destiné à souder la couche de présentation JSF (Java Server Faces) et les EJB3 en prenant en compte le contexte d'exécution. Il y a le contexte de la requête, avec l'envoi de paramètres aux serveurs. Le contexte de la session ensuite qui va gérer le temps de présence et de visite d'un visiteur avec ses différentes actions. JBoss Seam comme l'explique Thomas ajoute en plus 3 nouveaux concepts de contexte. Ci-dessous un extrait de son article. "(*extrait de <http://www.jaib.org/content/view/2111/1111/>)*)  
*JBoss Seam reconnaît l'importance de ces contextes et en apporte de nouveaux:*

- **Conversation:** *Différentes requêtes consécutives font souvent partie de la même conversation. Par exemple, une conversation peut requérir qu'un utilisateur passe par le cycle requête/réponse plusieurs fois. Un exemple de scénario serait un Wizard. JBoss Seam permet la définition et la gestion de conversations de manière simple.*
- **Process:** *L'intégration de BPM (Business Process Management) dans une application web tel que permise par jBPM est également introduite. Considérer la situation suivante: un business process requiert un workflow avec deux acteurs: un acteur crée un élément, alors que le second a la responsabilité de l'approuver. Les deux acteurs auront chacun besoin d'une ou plusieurs conversations avec l'application pour compléter leurs tâches respectives. JBoss Seam permet*

facilement et de manière transparente de définir ces tâches.

- **Application:** Il y a toujours besoin d'un contexte disponible depuis le début de l'application jusqu'à la fin de celle-ci, JBoss Seam le permet également.

A suivre donc. Nul doute que dans quelques temps JBoss Seam va commencer à faire parler de lui. A retenir: JBoss Seam est une nouvelle manière d'écrire et de décrire une application J2EE en se reposant sur les annotations pour lier les EJB3 et JSF. Simple, puissant et efficace.

## Comparaison de différents logiciels d'intégration continue

Utilisateur de [CruiseControl](#) depuis presque 2 ans, au moment d'installer et de mettre celui-ci en place dans ma nouvelle équipe je me suis demandé s'il n'existe pas d'autres solutions [d'intégration continue](#). J'avais entendu parler de Luntbuild et Anthill. Mais bon, quelques tests plus tard je pense que CruiseControl pour de l'intégration continue d'un projet Java est le plus puissant et le plus simple à installer. Il a l'avantage d'être gratuit et open source. Sur [cette page](#) vous trouverez un tableau récapitulatif des différentes fonctionnalités de CruiseControl comparées à ses concurrents.

J'ai rencontré un petit problème sur l'application web de Cruise Control. Lorsque le visiteur clique sur l'onglet "Metrics" une page blanche est affichée et une exception est levée sur Tomcat.

```
2006-01-25 09:09:30 StandardContext[/cruisecontrol]jsp: http://ptxsfrabench2:8080/cruisecontrol/buildresults/kiwi: (null)
java.lang.NoClassDefFoundError
    at org.jfree.chart.ChartFactory.createPieChart(Unknown Source)
    at de.laures.cewolf.taglib.CewolfChartFactory.getChartInstance(CewolfChartFactory.java:76)
    at de.laures.cewolf.taglib.SimpleChartDefinition.produceChart(SimpleChartDefinition.java:30)
    at de.laures.cewolf.taglib.AbstractChartDefinition.getChart(AbstractChartDefinition.java:81)
    at de.laures.cewolf.taglib.ChartImageDefinition.ensureRendered(ChartImageDefinition.java:131)
    at de.laures.cewolf.taglib.ChartImageDefinition.getBytes(ChartImageDefinition.java:125)
    at de.laures.cewolf.storage.SerializableChartImage.(SerializableChartImage.java:51)
    at de.laures.cewolf.storage.SessionStorage.storeChartImage(SessionStorage.java:57)
    at de.laures.cewolf.storage.SessionStorage.storeChartImage(SessionStorage.java:35)
    at de.laures.cewolf.taglib.tags.ChartImgTag.doStartTag(ChartImgTag.java:74)
    at org.apache.jsp.main_jsp._jspx_meth_cewolf_img_0(main_jsp.java:959)
    at org.apache.jsp.main_jsp._jspService(main_jsp.java:640)
    at org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:94)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:802)
    at org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:324)
    at org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:292)
    at org.apache.jasper.servlet.JspServlet.service(JspServlet.java:236)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:802)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:237)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:157)
    at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:214)
    at org.apache.catalina.core.StandardValveContext.invokeNext(StandardValveContext.java:104)
    at org.apache.catalina.core.StandardPipeline.invoke(StandardPipeline.java:520)
    at org.apache.catalina.core.StandardContextValve.invokeInternal(StandardContextValve.java:198)
    at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:152)
    at org.apache.catalina.core.StandardValveContext.invokeNext(StandardValveContext.java:104)
    at org.apache.catalina.core.StandardPipeline.invoke(StandardPipeline.java:520)
    at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:137)
    at org.apache.catalina.core.StandardValveContext.invokeNext(StandardValveContext.java:104)
    at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:118)
    at org.apache.catalina.core.StandardValveContext.invokeNext(StandardValveContext.java:102)
    at org.apache.catalina.core.StandardPipeline.invoke(StandardPipeline.java:520)
    at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:109)
    at org.apache.catalina.core.StandardValveContext.invokeNext(StandardValveContext.java:104)
    at org.apache.catalina.core.StandardPipeline.invoke(StandardPipeline.java:520)
    at org.apache.catalina.core.ContainerBase.invoke(ContainerBase.java:929)
    at org.apache.coyote.tomcat5.CoyoteAdapter.service(CoyoteAdapter.java:160)
    at org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:799)
    at org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.processConnection(Http11Protocol.java:705)
    at org.apache.tomcat.util.net.TcpWorkerThread.runIt(PoolTcpEndpoint.java:577)
    at org.apache.tomcat.util.threads.ThreadPool$ControlRunnable.run(ThreadPool.java:683)
    at java.lang.Thread.run(Thread.java:536)
```

3 mn sous Google plus tard, j'ai trouvé la solution. Le problème vient du fait que j'ai installé CC sur une SUN sans display et donc sans serveur X. Il faut simplement modifier le script de démarrage de Tomcat en ajoutant les options suivantes

```
# For X11 and CruiseControl Metrics tab
JAVA_OPTS=-Djava.awt.headless=true
export JAVA_OPTS
```

## Deuxième réunion du PJBUG à Paris

Mardi 1er février dernier à l'occasion du salon "Solutions Linux" j'ai participé à la première réunion du [PJBUG](#). Le PJBUG (Paris JBoss User Group) a pour objectif de permettre aux utilisateurs de JBoss de se rencontrer et de participer en exclusivité à des présentations techniques de la part de JBoss Inc. A l'occasion du salon, Luc Texier venu tout spécialement de Neuchâtel nous a présenté l'architecture et les API qui permettent de faire du clustering avec JBoss. François le Droff a fait un bon compte-rendu, je vous invite à aller le lire [directement sur son blog](#) dans la section JBoss.

J'ai bien aimé la présentation de Luc. Connaissant déjà bien JGroups, j'y suis plus allé par curiosité pour

voir "qui serait là" que par réel intérêt technique. Sa présentation était technique et complète. A la fin, Luc a demandé à l'assistance quelques retours sur nos propres expériences. Pour ma part, j'ai besoin de matières premières pour évangéliser les autres développeurs ainsi que le management.

Raphaël Semeteys, d'Atos Origin, a demandé à l'assistance si certains seraient intéressés pour présenter des solutions autour de JBoss lors d'une prochaine réunion. Je vais y réfléchir. J'ai bien envie de présenter le système de monitoring réseau développé l'an dernier autour de JGroups. Va falloir que j'y réfléchisse et que je valide si j'ai le droit ou non de présenter ce produit... A réfléchir.

## SCA Service Component Architecture en 10 minutes

Autour de SOA, je vous propose un résumé aujourd'hui de la spécification SCA (Service Component Architecture) ainsi que quelques liens vers des projets intéressants autour de SCA et SDO...

SCA signifie Service Component Architecture. Il s'agit d'une spécification ([en ligne ici](#)) qui vise à promouvoir un système découplant l'implémentation de l'assemblage des services. SCA est indépendant du langage. Une architecture de service basée sur des composants permet de fournir un service défini par un ensemble d'interfaces. Des propriétés de données permettent aussi de définir des attributs globaux pour plusieurs services. Par exemple la langue ou la monnaie de l'utilisateur. Un composant SCA est une brique logicielle qui est assemblée dynamiquement par un moteur. Elle utilise un service en entrée et fait appel via des références à d'autres services. Des attributs (les propriétés) permettent de configurer le contexte du SCA globalement. Imaginez un composant de type GestionCompte. Celui-ci configure une implémentation GestionCompteImpl, branche des références vers les services CompteDataService et CotationDevisesService. De plus une propriété "Monnaie" est mise à EURO pour que le compte soit géré en euro. Un composant SCA est un assemblage dynamique de plusieurs services autour d'un code de base, l'implémentation de références. En s'inspirant de l'injection de contrôle type Spring, SCA permet à un moteur externe de brancher les références des services dynamiquement. Il n'est donc pas de la responsabilité du programmeur qui écrit le composant de se lancer dans l'écriture de ServicesLocator par exemple. L'injection s'applique aussi aux propriétés (la monnaie "EURO").

Concernant la couche de transport, celle-ci est invisible pour le composant SCA. Ce qui permet donc d'utiliser des web-services pour discuter avec d'autres services, mais aussi JMS, Corba ou des appels locaux. C'est un gain de souplesse et d'évolutivité.

Par rapport à Java, SCA supportera des objets simples (POJO). Une implémentation basée sur des EJB sera aussi possible ainsi qu'une implémentation BPEL (OASIS WS-BPEL standard).

Les composants SCA sont déployés au sein de ce que la spéc appelle des "Module Assembly". Les "Entry Points" sont les portes d'entrées du module qui contient un ou plusieurs composants SCA. Lorsque les composants SCA nécessitent des services externes, ceux-ci sont appelés "ExternalServices". Plusieurs Modules sont regroupés par sous-système pour former une Application.

Le Binding est le système qui permet de brancher et d'assurer la couche transport d'une architecture SCA. Un des postulats de départ est de séparer la partie business logic de la partie transport. SCA supporte les Web Services, JMS, StatelessSession Bean ou encore des procédures stockées en base. Par rapport aux échanges entre service, l'architecture SCA permet des échanges asynchrones de type "one-way" comme pour envoyer un message, de type callback et de type conversational. La gestion d'une conversation est par exemple une demande de crédit pour notre service bancaire. Plusieurs étapes dans l'échange de message seront nécessaire pour obtenir un résultat.

Pour résumer, les caractéristiques de SCA sont les suivantes:

- 
- la logique d'une application est divisée en composant implémentant des services métiers
- un composant dispose d'interface orientées métiers et services. Il n'a pas d'interfaces liés au middle-ware
- les composants peuvent être branchés entre eux pour les réutiliser. Cette capacité d'assemblage d'une architecture SCA permet de créer un réseau de services réutilisables
- SCA force une séparation entre les besoins de la personne qui implémente un composant de celle qui assure l'assemblage système des composants.
- SCA peut être implémenté par dessus un grand nombre d'environnement middleware.
- Les composants sont implémentés et utilisés de la même manière quelque soit le langage ou la technologie utilisée.
- SCA permet une qualité de service en s'assurant que le support des transactions, de la sécurité et des messages asynchrones est appliqué de manière déclarative et dynamique sans que la programmation d'un service force l'écriture ou l'implémentation de briques systèmes. C'est un principe aussi repris par les EJB3.
- Que le composant soit déployé localement ou à distance, il est accédé via ses interfaces métiers. L'assemblage de composants peut s'effectuer à différents niveaux pour permettre une plus grande flexibilité et une meilleure visibilité des composants.
- Une variété de ressources comme les Web Services, les EJB's ou les EIS peuvent être modélisés comme des composants distants et utilisés sans avoir connaissance de l'implémentation ou de la couche transport. Cependant certains types de transport restreignent les possibilités de qualité de service supportées.
- SCA supporte différentes technologies pour exprimer les interfaces métiers des composants, que ce soit WSDL ou des interfaces Java.
- Des composants sans interface métiers peuvent être utilisés pour permettre l'accès aux données, en séparant le problème de la persistance des données de la logique métier. Cela facilite aussi le portage de composants entre différents runtimes.

- Les capacités d'infrastructure comme la sécurité et les transactions sont appliqués sur les interactions entre les composants (ProxyPattern) plutôt que d'utiliser du code dans le composant. C'est une idée que l'architecture JEMS de JBoss a mis en place avec succès depuis 2 ans déjà. Cela permet de bien séparer la partie métier de la partie infrastructure.
- Les composants applicatifs peuvent être reprogrammés à la volée en déléguant à des composants de type "[StrategyPattern](#)" certains traitements. C'est un pas vers les moteurs de Workflow ou les règles engines.
- SCA définit un modèle abstrait pour l'implémentation et un modèle pour l'accès au composant. Le modèle est découpé en différentes implémentations concrètes dans une grande variété de langage comme JAVA, C++, BPEL ou des scripts XSLT. SCA essaye d'être intrusif au minimum avec seulement quelques APIs. L'injection des dépendances permet à un composant SCA de ne pas dépendre d'API externe pour retrouver d'autres composants.
- Les objets échangés entre différents composants via des interfaces métiers distantes est défini dans la spécification Service Data Object (SDO)

Sur apache.org le projet [Tuscany](#) a été créé pour regrouper 5 projets destinés à fournir une architecture type SOA:

1. SCA Runtime for Java (service component architecture)
2. SDO 2.0 Runtime for Java
3. Data Access Service for Java
4. SCA Runtime for C++
5. SDO 2.0 Runtime for C++

Tuscany(en français la Toscane en Italie) est vraiment au tout début. Pour récupérer l'implémentation de référence il faut la charger via Subversion.

Un exemple commenté d'application basée sur SCA est disponible:

[http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-sca/SCA\\_BuildingYourFirstApplication\\_V09.pdf](http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-sca/SCA_BuildingYourFirstApplication_V09.pdf)

La sortie des EJB3 pour Java va permettre une intégration facile de cette architecture. La gestion de la persistance et l'injection de dépendances sont 2 mécanismes puissants d'ores et déjà disponibles avec la spécification des EJB3 pour Java. Les Sessions Beans simplifiés des EJB3 peuvent servir naturellement de composant (les composants dont j'ai parlé au début) comme brique de base. Il est possible d'injecter des dépendances simplement grâce aux annotations dans un EJB3. Tout ceci fait que SCA + EJB3 semble naturel et plus facile à mettre en place.

En conclusion, SCA est donc une spécification d'architecture qui vise à faciliter l'écriture de composants métiers pour les intégrer dans un conteneur global.

Pour terminer, lors de la lecture de la spéc., je suis tombé sur un nouveau barbarisme. Après POJO (plain old java object) j'ai vu dans la section 5.6.1 le mot **POX** qui signifie Plain Old XML... on arrête pas le progrès.

## JBoss Seam 1.0 beta 2 est sortie le 8 février

La version JBoss Seam 1.0 beta 2 est sortie le 8 février dernier. [L'annonce officielle](#) sur le blog de Gavin King (fondateur de Hibernate, contributeur de la spec des EJB3.0 entre autre) présente 3 nouveautés dans cette nouvelle beta. La première nouvelle fonctionnalité est la gestion des Workspaces. Ce système tire profit de la gestion du mode conversationnel de Seam. Elle permet de placer dans une page web un sélecteur pour laisser l'utilisateur se déplacer sur le site dans le cadre d'une conversation. Cela revient à simuler un deuxième navigateur qui visite votre site, ou un deuxième tab pour Firefox, tout en restant sur le même écran. L'avantage d'avoir un même contexte est que cela permet alors au visiteur de passer d'un écran à l'autre très simplement. Il est possible d'imaginer que grâce à ce système il devient possible de comparer 2 produits facilement. Imaginez que vous êtes sur Amazon. Vous commencez vos achats et placez 2 CD dans votre panier. En cours d'achat, un ami vous demande de lui faire aussi une commande pour lui, mais il ne veut pas exactement les mêmes CD. Sans ouvrir un autre navigateur, grâce à la gestion du mode Workspace de Seam vous pourrez dupliquer puis modifier la session, sans devoir gérer (en temps que développeur Java) les sessions vous-mêmes.

Deuxième nouveauté de cette version de Seam, l'intégration de jBPM, le moteur de workflow proposé par JBoss. La gestion d'un workflow vous permet de définir des tâches puis de gérer des états pour vos objets sur le serveur. Prenons un bug par exemple. Initialement avec un état NEW, pour le passer à l'état ASSIGNED il faut définir des transitions, des acteurs (qui a le droit de changer l'état) et des règles de validation. Grâce à jBPM il est possible de mettre en place facilement des systèmes de traitement des données asynchrones.

Enfin la troisième nouveauté est la fonction de *reverse engineering* qui vient en fait des outils d'Hibernate. Cette fonction vous permet en quelques minutes de créer votre première application Seam en partant de votre modèle Hibernate. Creation, Recherche, Mise-à-jour et Effacement (Create, Research, Update and Delete). Cet utilitaire est embarqué dans la version custom d'Eclipse destinée à JBoss Seam. J'avoue que je ne suis pas un grand fan d'Eclipse et que je préfère Idea IntelliJ. Mais là, ça vaut le détour. Des captures d'écrans ainsi que plus d'explications sont disponibles dans [la section 9.2 de la doc de Seam](#). Thomas Heute a fait aussi [une vidéo de cette fonctionnalité](#).

Enfin pour faire fonctionner cette version de Seam, il faut au préalable installer la dernière version de JBoss AS 4.0.4 RC1. Je vous recommande d'utiliser la version Java Web Start en cliquant sur "Run Installer" [sur cette page](#).

## AJAX, déjà un an, déjà 7 ans

AJAX existe officiellement depuis un an suite à l'article d'un consultant en mars 2005. Cependant d'un point de vue technique AJAX existe depuis 1999 comme l'explique l'article de [Wikipedia](#).

### Avez-vous entendu parler du Web 2.0 ?

Basé sur des technologies comme AJAX, DHTML, XML ou Soap, le concept de "Web 2.0" regroupe l'annonce de l'arrivée prochaine d'application légère dans votre navigateur. Microsoft ainsi commence à croire que Google pourrait être son plus sérieux compétiteur. La présentation de Google faites aux analystes financiers parle ainsi d'un nouveau concept de Web. Google proposerait un espace de stockage illimité pour vos fichiers, des applications légères embarquées dans votre navigateur pour les utiliser, et donc la fin des logiciels installés sur votre ordinateur personnel. Un espace disque infini serait alors offert et accessible de votre travail, de votre téléphone mobile et de votre télévision à la maison... Peut-être que d'ici 4 à 5 ans, Google prendra la première place devant Microsoft ?

En attendant celui-ci cependant n'attend pas les bras croisés de se faire distancer. Microsoft annonce ainsi quelques nouveautés basé sur AJAX dont Atlas, un framework de développement qui devrait permettre aux éditeurs de logiciels d'offrir une version "Web 2.0" de leurs logiciels. La version CTP d'Atlas est sortie au début du mois de mars. Allez faire un tour (avec Internet Explorer) aussi sur [Live.com](#) de Microsoft. Vous y trouverez un moteur de recherche à la Google (sans publicité apparente) utilisant AJAX. Le logo "Windows Live Beta" fait cependant un peu croire à la sortie d'un OS en ligne, alors qu'il n'en n'est rien.

**Et les développeurs...** Depuis quelques mois, on voit ici et là de plus en plus de librairies et de frameworks d'application clairement orienté Web 2.0. Leur objectif est de faciliter la création de site "AJAX" tout en conservant du côté du serveur un modèle qui tienne la route. Vous pouvez trouver sur [ToutAjax](#) une liste assez complète d'applications, de démos et de librairies pour se lancer.

Ce qui m'intéresse en temps qu'architecte J2EE, c'est la facilité de mise en oeuvre d'une ergonomie et d'une interactivité comme AJAX dans mes applications Webs, ainsi que la facilité de maintenance ensuite. Du côté du client, il existe désormais des librairies gratuites puissantes à intégrer dans une application qui nous permettent de construire nos interfaces en ayant l'interactivité d'AJAX. Quelques librairies comme [Script.aculo.us](#), [Rialto](#) ou [Dojo Toolkit](#) sont puissantes et efficaces mais pas toutes semblable. [Rialto](#) est celle qui m'intéresse en ce moment, notamment à cause du support de Java Server Faces.

**Il me semble que JSF + AJAX est un peu le couple à marier en 2006.** Java Server Faces apporte un modèle composant à une application et permet de définir ses composants tels qu'ils seront représentés dans le navigateur. Il n'a pas fallu attendre longtemps avant de voir apparaître ici et là des composants JSF utilisant AJAX [comme ce composant pour télécharger uploader un fichier](#) vers une JSP. Si vous avez lu aussi mon post sur Shale (Struts 2.0 mais pas vraiment) vous savez déjà que Shale aura un composant de Rendering et s'intégrera avec Apache MyFaces pour pouvoir au final offrir des composants JSF utilisant AJAX (entre autre).

Pour terminer, je vous donne un lien vers le fameux PetStore mais réalisé avec donc ces nouveaux composants qui font que dans quelques mois, le web va changer. [Ajax PetStore](#) vu sur l'excellent site [Ajaxian.com](#)

## Struts Shale ne sera pas Struts 2.

### Qu'est-ce que Struts Shale ?

Struts Shale (Schiste comme la pierre en français) est le successeur de Struts. Cependant comme l'a annoncé le groupe Apache, il n'y aura pas de Struts 2.0. Shale est un nouveau framework MVC basé sur Java Server Faces 1.1 qui se concentre sur la partie Contrôleur. La partie Vue reste déléguée à une implémentation de JSF telle qu'[Apache MyFaces](#) ou JSF-RI de SUN.

### Struts 1.0 est mort ?

Struts devient Struts Classic avec sa propre home-page et va continuer à être développé et maintenu. Struts n'est certainement pas mort pour l'instant. Voici une partie de l'annonce officielle:

*The Apache Struts team is pleased to announce the adoption of its latest subproject, Struts Shale, a JSF-based framework. Right now, there are no concrete plans for a 2.x release of Struts Classic. The people working on the 1.x series want to evolve the code base without breaking backward compatibility. If we do our job right, then a 2.x release may never be needed. It's an entirely new codebase. Struts Shale is an MVC framework for web applications, like Struts Classic, but it's a fresh start. Struts Shale is the best choice for developers eager to leverage the latest technologies, like Java Server Faces.*

### Qu'apporte Shale ?

Shale propose quelques ajouts à JSF pour faciliter le développement d'application Web.

- Un **ViewController** permet d'associer un simple Bean Java à la vue orientée composant proposée par JSF. Il s'agit en fait d'une class qui propose des callbacks standards pour qu'un Bean soit notifié selon les différentes phases de JSF.
- Le **DialogManager** permet l'ajout d'un mode "conversationnel" similaire à JBoss Seam. Un fichier de configuration XML définit l'enchaînement des écrans selon les états retournés par les Beans. Cela permet une solution plus robuste que l'utilisation du fichier navigation.xml de JSF qui est trop orienté composants. Ce mode de dialogue permet ensuite d'implémenter une machine à état et de définir donc un diagramme d'état pour l'application. La mise-en-place s'effectue via un fichier de configuration XML.
- L'**ApplicationManager** est un front controller qui permet l'ajout de la notion d'application à JSF

qui sans cela, reste très orienté "page". Typiquement, l'application manager sera un module activé à chaque requête avant et après l'exécution du code de l'application pour fournir des services métiers standards comme l'authentification et l'autorisation. Il permet aussi de filtrer des ressources à partir de l'adresse du client et donc d'exécuter ou non une partie de l'application.

- Le module **Validation** est un ajout à JSF qui vous permet de coder la validation de ce que l'utilisateur entre dans un formulaire, que ce soit du côté serveur ou du côté client. En pratique, de nouveaux tags TLD permettent d'ajouter un système de validation dans les pages JSP. Très pratique et simple à utiliser.
- Le module **Remoting** qui n'est pas encore documenté doit permettre de faciliter la mise en place et le dialogue entre une page contenant du javascript pour faire de l'AJAX et une partie serveur avec un Bean géré par JSF.
- L'intégration du moteur d'injection **Spring** permettra de créer facilement des Beans gérés par JSF et injecté par Spring dans toute l'application. Pas simple à mettre en oeuvre car Spring utilise son propre ClassLoader, c'est certainement un ajout majeur qui vise à faciliter l'intégration et la configuration dynamique d'une application.
- **Clay** est un module qui intéressera ceux qui travaillent avec des Web Designers. Il vous permet de coder simplement en HTML l'ensemble de vos pages et d'ajouter ensuite quelques balises HTML spéciales pour que le moteur transforme ces pages en vue JSF lors de l'exécution. Cela permet donc de faire le design des pages en dehors de la partie JSF.
- **TestFramework** est une extension à JUnit qui à l'aide d'objets simples, permet de tester le code Shale.
- **Tiger Extensions** est un module optionnel qui permet de faciliter l'écriture du code dans le cas où vous désirez passer à Java 5. Au passage, Struts Shale fonctionne avec Java 1.4 et Java 5.0

Une version alpha de Shale devrait sortir d'ici quelques jours d'après la mailing list des développeurs. Le projet n'est donc pas encore assez mature pour être utilisé en production, mais rien ne vous empêche de jeter un oeil. Et donc Shale n'est pas Struts 2.0

Site internet de Struts-Shale: <http://struts.apache.org/struts-shale>

Doc PDF sur l'architecture et les concepts de Shale <http://struts.apache.org/struts-shale/struts-shale.pdf>

## Script.aculo.us et fuites mémoires Javascript

Une nouvelle version de la librairie JavaScript Script.aculo.us est disponible [sur le site http://script.aculo.us](#). Cette librairie vous permet d'ajouter dans vos pages facilement et rapidement des comportements dynamiques et asynchrones afin d'ajouter une petite touche AJAX à vos pages.

Parmi les améliorations notables, on note une mise à jour de la version de la librairie Prototype qui corrige des fuites mémoires d'Internet Explorer. En effet, AJAX c'est bien beau mais lorsque la partie Javascript est mal programmée, cela conduit rapidement à des problèmes de fuites mémoires. J'en parle en connaissance de cause puisque notre moteur souffre actuellement de ce souci. Pour aider les développeurs à rechercher les fuites mémoires du côté client dans le navigateur, il n'existe pas grand chose. Le seul outil pratique que j'ai trouvé s'appelle Drip. Il permet de suivre l'usage de la mémoire d'Internet Explorer et de voir les problèmes de [Javascripts closure](#). L'adresse du site est: [http://outofhanwell.com/ieleak/index.php?title=Main\\_Page](http://outofhanwell.com/ieleak/index.php?title=Main_Page)

Une autre ressource pour suivre l'utilisation mémoire d'IE est expliquée [dans cet article du site Ajaxian.com](#).

Attention donc aux problèmes de mémoire avec Internet Explorer

## C'était y'a un an... souvenez-vous... formidable

A l'occasion du relookage de mon blog, j'ai repris quelques vieux posts dans la partie "Moi". Je rassure tout le monde tout d'abord. Je n'en n'ai pas parlé mais je suis bien papa depuis le 20 octobre 2005 d'un petit Robin. Aujourd'hui âgé de presque 7 mois, cela a changé beaucoup de choses dans notre vie. Tout va bien.

Revenons à cette histoire de **transition vers Bangkok**. L'an passé je travaillais dans une équipe franco-américaine sur un serveur java de gestion des alertes et de portefeuille en temps réel. Restructuration et diverses histoires plus tard, le logiciel a été envoyé à Bangkok en Thaïlande. Pas plus tard que ce matin j'ai parlé avec l'ancien chef de projet via Skype. Tout se passe bien là-bas, même s'il fait un chaleur du diable. De ce que j'ai appris sur les thaïlandais dans le monde professionnel: il y a un gros turn-over. En effet en général, les étudiants se forment pendant 2 ou 3 ans puis prennent ensuite un premier job. Quelques temps plus tard, ils reprennent alors d'autres études, puis reprennent un autre job plus tard, et ainsi de suite. Le premier truc qui étonne en tant que français donc, c'est la vitesse de rotation des équipes.

Différence aussi: la motivation. Là-bas les équipes sont en moyenne plus jeunes, plus dynamique et plus motivées que chez nous. Je ne sais pas exactement pour quelles raisons, c'est simplement un constat fait par 2 français qui y travaillent depuis presque un an...

La culture est aussi différente. La Thaïlande est un pays de l'Asie du Sud-Est qui n'a jamais été colonisée. La place de la famille royale est très importante, la culture/Pe est différente aussi par rapport aux voisins.

Le travail entre les français et les thaïlandais n'est pas évident au premier abord. Dans notre culture latine, les conflits font partis du paysage, les moments de coup de gueule ou de stress font presque partis du paysage. Notre formidable capacité à refuser puis accepter de nouvelles idées n'existe pas chez les thaïs. Ainsi une nouvelle fonction dans notre ancien logiciel, aurait susciter chez nous quelques discussions et

n'aurait pas abouti si celle-ci n'était pas bénéfique pour l'équipe. Au contraire chez nos autres collègues, la culture du "non" est moins forte. Ainsi une idée d'un supérieur sera mise en production alors qu'en France elle aurait suscité moult réunions et conflits... Différence donc...

Quoi d'autre ? le reporting est difficile. Basés sur des faits, il faut faire attention au sens que nous allons donner à la phrase "c'est terminé" ou à la phrase "tout va bien". Autant chez nous à cause de notre pessimisme et de notre culture, nous aurons un "c'est terminé" lié à un programme terminé, autant chez nos collègues thaïlandais le "c'est terminé" est subjectif... Il y a d'ailleurs 3 ou 4 façons de dire "oui" en thaïlandais, chacun de ces oui ayant une nuance différente...

Bref voilà pour cette transition ce que j'avais envie de raconter...

## Maven2... ou la ruée vers l'Ouest

J'ai rencontré [Jean-Laurent de Morlhon](#) de Valtech, venu passer une journée chez Reuters pour entre autre parler de Maven. A cette occasion, j'ai commencé à regarder et envisager la possibilité de migrer nos scripts ANT et notre système de release vers Maven.

Notre équipe développe un framework j2ee qui est ensuite relivré à 4 équipes différentes. La mise en place de Maven nous permettrait de gérer les versions ainsi que l'intégration avec les autres produits plus facilement. Chaque version de notre framework serait installée dans un repository local commun aux 4 équipes. Notre framework sera packagé comme un dépendance parent pour leur projet.

Après avoir téléchargé et installé Maven, j'ai commencé à écrire mon premier POM... Via le blog de Jean-Laurent j'ai trouvé un très bon livre en anglais au format PDF de Vincent Massol et Jason van Zyl qui est disponible gratuitement après vous être enregistré [sur le site de l'éditeur Mergere Inc.](#)

Pour revenir à notre projet, celui-ci est composé de plusieurs jar, ejb, d'une application web et enfin le tout packagé dans un ear. La première étape a été de réorganiser les sources. Merci à Subversion, il est possible de faire cela tout en conservant l'historique. En effet, Subversion contrairement à CVS ou VSS, versionne les répertoires. Il est facile de renommer un répertoire ou de le déplacer tout en conservant l'historique de vos fichiers. Pensez-y si vous devez choisir entre CVS et SVN. Pour revenir à Maven je me suis donc lancé dans un chantier de déménagement pour tout d'abord mettre en place mon arborescence.

Ensuite l'étape suivante est de placer un fichier pom (project object model) par module du projet. J'ai ainsi rangé mes ejbs d'un côté, les jars de l'autre. Parmi les modules nous avons aussi un connecteur JCA. Maven2 est aussi capable de générer un RAR pour ce connecteur.

La gestion des dépendances et des librairies est le **premier intérêt majeur** de Maven. J'ai ajouté en quelques lignes toutes les librairies que nous utilisons (Log4J, Castor, Commons-fileupload, Xerces, Xalan-J...). J'ai regroupé à la racine de mon projet dans le POM de départ mes dépendances. En fait il semble que si des modules fils soient les seuls à se servir de certaines librairies, il vaut mieux les déclarer dans le pom de ce module uniquement et pas à la racine. A vérifier...

J'ai ensuite regardé comment récupérer tous nos tests JUnit. Et bien c'est facile ! Il suffit de relocaliser les tests dans un répertoire src/test/java et hop, Maven les compile et les exécute alors aussi.

Après quelques heures, j'avais la quinzaine de module de notre projet qui compilait. La partie Web est cliente des EJB. Pour charger les dépendances comme des EJB client, il suffit d'utiliser la balise type dans la balise dependency:

```
<dependency>
  <groupId>com.reuters.karma</groupId>
  <artifactId>wfProfiling</artifactId>
  <version>1.0-SNAPSHOT</version>
  <type>ejb</type>
</dependency>
```

Le packaging de l'EAR n'est pas plus compliqué. Le connecteur JCA est packagé sous la forme d'un JAR dans la balise packaging de son POM mais il sera ensuite packagé comme un rar dans l'EAR via la configuration suivante:

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-ear-plugin</artifactId>

      <configuration>
        <displayName>Karma Framework</displayName>
        <description>Karma Framework Maven</description>
        <version>1.4</version>

        <modules>
          <rarModule>
            <groupId>com.reuters.karma</groupId>
            <artifactId>wfBLSetransport</artifactId>

            <bundleFileName>wfBLSetransport.rar</bundleFileName>
          </rarModule>
        </modules>
      </configuration>
    </plugin>
  </plugins>
</build>
```

```

<groupId>com.reuters.karma</groupId>

<artifactId>karma-war</artifactId>
<contextRoot>/karma</contextRoot>
</webModule>
</modules>

</configuration>
</plugin>
</plugins>
</build>

```

Maven est un framework de compilation qui permet aussi via des plugins de faire des choses assez sympathique comme **installer et déployer un JBoss AS 4.0 automatiquement** via l'API [Cargo](#). Cargo est une API qui permet de gérer les conteneurs J2EE comme JBoss, Geronimo ou BEA WebLogic en proposant des méthodes pour démarrer, arrêter et déployer des applications. En intégrant cela dans Maven, il devient alors possible d'automatiser l'installation d'un serveur J2EE dans l'environnement d'un développeur. Le déploiement sera fait aussi automatiquement. Cela vous évitera d'installer plusieurs fois JBoss sur différentes machines. Maven s'en chargera pour vous. Le support de BEA WebLogic est aussi un gros point positif. J'avoue que c'est assez bluffant. Bien entendu, ceci ne s'applique qu'à la partie Production. Il n'est pas question des clients finaux, mais plutôt des développeurs et de l'installation de leur environnement de travail.

Derniers tests réalisés avec Maven, la gestion qualité du projet. J'ai rajouté quelques lignes dans le POM de départ pour que Maven génère des rapports HTML sur la qualité du code source: tests unitaires, PMD, Checkstyle le tout avec JXRref pour accéder au code source. Et bien je dois dire que cela marche. Le seule souci pour l'instant, c'est que j'ai rencontré une erreur avec Maven. En effet, j'utilise des dépendances systèmes afin de charger des librairies Reuters (que nous ne pouvons pas distribuer) pour les données du marchés. Or il semble qu'il y ait un bug entre la gestion de ces dépendances systèmes et la dernière version de Maven.

Lorsque je lance la commande **mvn site** voici l'exception générée:

```

java.lang.NullPointerException
    at org.apache.maven.artifact.resolver.DefaultArtifactResolver.resolve(DefaultArtifactResolver.java:82)
    at org.apache.maven.artifact.resolver.DefaultArtifactResolver.resolve(DefaultArtifactResolver.java:63)
    at org.apache.maven.project.DefaultMavenProjectBuilder.findModelFromRepository(DefaultMavenProjectBuilder.java:467)
    at org.apache.maven.project.DefaultMavenProjectBuilder.buildFromRepository(DefaultMavenProjectBuilder.java:225)

```

Le POM du projet qui génère cette erreur a une dépendance système :

```

<!-- Special dependency for Reuters Report Server API -->
<dependency>
  <groupId>internal</groupId>
  <artifactId>reportserverapi</artifactId>
  <version>1.0.0</version>
  <systemPath>${basedir}/../libs/internal/reportserver/reportserverapi-1.0.0.jar</systemPath>
  <scope>system</scope>
</dependency>

```

Donc voilà où j'en suis aujourd'hui. Je vais aller voir sur le site de Maven2 si je peux ajouter un bug, quoique je doute d'être le premier avec ce problème.

Le premier bilan est très positif. En migrant sous Maven nous pourrions tout d'abord accélérer le temps de compilation du projet. En effet, avec ANT la compilation complète du projet prend dans les 5mn30. Avec Maven la compilation complète prend autour de 3mn. La compilation après modification d'un fichier seulement est par contre beaucoup plus rapide. Je pense que la gestion des dépendances dans notre projet entre les JAR et les EJB est mieux fait avec Maven.

Autre astuce intéressante: la génération d'un projet IDEA IntelliJ automatiquement. Je rassure mes collègues, utilisateur d'Eclipse, il est aussi possible de générer un projet Eclipse automatiquement. Là encore un gain de temps. **Maven fait gagner du temps** en évitant de laisser les développeurs gérer la plomberie d'un projet, que ce soit la compilation, l'intégration, le packaging, la mise en production sur un serveur de test ou même la mise en place du projet dans Eclipse ou IntelliJ... Cela achève de me convaincre qu'on devrait migrer vers Maven afin de ne plus perdre de temps à gérer tout cela

## Atelier JBoss ON the road et JBoss Seam 1.0.0 CR3

<P>Bonne journée hier. J'étais à Paris au premier atelier "JBoss ON the Road" animé par Luc Texier, Thomas Heute et Thibault Mallet de JBoss Inc. Dans la matinée nous avons entendu un retour sur expérience d'un client de JBoss, Norisbank avec la solution easyCredit qui offre des crédits à la consommation en Allemagne. Le déploiement sur 16 serveurs Solaris de JBoss 4.0.2 peut traiter environ 100 000 requêtes HTTP par jour, avec une capacité de montée en charge grâce à un déploiement simple et efficace. JBoss AS a été retenu pour l'économie réalisée sur les licences. Cependant le coût du support n'a pas été évoqué.</P><P>Luc a ensuite animé un atelier interactif sur JBoss ON la solution de monitoring et de remontée d'alertes pour les produits JBoss mais aussi Apache ou Tomcat. L'outil est une application J2EE utilisant un Agent installé sur chaque machine physique à monitorer. JBoss ON est à la fois simple et puissant. Il permet de suivre les paramètres de la JVM, de la machine et des applications en descendant jusqu'aux transactions sur les EJB ou JDBC par exemple. D'autre part le monitoring d'Hibernate est aussi possible. Bon outil même si le lab s'est déroulé plus ou moins bien (coupure de courant). Dommage que JGroups n'ait pas été utilisé pour que les Agents soient découverts automatiquement par le serveur. Cela me rappelle ce que j'ai fait l'an dernier (<a href="http://www.jroller.com/page/Trecollo/20050628">voir article sur la PDS Console</a></P><P>JBoss ON est un produit à part qui n'est pas disponible en open source. Il est distribué lorsque le client souscrit aux offres de support de JBoss. J'ai demandé si la partie Agent pouvait être redistribuée avec une application commerciale utilisant JBoss AS par exemple. Typiquement en tant qu'éditeur, j'aimerais offrir cet Agent dans l'application afin que nos clients, s'ils désirent activer et installer JBoss ON, n'aient rien à faire de plus. D'après la EULA actuelle cela ne semble pas possible.</P><P><B>Présentation de JBoss Seam</B><B> Thomas Heute est venu nous présenter JBoss Seam. Tout d'abord voir "en vrai" thomas est intéressant, sa présentation était sympa et intéressante. Ce que j'ai retenu de nouveau: <UL><LI>Sortie de JBoss Seam 1.0.0 CR3 fin mai</LI><LI>Sortie de la version finale JBoss Seam 1.0 cet été, peut-être même dès la semaine prochaine à Las Vegas</LI><LI>Ajout du support de l'118N terminé</LI><LI>Ajout de Seam Remoting avec une démo d'une Chat room pour

avoir de l'AJAX dans Seam via JSF. Je vais tester dès que possible cette partie</LI> <LI>Intégration de jBPM afin d'avoir une gestion du workflow pour la navigation et pour la gestion d'état. Pour cela, ajout d'une annotation PROCESS dans Seam (COOOOOOL !!)</LI> <LI>Ajout support client lourd l'année prochaine. C'est une demande de nos clients qui sont des traders financiers avec du Java Web Start.</LI> </UL> Voilà le plein de bonnes nouvelles pour cet été. Si vous voulez télécharger JBoss Seam histoire de passer un bon moment: <a href="http://www.jboss.com/products/list/downloads#seam">http://www.jboss.com/products/list/downloads#seam</a> Voici aussi <a href="http://www.jroller.com/page/Trecollo?entry=premiers\_tests\_de\_jboss\_seam">l'article sur JBoss Seam</a> posté en janvier dernier. </p>

## Installation de JBoss Seam

Pour tester JBoss Seam 1.0.0.RC3 vous devez installer JBoss AS 4.0.4 GA. Sur la page <http://labs.jboss.com/portal/jbossas/download> sélectionnez l'installation via l'installateur en cliquant sur **Run Installer**. Ne téléchargez pas l'archive (via le bouton Download). En effet les EJB3 n'étant pas encore certifiés JEE5, l'ejb3 deployer n'est pas livré par défaut.

Une fois le téléchargement terminé, et après avoir accepté la licence, pensez à choisir l'installation de type EJB3 lorsque le programme d'installation démarre. Il vous faudra aussi Java 5 si vous ne l'avez pas déjà. A noter que si vous téléchargez *jboss-4.0.4.GA-installer.jar*, vous pouvez ensuite patcher votre installation de JBoss 4.0.4 de la manière suivante pour y ajouter le support des EJB3:

```
java -jar jboss-4.0.4.GA-installer.jar -installGroup ejb3
installpath=C:\software\jboss-4.0.4-eb3
```

C'est tout pour aujourd'hui.

## Migrer de JSF 1.2 vers JBoss Seam

<P>Sur cet <a href="http://weblogs.java.net/blog/bleonard/archive/2006/05/trying\_out\_jbos.html" target="new2">article</a> vous trouverez des explications claires et complètes pour intégrer JBoss Seam sur une application utilisant JSF 1.2. L'article comporte une deuxième partie <a href="http://weblogs.java.net/blog/bleonard/archive/2006/05/trying\_out\_jbos\_1.html" target="new2">ici</a>. Seam permet de remplacer les backed-bean de JSF par des ejb3 rapidement et facilement.</P> <P>J'ai commencé hier une maquette en reprenant une application financière qui utilise notre propre framework pour évaluer le coût, la complexité et l'intérêt de passer à Seam. A suivre...</P> <P>Sinon la Suisse joue contre la France mardi soir... clin d'oeil à Luc et Thomas de Jboss inc.</P>

## Red Hat rachète JBoss pour 350 millions de dollars

Et non ce n'est pas un poisson d'avril. Nous sommes le lundi 10 avril et je suis bien réveillé. D'après donc une news tombé chez Reuters (cocorico c'est chez moi) Red Hat annonce son intention de racheter JBoss Inc. pour 350 millions de dollars. 40% de la transaction se fait en cash et 60% en action Red Hat.

Red Hat comme JBoss se finance sur un système qui utilise le support et l'expertise technique. Il semble donc logique et cohérent qu'ils s'associent. JBoss propose de son côté un bon nombre d'API et d'outils comme JBoss Rules 3.0 basé sur Drools, jBPM pour la gestion des workflows ou la gestion des transactions avec l'ex-moniteur transactionnel de la société Arjuna.

Sur le blog de Marc Fleury (CEO de JBoss), [on peut voir celui-ci entrain de signer l'accordement](#).

## JBoss Seam 1.0 GA est disponible depuis le 13 juin

A peine une semaine après le séminaire JBoss, la version 1.0 GA de JBoss Seam est disponible. En fait les archives étaient disponibles dès samedi 11 juin. Suite à cela nous avons vu cette semaine une pluie d'article sur les sites anglophones. Parmi ceux-ci, je vous conseille la lecture d'une [interview de Gavin King](#) le créateur d'Hibernate et de JBoss Seam sur InfoQ complète et intéressante. En deux pages il présente les idées phares de JBoss Seam qui en font un Framework de 3ème génération.

Pour ma part, depuis la semaine passée j'ai avancé sur ma maquette. J'ai intégré les tags JSF de l'api [Jenia.org](#) pour permettre d'avoir accès facilement dans JBoss Seam à JFreeChart. J'en aurai besoin pour la partie charting de notre application financière. Promis dès que j'ai un peu de temps je poste ici les modifications effectuées dans la build pour tester tout cela. Il a fallut notamment changer le build.xml pour pouvoir packager des JAR à la racine de l'EAR et changer le Manifest.mf pour le jar packagé dans le WAR... mais bon c'est du charabia et je prendrai le temps d'expliquer les modifications pour intégrer ces différents tags.

Pour tester JBoss Seam Remoting, en fait il ne m'a fallut qu'une petite heure pour reprendre "helloWorld" et créer un tag inputSearch à la AJAX. C'est vraiment simple. Seam = Simplicité.

La prochaine étape sera de voir comment appeler un web service à partir de la partie Seam. Il y a des spécifications en cours dans EJB3 mais pour l'instant je n'en sais pas trop... A suivre donc

## Comment AJAX-iser ses composants JSF

[Ajax Anywhere](#) est une librairie open source qui permet de transformer des composants JSF existant en composant compatible AJAX. Ainsi au lieu de rafraichir l'ensemble d'une page lorsque l'utilisateur soumet un formulaire, vous pouvez délimiter des zones dans votre page et enchaîner un rafraichissement uniquement de cette zone. L'avantage d'AJAX Anywhere contrairement à d'autres JSF existant, c'est que

cette API n'est pas orientée composant mais plutôt "zone". D'autre part il est possible d'ajouter ou retirer AJAX anywhere facilement car les composants JSF eux-même ne sont pas impactés. Vous pouvez donc ajouter une couche "AJAX" sur votre application JSF existante, et ce, très facilement

#### Mise en oeuvre:

- Téléchargez l'api [ici](#)
- Placer le fichier aa.js à la racine de votre répertoire web, les fichiers .class dans le répertoire WEB-INF/lib, le fichier ajaxanywhere.tld dans le répertoire META-INF/tlds.
- Configurez un nouveau filter:

```
<filter>
  <filter-name>AjaxAnywhere</filter-name>
  <filter-class>org.ajaxanywhere.AAFilter</filter-class>
</filter>

<filter-mapping>
  <filter-name>AjaxAnywhere</filter-name>
  <url-pattern>*.jsp</url-pattern>
</filter-mapping>
<filter-mapping>
  <filter-name>AjaxAnywhere</filter-name>
  <url-pattern>*.do</url-pattern> <!-- default Struts mapping -->
</filter-mapping>
<filter-mapping>
  <filter-name>AjaxAnywhere</filter-name>
  <url-pattern>*.htm</url-pattern> <!-- other frameworks mapping-->
</filter-mapping>
```

Il faut maintenant brancher AjaxAnywhere sur vos composants JSF dans une page JSP. Vous pouvez voir [ici](#) une démo avec les composants MyFaces et l'intérêt de ne pas recharger toute la page. Pour ajouter AjaxAnywhere à une page JSP existante il faut:

- Importer la librairie de tag
- Appeler la methode AAUtil.addZones sur l'élément JSF à ajaxiser
- Importer les librairies JS et définir la FORM qui contient le composant
- Changer la target action de la balise FORM pour appeler du code JS d'AjaxAnywhere
- Pre et post fixer votre composant avec le tag JSF <aa:zone>

Si vous avez déjà une application avec JSF et que vous désirez tester le principe d'AJAX rapidement, alors AjaxAnywhere est pour vous

## Comment configurer l'encoding en UTF-8 dans JBoss

J'ai rencontré un souci avec JBoss et l'UTF-8. En effet, Tomcat par défaut encode le contenu en ISO-8859-1. Si vous développez une application avec par exemple des traductions en chinois, et que vous faites en plus appel à Javascript, vous rencontrerez un problème dans le navigateur. Voici comment s'assurer que l'UTF-8 est bien utilisé dans votre application.

Tout d'abord il faut spécifier l'encoding à utiliser dans chaque page JSP:

Ensuite, reconfigurez Tomcat en éditant le fichier \$JBOSS\_HOME/server/default/jbossweb-tomcat55.sar/service.xml. Pour chaque Connector que vous utilisez (http, https) ajoutez un attribut **URIEncoding="UTF-8"** dans la balise Connector.

```
<Connector port="8080" address="{jboss.bind.address}"
  maxThreads="250" strategy="ms" maxHttpHeaderSize="8192"
  emptySessionPath="true"
  enableLookups="false" redirectPort="8443" acceptCount="100"
  connectionTimeout="20000" disableUploadTimeout="true"
  URIEncoding="UTF-8" />
```

Et voilà c'est tout. Maintenant Tomcat fonctionnera en utilisant l'UTF-8 pour l'encoding des URL. Si vous faites appel à du Javascript avec AJAX pour envoyer des url uencodées vers le serveur, vous verrez alors que cela fonctionne correctement

## String et intern() on ne sait pas tout sur les Strings

Hier soir, plongé dans du code de Xerces-J pour notre parseur XML j'ai vu plusieurs fois ce bout de code:

```
protected final static String A_CONSTANT = "constantTest".intern();
```

Quel est le fonctionnement de la méthode `intern()` de la class `String` ? Comme vous le savez en Java, il y a deux moyens de comparer deux objets. La méthode `equals()` et l'opérateur `==`. La méthode `equals()` s'assure que les deux objets contiennent les mêmes données alors que l'opérateur `==` vérifie si les références de l'objet pointent sur le même objet. D'ailleurs une des premières leçons des débutants en Java est qu'il faut en général utiliser la méthode `equals()` pour comparer 2 `String` en Java. En effet à votre avis quel est le résultat de l'exécution de ce code ?

```
System.out.println("Comparaison: "+ new String("Touilleur")==new String("Touilleur") );
```

Si vous avez répondu `Comparaison: true` vous vous êtes trompé. S'agissant de deux objets distincts, ils ne sont pas égaux. Le code retourne `false`. Voyons maintenant avec l'opérateur `equals` de la class `String`:

```
System.out.println("Comparaison: "+ new String("Touilleur").equals(new String("Touilleur")) );
```

Dans ce cas précis le programme retournera `Comparaison: true` car la méthode `equals` compare les contenus des chaînes. Cependant cette méthode qui itère les caractères peut être très lente. Alors comment Java optimise ce pattern en interne ? N'en déplace à mes amis fan de C/C++ les gens qui codent la virtual machine de Java ne sont pas plus bêtes que d'autres, et bien souvent ils ont aussi de bonnes connaissances en C/C++.

L'opérateur `==` compare l'identité en mémoire des objets, et donc cet appel sera vraiment plus rapide que l'appel de la méthode `equals`. Si comme dans Xerces-J la comparaison de `String` est le cœur de votre moteur, il est indispensable de l'optimiser, les gains en terme de performances seront importants. Pour cela ce qu'il faut c'est une liste de `Strings` à comparer en mémoire dans laquelle nous viendrons piocher lorsque nous devons comparer une `String` à notre liste de référence. D'autre part si la `String` comparée n'existe pas nous voudrions stocker sa référence en mémoire. Avec ce mécanisme il sera possible d'utiliser l'opérateur `==` pour comparer des instances. C'est le rôle de la méthode `intern()` de la class `String` qui gère cette liste et qui nous retourne la référence de la `string` en mémoire.

```
public static void main(String[] args)
{
    System.out.println("Test 1 "+(new String("Touilleur")==new String("Touilleur")));
    System.out.println("Test 2 "+( new String("Touilleur").intern() == (new String("Touilleur").intern())));
}
```

Attention aux parenthèses qui sont importantes ici pour que le test soit valide. L'exécution de ce programme affichera:

```
Test 1 false
Test 2 true
```

La question qui reste en suspend est pourquoi alors une `String` qui est `static` et `final` est instancié avec un appel à `intern()` ? Nous sommes habitués à voir ce code là:

```
public static final String MY_PARAM="params";
```

moi ce qui m'a étonné c'est

```
public static final String MY_PARAM="params".intern();
```

Il se trouve que Java fait en sorte que ce genre de pattern soit déjà optimisé. Toutes les `Strings` d'une class sont par défaut automatiquement "internés" de sorte que vous pouvez avoir quelques optimisations. Le code ci-dessous montre que les `String` qui peuvent être résolues à la compilation seront optimisées. Si par contre vous lancez le programme en passant `test` (`java QuickTest test`) vous verrez que les paramètres dynamiques ne peuvent pas être optimisés et que l'exécution du code retournera `false`.

```
public class QuickTest
{
    public static final String MY_PARAM="test";

    public static void main(String[] args)
    {
        // Est-ce que test est identique a MY_PARAM ?
        System.out.println("test" == MY_PARAM); // VA AFFICHER TRUE

        // Et si nous concaténons la string
        String s2="te"+"st";
        System.out.println( s2 == MY_PARAM); // VA AFFICHER TRUE

        // Si nous spécifions test en tant que premier argument de ce programme
        // afin de voir a l'exécution si nous avons une différence
        String s3=args[0];
        System.out.println(s3 == MY_PARAM); // VA AFFICHER FALSE

        // Maintenant si nous passons par l'appel a intern()
        System.out.println(s3.intern() == MY_PARAM); // VA AFFICHER TRUE
    }
}
```

Le 4eme appel dans le code ci-dessus est donc la raison pour laquelle nous pouvons tout à fait ajouter un call intern() sur une constante en Java. Il faut donc simplement que notre code force un appel à intern() pour pouvoir utiliser l'opérateur ==. D'autre part pour revenir au début de ce post, forcer un appel à intern() sur une constante permet de transformer cette constante résolue à la compilation en constante résolue à l'exécution. (Voir ) bien que finalement ceci est un intérêt assez limité.

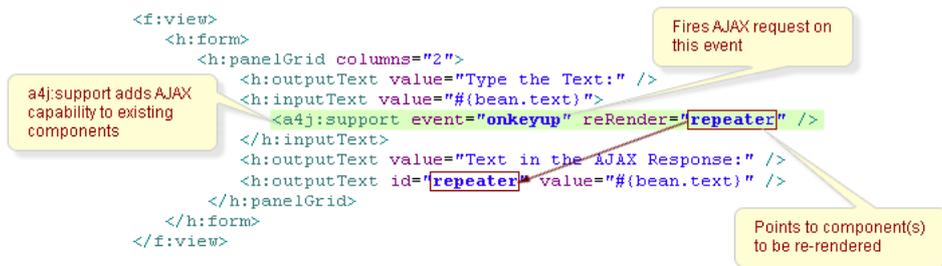
La méthode intern() de la class String est une méthode native. L'implémentation est donc déléguée à la JVM, ce qui fait qu'il y a aussi une différence en terme de performance entre BEA JRockIt et SUN JVM

## Idea IntelliJ 6.0 est disponible

Aaaah grand moment de bonheur ce matin: IDEA IntelliJ 6.0 est enfin disponible ! [Parmis les nombreux nouvelles fonctionnalités](#) on note le support des EJB3, de JBoss Seam, de JSF, du Google Web Toolkit, une meilleure gestion des gestionnaires de code source... etc. Bref si vous utilisez Eclipse, et que vous avez envie d'essayer l'autre éditeur Java et bien je vous encourage à récupérer une version sur [le site de JetBRAINS](#).

## Ajax4jsf 1.0.2 fonctionne avec JBoss Seam

Bonne nouvelle. [Ajax4jsf](#) fonctionne sans aucuns problèmes avec JBoss Seam. Ajax4jsf est une api open source qui offre à JSF (Java Server Faces) des possibilités AJAX sans devoir écrire une ligne de Javascript. Ajax4jsf utilise cependant toutes les fonctionnalités de l'architecture JSF (validation, conversion, gestion du cycle de vie) tout en facilitant la mise en place de composants dynamiques.



Le site Ajax4jsf: <https://ajax4jsf.dev.java.net/nonav/ajax/ajax-jsf/>

## Après CruiseControl et Borland Gauntlet, TeamCity

J'ai installé la semaine passée TeamCity de JetBrains, le moteur d'intégration continue professionnel de JetBrains. Ce nouveau produit permet de faire de l'intégration continue de code Java ou .NET et de suivre via IDEA IntelliJ (l'éditeur) en temps réel si le code de votre produit compile ou non. Tout d'abord l'installation et la configuration de TeamCity se sont déroulées sans problèmes. Par rapport à CruiseControl, le produit est plus simple à installer et ensuite à configurer. Notre code étant sous SVN, j'ai réussi à configurer un projet qui récupère le code Java de SVN puis qui lance notre build Ant sans aucuns soucis. Il faut dire que j'avais déjà écrit des targets ANT spécifique pour CruiseControl afin de permettre à notre logiciel de compiler sans devoir configurer quoique ce soit.

TeamCity gère SVN, CVS, PerfForce et d'autres repository de code source. Au niveau de l'intégration avec IDEA IntelliJ, un plugin permet de poster vers TeamCity son code et d'enchaîner un commit si, et seulement si, le code compile sur TeamCity. C'est le principe de la compilation différée. Le cas d'usage: un vendredi soir vous terminez de corriger 3 fichiers importants. Vous aimeriez partir en week-end l'esprit tranquille et ne pas casser la compilation de votre logiciel. Grâce à Teamcity vous pouvez alors pré-commiter vos fichiers vers TeamCity. Celui-ci va alors lancer la compilation et si celle-ci passe, enverra vers SVN vos 3 fichiers. Si par contre vos modifications venaient à casser la build ou à ne pas être compatible avec d'autres mise à jour, alors le serveur mettra de côté votre commit.

TeamCity est administré via une interface Web déployée sur Tomcat. La licence coûte 249 USD. Cela ne me semble vraiment pas cher. En effet, pour faire tourner CruiseControl correctement il vous faudra pas mal de tests, et prendre le temps de comprendre la configuration de CC via le fichier config.xml. D'un autre côté, TeamCity propose une interface Web très claire et simple qui vous fera gagner du temps.

### Les ressources:

- Le site de JetBrains TeamCity <http://www.jetbrains.com/teamcity/>
- Features et Screenshots <http://www.jetbrains.com/teamcity/features/index.html>

## Plus de développement en .NET qu'en Java

C'est un fait qui est confirmé par des chiffres publiés ici ou là sur le web, les grandes compagnies se tournent de plus en plus vers la plateforme Microsoft .NET au dépend de la plateforme J2EE Java. Dans [cet article](#) de Steve Hamm publié sur BusinessWeeks, nous apprenons tout d'abord que la part de développeur Java par rapport à la part de développeur .NET est passé, selon Evans Data Corp. de 51.4% en 2002 à 47.9% fin 2005. Pendant ce temps, la part des développeurs .NET est passée de 40.3% à 54.1% avec une forte croissance en Amérique du Nord.

En Europe la part des développeurs .NET est majoritaire par rapport à celle de Java (toujours d'après cette enquête). Ok moi aussi j'ai du mal à y croire, surtout en voyant les offres d'emploi en France. Mais saviez-vous que c'est encore une particularité de notre pays ? Il faut voir aussi qu'en France peu d'écoles d'ingénieur ou de faculté offrent de formation Microsoft .NET alors qu'au final, il y aurait du travail pour des juniors.

Ce qui m'inquiète (je peux donner mon avis, je suis chez moi non ? ) c'est que Java est de moins en moins utilisé pour des développements de type Web, au profit de la plateforme Microsoft .NET ou sinon, de ce que l'on appelle LAMP (Linux + Apache + MySQL + PHP). PHP est passé ainsi de 0 site en 2000 à plus de 23 millions en 2006. Le PHP est un langage de script adapté aux petites applications Web simple et rapide. Attention cependant, il faut savoir qu'à fonctionnalité égale, le PHP est beaucoup plus lent que le Java qui lui-même est plus lent que le C++. Si vous voulez en avoir le coeur net [vous verrez sur ce site](#) que certains exemples en PHP sont entre 2 et 122 fois plus lent (les recursives) que Java. Celui-ci est aussi plus lent que le C++, il n'y a pas de miracle.

## Construire une application d'entreprise nouvelle génération avec AJAX en utilisant echo2, EJB et ServiceMix ESB

Le titre de ce post est long mais c'est la traduction d'un témoignage très intéressant d'un architecte sur un projet **nouvelle génération**. L'article [Building an Enterprise Scale Mass Market AJAX Web Site using echo2, EJB3 and ServiceMix ESB](#) présente la réalisation d'une application de recherche de prêt immobilier.

Les choix techniques pour l'architecture de la solution sont:

- o User Interface: echo 2
- o Application Server: JBoss Application Server 4
- o Integration with Mortgage Providers: ServiceMix
- o Persistence Mechanism: MySQL
- o CMS System: Hippo
- o Object Relational mapping: Hibernate 3
- o Dependency Injection: Spring

Pour la couche UI, RDF Group utilise [echo2](#) un framework open-source Ajax très sympa. L'intérêt de ce composant est de proposer un modèle orienté widgets/composants qui est à la fois simple à programmer et puissant. Allez voir les démos echo2 pour vous faire une idée de ce qu'il est possible de faire faire à votre navigateur... Je me sens un peu chez moi puisque le projet que j'encadre est similaire à echo2 ou encore [ZK Ajax](#).

Pour la couche Business Logic, RDF Group a fait le choix d'utiliser [Apache ServiceMix 3.0](#), projet open-source de la communauté Apache. Ce framework propose un bus de données d'entreprise (ESB) pour s'intégrer dans une architecture SOA. Il est construit à la base sur la spécification JBI JSR208 qui est dit comme étant l'implémentation Java de référence pour l'orchestration de services. J'ai parlé il y a un temps de SCA, la proposition des gros éditeurs (BEA, Sybase, Oracle...) et je ne sais plus si j'ai parlé de Microsoft Indigo, l'architecture proposée par Microsoft qui est aussi intéressante.

**ServiceMix** utilise Spring et intègre des API très intéressante comme le moteur de règles [Drools](#). Ce moteur de règle devrait donc permettre de proposer des services d'évaluation de Business Object vraiment intéressant.

La couche de Binding propose les protocoles suivants:

- o HTTP
- o JMS
- o JavaMail
- o WSIF
- o Jabber (instant messaging)

La couche Services package les couches métiers SOAP suivantes:

- o WS-Notification
- o WS-Reliable Messaging
- o WS-Eventing
- o WS-Resources Properties
- o WS-Distributed Management

Quant à la couche Components voici ce que l'on y trouve:

- o ActiveMQ pour la fabrication de message
- o ActiveSOAP
- o Mule (Object Brooker)
- o Groovy pr le scripting

- Drools (moteur de regles)
- PXE (BPEL)
- Agila (Workflow)

Concernant les serveurs J2EE, ServiceMix s'integre avec Apache Geronimo facilement. Il y a aussi [le support de JBoss](#) et donc la possibilité de construire facilement un bus de données.

Pour vous faire une idée de ServiceMix je vous conseille la lecture [du UserGuide sur leur Wiki](#).

Dès que j'ai un peu de temps je testerai ServiceMix et je ne manquerai pas de revenir ici autour d'un café en parler avec vous.

## JBoss Seam: l'intérêt de la conversation

Je vous propose en 10 minutes de vous expliquer un principe puissant et simple de JBoss Seam. Ouvrez l'url suivante <http://seam.demo.jboss.com> et suivez mes explications.

Tout d'abord cliquez sur le lien "**Register New User**" pour créer un compte. Remplissez le formulaire et cliquez sur Register. Vous revenez ensuite à la page d'accueil, sur laquelle vous pouvez désormais vous authentifier. Après avoir cliqué sur "Account Login" la page principale est affichée. La section "Search Hotel" permet de rechercher un hôtel puis d'effectuer une réservation. La section "Current Hotel Bookings" est initialement vide. Allons-y, nous allons commencer notre première conversation JBoss Seam.

Tapez Hotel puis cliquez sur le bouton "Find Hotel". Une liste d'hotel s'affiche.

Prenons Hotel Rouge à Washington DC, nous allons voir les détails en cliquant sur le lien View Hotel. A cet instant vous entrez dans une conversation. La page suivante affiche les détails de l'hôtel et je peux enfin parler de ce qui va m'intéresser dans un instant: **la section workspace**. Cette section vous montre une session en cours que vous avez initialisée.

**Ouvrez une autre fenêtre de navigateur et loguez-vous à nouveau.** Que remarquez vous ? **le résultat de la recherche est affiché sur la nouvelle session.**

JBoss Seam vous redonne l'état qu'il a persisté dans une autre fenêtre. Maintenant si vous avez bien suivi, votre premier navigateur doit être à la page affichant les détails de 'Hotel Rouge' et la deuxième fenêtre affiche la liste des hôtels dont le nom contient "Hotel". Sur la deuxième fenêtre, cliquez sur le premier Hotel de la liste. Vous pouvez alors voir les détails de cet hôtel mais surtout, et c'est là que j'insiste, vous pouvez voir dans la partie "Workspace"

### Workspaces

```
View hotel: W Hotel [current] 09:43 -09:43
View hotel: Hotel Rouge      09:43 -09:43
```

En cliquant sur un lien vous passerez d'une conversation à l'autre. Imaginez ce concept dans une application de site marchand. Vous pourriez par exemple commencer vos achats dans une fenêtre, puis dans l'autre regarder les rayons DVD pour comparer des articles. Les Workspaces sont des espaces de conversation. Une conversation au sens JBoss Seam est en fait ce que nous faisons à chaque fois que nous achetons un livre sur Amazon ou que nous consultons un site Boursier. C'est un enchaînement d'écran avec différents points de sortie. Sur la démo JBoss Seam vous pouvez terminer une conversation en effectuant une réservation. Vous pouvez aussi annuler votre transaction ou commencer une autre réservation sur un autre onglet de votre navigateur. **N'est-ce pas tout simplement génial ?** Vyons autre chose: le site internet Opendo.co.uk utilise **Struts comme le prouve ce lien**. Lorsque vous effectuez une réservation, si vous ouvrez une autre fenêtre de navigateur, chaque navigateur est un nouveau client pour l'application. A noter que cette application fonctionne plutôt bien. Vous pouvez essayer d'utiliser le bouton Back, de cliquer plusieurs fois sur un bouton, il semble qu'elle tienne bien le coup. Simplement j'imagine le travail derrière pour que la gestion du bouton Back ait été correctement effectuée.

J'espère vous avoir convaincu après avoir testé la démo BookingHotel de JBoss Seam:

## Conférence QCon à Londres

Un petit post pour vous parler de la conférence **QCon** qui aura lieu du 12 au 16 mars 2007 à Londres. Cette conférence est organisée par le site [InfoQ.com](http://InfoQ.com) dont les articles sont en général d'excellente qualité. La conférence est divisée en 2 jours de tutoriaux et cours pratiques et 3 jours de conférence "à la JavaOne". Vu le désert d'événement pour l'instant sur Paris en ce moment, c'est peut-être l'occasion d'aller y faire un tour pour se tenir au courant de ce qui se fait dans le monde Java, .NET, SOA ou les méthodes Agile.

Parmi les orateurs de la section Java, A.Popescu l'auteur de TestNG, G.Laforge un des auteurs de Groovy, Gavin King l'auteur d'Hibernate et de JBoss Seam, Martin Fowler une des grandes figures des Design Patterns, Rob Harrop le lead developer du framework Spring ainsi que Rod Johnson, bref que du beau monde.

Il y a 4 groupes de conférences: .Net, Java, Architecture et Software Usability. Les conférences commencent le mercredi mais vous pouvez aussi assister le lundi et le mardi à des tutoriaux vraiment intéressants. Quelques conférences sur les méthodes de gestion de projet Agile, ou d'initiation à EJB3 peuvent servir d'initiation pour les béotiens qui veulent avoir tout de suite des présentations de qualité.

Le mercredi commence la partie Conference avec un ensemble d'orateur de qualité. Vu le nombre de conférence il faudrait y aller à 2 voir 3 personnes et se répartir les participations aux différentes présentations.

J'espere (secretement) qu'un de mes managers lira ces quelques lignes et se dira, tient il est temps d'investir un peu de temps et d'aller voir ce qu'il s'y passe... en m'emmenant là-bas. Cela me

rappellera JavaOne en 2000 (soupir... merci encore à Guillaume et David pour ce cadeau).

## JSR-296 Swing Application Framework

La spec JSR-296 est un ensemble de class qui vise à simplifier le développement d'applications Swing. Encore à l'état de prototype, une premiere version est disponible sur <https://appframework.dev.java.net/>.

Cette spécification définit un socle commun que toute application Swing utilise comme:

- o Gestion du cycle de vie, démarrage et arrête de l'application
- o Support pour la gestion des ressources comme les images, les ressources Bundles pour l'internationalisation, les polices et tout autre élément de configuration.
- o Support pour la définition, la gestion et l'enregistrement des Swing Actions, en incluant les actions qui doivent fonctionner de manière asynchrone.
- o Persistance de la session, enregistrement des préférences, des positions des fenêtres automatiquement.

Cette spécification n'est pas encore à l'état de "Early Draft" dans le processus de revue JCP. Mais cependant elle pourra intéresser ceux qui développent des applications Swing. A voir.

## Google a de belles courbes, comme Paris Hilton

Les salariées de Google ont la chance de ne rien faire comme les autres. Dans la série un jour mon fils aura une tétine Google, le site [ZorGloob.com](http://ZorGloob.com) présente quelques nouveautés à suivre, surtout lorsque l'on voit le succès de Gmail et Picasa. Gmail devrait d'ailleurs être rebaptisé prochainement *Google Mail* suite à un problème de droit avec G-Mail, société allemande de publipostage.

Parmi les liens les plus sympas à regarder, la liste des recherches les plus populaires sur Google par pays et par mois. On appelle cela **Zeitgeist**. [Les Francophones](#) sont plutôt branchés Noël, Darty, Nintendo Wii et SFR en décembre. Ouf on a évité "Paris Hilton nude" et autres blondes à petite poitrine (la fameuse paris précédemment citée).

Pour terminer sur des histoires de belle courbe, en voici une qui en dit long et qui fait réfléchir. D'après [cette courbe](#) la société Google emploie 10670 employés dans le monde au 31 décembre 2006. La société Reuters fondée en 1870 emploie environ 13500 personnes dans le monde... Peut-être qu'un jour Google s'offrira un groupe comme Reuters. Après tout, nous avons plus de bureaux dans chaque pays qu'il n'y a de pays à l'ONU... Google couvrirait alors toute la Terre, Google everywhere... Je l'aurai mon teeshirt Google, je l'aurai...

## JBossESB 4.0 est sorti

### JBoss annonce la sortie d'une solution ESB Open-Source: JBossESB 4.0

Annoncé depuis quelques temps, JBossESB 4.0 est sorti. Suite au rachat de l'ESB Rosetta au Canada en juin dernier, JBoss a maintenant une solution d'ESB qui est déjà en production dans un groupe d'assurance au Canada.

D'un point de vue technique, au premier coup d'oeil rapide j'ai l'impression qu'Apache ServiceMix couvre plus de domaines techniques que JBoss. Cependant d'un point de vue architecture et clustering, à mon avis la solution de JBoss basée sur JEMS doit être plus performante. C'est complètement subjectif, je sais :-)

Si vous vous demandez ce qu'est SOA [ce PDF](#) vous donnera quelques réponses. Pour comprendre ce qu'est un ESB, je vous renvoie à [l'excellente présentation de ServiceMix par Mark Richard \(InfoQ\)](#). Cela dure une bonne heure en vidéo mais franchement cela vaut le coup, surtout si vous êtes architecte.

JBossESB 4.0 propose les fonctions suivantes:

- o Support pour un système de notification utilisant entre autre JMS, email, la base de données ou le système de fichier. A priori donc, pas de TCP, ou de protocole propriétaire.
- o Transformation des données (rôle du médiateur) avec Smooks ou XSLT.
- o Système de listener et d'action model pour découpler les acteurs du modèle
- o Routage basé sur Drools (JBoss Rules) ou sinon XPatch en analysant le contenu
- o Support pour une table de registry avec JAX-R et jUDDI (implémentation open-source des annuaires UDDI)
- o Passerelles pour injecter du trafic non-ESB dans l'ESB, ce sont les connecteurs.
- o Haute-performance et tolérance aux pannes, en production dans la 2eme compagnie d'assurance au Canada depuis 3 ans.

JBoss complète son architecture orientée services et propose maintenant un ESB basé sur le coeur de JBoss, JEMS. Sachant aussi la qualité de cette architecture, il n'y a aucuns doutes que cette solution est à envisager si vous cherchez un moteur ESB pour votre architecture.

[Home-page JBossESB](#)

## Eclipse est mort, vive Rational Jazz

Avec un titre comme celui-ci, je suis certain que la majorité des lecteurs vont se dire que j'ai encore fondu un plomb et que dans 3 lignes je vais vous parler d'[IDEA IntelliJ](#)... perdu ! Non ce matin j'ai envie de parler du projet Rational Jazz sur lequel travaille Erich Gamma, l'un des auteurs d'Eclipse. Eclipse ne va pas être abandonné, loin de là.

Voyons juste ce qu'IBM prépare pour notre environnement de travail dans 2 ou 3 ans.

Tout d'abord, connaissez-vous **Erich Gamma** ? Erich est l'un des auteurs d'Eclipse, l'un des co-auteurs du livre [Design patterns. Catalogue des modèles de conception réutilisables](#) et enfin l'un des co-auteurs du framework de test unitaires JUnit. Autant dire donc que sans le savoir vous avez certainement déjà utilisé l'un des outils auquel il a contribué ou dont il est l'auteur.

Erich Gamma travaille sur un nouveau projet en collaboration avec IBM, **Rational Jazz**. Il s'agit d'un nouvel outil et d'une nouvelle plateforme de développement collaboratif. Fortement orienté vers un travail en réseau, Jazz sera à la fois un éditeur mais aussi un environnement de travail afin qu'un développeur à Bangkok puisse travailler avec un développeur à Paris ou à Palo-Alto. Il y aura à coup sûr donc des outils comme un serveur central de code source (CVS,SVN), un système centralisé de gestion d'artefacts (comme CollabNet, SourceForge.net, Jira). Je pense qu'il y aura aussi un système de Wiki pour documenter (comme [Confluence](#)) et échanger des documents ([à la Microsoft SharePoint](#)). Que manque-t-il ? Sans doute un serveur d'intégration continue ([CruiseControl](#) ou mieux, [TeamCity](#) de JetBrains.). Quoi d'autre ? J'y ajouterai avec un peu de Websphere un environnement pour mettre en ligne automatiquement les applications J2EE compilées afin d'avoir en permanence une démo qui tourne par exemple...

Rational Jazz ce sera donc un peu tout cela: faire jouer ensemble des musiciens complètement différent, pouvoir s'adapter au changement facilement tout en restant simple et puissant. Le message fort derrière ce nouveau projet est qu'avec des équipes de développement localisées aux 4 coins du monde, des outils modernes et puissants sont nécessaires afin d'améliorer la productivité et la qualité. En disant cela je pense très fort à un produit chez nous, hop voilà c'est fini... vive les pays de l'Est.

Pour résumer, Jazz Rational est un nouvel outil de développement collaboratif qui ne sortira que dans quelques mois. Il proposera une réelle innovation pour s'adapter aux nouvelles méthodes de travail en équipe à travers Internet qui n'existaient pas il y a 6 ou 7 ans.

Pour terminer quelques liens:

- o Article [sur eWeeks](#)
- o [Le site Rational Jazz](#)
- o La conférence [eclipseCon 2007](#) à Santa-Clara, CA, du 5 au 8 mars 2007.
- o [Témoignage de Steve O'Grady sur une démo de Jazz](#)

## Nouvelles de DotVision

Quelques nouvelles de [Dotvision](#). J'ai vu leur participation à la célébration au Sénat le 13 février 2007 dernier de IDEES, le programme d'aide aux start-up de Microsoft. Extrait: *Même constat chez Dotvision, où le support technique de Microsoft est particulièrement apprécié. La société était au départ spécialisée dans les développements en langage Java. « Elle est depuis passée aussi au C# de Microsoft, mais même sans l'accord, nous aurions fait ce choix », nous explique Guillaume Pelletier, son dirigeant, « Aujourd'hui, l'activité se partage entre les deux langages ».* (Extrait [de cet article](#))

A part JC, Djam et Tanguy, plus aucunes nouvelles des autres DotVision-boys. Sur ce, bon weekend.

## Viadeo (ex-Viaduc) ça y est j'en suis

Je me suis inscrit sur Viadeo. J'étais un peu réfractaire à utiliser Viadeo, ex-Viaduc, car les français en font une utilisation un peu curieuse par rapport aux américains. Je vais vous expliquer cela.

Je me suis référencé sur [LinkedIn](#) il y a presque 4 ans. Yan Pujante, l'un des responsables techniques de LinkedIn m'a formé lorsque je suis arrivé chez Reuters à Palo-Alto. Grâce à lui ensuite, lorsqu'il a quitté Reuters pour fonder LinkedIn, j'ai eu la chance de découvrir cet outil vraiment intéressant.

Yan expliquait que les personnes que vous mettez en tant que contact doivent, pour que votre réseau soit de qualité, respecter quelques critères. Tout d'abord connectez-vous uniquement avec des personnes que vous avez rencontré physiquement. Ensuite, idéalement vous devez référencer les personnes que vous seriez prêt à recommander par exemple pour un job. Typiquement je n'ai pas ajouté deux ou trois personnes de notre ancienne équipe (john n. si tu lis cela, tu comprendras) non pas parce que j'aime/j'aime pas la personne, mais parce que je ne me sens pas capable de la recommander. La qualité d'un réseau est plus important que sa taille. Un utilisateur avec 300 connexions,est-ce réaliste ? Il passe sa journée à mettre en relation ses contacts ? A-t-il le temps

de faire suivre ou de compléter les demandes de mise en relation ? J'en doute.

Viadeo en bon site franco-français, vous place en premier sur la partie formation/diplôme pour remplir votre profil. LinkedIn vous invite tout d'abord à détailler votre poste actuel, puis ensuite les entreprises où vous avez déjà travaillé. A la fin seulement, la partie formation vous permet de renseigner vos diplômes. J'ai envie de dire à Monsieur Viadeo que la formation quoique importante, ne fait pas tout. Et en tant que recruteur moi-même, je regarde d'abord l'expérience et le parcours professionnel avant de regarder la formation. En informatique il est illusoire de croire que les écoles d'ingénieurs ou l'université en France forme des personnes prêtes à travailler. Dur pour un étudiant qui lit ces quelques lignes, mais c'est vrai. Java et .NET sont deux technologies qui vous donne un travail et un salaire correct (> 30 KEUR sur Paris pr un débutant).

Voilà monsieur Viadeo, j'ai craqué et je vais remplir mon profil mais je vous préviens, si je trouve que votre réseau n'est pas top, bye bye...

## Salon Plein Emploi, ma journée vu du côté recruteur

Nous étions aujourd'hui au CNIT à la Défense à Paris au salon [Plein Emploi](#) pour recruter des ingénieurs de développement C++ et Java. D'ailleurs si vous cherchez un poste chez un éditeur de logiciel, [je fais un peu de pub pour la maison](#) au passage.

Pour la première fois je suis assis de l'autre côté du stand, à accueillir les candidats, regarder les CV, répondre à des questions techniques ou sur notre activité. Une grande partie des exposants est constituée de SSII et de cabinet de consultant. Finalement peu d'éditeur présent. En faisant le tour des stands pour regarder les offres des SSII, nous nous sommes fait aussi attraper à coup de fraise Haribo, de bonbons, de carte de visite ou de pseudo-questionnaire histoire d'attirer les candidats sur les stands. Une grosse société de service dont le nom commence par A m'a même tenu la patte un bon moment, tellement ils cherchent des candidats. Les sociétés de service ont visiblement l'habitude et la culture de recrutement adéquate pour ce genre de salon. De notre côté, nous avons rencontré des candidats intéressants, mais prévoir la prochaine fois un stock de goodies pour attirer plus les visiteurs. Autre remarque: j'ai vu que les panneaux des sociétés de consultant mettent en avant les technologies (J2EE, .Net, C++, Hibernate, JSF chez nos voisins en face) ce qui forcément attire l'oeil des candidats. Il faudra peut-être que l'on mette un peu plus en avant les technologies que nous utilisons.

Du côté candidat, nous avons vu passer un CV photocopié de travers, un candidat qui nous a un peu agressé car nous ne faisons pas de CSM SAP Je-Ne-Sais-Quoi (désolé mon vieux), un autre qui nous a sauté dessus en disant "vous cherchez des informaticiens ?" (dans ma tête: *non on est là pour faire beau et on va aller au salon infirmière à côté...*) ou d'autres trop timide qui s'enfuyaient si je m'approchais d'eux... "Hé les gars, votre boulot on va pas vous le mettre dans la bouche non plus hein ?". Intéressant comme expérience.

Voilà voilà

## SUN Tech Days, la pilule bleu qui vous veut du bien

**Compte-rendu de notre journée de mardi dernier aux [SUN Tech Days 2007](#) à Paris. Allez-vous prendre la pilule **bleu** de SUN, le progrès et la lumière ou la pilule **rouge**, placebo qui vous ne donnera aucunes informations ? Aucunes nouveautés techniques ?**

8h20... devant la FNAC au CNIT... personne. Pourtant la veille j'ai donné RDV à 3 de mes collègues pour aller passer notre journée ensemble aux SUN TechDays la conférence technique de SUN sur Java. Vérification faite (merci superman) il fallait aller à l'Esplanade de la Défense. Bref me voilà arrivé avec 2mn de retard s'il vous plait, score honorable lorsque l'on sait qu'en fait je suis arrivé le premier car les autres ont été retardés par les grèves dans le métro. Bref passons et parlons maintenant de ce qui vous intéresse plus: **qu'y-avait-il à voir ou à écouter cette année au SUN Tech Days ?**

La matinée commence par une introduction d'Eric Mahé, suivit d'une allocution de Bruno Houdel, le Mr Marketing de SUN et enfin, James Gosling himself. Clairement James a mal tourné. Je l'ai vu en 2000 à San-Francisco, ses présentations étaient bof mais le fond de ce qu'il dit est intéressant. Rebelote en 2003, mais là encore après une présentation pâlotte (franchement c'est un mauvais orateur) il nous avait finalement intéressé sur le fond. Ouf je suis rassuré, en 2007 il est toujours aussi soporifique mais le fond de ce qu'il dit est toujours aussi intéressant. Que faut-il retenir ?

James Gosling a d'abord parlé de la situation de Java sur le marché et dans l'industrie aujourd'hui. Quelques exemples : Java est utilisé sur l'Airbus A380 pour l'embarqué, plus d'1.5 milliards de téléphone portable dans le monde avec une JVM. La déclaration d'impôt en 2006, encore du Java avec JBoss Application Server, aucuns problèmes de performances, n'en déplaise à certains détracteurs des solutions open-source. Java est partout.

Il a ensuite abordé le thème de « Java fonctionne partout » qui bien qu'encore d'actualité, se transforme pour les développeurs en « *Learn once, develop anywhere* » ce qui pourrait se traduire par « apprenez sur une plateforme et développez pour différentes plateformes ». Dans la série les petites phrases qui font sourire : Java 5.0 est sorti, les gens n'ont pas besoin d'attendre une hypothétique version 5.1 pour passer en production. La qualité et la puissance

de Java fait qu'il n'y a jamais eu besoin de version 5.1 ni de service pack. Par ailleurs la sécurité dans Java a été abordée dès la phase d'architecture, ce qui fait qu'aujourd'hui il n'y a pas de virus sur java, il n'y a pas d'éditeurs d'anti-virus pour Java. Pour information Microsoft Vista est déjà attaqué de toute part, et non Microsoft n'a pas réussi à sortir un système infaillible. *PatchGuard* dans Vista semble déjà avoir quelques petits soucis, je vous propose des liens pour en savoir plus : <http://www.eweek.com/article2/0,1895,2036585,00.asp> et aussi <http://www.eweek.com/article2/0,1895,2029031,00.asp> .

Parlons performance. James Gosling commence un slide par "**Java is slow because it's interpreted**". Mais comme il l'explique ensuite une ligne en dessous, il est lent la première fois que le code est examiné (et encore) mais il faut comprendre ensuite que le bytecode n'est pas interprété à chaque fois. Le PHP, Perl ou Python sont interprétés à chaque fois, pas Java ! Ouvrez les yeux il est temps de changer de point de vue.

Pour revenir sur un plan plus neutre, des tests ont été fait pour quantifier les progrès entre Java 1.4 et Java 5 et aussi démontrer l'avancé en terme de performance de Java. **Le code n'est pas seulement interprété, il est en plus optimisé selon la plateforme et l'architecture.** La loi de Moore montre que les fréquences des CPU augmentent tous les 18 mois. Cependant depuis fin 2003 elle tend à stagner et aujourd'hui ce n'est plus la fréquence CPU qui fait la différence mais le nombre de core. Or Java est plus facile à optimiser sur une architecture multi-core 32 bits ou 64 bits et surtout, surtout, la HotSpot de SUN optimise mieux ce que ce pourrait faire un développeur très calé en C++ avec des paramètres de compilation. Pourquoi ?

Tout simplement parce que la HotSpot a été testée sur un grand nombre d'architecture et elle-même optimisée par les meilleurs développeurs C++ et C.

**La compilation dynamique optimisée est plus performante que la compilation statique.** Et avec l'arrivée de multi-core cela va encore plus être visible de manière flagrante. C'est un message fort qu'il faut comprendre pour accepter l'idée que Java est finalement en 2007 un langage rapide.

Dur non à entendre ? Ca va vous lisez toujours ? On fait une pause ?

Allez on se regarde quelques benchmarks pour se rassurer un peu :

<http://www.shudo.net/jit/perf/> et plus particulièrement des résultats comme le SciMark 2004 où la SUN Hotspot Java 1.5 bat le langage C (gcc 3.2.4) ce qui me laisse penser que finalement, on arrive enfin en 2007 à faire quelque chose qui tourne.

James présente ensuite un étude qui montre que le garbage collector optimisé d'HotSpot est plus rapide que des opérations de malloc/free programmées par un développeur.

Ceci s'explique car le GC est optimisé et surtout s'adapte au programme dynamiquement. La question que l'on se pose est: comment peut-on envisager d'atteindre le même type d'optimisation avec un langage statique si la structure ou les paramètres de compilation n'ont pas été sélectionnés correctement ? Bref vous l'aurez compris, James Gosling a voulu faire prendre conscience qu'on est en 2007 et qu'à ce titre les bonnes vieilles croyances devraient elle-aussi évoluer.

Nous avons ensuite eu 6 présentations rapides de quelques points forts de Java 5 et de Java EE5. Une présentation de Romain GUY sur Swing nous a montré qu'en Java, moyennant quelques efforts, il était possible d'avoir des applications de type client lourd qui « déchirent » car je ne vois pas d'autres mots. Romain a codé sous nos yeux en 5mn une application swing utilisant GoogleMap et Flickr, c'était assez impressionnant.

Ensuite la présentation de Java **Derby** m'est passé complètement au dessus, j'avoue que je n'ai pas tout compris. J'irai me documenter sur Derby pour comprendre. Ensuite **GlassFish** et JEE5 avec bien entendu JSF, présenté avec jMaki, c'est un peu notre domaine. J'ai bu du petit lait, c'était bien présenté et la présentatrice nous a codé un magasin de ventes de fleur en 3 clicks.

Mon impression est que **jMaki** ressemble au Google Web Toolkit. Du code Java placé dans des pages génère du javascript cross-browser et facilite la vie du développeur. Le javascript justement. La présentation sur **Phobos** et Rhino nous a montré que le Javascript pouvait être intégré du côté serveur pour écrire des pages dynamiques sans recompiler. Je n'ai pas compris si nous avions à faire à un clone d'ASP par exemple. Cela dit, je vais être honnête : je n'aime pas le javascript et je n'envisage pas de m'en servir du côté serveur.

La démo que nous avons vu consiste à créer une page de type JSP, y mettre du code Javascript et donc sans recompiler, voir en direct live les changements dans le navigateur. Tout d'abord concernant la syntaxe Javascript, à part IDEA IntelliJ qui s'en sort, les IDE Java ne gèrent pas correctement le JS.

Ensuite, comment déboguer ? Comment construire des composants riches comme dans JSF ? Non, il faut que le mouvement soit Java à 100% avec JSF et que le JavaScript reparte dans le navigateur, où il fait très bien son boulot. Je n'en veux pas sur mon code, ce n'est pas réaliste.

J'ai assisté ensuite à une présentation sur le serveur d'application de SUN, gratuit et open-source, à savoir **SUN JSAS** mais que tout le monde appelle « **Glassfish** » du nom d'un poisson tropical transparent. JSF 1.2 en standard, JAXB 2.0 pour le binding, JSP2.1 et aussi JAX-WS 2.1 pour la couche web service. Il ressort que Java EE 5 est la version à utiliser aujourd'hui. Parmi le message passé par l'orateur, la mort de J2EE 1.4.

**J2EE 1.4 est mort, ne commencez pas de projet avec cette architecture. Vous allez écrire beaucoup de code, vous allez mal l'écrire alors vous allez prendre des patterns. Ensuite vous serez déçu, donc vous allez retirer les EJB2.1 et mettre Spring. Et finalement on vous dit aussi que les Session Stateful Bean c'est lent, alors vous envoyer dans la couche http la gestion de la session (au passage, une servlet ou un SessionStatefullBean ça fait pas la même chose?). Enfin votre projet meurt, obèse, lent et incapable de monter en charge car le transfert des sessions http explose votre réseau.**  
Voilà vous me devez 100 EUR j'ai prédit l'avenir de votre logiciel.

Sinon vous pouvez aussi prendre le temps de lire un peu ce que **JEE5** propose. Tiens des EJB3 avec des annotations, on passe de 9 fichiers XML sur la démo PetStore à un seul. Tiens un moteur de mapping puissant dans Glassfish, Oracle TopLink. En fait Oracle a donné à la communauté Glassfish le moteur de mapping TopLink. Tout comme JDBC est une spécification pour qu'ensuite les vendeurs proposent une implémentation, **JPA** (Java Persistence Api) est aussi une spécification qui doit permettre dans une solution JEE5 de choisir TopLink ou Hibernate.

Chez nous un architecte à deux balles a écrit dans des documents qu'il ne fallait pas utiliser JPA mais Hybernate (*notez au passage le y dans Hybernate, on sens qu'on a tiré un champion là*).

Rigolez-pas, le gars semble y croire, alors on le laisse dire et on se marre bien en attendant.

Pour revenir à GlassFish, franchement je vais tester. Je suis un gros fan de JBoss mais là en voyant l'intégration de JSF dans NetBeans 5.5 avec un serveur Glassfish, je me dis qu'il faut absolument regarder ce qu'il y a dedans. GlassFish v2 est le premier serveur qui sera certifié Java 5, la version v2 finale est prévue pour le mois d'août.

Il y a aussi une partie ESB, JBI avec OpenESB et l'authentification d'entreprise avec OpenSSO. La partie clustering et réplication semble aussi très prometteuse. Pour la couche base de données il est possible de prendre des solutions comme SUN HADB qui garantissent 99.999% de disponibilité. Oui avec 5 fois le chiffre 9, et le tout certifié par un organisme ISO s'il vous plait.

Contrairement à JBoss qui propose maintenant une console d'admin payante, Glassfish dispose déjà d'une interface d'administration qui franchement est mieux que celle de BEA WebLogic 8.2. Quoi d'autre ? pour JSF j'ai noté aussi un projet qui regroupera les composants JSF qui ont été « ajaxisés ». Cette perle (pour nous) est le projet **Woodstock** (<https://woodstock.dev.java.net/Preview.htm>). Site à aller voir absolument si vous faites du JSF.

Je suis ensuite allé voir [la présentation de Romain GUY sur Swing](#). Je vous fais un résumé rapide : ça déchire. Ce que Romain a présenté est franchement impressionnant et c'est assurément une réponse possible à Microsoft Expression et à la couche Windows Foundation Classes de Vista. Le site de Romain donne plus de démos (<http://www.curious-creature.org>) et mérite le détour.

Avec Java Web Start il est possible de tester quelques unes des réalisations de l'équipe d'architecture de Swing.

**Troisième présentation sur NetBeans 5.5** très intéressante. NetBeans c'est un IDE, une plateforme comme Eclipse RCP et enfin une vaste communauté open-source. Je retiendrai que Matisse, l'outil de développement des interfaces Swing de Netbeans est très puissant, simple à utiliser. Le Profiler de NetBeans qui est un plugin à installer, est franchement un concurrent sérieux de JProfiler. S'en est suivi une présentation de Ruby. J'aime bien ce langage mais franchement, pour notre domaine de développement, c'est en dehors du sujet.

Une info intéressante pour vous : l'index TIOBE donne **une indication à relativiser de la popularité d'un langage** : <http://www.tiobe.com/tpci.htm>. Ruby est celui qui a le plus progressé en 2006. Le Javascript aussi (merci Ajax). Le Java semble redescendre au profit de nouveaux langages comme le Python, Ruby.

Dernière présentation à laquelle j'ai assisté, **JSF Visual Pack**. Je connais très bien JSF (java server faces) et je voulais surtout que les développeurs avec moi aient une présentation en 1h de ce qu'est JSF et comment le mettre en oeuvre. **Déception**. L'orateur visiblement stressé nous a noyés dans ses slides, et je le regrette fortement. Franchement c'était lourd, et mal présenté. Très déçu par cette présentation.

Enfin sinon j'ai terminé par un tour des exposants. Le stand d'Adobe est celui qui a le plus retenu mon attention. Nous avons assisté à une présentation de **Flex**. Le SDK est gratuit pour compiler ses propres fichiers MXML. Le studio de création des écrans est payant (479\$ je crois) mais il vaut le coup. Si vous acceptez Flash sur le poste client, au lieu de se prendre la tête avec du Javascript et du DHTML, prenez du Flex. C'est simple à écrire, le résultat et l'ergonomie sont vraiment impressionnants.

Le site d'Adobe sur Flex se trouve <http://www.adobe.com/products/flex/> et le site de la communauté open-source est <http://www.flex.org>.

Enfin un exemple de site réalisé avec Flex :

<http://examples.adobe.com/flex2/inproduct/sdk/flexstore/flexstore.html>

Pour conclure, cette journée nous a donné l'occasion de voir rapidement l'avancé de Java et de Java EE5. Mieux vaut passer quelques heures à écouter une présentation technique d'un évangéliste de SUN que d'attendre la sortie d'un livre et finalement ne pas bénéficier du recul des autres développeurs qui ont déjà essayé les plâtres des nouvelles technologies.

**Java 5 est rapide, l'architecture J2EE 1.4 est morte et enterrée, NetBeans 5.5 mérite le détour, les EJB2.1 sont un lointain souvenir, nous sommes en 2007 et vous venez de prendre la pilule bleu.**

## Flex2, premiers pas...

Suite au SUN TechDays j'ai téléchargé et installé Flex Builder. Flex est une solution puissante développée au départ par Macromédia en 2004, puis maintenant par Adobe suite au rachat de Macromédia. Flex permet de construire des applications internet riches multi-plateforme car il utilise Flash 9. La majorité des navigateurs ont maintenant Flash installé ou avec l'explosion des sites comme YouTube ou DailyMotion qui font appel à cette technologie, il est clair que l'installation d'un plugin n'est plus un problème.

Une application Flex s'écrit avec du XML, plus précisément du MXML. Il suffit de décrire la composition de l'application, et de déclarer ce que voit en fait l'utilisateur. Il est aussi très facile de créer des composants ou des vues réutilisables. MXML me fait penser à Xaml de Microsoft ou encore à la syntaxe XUL de Mozilla. Cependant il y a aussi ActionScript, un langage de script qui vous permet de brancher des événements, d'appeler des Web Services ou des applications de type REST via HTTP.

J'ai bossé tard hier soir après avoir installé le Flex Builder basé sur Eclipse RCP. Je me suis lancé sur l'écriture de quelques écrans standards de nos applications utilisant notre framework Java "pour voir". Et franchement le résultat est très sympa. En terme d'ergonomie, on est loin devant AJAX, HTML et CSS. Nous utilisons même des composants HTC de Microsoft pour faire des tableaux éditables ou le support du drag-and-drop. Avec Flex tout est plus facile, simple et propre. Et évidemment je peux faire tourner mon application dans Firefox, comme dans Internet Explorer.

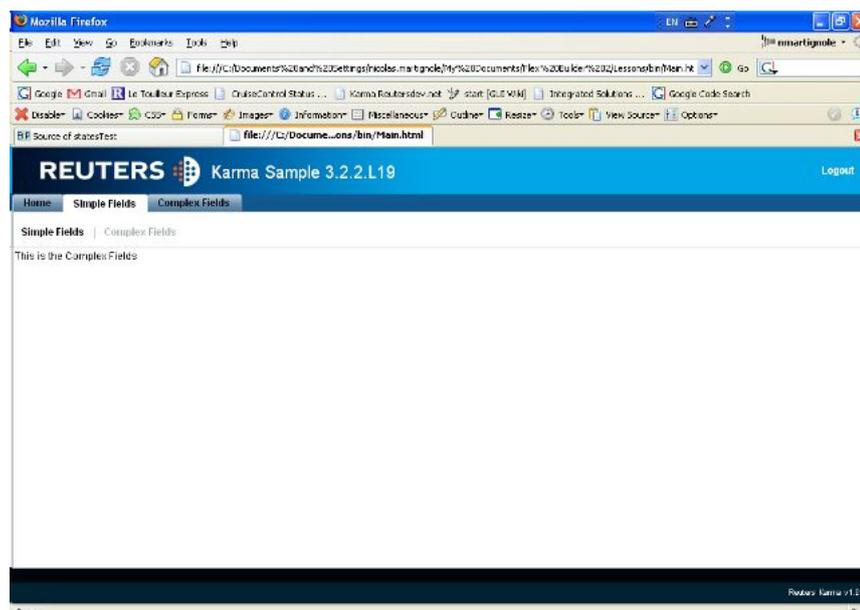
En fouillant sur le web ce matin j'ai déjà reperé quelques blogs, et quelques sites sympas.

Kuler est un module pour sélectionner la couleur primaire et les couleurs secondaires d'un site web. Cela vous permet de faire un site "beau" et pas un site d'informaticien avec 3 couleurs moches.  
<http://kuler.adobe.com/>

Le blog de Bruce PHILIPS est très sympa car on suit au jour le jour ses progrès et ses découvertes avec Flex. Il a maintenant un sacré maîtrise et j'y ai trouvé un bon tutorial pour faire la partie login de mon application. <http://www.brucephillips.name/blog/index.cfm>

Si vous voulez voir une liste des composants, des layouts et des charts de Flex vous pouvez regarder sur ce site: <http://examples.adobe.com/flex2/inproduct/sdk/explorer/explorer.html>

Enfin pour terminer je voulais mettre ici en ligne le fichier SWF réalisé cette nuit, mais JRoller refuse que j'upload des SWF. J'ai donc simplement mis une capture d'écran... Désolé pour l'image, j'essayerai de mettre un fichier SWF sur un autre serveur.



Flex est simple à apprendre, et le Flex Builder me paraît franchement indispensable pour gagner du temps lors de la création des écrans. Cette technologie au premier coup d'oeil fonctionnera bien si du côté serveur, l'architecture est de type REST, si vous pouvez générer du XML ou encore mieux, si vous disposez de Web Services.  
 A tester donc.

## JBoss Seam 2.0 Beta ajoute le support de Groovy et du Google Web Toolkit

Traduction d'un post de Gavin King ([article en anglais](#))

3 mois après la sortie de JBoss Seam 1.2.1, JBoss Seam 2.0 est maintenant disponible en beta. JBoss Seam est un framework applicatif pour Java EE 5.0 qui regroupe et facilite la mise en place des JSF d'une part et des EJB3 d'autre part. En plus cela, JBoss intègre la librairie iText pour générer des PDF, le moteur de règles Drools ou encore le moteur jBPM. L'architecture stateful de Seam permet de travailler et gérer d'une nouvelle manière les applications d'entreprise. Gavin King (auteur d'Hibernate et maintenant lead developer sur Seam avec T.Heute) liste quelques unes des caractéristiques du framework JBoss Seam:

- o Seam WebService permet aux composants Seam de fonctionner comme des web services
- o Il est possible d'utiliser Groovy pour écrire des composants Seam
- o Le coeur de Seam est indépendant de la partie JSF
- o Support de GWT (Google Web Toolkit)
- o Intégration de la fonction de recherche d'Hibernate
- o Introduction de JBoss EL (Expression Language)
- o Amélioration du module asynchrone de Seam, avec l'intégration de Quartz (moteur de planification de jobs open source)
- o Migration à JSF 1.2
- o Configuration simplifiée (beaucoup moins de XML que JSF seul)
- o Support pour Facelets (système de templates pour la composition des pages)
- o Amélioration de l'intégration du framework de tests
- o Nouveau système de transaction abstrait avec le support des environnements non JTA
- o Javadoc
- o 2 nouvelles applications de démo
- o Migration à JBoss application server 4.2.0 GA

- Composant JSF Seam réimplémentés en utilisant Ajax4JSF
- Bugfixes...
- 

J'ajoute en plus de cela la facilité pour intégrer dojotoolkit et la possibilité de faire fonctionner JBoss Seam dans Tomcat (donc sans devoir utiliser le serveur d'application de JBoss). La documentation de JBoss Seam a été aussi réécrite.

## Maven2 dans la vraie vie

Nous avons migré sur Maven2 l'ensemble de notre projet en quelques semaines. Cela peut sembler beaucoup et je pense qu'en me lisant, certains n'imaginent pas en fait ce que nous entendions par vraiment migrer vers Maven2. Cela a eu un impact sur la structure du code, l'intégration continue et la mise en ligne de nos versions.

La structure du projet a changé pour bénéficier au meilleur prix de maven2. Merci SVN cette opération s'est effectuée sans aucuns problèmes. Nous avons conservé l'historique de notre projet tout en faisant le choix de commencer une nouvelle branche, et ce parce que la version ANT est encore en production

Maven2 pour simplement remplacer ANT, pourquoi pas ? En effet, terminé la maintenance à la main des scripts ANT, et finalement la nécessité d'avoir un gars un peu "touchy" en Ant pour que la build passe de 10mn à 4mn. Maven2 s'en sort bien.

Concernant les dépendances internes, les JAR propriétaires qui n'existent pas sur le repository ibiblio, rapidement nous nous sommes posés la question d'installer un proxy maven2. Après avoir testé le proxy maven2 proposé par défaut, j'ai trouvé un très bon soft à la fois proxy et à la fois repository pour Maven2: [Artifactory](#). Simple à installer, ce proxy nous permet de mettre en cache les libs standards. Il nous permet aussi d'effectuer nos releases vers ce serveur, afin que les produits utilisant notre framework puissent se compiler à partir de maven2.

Pour gérer la version d'un produit avec maven, le plugin release est vraiment pratique. Il se charge de créer un tag, mettre à jour l'ensemble des pom.xml et enfin d'uploader vers Artifactory la version compilée. Pas évident à mettre en place mais vraiment top au final.

L'intérêt de Maven est aussi de pouvoir intégrer facilement tout un ensemble de rapports standards sur les tests unitaires, la couverture du code, la javadoc, le code source, la qualité et les éventuels bugs. C'est facile à intégrer et le résultat est très sympa.

Notre framework a une application d'exemple, un genre de "HelloWorld" mais un peu plus compliqué. Une fois l'intégration continue mise en place avec cruisecontro, s'est posé le problème de mettre en ligne une version de démo. Celle-ci permet d'effectuer quelques tests de non-regression et de montrer à nos clients les possibilités du framework. Bref pour maven2 la meilleure solution semble être d'utiliser Cargo. Ce plugin de [Vincent Massol](#) permet d'installer, deployer un serveur JBoss directement à partir de Maven2. Pour l'instant nous en sommes là et en attendant une petite copie effectuée avec une tâche Ant nous permet de continuer à travailler. A suivre donc.

## Google Guice petite frere de JBoss Seam ?

[Google Guice](#) (prononcez "juice") est un nouveau venu dans la place, du côté des frameworks de nouvelles générations Java EE5. Google Guice est clairement orienté IoC (Inversion of Control). Cela me rappelle le meilleur framework à ma connaissance qui allie à savoir... **JBoss Seam**. La version Alpha 1.3 fonctionne sans problèmes avec JBoss AS 4.2. Je pense qu'une GA sera disponible cet été dès que les derniers bugs seront corrigés. J'avais posté en janvier lors de mes nocturnes sur jboss Seam quelques liens. Si vous voulez y jeter un oeil, <http://jroller.com/page/Trecollo/20070131> Sinon mon petite frère, Aurélien, a été embauché chez Google en Irlande. C'est cool !

## Commandes sympathiques sous Unix

Quelques sites sympas si vous voulez apprendre de nouvelles commandes unix ou quelques tips sympas pour briller en société:

- [SysAdmin Cool commande](#)
- [SUN Solaris big admin](#)

## L'été sera chaud, l'été sera chaud

Vacances puis mission chez IBM pour réaliser le portage vers IBM Websphere Application Server 6.1 de notre produit oblige, ces dernières semaines c'était silence radio par ici. Promis je reviens bientôt poster quelques billets sur WAS, XDoclet et la mise en production de Maven2 sur notre projet.

## JBoss Seam et OutOfMemory... mais non

Que tout le monde reste assis et calme, les problèmes de mémoire de la dernière version de JBoss Seam 1.2.1 et de JBoss 4.2.0 n'existent pas... Plus sérieusement pour utiliser JBoss Seam il faut un JDK 5 ou 6. Même si JBoss a un système vraiment sympa de déploiement à chaud, en raison de bugs dans la JVM il peut arriver de rencontrer un OutOfMemory PermGen space lorsque l'on redéploie plusieurs fois une application. Pour éviter cela il suffit d'éditer le fichier JBOSS/bin/run.bat ou run.conf si vous êtes sous unix ou mac os X et de configurer la ligne JAVA\_OPTS avec une valeur raisonnable pour la taille de la PermGen.

```
-Xms512m -Xmx1024m -XX:PermSize=256m -XX:MaxPermSize=512
```

## JBoss Seam 1.2.1 GA et l'encoding, allez la France

J'ai trouvé un léger problème lorsque vous utiliser JBoss Seam et des pages en français. C'est un souci connu avec Tomcat, le connecteur qui est utilisé sur JBoss pour la partie Web. En effet l'encodage des URI est effectué en ISO-8859-1 par défaut, ce qui pose certains soucis lorsque vous voulez faire de l'UTF-8. Si vous ne sauvegardez pas les fichiers en UTF-8 dans Eclipse ou IntelliJ vous allez rencontrer par ailleurs un autre problème lors du chargement de la page car les caractères ne sont pas chargés avec le bon encoding.

```
Invalid byte 2 of 3-byte UTF-8 sequence.
```

Pour éviter cela il suffit de configurer le connecteur Tomcat embarqué dans JBoss en éditant le fichier jboss-4.2.0.GA\server\default\deploy\jboss-web.deployer\server.xml. Ouvrir server.xml fichier et ajouter un tag useBodyEncodingForURI et un tag URIEncoding comme sur cet exemple:

```
<Connector port="8080" address="{jboss.bind.address}"
  maxThreads="250" maxHttpHeaderSize="8192"
  emptySessionPath="true" protocol="HTTP/1.1"
  enableLookups="false" redirectPort="8443" acceptCount="100"
  useBodyEncodingForURI="false" URIEncoding="UTF-8"
  maxKeepAliveRequests="50" />
```

Relancer le serveur et tester en éditant une page avec des caractères iso (accents, le signe EURO par exemple) [Voir la doc de tomcat pour plus de détails](#)

**Update dimanche soir** je précise que pour être complètement tranquille le mieux est de sauvegarder vos fichiers xhtml en UTF-8 et de spécifier l'encoding dans l'entête de la page :

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
```

## A propos d'ICEFaces 1.6, JBoss Seam et mon projet de dans deux ans

Je suis sur un nouveau projet personnel. Basé sur JBoss Seam, j'ai besoin de bibliothèques JSF assez puissantes pour gagner du temps. Ce matin j'ai vu qu'ICESoft Technologies, Inc a annoncé la sortie officielle de la bibliothèque JSF ICEFaces en version 1.6. Cette nouvelle version s'intègre avec JBoss Seam 1.2.1 GA. Des [tutoriaux](#) permettent d'apprendre à coder des pages rapidement. Le mieux cependant est peut-être de regarder [la page de démonstration](#) pour vous faire une idée.

Par rapport à d'autres bibliothèques comme tomahawk qui a du mal à s'intégrer avec Seam, ICEFaces a été testé et validé avec JBoss Seam. Pour ma part **je cherche un composant jsf d'agenda** qui tienne la route. J'aimerais bien tester [MyFaces Schedule](#) mais pour mettre en place tomahawk dans JBoss il faut désactiver l'implémentation jsf par défaut. Je n'ai pas encore testé, car je regarde une autre alternative utilisant Google Agenda. L'idée est de récupérer l'agenda via du HTML, ce que j'ai réussi à faire hier soir sans trop d'effort. Google met aussi à disposition une API Java pour poster des événements vers un agenda Google. Bref j'ai tout, sauf qu'au niveau de l'ergonomie l'agenda qui s'affiche en mode HTML est moins sympa à utiliser que la vraie version de Google Agenda... Donc me voilà revenu au point de départ.

Sinon JBoss Seam est vraiment puissant... mais chut, tant pis pour ceux qui se réveilleront trop tard...

## Alert Sécurité Java AusCert: je vous explique

## tout

Selon l'AusCERT (Australian Computer Emergency Response Team) un organisme chargé de faire suivre les alertes de sécurité, une grave faille de sécurité a été trouvée dans la JVM de SUN par des gens de chez Google. Attendez, restez jusqu'à la fin avant de sauter sur votre téléphone/email/blackberry pour prévenir vos chefs/amis/femmes/copainsDeMicrosoft car il faut prendre le temps de lire le bulletin d'alerte [sur le site de l'AusCERT](#).

La vulnérabilité concerne la JVM de SUN dans les versions suivantes:

- o Sun Java Runtime Environment (JRE) 6
- o Sun Java Runtime Environment (JRE) 5.0 Update 10 and prior
- o Sun Java Runtime Environment (JRE) 1.4.2\_14 and prior
- o Sun Java Runtime Environment (JRE) 1.3.1\_20 and prior
- o Sun Java Development Kit 6
- o Sun Java Development Kit 5.0 Update 10 and prior
- o Sun Java Software Development Kit 1.4.2\_14 and prior
- o Sun Java Software Development Kit 1.3.1\_20 and prior

L'impact de la faille de sécurité permet d'exécuter n'importe quel code sur le poste d'un navigateur Web équipé d'un JVM (plugin Java) par exemple.

La faille elle-même concerne un dépassement de pile (j'ai du mal avec le français: buffer overflow) dans le module de chargement des images JPG ou BMP ayant une partie IIC (Integrated Information Color). Le plantage lors du chargement entraîne ensuite un plantage de la machine ou l'exécution d'un code embarqué dans l'image. [Un exemple de code se trouve ici](#).

*A buffer overflow vulnerability in the image parsing code in the Java Runtime Environment may allow an untrusted applet or application to elevate its privileges. For example, an applet may grant itself permissions to read and write local files or execute local applications that are accessible to the user running the untrusted applet.*

Deuxième problème de sécurité, cette fois-ci lié au chargement d'une image BMP sur Unix et Linux uniquement. Elle peut causer une attaque de type déni de service (denial of service) faisant planter le poste en effectuant le chargement d'une image BMP spéciale.

*A second vulnerability may allow an untrusted applet or application to cause the Java Virtual Machine to hang.*

Maintenant mon avis sur la question: les applets Java, non signées ne sont plus vraiment utilisées. La technologie Flash a remplacé depuis longtemps Java sur le poste client de monsieur tout-le-monde. De ce côté là, peu d'impact. Mais bon, le risque existe. Je pense aussi que la JVM a encore une belle marge de progression si nous comparons celle-ci à la liste des soucis de sécurité d'Internet Explorer. [Le player Macromedia Flash](#) n'est pas plus épargné par les alertes de sécurité.

Bref si vous voulez corriger ce problème sachez que SUN a résolu ce problème dans les versions suivantes de la JVM:

- o JDK and JRE 6 Update 1 or later
- o JDK and JRE 5.0 Update 11 or later
- o SDK and JRE 1.4.2\_15 and later

Référence: <http://sunsolve.sun.com/search/printfriendly.do?assetkey=1-26-102934-1>

Le site de Chris Evans chez Google: <http://scary.beasts.org/security/CESA-2006-004.html>

## JBoss Seam 2.0 Beta 1 premiers tours de clefs

Hier soir j'ai compilé la dernière version de JBoss Seam 2.0 Beta 1 toute fraîche tombée de CVS. La release officielle est pour bientôt mais j'avais envie (et besoin) de vérifier comment mes applications passeront en 2.0

Tout d'abord au niveau des packages, comme Gavin l'a expliqué dans un post (voir forum JBoss), il y a eu une réorganisation et un nettoyage des packages. De manière très concrète, il faut prévoir un peu de refactoring au niveau des imports mais rien de très compliqué.

Au niveau de la configuration, attention au fichier /resources/WEB-INF/components.xml qui a pas mal changé. L'intégration du système d'Identity de jBPM fait qu'il faut reprendre ce fichier, sous peine sinon de se retrouver avec ce genre d'exception:

```
11:20:48,973 ERROR [org.apache.catalina.core.ContainerBase.[jboss.web].[localhost].[/seam-matstan]] Exception sending context initialized
java.lang.RuntimeException: error while reading /WEB-INF/components.xml
    at org.jboss.seam.init.Initialization.initComponentsFromXmlDocument(Initialization.java:136)
    at org.jboss.seam.init.Initialization.create(Initialization.java:86)
    at org.jboss.seam.servlet.SeamListener.contextInitialized(SeamListener.java:34)
    at org.apache.catalina.core.StandardContext.listenerStart(StandardContext.java:3854)
    at org.apache.catalina.core.StandardContext.start(StandardContext.java:4359)
    at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:761)
    at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:741)
    at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:553)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
```

```
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
```

Attention aussi au niveau de la qualification de nommage des composants dans Seam. Exemple trivial: j'ai une class Session qui traîne dans ma conversation. Or il faut savoir que certains composants Seam sont branchés avec une @Factory... Si vous avez un attribut "session" que vous voulez injecter comme sur cet exemple:

```
@In
MyObject session;
```

... cela ne marche pas car Seam enregistre aussi ce même nom pour la session Seam, en l'occurrence ici org.jboss.seam.web.session. En général il est bien de forcer les noms des composants pour éviter ce genre de problèmes.

Sinon je suis à fond dans jBPM. La version 3.2.1 que Gavin intègre (il y a des commits sur CVS en ce moment) offrira quelques nouvelles fonctions:

- o Notification par email lors de l'exécution d'une tâche en spécifiant notify="true" dans une task-node
- o Gestion de modèles d'emails pour cette partie
- o Gestion des Calendriers. Cela permet par exemple de programmer une task durant les jours ouvrés d'une société. Ou pour la finance, d'utiliser des calendriers cambistes afin d'affecter des tasks selon les dates d'ouverture de la bourse... (grand sourire)
- o Gestion des "due Date", "start Date", "end Date" sur les tasks. Il devient facile aussi de demander au système d'envoyer un email de rappel 2 jours avant l'échéance d'une task par exemple. Tout ceci grâce à jBPM.

Concernant la partie JSF, je déconseille d'essayer d'installer Tomahawk. Dommage car l'un des composants d'agenda est sympa. Mais Gavin King sur les forums de JBoss a expliqué les problèmes rencontrés pour faire fonctionner cette API avec Seam 2.0

Coté Ajax4jsf et RichFaces, ces deux API sont très simples à utiliser et vraiment mature. Pour se faire une idée des applications Seam 2.0, il faut vraiment tester **seamdiscs**. Sur le modèle de MySpace, la démo montre quelques composants de RichFaces. Je vous laisse tester par exemple la SuggestionBox en allant sur le site d'Exadel: <http://livedemo.exadel.com/richfaces-demo/richfaces/suggestionBox.jsf?c=suggestionBox>

Merci aussi sinon pour les emails reçus, j'ai de plus en plus de monde qui vient sur ce blog, avec aujourd'hui plus de 400 visiteurs uniques différents. Cela commence à faire du monde, je vois que les gens viennent de partout, cela fait vraiment plaisir de lire vos commentaires et vos encouragements. Merci !

## Naissance de Cloé

Un petit post personnel pour annoncer la naissance de notre deuxième bébé: Cloé (oui Cloé sans H). La puce est arrivée lundi 23 juillet, la maman se porte bien, le grand-frère est très content

## Convertissez vous à JBoss Seam mes frères

Ce matin sur [The ServerSide](#) j'ai trouvé un article de Michael Yuan sur JBoss Seam 2.0 qui reprend les nouveautés et présente le framework JBoss Seam. J'appelle à la conversion les développeurs qui ont un projet avec des EJB3 pour la partie business et JSF pour la partie Web. Vous verrez à quel point JBoss Seam vous simplifie le travail.

Voici quelques questions/réponses sur ce que l'on peut faire aujourd'hui avec JBoss Seam pour compléter l'article de Michael Yuan. J'espère vous apporter ici des nouvelles fraîches par rapport à ce que l'on trouve sur le net.

### Est-ce que JBoss Seam est payant?

Non, JBoss Seam est distribué sous licence LGPL.

### Est-ce que je dois absolument utiliser le serveur d'application JBoss pour profiter de Seam?

Non, JBoss Seam peut être aussi déployé dans un conteneur léger de type Tomcat. Des portages sont en cours pour IBM Websphere, des utilisateurs ont aussi déployés Seam sur le serveur SUN Glassfish. Bref les développeurs de JBoss Seam ont clairement envie de fonctionner sur le plus de serveur d'applications du marché possible. Voir aussi [ce post pour Glassfish](#)

### Est-ce que JBoss Seam sait générer des applications Web 2.0 avec Ajax ?

La réponse est oui. A la base, Seam utilise JSF (Java Server Faces) et la communauté open-source propose des API qui sont maintenant intégrées de facto dans JBoss Seam. Il devient ainsi très simple d'ajouter des formulaires qui se mettent à jour partiellement via Ajax par exemple. D'autre part le module Seam Remoting permet au développeur d'implémenter une vraie application Web 2.0 facilement

### Est-ce que Seam peut envoyer des emails ? faire des traitements la nuit ?

JBoss Seam peut en effet envoyer des emails. Il est possible de générer un fichier PDF par exemple et de l'envoyer chaque nuit. Pour l'envoi de l'email, l'api est simple et facile à utiliser. Le traitement de batch avec Seam se repose sur un moteur Asynchrone. Seam peut par exemple faire une tâche chaque heure, chaque nuit et déclencher un traitement. Tout ceci soit via l'api EJB3 d'asynchronisme, soit via l'api Quartz qui est plus puissante. En effet celle-ci permet de créer des

tâches avec une date d'échéance par exemple.

Concernant l'envoi d'email, jBPM 3.2 qui est en cours d'intégration dans JBoss Seam 2.0 Beta 1 permet aussi d'envoyer un email lorsqu'une tâche est terminée.

#### Est-ce qu'il est possible de changer de langue, de locale à la volée ?

Oui en effet le support de l'internationalisation (i18n) est fortement intégré dans Seam, en partie via JSF. Par ailleurs il est possible d'ajouter un composant en 2 clics pour permettre de changer de langue et de Locale à la volée. Ce choix peut être persisté via un cookie. Il existe aussi une possibilité de changer le thème (la skin) d'une application (je pense au module Stylizer chez Reuters...) afin de changer l'aspect d'un site avec un seul clic.

#### Est-ce qu'Eclipse peut me permettre d'écrire des pages graphiquement ? Existe-t-il un studio ?

J'ai lu que Macromedia prépare une version de DreamWeaver qui reconnaît parfaitement JSF 1.2. Cela devrait permettre de coder les pages de la vue plus rapidement. Attention cependant, avec richFaces, ajax4jsf et iceFaces de toutes les façons ces studios ne fonctionnent pas. Il vaut mieux travailler en mode "explode". Ce mode consiste à compiler votre application, à déployer sur JBoss une arborescence explosée (EAR dans lequel on retrouve un repertoire WAR...). Cela permet alors de travailler sur les fichiers .xhtml (avec Facelets) sans devoir reconstruire tout l'EAR à chaque fois. Vous n'avez alors qu'à lancer un "ant restart" de temps en temps si vous touchez à la partie Java. C'est très pratique.

Par ailleurs sinon je vous conseille de dépenser un peu d'argent pour acheter IDEA IntelliJ. C'est mon quart d'heure sponsor ce soir. Mais entre Eclipse+MyEclipse vs IntelliJ, il n'y a pas photos. IntelliJ reconnaît le format xhtml, les JSF **et surtout les annotations de JBoss Seam** ce qui au passage devrait convertir j'espère quelques lecteurs ici. Ajoutez-y le support de JBoss AS en natif pour contrôler, débiter JBoss (comme JBoss IDE mais en mieux) et voilà, vous allez craquer. Si vous avez d'autres questions, n'hésitez pas à commenter.

## Aide mémoire pour Java, C++, C#, Python et ses amis

Le site [Cheat-Sheets.org](http://Cheat-Sheets.org) propose des liens vers différents aide-mémoires pour tout un tas de langage informatique comme Java, C++, C#, Python etc. Il y a aussi des liens vers des mémos pour tout ce qui est Web (CSS, HTML, DHTML...) ou d'autres choses comme [les extensions de fichier](#) ou des raccourcis [Photoshop](#) par exemple. Enfin mon préféré [spécial dédicace à flavien](#)

## JBoss 4.2.2 JspFactory error

Une erreur bête après avoir migré de JBoss 4.0.5 à JBoss 4.2.1 GA avec une simple page JSP:

```
org.apache.jasper.JasperException: Unable to compile class for JSP:
```

```
An error occurred at line: 22 in the generated java file
The method getJspApplicationContext(ServletContext) is undefined for the type JspFactory
```

Stacktrace:

```
org.apache.jasper.compiler.DefaultErrorHandler.javacError(DefaultErrorHandler.java:92)
org.apache.jasper.compiler.ErrorDispatcher.javacError(ErrorDispatcher.java:330)
org.apache.jasper.compiler.JDTCompiler.generateClass(JDTCompiler.java:415)
org.apache.jasper.compiler.Compiler.compile(Compiler.java:308)
org.apache.jasper.compiler.Compiler.compile(Compiler.java:286)
org.apache.jasper.compiler.Compiler.compile(Compiler.java:273)
org.apache.jasper.JspCompilationContext.compile(JspCompilationContext.java:566)
org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:311)
org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:320)
org.apache.jasper.servlet.JspServlet.service(JspServlet.java:266)
javax.servlet.http.HttpServlet.service(HttpServlet.java:803)
com.rfs.j2ee.KiWI.util.CompressFilter.doFilter(CompressFilter.java:86)
com.rfs.wfsignon.web.SignOnFilter.doFilter(SignOnFilter.java:186)
com.rfs.j2ee.KiWI.util.EncodingFilter.doFilter(EncodingFilter.java:38)
org.jboss.web.tomcat.filters.ReplyHeaderFilter.doFilter(ReplyHeaderFilter.java:96)
```

Cette erreur est toute bête: l'implémentation de la class JspFactory retournée par le classloader charge une version qui se trouve dans votre EAR ou dans votre WAR. Pour nous en effet nous avons servlet-2.3.jar à la racine de l'EAR (construit avec maven2). Bref il suffit de retirer cette archive de l'EAR et l'application fonctionnera à nouveau. Pensez aussi à retirer un éventuel j2ee.jar qui n'a rien à faire à la racine de votre EAR.

Pour exclure **servlet-2.3.jar** et **jsp-api.jar** avec Maven2 lors du packaging, ajouter dans le pom.xml qui genere l'EAR une dependency en forçant le scope à la valeur "**provided**":

```
<dependencies>
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>servlet-api</artifactId>
    <version>2.3</version>
    <scope>provided</scope>
  </dependency>

  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jsp-api</artifactId>
```

```

<version>2.0</version>
<scope>provided</scope>
</dependency>

</dependencies>

```

J'en profite pour vous donner l'adresse d'un article de présentation et de mise en route de Maven2 très bien écrit: [sur le Blog de Jean-François Helie](#)

**English version:** The JspFactory that is loaded by the Jasper is not the correct one. Check that you don't have a JspFactory impl in your EAR or WAR/WEB-INF/lib subfolder. In Tomcat 6, the JSP API classes are found in jsp-api.jar and servlet API classes are found in servlet-api.jar, with both located in the lib directory of the Tomcat 6 installation. Any servlet.jar you find involved with this server would contain out of date classes for this server. Delete servlet-2.3.jar from your EAR.

## Le Lapin Nabaztag et CruiseControl



Un peu de détente et d'humour, c'est ce qui me manque parfois au boulot. Je vous posterai une fois les plus gros délires de ma vie d'avant en start-up chez Coplanet et DotVision. Mais en attendant je voulais parler ce matin d'autre chose. Profitant de mon anniversaire, je me demande si je ne vais pas adopter [un Nabaztag](#), ce petit lapin magique qui permet de faire tout et n'importe quoi. Ce lapin relié à internet vous réveille le matin, vous donne la météo et peut vous lire vos emails. Il sera capable de vous signaler ce matin un incident sur la ligne A du RER. Pendant le petit déj il pourra vous faire écouter une web radio comme France Info. Ensuite une fois que vous serez au bureau, vous pourrez envoyer un email à votre Nabaztag, qui est resté à la maison. Il vous informera aussi que votre femme est rentrée avec les enfants, elle pourra même vous envoyer un petit message vocal... Bref tout un ensemble de choses complètement inutiles mais sympas. Qu'en est-il maintenant pour une application au bureau ? Je me suis demandé si je ne pourrai pas discrètement ramener la bête au bureau, quitte à passer encore pour un original.

Voici les idées en vrac que je me vois bien appliquer ici, histoire d'avoir un peu plus de fun dans l'équipe:

- **Lapin rapporteur de build:** imaginez ce lapin posé sur votre bureau. A chaque compilation automatique de votre projet Java avec CruiseControl, le lapin vous signale avec des lumières et une mélodie si la compilation est passée ou non. Vous pouvez aussi l'interroger afin de connaître les derniers messages de commit. C'est déjà possible [d'après ce post sur le blog de Nabaztag](#). Voir aussi [ce post de Tom](#)
- **Lapin veilleur de bug** le lapin est inscrit à une liste de diffusion d'emails que nos clients internes utilisent actuellement pour nous signaler des bugs. Lorsqu'il voit certains mots clés dans le sujet, il s'active et signale l'arrivée d'un nouveau bug à l'équipe
- **Lapin Mangeur** le lapin sonne le rassemblement vers midi pour proposer d'aller manger. Il récupère le menu sur l'extranet d'Eurest qui est à notre disposition.
- **Lapin Fetard** le lapin souhaite un bon anniversaire aux membres de mon équipe le jour J
- **Lapin Veilleur** en étant inscrits sur certains flux RSS, le lapin sait que JBoss, BEA Weblogic ou IBM Websphere sortent une nouvelle version... il nous le signale pour que nous allions voir
- **Lapin Meeting** connecté à mon agenda outlook, il me rappelle les réunions d'équipes, me donne la salle et sonne quelques minutes avant la réunion.

Bref voilà comment il serait possible d'instiller un peu de fun, de déstresser les gens et de rendre moins sérieux, plus productif et plus cool notre travail. Si vous avez au boulot un Nabaztag, contactez-moi par email pour me raconter votre expérience. Si vous cherchez un architecte Java/J2EE et chef de projet orienté Fun pour votre start-up, vous pouvez aussi me contacter. Les messages importants me seront lus par mon Nabaztag !

## Tranche de vie au boulot, mardi prochain dans l'avion

Imaginez la scène: c'était vendredi vers 15h45. Je viens de mettre à jour le planning. J'ai eu le temps de voir chacun des développeurs et de préparer une réunion pour lundi. Les bugs et les derniers mails sont traités, reste plus qu'à attendre 17h00 et à passer à un pot de départ. Je me suis même fait une petite heure de refactoring dans notre code (luxe) et j'ai enfin terminé de mettre à jour des scripts sur une de nos SUN. Bref j'étais fin prêt pour le week-end après cette semaine de reprise... et bien non. Dans la matinée un appel au support nous signale un souci sur un logiciel qui utilise notre framework. C'est à New-York. Problème en production. Discussion par messaging avec un gars de chez nous et les bureaux qui sont à New-York. Y'a du monde qui passe me voir

ensuite. On parle d'un problème avec un proxy ou un load-balancer F5 hardware. Bref sans pouvoir tout vous raconter (ca serait fun pourtant mais je ne peux pas), je prends l'avion mardi matin pour aller à New-York. Retour dans la nuit de vendredi à samedi. Je sens que la semaine prochaine va être fun. D'un côté je suis content de bouger un peu et d'aller sur site voir un client. D'un autre côté laisser la petite famille (cloé 2 mois/robin 23 mois) et laire se débrouiller c'est pas top. Mais bon, c'est le boulot. On pourra pas dire que je ne suis pas réactif :-)

## New-York, New-York

Voir Times Square en vrai, cela a son petit effet. Nos bureaux sont sur Times Square, et ce quartier ne dort jamais. J'ai eu le temps mardi soir de faire un tour à pied. Tout brille, tout bouge, les klaxons, l'odeur, le bruit, la foule... c'est assez speed. Voici quelques photos, on voit le bâtiment de Reuters qui est à l'angle de la 42<sup>ème</sup> et de Brooklyn, en plein dans Times Square. Il y a un énorme bandeau lumineux bleu (les couleurs Reuters) avec des news qui défilent.



Hier soir je me suis baladé en métro afin d'aller à Brooklyn voir de la famille. Je pourrai parler des heures des différences entre notre petit métro parisien et ici à new-york. Mais quelques mots seulement: vétuste, bruyant, climatisé, mal fleché. Cela dit, avec une carte dans la main à chercher mes stations, en fait les gens viennent vous aider spontanément. Jamais vu cela à Paris. Promis la prochaine fois que je vois un touriste perdu dans le RER, je fais ma B.A. Côté boulot ça va mieux. Côté boulot, après une première journée assez difficile hier mercredi, je pense avoir trouvé une solution pour les différents soucis rencontrés. Le code est fixé, il est 22h48 à Paris et 16h48 ici. J'attends de voir ce que donne la compilation mais ça semble bon. Quelque chose que j'avais remarqué ici par rapport à la France, c'est l'attitude des clients lorsqu'il y a des problèmes. A chaque fois nous avons rencontré un esprit positif où les gens cherchent à résoudre les soucis et ne nous tiennent pas personnellement responsable de ce qu'il se passe. Je me demande si dans la même situation nous aurions eu un aussi bon accueil finalement à Paris... Cela dit, vous êtes attendu au tournant. Pas question de raconter n'importe quoi. Le client sait ce qui ne va pas bien et attend une solution de notre part. Pas de politique, pas de passage de savon, on est là pour bosser... Dans ma boîte à outil j'ai ajouté 1 logiciel pratique capable de monitorer le trafic HTTP entre un navigateur et un serveur: [Charles](#) est un proxy qui enregistre le trafic, les temps de réponses, les headers HTTP. Très pratique. Je vous le conseille pour des soucis de cache, ou avec Soap. dtrace sur Solaris est aussi un outil à apprendre à maîtriser si vous voulez vraiment comprendre ce qui ne tourne pas rond sur une Sun. Retour vendredi dans la nuit... Même pas le temps de faire le touriste. Enfin je ramène un camion de pompier à Robin qui sera très content.

## Conseils pour acheter un appareil photo à New-York

Je termine ma mission aujourd'hui à New-york. Avant de partir je ne résiste pas à l'envie d'acheter de l'électronique. J'ai repéré le Canon Ixus 860 IS à 338 EUR chez Pixmania en ce moment. Il est à 400 USD ici ce qui donne 282 EUR, soit une économie de 56 EUR quand même. Pas mal non ? Bref me voilà sur Times Square à regarder les vitrines des magasins qui vendent de l'électronique. Ce sont des petits magasins qui montrent les appareils photos en vitrine avec des prix marqués à la main sur des étiquettes. On en voit tout autour de Times Square. Ce matin en quittant l'hôtel j'ai repéré ma proie dans une vitrine. Cependant une impression bizarre fait que je me suis dégonflé et que je n'ai pas acheté. Arrivé au bureau, j'en discute avec un de mes collègues. Or en effet, **n'achetez pas d'informatique ou d'électronique dans ces magasins**. Ce sont des arnaques ou de fausses bonnes affaires le plus souvent. [Cet article du Times](#) explique quelques principes pour attraper les gogos. Si vous voulez acheter je vous conseille d'aller sur la 5th Avenue à l'angle de la 43th Street chez BestBuy. Sorte de FNAC où les prix sont normaux. Sinon vous pouvez aussi aller chez B&H à l'angle de la 9th Avenue et de la 34th Street, au sud de Times Square. Le mieux est encore d'acheter à l'avance via Amazon et de se faire livrer au bureau à New-York ou chez un ami. J'ai donc acheté cet après-midi le Canon Ixus 860 IS avec une carte SD 1G0 le tout pour 472\$ taxes comprises, soit 333 EUR. Du côté du chargeur, un adaptateur à 2 EUR me permet de brancher le chargeur sur du 240v sans soucis à Paris. Voilà la fin de ma visite éclair à New-York. A peine le temps de faire du tourisme, je repars ce soir... Trop court, trop speed comme tout ici.

## CSS, Ajax et DHTML sont sur un bateau... Ajax tombe à l'eau, ça sent la javel



WE SMASH YOU WITH THE INFORMATION THAT  
WILL MAKE YOUR LIFE EASIER, REALLY.

Un site qu'il est bien de connaître si les mots CSS, Ajax, Javascript ou DHTML vous évoquent quelque chose: [Smashing Magazine](#) (prononcez "ce machine mag azine"). Voilà un petit nouveau que je ne connaissais pas et franchement, le contenu est très sympa. Pour les web designers, les graphistes ou les développeurs JSF/GWT, on y trouve de tout. Et pas trop de publicité pour une fois. On y apprend que 50.4% des visiteurs d'un site Internet seulement regardent en plein écran le contenu du site. Ce qui revient à dire qu'il faut éviter les designs bloqués à 1024/768 qui risquent de provoquer un agacement de vos visiteurs (je parle au web designer assis au fond à gauche). J'ai aussi vu [sur le site w3counter](#) qu'Internet Explorer ne représente **que** 66.28% des navigateurs utilisés aujourd'hui d'après ce site. Linux par contre est à des années lumières (1.38%) du monde Windows... Bref Microsoft rulez comme on dit chez les djeuns. Enfin un point important, ce sont les articles sur l'accessibilité et l'ergonomie. Ici nous travaillons avec un ergonomiste qui nous aide à créer des interfaces "intelligentes" ou à améliorer le design de nos écrans. Trop souvent oublié, le design et l'accessibilité sont aussi 2 points importants à prendre en compte lorsque l'on crée un site web.

## Java et JFreeChart sous Unix sans serveur X, sans écran

**Comment faire fonctionner un programme Java avec JFreeChart si le serveur sous unix n'a pas de serveur X ?**

Question d'un client: *notre application J2EE utilise JFreeChart pour générer des graphiques sur des pages webs. Or lorsque l'application est déployée sur une machine ne disposant pas de serveur X sous Unix, des exceptions Java apparaissent lorsque le moteur génère des images.*

Réponse: en effet sur un serveur en rack sans souris, sans clavier et à plus forte raison sans serveur X, Java doit être configuré pour fonctionner en mode *headless*. Si vous avez accès au code, vous pouvez ajouter une propriété système à l'initialisation:

```
System.setProperty("java.awt.headless", "true");
```

Vous pouvez sinon passer cette propriété système lors du démarrage de l'application:

```
java -Djava.awt.headless=true ...
```

Les classes Java de la librairie AWT sous Solaris et Linux ont des dépendances natives sur certaines classes X11. Lorsque JFreeChart génère une image PNG ou autre, il utilise (pour des raisons de performances) les librairies natives de la plateforme. Cela a un sens si votre serveur unix a une installation de ces librairies, ce qui n'est pas toujours le cas. Pour cela il est recommandé pour les applications web ou j2ee qui utilisent JFreeChart de forcer le mode *headless* pour éviter des soucis.

## Mac OS 10.5 est sorti mais Java 6 ne marche pas pour l'instant

L'événement la semaine dernière c'est la sortie de Mac OS 10.5, aka Leopard. Quelques heures plus tard des développeurs remarquent que Java 6 s'installe correctement mais crashes sur Leopard. Plus embêtant, la communication d'Apple sur ce point important est assez

**réduite pour l'instant. Qu'en est-il ce matin ?** Tout d'abord il faut comprendre qu'Apple a décidé de prendre en charge l'écriture de la JVM pour sa plateforme. [James Gosling](#) explique cela sur son blog dans un post. Il semble que c'est bien Apple que l'on doit blâmer aujourd'hui. Or il avait été annoncé au début de cette année à la conférence mondiale Apple (WWDC je crois) que Java 6 serait livré avec Leopard... D'où pour l'instant l'accueil glacial de la communauté. Tout le monde y va de son petit billet. Sur [JavaLobby](#), ou sur le blog [de David Herron](#). Sur le site [DamnHandy](#) on apprend qu'il est hors de question que Java6 soit abandonné sur Mac OS X. C'est plutôt le support du 64 bits qui explique pour l'instant le retard à l'allumage et donc le couac actuel. L'article le plus intéressant à mon avis est [l'analyse de Lisa Hoover sur CIO.com](#). Donc pour résumer il semble acquis qu'une version correcte de Java 6 sera dispo et finalisée sur Mac OS 10.5 et que l'impact pour l'instant est assez limité. A noter aussi que Java 6 est sorti il y a maintenant quelques temps et que l'on doit être à la révision 3 si je me souviens bien.

## Leopard, Mac OS 10.5 est sorti sans Java 6 pour l'instant

**L'événement la semaine dernière c'est la sortie de Mac OS 10.5, aka Leopard. Quelques heures plus tard des développeurs remarquent que Java 6 s'installe correctement mais crashe sur Léopard. Plus embêtant, la communication d'Apple sur ce point important est assez réduite pour l'instant. Qu'en est-il ce matin ?**

Tout d'abord il faut comprendre qu'Apple a décidé de prendre en charge l'écriture de la JVM pour sa plateforme. [James Gosling](#) explique cela sur son blog dans un post. Il semble que c'est bien Apple que l'on doit blâmer aujourd'hui. Or il avait été annoncé au début de cette année à la conférence mondiale Apple (WWDC je crois) que Java 6 serait livré avec Leopard... D'où pour l'instant l'accueil glacial de la communauté.

Tout le monde y va de son petit billet. Sur [JavaLobby](#) ou sur le blog [de David Herron](#). Sur le site [DamnHandy](#) on apprend qu'il est hors de question que Java6 soit abandonné sur Mac OS X. C'est plutôt le support du 64 bits qui explique pour l'instant le retard à l'allumage et donc le couac actuel. L'article le plus intéressant à mon avis est [l'analyse de Lisa Hoover sur CIO.com](#).

Donc pour résumer il semble acquis qu'une version correcte de Java 6 sera dispo et finalisée sur Mac OS 10.5 et que l'impact pour l'instant est assez limité. A noter aussi que Java 6 est sorti il y a maintenant quelques temps et que l'on doit être à la révision 3 si je me souviens bien.

## 10 profils de développeurs psychopathes ou comment on se soigne

**Disclaimer:** avant toutes choses je tiens à signaler que j'ai eu l'idée de ce post en lisant ce matin [cet article](#). J'essaye d'apporter ma touche personnelle sans trop reprendre ce qui est suggéré dans l'article original. Cet article s'adresse avant tout à ceux qui ne sont pas développeurs. Que vous soyez chasseur de tête, épouse/époux de développeur, commercial, chef de projet ou simple visiteur, je vais essayer de vous parler de quelques profils types que l'on rencontre dans la vie de chaque jour au bureau. Il existe plusieurs profils de développeur. On pourrait presque retrouver dans chacun de nous une ou plusieurs caractéristiques des profils ci-dessous. Mais après tout ceci n'est qu'une vue très subjective de notre petit monde à nous. Les profils de nos gentils développeurs:

- o le Résistant Zapatiste
- o le Jackass-men
- o l'Intermittent du spectacle
- o le Jeune qui aurait aimé faire la StarAc
- o le Transfuge Russe
- o le GPS
- o la Styliste
- o le Geek Royal
- o l'Artiste
- o l'Evangéliste

**Le Résistant Zapatiste** est un développeur souvent expérimenté. Il est vraiment doué dans son domaine. Vous lui demandez de coder une fonction rébarbative, il va prendre plaisir à le faire. C'est un bosseur. Il est râleur car il en a vu passer des technos et des machins qui font poum-poum. Mais lui c'est la vieille techno "old-school", celle qui marche qu'il aime. Alors il a peur du changement, son souci c'est l'avenir. Il excelle dans son domaine mais il a du mal avec la jeune garde qui ramène ses connaissances. Ses points forts sont l'expérience, le recul et la sagesse. Ses points faibles sont parfois sa difficulté à appréhender un problème avec de nouvelles technologies ou de nouveaux outils. Il va vous parler d'optimiser 23ms dans une boucle alors que de toutes les façons le temps de la requête est de 150ms à cause du réseau. Evidemment il a du mal avec Java ou C# car il pense encore que Java c'est lent et que C# c'est de la m... Bref il pense encore que la guerre n'est pas finie. **Le Jackass-Men** est souvent le jeune qui débute ou le vieux blasé qui tire à vue dans le code. Il est très productif. Avec lui les lignes de code et de commit sont envoyées par paquets de 10. Il corrige 2 bugs en même temps, vous code une fonction qui "déchire" en 2 heures. Son problème c'est qu'il va trop vite. Il ne fait pas de documentation de son code. C'est codé au fusil à pompe, souvent mal codé et pas testé. Il zappe rapidement d'un sujet à l'autre et parfois il laisse en plan avec des *TODO* dans le code. Il a 10 idées à la minute, il lance parfois des chantiers audacieux et c'est le souci. Il décide de réinstaller un serveur un vendredi soir sur un coup de tête (ou de gueule). Il réécrit une partie de votre produit, le résultat est bluffant en termes de

performances. Mais 2 semaines plus tard, impossible de relire son code. La doc ? C'est pour les stagiaires. Il code une superbe fonction dans votre produit mais il ne la teste pas sur tous les serveurs. Pas envie de s'embêter. Il est limite difficile à motiver sur des sujets importants comme mettre de la doc ou écrire des tests unitaires car il s'en fiche. Ses points forts sont la productivité, la réactivité et l'esprit d'initiative. Par contre au niveau qualité ou recul, il fait parfois des dégâts. C'est la personne qui restera jusqu'à 23h pour vous aider. En retour à vous de mettre quelques garde-fous pour ne pas le laisser flinguer le code. **L'Intermittent du spectacle** c'est le développeur qui a rejoint votre équipe un peu par hasard. Très sympa, mais il est là temporairement. Sa page d'accueil d'Internet Explore s'ouvre sur sur Monster.fr, il laisse trainer un CV sur l'imprimante. Il est content, il ne fait pas de bruit dans l'équipe. Mais on sent qu'il n'est pas là pour rester des années. Il fait son travail, ni plus, ni moins. S'il doit coder une fonction avant jeudi, elle sera codée. Par contre s'il a terminé en avance, vous ne le saurez jamais. Il travaille sur ses sujets à lui. On peut compter sur lui et il est content de travailler. Mais bon, il sera peut-être parti la semaine prochaine. Il suit le planning et rien que le planning. Il clique sur le bouton A et le bouton C mais il ne teste pas le bouton B car "on lui a pas dit". Il dit souvent "avant ça marchait" et il vient vous voir pour comprendre comment installer un éditeur de texte. Ses points forts sont qu'il bosse, il respecte le planning. Surtout pas de vagues. Ses points faibles: il n'est pas en retard 2 fois dans la même journée. S'il arrive à 10h il fait bien attention de partir à 17h30. **Le jeune qui aurait aimé faire la StarAc** c'est un profil qui fait du code, mais sans passions. C'est un bon développeur, pas exceptionnel mais tout à fait correct. Son bureau est ultra-customisé, son lecteur WinAmp/iTunes est toujours ouvert. Il code surtout en suivant du code existant, ou en regardant comment font les autres. C'est un esprit ouvert, il ne critique pas les nouvelles idées. Il a besoin de fun pour avancer. Par contre il manque un peu de recul et parfois il se fait entraîner sans oser être autonome. Ses points forts seraient sa capacité de travail, sa productivité. Ses points faibles sont qu'il n'est pas motivé, qu'il est suiveur et jamais moteur. C'est quelqu'un avec un fort potentiel s'il était motivé. C'est à son chef de se bouger pour l'aider et qu'il arrête de penser à *..la Mussiiiiique, oui la muusiiiiique...* **Le transfuge Russe** est un cas à part. Souvent une personne timide et silencieuse. On oublie qu'elle existe car elle ne parle que quelques minutes par jour. Par contre ce développeur est là et bien là. C'est au niveau des commits, de la quantité de code écrit que l'on sait qu'il est là. Il est très fort, c'est un zéro communicant qui code 15 fois mieux que la moyenne des développeurs. A lui les sujets dont personne ne veut entendre parler. Il va vous réinstaller une Solaris, compiler une librairie et terminer de réécrire un module en 3 jours. On sait qu'il a terminé une tâche non pas parce qu'il le dit, mais parce qu'avec lui: CA MARCHE. Ses points forts: il fait le squelette et le meilleur code de votre produit. Ses points faibles: très mauvais communicant. Même ses emails de 3 mots sont à déprimer, il ne parle pas et il est parfois difficile de savoir s'il est heureux, comment va la vie, bref c'est un transfuge... **Le GPS** c'est le Google de votre projet. Ce développeur sait où se trouve LA fonction qui fait papa-maman dans votre code. Demandez-lui qui a implémenté la fonction XBZ dans la lib Kondor, il le sait. C'est la seule personne qui peut déboguer en lisant du code, il sait analyser et vous orienter dans le code. C'est vraiment un plaisir. Il sait aussi les dates de vacances de l'équipe ou si vous avez eu une réunion avec un client la semaine dernière. Il sait que ce midi vous êtes chez le coiffeur. Il vous remplace sans soucis quand vous n'êtes pas là car il adore jouer avec Microsoft Project. Il passe son temps parfois à lire du code d'un oeil rêveur. Seul lui sait ce qu'il se passe dans sa tête lorsqu'il vampirise le code... Ses points forts sont qu'il constitue la mémoire de votre produit et une personne fiable pour communiquer. Ses points faibles c'est qu'il sera parfois tenté de garder certaines informations pour lui, ce qui représente un danger pour le projet. **La Styliste** est un développeur qui a pour habitude de rhabiller le code qu'il voit. Il ouvre un fichier Java, ou du C# et en 10mn il vous a rendu cela digne des spécifications de présentation du code de SUN. C'est celui qui va renommer les variables, mettre la javadoc, reformater le fichier. Limite maniaque, il a tendance à perdre du temps sur la forme sans aller regarder le fond. Ses points forts sont que c'est un esprit carré, qui est maniaque du détail. Le code qu'il écrit est documenté, c'est le seul de votre équipe qui sache écrire de la javadoc, faire des tests unitaires et mettre à jour le wiki de votre produit. Ses points faibles sont qu'il fait parfois trop de qualité et pas assez de quantité. Il aura tendance à énerver les autres développeurs à changer ou reformater leur code. Intéressant dans une équipe pour améliorer la qualité et entraîner les autres. **Le Geek Royal** est un profil assez courant. C'est un développeur, ou plutôt c'est le meilleur développeur de votre équipe. Et il le sait. Son ego n'a d'égal que son talent. Il est capable de répondre à une question simple d'un client par "RTFM" (Read the Fucking Manual/Lisez le putain de manuel). Il est vraiment mauvais en diplomatie. Il a aussi parfois des difficultés avec les autres membres de l'équipe. Du côté de son code, rien à redire. C'est le BOSS. Le meilleur. Que ce soit au niveau de l'architecture ou de l'implémentation il se balade. On peut rien trouver à redire. Ses points forts: c'est le meilleur. C'est tout. Ses points faibles: tendance à écraser les autres, ou à prendre un problème de haut. Difficile aussi de lui demander de remplir une fiche de bug, de suivre un planning ou d'encadrer un stagiaire. Il vit SA vie et vous devez vous adapter à SON rythme. Contrairement au Transfuge Russe qui fait tout en silence, lui il fait ce qu'il veut en manifestant sa présence et son talent. Mais il est bon... alors que dire ? **L'Artiste** c'est le perfectionniste. Ce développeur est un artiste car il aime les solutions élégantes, qui sont belles. Il fait un effort de présentation du code. S'il fait du web, il met de la couleur. Il a un don certain pour utiliser une palette de couleur dans une interface Swing pour que ce soit beau. Lorsqu'il code, et c'est parfois un défaut, il cherche à réaliser l'oeuvre parfaite. Il traite tous les cas possibles, même les plus improbables. Il écrit du code qui sera certainement trouvé en 2046 en archéologie numérique par un étudiant. Il code et programme pour que cela dure. Il a géré le bug de l'an 3000. Il sait par coeur les combinaisons de couleur hexadécimales. Il utilise Beautify sur son code. C'est lui qui vous a fait de superbes templates Word avec le logo de votre équipe. Et il se charge des JPEG de la page d'accueil. Ses points forts sont la qualité, la robustesse du code. Ses points faibles sont la vitesse de production, la réactivité. Il a du mal à "terminer" un programme. Il trouvera un cas d'usage dont personne n'avait imaginé. C'est un Artiste. A encadrer en respectant sa vision. **L'Evangeliste** c'est le gourou, le guide de votre groupe. C'est le seul qui lit des articles techniques ou des blogs pointus. Il a une grosse base de connaissances sur les produits, les frameworks ou les outils. Il a lu un grand nombre de livres. Il va sur Slashdot, puis sur InfoQ et enfin TheServerSide chaque jour. C'est lui qui vous a proposé de passer à maven2, qui sait comment configurer et installer un serveur Yota ou qui a déjà codé un peu avec Rails. Il est toujours en avance de 2

technologies. Ses points forts: c'est le poumon de votre équipe. Avec lui difficile d'ignorer l'existence d'Hibernate. Parmi ses points faibles, c'est parfois sa trop grande avance dans le temps par rapport aux réalités de l'industrie. Il évangélise tout le monde. Limite intégriste. Il croit parfois dur comme fer qu'une techno est bonne, alors qu'il n'y a aucun produit en production avec. Il installe déjà Weblog 11 alors que vos clients parlent à peine de passer en Weblog 9. Difficile à gérer lorsqu'il est frustré que ses idées ne soient pas mise en place tout de suite. Par contre il est indispensable dans l'équipe pour éviter de faire encore du *System.out* au lieu d'apprendre Log4J. Bref ce qu'il faut retenir c'est que les développeurs sont **des grands malades** et des grands enfants qui ont le pouvoir de créer. Ils ont forcément un égo, après tout ils créent et ils composent des choses que seul eux peuvent comprendre... Alors soyez gentil avec un Evangéliste ou un Geek la prochaine fois que vous en voyez un. Il y a toujours quelque chose de bon chez un développeur. Toujours.

## La FreeBox devient serveur d'impression et IPv6

La freebox est maintenant aussi capable d'être un serveur d'impression. Vous pouvez brancher une imprimante sur le port USB ou sur l'un des ports Ethernet afin de pouvoir imprimer depuis votre PC portable relié lui, en Wifi avec la freebox. Cela vous permet donc d'avoir une imprimante rangée dans un coin, et vous évite de devoir investir dans une imprimante Wifi. Par ailleurs vous pouvez aussi partager cette imprimante avec plusieurs machines, PC ou Mac. Vraiment pratique ! Pour brancher l'imprimante vous pouvez utiliser le port USB au dos de la Freebox. Ensuite sous Windows, il suffit de déclarer une imprimante locale, en spécifiant un port TCP. Vous mettez l'adresse de la freebox (192.168.0.254) et c'est tout ! Vous trouverez un tutorial sur le site FreeNews [à cette adresse](#)

## Prédictions pour 2008

Chaque fin d'année voit sa sortie de publication d'horoscope pour l'année 2008. Pas un magazine ne manque l'occasion de nous prédire notre avenir, la santé, l'amour et l'argent pour 2008. Mais pour nous, informaticien de base, ce qui compte c'est plutôt d'imaginer ce que 2008 nous réserve. Je me lance, mais si vous avez des idées n'hésitez pas à commenter. Les posts des commentaires sont modérés afin d'éviter le spam. **Pour 2008 je prédis que :**

- o nous entendrons parler de **la sortie du mZune** l'équivalent de l'iPhone mais chez Microsoft. Basé sur le baladeur Zune de Microsoft, ce téléphone de nouvelle génération sera basé sur Windows Mobile. Accès Wifi à la plateforme Live! de Microsoft, Hotmail, Messenger, etc.
- o nous parlerons de l'arrivée d'ActiveSync pour l'iPhone. Apple aurait fait une demande à Microsoft afin de pouvoir installer un client ActiveSync sur l'iPhone. A suivre donc.
- o Une annonce au MacWorld dans quelques semaines du premier lecteur portable vidéo intégrant une puce TVB pour la télé numérique terrestre.
- o Google annoncera le lancement de son offre de téléphonie aux USA sur le réseau 700Mhz. Voir [cet article](#).
- o Sony annoncera le support de la lecture des DivX sur la PS3. Parait que c'est déjà le cas en patchant la console officiellement.
- o enfin Java 1.4 sera obsolète en novembre 2008. Et ça c'est pas une blague.
- o **Java 6** sur Mac d'ici à février sera disponible. En principe [c'est en bonne voie](#) puisqu'une preview 8 est disponible au moment où j'écris ces lignes
- o comme cette année, LeWeb3 sera encore un gros événement réussi. Le concours des startups est vraiment une bonne idée et j'ai découvert de nouvelles sociétés avec des offres vraiment top.
- o je prédis aussi que le phénomène du blog va se tasser. Tweeter et autre instant-chat vont vraiment exploser en 2008. Le mail devient une voie de communication "lente" pour les "vieux" alors que les jeunes utiliseront massivement les messageries instantanées.
- o Explosion des sites de personnalisation (tee-shirt, mug, calendrier, photos, livre pour enfant...). C'est la tendance en ce moment. On cherche à personnaliser les cadeaux que l'on offre à ses proches

Voilà pour les quelques idées pour 2008.

## Google Disk: la lettre au Père Noël

**Lettre au petit Papa Noël de Google** Cher Google, j'ai été sage cette année. J'ai bloqué à propos de toi plusieurs fois et je parle de toi autour de moi. Du coup, pour Noël, je souhaite un espace de stockage, un genre de disque dur virtuel. J'aimerais y stocker mes papiers scannés, mes différents documents administratifs et personnels. Je souhaite aussi un logiciel sur Windows et Vista à installer afin que mon gdrive soit accessible via une lettre réseau sur mon bureau (par exemple g:). Tiens, tu as vu, je l'ai baptisé **GDrive**. Je serai prêt à payer un petit peu pour avoir un accès sécurisé et un espace plus grand pour les fichiers importants. Alors s'il-te-plaît, réalise mon souhait pour 2008. Nicolas... Sites parlant de Google Disk: [Blogs Computer WorldGDisk](#) est un service qui transforme votre compte Gmail en espace de stockage.

## Bilan de 2007

Le bilan de 2007, au niveau boulot ou perso:

- A la maison, naissance le 23 juillet de notre deuxième enfant, Cloé
- J'ai formé 4 personnes à Java, expérience vraiment sympa.
- Ma maquette de JBoss Seam pour le boulot a pas avancé.
- Ma maquette 2 avec Adobe Flex vient de ressortir des cartons mais c'est pas encore vendu. Trop novateur, des incertitudes pour le binding avec les données. Même s'il est possible de faire une belle interface utilisateur, si l'architecture derrière ne suit pas c'est de l'argent jeté par la fenêtre.
- Un voyage à New-York chez un client en octobre. On change d'air, content de voir du pays, de parler avec un "vrai" client
- Un pari perdu sur un gros bug de synchronisation de Thread, le genre de problème sympa que vous êtes deux à comprendre dans la boîte. Je t'oublie pas humberto, tu l'auras ta bouteille !
- Je ne sais pas combien de commit, je compte plus. Vive SVN. Ah non, c'est plus à la mode. Vive [Git](#)
- 58 articles publiés sur mon blog en 2007. Les 3 articles les plus consultés sur le blog cette année sont [Sun TechDays, la pillule bleu](#), ensuite [Flex2, premier pas](#) et enfin [Convertissez-vous à JBoss Seam mes frères](#)
- Une mission chez IBM pour faire un portage WebSphere 6.1. J'ai maintenant des idées très arrêtées sur Websphere, et je peux parler en connaissance de cause.
- Un projet perso pour mon frère réalisé avec JBoss Seam cet été.

En attendant, **BONNE ANNEE 2008 !**

## Envie d'iMac pour Noël



C'est une envie de geek et de développeur Java. Envie de passer à un iMac et de jeter Windows. Comme vous le savez sans doute pas, j'ai été longtemps un utilisateur de linux. 3 ans chez Dotvision avec Mandrake. Puis ma première année chez Reuters avec Mandrake+Wine pour faire fonctionner Lotus Notes. Un jour j'en ai eu un peu ras-le-bol de bidouiller tout le temps sous Linux. Stop, pas de commentaires, moi je parle avec mon expérience d'utilisateur. Je ne polemique pas. J'ai une Fedora à droite de mon portable au bureau, je m'en sers toujours. Mais à la maison ?

Apple finalement propose ce que je cherche: tout d'abord un esthétisme, un look que j'aime beaucoup. Ensuite Mac OS X Tiger me semble vraiment correspondre à ce que j'ai envie de faire. Et pour développer, IDEA IntelliJ tourne sans problèmes sur Mac.

En attendant ce week-end j'ai remonté notre PC avec Vista à la maison. Pas plus tard qu'hier, Robin, 2 ans, a fait tomber un bol d'eau sur l'unité centrale. Celle-ci s'est éteinte. J'ai séché l'uc au sèche-cheveux mais je n'y crois pas trop... C'est un signe du destin. Je dois quitter Windows... Allez, au revoir...

**Bonus** [La publicité "Meilleur Rendu" de la campagne Apple](#)

## J'ai acheté un iMac 20"

Ca y est... l'iMac est arrivé chez nous pour remplacer notre pauvre PC qui ne marche plus depuis que mon fils de 2 ans a versé de l'eau dedans... Papa Noël nous a donc apporté un bel iMac 20" 2,4 Ghz avec 260 Go de disque dur, et 2 Go de mémoire grâce à une offre Surcouf. Les premiers jours avec la bête se passent bien, je dois juste passer de Tiger à Léopard car le dernier système d'exploitation d'Apple n'était pas encore installé sur mon appareil. J'en profite pour créer une nouvelle section Apple sur mon Blog afin de regrouper mes messages. Et en plus c'est le dernier post de 2007 !**Bonne année 2008 et à bientôt!**

## Comment Steve Jobs a présenté l'iPhone à MacWorld

Chaque année a lieu à San Francisco la fameuse conférence **MacWorld Expo**. Cette année elle se tiendra à partir du 14 janvier 2008. Et chaque année, la présentation de Steve Jobs est l'événement que toute la communauté attend avec impatience. Je vous propose de revenir sur comment Steve Jobs a présenté l'iPhone et Leopard cette année. Le 9 janvier 2007, Steve Jobs commence la journée par sa présentation habituelle. Évidemment de grandes rumeurs courent sur Internet depuis quelques semaines. On attend avec impatience les nouveautés d'Apple cette année. Ce qui vaut le coup, c'est de revoir comment Steve Jobs a présenté à la presse et aux visiteurs l'iPhone. Vous trouverez [à cette adresse](#) une transcription de sa présentation. Je vais la traduire en partie en français. 9h41, Steve Jobs après avoir parlé des ventes sur iTunes et présenté quelques informations générales, passe à un slide représentant le logo d'Apple.

Steve Jobs: "Aujourd'hui est un jour que j'attends depuis 2 ans et demie. Il y a de temps à autre un produit révolutionnaire sur le marché."

La foule comprend alors d'un seul coup l'idée géniale: un iPod, un téléphone et un appareil connecté à Internet, dans un seul appareil... l'iPhone est né ! Fort non la mise en scène ? Dans la suite de la présentation, Steve montre les claviers des Smart-phones. Sont-ils si Smart ? Avons-nous besoin de ces petites touches où de toutes les façons nous n'arrivons pas à taper ? Non ! Et il présente ensuite l'écran tactile et une nouvelle révolution en terme d'ergonomie et d'interface utilisateur. Moi je dis: Merci à Apple d'exister, nous n'aurions peut-être pas de souris sans eux... Rendez-vous dans un mois pour dévoiler les secrets d'Apple pour MacWorld Expo.

## Envie d'iMac

 C'est une envie de geek et de développeur Java. Envie de passer à un iMac et de jeter Windows. Comme vous le savez sans doute pas, j'ai été longtemps un utilisateur de linux. 3 ans chez Dotvision avec Mandrake. Puis ma première année chez Reuters avec Mandrake+Wine pour faire fonctionner Lotus Notes. Un jour j'en ai eu un peu ras-le-bol de bidouiller tout le temps sous Linux. Stop, pas de commentaires, moi je parle avec mon expérience d'utilisateur. Je ne polemique pas. J'ai une Fedora à droite de mon portable au bureau, je m'en sers toujours. Mais à la maison ? Apple finalement propose ce que je cherche: tout d'abord un esthétisme, un look que j'aime beaucoup. Ensuite Mac OS X Tiger me semble vraiment correspondre à ce que j'ai envie de faire. Et pour développer, IDEA IntelliJ tourne sans problèmes sur Mac. En attendant ce week-end j'ai remonté notre PC avec Vista à la maison. Pas plus tard qu'hier, Robin, 2 ans, a fait tomber un bol d'eau sur l'unité centrale. Celle-ci s'est éteinte. J'ai séché l'uc au sèche-cheveux mais je n'y crois pas trop... C'est un signe du destin. Je dois quitter Windows... Allez, au revoir... **Bonus** [La publicité "Meilleur Rendu" de la campagne Apple](#)

## Comment tester un site avec IE7 sans l'installer

Problème: j'utilise Microsoft Internet Explorer 6 SP2 sur mon poste de travail, installé en XP Pro. Comment tester avec IE7 sans devoir l'installer ? J'ai trouvé une solution via Google, que je vous traduis en français ici.

- o **Etape 1** télécharger Internet Explorer 7 à partir du site de Microsoft ou de FileHippo par exemple. Le fichier porte l'extension .exe et doit s'appeler en principe IE7-WindowsXP-x86-fra.exe.
- o **Etape 2** Créez un répertoire IE7 à la racine de votre disque. Placez-y l'archive de IE7 téléchargée et renommez-la en **IE7-WindowsXP-x86-enu.exe**.
- o **Etape 3** télécharger [le package IE7 standalone](#) créé par John Galloway. Décompressez ensuite cette archive dans le répertoire IE7 créé à l'étape précédente. Vous devez donc avoir au final 7 fichiers dans ce répertoire, dont l'archive de IE7.
- o **Etape 4** exécutez maintenant le programme *IE7 Standalone Setup.bat* et cliquez sur OK pour installer xmlitesetup.exe. Une mise à jour sera aussi déclenchée automatiquement durant le setup afin d'installer les tabs à IE7.

Voir [l'article de John](#) pour plus de détails (en anglais)

## J'ai mis Windows à la porte de chez moi

Au moins 10 ans de vie commune, vous imaginez ? La semaine dernière j'ai mis Windows dehors. Hop j'ai pris ma grosse unité centrale, le Vista Home Premium machin installé dessus, et zou, dehors. A la place dans le salon, l'iMac 20" acheté à Bordeaux Surcouf pendant les vacances. Rien à dire, après avoir réalisé ce week-end mon premier montage iMovie et avoir gravé avec iDVD, je réalise enfin le temps que je vais gagner. Avec 2 enfants en bas âge, j'ai passé l'âge de bidouiller soit avec Windows, soit avec Linux. Pas le temps de chercher un patch pour ma carte wifi Belkin neuve qui ne marche pas sur Vista. Pas envie de bloquer ma chère et tendre épouse en lui imposant Linux, même si j'ai fait une tentative avec Ubuntu il y a un an. Elle n'a pas accroché du tout. Bref nous sommes allés tout les deux à Surcouf afin de prendre un portable ou un imac. J'avais déjà en tête une idée du Mac, mais j'ai vraiment craqué pour la simplicité, la puissance et le rapport qualité-prix. Nous nous sommes donc offert un iMac 20" avec 2Go de mémoire, un processeur à 2.4 Ghz et un disque de 300Go. Mon épouse a adopté littéralement la machine. Que ce soit pour

écouter des musiques, utiliser **Mail** ou **Safari** elle adore. Et elle s'en sert ! Bref pour l'instant nous sommes vraiment contents.

Je vois ce matin un article intéressant sur le blog [de Jean-Francois Helie](#) qui s'est acheté un Mac Book Pro (portable). Il donne des liens intéressants pour les switchers, les personnes qui passent de Windows à Mac dont le site <http://www.debutersurmac.com/> qui liste un grand nombre de logiciel, dont de quoi ripper un DVD ou encoder pour ipod une vidéo.

## Petites réflexions autour du client riche, l'heure du bilan

**Après avoir réalisé une maquette avec [Adobe Flex](#) j'ai testé il y a quelques jours la version Beta 2 de Microsoft Expression Blend. Quel est le positionnement de ces technologies aujourd'hui ? Le client léger basé sur les technologies du web (html,css,javascript) a-t-il encore un avenir ? Qu'en est-il de l'architecture service ?**

Le coeur de mon métier, c'est le client riche. Avec mon équipe nous nous occupons d'un framework java propriétaire qui fonctionne en client léger dans IE ainsi qu'en mode client lourd (swing). Aujourd'hui je me rends compte qu'il est encore difficile de faire accepter l'évolution de l'architecture qui est en place. Nous passons d'un client lourd installé sur le poste de nos clients à une architecture déployée J2EE dont l'interface fonctionne dans un simple navigateur. Nous avons fait de gros efforts pour que l'application web soit aussi interactive que possible. En quelques sortes, cela fait plus de 2 ans que nous faisons de l'AJAX avant même que ce terme ne soit publié il y a maintenant 2 ans.

Et honnêtement le résultat est là.

Des tableaux éditables qui s'exportent vers Excel ou PDF, des graphiques avec JFreeChart, des blotters de données du marché qui se mettent à jour en temps réel... le tout avec du HTML, du CSS, des Javascripts et des composants HTC. Bref on est au limite de ce que peut faire un navigateur tel qu'IE6 sans aucuns plugins. Mais ça marche !

Aujourd'hui il y a encore quelques blocages qui font que nous avons ressorti le client lourd des cartons (Swing). Il y a aussi un coût plus important pour développer une interface riche web qu'une interface classique car les technologies actuelles (css,html,js,htc...) si l'on peut parler de technologies, ne sont pas très bien maîtrisées ou enseignées.

Ensuite sachez que tout n'est pas rose dans le monde AJAX. Ne vous emballez pas trop vite pour toutes ces librairies javascript comme Scriptaculous ou Prototype. Certes, elles vous font gagner du temps. Mais malheureusement elles ne sont pas toujours exemptes de bugs. Attention aussi au navigateur de Microsoft. Je pense particulièrement à Internet Explorer 6 (avant le SP2) et son moteur javascript. J'ai rencontré 2 fois un problème de fuite mémoire en javascript. Très difficile à identifier.

Pour revenir au débat "client riche/client lourd", je vois 2 soucis: je pense qu'il faut maintenant se tourner vers des technologies plus avancées pour faire un vrai produit fonctionnant dans un navigateur. J'ai testé [Adobe Flex2](#) ainsi que [Microsoft Expression Blend](#). Ces 2 produits sont adaptés pour construire un produit avancé qui fonctionnera en mode client lourd comme en mode client léger. Pour la technologie d'Adobe, [Apollo AIR](#) permet de faire fonctionner une application Flex en mode client lourd. Une syntaxe XML permet de définir l'interface, le MXML.

Chez Microsoft, la Beta 2 d'Expression Blend m'a agréablement surpris. Elle permet de construire une application qui fonctionnera avec le .Net framework 3.5. Elle peut fonctionner en client lourd comme en client léger grâce à Silverlight.

[Adobe Flex 3](#) m'a semblé plus pratique au niveau des Layout Manager pour construire mon interface. Il est possible de définir un ensemble de composant et de les sauver dans sa boîte à outil afin de pouvoir s'en resservir. C'est aussi vrai pour Expression Blend. Du côté de Microsoft, l'éditeur est plus puissant. Il est moins facile à prendre en main mais la séparation entre l'interface et la partie dynamique est plus simple que chez Adobe. Je pense qu'il ne faut pas négliger ce que propose Microsoft. Nous arrivons à une solution mature pour développer à la fois en mode client lourd et en mode client léger.

Maintenant réveillons-nous un peu: je ne suis pas entrain de jeter aux orties tout le boulot web, et ce qui se passe derrière le décor. Je pense qu'il faut investir et commencer à regarder sérieusement Adobe Flex et/ou la couche Windows Presentation Framework avec Silverlight de Microsoft. Par contre je ne pense pas qu'il faille jeter l'architecture service J2EE pour autant. Et c'est l'un de mes soucis. On me demande de faire du client lourd. Du vrai client lourd 2-tiers avec la base de données et des services financiers déportés. Pour ma part, ma vision est que l'on devrait conserver un serveur d'application afin d'effectuer un certain nombre de services (profiling, authentification, autorisation...). Sur le serveur d'application, le coeur du produit continue à fonctionner. EJB3 ou pas, peu importe. En conservant une architecture Web+J2EE il reste possible de fonctionner en mode client léger dans un navigateur comme en mode client lourd déporté. Je pense que la méconnaissance de l'architecture J2EE nous conduit aujourd'hui à revenir en arrière, ce qui est vraiment dommage...

Deuxième réflexion: nous avons plusieurs produits. Nous offrons à nos clients une Suite qui regroupe tout ou partie de ces produits. Or si vous pensez en terme de client lourd, il devient rapidement compliqué de construire et livrer le binaire adapté à ce que le client a acheté. Pareil pour ce qui concerne les installations de patches... Avec 6 produits majeurs financiers, je peux vous assurer que vos équipes vont passer leur temps à mettre à jour tel ou tel produit... Mais bon, c'est envisageable.

En conservant une architecture 3 tiers, il est possible facilement d'ajouter ou de retirer tout ou partie d'un produit, car le coeur du produit n'est pas dans le client riche mais déporté sur le serveur d'application J2EE. Par ailleurs on concentrera les efforts sur l'ergonomie, l'interface du client, mais on se contentera d'effectuer des appels vers le serveur pour les traitements...

Voilà c'est tout... ce sont quelques idées qui me trottent dans la tête. L'arrivée des technologies de type client riche va bousculer une nouvelle fois nos connaissances d'architecture. Il faut penser maintenant à ce qui se passe derrière l'écran et la jolie fenêtre animé dans un écran Flex... Pensez-y si vous faites du client riche...

## Technorati, j'y arrive

Hop c'est décidé, en 2008 je fais un effort pour faire connaître mon blog. Mais comment s'y prendre ? J'ai commencé par [Technorati Profile](#). Ce site est très populaire dans la blogosphère. Il indexe plusieurs millions de blogs. Un moteur de recherche et de publication de flux RSS permet ainsi de s'abonner et de suivre l'actualité sur un sujet facilement. Ainsi dans Firefox il est possible de créer un favori dont le contenu est un flux RSS sur un sujet précis. Vraiment pratique

## Apple MacWorld Expo, les annonces possibles de Steve Jobs

Plus que quelques jours avant l'exposition MacWorld Expo. Le 15 janvier prochain, tout le monde suivra avec attention la présentation annuelle de Steve Jobs. [L'an passé](#), c'était l'annonce de l'iPhone. Cette année il y a plusieurs rumeurs ou pistes sur ce que Steve devrait annoncer.

En premier lieu, un mini mac. Basé sur l'idée d'Asus et du [eee PC](#) à moins de 300 EUR, les journalistes attendent un appareil léger avec en plus **un écran tactile** afin de remplacer la souris. Apple a déposé plusieurs brevets dans ce sens aux USA, et donc on peut s'attendre à l'arrivée de cette mini machine très prochainement.

Peut-être aussi l'iPhone 2 avec le support de la 3G pour les européens et le marché asiatique, bien plus en avance que les réseaux aux Etats-Unis. Cela risque peut-être de décevoir les nouveaux acheteurs de l'iPhone. Cependant c'est une étape quasi nécessaire pour attaquer le marché Japonais. Des rumeurs parlent d'un nouvel iPhone en juin chez nous autour de 500 EUR. A prendre au conditionnel.

Sinon d'autres rumeurs ce matin [en lisant le Guardian](#) parlent d'une mise à jour de Leopard ainsi que des portables MacBook qui ont été rejoint par Sony, Dell et Toshiba (voir SVM Mac en kiosque ce mois-ci). Vu l'échec d'AppleTV (à peine 400 000 clients), il est possible que Steve annonce un nouveau système pour la VOD (Vidéo à la demande) et/ou un système pour stocker ses vidéos personnelles sur un disque réseau distant. Clairement, on vise la mort du magnétoscope, pour le remplacer par un système avancé afin que notre télévision puisse lire nos vidéos le plus simplement possible.

Ce qui n'est pas forcément une réussite concernant l'iPhone, c'est ce système de blocage. En effet, une forte communauté de développeur fan est prête à développer de nouvelles applications pour l'iPhone. Apple afin d'éviter virus et problèmes, a fait le choix de fermer pour l'instant cette possibilité. Peut-être que nous aurons l'annonce d'un SDK dédié à l'iPhone et piloté par Apple.

Bref le 15 au soir vers 18h00, connectez-vous sur <http://www.svmmac.fr> pour suivre la Keynote de Steve Jobs... on ne sait jamais !

## Joomla, un gestionnaire de contenu pour votre site

[Joomla](#) est un système de gestion de contenu pour votre site web. Il est gratuit, simple et destiné à vous faciliter la vie. Basé sur PHP, Joomla permet de construire des sites Internets complexes (forum, système de vote, mailing...) sans taper une ligne de PHP ou d'HTML. Pour ma part, j'ai commencé à travailler avec la nouvelle version 1.5 RC4.

Vous avez besoin d'un serveur avec une base de données MySQL et le support PHP. Ensuite concernant le look, les templates permettent d'habiller d'un seul clic votre site. Et vous pouvez changer ensuite à tout moment d'habillage sans une ligne de code. Sous Mac, aucuns problèmes pour installer Apache+MySQL grâce à [MAMP](#) (Mac, Apache, MySQL et PHP). Sous Windows vous pouvez télécharger [XAMPP](#). Comptez quelques heures pour maîtriser Joomla si vous avez déjà une expérience avec Internet. Sinon tout au plus une semaine je pense.

**Concernant la compatibilité avec la version 1.0** vous pouvez activer le plugin **System - Legacy** afin de pouvoir utiliser un template 1.0 avec la version 1.5 de Joomla. Cela dit, les templates sont en fait une simple page PHP et quelques CSS.

Si vous utilisez Joomla et que vous cherchez des templates gratuits, je vous conseille les sites suivants:

- <http://templates.joomlafacile.com/>

- <http://www.siteGround.com>
- <http://www.joomlashack.com/>
- <http://www.joomladesigns.co.uk/>

Pour le Touilleur, je suis hébergé sur le site JRoller. Il a l'avantage d'être gratuit et bien référencé. Mais je pense qu'à terme je prendrai un hébergement dédié afin de pouvoir d'une part mon blog, et d'autre part un site plus classique pour le boulot. Voili voilà

## Conférence de Steve Jobs dans 2 heures... On parle d'un MacBook Air

Plus que quelques heures avant que Steve Jobs ne donne sa conférence annuelle à [MacWorldExpo](#) à San Francisco. Le site [AppleInsider](#) reprend un article de Wired qui présente le **Mac Air**, toujours autour de rumeurs sur un nouveau portable ultra-fin proposé par Apple.



Le site [AppleInsider](#) propose de suivre en temps réel la conférence de Steve Jobs à partir de 18h00 ce soir. Voir [ce lien](#) et sinon pour suivre la KeyNotes ce soir, vous pouvez aller sur le site de [SVM Mac](#).

## Présentation de Steve Jobs 2008 (en direct)

Soyons fou: je suis connecté à 5 sites en anglais qui retranscrivent la conférence de Steve Jobs qui a commencé il y a quelques minutes. Je vais vous faire suivre ici ce que je lis en live...



Pour nous faire patienter... **Note: ceci est une transcription en live avec donc ses fautes de frappes et d'orthographe...**

- 18h15 c'est parti...
- 18h15 La pub avec les 2 personnes représentant le PC et le Mac apparaît à l'écran. PC :

"Mac, 2008 sera l'année du PC !" Mac: "Ah bon ? et que comptes-tu faire?" PC: "Et bien faire tout ce que toi tu as fait en 2007" Tonnerre d'applaudissement dans la salle !

- o 18h16: Entrée en scène de Steve Jobs. Steve dit que l'année 2007 a été formidable pour Apple
- o 18h18: Jobs dit que 20% des Mac installés fonctionnent avec Mac OS X 10.5 Leopard. C'est un succès
- o 18h19: "Nous avons vendu 4 millions d'iPhone. Aujourd'hui cela fait 200 jours que l'iPhone est en vente. Soit 20 000 ventes par jour..."
- o 18h20: Jobs parle de Time Machine: "...Vous vous dites que c'est embêtant de devoir avoir un deuxième disque dur pour Time Machine... Ok nous vous avons entendu... Maintenant vous l'avez et c'est **Time Capsule**. C'est un système de sauvegarde déporté. Cela fonctionne via AirPort Extreme, c'est une station avec un disque interne (SAS) ainsi qu'un disque dur de qualité serveur. Vous pouvez backuper votre notebook dessus sans problèmes. Une prise électrique et ça marche..."
- o 18h21: Deux versions: une avec 500 GB et une avec 1 TB. Prix: 299\$ et 499\$ (c'est tres

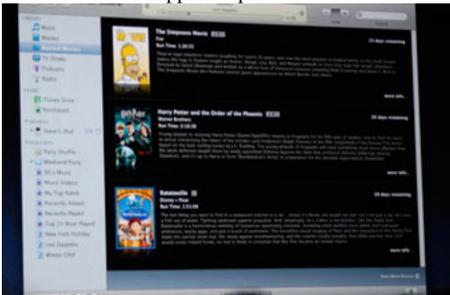


agressif) à partir du 1er février.

- o 18h23: (je zape le speech sur les parts de marchés de l'iPhone...)
- o 18h24: Steve Jobs: "...Tout le monde attend maintenant le kit de développement pour l'iPhone. Il arrivera **fin février**. Mais nous voulions donner à tout le monde quelque chose aujourd'hui. Je vous annonce la disponibilité de plusieurs nouvelles applications pour l'iPhone. Quelques possibilités: cartes avec geolocalisation, clips webs, page d'accueil personnalisées, sms envoyés à plusieurs, support pour les vidéos, les chapitres, les sous-titres... etc"
- o 18h29 Démo de la nouvelle application **Maps** pour iPhone. Application GPS avec le look apple. Ensuite démo SMS à plusieurs. Webclips...
- o 18h30 Pour l'iPod Touch de nouveaux logiciels sont annoncés: Mail, Maps, Stocks, Notes et Weather. Maps fonctionnera avec le wifi pour localiser l'iTouch. Webclips et tout ce qui est vidéo, identique à l'iPhone... Pour les clients actuels de l'iTouch la mise à jour sera à 20\$



- o 18h31 Mise à jour de l'iPod touch via l'iTunes, et donc on va parler de l'iTunes store maintenant.
- o 18h32 Annonce de la possibilité d'acheter des films via l'iTunes, **2.99 \$** le film sur l'iTunes Store et **3.99\$** pour les nouveautés. On ne sait pas si pour l'Europe il y aura quelque chose...
- o 18h40 30 jours après la sortie en DVD on pourra voir le film sur l'iTunes Store.
- o 18h42 Version internationale de l'iTunes Store s'engage pour 2009. Youpi pour l'Europe, nous aurons aussi iTunes store + video l'an prochain
- o 18h44 Il parle du Tivo, de Microsoft et finalement il annonce une version 2 de l'appleTV. Plus besoins d'un ordinateur. Il sera possible de louer des films directement avec l'appleTV comme c'est le cas avec la Freebox. On pourra faire de la HD... Il parle de voir des films, on ne sait pas s'il sera possible de louer...
- o 18h45: Location de film HD à partir d'un dollar. 3.99\$ pour une librairie de titre et donc 4.99\$ pour les nouveautés. Plus de 100 titres seront disponibles. Nous allons voir aussi l'interface de l'appleTV
- o 18h48 Démo de l'appleTV par Steve Jobs



- o 18h49: Tout le monde regarde un film en vidéo HD joué sur le grand écran... on attend la fin

pour les meilleurs annonces...

- o 18h55: Conclusion de la démo de l'AppleTV avec PodCast en HD... Tiens ça ressemble à Free HomeVideo. Apple veut que la solution soit autonome. Pas besoin d'un ordinateur, d'un PC ou même d'un Mac. Cette partie de la présentation est assez longue...
- o 18h57: Après le podCast, les photos en HD... nous pouvons voir des poissons... Partage avec iMac
- o 18h59: Une annonce...
- o 19h00: La mise à jour de l'Apple TV sera **GRATUITE**.
- o 19h00: **Le PRIX de l'AppleTV sera 229\$** mais on ne peut pas enregistrer...
- o 19h02: Steve Jobs amène une transition... on se demande ce qu'il veut annoncer.
- o 19h04: Le PDG de 20th Century monte sur scène afin de parler avec plus de détails de l'appleTV et de l'accord avec Apple pour que les films de cette major soient disponibles sur l'AppleTV.
- o 19h08: Le patron de la Fox s'en va... on passe à la suite
- o 19h09: On entend dans la salle que l'ambiance monte, changement de lumière. Steve Jobs disparaît un instant et va chercher quelque chose.
- o 19h10: On parle de MacBook... "Après le MacBook, le MacBook Pro, voici le **MacBook AIR**. C'est tout simplement l'ordinateur le plus fin du monde."
- o 19h11: Il compare avec une machine de Sony... L'épaisseur sera de 0.76 pouces... (1.9 cm !!!) et la partie la plus fine sera aussi fine qu'une enveloppe. Steve Jobs fait la démo en mettant le MacBook AIR dans une enveloppe. Face avant en aluminium et le dos noir. Ecran 13.3 pouces.



- o 19h12: Système magnétique pour ouvrir/fermer. Ecran OLED. Le clavier sera aussi éclairé.
- o 19h13: Clavier full-size qui ressemble à celui de mon iMac :-). Et il est rétro éclairé...
- o 19h14: Arrivée d'un trackpad multi-point. Rectangulaire. Steve Jobs fait ensuite une démo. Deux-doigts en parallèle sur le track-pad, vous écartez les doigts et ça zoome !
- o 19h15: Pour passer d'une photo à l'autre, vous faites glisser vos doigts sur le pad... Comme l'iPhone ! Cette technologie est maintenant disponible pour le Mac AIR.
- o 19h16: Disque 80Go en 2.5 pouces. Une version Flash sera aussi disponible en option... La carte mère a la taille d'un iPhone. Le processeur Intel sera de 1.6 Ghz et 1.8 Ghz.
- o 19h18: Il passe la parole à une personne d'Intel: Paul Otellini "on a été contacté il y a un an pour faire une puce qui tienne dans cet appareil... Et ce sera une exclusivité pour Apple pour l'instant".
- o 19h21: Steve Jobs: "ok, le disque dur est peut-être un peu petit non ? Et vous avez Times Capsule pour sauver vos fichiers non ? Et là je vous annonce que je retire le lecteur de DVD. Il reste disponible cependant en option..." Pas de lecteur de média dans l'appareil par défaut. Il semble qu'il sera possible de brancher un graveur externe via les ports USB/Firewire. Mais Steve Jobs parle d'une nouvelle techno pour monter et lire à distance via Wifi un DVD.
- o 19h22: "Il sera possible de monter un DVD sur un iMac. Le disque sera accessible via Wifi à partir du bureau du Mac AIR. C'est de cette manière par exemple que vous pourrez installer des applications"
- o 19h24: "5H d'autonomie, un trackpad touch, une webcam intégrée, 2 Go de mémoire..."



- o 19h25: Une publicité passe... on voit le MacBook AIR qui sort d'une enveloppe.
- o 19h26: 1799\$ disponible dans quelques semaines
- o 19h26: Steve Jobs revient sur la partie écologique du Mac Air. Il n'y a pas de trucs toxiques, de mercures. Apple a pris note des remarques de Greenpeace.
- o 19h27: 56% plus petit qu'un MacBook: c'est moins de déchet
- o 19h28: Le slogan: "Le Mac le plus fin du monde"
- o 19h29: Steve Jobs termine sa présentation en reprenant les annonces.
- o 19h30: ... Concert privé de ??

**FIN DE LA TRANSCRIPTION** je rends le micro... Merci et A+

- o Retrouvez en anglais la transcription en anglais ici
- o [660 personnes ont suivi sur le site SVMMac la KeyNote](#). Bravo à l'équipe SVM Mac.

## Développer en Java avec un Mac et Leopard:

## il y a tout

[Update: ce post a été édité et réécrit après une nuit de sommeil]

J'ai hésité à titrer ce post "SVN,MVN et VPN" car ce sont les 3 mots qui définissent ce soir ma soirée.

### Comment accéder à un réseau VPN sécurisé avec une clé RSA SecureId à partir d'un Mac ?

Il existe plusieurs solutions. Tout d'abord Mac OS X 10.5 propose un client VPN en standard. Cependant je n'ai pas réussi à le faire fonctionner avec mon système d'authentification. C'est pourquoi j'ai installé [Apani Contivity VPN Client](#) pour Mac OS X. Ce produit commercial fonctionne très bien et permet d'accéder à un VPN tout en utilisant une clé RSA SecureId.

### Leopard dispose-t-il d'outils comme SVN (Subversion) ou Maven ?

Par défaut il faut savoir que Leopard est livré et configuré avec un bon nombre d'outils de développement. J'ai été agréablement surpris de voir que svn et mvn sont déjà configurés avec Leopard. La version de maven est assez récente (2.0.6) et je suppose que l'utilitaire Ant est aussi installé. C'est vraiment un point fort de Leopard: pas besoin d'effectuer d'installations d'outils pour commencer à coder en java. Java est aussi déjà installé, il n'y a rien à modifier pour tester.

### Quel éditeur utiliser ?

Eclipse n'est pas installé en standard, et à part vi ou TextEdit, il n'y a pas d'éditeurs dédiés à Java. Cela dit je vais en profiter pour vous parler d'un éditeur moins connu et pourtant vraiment bien: [IDEA Intellij](#).

J'ai installé la version Mac d'Intellij 7. Franchement si vous aimez Eclipse, et que vous vous plaignez de sa lenteur/lourdeur/mocheté alors je vous conseille Intellij. Il est capable d'ouvrir les projets Eclipse ainsi que d'importer ses plugins. Il est possible de configurer les raccourcis d'Eclipse, ce qui permet de retrouver ses réflexes rapidement.

S'agissant d'un logiciel commercial, en effet il est payant. J'en profite pour donner mon avis sur les réflexions "franco-français" du style "il est payant". Comme le disait un de mes collègues ce midi, dans le monde C++ ou Microsoft tout est payant et personne ne s'en offusque. Au nom de quoi devrions-nous hésiter à payer une licence pour un éditeur Java qui est performant ? Le gratuit n'est pas forcément la meilleure solution industrielle. En terme de productivité, il n'y a pas qu'Eclipse mais aussi d'autres outils souvent plus performant. En tant que chef d'équipe je préfère investir le coût d'une licence pour un outil puissant que de compter les heures perdues par mes développeurs à bidouiller avec Eclipse et ses plugins.

Je referme la parenthèse sur Eclipse et cet avis n'engage que moi. Je ne dis pas qu'Eclipse c'est de la m... car c'est un excellent éditeur. Intellij est fait en partie par d'anciens développeurs d'Eclipse. Mais c'est un outil plus puissant et plus stable qu'Eclipse pour le développement Java.

## Changement d'hébergeur

J'ai commencé à migrer mes articles de mon ancien site vers celui-ci.

## Hibernate Shards: partitionnement horizontal des données

Dans la finance, les progiciels financiers gèrent des volumes de données énormes. Et souvent pour des raisons de sécurité, les données doivent être réparties sur plusieurs sites physiques. Imaginez si une banque privé ou un hedgefund perdait ses données dans un incendie... Chez nous, notre produit historique est capable de gérer des données multi-sites (multi entities en anglais). Depuis que nous avons commencé à migrer vers une architecture SOA basée sur Java, l'accès aux données s'effectue via un ORM (Hibernate). J'ai trouvé ce matin sur le blog de Xebia qu'Hibernate dispose d'un module permettant d'adresser le partitionnement des données horizontales: [Hibernate Shards](#).

Hibernate Shards (fragements en anglais) était à l'origine un projet de Google. C'est une extension d'Hibernate Core conçu pour encapsuler et minimiser la complexité de travailler avec des données fragmentées (horizontalement partitionnées). L'API est assez simple et nécessite Java 5 minimum.

Si vous connaissez les interfaces par défaut d'Hibernate qui sont:

- `org.hibernate.Session`
- `org.hibernate.SessionFactory`
- `org.hibernate.Criteria`
- `org.hibernate.Query`

Hibernate Shards ajoute les objets suivants qui permettent d'interagir avec les données:

```
org.hibernate.shards.session.ShardedSession
```

- `org.hibernate.shards.ShardedSessionFactory`
- `org.hibernate.shards.criteria.ShardedCriteria`
- `org.hibernate.shards.query.ShardedQuery`

Cependant chaque application Java étant architecturée différemment, Shards propose aussi des classes supplémentaires qui vous permettront de mettre en place votre stratégie de partitionnement personnalisée.

- `org.hibernate.shards.strategy.selection.ShardSelectionStrategy`
- `org.hibernate.shards.strategy.resolution.ShardResolutionStrategy`
- `org.hibernate.shards.strategy.access.ShardAccessStrategy`

Hibernate Shards permet aussi de mettre en place ce que l'on appelle la "refragmentation" (resharding). Imaginez que vous créez votre site de ventes aux enchères ou de rencontre. La base de données explose car votre site a du succès. Grâce à ce concept vous pourrez mettre en place des bases supplémentaires sans trop de soucis... Attention c'est loin d'être facile mais c'est faisable.

## Avancement de la migration de l'ancien blog vers celui-ci

C'est parti... après avoir configuré et créé le look du Touilleur Express 2008, j'ai commencé à migrer [mes anciens articles de JRoller](#) vers ce nouveau serveur. Je ne me souvenais pas que j'avais commencé en septembre 2003. Je pensais que c'était un peu plus tard. Bref j'ai repris tout les articles entre 2003 et 2008, soit presque 134 billets. Il y en a encore presque 60 entre janvier 2004 et début 2005 à reprendre.

C'est amusant de se relire. Au début du Touilleur, j'essayais de parler vraiment que de Java. Je donnais ainsi quelques lignes de code par ci par là pour épater la galerie. En 2004 j'ai pas mal blogué à propos de Linux et de Mandrake. C'était ma période Linux 8h par jour.

En 2005 je vois que je parlais déjà de JSF et d'Ajax vers mai-juin 2005. C'était à la fin de mon premier poste, avant de rejoindre mon équipe actuelle en tant que chef de projet. Septembre 2005, c'est un changement: je passe dans ma nouvelle équipe et je deviens chef de projet. J'ai relu de vieux posts de fin 2005 sur le client riche, les EJB3 et certains sont encore d'actualité je trouve.

2006 est marqué par JBoss et JBoss Seam. C'est plus un projet perso qu'en rapport direct avec mon travail. Je blogue aussi sur Ajax, JSF, Java, J2EE et encore et toujours JBoss.

2007, toujours du JBoss Seam et un peu de Google.

Et depuis début 2008, j'achète un iMac... Et là vous avez dû vous rendre compte qu'en ce moment je suis à fond Mac...

J'aimerais parler de client riches cette année. Un peu d'Adobe Flex ou de Microsoft Silverlight. Un peu de Swing, et plus particulièrement de Java FX. Du côté architecture, j'en parle très peu. Dommage car je devrais prendre le temps de parler web services, soa, esb... mais c'est trop loin de ce que je fais en ce moment. Dommage.

Sur ce, bonne lecture, je retourne à ma migration d'article (je dois les faire un par un...)

## Astuces pour identifier la cause d'une java.io.NotSerializable et tuning JBoss

**L'exception java.io.NotSerializable se rencontre tard le soir au détour d'une petite période d'inactivité avec votre belle application J2EE... Le serveur essaye de passer un EJB et boum, voici notre belle exception qui sort du bois. Vous pouvez aussi rencontrer notre amie alors que vous essayez d'envoyer par le réseau un objet java. Nous allons voir comment dompter la bête et aussi facilement identifier ce qui n'est pas sérialisable. Je commence par faire un rappel sur la sérialisation, et ensuite je parlerai de JBoss pour vous montrer comment configurer celui-ci pour identifier les problèmes de passivation avec les EJB.**

Rappel rapide sur la sérialisation en Java: dans le domaine de la programmation, du stockage de données et de la transmission par le réseau, la sérialisation consiste à sauver un objet Java au format binaire. Il est ensuite possible de transmettre cette objet via le réseau, ou de le sauvegarder sur le disque par exemple. Les données binaires respectent un format spécial qui permet à Java de recréer un Objet identique avec le même état que l'original. En anglais, sérialiser un objet se dit "marshalling" et le désérialiser se dit "unmarshalling".

Avant tout, certains objets en Java ne sont pas sérialisables: Socket, Thread, OutputStream ou

Image par exemple. Cela peut poser un problème à votre application: que se passe-t-il si votre code contient une instance de l'une de ces classes non sérialisable ? Java vous le signale avec l'exception java.io.NotSerializable. Il existe cependant des solutions à ces problèmes.

## La sérialisation par défaut

Pour persister un objet à l'exécution, nous devons le rendre sérialisable. Java fait appel à un design pattern appelé "Marker". Ce design pattern consiste à créer une Interface java vide sans méthodes afin de marquer une class Java comme étant capable d'effectuer un certain traitement. J'aime bien demander en entretien d'embauche: "Peut-on avoir une Interface sans méthodes, sans instances ? Si oui, à quoi cela sert-il ?"

Prenons une class SoccerPlayer qui vient de mes cours sur java:

```
import java.util.Date;
import java.io.Serializable;

/**
 * Created by IntelliJ IDEA.
 * User: nicolasmartignole
 * Date: 22 janv. 2008
 * Time: 23:07:31
 */
public class SoccerPlayer implements Serializable {
    private String name;
    private Date birthDate;
    private int number;

    public SoccerPlayer(String name, Date birthDate, int number) {
        this.name = name;
        this.birthDate = birthDate;
        this.number = number;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Date getBirthDate() {
        return birthDate;
    }

    public void setBirthDate(Date birthDate) {
        this.birthDate = birthDate;
    }

    public int getNumber() {
        return number;
    }

    public void setNumber(int number) {
        this.number = number;
    }
}
```

En marquant notre classe avec l'interface java.io.Serializable, celle-ci est prête à être sérialisée. Nous allons écrire un peu de code pour tester en l'état la sérialisation de cette class.

```
import java.util.Locale;
import java.text.DateFormat;
import java.io.FileOutputStream;
import java.io.File;
import java.io.ObjectOutputStream;

/**
 * This class marshalls the SoccerPlayer class to a file.
 * User: nicolasmartignole
 * Date: 22 janv. 2008
 * Time: 23:11:07
 *
 * See online Touilleur Express http://www.touilleur-express.fr
 */
public class FileSystemStorageEngine {
    private static DateFormat df = DateFormat.getDateInstance(DateFormat.SHORT, Locale.FRANCE);

    /**
     * Creates a new player, performs a serialiation test.
     * @param args is cl args.
     * @throws Exception for any Exception, I don't want to deal with it in my sample.
     */
    public static void main(String[] args) throws Exception{
        SoccerPlayer player = new SoccerPlayer("Nicolas Martignole", df.parse("24/09/1975"), 10);
        FileOutputStream fos=new FileOutputStream(new File("test.ser"));
        ObjectOutputStream out=new ObjectOutputStream(fos);
        out.writeObject(player);
    }
}
```

La class ObjectOutputStream est la classe qui se charge de sérialiser notre class SoccerPlayer vers

le fichier. Si la class SoccerPlayer n'était pas marquée avec l'interface Serializable, Java lève une "java.io.NotSerializable" exception. La class SoccerPlayer est simple, et ses variables d'instance (une String et une Date) peuvent être sérialisées par Java sans problèmes.

## Désérialiser

La désérialisation s'effectue en utilisant la class ObjectInputStream. Dans l'exemple ci-dessous, je relis le fichier test.ser et je vais afficher le nom du joueur dans la console.

Nous modifions un peu notre class afin d'appeler une méthode chargée de désérialiser notre fichier. Dans mon exemple je ne fais aucuns traitements d'erreur, et donc si le fichier n'existe pas le programme lèvera une FileNotFoundException. Mais ici le plus important c'est juste de montrer qu'il est simple de recharger un Objet Java en mémoire.

```
public static void main(String[] args) throws Exception{
    SoccerPlayer loaded=reloadSoccerPlayer();
    System.out.println("Soccer player name: "+loaded.getName());
}

private static SoccerPlayer reloadSoccerPlayer() throws IOException, ClassNotFoundException {
    FileInputStream fis=new FileInputStream(new File("test.ser"));
    ObjectInputStream in=new ObjectInputStream(fis);
    SoccerPlayer sp=(SoccerPlayer)in.readObject();
    return sp;
}
```

L'exécution de mon code sur Mac donne

```
/System/Library/Frameworks/JavaVM.framework/Versions/1.5.0/Home/bin/java -Dfile.encoding=MacRoman -classpath /System/Library/Frameworks/Java
```

```
Soccer player name: Nicolas Martignole
```

```
Process finished with exit code 0
```

## Lorsque la class n'est pas sérialisable

Nous allons parler de l'exception java.io.NotSerializable. Mais avant tout, je dois continuer à rappeler quelques mécanismes de Java.

Reprenons notre class SoccerPlayer et modifions-là afin qu'elle ne soit pas sérialisable. Je vais créer un objet "PlayerPosition" qui représente sur le terrain le poste du joueur. Cet objet ne sera pas sérialisable car il est dynamique. Un joueur défenseur peut jouer ailier (enfin je crois) et donc nous imaginons qu'il n'est pas possible de sérialiser cette information. Peu importe le code de la class SoccerPosition, nous avons simplement besoin d'avoir une instance de celle-ci dans la class SoccerPlayer. La class SoccerPosition n'implémente pas l'interface java.io.Serializable.

La class PlayerPosition est très simple:

```
public class PlayerPosition {
    String positionName;

    public PlayerPosition(String s) {
        positionName =s;
    }

    public String getPositionName() {
        return positionName;
    }

    public void setPositionName(String positionName) {
        this.positionName = positionName;
    }
}
```

Nous ajoutons une instance dans SoccerPlayer:

```
public class SoccerPlayer implements Serializable {
    private String name;
    private Date birthDate;
    private int number;

    private PlayerPosition position;

    public SoccerPlayer(String name, Date birthDate, int number, PlayerPosition p) {
        this.name = name;
        this.birthDate = birthDate;
        this.number = number;
        this.position=p;
    }
}
```

Je modifie le constructeur de SoccerPlayer afin de passer une instance de PlayerPosition. Voyons maintenant lorsque l'on relance le programme pour sérialiser ce qui se passe:

```
public class FileSystemStorageEngine {
    private static DateFormat df = DateFormat.getDateInstance(DateFormat.SHORT, Locale.FRANCE);
    /**
```

```

* Creates a new player, performs a serialiation test.
* @param args is cl args.
* @throws Exception for any Exception, I don't want to deal with it in my sample.
*/
public static void main(String[] args) throws Exception{
    SoccerPlayer player = new SoccerPlayer("Nicolas Martignole",
        df.parse("24/09/1975"),
        10,
        new PlayerPosition("Goal")
    );

    performSerialization(player);

    SoccerPlayer loaded=reloadSoccerPlayer();
    System.out.println("Soccer player name: "+loaded.getName());
}

private static void performSerialization(SoccerPlayer player) throws ParseException, IOException {
    FileOutputStream fos=new FileOutputStream(new File("test.ser"));
    ObjectOutputStream out=new ObjectOutputStream(fos);
    out.writeObject(player);
}

private static SoccerPlayer reloadSoccerPlayer() throws IOException, ClassNotFoundException {
    FileInputStream fis=new FileInputStream(new File("test.ser"));
    ObjectInputStream in=new ObjectInputStream(fis);
    SoccerPlayer sp=(SoccerPlayer)in.readObject();
    return sp;
}
}

```

Avant tout, notez que si vous n'instanciez pas de PlayerPosition pour la class SoccerPlayer, tout va bien se passer. C'est pour cette raison que j'ai mis à jour le constructeur de SoccerPlayer pour forcer l'affectation d'une PlayerPosition.

Après avoir compilé, le programme plante:

```

Exception in thread "main" java.io.NotSerializableException: PlayerPosition
    at java.io.ObjectOutputStream.writeObject0(ObjectOutputStream.java:1081)
    at java.io.ObjectOutputStream.defaultWriteFields(ObjectOutputStream.java:1375)
    at java.io.ObjectOutputStream.writeSerialData(ObjectOutputStream.java:1347)
    at java.io.ObjectOutputStream.writeOrdinaryObject(ObjectOutputStream.java:1290)
    at java.io.ObjectOutputStream.writeObject0(ObjectOutputStream.java:1079)
    at java.io.ObjectOutputStream.writeObject(ObjectOutputStream.java:302)
    at FileSystemStorageEngine.performSerialization(FileSystemStorageEngine.java:34)
    at FileSystemStorageEngine.main(FileSystemStorageEngine.java:24)

```

Ici Java identifie facilement que la class PlayerPosition ne peut pas être sérialisée. Là où souvent vous aller avoir des soucis, c'est avec un serveur d'application comme JBoss ou Weblogic. Le mécanisme de passivation permet à un serveur d'application de libérer de la mémoire en stockant sur le disque des objets sérialisés. Lorsque la passivation échoue, les objets de la session EJB de l'utilisateur ne pourront pas être restaurés. D'où cette exception.

### **Avec un serveur J2EE et des EJB 2.1, comment retrouver quelle classe n'est pas sérialisable ?**

Voici l'astuce du jour (et l'objet de ce billet):

1. Lancer le serveur d'application en mode debug sur une socket
2. Se connecter via Eclipse ou IntelliJ sur le serveur J2EE en démarrant une session de remote debug avec Java 5.
3. Charger les écrans dans l'application Web en se connectant
4. Configurer Eclipse ou IDEA IntelliJ pour que le débogueur s'active lorsque l'exception java.io.NotSerializable est levée (voir plus bas)
5. Attendre que l'EJB se passive
6. Votre IDE doit alors se réveiller et vous êtes arrêtés à l'endroit où la Thread du serveur d'application est entrain de faire la sérialisation. Avec l'inspecteur d'objet d'IntelliJ il est facile d'identifier quelle classe n'est pas serializable

### **Sur JBoss la passivation d'un Session Statefull Bean est réglé à 10 minutes, je ne vais pas attendre aussi longtemps**

En effet, il est facile de configurer JBoss afin de passiver tout de suite des Beans. Je vais donner l'explication pour JBoss 4.2.2.

Ouvrez le fichier \$JBoss\_HOME/server/default/conf/standardjboss.xml avec votre éditeur de texte favori (vi). Dans ce fichier de configuration, la section //container-configuration// définit la configuration du conteneur. Chaque configuration spécifie des éléments tels que l'invoquer à utiliser, la politique d'interception de jboss, le nombre d'instances de caches et les files d'attente de thread, la configuration du persistence manager et enfin la partie sécurité.

Nous allons regarder plus en détail la section « container-cache-conf ». Cet élément est passé à l'implémentation de l'InstanceCache si celle-ci supporte l'interface XmlLoadable. Vous pouvez donc implémenter votre propre mécanisme de cache si vous le désirez. Les implémentations actuelles de Joss InstanceCache dérivent de la classe abstraite « org.jboss.ejb.plugins.AbstractInstanceCache ». Cette classe implémente l'interface XmlLoadable et donc sera capable de charger notre configuration.

Examinons le fichier standardjboss.xml

```
org.jboss.ejb.plugins.LRUEnterpriseContextCachePolicy
50
1000000
300
600
400
60
1
0.75
```

Deux implémentations possible de CachePolicy sont utilisées dans le fichier standardjboss.xml avec le support d'une configuration que nous allons voir dans un instant  
Les deux implémentations sont

- o org.jboss.ejb.plugins.LRUEnterpriseContextCachePolicy
- o org.jboss.ejb.plugins.LRUStatefulContextCachePolicy

Dans JBoss 4.2, les Entity beans utilisent LRUEnterpriseContextCachePolicy. Les stateful session bean utilisent LRUStatefulContextCachePolicy.

Voici ce qu'il est possible de configurer:

- o min-capacity nombre d'éléments minimum du cache
- o max-capacity nombre maximum d'éléments du cache
- o overager-period : durée en secondes entre 2 exécutions du système de nettoyage du cache. Lorsque ce système s'active, les beans dont l'âge est supérieur à max-bean-age sont retirés du cache pour être passivé. Vous voyez donc où nous allons en venir.
- o max-bean-age : donne le temps maximum en seconde Durant lequel un bean peut être inactive avant d'être passive.
- o resizer-period: durée en secondes entre 2 exécutions de la thread du "resizer". Cette tâche se charge de redimensionner le cache en se basant sur les 3 propriétés suivantes. Lorsque cette tâche s'exécute, elle vérifie le temps écoulé entre 2 caches miss et si cette période est inférieure à « min-cache-miss-period » alors le cache est agrandi à la valeur « max-capacity » en utilisant le facteur « cache-load-factor ». Si la période par contre entre 2 miss caches est plus grande que la valeur de "max-cache-miss-period » alors la taille du cache est réduite en utilisant la valeur de « cache-load-factor ». Ce système permet d'avoir un cache de beans adapté à la charge du serveur avec JBoss.

En plus des paramètres précédents, LRUStatefulContextCachePolicy pour les Session Stateful Bean a les éléments suivants

- o remove-period Spécifie une période en secondes à attendre entre l'exécution de la tâche de nettoyage du cache. La tâche « remove task » se charge de passer les EJB qui n'ont pas eu d'accès depuis « max-bean-life » secondes. Ce système différent permet de vider les EJB qui ont été passivés afin d'éviter de remplir par exemple le disque. Ce n'est pas le mécanisme de la passivation. C'est un mécanisme de nettoyage des beans sérialisés sur le disque et qui n'ont pas été réactivé depuis « max-bean-life ».
- o max-bean-life Spécifie la période maximale en seconde d'inactivité pour un bean avant que celui-ci ne soit détruit de l'espace de passivation.

## Comment voir si un EJB peut être passivé et restauré facilement avec JBoss ?

Nous arrivons enfin à l'explication, après avoir parlé sérialisation et JBoss. Editez la section container-cache-conf pour le container « Standard Stateful SessionBean ».

Vous pouvez limiter le nombre de beans en cache, et donc forcer la passivation pour les beans les plus anciens :

Ici je déclare que les beans doivent être passivés après 30 secondes (max-bean-age), et qu'ils ont une durée de vie de 1800 secondes(max-bean-life). Cela me laisse du temps pour tester l'activation après passivation. Toutes les 25 secondes je vérifie mon cache afin de voir ce que je dois passer (overage-period). Toutes les 120 secondes (remove-period) je regarde si je ne peux pas effacer du disque de vieux beans passivés. Je ne touche pas à ce qui est redimensionnement du cache (resizer-period).

```
org.jboss.ejb.plugins.LRUStatefulContextCachePolicy
10
30
25
30
400
60
1
0.75
120
1800
```

## Comment désactiver la passivation des EJB avec JBoss ?

C'est possible, quoique par forcément souhaitable. JBoss fournit une implémentation de

CachePolicy appelée org.jboss.ejb.plugins.NoPassivationCachePolicy.  
 Cette police ne passivera jamais une instance. Elle est basée sur une Map en mémoire qui ne retire jamais les beans sauf si ceux-ci sont explicitement retirés

Pour désactiver la passivation :

```
Standard Stateful SessionBean
    org.jboss.ejb.plugins.LRUStatefulContextCachePolicy
```

J'espère que ce petit guide vous sera utile, n'hésitez pas à me contacter si vous cherchez un expert JBoss pour faire du tuning ou si vous avez besoin de conseils pour votre application.

## serialVersionUID mythes et légendes

Suite à [mon article sur la sérialisation](#), je souhaite préciser quelques détails sur la sérialisation et parler du champ "serialVersionUID". Qu'est-ce que ce champ ? A quoi sert-il ? Faut-il le mettre ou non dans une class Serializable ?

### Quelques rappels:

Toutes les classes dérivées d'une class sérialisable sont aussi sérialisables. L'interface java.io.Serializable ne sert qu'à marquer la classe comme étant sérialisable comme on l'a vu. Cependant pour permettre à des types dérivés d'être sérialisable alors que leur super classe n'est pas sérialisable, il faut que la super classe déclare au moins un constructeur vide sans argument. De plus, la classe dérivée doit prendre en compte la sérialisation des champs protected ou public de sa super classe.

Lorsqu'une classe est désérialisée par Java, les champs de la super-classe qui ne serait pas sérialisable seront initialisés dans le constructeur vide sans argument. D'où son importance. Faute de cela, vous aurez une exception au runtime alors que votre classe fille est marquée comme étant Serializable... Ce n'est pas toujours visible au premier coup d'oeil. Des outils comme PMD sont capables de détecter ce type de problèmes et de vous aider.

Sachez aussi qu'il est possible de prendre en main le mécanisme de sérialisation/désérialisation en définissant 2 méthodes Java dans votre classe. Ce qui est intéressant à retenir, c'est que ces méthodes ne sont pas définies dans une interface... A votre avis pourquoi ? Il y a une raison très précise...

Les 2 méthodes sont:

```
private void writeObject(java.io.ObjectOutputStream out)
private void readObject(java.io.ObjectInputStream in)
```

Ne modifiez pas la signature de l'une de ses méthodes, Java utilise l'introspection pour les identifier lors de la sérialisation et désérialisation.

En général, le code d'une méthode de sérialisation lorsque la classe contient des membres non sérialisables ressemble à ceci:

```
private void readObject(ObjectInputStream in) {
    // Déserialise normalement la classe
    in.defaultReadObject();
    // Instancie un objet qui ne pouvait pas être sérialisé
    this.socket = new Socket();
}
```

Quoi d'autre ? Vous pouvez aussi avec Java 5 utiliser un autre mécanisme de sérialisation. Pour cela il faut définir 2 méthodes dans votre class:

```
public Object writeReplace() throws ObjectStreamException;
public Object readResolve() throws ObjectStreamException;
```

Ce mécanisme permet de retourner un objet autre que le stream par défaut lors de la sérialisation et de la désérialisation. Vous pouvez même marquer ces 2 méthodes en tant que private, voir la javadoc à ce sujet.

### A propos de serialVersionUID :

#### Qu'est-ce que le champ serialVersionUID ?

C'est une clé de hachage SHA qui identifie de manière unique votre Classe. Cela permet lorsqu'elle est sérialisée, de la marquer avec une somme de contrôle (checksum) pour que lors de la désérialisation, votre programme soit certain de la version de la classe Java qu'il manipule. C'est un gestionnaire de version de votre classe si vous préférez.

#### A quoi sert-il ?

Imaginez un serveur et un client qui s'échangent des objets sérialisés. Ce système permet de s'assurer que les versions des classes envoyées d'un côté, existent bien de l'autre côté. Si la version est différente, Java lève une java.io.InvalidClassException pour vous dire qu'il y a un souci de

version. Si je le déclare, est-ce que la sérialisation va plus vite ? Et non... c'est une des légendes urbaines en Java. Avant tout, le calcul de ce serialVersionUID n'est effectué que lors du chargement de la classe par le ClassLoader. Typiquement sur un serveur démarré, je ne pense pas que vous verrez un quelconque gain de performance... La sérialisation d'une classe est lente, ne confondez pas avec le calcul d'un checksum SHA, opération assez rapide.

### Comment le déclarer ?

Il faut que cette variable respecte à la lettre la signature afin que Java puisse la voir lors de la sérialisation et de la désérialisation. En général on utilise des Long, en prenant soin de mettre un bon nombre de chiffres, histoire de ne pas tomber sur la même clé que le voisin.

```
private static final long serialVersionUID = 354054054054L;
```

Notez le "L" à la fin de la ligne pour indiquer qu'il s'agit d'un long et éviter un cast fantôme. A noter que vous pouvez déclarer ce champ protected. C'est une très mauvaise idée, sauf à savoir ce que vous faites.

### Faut-il déclarer ce champ ?

Sur la nécessité de le faire, la réponse est non. Java se charge de calculer le serialVersionUID lors du chargement de la class pour sérialiser/désérialiser. C'est donc fait automatiquement et vous n'avez pas à le déclarer pour que la sérialisation fonctionne. C'est pour cela que je m'en suis passé dans [l'article d'avant-hier](#).

Maintenant sur l'opportunité, voici ce que dit la spécification Java 5 sur la sérialisation: *Note - It is strongly recommended that all serializable classes explicitly declare serialVersionUID values, since the default serialVersionUID computation is highly sensitive to class details that may vary depending on compiler implementations, and can thus result in unexpected serialVersionUID conflicts during deserialization, causing deserialization to fail.*

### Que se passe-t-il si je le déclare ?

Je vous recommande de déclarer ce champ si vous savez ce que vous faites. En spécifiant ce tag, vous prenez en quelque sorte la responsabilité de gérer la version de votre classe. Il existe pleins de cas que j'ai rencontré où cela facilite la vie. Par exemple avec JBoss. Lorsque vous travaillez avec une arborescence explosé de votre EAR, et que vous ne modifiez qu'une partie, vous êtes souvent obligé de forcer ce tag pour être tranquille. Sans cela, JBoss et le ClassLoader (qui est lui aussi versionné, j'en parlerai une fois) va jeter des NoClassDefFoundError... difficile de faire le lien avec la sérialisation mais pourtant c'est le cas... C'est aussi pratique en phase de développement, lorsque vous travaillez sur la classe qui doit être sérialisée dans une architecture client-serveur avec RMI. Cela permet par exemple de déployer une nouvelle version de la classe sur le client 1 et ne vous force pas à mettre à jour le client 2 distant. Sans cela, à chaque nouvelle version vous seriez obligé de livrer la même version à vos différentes JVM ce qui n'est pas forcément possible. La Javadoc recommande de déclarer ce champ, car sa valeur est très sensible aussi au compilateur qui génère ce nombre. **Afin de garantir que la valeur d'un serialVersionUID soit constante quelque soit le compilateur, il est recommandé de le fixer une fois pour toute.**

### Mini quizz pour tester ses connaissances

Pour terminer, quelques petites questions:

- o un champ static est-il sérialisable ?
- o un champ volatile est-il serialisable ?
- o Soit une classe Player qui implémente Serializable. Est-ce qu'une inner-class dans Player doit aussi implémenter cette interface ? Est-ce que l'inner-class sera aussi sérialisée ?

### Références:

Article Java World "[Into the mist of serialization](#)"  
["Java Object Serialization" version 1.5.0](#)

## Trouver un nouveau job: les sites Web 2.0

**Je vous propose de retrouver une série d'article sur les meilleurs sites Web 2.0. L'idée est de vous présenter des sites novateurs, facile à utiliser et qui utilisent des technologies Web 2.0.** Cette semaine je vous parle de Jobfox.com, Notchup, Emurse, Gigtide et Resolio sur le thème des nouvelles techniques de recherche d'emploi qui nous arrivent des USA et de Grande-Bretagne.

### I. Jobfox.com

**Le principe:** vous vous inscrivez et définissez une liste de vos compétences. Le moteur notifie les employeurs et les chasseurs de têtes de vos compétences. Les entreprises qui viennent consulter votre profil voient visuellement vos points forts et vos sujets de prédilections. Par ailleurs, vous êtes notifié par SMS que l'entreprise ZeWorldCompany a visualisé votre profil. Le site suit le standard de ce que j'appelle un site Web 2.0. La page d'accueil donne toutes les informations d'un seul coup d'oeil, sans qu'il soit nécessaire de scroller. Les utilisateurs habitués identifient immédiatement où s'authentifier. L'inscription prend 5 secondes dès la page d'accueil. Intelligent et

simple. Une fois authentifié vous pouvez télécharger votre CV ou remplir votre profil. Au niveau techno, c'est du Microsoft .NET ASP. J'ai commencé à remplir mon profil mais évidemment c'est complètement dédié au marché nord-américain. Les points positifs: excellente navigation dans le site, l'utilisateur est guidé pas à pas. Seul l'information importante est affichée. Il n'y a pas plusieurs points d'entrées dans le site, c'est fluide et bien pensé. Au niveau techno, l'utilisation d'ajax (ajax.net ?) rend la saisie des formulaires agréable et rapide. J'ai par contre réussi à planter le site en entrant des accents dans une zone de texte... **Les principes Web 2.0:** Vous ne postulez pas à un job, ce sont les entreprises qui viennent à vous. Vos compétences sont représentées sous la forme d'un arbre graphique. En remplissant des questionnaires vous pouvez affiner votre profil, faire un mini-bilan de compétences et aussi donner à des recruteurs potentiels une idée de votre profil. Cette information est contrôlée et si vous ne souhaitez pas publier une donnée, libre à vous de ne pas l'afficher. Pour ma part j'ai passé presque 30mn pour remplir les différents questionnaires techniques. Cela permet de générer un arbre de vos compétences techniques... Cela dit il n'y a pas JBoss par exemple, ce qui fausse un peu mon "Experience Map" Exemple créé rapidement avec mon profil (je n'ai pas tout rempli): <http://www.jobfox.com/people/nmartignole>. L'url <http://www.jobfox.com> En conclusion, quelques nouvelles idées qui peuvent intéresser les recruteurs.

## II. Notchup, être payé pour passer un entretien

Notchup est un site internet qui vous propose d'être payé pour passer un entretien.

1. Si vous êtes actuellement content de votre poste
2. Si vous n'avez pas envie de changer de société
3. Si vous pensez être bien payé.... dites, vous êtes pas bien vous...faut consulter...

**alors Notchup est pour vous !** En effet l'idée de ce site est de mobiliser et d'enregistrer les gens qui n'ont pas envie de changer de travail mais qui seraient prêt à aller passer un entretien, en échange d'une indemnité. Imaginez recevoir des offres et des propositions que les chasseurs de têtes seraient obligés de vous payer... "*Mr Martignole nous aimerions bien vous rencontrer, nous avons vu que vous vendez 100 EUR l'interview. Nous sommes intéressés car nous voulons vraiment vous rencontrer...*". Bizarre non ? En même temps si nous regardons le marché de l'informatique, et en particulier tout ce qui est Java et .NET, nous sommes d'accord pour dire qu'en ce moment, il est difficile de trouver des profils intéressants. Alors pourquoi pas ? Si vous souhaitez vous inscrire ou tester ce service, il faut être invité. Envoyez-moi un email avec "demande notchup" dans le sujet (et uniquement cela) et je vous donnerai une invitation. Ce qui est assez génial, c'est que Notchup vous permet d'importer votre profil [LinkedIn](#) ! Très pratique afin d'éviter de devoir tout saisir à nouveau. Bref me voilà inscrit... et je viendrai avec plaisir à un entretien pour 200 \$ d'après l'outil de calcul de Notchup... Ah la la... la finance et l'informatique c'est un sujet qui intéresse les recruteurs.

## III. Outils de création de CV (resume) en ligne

Lorsque j'ai reçu des CV pour des candidats l'an passé, c'était un peu du n'importe quoi. J'ai eu droit au fichier Word de 3Mo (à se demander ce qu'il y avait dedans), au fichier RTF limite ASCII Art, à "j'ai-pas-eu-le-temps-mais-regardez-sur-viadeo" et aussi à mon spécial "CV à la Monster" (illisible). Le spécial "J'ai pas le temps regardez sur viadeo" c'était plutôt "*ché pa si g word, sinon c sur ww.viadeo.com. merci je vous apelle*" (candidat à un stage qui nous a contacté le 13 décembre) Et moi de répondre "lol, je viens de jeter votre mail à la p\_belle" En général, lorsqu'un CV arrive dans l'équipe (en ce moment nous cherchons un stagiaire pour avril 2008), croyez moi je le lis en 2 mn. Ensuite je le fais lire sans donner mon avis à quelqu'un de mon équipe. D'où l'importance de la présentation, et ensuite du contenu. Les sites suivants vous permettent de créer un CV en ligne accessible ensuite au format PDF (notre préféré), Word ou même HTML. Le site [Emurse.com](#) se propose de stocker gratuitement votre CV en ligne, et la qualité d'affichage est franchement top. Au passage si vous souhaitez rédiger un CV en anglais, il y a un grand nombre d'exemples, soit américains, soit anglais. Sinon il y a aussi [Resolio.com](#) et aussi [GigTide](#) qui est payant. GigTide offre des services supplémentaires comme de vous informer lorsque quelqu'un visionne votre CV. **Conclusion** Il n'y a pas que Monster ou lesJeudi.com. L'an passé c'était la mode "SecondeLife" jusqu'à ce que les gens réalisent qu'il n'y a pas de services à valeur ajoutée. Quand je vois LinkedIn ou Notchup, j'imagine que nous allons arriver aussi à ce type de solution. Le gain est double: un chasseur de tête peut identifier les quelques 80% de personnes qui dorment dans les entreprises, sans réseau, sans identité numérique sur Internet et qui pourtant, avec la motivation d'être payé, pourrait sortir voir dehors et rencontrer un recruteur... Vous n'imaginez pas les pepites qui dorment dans les entreprises, surtout en informatique.

## Java J2EE + Poker = xebia poker

Je suis tombé par hasard via [Wikio](#) ce matin sur une idée que j'ai trouvée géniale: un tournoi de **poker** organisé par la société [Xebia](#). A la clé, la possibilité de gagner une participation à JavaOne à San Francisco ! Le principe est simple: tout d'abord vous devez vous inscrire sur le site [XebiaPoker.com](#). Une fois inscrit, vous pouvez commencer à répondre à plusieurs Quizz en Flash sur le thème de Java, J2EE, Scrum ou Hibernate par exemple. Le quizz se déroule dans un temps limité. Croyez-moi c'est bien stressant. Et vous ne pouvez pas recommencer un quizz une fois lancé. Tout ceci pour ne pas tricher et que le système soit équitable. Les bonnes réponses aux différents quizz permettent d'établir un classement des joueurs. Les 50 premiers se verront inviter pour participer à un vrai tournoi. Etant aussi joueur de poker débutant, je me dis que c'est vraiment une excellente idée. Je me suis inscrit et j'ai commencé à concourir... je suis 21ème pour l'instant. Il me reste encore 2 quizz sur Spring et Hibernate pour espérer grimper un peu plus, mais ça va être difficile. Je vais potasser car je ne connais pas Spring par la pratique. J'ai lu un bon nombre d'articles mais j'avoue ne pas avoir eu l'occasion de m'en servir. Sur le quizz Java niveau 2 j'ai eu un plantage lors de l'envoi d'une réponse, avec affichage d'un message d'erreur... tant pis pour

moi. Les quizzes sont limités dans le temps, vous n'avez que 20 secondes pour répondre à chaque question. Et il n'est pas possible de repasser un quizz. Bref c'est bien stressant et très sympathique comme idée. Si vous êtes intéressés, rendez-vous sur <http://www.xebiapoker.com> JavaOne... me



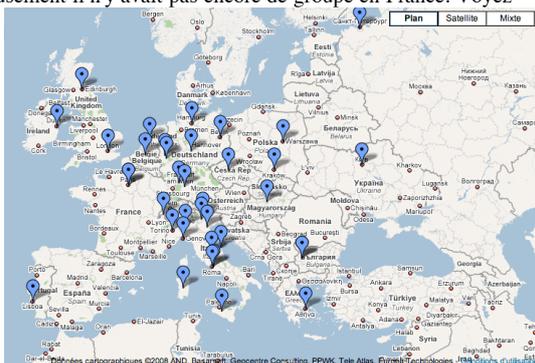
(re)voilà !!!

## Paris Java User Group (pjug)

La première réunion du Paris Java User Group (pjug) se tiendra ce soir à 19h00 dans Paris. Retrouvez toutes les informations sur le site [du pJUG](http://du.pJUG). A ne pas confondre avec le PJBUG (paris jboss user group).

## PJUG Paris : compte-rendu de la réunion de lancement

Mardi 12 février 2008 avait lieu à Paris la première réunion du Paris Java User Group (PJUG), grâce à [Antonio Goncalves](#), l'auteur du livre [Java EE 5](#) ainsi que [David Dewalle](#). Organisé dans les locaux de [Port Parallèle](#) j'ai compté plus de 65 personnes en tout. C'est donc un succès, et nous étions un peu à l'étroit pour la présentation. Mais tant mieux. Le principe des [Java User Groups](#) est de permettre à des passionnés de Java de se rencontrer. Avant tout un moyen convivial de rencontrer de nouvelles personnes et de discuter autour de Java et des technologies Open-Source. Comme l'a expliqué Antonio, curieusement il n'y avait pas encore de groupe en France. Voyez



vous-même sur la carte ci-dessous:

Il y a 15 groupes en Italie, dont la Sardaigne, et rien en Espagne par exemple. La Belgique est plus dynamique aussi, avec quelques grands JUGs. Enfin depuis ce soir, c'est parti. Il y a maintenant un groupe de passionné de Java qui se réunira les deuxièmes mardi de chaque mois à Paris. Au programme: présentation puis ensuite apéritif. Le tout sponsorisé par des sociétés. Merci pour ce soir à SUN, ainsi que le buffet offert par la société [Anéo](#). Après la présentation, le buffet convivial m'a permis de faire connaissance avec quelques personnes. J'ai bien discuté avec [Jérôme Grodziski](#) autour du métier d'indépendant et du fonctionnement du marché lorsque l'on décide de se mettre à son compte. Ensuite j'ai fait la connaissance d'Alexander von Zitzewitz, fondateur et directeur de la société [Hello2MORROW](#), éditeur de SonarJ, ainsi que de son distributeur en France, [Frédéric Brachfeld](#). SonarJ est un outil d'analyse de code capable de calculer des statistiques de couverture de code ainsi que de qualité. Il peut calculer la complexité cyclomatique d'une classe par exemple. Intégré à Maven ou ANT, SonarJ est donc un outil d'analyse de qualité du code. J'ai ensuite rencontré Cyril Le Clerc et une autre personne de [Xebia](#). Enfin des Xebia-boys en live ! Nous avons parlé du jeu-concours XebiaPoker de Xebia, puis de l'organisation de Xebia. Vraiment intéressant comme rencontre. En principe si tout va bien je dois participer à la formation Scrum Master fin mars. J'aurai donc l'occasion d'aller à la rencontre de Xebia. Sinon pour le jeu XebiaPoker, j'ai remonté de 17 places, et ce soir je suis 43ème... Snif laissez-moi repasser les tests de performances que j'ai planté en répondant trop vite et le test J2SE advanced qui a planté par contre (message "connection time-out" avec le serveur dans la page Flash). Enfin j'ai retrouvé une vieille connaissance de Coplanet: Florent Ramière. Cela remonte à 1998 je crois du temps où il était en stage de fin d'étude de l'EPITA. Il est aujourd'hui co-fondateur et responsable de [Jaxio](#). Jaxio est l'éditeur d'une solution de génération de code source Java qui permet de générer du code utilisant les dernières technologies Java en partant du modèle de base de données. L'intérêt est de s'affranchir du code spaghetti sans valeur ajoutée et bien souvent non suffisamment testé pour se concentrer sur la vraie valeur d'un logiciel. Soirée sympa, où l'on peut rencontrer des personnes passionnées de Java qui sont venues pour partager un moment avec d'autres fans. Si vous souhaitez venir à la prochaine réunion, rendez-vous sur le site <http://www.parisjug.org/>

## Présentation de la JSR-296, Swing Application

# Framework

L'une des nouveautés de Java 7 sera la mise à disposition d'un framework applicatif pour Swing, spécifié dans la [JSR-296](#). Voici un résumé qui vous intéressera si vous souhaitez en savoir plus pour la plateforme Desktop de Java. **Quel est avant tout la justification d'un framework pour swing ?** Swing est disponible depuis presque 10 ans. Il existe un grand nombre d'applications Java aujourd'hui qui ont toutes été écrites avec les moyens du bord. En conséquence (et c'est le cas chez nous) de nombreux développeurs ont implémenté une couche de framework afin de fournir des services de base par dessus Swing. Le constat est que toutes les applications Java qui utilisent Swing se retrouvent à devoir implémenter une partie de plomberie pour fonctionner. C'est l'une des raisons aussi du succès de certaines plateformes comme Eclipse RCP et Netbeans Platform. **Swing a-t-il évolué depuis 5 ans ?** Je mets au défi un développeur Java expérimenté de passer sans erreurs l'un des tests du site [Java Black Belt](#) sur Swing. Vous verrez que si vous êtes un maître Jedi en Spring ou EJB3, vous risquez une douche froide en allant faire un tour du côté de Swing. Spécialement depuis Java 5 et Java 6 qui, ne l'oublions pas, ont fait avancer Swing d'un grand pas en avant. La réponse à la question initiale est donc oui. Avec la JSR-296 et aussi la JSR-295 (Beans binding) nous arrivons à un framework simple et disponible de facto dans Java. Je parlerai de la JSR-295 la semaine prochaine. **Y-a-t-il une comparaison avec Eclipse RCP, Spring RCP ou Netbeans Platform ?** Le framework de la plateforme Eclipse est un succès car il comble un manque dans l'architecture de base de Java. De même pour Netbeans et son moteur très puissant. En même temps ces frameworks ne sont-ils pas trop gros pour des applications simples ? Et pourquoi devoir se reposer sur une dépendance supplémentaire ? Le message des leaders de la spécification JSR-296 est : plus simple, moins riche et livré en standard avec Java.

**Caractéristiques du framework** Voici maintenant ce qui est couvert par la JSR-296 : - gestion du cycle de vie d'une application - chargement de ressources et gestion de l'internationalisation (i18n) - persistance de la configuration de l'affichage, de la position des fenêtres automatiquement - définition de la notion d'Action au sens cas d'utilisation afin de faciliter l'écriture du code. Utilisation appropriée du moteur de thread de Swing pour effectuer de manière asynchrone certaines tâches. **Gestion du cycle de vie** En premier lieu, une gestion du cycle de vie de l'application. Combien de fois avons-nous eu à développer des shutdown hook afin d'intercepter la fermeture d'une fenêtre ? La JSR-296 propose une class Application qui vous permet de suivre le cycle de vie de votre programme. Parmi les méthodes de cette class nous retrouvons : - launch() signifie que l'application s'initialise. Vous pouvez afficher un splash-screen par exemple - startup() - shutdown() - exit() Quoi d'autre ? Un système événementiel de MileStone permet à votre application de signifier à qui veut bien l'écouter (une progress bar) l'avancement du chargement initial. Enfin un système standard permet de gérer la sortie de l'application, lorsque l'utilisateur clique et que vous souhaitez lui demander s'il veut sauver son fichier par exemple. **Les Ressources** Je ne sais pas si vous avez déjà manipulé des ResourcesBundles en Java. Ceux-ci permettent de charger typiquement des chaînes de caractères à afficher pour les labels de votre application. C'est limité car basé uniquement sur des Strings, et unidirectionnel : il n'est pas possible d'enregistrer des préférences dans un ResourceBundle par exemple. Enfin, un ResourceBundle n'est qu'un fichier plat de type Properties, sans notions de hiérarchie. Quels sont les apports de la JSR-296 ? Tout d'abord la possibilité de stocker des chaînes, des images, des URL, des valeurs numériques, ou des valeurs hexadécimales (code couleur). Ensuite une notion d'héritage pour regrouper certaines ressources générales (le nom de votre produit) et de spécialiser dans des fichiers dérivés des propriétés particulières (nom d'une fenêtre, d'un label). La class ResourceMap permet de convertir et de mettre en cache les propriétés de l'application. Pour cela elle se base sur la class StringConverter qui fait partie de ResourceMap. ResourceMap est un décorateur pour la class ResourceBundle qui elle-même finalement, est un décorateur pour Properties. Le plus important (et je crois le moins simple à expliquer) c'est la notion d'héritage. Basé sur la représentation Package>Class. Si vous voulez utiliser « gratuitement » l'injection de propriété, il faut suivre quelques règles. Il est temps de sortir un petit exemple : Soit la class suivante qui affiche le nom et la couleur d'un fruit dans un JLabel

```
/src/main/java/com/touilleur/demo/TestJSR.java
```

```
package com.touilleur.demo ;

public class TestJSR {
    private JLabel myLabel = new JLabel("myLabelTest","Fraise");
    ...
}
```

Pour pouvoir spécifier par exemple le label, mais aussi la font et la couleur du JLabel, il suffit de déclarer un fichier de Properties dans un répertoire resources en respectant la structure de package de la class TestJSR /src/main/java/com/touilleur/demo/resources/TestJSR.properties

```
# Fichier de test
myLabelTest.text=Une Orange de Seville
myLabelTest.font=Arial-BOLD- 16
myLabelTest.foreground=142,20,65
```

Les ressources ne sont donc pas qu'internationalizable mais aussi configurable sans taper de code Java. Vous imaginez ici facilement la possibilité de passer d'une charte graphique à une autre en modifiant simplement les propriétés des éléments. Ce qu'il manque, c'est une notion d'héritage de propriétés comme en CSS par exemple pour vraiment pouvoir fixer facilement l'aspect d'une interface. **Exemple complet** Voici un exemple qui utilise la classe SingleFrameApplication ti

```
package com.touilleur.demo ;
/**
 * Exemple simple JSR-296
 */
public class SayHello extends SingleFrameApplication
```

```

{
    protected void startup(String[] args)
    {
        JButton btn = new JButton();
        btn.setName( "testButton" );
        JPanel panel = new JPanel();
        panel.add( btn );
        show( panel );
    }
    public static void main( String[] args )
    {
        Application.launch( SayHello.class, args );
    }
}

```

Nous pouvons associer le fichier de propriétés suivant afin de définir les caractéristiques de l'interface facilement :

```

Application.title=SayHello
Application.id=SayHello1

testButton.font=times italic 12
testButton.foreground=255,0,0
testButton.Action.text = &Afficher un message

```

Ceci va créer un JLabel en rouge, sur lequel l'utilisateur pourra cliquer pour afficher un message. Justement parlons maintenant des Actions. **Gestion des Actions** Les actions en Swing sont les événements que l'utilisateur déclenche dans l'interface. Pour cela une nouvelle annotation @Action permet de marquer une méthode comme étant une action. Si j'ajoute une méthode sayHello à mon code pour changer le label du bouton, voici comment la déclarer. J'ai récupéré un exemple publié par l'un des auteurs de la JSR-296 et je l'ai adapté. Notez l'ajout d'une méthode sayHello qui retourne une SayHelloTask.

```

package com.touilleur.demo ;
/**
 * Exemple simple JSR-296
 */
public class SayHello extends JFrameApplication{
    protected void startup(String[] args){
        ApplicationActionMap applicationMap = ApplicationContext.getInstance().getActionMap( getClass(), this );
        ResourceMap resourceMap = ApplicationContext.getInstance().getResourceMap( getClass() );

        JLabel label = new JLabel();
        label.setName( "titleLabel" );

        JButton button = new JButton();
        button.setName( "testButton" );
        button.setAction(applicationMap.get( "startCounting" ) );

        JPanel panel = new JPanel();
        panel.setBorder( BorderFactory.createTitledBorder(resourceMap.getString( "countingDemo" ) ) );
        panel.setLayout( new BorderLayout() );
        panel.add( label, BorderLayout.NORTH );
        panel.add( button, BorderLayout.CENTER );
        show( panel );
    }

    @Action
    public SayHelloTask sayHello( ActionEvent ev ){
        return new SayHelloTask();
    }

    public static void main( String[] args ) {
        Application.launch( SayHello.class, args );
    }
}

```

Vous pouvez aussi conditionner l'exécution d'une Action à la valeur d'une propriété. Comment ne pas appeler la méthode « sayHello » si la propriété « speak » est à false ? Simplement en utilisant l'attribut « enabledProperty » comme sur cet exemple

```

@Action(enabledProperty = "speak")
public SayHelloTask sayHello( ActionEvent ev ) {
    return new SayHelloTask();
}

```

Voyons pour terminer la partie Task et j'aurai fait le tour des caractéristiques les plus importantes du framework Swing.

```

public class SayHelloTask extends Task{

    public SayHelloTask (){
        super(SayHelloTask.class);
    }

    public Void doInBackground() throws InterruptedException {
        System.out.println("Je prépare ma réponse tranquillement...");
        while(!isCancelled()) {
            Thread.sleep(2000);
            publish((Void)null);
        }
        return (Void)null;
    }

    public void process(List ignored) {

```

```

        long elapsedTime = getExecutionDuration(TimeUnit.MILLISECONDS);
        System.out.println("Elapsed" + elapsedTime);
    }
}

```

Cependant ici, si je viens à cliquer plusieurs fois sur mon bouton, je vais déclencher autant d'événement que de click dans l'utilisateur. Or parfois vous voulez bloquer l'interface ou au moins une partie le temps du traitement. Grâce au framework nous pouvons facilement désactiver par exemple notre bouton en utilisant l'enum `Block.COMPONENT` comme sur cette exemple

```

@Action(enabledProperty = "speak", block = Block.COMPONENT)
public SayHelloTask sayHello( ActionEvent ev ) {
    return new SayHelloTask();
}

```

**Enregistrement de l'état** Cette fonction permet de persister la position des fenêtres, vos préférences d'affichage, le fait d'avoir changé la taille d'une colonne d'un JTable par exemple. Très pratique et évite de devoir tout implémenter soit-même. **En conclusion** Swing Application Framework est un framework léger et simple qui se propose de simplifier l'écriture des applications Swing. Des abstractions simples permettent d'écrire plus rapidement du code robuste et performant, car la gestion des threads est simplifiée. Je n'ai pas parlé de l'injection avec le tag `@Resource`. C'est une initiative de Romain Guy au départ (voir mes anciens posts) et cette idée permet vraiment de simplifier l'injection de ressource comme des icônes dans votre code. Comme je dis à chaque fois : à tester donc. **Ressources supplémentaires:** Présentation complète de JSR-296 <http://developers.sun.com/learning/javaoneonline/2006/desktop/TS-3399.pdf> [http://weblogs.java.net/blog/diverson/archive/2007/04/swing\\_applicati.html](http://weblogs.java.net/blog/diverson/archive/2007/04/swing_applicati.html) Voir aussi [le blog d'Hans Muller](#) de SUN.

## Les site Web 2.0: le travail collaboratif

Deuxième article sur le thème des sites Web 2.0. Après [comment trouver un nouveau job](#) je vais vous parler ce soir de quelques outils pour la gestion d'une petite entreprise, la votre. En tant que Freelance, que pouvons-nous trouver sur le web pour nous faciliter la vie ? Quels outils pour surveiller un site ou versionner du code ?

Commençons par [HomeCourt](#). Ce site est un extranet de gestion de la relation client simple et rapide. Vous et votre client listez des actions ou des éléments importants. Ensuite je prends de mon côté ce qui est de ma responsabilité. Mon client prend de son côté les actions à effectuer. Nous pouvons ainsi tenir à jour une liste de tâches simples. Avant tout destiné aux web designers ou aux freelancers, HomeCourt est simple et les vidéos d'explication permettent de se faire une bonne idée de la puissance de cet outil de collaboration. Il est aussi intéressant de noter qu'il permet de conserver une trace des échanges et des tâches à effectuer.

Pour gérer un projet, si vous cherchez une solution similaire à Microsoft Sharepoint sachez qu'il existe de plus en plus de site de gestion de tâches et de micro-projet. Par exemple le site [ActionThis](#) propose des outils de collaboration et de gestion de documents en ligne. Voir la [démonstration](#) pour vous faire une idée plus précise. Cet outil vous permet de gérer les tâches de votre équipe et d'effectuer le suivi des coûts et de l'avancement du développement. L'intérêt est aussi l'intégration avec Outlook et son agenda pour la gestion des réunions. Voir aussi le très connu [BaseCamp](#).

Une fois l'équipe sur les rails, et les premières lignes de code écrites, il est peut-être temps de sauver et versionner votre travail non ? [Beanstalk](#) est un serveur Subversion hébergé. Il vous libère de la contrainte d'administration d'un serveur. Il propose en plus des outils graphiques pour la gestion du repository. Voir ensuite si vous n'avez aucuns soucis à héberger votre code source en dehors de chez vous. A noter qu'il est possible d'avoir un compte gratuit pour 3 personnes et limité à 20 Mo. Pas mal si vous souhaitez tester avant de vous engager.

Votre projet est maintenant sur les rails, mais avez-vous déposé un nom de domain ? Vous n'avez pas d'idées ? Allez faire un tour sur [Bust A Name](#). Avec 3 ou 4 mots, il sera capable de vous proposer quelques combinaisons de nom de domaine libre. Pas de FR malheureusement. Mais fun.

Une fois votre site en ligne, il est important de le monitorer. Pas plus tard que la semaine passée, mon blog a été inaccessible pendant 48H.... La catastrophe ! Afin d'éviter cela il existe un bon nombre de solutions. Par exemple [Mon.Itor.Us](#). Ce site est gratuit. Vous placez l'url de votre serveur, un contact d'urgent et c'est tout. Il surveille ensuite votre serveur HTTP régulièrement. Vous pouvez aussi ajouter par exemple un rapport de performance afin de suivre dans le temps les fluctuations de performances de votre application par exemple.

Chacun des sites est construit sur les principes Web 2.0 : simple, efficace et clair. Vous verrez que la page d'accueil de chacun des produits vous donne d'un coup d'oeil l'utilité du site et du service. C'est important pour les visiteurs qui viennent sur le site la première fois. Pour vos clients habituels, il est important aussi d'isoler une zone d'atterrissage clairement identifiée pour qu'ils puissent s'authentifier rapidement et entrer dans l'application. Regardez par exemple [GorillaSpot](#). Combien de temps faut-il pour comprendre l'utilité du site ? Par contre difficile de trouver la zone d'entrée pour les clients du site.

D'autres sites fonctionnent aussi très bien sans devoir s'enregistrer. Prenez par exemple [Drop.io](#). Il vous permet de placer un fichier pour l'échanger rapidement avec une autre personne. Un genre de rapidshare plus simple et sans publicité s'il vous plaît...

Sur ce, je vous souhaite une bonne soirée et à plus tard.

## Scrum: le planning poker

Avant de parler du Planning Poker, quelques mots sur Scrum. Cette méthode Agile lancée en 1992 par Jeff Sutherland permet de travailler plus rapidement et efficacement dans la gestion de projet. Elle peut s'appliquer au développement d'un logiciel informatique mais pas uniquement. J'en parlerai certainement plus en détail dans un autre post, car je participe à la formation Scrum Master de Xebia qui aura lieu le 25 et 26 mars prochain. Ce matin donc, parlons du planning Poker. J'ai entendu ce mot pour la première fois il y a quelques semaines. Quel rapport entre un jeu de carte et une méthode Agile ? J'ai trouvé la réponse sur le site d'une société qui fabrique ce jeu, et je vais donc traduire en partie la présentation du planning poker (références du site à la fin de cette page). Le Planning Poker est l'un des outils de la méthode Scrum. Il permet à une équipe lors d'une réunion de planification de donner des quotations pour le développement d'une fonctionnalité. Lors de la réunion, le product owner (directeur de produit) représente le client, le commanditaire. Il donne l'ordre dans lequel doit s'effectuer les développements des nouvelles fonctionnalités. L'équipe a pour responsabilité de donner des quantifications au directeur de produit afin que celui-ci sache aussi que telle fonctionnalité coûte 10 unités et qu'une autre fonctionnalité ne coûtera que 2 unités. A noter que je ne parle pas de jour/homme pour l'instant. Et c'est là qu'intervient le Planning Poker. Mettons-nous en situation. Vous réunissez votre équipe de 5 développeurs dans une salle. Vous demandez à tout le monde "combien de temps pour développer l'ajout de la fonction X sur mon logiciel ?" . Mr A pense qu'il sait exactement ce qu'il faut faire, donc il pense 3 jours. Mr B et Mr C sont plus prudents. Mr D et E ne participent pas activement à la conversation et n'écourent pas. Bref c'est Mr A qui prend la parole et qui vous répond "**Je pense 3 jours**". L'équipe est influencée par la première personne qui s'exprime. C'est un risque important car en fait on voit que B et C pensaient que cette fonction demande plus de temps à développer. Nous risquons de perdre de l'information ici et leurs doutes doivent être exprimés. Reprenons la même scène avec le planning poker. Le principe est de donner à chacun de vos équipiers un jeu de carte spécial, le fameux Planning Poker dont voici une photo : Vous distribuez un jeu complet à chacun de vos équipiers. Comme vous pouvez le constater, les cartes ne se suivent pas. Elles sont basées en partie sur la suite de Fibonacci afin de forcer les personnes à trancher. L'unité n'est pas le nombre de jour/homme, mais une unité arbitraire. Il est important de donner un jeu à chaque personne de l'équipe. Ensuite voici comment se déroule la séance : le directeur de produit exprime son idée, et l'équipe lui pose des questions afin d'affiner la demande pendant environ 3 mn. Ensuite commence la phase du vote. Chacun prend une des cartes et la pose devant lui face cachée. Cette fois-ci chacun pense librement et activement. Mr D et Mr E sont obligés aussi de participer. Une fois que tout le monde a sélectionné une de ses cartes, on retourne celle-ci en même temps et le résultat du premier vote est discuté par l'équipe. Wow cette fois-ci nous voyons bien qu'entre Mr A et Mr C, une discussion s'impose à propos de l'histoire. Après quelques discussions, Mr A réalise qu'il a oublié une partie du développement et qu'il faut l'inclure. Mr C réalise qu'avec la proposition de Mr A en effet il est possible de réaliser plus rapidement la fonction et que donc son estimation est un peu trop pessimiste. Après 3 mn, toute l'équipe refait une estimation en reprenant les cartes, et en sélectionnant une nouvelle carte, ou la même s'ils n'ont pas changé d'avis. Voilà maintenant une quotation réaliste et réalisable de la demande initiale. Les développeurs se sont parlés activement afin de présenter une vision sur la solution envisagée. Et tout le monde y a participé. Ce qui est aussi efficace en terme d'échanges et de communications. Il existe quelques cartes spéciales: - la carte 0 représente une fonction qui est déjà implémentée - la carte ? représente le fait que vous n'avez vraiment aucunes idées sur la question - la carte "tasse de café" c'est le joker pour signaler que vous voulez faire un break

### Ressources supplémentaires:

Concernant la valeur des cartes et l'attribution des unités, voici [l'article original de Mike Cohn sur le Planning Poker](#) en anglais. Très complet. [Frédéric Doillon](#) est un Scrum Master dont le blog est une mine d'or en français car il partage son expérience de Scrum master avec son équipe. Allez lire [son billet sur le Planning Poker](#). Vous pouvez fabriquer un jeu de carte avec quelques fiches cartonnées ou en acheter un à cette adresse <http://www.crisp.se/planningpoker/>. Vous retrouverez aussi la version en anglais de ce que j'ai traduit [à cette adresse](#) et donc le crédit de cet article va en premier à Crisp SE. Merci à eux.

## PJUG : présentation de Selenium... sans moi

Contre-temps de dernière minute, je n'ai pas assisté ce soir à la deuxième réunion du Paris Java User Group (PJUG) et à la présentation sur [Selenium](#). Ce sera pour une prochaine fois. Tout cela pour une histoire de correction en urgence pour un client... vous connaissez l'histoire : branche svn, correction et test rapide en croisant les doigts... Mais si nous avions fait comme cela pour construire la Tour Eiffel, croyez-moi il n'y aurait rien... Lisez bien vous allez comprendre... J'ai la confirmation que j'assisterai fin mars à la certification Scrum Master avec Jeff Sutherland, organisée par [Xebia](#). Passer d'un mode de développement à l'ancienne (spécifications, codage, livraison, recette) à une méthode Agile comme Scrum, c'est une petite révolution qui se prépare. Pour moi c'est vital. Il nous faut de l'Agilité pour répondre efficacement à la complexité de notre secteur. C'est un peu bateau je l'avoue. Il me faut aussi de quoi évangéliser (car je ne vois pas d'autre mot) les 3 équipes autour de moi, puis ensuite les 4 clients actuels internes qui utilisent notre framework. Par ailleurs nous avons un projet qui sera le candidat idéal pour mettre en application tout cela. J'ai cependant le sentiment que finalement, le plus gros effort sera la partie humaine et la communication. Il va être difficile de faire passer un message et de demander à notre environnement de s'adapter à notre nouvelle méthode. Mais comme disait un

grand savant chinois : "Qui n'avance pas... recule". Le développement d'un logiciel est finalement assez proche de la croissance d'un arbre. Une branche pousse, des fleurs puis des fruits sortent sur les branches. Parfois la branche est malade, et on tente de la réparer avec un bout de ficelle... alors que finalement il vaut mieux parfois couper net et proprement afin que la maladie contractée par cette branche ne se propage pas à l'arbre tout entier... (je suis le roi de la métaphore de supermarché). J'ai appris il y a quelques jours que Gustave Eiffel avait construit sa Tour en 26 mois avec à peine 200 personnes. Vous imaginez un Gustave Eiffel ingénieur ? qui dessine ? Pas du tout. Mr Eiffel était à la fin de sa carrière. Plus rien à prouver, 25 ans d'expérience. Par contre un sens de la communication et de la politique ultra-développé. Bref lorsqu'il entend parler d'un projet pour l'exposition universelle de 1889, il se débrouille pour faire gagner son projet au concours organisé par la ville de Paris. En 1887 il demande à 2 de ses meilleurs ingénieurs de travailler sur les plans (il fait de la délégation) en prenant pas mal de temps pour regarder ce que font les concurrents (veille technologique). Les premiers plans sont prometteurs mais... moches ! Il contacte un architecte afin de demander un dessin qui donne une touche esthétique à son idée de Tour (appel à un prestataire qualifié externe). Par la suite il subit 2 grèves pendant la construction. A la première grève les ouvriers veulent 40 centimes d'augmentation. Il fait une promesse plus généreuse à 50 centimes s'ils reprennent le boulot... et les ouvriers n'ont rien finalement (exploitation acharnée de prestataires). Deuxième grève, avec cette fois-ci des ouvriers bien décidé à obtenir un peu plus par mois. Il refuse encore mais promet une prime énorme de 100 Fr si le chantier est terminé à temps pour l'exposition universelle de 1889. Les ouvriers, pour le coup vraiment motivés par cette prime, termineront le chantier avec 2 mois d'avance en mai 1889! Gustave Eiffel payera la prime de 100 Fr (tenir ses promesses et motiver par une prime). Pour revenir aux méthodes Agile, une chose m'a marqué. Lors de la construction de la Tour Eiffel, Gustave Eiffel avait demandé aux personnes manuelles qui montent la Tour de refuser toute pièce comportant le moindre défaut. Le zéro défaut à la Toyota avant l'heure. Un journaliste demande à Eiffel : - "Attendez... vous payez cher ce fer qui vient de Lorraine, vous faites fondre les pièces à Levallois, et si la pièce a un défaut, vous la jetez ?" - "Oui tout à fait..." - "Mais pourquoi par exemple ne pas la raccourcir lorsqu'elle est trop longue ?" - "Vous avez raison... et vous avez tort. Si nous demandons à un monteur manuel de recouper la pièce dans l'urgence, sur les calculs d'un ingénieur sur le site qui sera stressé, qui aura le vertige et froid, pensez-vous que la correction de la pièce sera parfaite ?" - "...euh non... et vous ne pouvez pas rallonger la pièce ?" - "Les monteurs sont des personnes qui sont dédiés à une tâche: assembler les éléments. Les ingénieurs font les calculs et s'assurent que la Tour ne va pas s'écrouler. Donc ce n'est pas au monteur de décider ce qu'il faut faire" - "Quelqu'un surveille le travail des monteurs ? Qu'ils ne font pas exprès de rejeter trop de pièces ?" - "Nul besoin. Les monteurs sont fiers d'avoir la possibilité de s'opposer aux Ingénieurs. Et croyez-moi, pour rester crédible ils ne refusent les pièces que lorsque c'est vraiment nécessaire..." - "...". Gustave Eiffel venait d'inventer une méthode utilisée ensuite 100 ans plus tard par Toyota pour construire leurs véhicules: le Zéro Défaut. Et si nous revenons à nos logiciels ? Imaginez parfois un bout de code que vous avez écrit tard comme ce soir pour moi... Vous êtes à 95 mètres du sol, entrain de tailler votre bout de code Java pour le faire tenir tant bien que mal... vous savez que si vous ne mettez pas cette correction, il ne sera pas possible de continuer à monter la suite de votre logiciel... Le client est en bas qui attend une solution miracle pour traiter de gros volumes... Alors vous codez, vous sciez, et vous pensez qu'en 1888 s'ils avaient fait comme vous, la Tour serait certainement aujourd'hui une horreur bien couteuse à maintenir. Et pour le coup, ce n'est plus la branche mais l'arbre qu'il faut abattre... Allez je retourne à la mine. Faire du code rend humble.

## Vendredi j'achète un iPod nano 3G, samedi il est planté

**Si vous votre iPod nano 3ème génération (3G) se met à redémarrer à chaque fois que vous le branchez sur votre machine via USB, alors cet article est pour vous.**

C'est beau la technologie.

Jeudi matin j'achète sur l'AppleStore un balladeur MP3 Apple iPod nano 3G de 8Go. Livré le lendemain par UPS s'il-vous-plaît ! le tout gravé avec mon nom au dos de la bête. Bref me voilà parti pour le brancher sur le Mac afin de pouvoir enfin écouter dans le RER de la musique et voir des vidéos. Tout se passe bien jusqu'au téléchargement d'une nouvelle version du firmware. Après l'installation du nouveau logiciel, l'iPod reboot et là, *c'est le drame...*

L'appareil s'initialise (affichage du logo Apple) puis une icône de connexion apparaît, et l'appareil redémarre... le logo s'affiche, logo prise branchée, et il redémarre... et ce, tant qu'il est connecté via USB au Mac.

J'installe alors iTunes sur un PC, je reteste... même problème. Facheux...

Merci Google et [le site du support d'Apple](#), j'ai trouvé la solution. Cependant elle est un peu difficile à mettre en oeuvre et je me demande même si je ne vais pas en faire une vidéo afin d'expliquer comment réinitialiser votre iPod nano ou iPod classic.

Tout d'abord il faut passer l'appareil en mode disque. Pour cela vous pouvez vous devez tout d'abord effectuer un reset de l'appareil (cela n'efface rien pour l'instant).

- o Débranchez tout d'abord l'iPod du câble USB. Il faut que la batterie soit chargée, même partiellement, afin d'effectuer la manipulation ci-dessous.
- o Déplacer le curseur "Hold" situé sous l'iPod à droite puis replacez-le à gauche (donc unhold)
- o Appuyez sur la touche "Menu" et sur le curseur central sans toucher les autres zones du touch-pad. Conservez les boutons appuyés, cela va éteindre votre iPod. A cet instant vous avez toujours les 2 doigts appuyés sur chacun des boutons.

- L'icône pomme "Apple" s'affiche pour indiquer que l'appareil démarre. Sans lâcher le curseur central, relâchez la touche "Menu" et appuyez sur la touche "Lecture/Pause" en plaçant votre doigt assez à l'extérieur de la roue, afin de ne pas appuyer sur une autre touche. Conservez alors appuyés vos 2 doigts jusqu'à ce qu'un écran blanc s'affiche et vous indique (en anglais) "Disk Mode".
- Vous pourrez alors rebrancher le câble USB sur votre Mac ou sur Windows.
- Lancez iTunes, l'iPod sera détecté comme étant vierge. iTunes doit vous proposer de le réinitialiser. Arrivé à cette étape votre petit iPod nano est sauvé.

En deuxième partie de la solution, il faut aussi recréer proprement sa librairie sur iTunes, sur votre Mac ou sur Windows. Pour cela [le mieux est de suivre cet article](#) du support Apple qui explique comment réinitialiser sa librairie.

Une fois mon appareil initialisé, j'avoue qu'iTunes et un ipod nano, c'est vraiment bluffant. J'ai ainsi récupéré des podcasts vidéos sur AlloCine,M6, des podcasts en anglais sur Java, et le tout se transfère sans problèmes vers l'ipod. La qualité d'affichage est correcte, le son pas mal même si des écouteurs intra-auriculaires donnent un meilleur son.

## Buildix: SVN + CruiseControl + Mingle + Trac

Je pars sur un nouveau projet Agile en Java/J2EE et notre besoin est le suivant: je veux un serveur SVN pour versionner mon code. Je souhaite aussi faire de l'intégration continue, donc CruiseControl car il est facile à installer et suffisant pour mes besoins. Pour la documentation idéalement j'aimerais bien utiliser un wiki, car cela fonctionne bien mieux que Microsoft SharePoint Machin et Microsoft Word-j'écris-de-la-doc-que-personne-ne-lira

Bref allons chercher un peu si ce produit n'existerait pas sur le net...

Google.... 3mn plus tard je me dis que je vais encore faire économiser de l'argent à mon employeur.

--> [Buildix de ThoughtWorks](#).

Dans un écosystème Agile où vous avez besoin d'outils rapidement, c'est le produit qu'il vous faut. Voyez un peu : intégration continue, gestionnaire de code source, wiki, gestion des bugs et des demandes de changement à la sauce Agile... c'est pour vous.

L'intérêt est aussi l'intégration de ces outils. Nous utilisons la solution de Collabnet ici chez Reuters. Puissant mais limité à la gestion du code source, la gestion des bugs et des reqs. Il nous a fallu installer un système de wiki, et chaque équipe gère son serveur CruiseControl... gaspillage de temps et d'énergie. Je pense que l'idée de Buildix répond à un vrai besoin : notre métier c'est de réfléchir et de coder. Pas de patcher Solaris 10 pour installer gcc et openssl afin d'avoir un serveur SVN avec https... (c'est du vécu).

Buildix est donc composé des produits open-sources suivants:

- \* Subversion pour la gestion du code source
- \* Mingle pour la gestion de projet Agile
- \* Cruise Control pour l'intégration continue
- \* Trac comme wiki et système de gestion des bugs

Buildix est distribué sur un LiveCD avec un noyau Linux basé sur Ubuntu... oui vous ne devez pas: rien à installer ou si peu... (enfin on verra dans quelques lignes comment je vais changer d'opinion... suspens)

Enfin je termine par un mot à prononcer à haute voix: G R A T U I T

Et l'url aussi : <http://buildix.thoughtworks.com/>

### 3H plus tard... derrière mon PC, avec Buildix lancé sur une machine secondaire...

L'installation s'effectue en 40 mn. Basé sur Ubuntu, la distribution installe en effet proprement les produits suivants:

- Serveur SVN
- CruiseControl
- Mingle (qui ne semble pas être gratuit)
- Trac

Maintenant une information pour ceux qui vont essayer Buildix : **Enorme bug qui vous empêche de vous authentifier**. Je m'explique:

Durant l'installation je crée un utilisateur Nicolas, avec un mot de passe qui tient la route. Après l'installation, impossible de m'authentifier... je suis sceptique. Je reboote alors en single et après avoir un peu craqué ma propre machine (enfin là j'en rajoute à mort... booter en single vous donne l'accès root... mais bon, passons). Donc après avoir hacké pendant 3mn comme dans Matrix ma machine, je vois dans /etc/passwd un utilisateur **buildix**. Un petit coup de Google plus tard, en effet il y a un gros bug dans l'installateur. L'installateur ne va pas créer le user demandé. Il faut s'authentifier en tant que **buildix** avec comme mot de passe... **buildix**.

Une fois l'authentification effectuée, je me retrouve donc sur un bureau type Ubuntu 7.04 je crois.

Via Firefox, en se connectant sur <http://localhost> nous avons accès à une page d'accueil regroupant chacun des produits. D'ailleurs Mingle aura besoin d'une petite phase de configuration avant de marcher correctement.

C'est l'heure du bilan :

Le contrat est rempli, serveur SVN et CruiseControl correctement installé. Il faudra créer les utilisateurs manuellement, sachant que l'utilisateur unix buildix peut utiliser la commande sudo pour ajouter des utilisateurs. Ensuite Trac est un excellent produit. Il demande un effort de customisation, mais il est assez sympa. Mingle dispose de templates pour Scrum. Vous pouvez donc installer Mingle afin d'avoir un Product Backlogs ou un Burndown Chart en ligne. Voir ensuite si c'est vraiment l'outil adapté.

Du côté négatif : je pense que le LiveCD est une bonne idée... marketing. En vrai j'ai trouvé que cela ramait un peu pour le peu de serveur configuré. Une installation en vrai sur une machine dédiée me semble être une meilleure idée. D'autre part ce bug dans l'installeur fait que beaucoup risquent d'abandonner dès l'installation... Produit donc encore jeune mais qui gagne à être connu. ThoughtWorks a la bonne idée de répondre à un réel besoin : installer proprement un serveur SVN, bien que cela ne soit pas difficile, demande un minimum de connaissances systèmes.

Je surveillerai l'évolution de Buildix, même si pour moi finalement je continuerai d'installer à l'ancienne pour l'instant.

## J-1 avant la formation Scrum Master

Demain matin grosse journée : rendez-vous chez Xebia afin de participer pendant 2 jours à une formation certifiante sur Scrum délivrée par Jeff Sutherland. Je vous raconterai mon expérience personnelle et mes impressions sur la formation, sur Scrum et sur Jeff Sutherland. Stay tuned... Un peu de lecture en attendant pour vous: [C'est quoi Scrum ? Qui est Jeff Sutherland ?](#) et voir aussi [mon dernier post sur le planning poker](#)

## I'm now a Scrum Master

**Si une sorte de messie existe aujourd'hui, je dirai qu'il voyage en classe affaire, évangélise ses brebis et porte un polo Ralph Lauren noir.** Pendant ces 2 jours de formation avec Jeff Sutherland, nous avons couvert les différents aspects de Scrum avec une partie pratique et une partie partagée par Jeff avec son expérience. Scrum est avant tout un framework de développement Agile orienté produit et pas projet. Basé sur la simplicité, sur la transparence et l'efficacité. L'organisation parfaite nous a aussi permis de discuter et rencontrer pas mal de monde puisque nous étions 30 personnes. Suisses, Belges, Luxembourgeois et Français. **Déroulement de la formation** Durant ces 2 jours nous avons mis en oeuvre, non sans mal et avec pas mal de rires, les différents ateliers de Scrum. 30 personnes étaient présentes, avec quelques personnes de Xebia. Divisé par table de 6, nous avons effectué différents exercices, l'alternance entre présentation formelle, séance de questions, exercices et pauses cafés étant gérée par Jeff avec un grand sens de l'organisation. En fait son secret était affiché dans un coin de la salle. Il a déroulé les différents points de la formation sous la forme de post-it accrochés au mur. Durant les 2 jours, selon l'avancement, il mettait à jour son tableau en déplaçant ses post-its... Et nous avons tenu le programme sans problèmes, le dernier jour se terminant à l'heure. Qui était présent à cette formation ? Finalement des personnes de différents horizons. Des français mais aussi des personnes du Luxembourg et de Suisse. Des développeurs, des chefs de projets, des managers opérationnels... les profils étaient variés. Et les échanges ont été très constructifs. Sur le déroulement de la formation, sans en révéler tous les détails, je dirai que vous repartez avec un bon bagage. Suffisant pour commencer dès le lundi suivant à travailler en tant que Scrum Master. L'objectif est rempli. Finalement Scrum est simple. Il n'y a pas de choses compliquées à essayer de comprendre. Simplement 3 outils de reporting, 3 artefacts, quelques définitions simples... et vous pouvez vous lancer. Qu'en est-il de la part de "buzz commercial" par rapport à la réalité ? Finalement pour justifier la réussite de Scrum, le plus simple est de regarder les exemples de sociétés comme Yahoo, Google, GE HealthCare, qui sont passés d'un mode de développement à l'ancienne (Waterfall, RUP) à une méthode Agile comme Scrum. Les chiffres sont là pour répondre aux sceptiques. Et c'est assez impressionnant... Le plus gros du travail maintenant va être d'évangéliser, de proposer et de faire adopter Scrum. Car pour moi il est clair qu'un produit développé dans un mode Waterfall est un produit qui a de fortes chances de se planter. Alors certes, les sceptiques ou les maîtres de RUP risquent de bondir et de me tirer dessus à vue. RUP par exemple définit 27 rôles. Scrum fonctionne avec 3 rôles : Product Owner, Scrum Master et The Team. **CMMI 5 et Scrum, comment économiser de l'argent ?** Un graphe qui m'a marqué : un projet [CMMI](#) développé par une entreprise de niveau 1 qui a une productivité de 100, en mode Waterfall, coûte 50% de ce temps en correction de problèmes et 50% en développements innovants. Une entreprise évaluée CMMI niveau 5 améliore sa productivité et peut réaliser la production d'un logiciel pour 69% d'effort au lieu de 100% auparavant. La correction d'incidents, la réécriture passe de 50% à 10% sur la même échelle grâce au processus d'Optimisation mis en place. La productivité est inchangée, à 50%. Et par contre, 9% de coûts de structures, de gestion des processus s'ajoutent. 9% de l'activité est consacrée à se regarder le nombril pour s'améliorer sans cesse (principe CMMI 5). Appliquons Scrum dans cette structure et regardons ce qu'il se passe. Au total, l'effort passe de 69% à 35%. Cela veut dire, que pour 2 fois moins d'effort, vous produisez la même quantité de lignes de codes. Les coûts de gestion de processus sont divisés par 2 (4% au lieu de 9%). Les coûts de refactoring passent de 10% à 6%, vous réduisez de 40% le temps passé en correction de bugs, refactoring et bidouillages divers... Enfin la partie production pure passe de 50 à 25. Votre société a donc doublé sa productivité... Ces chiffres ainsi que de biens meilleurs explications en anglais se trouvent [ici](#) si vous souhaitez en savoir plus. **Et la suite ?** Je

compléterai à partir de maintenant en vous faisant partager mon expérience de mise en place de Scrum chez nous... C'est le début d'une longue série de Post je pense. **Photos souvenirs** Sans vous dévoiler pourquoi nous sommes entrain de faire ce château de carte, voici quelques photos prises par Alexander Papanastassiou de Business Quests et par moi-même. Le monsieur avec la chemise



noir, c'est Jeff Sutherland. Jeff présente les consignes du dernier



atelier Avec une velocity assez rapide nous complétons les 3 itérations et terminons même un peu vite. Alexander propose d'essayer finalement de faire l'exercice le plus difficile avec les cartes à jouer... Quasiment impossible.



Jeff Sutherland debout à côté de Guillaume Bodet de Xebia.



Damien trie les tâches du Product BackLog



Changement de tactique: sur une idée de Denis nous utilisons des



Post-it pour coller les cartes... et là ça avance vite. Un des exercices (sorti du contexte les gens vont se demander ce que l'on a fait...). Le monsieur avec le chemise blanche,



c'est Cyrille de Xebia. 4 étages !!! Une photo de Jeff Sutherland



et enfin une petite photo de moi en plein Sprint final, avec la



casquette de Scrum Master.

**Autres liens** Voir le post sur le blog d'Alex <http://blog.businessquests.com/2008/03/done.html> L'article de [David Andriana](#) est très clair et vous expliquera ce que nous avons fait pendant ces 2 jours. Si vous êtes intéressés, les places à la formation partent très vite, donc n'attendez pas, [inscrivez-vous sur le site de Xebia](#). Pour ma part j'ai même utilisé mon DIF (droit individuel à la formation).

## Premiers jours avec Scrum : une histoire de Sprint Backlog qui disparaît

C'est parti... Nous allons effectuer notre premier sprint la semaine prochaine. En attendant j'ai commencé à remplir un Product Backlog. Nous allons gérer pendant une semaine les tâches en cours en utilisant notre petit tableau sur le framework Karma dans un premier temps. Le tableau est déjà couvert de Post-It, avec les corrections de bugs et les développements en cours. L'idée est d'abord de m'adapter et de bien me roder avant d'attaquer le Sprint Planning la semaine prochaine. je n'ai pas encore fait tous mes choix d'outils, mais le plus simple c'est un tableau...

La première bonne nouvelle pour l'équipe c'est qu'en tant que Scrum Master, je leur ai retiré l'obligation de répondre aux emails que nous recevons sur un alias. Je vais me charger de faire le proxy entre les demandes de nos clients internes, souvent des développeurs, et mon équipe de développeur. Les questions étant souvent techniques, j'ai pris conscience qu'à chaque email intempestif, celui-ci génère une perte de temps pour l'équipe, plongée dans son travail. De plus la règle était "vous vous auto-gérez". Mais pas évident car tous n'ont pas la même charge de travail au quotidien. Nous allons essayer pendant quelques temps de cette façon. Cela ne devrait plus ralentir l'équipe. La fonction du Scrum Master est de retirer tous les obstacles (impediments en anglais) empêchant le bon déroulement du sprint.

Question dont je n'ai pas la réponse : **Comment gérer les bugs et les demandes de corrections "urgentes" ?**

J'ai le sentiment que durant un Sprint, l'équipe doit s'immerger complètement, comme l'équipage d'un sous-marin nucléaire. Ils ne doivent recevoir de l'extérieur que très peu de stimuli. Et ils savent ce qu'ils ont à faire. Notre durée de sprint sera de 3 semaines. Cela colle avec les sorties d'un gros logiciel chez nous. Et cela convient à mon avis à la taille des développements que nous effectuons. Donc au pire, un produit peut attendre 3 semaines. C'est réaliste avec le temps nécessaire ensuite pour les autres équipes de repackager leur produit, faire passer une campagne de test QA, faire valider la correction par le support et enfin effectuer une installation chez le client... On ne travaille pas à la journée prêt, ce qui serait irréaliste avec une démarche qualité. Maintenant j'attends plutôt de commencer un vrai sprint afin de voir comment rester efficace.

### L'aventure du tableau volé...

Le tableau avec notre pseudo sprint backlog a fait son effet. J'ai pris un paper-board avec quelques feutres. Après avoir tracé 3 colonnes (A faire, en cours, fait) j'ai utilisé des post-its pour les activités en cours cette semaine, afin de travailler sur cet outil avant de lancer un vrai sprint. Ce tableau est trop petit, et n'importe qui peut l'embarquer dans une salle de réunion.

Du coup, j'essaye de récupérer un énorme tableau blanc. Pour cela je dois aussi me battre avec les services généraux pour faire poser 4 chevilles dans un mur... Enfin passons.

Pour notre tableau actuel j'ai eu une désagréable surprise en revenant de notre pause déjeuner toute à l'heure avec l'équipe :

**La feuille avec nos post-its a disparu ! Plus de sprint backlog !**

Vous y croyez vous ?

Je serai victime du premier attentat anti-scrum...

Bref j'ai pas essayé de comprendre, me disant que le message est clair, va y avoir du sport... Ca commence fort

Quelques minutes plus tard, derrière moi, notre ergonomiste assez embêté s'approche discrètement :

- "...Euh... je crois que ta feuille là, elle est juste cachée là derrière... c'est moi qui fait une blague

Ah là là... vivement qu'ils me fixent un vrai tableau dans le mur et que l'on puisse travailler.

Bon et vous ? Ca Scrum ou pas ? racontez-nous un peu

## Xebia Poker : dernière ligne droite

Xebia organise le jeu XebiaPoker depuis quelques semaines. Ce soir c'est la dernière ligne droite. Les 50 premiers se retrouveront à Paris le 10 avril au Café Barge pour un vrai tournoi de poker afin de départager les concurrents. Le vainqueur aura la chance d'aller à JavaOne à San Francisco qui se déroulera du 6 au 9 mai 2008. Ce soir j'ai terminé mon dernier quizz. Pour l'instant je suis 16<sup>ème</sup> et donc c'est bon pour moi pour le 10 avril.

## Ventura attaque SUN Microsystems sur la marque Java

J'ai un peu halluciné en voyant la news tomber sur un flux Reuters ce matin : Ventura Ltd., une société de droits panaméens a déposé une requête ce matin au tribunal de San José en Californie à l'encontre de SUN Microsystems. La plainte porte sur l'utilisation du nom Java par SUN Microsystems depuis 1995 en tant que marque déposée, alors que celle-ci n'était pas protégée. La société Ventura Ltd a eu l'idée de déposer il y a quelques semaines discrètement la marque Java dans un périmètre plus important que celui initialement effectué par SUN Microsystems. Par ailleurs SUN n'avait pas protégé la marque sur tous les domaines d'exploitations (informatique, média, sport, alimentation, automobile...) et donc la société Ventura en a profité. Bref nous risquons nous aussi de ne plus avoir le droit d'utiliser le mot "Java" car il est maintenant la propriété de Ventura Ltd, qui s'avère être une société de pêche aux poissons.

## Hudson, l'intégration continue certifiée Bio

Sur une idée de Guillaume Carré de Xebia, j'ai testé [Hudson](#). Il s'agit d'un serveur d'intégration continue basé sur un moteur d'exécution d'activités répétées et de surveillance de tâches. Vous pouvez donc vous en servir comme un CruiseControl pour effectuer de l'intégration continue... mais pas seulement !

Chez nous ce n'est pas une ferme, mais vraiment des fermes de serveur avec des serveurs [CruiseControl](#) élevés en batterie qui compilent, intègrent et déploient chaque nuit nos applications. En majorité pour des projets Java, il y a des intégrations réussies de C/C++.

Le souci je crois, c'est que nos poulets bios sont plutôt devenus des poulets industriels élevés en batterie... CruiseControl c'est bien si vous apprenez à vous en servir et si vous faites l'effort de le nettoyer de temps en temps... Or je me rends compte que, faute d'une interface de configuration simple, les développeurs tendent à laisser tourner CruiseControl comme un hamster dans sa roue.

Hudson en premier séduit par son interface Web d'administration et de gestion. En quelques clics de souris vous pouvez préparer votre projet et lancer vos premières compilations... La classe.

Pour lancer Hudson via Java Web Start [cliquez-ici](#) puis une fois l'installation effectuée, regardez sur <http://localhost:8080>.

Ensuite notre machine à élever le bon grain est assez bien architecturée car j'ai vu que le système pouvait fonctionner en effectuant des builds parallèles. CruiseControl est synchrone et n'effectue qu'une seule build à la fois. Souvenez-vous de [TeamCity](#), le produit de JetBrains, qui faisait cela très bien, sous la condition de dépenser quelques dollars (il existe une version gratuite, allez vérifier sur le site de JetBrains).

Hudson est capable par exemple d'effectuer un job de vérification régulièrement et de vous donner un état. Première application pour nous : vérifier que nos urls de démonstrations sont actives et fonctionnent avant que ce ne soit les gens de l'équipe qualité qui s'en rendent compte.

Du côté de l'intégration continue, Hudson peut surveiller votre repository et s'activer lorsqu'un utilisateur effectue un commit (classique). Il peut ensuite sauvegarder les artefacts générés. Et la fonction la plus intéressante que je n'ai pas testée : la possibilité de rappeler une build. C'est un système très pratique qui permet de relancer une build de release pour les clients qui souhaitent une version un peu ancienne de votre produit. Pour notre part, nous ne faisons pas que de la compilation. Chaque nuit nous effectuons la publication sur notre repository d'une snapshot afin que nos clients puissent intégrer en continu sans attendre une release. Bien entendu il y a des releases officielles, mais nous effectuons celles-ci manuellement avec Maven.

Voici par exemple le tableau de bord de JBoss pour vous donner une idée en live de ce que cela peut donner :

<https://hudson.jboss.org/hudson/>

Bref je pense qu'il sera intéressant de généraliser Hudson, au moins pour les nouveaux projets, car l'administration est vraiment plus simple.

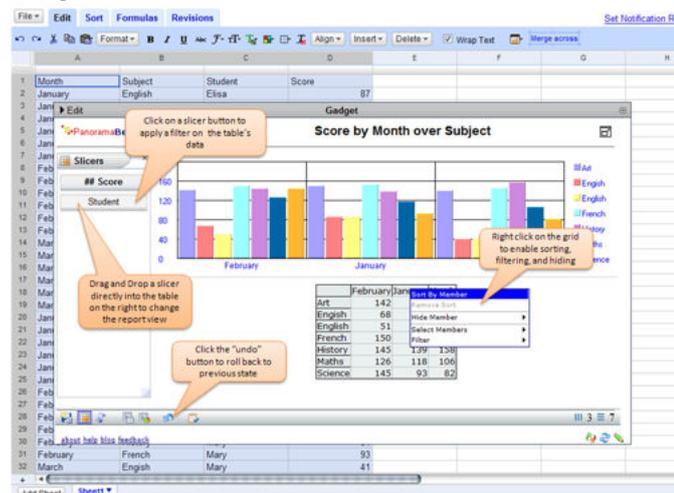
Merci à Guillaume.

A tester donc chez vous avec vos poulets.

## Analyse de données avec PivotTable

Avez-vous déjà testé la fonction [PivotTable de Microsoft Excel](#) ? C'est une fonction d'analyse des données très pratique qui permet de visiter des données et d'effectuer une analyse des données en direct. J'ai entendu sur [The Java Posse](#) que Google avait ajouté à la fonction [Google Spreadsheets](#) la possibilité de créer des PivotTables. Google Docs intègre un module développé par [la société](#)

[Panorama](#) dans Google Spreadsheets. Mes premiers tests avec Safari et Firefox ne sont pas très concluants car l'interface est en partie traduite en français. Il manque quelques icônes. Mais c'est assez prometteur.



Pour tester, [voir le tutorial en anglais](#).

## Mardi 8 avril Réunion PJUG

Avis à la population : mardi soir rendez-vous à l'Espace Lancelot (29 rue des Trois Bornes, 75011 Paris) à partir de 19h30 pour assister à la présentation sur les performances de Kirk Pepperdine.



Vous pourrez retrouver sur son [blog](#) des billets intéressants sur l'évaluation des performances sur la plateforme Java. Kirk tiendra aussi une conférence à JavaOne en mai, autant dire que nous verrons ce soir un grand monsieur. J'aurai des tonnes de billets à écrire sur les performances et les légendes urbaines autour de Java et de ses performances. Je ne compte pas le nombre de fois où j'ai été tenté de vous repeindre le Touilleur avec des articles sur les performances. J'ai arrêté de me battre car je me suis rendu compte que finalement les causes de ces discussions sont souvent: - une grosse méconnaissance de la plateforme Java, - une ignorance de l'écosystème open-source autour de Java - un désert d'objectivité - deux ou trois idées vieilles de 10 ans basées sur le Java de cette époque ... et la liste est longue. C'est dommage car le langage lui-même ne représente que 50% de ce qu'une personne qui fait du Java doit connaître. Je mettrai dans les 50% restant une connaissance des outils de développement, de gestion de projet, de moteur applicatif ou de serveurs d'applications, qui font partis pour moi maintenant de la boîte de survie du développeur Java. Imaginez un monde sans Ant, Maven, ApacheTomcat, Eclipse, Hibernate, JBoss Application Server, Spring, JSF, Log4J, Xerces, Xalan, Struts, Wicket... Allez je vois que je suis déjà parti sur un autre sujet... j'en garde pour un autre billet. Pour revenir à la soirée PJUG, si vous venez ce soir, faites-vous connaître, je serai content de discuter avec vous.

Hop une petite photo pour m'identifier :  A mardi soir

## Présentation de Kirk Pepperdine au Paris Java User Group

**Débriefing à chaud de la présentation de Kirk** Je rentre juste de la soirée du PJUG organisée ce soir autour de la présentation de [Kirk Pepperdine](#) sur Java et les performances. Je vous livre donc en vrac ce que j'ai vu, ce que j'ai entendu et aussi mes rencontres de la soirée. Tout d'abord au niveau logistique, nous étions une bonne centaine ce soir... Un peu de math à calculer le nombre de personnes par rangées, multiplié par le nombre de rangées... oui il devait y avoir au moins 80 personnes. Des personnes sont restées debout. Le buffet était offert ce soir par la société [Jaxio](#). Un nouveau sponsor était présent car j'ai reconnu une personne de Valtech qui était passée chez nous faire un audit performance. Il y avait aussi des personnes de Xebia. La présentation de Kirk était du caviar pour les geeks dans la salle. J'ai eu peur parfois qu'en allant trop dans les détails, il fasse décrocher l'assemblée. Mais les gens ont suivi, et la participation a été bonne. A propos on passera sous silence mon aller-retour au tableau, où je n'ai pas décroché un livre ou un tee-shirt... Cela m'apprendra à sortir du bois. La présentation était vraiment dynamique, et Kirk est un orateur surprenant qui n'est pas là pour nous vendre de la soupe, mais pour nous donner des nouvelles du front... Et il y a du boulot qui nous attend. Kirk Pepperdine a aussi donné son avis sur les bases de

données. De son point de vue, le prix de la mémoire et les évolutions technologiques (mémoires de type NAND ou Flash) font que d'ici quelques temps l'utilité d'une base de donnée se pose. On se force à utiliser un monde relationnel qui n'est pas familier au monde objet Java. Vous êtes vous déjà rendu compte de la différence entre un modèle Java, avec héritage, et un modèle relationnel en base ? Il a aussi conseillé pour les malheureux condamnés à travailler sur Windows de faire l'expérience de désactiver le Swap disque puis de lancer Eclipse... Oh mais comme c'est rapide... Dingue non ? L'explication est que la gestion de la mémoire Java ne peut pas déterminer si un élément paginé est stocké dans la mémoire vive ou sur le disque. Donc si vous avez des comportements bizarres avec Eclipse vous pouvez tenter de désactiver le swap de Windows, vous pouvez passer au Mac (très bonne idée) et vous pouvez aussi passer à [IDEA Intellij](#) qui est l'éditeur qu'utilise Kirk... J'ai eu envie de dire à tout le monde : *jetez Eclipse, dépensez de l'argent et achetez IDEA Intellij... On est en 2008 et il y a la télé couleur aussi dans le monde des éditeurs.* Mais finalement je suis resté assis. J'ai été ridicule une fois, j'allais pas remettre cela. Je vous donnerai un cours la prochaine fois sur le fait qu'une String Java, contrairement à ce que vous croyez, n'est pas un objet immuable... On peut changer sa valeur sans problèmes... Je me réserve un billet là dessus. Revenons à nos moutons. J'ai retenu durant sa présentation qu'avec l'arrivée des processeurs multi-coeurs, les choses vont changer. Java est le premier langage à proposer un système de virtualisation du code, avant même que l'on ne parle de virtualisation de machine. Il ajoute une abstraction entre le code machine et le code du logiciel. Ce faisant, les éditeurs de JVM ont optimisé et tiré parti de l'architecture matériel sous-jacente. Une JVM génère au final un bytecode qui sera adapté au processeur, à l'architecture et au système d'exploitation qui fait tourner la class Java. C'est faisable aussi en C++ ou en C, au prix d'une optimisation et d'un travail qui est à la charge du développeur. Cependant, et c'est ce qu'il a tenu à faire passer, avec l'arrivée des processeurs multi-coeurs nous risquons de voir surgir de nouveaux soucis apportés par l'architecture parallèle. En effet, il n'existe pas encore dans le langage Java de quoi délimiter du code afin d'aider la JVM ou même le processeur, à exécuter en parallèle des bouts de code. Enfin il nous a cependant montré que sur Java 7, des marqueurs ainsi que des mots clés de la JVM permettront de modifier le comportement et de l'optimiser... J'ai l'impression de ressortir des soucis qui n'existaient pas il y a quelques années... De son point de vue, le matériel est maintenant quelque chose que nous devons à nouveau prendre en compte dès l'écriture de code, lorsque nous cherchons les meilleurs performances... Cela me fait penser qu'aujourd'hui nous arrivons à peine à faire écrire du code correctement et que des outils comme JProfiler se vendent très bien pour identifier et corriger les erreurs de codage... Donc je ne sais pas. C'est à réfléchir. Nous étions ce soir en face d'une personne dont le coeur de métier est la performance. Il a distillé des réflexions très intéressantes sur la manière de prendre des mesures. Qu'est-ce qu'après tout, effectuer un benchmark ? Qu'entends-t-on par audit et mesures de performances ? C'est un vrai métier, et si j'avais un conseil à donner à mon big boss, c'est d'appeler un vrai médecin pour soigner le malade. Pas de faire du bricolage avec 3 graphes... Je referme la parenthèse. Revenons à Kirk. Je vous laisse un lien à lire car je crois que chaque développeur Java devrait l'avoir lu une fois. Cela vous donnera une description fine du fonctionnement de la JVM.

[http://www.jucs.org/jucs\\_11\\_7/an\\_experimental\\_evaluation\\_of\\_jucs\\_11\\_7\\_1291\\_1310\\_faustino.html](http://www.jucs.org/jucs_11_7/an_experimental_evaluation_of_jucs_11_7_1291_1310_faustino.html)

La page des options de réglage de la JVM de SUN, toujours utile lorsque vous faites des micro-benchmarks afin de voir comment désactiver le Garbage collector en augmentant la taille de la heap. <http://java.sun.com/javase/technologies/hotspot/vmoptions.jsp> Bon, et donc je terminerai par une petite private joke : "Franck" ... "Franck" "is" "very sad about immutable String" ... true false ... "very sad about immutable String" On en parle la prochaine fois... Bonne nuit ou bonne journée

## Les news du Jeudi: Xebia Poker, Scrum, la vie...

Ce soir aura lieu la soirée Xebia Poker. Arrivé 16 ème j'ai gagné mon ticket d'entrée et donc je serai ce soir au Café Barge pour tenter de gagner une place en finale... On verra bien. Comme plusieurs personnes me l'ont fait remarqué, le dernier article publié sur la page d'accueil du Touilleur datait du 11 mars, alors que j'ai blogué pas mal depuis. Désolé pour ce couac. C'est maintenant réparé, vous devriez retrouver une page d'accueil à jour. Vous pouvez me signaler les soucis en utilisant mon email qui est en haut à droite sur le logo du Touilleur. Et vous pouvez aussi me contacter pour me parler de la pluie et du beau temps, il n'y a pas de soucis. Je sponsorise aussi les candidatures motivées, comme pour une personne qui m'a contacté et qui cherchait du boulot dans la finance avec du Java. Avis aux amateurs donc. Qui ne tente rien n'a rien :) Pour revenir au look du blog, j'ai préparé un nouveau thème pour le Touilleur. La section Archives prenant un peu trop de place, je vais la remplacer par un nuage de tags, ce qui me semble plus pertinent. Migration prévue ce week-end. Côté Scrum, c'est sur les rails pour mon équipe depuis [la certification Scrum](#). Après quelques aventures, je trouve que la semaine dernière a été assez sympa. J'ai conscience que pour l'équipe c'est une révolution, et qu'il y a beaucoup de travail de communication de ma part à effectuer. Mais je pense qu'ils y croient. Ici au travail, depuis la formation, j'ai pris conscience que trop d'email tue l'email. Je serai tenté de proposer un "[No Mail Friday](#)", "Pas de mail le vendredi". L'outil électronique a tendance à dégrader la communication et les relations entre les gens. Je n'ai jamais autant pris l'ascenseur et discuté avec les gens qu'avant. Il est vital de discuter, je pense que nous souffrons d'un problème de communication où le mail devient le rempart pour ne pas s'engager, pour ne pas s'exposer. J'ai décidé de me déplacer, discuter avec les gens, ou téléphoner. Terminé la perte de temps à jouer au "Reply-All" avec une pile de manager... Hier je suis entré dans le bureau du nouveau big boss. Madame Boss est partie, remplacée par Monsieur Boss. Et au dessus, GrandBoss qui est reparti au Luxembourg a été remplacé par MisterHeadBoss. Bref vu que MisterHeadBoss est un ancien de Netscape, je suis passé discuter avec lui à propos d'archéologie de l'informatique : Netscape Applcation Server, la suite iPlanet, Netscape Directory Server qui était l'un des premiers annuaires LDAP... A l'époque cette suite iPlanet était la meilleure offre pour avoir un serveur web, un annuaire LDAP, un agenda d'entreprise ou un serveur de certificats. La phase 2 sera de lui parler de Scrum. S'il veut soigner la bête, il faut changer de médecin et de médicaments. Basculer toutes les équipes n'est pas une bonne idée. Par contre, en parler et mettre en place des

actions pour certains projets, à mon avis cela pourrait apporter une bouffée d'air frais à nos développeurs qui tournent en rond comme dans un bocal. Et vous ? Est-ce que ça scrum ? Comment va la vie ?

## Résultats de la soirée Xebia Poker

Imaginez une cinquantaine de personnes ayant une bonne connaissance de Java et de J2EE... Ajoutez ensuite un jeu de poker, le Texas Hold'em. Et pour terminer, une organisation très pro par [Xebia](#), c'était le premier tournoi Xebia Poker 2008. Voilà la recette magique de la soirée de ce soir. L'heureux gagnant d'un voyage à JavaOne est Sébastien Guerlet avec qui j'ai eu le hasard de discuter au début de la soirée, sans savoir qu'il deviendrait le gagnant. Il travaille dans une société de consultant qui intègre pour des grands comptes le logiciel [SalesForce.com](#), un moteur de CRM [SAAS](#) (Service As A Service). Arrivé 47<sup>ème</sup> au Quizz, il a remporté le tournoi après 3 heures de jeu. La fin était vraiment sympa, je pense que vous trouverez des photos sur [le blog de Xebia](#) demain matin. J'ai croisé des lecteurs du Touilleur et c'est toujours sympa de discuter avec les personnes qui lisent mon charabia. Je suis tombé aussi sur Dimitri Baeli, avec qui j'ai travaillé il y a 6 ans dans l'une des start-ups de l'époque (Dotvision) qui avait fait la plus grosse levée de fond de la place de Paris en 99... Souvenir... JavaOne 2000... Ah c'était cool. Quant à moi j'ai terminé 24<sup>ème</sup>... 56 minutes de jeu... Mais du coup j'ai ensuite eu le temps de discuter avec pas mal de monde. Et donc voici ce à quoi je voulais venir : je voulais partager les idées et les suggestions des joueurs, qui ont tous été bluffés par l'organisation. Diner et buffet très sympa. - Guillaume : "Ca aurait été bien d'avoir un goodies comme un tee-shirt" - Dimitri : "L'an prochain, si vous pouvez faire jouer 30mn les gens sans commencer le tournoi, ça sera plus facile" - Nicolas : "Je propose que Sébastien, le vainqueur, soit requalifié l'an prochain automatiquement" - Christophe "On a tous des badges avec nos noms et marqué Xebia... du coup je ne vois pas qui sont les gens de Xebia." Et moi de répondre : "ce sont ceux qui n'ont pas de badge :-)" - Le croupier : "Pour des amateurs, ça joue bien. Y'en a qui se débrouillent vraiment bien." - Un gars : "On pourrait avoir des cartes de poker avec le logo Xebia ?" Allez je vais dormir car demain Scrum Stand-up meeting à 09h30, suivi d'une réunion et ensuite vendredi soir, encore soirée Java (j'ai pas arrêté cette semaine) Mission remplie, billet posté à 1h36, encore un point de santé mentale en moins, à vous les studios, Bye

## Changement de look du Touilleur

C'est parti, je m'attaque ce week-end à la refonte du Touilleur, que ce soit au niveau de l'apparence comme de la construction des pages. J'ai pris le template [Yoghourt](#) distribué sous licence Creative Commons Attributions v2.5 dont l'auteur original est Jeremie Tisseau web designer de la société [Web-Kreation](#). Vive le Mac, avec Adobe Illustrator et Photoshop on arrive à faire de beaux logos



(enfin de mon point de vue d'informaticien) comme celui-ci: J'ai créé la tasse de café avec Illustrator, 2 bonnes heures de travail pour arriver à un "machin" qui ressemble à une tasse comme je voulais. Allez j'y retourne, je reposterai dans la journée. [update] Voilà c'est terminé. Quelques bandeaux Google à remettre par ci, par là (qui me permettent de couvrir les frais d'hébergement) et aussi quelques vérifications sur les vieilles pages. J'ai ajouté un nuage de tag sur la page principale, c'est finalement plus intéressant que les archives, et cela donne une idée du contenu du blog depuis que je suis passé à WordPress en janvier dernier. Côté statistiques, sur Google Analytics il y a entre 80 et 100 visiteurs uniques par jour. Par mois, il y a eu 1789 visiteurs différents en mars. Etape suivante : faire des cartes de visite sur VistaPrint.fr afin de distinguer le Nicolas de Reuters de celui du Touilleur.

## Scrum et l'Agile Alliance

Suite à ma formation Scrum, j'ai reçu mercredi la certification ainsi que mes codes d'accès au site [Scrum Alliance](#).

Attention les yeux, j'ai maintenant le droit d'afficher le logo sur mon site ou sur mon CV:



Hop la partie "je vous montre mon beau logo, fier comme un prince" est terminée.

Concernant le site [Scrum Alliance](#), j'aimerais vous en parler car le site contient de nombreux documents utiles dans la section [Resources](#). Voici parmi ceux que j'ai lu rapidement, des documents qui peuvent vous aider à apprendre les principes de Scrum, ou tout du moins à savoir de quoi cela parle :

1. [Scrum Primer](#) est un document d'une vingtaine de page, écrit par Yahoo! Inc. Excellent guide qui vous donne toutes les caractéristiques de Scrum. Il mériterait une traduction en français.
2. [Explaining Scrum to your Management Team](#) est un document qui vous permet de présenter, justifier et argumenter l'intérêt de Scrum. Je parlerai de ce document plus tard dans un billet à part, car j'ai des idées sur la question...

Pour les courageux qui ont regardé les 14 pages de "Resources", vous trouverez à la fin quelques images humoristiques dont celle-ci, que j'ai reprise le blog :

A gauche Jeff Sutherland et à droite Ken Schwaber



Si par ailleurs vous cherchez des livres sur Scrum, allez jeter un oeil sur [la liste Amazon d'Eric Lefevre](#).

## Jazon, conférence sur Java en Suisse en juin



Avis aux amateurs, [Jazon](#), une conférence internationale aura lieu en Suisse du 23 au 27 juin 2008. Si vous n'avez pas la chance d'aller à Java One à San Francisco d'ici quelques semaines, ou Java Polis en Belgique en décembre, c'est peut-être l'occasion de participer à un conférence sympathique, à en juger par le programme.

### Les orateurs:

La liste complète est [sur le site](#). Et elle risque d'évoluer d'ici au mois de juin. Parmi les personnes listées, j'ai noté la présence de:

- o Bela Ban, JBoss - Red Hat, l'un des auteurs de JGroups. Il présentera l'actualité sur les Clusters.
- o Rod Johnson, Monsieur Spring, interviendra sur l'un de ses sujets favoris. A ce propos je vous informe que [SpringSource France](#) a ouvert ses portes, dirigé par Julien Dubois, coauteur du très bon livre "[Spring par la pratique](#)"
- o Joshua Bloch de Google
- o Martin Marinscheck sur l'intégration d'Ajax dans Apache MyFaces (c'est pour moi)
- o James Ward sur les applications clients riches (RIA, encore mon domaine)
- o Emmanuel Bernard, JBoss - RedHat, Hibernate Validator / Hibernate Search / EJB3 et grande figure française de JBoss (à mon sens)
- o Tout un groupe de personne de SUN Microsystems, dont quelques figures comme Jerome Dochez (GlassFish) ou Simon Phipps que j'ai rencontré chez Reuters.
- o Dion Almaer, fondateur de l'excellent site [Ajaxian](#) qui est l'une de mes sources de travail au quotidien.

### Le lieu:

Les conférences auront lieu en Suisse à Zurich. Un billet d'avion ou de train sera certainement plus accessible qu'un voyage jusqu'à San Francisco.

### Le prix:

Le prix d'entrée est de 1085 EUR pour les 3 jours ou 835 EUR si vous vous inscrivez avant le 30 avril prochain. Si vous venez à 5, une entrée est offerte. Attention je vous invite à vérifier sur le site. J'ai repris ces informations dans mon post pour votre Google Reader. Que les étudiants s'assoient et respirent, il y a des prix spéciaux pour eux et donc c'est jouable. C'est 3 fois moins cher, si vous avez moins de 24 ans cependant. Moi j'ai juste un peu plus... hop...

### JUG (Java user group):

Si vous venez de la part d'un Java User Group comme le notre, [Paris Java User Group](#), vous pouvez obtenir un bon de réduction de 25% et avoir une entrée pour les 3 jours à 785 EUR, voir même 625 EUR si vous avez un bon avant le 30 avril. Pour obtenir un voucher (réduction) il faut s'abonner à [la mailing list du PJUG](#) puis envoyer un email à david@parisjug.org.

### Bon, et toi tu y vas ?

Je vous tiens au courant mais j'y travaille.

### Et la France ?

Voyez un peu ce que j'ai listé : [JavaPolis](#) en Belgique en décembre 2007 et dont la nouvelle édition est annoncée pour fin 2008. [Jazoon](#) à Zurich du 23 au 26 juin 2008 en Suisse. [SpringOne](#) le 11 et 12 juin 2008 à Anvers en Belgique. En mars dernier avait lieu à Londres [QCon 2008](#) du site [InfoQ](#). ... et chez nous ? ? ? En étant tout à fait sérieux, à part les [Sun Tech Days en mars 2007](#) l'an passé à Paris, chez nous il ne se passe pas grand chose. Moi cela m'inquiète... Dîtes-moi que je suis passé à côté d'un événement cet année... Est-ce la barrière de la langue ? le peu d'intérêt de nos compatriotes pour Java ? Je rejoins Antonio qui a fait remarquer qu'en Italie il y a 6 JUG et seulement 3 en France (Tours, Paris et le [JUG virtuel de developpez.com](#)) Vu le nombre de français dans la communauté francophone, il est dommage de ne pas avoir d'événements de qualité à Paris... Mais tant pis, on ira voir nos voisins avec un grand plaisir, j'adore la Belgique et je ne suis pas allé en Suisse depuis 15 ans. Alors c'est parti.

## Spring 2.5 : une histoire

Il est temps de réparer un oubli : je n'ai presque jamais parlé de Spring ici. Depuis février nous avons mis en production un projet basé sur Spring 2.5. La mise en production s'est effectuée cette semaine à New-York, chez un gros hébergeur financier, client de nos produits. Un de nos développeurs est là-bas et termine les tests.

Tout a commencé en décembre. Notre framework web génère des fichiers Excel et PDF à partir de rapports financiers qui sont dans un format propriétaire. Cela fonctionne sans soucis pour des petits rapports. L'utilisateur navigue dans l'interface web, sélectionne un rapport puis lance une action « Export to Excel ». Une servlet effectue alors le traitement. Classique, pas de quoi se relever la nuit.

Cependant lorsque le volume de données est trop important, tout un ensemble de soucis apparaissent : l'utilisateur est bloqué pendant le traitement de l'export. La mémoire et le CPU du serveur d'application est fortement sollicité. Ce qui conduit même à ralentir les autres utilisateurs connectés au client léger web...

Nous avons tout d'abord désynchronisé cette partie. Via un MDB, le code de l'export est exécuté de manière asynchrone. L'utilisateur est notifié dans son navigateur par un icône lorsque le traitement est terminé. Cette piste n'est pas la bonne : vous ne retirez pas la charge et le besoin en puissance pour créer l'export. Bref retour à la case départ.

Je cherche une solution facile à mettre en place par l'équipe en 4 semaines. Le client attend une solution qui tienne la route pour passer en production, donc pas le temps de tergiverser. Et là, la solution la plus simple et la plus rapide : Spring 2.5 !

Dans la même semaine je croise Julien Dubois qui est l'auteur du très bon livre [Spring par la Pratique](#) aux éditions Eyrolles. En effet toute la partie JMS de Spring nous servira ensuite à implémenter une solution propre.

En 7 jours, nous écrivons une application Spring simple, qui utilise Spring-JMS. Cette application que j'ai baptisé « Kocktail » se connecte au serveur JMS du serveur d'application, dépile d'une Queue un message, génère le fichier Excel ou PDF puis reposte sur une Topic un message de résultat, indiquant que le rapport est prêt. Kocktail car l'application charge un rapport financier, un template de rendu et génère un fichier Excel ou PDF assez complexe.

Les fichiers générés sont assez volumineux. Nous avons donc décidé de partager un répertoire entre le serveur d'application et la partie Kocktail. Il était or de question d'envoyer via JMS des fichiers Excel de 3 à 30 Mo.

Au final nous avons fait un passage en production au début de cette semaine. Avec quelques petits ajustements mais on est resté sur notre solution Spring. La solution peut monter en charge car il est possible de lancer plusieurs processus Java « Kocktail ». Sur un serveur multi-core cela donne de biens meilleurs résultats. Et surtout, du côté de l'interface utilisateur, nous n'avons plus de ralentissement. Je sais que c'est la base d'un ESB. Mais allez tout changer en un mois... impossible pour moi. J'ai fait au plus simple.

Mes impressions sur Spring : facile à mettre en œuvre, souple et pratique. La première piste avec les MDB a mis en lumière les soucis de la plateforme J2EE : obligé de déclarer tout un ensemble de XML, de s'assurer que cela fonctionne ensuite avec JBoss, Weblogic et Websphere... Et le nom des Queues/Topics fixés en dur dans l'EAR... non merci.

Avec Spring, nous avons ajouté une partie Spring client JMS dans notre framework afin que lors de l'export, un message soit posté sur une Queue configuré dans Weblogic ou JBoss. Facile, simple et pratique. Nous n'avons pas eu besoin de sortir une solution plus puissante basée sur [Jencks](#) par exemple. Notre besoin est assez simple.

En résumé, je pense qu'il est important d'apprendre Spring, qui permet d'attaquer des problèmes J2EE en partie causés par la lourdeur de cette architecture.

## Fin de notre premier Sprint, débriefing

C'est fait. 2 semaines se sont écoulées et nous avons terminé notre premier Sprint Scrum avec mon équipe. Vendredi nous avons donc fait une Sprint Retrospective et ensuite une Sprint Review. Voici un premier retour sur expérience.

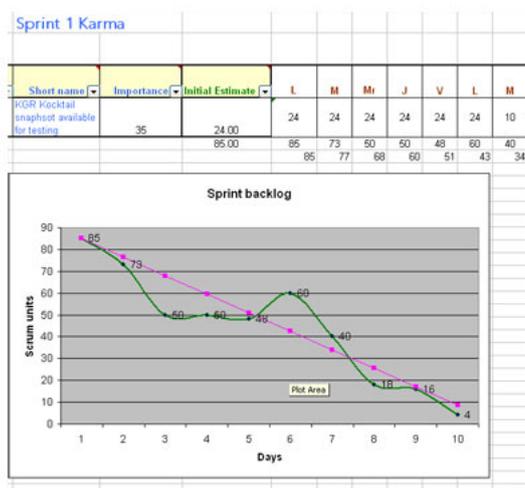
J'ai commencé pendant ces 2 semaines par mettre en place Scrum au sein de l'équipe que j'encadre. 3 développeurs et un stagiaire. Nous avons défini les différentes tâches il y a 2 semaines. Ensuite effectué les estimations pour appliquer Scrum. L'objectif du sprint était de sortir la fonctionnalité [Kocktail](#) pour un client à New-York ainsi que de mettre en place un système pour construire une suite de produit (plusieurs de nos produits ensemble) à la Microsoft Office.

Notre Sprint backlog initial contenait aussi quelques éléments mal définis que l'équipe a refusés. Et ils ont bien fait. Au moment où je les avais ajoutés, ces demandes étaient "super-urgentes". Finalement nous aurions été perturbés par ces tâches qui, d'un point de vue business, n'apportaient rien.

Pour les rôles j'ai joué le rôle de Product Owner. En allant voir les différentes équipes qui utilisent notre framework pour construire leurs produits, et avec les chefs de produits, je définis les demandes de corrections et d'améliorations, et nous nous accordons sur la priorité. Cela afin d'éviter que les développeurs des équipes tierces passent leur temps à venir nous voir ou nous téléphoner. De ce point de vue, j'ai déchargé l'équipe de répondre directement aux emails et aux sollicitations. L'idée est de ne pas les perturber pendant le travail... en théorie.

Avant de vous raconter ces 2 semaines, voici l'export Excel du Sprint Backlog terminé. Pour ceux qui ne connaissent pas, voici le principe : en début de Sprint, l'équipe s'engage à réaliser un certain nombre de points Scrum. Ici 85 pour l'équipe. Chaque matin nous effectuons une réunion debout de 3 mn devant un tableau. Je demande à chaque développeur : "Peux-tu raconter ce que tu as fait hier **aux autres** ? Que vas-tu faire aujourd'hui ? Est-ce que tu as tout ce qu'il te faut pour travailler aujourd'hui ?"

J'insiste pour que chacun parle aux autres et pas uniquement à moi. Cela force l'équipe à bien faire circuler l'information. A l'issue de ce tour rapide, nous mettons ensemble à jour le "reste à faire". La courbe donne donc chaque jour la quantité estimée de travail à effectuer. Cela ne prend pas de temps car chaque personne ne donne une mise que sur la tâche qu'elle effectue.



Les premiers jours montrent que l'équipe part plus rapidement que la courbe théorique. En effet, les 3 premiers jours les tâches à effectuer sont les plus importantes, et c'est plus parce que l'équipe doit livrer quelque chose au client à New-York que parce qu'ils seraient allés plus vite. Le 6ème jour, un de mes clients favoris sort du bois avec un bug important et urgent pour le coup. Nous sommes alors obligés de perturber le Sprint pour qu'un des développeurs travaille sur ce point. J'ai hésité entre ne pas le modéliser et l'ajouter. J'ai décidé d'ajouter au Sprint Backlog cette activité, et de demander à la personne de l'estimer. Forcément cela fausse la quantité de travail initial à faire... Mais d'un autre côté je me disais que nous pourrions ainsi voir l'impact de l'arrivée de cette demande urgente de ce client. Ensuite on constate que la courbe reprend la tendance, l'équipe terminant chacune des tâches petit à petit.

Durant le sprint par contre, rien n'interdit aux développeurs de modifier la quotation d'un élément. L'équipe le matin me disait que l'élément "Installation Weblogic - 12 points" coûtera plutôt "16 points" car il faut prévoir l'installation d'une SUN, ce qui n'avait pas été fait. La première estimation était fautive, sans que ce ne soit de notre faute. Et ainsi de suite...

Enfin on voit qu'il reste 4 points à faire. En effet une des tâches ne s'est pas terminée car les intervenants extérieurs dont nous avons besoin pour le packaging n'étaient pas prêts. Elle est donc repoussée au sprint suivant.

#### Le bilan du Sprint pour le Scrum Master :

En tant que Scrum Master (et Project Leader) j'ai une vision quotidienne très claire de l'activité de l'équipe. Aucun micro-management nécessaire. Inutile d'utiliser Microsoft Project ou un diagramme de Gantt. A quoi bon perdre une heure chaque matin à jouer avec 3 cases grises sur Project ? Est-ce que cela sert ? Faire un suivi de projet avec du Gantt me semble complètement débile. Vous allez faire le tour de l'équipe et chacun vous dit ce qu'il a fait. Ensuite vous remplissez ce magnifique outil et finalement, au moment où votre diagramme est prêt il est déjà faux car l'équipe, elle, avance... Ok pour la gestion macro-projet. Par contre, débile pour suivre au quotidien l'activité d'un développeur.

Sinon durant le sprint, difficile pour moi de ne pas dire "*toi tu peux prendre ce bug, et toi tu peux aider ce client sur ce point...*". L'équipe s'est partagé le travail, je ne les ai pas trop perturbés là-dessus. Cependant, lorsqu'un des membres de l'équipe termine une tâche et que le sprint backlog est vide, il faut que j'alimente une drop-zone sur notre tableau avec d'autres tâches non prévues. Ce que j'ai fait la deuxième semaine. Cela permet de laisser l'équipe s'auto-gérer. J'ai essayé de demander aux développeurs de s'aider mutuellement et de ne pas dépiler du tableau toutes les tâches, ce qu'ils ont fait.

Les limites de l'auto-gestion, surtout pour un framework comme le nôtre, c'est la gestion des incidents qui viennent de l'extérieur. Avec une durée de Sprint de 2 semaines, je pense couvrir et masquer à l'équipe une partie des bugs et des demandes classiques. Donc objectif rempli. Par contre, le souci c'est "*faites-moi un patch ce soir sinon mon client se coupe un bras*". Et là c'est la vie, il faut gérer. La solution 1 : c'est un obstacle et le Scrum-Master se débrouille. Solution 2 : on arrête le Sprint et toute l'équipe se consacre à ce problème. Solution 3 : je sors un des joueurs de l'équipe, et il s'en charge. Un peu comme sur un terrain de foot. C'est la solution 3 qui me semble la moins violente pour le Sprint.

#### **La Sprint Retrospective :**

Le dernier jour d'un Sprint Scrum, le Product Owner, le Scrum Master et l'équipe se retrouvent pour faire la démo des éléments développés pendant les 2 semaines. Vendredi après-midi nous sommes donc installés dans une salle et avec un vidéo projecteur, nous avons passé en revue les items. Dans ma tête au départ, je me suis dit que cela allait être une perte de temps... Mais non !

Durant les présentations, chaque personne a expliqué et surtout répondu aux questions des autres sur son boulot. On a fait passer l'information bien plus efficacement que d'habitude. Nous avons même parlé pour une correction de bugs d'idées supplémentaires si la correction ne convenait pas...

Le plus heureux dans l'histoire c'est le stagiaire. Arrivé il y a 3 semaines, il a la chance de voir de A à Z le travail d'une équipe de développement, sans que nous soyons obligés de faire un gros tutorat comme l'an passé. Il nage un peu, mais je pense qu'il est super content de voir comment nous travaillons.

Donc ne surtout pas annuler la Sprint Retrospective. C'était un moment sympa.

#### **La Sprint Review :**

Ensuite sur un tableau blanc, je dessine 2 colonnes. Améliorations/Dégradations apportées par Scrum. Je fais un tour de table, c'est un peu le silence. Le premier point sur lequel nous discutons est la trop grande fréquence des stand-up meetings. L'équipe se demande si nous devons vraiment se voir chaque matin.

L'équipe n'est pas contre la réunion et son fonctionnement : elle prend 10mn avant le café de 10H, et ensuite on est tranquille pour la journée. C'est juste le fait de se réunir pour dire... quoi ?

N'ayant pas encore assez d'expériences, difficile pour moi de dire s'il faut modifier ou non. Cependant je reste convaincu qu'il faut maintenir ce rythme quotidien. Je retiens de la certification Scrum, que Jeff disait de ne pas changer les règles du jeu. Lorsque vous jouez au foot, vous n'allez pas arrêter de jouer à 60mn de jeu ou décider que le défenseur peut prendre la balle avec la main... Donc on reste sur l'idée d'un stand-up meeting chaque jour, et surtout de mettre à jour le backlog pour suivre l'évolution du projet. J'en reparlerai certainement dans quelques semaines.

#### **Scrum on continue ?**

J'ai posé la question. Tout d'abord nous travaillons depuis 2 ans en mode itératif, avec une release toutes les 3 semaines environ. Donc ce n'est pas une grosse révolution pour ma propre équipe. Nous nous sommes accordés pour dire que de toutes les façons, nous étions déjà dans un processus Agile.

L'équipe est ok pour continuer, moi je vais avoir encore plus de travail pour affiner mon processus et simplifier le travail de l'équipe. C'est aussi mon premier "sprint" et il faut être patient pour voir le retour. Là dessus tout le monde est d'accord.

#### **Et les autres équipes ?**

J'en ai discuté avec d'autres chef de projets... Scrum (ou scrumble comme m'a dit un collègue) est encore assez abstrait pour une grande partie des différents départements. J'ai une présentation de prévue en mai à un ensemble de chefs de projets, avec lesquels j'ai effectué une formation. Je vais pas présenter Scrum en mode "*Convertissez-vous mes frères*" mais plutôt en mode "*Eh ! au fait si vous voulez tenter quelque chose...*". Un autre axe est d'arriver à en parler au top-management. C'est en bonne voie mais je marche sur des œufs.

Enfin voilà. Si vous êtes encore sur Word et Visio à essayer de décrire votre logiciel ou votre architecture, pensez que votre document ne va exprimer que 60% de vos idées, que 5 personnes au maximum seront capables de le lire et de le comprendre, 2 personnes seulement le liront en entier, et que le client final de toutes les façons n'aura rien compris. Donc vous me faites plaisir, vous arrêtez d'écrire du charabia et de l'UML et dès lundi vous rediscutez avec le client. Vous verrez qu'il a encore changé d'avis...

Bon week-end

## **SpringSource et Rod Johnson**

Billet intéressant à lire pour les architectes JEE : Rod Johnson le créateur du framework Spring expose dans [ce billet sur son blog](#) sa vision à plus long terme sur la faune Java et le monde de l'entreprise. Rod est à fond pour [le support des modules dans JEE 6](#), même s'il est encore tôt pour savoir le contenu de ces profils. J'étais étonné de voir qu'en décembre Rod blogait ceci à propos de JBoss et de Tomcat:

"  
*Also from our observations in customer accounts, I would expect that a large part of the JBoss numbers are actually Tomcat. Many users who think they're using JBoss are actually using a less efficient form of Tomcat. One example—the French Tax Office's online taxation service—is a public reference for both JBoss and Spring.*" ([article complet ici](#))

J'ai un peu de mal avec l'affirmation ci-dessus. Tout d'abord JBoss est un sponsor de Tomcat depuis 3 ans. [Mladen Turk](#) a écrit le module APR qui permet de faire fonctionner Apache avec un serveur Tomcat, et il fait partie de l'équipe JBoss. Enfin la version JBoss Web est une version avec support JBoss qui est composée d'Apache et de Tomcat, justement utilisé par nous, gentil contribuable, lorsque nous effectuons notre déclaration d'impôts. Il serait intéressant d'avoir le retour d'une personne du ministère des finances ou de JBoss pour nous éclairer là-dessus.

Aujourd'hui il serait dommage de passer à côté de Spring, de la puissance de l'inversion de contrôle et de la souplesse de ce framework. Avant tout, pour moi, c'est un framework d'assemblage. Il ne m'offre pas de solution de mapping object-relationnel. En même temps, j'ai le choix d'utiliser librement Hibernate, TopLink ou autre. La première chose qui me séduit c'est que je n'enferme pas ma solution dans une architecture compliquée, sans possibilité d'évolutions. Je détache mon produit des griffes des éditeurs comme BEA ou IBM. En effet aujourd'hui, nous payons une fortune des accords OEM, qui, je le dis sans problèmes, ne me servent à rien. Même le support ne sert à rien... Il est payé par le client final qui achète la solution. Mais quelque part, j'ai un peu de mal à me dire que j'engraisse un éditeur avec une rente viagère, le temps que l'architecture J2EE basée sur les EJB 2.1 s'arrête d'exister... On en est là aujourd'hui.

SpringSource, l'entreprise, est devenu une entreprise qui doit maintenant monétiser Spring. Cela passe par des formations, des certifications, du support d'architecture. SpringSource va sur le même chemin que JBoss Inc., l'entreprise. Le support de JBoss est indispensable à nous, éditeurs de logiciels. En effet sans l'existence de la société JBoss et d'un accord entre ma société et JBoss, nous ne pourrions pas vendre nos produits à nos clients. Tout simplement.

Donc en terme de "faut-il un SpringSource..." si demain nous avons des produits architecturés sur Spring, il sera obligatoire pour nous, en tant qu'éditeur, d'aller taper à la porte de SpringSource afin de mettre en place un accord.

La difficulté que je vois, c'est que JBoss offre plusieurs produits, SpringSource offre un framework d'intégration pour grouper plusieurs technologies. Il y a cependant des projets 100% Spring qui sont vraiment biens, à mon sens. Mais on ne peut pas imaginer que Spring va réécrire un moteur de persistance, de transactions ou de sécurité. Spring est avant tout un outil d'assemblage très flexible. Du coup, lorsque nous envisageons le support, qu'allons-nous vraiment couvrir ? Lorsqu'un client signale un bug, comment identifier ce qui est Spring de ce qui est Hibernate ?

En conclusion donc, il est intéressant de suivre ce qu'il va se passer autour de Java EE 6 et de la position de SpringSource et de JBoss. Il n'y a pas de compétition entre les 2, mais même s'il y avait un embryon de compétition, c'est toujours bon pour le business...

## Actu des blogs Java français

Petit tour rapide ce matin des différents blogs auxquels je suis abonné. Arnaud Héritier d'Octo effectuera [une présentation sur Maven](#) le 13 mai au Paris Java User Group. Xébia cite [un article d'InfoQ](#) à lire sur la mise en place de Scrum en Chine. Le retour sur expérience et les témoignages de l'article sont intéressants. Il est intéressant de voir que sur les projets où Scrum n'a pas fonctionné, l'une des raisons est le fait que l'on ne devient pas Agile en lisant 3 articles sur Scrum. Il faut vraiment prévoir un effort et un travail complet pour que cela fonctionne. Sur le blog "[Scrum - Méthodes Agiles](#)" de Claude Aubry, vous trouverez un article très intéressant pour contractualiser les prestations Agile. Comment répondre à un contrat au forfait avec une méthode Agile ? Son blog est une mine d'or, j'avais parlé à la formation Xebia d'un outil développé par des étudiants encadrés par Claude Aubry : IceScrum. Allez voir [leur site](#), l'outil est français, il faut en parler. Sinon j'ai lu aussi "[J2EE Short-comings paved way for high-performance choices](#)" sur [TheServerSide.com](#) à propos des performances et de J2EE. On y parle de [Tangosol](#) dont Kirk Pepperdine nous avait touché un mot. Une petite start-up de geeks rachetée à un prix de fou par Oracle. Tangosol propose une solution de grilles en mémoire. Là où en tant que monsieur [Thomson-Reuters](#) je lève un oeil c'est lorsque je lis le texte suivant:

*In fact, at TSS JS, Davies described a complex transaction handler his team built that combined Artix Data Services, Azul hardware, GigaSpaces, and Mule software.*

Artix et les solutions de Iona sont déjà dans la place chez nous... Sinon exercice sympa ce matin : je lance Monster.fr ou le site lesjeudi.com. Je tape simplement "Scrum" afin de voir la liste des offres d'emploi qui contiennent "scrum". Je me demandais hier soir si des offres avec simplement le mot clé "Scrum" existaient ? Résultat : 10 offres sur Monster et 9 sur lesjeudi.com. Je vais mettre à jour ce billet dans la journée... A toute à l'heure.

## JSR-311 JAX-RS REST une histoire de

# RESTaurant

Je vous propose de vous parler de REST, une architecture intéressante pour les services Webs. Après avoir expliqué les principes de REST, je vous propose de jeter un oeil à la spécification [JSR-311 JAX-RS Java API for RESTful Services](#) (mince encore du java...).

## Introduction à REST

Representational State Transfert (REST) est un type d'architecture pour les systèmes distribués qui propose de représenter le modèle de données et les services sous la forme de ressources. L'URI (qui fait partie d'une URL) doit représenter une ressource unique sur le système. L'idée est de se reposer sur des méthodes standards du protocole HTTP pour manipuler une ressource. C'est une architecture presque sans état du côté du serveur. Les objets sont manipulés plutôt que d'utiliser des verbes pour modifier des ressources.

### Qu'est-ce qu'une ressource ?

Une ressource est une entité facilement identifiable dans un système informatique. Ce peut être un objet du domaine comme un Restaurant mais aussi un service comme un RestaurantFormSearch par exemple. Prenons un exemple simple: un guide des meilleurs restaurant d'Ile-de-France. Vous devez concevoir un site internet listant les restaurants par département, par type de cuisine, par évaluation des visiteurs... Imaginons que chaque restaurant est classé par type et par departement. Si je veux désigner Nagoya, un excellent Japonais à Créteil-Soleil (et c'est vrai), l'idée de REST est de déclarer une URI pour représenter cette ressource :

```
/idf/val-de-marne/creteil/japonais/nagoya
```

Donc par extension, si vous souhaitez voir tous les restaurants du val-de-marne, le système doit être capable de répondre si j'utilise cette ressource :

```
/idf/val-de-marne
```

Quand je dis "répondre", je veux dire, ouvrir l'url `http://localhost/idf/val-de-marne` dans un navigateur et recevoir en retour une liste des restaurants, structurée au format XML, HTML ou autre.

### Comment lire une ressource ?

REST se repose sur HTTP et les notions de bases de l'architecture du web. La lecture d'une ressource s'effectue via la méthode GET du protocole HTTP/1.1. Oui, REST est lié à HTTP car c'est une architecture basée sur le web.

### Comment créer une nouvelle ressource ?

Vous devez créer localement une nouvelle instance puis l'envoyer vers le serveur en utilisant la méthode PUT. Par analogie, lorsque vous écrivez un courrier électronique, c'est exactement le même principe. Votre email existe d'abord localement. Ensuite vous le postez au serveur. Imaginons que vous vouliez me faire part d'un autre bon restaurant à Noisy-le-Grand. Voici comment votre système enverra au serveur une nouvelle ressource.

```
POST /idf/seine-saint-denis/nosiy-le-grand/japonais/nigiri
```

### Comment effacer ? Comment mettre à jour une ressource ?

L'effacement d'une ressource se fait via la méthode DELETE du protocole HTTP. La méthode PUT permet en principe de mettre à jour une ressource existante. Cependant certains navigateurs ne gèrent pas correctement ces méthodes. En particulier il n'est pas toujours possible de spécifier la méthode DELETE dans la balise FORM. Pour cette raison on contourne le problème en ajoutant un champ caché dans le formulaire et en utilisant la méthode POST.

### Comparaison avec une architecture RPC classique

Prenons une application pour créer des restaurants et des types de gastronomie. Imaginez les méthodes d'un web service capable d'effectuer ces traitements:

```
getRestaurant()
addRestaurant()
removeRestaurant()
updateRestaurant()
getCuisine()
addCuisine()
removeCuisine()
updateCuisine()
listRestaurants()
listCuisines()
findRestaurant()
findCuisine()
```

Cela fait beaucoup de méthodes pour le même type de service : création, mettre à jour, effacer et lire. De plus, à chaque nouveau service vous devez déployer sur le client un nouvel adaptateur ou mettre à jour la partie cliente... Que se passe-t-il si nous basculons en architecture REST ? Nos ressources seront accessibles avec des URI simples et lisibles comme celles-ci

```

http://www.touilleur-express.fr:8080/restaurants/
http://www.touilleur-express.fr:8080/restaurants/LaFourchetteDor
http://www.touilleur-express.fr:8080/findRestaurantForm
http://www.touilleur-express.fr:8080/cuisines/
http://www.touilleur-express.fr:8080/cuisines/francais

```

Si vous voulez lire les données, celles-ci sont donc désignées par une URI unique. Si vous souhaitez mettre à jour la fiche d'un restaurant par exemple, cela pour s'effectuer soit via une URI de type service, soit directement en spécifiant l'URI du restaurant. Pas mal non ?

## Services Webs RESTfull

C'est une notion vraiment intéressante. Des Web Services basés sur l'architecture REST sont de plus en plus envisageables par rapport à l'utilisation de SOAP pour déployer des services. On distingue 2 grands types de Web services: ceux de type appels distants de procédures (RPC, Remote Procedure Call) et ceux de type échange de document (Document-based). Un Web Service de type RPC est un ensemble de méthodes destinées à effectuer des appels distants. Par exemple nous pourrions voir ce genre de méthode (je caricature un peu, et évidemment qu'il ne faut pas faire ce style de code)

```
public void String getJapaneseRestaurantName(String country,String area, String kindOfFood)
```

Si nous prenons une architecture de type Document-Based, comme il est possible de le faire avec SOAP, à lieu de définir des méthodes RPC vous définissez des méthodes d'échange de ressource. Dans notre cas à lieu d'envoyer une String contenant le nom du restaurant, nous allons envoyer un objet Restaurant. Cette architecture est basée sur l'échange de document, ce qui est très proche de REST. Un service web de type REST doit suivre les principes suivants:

- o **Représentation des ressources:** au lieu de ne fournir qu'un seul point d'accès (un port) et ensuite un grand nombre de méthode dans le web service, l'idée est de donner accès la ressource elle-même. Une ressource est une partie de l'application Web que l'on désire rendre accessible. Cela peut être un avion, un vol, un aéroport, le nom d'une compagnie aérienne, un billet électronique... La ressource sera accessible du client qui en fait ce qu'il veut. Quant à l'état d'une ressource, comme par exempl "billet électronique réservé, billet annulé" c'est un état qui sera stocké dans la ressource elle-même. Faire la distinction entre le modèle de base et son état est l'une des difficultés de REST.
- o **Adressable et accessible sur le réseau.** Une ressource est caractérisée par son type (aéroport) et par son URI (france/paris/orly) afin de l'identifier et de la localiser. De même un web service fournira des services pour des ressources de manière moins visible qu'une architecture de type RPC.
- o **Accessible sur le réseau** Une URI n'est pas suffisante pour qu'un client accède à une ressource. Il faut que l'interface d'accès de la ressource soit identifiable et unique. Si c'est la méthode GET du protocole HTTP qui est utilisée pour lire une ressource, il faut que cette manière soit uniforme dans l'architecture.
- o **Sans état.** Ah c'est un vaste sujet difficile. La puissance d'une architecture REST est de dire que le serveur n'est pas responsable de l'état d'une ressource. Dans une architecture REST, il n'y a aucuns états sur le serveur. C'est au client de stocker l'état de la ressource, et de la lire ou de la retourner au serveur. Ce système permet de bénéficier d'une architecture en cluster très facilement. Votre application va créer une ressource de type "Restaurant". Votre client gère son état ("en attente de revue", "testé", "resto fermé") et il se contente d'envoyer ou de lire la ressource avec le serveur. Dans une architecture de type RPC, vous auriez des méthodes (des verbes) pour modifier la ressource qui est **sur** le serveur. Dans une architecture REST, vous utilisez des ressources (des noms) pour modifier les objets du côté du client. Le client est responsable de l'état de la ressource. Le serveur donne accès à une version de la ressource.

## Cache, cluster et grids sont sur un radeau...

A partir du moment où l'on parle de ressource, tout ce qui est mise en cache du côté du client prend un vrai sens. En effet vous pouvez utiliser les spécifications du protocole HTTP/1.1 pour réduire les échanges entre un client et un serveur. Il suffit lorsqu'une ressource est retournée d'ajouter une date d'expiration ou un ETag HTTP/1.1. Cela permet alors d'utiliser les principes du web pour une application enfin intelligente.

## La spécification JSR-311

Cette spécification vise à proposer une nouvelle librairie légère pour Java afin de permettre d'utiliser une architecture REST. Pour cela, les idées des personnes qui rédigent cette spécification est d'utiliser des annotations pour marquer une class simple et la rendre donc accessible via REST. Au niveau de la couche de protocole, REST est lié à HTTP, même si des extensions comme Webdav seront supportées. Au niveau du contenu échangé entre le navigateur et le serveur, la bonne idée est de n pas nous forcer à utiliser soit du XML, soit du JSON, soit autre chose. Le contenu négocié sera donc indépendant de ce que l'API propose. C'est une bonne chose. Enfin l'objectif de l'équipe est que cette JSR fasse partie de Java EE6. J'ai lu la spec, mais j'avoue que je n'ai pas tout repris ici. Si vous voulez des détails, le mieux est d'aller la lire en ligne sur le site de la JSR.

## Détails sur l'implémentation (une histoire de resto...)

JAX-RS fonctionnera en mode client seul ou en tant que Servlet embarquée dans un conteneur léger. Concernant les annotations, une ressource sera une classe avec au moins une

annotation `@Path` présente sur l'une de ses méthodes ou sur la classe. Par exemple une ressource qui représente des Restaurants :

```
@Path("restaurants")
public class Restaurants {
    ...
}
```

Après compilation et déploiement dans un conteneur de Servlets comme Tomcat, votre liste de restaurant sera donc accessible avec une url de ce type: `http://localhost:8080/rest/restaurants` Cependant ce n'est pas suffisant. Ensuite les annotations suivantes permettront de déclarer les méthodes de type REST: `@GET`, `@POST`, `@PUT`, `@DELETE` et `@HEAD`. Si je souhaite une uri `/restaurants/chinois` par exemple il faut que j'ajoute une méthode et qu'elle soit marquée avec un `Path` comme sur cet exemple :

```
@Path("restaurants")
public class Restaurants {
    @GET
    @Path("chinois")
    public List getListOfChineseRestaurants()
```

Concernant l'injection de paramètres à une méthode, si par exemple vous avez déclaré une méthode `findRestaurant` et que vous souhaitez injecter des paramètres, JAX-RS vous permettra d'annoter la méthode afin de recevoir par injection, les paramètres décryptés de l'URI... ok je pense que là personne n'a compris... Soit l'uri suivante: `/restaurants/laTourDargent` Imaginons que votre Pojo dispose d'une méthode générique pour lister les restaurants. Comment injecter alors le critere "laTourDargent"?

```
@Path("restaurants")
public class Restaurants {
    @Path("{name}")
    public Restaurant getRestaurantByName(
        @UriParam("name") String restaurantName) {
        ....
    }
}
```

## Gestion des formats

Selon que vous souhaitez envoyer du xml, du html ou autre, vous pouvez aussi annoter chaque méthode ou chaque class avec une annotation de contenu. Si je souhaite afficher une photo du restaurant, je peux déclarer une méthode `getPhotoForRestaurant`.

```
@Path("restaurants")
@ProducesMime("text/html")
public class Restaurants {
    @Path("{name}")
    public Restaurant getRestaurantByName(
        @UriParam("name") String restaurantName) {
        ....
    }
    @Path("{name}/photo")
    @ProducesMime("text/jpg")
    public String getPhotoForRestaurant(
        @UriParam("name") String restaurantName) {
        ....
    }
}
```

Attention cependant, ici je retournerai un buffer binaire emballé dans une `String`, ce n'est pas ce qu'il faut faire. Pour gérer les différents types de média, la JSR-311 introduit la notion de `Provider`. Un `provider` peut recevoir ou envoyer des ressources selon différentes formats (xml, text, binaires...). Le `Provider` se charge aussi de l'encoding lorsque c'est nécessaire.

## Les contextes

Il est possible d'injecter les paramètres HTTP ou de lire des valeurs des entêtes HTTP en utilisant une annotation `@Context`. Cela permettra par exemple de retourner du XML si le client est un RSS Reader, ou de retourner du HTML pour un client web. Il sera aussi possible de récupérer les paramètres HTTP afin d'effectuer un traitement.

```
@Path("restaurants")
@ProducesMime("text/html")
public class Restaurants {
    public List findRestaurants(@Context UriInfo info) {
        // info contains bookmarked uri with search
        // criteria for my touilleur restaurant...
        // Call the restaurant service
    }
}
```

Les contextes qui peuvent être injectés : - les headers HTTP - la requête courante - `ServletConfig`, `ServletContext`, `HttpServletRequest`, `HttpServletResponse` En principe nous pourrions aussi injecter des références vers des EJB3 dans un serveur d'application. J'imagine qu'il est prévu de faire de même pour les datasources... Cela dit là on marche sur les plates-bandes de la JSR-299, Web Beans, largement inspiré du framework JBoss Seam. Je m'arrête là, j'espère avoir donné l'envie à quelques uns d'aller tester ou de s'intéresser à la JSR-311 pour leurs applications.

## Je veux tester !

Il vous faut un serveur Tomcat, un peu d'huile de coude et enfin l'implémentation de référence de la JSR-311, à savoir Jersey. JBoss Seam travaille actuellement pour offrir un support compatible à la JSR-311. Pour cela donc, on pourra utiliser les annotations dont j'ai parlé directement dans Seam. <http://www.seamframework.org/Documentation/RESTfulApplicationsWithSeam> A noter qu'en 2007, JBoss Seam et la JSR-311 faisaient partis tous les deux des 10 sujets recommandés dans l'article [Top 10 Destinations for Enterprise Developers at the 2007 JavaOne Conference](#)

### Ressources:

Le site de la JSR-311 <https://jsr311.dev.java.net/> Jersey, reference implementation of JSR-311 <https://jersey.dev.java.net/> The JBoss Seam framework and REST <http://www.seamframework.org/Documentation/RESTfulApplicationsWithSeam>

## Retour de vacances

Billet plus perso que très java aujourd'hui : après un break de 10 jours au soleil, j'ai repris les commandes du toulleur. Pas de réunion PJUG ce soir pour moi, emploi du temps trop chargé. Ce matin dans ma boîte aux lettres 2 livres livrés par Amazon.com : "Agile Project Management with Scrum" et "The Entrepreneur and Scrum" par Ken Schwaber. Ces 2 petits livres de 150 pages chacun sont les 2 piliers qu'il me manque pour compléter ma formation Scrum. Pendant les vacances l'équipe a déconnecté, je vois que le tableau n'a pas bougé. On ne peut pas leur en vouloir, avec la pression d'un client il n'est pas facile de gérer le projet... Et au passage je me rends compte que sans Scrum Master, le projet part dans le décor très rapidement... Va falloir batailler cette semaine pour remettre les autres équipes au pas, qui se sont un peu crues chez Disney cette semaine : et que je te change l'ordre d'un req, et que je décide de décaler une load, et que je rajoute un ou deux bugs par ci-par là... Il faut voir que le rôle de Scrum Master, c'est aussi d'aller mettre un ou deux bourre-pifs aux voisins qui font n'importe quoi. Je caricature, mais franchement, parfois entre essuyer la purée que ma petite fille de 9 mois a craché partout, et envoyer un email pour demander à un client d'entrer un bug, on est pas si loin... Sinon cette semaine : Ze Presentation. Je dois faire une introduction à Scrum à quelques chefs de projets intéressés. Comment présenter sans trop évangéliser ? comment être incollable avec peu de recul ? J'ai bien préparé mon affaire avant les vacances car j'applique Scrum aussi à mon boulot perso, et cela marche bien. Beaucoup de travail, des changements à venir ici et ailleurs... oui c'est une semaine spéciale.

## JBoss RESTeasy

JBoss et plus particulièrement Bill Burke développe une implémentation open-source de la spécification JSR-311 ([voir ancien article ici](#)) Je me suis permis de traduire [cette page](#) en français pour vous donner un petit aperçu de l'api. **JAX-RS Overview** JSR-311, JAX-RS est simplement un moyen de mapper une requête et une réponse HTTP à un appel de méthode dans une class Java. Vous annotez la class ou l'interface avec une annotation qui branche cette classe sur une URI, puis vous mappez les méthodes de cette classes à des requêtes HTTP entrantes, comme sur cet exemple:

```
@Path("/library")
public class Library {

    @GET
    @Path("/book")
    public String getBooks() {...}

    @GET
    @Path("/book/{isbn}")
    public String getBook(@PathParam("isbn") String id) {
        // search my database and get a string representation and return it
    }

    @PUT
    @Path("/book/{isbn}")
    public void addBook(@PathParam("isbn") String id, @QueryParam("name") String name) {...}

    @DELETE
    @Path("/book/{id}")
    public void removeBook(@PathParam("id") String id {...}

}
```

Dans cet exemple, nous avons une ressource **Library** qui est accessible à travers le web. L'annotation **@Path** spécifie un chemin relatif pour localiser la ressource et invoquer une méthode. **@GET**, **@PUT**, et **@DELETE** correspondent aux méthodes standards du protocole HTTP. **@PathParam** récupère des arguments passés dans la ligne d'URI. **@QueryParam** récupère des données de la requête standard. En conséquence pour récupérer la liste des livres nous pourrons spécifier l'URL suivante: **GET** <http://localhost:8080/library/book> La méthode **Library.getBooks()** sera appelée pour lister les livres de la bibliothèque (Library).

La requête **GET** <http://host.com/library/book/332-444-111> affichera le livre dont le numéro ISBN est "332-444-111" en invoquant la méthode **Library.getBook()**

L'appel suivant: **PUT** <http://host.com/library/book/332-444-111> aura pour effet d'ajouter un nouveau livre dans la bibliothèque avec pour ISBN le numéro "332-444-111". Pour cela l'appel de type **PUT** HTTP sera mapé vers la méthode **Library.addBook()**.

L'appel DELETE <http://host.com/library/book/332-444-111> effacera le livre spécifié

Lorsqu'un client demande une ressource, le comportement par défaut est d'allouer un composant pour traiter la requête puis de l'effacer. Ceci peut aussi s'apparenter à une instance par requête. L'ambition de la spécification JAX-RS est aussi de fournir un mapping pour des EJB. RESTeasy de JBoss implémente donc ce modèle de comportement mais fourni aussi une intégration avec un objet enregistré via JNDI ou une intégration avec Spring. Cet exemple ne montre qu'une partie des possibilités de JAX-RS. Il existe d'autres annotations qui permettent d'accéder facilement aux différentes parties d'une requête HTTP. Voir le [wiki](#) de RESTeasy pour d'autres exemples. (Traduction de l'anglais par Nicolas Martignole, article original <http://wiki.jboss.org/wiki/RESTeasyOverview>)

## Présentation d'Esper, moteur de gestion de flux d'événements en Java

### Contexte

Je cherchais une solution pour traiter rapidement un très gros flux d'événements afin d'implémenter un moteur d'alerte. La première idée qui vient en tête serait d'enregistrer dans une base de données mes événements puis ensuite d'écrire un certain nombre de requêtes afin de détecter lorsqu'une alerte sur un service est levée. Le problème est que cela représente un énorme volume d'information, et que donc le chemin « base-de-données-application » n'est pas le bon.

### Introduction à Esper

Esper (<http://www.espertech.com/products/esper.php>) est un moteur open-source de gestion de flux d'événement (ESP en anglais, ce qui signifie Event Stream Processing) ainsi qu'un moteur de corrélation d'événement (si serveur A tombe et serveur B tombe alors le service « paniers » ne marchera plus par exemple...). Un support commercial est disponible ainsi qu'une version avancée lorsque des besoins de haute-disponibilités s'ajoutent à votre projet. Le tout est fourni par la société EsperTech basée en Californie.

L'idée d'Esper est de détecter des événements en temps réel et ensuite de déclencher des actions lorsqu'un événement, ou plusieurs événements se produisent. Le traitement d'alerte en temps réel fait parti du cœur de logiciel financier complexe, mais aussi d'outils très puissant de supervision de réseau et applicative.

ESP (Events stream processing) et CEP (consolidated events processing) sont 2 domaines qui viennent de la recherche. Esper propose en fait un langage appelé Event Processing Language (EPL) pour décrire un filtre d'alerte, pour agréger des alertes et travailler sur des fenêtres temporelles. Je vais reprendre un des exemples de la doc d'Esper pour vous expliquer cela. L'idée n'est pas de stocker des données, et ensuite de les analyser, selon des requêtes à écrire sur la base, ce qui sera lent. L'idée est de stocker des filtres, des alertes, puis d'appliquer ces filtres sur les données surveillées en temps réel. C'est un moteur de règle appliqué aux données. Esper est donc un système qui vous permet d'écrire avec la syntaxe EPL des requêtes et de déclencher ensuite en Java des événements dans votre code lorsque les filtres sont activés.

### Exemple avec la surveillance de la charge CPU d'un serveur

Prenons un objet dans une application de surveillance de machine qui serait capable de demander à un démon distant la charge du CPU. Nous allons dire que nous voulons une alerte si pendant 10 minutes, le CPU reste au-delà de 90% de charge machine. Nous ne voulons pas d'alerte si pendant cette fenêtre de 10 minutes, le CPU passe une seule fois au dessus de 90%, mais si cela se reproduit disons 10 fois au moins, afin de détecter un problème de charge.

Cependant dans un premier temps, il faut mettre en place une alerte si le serveur interrogé ne répond pas du tout pendant disons 10 minutes. Pour communiquer entre mon serveur de surveillance et le démon distant, j'envoie un message simple, et le démon distant me répond avec un message de réponse. Si je n'ai pas de corrélation avec un requestId pendant 10 minutes, alors je veux lever une alerte. Le fait d'attendre 10 minutes évitera de lever une alerte pour un problème réseau temporaire par exemple.

La requête est exprimée avec cette classe Java simple que j'envoie au démon :

```
public class CPULoadRequest {
    public int requestId;
}
```

La réponse retournée par le démon contient un timestamp ainsi que la valeur courante de la charge CPU au moment où nous avons enregistré celle-ci :

```
public class CPULoadResponse {
    public int responseId;
    public long timestamp ;
    public int cpuLoad ;
}
```

J'ai besoin de ces objets en Java car je vais m'en servir avec Esper pour exprimer une requête, qui

correspond à ce que je veux surveiller.

En regardant les événements CPUloadRequest et CPUloadResponse, Esper sera capable de nous alerter lorsque mon seuil est déclenché. Si par exemple je veux détecter que je n'ai pas reçu de CPUloadResponse durant les 10 dernières minutes, je peux exprimer cela avec cette requête EPL :

```
every r=CPUloadRequest -> (timer:interval( 10 min) and not CPUloadResponse(responseId = r.requestId))
```

Le symbole "->" signifie "suivi de". Ici pour chaque CPUloadRequest suivi dans un intervalle de 10 minutes et dont on ne retrouve pas l'id, alors lever une alerte.

Pour être notifié de toutes les quotations boursières de Reuters supérieures à 45 des 60 dernières secondes on peut ainsi écrire :

```
every StockTickEvent(symbol="RTR.L", price>45) where timer:within(60 seconds)
```

Voyons maintenant comment exprimer l'alerte suivante : je souhaite qu'une alerte soit générée si le CPU dépasse au moins 10 fois 90% de la charge machine pendant une fenêtre de 10 minutes.

```
select count(*)
  from CPUloadResponse(cpuLoad > 90).win:time(10 minutes)
 having count(*) >= 10
```

Le mot clé "win:time" déclare une fenêtre de temps glissante de 10 minutes. Ici on cherche à compter combien de CPUloadResponse supérieur à 90 sur les dix dernières minutes, et se répétant au moins 10 fois existent... Esper ne consulte pas de base de données pour retourner la valeur. Il se débrouille pour mettre en place de quoi enregistrer sur les 10 dernières minutes seulement, un petit bout de filtre qui correspond à ce que je demande. Vous voyez la puissance ? Nous ne stockons rien.

## Mise en place avec du code

Pour représenter ce statement en Java voici à quoi va ressembler votre code :

```
EPStatement statement = service.createEQL(
    "select count(*) as cnt " +
    "from CPUloadResponse (cpuLoad > 90).win:time(10 minutes) " +
    "having count(*) >= 10";
```

Ensuite pour être notifié lorsque mon seuil est déclenché, je vais implémenter l'interface UpdateListener d'Esper et mon code sera appelé par Esper lorsque la condition est réalisée.

```
public class CPUListener implements UpdateListener {
    public void update(EventBean[] result, EventBean[] prior) {
        System.out.println("Number of CPU over 90% for the last 10 min is " + result[0].get("cnt") );
    }
}
```

Pour enregistrer ma classe CPUListener dans Esper :

```
MyStmtObserver observer = new CPUListener ();
statement.setObserver(observer);
```

Et pour terminer il me reste à implémenter un peu de code pour créer des objets CPUloadRequest et les envoyer à mon démon distant avec par exemple :

```
CPULoadRequest event = new CPULoadRequest() ;
myService.sendToDaemin(event, daemonId) ; // Send this event to a specific daemon
```

## Domaine d'application

Esper pourrait être utilisé dans la finance pour implémenter un logiciel d'alerte et de notification. Il pourrait servir aussi dans un site marchand pour surveiller les données applicatives comme le nombre d'achat sur un site marchand, l'encaissement par un site bancaire, l'envoi d'un bon de commande à un transporteur par exemple... Dans une architecture SOA Esper apporterait un moyen de modéliser les règles de surveillance du bon fonctionnement de différents services...

## Conclusion:

Au lieu d'enregistrer des événements temporels et ensuite de les analyser, Esper permet à une application Java de décrire et d'enregistrer des requêtes d'alerte. Par sa structure et ce mode de fonctionnement, Esper est donc beaucoup plus adapté à gérer un grand nombre d'événement. Par ailleurs le mode d'exécution d'Esper est continu, dans le sens qu'on peut réellement parler de fenêtre temporelle, et pas de surveillance par pooling d'une base de données.

Spéciale dédicace à Gik pour ce billet ;-)

## Ressources supplémentaires :

Excellent article sur OnJava <http://www.onjava.com/pub/a/onjava/2007/03/07/esper-event-stream-processing-and-correlation.html>

Article dont je me suis largement inspiré pour écrire ce billet : <http://www.theserverside.com/tt/articles/article.tss?l=ComplexEventProcessing>. Auteurs : Thomas

Bernhardt et Alexandre Vasseur.

## Le jeu Bioshock tourne sur mon iMac 20" sans problèmes

Un peu d'informations Mac et Apple pour tenter de vous convertir à autre chose que Windows et Linux. Oui cela faisait un peu longtemps que je n'avais pas tenté de vous évangéliser mes frères. Tout d'abord à la maison, j'ai installé Microsoft Vista sur mon iMac 20" puis dans la foulée, soyons fou, le jeu BioShock. Pour ceux qui ne sont pas trop jeux, disons que BioShock est un excellent jeu de tir se déroulant dans une cité engloutie construite en 1960. Après avoir créé une partition de 30Go avec BootCamp, j'ai installé Vista puis ensuite [BioShock](#) (20 EUR à la Fnac mardi soir). Je rebooto donc complètement sous Windows, je n'utilise pas la virtualisation. Mon imac est équipée d'une carte ATI Radeon 2600 HD pro avec 256Mo de mémoire, compatible DirectX 10. Ce n'est pas la toute dernière carte mais franchement, le jeu tourne sans problèmes en niveau de qualité "supérieure" d'après les réglages de BioShock. J'ai la version avec CPU Intel core 2 Duo à 2.6Ghz et 3Go de mémoire vive, le tout pour 1300 EUR mon bon monsieur. Oui un mac ce n'est pas si cher contrairement à ce que vous pouvez penser. Aaah mac... vivement que je m'achète un petit macbook pro.

## Comment trouver la charge CPU de la machine en Java ?

Voici la réponse à une question posée par un développeur : comment puis-je trouver la charge moyenne du ou des CPU d'un serveur en Java ? Et le tout, sans écrire de JNI ou d'appel natif ? Java 5 a introduit avec JMX des MBeans simples pour la gestion de mémoire, l'analyse des Threads java, l'accès au système d'exploitation ou encore les paramètres du processus Java en cours. Dans le package **java.lang.management** nous avons les beans suivants :

- o java.lang.management.ClassLoadingMXBean
- o java.lang.management.CompilationMXBean
- o java.lang.management.MemoryManagerMXBean
- o java.lang.management.GarbageCollectorMXBean
- o java.lang.management.MemoryMXBean
- o java.lang.management.MemoryPoolMXBean
- o java.lang.management.OperatingSystemMXBean
- o java.lang.management.RuntimeMXBean
- o java.lang.management.ThreadMXBean

Pour notre problème de CPU et de charge, le bean OperatingSystemMXBean permettra de lire quelques informations sur le système d'exploitation. Ce bean est disponible à partir de Java 5. Mais il vous faudra Java 6 obligatoirement pour trouver la charge moyenne de la machine. Nous allons utiliser tout simplement la méthode `getSystemLoadAverage()` qui retourne la charge CPU moyenne des dernières 60 secondes. Le code sera le suivant :

```
import java.lang.management.ManagementFactory;
import java.lang.management.OperatingSystemMXBean;

/**
 * Sample that uses the JMX OperatingSystemMXBean bean to collect various information
 * about the operating system, the number of CPU, and the load average from Java.
 * This code works with Java 6 cause getSystemLoadAverage is available only with Java 6.
 *
 * @author nicolas.martignole
 * @version created on 23-May-2008
 * @since 1.6
 */
public class TestLoadAverage {
    public static void main(String[] args) {
        OperatingSystemMXBean bean = ManagementFactory.getOperatingSystemMXBean();
        if (bean == null)
            throw new NullPointerException("Unable to collect operating system metrics, jmx bean is null");
        System.out.println("System architecture : " + bean.getArch());
        System.out.println("Number of processor(s) : " + bean.getAvailableProcessors());
        System.out.println("Operating system name : " + bean.getName());
        System.out.println("Operating system version : " + bean.getVersion());

        // Returns the system load average for the last minute.
        // Java 6 only
        System.out.println("System load average for the last minute : "+bean.getSystemLoadAverage());
    }
}
```

Lorsque j'exécute ce code sur mon portable (un vieux T42) avec un processeur mono-coeur, voici l'affichage retourné : System architecture : x86 Number of processor(s) : 1 Operating system name : Windows XP Operating system version : 5.1 System load average for the last minute : -1 La mauvaise nouvelle est que sur Windows, la méthode **getSystemLoadAverage** n'est pas implémentée. Elle retournera -1. Sur une machine double-coeur, le nombre de processeur retourné est bien 2. J'ai fait le test avec un T60 d'IBM équipé d'un Intel Core 2 Duo.

Pour notre problème de CPU, nous travaillons sur Solaris. Voici le résultat d'un test sur une Sun. II

```
s'agit d'une V440 quadri processeurs avec Solaris 10. On peut constater que cette fois-ci, Java 6
retourne la charge moyenne du CPU des 60 dernières secondes. [nicmarti@ptxskarma
nicmarti]% uname -a SunOS ptxskarma 5.10 Generic_118833-36 sun4u sparc SUNW,Sun-
Fire-V440 [nicmarti@ptxskarma nicmarti]% java -version java version "1.6.0_01"
Java(TM) SE Runtime Environment (build 1.6.0_01-b06) Java HotSpot(TM) Server VM
(build 1.6.0_01-b06, mixed mode) [nicmarti@ptxskarma nicmarti]% java
TestLoadAverage System architecture : sparc Number of processor(s) : 4 Operating
system name : SunOS Operating system version : 5.10 System load average for the
last minute : 0.328125 [nicmarti@ptxskarma nicmarti]%
```

## Statistiques sur les Threads

De la même manière vous pouvez aussi écrire rapidement du code en java pour surveiller les Threads Java de votre application. Il est ainsi possible de compter le nombre de threads java démarrées dans la JVM, de voir si une thread est active ou en attente par exemple. Voici un autre exemple de code que vous pouvez très facilement adapter en lisant la javadoc de [ThreadMXBean](#)

```
import java.lang.management.ManagementFactory;
import java.lang.management.ThreadMXBean;
import java.lang.management.ThreadInfo;

/**
 * This sample collects java Thread details using Java 5 Managment capabilities.
 * Java 5 or 6 is required.
 *
 * @author nicolas.martignole
 * @version created on 23-May-2008
 */
public class TestThreads {
    public static void main(String[] args) {
        ThreadMXBean bean = ManagementFactory.getThreadMXBean();
        if (!bean.isThreadCpuTimeSupported()) {
            System.out.println("Sorrrt, the Java virtual machine implementation does not support CPU time measurement for any thread.");
            System.exit(-1);
        }
        if (!bean.isCurrentThreadCpuTimeSupported()) {
            System.out.println("Sorry, JVM does not supports CPU time measurement for the current thread");
            System.exit(-1);
        }
        if (!bean.isThreadCpuTimeEnabled()) {
            System.out.println("Sorry, Thread CPU measurement is not enabled");
            System.exit(-1);
        }
        showThreadIds(bean);
    }

    /**
     * This method shows some threads informations.
     *
     * @param bean is the ThreadMXBean.
     */
    private static void showThreadIds(ThreadMXBean bean) {

        for (long id : bean.getAllThreadIds()) {
            System.out.println("ID " + id);
            showThreadInfo(bean, id);
        }
    }

    /**
     * Show interesting details about a thread.
     *
     * @param bean
     * @param id is the thread id.
     */
    private static void showThreadInfo(ThreadMXBean bean, long id) {
        ThreadInfo info = bean.getThreadInfo(id);
        System.out.println("java thread name: " + info.getThreadName());
        System.out.println("java thread state: " + info.getThreadState());
        System.out.println("Thread CPU Time : "+ bean.getThreadCpuTime(id));
        System.out.println("Thread peak time: "+bean.getPeakThreadCount());
    }
}
```

## Information sur la JVM comme le Classpath

Le bean [RuntimeMXBean](#) permet de lire un certain nombre d'informations intéressantes sur le processus Java en cours. Vous pouvez ainsi retrouver le classpath, le bootclasspath ce qui est utile lorsque vous jouez avec Xalan ou Xerces. Vous pouvez aussi lire depuis combien de temps le processus Java a démarré (uptime) ainsi que l'implémentation de la JVM en cours (SUN ? Bea ? IBM ?). Vous avez accès aux propriétés systèmes sous la forme d'une Map et donc, au PATH sous unix par exemple. Vous avez aussi accès aux paramètres de la ligne de commande du processus java, afin de voir par ailleurs ce qui a été passé en argument au processus Java.

## Conclusion

Je vous conseille de prendre quelques minutes pour voir la liste des différents MBean d'administration et de surveillance qui sont disponibles en Java. Ceux concernant la gestion de la mémoire par exemple sont aussi très puissants.

Voir la liste complète sur cette page : [Package java.lang.management Java SE 6](#). En espérant que ce post aidera quelques uns d'entre nous.

## Atelier clients riches Valtech 19 juin 2008

J'ai eu la confirmation ce matin que je participerai le 19 juin prochain à l'Atelier Client Riches organisé par Valtech. Si vous êtes intéressés, voici en quelques mots les sujets qui seront traités durant cette journée. Dans un premier temps, présentation de Flex par Yann Chevalier, puis de Microsoft Silverlight par Angelo Pacifico et enfin Google Web Toolkit par Sami Jaber. Comme d'habitude je posterai ici un compte-rendu de cette journée afin de vous faire partager mon expérience.

## Macro-planning ou l'art de la voyance

Billet d'humeur cet après-midi (ou comment se défouler gratuitement...)

Chaque projet passe un jour par le petit moment magique où vous devez vous engager sur une date et sur un contenu.

Le client :

"- Combien de temps pour ce superbe site internet de réservation de billets pour la Lune ? "

Et vous, après des semaines de travail, vous répondez fier comme un pape:

- "457 jours / hommes..."

Votre client est content, ou votre chef est content. Et ils vous fichent la paix après avoir cependant lu votre document/regardé votre Microsoft Projet Plan/discuté avec vous au café...

En ce moment à la maison je fais venir des entreprises afin de chiffrer les réparations nécessaires suite à un dégât des eaux. Très marrant de demander un devis, sachant que les 4 entreprises contactées m'ont rendu des devis complètement différents à chaque fois... Sachez qu'entre chacune des visites je n'ai pas explosé un mur ni creusé une piscine... Cela me fait penser que du coup, lorsque la veille j'ai vendu 457 jours/hommes mon paquet de lessive, j'aurai annoncé 450 ou 480, ça serait passé non ?

De là à dire que l'estimation des travaux dans le bâtiment et l'estimation des projets dans mon métier se font au pifomètre... il n'y a qu'un pas que je franchis sans problèmes.

Non mais franchement, n'est-il pas urgent d'arrêter de vendre et de planifier des projets en essayant de donner des estimations en jour/homme ? Même les meilleures méthodes d'estimation sont pour moi une forme d'arnaque intellectuelle. Disons que je vends mon bidule pour 457 jours et donc je récupère 3 développeurs. C'est tout simplement faire de la voyance et raconter à son supérieur que le projet sera réalisé dans cette enveloppe de jours, ni plus, ni moins. Arrêtons cinq minutes... Il y a des numéros surtaxés qui pour quelques euros vous prédisent votre avenir. Cela coûterait moins cher non ?

Il est temps que les chefs de produits se prennent en main comme de grands enfants. Lorsque votre projet passe à Scrum, les personnes à l'extérieur ont le sentiment de ne plus suivre ce que réalise votre équipe. On me parle de "perte de visibilité" ici... Je réponds alors qu'en effet j'ai arrêté la voyance, car depuis presque 3 ans c'était pas un franc succès. Depuis que nous pouvons annoncer 1 ou 2 sprints Scrum à l'avance le menu des livraisons, les clients directs sont heureux. Et finalement c'est ce qui compte.

Par contre, la masse des macro-planneurs perd complètement les pédales car nous ne faisons plus de planning sur les 3 ou 6 mois à venir. Certes il y a bien un ou deux objectifs qui donnent une tendance. Mais là où il fallait 457 jours pour réaliser une tâche, en passant à Scrum on est maintenant capable de livrer par itération une partie de cette demande dès la fin d'un sprint de deux semaines.

Du coup ces 457 jours ne veulent plus rien dire, et il est facile de prévoir que le projet sera terminé avant, qu'il répondra mieux à la demande du client et qu'en plus, il prendra en compte les demandes de changements qui peuvent survenir durant ces pseudos 457 jours/hommes... dingue non ?

Et tout ceci grâce à Scrum...

(soupir)

## Wizzgo enregistre la télé

Un nouveau service qui devrait faire du bruit je pense dans les semaines qui viennent : [Wizzgo](#). Le principe : le site est un magnétoscope numérique qui vous permet d'enregistrer les programmes de la TNT (17 chaînes) afin de les visualiser ensuite plus tard soit sur votre PC, sur votre iPod via iTunes ou même via l'iPhone. Le service gratuit vous permet donc de revoir par exemple les 3 épisodes de Prison Break d'hier soir dès le lendemain matin sur votre ipod, dans le RER.

Se pose cependant la question de la rémunération des droits d'auteur, comme pour le foot par exemple. En effet, TF1 ayant acquis par exemple 20 matches de la L1 pour 2008, pour un montant assez important, et Orange ayant acquis les droits de diffusion sur Internet, comment Wizzgo compte-t-elle cohabiter ? Comme l'explique Jérôme Taillé-Rousseau, Wizzgo se repose simplement sur le droit à la copie privée qui permet à chacun d'effectuer une copie pour usage

strictement privé. Les fichiers au format Quicktime sont ainsi marqués afin d'identifier les contrevenants en cas de diffusion sur les réseaux Peer-to-peer. Wizzgo n'est pas un diffuseur mais un service de stockage personnalisé. D'où la parfaite légalité du système.

Le service vous permet d'enregistrer 15 heures par mois, et fonctionne sur le serveur. En partant du bureau donc vous pouvez programmer l'enregistrement d'une émission entre 19h00 et 20h00, puis la visualiser à la maison sur votre PC ou sur votre Mac. L'espace de stockage est limité à 4Go d'après ce que j'ai vu pour l'instant.

Testez-le pour le fun en téléchargeant le client pour Mac ou PC sur <http://www.wizzgo.com>. Petits bémols : pas de client linux et seul le format Quicktime est géré.

## iPhone2 dans quelques jours ?

Selon des rumeurs l'iphone2 sera disponible à partir du 15 juillet aux Etats-Unis. C'est une rumeur qui devrait être confirmée dès le 9 juin prochain par Steve Jobs lors de [la grande conférence des développeurs Apple](#) qui se déroulera du 9 au 13 juin prochain à San Francisco. Sachant que Steve Jobs ne sort que 2 fois par an, en février et en juin, il est presque acquis que l'annonce sera faite lors de cette conférence.

C'est tout d'abord l'opérateur téléphonique AT&T qui a demandé à ses salariés de ne pas prendre de congés entre le 15 juillet et le 15 août. Ensuite une publicité tournée il y a quelques jours dans l'AppleStore de la 5th avenue à New-York. Le site [MacRumors](#) explique ainsi que :

L'AppleStore n'a été fermé que 2 fois depuis son ouverture : pour la sortie de Léopard et pour la sortie de l'iPhone 1...

En effet cet AppleStore est ouvert 24h/24h en principe.

Pour moi ce qui m'intéresse en ce moment c'est le Mac Book Pro. Voilà je l'avoue, je passe mon temps sur l'appstore en ligne. Je clique sur le panier, je rajoute de la mémoire et un peu de disque dur, je sélectionne telle ou telle option et je clique sur valider... et ensuite au moment où en principe je dois l'acheter je m'arrête... 4 fois que je fais cela, je vais bien finir par craquer non ?

En attendant Intel annonce que sa plateforme Montevina destinée aux ordinateurs portables aura [un petit peu de retard](#). Sachant qu'Apple intègre très rapidement les dernières innovations d'Intel, il est peut-être plus sage d'attendre le 1er septembre pour acheter un Mac Book Pro... Le 1er septembre... le 1er septembre...

Allez je retourne sur [le site d'Apple](#) pour rêver encore une fois... 15" CPU 2.6Ghz 4Go de mémoire, disque 200GB à 7200 tours/mn (important pour mes compilations), écran mat, iWorks pour faire de belles présentations avec KeyNote, un clavier en Anglais international pour travailler et enfin j'hésite à prendre l'AppleCare pro car cela rajoute 439 EUR... Total 2624 EUR TTC et donc 3000 EUR avec le support 3 ans sur site. Ah ah ah...

Si seulement je pouvais retourner à New-York comme en octobre dernier pour m'en prendre un... (soupir)... On va attendre le 1er septembre... le 1er septembre... le 1er septembre...

## Rencontre avec différentes sociétés

Voici une expérience que j'aimerais bien tenter en juin et juillet. C'est une idée qui est arrivée suite aux différentes rencontres effectuées au PJUG. Dites-moi ce que vous en pensez... **Vous** : vous êtes un éditeur de logiciel autour de Java avec un produit que vous souhaitez faire connaître. Vous êtes aussi à la recherche de nouveaux développeurs pour renforcer vos équipes. Le souci : comment se faire connaître et identifier les meilleurs candidats ? **Moi** : passionné par Java, à la recherche de nouveaux produits, aimant faire de nouvelles connaissances, j'aimerais aussi faire parler de mon blog, et évidemment me créer un réseau de partenaires. **Le blog** : c'est environ 400 visiteurs différents par mois. 2000 visites en avril, 4223 pages vues toujours en avril. Les lecteurs sont francophones, composés de développeurs, d'architectes ou de consultants. **Le concept** : Rencontrons-nous ! Dans vos locaux autour d'un bon café, l'idée est de présenter votre société, votre produit et vos besoins. Ensuite je publie un post sur le blog. Je vise dans un premier temps les éditeurs de logiciels sur Paris uniquement. Peut-être ensuite les cabinets d'expertises et de consultants ainsi que les sociétés de service. Mais dans un premier temps, juste les éditeurs de la place de Paris. Java, .NET, Scrum, clients riches, méthodes Agiles... J'ai pris contact avec 2 sociétés pour commencer à amorcer le concept. Qu'en pensez-vous ? PS : les commentaires ne sont pas modérés mais vous devez être inscrit, et ce, afin d'éviter le spam. Vous êtes libre de vous inscrire pour poster ensuite vos messages. N'hésitez pas à me contacter par email pour prendre RDV.

## Rencontre avec la société Jaxio et le produit Celerio

J'ai eu le plaisir de rencontrer pour un premier [interview](#) la société [Jaxio](#), éditeur du logiciel **Celerio**. Jaxio a été créé par Nicolas Romanetti et Florent Ramière pour développer et proposer un nouvel outil de génération agile pour Java appelé Celerio.

Voici un résumé de l'interview et ensuite mes impressions à la suite de celui-ci

---

### Quel est le besoin auquel répond Celerio ?

Nicolas Romanetti explique tout d'abord que le constat sur le marché aujourd'hui est que :

- les technologies Java du monde open-source évoluent vite.
- les compétences pointues en Java sont difficiles à trouver.
- les décideurs sont de plus en plus exigeants et exigent un engagement sur la qualité et les performances pour le même prix. Il n'est plus envisageable d'intégrer à posteriori les outils de qualités dans le code au risque de mettre sinon le projet en danger.
- le temps de développement des projets tend à se réduire, les décideurs veulent une solution en ligne rapidement pour un budget limité.
- les développeurs écrivent du code d'architecture technique et ne passent peut-être pas assez de temps sur la partie métier. La recherche d'une solution à un problème sur Hibernate ou Spring lorsque l'on ne maîtrise pas ces technologies peut accaparer un temps précieux
- une fois en production, le tuning est difficile car il demande une expertise sur les technologies que les développeurs n'ont pas tous
- le turn-over sur les projets de 6 mois à 12 mois est un facteur de risque. La connaissance n'est pas assez diffusée.
- la mise en place d'outils comme maven, findbugs, pmg, checkstyle n'est pas vu par la maîtrise d'ouvrage comme un élément à valeur ajoutée.
- le choix parmi les technologies open-source Java n'est pas toujours évident
- il existe un fort besoin d'industrialiser le développement des applications en mettant en place de l'intégration continue, du développement par les tests, de la génération de code
- presque tous les projets d'applications webs en Java se basent sur les mêmes technologies. Le marché a déjà sélectionné les meilleures solutions open-source pour nous.

### Pouvez-vous citer une référence client ayant mis en place Celerio ?

La société EPTICA a développé une nouvelle application Web destinée entre autre à répondre à un appel d'offre d'une grande banque française. Le tout en moins de 6 mois. Cette application a été prototypée puis développée grâce à Celerio. Elle permet aujourd'hui chaque matin aux quelques 18 000 conseillers clientèles de la Grosse Banque de répondre aux courriers électroniques envoyés par les clients via le portail Web de la banque. Grâce à Celerio, EPTICA a répondu avec succès à la demande de la Grosse Banque et le produit a été mis en place très rapidement.

### Comment ont-ils utilisé Celerio ?

Les développeurs d'EPTICA ont utilisé Celerio pour générer le code d'architecture de leur projet. Ils ont ensuite implémenté la partie métier sans perdre de temps à configurer Spring, Hibernate ou Struts par exemple.

Celerio est intégré au cycle de production car il permet de générer une partie de l'application lors de l'intégration continue. Il facilite la conception par itération et il est donc complètement Agile. Scrum serait une excellente méthode projet pour piloter un développement avec Celerio. On évite l'effet Tunnel durant lequel un client n'a aucunes visibilité sur le développement en cours. Celerio facilite le travail des développeurs. Le développement peut donc démarrer rapidement sans attendre la fin de la rédaction des spécifications ou de la définition du schéma de la base par exemple...

### Comment mettre en place Celerio ?

En fait Celerio est livré sous la forme d'un ensemble de plugins maven. Il permet tout d'abord à l'aide d'un bootstrap développé par l'équipe de Jaxio de préparer une arborescence de travail avant de commencer à écrire le code. Cette structure normalisée basée sur Maven sera familière aux développeurs utilisant déjà Maven.

Ensuite le développeur travaille sur le modèle de la base de données : un simple script SQL permet de définir dans un premier temps la structure relationnelle de son application.

Une fois son modèle exposé, il peut alors faire fonctionner Celerio afin d'effectuer les étapes suivantes :

- génération de la couche Hibernate
- création des beans DAO, model, manager représentant le modèle
- création d'un ensemble d'action Struts standards (CRUD)
- génération des tests unitaires avec junit et EasyMockup
- Maven ensuite se charge de la compilation du code, des tests unitaires et d'intégration

- exécution des tests
- génération du site maven de rapport avec un ensemble d'outils de rapports
- génération de la web application avec toutes les pages jsp, le code, les dépendances externes
- mise en place d'un serveur Jetty minimal afin de tester ensuite directement sur son poste le résultat

**NDLR : Celerio s'active en ligne de commande avec maven en spécifiant des profils d'exécution selon ce que l'on désire faire. Le cousin de Celerio sur ce mode est le générateur seam-gen de JBoss Seam bien que celui-ci ne soit pas aussi puissant que Celerio.**

**Si par exemple je surcharge l'action insert() dans la class UserAction de Struts, comment Celerio ensuite peut fonctionner ? Est-ce que le code généré écrase mon code ?**

Justement non, nous avons fait en sorte que la personnalisation métier soit très facile à mettre en place. C'est au générateur de s'adapter à votre code et pas à vous de rentrer dans un canevas trop rigide. Il n'était pas envisageable de mettre partout des tags spéciaux ou des annotations pour rendre magique du code.

Au lieu de cela nous avons cherché longtemps une solution qui permette de surcharger proprement du code tout en laissant la possibilité à Celerio d'ajouter du code. Cette partie fait l'objet d'un dépôt de brevet et je ne peux pas encore en parler.

(Cela dit Nicolas et Florent m'ont fait une démonstration qui m'a vraiment convaincu, en temps que développeur.)

**Est-ce qu'il faut installer un éditeur spécial ? un plugin dans Eclipse ?**

La réponse est non aux deux questions. Le développeur Java doit être libre d'utiliser l'outil qu'il veut. Dès le départ par rapport au code généré (java, xml, proprietes, jsp, tld...) nous avons fait en sorte de respecter une structure simple telle que celle proposée par Maven. Cela nous permet d'éviter de réinventer la roue. D'ailleurs c'est l'une de nos phrases favorites : rester simple et ne pas réinventer ce qui a déjà été fait autre part.

**Quelles sont les techniques de rendu pour la couche Web ?**

Aujourd'hui html simple et extJS. Nous intégrons cependant dans l'application générée des taglibs standards de Struts2. Pour la partie rendu on suit l'approche struts2 : création de nouveau composants en freemarker. Pour certains composants, comme ce paginateur dans un tableau, nous avons ajouté des composants maisons.

Nous disposons aussi d'un rendu basé sur extJS plus orienté client riche. Et nous travaillons sur une intégration de Google Web Toolkit.

**Au niveau couche web ?**

Aujourd'hui Struts2 et Spring pour l'accès au code métier. Nous travaillons sur Spring Web Flow pour apporter une architecture orientée composant, là où Struts est orienté page/action.

**Et donc JSF ? envie, pas envie ?**

Honnêtement notre premier besoin est de générer du code rapide, simple à maintenir et qui permette une montée en charge de l'application. Cependant JSF est une techno à laquelle nous réfléchissons aussi pour d'autres types de projet, plus orienté client riche que purement web. Struts2 reste le moteur le plus simple et le plus puissant pour répondre à la majorité des besoins de nos clients.

**Merci de m'avoir reçu et à bientôt**

A bientôt !

-----  
**Conclusion**

Séduit, clairement séduit. En effet en tant que décideur j'ai là une solution qui permet à mon équipe de générer le squelette de mon projet en suivant les meilleurs pratiques en cours.

Pas facile de comprendre l'intérêt de Celerio tant que l'on ne l'a pas vu en action. Ensuite on ne peut qu'être séduit. Certes le gourou refusera qu'un produit tiers vienne sur son terrain lui montrer comment travailler. Mais moi en tant que payeur sur un projet, je serai même demandeur pour voir une alternative à mon projet web basé sur Celerio. J'imagine même que je pourrai challenger mon équipe d'architecture et obtenir le meilleur d'un point de vue métier. En tant que décideur, les solutions techniques sont pour moi une source de coûts. Et de coûts aussi une fois en production. Là sur ce point, Jaxio marque un point énorme par rapport au marché.

Du côté des recherches de compétences pointues sur Java, aujourd'hui il est très difficile de trouver un expert Spring, Hibernate, Struts2 qui pour un coût modique (son salaire) vous propose le meilleur assemblage technique pour répondre à vos besoins. Florent et Nicolas sont tous les deux de vieux routiers de Java qui ont travaillé à San Francisco avant de rentrer en France. Ils

connaissent parfaitement l'ensemble des technologies qu'ils proposent, ce qui est aussi le résultat d'un gros travail de veille technique, d'une certaine curiosité et de connections privilégiées comme avec Spring-Source. La valeur de Celerio est aussi basée sur la valeur de leurs domaines d'expertises.

J'ai aussi compris que leur logiciel permet de répondre très rapidement à un appel d'offre. En effet, celui-ci permet de générer les quelques 70% du code d'architecture et d'infrastructure d'une application, pour qu'ensuite un développeur fonctionnel s'attaque à la partie métier.

Je disais à Nicolas et Florent que là où aujourd'hui un junior va apprendre Spring en regardant le fameux PetStore, Celerio permet de générer du code en rapport direct avec le projet sur lequel le développeur va travailler. Or à mon avis pour monter en compétence, il sera plus facile de regarder du code en rapport avec son projet qu'un bête projet sur les...petites bêtes !

Imaginons un instant qu'ici chez Thomson-Reuters nous soyons utilisateur de Celerio. En tant que développeur je lance l'outil sur ma base et mes concepts financiers. Celui-ci me génère une application web en 2mn qui fonctionne réellement et que je peux tester. Si ensuite je veux commencer à écrire mes propres écrans je pourrai partir de la structure écrite par Celerio, au lieu de m'inspirer d'un exemple sur le web. Cependant je reste maître de mon code et le générateur n'est pas intrusif. Et je peux continuer à utiliser Celerio pour générer d'autres parties de l'application au fur et à mesure que mon modèle en base évolue. Celerio n'est pas un outil "one shot" mais un vrai outil logiciel qui accompagne le développeur.

Je cherche quelque chose de moins enthousiaste à dire mais je n'ai pas grand chose. Bon je me force pour Nicolas: les diagrammes des slides powerpoints sont pas très beaux (arf...). Non franchement ce que j'ai vu m'a intéressé en tant que développeur et aussi en tant que client potentiel. On n'est pas dans la poudre de perlin pin pin qui fait la pluie et le beau temps, mais bien devant un outil qui crée une grosse valeur ajoutée à un projet pour une mise initiale raisonnable. Je pense que le mieux est encore de se faire accompagner par l'équipe de Jaxio. Je vois qu'en 4 heures nous avons couvert le produit, ce qui montre qu'il est simple à comprendre. Mais derrière le gros effort de simplification se cache des heures et des heures de réflexions, d'essais, de tuning et enfin d'implémentation.

Le code est propre, documenté, léger. Pas de fouillis, classé par package sans dépendances cycliques... tout a été pensé minutieusement. Faire simple c'est à mon avis plus dur que faire...

Quoi d'autres ? On aimerait une version allégée open-source pour tester et faire du buzz autour de Celerio. Peut-être un jour ?

Au niveau du rendu, il y a tout ce que l'on a besoin dans une application Web. Mon œil d'architecte aurait aimé bien sûr plus de composants riches, car je suis habitué à voir non pas des applications webs mais des applications type client léger.

En tout les cas l'équipe est très sympa et dynamique. Ils sont situés à Suresnes mais un déménagement est prévu cet été pour aller dans Paris.

J'ai demandé pour terminer à l'équipe le profil de développeur Java qu'ils recherchent :

- « Quelqu'un de dynamique, motivé, qui aime coder, qui aime la technique et bien entendu qui soit un vrai fan de Java. Aussi quelqu'un capable d'aller en clientèle pour assurer des prestations techniques chez nos clients. »

Si vous souhaitez postuler, vous pouvez contacter Jaxio via leur site Internet <http://www.jaxio.com>

Si vous êtes éditeur et vous voulez que l'on se rencontre, envoyez-moi un email, mon adresse est sur le logo du touilleur en haut à gauche de cette page.

Prochaine interview du Touilleur : la société Prima Solutions à Paris avec Dimitri Baeli.

Jaxio est partenaire du [PJUG](#), Paris Java User Group.

Vous pourrez les croiser à la prochaine réunion PJUG le [mardi 10 juin sur Spring](#).

## JSR 303, vous avez validé votre bean ?

Un petit billet sur la JSR 303, que j'avais commencé il y a plus d'un mois. Tout d'abord en quelques mots, la JSR 303 propose de standardiser en Java l'expression des contraintes du monde objet ainsi qu'un moteur de validation. En effet nous vivons dans un monde de contraintes dès que nous commençons à programmer. Sur une classe Personne, l'attribut âge ne peut être qu'un entier positif par exemple. Dans le monde de la finance, des règles bien plus complexes sont aussi implicites pour les personnes qui manipulent des concepts financiers. La validation de ce que peut saisir l'utilisateur peut constituer une grosse partie métier d'une application.

Surgit un débat : où peut s'effectuer la validation par exemple dans une application web ? Dans la couche de présentation ? Dans la couche de service ? Au niveau du moteur de mapping objet-relationnel ? Ou dans la base de données ?

La JSR 303 propose 3 domaines dans sa solution :

- Comment exprimer une contrainte ?
- comment valider un objet ?
- comment retrouver et utiliser les contraintes définies sur un objet ?

**Dis papa c'est quoi cette petite contrainte ?**

Pour moi une contrainte c'est quelque chose en Java que le type d'attribut ne contraint pas assez. Dans une classe Java, un attribut de type Integer au lieu de String nous donne déjà une idée sur ce que l'on veut stocker. Le problème est que ce n'est pas suffisant pour donner de la consistance.

Prenons une classe Utilisateur avec un attribut email qui représente une adresse email.

```
/**
 * User pour tester la JSR 303...
 * Created by IntelliJ IDEA.
 * User: nicolas.martignole
 * Date: 05-Jun-2008
 * Time: 17:26:57
 */
public class Utilisateur {
    private String email;

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }
}
```

Je ne souhaite pas stocker un email qui serait null ou une chaîne de caractères de plus de 50 caractères. Pour cela je peux modifier mon bean et faire en sorte que la méthode setEmail jette une exception lorsque l'argument passé n'est pas correct :

```
public void setEmail(String email) {
    if (email == null)
        throw new IllegalArgumentException("Email ne doit pas être null");
    if (email.length() > 50)
        throw new IllegalArgumentException("Le champ email ne doit pas faire plus de 50 caractères");
    this.email = email;
}
```

Jusqu'ici cela fonctionne très bien. Les remarques sont que ma contrainte ne saute pas aux yeux, sauf à lire le code. Et en plus je ne peux pas l'introspecter et l'extraire du code pour m'en servir dans mon application par exemple.

Le type de message pour l'utilisateur ici est un peu violent. Une exception Java pour du code métier... Je ne suis pas très sympathique pour ceux qui se serviront de mon bean non ?

#### Est-ce qu'il faut exprimer la contrainte dans le modèle ?

Après tout je pourrais avoir le droit d'avoir une class java Utilisateur sans contraintes sur l'attribut email, mais une règle dans ma couche de présentation ou de service pour restreindre la puissance non ?

Si mon Utilisateur était un Entity bean, persisté vers une base, après tout je pourrais utiliser les types de colonnes dans ma base afin d'empêcher de stocker une adresse email null et de plus de 50 caractères en me basant sur des contraintes dans la base. Vous pensez pas que c'est un peu trop bas vous ?

```
CREATE TABLE UTILISATEUR {
    id bigint not null primary key,
    email varchar(50) not null
}
```

Je peux aussi dans ma couche de présentation ajouter une contrainte afin de tester les arguments saisis par l'utilisateur avant d'appeler ma couche de service. Ou dans ma partie métier, je peux encore exprimer une contrainte...

Bref le souci c'est qu'à en mettre partout, surtout entre la base et le modèle, on finit par ne pas voir exactement les contraintes sur les objets. En plus le souci est que l'on aura un empilement de contraintes les unes après les autres. C'est parfois souhaitable dans certaines architectures cependant.

Le meilleur endroit pour stocker les contraintes, tel qu'il est proposé par la JSR 303 est dans le bean lui-même. Donc au niveau du modèle.

La JSR 303 propose d'utiliser les annotations introduites en Java 5 pour tout simplement exprimer les contraintes sur le bean. Donc ma class User deviendra tout simplement :

```
/**
 * Petit classe pour demontrer le principe de la JSR 303.
 *
 * Created by IntelliJ IDEA the best IDE in the world. Eclipse sucks.
 * User: nicolas.martignole
 * Date: 05-Jun-2008
 * Time: 17:48:09
 */
public class Utilisateur {
    @NotNull(message = "Le champ email ne peut pas être null")
    @Email(message = "Merci d'entrer un email valide")
    @Length(max = 50, message = "Pas plus de {max} caracteres svp")
    private String email;
```

```

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }
}

```

On constate que l'on peut maintenant indiquer que l'attribut email ne peut pas être null, doit être une adresse de courrier électronique valide et d'une longueur maximum de 50 caractères.

La JSR 303 permet d'annoter une class complète (Utilisateur) un attribut (email) mais pas encore les arguments passés à une méthode.

Vous pouvez créer vos propres contraintes et utiliser aussi des expressions régulières pour exprimer une contrainte. Je ne suis pas très inspiré donc disons qu'une méthode setAge prend en argument une String. Cette String doit être de type numérique, sur 2 digits. Par exemple « 03 » ou « 43 ». Mais pas « 2 » ou « toto »

L'annotation Pattern vous permet de déclarer votre pattern de validation :

```

@Pattern(regex = "\\d+\\d+")
@NotNull
public int parseAge(String age) {
    return Integer.parseInt(age);
}

```

Les contraintes peuvent être déclarées dans des classes abstraites, donc héritées par des classes filles. Lorsqu'une classe fille surcharge une méthode d'une super classe qui a une contrainte, la règle est additive sauf si vous redéfinissez la même contrainte. Je m'explique. Si dans une class « PetitUtilisateur » je surcharge à nouveau la méthode parseAge je peux tout à fait changer ma contrainte pour qu'elle soit moins restrictive.

Si nous souhaitons maintenant définir une contrainte pour qu'un numéro de carte bleue soit valide, il est possible de se lancer dans l'écriture de sa propre contrainte. Pour faire simple disons que l'on souhaite vérifier que le numéro comporte 16 digits et commence par 4973 (c'est le cas de la Société Générale).

Dans un premier temps on va déclarer une nouvelle annotation comme d'habitude en Java 5

```

@java.lang.annotation.Documented
@ConstraintValidator(value = CarteBleueValidator.class)
@java.lang.annotation.Target(value = {java.lang.annotation.ElementType.FIELD})
@java.lang.annotation.Retention(value = java.lang.annotation.RetentionPolicy.RUNTIME)
public @interface CarteBleue {
    String message() default "";
    String[] groups() default {};
    String bankName() default "";
}

```

ConstraintValidator permet de déclarer la classe Java à instancier afin d'effectuer la validation. Je déclare ma contrainte comme étant valide pour un champ (FIELD) et aussi comme étant retenue à l'exécution. Vous êtes obligés par contre d'utiliser la retention policy RUNTIME pour que la validation soit effective. Il faut aussi que les attributs message et groups soient déclarés. Message permet d'indiquer un message d'erreur par défaut et groups est utilisé pour les contraintes cascadées.

Voyons maintenant comment implémenter CarteBleueValidator pour donc effectuer une validation de nos numéros de carte bleue :

Attention : je n'ai pas testé le code suivant. Je l'ai écrit à la lecture de différents articles publiés par Emmanuel Bernard et sur les présentations de Java One 2008. A tester donc

```

/**
 * Validator for CarteBleue
 * Created by IntelliJ IDEA. No you really use Eclipse ? booh.
 * User: nicolas.martignole
 * Date: 05-Jun-2008
 * Time: 18:14:50
 */
public class CarteBleueValidator implements Constraint {
    private String bankName;

    public void initialize(CarteBleue annotation) {
        bankName=annotation.bankName();
    }

    public boolean isValid(Object value) {
        if (value == null) return true;
        if (!(value instanceof Double))
            throw new IllegalArgumentException("@CarteBleue only applies to Double");
        Double cbNumber = (Double) value;

        // Si moins de 16 digits alors le numero n'est pas complet
        if(cbNumber < Double.parseDouble("1000000000000000")) {
            // question : comment positionner un message d'erreur ? pas possible ici
            // car nous traitons un NotValid.
            return false;
        }
        if(bankName==null){
            // pas de banque dont on retourne true. Le numero est bon

```

```

        return true;
    }
    // Dans l'annotation on a mis bankName = "SG"
    if(bankName.equalsIgnoreCase("SG")){
        if(cbNumber.toString().startsWith("4973")){
            return true;
        }
        return false;
    }
    return true;
}
}

```

Une fois tout ceci écrit je peux alors ajouter un attribut Double dans ma classe Utilisateur et l'annoter avec mon Validateur :

```

@CarteBleue(bankName = "SG")
private Double creditCardNumber;

```

Comme vous pouvez l'imaginer il existe un certain nombre de validateurs par défaut afin de vous éviter de réécrire la même logique plusieurs fois.

Vous pouvez déclarer :

```

@NotNull, @AssertTrue, @AssertFalse et @NotEmpty pour des tests simples.
@Length, @Size pour les chaînes, les collections et les tableaux
@Min et @Max pour les Number.
>Email par exemple
@Pattern pour utiliser une expression régulière
@Valid pour simplement valider un objet ou un attribut.

```

La validation peut aussi maintenant être appelée par votre code métier, comme c'est le cas dans JBoss Seam qui utilise le moteur de validation d'Hibernate.

L'équipe de la JSR 303 travaille maintenant sur des problèmes plus pointus comme le support de l'internationalisation et sur le support de la validation des paramètres d'une méthode, ce qui n'est pas encore possible pour l'instant.

Je me pose pas mal de questions sur justement tout ce qui tourne autour de l'i18n. Prenons un attribut de type Date appelé « dateDeNaissance ». Vous serez d'accord avec moi pour dire que lorsque l'utilisateur effectue la saisie de la date de naissance dans la couche de présentation, si l'on veut valider cette date il va falloir implémenter des Validators en mesure de détecter la Locale de l'utilisateur. Moi en tant que français je vais utiliser jj/mm/aa. Un anglais par exemple va saisir mm/jj/aa. La question que je me pose : si le Validator se base sur la Locale courante de Java, qui est en fait celle du serveur d'application par exemple, celle-ci peut être différente de la Locale de l'utilisateur... Bref notre beau système de validation ne marchera pas dans ce cas...

Pour conclure je sais que je n'ai pas couvert toute la spéc JSR 303 et que le sujet est encore vaste. Il y a aussi des débats sur le risque de se dire que trop d'annotations tue l'annotation... J'en discutais avec Florent Ramière lundi dernier.

Ne risque pas-t-on de se retrouver avec des classes hyper annotées avec peu de code ?

Flavien (un autre lecteur assidu) va me demander encore si au niveau performance, cela ne coûte pas plus cher que de coder à l'ancienne. S'agissant d'annotation de type RUNTIME la réponse est non à priori. Il y a un peu d'introspection à l'exécution pour instancier le bon bean de validation, puis ensuite le reste ne doit pas coûter trop cher. D'ailleurs la méthode isValid du validateur doit être implémentée pour supporter des appels concurrents. Vous voyez ce que cela veut dire ?

Sur ce je vous souhaite une bonne soirée.

Mots clés : jsr303, jsr-303, jsr 303

## Interview de Dimitri Baeli, Prima Solutions

Mardi rencontre avec Dimitri BAELI, Architecte Expert [chez Prima Solutions](#). Dans la foulée de mes interviews avec des éditeurs, voici une rencontre où l'on a plus parlé Java stuff finalement que de Prima Solutions. Mais c'était bien fun.

Je connais Dimitri depuis 6 ans. Nous avons travaillé ensemble chez Dotvision il y a pas mal de temps avant que je ne rejoigne Thomson-Reuters en 2003.

### Nicolas : Salut Dimitri, d'abord quelques mots sur Prima Solutions ?

Dimitri: Prima Solutions est un éditeur français spécialisé dans les solutions pour le monde de l'assurance. Nous sommes une trentaine, avec un bureau à Paris dans le 17ème et un bureau à Chicago. Nous proposons 3 produits différents : Prima IBCS, Prima Repository et Prima Vanilla. Je travaille sur Prima IBCS, qui est un modèle objet permettant de définir les concepts types du monde de l'assurance. Cela facilite l'écriture ensuite d'un socle de services pour n'importe quel type d'application du monde de l'assurance.

### Quel est ton rôle chez Prima ?

Je suis Architecte en charge de la définition des outils et des interfaces de nos logiciels. Je suis aussi amené à effectuer des entretiens d'embauche.

### Tu étais à JavaOne 2008 cette année, alors quelles sont tes impressions ?

Je dirai qu'en premier nous avons vu le retour des Applets via une nouvelle implémentation du Plugin Java qui est enfin quelque chose de très mature. Il était temps avec sur le marché des concurrents comme Microsoft Silverlight et Adobe AIR. Les présentations et la qualité des

intervenants à JavaOne en font un événement important de la communauté Java. Et puis j'en ai profité pour aller voir quelques amis français installés dans la Silicon Valley.

J'ai assisté à plus de 30 sessions. Entre autre à la soirée très sympa [des Java Posses](#), les célèbres Podcasteur qui chaque semaine réalisent un podcast en anglais sur Java et le monde de l'open-source.

Il y avait quelques monstres comme le créateur de Mylyn par exemple dont la présentation valait le détour. On va reparler de Mylyn tout à l'heure je te ferai la démo.

### **C'est quoi ce stylo que tu as sur la table ? (je vois un gros stylo noir avec un écran LCD)**

Arf. Là j'ai craqué. 150 \$ c'est pas cher quand tu vas voir ce que l'on peut faire avec ce stylo magique. Il s'agit d'un stylo [LiveScribe](#) dont d'ailleurs les Java Posses parlent dans l'épisode #189 de la semaine dernière. Ce stylo te permet tout d'abord de prendre des notes... Hop j'écris sur ce bloc. Cependant comme on peut le voir, le papier de mon bloc de feuille est un peu spécial.

*(En effet en regardant de pres, des petits points bleus sont dessinés)*

Le stylo est doté d'un scanner et d'un microphone. Tu peux donc enregistrer par exemple la voix d'un orateur pendant que tu prends des notes... Et là où c'est vraiment fort, c'est que lorsqu'ensuite tu repointes une ancienne note dans ce carnet... (il tourne des pages et arrive à un plan de jardin) tu peux réécouter le son attaché à la note....

*(on entend alors une discussion avec Vincent Massol sur une tondeuse automatique. Vincent avait dessiné un plan de jardin)*

Pour la démo : <http://www.livescribe.com/cgi-bin/WebObjects/LDApp.woa/wa/MLSOVeriewPage?sid=t5S11hblGhQV#>

### **Est-ce que la reconnaissance de la page par le stylo est rapide ?**

Le LiveScribe est vraiment rapide. Tiens hop là je pointe le mot "JSR-303" sur mes notes, et tu peux entendre la présentation d'Emmanuel Bernard enregistré à JavaOne... Pas mal non ?

### **C'est assez bluffant... J'ai voulu acheter un dictaphone numérique pour le Touilleur... entre 200 et 400 EUR... Et là pour 150 \$ tu me montres une solution vraiment top. Mais le bloc papier est spécial non ?**

Oui en effet. Cependant on parle de pouvoir imprimer ces feuilles spéciales à la maison avec une imprimante capable de sortir du 600 dpi. Donc à priori cela veut dire que l'on ne sera pas attaché à acheter des blocs spéciaux.

### **Pour résumer : le papier a des petits points à peine visibles qui font une sorte de code barre unique. La caméra du stylo à priori associe donc les points visualisés avec un marqueur de temps sur un fichier enregistré avec le micro, qui est dans le stylo...**

En effet c'est cela.

La capacité est de 1 Gb pour 149 \$ et 2 Gb pour 200\$. Il est possible d'en acheter en ligne sur [LiveScribe.com](#)

### **C'est vraiment intéressant. Passons à la démo de Mylyn dont tu m'as parlé lors de la soirée Xebia Poker si tu veux bien.**

[Mylyn](#) n'est pas facile à décrire sans utiliser les mots "génial", "trop de la balle", "ça déchire" mais je vais essayer. Donc le principe de Mylyn, outil open-source pour Eclipse, est de permettre à un développeur d'associer sa vue courante dans Eclipse à une tâche. Par exemple je décide de commencer à implémenter la tâche #165320 que tu vois ici sur [XPlanner](#)...

### **XPlanner c'est ?**

... un gestionnaire de projet et de suivi de tâche orienté eXtreme Programming simple et puissant que nous utilisons chez Prima. (Voir le site [XPlanner](#)).

### **Donc dans ton écran d'Eclipse, je vois que tu cliques dans un panel "Tasks" sur cette tâche pour l'activer ?**

Exactement. Sur mon écran d'Eclipse comme tu peux le voir il n'y a pas de volumes montés, pas de code source... en fait il n'y a rien d'affiché. Lorsque je vais commencer à fixer mon bug, je vais ouvrir petit à petit mes packages Java au lieu d'avoir à gauche toute l'arborescence de mes packages. Sur notre projet nous dépassons les 150 000 lignes de code. Je serai pollué visuellement avec tous ces packages et classes qui ne me servent à rien.

### **J'ai un peu de mal à comprendre pour l'instant**

Il faut savoir que Mike Kersten, le concepteur de Mylyn, est un universitaire de haut niveau spécialisé en Psychologie. Il a expliqué lors de sa présentation que notre cerveau tend à se perturber lorsque trop d'informations sont affichées à l'écran. La mémoire à court terme, d'ailleurs comme en Java, est limitée en capacité. Ainsi si nous voulons faciliter le travail et obtenir les meilleurs conditions, il faut que les IDE allègent la vue. Donc ici Mylyn me permet de ne rien voir de mon projet dans un premier temps. Au fur et à mesure que ma correction de bug va prendre forme, je vais ajouter les quelques classes associées à ma tâche. Là où tout ceci est vraiment fort c'est que je peux échanger mon environnement de travail avec un autre développeur.

### **Donc si je travaille avec toi et que tu me demandes de terminer ta correction de bug ?**

Et bien tu lances ton Eclipse, tu sélectionnes le bug dans ta liste, et lorsque tu cliques dessus, tu peux récupérer exactement le même environnement que moi. Même vue, même espace de travail dans Eclipse.

Ensuite une fois la correction terminée, lors du commit vers SVN comme tu peux le constater Mylyn prépare une liste très précise des fichiers à commiter et même le message de commit avec l'id de la tâche .

Mylyn est maintenant de base dans les distributions Eclipse Europa et Ganimède à venir.

J'ajoute en plus que Mylyn permet aussi d'accéder à Mantis et de gérer ses tâches personnelles. Tu peux aussi n'exécuter que les tests unitaires concernés par la tâche. La complétion dans Eclipse se concentre uniquement sur les classes et les méthodes récemment utilisées. L'architecture de Mylyn

est ouverte. Je pense que nous devrions voir des plugins pour NetBeans et IDEA IntelliJ très rapidement.

*(Difficile de reprendre ici l'intégralité de la présentation de Dimitri. Je vous propose d'essayer par vous-même Mylyn pour en comprendre l'intérêt. Voir le site <http://www.eclipse.org/mylyn/>)*

### **Merci Dimitri pour ces informations. Est-ce que tu recherches des personnes pour Prima Solutions ?**

Nous sommes toujours à la recherche de candidats motivés ayant un bon socle de connaissances en UML. Nos produits utilisent l'approche MDA et donc une bonne maîtrise d'un outil comme MagicDrawUML est un plus. Ensuite quelqu'un qui aime coder et qui veut bien rejoindre une société sympa.

### **Le dernier achat que tu as fait ?**

On s'est acheté une WiiFit pour le bureau pour accompagner la wii achetée l'année dernière. Vraiment très sympa entre midi et deux pour se détendre entre développeurs. On a notre petite salle de FitNess !

### **Excellent ! Merci pour ton temps et à bientôt !**

A bientôt

## **iPhone 2 disponible**

C'est fait. Comme il sait le faire si bien à chaque fois, Steve Jobs annonce la sortie de l'iPhone 2 avec donc un module 3G, le GPS et deux capacités de stockage : 8 Go et 16 Go. L'écran et l'appareil photo restent inchangés. L'autonomie en 3G est de 5H, ce qui représente en effet une petite autonomie. L'appareil est plus fin selon Steve Jobs. Du côté entreprise, intégration avec ActiveSync ce qui permettra de synchroniser son agenda avec une messagerie Outlook. Excellente nouvelle pour le déploiement en entreprise. Le module GPS est intégré à Google Maps. Il permet une géolocalisation très fine.

Aux USA le modèle 8Go sera vendu 199\$. Le modèle 16Go sera vendu 299\$. Il sera disponible en blanc. Le modèle est déjà disponible [sur l'AppleStore Français](#) qui ne donne cependant pas le prix en EURO mais simplement une liste de points de ventes [sur cette page](#).

Orange annonce la sortie [le 17 juillet prochain](#) en France de l'iPhone2 dans leurs magasins. Par ailleurs une offre de reprise pour l'iPhone 1 sera aussi proposée aux premiers acheteurs de l'an passé.

Moi je craque complètement... quand je vois que je me traîne mon vieux Siemens-Bouygues, j'ai vraiment envie de dégager cet abonnement et de me prendre un petit iPhone2... Avec le MacBook Pro je crois que je suis entrain de me transformer en switcher fou...

## **Compte-rendu de la soirée Spring PJUG**

Je vous le dis simplement : hier soir vous auriez dû venir. Oui vous auriez vu tout d'abord un phénomène assez curieux. Arrivé à 19h30 je retrouve Antonio à l'extérieur, et à cet instant il n'y a pas encore beaucoup de monde. Avec 189 inscrits, Antonio préfère prendre les devants et demande une salle plus grande. A 19h50 c'est encore calme, une trentaine de personne. A 20h00 il y a plus de 100 personnes... A 20h15 la salle est presque complètement remplie. 189 inscrits sur le site du PJUG. Du coup [Jean-François Hélie](#) ([Octo](#)) et Christian Blavier ([Octo](#)) sont un peu impressionnés devant tout ce monde.

Les thèmes de la soirée : Spring Test Driven Development présenté par Octo, puis ensuite présentation de Spring Source, Spring Security 2 et de SpringSource Application Platform par Julien Dubois et [Michaël Isvy](#) de la société SpringSource France.

Après une introduction par Antonio, qui rappelle l'importance des sponsors pour le [ParisJUG](#), Jean-François Hélie et Christian Blavier s'installent pour la démonstration du développement piloté par test sur Spring 2.5. Test Driven Development en anglais dans le texte. Le projet sera le suivant : écrire en 45 minutes une application de Blog simple capable de lister et poster des articles. Jean-François est aux commandes d'Eclipse sur son MacBook (yeah Mac rulez...) et Christian fait les commentaires. Jeff a réalisé un exercice loin d'être évident devant tant de monde : il a codé sous nos yeux l'application en appliquant simplement les principes du TDD.

Christian explique que l'on commence par un test JUnit qui nous sert de cahier des charges. Dans ce test, le nom de la méthode exprime l'intention que l'on souhaite développer. En l'occurrence pour ce premier test, une méthode « shouldListArticles » si ma mémoire est bonne. Pour cela, Jean-François implémente un BlogDAOTest dans lequel il place son premier test. Ce que l'on voit est très technique. Un interlocuteur dans la salle qui n'est pas développeur décroche forcément. En même temps je me dis que c'est un excellent moyen de montrer à une personne peut-être moins technique cette méthode d'implémentation pilotée par les tests.

Impossible ici de reprendre le code complet de Jean-François. En une petite heure il aura écrit un DAO, un bean BlogService et son test et enfin un BlogController avec son test. L'accès à la base de données Hypersonic s'effectue à travers Hibernate. La partie rendu web est gérée par Spring MVC.

Du côté des tests, nous avons vu que Spring permet d'alléger la quantité de code à écrire, en particulier dans la partie « plomberie ». Quelques annotations comme `@InjectIntoByType`,

@Mock, @TestedObject, @Service ou @Repository permettent de marquer le code et d'écrire des tests d'intégration simplement.

Nous avons vu l'utilisation de bouchon dans les tests. Jean-François parle d'Unitils (<http://www.unitils.org/>). Cette librairie open-source de tests unitaires facilite l'écriture de tests unitaires et d'intégrations. En particulier lorsque vous souhaitez tester des DAO sans faire d'accès à la base de données.

Durant la présentation, je pense à Seam. En fait je m'imagine à la place de Jean-François entrain de réaliser exactement le même exercice avec Seam Framework (JBoss Seam). La chose qui manque pour l'instant à Spring c'est une couche de rendu Web plus puissante. C'est ce qu'offre déjà Seam avec l'intégration d'IceFaces et de RichFaces, ainsi que de Facelets pour la gestion propre des pages. Cependant Julien Dubois expliquera plus tard dans la soirée qu'il y a des projets chez Spring afin d'y intégrer un moteur Ajax puissant d'ici à la fin de cette année. J'ai vu que Spring Web Flow propose une intégration avec JSF, ce qui en fait répond à mes interrogations.

En conclusion pour cette partie, avant tout bravo à Jean-François et Christian. La présentation était pointue, concrète et donne envie de suivre ce chemin pour l'implémentation d'applications webs autour de Spring 2.5

Après un petit break d'une quinzaine de minutes où j'ai croisé pas mal de monde, Julien Dubois et Michaël Isvy de SpringSource ont pris la parole. Julien est responsable de SpringSource pour la France et membre de l'OSSGTP. Il est aussi le coauteur de « Spring par la pratique » aux éditions Eyrolles dont j'ai parlé dans plusieurs articles sur le Touilleur. Michaël Isvy est formateur et consultant chez SpringSource, ainsi que contributeur sur Spring Security 2.

Spring c'est 2.5 millions de téléchargement par mois. Les offres d'emplois avec « Spring » dépassent les offres purement J2EE aux USA. Julien résume assez bien la philosophie de Spring : « dites-nous de quoi vous avez besoin, nous vous le fournissons » qui tranche un peu avec « voilà tout ce que vous pouvez faire, à vous de ne prendre que ce dont vous avez besoin ». D'ailleurs un excellent petit clin d'œil aux « vieux » éditeurs J2EE qui vivent d'une rente établie en 2000 avec l'architecture J2EE EJB 2.1. J'ai vu deux personnes de BEA dans la salle qui rigolaient et qui donc semblaient d'accord. Et j'ai gagné un superbe tee-shirt dans la série questions-réponses sur Spring.

Ce matin sur Monster.fr j'ai fait un test assez simple. J'ai tapé simplement un mot clé sur la zone de recherche et j'ai listé les résultats.

Dites-moi ce que vous en pensez :

Mot clé Nombre d'offre d'emplois affichés :

```
Spring.....145
EJB.....84
Seam.....1
JBoss.....83
J2EE.....634
websphere.....201
TDD.....2
RIA.....11
Weblogic.....157
.NET.....600
Java.....1097
C++.....609
Scrum.....17
Agile.....48
```

Source: *Le Touilleur Express sur Monster.fr mercredi 11 juin 2008*

Premier constat : il y a plus d'offres d'emploi contenant le mot clé Spring que le mot clé EJB sur Monster.fr. On constate que Java arrive en tête devant .NET, ce qui me rassure et me fait très plaisir. J'aime bien .NET mais si vous voulez, mon truc c'est plutôt Java.

Après la présentation de la société SpringSource, qui assure des formations, du support et des missions de consulting en France, Michaël Isvy présente Spring Security 2. Première information pour moi, béotien de Spring pour l'instant, c'est que Spring Security 2 était connu sous le nom de... Acegi Security 1.0. Spring Security 2 c'est donc Acegi 2 si vous voulez.

Ce module de Spring couvre les manques de l'architecture J2EE actuelle sur la sécurité sans cependant introduire des dépendances avec un serveur d'application. Spring Security est développé par Ben ALEX et Luke TAYLOR. La version 2.0.2 est sortie le 5 juin dernier. Parmi les fonctionnalités de Spring Security j'ai noté

- support de l'authentification par cookie
- possibilité de bloquer 2 authentifications simultanées avec le même profil
- Support d'OpenID (Google, SUN, IBM...)
- Support de la JSR 250 (EJB3) et donc de la partie sécurité des EJB 3.
- WS-Security pour les Web Services

Voir le site pour plus de détails : <http://static.springframework.org/spring-security/site/>

Nous avons vu quelques exemples qui donnent envie de s'intéresser à Spring Security 2 pour les applications webs.

Ensuite Julien a présenté SpringSource Application Platform. Il s'agit d'un moteur d'exécution (PAS d'un serveur d'application) basé sur Tomcat, Spring et OSGi. Il permet de faire fonctionner une application Spring en utilisant le moteur de modules d'OSGi. D'ailleurs j'ai noté un jeu de mots de Julien « OS-JEE-Aie... » pour OSGi.

A mon sens c'est une grosse annonce, un gros projet et qui va faire bouger l'écosystème des éditeurs J2EE. S2AP (SpringSource Application Platform) a décidé de prendre le système de

modules d'OSGi car celui-ci est mature et c'est ce que le marché demande.

Julien Dubois explique ensuite qu'entre les membres d'OSGi et les membres de la JSR 277, il y a des discussions assez franches. La spécification JSR 277 porte sur Java Module System. La JSR 316 de JEE6 regroupe 27 spécifications comme l'explique Antonio. Les discussions portent sur le fait qu'OSGi qui est aussi une JSR et la JSR 277, certaines parties sont identiques. OSGi est déjà en production puisqu'Eclipse s'en sert alors que la JSR 277 est encore en phase de définition. Julien explique que le Java Community Process (JCP) n'est pas le seul groupe à avoir le droit de faire des spécifications contrairement à ce que l'on pense. La fondation OSGi est aussi autorisée à proposer ses propres spécifications. C'est ce qu'elle a fait avec la JSR 291 justement... Et cela provoque des frictions entre les 2 groupes d'expert de chacune de ces 2 JSR.

SpringSource Application Framework met un coup de pied dans la fourmilière. Sa licence GPL v3 provoque des débats riches et passionnés dans la communauté Spring. Julien explique cependant que même si la licence est GPL v3, une application web déployée sur le conteneur S2AP n'est pas obligée de devenir GPL à son tour. C'est clairement expliqué dans la GPL v3. Un site internet qui tourne sur un serveur GPL v3 n'est pas obligé d'être lui-même en GPL. Par contre, un développement pour ajouter une fonction particulière à un serveur d'application ne pourra pas être propriétaire et doit être rendu à la communauté open-source. J'avoue que je suis perplexe mais que je pense que tout ceci est plutôt positif pour nous, en tant que développeur et intégrateur. N'étant pas moi non plu avocat, je ne m'avancerai pas trop dans le débat pour l'instant.

Sinon pour continuer, j'ai lu sur InfoQ que l'an passé, Sun a voté contre la JSR-291 (OSGi) et propose de prendre la JSR-277 qui quoique plus ancienne, est encore en phase bêta. La JSR-291 est en production, la 4ème implémentation de référence est disponible. Si vous codez avec Eclipse, vous vous en servez sans le savoir. Cependant Sun (enfin une partie de Sun) soutient la JSR-277 car celle-ci propose une modularisation au niveau de Java SE lui-même, là où la JSR-291 se présente comme un module fonctionnel à la maven en quelques sortes. Il semble aussi que les membres de l'OSGi sont assez inquiets ou furieux que la JSR-277 ne reprenne pas stricto sensu leurs idées déjà implémentées dans OSGi...

J'imagine que si l'expert group de JPA n'avait pas travaillé avec Hibernate, que la spécification WebBeans n'était pas la suite de SeamFramework, nous aurions les mêmes discussions animées. Mais au-delà de ces discussions, le fait que Spring propose en plus maintenant un vrai serveur basé sur OSGi fait que l'on risque un schisme entre les pro-JSR277-SUN et les pro-JSR291-OSGi-Spring...

Les dernières discussions chaudes du côté de la JSR-277 : Peter Kriens membre de la JSR-277 explique que SUN pilote dans son coin cette spécification et que l'Expert Group de la JSR est là pour valider à posteriori, sans aucuns débats. Il explique que la proposition des modules Java (JAM) n'est pas une bonne proposition et qu'il serait plus logique de reprendre les propositions de la JSR-291... Ambiance...

(voir <http://www.infoq.com/news/2008/05/jsr277-osgi>)

Bon pour revenir à nos moutons, car là je m'éloigne, Julien a expliqué que S2AP propose un système modulaire (OSGi donc) qui permet à un développeur de ne redéployer qu'une partie de son application. Par exemple il sera possible de ne mettre à jour que le Contrôleur d'une application Web, sans devoir mettre à jour le reste de l'application. Cela répond à des besoins exprimés par les développeurs : facilité de déploiement, moins d'attente et possibilité en production de changer à chaud qu'une partie de l'application. En effet c'est un vrai cas industriel, même si les développeurs dans la salle semblaient sceptiques. Bon j'avoue que chez nous c'est encore l'âge de pierre, et que l'histoire de ne changer qu'une partie de l'application, j'attends de voir. Certains développeurs fonctionnels ne savent pas se servir de Weblogic ou compiler en mode « exploded » pour travailler avec un JBoss. Par contre je vous explique une chose : ils vous mettent une petite claque en finance et en modélisation des données du marché. Donc chacun son boulot. Mais du coup, j'avoue être sceptique pour l'histoire de « je ne déploie qu'une partie de mon code qui a changé » même si cela semble plus simple. Julien explique que nous pourrions arrêter à jouer à la sauce de JAR dans le WAR pour faire tout et n'importe quoi.

La beauté de S2AP est qu'il n'y aura pas de dépendances dans notre code sur OSGi. La complexité d'OSGi est gérée par S2AP. Nous nous concentrons sur le code métier et pas sur la plomberie. « Dites-nous de quoi vous avez besoin et on s'en occupe ».

Je passe une discussion sur JEE 6 et les 3 profils (A, B et C). Mais clairement S2AP vise la compatibilité avec les profils A et B de JEE6, pas le profil C qui contient des dinosaures comme EJB 1 et 2/x.

La bêta 6 de S2AP est disponible et SpringSource annoncera la sortie de la version 1.0 très bientôt. Malgré mes clins d'œil pour que Julien lâche une date, il a répondu « désolé Nicolas, tu ne pourras pas bloguer dessus ce soir... ».

En même temps avec la tenue aujourd'hui et demain de SpringOne à Anvers, je pense que nous devrions avoir une petite annonce très bientôt !

Pour plus d'informations :

<http://www.springsource.com/products/suite/applicationplatform>

J'ai noté ensuite les questions/réponses mais nous sommes plus dans le débat qu'une vraie discussion de fond. Je crois qu'il faudra que je vous présente les idées d'OSGi dans un prochain article car il y a pas mal de choses à dire.

Pour faire simple et expliquer OSGi : les modules sont la représentation à l'exécution de ce que Maven2 propose à la compilation. Un package applicatif qui tourne dans un moteur OSGi isole ses

dépendances de ses voisins. Chaque partie de votre application web pourra par exemple utiliser sa version d'un parseur XML sans problèmes.

Sur ce je vous laisse, je crois que je vais finir par acheter un Pulse SmartPen de [LiveScribe.com](http://www.livescribe.com) pour prendre des notes.

## Quelques sites indispensables pour vos développements Webs

Ce matin un post qui n'a rien à voir avec Java, ni Apple... pour une fois ! J'ai en réserve un bon nombre d'adresse que j'utilise lorsque je dois valider un site, créer rapidement un logo ou trouver une mise en page CSS sympathique. Je n'ai pas tout repris ici mais voici déjà quelques uns de mes liens. [BrowserShots](http://www.browsershots.org) est un site qui vous permet de tester votre site internet avec différents navigateurs. Pour cela le site visite votre page et effectue une capture d'écran, que vous pouvez ensuite télécharger. Le Touilleur Express par exemple ne marche pas très bien avec certaines versions d'Opéra. Pratiquez et rapide. <http://browsershots.org/> L'adresse de courrier électronique poubelle, vous connaissez ? Disons que vous avez besoin de vous inscrire afin de recevoir un email d'un site marchand un peu douteux, le site [SpamBox](http://www.spambox.com) propose de créer un email valable une heure seulement. Cependant avant de m'en servir, je vérifierai bien si ce site lui-même n'en profite pas pour récupérer votre vraie adresse email pour la revendre... On est jamais trop prudent. Le logo en flamme, le logo en flamme ! Si vous cherchez quelques tutoriaux afin de créer votre logo en flamme, allez jeter un oeil sur [le blog de DesignVitality](http://www.designvitality.com). J'aime particulièrement les designs Web 2.0. Le site [DaFont](http://www.dafont.com) offre une très grande collection de polices pour réaliser des applications webs, des sites et des logos. Il y a des polices très originales qui permettent de réaliser la patte d'un site en quelques instants. Le site qu'il faut avoir dans ses bookmarks car il contient finalement un peu prêt tout ce dont j'ai besoin lorsque je crée un habillage : [http://webdevelopersfieldguide.com/](http://www.webdevelopersfieldguide.com/). Par exemple [une roue colorimétrique](http://www.colormetric.com) pour l'équilibre des couleurs. Ou un système comme [DeGraeve.com](http://www.degraeve.com) qui se base sur une image pour vous proposer une carte de couleurs. Pour suivre l'actualité autour d'Ajax et des librairies comme [Prototype](http://www.prototype.com) ou [Script.aculo.us](http://www.scriptaculo.com), je vous avais déjà parlé d'[Ajaxian.com](http://www.ajaxian.com). Le site propose souvent des articles assez pointus sur Ajax et Javascript. En ce moment nous corrigeons des problèmes de Closures en Javascript. Je cherchais sur le net un outil que j'avais testé il y a 2 ans pour identifier des closures... Je tappe quelques mots et je tombe... sur mon blog... Ca fait bizarre de se répondre à soi-même... Si vous voulez briller en société et expliqué à un client ou à un consultant ce qu'est une Closure, je vous redonne [cette page](#) très bien faite, avant de vous proposer à mon tour une explication en français dans le texte prochainement. Oui on ne le dit pas assez, mais un certain nombre de librairies Ajax peuvent causer des problèmes de mémoire du côté du navigateur.

## Compte-rendu de la journée Valtech RIA

Hier grosse journée. Tout d'abord chez Valtech pour assister à une journée de présentation et de formation sur les 3 technologies phares des applications riches : Microsoft Silverlight, Adobe Flex et Google Web Toolkit. Ensuite réunion avec l'OSSGTP (où j'ai fait la connaissance de [Didier Girard](#) qui était aussi à session Valtech). Voici un petit compte-rendu rapide sur mes impressions de cette journée.

L'organisation de Valtech était à la hauteur de l'événement. Environ 90 personnes étaient présentes. En majorité des développeurs Java. Il y avait quelques développeurs .NET, PHP et des graphistes. Mais le premier constat, c'est que soit le fichier de prospection de Valtech n'a contacté que des équipes Java, soit clairement un développeur client riche en France n'est pas un graphiste mais un développeur...

Nous étions environ une cinquantaine dans mon groupe. Tout le monde était équipé d'un portable avec les différents environnements de développement installés grâce à un CD donné par Valtech. La première session sur Google Web Toolkit était animé par Sami Jaber de DNG Consulting, ancien directeur technique chez Valtech Toulouse. Sami a été interviewé plusieurs fois dans 01 Informatique ([ici](#), là sur [JavaFX](#) et [ici sur RDA](#)). [Son blog](#) parle aussi de .NET. Je retiens plusieurs points sur sa présentation. Tout d'abord il explique que GWT n'est pas réellement comparable à SilverLight ou Flex. Il n'y a pas de plugin et pas de rendu vectoriel. GWT est orienté application web riche. L'avantage de GWT est son intégration complète dans l'espace de travail du développeur. Le développeur écrit en Java une application similaire à de la programmation avec Swing pour décrire votre interface et vos actions. Ce code génère du Javascript avec l'outil GWT qui est ensuite déployé vers le client. L'un des points forts de GWT est la génération optimisée du Javascript selon le navigateur ainsi que selon la langue de l'utilisateur. GWT prend en charge la complexité de gestion des différents navigateurs et permet donc de ne se concentrer que sur la partie métier. A peu de chose prêt c'est ce que nous faisons avec mon équipe sur le framework Karma depuis 3 ans, pour la finance. Là où par contre GWT va plus loin, c'est son intégration dans l'environnement de travail. Chez nous la productivité n'est pas mirobolante car la phase de debug est assez délicate. Sur GWT la productivité est bien meilleure que notre solution.

GWT seul est pauvre en terme de composants graphiques. Cependant il faut savoir qu'un bon nombre des librairies Javascript sont en court de portage vers GWT. Sami a présenté une intégration de la librairie [ExtJS](#) dans le projet [Ext GWT](#) (ou GXT) qui m'a particulièrement intéressé. ExtJS est une librairie très puissante mais difficile à programmer proprement, car c'est du Javascript. Il existe bien une taglib pour ExtJS mais je ne suis pas convaincu. Ext GWT par contre est vraiment une librairie qui a un avenir au sens industriel du terme. Je vois déjà les applications chez Thomson-Reuters qui pourraient en bénéficier pour pas très cher. La librairie est disponible soit sous licence GPL v3, soit sous licence commerciale si vous souhaitez l'intégrer dans un

produit. C'est la mode en ce moment.

GWT est donc une librairie et un moteur complet qui permet de travailler d'Eclipse (ou d'IDEA IntelliJ) rapidement, sans devoir déployer son application et perdre du temps. L'exemple de Sami utilisait [Apache OpenEJB](#), un tout petit moteur EJB3 qui a l'avantage de se lancer très vite par rapport à un JBoss. Je ne parle même pas d'IBM Websphere et compagnie...

**Deuxième séance: Microsoft Silverlight beta 2 release 2.** Je ne vais pas m'étendre. Très décevant. La présentation était moyenne, les 2 présentateurs ont proposé d'axer la présentation sur l'intégration entre l'utilisation de Microsoft Blend et d'un outil de développement C#/XAML afin d'expliquer la séparation entre le design graphique et la partie codage. Mon sentiment sur ce que j'ai vu, c'est que Silverlight n'est pas encore terminé. Certes on parle d'une bêta. Mais en fin de présentation, l'un des speakers a expliqué que dès aujourd'hui il était possible d'aller en production avec un support qui sera assuré par Microsoft... Moi désolé mais je n'ai rien compris.

J'ai apprécié Microsoft Blend qui est un outil puissant pour construire son interface. Il écrase Flex Builder car il permet vraiment d'aller très loin sur les effets graphiques et de présentation. En terme d'ergonomie, je trouve qu'il y a encore quelques points à clarifier, mais l'outil est utilisable pour faire une interface très sympa. Du côté de XAML et du pont avec la partie métier et service, une application cliente Silverlight est agnostique avec la partie serveur. Elle peut donc être client d'un Tomcat sans soucis. Silverlight supporte un grand nombre de modèle de communication comme REST ou RPC. Il y a une couche de sécurité que je n'ai pas vu du côté de Flex. Anecdote cependant qu'un des présentateurs a lâché : le contenu XAP d'un fichier silverlight est placé dans le cache du navigateur. Il serait possible de récupérer ce contenu pour le décompiler et en faire ce que l'on veut. Il explique donc qu'il faut faire attention à ne pas stocker de choses sensibles dans l'application, mais à aller les chercher sur le serveur. Ce qui me paraît logique.

Quoi d'autre sur Silverlight ? je vous passe les manipulations assez acrobatiques que nous avons dû faire car il y a un souci de Namespace sur la dernière version de Blend. C'était folklorique. Moi en tant que client j'attendrai encore un peu avant de m'en servir pour un client final. Je n'ai pas du tout aimé la partie C#. Cela dit, Silverlight a aussi un support pour PHP et Ruby. Donc on peut envisager d'écrire autrement la partie Page de l'application. Voili voilà.

Nous avons terminé la journée par la présentation qui m'a le plus bluffé : Adobe Flex. Je connais bien la techno puisque l'an passé j'ai réalisé tout un prototype avec Flex de l'une de nos applications. Je voulais rester afin de voir comment présenter Flex aux gens qui ne connaissent pas. [Yann Chevalier de Baao](#) nous a fait une superbe présentation. C'est un très bon speaker et je pense que toute la salle a réussi à réaliser les ateliers qu'il présentait. Nous avons codé un player de vidéo en Flex, avec une timeline, des boutons Pause et Stop et un compteur de temps. Cela permet de voir comment utiliser le Designer pour créer l'interface. Ensuite comment binder un composant à un événement. Comment déclarer un composant réutilisable... Bref une très bonne présentation. Rien à dire, je reste convaincu que pour l'instant Flex a une bonne longueur d'avance sur Silverlight. Flex est pensé par des graphistes, des ergonomes, des développeurs Flash qui ont une grande expérience du Web là où Silverlight est orienté développeur. Cependant j'ai vu qu'en terme de rendu, Silverlight est plus puissant dans ses effets graphiques. Là où pour l'instant cela pêche un peu, c'est que des applications avec de la vidéo et des cubes qui tournent, il y a peu d'intérêt à en faire... Nous sommes encore en phase de transition vers du Web un peu plus riche, ce que GWT qui reste sur du HTML et du JS fait très bien.

En conclusion, je suis reparti avec un tee-shirt GWT. C'est pas marrant à dire mais les teeshirts Silverlight sont restés sur les tables de la salle d'accueil. Problème de publique je pense. Les gens présents étaient en majorité des pro-Java et malheureusement le fossé continue à se creuser. J'en suis responsable aussi en faisant du bashing ici. Mais je ne vais pas pipoter en racontant que c'est une techno mature et prête à être mise en production chez un client, un trader, dans une salle de marché... C'est une bonne techno mais avec un concurrent comme Adobe Flex et un produit comme [Thermo](#) qui devrait bientôt arriver, la concurrence va être rude pour Silverlight. A noter en juillet nous devrions avoir une présentation sur Flex par l'équipe d'Adobe France aux Paris Java User Group (Paris JUG).

## OSSGTP et présentation d'Elastic Grid

Update: cet article a été modifié après sa publication suite au retour de Jérôme.

Jeudi soir sur invitation de Dimitri Baeli, j'ai assisté à la réunion mensuelle de [l'OSSGTP](#). Après avoir présenté l'association je vous parlerai d'Elastic Grid, un projet très intéressant de Jérôme Bernard.

L'[OSSGTP](#) (Open source software get together in Paris) est un groupe français de développeurs de la communauté open-source francophone. Chaque mois l'idée est de se réunir et de parler de l'actualité des projets open-source, d'assister à des présentations en avant première et enfin de pouvoir échanger entre fans de la communauté open-source en France. C'est l'occasion de nouer de nouveaux contacts et de rencontrer des personnes actives dans la communauté open-source, majoritairement Java. Pour moi c'était aussi l'occasion de mettre des visages sur des noms que je connais depuis quelques temps, à force d'écumer les blogs sur Java.

Parmi ses membres, je connais déjà [Julien Dubois](#) (Spring Source) et [Dimitri Baeli](#) (Prima Solutions). J'ai fait la connaissance IRL de [François Le Droff](#) (Adobe, Flex) dont je lis le blog depuis pas mal de temps. Etaient aussi présent : Didier Girard auteur du blog [application-servers.com](#) que j'ai croisé plus tôt dans la journée [chez Valtech](#), [Jérôme Bernard](#) l'auteur du projet open-source Elastic Grid qui travaille avec Amazon EC2. Charles Gay auteur de [JGuard](#). Je n'ai pas retenu tous les noms des personnes présentes mais nous étions 9 en tout. Octo était l'hôte.

de la réunion.

Avant de vous parler d'Elastic Grid, je vais vous parler d'Amazon Elastic Cloud Computing (EC2). EC2 permet de louer une ou plusieurs machines sur la plateforme d'hébergement d'Amazon à l'heure tout en offrant un moyen via des Web Services pour démarrer autant de machines que vous souhaitez à partir de votre image ISO, les arrêter, gérer les règles du firewall, etc. Imaginez que vous ayez une image de votre serveur Linux avec votre Fedora, un serveur Tomcat et une base MySQL. Avec EC2 il est possible de déployer en quelques minutes cette image sur la grappe de serveur d'Amazon. Le prix dépend de la puissance et du nombre de machines que vous souhaitez. Je vous conseille de lire [ce post](#) qui donne un exemple de prix. EC2 est donc un service d'hébergement en Cloud computing qui vous permet de louer à l'heure un énorme nombre de machine pour faire quelques tests par exemple.

EC2 utilise Xen pour faire de la virtualisation. Il existe 5 profils de serveur différents. Du [petit serveur](#), processeur Xeon 2007 à 1.2Ghz, 160 Gb d'espace disque et 1.7Go de mémoire jusqu'au gros serveur 64bits avec 15Go de mémoire et l'équivalent d'un octo-processeur dual-core... De quoi bien s'amuser donc.

Pour travailler avec [Amazon EC2](#), vous devez créer une image AMI (Amazon Machine Image). Ensuite vous envoyez vers [Amazon S3](#) (simple storage service=s3) cette image iso. Via des webs services ensuite vous configurez le nombre de machine et vous pouvez alors les piloter à distance. Pour cela la communauté a développé des plugins pour Firefox ainsi que des plugins pour IDEA IntelliJ.

Jérôme Bernard d'Elastic Grid, LLC. est justement l'auteur d'un plugin pour IntelliJ, mais surtout le co-auteur du projet [Elastic Grid](#) avec Dennis Reedy et contributeur sur Rio. Je vais essayer d'explique le mieux possible son projet qui vaut le détour. Dennis Reedy est l'auteur du [projet RIO](#). Rio est un outil qui permet de déployer des applications, de les monitorer et de démarrer par exemple une application lorsqu'une panne est détectée sur un serveur.

Le projet Elastic Grid propose de réaliser pour vous du provisioning et de la reprise sur incident pour toute application déployé sur Amazon EC2. Prenons un cas typique que je proposais à Dimitri dans le métro. Je suis un site marchand développé en Java, hébergé avec un petit Tomcat et une base MySQL. Je suis un vendeur de ski. Donc durant le mois de juillet, je n'ai vraiment pas besoin de beaucoup de CPU et de mémoire, puisque mes ventes seront basses. Au contraire, entre novembre et décembre il faut que mon site marchand fonctionne au mieux, et que même idéalement il augmente tout seule ses capacités en terme de CPU et de mémoire, si mon site est surchargé. Je ne veux pas louper de ventes, et j'ai donc besoin d'un système assez intelligent pour démarrer ou arrêter au besoin des machines. Elastic Grid permettra de piloter automatiquement le démarrage/arrêt de machines et le déploiement de vos applications/services pour s'adapter à la charge et donc, d'effectuer une montée en charge pour vous. Classe non ?

Jérôme a montré une application de découpage de séquences vidéos. Il travaille pour la chaîne de télévision Vivolta. Le découpage et la transformation d'un fichier MP4 en FLV pour le web demande pas mal de puissance machine, surtout du CPU. Grâce à Elastic Grid, lorsque beaucoup de demandes de découpages de vidéos arrivent, nous avons vu sur Amazon EC2 que les serveurs démarraient automatiquement pour effectuer le traitement. Assez génial... Et il va falloir que j'aille à Bry-sur-marne en parler un peu... (secret dont je parlerai dans quelques jours)...

Jérôme est allé à JavaOne 2008 cet année afin de présenter le projet Elastic Grid à la communauté Java. Vous pouvez retrouver sa présentation à [cette URL](#).

Voilà c'était une journée assez dingue. D'abord durant la présentation GWT, la présentation de GXT qui tombe pile poil au moment où je retravaille l'interface d'un logiciel. Ensuite la rencontre avec Jérôme et son projet Elastic Grid... Il y a une étoile qui surveille ce que je fais ou quoi ? (secret 2 : revenez lire ce post après la sortie du post que je publierai plus tard...)

## Un nouveau départ

Voilà un post que j'ai commencé à écrire le 15 juin. Le jour où avec une belle lettre je suis allé annoncer mon départ de Reuters aux Ressources Humaines.

Voici les différents styles pour annoncer donc ma démission:

- o le style avec images d'archives en noir et blanc: *Arrivé en février 2003, Nicolas a tout d'abord commencé à Palo-Alto...bla bla blaaa gnaaa gnaaa..*
- o le style presse people : *"...Reuters : pourquoi elle et moi c'est fini"*
- o le style geek : *"ps aux | grep "motivation" -exec \$! wc -l" return 0*
- o le style intégration continue: *"résultat du sprint 4 : départ"*
- o le style Microsoft : *"votre formation Scrum a provoqué une erreur fatale dans le système. Vous devez quitter l'application. Souhaitez-vous envoyer votre démission ? oui - non"*
- o le style annonce officielle *"Nicolas a annoncé son départ pour créer sa propre société. Nous le remercions pour le boulot sur le produit XXX et son implication."*
- o le style en anglais : *"I announced this morning to the team that I had resigned. I sent my resignation letter yesterday."*
- o le style radio moquette *"...en fait il va à la GrosseBanque car on lui offre 40K de plus... tu te rends compte ? si c'est vrai c'est machine qui me l'a dit..."*
- o le style de robin, mon fils, 2 ans 1/2 *"... papa nouvo ta vailleu... plu train matinnn..."*
- o le style de ma gardienne *"ah c'est bien msieu martignole, mon fils aussi il fait de l'ordinateur comme vous... c'est bien ça l'interneuteu..."*
- o le style Java : `SoftReference currentjob=new SoftReference(new Job("Reuters"));`

```
currentjob.clear();
```

Le bilan de ces 5 ans et demi : vraiment positif.

J'ai eu l'occasion de travailler sur deux projets complètement différents. Le premier dans une équipe américaine m'a apporté de solides connaissances en architecture service et en traitement des flux de données temps réels. Le second, le framework Karma, m'a permis de piloter une équipe de développeur et de créer un framework java de présentation pour la finance. Deux expériences riches. Je vais regretter mon équipe et les gens que j'ai embauchés et formés.

**Maintenant que vais-je faire ?** Je travaille sur un projet depuis quelques mois. Encore un peu tôt pour en parler mais dès septembre j'envisage de travailler à plein temps sur ce projet, tout en effectuant des missions de consulting. Et c'est là, toi cher lecteur, que tu intervies.

Je te propose le produit suivant : un gars sympathique, anglophone avec des connaissances Java, du monde de la finance et de Scrum. Ce charmant bonhomme a encadré des développeurs. Il connaît aussi très bien les outils de production pour faciliter la vie de tes développeurs comme maven, subversion, cruisecontrol ou hudson. Par ailleurs tout fraîchement estampillé ScrumMaster, il peut aussi venir t'aider, toi client, pour mettre en place Scrum. Si tu le mets sur une application J2EE qui a quelques soucis de performances, il peut te régler tes soucis Unix car coup de bol, il est aussi pas mal à l'aise devant un xterm (utilisateur chevronné de linux) ou un JProfiler pour chercher des problèmes de performances. Pour terminer, et c'est offert par la maison, si tu souhaites avoir un petit avis éclairé sur tout ce qui est RIA (GWT, Flex, Silverlight...) tu es devant la bonne boutique...

Je termine le 31 juillet, puis ensuite c'est le grand saut vers une nouvelle aventure.

## Ext-GWT ou GWT-Ext ?

**Après avoir présenté Google Web Toolkit, la librairie Ext-GWT (ou GXT) et la librairie GWT-Ext, je souhaite vous parler des essais que j'ai fait cette semaine. Le choix n'est pas si simple...**

La semaine dernière, à l'occasion de [la journée Client Riche organisée par Valtech](#), je vous ai reparlé un petit peu de Google Web Toolkit. En quelques mots, GWT permet d'écrire en Java du code qui sera transformé en Javascript pour fonctionner du côté du navigateur du client. L'intérêt de GWT est son mode hébergé. Lorsque vous développez, ce qui est bluffant c'est la rapidité à laquelle vous pouvez construire votre interface. En effet, nul besoin de compiler quoique ce soit. Vous éditez simplement votre code Java, vous sauvez votre fichier, un rechargement dans l'outil GWT et votre modification apparaît immédiatement dans un navigateur utilisant la DLL d'internet explorer.

GWT ne dispose pas d'une librairie de composants très riche. Il existe dans le monde Javascript des librairies très puissantes. La librairie [Ext-JS](#) est l'une de mes favorites. Le problème de cette librairie est qu'il faut une bonne maîtrise du langage Javascript et pas mal d'énergie pour développer une interface complète. Croyez-moi, je sais de quoi je parle.

Pour rejoindre le meilleur de ces 2 mondes, [Sanji Jivan](#) a développé GWT-Ext, une librairie qui se repose sur GWT. Ext-JS 2.0.2 ainsi que quelques composants supplémentaires comme les courbes en 2D (YUI) et le module Pagebus destiné à s'interfacer avec Tibco. Cette librairie est distribuée sous licence LGPL v3.

Je vous conseille de regarder [la page d'exemple](#) pour vous faire une idée de ce qu'il est possible d'écrire en Java, et qui se transformera en Javascript au final pour le client.

Le souci et le débat est que Jack Solum, qui travaille sur la librairie Ext-JS a effectué un changement de licence en avril dernier. La librairie Javascript est passée d'une licence LGPL à une licence GPL v3. La conséquence pour les logiciels développés sur cette librairie, est qu'ils doivent soit adopter une licence virale comme la GPL, soit acheter une licence commerciale à l'entreprise de Jack. Cela a provoqué pas mal de remous dans la communauté des utilisateurs de la librairie Ext-JS. A priori les choses sont entrain de changer et on parle de mettre en place des exceptions pour les produits open-source quant à l'utilisation de la librairie Ext-JS. Voir [ce post sur Ajaxian](#). En tout les cas Jack Solum s'est fait largement critiquer pour le passage d'une licence LGPL à une licence GPLv3, [comme il l'explique sur son blog](#) (note: il y a un post de Ludovic Dubost, OSSGTP, XWiki dans les comments). Quelque part c'est très informatif sur le côté épidermique des licences et de ce que les gens en font... On entre dans le débat LGPL/GPL...

Toujours est-il que le problème pour GWT-Ext est que ce projet reste à la version 2.0.2 de la librairie Ext-JS et ne pourra pas passer à la version 2.1 d'Ext-JS distribué en GPL. Dans le même temps, le groupe Ext-JS a lancé sa propre version du portage vers GWT de la librairie Ext-JS (honnêtement si vous suivez et vous êtes encore entrain de me lire, chapeau...). Actuellement la couverture fonctionnelle n'est pas encore très complète. Voici les deux liens pour comparer la première librairie (GWT-Ext) et donc la deuxième que j'appelle GXT mais dont le nom officiel est Ext-GWT... Comment embrouiller le développeur lambda...

- o Démo GWT-Ext <http://www.gwt-ext.com/demo/>
- o Démo Ext-GWT dit "GXT" <http://extjs.com/products/gxt/>

Je travaille sur un projet pour le mois de septembre pour lequel, après avoir fait un prototype avec JBoss Seam et IceFaces, j'ai eu envie de tester un rendu basé sur GWT. J'ai donc développé 2 prototypes avec chacune de ces librairies (GWT-Ext et GXT) afin de me faire une idée précise des lacunes et des points forts de chacune d'elle.

La librairie GWT-Ext est sortie en juillet 2007. La documentation est assez complète et la prise en

main assez rapide. Complètement basée sur Ext-JS, on retrouve donc la manière de programmer cette librairie. Simplement, le fait de travailler en Java me permet de travailler plus rapidement qu'en pur Javascript. De plus, Sanji le développeur derrière cette librairie a intégré aussi des composants de Charts (de YUI) ce qui m'évite d'utiliser une librairie propriétaire.

La librairie GXT est encore en phase de bêta. La documentation et les exemples sont moins fournis. Elle apporte un modèle léger événementielle qui fait que le code écrit me semble plus clair. La javadoc est plus complète, et lors des phases de debug, une partie de cette librairie étant écrit en Java, il est plus facile de travailler qu'avec GWT-Ext.

GWT-Ext est un wrapper complet qui utilise JSNI pour effectuer des appels Javascripts vers la librairie Ext-JS directement. Au contraire, GXT dispose de sa propre implémentation en Java. Au final c'est bien du javascript qui est généré.

Le post suivant de Didier Girard propose un petit tableau pour comparer les 2 librairies.  
<http://www.ongwt.com/post/2008/04/22/MyGWT-is-dead-Long-live-Ext-GWT>

Au final je vais m'arrêter là... J'ai un prototype développé avec les 2 librairies. Je me suis cassé la tête sur GWT-Ext en particulier lors de la mise en place de la partie dynamique. Ext-GWT par contre me permet d'écrire du code avec une architecture plus propre. Donc c'est fait.

Fin du débat

## Prochaine soirée du PJUG sur Adobe Flex

Mardi 8 juillet la prochaine présentation au Paris Java User Group sera sur Adobe Flex. A quelques pas de Denfert-Rochereau à Paris à la FIAP. Ce sera l'occasion de voir deux présentations sur MDA et sur Adobe Flex organisées par l'équipe d'Adobe France (dont François le Droff). Le mois dernier nous étions presque 150 personnes, je vous recommande [de vous inscrire](#) pour avoir une place La présentation sur Flex sera animée par James Ward. Sur [son blog](#) vous pouvez retrouver quelques articles sympathiques dont un proto Google Apps, Adobe Air développé avec Dick Wall, l'un des orateurs du podcast [Java Posse](#). Avec quelques lignes en python pour faire la partie backend serveur, et quelques lignes de MXML pour charger l'objet via RemoteObject. Elegant et simple. J'en profite pour dire un mot sur une application qui permet de traiter un grand volume de données, et qui m'intéresse particulièrement : [NASDAQ Replay](#) développé avec Flex. C'est [François](#) qui m'en a parlé lors de la réunion OSSGTP, et j'ai bien fait d'aller y jeter un œil. Cette application légère fonctionne avec Adobe AIR sur votre machine et vous permet de rejouer les évolutions des cotations d'actions sur le NASDAQ. Pour cela l'ensemble des Quotes ont été stockées sur Amazon S3, un énorme volume de stockage virtuel proposé par la société Amazon. Les milliers d'enregistrement sont stockés sous la forme de simples fichiers textes afin que l'accès aux données soit très rapide. L'application permet à un Trader de rejouer des prises de position sur le NASDAQ et de valider par exemple, des moteurs d'évaluation de Risk. Ce prototype montre à mon avis le futur des applications d'analyse en ligne. J'imagine des cubes OLAP stockés de manière organisée sur S3 avec un frontal développé en Flex... Il y a des millions d'euros sur ce marché dans la finance pour qui veut bien se pencher sur la question... Avis aux amateurs, l'analyse du Risk est un marché porteur dans la finance. Thomson Reuters avait racheté une entreprise appelée Applications Networks en juin 2006 pour récupérer un produit appelé JRisk. Ce produit à l'époque était un moteur d'analyse de risque avec une partie OLAP. Le montant du deal publié [sur l'annonce presse](#) était de 42 millions de dollar. Pour information, le secteur du Trade and Risk Management chez Thomson Reuters représente un chiffre d'affaire supérieur à 100 millions de livres sterling... de quoi payer une petite licence Adobe Flex Builder non ? Pour mon "projet-de-dans-deux-ans" que je commence en septembre si tout va bien, je crois que je vais reprendre cette idée pour stocker les données que je collecte avec mon moteur... Je crois bien qu'Amazon S3 vient de résoudre mon problème de volumétrie des données. Allez on se voit la semaine prochaine, passez une bonne semaine.

## Scrum, début du 6ème sprint

Là, tout en trempant une tartine de pain dans votre café, tout en lisant votre Touilleur Express, vous vous dites : "*le coco ne parle plus de Scrum sur son blog... c'est bizarre...*". Et bien pourtant Scrum continue à faire partie de la vie de notre projet. Voici donc quelques nouvelles du front. Nous avons terminé le 5ème sprint la semaine dernière. Il y a eu un couac. 2 tâches sur le Sprint backlog ont été annulées. En commençant à travailler sur ces 2 développements, l'équipe a identifié des lacunes dans la demande du client... Bref nous nous sommes retrouvés avec un trou d'air en milieu de Sprint. Ensuite que se passe-t-il si le Scrum Master ne fait pas son boulot en réunissant l'équipe chaque matin ? Et bien on ne fait plus de Scrum. Or mes collaborateurs/collègues/amis ont vite fait de lever le pied et de ne pas faire le scrum meeting sans moi. J'en suis grandement responsable, en étant complètement parti et occupé par le projet sur lequel je travaille pour ma vie "après Reuters" ([plus que 5 semaines](#)). Il y a 2 semaines nous avons repris contact avec la Terre... pardon... avec l'équipe du business qui décide en amont du contenu des produits chez nous. Cela nous a relancé sur les rails et j'ai ainsi présenté Scrum à notre nouveau chef d'équipe ainsi qu'au responsable produit. Nous avons construit ensemble un nouveau Product Backlog avec des items importants pour une démo en septembre. J'ai aidé le responsable produit pour qu'il prenne le rôle du Product Owner sans trop parler de Scrum. J'ai essayé de proposer ce processus plus simplement qu'en attaquant avec mes slides PowerPoint et mon bâton d'évangéliste. Et pourtant croyez moi j'arrive très bien à faire le pasteur évangéliste à propos de Scrum... Cela m'amuse pas mal. Hier nous avons tenu une nouvelle réunion de planification pendant 2H30 avec l'équipe (les 3 karma-boys) ainsi que l'ergonome, le chef de l'équipe et le responsable produit. Nous avons discuté et ensuite effectué les cotations à l'aide [du Planning Poker](#), qui a toujours son petit effet. J'ai eu le

sentiment que le chef de produit était surpris par Scrum. Oui forcément ça fait un peu bizarre. J'ai bien aimé le petit moment où mon équipe discutait sur le nombre d'items du product backlog sur lequel ils allaient s'engager. Le product owner essayait de charger un peu plus le sprint log mais il a compris qu'il n'était qu'un spectateur. Nous commençons donc ce matin un nouveau sprint de 10 jours. La vélocité moyenne est de 98 scrums units pour mon équipe. En tenant compte des jours de congés des 10 prochains jours, l'équipe dispose d'un potentiel de 110 scrum units, et ils se sont engagés sur 120 unités. Je vous raconterai ici les 5 premiers sprints rapidement dans un prochain billet. 5 sprints c'est 10 semaines de Scrum... pas mal non ? Concernant mon poste, je devrai avoir un remplaçant d'ici une semaine et je compte bien utiliser Scrum pour effectuer la transition durant les dernières semaines qu'il me reste à faire chez Thomson-Reuters-Effix à Puteaux, France, Terre... Après une grosse nouvelle aventure m'attend... Je ne sais pas encore si cela va fonctionner mais vous pouvez retremper votre tartine dans le café et tourner la page suivante, je continuerai à vous raconter ici "le projet de dans 2 ans qui commence en septembre" avec Scrum bien entendu .

## Rich Internet Application ou pourquoi on vous parle de ça

Que ce soit Adobe Flex, Google Web Toolkit ou Microsoft SilverLight, ces technologies pour client riche vont bouleverser le métier cette année. Même si la prise en main de GWT est relativement facile, le souci est que l'on se retrouve à coder en Java ce que finalement nous pourrions faire en client Web classique. Même mieux, comme me disait Nicolas Romanetti de [Jaxio](#) : *Pourquoi ne pas faire cela en Java Web Start ou avec des Applets au lieu d'utiliser GWT ?* Vous pouvez développer aujourd'hui 2 types d'applications. Un site web classique pour des utilisateurs tout public, pour un extranet, pour un intranet. Et vous pouvez aussi développer des logiciels complets basés sur ces technologies, ce qui est plus mon domaine. Entre la page web classique, la gestion du bouton back, des onglets du navigateurs, et une application riche embarquée dans un navigateur, il y a un monde. Dans le monde de la finance, les contraintes en terme de temps réel sont importantes. La volumétrie des données est aussi un critère important. Il faut limiter et bien réfléchir au découpage en couche de toute l'architecture. En terme d'ergonomie, les traders doivent être les homo sapiens les plus exigeants en terme de réactivité, d'utilisation du clavier et de "...ça doit marcher ton machin...". Après avoir travaillé avec GWT, un prototype .NET et Adobe Flex, je mettrai mon argent sur Flex. La technologie simple et efficace proposée pour l'échange de messages entre le serveur et le client Flex m'a complètement séduit. Si vous téléchargez BlazeDS et que justement, vous jetez un œil sur la démo "trader", vous verrez qu'il est tout à fait possible d'avoir du pseudo-temps réel sans trop d'efforts sur un poste client.



**BLAZEDS** [BlazeDS](#) est un composant serveur qui fonctionne dans un conteneur de servlets et qui permet de pousser vers un client Adobe Flex ou Adobe AIR des flux de données ou des objets sérialisés. Ce serveur est open-source et distribué en licence [LGPL v3](#). J'ai fait quelques tests avec Flex et un chargement d'un flux HTTP avec un tag mx:HTTPService et un tag mx:RemoteObject afin de comparer les performances avec [Firebugs](#). Le chargement via RemoteObject est 2 fois plus rapide sur la démo "testdrive-remoteobject" par rapport à la démo "testdrive-httpservice" distribué dans la version Turnkey de BlazeDS. Vous allez m'entendre parler Adobe Flex pas mal ici dans les semaines qui viennent. J'ai fait aussi un proto avec Spring 2.5 en prenant Spring MVC, une factory particulière pour Flex et IDEA IntelliJ. Il faudrait faire un petit screencast pour vous montrer tout cela car c'est long à raconter par écrit... **Lien supplémentaire:** <http://flexbox.mrinalwadhwa.com/>

## Présentation de SonarJ

Petit-déjeuner jeudi dernier avec la présentation de [SonarJ](#) par Frédéric Brachfeld de [PcMetric](#). Je vous propose de vous parler tout d'abord des besoins et du contexte d'utilisation de SonarJ avant de vous expliquer son utilisation et sa puissance. Je vous propose de ne pas vous faire une brochure commerciale mais d'essayer de vous donner un nouvel angle sur SonarJ En une phrase, SonarJ est un outil qui permet de vérifier en temps réel qu'un ensemble de classe Java respecte une architecture au moment où l'on saisie une ligne de code. Après avoir décrit votre architecture par tranche et par couche dans SonarJ, un plugin permet dans Eclipse de rendre votre code "intelligent" afin qu'il respecte l'architecture de votre projet. **Le besoin et le constat** En Java, le nombre d'objet et la structure évolue de plus en plus au fur et à mesure des besoins. Le constat est que les coûts de maintenance, sans outils adaptés, vont en augmentant de manière exponentielle. Frédéric Brachfeld pendant la présentation utilise le terme d'érosion d'architecture, que je trouve très juste pour expliquer qu'un projet s'use. Dans un premier temps, les changements d'équipe font que l'information d'architecture tend à disparaître. On ne compte plus les projets qu'il faudrait reprendre à zéro après quelques années. La documentation de l'architecture n'empêche pas l'explosion des coûts de maintenance. Et les demandes de modification vont coûter de plus en plus cher. En fait, et je reviendrai là dessus tout à l'heure, au moment même où le développeur tape sa ligne de code, finalement rien ne l'empêche d'importer une classe Java d'un autre module de n'importe quelle partie de son application. Or le souci est là. Sans architecture et découpage, le coût de maintenant d'une application explose. Grâce aux patterns, les développeurs Java sont capables de concevoir du code architecturé afin de séparer les couches présentation, métier et accès aux données. Encore une fois cependant, il n'y a pas de documentation des relations inter-packages dans un projet Java. En clair, il est facile de transformer le plus beau code en une soupe difficile à maintenir. Le constat est

donc qu'il faut un cadre pour documenter l'architecture, s'assurer que les développeurs respectent l'architecture du logiciel et donc réaliser des économies sur le long terme. **Qu'est-ce que SonarJ ?** SonarJ est un produit développé par la société [Hello2morrow](#) en Allemagne. L'outil est utilisé par exemple pour le développement de Spring MVC pour en renforcer la qualité. Il s'agit donc d'un outil de rétro-ingénierie qui travaille au niveau du bytecode. SonarJ se présente sous la forme d'une application Java développée avec SWT. Un plugin très pratique permet ensuite d'ajouter les fonctionnalités de SonarJ directement dans Eclipse. Après avoir lancé l'application, je crée un premier projet. Je décide d'importer le code de notre framework, Karma. 1800 classes Java environ et une dizaine de modules différents. SonarJ vous propose de découper verticalement votre application par tranche (slice en anglais). Par exemple une partie "Sécurité", une partie "Rapports", une partie "Administration". Ensuite vous devez découper par couche applicative votre application. Par exemple "web layer", "ejb layer" ou "data access layer". Vous avez saisi l'idée. Ce découpage horizontal et vertical constitue une matrice. A vous ensuite de définir les relations entre les différentes couches ou les différentes tranches. SonarJ peut alors vous donner des détails sur les violations d'architecture et les problèmes rencontrés. **Notification en temps réel** SonarJ m'a réellement convaincu lorsqu'au moment de taper un peu de code dans Eclipse, j'ai vu une erreur contextuelle apparaître dans Eclipse me signalant que la couche DAO ne devait pas être accédée directement de la Vue. Clairement, SonarJ rend votre code intelligent au moment même où vous tapez une ligne de code. **Une critique ?** Tout n'est pas rose et il faut aussi que je vous donne mon sentiment sur les points à améliorer. Tout d'abord dans la vue slice ou layer, le moteur de positionnement des boîtes n'est pas très facile à utiliser. Il faut s'y prendre à plusieurs fois pour connecter ses boîtes. Il serait souhaitable d'avoir un système de rangement de ces boîtes UML-like afin de faciliter notre travail. Le modèle matricielle ne remplace pas à mon avis une grosse réflexion sur le découpage de son application. Avec 5 tranches verticales et 4 couches horizontales, cela fait autant de case où ranger vos packages. Et cette partie prend du temps. Ce n'est pas une critique, simplement un bémol essayant de dire qu'il faut prévoir du temps pour cette partie qui est très importante avant de commencer à utiliser le plugin dans Eclipse. **Un très bon point: le suivi des corrections** Avoir un outil qui vous signale 46 erreurs dans votre application c'est une chose. Mais là où SonarJ m'a aussi séduit c'est la capacité à créer des mini-tâches afin de les affecter à un développeur. Il est possible de s'intégrer à un outil comme Jira afin de créer des artefact directement à partir de SonarJ. Là encore on voit que le logiciel est clairement orienté industrialisation. Le suivi des corrections est une chose importante pour l'architecte ou le chef de projet. **La revue de code ne sert à rien** J'enfonce une porte ouverte mais c'est dans l'air du temps : la revue de code à posteriori pour vérifier la qualité du code. C'est une chimère qui repose sur le fantasme que deux personnes seraient plus performantes qu'une seule pour relire du code. En effet cela part d'une idée louable mais mieux vaut travailler en amont en programmant en binôme que de faire intervenir une personne pour relire son code. SonarJ remplace je pense le suivi nécessaire d'un architecte ou d'un coordinateur qui s'assure que les personnes n'ajoutent pas d'importation de la vue vers la couche d'accès aux données par exemple. Un très bon point pour SonarJ. **Etablir des règles dans un document Word est complètement inutile** Pourquoi ne pas établir des règles de codage dans un document ou sur un wiki ? Nous pourrions y décrire avec beaucoup de détails qu'une classe ne doit pas faire plus de 300 lignes, qu'il est interdit d'utiliser plus de 15 méthodes publiques. Qu'il faut préférer la délégation ou l'injection de contrôle... et bla bla bla et blaaa blaa bla... En fait l'idée c'est de mettre dans un document une forme d'intelligence en espérant que le développeur respecte à la lettre ces consignes... Je crois que c'est aussi stupide que de donner un gilet de sauvetage dans un 747 qui ne servirait pas à grand chose en cas de crash. Une fois que l'avion s'est écrasé, vous aurez fier allure en disant "*et pourtant je vous avais donné les consignes de sécurité...*" SonarJ embarque des outils d'analyse de code et de définitions de métrique. J'ai cru comprendre qu'à la manière d'un Checkstyle ou FindBugs, SonarJ s'intègre à Ant ou Maven afin de générer des rapports au format HTML sur la qualité du code. Et ce, en temps réel. C'est là qu'il faut définir les règles et c'est là qu'il faut les implémenter. Pour moi un code pourri doit littéralement faire exploser une intégration continue. Il ne faut surtout pas fantasmer sur un document de "règles d'architecture". Cela dit si vous souhaitez vous lancer dans l'écriture d'un tel document, je cherche quelque chose pour allumer le barbecue cet été. **En conclusion** Objectivement, lorsque vous souhaitez changer l'architecture ou auditer du code, il existe 3 façons différentes de le réaliser. En premier vous faites appel à une équipe en interne, avec un œil neuf sur la question. A mon avis la plus mauvaise idée qu'il soit. En second, vous faites appel à une SSII. Si celle-ci n'est pas outillée avec SonarJ je pense que vous allez régler une facture douloureuse. De plus, une fois le contrat terminé, qui continuera à vous garantir de la qualité du code ? En troisième solution je pense qu'en effet SonarJ apporte des barrières de qualité qui rende le code plus robuste, plus facile à maintenir et que donc nous pouvons réaliser des économies sur le long terme. Il est possible de télécharger une version d'évaluation limitée à 30 jours pour vous faire une idée de SonarJ. **Autres articles:** Article de Erwan Alliaume de Xebia : <http://blog.xebia.fr/2008/06/25/sonarj-comment-gerez-vous-votre-architecture-et-votre-qualite-technique/>

## Petite révélation sur l'iPhone 3G

Je ne vous cache pas qu'il y a de fortes chances pour que je fasse l'acquisition d'un iPhone 3G dès la fin du mois de juillet. Il faut attendre le 17 juillet pour enfin avoir accès au Saint Graal de la télécommunication... On en est même venu avec un ami à créer une association de "*ceux-qui-ne-craquent-pas-pour-l-iphone*" mais je pense que notre groupe va exploser dès le 17 juillet... Petite révélation donc pour ceux qui s'intéressent au sujet : l'iPhone 3G ne fait pas que de la 3G. Il marchera aussi très bien avec la 3G+. Quelle différence me demandez-vous ? La 3G+ est une évolution logicielle de la 3G (UMTS) et permet d'atteindre des débits descendants jusqu'à 10 fois plus rapide que la 3G. Les 3 opérateurs ont déployé en France un réseau compatible 3G+, Bouygues Telecom l'ayant fait discrètement [en mai 2007](#). Or nos voisins de l'autre côté de l'Atlantique découvrant à peine la 3G, nous n'allions pas en plus leur faire l'affront de leur dire gentiment que l'iPhone tournera entre 2 et 10 fois plus rapidement dans notre bon vieux pays non ? (source : Europe 1, Benjamin Vincent) Surfer en 3G+ (HSDPA) c'est se balader sur internet à un

débit entre 1.1 Mbits/seconde et 14.4 Mbits/seconde. En moyenne 7.2 Mbits/seconde... De quoi visualiser une belle vidéo non ? Là où je pense qu'il y a un souci, c'est que l'accès à l'iTune Store ne pourra s'effectuer que via le Wifi, lorsque votre iPhone sera à proximité d'une borne Wifi. Dommage... Enfin pour terminer, notez que votre iPhone 3G peut aussi vous servir de modem d'appoint pour que votre MacBook Pro se connecte à internet. Donc oui, vous pourrez surfer cet été de la plage pendant que Madame se baigne. Et vous pourrez relever vos emails le soir de chez vos beaux-parents qui n'ont même pas Internet (oui ça arrive...). Cela dit par défaut, le Bluetooth de l'iPhone 3G semble bridé pour n'utiliser que certains accessoires comme des casques sans fil. Donc à priori il sera impossible d'utiliser l'iPhone en mode bluetooth avec le profil modem et un ordinateur portable, via Bluetooth. Mais bon, ce n'est qu'[une limitation logicielle](#)... PS: pour les lecteurs assidus, j'ai retiré l'article sur flex/blazeds ce midi car il y a quelques grosses coquilles dans la partie "code". Désolé !!! je le publierai ce soir à nouveau.

## Flex, BlazeDS, Spring, Hibernate all together with Maven

Note: first post ever in english but I really thought it would matter.

For those who wants to play with Flex, BlazeDS, Spring, Hibernate using a set of Maven projet, François le Droff wrote a good article. He managed to set-up a nice environment that was really helpfull for me when I started to add maven support to my own project.

Read [http://blogs.adobe.com/francoisledroff/2008/05/my\\_blazeds\\_xdoclet\\_spring\\_hibe.html](http://blogs.adobe.com/francoisledroff/2008/05/my_blazeds_xdoclet_spring_hibe.html)

However my project use Charts and when I tried to add support for Charts, it doesn't compile anymore with Maven. I managed to fix various issues and I will report there how to make it work.

### Add new mirror to your maven configuration

First, as explained on its post, add its repository to your maven configuration file  
Here is my settings.xml :

```
<settings>
  <localRepository>/Users/nicolasmartignole/.m2/repository</localRepository>
  <offline>>false</offline>

  <proxies>
  </proxies>

  <servers>
  </servers>

  <mirrors>
    <mirror>
      <id>ibiblio</id>
      <mirrorOf>central</mirrorOf>
      <name>Ibiblio central repos.</name>
      <url>http://www.ibiblio.org/maven2</url>
    </mirror>
  </mirrors>

  <profiles>
    <profile>
      <id>defaultProfile</id>
      <activation>
        <activeByDefault>true</activeByDefault>
      </activation>

      <repositories>
        <repository>
          <id>fna-repository</id>
          <name>fna-repository</name>
          <url>http://fna.googlecode.com/svn/trunk/fna/fna_m2_repository/</url>
          <releases>
            <enabled>true</enabled>
          </releases>
          <snapshots>
            <enabled>>false</enabled>
          </snapshots>
        </repository>
      </repositories>

      <pluginRepositories>
        <pluginRepository>
          <id>fna-repository</id>
          <name>fna-repository</name>
          <url>http://fna.googlecode.com/svn/trunk/fna/fna_m2_repository/</url>
          <releases>
            <enabled>true</enabled>
          </releases>
          <snapshots>
            <enabled>>false</enabled>
          </snapshots>
        </pluginRepository>
      </pluginRepositories>
    </profile>
  </profiles>
</settings>
```

## Create a simple project from scratch

Now it's time to create your first maven project with Flex. I will call my new project "Lombok" and it will be part of "Treck" which is my company name.

```
mvn archetype:create
-DarchetypeGroupId=com.droff
-DarchetypeArtifactId=blazeds-xdoclet-spring-hibernate-archetype
-DarchetypeVersion=1.0
-DgroupId=org.treck
-DartifactId=lombok
```

## Create a new IDEA IntelliJ project

Go to the new created folder (cd lombok) and type-in

```
mvn idea:idea -DjdkName=1.5
```

This will create a new IntelliJ Project.  
You can also create a new Eclipse project with :

```
mvn eclipse:eclipse
```

## Compile time !

It's now time to build the first project. There's nothing special here since François's project come with a nice CRUD application that works with HSQL database and Hibernate.

Simply type "mvn install" and wait for compilation

## Deploy time

Once your project has been compiled you can now start the local Jetty servlet container. BlazeDS has already been configured for you with a simple amf service.

Enter

```
cd java_webapp
mvn jetty:run-exploded
```

If everything went well you should see something similar to the following output :

```
...
6 jul. 2008 09:34:50 org.springframework.orm.hibernate3.HibernateTransactionManager afterPropertiesSet
INFO: Using DataSource [org.springframework.jdbc.datasource.DriverManagerDataSource@bc5edf] of Hibernate SessionFactory for HibernateTrans
6 jul. 2008 09:34:50 org.springframework.web.context.ContextLoader initWebApplicationContext
INFO: Root WebApplicationContext: initialization completed in 1465 ms
2008-07-06 09:34:50.252::INFO: Started SelectChannelConnector@0.0.0.0:8080
[INFO] Started Jetty Server
[INFO] Starting scanner at interval of 10 seconds.
```

Open a web browser and check <http://localhost:8080/lombok>

## Time to add some more difficulties

So far, so good. Now when I tried to import my existing Flex application into this sandbox, I started to encounter some difficulties. I will explain here how I manage to fix them.

First, let's edit the simple MXML application and add a new Chart component. I let you create a working MXML from Flex Builder. My sample is based on a dataProvider that is an ArrayCollection of values.

```
...
<mx:LineChart id="linechart1" width="617" height="210" dataProvider="{todoItems}" showDataTips="true">
  <mx:series>
    <mx:LineSeries yField="cpuIdle" form="curve" displayName="Idle"/>
  </mx:series>
  <mx:horizontalAxis>
    <mx:CategoryAxis categoryField="name"/>
  </mx:horizontalAxis>
</mx:LineChart>
```

Now if you try to compile with mvn install the project you should see this error

```
...
[ERROR] Could not resolve <mx:LineChart> to a component implementation.
...
BUILD FAILURE
```

## Add 2 dependencies to your pom.xml

Edit `flex_app/pom.xml` and add the following dependencies

```
<dependency>
  <groupId>com.adobe.flex.sdk</groupId>
  <artifactId>datavisualization</artifactId>
  <type>swc</type>
```

```

    <scope>merged</scope>
    <version>3.0.0.477</version>
  </dependency>
  <!-- Add this to avoid the Unable to resolve resource bundle charts for locale en_US error //-->
  <dependency>
    <groupId>com.adobe.flex.sdk</groupId>
    <artifactId>datavisualization</artifactId>
    <version>3.0.0.477</version>
    <type>resource-bundle</type>
    <classifier>en_US</classifier>
  </dependency>

```

Update also the rvin.mojo plugin version to read (still editing flex\_app/pom.xml)

```

<plugins>
  <plugin>
    <groupId>info.rvin.mojo</groupId>
    <artifactId>flex-compiler-mojo</artifactId>
    <!-- Update here to the correct version -->
    <!-- <version>1.0-beta4</version> -->
    <extensions>true</extensions>
  </plugin>

```

At the end of the file you can also add a new repository (or edit your settings.xml)

```

<repositories>
  <repository>
    <id>flex-mojos-repository</id>
    <url>http://svn.sonatype.org/flexmojos/repository/</url>
    <releases>
      <enabled>true</enabled>
    </releases>
  </repository>
</repositories>

```

### Update your local repository

To fix this issue you need to manually import a SWC component and a resource bundle into your local maven repository. You also need to update the pom.xml so that the flex compiler add at compile time in the external-libs the chart component.

First, I read that you need to import the SWC component that are bundled with Flex Builder. Don't know exactly why but it doesn't work if you import datavisualisation component from an external flex SDK 3.0. So here is how to import the Chart component into maven :

```

mvn install:install-file -DgroupId=com.adobe.flex.sdk
-DartifactId=datavisualization
-Dversion=3.0.0.477
-Dpackaging=swc
-Dfile="/Applications/Adobe Flex Builder 3/sdks/3.0.0/frameworks/libs/datavisualization.swc"

```

Then you need to import the resource bundle

```

mvn install:install-file -DgroupId=com.adobe.flex.sdk
-DartifactId=datavisualization
-Dversion=3.0.0.477 -Dclassifier=en_US
-Dpackaging=swc
-Dfile="/Applications/Adobe Flex Builder 3/sdks/3.0.0/frameworks/local/locale/en_US/datavisualization_rb.swc"

```

If you forget to import the resource bundle you'll see this error message:

```
Unable to resolve resource bundle "charts" for locale "en_US".
```

### Rebuild

The application should now build completely with the Chart. It's really important to specify in the pom.xml a scope of "merged" for the datavisualization plugin as shown below:

```

<dependency>
  <groupid>com.adobe.flex.sdk</groupid>
  <artifactid>datavisualization</artifactid>
  <type>swc</type>
  <scope>merged</scope>
  <version>3.0.0.477</version>
</dependency>

```

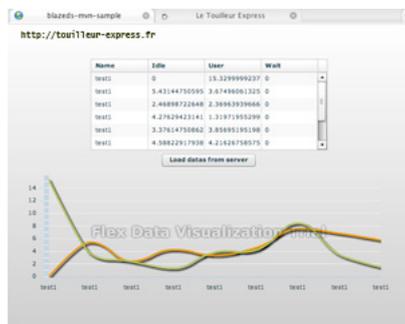
If you don't specify this scope, everything will compile but when you deploy the SWF file on Jetty and press refresh, Flash shows an error message with something like

```
VerifyError: Error #1014: Class mx.charts::LineChart could not be found.
```

The reason is that the datavisualization needs to be included within the SWF and shall not be loaded as an external library.

I hope that it will help some of you if you play with this nice mojo and with Flex Chart

To conclude a screenshot I took this morning of Chart:



## Ce soir c'est petits fours et images en couleur

2 Informations importantes :

Ce soir rendez-vous dans Paris non loin de Denfert-Rochereau pour [la soirée Paris Java User Group](#) (PJUG) consacrée à MDA et Adobe Flex. Pour Flex nous devrions parler ce soir de la partie architecture et des échanges de messages. [James Ward](#) est un speaker très calé sur le sujet.

Cher lecteur comme tu peux le constater les soirées PJUG sont ouvertes car ce soir nous ne devrions pas trop parler Java. Une raison supplémentaire pour venir.

L'accès s'effectue via la station RER B Denfert-Rochefeuu, ensuite passage par le MacDo vers 19h30 et enfin arrivée dans les locaux [de la FIAP](#) où se déroulera l'événement à partir de 20h00. En entrant la dernière fois il fallait prendre un escalier qui descend sur la gauche à côté du bar dans le Hall d'entrée.

J'ai reposté ce matin mon article sur BlazeDS, Maven, Flex pour compléter le sujet de ce soir.

Deuxième information à diffuser: **Google est en mesure d'indexer le contenu texte des fichiers Flash**. Voir les détails sur [ce post](#) sur le blog de Google. Il est donc faux aujourd'hui de dire que Google n'indexe pas les fichiers Flash.

Mercredi je participe à la journée [Tremplin Entreprise](#) sur invitation de [Jaxio](#). Comme d'habitude je vous ferai un petit retour ici sur le blog... Je vais vraiment avoir besoin de [ce stylo Livescribe](#) que Dimitri Baeli m'a montré je crois...

A ce soir pour ceux qui viennent.

J'ai vu 168 inscrits ce matin sur [la page PJUG](#)

## Débriefing comme d'habitude à pas d'heure de la soirée Paris JUG

Débriefing à chaud dans le RER du Paris JUG. J'essaye toujours autant que possible de vous donner à chaud un retour sur la soirée. C'est un peu comme une photo argentique de la soirée... Pas bidouillée, pas retravaillée... On travaille à l'ancienne ici. [Eric Lefevre](#) de Valtech a proposé d'organiser à la rentrée [un Open Space Technology](#). L'idée est de se retrouver afin de discuter entre personnes du Paris JUG sur des sujets technos en suivant un ordre libre. Lisez son article pour comprendre mieux le principe et ensuite vous pouvez en discuter sur la mailing liste du Paris JUG (enfin je crois qu'il y a une mailing list non ?? quelqu'un pour confirmer ?). La soirée était découpée en 2 présentations : tout d'abord MDA puis ensuite Adobe Flex et les échanges client-serveur. MDA a reçu un accueil un peu frais de la part de la centaine de personnes présentes ce soir. Pourtant j'ai trouvé la présentation assez précise et bien menée par Grégory Weinbach [d'Objects Directs](#). Il connaît son sujet. Pour ma part je n'ai pas expérimenté MDA bien que l'équipe installée à côté de notre équipe travaille avec MDA depuis 6 mois. Grégory Weinbach après avoir présenté le principe de MDA explique qu'il faut faire la distinction entre les frameworks MDA et les outils MDA. Il explique que la modélisation apporte de l'Agilité dans la création d'un logiciel. Il est possible de travailler par itération et l'expression des besoins fonctionnels rapproche le développeur du client. Ce client est alors plus satisfait car il est aussi acteur de cette modélisation. Il explique aussi que l'on retrouve des modélisations types qui sont soit trop techniques, soit qui cherchent à modéliser à outrance, soit qui sont "PSM"... Je ne sais pas ce que c'est que PSM, donc je ne pourrai pas vous l'expliquer ici. MDA propose aussi une indépendance technique. On code en quelque sorte au niveau du modèle en modélisant les composants métiers. Par exemple une Liste, une Form, une Vue. Ensuite les interactions et les règles métiers sont ajoutées, ce qui permet d'avoir un programme cohérent. Pour la partie dynamique, l'architecte peut utiliser UML 2 ou un langage abstrait (ASL) pour donner vie à son Modèle. Là dessus il explique que la modélisation uml dynamique n'est pas facile, et que les outils ne contraignent pas assez le développeur. Tout ceci pour rester indépendant du langage cible qui n'est alors pas important. Le modèle exprimé, il est enfin possible de générer du code afin de le compiler... ou pas (VBScript ça se compile ?) Les questions de l'assistance du Paris JUG étaient assez précises. Dont une question monologue tartine et vas-y que je te raconte ma vie très intéressante... Il y a eu des questions sur les outils, et il est vrai qu'en tant que développeurs nous aurions souhaité voir un peu cela de plus près. Un peu d'expérience de l'Architecte nous aurait intéressé. **Bon et ton avis dans tout cela ?** Mes sentiments sur MDA : non MDA n'est pas une méthode Agile. Adepte de Scrum je crois que ce soir c'était un

petit tirage de couverture. Il ne me semble pas que la modélisation apporte une quelconque agilité. MDA n'est pas encore maîtrisé par les développeurs et demande un effort de formation pour fonctionner correctement. Ensuite j'ai eu un drôle d'impression en entendant que MDA c'est "les Architectes" et que la caste des "Développeurs" passe au second plan. Le besoin d'avoir un Analyste, est-il encore d'actualité ? Un Analyste qui ne ferait que de la modélisation avec une compétence financière, ça doit valoir une fortune... Je reste très perplexe sur l'offre "indépendance du langage". Dans le domaine des applications webs, faut-il rappeler la part de marché de Java ? Est-ce que nous avons déjà vu un projet passer d'un langage à un autre ? Je comprends très bien qu'il est important de montrer que l'effort doit se faire au niveau de la modélisation du PIM (Platform Independent Model). Mais où sont les contraintes de mise en production ? Les contraintes d'architecture ? Les tests fonctionnels et unitaires font-ils partie aussi du code généré ? Pour ma part je creuserai le sujet MDA, bien que j'en ai entendu et fait l'expérience sur la modélisation d'instruments financiers l'an passé. A suivre donc... **Flex, la petite claque** Cher lecteur, il y avait une personne dans la salle avec un grand sourire. Cette personne c'était moi. La présentation de Flex nous a montré la puissance et la simplicité des échanges entre le client Flash ou AIR et un serveur distant utilisant BlazeDS. Tout d'abord pour les personnes qui étaient là ce soir, nous avons eu un sacré orateur. Avez-vous la technique de "*je vous tape 5 lignes de MXML de la main droite tout en parlant au micro tenu de la main gauche ?*". James Ward est l'auteur du fameux Census dont je vous ai déjà parlé. Allez on prend sa souris et nous allons tous regarder cette url : <http://www.jamesward.com/census/>. Cette application vous explique les temps de chargement relatif entre votre propre navigateur et son serveur selon la technologie utilisée. Vous pouvez donc comparer du HTML simple, du Dojo, du JSON, du XML chargé avec AJAX ou encore de l'AMF, le format binaire compressé et optimisé de Flex. Allez y faire un tour je vous attends ici pour continuer... James nous a ensuite montré successivement comment lire un flux XML avec l'objet XMLHttpRequest. Ensuite comment charger un objet Flex directement via RemoteObject. Ensuite une démonstration assez bluffante de DataService avec 2 pages webs qui s'échangent des données. Il a montré la partie Producer et Consumer en codant un chat en 10 mn à peine, de la main droite et en parlant au micro je vous le rappelle... Comme je disais à mon voisin : "*c'est moi ou j'ai l'impression que tout est accéléré ?*" Les personnes de l'assemblée ont été franchement séduites. Je crois qu'un bon nombre de développeurs vont essayer eux-même Flex afin de s'en faire une idée encore plus précise. Cela tombe bien cher lecteur, je risque fortement de t'en parler de plus en plus dans les jours qui viennent... Stay tuned ! Demain c'est costard cravate et [journée au Sénat](#). Je vais voir Jaxio. Ca va être fun !

## Les commentaires sur le Touilleur

Un petit mot pour vous informer que j'ai réactivé ce matin l'obligation de confirmer son inscription avant de pouvoir poster un commentaire. Cela en raison des spams que je supprime chaque matin depuis 9 jours... Désolé si vous avez vu des liens vers des sites sans rapport avec le touilleur... Moi ça m'énerve de le découvrir chaque matin. Donc pour poster un commentaire, en principe vous devez vous inscrire, le site confirme votre inscription et ensuite vous pouvez poster. Je ferai quelques tests dans la journée. Merci et désolé de réduire la facilité pour poster les commentaires

## Une journée au Sénat

Mercredi dernier avaient lieu au Sénat les 10 èmes rencontres entrepreneurs-investisseurs, [Tremplin Entreprises](#). Co-organisé par l'ESSEC et le Sénat, c'est le premier événement du capital-risque en France. 30 projets innovants pré-sélectionnés étaient présentés par chacun des entrepreneurs devant des investisseurs et des invités. Depuis 1999 c'est près de 300 projets innovants qui ont ainsi été primés.

Sur invitation de [Jaxio](#) j'ai participé à la matinée du 9 juillet. L'occasion pour moi d'assister en tant que spectateur à la présentation d'une dizaine de projets.

Organisé en 4 catégories (Internet et Services, Logiciels, Matériaux-Composants-Systèmes et Science de la Vie) j'ai préféré suivre les présentations sur "Internet et Services" ainsi que "Logiciels" où Celerio, le logiciel de Jaxio, a été sélectionné parmi les finalistes au concours pour cette catégorie.

**aiderdonner** (<http://www.aiderdonner.com>)

Il s'agit d'un service web pour les associations et les organisations désirant collecter des fonds et faire appel à la générosité publique. Le site vous permet de créer votre page de collecte afin de faire appel à vos amis par exemple.

Le créneau d'aiderdonner est de dématérialiser la collecte de dons et d'accélérer ensuite le versement à des associations. Des sites similaires existent aux USA et rencontrent un grand succès. Je trouve l'idée intéressante et porteuse, la présentation était très sympa.

**CreditDomus** (<http://www.creditdomus.com>)

CreditDomus est un courtier en crédits immobilier sur Internet qui se distingue de ses concurrents ([meilleurtaux.com](#) ou [empruntis.com](#)) grâce à un logiciel novateur d'optimisation de prêt grâce à des calculateurs de lissage automatique. En principe lorsque vous empruntez pour acheter un appartement, vous cumulez plusieurs prêts. Un prêt principal, peut-être un prêt à 0%, un prêt employeur etc. Or l'intérêt entre les prêts aidés (prêt à taux zéro) et le prêt principal est d'effectuer un lissage afin que le montant des mensualités soit constant.

C'était la présentation la plus concrète pour les investisseurs présents, l'équipe semble solide et bien organisée. C'est un marché concurrentiel mais qui reste porteur malgré le ralentissement des offres de crédits au premier semestre 2008.

**Limonetik** (<http://www.limonetik.com>)

Ensuite l'équipe de [Limonetik](#) a présenté un nouveau moyen destiné à rendre facile l'utilisation de moyen non bancaire comme une Carte Lafayette Mariage ou une carte cadeau (Kadéos) sur des sites comme la Fnac par exemple. L'idée est qu'en tant que client, je peux acheter par exemple une télé sur le site de la Fnac en payant une partie avec ma carte bleue, une partie avec ma carte mariage et le solde avec ma carte crédit-révolving. L'équipe est constituée de 2 anciens de PriceMinister. Elle est financé aussi par Pierre KOSCIUSKO-MORIZET le fondateur de PriceMinister.

Présentation sympathique mais un peu courte. Une démo aurait été parfaite pour comprendre comment fonctionne le système.

Je vais essayer de vous l'expliquer ici rapidement :

Le problème pour un site de paiement lorsqu'il doit ajouter le support d'une carte, c'est le temps nécessaire et les modifications à apporter pour modifier le site, mettre en place les scripts sécurisés pour la carte et enfin effectuer un passage en production. L'idée de Limonetik est donc de référencer des sites partenaires sur un site portail, puis d'effectuer ses achats via le portail. Au moment de payer, le site "décoré" par le moteur Limonetik n'est pas modifié. La carte bleue de Limonetik effectue le débit sur le site distant en appelant à votre place le paiement du site marchand. Pour cette raison la technologie n'est pas intrusive et se comporte comme un client lambda qui effectue son achat sur le site de Darty ou de la Fnac.. De votre côté en donnant accès à votre carte Lafayette mariage à Limonetik, celui-ci pourra donc effectuer un transfert de fonds. Tout ceci est rendu possible depuis la loi de mai 2007 qui autorise des tiers à recevoir des fonds et à en envoyer vers d'autres tiers. Auparavant seules les banques et Orange étaient autorisés à conserver des fonds. Dingue...

Bon, le point fort est l'intégration avec les sites existants sans que ceux-ci ne soient impactés, puisque l'on utilise la carte bleue. Le point sur lequel je m'interroge c'est l'obligation de partir d'un portail utilisant Limonetik pour effectuer ses achats. C'est obligatoire pour des raisons techniques, et Limonetik est donc un proxy applicatif entre le site distant et vous. (Si je me trompe, corrigez-moi)

#### **Présentation de rtgi (<http://www.rtgi.fr>)**

rtgi développe des outils spécialisés de collecte et de mesure de l'information sur Internet. Concrètement, Alain LE BERRE nous a montré [une carte en 2D](#) représentant actuellement les liens inter-sites autour de la campagne américaine pour les élections de 2008. Une cartographie permet de comprendre les liens entre par exemple les sites médias d'une part, et les blogs politiques. Une interface de navigation simple permet d'ajouter des critères et de visualiser ensuite les liens inter-sites. Le moteur linkFluence permet donc de modéliser les liens sociaux, de suivre des sujets et d'aider des cabinets de conseils ou de communication pour visualiser les acteurs majeurs sur un thème donné.

Je donnerais cher pour voir une cartographie des sites autour de Java en France, des relations entre blog par exemple.

Allez voir surtout la démonstration <http://presidentialwatch08.com/>

#### **Présentation de Multiposting.fr**

Ensuite la société Multiposting.fr devrait intéresser les recruteurs et les chasseurs de tête. Cette jeune pousse propose un outil permettant de poster une offre d'emploi sur plusieurs sites, puis d'en assurer le suivi et la gestion via un site centralisé. Gain de temps pour les recruteurs, retour sur investissement important. J'ai appris par exemple que le coût d'une annonce sur Monster.fr serait au delà de 750 EUR aujourd'hui. Le marché étant tendu, il n'est pas irréaliste de penser qu'une annonce sur Cadreemploi.fr coûtera dans les 1000 EUR pour 3 semaines d'ici quelques temps, d'après leur témoignage.

Le marché représente 1.5 millions d'annonces d'emploi en France. Le coût moyen d'une annonce est de 400 EUR. Remplir un formulaire sur un site prend pas mal de temps. Cela se sent parfois lorsqu'en lisant les annonces nous sommes un peu frustrés de retrouver la tartine habituelle et rien de novateur qui donne envie d'envoyer son CV.

Des sites similaires aux USA existent et marchent très bien. Le multi-posting d'annonces d'emploi est un marché porteur. Voir le site <http://www.multiposting.fr/> pour plus de détails.

#### **Novapost (<http://www.novapost.com>)**

Novapost propose une nouvelle solution de dématérialisation des documents papiers, développée en partenariat avec IBM. Le principe : vous avez des factures, des documents comptables à conserver. Vous manquez de place dans votre bureau. Novapost permet d'archiver électroniquement vos documents, vous assure un stockage sécurisé et tout ceci via un portail internet. L'application hébergée (Saas / Software as a Service) permet d'accéder à distance à ses documents.

J'ai vu qu'un projet pour les particuliers se prépare : <http://www.sisimple.fr>. Il sera donc possible pour nous, de stocker nos documents afin d'éviter la paperasserie à la maison.

#### **BinarySEC (<http://www.binarysec.com>)**

Certainement le produit qui m'a le plus intéressé. Il s'agit d'un module de sécurisation pour Apache qui permet de se protéger contre le trafic suspect qui entre dans un site. L'intérêt du produit c'est tout d'abord son moteur d'apprentissage et de règles. En effet, nul besoin de se compliquer la vie comme avec mod-security.

L'équipe de BinarySEC est constituée de Richard TOURET et de Michael VERGOZ. Michael est l'un des deux core contributeur français de PHP. Autodidacte et très fort, à 25 ans il a déjà une très grosse expérience derrière lui. Il a bossé chez Intel, il est un contributeur du système de détection d'intrusion [Prelude](#)... Bref vraiment intéressant.

[Son blog](#) est très pointu, j'ai eu quelques frayeurs en lisant les trous béants sur Wordpress (que j'utilise) et je vais m'empresse d'effectuer une mise à jour. Est-ce que cela explique le fait que le Touilleur tombe souvent ? J'ai régulièrement des soucis, et je dois relancer Apache.

En tous les cas, BinarySEC propose une version d'essai limitée à 5000 transactions par jour, que vous pouvez installer sur votre serveur Apache. <http://www.binarysec.com/page-eng-freetrial.html>

## Présentation de Jaxio (<http://www.jaxio.com>)

Je termine par les meilleurs :-)

Nicolas Romanetti a présenté Celerio, dont je vous ai déjà parlé sur le Touilleur lors d'un [interview](#).

Celerio est un outil de génération de code qui part du modèle le plus simple : le modèle relationnel. Il permet de générer le socle technique d'une application web en utilisant les meilleurs patterns et outils open-source. En effet on peut se dire que c'est similaire à AppFuse... Mais non ! Celerio ajoute en effet un moteur breveté de reprise de code. Ce n'est pas un outil que vous lancez une fois et ensuite hop, je commence à coder. C'est un moteur très puissant qui permet de gagner du temps. Et du temps, en SSII, on en a pas tellement.

Son intérêt aussi est de rendre réalisables des projets webs qui n'auraient pas pu être réalisés faute de temps, de ressources ou de connaissances. On est d'accord pour dire que nos clients en veulent toujours plus, pour un prix qui ne doit pas être élevé, et que nous ne disposons pas de beaucoup de temps... Du coup comment gagner un appel d'offre sans être correctement outillé ?

Durant sa présentation, Nicolas Romanetti a expliqué ainsi que la SSII Eptica avait remporté un marché grâce à Celerio et au prototypage rapide. Pour nous, geeks de Java, derrière Celerio on retrouve des technologies bien sympathiques comme Spring ou Hibernate par exemple.

Ce que j'en pense c'est que Celerio est vraiment différent par rapport aux solutions open-source et gratuites. Il s'adresse aux personnes qui doivent gagner du temps. Et il a un concept unique que l'on ne trouve pas dans AppFuse. Avec un marché tendu où les SSII peinent à recruter des experts, c'est un outil qui permet aussi aux juniors de monter en compétence.

### Conclusion

Le format de présentation en 10 minutes, devant des investisseurs, forcément que cela tape dans l'œil et force chacun à être concis.

## Nouvelles de cette semaine

**Problèmes avec Wordpress** J'ai effectué une mise à jour de [Wordpress](#) ce matin et visiblement le plugin de cache a quelques soucis. En attendant que l'auteur publie une version qui tienne la route j'ai donc désactivé celui-ci. Merci à Flavien et à Samuel qui dès 08h10 m'avaient envoyé un email !

**Nouvelles des Java posses** Si vous voulez suivre l'actualité de Java et des technologies open-source, je vous recommande de vous abonner et d'écouter le podcast [JavaPosse](#). Créé il y a 3 ans, ce podcast propose presque chaque semaine des discussions sur l'actualité de Java, des éditeurs et des langages cousins comme Ruby on Rails ou [Scala](#). Je retiens cette semaine l'annonce de JSF 2.0 qui devrait nous proposer des nouveautés sympathiques. [Alexis Moussine Pouchkine](#) en parle sur [cette page](#). Enfin un peu de simplification dans JSF. **OSSGTP** Réunion ce soir de **OSSGTP**, le groupe des acteurs du monde open-source sur Paris. J'ai vu Jérôme Bernard hier soir et le thème [ce soir sera sur les licences](#). Pour le projet sur lequel je travaille, la question se pose. Je suis parti sur une GPLv3 pour l'instant. Je vous demanderai votre avis sur la question après vous avoir expliqué ce que je vais faire en septembre. Je n'aurai pas la chance d'assister ce soir à la réunion de l'OSSGTP comme la dernière fois, mais j'espère un jour gagner mes galons une fois mon projet sur les rails. C'est tout... pour le moment !

## iPhone 3G en rupture de stock

Ne cherchez pas à aller sur le site de la FNAC ou même dans les magasins parisiens, les boutiques Orange ont été dévalisées. Les vendeurs expliquent que chaque magasin a reçu dans les 100 téléphones. Les personnes achètent majoritairement la version 16 Go à 199 EUR. La FNAC digitale, Boulevard St Germain, tout vendu en 4H. En 1h [le stock de 40 appareils a été vendu](#). J'ai une piste pour vous : le magasin Orange de Noisy-le-Grand ce matin en a encore en stock... hé hé. Je vous laisse, le temps que vous lisiez ce message j'espère être l'heureux propriétaire de la petite bête ! Je vous raconte tout à l'heure vers 11h00. [Update 10h00] => plus d'iPhone 3G ni à la Fnac, ni chez Darty, ni au magasin Orange... je laisse tomber.

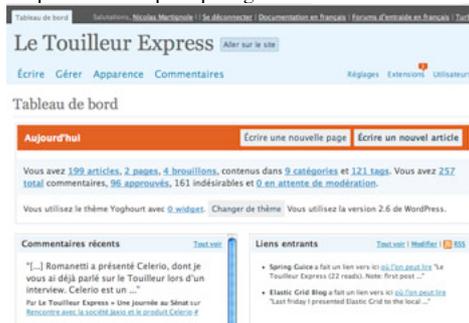
## Scrum : débriefing du sprint 6

**Si je vous dis que lorsque vous faites le plein de votre voiture, vous faites du Scrum, est-ce que vous me croyez ?** Instant particulier vendredi dernier : fin du sprint 6 mais surtout dernier Scrum Debriefing pour moi. En effet c'est Sylvain qui prend la suite de Karma. Je me rends compte que Scrum va faciliter le passage de connaissances juste avant de partir. **De l'importance de faire une démonstration au client** Vendredi nous avons donc effectué la démonstration de plusieurs fonctionnalités en présence du responsable du développement et de l'ergonome. Nous devions implémenter entre autre une fonction de navigation qui permet de charger rapidement une donnée (une contre-partie) à partir d'un écran de capture de deals standard. Bref nous voilà partis dans la démonstration... clique, clique.... hop, clique... (vous voyez ?).... clique Et c'est là que l'ergonome demande "...et si la personne ne met rien dans votre champ ? ...". Et bien nous avons beau avoir regardé ce changement dans tous les sens, nous avons complètement oublié un cas simple. Tout ceci pour dire que grâce à la revue de sprint nous avons trouvé un nouveau cas particulier qui sera corrigé dès demain. Merci Scrum. En effectuant une démonstration devant le client, le développeur (ou les développeurs) qui ont implémenté la fonctionnalité reçoivent directement un "merci" ou un "c'est pas ce que j'ai demandé" du client... Exercice parfois délicat, il faut rappeler à notre client préféré qu'il y a 10 jours c'est lui qui a demandé cette fonctionnalité. D'où l'importance de lui demander de raconter comment il fera la recette lui-même de la fonctionnalité. **Raconte moi une**

**histoire** Il faut que le client vous raconte une histoire. C'est de cette façon que les questions surgissent. Prenons un exemple : Le client: - "En tant qu'utilisateur je souhaite changer la langue de l'interface sans devoir quitter l'application. J'en ai besoin pour charger des concepts financiers exotiques et je ne veux pas perdre ma session." Moi: - "Très bien. Comment veux-tu changer de langue ?" - "Je veux cliquer sur un drapeau sur l'écran principal. Là une liste de drapeaux apparaît. Je clique sur le drapeau Japonais et toute l'interface bascule en japonais." - "Les labels s'affichent verticalement sur cette barre ?" - "En fait non, car cela n'a pas de sens pour cet écran" - "Et est-ce que le choix de la langue doit être sauvegardé si tu quittes et tu reviens dans l'application ?" - "Oui c'est important" - "Comment faire si nous n'avons pas la traduction d'un champ ?" - "Affichez alors le label par défaut. En anglais comme actuellement" Il faut demander au client de se projeter et d'imaginer la solution complètement terminée, afin qu'il identifie tous les besoins et que rien ne soit oublié. Ici il faudra implémenter la persistance du choix de la langue. Cela peut se faire en base de données mais nous n'avons pas besoin d'en parler ici. **La durée du sprint, tu peux la mettre à 13 jours ?** Non. Mise en situation : ce matin avec 2 personnes importantes de chez nous, nous avons effectué notre planification pour le Sprint 7. J'ai donc piloté les discussions entre les développeurs, l'ergonome, l'ingénieur financier et le responsable du produit chez nous. C'était assez acrobatique. A noter que c'est la première fois que le client final (le chef du produit) vient à notre réunion. Je me demandais comment Scrum serait perçu. Et bien cela s'est très bien passé. J'ai détecté quelque chose de nouveau avec Scrum. Lorsque le client final nous demande quelque chose, nous avons tendance à chercher trop tôt à résoudre tout le problème et donc à couvrir trop tôt l'ensemble de la demande. Je m'explique. L'une des demandes est d'implémenter une fonction de macro pour enregistrer à l'écran une saisie, afin de la rejouer automatiquement plus tard. Rappelons que nous sommes dans du web. Donc la solution est du côté du serveur dans notre framework. Et bien les développeurs discutaient de la manière de sauver en base ce scénario... alors que le client a un besoin fonctionnel. Il a fallu à cet instant recadrer et expliquer que le choix technique n'a aucune importance pour l'instant. En 10 jours, implémentons une partie sans la persistance, effectuons la démonstration au client, et nous serons capable d'aborder la persistance plus tard. Peut-être qu'en voyant cette fonction de macro nous allons décider de l'abandonner. D'où l'importance de ne pas consommer d'énergie pour rien pour l'instant. **L'histoire du plein d'essence** Pour moi Scrum peut se comparer au plein d'essence de ma voiture. Je fais le plein une fois, je roule avec le crédit d'essence que j'ai en faisant attention lorsqu'il ne reste plus beaucoup d'essence. Je peux suivre la consommation à l'aide d'une jauge qui me dit "ce qu'il reste" et pas ce que j'ai consommé. Réfléchissez à ceci : à quoi cela sert-il de savoir que pour venir de Paris, vous avez consommé 10 litres si ensuite vous devez passer le week-end à la campagne ? A rien. Ce week-end vous êtes à la campagne. Votre consommation sera différente. Ce qui est important c'est le "reste à consommer" qui est représenté par le "reste à faire" en Scrum. La manière dont vous avez consommé une partie de votre énergie sur un projet n'est pas une certitude sur comment vous allez dépenser le reste... D'où mes grandes idées sur l'inutilité d'un suivi avec un diagramme de Gantt... On ne peut pas prédire l'avenir d'un projet en regardant le passé. Il faut chaque jour se remettre en question afin de voir si le projet est sur les rails ou s'il part dans le décor. Je termine sur l'image du réservoir d'essence pour rappeler l'importance d'être constant dans la durée d'un Sprint. En effet, il n'y a rien de pire que de changer la durée d'un sprint. 10 jours c'est une durée qui ne doit pas changer. Votre réservoir fait 45 litres et pas 60 litres... Le fait de ne pas pouvoir modifier ce volume vous force à prendre des décisions. Lorsqu'il ne reste que 6 litres et que vous êtes à 20 km d'une pompe, vous prenez une décision. Scrum c'est la même chose. Pensez à moi la prochaine fois que vous ferez le plein, vous êtes entrain de faire du Scrum.

## Le 200ème article

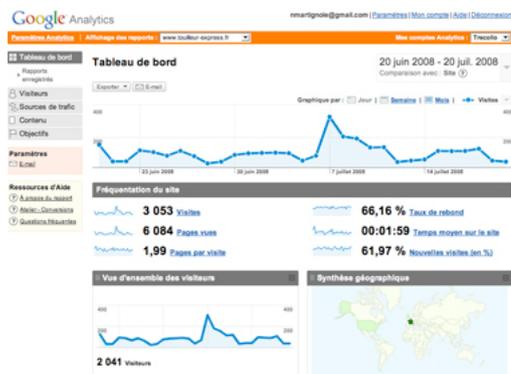
C'est le 200 ème billet que j'écris sur le Touilleur ! Je ne blague pas. J'étais entrain d'écrire un article pour vous dire que NASDAQ a sorti [une deuxième version de leur application](#) "Market Replay" utilisant Adobe AIR. Et là, au moment de sauvegarder, je vois sur le tableau de bord de WordPress que je viens de faire le 200ème billet.... Séquence émotion. Du coup me voilà à annuler ce billet et à revenir à une petite page de commémoration. Désolé mais ça fait quelque chose, après avoir commencé en novembre 2003 sur [JRoller](#) et maintenant sur mon serveur dédié, il fallait bien faire une petite pause ce soir pour partager avec vous cet instant. Regardez-moi cette belle



capture d'écran :

"Vous avez 199 articles, 2 pages, 4 brouillons, contenus dans 9 catégories et 121 tags. Vous avez 257 total commentaires, 96 approuvés, 161 indésirables et 0 en attente de modération." Allez on débouche le champagne... Nous en sommes à 2041 visiteurs différents depuis le 20 juin dernier... Cela continue à grimper... Mais de quoi voulez-vous que je parle pour vous passionner encore plus ? Plus de Scrum ? Plus de Java ? Plus de "il raconte n'importe quoi mais on se marre bien" ?

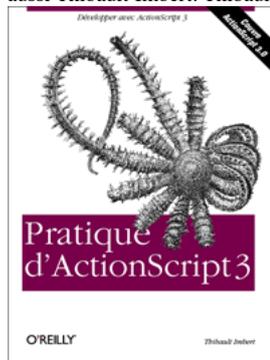
"Vous avez 199 articles, 2



Le premier article du Touilleur date du 22 novembre 2003 et à l'époque j'expliquai [pourquoi Java s'appelle Java](#). Me demande si le blabla de cette époque est vrai ou pas... Où suis-je allé chercher cela ? Allez tchin, et restez fidèle au Touilleur Express !

## RDV chez Adobe

Ce midi je suis allé rencontrer une petite partie de l'équipe d'Adobe Systems France, basée à Paris près du Trocadéro. François le Droff que j'ai rencontré à l'OSSGTP mais que je connais depuis pas mal de temps par blog interposé, travaille en tant que consultant et architecte sénior. Il est l'un des auteurs d'un archetype Maven très pratique que j'utilise pour mon projet. "[My BlazeDS](#), [XDoclet](#), [Spring](#), [Hibernate](#)" est un archetype maven qui permet de créer et compiler un projet maven pour démarrer rapidement un projet Flex constitué d'un noyau Java pour la partie service, de BlazeDS pour la partie transport et de Flex pour la partie rendu. Si vous souhaitez ajouter le support des composants de Chart je vous renvoie vers l'article du début de ce mois [sur le blog](#). J'ai rencontré Xavier AGNETTI, Consultant, qui travaille sur un projet intéressant : flex-pmd. Ce projet permet d'ajouter la recherche et l'analyse de code action script à l'intégration continue avec Ant ou Maven, afin de détecter des erreurs que nous ne pouvons pas toujours voir à l'écriture du code. Xavier a fait aussi une démonstration de l'intégration avec Hudson et de la génération des rapports sur le code flex sample. Nous avons parlé de [VELO](#), ce contributeur brésilien qui participe et réalise le projet [Flex-Mojos](#). Ce projet permet d'utiliser facilement Maven avec le Flex SDK et BlazeDS. C'est un projet indispensable pour industrialiser correctement un projet d'envergure. J'ai rencontré aussi Thibault Imbert. Thibault ([blog](#)) est l'auteur du livre "Pratique d'Action Script 3":



Le livre sera édité par Pearson Education à la fin de cette année, après que l'éditeur O'Reilly France a décidé de [fermer ses portes](#). C'est bien triste car les livres O'Reilly sont d'excellente qualité. Le livre de Thibault d'après la table des matières m'a tout l'air d'être très complet. Voyez plutôt : **Chapitre 1 (5 pages) - Qu'est ce que l'ActionScript 3 ?**

- o Histoire
- o 10 raisons de coder en ActionScript 3
- o Outils
- o La plateforme Flash

### **Chapitre 2 (36 pages) - Langage et API ?**

- o Le langage ActionScript 3
- o Machines virtuelles
- o Traduction dynamique
- o Gestion des types à l'exécution
- o Erreurs à l'exécution
- o Nouveaux types primitifs
- o Valeurs par défaut
- o Nouveaux types composites
- o Nouveaux mots-clés
- o Fonctions
- o Contexte d'exécution
- o Boucles
- o Enrichissement de la classe Array
- o Ramasse-miettes
- o Bonnes pratiques
- o Autres subtilités

**Chapitre 3 (33 pages) - Le modèle événementiel**

- L'histoire
- Un nouveau modèle événementiel
- Tout est événementiel
- Ecouter un événement
- L'objet événementiel
- La classe Event
- Les sous-classes d'Event
- Arrêter l'écoute d'un événement
- Mise en application
- La puissance du couplage faible
- Souplesse de code
- Ordre de notification
- Références faibles
- Subtilités

**Chapitre 4 (41 pages) - La liste d'affichage**

- Classes graphiques
- Liste d'affichage
- Instanciation des classes graphiques
- Ajout d'objets d'affichage
- Réattribution de parent
- Liste interne d'objets enfants
- Accéder aux objets d'affichage
- Suppression d'objets d'affichage
- Effondrement des profondeurs
- Gestion de l'empilement des objets d'affichage
- Echange de profondeurs
- Désactivation des objets graphiques
- Fonctionnement interne de la tête de lecture
- Subtilités de la propriété stage
- Subtilités de la propriété root
- La notion de \_level

**Chapitre 5 (30 pages) - Symboles prédéfinis**

- Les types de symbole
- Le symbole clip
- La propriété name
- Instanciation de symboles par programmation
- Instanciation de symboles prédéfinis
- Extraire une classe dynamiquement
- Le symbole bouton (SimpleButton)
- Le symbole graphique (Graphics)
- Les images bitmap (BitmapData)
- Afficher une image bitmap
- Le symbole Sprite
- Définition du code au sein d'un symbole

**Chapitre 6 (33 pages) - La propagation événementielle**

- Concept
- La phase de capture
- La notion de noeuds
- Déterminer la phase en cours
- Optimiser le code avec la phase de capture
- La phase cible
- Intervenir sur la propagation
- La phase de remontée
- Ecouter plusieurs phases
- La propriété Event.bubbles
- Suppression d'écouteurs

**Chapitre 8 (74 pages) - Programmation orientée objet**

- Concevoir autrement
- Tout est objet
- Notre première classe
- Introduction aux paquetages
- Définition de propriétés
- Attributs de propriétés de classe
- Attributs de classe
- Le constructeur
- Utilisation du mot clé this
- Définition de méthodes
- L'encapsulation
- Mise en pratique de l'encapsulation
- Les méthodes d'accès
- Contrôle d'affectation
- Méthodes en lecture-écriture

- Cas d'utilisation de l'attribut static
- La classe JoueurManager
- L'héritage
- Sous-type et super-type
- Spécialiser une classe
- Le transtypage
- Surcharge

#### **Chapitre 9 (83 pages) - Etendre les classes natives**

- Intérêts
- Le vieil ami prototype
- Etendre les classes non-graphiques
- Etendre les classes graphiques
- Accéder à l'objet Stage de manière sécurisée
- Ajouter des fonctionnalités
- Réutiliser le code
- Classe dynamique
- Un vrai constructeur
- Créer des boutons dynamiques

#### **Chapitre 10 (29 pages) - Diffusion d'événements personnalisés**

- L'histoire
- La classe EventDispatcher
- Mise en application
- Choisir un nom d'événement
- Etendre EventDispatcher
- Stocker EventDispatcher
- Passer des informations
- Menu et événement personnalisé

#### **Chapitre 11 (27 pages) - Classe de document**

- Intérêts
- La classe MainTimeline
- Classe de document
- Limitations de la classe Sprite
- Actions d'images
- Déclaration automatique des occurrences
- Déclaration manuelle des occurrences
- Ajouter des fonctionnalités
- Initialisation de l'application
- Accès global à l'objet Stage
- Automatiser l'accès global à Stage

#### **Chapitre 12 (81 pages) - Programmation bitmap**

- Bitmap et vectoriels
- Couleurs
- Manipuler les couleurs
- La classe BitmapData
- Codage des couleurs
- Gérer les ressources avec le profiler
- La classe Bitmap
- Réutiliser les données bitmap
- Libérer les ressources
- Calculer le poids en mémoire
- Limitations mémoire
- Images en bibliothèque
- Peindre des pixels
- Lire des pixels
- Accrochage aux pixels
- Le lissage
- Mise en cache des bitmap à l'exécution
- Effets pervers
- Filtrer un élément vectoriel
- Filtrer une image bitmap
- Animer un filtre
- Rendu bitmap d'objets vectoriels
- Optimiser les performances

#### **Chapitre 13 (84 pages) - Chargement de contenu**

- La classe Loader
- Charger un élément externe
- La classe LoaderInfo
- Interagir avec le contenu
- Créer une galerie
- La composition
- Redimensionnement automatique
- Gestion du lissage

- Précharger le contenu
- Interrompre le chargement
- Communiquer entre deux animations
- Modèle de sécurité du lecteur Flash
- Programmation croisée
- Utiliser un fichier de régulation
- Contexte de chargement
- Contourner les restrictions de sécurité
- Bibliothèque partagée
- Désactiver une animation chargée
- Communication AVMI et AVM2

#### Chapitre 14 (98 pages) - Chargement et envoi de données

- La classe URLLoader
- Charger du texte
- L'encodage URL
- Charger des variables
- Charger des données XML
- Chargement de données et sécurité
- Charger des données binaire
- Concept d'envoi de données
- Envoyer des variables
- la méthode GET ou POST
- Envoyer des variables discrètement
- Renvoyer des données depuis le serveur
- Aller plus loin
- Envoyer un flux binaire
- Télécharger un fichier
- Publier un fichier
- Publier plusieurs fichiers
- Création de la classe EnvoiMultiple
- Retourner des données une fois l'envoi terminé
- Aller plus loin

#### Chapitre 15 (32 pages) - Communication externe

- Conteneur et contenu
- Passer des variables
- Intégration par JavaScript
- La propriété parameters
- Les FlashVars
- Passer des variables dynamiquement
- Accéder facilement aux FlashVars
- Appeler une fonction
- L'API ExternalInterface
- Appeler une fonction externe depuis ActionScript
- Appeler une fonction ActionScript depuis le conteneur
- Communication et exceptions
- Communication et sécurité

#### Chapitre 16 (82 pages) - Le texte

- Le texte dans Flash
- Afficher du texte en Flash
- Afficher du texte par programmation
- Les types de champ texte
- Formatage du texte
- Rendu HTML
- La classe TextFormat
- Étendre la classe TextField
- La classe StyleSheet
- Modifier le contenu du champ texte
- Remplacer du texte
- L'événement TextEvent.LINK
- Charger du contenu externe
- Exporter une police dans l'animation
- Charger dynamiquement une police
- Détecter les coordonnées d'un caractère
- Créer un éditeur de texte enrichi

#### Chapitre 17 (73 pages) - Son et vidéo

- Lecture de sons
- Lire un son provenant de la bibliothèque
- Lire un son dynamique
- La classe SoundLoaderContext
- Transformation du son
- Modification globale du son
- Lire le spectre d'un son
- Transformée de Fourier
- Le format MPEG-4 audio (AAC)

- La vidéo dans Flash
- Le format MPEG-4 vidéo (H.264)
- La classe Video
- Transformation du son lié à un objet NetStream
- Mode plein écran

#### Chapitre 18 (39 pages) - Les sockets

- Une passerelle universelle
- Créer un serveur de socket XML
- La classe XMLSocket
- Créer un forum de discussion
- La classe Socket
- Créer un serveur de socket binaire
- Echanger des données

#### Chapitre 19 (74 pages) - Flash Remoting

- La technologie
- Un format optimisé
- Passerelle Remoting
- Déploiement
- Le service
- Se connecter au service
- La classe Responder
- Appeler une méthode distante
- Echanger des données primitives
- Echanger des données composites
- Echanger des données typées
- Envoyer un e-mail
- Exporter une image
- Se connecter à une base de données
- Sécurité
- La classe Service

#### Chapitre 20 (44 pages) - ByteArray

- Le codage binaire
- Position et poids du bit
- Octet
- Ordre des octets
- La classe ByteArray
- Méthode de lecture et d'écriture
- Copier des objets
- Ecrire des données au format texte
- Lire des données au format texte
- Compresser des données
- Sauvegarder un flux binaire
- Générer un PDF

#### Chapitre 21 (30 pages) - Application finale

- Décomposer l'application
- Gérer les chemins d'accès
- Création d'un fichier XML de configuration
- Utiliser les bibliothèques partagées
- Générer et charger les bibliothèques
- Utiliser les bibliothèques dans l'ensemble de l'application

Vous pouvez librement [télécharger le travail de Thibault](#) sans oublier de faire une donation via PayPal. Quand on sait la quantité de travail nécessaire pour écrire un livre, je trouve cela très généreux de le partager et de le mettre à la disposition de la communauté.

Bon j'y retourne, j'ai une tonne de travail.

## Mes projets pour septembre

Il est temps d'en dire un peu plus sur mon avenir. A quelques jours de la fin de mon contrat chez Thomson-Reuters, voici quelques informations sur les projets pour la fin de cette année. **Création d'une Entreprise** J'ai fait le choix de créer "Innoteria", une structure sous la forme d'une EURL. Les démarches de recherche et de création d'une entreprise, vous imaginez que j'ai de quoi bloguer pour les mois à venir. L'entreprise sera sur les rails début août. Je vous raconterai les démarches, les questionnements et les grandes interrogations lorsque l'on se lance tout seul. C'est passionnant, si vous souhaitez vous mettre Freelance j'espère vous faire partager ici mon expérience. Voici le logo d'Innoteria: [caption id="" align="alignnone" width="262" caption=" "]



[/caption] En plus de cette mission, j'ai commencé quelques démarches pour contacter des amis, des cabinets de conseils avec qui j'aimerais travailler.

Je souhaite intervenir en tant que chef de projet Scrum, architecte Java ou dans le domaine des applications internet riches (RIA). Mes éléments différenciateurs : en terme de veille technologique, j'ai une bonne culture des produits, des outils et de tout ce qui peut faciliter la vie d'un développeur. Grâce à cela j'ai par exemple apporté des outils chez Reuters, afin de réduire les coûts et d'améliorer la qualité des développements. Si vous êtes intéressés, vous pouvez retrouver mon profil sur le site [Linkedln.com](http://www.linkedin.com) **Le projet Lombok** J'ai commencé à travailler sur mon temps libre sur un projet basé sur Adobe Flex, BlazeDS, Spring, Hibernate et JBoss Seam. L'idée est de proposer une interface riche de visualisation de données réseaux et métiers, collectés par un système dont je suis simplement client. Je vous montrerai au fur et à mesure de l'avancement de notre projet quelques écrans. Cela me permet de réaliser un projet métier basé sur Flex/AIR tout en utilisant mes connaissances du côté serveur. Ce projet a été initialisé sur l'idée d'un ami, et nous travaillons pour un premier client dès le mois d'août. Nous regardons pour mettre en licence GPL une partie du projet global, sur la couche métier et présentation. Pour l'instant c'est un sujet sur lequel nous travaillons encore. La suite des aventures dans quelques mois...

## Java sur l'iPhone

[caption id="" align="alignnone" width="320" caption="Compiler du java sur l'iphone"]

```

Last login: Sat Jun  7 02:36:55 on ttty2
iphone:~ mobile$ java -fullversion
java full version "jamvm-1.5.0"
iphone:~ mobile$
iphone:~ mobile$ jamvm -fullversion
java full version "jamvm-1.5.0"
iphone:~ mobile$
iphone:~ mobile$ jikes
use: jikes [options] [@files] file.java...
For more help, try -help or -version.
iphone:~ mobile$
iphone:~ mobile$ cd /tmp
iphone:/tmp mobile$ ls
Java4IPhone.java  MediaCache/  launchd/
iphone:/tmp mobile$ jikes Java4IPhone.java

Found 1 system error:

*** Semantic Error: You need to modify your
classpath, sourcepath, bootclasspath, and/or
extdirs setup. Jikes could not find package
"java.lang" in:

.

iphone:/tmp mobile$ jikes Java4IPhone.java -
cp /usr/lib/rt.jar
iphone:/tmp mobile$ java Java4IPhone
Hello from www.Java4IPhone.com
iphone:/tmp mobile$
  
```

[/caption] Entendu ce matin sur le podcast [des JavaPosses](#), il est possible d'installer une JVM sur un iPhone 3G. Attention cependant, pour cela il faut jailbreaker son "précieux" et donc, à réserver pour les geeks pour l'instant ([voir cet article complet](#)). On peut s'interroger pourquoi Apple n'est pas un sponsor fort de Java pour l'iPhone. Cela s'explique par le fait qu'Apple souhaite que les développeurs se basent sur la plateforme de développement Apple, plutôt que de proposer le support de Java. [Le kit de développement de l'iphone](#) a tout son sens, lorsque l'on regarde l'architecture de la plateforme de l'iPhone. J'avoue que s'agissant d'applications clientes "riches" il me semble plus logique d'utiliser le kit d'Apple que de sortir ma boîte à outil de développeur Java. Ensuite si l'on désire offrir le support des particularités de l'iPhone (interface multi-touch, reconnaissance du sens d'affichage...) il sera plus pratique de travailler nativement dans le format proposé par Apple. Si l'aventure vous tente : enregistrez-vous tout d'abord sur le site <http://developer.apple.com/iphone/program>. Il vous faudra par contre un Mac pour développer... désolé !

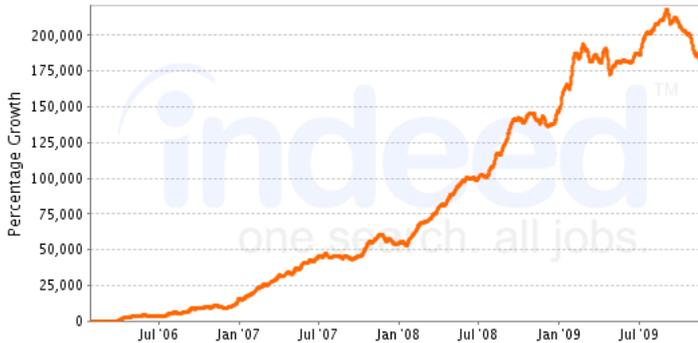
## Adobe Flex : des liens pour toi, public

Des URLs sur Flex, en veux-tu, en voilà. Je crois que j'ai bien explosé mon répertoire "Flex" sur Firefox et avant de tout perdre, je me suis dit qu'il serait pas mal pour une fois de mettre ici une tartine de liens utiles. Je vous épargne les liens qui ne servent pas trop. Enfin j'essaye. J'en profite pour rappeler 2 choses à propos de Flex et de Flash : le contenu Flash est indexé [par Google](#) depuis quelques temps et le bouton "Back" du navigateur est très bien géré dans les applications Flex. **ScaleNince** <http://www.scalenine.com> Ce site propose un large éventail d'habillages pour les applications Flex et AIR. J'ai trouvé marrante l'idée d'avoir une skin (un habillage) [aux couleurs de Windows XP](#). **Nasdaq Replay** <https://data.nasdaq.com/MR.asp> Cette application financière en Flex permet de revoir les données du marché de l'indice NASDAQ. Pour les Traders cela permet de rejouer les fluctuations de cours sur le marché. Un peu comme un magnétoscope qui vous permettrait de revoir n'importe quel but des dernières coupes du monde. Vous pouvez même visualiser en pseudo-temps réel les données, si vous voulez tester des moteurs de valuation par exemple. Ce serait un peu comme si un trader pouvait se visionner n'importe quel instant de la vie d'une cotation pour qu'il puisse s'entraîner et voir si ses systèmes de calculs cambistes sont bons. L'application est basée sur Adobe AIR et donc fonctionne en application standalone. Les données du NASDAQ sont stockées sur Amazon S3, la méga ferme de stockage d'Amazon dont j'ai déjà parlé. J'ai lu [dans cet article](#) que NASDAQ charge 300 000 entrées par jour vers Amazon S3 et que pourtant, les temps d'accès sont linéaires. **Tutorial Livecycle Data Synchronization**

(<http://www.insideria.com/2008/07/air-data-synchronization-via-l.html>) Pour ceux qui étaient là avec moi **le 9 juillet dernier**, est-ce que vous vous souvenez de la démonstration de l'application CRUD légère basée sur Flex et Livecycle Data Service ? J'ai mis la main sur un tutoriel qui mériterait que je le traduise en français et que je vous conseille de lire. Vous êtes guidé pas à pas dans la construction d'une application AIR capable de fonctionner en mode déconnecté, et utilisant le serveur Adobe LifeCycle Data Services 2.6. Les données du côté du client peuvent être stockées automatiquement dans une base SQLite, une sorte de Google Gears. Le tout en utilisant un simple mot clé dans votre fichier MXML. Elle est pas belle la vie ? **Le blog de Christophe Coenraets** <http://coenraets.org> Vous trouverez sur le blog de cet évangéliste un petit aperçu de ce que la technologie Flex peut faire. D'une application de chat à distance avec webcam en passant par une démonstration de LiveCycle avec GoogleMaps. **Le blog de Serge Jesper** <http://www.webkitchen.be/> Ce blog contient d'excellents articles, dont [une comparaison entre Flex et Microsoft Silverlight](#). Serge a assisté à un training de 3 jours chez Microsoft, et avec 12 ans d'expérience comme développeur Flash, il apporte un éclairage intéressant sur Silverlight. Pour ma part vous savez déjà ce que je pense pour l'instant de Silverlight. J'ai aussi apprécié [son étude sur l'évolution des demandes de compétence Adobe Flex](#) ces derniers mois. En France nous sommes encore au calme et je sais que d'ici 6 mois, un profil avec une compétence Flex aura toutes ses chances sur le marché. [caption id="" align="alignnone" width="540" caption="Poste adobe flex"]

Job Trends from Indeed.com

— "adobe flex"



[/caption] Vous

pouvez observer sur le site [Indeed.com](http://Indeed.com) la tendance en terme de demande d'emploi pour un mot clé donné. Ainsi "[java](#)" est en légère chute de 5% depuis septembre 2008 aux USA. La plateforme [.NET](#) aux USA est en forte demande avec 30% d'évolution. Je suis content de voir que "[Scrum](#)" est aussi en forte augmentation. C'est ce que j'ai expliqué un ami qui reste sur d'anciennes méthodes de gestion de projet. Nous n'avons pas conscience en France que les projets "à l'ancienne" c'est terminé. Tapez RUP sur [indeed.com](http://indeed.com), c'est 30% de baisse en 2 ans... Plus personne n'en veut !!! Avis aux x-mens : voilà de quoi muscler vos slides pour convertir les gens à Scrum. Voilà c'est tout pour ce soir. Je compléterai avec d'autres URLs dans les jours qui viennent. Sinon avez-vous vu ce lien de Google ? ([http://www.google.com/help/netneutrality\\_letter.html](http://www.google.com/help/netneutrality_letter.html)) Très bizarre... je n'en n'ai pas entendu parler. Demain c'est mon pot de départ chez Reuters. Plus que 4 jours... cela fait bizarre de se dire que jeudi soir, c'est fini. Ah là là...

## Time Capsule

Mardi soir, c'était mon pot de départ. Aujourd'hui jeudi, dernier jour chez RFS (Reuters Financial Software, Thomson Reuters, Puteaux) Mes collègues m'ont gâté puisqu'ils m'ont offert [une Time Capsule](#), le système de sauvegarde d'Apple. J'ai pris quelques photos car franchement, l'ouverture de la boîte c'est le spectacle assuré. J'ai branché la bête dans le salon et l'iMac est actuellement en cours de sauvegarde. La première sauvegarde est un peu longue. Je l'ai lancé hier soir, et ce matin environ 70% de 105 Go étaient sauvegardés. Le boîtier chauffe pas mal, je verrai ensuite une fois le backup terminé si je ne vais pas déplacer la capsule autre part. Pour la sauvegarde des données de mon entreprise, c'est parfait. Le boîtier est très silencieux. Un grand, grand merci à toutes les équipes de Reuters avec lesquelles j'ai travaillé pendant ces 5 ans 1/2 : TradeAccess, WebAccess, KGR, K+TP, KSP, Kondor+, les gens de PDS... Beaucoup de monde. A bientôt **Update** : Petit détail qui tue : "Designed by Apple in California" marqué sur le dessus de la boîte...



Hop



Enfin un dernier souvenir de l'open-space avec le fameux tableau Scrum utilisé comme Sprint Backlog. Et en prime une photo d'Hasan qui a



bossé avec moi sur Karma.

## Exadel Flamingo : JBoss Seam et Adobe Flex ensemble

Votre souci : vous souhaitez une interface cliente riche tout en disposant d'une architecture puissante du côté du serveur. Comment faire ? Regardons ce qu'il se passe aujourd'hui : terminé le temps de la page HTML qui se recharge complètement. Les utilisateurs finaux veulent de l'interactivité, une interface riche et une ergonomie fantastique. D'un autre côté, le souci pour vous en tant qu'architecte est de conceptualiser et écrire une couche de service correcte. Pour le projet sur lequel je travaille, nous avons une application développée avec JBoss Seam en début d'année. Le développement avec IceFaces des écrans principaux ne m'a pas pris beaucoup de temps. Par contre, s'agissant d'une application de surveillance et de monitoring, j'ai pas mal de difficultés à obtenir un résultat puissant en terme d'ergonomie et de réactivité. Je me suis donc tourné avec bonheur vers Adobe Flex. Le fait aussi de pouvoir fonctionner en mode déployé avec Adobe AIR est un plus. En prenant JBoss Seam et Adobe Flex, il est possible de faire une application vraiment puissante. Pour cela, [Exadel Flamingo](#) est vraiment la dernière brique qu'il me manquait pour finaliser l'architecture. Flamingo est un outil très puissant qui permet tout d'abord de générer rapidement une application type CRUD avec une interface utilisateur en Flex. Si vous avez testé seam-gen,

vous voyez ce que je veux dire. Flamingo utilise le protocole binaire AMF d'Adobe pour la communication entre l'interface cliente déployée en Flash du côté navigateur et donc, la partie serveur basée sur JBoss Seam. C'est similaire à un déploiement avec BlazeDS ou LiveCycle. Cela dit, en terme d'échanges d'objets, je me demande si JBoss Seam va aussi loin que LiveCycle. L'intérêt de Flamingo est qu'en générant du code pour vous, il facilite l'écriture en ActionScript du côté du client. Après avoir défini une Entité du côté de Seam, vous pouvez facilement appeler un ensemble de méthode simple pour effectuer la recherche ou la mise à jour de votre entité à partir du code ActionScript. Flamingo est donc avant tout un générateur de code destiné à réduire la quantité de code à écrire, et qui facilite l'accès aux Entity EJB3 gérées par Seam. Les échanges entre le client en Flex et le serveur JBoss Seam peuvent s'effectuer soit via une Servlet spéciale de Flamingo, soit via la partie Remoting de JBoss Seam. Flamingo travaille avec Maven2 pour générer la glue entre JBoss Seam d'une part et les technologies RIA comme Flex ou JavaFX d'autre part. Le mieux sera de vous écrire un petit exemple complet pour que vous puissiez vous faire une idée de l'intérêt de Flamingo. Désolé faute de temps je vous laisserai que deux URL pour que vous puissiez regarder : - [version JBoss Seam avec JSF seul](#) - [version Adobe Flex avec protocole AMF](#)  
Retrouvez d'autres urls à cette adresse : <http://demo.flamingo.exadel.com/booking/booking.html>

## Mac Book Pro

Brève du jour : je me suis commandé un Mac Book Pro 15" sur l'AppleStore. Je vous en reparle dès que le colis est arrivé, ce qui ne saurait tarder.

## Google Developer Day le 18 septembre

Sur invitation de la société SpringSource, je participerai au [Google Developer Day](#) le 18 septembre à Paris. Cette journée de présentation des différentes technologies de Google permettra aux développeurs de se tenir informés sur les API de Google, de se rencontrer et de faire des échanges. [L'agenda](#) de la journée contient des sessions sur Android, les Mashups, Google App Engine, les API YouTube, Open Social et surtout GWT. Avis à la population : session sur GWT faite par des pointures de Google. L'inscription est gratuite, même si le nombre de places est limité. Donc si vous êtes intéressés, et pas en vacances, profitez-en pour vous inscrire. Merci à Julien Dubois de [SpringSource](#), et donc je ne manquerai pas de faire un compte-rendu ici sur le blog de cette journée.

## Mac Book Pro arrivé

Le MacBook Pro est arrivé vendredi dernier. Trop de boulot et à peine eu le temps de le déballer... Mais c'est maintenant chose faite. Ce premier message tardif vous est envoyé en direct de la bête. Clavier rétro-éclairé, je peux taper dans le noir. Le bonheur avec un mac c'est que Java 5 et 6 sont déjà installés, ainsi que SVN, MVN et Ant. Sitôt la bête sortie du carton, vous pouvez donc travailler. Cela laisse rêveur non ? [IDEA IntelliJ 8](#) (mon éditeur Java favori) est disponible en bêta M1. Au menu : support de Spring 2.5, JBoss Seam, Adobe Flex et Google Web Toolkit. Vous l'aurez deviné, je l'ai déjà installé. Je passe le support de maven qui est parfait (à comparer à son voisin de palier, monsieur eclipse) et j'arrête de bosser pour ce soir. Qui a dit qu'indépendant on travaillait moins ? hein qui ?

## La rentrée : groovy

Un peu d'actualité sur Paris et quelques liens à regarder. La prochaine soirée du Paris Java User Group aura lieu [mardi soir prochain à Paris à l'ISEP à partir de 19h15](#) autour du langage Groovy présenté par Guillaume Laforge, chef de projet de Groovy et membre de l'OSSGTP. L'intérêt de Groovy est à mon avis de dépasser l'étape de langage de script pour vraiment apporter à la plateforme Java un langage dynamique. Guillaume est aussi spec-lead de la [JSR-241](#) qui vise à standardiser Groovy au sein de la plateforme Java. Groovy permet de retirer pas mal de code d'assemblage et de s'approcher de la mouvance des langages de script amorcée avec Java 6. On ne peut pas parler de Groovy sans parler de [Grails](#), le framework MVC basé sur Groovy. La deuxième partie de la soirée sera donc consacrée à la présentation de Grails par Fabrice Robini, architecte chez Octo. J'attends avec impatience de voir un peu une présentation en live de Grails. J'espère que Fabrice pourra refaire ce que Jean-François Hélie et Christian Blavier nous avaient proposé [en juin dernier](#). Le 18 septembre se déroulera aussi le [Google Developer Day 2008](#) à l'ENSA. Les sessions sur GWT et sur AppEngine pourraient intéresser pas mal de monde. Je ne manquerai pas de vous en faire un petit compte-rendu. Si vous voulez vous regarder quelques petites vidéos sympathiques, je vous conseille de regarder les vidéos de [Jazoon 2008](#) une conférence dont je vous avais parlé [il y a quelques mois déjà](#). A propos de vidéo, si vous souhaitez regarder tranquillement un screencast de votre MacBook Pro préféré, le site [Parleys.Com](#) héberge un certain nombre de vidéo sur l'écosystème Java, Spring, Mule, JBoss et compagnie. Je vous recommande par exemple [la présentation de Mule 2](#) de Ross Masson ou [l'intervention de Rod Johnson](#) sur Spring 2. Au passage on note que le site a été réalisé avec Adobe Flex et qu'il existe même une version AIR. Pas mal non ?

## Se mettre à son compte

Je regarde une pile de magazines entassés sur l'imprimante : "Se mettre à son compte, L'Entreprise mars 2006", "S'installer comme Consultant, L'Entreprise juin 2006", "Toutes les

*aides pour se mettre à son compte, Managment Mars 2005" et enfin "Créer sa boîte chez soi, L'Entreprise aout 2005". Cher lecteur, voici maintenant "Je me mets à mon compte, le Touilleur Express, septembre 2008". Bonne lecture ! Parlons tout d'abord de ce qu'il est possible de faire le jour où vous déciderez de vous lancer en tant que Freelance. Je ne parlerai ici que de l'activité de consultant en informatique, sujet qui nous intéresse. Si vous souhaitez vous lancer en tant que Freelance dans un autre secteur (graphiste, création de site web, traducteur...) je vous conseille vivement de regarder le site <http://www.freelance-info.fr>. Son Forum est une mine d'or. Lorsque vous quitterez le nid douillet du statut de salarié, vous pourrez exercer votre activité sous différents statuts. Je ne vais pas tous les décrire, mais simplement vous parler ici de ce qu'il est possible de faire. Tout d'abord le portage salarial, ensuite l'activité en tant que profession libérale et enfin la création d'une entreprise, sujet qui m'a occupé cette semaine. **Le portage salarial** Le portage salarial est un moyen simple, rapide et efficace pour démarrer une activité sans devoir effectuer un grand nombre de démarche. En effet vous restez salarié, tout en bénéficiant d'une protection sociale. Les frais de gestion varient aujourd'hui entre 5% et 12% du chiffre d'affaire facturé au client. Ainsi si vous êtes payés 100 euros la journée, vous pouvez donc espérer gagner 90 euros. A cela s'ajoutera la commission de la SSII ou du cabinet de placement qui peut aller de 15% à 25%. Au final donc vous devez réaliser que pour 100 EUR facturé à votre client final, vous ne touchez en fait que 75 EUR par exemple. ITG est une bonne société de portage salarial qui existe depuis pas mal de temps, avec environ 2500 personnes "portées". Les avantages : c'est le moyen le plus rapide pour démarrer une activité. Les formalités sont très simples. Vous pouvez vous concentrer sur le métier sans perdre de temps avec les démarches. Pas de comptabilité. Les inconvénients : vous faites une croix sur une partie de votre chiffre d'affaire. Attention aussi aux sociétés de portage douteuses. **L'entreprise individuelle, exercer en tant que profession libérale** Ensuite vous pouvez vous enregistrer comme profession libérale à l'URSSAF. C'est une démarche rapide et gratuite que vous pouvez faire [partiellement par Internet](#). Fin juillet à la fin de mon contrat chez Reuters, c'est ce que j'ai fait afin de pouvoir démarrer mon activité en août. Une fois le formulaire P0 de l'URSSAF rempli, j'ai posté le tout et j'ai reçu 6 jours plus tard une attestation URSSAF ainsi que le courrier de l'INSEE avec mon numéro de SIREN/SIRET. L'avantage de cette procédure : très simple. Vous ne créez pas d'entreprise, puisque c'est vous en tant que personne qui exercera l'activité. Les inconvénients : vos bénéfices seront déclarés fiscalement sur votre déclaration de revenu comme BNC (bénéfice non commerciaux) et vous risquez d'avoir une surprise concernant votre impôt sur le revenu. En effet, imaginons que chaque journée de travail est facturée 400 euros. Pour 230 jours d'activité dans une année que vous allez effectuer, cela représente un chiffre d'affaire de 92 000 euros. Les charges sociales et vos frais de fonctionnement vont retirer environ 35% de ce montant. Vous serez alors imposé sur la somme de 59 800 euros. Belle somme, mais attention selon votre situation personnelle (marié, enfants...) vous risquez de payer beaucoup d'impôts sur le revenu. L'activité en Entreprise Individuelle est donc simple, il y a peu de démarche à effectuer, c'est gratuit. De plus vous n'êtes pas tenu de publier un bilan, et donc la comptabilité est très simple. Au niveau de la TVA, l'activité s'exerce en franchise de TVA. Donc vous facturez sans TVA vos clients (en HT) et en retour vous ne pouvez pas déduire la TVA de vos achats. Un ordinateur sera payé 2000 euros TTC, comme un particulier. Au niveau bancaire vous n'êtes pas tenu d'ouvrir un compte pro. Cependant votre banquier, voyant des chèques assez importants, ne manquera pas de venir vous solliciter. A voir donc. Les avantages : vous exercez en tant qu'entreprise, tout en conservant la souplesse et la flexibilité. Les inconvénients : vous perdez le statut de salarié, vous exercerez en tant que Travailleur Non Salarié pour la caisse d'assurance maladie. Il faudra prévoir une complémentaire santé et une prévoyance s'il vous arrive quelque chose. Compter dans les 150 EUR par mois, charge déductible cependant de votre chiffre d'affaire. Le gros inconvénient du statut de EI, c'est que vous êtes responsable sur vos biens personnels indéfiniment. Si vous êtes mariés, votre conjoint est aussi engagé. Il n'y a pas de distinctions entre votre patrimoine et celui de votre entreprise. En août j'étais parti sur ce schéma de fonctionnement. Simplement, je souhaite plus tard pouvoir gérer plus finement ma fiscalité. Pour cette raison j'ai effectué quelques démarches pour créer une EURL à capital fixe. C'est là que la partie de plaisir commence. **La création d'une EURL 1) Première possibilité : se rendre au CFE** - Vous pouvez vous rendre au CFE de la Chambre des Métiers et de l'Industrie du département où vous souhaitez domiciler votre entreprise. Sur place il est possible de retirer les formulaires nécessaires ou même d'effectuer l'ensemble des démarches si votre dossier est complet. Compter 60 euros pour se faire aider d'un assistant sur place. Ensuite le transfert du dossier au greffe prendra 10 jours. C'est un peu long. **2) autre alternative, passer par Internet** - Se rendre sur le site Internet <http://www.cfenet.cci.fr> afin d'effectuer les mêmes démarches qu'au CFE. Vous allez remplir tout un ensemble de formulaire afin de créer l'entreprise (formulaire M0) et de déclarer le statut social des dirigeants (TNS). Le souci c'est qu'il faut ensuite poster le tout en recommandé. Certes vous pouvez en principe tout effectuer via Internet, si et seulement si, vos documents sont scannés et certifiés conforme. Il faudra donc aller à la mairie faire certifier votre photocopie de carte d'identité ou les statuts. **3) dernière alternative : tout faire soit même** Finalement c'est ce que j'ai fait. J'ai trouvé les formulaires CERFA M0 et TNS sur le site de l'URSSAF. Ces formulaires sont faciles à remplir. Le reste des démarches est commun à mes 3 propositions citées ci-dessus. - Ensuite se rendre sur un site d'annonces légal comme le Parisien et faire publier une annonce légale. Compter dans les 130 euros environ. La mienne devrait être publiée demain vendredi. Conserver précieusement les attestations de publication de l'annonce légale. Ensuite avec **ce gros dossier** (3 copies des statuts tamponnés par les impôts, les formulaires M0 et TNS, un extrait acte de naissance, un extrait acte de mariage, attestation de non condamnation, attestation de publication de l'annonce légale, justificatif de domicile de moins de 3 mois comme EDF ou Free, photocopie de l'acte d'achat de notre appartement, chèque de 83,96 euros, un café, une boîte de tic-tac, un peu de musique sur votre iPod, votre meilleur sourire...) vous êtes prêt pour vous rendre au Greffe du Tribunal de Commerce pour effectuer le dépôt et donc la création de votre entreprise. **L'aventure de la journée** Arrivé ce matin à 09h10, c'est avec un grand plaisir que j'ai découvert la magnifique architecture post-soviétique du tribunal de commerce. Avec un peu d'attente (1h10) qui vous permet de profiter de la vie dans le hall du Tribunal de commerce, je passe enfin au guichet tant attendu. Derrière un hygiaphone qui me cache le doux visage de ce gentil fonctionnaire, commence la séance de torture. Car oui, après 1h10 ce n'est pas encore gagné. Non je ne suis pas un artisan, oui mon dossier est complet, non je n'ai pas de facture EDF car je n'en reçois qu'une fois par an,*

oui je sais qu'il faut parapher les pages des statuts, non je ne suis pas désagréable. Si j'étais désagréable je montrerais que l'hygiaphone conçu pour intercepter les postillons, n'arrête pas le passage de mots d'oiseaux, comme l'artisan taxi avant moi qui avait l'air vraiment en colère... Bref remettons nous dans le contexte : je suis en pleine semaine, l'activité est à son paroxysme, et il y a 6 guichets. Seul le guichet 2 fait les procédures d'enregistrement... C'est marqué dessus. Clairement quel dommage de ne pas avoir plus de moyen. Les impôts m'ont reçu parfaitement, accueil, sourire, fonctionnaires sympas et motivés. Les locaux sont en bon état, les gens semblent content de bosser là. Au greffe c'est comme si finalement l'URSS avait réussi à envahir Bobigny. Gris, moche, sale, déprimant. Le pauvre bougre derrière sa vitre doit se faire dégommer toute la journée... Enfin voilà. C'est fait le dossier est déposé. Reste maintenant à prévoir une complémentaire santé (110 eur par mois), une prévoyance (20 eur par mois) et une assurance pro (cher). J'ai trouvé un excellent comptable sur Paris qui travaille avec des freelances et des indépendants, n'hésitez pas à me contacter si vous souhaitez en savoir plus. Les inconvénients : beaucoup de démarches à effectuer, loin d'être simple finalement. Obligation de prendre un expert comptable car les déclarations sont trop compliquées (je pense). Les avantages : vous êtes le patron, vous pouvez ensuite vous verser un salaire raisonnable afin de limiter votre impôt sur le revenu, et réinvestir une partie de votre capital dans l'entreprise. Cela permet avec de savants montages fiscaux de ne presque plus payer d'impôt sur le revenu. En terme de crédibilité pour vos futurs clients, je vous le recommande. C'est un gage de sérieux, vous pouvez ensuite imaginer travailler directement avec un client final sans passer par une société de service. Attention cependant, moi je préfère travailler avec un cabinet de placement car eux m'apportent beaucoup plus de missions. Je vous en reparlerai plus tard. Dans la voiture en revenant j'ai entendu une bonne blague à propos de la Justice : "*Le métier de ministre n'attend pas : alors que Rachida Dati est à la Justice, son bébé est déjà à l'Interieur*"

## Mule 2 : présentation à JavaPolis 2007

**Voici un compte-rendu en français de la présentation de Mule 2 de Ross Mason, le CTO et cofondateur de MuleSource Inc. Vous pouvez retrouver la vidéo en anglais [à cette adresse](#). Bonne lecture.** Ross Mason a débuté le projet Mule en 2003. Avec un background en Finance et en Telco, Ross a débuté le projet Mule dans l'idée de créer un ESB open-source. Un ESB (Entreprise Service Bus) est un bus d'information, d'échange et de transformation de message. Mule remporte un succès car plus d'un million de téléchargement ont été effectués depuis la première version. 7 des 50 plus grosses entreprises dans le monde utilisent Mule. Il y a plus de 2500 développeurs sur les listes de diffusions, la communauté est donc active. 2000 applications sont en production avec Mule aujourd'hui. La définition de Ross de Mule est "de permettre l'intégration de systèmes hétérogènes dans un même environnement". Il s'agit de connecter différentes applications dont l'architecture est complètement différente. Un portail Web avec des sources de données. Des Services Adapters avec un bus de messagerie, etc. Mule est composé d'un diffuseur de messages (Message Broker), d'un repository pour la configuration, d'un système de routage et de diffusions de messages, d'un moteur transactionnel, d'un moteur de transformation et enfin d'un moteur de gestion de workflow. Mule peut donc recevoir un fichier, le transformer, diffuser via le routage des messages ou une partie de ce fichier à plusieurs applications, s'entourer d'une transaction pour réaliser le tout. La gestion de la sécurité est souple mais permet de gérer l'authentification et l'autorisation. **Pourquoi choisir Mule ?** Mule tout d'abord ne force pas un type de message. Que votre architecture se base sur l'échange de fichiers XML, CSV, d'objets sérialisés ou de flux binaires, le moteur de transformation de Mule permet d'ajouter ces différents formats. Au niveau du code, Mule n'est pas intrusif et n'impose pas d'API externes. Les objets sont complètement portables, ce qui facilite les tests. Le fait que Mule ne force pas le format des messages en fait un outil non compatible avec la spécification JBI de Java. Cependant je pense que c'est un avantage qui permet d'avoir un système plus ouvert. Renseignez-vous sur ce sujet, je n'ai pas plus creusé. Mule a été comparé à un couteau suisse. Il offre un grand nombre d'architectures de services. ESB est l'une des topologies, mais Mule peut très bien faire du Client-Serveur par exemple, en se basant sur des échanges synchrones, asynchrones de type message ou de type commande (rpc). Pour cette raison c'est une solution souple. Un service web de type requête-réponse synchrone peut ensuite faire suivre un message asynchrone à une Queue JMS par exemple. C'est le principe de Mule. L'intégration avec l'architecture J2EE est optionnelle. Le support de JBI, JMS, EJB, des connecteurs JCA, de JTA et des Servlets permet à Mule de s'intégrer dans une architecture J2EE existante sans problèmes. La partie routage des messages a été écrite en se basant sur le livre "[Enterprise Integration Pattern](#)" de Gregor Hohpe, Bobby Woolf et Kyle Brown. C'est un livre de 2004 qui a encore beaucoup d'idées d'architecture simple. La distribution transparente est une fonctionnalité de Mule qui permet de définir les règles d'échanges de messages dans la configuration. Ceci permet de faire des tests unitaires avec échanges de messages entre virtual machine, puis une fois en production d'utiliser un autre fichier de configuration avec des règles de distribution différentes. Très pratique dans le cadre des tests unitaires ou pour simuler (mock pattern) des interfaces clientes lorsque celles-ci ne sont pas disponibles. Pour la partie sécurité, Mule supporte JAAS ou Acegi pour l'autorisation. Mule est capable de gérer des objets existants, des POJOs ou des EJBs. Enfin Mule est réputé pour faciliter les tests et un framework complémentaire à JUnit facilite l'écriture des tests. **L'architecture composant de Mule** Une instance de Mule est composée d'un Routeur entrant (Inbound Router), d'un composant unitaire (UMO Service Component) et enfin d'un Routeur sortant (Outbound Router). Les routeurs sont connectés à différentes applications via des Channels. Un Channel peut être un client web service ou un fichier plat sur un répertoire par exemple. **Les nouveautés de Mule 2.0** La première nouveauté est un nouveau format de configuration XML basé sur des schémas, en lieu et place des anciennes DTD. Cela apporte pas mal de nouveautés pour la complétion de code dans IDEA IntelliJ par exemple. La configuration est plus simple et il n'est plus nécessaire de décrire toutes les classes. Un ensemble de XSD permet de simplifier la configuration des points d'entrées (endpoint). Il est possible d'associer un Endpoint avec une méthode Java, via un Proxy Java. Il y a une Registry de configuration et donc il n'y a plus de Mule Manager. La gestion des messages a été améliorée. Un nouveau type de transformation peut être appliqué automatiquement lorsqu'un type

de message est appelé. Mule peut prendre en charge certaines transformations simples. Le Streaming est maintenant implicite pour les endpoints. Mule est gérable via JMX, ce qui facilite la gestion. Mule est basé sur Spring pour sa partie configuration, cela se voit dès la lecture des fichiers de configuration. On retrouve l'idée de Spring 2.5. Son architecture est composée d'un Runtime Mule qui s'assure de la diffusion, de la gestion des Threads et des Transactions. La partie Transport supporte plus de 60 types différents. Cela inclus JMS, SOAP, Fichier plat, HTTP, FTP, SSH, JDBC, etc. La partie de routage permet l'échange de message. La partie transformation est capable d'effectuer un sacré nombre de tâches : XSLTC, CSV, Fix, ISO20022, Swift, objets Java... Il est possible d'appliquer un grand nombre de transformation. Mule peut enfin démarrer un petit conteneur web pour simplifier la configuration. **Mule 2 configuration** Mule 2 se base sur le système de gestion d'espace de nomage de Spring 2. Chaque module a son propre espace de nom. Spring Bean peut être aussi ajouté dans la configuration de Mule 2. Le support de Seasar (sorte de Spring japonais) est aussi possible tout comme Hivemind ou Picocontainer. La différence que j'ai vu entre Mule 1 et Mule 2 est que dans la première version de Mule, il fallait connaître le nom des classes Mule pour déclarer les connecteurs. Mule 2 se basant sur des espaces de noms XML comme Spring, vous pouvez déclarer directement un attribut xml correspondant au connecteur qui vous intéresse. De plus les propriétés de chaque connecteur sont différentes selon le nom du module et la validation est possible dès l'écriture de votre configuration Mule. Ensuite Ross présente un exemple simple de service Mule, qui permet de lire un message sur l'entrée standard pour créer un objet Order, puis ensuite d'appeler un service de gestion de la commande pour créer un objet ProcessedOrder. Je retiens que Mule apporte toute la colle qui permet de créer le chemin suivant : lecture sur la console d'un message, transformation du message en objet Order, appel de la méthode process d'un service, récupération de l'objet ProcessedOrder et transformation de cette objet vers une String pour afficher un message de résultat. Le rôle de Mule via la configuration est donc de créer tous ces liens et d'orchestrer les appels des objets entre eux. Zéro ligne de code pour vous. Ensuite un des intérêts ici est de montrer comment il est possible d'ajouter un nouveau point d'entrée sous la forme d'un service Web. Pour cela il déclare tout simplement une petite ligne dans le fichier de configuration, et via CXF un nouveau service web est automatiquement créé pour vous. Au lieu d'utiliser la console et l'entrée standard, vous pouvez maintenant appeler une méthode d'un service web pour refaire le même parcours. Si je résume : Mule permet d'ajouter via une ligne de configuration un nouveau point d'entrée à votre application. Il montre ensuite comment appeler un service de vérification de crédit durant la transaction. Lorsque la commande est créée, on souhaite effectuer un appel à un service HTTP distant capable de nous dire si la carte de crédit est créditrice ou non pour valider le paiement. Le service est modifié pour recevoir par injection un objet CreditChecker. Le code est vraiment simple et on ne voit aucunes informations de transport, il est impossible de voir qu'il s'agira d'un appel HTTP ou via un Web Service. Mule crée du code agnostique et la partie transport/communication est masquée de son côté. Cela facilite en effet les tests unitaires. Il montre ensuite un nouveau système d'extraction et de création des propriétés à l'exécution. L'exemple qu'il montre est l'envoi d'un email à un destinataire, l'email est extrait avec une expression XPath à l'exécution.

**Autres nouveautés** Mule 2 a été réécrit pour faciliter la gestion des clusters. Mule 2 est prêt aussi à fonctionner avec OSGi car cela permettra d'apporter la gestion de version d'OSGi à Mule, ainsi que le déploiement à chaud. La gestion de version permettrait de ne redéployer qu'une partie de votre application et de ne mettre à jour que ce qui est nécessaire. **La gestion du Streaming** Mule 2 est capable de gérer de gros volumes de données sans essayer de tout charger en mémoire. C'est indispensable par exemple lors de la génération de rapports financiers. Les données étant énormes, lorsque nous avons travaillé avec Kocktail chez Reuters je me souviens que cette partie était délicate. Si c'était à refaire aujourd'hui, je serai tenté de regarder si Mule ne peut pas aider et faciliter l'architecture de ce composant. Hasan si tu me lis... tu sais quoi faire. **La gestion des erreurs** La gestion des exceptions est importante car pour la gestion des connecteurs c'est ce qui apporte des règles de tolérance de pannes. Pour les exceptions des connecteurs (un service web distant est tombé...) il est possible de configurer dans Mule des règles de routage différente pour gérer les problèmes réseaux par exemple. JMX permet de surveiller un composant Mule afin de détecter des pannes, chose importante pour l'exploitation et les équipes de production (spéciale dédicace à Gik de l'INA). **Les transactions** A mon avis c'est la partie la plus délicate de Mule. Tout se configure via les fichiers de configuration. Mule supporte les transactions de type Local ou XA. Plusieurs serveurs d'applications sont supportés comme Weblogic ou JBoss. Le support d'Arjuna ou Atomika est aussi possible. Mule peut utiliser les gestionnaires de transaction des serveurs d'application comme Weblogic par exemple, en passant via JNDI pour retrouver le gestionnaire de transaction. **La montée en charge** Dans ce slide, Ross explique pourquoi Mule est scalable. Mule est basé sur le modèle **SEDA** (Straged Event Driven Architecture). Je fais une aparté pour parler de SEDA, ou de ce que j'en ai retenu en lisant le PDF. Le problème : comment gérer un grand nombre de requêtes dans un serveur ? Quelle est la meilleur architecture pour traiter 1000 requêtes à la fois ? Tout d'abord voyons l'architecture multi-threadée classique : Un serveur Web classique peut par exemple démarrer une thread par requête. Si cela semble une bonne idée au premier abord, charger une image du disque ou faire un appel dynamique à un EJB ne coûte pas la même énergie. Et ce modèle threadé est donc "bête" car il consomme des Threads pour des choses simples. Pour répondre à cela, est apparu les modèles orientés événement comme celui au coeur de Flash. A chaque événement système un ordonnanceur envoie des événements à un nombre limité de threads qui gèrent une machine à état. Imaginez la distribution de cartes dans une partie de Poket. Il y a 5 joueurs et le donneur donne des cartes. Certains joueurs vont terminer le boulot avant les autres (se coucher...) et d'autres vont travailler plus longtemps. Le souci de cette 2ème architecture de distribution est que l'on assume que la lecture des événements n'est pas bloquant. En clair, cela veut dire que lorsque je distribue les cartes, mes joueurs ne me bloquent pas et je passe au joueur suivant. Or c'est parfois difficile en vrai. De plus, si l'une de mes Threads prend 90% du CPU pendant que je distribue mes tâches, je suis impacté et toute l'application se met à ramer. Pour revenir à mon exemple avec le poker, la limitation de l'architecture orienté message est que lorsque l'ordonnanceur (le donneur) distribue les cartes il n'attend pas la réponse de chaque joueur et donc il va très vite, sans s'arrêter. Si ensuite il doit être bloqué par un joueur,

comme lorsqu'il demande ce que veut faire le joueur, alors le modèle est bloqué. Donc ce type de distribution des tâches fonctionne tant que les machines à état ne bloquent pas le scheduler. Pour cette raison, SEDA propose une 3ème architecture. Tout d'abord on ajoute la notion d'étape, pour que la gestion d'un événement soit effectué par petite étape très rapide. Ensuite on ajoute des queues pour effectuer la gestion des événements. Enfin SEDA fonctionne avec un système d'allocation dynamique des ressources afin d'ajuster à la charge le nombre de Threads. Tout ceci mis en place dans Mule promet donc une architecture mature et capable de monter en charge. La gestion des Threads est un cauchemar, je n'ai jamais vu une application parfaite en Java 1.4. Je me souviens d'une discussion avec un ami, Flavien, qui m'expliquait que son équipe avait écrit un moteur d'envoi de MMS massivement parallèle. L'implémentation initiale en Java se trainait à 100 MMS par seconde. Une nouvelle implémentation en C avec une architecture différente permet maintenant d'envoyer entre 2000 et 3000 MMS à la seconde... Pour Mule, le modèle SEDA est appliqué à JMS dans Mule ainsi que la notion de Spaces emprunté à Java Spaces. **Load balancing et Failover** Ross présente dans ce slide les capacités de répartition de charge et de gestion des pannes de Mule 1 et Mule 2. Mule 2.1 aura un support natif du clustering. **MuleForge** [MuleForge](#) est une communauté destinée aux utilisateurs de Mule pour partager des projets basés sur Mule. Par exemple des Transformers spéciaux, des connecteurs pour SAP, Salesforces.com, SIP ou SMS par exemple. Il y a déjà 80 projets hébergés sur ce site. Allez voir la page <http://www.muleforge.org/activeprojects.php> qui contient des connecteurs pour du Single Sign On, du LDAP, du SFTP, etc. **Mule HQ** Il s'agit d'une console de surveillance et d'administration qui permet via JMX de surveiller des instances Mule. C'est un peu le domaine sur lequel j'ai travaillé pour l'INA afin de surveiller et d'administrer des serveurs avec une application en Flex. Intéressant de voir ce que Mule propose, c'est une réelle demande des équipes d'exploitation et de production. Ensuite Ross présentera MuleIDE, un plugin pour Eclipse qui facilite l'écriture de modules Mule. **Conclusion** Voilà c'est tout pour cette transcription, j'espère que cela vous aura donné envie de tester Mule et d'y penser pour vos prochaines architectures. A bientôt. **Annexes**

- o Le site de Mule : <http://www.mulesource.com/>
- o Blog Octo : <http://blog.ippon.fr/tag/ross-mason>
- o [Livre Enterprise Integration Pattern](#) sur Amazon.fr

## Groovy

**Soirée Groovy au Paris JUG** Guillaume Laforge, chef de projet de Groovy, et Fabrice Robini Architecte chez [OCTO Technology](#) ont fait deux présentations très sympathiques hier soir à l'ISEP. Après une introduction à Groovy, la démonstration de Grails en 1h a achevé de convaincre le plus sceptique de la simplicité de Groovy et de Grails. **Qu'est-ce que le Paris JUG ?** Pour ceux qui ne connaissent pas [le Paris JUG](#), en quelques mots voici un résumé des derniers épisodes. Le Paris JUG (Java User Group) est un groupe de passionné de la plateforme Java qui se retrouve chaque deuxième mardi du mois à Paris. L'idée est de se retrouver pour partager et se rencontrer, tout en assistant à des présentations sur Java ou des domaines apparentés. Des sponsors (Xebia, Aneo, Valtech, BK Consulting, Zenika...) permettent d'aider et de financer chaque soirée afin que le concept reste gratuit. Vous pouvez retrouver [les anciens articles du touilleur sur le jug](#). Hier soir la soirée s'est déroulée dans [les locaux de l'ISEP](#), école en informatique et électronique. Le thème était [le langage Groovy](#). Pour une fois que nous avons la chance d'avoir un contributeur français majeur comme Guillaume, c'était l'occasion d'assister à une présentation plutôt technique. **Le Paris JUG a besoin de volontaires** L'équipe recherche des contributeurs pour participer aux blogs du Paris JUG, ainsi que pour effectuer des revues de livres. Si vous souhaitez participer et que vous vous sentez prêt pour écrire, profitez de la plateforme du JUG pour lancer votre Trifouilleur Express :-). **La présentation Groovy** Je vous parlerai plus de l'ambiance et des têtes que j'ai croisé, je ne vais pas refaire une présentation de Groovy sur ce billet, juste quelques lignes et un peu de code. Nous étions environ 80, avec la présence d'[Alexis Moussine-Pouchkine](#) au fond de la salle (monsieur Glassfish). Pas mal de monde sans battre le record de la soirée MDA/Flex. [Guillaume Laforge](#) que l'on ne présente plus, ou alors très vite (OSSGTP, Project Lead Groovy, SpecLeader de la JSR-251 et donc auteur de Groovy), a commencé par le HelloWorld classique sous la forme d'une classe simple. Premier point à retenir : **Groovy génère du byte-code Java**. Un fichier Groovy avec la syntaxe Java, compile et fonctionne sans problèmes avec une JVM, le résultat de l'interprétation c'est du byte-code. Il charge un fichier groovy, l'exécute, et un petit "Hello World" s'affiche. L'exercice ensuite consistera à alléger le fichier jusqu'à ce que le code soit 100% groovy. Il explique de cette façon plusieurs concepts du langage Groovy. Orienté objet, avec un support des closures plus simple que ce qui se prépare pour Java 7 ou 8, Groovy est orienté productivité et simplicité. Attention cependant, derrière cette simplification on peut penser que l'on retire des bequilles et une certaine forme de qualité à l'écriture du code. La simplification s'accompagne d'un déshabillage du code Java. Voici l'exemple de code avant/après afin de vous donner une idée.

```
/**
 * Simple HelloWorld
 */
public class HelloWorld {
    private String name;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void sayHello() {
        System.out.println("Hello "+getName());
    }
}
```

```

public static void main(String[] args) {
    HelloWorld hello=new HelloWorld();
    hello.setName("Nicolas");
    hello.sayHello();
}
}

```

Guillaume s'attaque à transformer ce fichier en programme Groovy en retirant tout ce qui n'est pas nécessaire. Groovy travaille par convention, l'ensemble de ce qui n'est pas précisé est implicite. Une class est publique, un attribut a un getter et un setter, etc. Voici le résultat en Groovy

```

class HelloWorld {
    def name
    def greet() { "Hello ${name}" }
}

def helloWorld = new HelloWorld()
helloWorld.name = "Nicolas"
println helloWorld.greet()

```

J'ai retrouvé la présentation de Guillaume sur Internet avec ses explications [sur dZone](#). Groovy est un langage dont la syntaxe est dérivée de Java, tout en étant plus simple. La présentation de Guillaume était claire quoiqu'un peu courte. Les questions du public : quid des performances ? Guillaume reconnaît qu'il y a un impact bien que Groovy soit le plus rapide des langages de scripts avancés de la future plateforme Java. Une autre question : quelle est sa préférence sur le chantier des Closures en Java 7/8 ? quelle serait sa préférence d'implémentation ? Il explique que les Closures pour Java apportent une complexité qui n'est pas adaptée à ce qu'est Java aujourd'hui. Groovy propose des closures implicites et faciles, dont la syntaxe reste simple. Il pense que les closures dans Java 7/8 apporteront trop de complexité, surtout sur la syntaxe, avis que je partage largement. **Grails, Grails, Grails** Fabrice d'OCTO prend ensuite les commandes après un buffet très sympathique organisé entre les 2 présentations. Au passage je croise [Dimitri](#) qui m'explique qu'il quitte Prima Solutions pour rejoindre [eXo Platform](#). Je croise Patrice un des développeurs d'eXo qui m'explique que l'équipe est constituée d'une dizaine de personnes en France. Un petit verre avec [Nicolas et Florent de Jaxio](#), ainsi que des personnes de Xebia et de Valtech et il est temps de reprendre la séance. Revenons à la présentation de Grails : simple, puissante et intéressante. En une heure Fabrice et Guillaume jouent une petite pièce de théâtre afin de nous proposer de développer une application Web de gestion de CD en quelques clics. On voit la simplicité et la puissance de Grails, et surtout un gain en productivité. La présentation a été effectué aux universités du SI je crois. Sur ce je vous laisse, boulot, boulot, boulot... Pas assez de



temps. Le monsieur avec la chemise blanche, les lunettes et le bouc au premier rang : c'est moi. Assis à côté de Nicolas Romanetti sur ma gauche, le cofondateur de Jaxio.

## Evénement Spring le 13 novembre 2008

Je fais suivre un email reçu ce soir de Julien Dubois : Le 13 novembre SpringSource et la société SFEIR organisent "**les rencontres Spring**" un événement à Paris à la Défense. Parmi les orateurs prévus:

- o [Juergen Hoeller](#) (SpringSource), co-fondateur et principal développeur du framework Spring, qui parlera de Spring 3.0
- o [Peter Cooper-Ellis](#) (SpringSource), senior vice-president of engineering de SpringSource, qui donnera des informations sur la roadmap des produits Spring
- o [Mark Thomas](#) (SpringSource), principal contributeur au projet Apache Tomcat, qui parlera de l'utilisation de Tomcat en production avec Spring

Rendez-vous sur le site de Spring afin de vous inscrire : <http://www.rencontres-spring.com/>. L'événement est gratuit et sera l'occasion de discuter autour d'une table ronde sur l'utilisation de Spring en entreprise. Merci Julien pour l'info :-)

## Revue de presse (agrume)

Je pique l'idée à mes voisins de Xebia : une mini-revue de presse sur les lectures de cette semaine et les choses remarquables de la communauté Java. Cela vous donne une petite radiographie de ce qui traîne dans ma tête en ce moment. Sur le blog d'Octo, j'ai lu [un article intéressant sur Comet](#). Comet est un principe de streaming utilisé pour pousser du serveur vers le navigateur des informations sans que le client web n'ait explicitement demandé ces informations. Après le polling HTTP, le streaming avec connexion persistante, c'est donc un 3ème moyen d'envoyer des données vers le navigateur. Comet ou "Reverse Ajax" est un mécanisme qui demande cependant une architecture de Thread particulier sur le serveur. En effet on n'imagine pas conserver une thread par client connecté, il faut un algo de répartition des tâches afin d'éviter d'écrouler le serveur. Intéressant comme article. Les geeks de [JavaPosse](#) ont parlé longuement de la sortie du navigateur de Google, à savoir [Google Chrome](#). Ce que vous devez retenir : chaque onglet de navigation fonctionne dans son propre processus. Lorsqu'un site prend trop de ressources, vous pouvez "tuer" le processus dans le gestionnaire de tâche. La version Mac et Linux build sur l'intégration continue mais Google n'a pas encore sorti ces 2 versions. Erwan Alliaume a résumé bien mieux que moi la soirée du Paris JUG de mardi dernier, spéciale dédicace au Trifouilleur Express. Allez lire [son compte rendu](#) sur la soirée Grooovoy sur le blog de Xebia. La prochaine présentation du Paris JUG aura pour thème "OSGI". La soirée se déroulera [le 14 octobre](#), animée par Cyrille Le Clerc et Nicolas Griso de Xebia. A propos d'authentification, j'ai lu un petit article intéressant qui propose "[5 fonctionnalités que votre page d'authentification doit avoir](#)". Voici cette liste traduite :

- o Une fonction "Se souvenir de moi" basée sur des cookies : cela évite de taper son mot de passe à chaque fois
- o Faire apparaître de façon claire le lien "Login" afin que l'utilisateur s'authentifie facilement sans trop devoir chercher sur la page
- o La redirection après l'expiration de la session : vous lisez une section protégée, vous partez déjeuner et en revenant votre session a expiré. Idéalement après avoir passé la page de réauthentification, le système doit vous remettre sur la même page.
- o "J'ai oublié mon mot de passe" : donnez à l'utilisateur un système rapide et simple pour retrouver son mot de passe
- o Eviter le javascript sur la partie d'authentification. Cela réduit l'accessibilité.

Ensuite dans l'air du temps : [une petite comparaison de GWT et de Wicket](#). L'article est vraiment très complet, quoiqu'un peu biaisé. A lire si vous connaissez Wicket. Sur le thème du Grid computing, au boulot hier Nicolas Griso (on est 3 Nicolas sur 12 personnes dans ma nouvelle équipe) me parlait d'[Hadoop](#). Il s'agit d'un projet open-source d'implémentation de l'algorithme de Map-Reduce destiné à traiter de très gros volumes de données en cluster. Ce projet Java est une sorte de plateforme de distribution et de réconciliation de tâches. La complexité lorsque l'on effectue du traitement de Batches, donc pour effectuer des traitements séquentiels, c'est que la réconciliation dans la phase de réduction est parfois complexe à gérer. Pour cette raison, Chris Wensel a eu l'idée d'apporter une couche applicative simple fonctionnant sur Hadoop et facilitant l'écriture d'applications : le projet [Cascading](#). Cascading dispose d'un scheduler topologique pour piloter l'enchaînement de tâches, d'un système de notification pour être notifié lorsqu'une tâche est terminée, sans qu'il ne soit nécessaire de passer par la base de données. Il est possible d'ajouter un moteur d'assertion simple pour effectuer de la validation bas niveau au fil de l'eau, et enfin le support de Groovy permet d'écrire des scripts facilement. Le site est vraiment complet, les "[Cook Book](#)" donnent une idée de l'intérêt de cette API par dessus Hadoop. Enfin j'ai commencé à coder un petit proto avec GigaSpaces... wow que c'est bien. Cela me rappelle mes années chez Dotvision avec Guillaume Pelletier : il avait déjà raison en parlant de JavaSpace à l'époque. J'avais même écrit à l'époque un bout de code de partage de space distribué fonctionnant avec du multicast, c'était fun. [GigaSpaces](#) c'est un gros truc, le genre de projet vraiment intéressant qu'il faut expliquer avec un grand billet, des schémas et du code. Pour cette raison, je m'arrête là ce soir et j'en garde sous le pied pour plus tard.

## Apple m'a tueR

"Ah ah ah" c'est moi devant la boutique d'Orange samedi, en me disant qu'il n'y a aucune chance que je craque pour acheter l'iphone 3G. Oui soyons réaliste : à part le mail, qu'est-ce que je pourrai bien faire d'un iphone ? Franchement ? si seulement c'était si simple... J'ai ensuite passé un moment agréable avec le vendeur. Ils sont forts ces gens quand même. J'étais admiratif des efforts commerciaux pour me faire acheter "le précieux". Heureusement que je n'avais pas pris de RIB, un cheque annulé, un extrait de KBis et ma pièce d'identité. Enfin disons que ce samedi il se trouve que j'avais ces pièces puisqu'ensuite j'allais à la banque. Si si je vous assure. La démo de l'iPhone achève de convertir n'importe quel informaticien normalement constitué. C'est vrai qu'à ce moment là j'étais déjà assis dans le magasin et raisonnablement il devenait clair que j'allais craquer. Mais bon, soyons raisonnable, à part ces 342 trucs géniaux je ne vois pas ce que je pourrai en faire. Grave erreur. Je vous le dis : j'ai craqué. Moment de faiblesse. Faux scoop bien entendu pour ceux qui lisent le blog. Avec ce qu'Apple m'a ramassé ces derniers mois (imac, ipod, mac book pro) je crois que je vais ouvrir un Apple Store chez nous. L'installation et la configuration s'effectuent comme toujours chez Apple en quelques instants. Le nombre d'idées géniales et ergonomiques de l'interface... c'est difficile à expliquer en quelques lignes. Certes l'iPhone ne fait pas tout : MMS. Mais ce qu'il fait par ailleurs, il le fait très bien. Par exemple vous avez une fiche Contact avec le numéro de téléphone, l'email et l'adresse d'un ami. Un clic sur l'email et vous pouvez lui envoyer un message. Un clic sur le téléphone et vous l'appellez et un clic sur l'adresse, le GPS s'active avec Google Maps, et vous pouvez vous rendre chez lui par navigation. Classe non ? Je l'ai configuré avec ma boîte email par IMAP, les pièces jointes se lisent toutes sans problèmes, les fichiers Excel comme PDF. La configuration avec iTunes me permettra de récupérer mes podcasts automatiquement le matin ou dans le RER. Je peux même administrer le Touilleur via Safari et donc, valider les commentaires pendant la journée. C'est essentiellement la partie Internet et Mail qui est vraiment bien faite. C'est bien simple : n'importe quel site s'affiche sans problèmes et très rapidement. Je n'ai fait fonctionner mon iphone qu'en mode Wifi pour l'instant, l'activation n'étant pas encore effective. Lundi je serai donc raisonnable : je laisse l'appareil à la maison et on attendra

que le portage de Bouygues vers Orange soit effectif. ... sont forts ces vendeurs... mais bon, je suis un bon client. Apple m'a tueR

## Rencontre avec Julien Dubois, responsable de SpringSource France

J'ai eu le plaisir d'interviewer Julien Dubois, le responsable de SpringSource pour la France. SpringSource est la société qui a créé et qui développe le framework libre Spring. C'était l'occasion de faire un point avec lui sur les nouveautés prévues jusqu'à la fin de cette année et de parler des événements en France autour de la communauté Spring.



*Saviez-vous que pour la seule journée du 16 septembre il y a eu 3740 téléchargements de Spring sur SourceForge.Net ? ( ) Cela représente près d'1,3 millions de téléchargements par an, sans compter l'activité des repositories Maven... Il est clair que Spring est l'un des projets les plus actifs de la communauté open-source Java. Aucune offre d'emploi sans que Spring ne soit mentionné, c'est devenu un standard de facto pour tout développeur Java.*

**Nicolas : En quelques mots, quelle est l'activité de SpringSource en France ? Comment avez-vous démarré l'activité en France ?** **Julien :** Spring Source France a démarré au début de cette année. Nous proposons des formations complètes sur Spring et les différents modules du framework. Nous travaillons aussi avec la Belgique, la Suisse, le Maroc, bref l'ensemble des pays francophones. Nous travaillons ensuite avec un ensemble de partenaires (Accenture, Steria, Ippon, Zenika, SFEIR, Xebia, Jaxio...) qui offrent des prestations de conseil autour de Spring. Nous formons et certifions nos partenaires sur nos technologies ce qui permet d'offrir des prestations de conseils. Nous proposons un support d'excellente qualité puisque bien souvent, les développeurs de Spring eux-même assurent le support. Enfin nous sommes actifs dans la communauté open-source francophone, avec des participations régulières au [Paris JUG](#), par exemple. **N: Comment a commencé l'aventure SpringSource pour toi ?** **J:** En 2006 j'ai coécrit le livre "[Spring par la pratique](#)" avec Jean-Philippe Retailé et Thierry Templier. Début 2008 l'équipe de Rod Johnson m'a contacté afin de me proposer d'ouvrir l'entité française de SpringSource. J'ai dit oui. Et depuis ce jour, j'ai un emploi du temps vraiment chargé, tout comme l'ensemble des personnes qui travaillent avec nous. **N: j'ai vu que Neelan Choski, votre COO a quitté SpringSource après 2 années intensives. On parle de quelqu'un de JBoss qui viendrait vous rejoindre pour le poste de COO. Est-ce que tu confirmes ?** **J:** Concernant Neelan, je te confirme qu'il a quitté le poste de COO, mais je n'ai pas plus d'infos pour le moment. C'est quelqu'un de très brillant, que je connais depuis des années, je ne me fais pas de soucis pour lui. Pour notre nouveau COO, en effet il y aura une annonce très prochainement dans ce sens. SpringSource est déjà constituée en partie par quelques anciens de JBoss, de BEA et d'IBM. Ce ne serait donc pas une surprise qu'il vienne de l'une de ces sociétés. **N: Parlons un peu de S2AP, quoi de neuf ?** **J:** Spring Server Application Platform (S2AP) est constitué de 2 éléments : SpringSource Enterprise et SpringSource dm Server. SpringSource Enterprise est constitué du framework Spring, d'outils de tests et de surveillance de performance et d'un package pour le support. SpringSource dm Server est un serveur d'application Java modulaire, basé sur SpringSource Dynamic Module Kernel(tm). Je t'annonce que la version GA sera prête d'ici à la première semaine d'octobre. Distribué sous licence GPL v3, nous offrons aussi la possibilité d'acquérir une licence classique pour les institutionnels. Comme je l'avais dit au [Paris JUG](#) une application web hébergée sur dm Server ne devient pas GPL de facto. **N : Où en êtes-vous concernant les dates de sortie de S2AP SpringSource dm Server** **J:** Nous avons annoncé il y a une semaine la disponibilité de la version 1.0.0.RC2, il est possible de la télécharger après s'être enregistré sur le site [SpringSource.com](#). **N: Par rapport à la spécification EJB 3.1, comment se situe S2APdm Server ?** **J:** Très proche, Rod Johnson a annoncé que dm Server devrait être compatible avec les profils A et B de la spécification JEE 6. Par contre pas avec le profil C, c'est à dire les vieux EJB 2.1, entre autres. Qui voudrait faire tourner des Entity Beans en 2008 avec dm Server? SpringSource dm Server devrait interpréter les annotations EJB 3.1, ainsi que les annotations Spring bien évidemment. SpringSource dm Server apportera aussi la possibilité d'avoir un serveur d'application sans conteneur web par exemple, l'intégration d'OSGi, la gestion par module... **N: Concernant Spring 3.0, as-tu des informations supplémentaires pour les développeurs qui suivent l'actualité de Spring ?** Juergen Hoeller (cofondateur de Spring) a expliqué que Spring 2.5 avec le support des annotations est une version majeure pour les utilisateurs. Par exemple un bean métier est marqué @Service, un DAO est marqué @Repository, et toute la partie MVC est aussi considérablement simplifiée grâce aux annotations.. La version 3, c'est une refonte du noyau interne en JDK 5 : en tant qu'utilisateur cela ne vous changera pas grand chose, mais pour nous c'est vraiment un travail important. Sinon, il y aura quand même quelques nouveautés, la plus importante je crois étant le support de REST. Si cela vous intéresse, Juergen fera une présentation des nouveautés Spring 3.0 aux prochaines « [Rencontres Spring](#) ». **N: SpringSource propose des outils pour les équipes de production afin de surveiller des serveurs, en quoi cela consiste-t-il ?** SpringSource c'est aussi des consultants expérimentés qui ont travaillé en équipe de Production. Or aujourd'hui, l'exploitation et la production des plateformes Java sont des activités complexes. En plus de ce que Java offre via JMX, qui te permet de surveiller la mémoire et l'activité du Garbage Collector, nos outils de monitoring et de surveillance sont orientés métiers. **N : C'est-à-dire ?** Il est possible avec SpringSource Application Management ServiceSuite (AMS) d'auditer une application par introspection et par aspects. Typiquement, tu peux te créer une métrique sur AMS afin de suivre le nombre de paniers achetés sur ton site marchand ou le nombre de batchs traités par ton application de backoffice dans la finance. **N : Et dans le domaine du Grid computing ? C'est l'un des sujets sur lesquels je travaille chez mon client en ce moment** SpringSource est partenaire de sociétés telles que Terracotta et GigaSpaces. Nous sommes d'ailleurs entrain de voir pour établir un nouveau partenariat avec [Fast Connect](#), principal intégrateur de la solution GigaSpaces en France. Spring fait souvent partie de la solution de départ. Ensuite en terme d'infrastructure de calculs en grille, des produits comme

Terracota et GigaSpaces complètent ces choix. Allez voir [le site de FastConnect](#) pour plus d'informations: **N: Et ton actualité pour les jours qui viennent ?** Je vais animer plusieurs formations, donc je vais faire relativement peu d'événements publics par rapport à d'habitude. Cependant voici notre agenda : Jeudi 18 : Google Developer Days, Semaine du 24 : nous effectuons des formations. Mardi 30 septembre : [BarCamp Java](#) chez Octo Technologies Jeudi 2 Octobre : [Soirée Terracotta chez Zenika](#) Jeudi 16 Octobre : [Soirée Spring au YaJUG](#), le JUG du Luxembourg Jeudi 23 Octobre : Petit Déjeuner « Spring » à Lille, sponsorisé par la société [Proxiad](#). Avis à nos voisins Belges! Je profite de l'occasion pour annoncer l'événement que nous organisons avec Didier Girard (NDLR : l'auteur du blog [Application Servers.com](#), Directeur Technique de SFEIR). Le 13 novembre aura lieu « [Les Rencontres Spring](#) » à Paris. C'est **un événement à ne pas manquer** avec la venue de Juergen Hoeller, de Peter Cooper-Elis et de Mark Thomas. Voir notre blog pour plus d'informations sur tous ces événements : <http://www.springsource.com/fr/blog> **N : Merci pour ton temps et bonne continuation ! J :** Merci à toi et à bientôt, au Paris JUG ou à un événement Spring !

## Un Cache en 2008

Un cache. Nous savons tous à quoi sert un cache d'objet. Réduire les temps d'accès à un tiers plus lent en conservant une copie temporaire dans un endroit indexé. La taille d'un cache doit être limitée afin de ne pas utiliser trop de mémoire. La recherche dans un cache doit être très rapide. Idéalement, un cache ne doit conserver que les données les plus souvent accédées. Il faut donc mettre en place un algorithme de tri afin que les données qui sont peu utilisées soient effacées. Curieusement, à chaque fois que je change de boulot j'ai l'occasion d'implémenter un Cache. Je vous propose de passer en revue ici rapidement quelques implémentations simples de cache en Java. **Le cache pour les Nuls** Forcément, mon premier cache qui remonte à 2000 était à l'époque un peu limité. Le besoin était de stocker des objets Java représentant des données géographiques. A l'époque j'avais implémenté cela avec une HashMap toute simple. Je vous laisse profiter du paysage :

```
/*
 * Copyright(c) 2008 Nicolas Martignole - Le Touilleur Express
 * http://touilleur-express.fr
 * Distributed under Creative Commons License 2.0
 * See http://creativecommons.org/licenses/by-sa/2.0/fr/deed.fr
 */
```

```
/**
 * A very simple cache that drop the oldest Entry.
 * http://touilleur-express.fr
 * @author Nicolas Martignole
 */
public class FIFOCache {
    private HashMap cache;
    private LinkedList indexes;
    private final int cacheSize;

    public FIFOCache(int cacheSize){
        this.cacheSize=cacheSize;
        cache=new HashMap(cacheSize); // initialCapacity
        indexes=new LinkedList();
    }

    public synchronized void cache(String key, Object o){
        if(cache.size() >= cacheSize){
            // remove the oldest element
            String keyToRemove=(String)indexes.removeLast();
            cache.remove(keyToRemove);
        }
        indexes.addFirst(key);
        cache.put(key,o);
    }

    public synchronized Object get(Object o){
        return cache.get(o);
    }
}
```

Cette implémentation est ultra simpliste : lorsque le cache est plein, on se contente d'effacer l'entrée la plus ancienne. Pour cela on stocke dans une liste chaînée la clé. L'avantage ? Pour un cache utilisé en lecture, c'est efficace. Notez que nous aurions même dû utiliser une Queue, tout simplement. Mais bon, c'était à l'époque. **Cache II, le retour** Un peu plus tard, chez Reuters, je crois avoir ensuite écrit un cache avec cette fois-ci un algorithme de LRU, Least Recently Used. L'idée est de n'effacer que l'entrée la moins accédée dans le cache. Pour que cela fonctionne, il faut donc réordonner la liste chaînée. Lorsqu'un élément existe dans le cache, avant de le retourner nous déplaçons sa clé au début de la liste chaînée. Les entrées les moins accédées se trouvent à la fin de la liste, alors que les entrées les plus accédées sont au début de la liste. Lorsque le cache est plein, on efface la dernière entrée de la liste pour pouvoir insérer à nouveau une entrée. Voici le bout de code, à peu de chose prêt :

```
/*
 * Copyright(c) 2008 Nicolas Martignole - Le Touilleur Express
 * http://touilleur-express.fr
 * Distributed under Creative Commons License 2.0
 * See http://creativecommons.org/licenses/by-sa/2.0/fr/deed.fr
 */

/**
 * A cache that implements the Least Recently Used algorithm, using a LinkedList.
```

```

*
* http://touilleur-express.fr
* @author Nicolas Martignole
*/
public class LRUSimpleCache {
    private HashMap cache;
    private LinkedList indexes;
    private final int cacheSize;

    public LRUSimpleCache(int cacheSize) {
        this.cacheSize = cacheSize;
        cache = new HashMap(cacheSize); // initialCapacity
        indexes = new LinkedList();
    }

    public synchronized void cache(String key, Object o) {
        if (cache.size() >= cacheSize) {
            // remove the oldest element
            String keyToRemove = (String) indexes.removeLast();
            cache.remove(keyToRemove);
        }
        indexes.addFirst(key);
        cache.put(key, o);
    }

    public synchronized Object get(String key) {
        Object cached = cache.get(key);
        if (cached == null) return null;
        if (indexes.getFirst().equals(key)) {
            return cached;
        }
        indexes.remove(key);
        indexes.addFirst(key);
        return cached;
    }
}

```

L'idée est que lorsque l'entrée existe bien dans la Map, on déplace sa clé au début de notre table d'index. Vous allez me demander : pourquoi ne pas faire la recherche sur la liste chaînée d'index au lieu de faire un `map.get(key)` ? Simplement parce que le temps de recherche dans une hashmap tend vers  $O(1)$  là où le temps de recherche dans une liste chaînée augmente de manière linéaire ( $O(n)$ ). C'est vrai en règle générale, la recherche dans une HashMap sera plus rapide que dans une LinkedList. C'est évidemment faux si vous avez 5 entrées dans votre cache. Jusqu'ici, rien d'extraordinaire. Simplement, les accès coûtent chers lorsque le cache est plein. **FBI Fausse Bonne Idée** Ensuite j'ai pensé m'inspirer du Garbage Collector. L'idée est la suivante : on crée 2 Maps. Une Map destinée à recevoir les objets jeunes, une Map destinée à recevoir les objets qui ont été demandés de nombreuses fois. Lorsque le cache est plein, on détruit la Map « jeune » en déplaçant vers la Map ancienne les objets fréquemment demandés. Lorsque le cache est utilisé, on recherche d'abord dans la Map des « anciens » avant de chercher dans la Map des petits nouveaux. Le problème est de mettre en place une politique pour vider la Map ancienne. Pour cela on peut reprendre l'idée du FIFO ou même ajouter un peu de LRU... Bref c'est faisable mais un peu compliqué. **Cache III la revanche** Et puis ensuite, mission à la BNP. Même souci : un cache constitué de 2 Hashtables a un souci de synchronisation. Plutôt que de fixer le code, ce qui était possible, j'en ai profité pour regarder aujourd'hui comment je pourrai implémenter toujours le même cache le plus simplement du monde. Tout d'abord le premier réflexe : consulter l'API de Java. Est-ce que cela vaut le coup de réinventer la roue ? Je me suis dit aussi : prenons une API open-source de cache et hop c'est réglé. Mais pour notre besoin, une seule classe est nécessaire. Pour ma nouvelle implémentation je suis donc parti de la class `LinkedHashMap`. Cette class qui étend `HashMap` ajoute une liste ordonnée afin que l'ordre d'enregistrement des paires clés/valeurs soit prédictif. En clair : le même code que ce que je vous ai montré mais disponible dans l'API Java. La classe `LinkedHashMap` fournit un temps constant d'accès comme toutes les `HashMaps` pour les méthodes `add`, `contains` et `remove`, ce qui en fait donc une classe parfaite pour le cache. Un constructeur spécial permet de créer une `LinkedHashMap` dont l'ordre d'itération correspond à l'ordre d'accès des entrées de la Map, de la donnée la plus accédée en premier à la donnée la moins accédée en dernier. Soit exactement l'algorithme LRU dont je vous parlais plus haut. A chaque fois que vous allez utiliser la méthode `get()` de cette Map, l'entrée accédée est promue au début de la liste. Ainsi un hit cache entraîne un tri du cache. Petit à petit, les entrées du cache qui ne sont pas accédées se retrouvent à la fin de votre cache. C'est là qu'intervient une méthode à surcharger afin de décider de la politique de nettoyage. La méthode `removeEldestEntry(Map.Entry e)` est appelée par l'API à chaque insertion. A vous d'implémenter un bout de code qui retourne true lorsque vous souhaitez effacer l'entrée la plus ancienne de la Map. Pour ma part je me contente de regarder si la taille de la `LinkedHashMap` est supérieure à ma taille configurée lors de la création de l'objet. Dans ce cas, l'API effectue le ménage et retire l'entrée la plus ancienne. A cela j'ai ajouté l'utilisation des Generics. En effet, il est dommage en Java 5 de ne pas s'en servir, cela évite des typages à ajouter dans le code partout. Enfin comme je suis ami avec la mémoire, j'ai décidé de décorer les valeurs cachées d'une `SoftReference`. L'idée est la suivante : lorsque dans la JVM la mémoire vient à manquer, autant faire le ménage dans le cache et donc, effacer toutes les entrées afin de laisser respirer l'application. Pour faire cela, le mieux est donc de simplement décorer les valeurs enregistrées dans la `LinkedHashMap` avec une `SoftReference`, ce qui n'est pas trop agressif et permet de conserver les données en cache. Une `WeakReference` sera trop faible et dès l'activation du GarbageCollector, votre cache sera vidé. Une `SoftReference` d'après mes tests, ne sera détruite que si tout d'abord la donnée pointée n'est plus disponible ou en cours d'utilisation, et lorsque le GC se réveille pour faire un gros nettoyage. Pour tester cela, j'ai fait un test unitaire avec une petite boucle infinie. Sans `SoftReference`, une exception `OutOfMemory` est levée au bout de 50 objets stockés. Avec une `SoftReference`, le test tourne sans fin. Un test avec `YourKit` montre l'activité du GarbageCollector, et que donc le cache se nettoie bien tout seul. Voilà j'espère que ce code fonctionnera correctement et longtemps, et que donc à

mon prochain job je n'aurai pas à nouveau besoin d'écrire un cache.

```

/**
 * Copyright(c) 2008 Nicolas Martignole - Le Touilleur Express
 * http://touilleur-express.fr
 * Distributed under Creative Commons License 2.0
 * See http://creativecommons.org/licenses/by-sa/2.0/fr/deed.fr
 */

/**
 * New LRU Cache constructed with a LinkedHashMap, a Map with a predictable iteration order that
 * stores a key/value pair, the value is decorated with a SoftReference so that we can release
 * memory when the system is overloaded.
 * I also use a special constructor from LinkedHashMap so that I can have an order in this map
 * provided by access
 *
 * @author Nicolas Martignole
 * @version created Sep 16, 2008
 */
public class NewLRUCache<K, V> {
    private LinkedHashMap<K, SoftReference<V>> cache;
    private int cacheSize;
    private int numberOfElements = 0;

    /**
     * Creates a cache with a fixed size of 1000.
     */
    public NewLRUCache() {
        this(1000);
    }

    /**
     * Create a new LRU Cache with a LinkedHashMap, override the removeEldestEntry
     * so that we can inject the current cacheSize.
     *
     * @param cacheSize is a positive integer.
     */
    public NewLRUCache(int cacheSize) {
        this.cacheSize = (cacheSize < 1) ? 1000 : cacheSize;
        int initialCapacity = (int) (cacheSize * 0.75);
        cache = new LinkedHashMap<K, SoftReference<V>>(initialCapacity, 0.75f, true) {
            @Override
            /**
             * Returns true if the current map size is greater than NewLRUCache, wich means
             * that the cache is full and we should drop the oldest entry.
             */
            protected boolean removeEldestEntry(Map.Entry<K, SoftReference<V>> eldest) {
                return size() > NewLRUCache.this.cacheSize;
            }
        };
    }

    /**
     * Returns the cache size.
     *
     * @return the cache size.
     */
    public final int getSize() {
        return cacheSize;
    }

    /**
     * Returns the number of elements currently stored into the cache.
     *
     * @return a number of elements.
     */
    public final int getCurrentUsage() {
        return numberOfElements;
    }

    /**
     * Stores into the cache the specified entry, remove the less recently used entry
     * if the cache is full.
     *
     * @param key is the new unique key to store.
     * @param entry is the object to put into the cache.
     */
    public void cacheEntry(K key, V entry) {
        if (key == null) return;
        if (entry == null) return;
        synchronized (this) {
            cache.put(key, new SoftReference<V>(entry));
            numberOfElements++;
        }
    }

    /**
     * Lookup for the specified key, update the list of less recently used items.
     *
     * @param key is the key to lookup
     * @return the associated object or null if it was not found.
     */
    public V getEntry(K key) {
        if (key == null) {
            return null;
        }
        SoftReference<V> ref;

```

```

synchronized (this) {
    ref = cache.get(key);
    if (ref == null) {
        // The value has been garbage collected so we must delete the key
        cache.remove(key);
        numberOfElements--;
        return null;
    }
}
return ref.get();
}
}

```

Références : Javadoc de [LinkedHashMap](#) Javadoc de [SoftReference](#)

## BarCamp Java ce soir

Rendez-vous ce soir à Paris chez Octo pour participer au premier [BarCamp Java](#). L'idée du BarCamp : pas de spectateurs, que des acteurs. Chacun vient avec 3 sujets qu'il aimerait traiter, discuter. La séance commence par un tour de salle rapide, chacun se présente par son nom, sa société et donne 3 mots clés que l'on marque sur une feuille. Ensuite tout le monde se sépare par petit groupe afin de profiter de l'heure pour parler sur un sujet sélectionné. Nous verrons bien ce que cela va donner.

## Compte-rendu du BarCamp Java

Un peu au dernier moment sur une suggestion de Florent [de Jaxio](#), je suis allé au premier [BarCamp](#) organisé [entre autre par Luc Bizeul d'OCTO Technology](#), dans leurs locaux sur les Champs-Élysées. Organisation excellente de l'équipe d'OCTO, environ 50 personnes étaient présentes. Parmi les têtes connues de la communauté Java : Antonio Goncalves (Paris JUG), Alexis Moussine-Pouchkine, Eric Mignot Scrum Master chez Pyxis, Jean-Laurent de Morlhon que j'avais croisé chez Reuters [il y a longtemps](#), bref beaucoup de monde présent pour cette première. Après un tour de salle, où chacun se présente par son nom, son entreprise et 3 tags qui l'intéressent, nous avons tous rempli des post-its afin de préparer un tableau et de répartir par salle les discussions. Difficile de choisir un sujet sur un thème précis. Les thèmes les plus demandés : TDD,



Spring et JEE. [touilleur-express.fr](#) **Spring vs JEE ? JEE est-il mort ?** Pour ma part j'ai décidé de suivre en premier la discussion "Spring vs JEE" très intéressante, animée entre autre par Antonio. Dans une salle d'une dizaine de personne, nous avons bien entendu parlé de l'actualité de SpringSource concernant la politique de licence. Pour Julien : les discussions ont été très constructives, la majorité des personnes présentes partagent la volonté de SpringSource d'offrir un service de qualité et de ne pas devoir maintenir un nombre incalculable de version. Finalement c'était une petite tempête dans un verre d'eau. Les discussions les plus intéressantes ont été amenées par une personne, développeur Python chevronné (désolé je n'ai pas pris ton nom) qui posait des questions autour de Spring par rapport à JEE. Antonio a apporté son expérience sur JEE 5/6 et le travail de la JSR afin d'expliquer les arbitrages lors de la sélection de ce qui doit être un standard par rapport à ce qui ne doit pas l'être. Ainsi les fonctions de clustering, de cloud-computing et de cache ne font pas parties de la spec JEE 6. Par contre, Spring a dynamisé le travail sur JEE 5, Antonio était interrogatif (si c'est le mot) sur l'annonce de Rod Johnson de supporter le profil A et le profil B pour JEE6. Le profil A : servlets/jsp. Le profil B : on ajoute EJB3.1. Comme il l'explique, Spring va apporter son support sur cette partie. Le profil C contient 32 spécifications, dont le support des EJB2.1, plus pour des raisons historiques. Une question : pourquoi Spring est aussi dynamique ? pourquoi cela marche ? A cela, Florent Ramière répond tout simplement : ce sont les meilleurs développeurs, qui forment une équipe soudée de qualité. De plus, une autre personne ajoute que sans le devoir de formalisme de devoir faire une spécification comme JEE, Spring est forcément plus dynamique. Est-ce que trop de Spring tuera Spring ? Quelqu'un parle de l'idée d'un projet en trop un jour dans le panier de Spring qui risquerait de les décrédibiliser. Est-ce que Spring avec S2AP ne joue pas la carte du solo player ? On reparle de l'idée de travailler avec un standard pour un client institutionnel par rapport à travailler avec un seul vendeur, SpringSource. Les discussions fusent car les avis sont partagés. En fait je vois que pas mal de monde méconnaît l'activité de SpringSource, et tout est un peu mélangé. Faut-il attendre JEE6 ou pas ? Question un peu vague, qui n'a pas de réponse toute faite. Antonio explique que des avancées majeures comme WebBeans, venant de JBoss Seam, ou même JSF2.0 feront de la future spec de Java un socle solide et standard. Comme dirait Alexis de SUN : les standards, y'a que cela de vrai (il bosse pour SUN). Globalement j'ai trouvé la discussion vraiment intéressante, chacun essayant d'apporter son point de vue sur le sujet. Je pense que Spring est devenu un standard de facto, et que comme tout excellente plateforme, peut-être que les jaloux ont décidé de taper dessus histoire

de s'occuper. Mais vue la qualité de l'équipe, je pense que tout ceci va retomber. De mon côté j'ai expliqué à tout le monde le positionnement de SpringSource tel que Julien [me l'a expliqué](#). J'en ai profité aussi pour parler de la production afin que les développeurs n'oublient pas que ce qu'ils codent, doit être administrable. Pour cela j'ai parlé de la démo des produits de SpringSource, en défendant qu'il y a un sens à ce que ce soit des outils payants. Le public visé n'est pas le même que le framework Spring. Enfin tout le monde a parlé d'IOC, le mot pour définir Spring, la puissance de Spring et comme dit Antonio, quelque chose que JEE n'a pas à ce niveau. **Scrum dans la vraie vie** Pour le coup j'étais content, c'est l'un de mes petits post-its :-). J'ai retrouvé entre autre Eric Mignot de Pyxis, excellent orateur, ainsi que Jean-Laurent de Morlhon. Autour de la table, une dizaine de personne, mon voisin Olivier pose des questions pointues sur Scrum, François s'interroge sur comment facturer un projet développé avec Scrum, mon voisin en face explique Scrum dans son équipe, il travaille pour une administration, etc. Mais avant cela, Antonio qui ne connaissait pas Scrum a joué le rôle de candide. Exercice donc en premier : expliquer et définir Scrum. Pourquoi on en parle ? A quoi cela sert-il ? Pour quel public ? Des questions tournent sur les projets traditionnels, en particulier pour les marchés publiques. Est-ce que Scrum peut s'envisager lorsqu'il faut répondre par un cahier des charges et un engagement fixe ? J'avoue avoir pas mal parlé, passionné par Scrum. Mais indéniablement, Eric a fait passer son expérience de coach Agile, avec une gymnastique bien rodée pour expliquer et répondre aux questions phares sur Scrum. **Le buffet** Le buffet après 2 heures de discussions était le bienvenue. De mon côté j'ai discuté de la place des développeurs Java, nous avons échangé sur les différents profils de gens qui font du Java. Comme le disait l'un de nous, l'informatique offre un accès assez large et le premier venu peut commencer à "faire de l'informatique". Ensuite pour monter la chaîne technique, le travail au quotidien n'offre pas sur un plateau de la connaissance. Nous étions d'accord pour dire qu'en Java, il faut se tenir au courant, et se remettre en question régulièrement. Je ne dis pas essayer chaque semaine un nouveau framework. Je dis simplement profiter d'un petit événement comme celui-ci pour prendre un verre et partager ses connaissances et sa passion. La passion, le moteur qui anime les personnes qui étaient là ce soir. A un BarCamp il n'y a rien à prouver, rien à gagner. Même si j'ai vu/senti la volonté de certains de s'exposer et de parler d'eux plutôt que des sujets, finalement j'ai trouvé l'ensemble de la soirée très ouverte et sympathique. A quand le prochain BarCamp ? Quelques photos prises avec l'iPhone :



(OFF) Félicitations à Julien

Dubois qui je crois savoir est papa d'une petite fille.

## SpringSource rachète Terracotta... ou pas

Mais d'où cette rumeur (fausse) de "SpringSource rachète Terracotta" a-t-elle émergée ? **Prenez les ingrédients suivants** : - une présentation de [Terracotta](#) organisée de manière parfaite par [Zenika](#), cabinet de conseils et d'architecture. - une personne de SpringSource qui tient à garder l'anonymat qui nous annonce que la semaine prochaine, SpringSource fait ZE annonce. Témoins : Erwan de Xebia. - un champagne sommes toutes très correct. - un esprit tordu et qui pose la question à Ari Zilka (moi) - une deuxième coupe de champ... de coca - le désir d'avoir un scoop d'enfer pour le Touilleur - une présence journalistique de ma pomme qui n'hésite pas à aller 2 fois cette semaine à des événements pour toi lecteur Agitez le tout : hop SpringSource rachète Terracotta. Cher lecteur, ce que vous ne savez pas encore et qui est complètement vrai : SpringSource a annoncé [cette nuit](#) une nouvelle modification de la politique de licence. Après la petite vague provoquée en grande partie par des personnes un peu mal lunées sur TheServerSide.com, SpringSource a réagi et annoncera donc les détails d'aménagement de licence afin que les éditeurs ayant intégré SpringSource ne soient pas pénalisés. Dommage du coup on a presque pas parlé de la sortie de [SpringSource dm Server](#) mardi dernier (voir aussi [cet interview](#) sur InfoQ). Bon alors SpringSource rachète Terracotta ? Non d'après Ari Zilka, le CTO de Terracotta à qui j'ai posé la question. La raison : SpringSource et Terracotta partagent les mêmes VentureCapital Risk. En effet, peu de raisons de se vendre à soi même une entreprise A vers une entreprise B... non ? Alors mince, qu'annoncera SpringSource d'ici à la semaine prochaine ? Un rachat ? Oui certainement. Avez-vous une idée ? Postez vos commentaires ci-dessous. Le plus original gagnera un mug avec le logo du Touilleur Express !!! Fermeture des commentaires à l'annonce de SpringSource **Note** : Thomas Queste a fait un résumé complet de la soirée [sur son blog](#).

## Quelques photos de la soirée Terracotta

Quelques photos prises avec l'iphone : Ari Zilka, CTO de Terracotta



[Julien Dubois](#), responsable de SpringSource France en discussion avec Florent Ramière et Nicolas Romanetti [de Jaxio](#).



Ari c'est

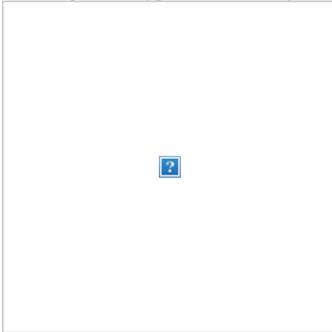


un vrai : il utilise un Mac. Apple Rulez !!!

## Comment trouver un logo pour son entreprise

?

Pour réaliser le logo, l'identité d'un site ou d'une entreprise, c'est tout un travail de création et de recherches. Il est intéressant de regarder quelles sont les différentes tendances en ce moment. [Sur ce site](#) vous pourrez constater que les logos à la mode cette année sont largement inspirés du Web 2.0. Regardez le logo de Skype et son effet glace, ou le logo d'Adobe AIR, volontairement



organique. Le logo de Microsoft Silverlight utilise un effet



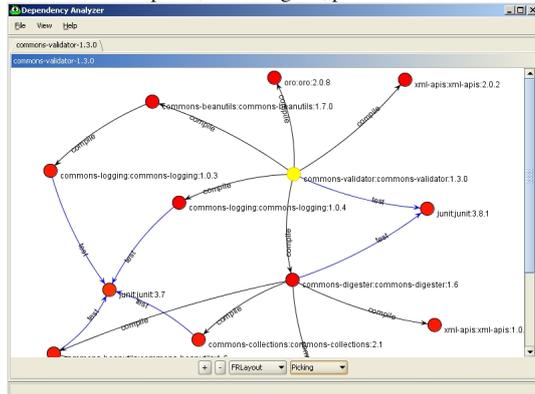
3D matricielle. A la recherche d'idées, vous pouvez aussi jeter un oeil [sur ce site](#) qui liste une cinquantaine de logos différents, à dominance anglo-saxonne. Le logo d'InnoTeria



pour terminer :

## Accélérer Maven2 : l'analyse des dépendances

Votre projet utilise Maven2 ? Cet article est pour vous. Un projet compilé avec Maven2 fait appel à un certain nombre de dépendances externes (spring, hibernate, commons-\*)... Il est aussi courant de rencontrer des "super pom", c'est à dire des pom.xml appelant par dépendances d'autres pom. Par expérience, j'ai vu qu'il devient difficile de maintenir dans le temps l'ensemble de l'arbre des dépendances. La conséquence ? Le projet devient plus compliqué, donc plus lent à compiler. Il est difficile d'identifier rapidement les dépendances inter-modules dans le projet. J'ai trouvé un petit outil bien pratique hier : [Dependency Analyzer](#) est un outil d'analyse graphique des dépendances Maven d'un projet. Après l'avoir installé, cet outil écrit en Java prend un fichier pom.xml en entrée, pour ensuite afficher en étoile ou en arbre, les dépendances. Grâce à cet outil, je me suis rendu compte qu'il y a un trop grand nombre de versions de JUnit dans le projet, que certains modules sont trop liés, donc fragiles, pour effectuer du refactoring facilement dessus.



Vous allez me dire : quel intérêt y-a-t-il à regarder la build Maven qui après tout, fonctionne très bien sur votre projet ? Réponse : la productivité. Si la compilation complète d'un projet dépasse les 2 minutes, l'équipe aura souvent mis en place des systèmes plus rapides pour fonctionner comme le déploiement à chaud, ou le fonctionnement en mode "exploded" afin de ne recharger que les classes modifiées. Bien que très efficace, il est parfois délicat de savoir les réelles dépendances du code, surtout lorsque le volume des fichiers est important. Je pense aussi à l'intégration continue. Idéalement, un projet en journée doit fonctionner en incrémental, sans effectuer de clean. Ce projet compile rapidement afin d'assurer que l'ensemble du code compile. Ensuite, et seulement ensuite, je pense qu'une tâche de l'intégration continue doit être les tests unitaires. Pour que tout ceci s'effectue le plus rapidement possible il faut - une machine d'intégration puissante - un repository SVN local ou un proxy svn - un projet qui compile l'ensemble du code de manière incrémental, sans qu'un clean ne soit

nécessaire - un projet d'exécution des test unitaires Plus globalement, faire respirer son intégration continue est un investissement indispensable. J'ai lu avec intérêt [les article de Jérôme Van Der Linden](#) s d'OCTO sur la mise en place d'usine de production. Articles que je vous conseille si vous réfléchissez à la mise en place d'une intégration continue, de la compilation à la mise en production. Xebia propose aussi [un pointeur sur un article de Kevlin Henney](#) sur des sujets moins terre à terre que Maven, intéressant pour prendre un peu de recul. Enfin je garde en mémoire [SonarL](#), l'outil dont je vous ai parlé cet été qui permet d'apporter de la structuration d'architecture entre différents modules d'une application.

## TDD : avantages et dangers

Question de Florent ce matin en arrivant à la BNP : "TDD en vrai ? ça marche ? à quoi cela sert-il ?". Bonne idée je me dis : réfléchissons un peu sur la démarche de développement pilotée par les tests, ou Test Driven Development (TDD). Tout d'abord un constat : au [BarCamp](#) la semaine passée c'était l'un des sujets discutés, même si finalement 3 personnes sur 10 avaient réellement mis en place TDD au jour le jour. Je tiens à signaler que je n'ai pas d'avis sur la question. Je serai plutôt attentiste que suiveur. **TDD c'est quoi ?** Le Développement piloté par les tests est une pratique de développement qui préconise d'écrire les tests unitaires avant d'écrire le code. Nous en avons eu une démonstration brillante [au Paris JUG](#) en juin dernier. Le cycle préconisé par TDD comporte cinq étapes : 1. écrire un premier test ; 2. vérifier qu'il échoue (car le code qu'il teste n'existe pas), afin de vérifier que le test est valide; 3. écrire juste le code suffisant pour passer le test, 4. vérifier que le test passe, 5. puis refactoriser le code, c'est-à-dire l'améliorer tout en gardant les mêmes fonctionnalités. **La critique** Je m'appuie sur un billet écrit en anglais "[Cram your test driven development up your ass...](#)" dont voici un résumé des critiques de la méthode TDD telle que l'auteur l'explique (*note : j'ai traduit et librement interprété l'article*) : **DEBUT DU RESUME Développement de code en avance, souvent sur-développement (Extra, often useless, up-front development)** L'ensemble des développements suit un cycle d'écriture de code basé sur ce que veut le client, en partant d'un prototype jusqu'à atteindre le code final que le client accepte. La transition d'une étape à l'autre entraîne de la réécriture de code et donc, entraîne l'inutilité du code initial, qui reste à l'état de prototype. Peu de développeurs peuvent se vanter d'écrire dès le premier coup une architecture complète et parfaite. C'est faux. Le client ajoute ou retire des demandes, vous butez sur une difficulté et vous changez de librairie, c'est la vie d'un développeur. En conséquence, l'auteur de l'article suggère que les TDD en forçant la pratique de l'écriture des tests dès le départ, sont inefficaces car ces tests seront effacés lorsque le code évoluera. **Le développement avec des visières rétrécit la vision du développeur** Encore une fois l'auteur explique que de même que les chevaux de courses portent des visières pour courir droit, les TDD tendent à restreindre le regard du développeur. Le risque : à trop se focaliser sur les tests unitaires, on en oublie le reste. Le résultat est que l'on tend à accepter une solution juste correcte pour se coller à ses tests, et que l'on ne s'oriente pas vers la meilleure solution. C'est d'autant plus vrai si les tests représentent beaucoup plus de lignes de code que le code testé. **Les tests pèsent plus que le code testé, et sont plus importants** Ce que veut dire l'auteur : après avoir écrit 500 lignes de tests, vous sentez-vous capable de tout casser pour reprendre une autre idée ? Vraiment ? A passer beaucoup de temps sur les tests, n'est-ce pas se détourner de l'objectif initial qui est avant tout d'écrire une application ? (*note : je ne partage pas l'avis de l'auteur*) **S'enfermer dans un coin** L'importance que l'on donne aux tests unitaires apporte le risque de croire que le test unitaire fait tout, et qu'en conséquence on adapte le code de la classe principale au test. Même pire, l'importance que l'on donne aux tests nous fait croire que les tests sont la vérité absolue, alors que l'on peut aussi écrire un test unitaire complètement faux. Des tests unitaires avec des bugs, cela existe. **L'application limite des TDD** Il est admis que les TDD s'appliquent difficilement à certains développements comme les développements d'interface utilisateur ou les interfaces Webs. Faut-il mobiliser toute son énergie pour mettre absolument des tests unitaires, difficile à maintenir, pour dire que l'on a fait des tests unitaires ? **Les tests résolvent des problèmes qui n'existent pas** TDD n'empêche pas l'apparition des Bugs. C'est une bêtise que de penser que les TDD empêchent l'apparition de bugs, car c'est ignorer les traitements fonctionnels et les cas limites. Une fois un bug trouvé, il est intéressant d'écrire un test unitaire afin de s'assurer que le bug ne va pas réapparaître. Mais que se passe-t-il si un développeur en fixant un bug, casse complètement les tests unitaires ? Qui a raison ? le test unitaire initial ou le développeur qui corrige un bug ? **La couverture des tests unitaires** Une partie de la pratique TDD consiste à diviser en petits tests facilement testables des gros tests. Le problème est qu'à diviser en petits éléments, on test trop bas et donc sur un focus bien trop bas. **Inefficace** Si une partie de code doit être réécrite avec des tests unitaires pilotés par le développement, et qu'il faut très souvent changer les tests unitaires pour travailler sur une partie, n'est-il pas plus efficace d'écrire le code complètement puis de le tester à posteriori ? **Il est impossible d'estimer le temps de développement** Un planning précis ne peut être mis en place que si l'on a une idée assez précise de la quantité de travail à effectuer. Les TDD ajoutent une surcharge au développement qui représente plus de code à écrire que la classe développée elle-même. Si c'est envisageable pour un projet open-source où la contrainte de temps est vague, est-il possible de faire du TDD chez un client au forfait ? **FIN DU RESUME** Retrouvez l'article original : <http://ww2.samhart.com/node/125> Après cette première partie qui tire à boulet rouge sur TDD, voyons un peu si nous pouvons répondre par quelques contre-arguments. Je vous traduis en partie [cet article](#) qui répond directement à "Cram your test driven development..." **Premier argument: Développement de code en avance, inefficacité et trop de codes** En effet il est facile de dire que la pratique TDD entraîne l'écriture de code qu'il faudra maintenir, et que lorsque l'on travaille sur un prototype, cela revient à écrire du code pour rien. Soyons honnête : j'ai vu souvent des parties de code qui n'évoluent pas ou peu. Et rappelons que l'on parle de tests unitaires : certaines classes finalement ne seront jamais réécrites. Donc réduire TDD à un machin qui entraîne trop de code, c'est faux. Le prototype est évidemment loin de l'application finale, mais un Controller reste un Controller, et on peut tout à fait placer des tests et développer son application en s'appuyant sur la pratique TDD. **Deuxième argument: Development with blinders on, Tests are weighted more heavily than the code and Coding yourself into a corner** L'idée de ces arguments est de dire que des tests incorrects vous entraînent vers une pente abrupte et que vous risquez de ne croire que vos

tests en essayant de corriger votre code, là où des tests seraient faux. Faut-il des TDD de TDD ? Je vous donne mon avis : encore une fois, écrire des tests unitaires est faussement facile. C'est une tâche qui mise dans les mains d'un développeur inexpérimenté risque d'entraîner encore plus de cauchemar. Je vois la phrase sortir du bois : "... le test ne passe pas, corrige ton code..." alors que c'est peut-être : "ton test pourri est complètement dépassé, mets le à jour et reviens me voir quand tu l'auras fait" TDD est une pratique intéressante car l'idée est de capturer dans des tests unitaires des postconditions souhaitées pour du code (NDLR: belle phrase qui n'est pas de moi). Le problème est qu'un test unitaire représente à l'instant où l'on écrit ce test, la vision du client. Pour peu que celui-ci change d'avis, il est évident qu'il faudra reprendre le test. En effet c'est un effort.

**La conclusion du Touilleur** Mon avis sur la question : TDD est une pratique intéressante, à appliquer peut-être avec un peu de recul, en évitant de croire que TDD est le nouveau Saint-Graal qui va faire repousser vos cheveux. Chaque méthode a ses évangélistes. Laissez moi 10 minutes avec Scrum, je vais vous démontrer que je suis un grand intégriste prêt à essayer de vous convertir. TDD tombe dans le même panneau : pour peu que votre chemin croise un intégriste passionné de TDD, vous serez illuminé et perdu. TDD c'est pourtant le gilet jaune de signalisation qu'il faudrait avoir dans sa voiture. A appliquer parfois sur certaines parties de votre code où il est clair, facile et intelligent de placer de solides échafaudages, les tests unitaires. Après tout c'est une gymnastique qui vise à écrire des tests unitaires "utiles" et intelligents. Je pense enfin que pour bien écrire des tests unitaires il faut investir du temps pour apprendre toute la boîte à outil suivante : - [JUnit](#) - [dbUnit](#) <http://dbunit.sourceforge.net/> - [EasyMock](#) <http://www.easymock.org/> - [Unitils](#) <http://www.unitils.org> Finalement je n'ai pas répondu à la question de florent : "Connais-tu quelqu'un d'assez neutre pour parler de TDD et pour donner son avis, sans que son œil se mette à briller et qu'il tente de te convertir ?" Je vous laisse répondre et donner votre avis sur la question. On pourra en parler au prochain Paris JUG [le 14 octobre](#) sur OSGI, organisé par Xebia. -----

----- **Références:** - "[Cram your test driven development up your...](#)" de Samuel Hart - "[That sounds unnecessarily uncomfortable](#)" de Ben Tels - [Test Driven Development](#) - [Définition de TDD sur Wikipedia](#) - Livre "[The Art of Unit Testing](#)"

## SpringSource : nouvelle politique de maintenance

Hop avant tout le monde, petite annonce de la soirée : <http://blog.springsource.com/2008/10/07/a-question-of-balance-tuning-the-maintenance-policy/> Fin du débat, vous pouvez retourner à vos activités et continuer à travailler.

## L'annonce de SpringSource : collaboration avec Pentaho BI

La nouvelle que je surveillais la semaine dernière est tombée : SpringSource et la société Pentaho annoncent un accord commun visant à rapprocher Spring de la solution de business intelligence de Pentaho. Voir le communiqué de presse : <http://www.springsource.com/node/717> [Pentaho Corporation](#) est l'entreprise qui appuie la solution d'analyse métier, *Pentaho BI Suite Enterprise Edition*, avec du support et de la documentation. Avec des outils d'analyse OLAP, le moteur Kettle (ETL) et aussi le moteur de génération de rapports, Pentaho est une plateforme d'analyse et de transformation de rapports. Prenons l'exemple de rapports financiers (au hasard). Pentaho est en mesure de transformer des données brutes (type CSV) en données agrégées afin d'offrir un outil d'analyse temps réel aux personnes du back-office (au hasard et par exemple). Cette annonce de se rapprocher de SpringSource et de la solution SpringSource Application Platform, appliqué à la finance, fait qu'il va falloir regarder dans les mois qui arrivent ce que SpringSource va proposer.

## Présentation d'OSGI au Paris JUG : naphtaline et peinture fraîche

Le Paris JUG nous a proposé une présentation d'OSGI ce mois-ci, organisée par Cyril Le Clerc et Nicolas Griso [de Xebia](#). Après une présentation sur l'histoire d'OSGI, nous avons vu les différents champs d'applications, avec en point de mire la question posée sur le blog de xebia : "est-ce vraiment une plus value pour les développements ?" La présentation commence par une introduction à l'histoire de l'Alliance OSGI, constituée d'acteurs du monde des télécoms, de l'embarqué et des middlewares. OSGI existe depuis 1999, la dernière version de la spécification de ce framework est la release 4.1 (aka JSR-291) sortie en mai 2007. Je retiens qu'avant tout, OSGI a été spécifié pour des problèmes d'applications embarquées dans des téléphones, de l'équipement automobile (BMW) et finalement, le tout me fait penser que c'est assez loin de mon informatique de "gestion" comme dit Cyril. OSGI est entre autre constitué d'une spécification d'un framework. Ce framework répond à deux soucis en Java : la modularité afin d'exposer un contrat versionné de classe et le contrôle d'un service entre plusieurs applications. Nicolas explique qu'en Java, le mécanisme actuel basé sur les JAR n'est pas adapté. Un JAR n'a pas de consistance au runtime, toutes les classes se retrouvent chargées dans le Classpath de l'application. Il existe bien un mécanisme peu connu (protected / public) dans le Manifest.mf mais qui reste peu utilisé et inadapté. Il n'y a pas de protections des dépendances en Java à l'exécution. La première classe trouvée est chargée par le ClassLoader, ce qui entraîne parfois des ClassNotFoundException ou des NoClassDefFoundException (quizz : expliquez la différence). C'est un point faible de Java. **ClassLoader mon pote de 10 ans** Le classloader respecte une hiérarchie. Il faut savoir que là où BEA Weblogic ou JBoss utilise un class-loader "children first" où une classe de votre WAR sera chargée avant la classe du serveur d'application, IBM Websphere fait l'inverse : une classe du

serveur d'application se retrouve chargée avant la version de votre WAR. Ennuyant non ? J'en ai fait l'expérience [en 2007](#). Cela donne des choses assez amusantes comme l'explique Cyrille avec JBoss. Celui-ci package une version 3.2.4 d'hibernate alors que vous voudriez peut-être utiliser la version 3.3. Il explique aussi qu'avec Tomcat il est impossible à priori de faire tourner à la fois du Spring 2.0 et du Spring 2.5 au sein du même WAR. Bref le souci : la gestion des versions des dépendances dans une application peut vite tourner au cauchemar et Java n'offre rien (pour l'instant). **JCP et OSGI sont sur un bateau** Nicolas et Cyrille abordent ensuite l'historique entre OSGI d'une part et le Java Community Process (JCP) d'autre part. OSGI est une alliance fermée dont le droit d'entrée de 20 000 \$ (je crois) permet de filtrer les happy-fews. JCP est gratuit pour le commun des mortels (ce qui m'a valu une petite peluche Duke d'ailleurs pour avoir bien répondu). Il faut voir qu'historiquement, OSGI Alliance répond à un besoin bien différent de celui du JCP, piloté par SUN. Cependant OSGI et JCP travaillent main dans la main, puisqu'OSGI est spécifié sous la forme de la JSR 291. Bref je vous passe les détails mais ils se sont rabibochés et ils vont nous faire un petit. **Du code, du code, du code** Cyrille ouvre le capot de la belle afin de commencer à nous montrer en live ce que donne OSGI. Avant cela, Nicolas explique le fonctionnement d'OSGI d'un point de vue technique. Un JAR classique devient "osgi-fié" à partir du moment où son MANIFEST.MF s'est fait en quelque sorte visité par OSGI. Je m'explique. Un certain nombre de mots-clés sont ajoutés dans le fichier Manifest.MF afin de définir les dépendances, ce qu'expose le module et les bundles nécessaires à l'exécution. Ce système très simple permet d'ajouter des règles de version afin que votre module OSGI n'utilise qu'une certaine version d'une API. Imaginez une sorte de balise "dependencyManagement" à la Maven, au coeur du Manifest.mf. Voici un exemple pour illustrer ce charabia :

```
Manifest-Version: 1.0
Bundle-ManifestVersion: 2
Bundle-Name: HelloService Plug-in
Bundle-SymbolicName: com.javaworld.sample>HelloService
Bundle-Version: 1.0.0
Bundle-Vendor: JAVAWORLD
Bundle-Localization: plugin
Export-Package: com.javaworld.sample.service
Import-Package: org.osgi.framework;version="1.3.0"
```

Ici j'ai un bundle "HelloService" qui importe en dépendance le framework OSGI dans sa version 1.3. (à noter qu'une ligne du fichier Manifest.mf ne doit pas dépasser 72 caractères). Comme on le constate le mécanisme est simple et rustique. Ceci s'explique par le fait qu'OSGI fonctionne sans soucis avec Java 1.1 et J2ME. C'est prévu pour de l'embarqué au départ. Au niveau du code Cyrille montre ensuite le code d'un service OSGI afin de nous présenter avant la pause le concept d'Activator et de ServiceRegistration. Je vous donne un peu de code car je n'ai pas les slides de Xebia, mais je vous invite à surveiller plutôt sur le site du Paris JUG les slides afin de vous faire votre propre idée du code. Sur cet exemple j'implémente l'interface BundleActivator afin d'être à l'écoute et d'enregistrer mon service de test dans OSGI :

```
public class HelloServiceActivator implements BundleActivator {
    ServiceRegistration helloServiceRegistration;
    public void start(BundleContext context) throws Exception {
        HelloService helloService = new HelloServiceImpl();
        helloServiceRegistration = context.registerService>HelloService.class.getName(), helloService, null);
    }
    public void stop(BundleContext context) throws Exception {
        helloServiceRegistration.unregister();
    }
}
```

Voyons comment ensuite importer mon service afin de m'en servir :

```
public class Activator implements BundleActivator {
    ServiceReference helloServiceReference;
    public void start(BundleContext context) throws Exception {
        System.out.println("Salut le touilleur ");
        helloServiceReference = context.getServiceReference>HelloService.class.getName());
        HelloService helloService = (HelloService)context.getService(helloServiceReference);
        System.out.println(helloService.sayHello());
    }
    public void stop(BundleContext context) throws Exception {
        System.out.println("ByeBye le touilleur !");
        context.ungetService(helloServiceReference);
    }
}
```

La clé de voute qui semble un peu bizarre c'est ce ServiceReference qui n'est ni plus ni moins un pointeur vers notre service. Le contexte nous sert de transport. Avouez que ce code tout simple nous semble un peu "old-school" à l'époque d'IOC et compagnie non ? et les annotations ? A cela Nicolas apporte une explication importante : OSGI c'est au départ pour de l'embarqué, et donc pour une version de Java plus simple que Java 5 avec annotations et compagnie. D'ailleurs OSGI pose un certain nombre de soucis avec la programmation orientée AOP car ces mécanismes ne sont pas de bons collaborateurs pour OSGI. Il faudrait creuser pour en savoir plus. Je vous donne mon impression : ça sent la naphthaline. Et je crois bien que c'est le message que Nicolas et Cyrille font passer. Une partie de la salle a peut-être mal compris l'exercice qui se déroule ce soir : on explique OSGI, on entre dans le détail, mais on vous laisse vous faire votre propre opinion. J'ai beaucoup aimé. Cyrille qui pourrait vendre un brushing à un chauve propose à un moment une idée pour expliquer OSGI qui m'a bien plus : OSGI est un pare-feu (firewall) visant à d'abord isoler les classloaders, puis à apporter une couche de contrôle simple à une application. Quelque part, c'est une forme d'injection de dépendance avec un moteur de versionning dont finalement tout repose sur Manifest.mf. Si vous me suivez bien vous voyez un peu où je veux en venir : là où Spring fait de l'injection de dépendance pour vous brancher votre application, on s'attend à ce qu'OSGI fasse

la même chose mais au niveau des Class, une sorte de Spring mais pour le ClassLoader... Or ce n'est pas franchement le cas. Nicolas présente ensuite le système de configuration basé sur du XML après la présentation des Declaratives Services. Bien que plus facile à maintenir, c'est peut-être du côté de Spring DM finalement que l'on aura un système sympathique à utiliser. Et encore une fois je pense en effet que les génies de Spring vont nous pondre un système qui répondra à nos besoins. **Spring, spring, spring, la peinture fraîche** Spring DM avec la RFC 124 écrite par SpringSource propose un système intelligent de configuration qui n'est encore qu'à l'état de Draft au sein de l'Alliance OSGI. En même temps il n'y a pas d'alternatives (aussi intelligentes). Nicolas explique que Spring apporte une simplification d'OSGI grâce à un namespace qui permet de piloter les services OSGI très facilement au sein de la configuration XML de Spring. Et en effet, c'est vraiment ce qu'il manque pour séduire le tout venant. Attention ce n'est pas encore terminé comme l'explique Cyrille. Bref ça sent la naphthaline et la peinture fraîche **Pause glouglou** A peine le temps de discuter avec quelques têtes connues que nous repartons pour la deuxième partie, que je vais faire courte. La démonstration propose de voir comment un service résout une dépendance, comment un service s'expose à travers une Servlet, comment utiliser log4j et entre autre, passer un fichier log4j.properties au bundle de Log4j. Enfin la démonstration s'est terminée par le changement à chaud d'un bundle. C'était convaincant et simple. L'assistance a peut-être pensé que Cyrille et Nicolas allaient tenter de les convertir mais je pense pas que c'était le cas. J'ai pensé qu'ils ont montré du code en l'état, en laissant chacun se forger une opinion. Et évidemment que cela laisse perplexe. Que penser de la gymnastique pour remplacer à chaud une partie de l'application ? Cyrille le dit honnêtement : "en application de gestion moi je n'ai jamais eu ce cas d'usage". Le seul truc marrant que je trouve à dire à mon voisin pendant que cyrille arrête, relance, arrête son service : "...et je coupe le son..... et je remets le son... et je coupe le son... et je remets le son..." Est-ce que penser que l'on arrête une partie de l'application avec OSGI ce n'est pas faire du [Philippe Katerine ? OSGI dans le monde JEE](#) Nicolas apporte ensuite une explication à l'utilité d'OSGI et je pense que c'était le plus intéressant. OSGI se retrouve de facto dans les serveurs d'application comme Websphere 6.1 et Weblogic 10 (en partie). L'explication est assez simple : OSGI permet aux éditeurs de middleware de gérer et d'isoler les JAR du serveur d'application par rapport à ceux de l'application, et aussi de faire cohabiter sans soucis les EJB 2.1 et les EJB3 sans que le ClassLoader ne vienne perturber le tout. D'autre part la gestion du cycle de vie permet à Websphere par exemple de piloter l'orchestration de services facilement. L'ESB Apache ServiceMix 4 est basé sur Spring DM et apporte des fonctions de monitoring et d'exploitation. Les équipes de production vont être contentes. Glassfish bascule sur Apache Felix, une des implémentations Open-source d'OSGI comme l'explique Alexis MP qui était présent. Pour terminer les bonnes pratiques pour OSGI :

- o Préférer Import-Package à Require-Bundle
- o Utiliser des plages de version dans le manifest afin de ne pas trop restreindre les versions (gag)
- o Séparer les apis des implémentations afin de faciliter la maintenance
- o Penser orienté objet
- o Gérer le cycle des dépendances

Dans les antipatterns j'ai noté :

- o Utilisation des Class.forName au lieu de passer par une Factory
- o Utilisation des librairies non osgi-fiées

A ce propos un gros point bizarre, pouri, c'est les librairies osgi-fiées. Il s'agit d'un JAR dont le Manifest.mf a été trituré afin de devenir compatible avec OSGI. Or comme il n'y a pas de normes, chacun y va de sa version faite maison. Spring propose un repository, mais ils ne pourront pas obliger les gens d'Apache ServiceMix de repackager leurs Jar en module OSGI. D'où des millions de question dans ma tête dont la première est : ce n'est pas un risque de gros gros b... ? **La conclusion** Sur le contenu : OSGI est un socle robuste et éprouvé qui répond à des problématiques complexes que la plateforme Java n'adresse pas pour l'instant. Il manque de maturité dans le monde JEE et il est en pleine évolution, en partie grâce à SpringSource. C'est donc un acteur à surveiller du coin de l'oeil et il aurait été dommage de ne pas utiliser certains de ses principes. Maintenant s'agissant d'un framework "historique", certaines de ses implémentations sont un peu dépassées. Sur la forme je vais vous donner mon avis : pour une fois on a eu une présentation au caviar de Xebia qui a réussi à montrer à la fois les points intéressants d'OSGI mais aussi le côté qui "coince un peu" : à la fois le vieux et le trop jeune. Naphtaline et peinture fraîche pour ceux qui suivent. Et là, c'était un exercice pas forcément facile au Paris JUG devant une assemblée qui attendait peut-être une présentation "OSGI ça déchire et je vais vous convertir mes frères". C'était bien plus fin que cela. Enfin moi c'est ce que j'ai compris. Merci à Nicolas et Cyrille, de Xebia. **Actualités du Paris JUG** : A l'occasion de l'anniversaire du JUG en février prochain, Antonio recherche des speakers pour faire des quickies de 15mn. Etant donné que je devrai en faire une à Devvix en décembre prochain, je pense que je vais proposer un sujet à Antonio rapidement. JetBrains, l'éditeur de l'excellent IDEA IntelliJ, offre des licences gratuites pour les participants du JUG. Eyrolles offre des livres et le Paris JUG vous propose de participer à des lectures de livres afin de publier des articles. Enfin j'ai entendu l'appel d'Antonio pour bloger sur le blog du Paris JUG. Là il est 00h41 et je pense que je vais fermer la boutique pour ce soir. Je vais réfléchir encore un peu avant de proposer ma candidature. Enfin en décembre prochain ne manquez pas [la conférence Devvix](#) (ex Javapolis) qui aura lieu à Anvers (2h de paris en TGV) et qui se déroule sur 3 jours. Florent (de jaxio) m'a boosté afin que je fasse une présentation à Devvix. J'ai donc postulé et proposé de faire une présentation sur la mise en place d'un ESB dans une plate forme de traitement de flux financiers. Ensuite il faut que je sois accepté, ce qui n'est pas encore acquis. Pour terminer quelques photos. Cliquez sur les photos pour les afficher en grand : [caption id="attachment\_381" align="alignnone" width="225" caption="Nicolas Griso et Cyrille LeClerc"]

Nicolas Griso et Cyrille LeClerc



[/caption] [caption id="attachment\_385" align="alignnone"

Beaucoup de monde



width="225" caption="Beaucoup de monde"]

[/caption]



## Le Touilleur a changé d'hébergeur

Cette nuit j'ai déménagé le touilleur vers un autre hébergeur. Reconfiguration DNS, réinstallation de la base et de Wordpress, configuration des plugins, réinstallation du thème, bref pas mal de petites choses. Il y a encore quelques soucis entre autre avec les photos, et l'email nicolas at touilleur (sur le logo en haut à gauche du site) ne fonctionne pas pour l'instant. Donc comme on dit dans ce cas là : "Veuillez nous excuser pour ce désagrément" En principe vos comptes pour ceux qui ont déjà posté un commentaire devraient encore fonctionner. J'ai aussi changé la règle de modération (encore une fois). Quelqu'un qui a déjà posté une fois et qui a eu son commentaire approuvé peut ensuite poster librement comme il le veut. Par contre lorsque vous postez un commentaire pour la première fois, je dois le modérer manuellement.

## Devoxx : le salon sur Java de l'année



Devoxx est le nouveau nom de Javapolis. C'est 3 jours de conférences sur Java et 2 jours d'ateliers approfondis sur l'ensemble des technologies de la plateforme Java. Cette année les conférences portent sur Java EE6, [REST](#), les langages de scripts, OSGI, Scala, [Web Beans](#), BPM, et encore [pleins d'autres](#)

[sujets](#). Organisé par le Java User Group Belge, l'événement se déroule du 8 au 13 décembre à Anvers en Belgique; le tout à 2h de train en partant de Paris. Les organisateurs de Devovx ont préparé un clip sympa : <http://www.devovx.com/display/IV08/2008/10/07/Devovx+2008+Promo> A la 44 ème seconde vous pourrez même apercevoir Antonio du Paris JUG avec un teeshirt noir en train de poser une question ! [Le Paris JUG](#) propose une réduction pour l'entrée, si vous souhaitez y participer en tant que particulier. Pour en bénéficier, rendez-vous sur le site du Paris JUG. Une fois inscrit sur la mailing list, envoyez un email en demandant le code de réduction. Enfin à propos de JUG (Java User Group), signalons l'ouverture d'un nouveau JUG : [le JUG de Bordeaux](#). Avec des racines sur le bassin d'Arcachon et de la famille à Bordeaux, l'Acquittaine est ma région de coeur. Si j'ai l'occasion d'aller à une réunion du JUG sur Bordeaux, je ne manquerai pas d'en parler.

## Trouver une clé de Windows XP

Je n'aime pas Windows. Je n'aime pas les outils dangereux et stupides que l'on peut télécharger sur Internet. Prenez par exemple un petit logiciel qui permet de recenser les logiciels installés sur votre machine, [Belarc Advisor](#). Cet outil liste les périphériques et les logiciels afin de vous conseiller par la suite d'installer des patches. Imaginez que ce logiciel par exemple trouve la clé de votre installation de Windows XP, puis qu'il l'enregistre dans un fichier de résultat. Maintenant imaginez que vous êtes un mauvais administrateur système sur Windows. Votre serveur web préféré tourne sur votre machine, et Google se permet donc d'indexer votre machine. Vous me suivez ? Le miracle de la technologie fait que le rapport Belarc Advisor sera indexé par Google... avec votre clé de Windows. Du coup, c'est dommage, mais vos clés de logiciels se retrouvent toutes sur Internet. C'est pas beau la technologie ?

Ni une, ni deux, vous pouvez maintenant trouver une clé gratuite pour n'importe quel logiciel avec la recherche suivante sur Google par exemple pour XP:

["Belarc Advisor Current Profile" "Software Licenses" "Windows XP" key:](#)

Elle est pas belle la technologie ?

## Un modèle de Product Backlog et de Sprint Backlog avec Excel

La tenue du [product backlog](#) et du [sprint backlog](#) peut s'effectuer avec des logiciels comme [IcScrum](#), ou aussi tout simplement avec Microsoft Excel. Au début de cette année lorsque je travaillais avec Scrum sur le projet Karma chez Reuters, finalement le plus simple était d'avoir un fichier Excel, ni plus, ni moins. Avec quelques petites macros et deux ou trois courbes, cela permettait d'avoir un [burndown chart](#) à jour. Depuis quelques jours nous parlons Scrum avec l'équipe chez BNP-Paribas. Nous avons fait un premier sprint sur les sujets d'architecture. Cela m'a donné l'occasion de reprendre mon fichier Excel afin de l'adapter, et je voudrais le partager ici afin de l'améliorer, et de vous en faire bénéficier. Il est libre de droit, vous en faites ce que vous voulez. L'onglet Product Backlog vous permet de lister l'ensemble des fonctionnalités et des demandes, ainsi que d'indiquer le sprint où un item est activé. Dans l'onglet Sprint j'ai mis un petit exemple rapide afin de vous montrer comment la feuille commence. Chaque sprint dure 10 jours. Les valeurs numériques représentent le "reste à faire". Il n'est pas interdit d'augmenter le reste à faire si en cours d'itération, on se rend compte qu'une tâche a été sous-évaluée, ou que vous rencontrez des complications. Bref c'est ultra simple, à vous de l'adapter à votre activité et d'en faire ce que vous voulez. [lien mis à jour vendredi, désolé] [Télécharger le fichier "Scrum Excel Template Le Touilleur.xls"](#)

## Twitter et iphone, truc de geek

Pour ceux qui connaissent : je me suis inscrit sur [twitter](#), mon login est [letouilleur](#). Pour ceux qui ne connaissent pas le principe de Twitter, voici un petit résumé du concept de Twitter et de l'outil que j'ai installé sur mon iPhone. Twitter est un système de micro-blogging qui permet d'envoyer des messages courts type SMS afin de publier sur une page un statut de votre activité. Exemple : vous êtes au Paris JUG et vous souhaitez simplement publier un message, ou même une photo : quelques lignes dans un message, puis celui-ci se retrouve publié sur votre page. Ni plus, ni moins. L'intérêt ensuite est de créer une communauté et d'échanger avec vos contacts et vos amis. Vous êtes entrain d'installer la dernière version d'Eclipse ? de recompiler le kernel de votre souris ? de vous prendre la tête avec une exception NullPointerException (comment ça y'a des pointeurs en Java ?) ? Bref si vous vous sentez seul dans la stratosphère des geeks, si vous pensez que l'élevage des fourmis en haute-cilésie mérite un débat ou que Spring c'est le string de Duke, je pense que vous êtes prêt à clavarder avec Twitter. Pour aller jusqu'au bout de ma geekitude, j'ai installé une petite application bien pratique sur mon iPhone : [twitterrific](#). Cette application permet de poster des messages vers twitter, d'attacher des photos prises avec l'iphone et d'y associer une information de géolocalisation. J'ai pris mon bureau en photo, le tout se trouve sur [ma page Twitter](#). Enfin si vous scrollez en bas de cette page, dans la barre de navigation vous devriez voir les 3 derniers messages postés sur Twitter. Je suis un geek, j'assume à fond !

## Comment écrire son propre blog ?

Après avoir lu [le Touilleur](#), ce qui serait sympa c'est d'écrire à votre tour sur un blog. **Comment commencer ? De quoi parler ? Comment fonctionne un blog ? Quels sont les 10 conseils pour rendre votre blog attractif ? Combien de temps cela prend-t-il ?** La première chose pour commencer votre blog : le contenu éditorial. Allez-parler de java ? De Scrum ? Pour

ma part les premiers articles parlaient de Java et de Linux. Par la suite j'ai dévié vers les technologies comme JBoss Seam, Adobe Flex, le monde de Google, puis Apple... Quoique votre blog traite aujourd'hui, son contenu changera certainement au cours du temps. L'important est de ne pas perdre votre lecteur en changeant du tout au tout. Le principe pour écrire un article est simple : vous devez avoir envie d'en parler. Je vous avoue qu'en relisant de vieux articles, j'ai eu une période remplissage où les articles n'étaient pas top. Conservez cependant tous vos articles, évitez de revenir dessus pour chercher à faire un article parfait. Il est intéressant souvent de laisser reposer un article une journée ou deux lorsque vous traitez un sujet de fond, afin de vous relire et de voir si l'ensemble tient bien. Au niveau du style et du fond, j'ai appliqué un principe vu sur des blogs plutôt américains : soyez simple, soyez vous même. Les lecteurs ont horreur des blogs avec un style ampoulé, où au bout de 2 paragraphes vous sentez le mammifère supérieur qui a fait un effort pour sortir son costume et sa cravate pour vous parler de NullPointerException. Si vous portez des jeans, vous portez des jeans. Pas la peine de jouer un rôle sur votre blog, c'est intenable à la longue et les lecteurs ne s'y intéressent pas. Publiez un vrai contenu. Les premiers pas lorsque l'on blogue consistent souvent à reprendre des actualités sans y apporter votre touche. Profitez de chaque occasion pour réellement partager votre opinion et donner votre point de vue. Si vous publiez sur un sujet un peu polémique, renseignez vous bien avoir d'écrire. Après tout, ce que vous publiez peut avoir un impact non négligeable par la suite, le contenu de votre billet étant indexé par Google. **Quels sont les outils pour bloguer ?** Je vous parlerai que de ma propre expérience : j'ai commencé avec JRoller, service d'hébergement gratuit qui vous permet de commencer rapidement. Par la suite vous pouvez déposer votre propre nom de domaine pour 14 EUR chez Gandi.net par exemple. Ensuite je suis passé à un hébergement dédié pour 35 eur par mois. Pour ce prix vous avez une machine dédiée (pas virtualisée) sur laquelle vous pouvez aussi tester et installer vos programmes, le tout sous Unix. Je viens juste de déménager le Touilleur de la société Ikoula vers Gandi, où aujourd'hui je paye 14 euros par mois pour l'hébergement, ce qui est raisonnable. Pour ce prix j'ai maintenant une machine virtuelle, ce qui coûte un peu moins cher et suffit à mon besoin. Du côté des logiciels de blogging, je ne connais que WordPress. Il a l'avantage d'être simple et puissant. A noter que sur Gandi, il existe des packs préinstallés avec Apache2, PHP, MySQL et DotClear. Si vous n'êtes pas une brute sous Unix, je vous conseille cette option. **Comment gagner de l'argent avec son blog** Je rigole. Cet intertitre est complètement inutile et très racoleur. On ne gagne pas d'argent avec un blog et on ne doit pas essayer d'en gagner. Là vous vous dites que j'ai un peu oublié de vous parler des bandeaux de publicité de Google que vous avez sans doute remarqué à droite à gauche... Depuis 14 mois mon compte AdSense est créditeur de... 61 dollars ! Autant dire rien du tout. Google ne paye que lorsque votre compte est créditeur de 100 dollars. Et vu le taux de change... **Comment trouver des sujets ?** Pensez en premier à ce que vous faites au travail : ce matin une exception particulièrement compliquée sur Weblogic, un article vu sur InfoQ à propos de Spring dmServer, une discussion avec un ami sur un sujet d'actualité, le compte-rendu d'une soirée du Paris JUG... Bref il y a toujours quelque chose à raconter. **La liberté de dire n'importe quoi** Avant tout une grosse différence entre un blog d'un particulier comme [celui de Tom](#) et le blog d'une entreprise comme celui d'Octo, de Xebia ou d'Ippon. Les blogs des cabinets de consultant sont souvent impressionnants, voire intimidant. Une connaissance croisée à la soirée Zenika me dit texto : "j'ai eu peur de poster mon CV chez ZZZ car leur blog est écrit par des gens trop forts...". Il faut savoir plusieurs choses : tout d'abord en effet les consultants qui publient sur les blogs des cabinets de conseils sont forts sur le sujet traité. Mais ils parlent au nom de leur entreprise, et il y a donc une relecture et une correction importante avant de publier un article sur le blog d'une entreprise. C'est une affaire d'image, et il est difficilement imaginable de pouvoir parler de la pluie et du beau temps, toute la démarche s'inscrit dans un processus de crédibilisation, afin de séduire non pas le client final mais les candidats intéressés pour rejoindre le dit cabinet. Il faut donc dépasser le côté impressionnant et simplement se dire que chez ZZZ, il y a une démarche constructive pour parler de sujets d'actualités sur Java. **Conclusion** Je lis beaucoup de blogs et je serai vraiment content de lire le votre. N'hésitez pas à me contacter par email, celui-ci se trouve en haut à gauche de la page sur le logo le Touilleur.

## Il y a le bon informaticien et le bon informaticien

Souvenez-vous de ce sketch des Inconnus : "[Le Bon Chasseur](#)" et imaginons un instant une projection vers le monde de l'informatique... Notre journaliste est chez Durano Nuclear Industrie où Roger le reçoit dans un bureau, dans un foutoir impressionnant constitué de boîtes de pizzas, de vieux ordinateurs éventrés, d'une calculatrice TI82 posée sur le bureau en évidence, d'un clavier [Dvorak](#) et d'un poster de StarTrek datant de 79... **Ambiance...**  - Bonjour Roger, vous êtes développeur et inventeur du langage B, comment allez-vous ? - KarlaBrok Qapla' ! - Pardon ? - C'est de l'ancien [Klingon](#). Start Trek ! Vous connaissez pas ? Ca veut dire [Bonjour](#). - Ah ok, KarlaBrok alors... Dites-moi Roger, vous développez uniquement avec le langage B ? - Uniquement sur le langage B, dit le Bouchenois. Alors le langage B ou bouchenois, c'est un langage qui est connu par Paulo, déjà, et ça va jusqu'à Maurice à porte d'Ivry, il en fait en régie chez EDF. Alors attention, le langage B c'était avant le langage C !!! Tout y était déjà avant même votre langage, la rhumba, votre machin... *Java vous voulez dire. C'est le nom de notre langage, Java* - ... merde d'interprété... Pardon ? ... nan je disais que vu que c'est interprété votre machine, ça peut pas aller aussi vite que le langage B. Et va que je te fais un gros pâté de 500 lignes là où moi je fais la même chose en 30 lignes de B.... je me marre !!!! je me marre !!! - *Disons que chaque langage a son domaine d'utilisation et que votre langage B a son domaine bien à lui. Racontez-moi ce qui est arrivé à votre dernier programme ?* Bon et ben avec Momo on codait tard un soir vers 17 heures quand soudainement tout d'un coup, le Momo est parti comme un lapin sur un bug. Le programme est parti en vrille et les vannes se sont toutes ouvertes... - ... *les vannes ?* - Ben les vannes du circuit des eaux de refroidissement de la centrale. C'est pas qu'on aurait des fuites de liquide radioactifs mais ces derniers temps on la joue discret si vous voyez ce que je veux dire. Heureusement mon momo c'est un BON informaticien. - *Il a corrigé le bug et il a*

recompilé le programme ? - Nan, plus simple - *C'est à dire ?* - il a coupé le courant. Qui dit plus de courant, dit plus de problèmes. Pas bête le Momo hein ? - *Oui en effet, faut y penser. Parlez moi de votre point de vue sur les licences ?* - Le langage B est LE langage libre par excellence. Y'a deux types de licences, on appelle cela des licencettes dans notre jargon. La licence commerciale et la licence... non-commerciale. La première licence, c'est pour gagner de l'argent. La seconde c'est pour gagner à être connu... pour gagner encore plus d'argent ensuite. D'ailleurs votre Java là, c'est une arnaque commerciale de SUN pour vendre son matériel. - *Java est open-source depuis le 13 novembre 2006.* - Ah pardon... respect, respect... je savais pas... - *Bon et sinon parlez moi de la traque du bug en Langage B, comment trouvez-vous des bugs ?* - La chasse aux bugs en langage B, c'est avant tout une aventure humaine. On part le matin tôt vers 11h après le café-ricard du matin. Y'a bébere avec sa HP qui a développé un appât à bug. On descend chaque ligne de programme imprimé sur ces listings que vous voyez là. Des fois on voit un barkbukette ou un dépassement de pile de curseur. Avec un peu de chance on peut même faire un peu de pointeurs de mémoires infectés. C'est pas la saison mais si on sait où chercher... on trouve - *Les meilleurs coins pour trouver des bugs dans votre langage ?* - AH AH AH ! - *Pardon ?* - Tu me prends pour une quiche ? tu crois que je vais te donner mes meilleurs coins ? Sache que les coins à bugs se transmettent de développeur en développeur. Et SEUL les puristes ont accès à la connaissance ! nom d'une clé de 12 tu es un blanc bec toi - *Que pensent les utilisateurs de vos programmes ? Comment effectuez vous vos tests ?* - 2 Règles : le client est con car il ne sait pas ce qu'il veut et ensuite règle 2 : les utilisateurs ne peuvent pas comprendre ce que l'on fait. D'ailleurs ils ont pas besoin. Les tests c'est simple : si l'utilisateur trouve quelque chose et ben il nous appelle entre 14H et 14h30 sur la ligne d'urgence. - *Parlez moi de votre chef de projet, comment travaillez-vous ?* - Ah le con... Le chef de projet, enfin le dernier qu'on a eu c'était un gamin tout juste sorti de l'école. Le gars avait à peine 3 poils de barbe qu'il voulait déjà être chef de projet. Il a tenu 5 mois et puis il a arrêté l'informatique. - ... *et il fait quoi aujourd'hui ?* - Ingénieur d'affaire. Il vend des ressources à des grosses boîtes. - *Où vous voyez vous dans 1 an ?* - Oh bah avec Bebere le rêve c'est soit d'acheter la TI-83 sur eBay, soit que l'on soit enfin reconnu pour notre boulot et que l'on nous augmente. On est aussi pas mal remonté vous savez... - *Oui la lutte sociale, vos conditions de travail sont assez difficiles, tout ce foutoir...* - Nan ça c'est normal. Nous ce qu'on veut c'est des FORMATIONS et une AUGMENTATION car malgré notre 14 ème mois, nos 17 jours de RTT et ben moi je trouve qu'on se fait enfiler comme des clés à pipe. Marre de se faire EX-PLOI-TER. L'informatique est un secteur SI-NI-STRE avec notre salaire on a juste de quoi se payer l'abo à WolrdOfWarcraft. - *Je vois cela... bon et bien vous passerez le bonjour à Bébere. Il est pas là ?* - Nan le con est parti à une conférence à Reims sur le protocole RBM, mieux que TCP. Et il fait profil bas histoire que les médias l'embêtent pas, rapport à la centrale nucléaire si vous voyez ce que je veux dire. Vous me comprenez ? - *Oui, oui. D'ailleurs je pense à un mot en Klingon et j'ai envie de vous dire* "...KarlBruni..." - Ca veut dire quoi ? - *Je vous laisse imaginer...*

## Licence CodeWeaver gratuite aujourd'hui seulement

Vu sur le site MacNews et aussi notifié par Dimitri, CodeWeavers offre aujourd'hui seulement une licence pour soit CrossOver pour Mac, soit CrossOver pour Linux. Rendez-vous sur <http://lameduck.codeweavers.com/free/> Après avoir saisi une adresse email valide, vous recevrez par email un numéro de licence pour pouvoir télécharger CrossOver. CrossOver est un logiciel qui permet de faire fonctionner sur un Mac ou sur un PC une application Windows. Par exemple pour les sites internet ne fonctionnant qu'avec Internet Explorer, CrossOver permet de faire "tourner" Internet Explorer sans soucis sur Mac OS ou Linux. La raison de cette journée exceptionnelle semble être un pari que le patron de CodeWeavers a perdu comme l'explique cet article sur [EchosDuNet](#).

Il y a de cela trois mois, Jeremy White, le PDG de CodeWeaver, a lancé des défis à l'administration fédérale américaine, en particulier au cabinet du président Bush. Si un défi est réalisé, CodeWeavers s'engage à fournir gratuitement pendant 24h des licences de ses logiciels CrossOver. Ainsi Jeremy White a lancé [5 défis à réaliser avant 2009](#) :

- o réduire le prix de l'essence (en dessous de 2,79\$/gallon)
- o réduire le prix de la nourriture (le lait doit être en dessous de 3,5\$/gallon)
- o créer des emplois (le nombre d'emplois doit être supérieur à celui du début de l'année, soit 138 002 000 personnes aux Etats-Unis)
- o faire revenir le marché de l'immobilier (l'immobilier en moyenne de 12%. L'objectif est de revenir au prix moyen du début de l'année soit 233 000 \$ pour une maison moyenne)
- o traduire Oussama Ben Laden devant la justice

Si le dernier défi semble très peu réalisable, la crise financière qui secoue le monde aura raison du premier défi. En effet, le baril du pétrole a baissé jusqu'à 60\$, faisant passer le prix du gallon en dessous des 2,8\$. Par contre, la crise devrait faire échouer 2 défis (emploi et immobilier) mais le défi du lait pourraient être tenu. Le défi étant réalisé, Jeremy White a mis un nouveau défi en ligne : faire remonter les marchés aux plus haut de l'année 2008. C'est encore gagné! CodeWeaver édite une série de logiciel payant permettant d'utiliser des logiciels Windows sur [Linux](#) et/ou Mac OS X sans installer le système de Microsoft sur le disque dur. CrossOver utilise le projet Wine (Wine Is Not an Emulator) pour mener à bien son oeuvre. La gamme CrossOver est composée de 3 logiciels :

- o CrossOver Mac qui permet d'exécuter la plupart des logiciels Windows sur Mac OS X Intel (licence standard ou Pro)
- o CrossOver Linux qui permet d'exécuter la plupart des logiciels Windows sur une distribution Linux (licence standard ou Pro)
- o CrossOver Games qui permet d'exécuter la plupart des jeux Windows sur Mac OS X ou sur Linux.

Le prix des logiciels est de 37€ pour les éditions standard de CrossOver Mac et Linux ainsi que CrossOver Games (Mac/Linux) et de 64€ pour les éditions Pro de CrossOver Mac et Linux. Pour obtenir la licence gratuite, il suffira de se rendre sur le site de [CodeWeaver](#) durant la journée du 28 octobre (heures américaines) qui donnera automatiquement un numéro de série valable sur les produits CrossOver Pro. Il faudra l'enregistrer sur le site de CodeWeaver pour le rendre définitif. Actuellement, le site subit l'assaut des communautés Mac et Linux. Il est possible d'obtenir le numéro de série mais pas encore de l'enregistrer. Bonne chance!

(from EchosDuNet) Assez déliant je trouve comme concept.

## Mise à jour du firmware de l'iPhone pour bientôt

La rumeur tourne depuis quelques heures dans la communauté des fans d'Apple dont je fais parti : une nouvelle version du firmware pour l'iphone est en préparation. La version 2.2 beta 2 apporterait enfin le support du téléchargement des podcasts directement sur l'iphone, une amélioration de Google Maps avec l'intégration de Google Street View. On note aussi l'arrivée de Google Public transportation qui donne les horaires de train, de bus, pour les USA uniquement. Il sera possible de partager avec un groupe d'amis sa géolocalisation. En quelques sortes, un moyen simple pour se retrouver dans Paris entre amis. Nous pourrons aussi envoyer par email sa géolocalisation. Cependant avec Twitterrific dont je vous ai déjà parlé, c'est déjà possible. L'entrée ligne de l'iphone sera aussi activée afin de pouvoir y brancher un micro, ce qui en fera un enregistreur numérique tout à fait correct pour faire un podcast mobile par exemple. Je m'imagines très bien à Devoxx vous commenter ma journée et envoyer directement vers le blog mon podcast audio pour tout le monde... le tout via l'iPhone. **Références :**

<http://theappleblog.com/2008/10/27/new-features-in-iphone-software-22/>

<http://www.tuaw.com/2008/10/28/rumor-roundup-iphone-2-2-firmware-details/>

<http://www.macrumors.com/2008/10/28/apple-seeking-to-increase-participation-in-app-store-ratings/>

## Evénements Java sur Paris pour novembre

Petit point sur l'actualité des événements pour le mois de novembre : **Conférence Open Source Exchange** - le 12 novembre Xebia et SkillsMatter organisent une journée de présentation et de conférences à Paris, gratuitement. Nous y retrouverons que des têtes connues, jugez plutôt : [Guillaume Laforge](#) pour Groovy, [Alexis Moussine-Pouchkine](#) pour Glassfish, [Antonio Goncalves](#) pour JAVA EE 6, [Michael Isvy](#) de SpringSource pour parler de Spring dm Server, bref que du beau monde. Plus de détails sur le site de [Xebia](#) ou sur le site de [SkillsMatter](#). L'événement aura lieu à la FIAP où nous avons assisté avant l'été aux présentations sur Spring, sur Adobe Flex et sur MDA. J'espère pouvoir y participer. Je vous tiendrai au courant. **Rencontres Spring 2008** Le lendemain, le 13 novembre la société SpringSource France et SFEIR, organise [l'événement Spring de l'année](#), avec la venue de Juergen Hoeller (l'un des auteurs de SpringSource), de Peter Cooper-Ellis et de Mark Thomas. L'événement se déroule de 08h30 à 13h00 à la Défense, l'inscription se fait via le site <http://www.rencontres-spring.com/> Voir aussi sur le blog de SFEIR de Didier Girard [ce billet](#). **Soirée Paris JUG sur GWT** Mardi 4 novembre, la soirée du Paris JUG aura pour thème Google Web Toolkit. Voir <http://www.parisjug.org/xwiki/bin/view/Meeting/20081104> **Petit-déjeuner OCTO** Le mardi 18 novembre OCTO organise un petit-déjeuner sur l'Agilité et le retour sur expérience de la part de personnes de la maîtrise d'ouvrage. [http://www.octo.com/com/com\\_petitdej181108.html](http://www.octo.com/com/com_petitdej181108.html) N'hésitez pas à me faire part (et à m'inviter) à vos événements

## Scrum : une histoire de lave-linge

J'ai trouvé la métaphore ultime pour expliquer à un client le fonctionnement des Sprints de Scrum. Comme vous le savez le Client, le Scrum Master et l'Equipe se réunissent au début de chaque Sprint au cours d'une réunion de 2 heures afin de fixer la liste des développements pour le prochain Sprint. Autour de la liste des fonctionnalités à développer (le product backlog) l'objectif en 2 heures est que le Client finalise avec l'Equipe la liste des développements à venir pour le prochain Sprint. Ensuite un des principes de Scrum est que lorsque l'équipe développe, elle ne doit pas être interrompue. Ce principe permet à l'équipe de travailler efficacement sans être perturbée en permanence par ce que j'appelle des "fausses urgences". Le Client est en quelque sorte tenu de respecter cette règle. Et croyez-moi, au final cela fonctionne mieux que d'interrompre en permanence une équipe de développement. Revenons au titre de ce billet. Comment expliquer à un Client que l'équipe ne peut produire qu'une quantité limitée d'éléments dans les 2 semaines à venir ? Comment lui expliquer qu'il doit alors faire un choix et donner une priorité pour chaque élément ? Comment lui expliquer qu'il ne pourra pas changer le contenu du sprint ? Prenez une machine à laver le linge. Vous avez un tas de linge sale : voici votre Product Backlog. Votre famille "alimente" en permanence ce Product Backlog chaque jour. De la même façon en Scrum les clients peuvent mettre à jour le product backlog en permanence. Ensuite prenons notre lave-linge : il ne peut laver qu'une certaine quantité de linge à chaque programme. Par exemple 5 Kg. Et bien de la même façon, une équipe ne peut produire qu'une certaine quantité de fonctions par itération. Un Sprint de Scrum est donc égal à un lavage en machine. Pour expliquer à un client pourquoi l'équipe refuse de prendre plus d'éléments, je me sers de cette image. Je peux aussi lui dire que sa voiture n'a que 40 litres d'essence dans le réservoir. Il doit s'en contenter. Il faut donc que le client comprenne que "la capacité de production est limitée" et que c'est l'équipe qui annonce cette

capacité. Ensuite regardons ensemble ce que fait quelqu'un qui prépare une nouvelle machine : il prend le tas de linge sale et ensuite il va le trier en prenant soin de laver en premier les vêtements qu'il lui faut pour le lendemain, et il écartera les vêtements qui peuvent attendre la prochaine machine. C'est de la gestion de priorité. Tout comme Scrum. Je pousserai bien le vice jusqu'à expliquer qu'un Sprint de callibrage où l'on recherche une solution, un Sprint de release où l'on prépare une version, me font penser à un programme "Coton" et un programme "Blanc Programme Long"... Enfin vient le temps du lavage, ce qui nous permet d'expliquer 2 notions : la non-modification d'un sprint et la durée fixée. Tout d'abord il n'est pas possible d'arrêter un programme en cours de lavage pour y ajouter une chemise oubliée. Tant pis pour vous, elle sera lavée avec la prochaine machine. Même pire, je pense que si vous pouviez mettre en pause un programme, cela serait un risque pour l'ensemble du linge déjà en cours de lavage : risque de surcharge de la machine, risque que la chemise ne soit pas bien lavée finalement... Durant un Sprint de Scrum il n'est pas possible de changer le contenu du Sprint. Vous devez laisser l'équipe terminer son travail. La gestion des patches et des bugs urgents doit être prévue en dehors du temps alloué à Scrum. Si votre activité de correction de bug représente 4 jours sur 10 en moyenne, alors votre équipe ne doit "faire du scrum" que sur une base de 6 jours, même si le sprint dure bien 10 jours. La durée fixée en Scrum est aussi très importante car elle permet par observation de suivre l'activité de l'équipe et de créer des statistiques afin de pouvoir ensuite fonctionner de mieux en mieux. Enfin je ne pousse pas plus loin la métaphore, pensez à Scrum lorsque vous lancerez votre prochain programme en "Coton délicat 40 degrés avec essorage"

## Soirée GWT et Restlet au Paris JUG

Petit compte-rendu cher lecteur ce soir. D'abord il est tard (dans les 01h36 du matin) et ensuite je dois me lever tôt pour le boulot (dans les 06h50 du matin aussi). Tout d'abord j'ai fait 3 soirées ce soir : tout d'abord la présentation de Didier Girard sur GWT qui était vraiment sympa, la présentation de Jérôme Louvel sur RESTlet et enfin la troisième mi-temps avec toute l'équipe du JUG et une dizaine de geeks autour d'une petite bière. De l'avis de tout le monde, Didier a donné l'envie d'essayer GWT. La présentation rondement menée nous a permis de voir ce qu'est GWT. Cher lecteur ce soir je suis fainéant pour la première fois... Je ne vais pas bloguer des pages et des pages, je vous propose en attendant de relire des billets plus anciens sur GWT comme [celui-ci](#) ou [celui-la sur Ext-GWT vs GWT-Ext](#). Comme discuté avec Antonio, cela fait plaisir d'avoir quelqu'un qui apporte son expérience et partage sa passion sans concessions. En effet, Didier étant directeur technique de SFEIR, il parle en connaissance de cause, ce qui m'a intéressé. Nous avons vu entre autre un plugin dans Eclipse facilitant l'écriture de code pour GWT, que GWT fonctionne avec Java 5 depuis le 28 août 2008, et un ensemble d'URL de sites d'exemples qui montrent GWT en action. Mon avis sur la question c'est que GWT est destiné à réaliser des applications riches, et qu'en terme de productivité c'est un moteur puissant qui permet facilement de construire son application. Maintenant d'un point de vue web, je pense qu'il serait dommage de penser que GWT permet de faire une application type "vente de billet de trains"... Mais j'aimerais être convaincu du contraire. En terme de mise en place, là où Adobe Flex propose avec BlazeDS et LiveCycle des solutions du côté du serveur, il n'y a pas de solutions en GWT, car on parle bien d'un framework de présentation. Je trouve par contre génial ce concept d'écrire du Java, de débiter dans Eclipse (ou IDEA IntelliJ) puis ensuite seulement de produire son code Javascript pour chacun des navigateurs, chacune des langues supportées. Google est sur une niche et ce concept est réellement novateur. Donc je reste fortement intéressé mais mon coeur balance plus pour Flex. Quitte à faire une application type client riche, je préfère travailler avec Flex. Cela me coûte une phase d'apprentissage du langage ActionScript, mais je gagne quelque chose par rapport à GWT, qui est le support de différents protocoles d'échanges entre le client et le serveur. Soit un échange de XML basique, soit du JSON, soit encore mieux, un format binaire propriétaire à Adobe qui permet d'optimiser encore plus la bande passante et les échanges entre le client et le serveur. Merci en tout cas à Didier et vous pouvez retrouver [sur son site](#) les URLs des démonstrations. Ensuite un petit break pour le buffet organisé par la société [Novedia Solutions](#). Honnêtement merci à eux car j'ai réussi à me caser derrière le buffet avec [Cyrille Leclerc](#), Antonio Goncalves, [Thomas](#) et Alexis Moussine-Pouchkine. A nous 4 nous avons appliqué le pattern "Interceptor" et "Introspection" sur les plats de petits fours, que du bonheur. Ma conclusion c'est que Spring n'a rien inventé. Au lieu de devoir faire un get et un set sur les petits fours, le serveur nous a tendu le plat et nous n'avions plus qu'à nous injecter ces bons petits gâteaux. Vive l'inversion de contrôle. Bon reprenons la suite. Jérôme Louvel, membre de l'OSSGTP et auteur de Restlet nous a fait une démonstration plutôt pointue de Restlet. Tout d'abord ce que j'ai compris : Restlet est un framework léger REST en JAVA qui offre un système du côté client comme du côté serveur pour accéder à des Ressources. Je trouve que l'architecture REST est franchement intéressante, et si vous voulez relire une présentation à ce sujet, [voici un billet que j'aime bien](#) écrit il y a quelques mois sur REST. Parlons un peu des choses qui fâche, désolé Jérôme si tu me lis. La présentation était vraiment très complète, voir un peu trop détaillé je pense pour le sujet de ce soir. Les efforts de migration et de portage vers la dernière version de GWT, quoique vraiment intéressant d'un point de vue technique, ont peut-être fait décrocher une partie de la salle. Je pense qu'il est difficile en une heure de parler à la fois de Restlet et à la fois de GWT, c'était le format de la soirée qui donnait peut-être le cadre. En tous les cas la présentation nous a donné un bon aperçu de Restlet et j'aurais aimé te poser des questions sur les annotations. J'ai pensé un moment que Restlet se propose d'être un 'ESB' à la fois du côté du client et du côté du serveur, en simplifiant l'accès aux ressources et en utilisant les principes de REST. Durant la présentation, tu as cependant bien expliqué que grâce à Restlet, une application de type webmail a ajouté le support de GWT pour un coût de développement léger. Le système de sélection du type de rendu selon la source était bien expliqué. Après la présentation nous nous sommes retrouvés au Falstaff à côté de l'ISEP autour d'une bonne bière. Moment agréable où j'ai écouté pas mal de monde, très sympa. Sur ce, je vous donne rendez-vous la semaine prochaine pour un billet spécial pour fêter l'anniversaire du Touilleur Express ainsi que des nouveautés autour de Spring dès jeudi 13 novembre après la journée "[Les Rencontres Spring](#)". Je vous en reparlerai cette semaine.

## Barcamp2 et photos de la soirée du Paris JUG sur GWT Restlet

**Le BarCamp est de retour** Avis à la population : mardi 9 décembre aura lieu le deuxième BarCamp organisé par Luc Bizeul chez OCTO Technologies à Paris sur les Champs-Élysées. C'est un événement que je vous recommande. Pour une fois vous pouvez venir et participer en étant acteur. J'ai vraiment bien aimé le premier BarCamp et je vous le conseille fortement. A noter que c'est ouvert à tout le monde : Java, Python, Flex... tout le monde est bienvenue. Le nombre de place est limité à 60 personnes, premier inscrit, premier servi. Pour vous inscrire, il faut tout simplement éditer la page du Wiki à [cette adresse](#). Pour cela un code est donné sur la page d'inscription, en bas à gauche. Il faut être attentif. Je me suis mis en "pas certain de venir" car le lendemain je pars à Devovx à Anvers. Cela risque de faire un agenda un peu trop chargé. Voir aussi le compte-rendu [du premier barcamp](#) pour vous faire une idée de la dernière soirée. **Photos du Paris JUG** J'ai envoyé deux photos hier soir prises en direct du Paris JUG avec l'iphone et Twitter. Cliquez sur la photo pour la visualiser en plus grand. Photo 1: Didier Girard, présentation de GWT



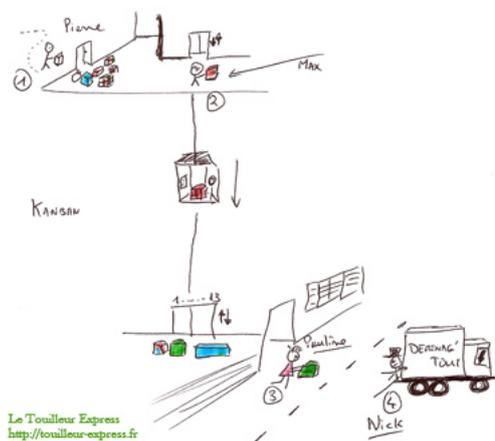
Photo 2 [d'Olivier Croisier](#) assis à côté de [Florent Ramière](#)



Je crois même apercevoir [Fabrice Robini](#) à côté de Florent. Enfin hier je me suis rendu compte qu'il faut que je mette une photo de ma tête. [Zouheir](#) m'a raconté hier soir qu'il ne savait pas que le Touilleur... c'était moi ! Marrant :-)

## Kanban ou Scrum ?

Scrum est un excellent outil léger pour gérer un projet et une équipe. Là où il n'y a aucunes méthodes de développement, c'est un principe simple pour mettre en route un projet. C'est réellement une méthode qui permet de créer de la valeur. Cependant je m'intéresse aussi à d'autres méthodologies de la communauté. J'ai entendu parler de Kanban et de Lean development il y a quelques mois mais jusqu'à récemment, c'était pour moi du chinois. Voyons en quoi consiste ces méthodologies. J'essaye ici de vous parler de concepts et de principes que vous pouvez mettre en place dans votre équipe. La méthode Kanban est une méthode industrielle de production ayant pour origine le Japon. Cela signifie "étiquette" en Japonais. Inventé dans les années 60 par Toyota, le principe est le suivant : une étiquette est attachée à chaque élément produit par l'équipe de développement. Lorsqu'un développement se termine, l'étiquette est retournée dans une liste d'attente. Lorsque le nombre d'étiquettes dans la liste dépasse un seuil, l'équipe de développement reprend alors une étiquette et l'attache à un nouvel élément à produire. Cela revient donc à limiter la production d'une équipe à un seuil. Limiter la production d'une équipe ? Cela peut sembler étrange non ? Je vais illustrer le principe par une image. Samedi prochain vous allez effectuer un déménagement. Voyons d'abord la méthode inadaptée pour effectuer ce déménagement, que j'appelle le "chacun pour soi" : vous prenez un carton, descendez 3 étages, traversez la rue, mettez le carton dans le camion. Au passage je signale qu'il est difficile de monter dans le camion et de le ranger au fond car vos amis font la même chose, et don c'est un peu compliqué. Tout ceci n'est pas très optimisé non ? Le principe de Kanban est de limiter la production, de laisser chacun être proactif en allant "piocher" une tâche à faire dans sa boîte d'attente. Pour revenir à notre déménagement voici comment j'explique cela : imaginez qu'avec 5 personnes vous vous répartissez le parcours de l'appartement au camion.



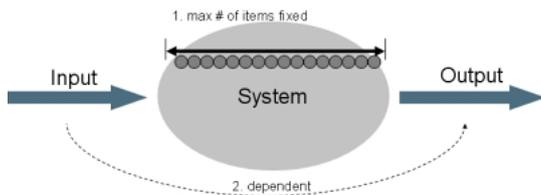
Pierre va chercher les cartons dans

l'appartement et les place devant la porte. Max appelle l'ascenseur, le charge, et l'expédie au rez-de-chaussée. Pauline vide l'ascenseur et amène les cartons au camion. Enfin Nick se charge de ranger les cartons dans le camion de manière optimale. Tout d'abord le principe vous l'avez compris : nous divisons les étapes du développement du logiciel. Chacun est responsable d'une partie, et travaille de manière autonome. Au lieu d'essayer de faire le parcours du déménagement complètement (de l'appart au camion) nous divisons l'activité globale en petites activités. Cela évite par exemple que 3 personnes essayent d'utiliser l'ascenseur en même temps. Au final le déménagement va plus vite. C'est le premier principe de cette méthode, chacun travaille sur une petite activité au lieu que tout le monde travaille sur tout le processus de production. Le deuxième principe : Pierre va plus vite pour vider l'appartement car il n'a pas beaucoup de chemin à faire. Pensez-vous qu'il doit alors entasser les cartons devant la porte ou doit-il attendre que Max ait chargé l'ascenseur ? La réponse Kanban est qu'il doit attendre. En effet, rien ne sert de déplacer un engorgement, il faut terminer un nombre limité d'activités, et surtout éviter de se lancer dans différents chantiers. Le signal qu'il peut reprendre sa production, c'est qu'un des cartons a été pris par Max qui charge l'ascenseur. Sinon il attend devant la pile de carton. Kanban est une méthode de restriction de la production par la consommation. Imaginez ce principe appliqué au développement d'un ensemble de fonctions dans un projet informatique. Tout d'abord rien ne vous empêche d'ajouter une gestion de priorité afin que les éléments les plus importants soient développés en premier. Ensuite ce que je trouve élégant, c'est que le système s'auto-régule. Je cherchais une image afin d'expliquer ce principe et j'ai pensé à la gestion d'une file d'attente. Dans un restaurant de restauration rapide américain, chacune se place devant un guichetier et attend son tour. C'est le principe de la distribution, que vous expérimentez aussi dans un supermarché lorsque vous faites la queue pour payer. L'inconvénient de ce système c'est le taux de passage, que des scientifiques ont calculés. L'alternative qui marche est de ne créer qu'une seule file d'attente puis ensuite de dépiler non pas plusieurs personnes en attente par plusieurs unités de traitement, mais bien une seule file. Ce principe est appliqué dans les administrations, à la Poste, à Orly devant les guichets Air France pour l'embarquement... Chacun attend dans une file unique, et dès qu'un guichet se libère, vous allez rapidement à ce guichet, et la file avance. Il a été prouvé que ce système de gestion est bien performant qu'un système de queue distribué. Revenons à la méthode Kanban. Afin d'illustrer ce système, les personnes qui réfléchissent à cette méthode proposent de ne faire qu'une seule file de tâches à faire pour toute l'équipe au lieu de distribuer aux membres des tâches. Si vous voulez, au lieu d'affecter des JIRA à chacun, et que cette pile de JIRA s'accumule sur le bureau de chaque développeur, vous laissez votre file unique en place. Ensuite en adoptant le principe de Kanban, c'est le développeur qui viendra chercher la tâche à effectuer, dès qu'il aura terminé son activité. La liberté de traiter plusieurs activités en même temps est finalement un frein à la productivité. Si rien ne me contraint à terminer ma tâche, et que je peux commencer un autre développement, je commence alors à ne plus gérer correctement mon travail. Répondez-moi honnêtement : est-ce qu'aujourd'hui vous avez travaillé sur une seule activité ? Oui ? pourtant ce bug urgent, cet email où l'on vous a demandé de vérifier un bug, cette explication de 20 mn à un collègue, la discussion avec un autre ami sur l'idée de mettre en place Maven ou pas... En fait notre journée est coupée d'un grand nombre d'événement qui cassent le rythme de production et votre énergie pour travailler. C'est un fait. Et à part travailler dans un aquarium, il faut s'en contenter. Par contre sur la gestion des tâches d'un projet, comme l'explique [Johanna Rothman](#) sur son blog, une organisation matricielle multi-tâche est l'anti-pattern de l'Agilité. Elle explique en substance que certains managers s'évertuent à gérer leur efficacité en répartissant les tâches entre les développeurs. Ils ne cherchent pas à améliorer le débit de production de l'équipe. L'un des principes de Kanban est qu'il faut terminer une tâche avant d'en commencer une autre. D'où l'importance de définir la notion de "terminé". J'y pense tout le temps. L'un des soucis dans la communication entre développeurs, c'est la définition du "j'ai terminé". En tant que développeur, je fais mes tests unitaires, je code et je valide avec la dernière version du code. En tant que chef de projet j'aimerais en plus que tu vérifies sur la version d'intégration que ton code fonctionne, et j'aimerais que tu mettes un peu de documentations sur le Wiki. Enfin en tant qu'ingénieur qualité, j'attends de l'aide pour tester la nouvelle fonction... Pensez à bien définir la notion de "terminé" et vos estimations seront plus justes. Comme le suggère Johanna, si votre manager vous demande de faire une nouvelle tâche, demandez-lui la priorité de cette tâche par rapport à celle que vous faites. S'il ne sait pas répondre, c'est qu'il n'a pas alors rempli son rôle de manager. Son boulot c'est d'identifier la priorité. Pas le vôtre. Le vôtre selon le principe de Kanban c'est de ne porter qu'un seul carton de déménagement à la fois, car vous avez alors plus de chances de terminer votre tâche, de la réussir et d'éviter de vous épuiser. C'est en regardant le principe du supermarché, où chacun se sert, que [Taiichio Ono](#) a eu l'idée de la méthode Kanban. Il a ainsi proposé un système où les producteurs viennent chercher les tâches à faire, au lieu qu'un manager affecte ces tâches. Les 6

règles du principe Kanban sont : - Les clients procèdent les éléments en respectant la quantité spécifié dans le Kanban - Les fournisseurs en amont, produisent alors de nouveaux éléments du nombre exact et dans l'ordre spécifié sur le Kanban - Aucun élément n'est déplacé sans être attaché d'un Kanban (étiquette) - Un Kanban doit accompagner chaque point tout le temps - Les défauts les montants incorrects ne sont jamais transmis à l'étape suivante du processus, on les jette. - Le nombre de Kanban peut être réduit afin d'identifier plus facilement les problèmes. **Le Kanban rapporté au développement informatique** Après tout ceci on peut donc définir ce processus comme étant une carte, une fiche, que l'on va attacher à une activité. Cette carte est un droit de travailler sur une activité, et le fait que chacun ne puisse en traiter qu'un nombre limité, limite la production. Chaque développeur est proactif, il notifie le producteur (le chef de projet) si la pile des Kanbans se vide. En principe cette carte de production doit lui donner toutes les informations pour réaliser sa tâche. Je reconnais qu'en pratique, dans le cadre d'un développement informatique, c'est peut-être moins évident. Pour résumer : chaque activité ne peut être complétée que si un Kanban est attaché. De manière concrète, imaginez 10 drapeaux pour une équipe de 15 personnes. Il est interdit de dépiler un élément de la liste "A FAIRE" tant qu'un drapeau "TERMINE" n'est pas disponible. Voici un dessin qui illustre cela

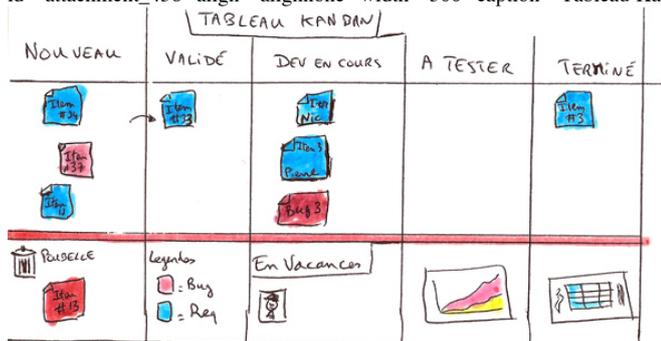
Kanban;

1. Makes the max # of items fixed in the system
2. Makes the input velocity depend on the output



**Le Tableau Kanban** Comme nous

avons vu, les Kanbans sont donc à la fois des marqueurs et des témoins que l'équipe se passe en remontant le flux de production. Dans le développement d'un logiciel, le plus simple est de mettre en place un grand tableau sur un mur, de prendre quelques post-it puis ensuite de tracer plusieurs colonnes, selon les étapes de votre production. Voici un exemple simple de Kanban : [caption id="attachment\_458" align="alignnone" width="500" caption="Tableau Kanban"]



<http://touilleur-express.fr> [/caption] L'usage est

de prendre un code couleur afin d'identifier visuellement les catégories des activités. Par exemple la correction d'un bug urgent sera sur une fiche rouge. Une amélioration pour un client sera sur une fiche bleu. Une amélioration proposée par un développeur sera sur une fiche jaune... Ensuite le principe est de respecter la taille limitée du tableau. Chaque colonne ne peut accepter qu'une quantité limitée de post-it. A partir de ce principe, vous venez de refaire le système de déménagement dont nous avons parlé au début : vous limitez le nombre d'éléments en cours de production, vous forcez chacun à terminer une tâche, vous identifiez visuellement l'activité de l'équipe, et vous pouvez afficher d'autres éléments comme des statistiques de WIP (Work In Progress) afin de montrer à votre management l'avancement de l'équipe. Enfin, l'une des valeurs de l'Agilité est la transparence. Et ce tableau expose à tout le monde votre activité, il permet de créer de la communication, et donc au final je pense qu'il améliore le développement. Maintenant pour répondre à la question, Kanban ou Scrum ? Je pense qu'il faut les 2. **Références:**

- <http://www.agilemanagement.net/Articles/Papers/KanbanAtLeanNPD.pdf>
- <http://www.infoq.com/articles/hiranabe-lean-agile-kanban>

## Terracotta et SpringSource

Bon et bien si c'est bien la nouvelle dont je vous avais parlé, comme pronostiqué [il y a quelques semaines](#), Terracotta et SpringSource ont annoncé cette nuit un partenariat. [Le communiqué de presse](#) ne donne pas plus de détails, simplement la mise en place d'un partenariat. Je pronostique donc que jeudi prochain l'invité surprise sera Amit Pandey, le CEO de Terracotta. Ou peut-être Ari Zilka ? Bref en tous les cas j'avais un peu exagéré en demandant à Ari Zilka si SpringSource allait racheter Terracotta, mais je n'étais pas si loin de la vérité... Honnêtement je m'attendais à un rachat, ce qui sera sans doute la prochaine étape pour SpringSource. Oracle rachète SpringSource... Soyons fou ! [Update : finalement SpringSource annoncera le rachat de G2One inc. Groovy and Grails quelques heures plus tard]

## LA grosse nouvelle : SpringSource rachète

## G2One Groovy

Merci à [Didier Girard et Twitter](#), finalement c'est grâce à lui que ce matin j'ai appris la nouvelle dont nous avait parlé Julien il y a quelques semaines : [SpringSource rachète la société G2one](#), la société qui édite le langage Groovy et l'outil Grails. Guillaume Laforge sera donc présent jeudi matin aux "[Rencontres Spring 2008](#)" après sa participation la veille à la journée "[Open Source Exchange](#)" organisée par Xebia et Skills Matter.

## Compte-rendu des Rencontres Spring 2008

SFEIR et SpringSource ont organisé l'événement "Rencontres Spring 2008" aujourd'hui à Paris. Après un accueil sympathique (teeshirt, carnet, stylo... le tout estampillé SpringSource) la salle s'est rapidement remplie. Didier Girard, directeur technique de SFEIR prend la parole. Tout d'abord gros succès, plus de 400 personnes inscrites, et environ 230 personnes présentes, la salle était pleine. SFEIR c'est d'abord 250 collaborateurs, environ 22 millions de CA, et depuis 1997 des développements Java au forfait et en régie. Je referme là la parenthèse "pub" mais il faut reconnaître que l'organisation était impeccable. [caption id="attachment\_475" align="alignnone"



width="300" caption="La foule était là"/> [caption]

**Introduction par Didier Girard** Didier commence par expliquer pourquoi Spring, pourquoi le framework est maintenant un acteur incontournable de la communauté Java. Tout d'abord un framework d'inversion de contrôle. Peu de personnes saisissent, mais il faut se souvenir qu'au temps des serveurs J2EE "lourd", celui-ci était en charge de la vie et de la mort de vos composants. Spring framework a retourné ceci, en redonnant une agilité au code par l'injection des dépendances, d'où l'inversion de contrôle. Ce n'est plus le serveur qui vous contrôle mais bien vous. La programmation orientée aspect est ensuite le deuxième point que Spring a facilité, je pense même démocratisé. L'ensemble des connecteurs de la couche Spring permet enfin d'intégrer des services tiers facilement. Enfin Spring est indépendant des éditeurs de logiciel, et indépendant des spécifications de Java, bref Spring est libre. [caption id="attachment\_476" align="alignnone" width="300" caption="Julien Dubois et Didier Girard"]



[caption] Didier parle ensuite de la démarche de SFEIR par rapport à la veille technique. Une cellule d'étude se charge de surveiller le marché afin d'assurer un niveau technique de qualité. Concernant Spring et le framework, il explique aussi l'apport en terme d'industrialisation. Comment séduire les personnes de la production. Il parle là des équipes qui gèrent aujourd'hui vos applications en production. Et comme il le dit justement, il va falloir expliquer et rassurer, afin d'appuyer le fait que Spring allège la complexité de la couche J2EE. L'équipe de production est souvent là pour s'assurer de la stabilité et de la pérennité des solutions. Faire certifier par votre équipe de production un serveur comme BEA Weblogic demande du temps. Spring se propose d'apporter plus d'agilité et de réduire cet effet "mamouth" que malheureusement l'architecture J2EE d'ancienne génération nous a légué. Il faut être réaliste : nous ne sommes pas forcément très populaire avec nos applications de première génération EJB2.1 et compagnie. Didier explique qu'il préfère parler "Java du côté serveur" plutôt que d'utiliser le mot "J2EE" car Spring c'est parfois aussi des déploiements plus simples, avec une petite empreinte mémoire de quelques mégas là où une pile serveur d'app complète vous demande 1 Go...

**Questions de la salle** Une question pertinente : SpringSource propose Spring dm Server, un serveur d'application. Cela ne va-t-il pas à l'encontre de la démarche de simplification de Spring ? Didier répond à ce sujet que Spring dm Server ajoute une flexibilité qui n'existe pas avec les serveurs classiques. La suite de la présentation par Julien répondra à cette question. **Julien Dubois, Regional Director for France** Julien présente l'activité de la société SpringSource. Avec environ 150 personnes à travers le monde, SpringSource se charge du support et des formations sur l'ensemble du framework Spring ainsi que Tomcat et Apache. La souscription aux services de SpringSource donne accès à un support de très haut niveau, constitué essentiellement par les développeurs de Spring. Grâce à l'architecture de support de la société Covalent, rachetée au début de cette année, SpringSource a maintenant une plateforme très efficace de support. Notez bien qu'en écrivant ces lignes, je fais de la publicité pour SpringSource, mais qu'il est important de dire qu'il existe une entité derrière les JAR que tout le monde manipule. C'est important. **Guillaume Laforge, chef de projet de Groovy et responsable de G2One.inc** La nouvelle annoncée mardi dernier est que la société SpringSource rachète G2One.inc, la structure créée par Guillaume, Graeme Rocher et Alex Tkatchman, ancien COO de JetBrains (IntelliJ) pour Groovy et Grails. Et donc Guillaume est venu sur scène afin d'apporter quelques informations. Pour l'essentiel, Groovy

est un langage open-source sous licence Apache 2 (et qui le restera). Grails est un framework léger web de génération similaire à Ruby on Rails.... mais en mieux ! Groovy est un langage de scripting dont je vous ai parlé à l'occasion de [la présentation de Guillaume](#) au Paris JUG. Grails et Groovy sont utilisés par exemple par la société LinkedIn.com pour développer des sites spécialisés pour des clients de LinkedIn. Guillaume rappelle quelques concepts clés de Groovy, tel l'usage des closures, présent depuis le début dans la syntaxe du langage. J'ai eu l'occasion d'interviewer ensuite Guillaume, je reviendrai donc sur certains détails dans un autre article. **Présentation de Peter Cooper-Ellis** Peter est l'une des personnes les plus importantes de SpringSource. Il est vice-président de Engineering and Product Management. C'est un ancien de BEA, et l'une des personnes qui s'est occupé de la mise en place et de la réussite du serveur d'application Weblogic Application Server. Il a rejoint SpringSource avant l'été et s'occupe de la roadmap et des produits de SpringSource. C'est donc une présentation très axée sur les produits professionnels de SpringSource, mais aussi quelques exclusivités mondiales... [caption id="attachment\_478" align="alignnone" width="300" caption="Peter Cooper-Ellis"]



[/caption] Sa présentation s'est axée sur 3 thèmes :

"Develop, Deploy and Support". SpringSource c'est avant tout le développement du framework, ensuite sa mise en place et enfin le support après vente. SpringSource est une compagnie open-source avec une offre commerciale, pour ajouter de la valeur au socle de base, qui est libre, gratuit et open-source. Clair, simple et limpide. Ensuite Peter a détaillé la roadmap des 12 mois à venir, qui est loin d'être définitive. J'ai réussi à prendre une photo de ce slide qui contient quelques révélations sur ce que SpringSource va proposer dans les mois à venir. [caption id="attachment\_477" align="alignnone" width="300" caption="SpringSource roadmap 2008 2009"]



[/caption] Ce que j'ai retenu : Spring dm Server sera

prêt en version M1 d'ici fin novembre, l'objectif est de sortir une version RC1 en janvier et enfin une version officielle vers le mois d'avril. Spring 3.0 est annoncé pour janvier (regardez la photo) et [Jurgen Hoeller](#) reviendra plus tard sur le contenu de ce que contiendra Spring 3.0. Enfin on note que les slides sont à jour puisque Groovy et Grails font partis de la présentation. Je me dis qu'à l'issu de cette présentation, nous avons un aperçu assez précis de ce que SpringSource va proposer d'ici à mi-2009. A noter l'apparition d'un projet dont je n'avais jamais entendu parler : Spring ROO qui est un générateur d'application basé sur les conventions par dessus la configuration. Impossible de savoir par rapport à Grails la marge de recoupement, s'il y en a une. A noter ensuite que grâce à la présence de Nicolas Romanetti de Celerio, nous avons parlé de génération de code. C'est un sujet sur lequel nous reviendrons dans les mois qui viennent. Je connais AppFuse, Seam-Gen, Celerio, Grails et donc maintenant Spring ROO. Il semble que la communauté se rend compte qu'écrire tout le temps la même chose n'est ni intéressant ni enrichissant. Passé les 3 minutes d'excitation devant son fichier hbm, honnêtement je préfère utiliser Celerio et bosser sur le contenu (notez que je fais une bonne grosse pub pour Celerio, mais on en parle pas assez, alors que c'est un très bon projet). Les questions portent ensuite sur "mais pourquoi vous faites un serveur d'app vous aussi ?". A cette question, la réponse de Peter est assez simple et sans appel : l'empreinte mémoire d'un weblogic, c'est 1Go, Spring et Tomcat, 6 Mo. Spring dm Server s'inscrit dans la démarche de simplification. Il semble simple à utiliser, je vous en parlerai prochainement car je l'ai testé la semaine passée. **Mark Thomas, Tomcat Tuning** Mark de retour de l'Apache Con et donc passablement "jet-lagué" nous a fait une présentation très technique sur les techniques d'optimisation de Tomcat. Avec 5 ans d'expériences sur Tomcat, 1500 bugs corrigés à son compteur, il sait de quoi il parle. Il présente d'abord le processus à appliquer pour faire du tuning de Tomcat. Je vous le résume : - Stabiliser le système, ne pas toucher au code durant la phase de tuning - Se donner des objectifs de temps de réponse à atteindre afin de savoir s'arrêter - Mesurer l'état actuel du système avec des scénarii réalistes - Identifier les goulots d'étranglement - Fixer en premier ce qui est le plus évident - Répéter jusqu'à atteindre l'objectif souhaité Les erreurs souvent rencontrées sont que l'on tend à optimiser trop de code, à ne pas tester assez la plateforme avec des données réalistes, à ne pas se fixer d'objectif précis pour effectuer ce tuning, à deviner où se trouvent les soucis au lieu de vraiment chercher et trouver, à fixer les symptômes plutôt que la cause... Par expérience il explique que 80% du tuning qu'il a eu à faire s'effectue dans l'application Java elle-même. Pour les 20% il y a maintenant de bonnes pratiques, afin d'éviter les soucis souvent rencontrés avec un serveur Tomcat mal configuré. Cela me rappelle un client à l'INA qui m'expliquait qu'il relance chaque matin ses serveurs. Par défaut Tomcat est livré en mode

"développeur" et il faut donc l'adapter et le régler avant de le passer en production. Je vous passe les détails de réglages et de tuning, même si c'était intéressant. J'espère que Didier Girard nous donnera les slides afin que chacun puisse relire les idées de Mark. **Présentation de Juergen Hoeller** Vous vous demandez qui est ce monsieur ? Et bien c'est tout simplement l'un des meilleurs développeurs de Spring, et surtout l'un des co-auteurs du framework Spring. C'est notre [Alan Cox](#) à nous pour les linuxiens si vous voulez, la barbe en moins. Après avoir fait un rappel des principes clés de Spring 2.5 il a présenté les nouveautés de Spring 3.0. A noter qu'il n'a pas parlé OSGI. Spring 2.5 a apporté le support des annotations afin de simplifier la configuration du framework. Je passe ce que tout le monde connaît, cette partie n'était pas intéressante. Les principales caractéristiques de Spring 3.0 : - basé sur Java 5 - proposition d'un expression language afin d'ajouter encore plus de choses sympathiques dans vos beans ou dans vos fichiers XML - support de REST avec annotations - Modèle de validation déclaratif basé sur Hibernate Validator et [la JSR-303](#) - Support pour les portlets 2.0 L'expression language (EL) va nous permettre de faire de l'injection de valeur... Grand silence dans la salle... Voyez ce code par exemple `@Value("#{bankAccount.balance}"); setMyBankAccountBalance(Double value) { ... }` Vous pouvez ainsi (et je me demande bien comment) récupérer la valeur d'un autre bean afin de l'injecter dans votre code. Bon je vous laisse sur ce bout de code et je vous invite à trouver un autre blog qui vous expliquera cette partie avec plus de détails. Juergen parle ensuite de l'ajout d'un mode "Conversation" au moteur Spring WebFlow. Fidèle lecteur du touilleur express, souviens-toi [dans cet article du 12 décembre 2006](#) de ce que j'avais raconté. Je trouve que ce concept est assez génial et bien que compliqué à expliquer, c'est un scope qui manque cruellement à Spring Web Flow. Juergen m'avouera ensuite plus tard que c'est fortement inspiré du framework Seam, car c'est un principe vraiment intéressant. Le support de REST permet dans Spring 3.0 d'utiliser des annotations, ainsi que l'expression language afin d'injecter des paramètres d'une URI dans votre bean. Bien que la soupe d'annotation ne soit pas ma tasse de thé, j'avoue que c'est intéressant. La validation du modèle est un principe intelligent afin de valider vos entités et vos beans. Le principe est d'ajouter des annotations sur les attributs afin de renforcer vos règles métiers. Je vous propose de lire l'article sur [la JSR-303](#), puisque Spring se reposera sur cette spécification. Enfin pour terminer, le support des Portlets 2.0 sera aussi disponible. La version Spring 3.0 M1 sera disponible fin novembre. La version 3.0 RC1 est prévue pour mars 2009 La version 3.0 final : avril 2009 en principe **Les clients** Didier Girard a eu l'idée d'inviter des utilisateurs du framework Spring, ce qui était très sympa. Les personnes dans la salle ont ainsi posé plusieurs questions, sans langue de bois. Étaient présents : Alexandre Navarro de la SGCIB, architecte sur des applications de trading, Benamar Kassini de Voyages-Snc.Technologie, François XXX d'HSBC, David le responsable d'Improve et enfin Guillaume Laforge, Groovy and Grails.



A la question "que pensez-vous de Spring dm Server et pensez-vous l'utiliser ?", la personne de la SGCIB répond qu'ils sont intéressés, car le besoin d'application en ligne 24h/24 dans la finance est un réel besoin. Le problème du versionning se pose réellement lorsqu'il n'est pas possible d'arrêter un système et qu'il faut cohabiter avec d'autres systèmes. Le site Voyages-sncf avec moins de 5 serveurs est une plateforme capable de gérer 5 à 7 millions de visiteurs par mois, avec des pics assez impressionnants par jour. Avec un très grand nombre de beans gérés par Spring, Benamar reconnaît que les temps de démarrage sont un peu longs. Mais en contre-partie, le coeur du système est capable de gérer un très grand volume de données. Spring apporte grâce à l'AOP entre autre et l'IoC, un très grand nombre d'avantages par rapport à une architecture J2EE à l'ancienne. **Conclusion** Enfin pour terminer, je ne regrette pas d'avoir pris le temps ce matin de venir assister aux présentations de SpringSource. L'organisation par SFEIR était parfaite, et en quelques heures nous avons couvert pas mal de sujets. Mais je reste sur ma faim... L'après midi j'ai donc été à la rencontre de Juergen et Peter ainsi que de Guillaume afin de les interviewer dans les locaux de SFEIR accompagné d'un autre blogueur : Olivier Croisier l'auteur du "[Coder's Breakfast](#)". J'ai enregistré avec mon MacBook Pro une interview de 48mn de Peter ainsi qu'une conversation très sympa avec Guillaume Laforge. Finalement la qualité du son pour l'interview de Guillaume étant assez correcte, après avoir nettoyé mon fichier je verrai si guillaume est d'accord pour poster ce podcast sur le Touilleur. En attendant je vous ferai un compte-rendu de ma rencontre avec Peter et avec Juergen dans de futurs billets. Merci à Julien Dubois et Didier Girard pour cet après-midi ! **Autres articles sur cette journée:** [Blog d'Octo](#) par Olivier Mallassi [Tuning Tomcat](#) par Tom's Quest [Spring dm Server le pari fou](#) par Tom's Quest Une photo de Juergen Hoeller, l'un des développeurs principal du framework Spring



## Nouveau : un podcast le Touilleur Express

[caption id="attachment\_501" align="alignnone" width="300" caption="N.Martignole et



G.Laforge"]  
 [/caption] C'est fait, le premier podcast du Touilleur Express est en ligne sur [sa page dédiée](#). Le premier épisode vous propose d'écouter Guillaume Laforge, le chef de projet du langage Groovy. SpringSource a annoncé le 11 novembre dernier le rachat de sa société, G2One.inc. A l'occasion de la journée "Les Rencontres Spring" nous avons été invité par SFEIR à rencontrer plusieurs personnes de SpringSource. J'ai enregistré une petite interview de guillaume avec Olivier Croisier, auteur du blog "[The Coder's Breakfast](#)". Voici des pointeurs vers des sujets abordés dans le podcast :

- [article sur InfoQ](#)
- [Le Blog de Guillaume](#)
- [The eXo Platform](#)
- [Benjamin Mestrallet, fondateur d'eXo, gagne le prix du jeune dirigeant de la Technologie](#)

A venir prochainement : interview avec Peter Cooper-Ellis. La qualité n'étant pas assez bonne, je vous proposerai une transcription de notre rencontre avec Peter prochainement. En attendant voici une photo prise chez SFEIR, avec donc de gauche à droite : Peter Cooper-Ellis, moi-même et enfin [Olivier Croisier](#). [caption id="attachment\_490" align="alignnone" width="300" caption="Interview Peter Cooper-Ellis avec Nicolas Martignole et Olivier Croisier"]



[/caption]

## BarCamp2 décalé au 16 décembre

Le [BarCamp2](#) est décalé au mardi 16 décembre, ce qui arrange pas mal de monde dont moi en premier, censé prendre le train tôt le lendemain matin. L'inscription s'effectue sur le blog après avoir pas mal bidouillé, j'ai réussi à mettre à jour la page sur le nouveau site, mis à jour il y a quelques jours.

## MDA, Scrum et l'Agilité dans tout cela

Le sujet est venu car le magazine papier "Programmez" a publié dans son [numéro 113](#) un article intitulé "Développeur, boostez votre productivité". [Emmanuel Chenu](#) sur son blog rapporte que l'auteur explique que la modélisation MDA est un processus améliorant la productivité et l'agilité du développeur. [J'avais déjà dit](#) que pour moi la modélisation n'est pas une méthode Agile, ou qui ajoute une quelconque agilité au développement. Emmanuel a écrit 2 billets que je vous recommande car il écrit très bien. En résumé, [son premier billet](#) et [son deuxième billet](#) peuvent se résumer comme suit : Après avoir rappelé que la modélisation MDA et les PIM ne sont qu'un autre moyen de représenter de l'information, Emmanuel pointe le fait qu'il faut des outils

supplémentaires pour effectuer cette modélisation. La question qu'il pose dans l'extrait ci-dessous est intéressante : Extrait de [son billet 2](#)

[...] Pour sonder une éventuelle compatibilité, on peut commencer par se demander si le MDE est compatible des valeurs du Manifeste Agile. Il me semble que le pilotage par les modèles met fortement l'accent sur les outils. L'éditeur UML, les transformations de modèles et les générateurs de code sont autant d'outils sur le chemin critique du développement. Ces outils particuliers viennent s'ajouter aux incontournables outils tels que le gestionnaire de versions, le compilateur, le framework de test, l'IDE, etc.

Je partage son idée lorsqu'il s'oppose à l'argument du journaliste. Celui-ci propose de dire que la modélisation simplifie la maintenance puisque le développeur n'a pas accès au code.

(Extrait de la page 69 de Programmez numero 113) Comment faire pour qu'un programmeur maintienne sans effort conséquent le code d'un autre? Comment faire pour qu'un programmeur retouchant celui d'un autre n'introduise pas, par simple incompréhension, des bogues? Réponse : **le transformer en créateur de modèles et générer le code à partir de ses modèles!"**

C'est ce que les éditeurs ont essayé de nous vendre il y a 7 ans. Prenez mon gros modelleur professionnel et vos informaticiens de base ne toucheront plus au code, seul l'Architecte pourra modéliser le logiciel. Nous sommes tous d'accord pour dire qu'aujourd'hui cette image a du plomb dans l'aile. Et pourtant je me souviens très bien avoir pensé un moment que nos diagrammes UML deviendront un vrai logiciel. En fait je vais vous avouer un secret : je fais de la modélisation. Pas comme vous l'entendez, au sens avec AndroMDA ou compagnie... Non pas du tout. Ma modélisation à moi c'est une feuille et un crayon. Je pense que les personnes qui travaillent avec moi depuis septembre ont remarqué que j'ai pour habitude de dessiner rapidement le système sur lequel nous discutons. Et croyez moi, c'est suffisant. Cela permet de mettre à plat une idée, d'expliquer un flux de données, d'affiner un point technique délicat, ni plus, ni moins. Pour revenir à l'article d'Emmanuel, après avoir passé en revue quelques associations comme Scrum et Modélisation ainsi que XP et modélisation, sa conclusion est la suivante :

En fait, je pense qu'un développement piloté par les modèles gagne à être conduit dans une démarche agile, comme Scrum. Ainsi, le projet gagnera en transparence et du logiciel opérationnel sera régulièrement disponible.

Il faut savoir que les pro-MDA-c'est-Agile sont allés jusqu'à penser (en 2004) que le modèle dynamique du PIM serait suffisant pour répondre à l'exigence d'agilité. J'ai lu [des vieux articles](#) dans lesquels les auteurs se gaussent d'UML2 et de la modélisation dynamique. Honnêtement, levez la main ceux qui connaissent la représentation UML2 des diagrammes dynamiques ? Ah y'a une personne là-bas au fond à gauche qui lève la main... Attendez vous vous appuyez sur le mur ? Désolé... **Quel est l'intérêt de la modélisation ?** C'est un moyen d'apporter de la réflexion, de faire travailler des gens qui ne sont pas développeurs sur la valeur d'un logiciel : le code métier. Je ne suis pas contre la modélisation, je trouve même cela élégant et très pratique lorsqu'il s'agit de modéliser des processus métiers complexes. En lisant en ce moment "[Enterprise Integration Pattern](#)" de Gregor Hohpe, je peux vous assurer que j'adore l'utilisation de la modélisation pour ce qu'elle apporte en terme de clarté et de concision. Je suis donc pro-modélisation mais je suis opposé à cette idée qui est de faire croire aux décideurs que la maintenance sera plus légère, que les coûts seront réduits, et que les développeurs arrêteront de faire n'importe quoi. Au contraire, malheureusement il faut trouver tout d'abord des personnes avec de bonnes compétences en UML, en utilisation des outils de modélisation, avec aussi un oeil d'architecte et d'intégrateur. Idéalement il faut un bon communicant car ce modelleur doit représenter la complexité métier d'un moteur de calcul d'assurance ou de gestion du risque en finance... Ne sommes-nous pas là entrain de créer un Dieu qui viendra ensuite vous demander un salaire exorbitant tout en se mettant une équipe de développeur sur le dos ? Bref attention à ce monsieur en face qui vous vend MDA. Soit c'est un génie et signez-lui un chèque, soit c'est le père Noël mais vous m'aviez dit que vous aviez arrêté d'y croire... Passons au sujet 2, je n'ai aucuns soucis à mélanger le tout dans un seul article. **La guerre anti-scrum est ouverte** Un autre article qui nous ouvre les yeux sur Scrum : "[The Decline and Fall of Agile](#)" de James Shore. Tout d'abord je vais vous résumer ici quelques idées clés, je vous conseille de le lire par vous même dans le but de vous faire votre propre opinion. *L'auteur explique tout d'abord qu'il constate un changement dans son métier de consultant sur l'Agilité. Là où il était appelé pour mettre en place de l'Agilité, il est aujourd'hui appelé pour venir sortir du trou des équipes noyées sous de la pseudo Agilité. Scrum est le responsable numéro un car c'est l'une des méthodes les plus populaires introduites ces dernières années. Des flots de Scrum Certified Master ayant suivi deux jours de formation sont lachés dans la nature. La difficulté de Scrum est que contrairement à ce que les gens pensent, c'est loin d'être facile. Scrum étant un processus qui fonctionne en petits cycles, il n'y a pas de pratique de développement. Vous pouvez coller du XP par dessus si vous le souhaitez. Il n'y a donc pas de spécifications ou de recherche (NDLR : je ne partage pas cet avis). Les équipes se creusent alors un puit sans fin, en n'ayant pas assez capitalisé sur la technique. Les projets sont alors certes en ligne, tel que Scrum le vend. Mais les coûts de maintenance ? Les effort de documentation ? Le changement coûte alors cher. [...] des équipes se disent Agile mais finalement elles ne font rien de plus que de la planification chaque semaine. Elles n'ont cependant pas changé leurs méthodes, il n'y a pas d'espace de travail en commun, le client n'est pas présent.... [...] Au final il y a des équipes qui commencent des éléments du sprint backlog, mais rien n'est terminé, le tableau se remplit. Il y a aussi les chefs qui changent les éléments du sprint en cours... car l'urgence c'est l'urgence... [...] Des équipes essayent d'adopter qu'une partie du processus de Scrum, car c'est la nature humaine. Il est plus facile d'accepter le côté sympathique et de laisser la partie rébarbative, en se couvrant au nom de l'Agilité. Bon et bien cela fait pas mal non ? C'est plutôt à charge non ? Et pourtant... [Rob Bowley](#) prédit même que les méthodes à la mode en cette fin 2008, Lean Development, Kanban, Minimal Marketable Feature, prendront le même chemin que Scrum. Attendons un peu et nous devrions*

voir des articles sur Lean Development Software qui se propose de remplacer Scrum... (note : je ne dis rien car j'avoue que l'idée m'a traversé l'esprit). Il se demande si tout simplement si **le problème ce n'est pas nous**

Many of the reasons people aren't being as succesful as they'd like with Scrum are exactly the same reasons they won't be any more successful with any other methodology. People tend to focus on tools because it's a lot easier than trying to tackle the often very difficult, challenging and more fundamental problems they grew from. Real change is hard and takes time, a very long time in some cases.

ce qui donne en français

Beaucoup de raisons pour lesquels les gens n'ont pas autant de succès avec Scrum qu'ils l'espèrent sont les mêmes pour lesquels ils n'auront pas plus de succès avec d'autres méthodologies. Les gens tendent à se focaliser sur les outils car c'est plus facile que d'adresser les problèmes plus profond, complexes et fondamentaux dont ils ont été entourés. Le vrai changement est difficile et prend du temps, vraiment beaucoup de temps dans certains cas.

**Scrum ? What else ?** Regardons de manière objective ce qu'est Scrum : une méthode simple et bien marché qui permet à des personnes intelligentes de bien gagner leur vie, je parle des personnes qui vous font passer la certification. Honnêtement je remercie Scrum de m'avoir ouvert les yeux sur la lourdeur du processus classique de développement, de m'avoir fait comprendre qu'il est important de communiquer, de mettre à plat une liste des demandes du client classées par priorité, de privilégier l'humain par rapport à l'outil, d'être honnête et transparent. J'ai un peu plus de recul et je me rends compte que cette méthode anglo-saxonne n'est pas la méthode miracle. C'est un très bon socle de départ, je recommande aux personnes n'ayant aucuns idées de l'Agilité de s'y intéresser. Je vous demande simplement de prendre le temps de bien réfléchir. Etes-vous prêt à engager toute votre équipe, votre manager, vos dirigeants dans une refonte de votre organisation ? de votre produit ? Avez-vous tous les moyens ? Avez-vous pensé à ce que sera votre projet après Scrum ? Au final il est important avant d'appliquer une méthode de bien comprendre l'environnement, l'écosystème d'un projet, d'un produit. Ensuite il ne faut pas s'interdire d'adapter le processus à son équipe et à son produit. Cependant je pense qu'il faut faire attention dans ce cas à ne pas tomber dans de la fausse Agilité, de casser l'organisation d'un projet ou de se mettre les managers à dos. Rien de pire au final, tout repose sur vous et sur ce que vous faites. Il n'y a pas de méthodes, il n'y a que vous pour faire avancer votre logiciel, votre produit.

## Le Touilleur Express a 5 ans

Hasard du calendrier, je me suis rendu compte que cette semaine cela fera 5 ans que le Touilleur existe... Du coup j'ai repris un peu mes vieux articles, afin de regarder l'évolution des messages, du ton, des grandes théories en carton pâte que j'ai écrit, des coups de gueule, des longs tutoriaux.... Cela m'a donné envie de vous parler un peu des articles les plus marquants, qui ont fait le plus de hit, qui ont été le plus lu, histoire de partager cela avec vous. Je vous préviens : c'est de l'auto-promotion à mort, je ne vous engage pas à lire si vous venez de découvrir le blog. Pas du tout envie de passer pour un peu mégalo, mais envie de parler de ces années passées à raconter et parler sur ce blog. **Les débuts** [Le premier article](#) pose les bases : ici on ne parlera que français. Le 23 novembre 2003 le premier message sur l'ancienne plateforme qui est encore en ligne ([JRoller](#)) annonce fièrement la couleur : welcome in a french world ! Les premiers billets sont courts, je cherche un peu quoi raconter. A l'époque je termine ma première année chez Reuters, dans une équipe américaine. Je suis encore très marqué Linux. 2 jours après mon arrivée chez Reuters je ne trouve rien de mieux que de casser ma partition windows afin d'installer Linux. Je me bats ensuite avec Wine [pour faire tourner Lotus Notes](#) et je dois passionner la terre avec la configuration de [mon écran plat](#)... Bref si je regarde les sujets de l'année 2003 : java et linux. A l'époque je propose des bouts de code, en parlant de tout et de rien. Je suis fier de vieilles bidouilles [comme celle d'afficher le nom du jar qui a chargé une class](#) ou [celle-ci aussi](#) par exemple, des bouts de code qui servent encore aujourd'hui. Il y a aussi une astuce pour utiliser les couleurs ANSI des xterm unix [avec Ant](#) ou Log4j, avis aux amateurs geeks. En 2004 petit clin d'oeil à [Benoit Moussaud](#) (Xebia) et son blog, l'un des rares à l'époque. Très actuel : on parle de James Strachan qui [en 2004](#) se fait récompenser pour avoir inventé Groovy. **Phase II** Fin 2004 je me souviens très bien avoir eu l'envie d'arrêter. Avec 50 lecteurs par mois, et encore les jours où le vent venait par derrière, à l'époque c'était vraiment pour le fun. Cela repart très fort début 2005 avec [la certification JBoss](#), des billets qui m'avaient demandé pas mal de travail à l'époque. A partir de là j'ai vraiment trouvé le ton et le contenu. Je tente des articles un peu geek comme la question du pattern [lazy initialization](#). Mi-2005 c'est [la fin de l'aventure pour le premier projet](#) chez Reuters, sauf si je pars à Bangkok avec femme et bébé à naître dans quelques mois... Finalement Robin est né en octobre 2005. Je fais un peu de JSF à l'époque, en travaillant [sur une console d'admin](#). Bon souvenir, mais un peu trop de XML pour faire marcher le tout. Je change d'équipe ensuite en septembre. **Etape 3 : là où tu penses devenir un chef de quelque chose** Hop me voilà jeune chef de projet, avec une équipe qui a codé un framework J2EE au fusil à pompe. Très belle architecture mais un gros bazar et aucunes méthodes. Entre [les conventions](#), la mise en place propre d'Ant, de SVN, de la documentation et de l'intégration continue, je m'éclate. Au passage je poste [mon premier billet en anglais](#) sur log4j. Ce bout de code permet dans un xterm sous Linux d'utiliser les couleurs ANSI pour afficher les messages de Log4j. Et il marche encore très bien aujourd'hui. Je crois en avoir refait une version plus propre, il faut fouiller sur le Touilleur. Linux Mandrake me lasse, je passe à [Ubuntu](#). Quelques mois plus tard je me souviens avoir laissé tomber linux. Les raisons ? Microsoft Project, Lotus Notes, des outils propriétaires... Bref j'ai perdu la guerre cette année là. Le premier billet qui parle de gestion de projet apparaît [fin 2005](#). En passant en revue les sujets des articles je vois que cela parle d'Ajax. C'est mon nouveau job : chef de projet d'un framework propriétaire Ajax destiné à réaliser l'interface web de produits financiers. Des tartines et des tartines d'ajax, [des](#)

[articles longs](#) sur XUL et Flash, bien avant la sortie de Flex et Silverlight. Je commence ensuite mes premières sorties dans la nature, en allant [chez Valtech regarder les EJB 3.0](#), à [un salon Linux à Paris](#) ou à [une présentation JBoss](#). Et tant qu'à faire je commence aussi à bidouiller avec [JBoss Seam](#) en 2005... Wow j'ai des cheveux gris ! On en parlera le 2 décembre prochain au Paris JUG sur JBoss. C'est [l'amour fou avec Seam](#). Dans les vieux billets marrants, je retrouve [Jean-Laurent de Morlhon](#), Valtech Boys à l'époque, aujourd'hui directeur technique chez Vidal. Avec le recul en 2006 je trouve que je n'ai pas tellement écrit d'articles. Peut-être trente articles, une majorité sur Seam et sur Ajax, plus rien sur Linux. **Étape 4** En 2007 dans les billets qui m'ont marqués, je me souviens du salon [SUN Tech Days](#). Rencontre avec Alexis Moussine-Pouchkine et Romain Guy. [Essais ensuite avec Flex 2](#), je m'interrogeais à l'époque sur la complexité de maintenir une application Ajax alors que Reuters fait essentiellement des applications. Le lecteur reprend au passage 3 articles sur Seam ([ici](#), [ici](#) et [là sur groovy](#)) Le top du top c'est mon petit [pétage de plomb](#) pour convertir l'assemblée à Seam. Tranche de vie de Reuters : [départ en urgence pour New-York](#) pour un client... Bon souvenir. **Étape je vais dépenser de l'argent** Avant de continuer il faut que je vous raconte quelque chose : fin 2007 mon fils renverse de l'eau dans mon unité centrale à la maison. Psshiiiiit, plus de PC et de carte mère. C'est dommage. Bref me voilà sans PC à la maison. L'idée d'acheter un mac apparaît [un jour](#) sur le blog. Je me lance aussi [dans la divination et la vision](#)... A croire que je tourne au pétrole parfois. Ma love story avec Apple a fait venir un nouveau public sur le blog. Je bassine un peu tout le monde avec mes articles [sur Steve Jobs](#) au boulot. Nous craquons avec ma femme sur un iMac en [décembre](#), le 3ème enfant de la famille. Et [je mets Windows à la porte](#) après 10 ans de vie commune... **Étape 12434 : l'année 2008** C'est une très grosse année. Je commence par installer Google AdSense dans le secret espoir d'être très riche grâce au Touilleur. Et bien j'ai gagné 62 \$ en 10 mois, sachant que je paye aujourd'hui 16 EUR par mois pour l'hébergement, le Touilleur est une aventure philanthropique. Avec le changement de serveur le 23 janvier 2008, je quitte JRoller pour prendre un hébergement dédié. J'installe aussi Google AdSense afin de me faire une idée du nombre de visiteurs. Je vais vous donner les chiffres mais attendez un peu... **La bonne blague** : En achetant le nom de domaine du touilleur express, un dimanche matin, je fais une grosse boulette qui me fait bien rire : j'ai d'abord acheté le domaine touilleur-express.fr avec 3 fois la lettre L. D'ailleurs je vous laisse regarder par vous même : <http://touilleur-express.fr> redirige vers le site actuel. Et bien sachez que 76 visiteurs se sont plantés cette année et qu'ils mettent 3 fois L dans le mot Touilleur.... **Les articles de 2008** Je parle du Paris JUG [dès février 2008](#), ce qui me paraît loin. A ce jour en 2008 j'ai publié 116 billets, avec une moyenne de 10,54 articles par mois. A part le mois d'août où je n'ai publié que 3 articles, le reste du temps il y avait toujours quelque chose à raconter. Que ce soit sur Scrum, sur ma transition de Reuters, sur le Paris JUG, sur les interviews réalisés avec différentes personnes... Pas mal de sujets différents. **Les chiffres, les chiffres, les chiffres** Parlons audience : quels sont les articles les plus lus ? d'où viennent les visiteurs ? Quels sont les sites qui génèrent le plus de trafic vers le Touilleur ? Qu'aimez-vous lire ? **Du 18 octobre 2008 au 19 novembre 2008: 5388 visites 9673 pages vues** 20% de trafic direct, 30% venant de sites référents comme Xebia ou d'autres blogs, 50% venant de Google. Les gens passent en moyenne 1mn20 sur le site. J'ai mis en ligne [le rapport Analytics](#) si vous voulez y jeter un œil. **Depuis début 2008** Bon je joue le faux modeste depuis tout à l'heure mais je suis vraiment content d'avoir autant de monde. Autant donc y aller franchement et ensuite je ne vous ennuie plus avec cela : **25 241 visites depuis le 1er janvier 2008, 46 788 pages vues** Les moteurs de recherche ne représentent que 45% de ce trafic, le reste c'est des personnes qui ont tout bookmark sur le Touilleur, soit cliqué sur un lien quelque part comme sur le site de [Xebia](#) par exemple. [Fichier complet Google Analytics avec chiffres de 2008](#) **Les voisins** Intéressant : quelles sont les sources de trafic qui apportent des visiteurs sur le blog ? En premier, sans surprise, c'est Google. 44% des gens qui viennent ici (et qui repartent rapidement) sont le résultat de recherches dans Google. Il y a 20% de personnes qui accèdent directement au Touilleur, sous entendu via un lien favori dans leur navigateur, ce qui fait 5075 visiteurs depuis janvier. Ensuite je souhaite dire un gros merci à mes voisins : [le blog de Xebia](#) a fait venir 972 visites. Soit 3,72% du trafic. Derrière on trouve aussi le blog de Didier Girard, [Application Server](#). Je suppose que Didier est le premier blog français sur Java, à ma connaissance c'est le plus connu. Si l'on creuse un peu j'ai vu qu'en fait 40% des visites sont des personnes qui reviennent sur le blog, il y a donc 60% de gens qui ne font que passer, souvent via un résultat de recherche sur Google. Du côté des pays, la France représente 75% des visites, ensuite la Suisse avec 5,53%, les USA avec 2,84%, la Belgique avec 2,34%. Après le Canada et l'Angleterre, c'est le Maroc et la Tunisie, avec environ 540 visites. **Les plus lus, les moins lus en 2008** L'article le plus en 2008 est cet article sur [le SerialVersionUID](#). Nous en avons parlé ce matin avec Guillaume Drot, un collègue d'Ippon Technologies. Ce qui m'amuse c'est que si l'on regarde le ton de l'article, c'est très sérieux. Ensuite voici les 3 articles les plus lus dans l'ordre

- [JSR-296 sur Swing](#)
- [Flex, blazeds, spring, hibernate...](#) article en anglais qui a attiré beaucoup de monde
- [Ext-GWT ou GWT-Ext](#) très d'actualité

L'article que j'ai eu le plus de mal à écrire : [mon départ de Reuters](#). Pour la petite histoire le projet dont je parle dans l'article est mort, je ne travaille plus dessus. L'article qui m'a le plus fait rire à écrire : [le lave-linge Conclusion](#) Voilà si vous avez lu jusqu'ici je vous tire mon chapeau. C'est très moi-je cet article. Mais bon, on me demande souvent s'il y a des gens qui lisent le touilleur. Ce à quoi je réponds non, et que je gonfle les chiffres afin d'être racheté par une Grosse Compagnie. Mon but dans la vie : être payé affreusement cher pour écrire toute la journée sur le blog. Je vous rassure : bien entendu que non. C'est un passe temps. J'ai des tonnes d'articles en tête, mais pas assez de temps pour tout mettre à place. J'ai la chance d'avoir une adorable épouse compréhensive et 2 enfants qui me lèvent à 07h30 le dimanche matin. Très pratique pour écrire. Le secret pour bloguer c'est de faire des enfants... Hem... j'ai encore dit n'importe quoi là. Allez, continuez à lire, à rigoler, à vous apitoyer, à sourire, à vous énerver, à me relire et me corriger (merci à Flavien sur ce coup) à me donner des idées (Florent de Celerio) à faire comme si on ne lisait pas le blog (les potes de reuters) à vous dire qu'il faut s'y mettre aussi (Dimitri) à écrire des blogs vraiment bien (Erwan de [Xebia](#), [Tom's Quest](#), [les gens d'OCTO Technology](#) et dernièrement, [Olivier](#)). Tant que j'aurai l'envie le temps et des choses à raconter, vous me verrez trainer dans vos lecteurs RSS, et même sur votre ipod. Là je dois vous parler de mon macbook pro qui a rendu l'âme, de mon retour

d'expérience sur l'iphone. J'ai envie de bloguer sur "Ecrire des logs pour les nuls" car j'en ai marre de voir des logs qui ne servent à rien. J'aimerais bien aussi parler de [cet excellent article](#) qui compare l'Architecte et Joe le plombier de Sami. C'est un billet vraiment intéressant... Bref il y a du boulot et des sujets à traiter. Merci de lire le Touilleur Express A bientôt

## 2 Soirées Paris Java User Group

Le Paris JUG organise 2 soirées début décembre :

- o Mardi 2 décembre : [soirée JBoss](#)
- o Jeudi 4 décembre : soirée "[Ma petite entreprise ne connaît pas la crise](#)" (devenez indépendant, créez votre entreprise)

La soirée JBoss se découpera en 2 parties : présentation de JBoss Application Server par Sacha Labourey, puis ensuite présentation de JBoss Seam par Malik Saheb. L'inscription est indispensable, la soirée se déroule dans les locaux de l'ISEP à Paris à partir de 19h15, le mardi 2 décembre. La soirée Paris JUG2B se déroulera chez [Port Parallèle](#) le jeudi suivant. L'idée : vous souhaitez vous lancer en tant qu'indépendant, vous souhaitez créer votre entreprise. Comment choisir la meilleure structure ? Qui peut nous accompagner durant la phase de création ? 2 Soirées sympathiques organisées par le Paris Java User Group, le tout gratuitement grâce aux sponsors :



## MacBook Pro : le coup de la panne et support du clavier international

**La petite panne** La semaine dernière, petite frayeur : mon macbook pro ne démarre plus. 3 bips répétitifs, et impossible de démarrer. **A la recherche de la panne** J'ai essayé d'activer le mode diagnostic (garder la touch D enfoncée, allumer le mac, voir ce que dit la machine) mais cela ne fonctionnait pas. Je récupère le CD d'installation. Il est possible de l'insérer. Je redémarre en gardant la touche C enfoncée afin de forcer un démarrage... Pas plus de succès. Ok ce n'est pas un souci de système d'exploitation. **Recherche sur le site du support d'Apple** Un tour sur les forums me donne [quelques informations intéressantes](#). Une procédure pour vider la PRAM du MacBookPro. Une autre pour remettre à zéro le gestionnaire d'alimentation... après avoir tenté ces manipulations, rien de neuf, toujours planté. **Identification des bips** Je trouve quelques vieux articles sur le site du Support mais qui ne me disent pas grand chose, à part qu'une de mes barrettes de mémoire serait en cause. **Appel au support Apple et réparation** Bref après avoir contacté le support client, j'ai amené mon MacBook Pro à Paris [chez Alis](#), métro Breguet-Sabin. Accueil sympa, service rapide et efficace. J'ai récupéré ma machine le lendemain. Barrette de mémoire défectueuse. Rien à payer grâce à la couverture AppleCare que je trouve un peu cher. **Clavier international** Voici quelques petites astuces si vous avez un clavier Qwerty comme moi. Sous Windows le mode US-International permet de taper des accents avec l'astuce suivante : je tappe "virgule" et "e" et cela donne un e accent aigu. Sur Mac OS X 10.5, le layout du clavier en mode "Américain étendu" permet aussi de saisir des accents. Pour taper un accent aigu il faut ainsi enchaîner la séquence suivante : Option+e suivit de e. Cela donne un accent aigu. Pour faire un "à" il suffit d'appuyer sur Option+ et la lettre a ensuite. Honnêtement c'est un peu fatiguant et pas vraiment pratique. Pour éviter cela il est possible d'installer sur Mac un mode "US-International" similaire à celui de Windows. : - téléchargez le layout sur [le site de Rainer Brockerhoff](#) - une fois l'image montée, déplacez sur le bureau le fichier "US International" - Ouvrez un terminal (Spotlight, tapez Terminal) - dans le terminal, effectuer la copie en tapant cette ligne de commande : `macbook-pro:~ nicolas$ cd /Library/Keyboard Layouts macbook-pro:~ nicolas$ cp ~/Desktop/U.S. International.bundle .` - quittez la session afin d'activer la nouvelle modification - le nouveau layout sera disponible dans la liste "International" dans le menu de configuration après le clavier Turc en fin de liste. Une fois ce clavier installé vous pouvez alors taper un "é" avec la séquence suivante : ' puis e

## Language Oriented Programming : ton boulot de dans 5 ans

Voici un concept intéressant à partager avec vous. Je vais vous parler aujourd'hui de LOP, Language Oriented Programming. Sergey Dmitriev est le cofondateur et CEO de JetBrains Inc., l'éditeur de l'excellent éditeur Java : [IDEA IntelliJ](#). Il propose une nouvelle manière d'envisager notre travail de développeurs pour les années à venir. En quelques sortes, pour lui le futur ne sera plus au développement orienté objet, mais bien au développement orienté par le langage. Jusqu'ici vous ne comprenez rien, c'est normal. Bienvenue à bord. Tout d'abord nous allons parler des raisons puis des moyens de la programmation LOP, à l'issue de l'article vous aurez une idée des concepts, et des liens vers des articles complémentaires. Cet article est librement adapté [de l'article original en anglais](#) de Sergey Dmitriev.

## L'origine de LOP : nous ne sommes pas libres

Sergey Dmitriev part de ce constat : aujourd'hui lorsque nous programmons, nous ne sommes pas libres. Nous devons nous adapter au langage Java. Malgré sa puissance, il n'est pas possible pour vous, en tant que développeur de sortir d'une zone délimitée par le langage, ses concepts et sa syntaxe. Ne serait-il pas intéressant de réduire la dépendance de la même manière que Java nous a libéré de la contrainte de la plateforme ? Il est naturel de coder sur Windows et de déployer sur Solaris. Pourquoi ne pas poursuivre vers cette liberté ? Les langages classiques (Java, C#, C++) se basent sur la modélisation objet, ce qui devient trop vague lorsqu'il faut modéliser des concepts métiers compliqués. L'alternative s'appelle DSL, mais ce concept donne un langage qui est trop fortement typé au problème. Exemple de DSL que vous connaissez tous : le SQL. Ce n'est pas un langage de programmation mais d'expression dynamique. Aujourd'hui nous vivons très bien avec ces contraintes. Pour cette raison on voit surgir des langages dynamiques ou plus réactif comme Groovy. Mais il n'y a pas de moyens pour un développeur d'exprimer son besoin de A à Z. Sergey pour obtenir cette indépendance que je vais vous expliquer, propose un système qui permet de créer, modifier et utiliser le langage et son environnement. Pour que cela fonctionne, le système est simple, et son idée est que cela soit vraiment la prochaine révolution du monde du développement informatique... Tout un programme. **Le passage vers le code texte** Il faut faire le constat suivant pour comprendre la suite : aujourd'hui entre l'idée pour résoudre un problème que vous avez en tête et le langage Java, il y a un pont que l'on traverse. Sur ce pont, commençons par utiliser la syntaxe du langage pour exprimer notre idée. Ensuite nous utilisons la programmation orientée objet avec des Classes, des Interfaces. Enfin en Java nous sommes particulièrement bien outillés : nous utilisons Spring, Hibernate, Struts... Entre l'idée que vous avez en tête et le programme final en Java, il y a donc bien un pont. Un bon développeur lisant votre code va se forger une représentation de ce que fait votre code. Mais en aucuns cas il ne pourra voir du premier coup ce que fait exactement votre programme. Merci la modélisation, nous cherchons aujourd'hui des palliatifs afin de représenter sous forme de diagramme ce que fait notre code. Nous ne sommes pas libres car nous devons adapter à la conception objet un problème qui n'a peut-être pas besoin de classes ou d'interface. Nous ne sommes pas libres car nous devons nous tenir à la syntaxe initiale. Enfin nous avons aujourd'hui de plus en plus de soucis avec le fait qu'il soit devenu indispensable d'être un expert en Spring, Hibernate ou autre... **L'éditeur et le format texte** Depuis des années sans même y faire attention, le résultat de notre travail n'est qu'un fichier texte portant une extension java. Certes nous nous amusons aussi avec le XML ou autre... mais au final nous laissons à nos suivants du code qui se lit avec un éditeur de texte. Grâce aux IDE puissants, ce texte reprend son sens, l'éditeur reconnaît la syntaxe et lui donne vie. Dans ce cas, pourquoi continuer à travailler sur des fichiers plats ? Prenez Microsoft Excel : l'outil le plus utilisé par le métier de la Finance par les analystes. Ceux-ci ne sont pas des programmeurs. Pourtant ils expriment des concepts financiers et le tout avec Excel... Et jusqu'à preuve du contraire, un fichier XLS n'est pas un fichier texte. Supprimons cette contrainte d'avoir un fichier texte. **Comment perdre du temps** Nous perdons beaucoup de temps. Je vais faire plaisir à mon meilleur ami qui est un fan de C. Il n'a jamais traversé le pont pour venir du côté de Java. Je me place de son point de vue : nous sommes une communauté très active, qui au fil des années s'est inventée des outils, des frameworks, des logiciels afin de résoudre toutes sortes de problème. Et il est vrai que c'est une énorme force que d'avoir une communauté libre aussi active. Mais d'un autre côté, est-ce que cela vous éclate toujours de tapoter du XML pour configurer vos beans ? de bidouiller un fichier hbm.xml afin qu'Hibernate s'exécute correctement sur votre classe ? de bidouiller avec Wicket là où parfois une JSP fait aussi bien et plus vite ? Est-ce que finalement nous nous sommes pas sur-outillés pour le plaisir ? **Est-ce que le développeur Java n'est pas devenu un Jacky roi du tuning ?** Je ne me moquerai plus des Cléo Super Sport avec "The Sport Touch" sur la vitre arrière. Après tout je dors bien avec un tee-shirt "SpringSource" et j'avais un sweet SUN vendredi dernier... Nous nous éclatons à perdre notre temps avec nos outils (Hibernate, Spring, Struts2) là où nous devrions bosser sur le code métier de nos applications. J'adore Spring et consors, mais je pense qu'il est temps de passer la seconde. Continuons à examiner la programmation orientée langage. **A quoi ressemble un programme LOP alors ?** Sergey explique clairement que dans son concept, un programme LOP n'est PAS une suite d'instruction. Déroutant non ? Pour lui un programme est la solution à un problème. Vous devez implémenter une machine à café ? Imaginez une syntaxe qui vous permet de construire votre machine, d'assembler les parties, de définir les règles, sans trop utiliser d'instruction. La programmation orientée langage devient simple : vous écrivez avec un langage que vous inventez ce que doit faire votre programme. Expression du besoin, rien de technique. C'est un story-board de cinéma qui raconte une histoire. Mais vous ne serez pas obligé de parler éclairage, prise de son, script, mise en scène, décor, habillage, dialogue... comme vous le feriez en Java "classique". **Un langage LOP** Un langage LOP n'est donc pas forcément exprimé avec du texte. Il y a bien une structure pour définir ses concepts et comment ceux-ci s'arrangent. Il dispose d'un éditeur pour exprimer comment il doit être rendu et arrangé. Sa sémantique, comme la grammaire française, exprime les règles et les exceptions.

## II. Comment maintenant modéliser son langage ?

Dans la première partie j'ai expliqué les idées de Sergey. Il explique en résumé qu'entre l'idée que vous avez en tête et son expression au final en Java, il y a un gouffre. Son point de vue est donc qu'il faut une alternative. Nous allons voir maintenant comment. **Le langage de structuration** Aussi compliqué que cela puisse paraître, LOP travaille avec plusieurs langages basés sur la syntaxe Java pour vous laisser créer votre langage. Dans un premier temps, le langage de Structuration permet d'exprimer la représentation de votre langage. Ainsi vous allez exprimer ce qu'est une méthode, une classe, pour peu que votre langage en ait besoin. Souvenez-vous : vous êtes entrain de créer votre langage. Vous pouvez ainsi inventer une "MethodeSimple" et une "MethodeAvecArguments" si cela vous semble utile. Oui vous pouvez inventer une "Closure" ou une "Annotation". Tout ceci ne se fait pas magiquement mais via un logiciel afin de vous aider : [MPS](#) (Meta Programming Model) est un logiciel de JetBrains encore à l'état de prototype. Il vous guide et vous permet donc d'écrire chacune des parties nécessaires pour modéliser votre langage.

Je sais, je n'ai pas encore décrit toutes ses parties. On y reviendra. La Structuration est donc la phase où vous sélectionnez dans votre grosse boîte de Lego les types de briques. **Définition de l'Éditeur** Une fois votre Structure en place, il est temps de définir comment nous allons éditer et donc écrire du Machin avec un Éditeur. Si vous vous souvenez du début : le texte n'est pas indispensable. De même que la modélisation utilise UML, nous allons utiliser un concept simple afin de définir et programmer... l'éditeur de votre langage. Il faut savoir qu'au final un programme écrit avec votre langage sera stocké sous la forme d'un graphe d'objets. Donc que ce soit un programme texte ou un diagramme avec des boîtes, cela n'a pas d'importance. On conserve bien la notion de graphe d'objet, les objets étant vos briques Lego, l'Éditeur c'est tout simplement l'outil qui va nous aider à assembler ces briques. Concrètement vous allez définir un ÉditeurDeMethode, un ÉditeurDeClasse ou par exemple un ÉditeurDeClosure. Visuellement dans le logiciel MPS cela ressemble à du code. Des zones éditables vous laissent définir l'enchaînement de l'édition, afin que lorsque l'utilisateur final puisse s'y retrouver. Vous venez donc d'écrire maintenant des choses pour éditer la Structure de votre langage. Sergey voit l'édition d'un langage comme un ensemble de cellules. Il explique qu'un éditeur est structuré afin de suivre une suite d'opération. La première cellule par exemple contiendra la portée (public, private, protected). La deuxième cellule pourra contenir un modificateur comme static. La troisième cellule sera le nom de la méthode... etc. Imaginez vous entraîné de préparer un formulaire de saisie pour qu'un développeur code avec votre langage. Cela permettra à MPS d'aider le développeur avec de la coloration syntaxique. J'ai du mal à expliquer ce concept. Mais c'est en quelques sortes le moyen utilisé aujourd'hui pour justement valider votre code Java dans Eclipse : il utilise un moteur d'édition afin d'analyser et de vous signaler au moment de la saisie les boulettes dans le code. Là il s'agit donc d'écrire cet éditeur. **Le langage de Transformation** Nous avons la structure, l'éditeur. Voici le Transformateur ou générateur. Concept similaire à MDA, il s'agit maintenant d'exprimer avec un langage les systèmes de transformation de votre langage. Comment allons nous passer de ce que l'utilisateur a édité précédemment à du code exécutable ? Il existe 2 systèmes : l'interprétation et la compilation. Soit votre transformateur va interpréter le code, soit nous allons vraiment le compiler en un langage classique comme Java, une fois pour toute. La première démarche fait penser à Groovy, là où la seconde me fait penser à la modélisation MDA. La compilation va s'exprimer avec le langage de Transformation. Je m'explique : votre objectif est d'utiliser l'une des 3 techniques de LOP afin d'expliquer comment transformer la Structure vers du code standard. Vous pouvez tout d'abord travailler par approche itérative : votre langage de Transformation sera donc du style : "Pour toutes les Briques Rouges LEGO tu me donnes tel bout de code". Cela consiste à introspecter le programme. Il y a ensuite l'approche par modèle et macro. Nous pourrions imaginer 2 Briques Rouges et une Brique Jaune, cela donne automatiquement tel bout de code, une Hashmap par exemple. Les modèles sont très souvent utilisés, mais il faut alors définir de manière exhaustive les assemblages de code supportés. Le troisième moyen est de se baser sur de la reconnaissance. Je regarde la Structure de ton langage et je suis alors capable de le compiler. Par exemple je vois une boucle "For"... je peux alors décider de faire de l'inline code pour toi lecteur... Je vois une Annotation, je me débrouille alors pour t'ajouter un comportement dans le code généré... Les Templates dans MPS sont des templates type Velocity ou Freemarker. Cela permet d'écrire rapidement du code de transformation. Les Patterns se basent sur votre propre langage, c'est donc un peu compliqué à expliquer ici en quelques lignes. **La boîte à outil** On sait que le succès de la plateforme Java est dû à son API qui, dès le départ, permet de gagner un temps fou et de ne pas réinventer la roue. Qui a encore la prétention d'écrire une Stack ou une FIFO ? c'est un peu comme ne pas reconnaître que ce problème a déjà été résolu et souvent de manière élégante. Dans ce cas, si nous parlons d'inventer un langage et d'avoir une approche de développement piloté par le Langage, n'y-a-t-il pas un risque de perdre ces bibliothèques ? Sergey n'hésite pas à parler de "Base Language" et de "Collections Language". Le Base Language offre des structures de bases comme les boucles, les itérations, les opérateurs mathématiques. Car il serait lourd et stupide de devoir inventer son "Addition" par exemple. Vous pouvez donc vous en servir comme socle de départ afin d'écrire votre langage. Mais ce n'est pas une obligation. Si vous pensez que les variables, les itérations ne sont pas nécessaires pour modéliser votre idée, libre à vous de ne pas vous en servir. Le Collection Language ressemble plutôt à une librairie pour moi. De même qu'en Java vous disposez du package java.util ou de la lib STL en C++, là il s'agit plus d'une librairie qu'autre chose. Enfin il parle aussi de "User Interface Language" destiné à exprimer l'interface utilisateur. Là j'avoue je me suis repris un café et j'ai continué ma lecture.

## MPS Meta Programming System

Cet outil de JetBrains est en cours de développement, il se destine à offrir un support pour la programmation orientée Langage (LOP). Je cherche un exemple simple pour vous montrer quelques concepts et essayer de voir si vraiment nous sommes face à une révolution... ou pas.

## Conclusion

A travers cette traduction des idées de Sergey Dmitriev, j'ai essayé de présenter d'une part pourquoi cette démarche apparaît, puis ensuite comment elle se concrétise. Je pense en effet que dans un avenir proche, même si cela vous dérange, nous arrêterons d'écrire du code "tuyau". Par code tuyau je parle bien de la soupe technique qui permet de mettre en marche correctement les outils du marché que sont Spring, Hibernate et Struts2. Avec la Crise nous aurons moins de temps. Nos clients vont réduire les budgets, la demande du client reste la même mais en moins de temps. Nous pensons que les outils nous font gagner du temps. C'est vrai. Mais il faut voir qu'aujourd'hui l'apprentissage et la maîtrise du langage et des frameworks demande de plus en plus de temps. Il y a même une certaine fracture entre l'expert formé sur les dernières technologies et le développeur qui débute. Comment bien choisir le framework afin de rester sur le marché de l'emploi ? Je me demande si nous aurons le temps d'enseigner correctement Hibernate ou Spring à l'université ou dans une école. Ces outils avancent plus vite que nous, et nous produisons de plus en plus de déchets, de code qui ne sert à rien... L'approche LOP est à mon avis un peu trop compliquée pour l'instant. Je suis pragmatique : je ne vois pas demain un cas précis où je pourrai m'en servir. D'un autre côté cela fait 10 ans que je vis, pense, travaille, respire avec Java. Comment alors pourrais-je

voir par le trou de la serrure si cette initiative a un avenir ou pas ? A vous le micro... Qu'en pensez-vous ? **Ressources** [Blog de Sergey Dmitriev Language Workbenches: The Killer-App for Domain Specific Languages?](#) article très intéressant de Martin Fowler [Article Wikipedia sur Software Factory Web Software Software Factory by Microsoft](#) [Le Générateur Celerio de Jaxio](#) propose une approche basée sur le modèle astucieuse. [VideoCast de Sebastien Arbogast](#) dans lequel il explique comment créer un langage très simple.

## Sondage : donnez votre avis sur le Touilleur Express

[Mise à jour mercredi : les résultats sont disponibles [sur cette page](#). Merci à ceux qui ont participé. ] Hop ce matin je compte sur vous pour remplir ce petit sondage. Je publierai sur le Touilleur les résultats d'ici à la fin de l'année. Merci pour votre participation. Vous pouvez soit remplir directement ci-dessous le formulaire, soit le remplir sur [le site de Google Docs](#).

### Sondage sur les lecteurs du Touilleur Express

Afin de mieux vous connaître, voici un petit sondage sur le blog le Touilleur Express. J'aimerais savoir un peu qui lit le blog, les sujets que vous aimez, ce que vous n'aimez pas...  
Merci de le remplir, je vous ferai partager les résultats dans quelques semaines.  
Nicolas Martignole

#### Depuis combien de temps connaissez-vous le blog le Touilleur Express ?

- c'est ma première visite
- depuis octobre 2008
- depuis juillet 2008
- depuis avril 2008
- depuis février 2008
- depuis l'année 2007
- avant janvier 2007
- Autre :

#### Comment avez-vous connu le blog le Touilleur Express ?

Plusieurs réponses sont possibles

- via un autre blog
- suite à une recherche Google
- par le Paris JUG
- par un ami, une connaissance, un collègue, votre mère...
- c'est toi qui m'en a parlé
- Autre :

#### Etes-vous abonné via RSS ? Google Reader ? Feedburner ?

Il est possible de s'abonner via la page principale en cliquant sur les icônes RSS à gauche

#### Parmis les sujets traités sur le Touilleur, que préférez-vous lire en premier ?

Une seule réponse possible, faites un choix

- Article sur l'actu de la communauté Java et les conférences
- Sujets sur Java, des frameworks, des outils ou des technos comme Flex GWT
- Articles sur Scrum, les méthodes Agiles, la gestion projet
- Linux, le monde de l'open-source
- Apple, l'iphone, le Mac
- Internet en général, Google, le Web 2.0
- Lorsque tu parles de ton travail et de toi

#### Parmis les sujets traités sur le Touilleur, quel sujet vous intéresse le MOINS ?

... et ce que vous lisez moins

- Article sur l'actu de la communauté Java et les conférences
- Sujets sur Java, des frameworks, des outils ou des technos comme Flex GWT
- Articles sur Scrum, les méthodes Agiles, la gestion projet
- Linux, le monde de l'open-source
- Apple, l'iphone, le Mac
- Internet en général, Google, le Web 2.0
- Lorsque tu parles de ton boulot

**De 1 à 10 comment notez vous la qualité (contenu, niveau technique, clarté) des sujets du blog ?**

1=nul, 5=bien, 10=exceptionnel

1 2 3 4 5 6 7 8 9 10

Nul           Exceptionnel**De 1 à 10 comment notez vous l'objectivité des articles ?**

1 2 3 4 5 6 7 8 9 10

Pas du tout objectif, trop engagé,  
pas assez neutre           Très  
objectif,  
neutre,  
impartial**Vous diriez plutôt que le Touilleur vous apporte pas beaucoup/un peu/beaucoup/énormément d'informations nouvelles ?**

Afin de juger le niveau de veille technique et d'information

- pas beaucoup d'informations nouvelles
- pas mal d'informations nouvelles
- beaucoup d'informations nouvelles
- énormément d'informations nouvelles
- sans opinion

**Aimeriez-vous avoir une version Podcast du Touilleur reprenant certains articles et complétés d'informations ?****Si vous souhaitez recevoir le résultat du sondage par email, merci de nous laisser votre email****Avez-vous un blog ?**

précisez l'adresse dans le champs commentaire à la fin du formulaire

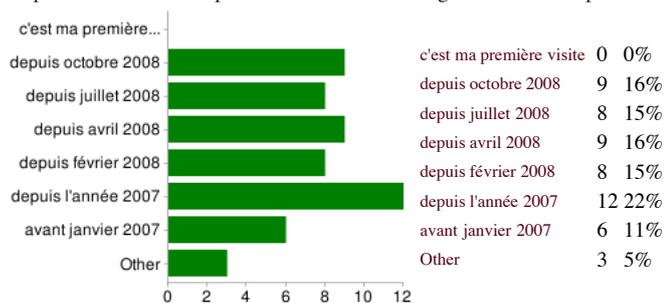
- Non je n'ai pas de blog
- Oui j'ai un blog perso
- Oui j'ai un blog d'entreprise (e.g. blog.xebia.fr)
- Les deux (perso et entreprise)

**Si vous souhaitez ajouter un commentaire à ce questionnaire**Fourni par [Google Documents](#)[Signaler un cas d'utilisation abusive](#) - [Conditions d'utilisation](#) - [Clauses additionnelles](#)

## Résultat du sondage

Voici les résultats du [sondage](#) sur le Touilleur extrait ce soir de Google Docs. **58** personnes ont participé au sondage, les résultats sont très intéressants. Je vous laisse regarder les résultats, j'ai ajouté mes commentaires à la fin. A noter que le copier-coller de Google Docs vers Wordpress permet de publier sans soucis les résultats, j'ai juste ajouté une balise style pour les couleurs... enfin Google Docs et Form c'est très pratique.

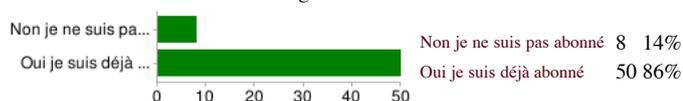
Depuis combien de temps connaissez-vous le blog le Touilleur Express ?



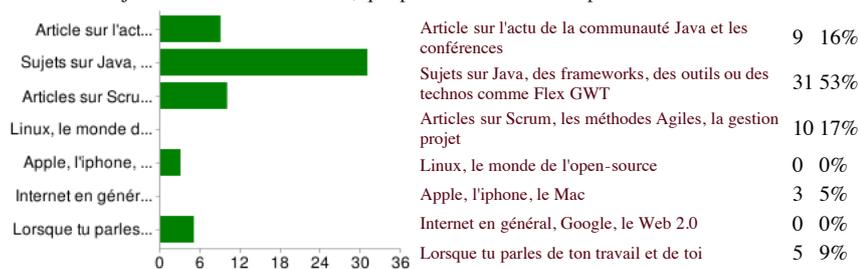
Comment avez-vous connu le blog le Touilleur Express ?



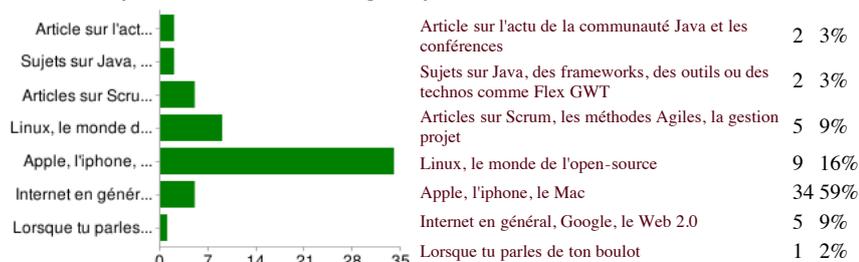
Etes-vous abonné via RSS ? Google Reader ? Feedburner ?



Parmis les sujets traités sur le Touilleur, que préférez-vous lire en premier ?

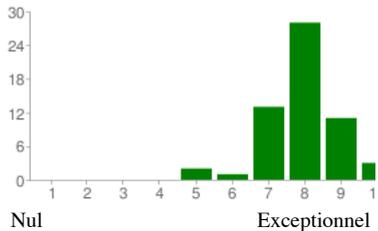


Parmis les sujets traités sur le Touilleur, quel sujet vous intéresse le MOINS ?



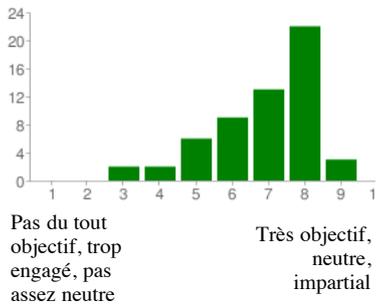
De 1 à 10 comment notez vous la qualité (contenu, niveau technique, clareté) des sujets du blog ?

1 - Nul 0 0%



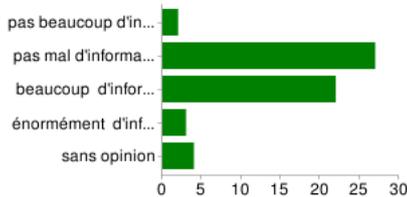
2	0 0%
3	0 0%
4	0 0%
5	2 3%
6	1 2%
7	13 22%
8	28 48%
9	11 19%
10 - Exceptionnel	3 5%

De 1 à 10 comment notez vous l'objectivité des articles ?



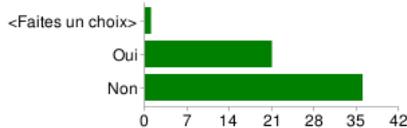
1 - Pas du tout objectif, trop engagé, pas assez neutre	0 0%
2	0 0%
3	2 4%
4	2 4%
5	6 11%
6	9 16%
7	13 23%
8	22 39%
9	3 5%
10 - Très objectif, neutre, impartial	0 0%

Vous diriez plutôt que le Touilleur vous apporte pas beaucoup/un peu/beaucoup/énormément d'informations nouvelles ?



pas beaucoup d'informations nouvelles	2 3%
pas mal d'informations nouvelles	27 47%
beaucoup d'informations nouvelles	22 38%
énormément d'informations nouvelles	3 5%
sans opinion	4 7%

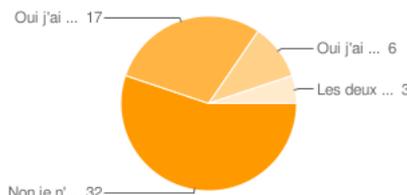
Aimeriez-vous avoir une version Podcast du Touilleur reprenant certains articles et complétés d'informations ?



<Faites un choix>	1 2%
Oui	21 36%
Non	36 62%

Si vous souhaitez recevoir le résultat du sondage par email, merci de nous laisser votre email ... j'ai retiré les emails...

Avez-vous un blog ?



Non je n'ai pas de blog	32 55%
Oui j'ai un blog perso	17 29%
Oui j'ai un blog d'entreprise (e.g. blog.xebia.fr)	6 10%
Les deux (perso et entreprise)	3 5%

**Conclusion** En majorité vous préférez les articles sur Java. Les articles sur Apple et l'iphone sont les moins demandés. La majorité des lecteurs a découvert le blog ces derniers mois, via un autre blog à 38%, via Google à 32% et via le Paris JUG à 4%. A une écrasante majorité vous êtes abonnés au flux RSS. La moitié des lecteurs préfèrent les articles sur Java ou les frameworks. A mon étonnement, personnes ne préfère lire en premier les articles sur Internet, le Web 2.0 et Google. Au niveau de la qualité, la note moyenne est de 8 sur 10, ce qui est très sympa et fait toujours plaisir à voir. Merci à vous. Du côté objectivité, bien que l'on parle d'un blog personnel sur lequel je donne mon avis, je suis étonné de voir qu'en majorité vous percevez une certaine objectivité dans le contenu des articles. Ce n'est pas bien/pas bien. Un blog peut être complètement subjectif, et les commentaires des lecteurs permettent à chacun de donner son avis, ce qui est important pour que chacun puisse donner son avis. Le Touilleur vous apporte pas mal d'informations nouvelles à 47% et beaucoup d'informations nouvelles à 38%. Notez qu'à 100 voix prêts on peut recompter si vous voulez :-). Le Podcast n'est pas un franc plébiscite puisque 62% ne souhaite pas de version podcast du Touilleur. Ok je le note. Et comme je suis motivé pour me lancer sur un podcast malgré tout, je vais essayer de faire un contenu sympa qui soit complémentaire du blog. 45% des personnes ayant répondu au sondage ont un blog perso ou professionnel, 5% ont à la fois un blog perso et un blog d'entreprise. Le blog d'entreprise représente 9 réponses sur 58, ce qui est pas mal du tout. Merci à toutes les personnes qui ont répondu, je vais laisser la feuille active et nous referons peut-être le point un peu plus tard.

## Brillez en société avec votre BlockingDeque

Encore un café qui se transforme en article sur le Touilleur. Avec Florent Ramiere de [Jaxio](#) nous discutons sur des techniques de synchronisation. Voici le problème à résoudre : soit une liste d'éléments à maintenir correspondant à des tâches. Des threads empilent des tâches à effectuer dans cette queue pendant que d'autres threads récupèrent à la sortie des tâches. Cependant on souhaite que cette liste chaînée offre des services supplémentaires afin que les consommateurs attendent qu'une certaine quantité de tâches soit disponible. Nous voulons aussi pouvoir prendre soit la tâche la plus ancienne (type FIFO) soit au contraire la tâche la plus récente de cette liste. Là vous êtes perplexe... ne faites pas les timides je sais que vous n'avez rien compris. Reprenons le problème avec un oeil différent : dans un restaurant, nous avons 5 serveurs qui se chargent de poser des commandes sur un passe-plat. Le passe-plat est une table où les serveurs posent les commandes. Les commandes les plus récentes sont posées à droite, de sorte que la commande du client qui attend depuis le plus longtemps se trouve à gauche. Les commandes sont enregistrées sur un tableau. Les 3 cuisiniers surveillent la liste des commandes. Un cuisinier dépile toujours la commande la plus à gauche posée sur ce passe-plat. Peu importe qu'un serveur l'ai posé à l'instant ou pas, il doit toujours traiter la commande la plus à gauche sur le passe-plat. Je sais que c'est simpliste comme image, mais vous allez comprendre pourquoi lorsque je vais sortir la nouveauté de la saison 2008... Ensuite nous voulons que les cuisiniers attendent qu'au moins 3 commandes soient présentes, afin d'optimiser la cuisson des aliments. L'idée est que lorsque 3 commandes sont présentes et que les cuisiniers n'ont rien à faire, ces 3 commandes sont retirées du passe-plat. Imaginez comment vos cuisiniers vont surveiller le passe-plat et attendre de manière synchrone que 3 commandes soient présentes... Enfin comme nous ne voulons pas non plus laisser les clients en attente trop longtemps, une commande qui a plus de 5 mn d'ancienneté a le droit d'être dépilée... sans qu'il ne soit nécessaire d'attendre 2 autres commandes... Ok ? Enfin pour terminer il faut aussi penser aux serveurs : nous souhaitons qu'ils attendent lorsque 5 commandes sont posées sur le passe-plat afin d'éviter d'engorger notre passe-plat pour rien. Dès qu'un cuisinier dépile une commande, le serveur sera alors débloqué et pourra poser sa commande sur le passe-plat. Donc nous avons une liste chaînée, avec de la synchronisation. Nous avons aussi une notion de seuil à respecter pour les consommateurs. Et enfin nous avons une notion de temps limite pour s'amuser encore plus... Il est temps de ressortir ses manuels et de regarder ce que nous avons en stock. Parlons d'abord "deque" (prononcer deck). **C'est quoi une deque ?** A prononcer "Deck", une deque est une "double-ended queue" soit une liste à double entrée en français dans le texte (on ne dit paspas double sortie en français, cela fait un peu trop tunning...). C'est une liste chaînée composée d'une tête et d'une queue. La différence avec une Fifo est qu'il est possible de dépiler soit au début, soit à la fin là où une Fifo ne vous laisse accéder qu'à la fin. Java 6 propose une interface [Deque](#). Les méthodes remarquables sont limpides :

	First Element (Head)		Last Element (Tail)
--	----------------------	--	---------------------

	<i>Throws exception</i>	<i>Special value</i>	<i>Throws exception</i>	<i>Special value</i>
--	-------------------------	----------------------	-------------------------	----------------------

<b>Insert</b>	<a href="#">addFirst(e)</a>	<a href="#">offerFirst(e)</a>	<a href="#">addLast(e)</a>	<a href="#">offerLast(e)</a>
---------------	-----------------------------	-------------------------------	----------------------------	------------------------------

<b>Remove</b>	<a href="#">removeFirst()</a>	<a href="#">pollFirst()</a>	<a href="#">removeLast()</a>	<a href="#">pollLast()</a>
---------------	-------------------------------	-----------------------------	------------------------------	----------------------------

<b>Examine</b>	<a href="#">getFirst()</a>	<a href="#">peekFirst()</a>	<a href="#">getLast()</a>	<a href="#">peekLast()</a>
----------------	----------------------------	-----------------------------	---------------------------	----------------------------

La méthode addFirst permet d'ajouter une nouvelle commande au début de notre passe-plat. Cependant si nous souhaitons faire passer une commande avant toutes les autres, la méthode addLast nous donne un droit d'écriture afin de placer notre commande à la fin. La méthode peekLast permet à un cuisinier de s'assurer que la dernière tâche en attente n'est pas dans la pile depuis un certain temps... Mais ce n'est pas encore la solution que je souhaite vous présenter. **BlockingDeque** Une BlockingDeque représentée par l'interface java.util.concurrent.BlockingDeque est une deque qui va attendre d'être non vide avant de laisser un consommateur dépiler un élément. Clairement : nos cuisiniers attendent sagement qu'une commande arrive. C'est aussi une Queue capable de faire attendre un serveur lorsque la liste est pleine, c'est de cette manière que nous allons faire attendre les serveurs lorsque la BlockingQueue est pleine. En Java, [l'interface BlockingDeque](#) est constituée entre autres des méthodes suivantes:

```
public boolean offer (E element, long timeout, TimeUnit unit)
public boolean offerFirst(E e, long timeout, TimeUnit unit) throws InterruptedException
public boolean offerLast(E element, long timeout, TimeUnit unit)
```

La méthode offer permet d'ajouter un nouvel élément à la fin de la liste, en attendant une certaine quantité de temps précisée par la valeur timeout précisé par l'unité de [TimeUnit](#). Si jamais finalement l'entrée n'a pas été ajoutée au bout du temps t, alors la méthode retournera false. Sinon elle retourne true. Pratique non ? La méthode offerFirst permet de faire passer une commande devant toutes les autres, tout en restant avec la contrainte du timeout comme précédemment. Du côté de nos consommateurs, les méthodes suivantes permettent soit de bloquer jusqu'à ce qu'une commande soit disponible, soit d'attendre tout en respectant un timeout. Les méthodes takeFirst et takeLast sont bloquantes :

```
public E takeFirst()
public E takeLast()
```

Les méthodes pollFirst() et pollLast() permettent d'examiner et d'attendre que la liste se remplisse, tout en restant dans un certain intervalle de temps comme précédemment :

```
public E pollFirst(long timeout, TimeUnit unit)
public E pollLast(long timeout, TimeUnit unit)
```

**LinkedBlockingDeque** Cette classe est une implémentation en Java de l'interface et permet de fixer une capacité maximum afin de limiter le nombre d'éléments dans la liste :

```
public LinkedBlockingDeque()
public LinkedBlockingDeque(java.util.Collection< ? extends E> c)
public LinkedBlockingDeque(int capacity)
```

**Un exemple complet** Voici un exemple complet avec un Serveur et un Cuisinier. Ces deux objets accèdent la même BlockingQueue. La capacité de la queue est de 5 commandes. Pour faire simple : le serveur ajoute un Integer à la BlockingQueue toutes les 300 millisecondes. Le cuisinier prend un élément toutes les 3000 millisecondes. Si le Serveur fonctionne correctement, au bout de 1500 ms (5 x 300) la queue des commandes sera pleine. Dans ce cas les 5 autres commandes suivantes seront mises en attente automatiquement, le Serveur sera bloqué.

```
import java.util.concurrent.BlockingDeque;

/**
 * A very simple producer class to demonstrate BlockingDeque fonctionnality.
 *
 * @author N.Martignole
 */
public class Serveur implements Runnable {
    private String name;
    private BlockingDeque<Integer> deque;
    // nous allons avoir 10 commandes a la suite
    private int[] numbers = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

    public Serveur(String name, BlockingDeque<Integer> deque) {
        this.name = name;
        this.deque = deque;
    }

    public void run() {
        for (int i = 0; i < 10; i++) {
            try {
                deque.putFirst(numbers[i]);
                System.out.println(name + " place une nouvelle commande #" + numbers[i]);
                System.out.println(name + " affiche l'état de la queue: " + deque + "\n");
                Thread.sleep(300);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Voyons la classe du cuisinier :

```
import java.util.concurrent.BlockingDeque;

public class Cuisinier implements Runnable {
    private String name;
    private BlockingDeque<Integer> deque;

    public Cuisinier(String name, BlockingDeque<Integer> deque) {
        this.name = name;
        this.deque = deque;
    }

    public void run() {
        for (int i = 0; i < 10; i++) {
            try {
                int j= deque.takeLast();
                System.out.println("Cuisinier "+ name + " prend " + j);
                System.out.println(name + " regarde la queue: " + deque + "\n");
                Thread.sleep(3000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Enfin il nous faut une class de Test pour tester le tout:

```
import java.util.concurrent.BlockingDeque;
import java.util.concurrent.LinkedBlockingDeque;

public class Tester {

    public static void main(String[] args) {
        // Creation d'une queue pour notre passe plat de 5 cases
        BlockingDeque<Integer> deque = new LinkedBlockingDeque<Integer>(5);

        // Creation d'un serveur et d'un cuisinier pour faire simple
        Runnable serveur = new Serveur("Serveur", deque);
        Runnable cuisinier = new Cuisinier("Cuisinier", deque);

        // Lance le tout
        new Thread(serveur).start();
        new Thread(cuisinier).start();
    }
}
```

Avez-vous constaté que le Serveur attend 300 millisecondes avant de poser une nouvelle commande là où le cuisinier attend 3000 ms ? Pour 10 éléments, au bout de 1500 ms le Serveur sera donc bloqué car la queue sera pleine. Nous devrions donc le voir attendre jusqu'à ce qu'une place se libère. Le résultat est intéressant :

```
Serveur place une nouvelle commande #1
Serveur affiche l'état de la queue: [1]
```

```

Cuisinier commence la commande #1
Cuisinier regarde le passe-plat []

Serveur place une nouvelle commande #2
Serveur affiche l'état de la queue: [2]

Serveur place une nouvelle commande #3
Serveur affiche l'état de la queue: [3, 2]

Serveur place une nouvelle commande #4
Serveur affiche l'état de la queue: [4, 3, 2]

Serveur place une nouvelle commande #5
Serveur affiche l'état de la queue: [5, 4, 3, 2]

Serveur place une nouvelle commande #6
Serveur affiche l'état de la queue: [6, 5, 4, 3, 2]

Cuisinier a terminé la commande #1
Cuisinier commence la commande #2
Serveur place une nouvelle commande #7
Cuisinier regarde le passe-plat [7, 6, 5, 4, 3]

Serveur affiche l'état de la queue: [7, 6, 5, 4, 3]

Cuisinier a terminé la commande #2
Cuisinier commence la commande #3
Serveur place une nouvelle commande #8
Cuisinier regarde le passe-plat [8, 7, 6, 5, 4]

Serveur affiche l'état de la queue: [8, 7, 6, 5, 4]

Cuisinier a terminé la commande #3
Cuisinier commence la commande #4
Cuisinier regarde le passe-plat [8, 7, 6, 5]

Serveur place une nouvelle commande #9
Serveur affiche l'état de la queue: [9, 8, 7, 6, 5]

Cuisinier a terminé la commande #4
Cuisinier commence la commande #5
Serveur place une nouvelle commande #10
Cuisinier regarde le passe-plat [10, 9, 8, 7, 6]

Serveur affiche l'état de la queue: [10, 9, 8, 7, 6]

Le serveur a terminé de placer les commandes
Cuisinier a terminé la commande #5
Cuisinier commence la commande #6
Cuisinier regarde le passe-plat [10, 9, 8, 7]

Cuisinier a terminé la commande #6
Cuisinier commence la commande #7
Cuisinier regarde le passe-plat [10, 9, 8]

Cuisinier a terminé la commande #7
Cuisinier commence la commande #8
Cuisinier regarde le passe-plat [10, 9]

Cuisinier a terminé la commande #8
Cuisinier commence la commande #9
Cuisinier regarde le passe-plat [10]

Cuisinier a terminé la commande #9
Cuisinier commence la commande #10
Cuisinier regarde le passe-plat []

Cuisinier a terminé la commande #10
Cuisinier a terminé de traiter les commandes

```

Comme on peut le constater, avec peu d'effort il est possible d'implémenter un système synchrone propre, sans faire appel à du vieux code comme `synchronized`, `wait` ou `join`. C'est l'une des forces de l'api `Concurrent` en Java qui permet de programmer des cas complexes comme celui-ci en quelques lignes. Pour terminer voyons le code du `Tester` avec 2 Cuisiniers. J'y ajoute une notion de `timeout` afin que les cuisiniers attendent 1000 ms au maximum lorsque la queue est vide. Pourquoi cela ? Car chacun des 2 cuisiniers va traiter 5 commandes, bien que cela ne soit pas déterministe. Je ne peux pas baser mon code uniquement sur la supposition que chacun des cuisiniers traite 5 commandes, rien ne me dit que par exemple le `GarbageCollector` ne va pas perturber mon code. Pour cela la solution est simple, il suffit d'ajouter un `timeout` comme nous l'avons vu au début de cette article, dans la class `Cuisinier`. La méthode `run()` de `Cuisinier` devient alors :

```

public void run() {
    for (int i = 0; i < 5; i++) {
        try {
            int j = deque.pollLast(1000, TimeUnit.MILLISECONDS);

            System.out.println(name + " commence la commande #" + j);
            System.out.println(name + " regarde le passe-plat " + deque + "\n");
            Thread.sleep(3000);
            System.out.println(name + " a terminé la commande #" + j);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
    System.out.println(name + " a terminé de traiter les commandes");
}

```

}

Le résultat de l'exécution donnera alors :

```

Serveur place une nouvelle commande #1
Serveur affiche l'état de la queue: [1]

Cuisinier 1 commence la commande #1
Cuisinier 1 regarde le passe-plat []

Serveur place une nouvelle commande #2
Serveur affiche l'état de la queue: [2]

Cuisinier 2 commence la commande #2
Cuisinier 2 regarde le passe-plat []

Serveur place une nouvelle commande #3
Serveur affiche l'état de la queue: [3]

Serveur place une nouvelle commande #4
Serveur affiche l'état de la queue: [4, 3]

Serveur place une nouvelle commande #5
Serveur affiche l'état de la queue: [5, 4, 3]

Serveur place une nouvelle commande #6
Serveur affiche l'état de la queue: [6, 5, 4, 3]

Serveur place une nouvelle commande #7
Serveur affiche l'état de la queue: [7, 6, 5, 4, 3]

Cuisinier 1 a terminé la commande #1
Cuisinier 1 commence la commande #3
Serveur place une nouvelle commande #8
Cuisinier 1 regarde le passe-plat [8, 7, 6, 5, 4]

Serveur affiche l'état de la queue: [8, 7, 6, 5, 4]

Cuisinier 2 a terminé la commande #2
Cuisinier 2 commence la commande #4
Cuisinier 2 regarde le passe-plat [9, 8, 7, 6, 5]

Serveur place une nouvelle commande #9
Serveur affiche l'état de la queue: [9, 8, 7, 6, 5]

Cuisinier 1 a terminé la commande #3
Cuisinier 1 commence la commande #5
Cuisinier 1 regarde le passe-plat [9, 8, 7, 6]

Serveur place une nouvelle commande #10
Serveur affiche l'état de la queue: [10, 9, 8, 7, 6]

Le serveur a terminé de placer les commandes
Cuisinier 2 a terminé la commande #4
Cuisinier 2 commence la commande #6
Cuisinier 2 regarde le passe-plat [10, 9, 8, 7]

Cuisinier 1 a terminé la commande #5
Cuisinier 1 commence la commande #7
Cuisinier 1 regarde le passe-plat [10, 9, 8]

Cuisinier 2 a terminé la commande #6
Cuisinier 2 commence la commande #8
Cuisinier 2 regarde le passe-plat [10, 9]

Cuisinier 1 a terminé la commande #7
Cuisinier 1 commence la commande #9
Cuisinier 1 regarde le passe-plat [10]

Cuisinier 2 a terminé la commande #8
Cuisinier 2 commence la commande #10
Cuisinier 2 regarde le passe-plat []

Cuisinier 1 a terminé la commande #9
Cuisinier 1 a terminé de traiter les commandes
Cuisinier 2 a terminé la commande #10
Cuisinier 2 a terminé de traiter les commandes

```

En conclusion nous avons vu un exemple simple de l'utilisation de la class `LinkedBlockingQueue`. Cette classe du package `Concurrent` de Java 6 offre des fonctionnalités puissantes suffisantes pour répondre à un grand nombre de cas pratiques et réalistes.

## Devoxx : c'est complet



Avec 3200 personnes inscrites la conférence Devoxx 2008 qui aura lieu du 8 au 12 décembre 2008 est complète ! C'est un vrai plébiscite et une grande réussite pour le JUG Belge, organisateur de cette conférence. **Les-gens-dont-je-vous-**

**parle-tres-souvent** Si vous regardez le panel des présentateurs, vous allez reconnaître quelques têtes connues dont nous avons parlé sur le touilleur : [Alexis Moussine-Pouchkine](#) (SUN/Glassfish) présente 2 sujets : \* [GlassFish in action](#) - Forget the app server for a minute! - University \* [10 reasons why Java EE development doesn't have to be painful](#) - Tools in Action Didier Girard (SFEIR/GWT) présente [GWT in Practice Dick Wall](#) et [Carl Quinn](#) sont les auteurs du Podcast "[JavaPosse](#)". Un enregistrement live est prévu pendant la conférence. Guillaume Laforge (Groovy/Grails/SpringSource) présente "[Groovy and Grails in Action](#)" Kirk Pepperdine, qui est venu faire [une présentation au Paris JUG](#), fera une présentation sur les [performances](#) de la machine virtuelle Java. Romain Guy présente 2 conférences sur les interfaces graphiques avancées sur la plateforme Java et sur [Android](#). Florent Ramière de [Jaxio](#) présente deux Quickies : \* [How to kickstart applications when Appfuse is not enough and MDA is too much ?](#) \* [When hype technologies are forbidden, what do you do to create web applications ?](#) Jérôme Bernard, [le cofondateur d'Elastic Grid](#), effectuera deux présentations : - [Dynamic deployment and scalability for the Cloud](#) dès lundi matin avec l'une des premières Bird of a Father pour commencer tôt - [Présentation sur les WebServices Amazon](#) Plus d'informations sur [le site d'Elastic Grid](#). **Même le Touilleur fait sa conférence** [Le Jeudi 11 décembre à 13h10](#) je présente un Quickie sur la gestion des performances, du traitement des données et des architectures orientées service appliqués à la finance. Vu le format très court de 15 minutes pour faire cette présentation, je travaille sur un format très proche d'un article de 4 à 5 pages. Je vous ferai partager le contenu de mes recherches avant la conférence afin d'avoir votre retour. L'objectif de cette présentation sera de donner une vision sur les dernières technologies, les frameworks et les pratiques utilisées pour gérer des volumes de données. La présentation dans l'idée que je m'en fais ressemblera à un article du touilleur : je souhaite parler avant des différentes technologies qui sont maintenant à notre disposition pour le traitement des données. **CNN Live en direct du terrain** Enfin j'emmène tout mon matériel : MacBookPro, iPhone, carte 3G Orange afin de vous faire partager à chaud l'actualité et de vous proposer chaque jour des reportages, des billets sur la conférence, comme si vous y étiez. Ce sera l'occasion j'espère d'interviewer du beau monde. Donc comme ils disent de l'autre côté de l'atlantique : Stay tuned !

## Petit compte rendu de la soirée JBoss au Paris JUG

Ce soir le dieu de l'informatique, le pigeon magique du JUG, le Duke ultime golden-platinum qui refait venir l'être aimé étaient tous aux abonnés absents. Panne de vidéo projecteur, panne de micro (c'est une contrepartie) et donc bilan mitigé. En première partie, [Sacha Labourey](#) le CTO de JBoss, une division de RedHat effectue une présentation sur JBoss AS 5.0. En deuxième partie, présentation de JBoss Seam par Malik Saheb. Le tout devant 200 personnes, la salle étant comble comme la fois précédente. Sacha Labourey présente d'abord un petit historique de JBoss Application Server. A ses débuts en 2000 la volonté des développeurs est de proposer un serveur d'app plus léger, avec la possibilité de recharger à chaud une application sans relancer le serveur. Pour cela l'architecture du micro-kernel basée sur JMX permet de gérer le chargement des ressources d'une application. J'en ai parlé en 2006 lors de ma formation à JBoss à Berlin. [4 articles du touilleur](#) vous donneront un petit aperçu du moteur de JBoss si vous souhaitez vous faire votre propre idée. Sacha enchaîne ensuite avec une justification sur la prochaine version de JBoss, JBoss AS5. En ligne de mire, la certification EE5, une infrastructure plus souple et plus puissante pour le clustering, une offre de messaging qui ne sera plus basée sur JBoss MQ mais sur JBoss Messaging, une réécriture de la couche Web Services avec les dernières innovations de JEE 5. L'un des objectifs de JBoss Application Server 5 est de s'affranchir du kernel basé sur JMX. Là où il y a quelques années, JMX était la solution pour connecter différents modules, aujourd'hui entre OSGI et l'injection de dépendances, Sacha explique que le nouveau moteur est basé sur des POJOs, accompagnés d'annotations. C'est donc un énorme travail de réécriture et de migration de code qui a entraîné le retard de la sortie de JBoss AS 5. **Quand sort JBoss AS 5 GA ?** Si l'on regarde JBoss 5, cela fait 3 ans que ce chantier a été lancé. La sortie est prévue dans les semaines qui viennent, vraisemblablement avant Noël. Le temps passé à écrire cette nouvelle version permet d'offrir une version plus mature de JBoss. Il faut se souvenir qu'il y a presque 3 ans, JBoss AS était l'un des premiers serveurs à offrir le support des EJB3 en l'état. La RC2 de JBoss AS 5 a été certifiée EE5 début septembre. Parmi les nombreuses nouveautés que Sacha nous présente dans ces slides, j'ai noté une nouvelle API de configuration, un nouveau moteur d'injection, une gestion des aspects, et beaucoup d'autres fonctionnalités. **OSGI ?** Sacha explique qu'OSGI ne donne pas tout. Il n'y a pas de gestion de classe unique, ni de gestion des méta-données. Enfin même si OSGI permet le rechargement partiel d'une application, il n'y a pas la notion de graphe d'état :INITIALIZING, RUNNING, STAND-BY... A la question "mais où est le marché pour ces besoins ?" la réponse est assez floue. Là où Spring dm Server nous promet de répondre à des soucis que nous n'avons pas encore, je pense que JBoss passe encore une étape supplémentaire vers le "on n'en n'a pas besoin mais c'est intéressant"... J'ai ensuite bien aimé une partie de la présentation de Sacha, qui est un orateur très intéressant et qui a vraiment séduit le public du JUG. Il explique par des chiffres le fait suivant : les éditeurs de serveur d'application J2EE se réduisent de version en version. - sur Java J2EE 1.2 : 18 serveurs d'application différents - sur la version 1.3 : 22 serveurs d'app - sur la version 1.4 : 17 serveurs d'app - sur la version EE5: 10 serveurs d'app - sur la future EE6 : 7 (IBM, Oracle, SUN, SAP, NEC, Kandlee et ITMakeSoft) Vous notez comme moi l'absence de Spring. Va savoir pourquoi. Comme Sacha l'explique, le marché des serveurs d'application est maintenant mature, d'où cette consolidation. La guerre est terminée. Quoique... Il présente ensuite la démarche de RedHat. JBoss AS est, et restera open-source. La politique de licence n'a pas changé. Open-source et open-standard. La version Enterprise offre une version industrielle indispensable pour mettre en production une application. Il est utopique de penser qu'aujourd'hui un client final met en production une version sans prendre du support. Ceci non pas pour des raisons techniques, mais avant tout pour des raisons... légales. Thomson-Reuters où j'ai travaillé 5 ans 1/2 a signé un partenariat avec RedHat afin que les versions des produits basés sur JBoss comme KGR s'appuient sur le support de RedHat en cas de problèmes avec JBoss AS... afin

d'assurer les clients finaux de la pérennité du socle technique. La version entreprise est supportée pendant 3,5 ou 7 ans. Les versions sont testées de manière intensive, et le support basé à Neuchâtel permet à RedHat d'offrir une qualité de service indispensable aux équipes de production. J'ai eu le plaisir de discuter avec Luc Texier [que j'ai déjà rencontré](#) en juin 2006, à propos du support et du travail de RedHat. La version Enterprise Application Platform (EAP) est strictement identique à la version open-source de JBoss, JBoss Application Server. EAP offre en plus d'autres outils pour les équipes de production comme JBoss ON, outil de supervision et de monitoring. Pour les esprits grincheux, il existe aussi [JOPR](#), outil libre et open-source qui est l'ancien JBoss ON, fruit d'un travail commun avec Hyperic. Sacha Labourey présente ensuite avec différents slides l'architecture de JBoss AS 5. C'est assez technique et trop long à raconter ici. Je retiens simplement l'ajout d'un Virtual FileSystem qui permet de créer virtuellement un EAR lors des tests par exemple. Les questions de l'assemblée tournent autour d'OSGI, des synergies entre les équipes de RedHat et les équipes de JBoss. Quelqu'un demande pourquoi ne pas standardiser le micro-kernel de JBoss ? Sacha répond que si la demande de la communauté est là, ils y travailleront. Il rappelle qu'Hibernate est sorti avant JPA, que Seam est disponible depuis 2006 et une partie de Seam est maintenant une JSR, la JSR WebBeans. Donc la démarche est plutôt : utilisons la technologie puis ensuite regardons comment en faire un standard. Contrairement à Spring, il dit clairement que pour lui les standards ont quelque chose de bien, ils évitent le syndrome "vendor-lock". Je vous laisse apprécier son point de vue, que je partage en partie. A une question sur "mais pourquoi autant de retard sur AS5 ?" il explique que la réorganisation du code source avec des équipes réparties dans le monde ne facilite pas ce travail. Imaginez que le noyau Linux décide de changer de la version 2.6.0.1.2.3.4.3402.beta1 à une version 3.0.... Enfin Sacha donne 2 informations intéressantes : - regardez le blog de Bob McWhister's <http://oddtthesis.org/> qui vous donne des nouvelles sur les discussions et les recherches de Bob - mod\_cluster est un connecteur http dont Sacha parle [sur son blog](#). Je reprends ici la présentation de mod\_cluster

mod\_cluster is an httpd-based load balancer. Like mod\_jk and mod\_proxy, mod\_cluster uses a communication channel to forward requests from httpd to one of a set of application server nodes. Unlike mod\_jk and mod\_proxy, mod\_cluster leverages an additional connection between the application server nodes and httpd. The application server nodes use this connection to transmit server-side load balance factors and lifecycle events back to httpd via a custom set of HTTP methods, affectionately called the Mod-Cluster Management Protocol (MCMP). This additional feedback channel allows mod\_cluster to offer a level of intelligence and granularity not found in other load balancing solutions. Within httpd, mod\_cluster is implemented as a set of modules for httpd with mod\_proxy enabled. Much of the logic comes from mod\_proxy, e.g. mod\_proxy\_ajp provides all the AJP logic needed by mod\_cluster.

En conclusion, présentation sympa de la part de Sacha dans des conditions inhabituelles : problème de vidéo projecteur, d'ordinateur, d'écrans au fond ne fonctionnant pas... Chapeau pour avoir animé la présentation et réussi à faire passer le message : venez voir JBoss Application Server 5.0

## JBoss Seam

Tout d'abord parlons du fond : présentation organisée par Malik SAHEB du framework léger d'assemblage, Seam. Malik est un senior solution architect qui vient du "back". Il nous explique qu'il va présenter JBoss Seam d'un point de vue technique, en parlant de son expérience. Comme il le dit lui-même il vient du côté obscur, il a travaillé sur le moteur transactionnel de JBoss et sur les EJBs. Le monde du Web était donc pour lui un challenge. Le constat est le suivant : JSF simplifie l'écriture d'application Web. Cependant il reste encore beaucoup de XML et de code technique pour accéder aux objets métiers et aux EJB3. Seam se propose dans un premier temps de simplifier le travail du développeur en apportant un moteur d'injection basé sur les annotations Java 5. Il évite d'ajouter des composants intermédiaires entre la vue et le modèle. Ensuite Seam est un framework d'assemblage, qui offre et simplifie au développeur un socle technique assez impressionnant : - AJAX, JSF, api RichFaces et IceFaces - EJB3 - JPA - BPM Je vous donne l'adresse de la RefCard que Malik nous a présenté : [Core Seam 2.1](#). La lecture de cette carte vous donnera une présentation complète et solide sur le coeur technique de Seam. D'un point de vue fonctionnalité, Seam offre différents outils simples comme la génération de PDF, l'envoi d'email, l'internationalisation, la gestion du bouton back, de l'history, de REST, l'intégration de Wicket, de Spring, bref vous l'aurez compris : c'est un bus où tout le monde se trouve embarqué afin de faciliter l'écriture d'application. Seam c'est ensuite des concepts novateurs comme l'introduction de deux scopes dans le domaine des applications Webs : le scope "Conversation" et le scope "Business". J'en ai déjà parlé sur le touilleur, je vous redonne [le lien](#) afin de vous laisser lire cette partie tranquillement. Seam c'est aussi un ensemble d'outils basés sur maven2 destiné à faciliter l'écriture et le démarrage d'une application. Tout comme Grails, Seam offre depuis 2 ans des outils pratiques pour créer une Entity avec ses pages CRUD, le tout en utilisant JSF+Ajax+une api comme RichFaces ou IceFaces. A noter que le support de GWT s'effectue aussi sans soucis, car Seam s'occupe de la partie serveur, là où GWT travaille sur votre poste. Il y a aussi quelques fonctionnalités très bien intégrées comme la Validation (JSR-303) ou Hibernate Search, ce qui évite toujours de réinventer la roue. Enfin Seam a aussi son moteur d'injection de dépendance et d'outjection... Un concept qui vous permet de replacer dans la session, dans la conversation ou dans le scope business un objet Seam. Si vous voulez, c'est un moyen très pratique pour gérer la persistance fine, au delà de la session HTTP qui reste assez pauvre. Dommage c'était très mal expliqué... et là mon cahier de note s'arrête ... Je suis sorti à la fin. Honnêtement, je vais me lâcher : présentation complètement ratée. Désolé Malik si tu lis cela, mais il y a plusieurs griefs que je vais justifier. Les gens qui lisent le touilleur savent très bien que je suis un fan de Seam, donc on est d'accord que je ne parle que de la forme et pas du fond de la présentation. Sur la forme : pas de chance avec ce @#\$ de vidéoprojecteur, il était impossible de lire correctement les exemples de code. Et là je parle de ma pomme, assis à côté de David qui filmait pour le JUG la session. Vraiment il faut peut-être revoir le template du Paris JUG et basculer sur un template avec un fond noir et du texte blanc, comme Devvxx. A tester au prochain JUG. Ensuite Seam est un

framework qui ne se démontre pas en listant ses fonctions, ce qu'il propose et s'il fait la pluie ou le beau temps. En tant que développeur j'aurai vraiment aimé avoir une présentation partant d'un cas concret, afin que pendant l'heure je puisse me faire ma propre idée de ce que je pourrai faire ensuite avec Seam. C'est un format plus adapté je pense pour faire passer le message. Enfin carton rouge pour pas mal de points : je n'ai pas entendu le nom de la personne qui a initié le framework Seam. Dommage car je pense que l'assemblée aurait aimé entendre que Gavin King, monsieur Hibernate, est l'auteur de Seam. Mince c'est pas un truc que l'on oublie de dire. J'aurai aimé aussi avoir une présentation de quelques minutes sur la différence entre Seam et la JSR-296, Web Beans, toujours initiée par le même Gavin King. Cela aurait mis en avant les éléments couverts par Seam, qui travaille sur un scope plus large que la spec JSR-296. Bref très déçu, j'ai peur que l'assemblée présente n'ait plus/pas envie d'aller tester Seam. Alors je pense qu'il faudra en remettre une couche, c'est pour cela que je suis motivé pour vous refaire une présentation sur le blog, à la mode de ce que j'ai envie de vous raconter. Donc stay tuned. **Conclusion** Soirée au final sympa. Au Paris JUG nous croisons toujours pas mal de monde, c'est l'occasion de faire des rencontres et de voir ou revoir de nouvelles têtes. J'ai bien discuté avec Thomas Parle de Prima Solutions, nous avons aussi parlé beaucoup avec David Dewalle. Enfin il est important que les sponsors continuent à apporter au Paris JUG toute l'énergie financière nécessaire afin que le JUG continue dans de bonnes conditions. David ne pensait pas qu'en 3 mois le JUG passerait de 30 à 200 personnes chaque mois... Voilà pour mon compte-rendu, pas eu envie de vider mon sac hier soir tard et de me lancer sur l'écriture d'un long billet. Donc là vous avez la version courte... à peine 2240 mots. Une paille.. A la semaine prochaine à Devvoox !!! Des articles plus anciens sur JBoss Seam du Touilleur : - [Convertissez vous à JBoss Seam mes frères](#) - [JBoss Seam 2.0 premiers tests](#) - [JBoss Seam : l'intérêt du scope CONVERSATION](#) - [Installer JBoss Seam](#) (article sans doute dépassé) - [Présentation de Luc Texier](#) de JBoss au PJBUG

## Mock JavaMail

A quelques heures de mon départ pour Devvoox je termine une petite démo basée sur Apache Camel et Apache Wicket. Je vous proposerai bientôt un grand article à tremper dans votre café mais là aujourd'hui je fais la version courte. Si vous avez déjà eu à tester JavaMail dans votre application, je parle de tests d'intégrations, sachez qu'il existe une petite librairie bien sympathique pour "mock" du code Java. Disons que vous souhaitez envoyer un email de votre interface Web pour remettre un mot de passe à zéro. Comment écrire un test unitaire sans utiliser un vrai serveur SMTP ? [Mock JavaMail](#) permet de simuler une boîte aux lettres et de se comporter comme un serveur SMTP pour l'envoi, un serveur POP3 ou IMAP pour la réception. Prenons par exemple un projet très simple. Voici tout d'abord la dépendance que j'ai ajoutée à mon fichier pom.xml :

```
org.jvnet.mock-javamail
mock-javamail
1.7
```

Prenons ensuite un test unitaire qui génère un email afin de tester l'envoi. Tout d'abord voici le code à améliorer qui génère un email afin de l'envoyer via smtp

```
private void sendSmtEmail(Address from) throws MessagingException {
    Properties props = new Properties();
    //props.put("mail.smtp.host", "localhost");

    Session session = Session.getDefaultInstance(props, null);
    session.setDebug(true);

    // create a message
    SMTPMessage msg = new SMTPMessage(session);
    msg.setSubject("A simple test email");
    msg.setFrom(from);
    msg.setContent("Email content", "text/plain");
    msg.setRecipients(MimeMessage.RecipientType.TO, "destinataire@yourdomain.com");

    Transport tr = session.getTransport("smtp");
    tr.connect("localhost", "anyUsername", "password");

    msg.saveChanges(); // don't forget this
    tr.sendMessage(msg, msg.getAllRecipients());
    tr.close();
}
```

Mock JavaMail se fiche pas mal du username et du password, donc vous pouvez mettre ce que vous voulez, tant que le serveur SMTP dans la construction de l'instance Transport est bien "localhost". Ensuite un exemple de code dans lequel je déclare un objet de type Mailbox, afin d'envoyer un email.

```
public void testSendMail() throws Exception {
    Address address = new InternetAddress("destinataire@yourdomain.com");
    Mailbox mailbox = new Mailbox(address);

    assertTrue("The mock mailbox is not empty whereas it should be", mailbox.isEmpty());

    // Classical code to send an email to a smtp server
    sendSmtEmail(address);

    // Get a fresh reference to the mailbox
    mailbox=Mailbox.get("destinataire@yourdomain.com");
    // Check that a mail has been received
```

```

assertFalse("The mock mailbox is empty", mailbox.isEmpty());

Message message=mailbox.get(0);
assertEquals("A simple test email",message.getSubject());
}

```

Vous pouvez maintenant facilement tester l'envoi des messages de votre application et vous assurer que tout fonctionne sans devoir passer par un FakeSMTP :-). Allez, rendez-vous à Devovx. J'ai déjà récupéré mon teeshirt du Paris JUG grâce à David, la fin de semaine s'annonce chargée.

## Devovx 2008 c'est parti

Devovx 2008 c'est parti. Arrivé au Metropolis, je récupère mon package : sac à dos, tee-shirt et surtout badge RFID. L'organisation a mis en place cette année ce système afin de suivre anonymement les flux et le nombre de développeur par conférence. L'événement se déroule dans un mégaplex type UGC. Les conférences ont lieu dans des salles de cinéma, tout à fait confortable et bien équipées. Il faut savoir que le soir après 19h00 l'organisation de Devovx est tenue de rendre les grandes salles car le soir... il y a de vrais films. En arrivant ce qui marque c'est tout d'abord la qualité de l'exposition. Certes c'est la partie marketing de la conférence. Mais chaque stand fait réellement l'effort de proposer du contenu intéressant pour nous, les développeurs. Après avoir croisé l'équipe du Paris JUG au complet, j'ai croisé aussi Thomas Heute de JBoss ainsi que Tugdual Grall d'eXo Platform. J'ai aperçu le monsieur du jug de Tours, et j'ai retrouvé Florent de Jaxio qui doit présenter un Quickie sur... SpringFuse. Je vous en parlerai en exclusivité très bientôt. Mais sachez qu'il y a un gros coup en préparation. Allez il est temps d'aller maintenant à ma première conférence... "Fingers in the Air : A Gentle Introduction to Software Estimation" Le sujet : comment améliorer l'estimation du temps de développement d'un logiciel. PS : notez que je blogue en live via le Wifi de Devovx... Ça semble fonctionner cette année.

## Devovx : présentation sur les estimations du temps de dév

Premier billet écrit entre 2 présentations et avant mon petit quickie tout à l'heure. J'ai assisté tout d'abord à une présentation sur les techniques d'estimation du temps de développement appliquées à notre secteur. Le speaker, Giovanni Asproni, a donné un ton très sympa à cette présentation. Dans un premier temps, un bon mot pour commencer:

Estimation : Prediction is a very difficult task especially about future ones..."

Giovanni demande à l'assistance qui donne des estimations de temps de développement ? la majorité de la salle. Qui se fait négocier par son management ces temps de développement ? Encore une fois pas mal de monde. Il explique ensuite la différence entre "Estimation" et "Engagement". L'estimation est la mesure objective du temps nécessaire pour faire une tâche. Cette estimation n'est donc pas **négociable**. L'engagement représente le but à atteindre et donc la promesse. Cette promesse regroupe un niveau de qualité, un certain nombre d'itérations, bref c'est bien la promesse de ce que le client aura. L'estimation c'est le prix à payer. Sur une route vous roulez avec un camping-car dont la hauteur est de 2m30. Arrive un pont dont la hauteur est limitée à 2m00. Est-ce que vous pensez réussir à faire passer votre camping-car sous ce pont ? non. Vous pouvez toujours estimer si "ça passe" mais dans ce cas, à vous d'assumer les conséquences. Donc l'estimation étant une mesure approximative, mais une mesure, vous ne devez pas négocier cette estimation. Vous devez donc négocier les engagements. Je reprends l'image du réservoir d'essence : lorsque le voyant s'allume car le réservoir est presque vide, sans le savoir votre cerveau bascule en mode "estimation". Et vous êtes d'accord pour dire que cette estimation n'est pas négociable. Il fait ensuite référence à un très bon livre de Steve McConnell "Software Estimation" afin d'expliquer finalement en quoi consiste une estimation. En tant que développeur nous allons nous efforcer de donner une valeur pour la taille, le coût, l'effort en jours et les risques pour réaliser une tâche. Lorsque votre responsable vous demande une estimation vous devez donc répondre "3 jours avec un risque important que la base ne tienne pas la charge, à 2 développeurs par jour avec une licence de GridGain" Les estimations sont par nature sources de problème et de frustration, car le management tient pour acquis le contrat passé entre vous et eux. Pour cette raison, Giovanni pense que les processus Agile par leurs cycles itératifs sont plus adaptés au développement logiciel que l'effet escalier, basé sur la contractualisation des échanges entre les développeurs et les personnes du business. Une spécification ne pourra jamais résoudre un problème de communication. Il présente ensuite un cône sur une échelle de temps afin d'expliquer qu'en général, les estimations données pour une activité sont fausses, et que la marge d'erreur va de 0.25 à 4 sur une échelle verticale. Si je réponds 3 jours pour faire ce développement, il pourra en fait en prendre 4 fois plus. Ce travail est basé sur l'analyse de gros projets dans l'industrie de l'armement américaine, et des travaux de la NASA. Ce cône est donc une fenêtre de tir pour nous clairement que nos estimations sont en générales fausses. **Comment alors réussir à travailler et répondre aux attentes des clients ?** Dans un premier temps, il faut passer beaucoup plus de temps sur l'engagement, sur la notion du fini, sur ce que désire le client. Regardez le temps que vous passez pour acheter une cuisine chez IKEA ou pour acheter une voiture. Pour quelles raisons le client refuserait-il de vous répondre sur un projet d'un million d'euros ? **Comment gérer les changements du client ?** Vous devez penser que les changements sont gratuits et qu'ils vont arriver en cours de développement. Pour cette raison, les estimations doivent aussi être revues à la hausse ou à la baisse. Il est aussi important de rappeler à votre client que s'il décide de prendre les jantes alliages, cela coûte alors 400 EUR de plus... Pensez à toujours ajuster vos estimations afin que le client ne découvre pas au dernier moment l'addition. **Faut-il surestimer ou sous-estimer ?** A cette question, Giovanni répond que les développeurs sont trop optimistes de 30%. Par sécurité il faut donc mieux ajouter un

peu plus à une estimation initiale. Cependant, l'effet inverse s'appelle [la loi de Parkinson](#) : le temps de travail temps à rejoindre le temps donné pour effectuer une tâche. Si tu me donnes 10 jours pour faire cette tâche, je vais tendre à remplir ces 10 jours, c'est humain. Avoir trop de temps disponible entraîne aussi le phénomène de [procastination](#) (décidemment avec le Touilleur Express vous faites aussi de la psycho). On tend à remettre à demain des tâches (payer ses impôts...) car on sait que l'on a jusqu'au "XXX pour payer". **Les techniques d'estimation** Il en existe un grand nombre, certaines basées sur l'observation du passé pour prédire le futur. A noter que dans la finance c'est une technique de calcul du risque sur un produit financier. Il y a des techniques scientifiques basées sur les probabilités, mais comme il le montre avec des formules un peu complexes, cela ne remplace pas l'humain. Son point de vue est que les techniques suivantes fonctionnent bien : - faire faire les estimations par les développeurs, surtout pas les chefs de projet - faire des estimations collégiales pour que tout le monde donne son avis - utiliser des analogies pour améliorer la précision d'une estimation - ne pas oublier les activités connexes comme les tests, la mise en production, les tests en UAT par exemple. - utiliser vos experts pour avoir de bonnes estimations mais ensuite pensez à augmenter ce temps. Tout le monde n'est pas un expert. Les domaines où les prédictions/estimations sont appliqués sont la Finance, avec le succès que l'on sait aujourd'hui et sinon la dame en mini-jupe le dimanche soir sur TF1 : la météo de la semaine. Il y a aussi Mme Irma dans sa roulotte et Mr Marabout à Barbès. Bref les estimations sont par nature fausses, il faut simplement travailler avec le client afin de le rassurer, de mettre en place des plans d'atterrissages d'urgence si le budget est dépassé. Et il faut aussi rappeler au client que le prix à payer n'est pas négociable. Voilà pour cette première présentation

## Devoxx : Android, Romain Guy

Après un déjeuner rapide, des rencontres avec les développeurs de SFEIR (ils sont venus à 12 avec Didier Girard), avoir croisé aussi des personnes d'OCTO, j'ai assisté à la présentation de Romain Guy sur la plateforme Android de Google. [Romain Guy](#) est maintenant développeur chez Google en charge du développement de la partie graphique et composant de la plateforme Android. Avec son background et son expertise sur Swing, il a donc basculé vers le monde de l'embarqué. D'un point de vue stratégique je pense qu'il a fait le bon choix : la plateforme qui sera le plus massivement utilisée dans les années à venir pour accéder à Internet sera le téléphone. Durant sa présentation il a montré quelques effets visuels du moteur de rendu d'Android, ainsi que son architecture. Avec des exemples de code parfois un peu trop détaillé, il a présenté des exemples d'utilisation et de mise en place pour Android. En tant que développeur avec de bonnes connaissances en design et de Photoshop, j'étais dans mes petits souliers. Cependant je doute que l'explication du dithering ait passionné tout le monde. Un reproche que je ferai à Romain, c'est que comme à Paris lors des [SUN TechDays](#), il nous embarque sur des présentations avec beaucoup de code. Il est clairement difficile de présenter les sujets sur lesquels il travaille. A sa place je ne sais pas si je m'en sortirai mieux. Cependant regardons comment les gens d'Adobe finalement présentent Flex. Ils partent d'un exemple, nous montrent quelques lignes de MXML et surtout, ils insistent sur la simplicité et les effets. Cela permet de sensibiliser peut-être 70% de la salle. En attaquant par la face Nord sur Android, nous avons d'un côté une excellente présentation sur le coeur de l'API mais d'un autre côté, la partie vulgarisation est peut-être un poil trop absente. Cela dit la présentation était complète et beaucoup de personnes ont posé des questions relativement... simples.

## Devoxx : Flex, Flash 10 and the Enterprise

Présentation pour continuer sur "[Future of Rich Internet Application](#)" par l'un des sponsors de Devoxx, Adobe Systems. Sur scène, pas mal de beau monde : Matt Chotin et surtout Chet Haase. Là je vous sens incrédule... Chet Haase était anciennement chez SUN et travaillait sur... JavaFX. Il est maintenant chez Adobe et il travaille sur Flash 10. Comme quoi, on ne va pas jouer les tabloïds à 2 euros mais cela donne une idée. C'est aussi le coauteur avec Romain Guy d'un bon livre sur Swing et les techniques avancées pour les clients lourds. **Intro** Où en sommes-nous de Flex aujourd'hui ? Quoi de neuf du côté du runtime ? La prochaine version de Flex "Gumbo" propose d'apporter un grand nombre de fonctionnalités en particulier dans le domaine de la gestion des effets 3D. Ce qui m'a plus intéressé c'est l'annonce d'un partenariat entre SpringSource et Adobe, afin d'améliorer le travail entre Adobe LifeCycle d'une part et Spring d'autre part. Est-ce à dire que SpringSource s'appuiera en partie sur la communauté d'Adobe ? En tous les cas cela permet dès maintenant des opérations simples comme utiliser un Bean Spring, le déployer dans un serveur léger type Tomcat, avec le moteur LifeCycle. Pour ceux qui découvrent Flex, il s'agit d'un moteur basé sur Flash de génération d'interface client riche, afin de construire des applications embarquées dans un navigateur ou aussi des applications déployées sur nos postes, via Adobe AIR. Pour écrire une application Flex, vous pouvez utiliser la syntaxe MXML ou le langage ActionScript. La création des écrans s'effectue avec Flex Builder. Enfin Flex propose 4 protocoles d'échanges entre le navigateur et le serveur : HTTP, SOAP, AMF et RTMP. AMF est le format binaire d'Adobe et le format RTMP est le format bas niveau basé sur TCP. Flex aujourd'hui c'est un ensemble d'applications et de sites déployés. J'en profite pour parler de l'excellent site [Parley's.Com](#) qui vous permet de suivre à distance Devoxx. Flex bénéficie d'un effet d'adoption massif en particulier depuis la sortie de la dernière version de la plateforme. Si vous souhaitez tester Flex, je vous conseille ce site : <http://flex.org/tour>. Si vous souhaitez tester un petit sympathique utilisant Adobe AIR, je vous conseille [Balsamiq Mockup](#). **Flash Player 10** Flash 10 s'améliore, donc Flex s'améliore. C'est en substance le message de cette partie. La dernière version sortie en octobre propose des effets assez bluffants montrés durant la présentation. J'ai eu du mal à retrouver cette démo, voici [une page](#) reprenant cette démonstration. Nous avons eu ensuite une démo des Pixel Blender (sorte de shader pour flash) ainsi que du moteur de rendu de texte et enfin la démo des fonctions sonores. J'ai été pas mal bluffé par la qualité des démonstrations. Il ne manque rien à Flash pour en faire un moteur tout à fait correct pour faire des jeux vidéos en 3D. Les fonctions de

traitement de l'image et de la vidéo étaient aussi intéressantes. **Du côté de Flex** Tout d'abord la sortie d'Adobe AIR 1.5, puis ensuite le présentateur a parlé de Gumbo, la version à venir de Flex. L'objectif est d'offrir le support de Flash Catalyst, le pont entre le monde des designers et le monde des développeurs. Peu de gens sont à la fois des développeurs et des designers. L'idée de catalyst est de vraiment forcer la séparation entre la partie représentation (l'image) et la partie logique (un bouton sur lequel je clique). Pour illustrer que l'équipe de développement de Flex travaille essentiellement sur ce sujet, Chess prend la photo d'un urinoir avec une fausse mouche collée au fond. Il explique qu'en tant qu'américain c'est un concept qu'il ne connaissait pas, mais qu'il comprend que cela aide les gens à se concentrer et à ne pas se disperser. Si tu ne veux pas que tes développeurs codent n'importe quoi, met une mouche au fond de l'urinoir. **Le cube qui tourne en 3D** Démonstration du modèle d'animation. Chess nous montre des carrés disposés en étoile. Ok. Ensuite il ajoute une fonction de rotation 3D en mxml et un slider. Ok. Et ensuite il s'amuse avec le slider... oh magie les carrés s'élevaient et nous sommes en 3D. Ce qu'il faut retenir : il sera très simple dans la prochaine version de Flex d'ajouter des animations 3D sur des objets simples en 2D. **Le vrai truc intéressant pour nous les geeks** [James Ward](#) est l'un de mes speakers préférés. Il avait déjà fait son petit effet au Paris JUG cet été mais là encore, même si Flex est un sujet facile à traiter en conférence, le bonhomme maîtrise son sujet. Comme il l'explique tout d'abord, en terme d'architecture, Flex revient sur le modèle client-serveur classique que les architectes connaissent. Idéalement, le serveur est sans état et le client Flash avec votre application conserve l'état. Le souci pour les gens venant du monde Java c'est le souhait de conserver notre modèle Java et nos couches métiers. Comment alors travailler avec Flex ? Adobe propose 2 solutions : - Adobe BlazeDS d'une part, open-source libre et gratuit - [Adobe LiveCycle Data Services](#) version commerciale avec plus de fonctionnalités LiveCycle DS est une web application Java déployée sur un serveur d'app léger comme Tomcat ou un Weblogic. Il facilite grandement l'échange d'objets entre le client Flex et le serveur, en utilisant des techniques de sérialisations d'objet et le protocole AMF d'Adobe (entre autre). James Ward en profite pour annoncer un partenariat entre SpringSource et Adobe, afin de renforcer les synergies autour de Flex. Et il se propose de faire une démonstration en live. (au passage je vous conseille [cet article](#) de Sébastien Arbogast) Ce nouveau projet s'appellera "**Spring BlazeDS**" et il permettra entre autre de bootstraper le MessageBroker de BlazeDS comme un bean managé par Spring. La servlet Spring Dispatcher sera aussi en charge avec un système de templates Spring du passage de message vers le client. Le code m'a semblé propre et clair. Difficile ici de vous donner les détails, mea culpa. Pour terminer James a parlé de **génération de code** et d'application rapidement de type CRUD. Pour cela nous avons vu la démonstration suivante : dans Flex Builder, une datasource est configurée. Il y a une table "PRODUCT". Je clique sur cette table et je la glisse vers mon Canvas Flex. Celui-ci me demande si je souhaite utiliser une DataGrid pour listes tous les produits ? je mets oui. Ensuite en regardant le code MXML généré on voit en fait que c'est le code standard que nous aurions écrit avec LiveCycle DS du côté serveur. Cependant cela fait gagner du temps. Il lance ensuite sa démo, la grille apparaît avec 10 produits. Il montre ensuite l'édition d'une des colonnes et il change le prix. Après avoir validé, on voit dans la base que le client (du flash) a renvoyé au serveur (webapp livecycle DS) un message et que le moteur de livecycle avec du spring a modifié la valeur dans la base de données... Ca c'est cool. Il lance ensuite une deuxième fenêtre Firefox sur la même page. Ok je vois le même tableau avec mes 10 produits et ma valeur modifiée. Il change encore une fois le contenu de la cellule, et valide. Et là, on voit la valeur se mettre à jour instantanément dans l'autre Grille dans l'autre fenêtre de navigateur. Il explique que LiveCycle dispose d'un système de notification avec du Comet et que le client est donc notifié en cas de changement... cela montre la puissance et la simplicité de Flex pour les sceptiques. En conclusion : encore de bonnes surprises par les personnes d'Adobe qui donnent l'impression d'en avoir sous le pied. On sent qu'un gros mouvement s'amorce du côté du navigateur. Je me demande si JavaFX sera sur le même type de marché ou pas... A suivre

## Devoxx : jeudi

Devoxx c'est un truc de malade. Je suis rentré mercredi soir à 23h00 en tram après une BOF (Bird of a Feather) sur Mule présentée par Antoine Borg, l'un des membres actifs de la communauté Mule. La première journée était vraiment intense, et on va pas se plaindre, mais ça fatigue. Organisation parfaite, honnêtement le rapport qualité/prix de Devoxx est imbatable. Pour 70 EUR de Thalys, une ou deux nuits d'hôtel et l'entrée, compter un budget de 500 EUR. 3200 personnes, 160 speakers, 38 nationalités... J'ai croisé Didier Girard et peut-être 12 développeurs de SFEIR. Il est clair que la participation à ces événements est un excellent moyen pour sensibiliser, former et motiver une équipe de développeur. Devoxx a aussi un ton que JavaOne n'a plus, c'est vraiment l'endroit où il est possible de rencontrer les gens qui font les produits que vous utilisez. A noter cette année l'absence des gens de SpringSource, en raison de l'événement Spring en Floride. Dommage. Ce matin la journée commence bien : je discute avec une autre personne qui attend un taxi comme moi, avec un sac à dos Devoxx... Tiens mais je connais ce monsieur : [Julien VIET](#) d'eXo Platform et anciennement responsable de JBoss Portal chez RedHat. On a fait le chemin ensemble et c'est toujours sympa de rencontrer des gens dont je lis les blogs ou dont je suis l'actualité via Tweeter. Aujourd'hui c'était le grand jour pour moi, petite présentation sur la vie d'un architecte dans le domaine de la finance autour des sujets d'actualités comme les ESB et les frameworks d'intégrations légers. La présentation s'est bien déroulée, et il y avait pas mal de monde. Je passais après la présentation vraiment intéressante de [Paul Fremantle](#) sur le sujet "Complex Event Processing". Paul est le **CTO** de la société WSO2. Son entreprise est aussi le support du projet open-source Apache Synapse. Je reviendrai sur sa présentation dans un autre article. Aujourd'hui j'ai donc assisté aux présentations suivantes : - Complex Event Processing (Paul Fremantle) - Project Fuji - Taking OpenESB to the next level (Andreas Egloff) - JavaPose.com LIVE (énorme...) - JBoss Seam and beyond (par les gens de [RealDolmen](#)) - JAX-RS : the Java APIs for RESTful Web Services (Paul Sandoz) La meilleure présentation pour moi était la dernière sur REST, durant laquelle j'ai appris pas mal de choses. La présentation sur Project Fuji m'a permis de découvrir le moteur d'ESB open-source supporté par SUN, massivement basé sur OSGI et donc très modulaire et très simple. J'en reparlerai un peu plus tard. La présentation sur

Seam était franchement trop compliquée, à croire que Seam ne s'explique pas ou que comme disait [Jean-Laurent de Morhlon](#), Seam est mauvais ? Bref nous sommes ressortis et j'ai croisé Christophe qui travaille à l'INA avec Gik, Florent Ramière qui passait en coup de vent et un monsieur que je n'ai pas reconnu (j'aurai dû) mais que j'ai ensuite vu aux côtés de Vincent Massol sur le stand XWiki... Monsieur X je vous passe le bonjour. Le meilleur moment de cette journée a été l'enregistrement du podcast [JavaPosse](#). Dick Wall et Carl Quinn en live, devant 500 personnes, ça déménage. Et alors la cerise sur le gâteau : voici la photo de monsieur le Touilleur entre les 2



JavaPosse !!!

JavaPosse c'est un podcast à écouter chaque semaine. 4 geeks parlent de l'actualité de Java, Scala, Groovy, JRuby, Python, SUN... bref tout ce qui nous intéresse. La séance a commencé par la distribution de bières à tout le monde ! S'en est suivi donc une course au décapsuleur, et ensuite l'enregistrement du podcast a commencé. Joe Nuxoll et Tor Norbye étaient tous les deux à San Francisco, connectés via iChat. Ce qui a donné par la suite une bonne tranche de rigolade car Joe n'a pas cessé de mettre l'ambiance. Très bon moment. On en avait les larmes aux yeux c'était cool. [caption id="attachment\_573" align="alignnone" width="300" caption="Devoxx 2008 JavaPosse"]



[/caption] Une autre photo



Après une bonne journée j'ai croisé des gens d'OCTO occupés [à bloguer](#). A noter que Fabrice (et d'autres) n'avaient pas fait le rapprochement que l'auteur du touilleur express c'était bibi. C'est grâce à mon tee-shirt que les gens m'ont reconnus plusieurs fois dans la journée. Des belges qui m'ont vu, j'ai croisé un suisse (laurent) qui m'a parlé du touilleur, c'est cool ! J'ai une photo sympa prise cet après-midi :



Enfin pour terminer on s'est fait un petit restaurant mexicain avec Zouer et David du Paris JUG, Jean-Laurent de Morhlon, [Sebastien Letélie](#) et Daou, un collègue de Jean-Laurent de Vidal. Je vous recommande aussi d'aller lire [le blog du Paris JUG](#). Antonio a rédigé des billets très complets qui expliquent un peu mieux ce qui s'est passé cette semaine à Devoxx. Demain matin 3 conférences et c'est terminé. L'année prochaine si je peux je viendrai plus longtemps. Restez connecté sur le Touilleur, d'autres message suivront demain et ce week-end.

## Devoxx : présentation Amazon et Elastic Grid

J'ai assisté pour commencer par une présentation très riche et très intéressante des services d'Amazon et ensuite de la solution open-source [Elastic Grid](#). Amazon offre des services en location vous permettant de louer de l'espace de stockage, de louer de la puissance machine, de la base de données ou encore de louer un système de messages. Le business model est similaire à ce que propos finalement une compagnie d'électricité : vous ne payez que la consommation. Les services disponibles : - [Amazon Simple Storage Service \(S3\)](#) est un espace de stockage répliqué automatiquement. - [Amazon Elastic Compute Cloud \(EC2\)](#) est une solution de location de puissance machine. Vous vous retrouvez avec la possibilité d'avoir 3000 serveurs le temps d'un week-end par exemple pour y faire tourner votre serveur d'application... Windows Server 2003, Linux RedHat Enterprise, Debian, Ubuntu, OpenSolaris... - [Amazon SimpleDB](#) offre une interface de type web service (REST ou Soap) pour stocker, rechercher et indexer des données, le tout basé sur EC2 et S3. - [Amazon CloudFront](#) est une offre d'hébergement pour les données statiques des sites webs comme les images ou les CSS. Il faut savoir que sur les gros sites internet, il est plus intéressant de faire héberger les données statiques en dehors du serveur afin d'économiser la bande passante et d'éviter de solliciter un serveur d'app ou PHP pour charger des images. - [Amazon Simple Queue Service](#) est un web service qui permet de poster et de récupérer des messages, comme sur une Queue. Les messages sont limités à 8KB au format texte, une application de ce système est de permettre à n'importe quelle de vos applications de s'échanger des événements simples. La communauté des utilisateurs de AWS (Amazon Web Service) est de plus de 440 000 développeurs. Il y avait en octobre 2008 plus de 25 milliards d'objets dans le cloud Amazon S3. En octobre 2008 il y avait 800 millions d'objets stockés dans S3... L'activité de la plateforme AWS est deux fois plus importante que la plateforme classique Amazon.com... qui tourne elle-même sur EC2 et S3... Un site de partage de photos comme [SmugMug.com](#) a réalisé une économie de 500 000 USD en arrêtant d'utiliser un datacenter classique pour basculer vers un système basé sur S3... Autant dire que lorsqu'un financier lira ses lignes, j'espère que nous aussi nous arrêterons de faire héberger un site de tirage photos chez un prestataire classique... S3 est assez intelligent pour se rendre compte aussi de l'usage des fichiers que vous y placez. S'il s'avère qu'un fichier est souvent demandé, il va alors le répliquer dans le nuage afin que son accès soit plus rapide. Il va aussi le déplacer des USA vers l'Europe si finalement la majorité des clients sont en Europe. Cher lecteur, imaginez ton site web entrain de déménager pendant la nuit vers l'Asie car des Indiens lisent tes articles, puis qu'ensuite il soit en Europe quelques heures après... Vive les déménageurs virtuels ! Du côté des adresses IP, le moteur d'Amazon a pensé à tout. Vous pouvez louer pour une partie de jeux en réseaux des serveurs, avec un lot d'adresse IP, le temps d'un week-end. Le site [Animoto](#) propose un système de création de slideshows avec vos photos. En bref : vous mettez les photos de l'anniversaire de votre fille, une petite musique, et Animoto réalise un montage automatiquement. Jusqu'ici tout va bien. Animoto utilise 80 serveurs sur Amazon EC2. Un lundi matin, un développeur a l'idée de coder un gadget pour FaceBook. Ce gadget a pour effet de multiplier les visites du site par 10 en quelques heures. 2 jours plus tard, le moteur d'Amazon est sollicité et les administrateurs d'Animoto activent alors 3500 instances d'EC2... Pensez-vous honnêtement que cela serait possible en vrai dans un datacenter ? en 2 JOURS !!! Pensez à un site qui vend des skis et des snowboards. En février, 50 serveurs. En juillet, 1 seul serveur... Amazon ne vous fait payer que ce que vous utilisez. De même qu'en hiver la facture d'électricité est plus élevée, ce site de ski a amélioré sa rentabilité avec peu d'efforts. Enfin le service EC2 a été lancé en Europe il y a quelques jours, et donc il n'y a plus de raisons de ne pas le tester. Il vous faut une carte bleu et c'est tout. Dans le future, les gens d'Amazon souhaitent améliorer les fonctions de monitoring, de répartition de charges et de montées en charge automatiquement. **Elastic Grid** Jérôme Bernard prend ensuite la parole afin de présenter Elastic Grid, projet open-source qui permet d'ajouter de la flexibilité intelligente au moteur d'Amazon. Tout d'abord, si vous commencez par vous même à déployer votre serveur avec votre application, il faut savoir que de bonnes connaissances Unix seront nécessaires. Il faut créer une image Amazon (AMI) qui contient le système d'exploitation et votre serveur installé. C'est ce principe d'image (similaire à une ISO) qui permet ensuite à Amazon EC2 de monter en charge rapidement et facilement. Jérôme commence par faire la démonstration d'[Elasticfox](#) un plugin pour firefox qui permet de piloter ses instances sur Amazon directement via son navigateur. Je vois qu'en quelques mois la communauté a été très active et qu'en effet ce plugin permet de voir son système démarrer, comme il l'a montré durant la démonstration. L'image EC2 AMI est donc un bootstrap. Une limitation d'Amazon EC2 est l'absence de multi-casting. Comme Jérôme l'explique, cela peut poser des problèmes pour les applications réparties basées sur des solutions comme JGroups, bien que JGroups soit aussi capable de faire du TCP (et pas que de l'UDP). Il faut ensuite faire attention que le système de fichier par défaut ne doit pas être considéré comme non résilient, et qu'à tout moment nous pouvons perdre nos fichiers. Pour cette raison, il faut associer l'espace de stockage Amazon S3 à une instance EC2 pour que les données qui doivent être persistées soient stockées dans le cloud. La solution Elastic Grid qu'il présente est donc un moteur qui améliore la productivité du développeur en fournissant des services de provisioning, de déploiement et de développement pour le développeur. Sans la solution Elastic Grid, ce sera à vous de créer votre image AMI à chaque fois, de vous débrouiller pour déployer et gérer votre image. La solution Elastic Grid améliore la productivité en utilisant un moteur unique basé sur Rino et JNI afin de piloter les instances EC2. Il est ainsi capable de redémarrer automatiquement une application qui est tombée, de doubler le nombre de machines afin de répondre à un pic de demande ou au contraire, de réduire le nombre de machine afin d'économiser de l'argent. Voici comment je vois ce concept : je rentre chez moi, les lumières s'allument automatiquement, la télé et la musique aussi. Je consomme alors 5Kw. Ensuite rentre mon épouse et les enfants. Le système le détecte et active alors les lumières des chambres, le chauffage de la salle-de bain ect. Finalement ma maison s'adapte au nombre d'occupant. Arrive la nuit, tout le monde dort, le système se charge alors de réduire la consommation d'électricité. L'un des concepts d'Elastic Grid est donc ce provisioning intelligent basé sur l'utilisation des ressources. De facto, la solution donne un niveau de service certifié (SLA) et permet de répondre aux mauvaises surprises sans intervention humaine. Jérôme termine par une démonstration basée sur Spring dmServer. Tout d'abord définir où se trouve l'image de dmServer, l'application à déployer et Elastic Grid se charge

de tout. Pour terminer Jérôme parle de la notion de CloudBursting. Ce principe serait d'offrir la possibilité temporaire d'augmenter la capacité machine de votre infrastructure en s'appuyant sur EC2 et S3. Il faudrait qu'il nous explique un peu plus cette idée car c'est intéressant. Bref j'ai bien aimé cette présentation qui m'a donné une nouvelle vision pour l'avenir sur l'hébergement des applications. Site d'Elastic Grid: <http://www.elastic-grid.com>

## Devoxx : que contiendra Java 7 ?

Un journaliste a fait la remarque : les éditeurs font maintenant des annonces à Devoxx, car le lieu est l'endroit idéal pour lancer un produit ou un nouveau service. Tout d'abord c'est l'annonce officielle de la sortie de JavaFX 1.0. De nombreuses conférences sur ce thème, une BOF, bref il y a eu un effet d'annonce très bien calibré. Ce qui m'a le plus intéressé, c'est les surprises de Java 7. Finalement la prochaine version de Java ne sortira pas avant début 2010, ce qui me fait penser que SUN et la communauté veulent prendre le temps nécessaire pour sortir une version vraiment innovante. Lors de la Keynote du jeudi matin, Mark Reinold a un peu changé le programme de la présentation, car au lieu de "Java SE 7 updated", sa présentation a commencé par "Modularity in Java SE". Avant tout, avis de décès : la JSR-277 (Java Modules) est morte et enterrée. Cela a bien fait rire tout le monde lors de la présentation des JavaPosse l'après-midi. On parle maintenant de la JSR-294. [Son blog](#) (avec un look un peu soviétique) vous donnera l'actualité de ce qu'il se passe... car il y a un côté un peu people qui ne m'intéresse pas tellement. De son point de vue, OSGi n'est pas suffisant pour répondre aux soucis d'obésité de la JVM. L'objectif est d'offrir un système plus modulaire pour que le kernel de base soit plus léger. Cela s'appliquera à la JVM, au JRE et au SDK. Le débat de fond reste de savoir si cette modularité doit faire partie de Java 7 de facto ou pas... Concernant les nouveautés du JDK 7, Mark a parlé de beaucoup de JSR dont voici la liste (sans doute incomplète car je n'ai pas tout eu le temps de récupérer) : - JSR 292 pour le support des langages Dynamiques - JSR 203 NIO2 (entrees / sorties nouvelle implémentation) - JSR 296 Swing Application framework ([voir cet article](#)) - JSR 308 vise à [étendre les endroits où nous pourrions utiliser les Annotations](#) en Java... - JSR 166 package Concurrency avec de nouvelles Fork pour ce package de Doug Lea (Fork/Join, Phasers, LinkedTransferQueue, ConcurrentReferenceHashMap, et Fences) - Smart Rethrow et surtout MultiCatchException, petite astuce qui permettra d'éviter les multiples catch(x) catch(y) et catch(z) qui n'améliorent pas la lisibilité du code. - JSR 295 Beans binding (?) Le plus fun sera d'apprendre que **il n'y aura pas de closures en Java 7** ce qui a bien fait rire tout le monde (JSR 295). Après tout ce temps perdu sur les blogs, on a pas vraiment compris ensuite ce qu'ils comptent faire... Bref longue vie à Groovy qui a déjà ce système depuis sa création. Plus sérieusement, cela veut peut-être dire que les closures... en s'en fiche en Java. Certes c'est élégant, mais moi ma bonne grosse application de gestion... elle s'en fiche. Au final ce qui faut retenir : l'objectif de Java 7 sera d'offrir un système modulaire afin que selon le type d'application, le nombre de classes et le temps de démarrage soit encore plus léger que Java 6. Il y a aura peu de révolutions du côté de la syntaxe, plus des chantiers comme un support des langages dynamiques. Tout ceci devrait occuper SUN et la communauté l'an prochain. **Références complémentaires:** <http://java.dzone.com/articles/java-7-update-mark-reinhold-de> <http://blogs.sun.com/darcy/> [Blog de Mark Reinhold](#)

## Devoxx : les Freelance et le ralentissement de l'activité économique

Chaque jour l'équipe Devoxx imprime et diffuse un petit journal papier, qui est super bien fait. Dans la dernière édition de vendredi ([disponible ici](#)) un des articles a retenu mon attention : Do Java Professionals feel the downturn of the Economy? Pour cela, Parleys a interviewé Roel Van der Paal, un freelance et Peter Voorneveld, Sales Manager en Belgique chez Computer Futures, l'un des plus gros cabinet de recrutement anglais qui ne travaille qu'avec des indépendants (dont moi-même soit dit en passant). Peter Voorneveld explique que l'activité de Computer Futures est d'être un broker (un agent) entre les indépendants et les employeurs. En Belgique, leur base dispose de plus de 13263 candidats, dont 60% de spécialistes Java. Sur cet ensemble, il y a en fait 3600 freelance Java (des personnes actives, structurées en entreprise indépendante, en portage ou en EURL). D'après ce qu'il a constaté depuis la rentrée, le marché se tend. Les entreprises préparent les budgets de 2009 et des efforts sont demandés aux indépendants. Les freelances sont appelés en renfort lorsque les entreprises ne peuvent pas facilement augmenter le nombre de poste de permanent, ou parce que les temps de recrutement sont trop longs. Le consultant qui est interviewé dans l'article pense qu'en raison de la crise, les entreprises feront appel plus facilement aux freelances, ceux-ci sont facile à trouver, souvent très expérimentés, et facile à remercier lorsque le projet ou le budget est finalement abandonné. Ce que je comprends c'est qu'entre faire une création de poste pour un projet ou prendre un freelance, une entreprise préfère faire appel à un freelance. Si le budget n'est plus suffisant, le coût pour se séparer du freelance est nul là où il est élevé s'il s'agit d'un permanent. Dans l'interview on apprend qu'un nombre important de contractor viennent d'Angleterre pour travailler en Belgique, là où le fait de ne parler qu'anglais n'est pas un handicap pour travailler. En France au contraire, il est plus difficile pour un anglophone de trouver une mission... Cela m'avait été confirmé par mes anciens collègues de Reuters qui sont indépendants (Malcolm, Paul, Peter...). Ils travaillent maintenant à Bruxelles et à Londres car à Paris, les employeurs veulent des personnes qui parlent... français. C'est un peu surréaliste. Les candidats qui viennent d'Angleterre cassent les prix. Ils sont de 50 à 75% moins cher que les candidats belges. Pour cette raison, demander un taux journalier trop élevé est devenu difficilement justifiable pour des candidats un peu juste. Enfin pour terminer, Voorneveld pense que certaines SSII seront obligées de se séparer de certains collaborateurs. Ceux-ci tenteront alors leur chance en tant qu'indépendant, ce qui devrait alléger la tension sur le marché. Plus d'indépendants égal des prix plus réalistes pour les entreprises. D'après lui, les professionnels expérimentés (> 5 années d'expérience) n'ont aucuns soucis pour trouver des missions, car la demande reste très forte. J'en

profite aussi pour vous donner un site peut-être pas très à jour mais qui recense les moyennes de tarif journalier enregistrées par les contributeurs du site : [Tarifs Freelance](#). Attention, il n'y a pas de Java, c'est une vue synthétique du marché.

## BarCamp mardi 16 décembre à Paris

Rappel pour les parisiens : mardi 16 décembre rendez-vous chez OCTO sur les Champs-Élysées à côté du Virgin (station Franklin Roosevelt) pour participer au deuxième BarCamp. Le site [JavaCampParis](#) vous donne les détails pour venir, ainsi que le formulaire d'inscription. Le petit souci c'est qu'il y a déjà 50 personnes inscrites et 12 personnes en attente... Il y a clairement une grosse envie de se retrouver entre geeks ! A mardi soir

## Devoxx : bilan côté organisation et ambiance

Encore un billet sur Devoxx... comme c'est original. Tout d'abord l'organisation de Devoxx était parfaite. Rien à redire. Nous avons un peu critiqué le Wifi, mais durant les conférences le réseau marchait très bien. Du côté des exposants, des repas, du confort des salles, de la qualité du son et de la vidéo, tout était parfait... A comparer avec les soucis cette semaine de l'événement [LeWeb](#) qui ont énervés [certains blogueurs](#). Du côté des présentations, de l'excellent et du très mauvais. Le site [Parleys.com](#) reprendra bientôt toutes les présentations et il sera donc possible de voir ou de revoir les meilleurs présentations. De l'avis de tous, avec le rythme des journées, le soir tout le monde est cuit. Le Metropolis est situé un peu loin du centre ville, mais à part cela, rien à dire. Du côté de l'ambiance, j'ai trouvé les présentations assez froides, voir studieuses. Certains présentateurs ont fait des efforts pour faire rire un peu l'assemblée, mais à chaque fois la salle était plutôt frigide. A part la présentation des JavaPosse qui a retourné la salle de 400 personnes, j'ai trouvé le ton un peu trop "geek" et peut-être pas assez décontracté. Je dis pas qu'il faut mettre l'an prochain des filles en mini jupes et de la bière partout, mais parfois c'était pesant. Sur le fond des présentations, globalement peu de blabla marketing, les présentations parlent souvent aux développeurs, et nous n'avons pas eu ce ton "show-off" que l'on voit à JavaOne. Il y a clairement une différence de présentation et de ton. Là où JavaOne est un salon massivement piloté par SUN, Devoxx a un ton plus indépendant, plus root mais plus véritable, plus intéressant et plus riche. Pour conclure, c'était un très bon événement, l'occasion de rencontrer du monde, de prendre la température de la communauté et de se rendre compte des tendances pour 2009.

## Java BarCamp 2 : merci OCTO



Très bonne soirée ce soir, lors du 2ème Java BarCamp. Avec une bonne cinquantaine de personnes, l'événement organisé chez [OCTO Technology](#) par Luc Bizeul était très sympa. C'est avant tout l'occasion de croiser des têtes connues, mais aussi l'occasion de discuter de sujets de manière plus approfondie par rapport au Paris JUG.





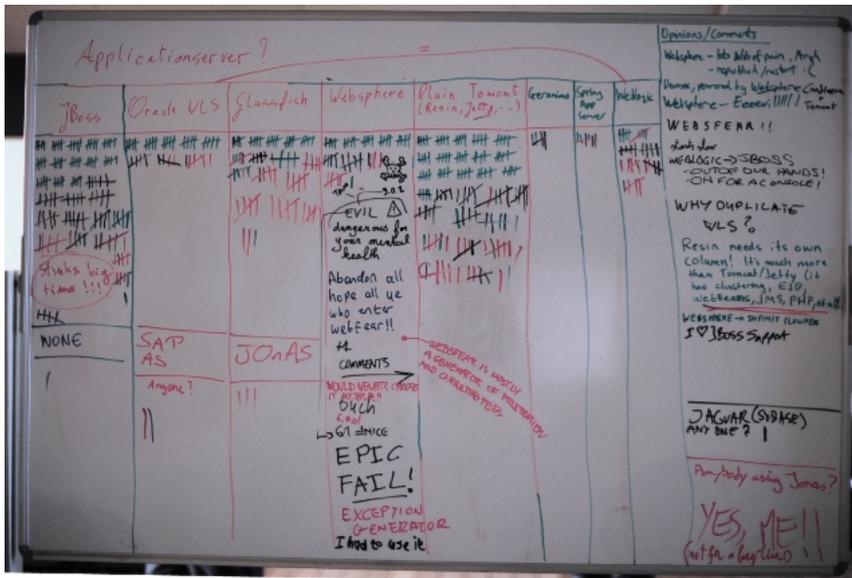
Dans un premier temps, chacun a inscrit sur un post-it un sujet dont il souhaite discuter. Ensuite après avoir placé l'ensemble des post-its sur un tableau, Luc a réparti les forums. Je me suis donc retrouvé (avec plaisir) pour parler de Devvxx et de JAX-RS. Je dois d'ailleurs terminer un billet sur cette présentation et ce soir c'était l'occasion pour moi de rejouer en direct devant une vingtaine de personnes ce que j'ai entendu lors de cette conférence. Avec Fabrice Robini d'OCTO, nous avons donc parlé tout d'abord de notre expérience sur Devvxx. Ensuite nous avons improvisé sans filet une présentation sur REST et sur JAX-RS, ce qui a donné un exercice un peu acrobatique, mais très sympa. A nous deux je pense que nous avons restitué aux personnes présentes notre vue sur REST et sur JAX-RS. J'ai eu l'impression de faire un article pour le Touilleur... en direct. Très fun. Dans un deuxième temps j'ai assisté aux discussions autour de Spring Integration et Mule. Grâce à Julien Dubois, nous avons eu une présentation de Spring Batch et de Spring Integration. Les personnes présentes ont contribué à la discussion ([florent ramière](#), [sébastien letétié](#)) et je serai vous, j'irai voir le compte-rendu d'Olivier sur le blog "[The Coder's breakfast](#)" car je pense qu'il pourra vous donner une synthèse complète sur cette présentation. Enfin grâce à l'organisation parfaite d'OCTO Technologies, nous sommes restés ensuite autour du buffet. J'ai discuté longuement avec Sebastien et [David Gageot](#), étant content de partager son opinion sur le Mac et sur le fait que passé un certain âge, on doit en avoir assez de bidouiller avec linux. Il nous a fait ensuite un démonstration du GPhone, fraîchement arrivé des USA pour la modique somme de 450 EUR. Durant la soirée j'ai aussi parlé politique économique avec luc, eric mignot et david, ce qui permet de s'aérer le cerveau et de déconnecter un petit peu de son monde de geek. Le Barcamp était encore un fois l'occasion de rencontrer du monde, de discuter avec des personnes autour d'un bon verre de vin (l'abus de santé est dangereux pour l'alcool) et donc de partager sur différents sujets de la communauté Java. Merci à OCTO de nous avoir reçu ce soir.

## REST, JavaRebel et nouvelles

Steve Jobs ne participera pas au prochain MacWorld en janvier prochain. C'est traditionnellement lors de ce genre d'événement que Steve Jobs annonce la sortie de nouveaux produits. Suite à la discussion d'hier au BarCamp sur REST, Duncan Cragg parle sur son blog dans "[REST, the business conversation](#)" des concepts de contractualisation pour faciliter les échanges de messages. Ces articles sont des discussions virtuelles avec un architecte d'eBay. Cette idée originale permet de parler d'architecture et de présenter les concepts phares de REST. [Le premier article](#) de sa série attaque de manière très simple la présentation et l'utilité d'une architecture de services REST. A lire. Si vous cherchez aussi des patterns et des anti-patterns autour de REST, une présentation effectuée à Devvxx est [en ligne](#). [JavaRebel](#) que j'ai vu à Devvxx propose un système de rechargement du byte-code Java à chaud avec la JVM. Le principe se base sur la déclaration d'un agent au moment du démarrage de la JVM. Lorsque vous recompilez par la suite votre code, cet agent recharge le code compilé sans qu'il ne soit nécessaire de relancer la JVM. J'ai fait un test ce matin avec Weblogic 9.2 et en effet, le système fonctionne plutôt pas mal. A tester si vous en avez marre de relancer votre serveur d'app... et aussi parce que tout le monde n'a pas encore Spring dm Server sous la main :-). A la conférence Devvxx, les organisateurs placent des tableaux blancs dans le hall central. Chacun de nous peut ensuite voter, participer à des sondages sur des thèmes sérieux et des thèmes loufoques. Sur son blog, [Stephen Colebourne](#) a synthétisé ces résultats et cela nous donne une idée de la tendance des attentes des gens pour Java 7 par exemple. On apprend ainsi que 49% des votants en dehors de Java s'intéresse à Groovy, ou encore que la très grosse majorité des développeurs travaillent maintenant avec Java 5 et 6. Je dois enfin rédiger et mettre en ligne sur le touilleur les slides de mon quickie de Devvxx sur les ESB appliqués au monde de la finance. Je vous fais cela la semaine prochaine.

## Devvxx: quel serveur d'application utilisez-vous ?

[caption id="attachment\_622" align="alignnone" width="640" caption="Which application server you use?"]



[caption] Petit sondage sur l'un des tableaux blancs installés au premier étage. Avec l'appareil photo de l'iphone mes photos étaient toutes plutôt ratées. J'ai trouvé sur flickr une image d'un des tableaux qui m'a valu une bonne discussion là-bas lorsque j'étais devant avec d'autres développeurs. Voici le nombre de résultat par serveur d'application:

- JBoss Application Server : 136 voix
- Oracle WLS + BEA Weblogic : 75 voix
- GlassFish : 78 voix
- IBM WebSphere : 46 voix (j'adore "web fear")
- Plain Tomcat, Jetty, Resin : 148 voix
- Geronimo : 5 voix
- Spring App Server : 7 voix
- SAP Application Server : 2 voix
- JONAS : 3 voix

Bref avec Spring dm Server qui arrive sur ce marché, je pense que nous allons voir l'an prochain des batailles de discussions sur "mon mien il est meilleur que ton tiens" comme il y a quelques années... Cela promet des heures de discussions sur les blogs.

## La méthode RESTO, une alternative à Scrum

Scrum est une méthode Agile de développement fortement anglo-saxonne. Par certains aspects, je me demande si les processus de Scrum sont adaptés à notre culture. Dans cet article, je déconnecte complètement votre subjectivité en vous proposant une nouvelle méthode virtuelle : la méthode Resto. Alors à table, il est temps de changer votre vision sur Scrum. Je me pose pas mal de questions sur Scrum. Une grosse partie de la méthode Scrum a révolutionné ma vision sur le développement informatique. A un point tel que j'ai vraiment du mal avec les méthodes classiques de gestion de projet. D'un autre côté la maîtrise de Scrum demande des efforts qui vont au delà d'une simple formation. Après quelques mois et un peu plus de recul, je commence à croire qu'il faudrait adapter et revoir culturellement la méthode avant de l'appliquer en France telle quelle. On peut reprocher à Scrum son cérémonial rigide. En effet malgré le peu de cérémonies, celles-ci sont des rendez-vous importants pour que le processus fonctionne. La cérémonie qui fonctionnait le moins bien pour nous (chez Reuters) était la Sprint Review Meeting. Je demandais à l'équipe de lister les points qui s'étaient bien déroulés et les points à améliorer. Et finalement tout tournait autour de Scrum. Je sais que Scrum met en avant les points délicats dans la gestion d'un projet, que c'est l'un de ses effets. Cependant l'équipe n'arrivait pas à passer à l'étape suivant et à voir un intérêt à ces événements. Alors réfléchissons à une autre méthode, la méthode RESTO (copyright Le Touilleur Express). Les bases de cette méthode : cycle itératif court et limité dans le temps, une liste de fonctionnalités à implémenter, aucunes cérémonies, la capacité pour chaque développeur de prendre n'importe quand n'importe quel élément d'un tableau. Techniquement il vous faut des post-it, des marqueurs de couleur, un "paper board" et un peu d'organisation. Ensuite j'y ajoute des rôles tournants afin de pouvoir palier aux soucis venant de l'extérieur. Il me faudra un barman, des cuisiniers et des serveuses. A noter que si votre équipe est trop petite, vous pouvez cumuler les rôles. **Faire le Menu de votre RESTO** Tout d'abord la première étape est de regrouper dans un fichier Excel toutes les demandes de vos clients, les bugs et les demandes de changement. Pour chaque élément, vous allez identifier par une phrase comment le client testera la fonction. Par exemple : "j'ouvre mon navigateur, je me connecte sur l'adresse de la prod, je vois une page d'authentification. Je mets mon login et mon mot de passe et je peux alors entrer dans l'application. Sur la page d'accueil je peux cocher une fonction 'Se souvenir de moi' et c'est tout". Ce fichier doit être mis à jour en permanence selon les demandes des clients. C'est ce que j'appelle le Menu, et c'est ni plus ni moins un Product Backlog. Les Serveurs auront pour mission durant la durée de la Mission de mettre à jour le contenu de ce fichier en y insérant les demandes des Clients. Les Cuisiniers seront chargés de donner le Prix à payer pour chaque élément, en respectant mes explications sur ce qu'est [une Estimation](#) dans le cadre du développement d'un logiciel. **Un peu de réunionite** Ensuite je verrai bien une première réunion à laquelle j'inviterai un client avec l'équipe. L'idée est qu'avant de commencer à développer, le client est présent afin d'expliquer en détail et de clarifier les points qui seraient mal expliqués. Appelons

cela la Commande. Une fois la Commande passée aux Serveurs, le Client doit attendre tranquillement que son plat soit préparé. **Le Barman : mister Cocktail** Le Barman est là pour le faire patienter. Il peut venir l'aider ponctuellement afin de l'aider à corriger un problème urgent. C'est la personne en charge de la production en quelques sortes. Il est disponible pour travailler avec le Client pendant que le reste de l'équipe bosse. L'idée par rapport à Scrum serait d'avoir une personne disponible et allouée afin de traiter les problèmes du Client en dehors du cycle de production. En fait le Barman se charge de faire boire le client pour le faire patienter. Je piquie cette idée à ce que nous faisons actuellement sur la plateforme de Prime Brokerage : l'un de nous est en charge de la production pour 5 jours et durant cette durée, ne participe pas aux développements de l'équipe. **La batterie de Cuisine** Les Cuisiniers ne voient pas le client, ce qui est peut-être une faiblesse par rapport à Scrum. D'un autre côté ils peuvent travailler en permanence sans devoir être présent à des réunions, ou être obligé de participer à des cérémonies comme pour Scrum. J'imagine ainsi accélérer la productivité en n'impliquant qu'une partie de l'équipe dans la relation client <-> équipe. **Il est important de ne rien faire une fois par semaine** Enfin j'y ajoute un concept qui vient de mon expérience chez Reuters : le vendredi c'est ravioli. Derrière cette phrase magique, l'idée est de dire à toute l'équipe qu'une journée de la semaine est consacrée à faire autre chose que le boulot habituel. Durant cette journée, chacun fait ce qu'il veut tant que cela reste en rapport avec son travail. Je reprends une idée appliquée chez Google. Aurélien mon petit-frère a travaillé 2 ans chez Google en Irlande. Il m'expliquait ainsi qu'une partie de leur évaluation et donc de leur bonus est calculé sur leurs initiatives pour la communauté. Mon frère a ainsi monté un groupe de rock/folk et il a participé à une œuvre caritative, en plus de son travail, ce qui lui a valu des points en retour par son management. Évidemment cela ne rapporte rien, si ce n'est un esprit, une ambiance et quelque chose que les autres compagnies n'ont pas. Je pense que la personne qui a l'idée d'un nouvel outil pour améliorer la productivité, ou qui a l'idée d'apporter un panier de basket pour que l'équipe se détende, devrait être récompensé. Attention je ne vois pas un système anglo-saxon basé sur la performance, plutôt un système bien français basé sur la bon humeur. Nous pourrions imaginer qu'à la fin de chaque Mission, tout le monde donne un point à une personne de son choix. Celui qui a le plus de point est alors proclamé Grand Manitou, et il peut alors affecter une tâche rébarbative dont il a la charge à un de ses collègues. Le Grand Manitou a le droit de refuser une Commande d'un client pour la prochaine Mission. A chaque fin de Mission on élit alors un **nouveau Grand Manitou** et ainsi de suite. Oui cela vous choque... pourtant j'ai entendu cette semaine le cas d'un grand cabinet de consultant qui a offert un iPhone à deux de ses collaborateurs les plus méritants, qui, par leurs contributions sur le blog de l'entreprise, ont contribué à améliorer la visibilité et la réputation de cette entreprise. Prenons ensuite une spécificité franco-française que nos voisins Belges ou Suisses nous envieient (peut-être) c'est la fin d'une mission. Après 2 semaines de développement, le dernier vendredi de ces 2 semaines sera une journée Ravioli, avec l'obligation de fêter la fin de la Mission. Pour cela vous pouvez apporter des croissants et faire un petit-déjeuner, vous pouvez faire un restaurant le midi ou même organiser un apéritif après le travail dans un bar à côté de votre travail... L'objectif est de FETER la fin de la Mission. Dans le bâtiment lorsqu'une équipe termine un chantier, les ouvriers, l'architecte et les conducteurs de travaux se retrouvent afin de faire un grand repas pour fêter la fin du chantier. Je pense que nous devrions faire la même chose et formaliser cela dans la **méthode RESTO**. Voilà un billet bien décalé qui nous sort de la morosité ambiante, qui nous permet d'envisager un processus adapté à nos habitudes, et qui je l'espère, vous donnera de l'inspiration pour vous épanouir et vous faire plaisir en travaillant. Je sais qu'au final derrière ce que je viens d'expliquer, ça sent le Scrum. Bonnes fêtes.

## Exemple Mule : lire un fichier CSV et écrire vers une base de données

Mule est un outil léger d'intégration qui permet de connecter différents systèmes. Par rapport à d'autres ESB plus complexes, Mule est un système rapide à mettre en place pour répondre par exemple aux cas d'usage suivants : lire un fichier CSV et l'importer vers une base de données. rerouter un appel d'un web service vers un envoi d'email, lire des fichiers plats puis les envoyer via FTP, lire le contenu d'une page via HTTP pour ensuite envoyer un email... Comme on le voit, Mule est donc avant tout un système d'interconnexion simple et conçu pour faciliter l'intégration de données dans une application Java. La version Entreprise de Mule propose un exemple complet d'utilisation du connecteur JDBC. Le code source de cet exemple n'étant pas distribuable ni libre de droits, je vous propose dans cet article un exemple simple dans lequel je vais lire un fichier texte de type CSV, le transformer puis ensuite l'importer dans une base HSQLDB. Cet exemple vous montre comment écrire un transformeur, déclarer le connecteur JDBC de Mule puis enfin mettre en oeuvre le tout.

### Installation de Mule

Téléchargez la version de Mule 2.1.2 via [le site de Mule](#) Décompressez dans un répertoire l'archive Déclarez une variable d'environnement MULE\_HOME Mettez à jour le script startMule.sh donné avec l'archive à la fin de cette article pour pointer vers le bon répertoire de Mule.

### Mise en place de la base de données

Dans un premier temps, si vous souhaitez tester par vous même, téléchargez l'archive suivante "mule-poc.tar.gz". Je vous ai préparé une arborescence avec l'ensemble du code source. Le répertoire database contient de quoi créer une base HSQLDB en mémoire mais autonome, car déclarée sous la forme serveur. HSQLDB nécessite un fichier "sqltools.rc" pour que le mode serveur autorise des connexions sur le serveur. Pour cela, copiez le fichier sqltools.rc vers votre répertoire \$HOME (ou C:\Document and Settings\ ) Le script *startDBServer.sh* assume que Java 5

ou 6 se trouve dans le Path, et démarre le serveur HSQLDB. N'étant pas un expert HSQLDB, j'ai fait au plus simple. Le script createDatabase.sql sera exécuté via le script shell "createDatabase.sh" afin d'importer le contenu du fichier sample.dsv.

## Structure simple de la base

L'unique table de la base est vraiment ce qu'il y a de plus simple, il n'y a pas d'index ni de clé primaire. Simplement des types Date, boolean, int et Varchar afin de montrer les capacités de transtypage de Mule. CREATE TABLE sampletable(id INT, lastUpdate DATE NOT NULL, msg VARCHAR, isValid BOOLEAN);

## Le fichier de configuration de Mule

Le fichier de configuration de Mule est le coeur du système. Basé sur Spring, Mule partage des concepts de Spring 2.5 qui en font un système très facile à apprendre. Contrairement à une solution basée sur Spring Integration, vous verrez qu'ici je ne vais écrire qu'une seule classe Java. J'aimerais refaire le même exemple avec Spring Integration plus tard afin de comparer. L'entête du fichier déclare les namespaces des connecteurs file, stdio, vm et jdbc ainsi que les entêtes pour Spring.

```
<?xml version="1.0" encoding="UTF-8"?>
<mule xmlns="http://www.mulesource.org/schema/mule/core/2.1"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:spring="http://www.springframework.org/schema/beans"
      xmlns:context="http://www.springframework.org/schema/context"
      xmlns:file="http://www.mulesource.org/schema/mule/file/2.1"
      xmlns:stdio="http://www.mulesource.org/schema/mule/stdio/2.1"
      xmlns:vm="http://www.mulesource.org/schema/mule/vm/2.1"
      xmlns:jdbc="http://www.mulesource.org/schema/mule/jdbc/2.1"
      xsi:schemaLocation="
        http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
        http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-2.5.xsd
        http://www.mulesource.org/schema/mule/core/2.1 http://www.mulesource.org/schema/mule/core/2.1/mule.xsd
        http://www.mulesource.org/schema/mule/file/2.1 http://www.mulesource.org/schema/mule/file/2.1/mule-file.xsd
        http://www.mulesource.org/schema/mule/vm/2.1 http://www.mulesource.org/schema/mule/vm/2.1/mule-vm.xsd
        http://www.mulesource.org/schema/mule/stdio/2.1 http://www.mulesource.org/schema/mule/stdio/2.1/mule-stdio.xsd
        http://www.mulesource.org/schema/mule/file/2.1 http://www.mulesource.org/schema/mule/file/2.1/mule-file.xsd
        http://www.mulesource.org/schema/mule/jdbc/2.1 http://www.mulesource.org/schema/mule/jdbc/2.1/mule-jdbc.xsd">
```

Ensuite je déclare simplement un DriverManagerDataSource en utilisant la base HSQLDB démarrée sur mon serveur. Le login par défaut est 'sa' et il n'y a pas de mot de passe.

```
<spring:bean id="toueurDB"
  class="org.springframework.jdbc.datasource.DriverManagerDataSource">
  <spring:property name="driverClassName"
    value="org.hsqldb.jdbcDriver"/>
  <spring:property name="url"
    value="jdbc:hsqldb:hsq://localhost"/>
  <spring:property name="username" value="sa"/>
  <spring:property name="password" value=""/>
</spring:bean>
```

Je déclare ensuite un connecteur mule de type JDBC avec pour datasource-ref, mon bean "toueurDB". Dans ce connecteur vous pouvez déclarer 3 types de requêtes, je ne déclare ici qu'une requête d'insertion appelée "writeTest". Cette requête va recevoir une List de Map. Chaque élément de la List correspond à une ligne de fichier. Chaque élément de la ligne séparé par des virgules, sera enregistré dans une Map. J'y reviens plus loin. La lecture des arguments utilise ici map-payload, l'une des techniques de Mule pour récupérer des couples clés-valeurs facilement. Une fois cette requête "writeTest" déclarée, elle est alors disponible pour mon service Mule

```
<jdbc:connector name="myJdbcConnector"
  dataSource-ref="toueurDB"
  pollingFrequency="1000">
  <jdbc:query key="writeTest"
    value="INSERT INTO SAMPLETABLE (ID, lastUpdate, msg, isValid) VALUES ([map-payload:id], [map-payload:lastUpdated], #>
</jdbc:connector>
```

Je déclare ensuite 1 transformeur :

```
<custom-transformer name="StringToListOfMap" class="com.innoteria.mule.simple.StringToListOfMap"/>
```

Le code de la classe StringToListOfMap n'est pas très élégant mais l'intérêt ici est de vous expliquer le principe, pas de vous donner un cours de Java. Mule lit un fichier texte simple et par défaut transforme son contenu en une String. A noter que ce comportement est bien entendu désactivable mais ici il me rend service car mon transformeur reçoit directement une String contenant tout le fichier. Je vais ensuite itérer et créer une Map clé-valeur correspondant aux paramètres de ma requête "writeTest" précédemment déclarée. Enfin je mets chacune de mes Map dans une List pour ensuite utiliser l'encoding transformer **map-payload**

```
public class StringToListOfMap extends AbstractTransformer {
  @Override
  protected Object doTransform(Object src, String encoding) throws TransformerException {
    String payload = (String) src;

    if (payload == null) return null;
    StringTokenizer stLine = new StringTokenizer(payload, "\n");
```

```

String line;
Map m;
List resultList=new ArrayList();
StringTokenizer st;
while (stLine.hasMoreTokens()) {
    line=stLine.nextToken();
    st=new StringTokenizer(line,",");
    m = new HashMap();
    m.put("id", st.nextToken());
    m.put("lastUpdated", st.nextToken());
    m.put("msg", st.nextToken());
    m.put("valid", st.nextToken());
    resultList.add(m);
}
return resultList;
}
}
}

```

Pour terminer la configuration de mon service Mule, je déclare ensuite un connecteur de type **file**. L'instance de Mule va scruter le répertoire out/source. Dès qu'un fichier y sera déposé, il sera lu et transformé ensuite via notre transformer "StringToListOfMap" dont nous venons de voir le code.

```

<model name="fromFileToDatabase">
  <service name="myImportService">
    <inbound>
      <file:inbound-endpoint path="/Users/nicolas/Dev/mule-poc/out/source"
        pollingFrequency="3000" fileAge="5000"
        moveToDirectory="/Users/nicolas/Dev/mule-poc/out/destination"
        transformer-refs="StringToListOfMap">
    </file:inbound-endpoint>
  </inbound>
  ...
  ...

```

Je n'utilise aucuns composants particuliers. Il est temps de déclarer maintenant un outbound endpoint de type JDBC, utilisant notre query "writeTest". La seule chose un peu spéciale est que j'utilise ici un list-message-splitter-router. Celui-ci prend en entrée une List d'élément et va automatiquement itérer cette List pour passer ensuite chacun des éléments au "jdbc:outbound-endpoint" en l'occurrence une Map. En cas d'exceptions je route les messages vers la sortie standard. Je pourrai aussi sauver vers un fichier les entrées invalides de mon fichier CSV.

```

<echo-component/>
<outbound>
  <list-message-splitter-router>
    <jdbc:outbound-endpoint queryKey="writeTest">
      <payload-type-filter expectedType="java.util.Map"/>
    </jdbc:outbound-endpoint>

    <payload-type-filter expectedType="java.util.List"/>
  </list-message-splitter-router>
</outbound>

<default-connector-exception-strategy>
  <outbound-endpoint address="stdio://OUT"/>
</default-connector-exception-strategy>

</service>
</model>
</mule>

```

Une fois le tout compilé, je démarre mon instance de Mule via le script "startMuleServer.sh" en l'exécutant dans le répertoire bin. A l'exécution, si je dépose le fichier "testDatabase.txt" dans le répertoire out/source, Mule le charge et le message suivant s'affiche dans la console : INFO 2008-12-23 15:24:48,927 [myImportService.2] org.mule.component.simple.LogComponent: \*\*\*\*\*  
 \* Message received in service: myImportService. Content is: '0,2008-12-12,[Un \*  
 \* exemple de message tr?s simple],true \* \* 1,2008-11-12,[Un autre exemple pour  
 tester],fa...[100 of 141]' \*  
 \*\*\*\*\*

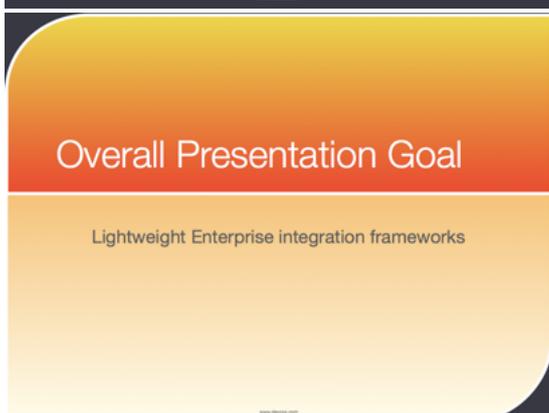
Un select dans la table retourne les 3 lignes du fichier testDatabase.txt comme attendu **Conclusion**  
 Dans cet exemple nous avons vu comment lire un fichier texte simple, comment transformer son contenu pour ensuite insérer directement des entrées dans une base de données. Mule Enterprise Edition propose des connecteurs avancés pour les traitements par Batch, plus efficace que l'import utilisé ici. Par ailleurs il est aussi possible de configurer les pools de connections en se basant sur les méthodes Spring standards. Mule est une solution qui évite l'écriture de code d'infrastructure, par essence du code souvent sans valeur ajouté. Cependant comme on peut le remarquer ici, il y a rapidement un besoin d'ajouter un peu de logique, ne serait-ce que pour la validation. Et cette logique s'écrit avec du code Java. J'ai aussi en stock un exemple avec Apache Camel, préparé pour Devvxx, que je vous proposerai prochainement dans un article sur le Touilleur. **Code source de l'exemple** : [mule-poc-touilleur.tar.gz](http://mule-poc-touilleur.tar.gz)

## Devvxx : ESB in Financial Application

Cette année j'ai eu l'occasion de présenter un sujet sous la forme d'un Quickie à [Devvxx](http://Devvxx). Il s'agit d'une mini-présentation limitée à 15 minutes, à l'heure du déjeuner. Au final j'ai bien aimé cette première expérience, en anglais et devant pas mal de monde mine de rien. Voici la présentation en français. **1. ESB in Financial Application**



## 2. Objectif de la présentation



En quinze minutes, l'idée de cette présentation est d'expliquer les besoins rencontrés dans le monde de la finance en terme d'architecture. Où en sommes-nous aujourd'hui et quels outils pouvons-nous utiliser afin de réduire les coûts et améliorer nos architectures ?

### 3. Speaker's Qualifications



- Thomson-Reuters
- BNP-Paribas consultant
- Certified Scrum Master by Jeff Sutherland
- Author of "Le Touilleur Express", a popular blog about Java in French
- Member of Paris Java User Group



Après 5 ans 1/2 chez Reuters, j'ai débuté une nouvelle activité en tant qu'indépendant. Je travaille actuellement chez BNP-Paribas sur une mission d'architecture. Je suis membre du Paris Java User Group.

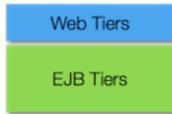
### 4. ESB



ESB. Ce mot est à la mode. A Devvix j'ai compté 4 conférences cette année. Mule est traité 3 fois par Antoine Borg avec qui j'ai eu le plaisir de discuter la veille, bref il se passe quelque chose. A noter qu'aucune conférence ne nous parle de téléphone portable, les thèmes cette année tournent autour de l'intégration, de la sécurité ou de SOA. Voyons un peu "dans la vraie vie" où nous en sommes aujourd'hui, car moi, plutôt que de parler de SOA ou d'ESB, j'aime bien parler du terrain avant tout.

### 5. A legacy

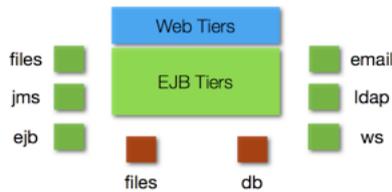
# A legacy application



## application

Ce que l'on constate c'est que les applications vieillissantes dans le monde Java des applications de gestion et de la finance, se présentent avec un socle d'architecture bien connu. Autour d'une architecture trois-tiers, nous retrouvons souvent du Struts pour la partie Web, un serveur d'application avec encore beaucoup d'EJB 2.1, et les innovations des dernières années que sont Hibernate, Spring et JMS. Et il est clair que ce type d'architecture aujourd'hui en production est encore amené à perdurer pour quelques années. Votre travail sera donc de faire vivre, d'améliorer ou de transformer ce type d'architecture. **6. A legacy application (cont.)**

# A legacy application

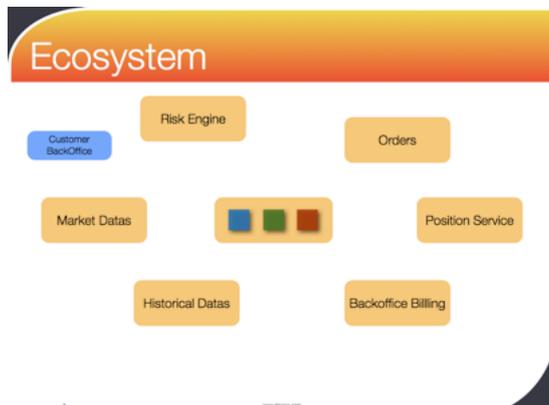


Dans la finance, une application web communique et échange des ressources avec le monde extérieur en utilisant un certain nombre de connecteurs techniques (boîtes vertes). Ce code est bien souvent du code développé au fur et à mesure des besoins. Il peut s'agir de lire un fichier, d'échanger via JMS avec une autre application, de faire appel à un web service, ou d'envoyer un email. Bref, du code massivement technique. Ce code est le socle de communication de chaque application de gestion. **7. Connect**

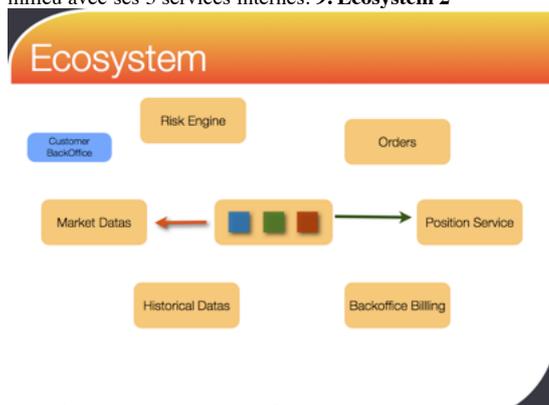
# Connect



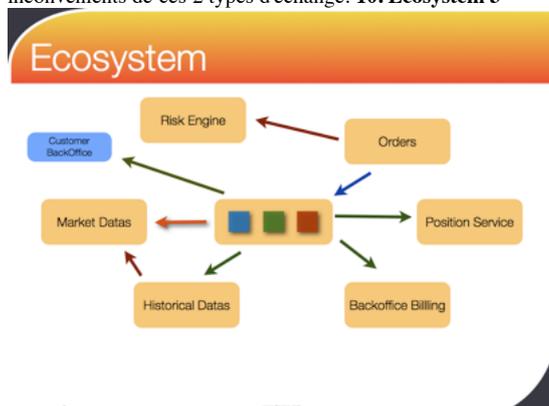
Connecter. Le besoin de Connecter différents systèmes. Ce besoin entraîne un travail de médiation, de transformation et d'interconnexion entre systèmes. Je veux montrer ici qu'il y a forcément un besoin de connexion, au minimum avec une base de données relationnelles, mais bien souvent avec d'autres systèmes informatiques ou techniques. C'est une composante fondamentale de nos solutions, de ce que nous sommes appelés à développer. **8. Ecosystem**



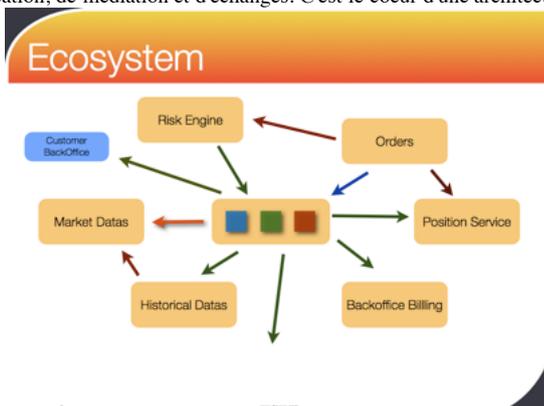
J'ai représenté sur ce slide une application financière typique de création de rapports, utilisé sur une plateforme de Prime Brokerage. Ce que nous voyons tout d'abord, c'est que cette application a besoin d'un environnement pour fonctionner, un écosystème. Cet écosystème est composé d'un grand nombre d'applications, maintenu par d'autres équipes que la mienne. Mon application est représentée au milieu avec ses 3 services internes. **9. Ecosystem 2**



Cette application utilise les données du marché ainsi qu'un service de tenue de Position. Aujourd'hui les échanges entre les systèmes s'effectuent avec des fichiers plats via des répertoires partagés ou via JMS. Dans le livre Enterprise Integration Pattern, une excellente présentation explique les avantages et les inconvénients de ces 2 types d'échange. **10. Ecosystem 3**



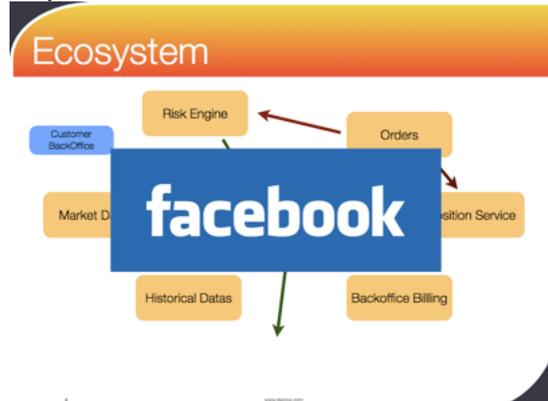
Au delà des échanges de mon application, ce qui est encore plus intéressant c'est de constater que les autres applications échangent elles-aussi des informations entre elles. Il faut donc bien voir que nous sommes là face à un vrai réseau de communication, de médiation et d'échanges. C'est le coeur d'une architecture



financière. **11. Ecosystem 4**

Au

final je dois prendre en compte que mon application cohabite, échange et transfère ses données avec un nombre assez important d'applications. Je ne rentre pas dans les détails de la partie métier, car l'idée ici n'est pas de vous donner un cours de finance des marchés, simplement d'expliquer que les systèmes sont massivement connectés en un réseau. **12. Ecosystem 5**



... En forçant un peu le trait, je peux faire une analogie avec un réseau social comme FaceBook : mon application fait partie d'un réseau, d'autres applications m'envoient des messages, observent même les données que je produis sans que je ne le sache, il y a donc clairement un besoin important à résoudre : ce besoin de communication et d'échanges avec mes "amis". **13. Collaborate**

## Collaborate



Collaborate en anglais, difficile à traduire avec le même sens en français. Collaborer c'est apprendre à se connaître : lorsque mon application échange avec d'autres systèmes financiers, au delà du problème technique à résoudre il y aura une interaction humaine. Et donc, ESB ou non, SOA ou non, ce projet d'inter-communication doit s'accompagner de la définition d'une langue pour se comprendre, d'un média pour discuter et d'un contenu pour s'entendre. Revenons un peu à notre application financière type pour parler d'un facteur très important : le temps. **14. Timebox and Adaptive**

## Timebox, Adaptive



Paris



Tokyo

J'ai sélectionné ces deux photos du métro afin d'illustrer un facteur important dans les applications financières : l'importance de l'heure de la journée. Prenons le métro de Paris : le matin entre 07h00 et 09h00 il y a un métro chaque minute. A 14h il y a un métro toutes les 6 minutes. A 23h00 il y a une rame toutes les 15 minutes. Le nombre de rame de métro par heure s'adapte selon les heures d'affluence. Qu'en est-il de mon application financière ? Chaque matin celle-ci a besoin de 10 fois plus de puissance qu'en pleine journée, car la génération des rapports doit s'effectuer à la fermeture du marché boursier, dans un intervalle de temps limité. Il n'est pas possible de lancer la génération tant que le marché est ouvert. D'autre part celle-ci doit s'effectuer avant l'ouverture du marché suivant, en Asie. De fait, nous nous retrouvons avec une architecture qui doit répondre à des demandes fluctuantes selon l'heure de la journée. Mais le plus important est que notre application n'a pas le droit à une deuxième chance. Lorsqu'un bug survient, il faut que l'erreur soit corrigée avant l'ouverture du marché, et ce phénomène n'est pas contrôlable. Enfin on constate que passé les quelques heures d'activité, la plateforme est ensuite peu sollicitée en cours de journée. Or le souci, c'est qu'aujourd'hui mon architecture n'est pas écologique : 16 serveurs de calcul de risk tournent en permanence, 4 serveurs d'applications sont sollicités pendant la journée mais l'utilisation des

ressources est assez inégale durant la journée. Il faut donc retenir que dans la finance, l'utilisation des ressources et la quantité des échanges avec les autres systèmes varient au cours de la journée. Cependant nous savons à l'avance nos besoins, et donc idéalement notre système devrait pouvoir s'adapter et se programmer comme le réseau de transport parisien. Un site internet qui vend des ordinateurs pourrait être surpris de l'affluence de visiteur. Pas une application financière : il est possible de connaître à l'avance les besoins en énergie. Mais pourtant, il est difficile aujourd'hui de régler et d'adapter cette énergie. **15. Crisis**

## 2009



2008 marque le début d'une crise financière, qui aura un impact dès demain sur les budgets des projets dans le monde de la banque et de la finance. Nos besoins de connexions avec les autres applications financières demeurent. Je pense même qu'il y aura de nouvelles réglementations pour éviter certains dérapages de 2008. Vu la nervosité des marchés financiers, les calculs de VaR, de risque, d'appel de marge, de collatéral sur notre application de Prime Brokerage vont aussi certainement être plus intensif... Bref il va y avoir du sport en 2009. **16. Predictions for 2009**

## Prediction for 2009



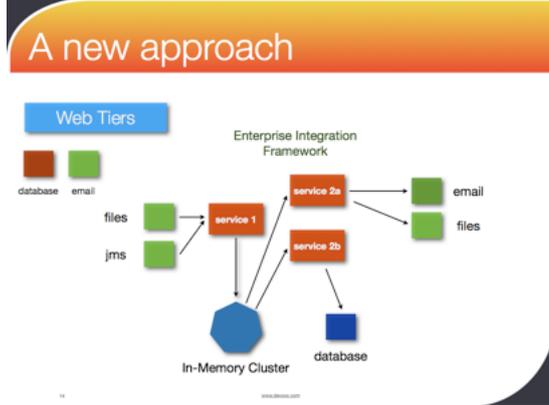
Il est donc facile d'imaginer ce qui nous attend : avec moins de budget, moins de liberté et plus de pressions, il va falloir trouver des solutions rapides, économiques et capables de faire réaliser des économies. Pour ma part, un calcul de risque c'est un appel de marge pour un client. Si la plate-forme sur laquelle je travaille peut traiter 50 000 calculs au lieu de 10 000 car mon architecture évolue, alors je ferai gagner de l'argent à mon client. D'autre part, vu la morosité et la crise du secteur de la finance, je pense que les solutions pragmatiques seront les gagnantes de 2009. J'ai peur que de gros projets basés sur les ESB des constructeurs souffrent du même phénomène que les constructeurs automobiles. Les gens veulent une voiture ? ils achètent une Logan et pas le dernier 4x4 à la mode. Votre application a besoin d'un ESB ? Peut-être que Spring Integration, Mule ou Apache ServiceMix seront suffisants pour vos besoins, que les gros projets sur 6 mois auront plus de mal en 2009 à être validé par les DSI. Il est temps d'être pragmatique et d'arrêter de rêver sur des besoins que les grands éditeurs d'ESB nous ont peut-être inventé, comme les éditeurs de serveur d'app en 2000... Etre pragmatique, économe, écologique, réaliste. **17. Toolbox**

## Your 2009 toolbox



En 2009 je pense que vous lirez des articles, que vous testerez, que vous entendrez parler de ces produits : Mule, Apache Service Mix, Apache Camel, Spring Integration, OpenESB project Fuji, Apache Synapse, Apache CXF, Jboss ESB... car ce sont des solutions qui répondent à un besoin : l'intégration légère en entreprise.

Après la guerre des serveurs d'application il y a quelques années, je pense que la prochaine grande bataille sera sur ce front. Les éditeurs du monde open-source savent que la demande va exploser, car lorsque la crise est là comme en 2001, le monde open-source tire son épingle du jeu. En 2002 personne ne se souvient mais c'était la grande époque des articles sur Linux dans OI informatique, le Monde informatique ect. L'administration français a alors fait un grand pas en sélectionnant des solutions comme JBoss Application Server pour traiter... nos déclarations de revenus. Economies, pragmatismes, simplifications, réductions des coûts... **18. A New Approach**



Alors si nous repensons à notre application de gestion du début, voici comment je la vois en 2009. Tout d'abord le Web tiers a été déporté de mon architecture. Je ne fais pas une application Web, je fais d'abord l'architecture d'une application de gestion. Le web ne me coûte pas cher, j'utilise des frameworks légers sur une recopie en lecture seule de ma base de données. Seule une petite partie du code métier est partagée avec mon espace de données principal. Ensuite on voit que j'ai cassé ce gros rectangle : je pense que des services sans états et légers sont suffisants pour effectuer les traitements de mon application. Là où auparavant j'utilisais la base de données pour stocker des données de travail, j'utilise désormais un espace mémoire en cluster afin de répliquer les données nécessaires à mes traitements de calculs. Cela demande plus de mémoire physique mais quintuple les performances de traitements. La base de données relationnelle n'intervient que via un service d'historisation qui effectue des captures de données afin de sauvegarder les données les plus sensibles, essentiellement pour un problème légal. Notez que la base est un petit carré bleu, car finalement elle ne me sert plus à grand chose. Ces services reprennent la boîte à outil dont nous avons parlé tout à l'heure. Plutôt que de chercher à faire une grosse boîte qui fait tout (un ESB au sens de certains) je crée un ensemble de services qui discutent et s'échangent des données. Et plutôt que de tenter de résoudre les problèmes de communication avec les applications extérieures, de tenter de mettre en place un gros chantier afin de mettre tout le monde sur la même plateforme, je conserve et respecte les systèmes extérieurs, je m'adapte à eux, eux ne s'adaptent pas à moi. Et là je sais que mon application pour 2009 pourra continuer à exister sans coûter une fortune. Il faut donc oublier ce rêve qui serait de migrer toutes les applications de mon réseau vers un seul et même système. Ce serait un risque si l'éditeur disparaît (ce qui n'est pas exclu en ce moment). Ce serait la plus belle décision pour dégrader massivement les performances : pourquoi un système extérieure serait plus rapide que ce que mes équipes ont développé ? Ce serait un risque technique, un risque en terme de sécurité, et finalement, il faudrait attendre 6 mois avant de voir ce type de projet de gouvernance sur les rails. Pour revenir à l'image du métro parisien, je rêve aussi de louer de la puissance machine plutôt que de l'acheter en utilisant par exemple Amazon EC2 et une solution comme Elastic Grid pour piloter mon parc informatique. Pour cette raison mon nouveau système est découpé en services sans état afin de faciliter une migration partielle et rapide de certains services vers ces nouvelles plate-formes de calculs et d'hébergement. La première banque qui aura déployé une plateforme de calcul de risque hébergée comme un service sur internet gagnera certainement beaucoup d'argent. Economies, pragmatismes, simples, réduire les coûts, arrêter de croire au père Noël, pensez sur 6 mois... **Code code code** Ensuite j'ai un slide technique pour montrer un exemple d'utilisation d'Apache Camel. Lors de la conférence je n'ai pas eu assez de temps pour les présenter. Je vous ferai un article complet sur Camel prochainement, donc revenez



```

/**
 * Generates an email from in-memory queue using Velocity template,
 * Stores this email in a folder
 * Then another service polls this folder, add an email header
 * and send to smtp server the generated email
 * Author Nicolas Martignole
 */
public class GenerateEmailStoreToFilePollFolderAndSendFile extends RouteBuilder {
    public void configure() throws Exception {
        from("direct:start")
            .to("velocityMailBody.vw")
            .setHeader(FileComponent.HEADER_FILE_NAME, "errorEmail.txt")
            .to("file:///Users/nicolas/Dev/DevboxDemo/wicket/target/demoCamel?append=false");
        // second part from the file backup -> send email
        from("file:///Users/nicolas/Dev/DevboxDemo/wicket/target/demoCamel")
            .setHeader("subject", constant("New incident reported"))
            .to("smtp://nicolas@localhost?password=secret&to=incident@noteria.fr");
    }
}

```

en janvier pour voir la suite.



# Keep it Simple



### it simple

Il est tentant de faire n'importe quoi, de faire au plus vite ou de chercher une solution chère et trop complexe pour un problème. Soyez simple et vous réussirez à développer une architecture qui fonctionne. **Ne pas faire de l'intégration pour en faire...** Comme ces messieurs dans la piscine : ne cherchez pas à faire de l'ESB pour en faire, et faites attention à votre environnement. Un ESB ne remplace pas une réunion de pilotage avec l'autre équipe en face qui

Pour connecter plusieurs

# Keep it Simple



fait "l'autre" produit.

Don't buy a

# Don't buy a solution



### solution

N'achetez pas les yeux fermés une solution. Au contraire, il faut que vos équipes retroussent leurs manches, repensez à des solutions simples. Dans le monde des applications de gestion, et de la Finance, le temps est certes à l'économie, mais aussi à un pragmatisme souvent contraire à nos envies d'architectes et de

N'achetez pas les yeux

# The End

<http://touilleur-express.fr>

[nicolas@innoteria.fr](mailto:nicolas@innoteria.fr)



consultants. **The End**

En

conclusion : nous avons discuté des besoins en architecture dans le monde de la finance : connecter, échanger et collaborer, inter-agir avec d'autres systèmes. Les solutions d'intégrations légères seront en 2009 un moyen de réduire les coûts en évitant d'écrire à la main du code technique sans valeur ajoutée. Je souhaite dépiler un message JMS et envoyer un email, Mule

m'aide à le faire en quelques heures au lieu de perdre mon temps avec du code sans valeur. Face aux réductions de budgets, il est logique de penser que les solutions d'intégration dans la finance auront le vent en poupe en 2009. Et face à cette crise, il y aura encore plus de travail pour nous, avec encore moins de budgets, d'où la nécessité de s'outiller enfin correctement. Fin de la présentation -----

## Comment simuler le navigateur de l'iphone avec Firefox ou Safari ?

J'ai installé [WPtouch](#), un plugin Wordpress sur le Touilleur Express afin que les visiteurs utilisant l'iphone puissent voir une version optimisée pour l'iphone. De fil en aiguille je me suis demandé comment tester ce plugin... si vous n'avez pas d'iphone sous la main. Il existe des sites internet comme [TestiPhone](#) ou [iPhone Tester](#) qui permettent de simuler la taille de l'écran de l'iphone. Cependant ces site ne sont pas capables de spécifier un User-Agent différent dans les entêtes HTTP puisqu'il s'agit de votre propre navigateur. Le site s'affiche alors normalement car le serveur ne reconnaît pas l'iphone.

### Firefox

Firefox propose un moyen de changer la variable User-Agent avec le plugin [User Agent Switcher](#). Il faut ensuite connaître la valeur de la variable User-Agent de l'iphone, ce que j'ai trouvé dans les logs d'accès du Touilleur : Mozilla/5.0 (iPhone; U; CPU iPhone OS 2\_2 like Mac OS X; fr-fr) AppleWebKit/525.18.1 (KHTML, like Gecko) Version/3.1.1 Mobile/5G77 Safari/525.20 Après avoir installé le plugin et relancé Firefox, dans le menu Outils / Agent Switcher sélectionnez Options. Dans l'onglet User Agent vous pouvez alors déclarer un nouveau navigateur. Ayant mis à jour mon iphone 3G en version 2.2 voici ce que j'ai saisi :

- o Description: iPhone 3G FR update 2.2
- o User Agent: Mozilla/5.0 (iPhone; U; CPU iPhone OS 2\_2 like Mac OS X; fr-fr)
- o App Name: AppleWebKit/525.18.1 (KHTML, like Gecko)
- o App Version: Version/3.1.1
- o Platform: Mobile/5G77 Safari/525.20

Une fois votre nouveau navigateur déclaré, vous pouvez alors l'activer via le menu Outils, et tentez de recharger la page du touilleur, vous devriez alors voir ceci :



Il ne reste plus qu'à charger la page via le site <http://www.testiphone.com> afin d'avoir un affichage très semblable à celui de l'iphone. Il reste simplement le souci que le navigateur n'est pas Safari mais Firefox. Certaines fonctions Javascript pourraient avoir un comportement un peu différent. Voici le résultat final



Voici une petite liste de sites spécialement adaptés à

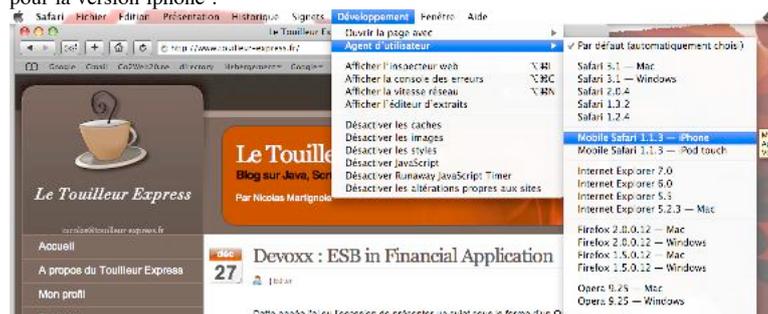
l'iphone :

- o [Google Calendar](#)
- o [Google Reader](#)
- o [Le Blog de Xebia](#)

qui utilise aussi le plugin WPtouch

### Safari

Si vous souhaitez tester avec Safari il est aussi possible de changer la variable User-Agent lorsque Safari est démarré en mode développeur. C'est vraiment très simple : - fermez Safari - ouvrez un Terminal - tapez dans la console la ligne de commande suivante : **defaults write com.apple.Safari IncludeDebugMenu 1** puis Enter - relancez Safari, un menu Développement est apparu. - dans le menu "Agent d'utilisateur" vous pouvez maintenant faire passer Safari pour Internet Explorer ou pour la version iPhone !



C'est le moyen le plus efficace pour tester la navigation d'un site développé pour l'iPhone sans s'embêter avec le development kit d'Apple.

## Création du French Scrum User Group

C'est parti ! Le French Scrum User Group est lancé, l'inscription est ouverte à tous. Un événement est organisé le 19 mars prochain lors de la venue de Jeff Sutherland (l'un des créateurs de Scrum) sur Paris. L'initiative de Luc Legardeur de Xebia se veut indépendante, libre et donc ouverte à tout le monde, sans aucuns liens avec Xebia. L'idée sera de se retrouver entre personnes intéressées par Scrum, afin d'échanger nos expériences avec Scrum. Pour vous inscrire : <http://www.meetup.com/frenchsug/> A suivre donc !

## Exemple d'intégration d'Apache Wicket et Apache Camel

[Update] 03/03/2009 : [English version available !](#) Dans cet article je vous propose de regarder un exemple simple d'utilisation du framework d'intégration léger Apache Camel avec l'un des frameworks webs les plus intéressants, Apache Wicket. Je vous ai laissé une archive avec le code source, prêt à être compilé afin que vous puissiez tester tranquillement chez vous. Si vous souhaitez réutiliser ce code ou publier un article sur un journal papier, je vous remercie de me contacter avant de le faire. **Qu'est-ce qu'Apache Wicket ?** [Wicket](#) est un framework web Java open-source orienté composant. Contrairement à un framework orienté actions comme Struts 1, Wicket vous permet de définir en Java vos pages et vos composants, d'intégrer ensuite dans votre code HTML des tags légers, afin enfin de rendre le tout dans un serveur d'application léger. Lorsque je travaillais chez Reuters, j'étais en charge du framework Karma, dont l'architecture et la philosophie est vraiment très proche de Wicket. Pour la création d'application web, c'est donc une solution intéressante à envisager. Dans l'exemple à suivre vous verrez comment créer votre première page, un composant particulier, et utiliser Spring avec Camel pour charger la configuration. Ce qu'il faut retenir de Wicket c'est que le modèle d'événements du côté du serveur est semblable à Swing. Vous implémentez des listeners et sur la base du pattern Observer-Observable, le moteur de Wicket se charge de vous notifier selon les actions de l'utilisateur dans la page Web. Avec un support d'ajax très correct, c'est donc une solution intéressante à connaître. **Qu'est-ce qu'Apache Camel ?** Apache Camel est un framework basé sur Spring d'intégration. Il implémente les EIP (Enterprise Integration Patterns) en proposant d'une part l'utilisation du Xml pour la configuration (comme Mule) ou d'autre part un DSL en Java très simple, ce que nous verrons dans cet exemple. Apache Camel est complémentaire de Mule. Mule est un framework léger d'intégration, Camel est une boîte à outils de patterns que vous pouvez d'ailleurs utiliser avec Mule, celui-ci se chargeant alors de la partie connexions, modèle de thread, là où Camel se charge du routage, de la transformation et de la médiation. Ici je n'utilise que Camel, afin simplement de vous montrer son fonctionnement. **Explication de la démonstration** J'avais préparé cette démonstration début décembre dans l'idée de la présenter lors de [ma présentation à Devoxx](#). Vu la durée de la présentation, il n'était pas envisageable de se lancer finalement dans une démo, j'ai donc gardé ce code de côté... jusqu'à aujourd'hui ! ... **et pourquoi tu n'as pas pris Mule/Spring Integration/Apache ServiceMix/OpenESB/MachinTrucPouetPouet** J'aimerais bien montrer une autre solution basé sur Mule uniquement, afin que chacun se fasse sa propre idée. Je dois tester Spring Integration avant de pouvoir vous dire et vous montrer un peu de code, donc patience ! Pour le reste on verra. L'application de démonstration Voici comment se présente l'application, une fois celle-ci déployée avec un "mvn jetty:run"

InnoTeria

New Forum

events

11th dec 2008

To contact me  
nicolas@innoteria.fr

## Devoxx Quickie demo

Welcome to my demo application based on **Apache Wicket** to demonstrate some concepts of **Apache Camel** with just few lines of codes.

In this demo you will see how to use **Wicket** to read Form input fields, how to use **Apache Camel** to transform a form to a plain text file using **Velocity**. You will also see how to poll a local folder and send by email a text file using **Camel**.

To send a request to the application, please fill the form and click the Submit button. Then click on the "Check mailbox" hyperlink at the bottom of this page.

Your name

Your message:

[Check incident mailbox...](#)

En tant que client, je mets mon nom dans le formulaire et un petit message afin de demander pourquoi ma superbe console PS3 n'est pas arrivée à Noël. Je clique sur envoyer et l'interface

To send a request to the application, please fill the form and click the Submit button. Then click on the "Check mailbox" hyperlink at the bottom of this page.

Your name  
Nicolas

Your message:  
I didn't receive my PS3 for christmas !

Thanks!

[Check incident mailbox...](#)

affiche "Thanks".

Que se passe-

**t-il derrière ?** Lorsque je clique sur envoyer, Apache Camel reçoit de Wicket un message. Il va transformer ce message en fichier et le stocker dans un répertoire. Ensuite, on imagine qu'un autre processus de Camel va surveiller ce répertoire, se réveiller, et lire le message. Il va charger un template Velocity afin de générer un email, puis enfin envoyer le tout via SMTP vers une boîte email de test. J'ai utilisé MockMail comme nous verrons plus loin tout à l'heure. **Vue administrateur** Pour démontrer le fonctionnement, j'ai ensuite implémenté une page qui liste les emails présents dans la boîte aux lettres. Nous verrons comment cela fonctionne tout à l'heure. **Premiers pas avec Wicket** Pour la suite, vous pouvez récupérer l'archive que j'ai mis à la fin de cet article, cela sera plus simple pour suivre le code. En premier lieu nous allons regarder la class `InnoTeriaWicketApplication`. Il s'agit du point d'entrée de l'application Wicket. J'utilise [Mock JavaMail](#) pour déclarer une boîte aux lettres. La class `ReportIncidentEndpoint` est l'un de mes composants Apache Camel, il représente ici la porte vers Camel pour Wicket.

```
public class InnoTeriaWicketApplication extends WebApplication {
    private ApplicationContext ctx;
    private ReportIncidentEndpoint endpoint;
    private Mailbox mailbox; // mock javamail

    private static ISpringContextLocator CTX_LOCATOR = new ISpringContextLocator() {
        public ApplicationContext getSpringContext() {
            return InnoTeriaWicketApplication.get().ctx;
        }
    };
}
```

```

    }
};

public InnoteriaWicketApplication() {
}

public Class getHomePage() {
    return HomePage.class;
}

public static InnoteriaWicketApplication get() {
    return (InnoteriaWicketApplication) Application.get();
}

@Override
protected void init() {
    ctx = new ClassPathXmlApplicationContext("applicationContext.xml");
    Address address= null;
    try {
        address = new InetAddress("incident@innoteria.fr");
    } catch (AddressException e) {
        e.printStackTrace();
    }
    mailbox = new Mailbox(address);
}

/**
 * This trick is used to decorate a specified template class with a SpringBeanLocator
 * so that I can use Spring to load my Apache Camel configuration.
 * @param clazz will be the endpoint to decorate
 * @param <T> template
 * @return a decorated class.
 */
private <T> T createProxy(Class<T> clazz){
    return (T) LazyInitProxyFactory.createProxy(clazz, new SpringBeanLocator(clazz, CTX_LOCATOR));
}

public ReportIncidentEndpoint getReportIncidentEndpoint(){
    if(endpoint==null){
        endpoint=createProxy(ReportIncidentEndpoint.class);
    }
    return endpoint;
}
}
}

```

Une fois votre application Wicket déclarée, il est temps de coder votre page d'accueil. La class HomePage étend WebPage. Nous voyons qu'un composant FeedbackPanel est déclaré avec 2 champs de type texte. Un formulaire est ensuite ajouté. La méthode onSubmit sera déclenchée lorsque le visiteur soumet ce formulaire dans le navigateur. On verra alors comment le service Apache Camel récupère l'instance de MyReportIncident pour la suite. En quelques lignes, vous voyez qu'il est simple de coder un formulaire de saisie et de déclencher une action lorsque l'utilisateur soumet celui-ci.

```

public class HomePage extends WebPage {

    private static final long serialVersionUID = 1L;

    public HomePage(final PageParameters parameters) {
        // un de mes composants
        FeedbackPanel feedbackPanel = new FeedbackPanel("feedback");
        add(feedbackPanel);

        final TextField yourName = new TextField("yourname", new Model());
        final TextField yourMessage = new TextField("yourmessage", new Model());
        yourMessage.setRequired(true);
        yourName.setRequired(true);

        Form form = new Form("formtest") {
            protected void onSubmit() {
                ReportIncidentEndpoint service = InnoteriaWicketApplication.get().getReportIncidentEndpoint();
                service.reportIncident(
                    new MyReportIncident(
                        yourName.getModelObject().toString(),
                        yourMessage.getModelObject().toString()
                    ));
                info("Thanks!");
            }
        };

        form.add(yourName);
        form.add(yourMessage);

        // Add a form with an onSubmit implementation that sets a message
        add(form);

        /* java code */
        add(new BookmarkablePageLink("link", CheckMailbox.class));
    }
}

```

Il faut ensuite aller voir du côté du code HTML pour comprendre comment Wicket fonctionne. Prenons cet extrait de HomePage.html

...

Your name

Your message:

...

On retrouve les id des composants déclarés dans la class Java. L'avantage de Wicket est qu'il permet de réaliser sa mise en page en HTML et d'utiliser la fonction aperçu HTML d'IDEA IntelliJ. Finalement le code de la page reste lisible et ce n'est que du HTML. Voilà pour cette mini introduction à Wicket, je vous laisse regarder le code tranquillement, et on reprend après la coupure pub. **Le côté Camel** Notre point d'entrée dans Camel est la class ReportIncidentEndpoint. La méthode reportIncident est notifiée par Wicket, elle publie sur une Queue en mémoire le message. La class ReportIncidentEndpoint est un bean spring qui est injecté un peu spécialement dans la class InnoteriaWicketApplication.

```
public class ReportIncidentEndpoint implements Serializable {
    private CamelContext context;
    private static final long serialVersionUID=2233L;

    public ReportIncidentEndpoint() throws Exception {
        // create the context
        context = new DefaultCamelContext();

        // append the routes to the context
        context.addRoutes(new ReportIncidentRoutes());

        // at the end start the camel context
        context.start();
    }

    /**
     * Callback method executed from the HomePage when the user
     * clicks on submit.
     * @param message is a message created by Wicket
     * @return a status message that is shown on HomePage.
     */
    public String reportIncident(MyReportIncident message) {
        // create the producer template to use for sending messages
        ProducerTemplate producer = context.createProducerTemplate();

        // Send the message to an in-memory queue and return
        // See GenerateEmailStoreToFilePollFolderAndSendFile
        // it will read the message from direct:start in-memory queue
        Object mailBody = producer.sendBody("direct:start", message);
        System.out.println("Body:" + mailBody);

        return "OK";
    }
}
```

Voyons maintenant deux exemples d'intégration avec Apache Camel. J'ai déclaré explicitement 2 routes, pour vous montrer l'un des principes de Camel. Regardez la class ReportIncidentRoutes. Celle-ci étend la class RouteBuilder de Camel qui vous permet de définir pour votre contexte le chemin de traitement. Notez qu'il est aussi possible de faire appel à du XML, mais ici je trouve élégant la définition de votre route en java, surtout lorsque finalement votre architecture ne changera plus. La première route lit la queue "direct:start", appelle ensuite un template Velocity pour générer un email, ajoute le nom d'un fichier et enfin, enregistre dans le fichier errorEmail.txt le message de l'utilisateur. Notez que je pourrais tout aussi bien envoyer vers une Queue JMS, peu importe. Ici l'idée est de vous montrer le fonctionnement de Camel.

```
public class ReportIncidentRoutes extends RouteBuilder {

    public void configure() throws Exception {
        from("direct:start")
            .to("velocity:MailBody.vm")
            .setHeader(FileComponent.HEADER_FILE_NAME, "errorEmail.txt")
            .to("file:///./target/demoCamel?append=false");
        ...
    }
}
```

La deuxième route surveille ce répertoire, prend le premier fichier déposé, ajoute un header qui sera le sujet de l'email et enfin expédie le tout à mon serveur SMTP. Souvenez-vous, j'ai lancé dans ma class InnoteriaWicketApplication une instance de Mailbox (mock mail) et j'ai donc bien un serveur de test qui écoute sur le port 25, ce que vous pouvez vérifier avec un telnet localhost 25. L'email est donc réellement expédié pour Camel.

```
from("file:///./target/demoCamel")
```

```
.setHeader("subject", constant("New incident reported"))
.to("smtp://nicolas@localhost?password=secret&to=incident@innoteria.fr");
```

Enfin voici un autre exemple un peu plus avancé de Camel qui vous montre sa puissance :

```
// exemple 1 si la cle foo de l'entete est egale a bar alors fait suivre a queue b
from("queue:a").filter(header("foo").isEqualTo("bar")).to("queue:b");

// exemple 2 de gestion de queue d'erreur si rien ne correspond
from("queue:c").choice()
    .when(header("foo").isEqualTo("bar")).to("queue:d")
    .when(header("foo").isEqualTo("cheese")).to("queue:e")
    .otherwise().to("queue:errorQueue");
```

Camel est donc intéressant car il vise à réduire la quantité de code technique habituellement nécessaire pour envoyer un email, lire un fichier, le transformer, envoyer un message vers une Topic JMS etc. L'idée de ce framework d'intégration est de faciliter les échanges avec les systèmes externes de votre application. En prenant l'exemple de l'envoi de mail je sais que certains vont me dire qu'il est possible de le faire autrement. Mais cet exemple est plus facile à écrire qu'un exemple avec du JMS par exemple. Donc si le sujet vous intéresse, allez voir sur le site d'Apache Camel qui contient pas mal d'exemples complets. Ce framework est facile à apprendre et il peut vous rendre service pour des cas d'utilisation relativement courant. **Comment lire le contenu de la boîte email et afficher les messages ?** Revenons à Wicket. Comment maintenant lire les messages en attente dans la Mailbox et les afficher à l'utilisateur ? Si vous cliquez sur le lien "Check Incident Mailbox..." sur la page d'accueil, vous verrez alors cet écran :



La class

CheckMailbox est une page Web wicket qui retrouve la liste des courriers électroniques envoyés à l'adresse "incident@innoteria.fr". Nous passons ensuite notre iterator à un composant fait maison qui liste les messages et les affiche. J'ai adapté une class de l'api Mail car celle-ci n'étant pas sérialisable, elle ne peut pas être utilisée par Wicket directement. Hop on a ici le pattern Iterator et le pattern Adapter. Elle est bas belle la vie ?

```
public class CheckMailbox extends WebPage {
    private static final long serialVersionUID = 1L;

    public CheckMailbox(final PageParameters parameters) {
        Address address = null;
        try {
            address = new InternetAddress("incident@innoteria.fr");
        } catch (AddressException e) {
            e.printStackTrace();
        }
        Mailbox mailbox = Mailbox.get(address);
        // javax.mail.message subclasses are not serializable
        // thus we need this little trick to adapt our collection.
        // I don't want use collections-utils and folks, it's too simple.
        InnoteriaMessageIterator imi=new InnoteriaMessageIterator(mailbox.iterator());
        MessagePanel rv = new MessagePanel("repeatingPanel", imi);
        add(rv);
    }
}
```

**My first composant** La class MessagePanel est un composant Wicket, définit avec son propre code HTML. DisplayMessagePanel est un Panel contenu dans MessagePanel. Lorsque j'ai écrit ce code, je pensais ajouter d'autres widgets Wicket dans MessagePanel. Bref passons. J'aime beaucoup l'élégance de wicket dans cette class. Sans chercher à être hyper générique, c'est simple, brutale et efficace. A l'exécution vous vous en doutez, ce code liste les messages et affiche donc la liste des réclamations des clients.

```
public class DisplayMessagePanel extends Panel {

    public DisplayMessagePanel(String id, final InnoteriaMessageIterator listOfMessages) {
        super(id);

        add(new RefreshingView("repeatingView") {
```

```

@Override
protected Iterator getItemModels() {
    return new ModelIteratorAdapter(listOfMessages.iterator()) {
        @Override
        protected IModel model(Object o) {
            return new CompoundPropertyModel((InnoteriaMessage) o);
        }
    };
}

@Override
protected void populateItem(Item item) {
    // Those 3 properties are read directly on InnoteriaMessage POJO
    // See CompoundPropertyModel javadoc for more details
    item.add(new Label("subject"));
    item.add(new Label("from"));
    item.add(new Label("content"));
}
});
}
}

```

**La conclusion de la fin pour terminer enfin** Voilà, j'ai passé pas mal de temps à écrire l'exemple complet. L'idée était de vous présenter un exemple complet basé sur Wicket d'une part et sur Apache Camel d'autre part. Il sera intéressant de faire le même code avec par exemple Spring Integration, Spring MVC afin de se faire son idée. Avis aux amateurs !!! Code source complet de cet article : [touilleur\\_wicket\\_camel\\_1.0.tar.gz](http://toulouse.wicket.camel.1.0.tar.gz)

## Compte-rendu de la soirée du Paris JUG sur JEE6 et Glassfish

La première soirée du Paris JUG avait lieu comme chaque deuxième mardi du mois à l'ISEP, avec le sponsor de SUN Microsystems ce soir. Les 2 thèmes présentés : JEE6 par Antonio Goncalves et Glassfish v3 "Prelude" par Alexis Moussine-Pouchkine. Je vous le dis d'entrée : c'était une très bonne soirée. Sur une échelle Touilleur de 1 à 10, nous sommes pas loin du 8. On reste à 7 mais honnêtement, le contenu était très intéressant. Absorbé par Devoxx j'ai zappé les présentations d'Antonio à la conférence [Open Source Exchange](http://www.opensourceexchange.com) organisée par Xebia, et à la présentation d'Alexis en décembre puisque j'étais encore en Belgique. C'est donc avec un œil neuf et affûté que j'ai écouté, noté et apprécié ce soir les présentations. J'ai pris la parole au début afin de présenter 2 informations importantes : la création du Scrum User Group, sur le modèle du Paris Java User Group. Cela se passe sur le site <http://www.frenchsug.org>, et nous nous verrons tous j'espère le 19 mars à Paris afin d'assister à l'inauguration du SUG en présence de Jeff Sutherland. Je compte sur votre présence si Scrum vous intéresse ! Deuxième chose que j'ai proposé au JUG il y a quelques semaines : reprendre l'idée des tableaux blancs vue à Devoxx et proposer aux gens de voter pendant le buffet. Les deux thèmes proposés ce soir : "Quel serveur d'application est votre serveur favori ?" et "En dehors de Java, quel est le langage qui vous intéresse le plus ?". Grâce à mon superbe iPhone je vous donnerai les résultats demain. Antonio a donc commencé sa présentation afin de nous donner un état des lieux de JEE6 ainsi que les prochains sujets à venir. Tout d'abord, un peu d'historique. J2EE a maintenant 10 ans. En 1999 c'est la sortie de la spécification 1.2 avec déjà les servlets, les JSP et JMS. En Novembre 2003, sortie de la version 1.4 de Java. Oui cher lecteur, votre superbe serveur d'application se base sur une spécification qui a presque 6 ans ! En mai 2006 c'est la sortie de JEE5, avec déjà 23 spécifications. Enfin la prochaine étape en juin 2009 sera la sortie de JEE6, si tout va bien. Alexis pense que la sortie officielle se fera pour JavaOne 2009. Le mot d'ordre pour JEE6 est : - plus riche - plus léger - plus facile JEE6 sera modulaire. Une version Lite (et pas light ?) sera proposée par exemple afin de donner aux développeurs uniquement ce dont il a besoin pour coder son application Web. Dans la version lite, si j'ai bien noté, pas de MDB, de RMI/IIOP, de Remote Interface, de Time Service, de CMP/BMP... Bref comme dit Antonio, c'est la version Coca light. L'effort de simplification vise à offrir un socle plus simple et plus prêt des cas d'usage que nous voyons au quotidien. Le Web profile est une spécification à part entière, avec Servlet 3.0, JSP 2.2, EL 1. et JSTL 1.2. Il est donc évident que quoique l'on puisse dire, JEE6 sera avant tout plus léger. Antonio explique ensuite la notion de "pruned". Certaines parties de la spécification (Entity Bean, JAX-RPC, JAX-R et JSR-88) seront marquées comme "pruned" ce qui correspond à un gros Deprecated. EN JEE7 ces modules disparaîtront car ils ne sont plus utilisés aujourd'hui. Ensuite nous avons voyagé dans la spécification JEE6. J'ai trouvé la présentation des Servlets 3.0 à la fois simple et efficace. Après avoir vu du code si clair et si mature, pourquoi aller chercher un framework d'injection alors que ce concept sera de facto dans JEE6 ? Quelques annotations comme @WebServlet ou @ServletFilter permettent au développeur de marquer son code Java, afin que celui-ci se comporte comme une servlet ou un filtre dans son application Web. La modularité et enfin le support d'un mode asynchrone ont achevé de convaincre le sceptique que j'étais en arrivant. J'ai été séduit par la clareté du code. J'avoue que depuis 4 mois je ne suis pas tellement l'activité du côté JEE6. Antonio étant un membre du JCP sur JEE6, il est bien placé pour nous dire ce qu'il se passe derrière en coulisse. On y reviendra tout à l'heure. La présentation sur JSF2.0 m'a permis de voir les efforts de simplification. @ManagedBean sera suffisant pour annoter un bean, le fichier faces-config.xml sera optionnel. JSF a viré les JSP et se base sur Facelet comme le confirme Alexis plus tard, autour d'une bière. JEE6 introduit quelques nouveaux concepts comme l'annotation @Singleton qui permet de s'assurer que le conteneur d'application n'instancie qu'une seule fois une classe. Rien de révolutionnaire mais c'est géré par le conteneur. Antonio enchaîne ensuite sur un souci vu et revu avec les EJB2.1 : comment ajouter des traitements asynchrones ? L'usage de la classe Thread n'est pas autorisé en EJB2.1. Pour palier cela il explique que les architectes ont fait appel à JMS afin d'introduire de l'asynchrone... Grave erreur. JEE6 propose une annotation toute simple : @Asynchronous. Cette annotation est placée

sur une méthode afin d'indiquer au conteneur d'application que son exécution doit être asynchrone. Ni plus, ni moins. Si vous devez ensuite récupérer le résultat de cette exécution, les Futures sont là pour vous aider. Du côté du packaging, l'EAR n'est plus obligatoire, vous pouvez packager votre application JEE6 dans un war. Nous continuons ensuite sur la présentation des portables JNDI. C'est très simple : les annotations permettent maintenant d'injecter automatiquement les références. Plus de narrow portable machin chose. Pour les tests d'intégration, JEE6 propose aussi la notion de conteneur embarqué. Cela permettra par exemple d'effectuer des traitements type Batch processing. D'ailleurs le Time Service m'a semblé un remplaçant tout à fait acceptable pour retirer du code basé sur Quartz et me reposer sur un standard. La syntaxe à la cron permet de scheduler l'exécution périodique d'une méthode, et donc, d'effectuer des traitements sous la forme de batches. En voyant tout cela et en pensant à notre plate-forme de traitement chez mon client, j'ai eu le sentiment de voir aussi une révolution pour nous, développeur. Cette simplification va entraîner un sacré effort de votre part pour bien vous former à JEE6. Cela tombe bien, Antonio a annoncé qu'il va sortir une nouvelle version de son livre en français. D'ailleurs (et après j'arrête la pub) SUN organise le 5 février un événement à son centre de formation à Paris à côté de la Madeleine. Il y aura des livres à gagner pour les personnes présentes. Je vous redonnerai plus de détails, ou si quelqu'un de SUN me lit, qu'il ajoute dans les commentaires les détails sur cet événement. Revenons à notre caviar. On sent un peu que la peinture n'est pas sèche, mais Antonio est transparent durant sa présentation. JPA 2.0 apporte encore quelques modifications, avec par exemple une annotation pour un meilleur support des collections (@ElementCollection). Par ailleurs j'ai noté que là où JPA 1.0 ne propose que le mode OPTIMISTIC LOCK, JPA 2.0 propose 5 modes différents de LOCK pour gérer finement la lecture des données transactionnelles. A suivre donc. Du côté de REST, puisque c'est l'un des dadas du moment, la spécification JAX-RS 1.1 est sortie. Elle propose via des annotations un système qui permet d'annoter un bean afin de le transformer en ressource REST. A ce propos, après avoir vu la simplicité de JAX-RS, je me demande pourquoi Spring a eu l'idée de refaire sa propre implémentation de Rest... Même Jérôme Louvel (framework RESTlet) offre un support de JAX-RS. Quel diable a piqué Spring pour qu'ils ne suivent pas le standard ? En conclusion JEE6 est simple, riche, modulaire même si je n'ai pas parlé des profiles, standard (hé oui) et robuste avec 10 ans derrière nous. Lors de la séance de question/réponse, Cyrille Leclerc de Xébia a mis le doigt là où il y a un gros point d'interrogation : quid de Web Beans ? La réponse il fallait aller la chercher à 23h35 après une petite bière. J'étais assis à côté d'Alexis et d'Antonio après la présentation au Falstaff, un bar où nous nous retrouvons après le JUG. La question c'est ce qu'il va se passer le 9 février. Les personnes du JCP vont voter pour ou contre l'adoption de WebBeans. En cas de refus, un deuxième vote sera alors proposé. En cas de refus, WebBeans sera retiré du scope de JEE6, ce qui peut très bien arriver. A Devv les 2 présentations que j'ai vu m'ont inquiété, et m'ont donné le sentiment que WebBeans propose son modèle là où les EJB 3 se débrouillent très bien depuis 2 ans... C'est donc un sujet qui représente en ce moment un gros travail pour les personnes du JCP, afin de mettre à plat et de s'assurer que la spécification finale JEE6 sera conforme aux attentes de tout le monde. **Deuxième partie :**

**Glassfish V3 "Prelude"** [Alexis Moussine-Pouchkine](#) est l'ambassadeur du projet Glassfish. Il fait le pont entre la communauté d'une part et le monde de l'entreprise d'autre part. Sa présentation porte sur la future version à sortir de Glassfish, le serveur d'application open-source de SUN. En introduction, Alexis explique qu'il y a presque 10 ans, SUN a donné à la communauté open-source le serveur d'application Tomcat. Aujourd'hui, il est intéressant de voir que certains partenaires, comme SpringSource, s'investissent sur Tomcat afin d'offrir un support professionnel et une offre de service. Il faut mentionner "[SpringSource te Server](#)" de SpringSource qui est un Tomcat avec des outils avancés de gestion et de diagnostics, utilisés par les équipes de production. La publicité a été supprimée après 20h30, je vais arrêter la pub pour Spring. En octobre 2008, le projet Glassfish a annoncé la sortie de la version V3 "Prelude", en prélude de la sortie de la spécification JEE6. Je pense que si tout va bien, Glassfish V3 "VAS Y MON GARS C EST LA VERSION CERTIFIEE JEE6" sortira cet été. Notez que pour le nom et la version on est pas tout à fait certain que SUN suive mon idée... Alexis d'un subtil clic de souris (yeah! Mac Rulez!) nous montre ensuite un slide sur le nombre de visiteurs du site Glassfish.org (450 000 par mois) et du nombre d'affichage d'une page HTML lancée lorsque vous démarrez Glassfish via Netbeans, dans les 45 000. Dans un slide suivant on constate que Glassfish est beaucoup plus téléchargé que JBoss Application Server. Au final peu importe les downloads, Glassfish intéresse la communauté et représente en France aussi des projets majeurs comme les sites Internet de RTL, de Fun Radio et aussi SFR. Glassfish v2 est en production sur des sites Internet majeurs en France. L'objectif de Glassfish V3 est de proposer un serveur d'app très léger. Cela ne veut pas dire forcément Tomcat, c'est surtout le souci du temps de démarrage. Une personne dans la soirée me confie qu'il a fait des tests avec la même application sur JBoss 5, Websphere 7 et Glassfish V3. JBoss demande 30 secondes pour démarrer, Websphere 7 demande 2 mn et Glassfish à peine 3 secondes... Un serveur léger c'est donc un serveur qui démarre rapidement et qui ne chargera ensuite le reste de ses modules que lors de la première utilisation. Pour cela, Glassfish est basé sur un noyau OSGI (Apache Felix). Je note sur mon petit cahier qu'Alexis insiste sur un concept de Glassfish : pas d'OSGI pour le développeur final. Et je vais vous donner mon avis : je suis d'accord. Aujourd'hui il faut arrêter de raconter aux développeurs que les gens des équipes de production vont arrêter une "partie" de l'application... Avec le foutoire d'une application J2EE 1.4, je peux vous dire que les équipes de production se sont adaptés à ce système, et que les mises en production ne s'effectuent pas avec de l'OSGI. Je n'y crois pas pour l'instant. Je crois à OSGI pour simplifier la récupération des dépendances, comme le système de Spring qui est élégant. Je n'arrive pas à croire au machin qui relance une partie de mon application... Pour continuer, Alexis explique ensuite que Glassfish V3 propose des modules afin de supporter par exemple Grails/Groovy ou encore JRuby/Rails. Et il insiste aujourd'hui sur les problèmes que rencontrent les développeurs Rails face à un Apache. Il explique que Glassfish est une réponse industrielle au problème du déploiement des applications Rails. Un seul process, un moteur de gestion des Threads et des entrées/sorties basé sur Grizzly, Glassfish enfonce Apache ou d'autres serveurs proposant le support de Rails. Parmi l'offre de Prelude, la console d'administration m'a semblé simple et claire. A comparer à la console de Websphere. La première fois que j'ai utilisé Websphere 6.1, après avoir lu les éléments de la page d'administration j'avais oublié pourquoi je m'étais logué sur cette page... Pour avoir mal au crâne, y'a pas mieux. Du côté de Glassfish on voit qu'un effort de simplification a été fait. Glassfish c'est

aussi et surtout l'Update Center. Ce système permet d'installer des modules supplémentaires dans Glassfish, comme la gestion des packages sur une distribution Linux. Très pratique pour Grails, Alexis explique que ce système permettra de mettre à jour son serveur Glassfish au fur et à mesure de la sortie des modules certifiés de JEE6. Enfin un mot rapide sur la version "embedded" de Glassfish, elle permet de faire des tests unitaires mais aussi d'embarquer un serveur léger dans une application java web start... Cependant il n'y a pas de support d'OSGI. Les démonstrations d'Alexis nous ont montré les concepts de rechargement à chaud de Glassfish ainsi que les temps de démarrage. Tout d'abord le temps de démarrage. Souvenir pour moi : en juillet 2007 à Noisy-le-Grand, je travaille 2 semaines sur le portage vers WebSphere 6.1 de notre framework pour un client... qui ne signera pas au final. Entre chaque déploiement, j'avais 4mn de temps libre. Du coup j'avais le temps de descendre prendre un café, parler avec le chef produit, regarder mes mails, et enfin voir mon serveur se planter comme une m... car il nous manquait un JAR. Cher lecteur, avec Glassfish, terminé la pause café et la discussion avec l'hôtesse de l'accueil : à peine 2 secondes pour démarrer... et encore sur un ordinateur portable... Dingue. L'explication de ce démarrage rapide est qu'il y a moins de code inutile et de recherche de dépendances pour s'injecter furieusement un Bean comme un drogué... A croire que l'injection de dépendance est devenu maintenant le shoot de votre code... Personne ne parle des applications avec 200 beans qui mettent 2 minutes à démarrer, comme chez BilletTel... Moi je vous en parle. Glassfish démarre tranquillement ce dont vous avez besoin, ni plus, ni moins. Effet numéro 2 : Alexis modifie du code Java dans une Servlet, effectue un joli Ctrl+S et un F5 dans Safari... hop la page s'est rechargée... Et je vous parle d'une Servlet, pas d'une JSP ! Et j'ajoute que Glassfish conserve la session de l'utilisateur, qu'il n'est pas nécessaire de redémarrer le serveur... qui de toutes les façons démarre rapidement. Si nous ne devons pas relancer le serveur, à quoi cela sert-il qu'il démarre rapidement ? En conclusion, j'ai vraiment aimé cette présentation de Glassfish. J'avais zappé ces sujets faute de temps à Devvxx, et c'est donc avec un oeil neuf que j'ai assisté ce soir aux présentations. Antonio nous a donné un état de l'avancement de JEE6 qui donne cependant à réfléchir, sur ce qu'il sera possible d'utiliser prochainement ou pas. Alexis a présenté Glassfish, un serveur que vous auriez tort d'ignorer, ou tout du moins à ne pas regarder. Prochain rendez-vous exceptionnel le 10 février à la FIAP afin de fêter les 1 an du Paris JUG avec la venue des JUG leader de toute la France (Tours, Nantes, Bordeaux, Nancy, Nice...) ainsi que la venue de l'organisateur de Devvxx qui présentera son site <http://www.parleys.com> sur lequel vous pouvez retrouver les conférences de Devvxx. Je vous parlerai demain du grand tableau blanc sur lequel chacun a voté, les résultats attendront demain... [Note] billet corrigé ce matin, merci à Emmanuel Servent pour sa relecture. J'ai terminé tard cette nuit. **Références:** Le site du Paris Java User Group : <http://www.parisjug.org/> Le site de Glassfish : <https://glassfish.dev.java.net/> Le blog d'Alexis <http://blogs.sun.com/alexismp> Le blog d'Antonio <http://www.antonioconcalves.org/> Cocktail le 5 février organisé par SUN Microsystems de 17h30 à 19h00 chez Demos (<http://www.demos.fr/actualites/Pages/cocktail-info-java-glassfish.aspx>) avec séance de dédicace par Antonio, 50 livres JEE5 à gagner, inscription obligatoire sur le site de demos.

## Paris JUG janvier 2009 : tableau blanc

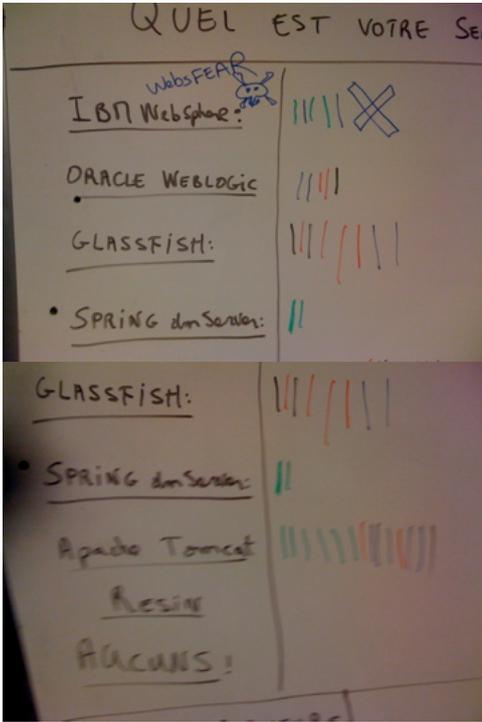
Hier soir sur une idée vue à Devvxx, j'ai proposé aux personnes présentes à la soirée du Paris JUG d'utiliser un tableau blanc au niveau du buffet afin que les quelques 210 personnes présentes puisse voter sur 2 questions autour de Java. Les 2 questions proposées étaient : - "Quel est votre serveur d'application favori ?" - "En dehors de Java, quel langage utilisez-vous ou aimez-vous ?"



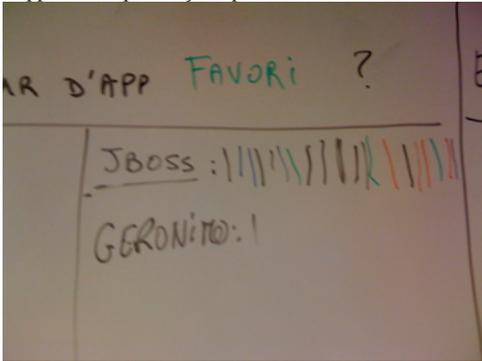
Les participants ont voté. Sur cette photo de gauche à droite, un monsieur avec un chapeau, [Noël Rocher](#) (JBoss-chemise blanche), chemise noir en retrait la personne qui va présenter Wicket en février au Paris JUG et enfin à droite, [Erwan Alliaume](#), l'un des gros contributeurs du blog de Xebia avec Cyrille Leclerc.



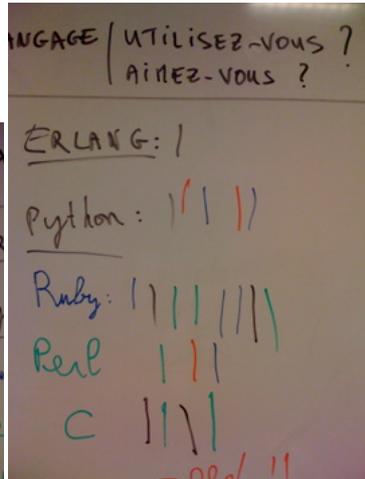
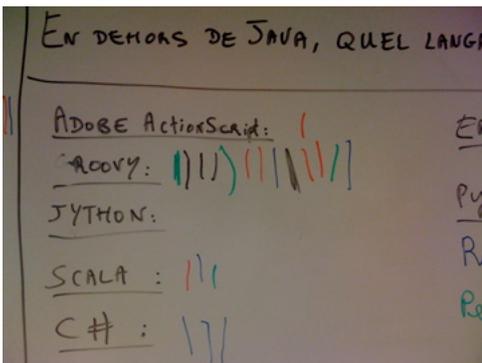
Au moment où tout le monde est remonté afin d'assister à la deuxième partie de la présentation, j'ai pris quelques photos des résultats que voici :

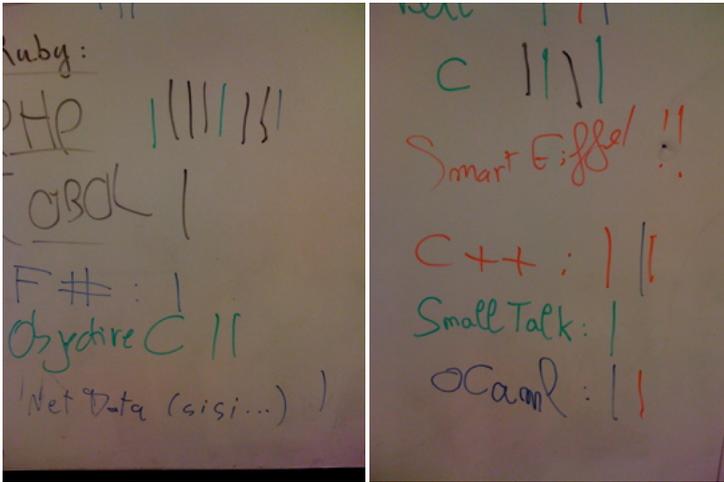


JBoss Application Server est le serveur d'application qui a reçu le plus de vote avec 20 voix :

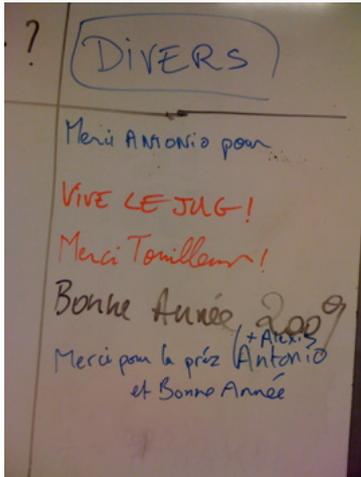


En dehors de Java, les langages utilisés ou qui intéressent les personnes présentes au Paris JUG ce soir là :





Enfin dans la



colonne divers :

**La synthèse des serveurs**

**d'application favoris :** JBoss Application Server : 20 - Apache Tomcat : 14 - Glassfish : 8 - IBM Websphere : 5 - Oracle Weblogic : 5 - Jetty : 2 - Spring dmServer : 2 - Geronimo : 1 - Jonas : 1 - .NET COM 1 Du côté des langages, le plus cité est Groovy : - Groovy : 13 - PHP : 8 - Ruby : 8 - Python : 5 - Langage C : 4 - Scala : 3 - C# : 3 - Perl : 3 - C++ : 3 - Objective C : 2 - Smart Eiffel : 2 - OCaml (corrigé au lieu de OCamel) : 2 - Cobol : 1 - SmallTalk : 1 - Erlang : 1 - F# : 1 - .Net Data (si... si...) 1 - Adobe ActionScript : 1 - Jython : 0 Merci à tous, et pour février, vous pouvez proposer ci-dessous dans les commentaires des idées sérieuses ou loufoques afin de refaire la même expérience.

## Soirée mercredi 21 janvier chez Zenika sur Lean

Rappel pour les personnes intéressées par les méthodes Agile, Scrum, le Lean development : une soirée sur ce thème est organisée la semaine prochaine par le cabinet de conseil Zenika. Le présentateur sera [Pascal Van Cauwenberghe](#). Rendez-vous à partir de 19h00 dans Paris au 54 rue Laffitte dans le 9<sup>ème</sup> arrondissement, l'inscription se fait sur le site de Zenika. J'en profite pour vous annoncer que le 14 avril prochain je présenterai Scrum avec Eric Mignot de Pyxis au Paris JUG, le 10 mars ce sera une soirée sur le Web et le 10 février, anniversaire du Paris JUG avec 3h de conférences au programme !!! le tout gratuitement ! A mercredi !

## Présentation de Lean chez Zenika

Zenika organisait ce soir une soirée de présentation sur les concepts et la méthodologie **Lean**. En quelques mots, Zenika c'est 16 personnes, des consultants Java/J2EE et surtout, un nom qui cherche ses racines dans les concepts de Lean. En effet Zenika est l'anagramme de Kaizen. "Kai" signifie "changement" et "Zen" signifie "bon". J'ai croisé quelques têtes connues, [Olivier Croisier](#) qui a rejoint Zenika à la fin de l'année 2008, [Thomas Queste](#), [Florent Ramière](#), Thomas Parle, [Pascal Thivent](#), et d'autres personnes du French Scrum User Group... Bref environ 110 personnes, ce qui représente une belle assemblée pour écouter [Pascal Van Cauwenberghe](#), consultant Agile basé à Bruxelles et fondateur des XP Days Benelux. La méthode Lean tire ses racines du constructeur automobile Toyota. Celui-ci applique des principes depuis une dizaine d'années, ce que l'on appelle **The Toyota Way**. Pascal nous propose simplement de dérouler et d'expliquer les 14 principes clés de Lean... Là je te sens sceptique lecteur. Tu te dis : super, on part pour 14 chapitres, je vais m'arrêter là. Grave erreur. Reste jusqu'à la fin, tu ne seras pas déçu. Sur les conseils de Thomas, j'ai bien saupoudré cet article d'exemple et de petites phrases afin de faire fonctionner tes muscles du sourire, et que le message passe, comme Pascal l'a fait ce soir.

## 14 points pour faire simple

Les principes de la méthode se découpent en 4 chapitres : - les processus - les personnes - l'amélioration - la philosophie

### 1) Les processus

Dans un premier temps, Pascal commence par expliquer que les bases de Lean sont de définir des processus exacts et justes, afin de produire le résultat attendu. Si votre développement dans votre équipe n'est pas guidé par des règles, le résultat ne peut pas être à la hauteur des espérances.

#### 1.1 Le débit

Visualisez votre méthode de production de logiciel. Retirez les temps d'attentes et de latences et donnez un rythme, des repères de temps. L'idée est de créer un mouvement, de donner des contraintes de temps. La sortie d'une nouvelle version prend 4 heures. Je passe 1h pour créer une branche, 2h pour le packaging et les tests, 1h pour l'installation. Je gagne du temps en m'assurant que les problèmes surgissent très rapidement afin de les résoudre. J'essaie d'éviter le "Muda" ou gachis. Pour cela je réduis l'attente, le transport d'un point à un autre et les mouvements excessifs. En clair : je supprime les étapes qui dépendent lors de ma release d'une approbation d'un email. Je réduis le transport en supprimant un upload FTP et en compilant le produit sur la machine d'intégration. Je supprime les mouvements excessifs en mettant en place des scripts d'intégration afin d'éviter de devoir faire trop d'opérations pour sortir ma release... L'objectif est de réduire le temps d'attente entre la commande du client et la livraison de sa demande. Cycle itératif court, livraison au fil de l'eau si possible plutôt que par paquet de 10... Pourquoi pas ?

#### 1.2 Basculer vers un système où le client "tire" la production

L'idée est de produire (donc de développer) uniquement selon la demande des clients. Pascal utilise l'image d'un rayon de supermarché : tant que le client n'a pas pris d'articles sur le rayon, on ne produit plus. Dès qu'il "consomme" un article, on lance alors l'approvisionnement. Rapporté à mon travail à la BNP, cela revient donc à dire que tant que l'utilisateur final n'a pas validé, commencé à utiliser ma superbe fonction, je ne m'embarque pas dans des développements. Cela a du sens. On évite ainsi la surproduction, le gâchis, et surtout d'avoir des fonctions jamais utilisées dans un logiciel. Pascal parle alors des cartes Kanban qui sont utilisés par Toyota afin de suivre la production. [Voir aussi un article écrit l'an passé sur le sujet.](#)

#### 1.3 Lissez la charge de travail/la production dans le temps

Appelé "Heijunka" en japonais, ce concept vise à éviter l'épuisement des personnes ("Muri") en s'assurant que la charge de travail soit constante, avec un rythme soutenu mais soutenable. Lorsque les équipes sont surchargées, elles peuvent certes répondre très ponctuellement à une demande. Si la situation perdure, on rentre alors dans un cycle d'épuisement, de démotivation et la chute est encore plus brutale. Pascal raconte l'histoire d'un développeur qui refuse de quitter son poste. Le premier soir, il lui demande de partir, le développeur répond qu'il doit encore terminer une tâche. Le deuxième soir, même chose. Le troisième soir, le développeur est encore là. Le lendemain, le coach le rencontre en entretien, et se rend compte alors d'un problème dans le logiciel dont le développeur ne souhaitait pas parler. Il est aussi important d'éviter le "Mura". En Français : coup de bourre. Dans la mesure du possible, si le cycle de travail de la personne n'est pas lissé, celle-ci va alors s'épuiser très rapidement. De plus, les temps d'arrêt et de redémarrage coutent chers. Amusez vous à interrompre un développeur toutes les 30 minutes, vous verrez à la fin de la journée la tête du bonhomme.

#### 1.4 Mettez en place une culture pour arrêter de résoudre les problèmes, car ceux-ci n'arrivent pas

Oooh que j'aime cette idée. Au lieu d'utiliser de l'énergie pour résoudre un problème, arrangez-vous pour qu'il n'arrive pas. Pascal explique qu'en 1924, un inventeur Japonais mets au point le premier métier à tisser qui s'arrête en cas de problèmes. Avant cela, il fallait employer à plein temps une personne pour surveiller le métier à tisser et dire "... jusqu'ici tout va bien" afin d'arrêter la machine en cas d'erreur... La qualité n'est pas une option. Les tests unitaires ne sont pas un gadget, ce n'est pas négociable. Pensez votre logiciel afin qu'en cas d'erreur, celui-ci soit capable de se rendre compte que quelque chose cloche et qu'il s'arrête d'envoyer des messages JMS par exemple... Mettez une grosse lumière rouge dans votre open-space. Lorsque l'intégration continue explose, il faut que toute l'équipe se mobilise, il faut que le leader en soit informé. Pensez à réduire le nombre de fonctions que vous livrez, afin d'en livrer moins mais de meilleures qualités. Vous gagnerez du temps, et finalement vous pourrez en livrer plus, contrairement à ce que vous pensez au départ.

#### 1.5 Les tâches standardisées mais avec de l'humain

Toyota utilise des robots pour construire ses voitures. Les robots sont utilisés pour les tâches pénibles et dangereuses, mais pas pour tout. Un humain a un cerveau, il sait s'adapter. Il faut donc conserver le pouvoir et le donner aux employés, au développeur. Enfin quelque soit le standard, celui-ci doit être amélioré en permanence. La personne qui fait de la production a le droit et le devoir d'améliorer le processus, chaque jour. Le gachis est de ne pas utiliser l'énergie des gens. Pour cette raison, Toyota a mis en place des boîtes à idées et récompense les ouvriers qui ont des idées, ce qui est le cas de tout le monde. Les experts sont les ouvriers, pas les managers.

## 1.6 Utilisez le contrôle visuel, afin que les problèmes ne soient pas cachés

Le principe ici est de mettre en place des indicateurs que toute l'équipe voit, ainsi que les managers et les clients. Concrètement, mettez au mur une feuille blanche, prenez des post-its de couleur, mettez une colonne "incidents en cours" ou "bugs urgent" et marquez les problèmes sur ces post-its. Oui vous pensez faire des loisirs créatifs, mais réfléchissez à l'utilité de ce genre d'indicateur. Un tableau se voit tout le temps. Les post-its représentent une quantité de problèmes à résoudre. Les clients voient au mur que votre équipe est surchargée. Vous pouvez aussi faire un suivi du temps de résolution. Il suffit de déplacer le post-it dans une colonne "Terminé" et d'indiquer le temps perdu sur la tâche. Pascal va encore plus loin en montrant la photo d'un urinoir. Au dessus de celui-ci on peut lire "11 bugs de Sev1, je sais que j'ai ton attention, alors au boulot!". Pas mal non ? Les indicateurs visuels chez Toyota c'est un panneau lumineux placé sur la chaîne qui indique aussi les facteurs de productivité, les objectifs à atteindre, les problèmes en cours...

## 1.7 Utilisez des technologies fiables, éprouvées et testées qui aident vos clients et vos développeurs

Certainement le point qui fait le plus mal... Arrêtons 5 minutes avec nos frameworks de double-inversion de contrôle où tu configures en XML ce que tu faisais en Java... Revenons sur Terre 2 minutes, prenons des outils qui ont fait leurs preuves et arrêtons de nous faire plaisir, avec ce framework "jante alliage" qui aura disparu dans 6 mois, ou cette superbe librairie de parsing pour lire un fichier texte de 3 lignes... Il faut stimuler la recherche selon 3 étapes, de la plus facile à mettre en place à la plus révolutionnaire : - conserver la solution existante - améliorer par incrément - faire une percée radicale Il y a souvent peu de problèmes de personnes, bien plus souvent un problème de processus. N'hésitez pas à investir du temps pour réaliser un prototype si vous souhaitez prendre une nouvelle technologie. Rejetez aussi les technologies qui ne sont pas compatibles avec votre culture, et encouragez les développeurs à toujours regarder ce qu'il se passe sur le marché, afin d'éviter de tomber dans un cul-de-sac technique... Et surtout, pensez à vous outiller. Je vous parlerai prochainement de SpringFuse, le produit de Jaxio. Il est temps d'arrêter de jouer aux Legos et de passer à des jeux un peu plus industriels.

## 2) Les personnes

La force d'une entreprise c'est son patrimoine humain. Sortons un instant de Lean. Regardez les cabinets de consultants sur Paris. Ce qui fait la valeur d'une entreprise c'est ses équipes. Pas son logiciel, pas ses méthodes. Non, juste ses équipes. Le secret des cabinets de consultant (SSII) qui marchent, c'est l'investissement dans les valeurs humaines. Le ticket d'entrée est plus cher, le retour sur investissement est bien plus important.

### 2.1 Fait grandir des leaders

Pascal explique que l'idée est de faire grandir des leaders en prenant des personnes du vivier de l'entreprise. Rien de pire qu'un manager parachuté à la tête d'une équipe d'après son expérience. Il explique que les leaders doivent servir de modèle pour les arrivants, et qu'un bon leader doit connaître les processus de son équipe en les ayant pratiqués. J'ai eu en tête le responsable du MacDo à côté de chez moi. Le gars a commencé il y a 7 ans comme équipier. Il gère aujourd'hui une équipe de 21 personnes... Et il n'a que 26 ans...

### 2.2 Développe des gens exceptionnels et des équipes qui suivent la philosophie de ton entreprise

Pascal explique qu'il est plutôt souhaitable de faire en sorte que les équipiers ne soient pas trop spécialisés dans une équipe de développeur. Pour cela, il faut mettre en place une dynamique et faire en sorte que le brassage des développements soit une réalité. Je suis un peu plus sceptique sur ce point, mais je comprends l'idée.

### 2.3 Le respect est une valeur essentielle

Tout d'abord respectez vos clients, mais aussi vos fournisseurs. Toyota rapproche géographiquement ses fournisseurs et montre toujours un respect important. L'exemple que Pascal utilise est le suivant : un constructeur automobile va voir un fournisseur de moteur. Il demande 5% de réduction, sinon c'est la fin du contrat... Aucuns respects du client. Toyota va voir le même fournisseur et propose de prêter 3 de ses ingénieurs afin de réduire les coûts de développement de 10%, et donc d'obtenir 5% de réduction aussi. Sauf que Toyota ne sera alors que le seul à bénéficier de cette réduction. Respect et intelligence...

## 3) L'amélioration

L'amélioration en permanence nous aide à résoudre les problèmes au fur et à mesure. En s'introspectant après chaque itération, une équipe de développement qui fait du Scrum s'autocritique et s'améliore. Pascal insiste sur le fait que l'introspection est peut-être le concept le plus important de Lean. Je pense à cet instant dans ma tête que chez mon client en ce moment, c'est le point que nous ne traitons pas. Il est temps de remettre en place ce processus d'urgence.

### 3.1 Allez voir par soi-même les sources des problèmes "Genchi Genbutsu"

Pour améliorer la production, Toyota préconise aux managers de descendre et d'aller sur la chaîne afin d'observer et de regarder en situation ce qu'il se passe. Il faut que le manager aille voir la

production et qu'il identifie les problèmes. Pascal raconte l'histoire de la Poste Belge qui perdait du temps pour traiter l'envoi de colis internationaux à cause d'un mur (un vrai avec des briques) entre le quai de déchargement des camions et la pesée, obligatoire pour les envois et la facturation. En détruisant le mur, la productivité a augmenté. Il faut que votre chef passe un peu de temps avec vous afin de comprendre ce qu'il se passe.

### 3.2 Prenez des décisions avec un consensus et le plus tard possible

Dans un premier temps, listez toutes les options possibles. La Toyota Prius avait ainsi 80 projets de moteur différent au départ. Dessinez un rétro-planning en partant de la date de sortie souhaitée. Revenez en arrière et représentez à chaque fois la date limite où il faudra prendre une décision (Spring ou EJB 3.1 ?). Ne prenez la décision qu'au dernier moment, pas au début. En cas de problèmes, discutez avec toutes les personnes impliquées afin de prendre l'avis de tout le monde, et qu'une énergie commune fasse que les choses changent ("Nemawashi"). Le consensus prend peut-être du temps mais vous donnera un résultat de meilleure qualité. Le planning poker pour estimer le temps de développement d'une tâche est un exemple de consensus.

### 3.3 Réflexions et améliorations

La réflexion ("Hansei") et l'amélioration continue ("Kaizen") sont deux facteurs clés de la méthode Lean. Préparez des processus métiers qui ne demandent pas beaucoup de travail d'inventaire. Cela forcera les équipes à s'améliorer en permanence. Lorsqu'un process est stable mais qu'un problème survient, pensez à demander 5 fois "Pourquoi" à votre développeur. - "Je ne peux pas corriger ce problème car je n'ai pas le temps" - Pourquoi n'as-tu pas le temps ? - Je suis occupé par une tâche de production que je dois faire, je dois envoyer un nouveau mot de passe à un utilisateur - Pourquoi dois-tu envoyer un nouveau mot de passe ? - Parce que le système ne sait pas le faire - Pourquoi ne sait-il pas le faire ? - Nous n'avons pas développé de fonction "me renvoyer un nouveau mot de passe" sur la page d'accueil - Pourquoi ? - Parce que nous avons oublié de le marquer dans le Product Backlog et que nous n'avons pas écouté le chef de projet. La prochaine fois nous le marquerons dans le product backlog et nous ferons en sorte de l'implémenter.

## 4) La philosophie

Pascal Van Cauwenberghe termine par le point le plus important pour lui : la philosophie à long terme. **Basez vos décisions de manager sur du long terme, même si le coût à payer à court terme est important.** Pour cela il évoque Toyota qui fait la promesse d'offrir un emploi à vie à ses salariés. En raison de la crise il se demande si cette philosophie pourra perdurer. Il explique que Toyota en accord avec ce principe, tente de former les gens en ce moment au lieu de les faire travailler sur des chaînes, afin que lorsque la crise sera passée, ils soient encore meilleurs... C'est beau...

## Conclusion et réflexions

Après vous avoir présenté Lean tel que je l'ai vu ce soir, j'ai quelques réflexions. Tout d'abord je visualise maintenant mieux comment une méthode comme Lean est compatible avec Scrum ou XP. Pour tout vous dire, je vois Lean comme des jalons et des repères pour travailler mieux. Ensuite Scrum apporte un framework qui aide à piloter le cycle de développement d'un logiciel. Enfin XP est pour moi maintenant dans les habitudes des développeurs, et je pense par contre que vouloir caler ses 13 principes reste assez difficile. Donc on a un escalier avec 3 marches, la marche la plus haute étant XP, et la première étant Lean. Qu'en pensez-vous ? Merci à Zenika qui a organisé une soirée très riche. A la sortie, remise d'un résumé de Lean, apéritif, accueil sympathique.. rien à redire. Bravo à eux. Zenika est sponsor du Paris Java User Group, ainsi que sponsor des [XP Days](#) le 25 et 26 mai 2009. Enfin j'ai appris aussi comme Olivier un peu au dernier moment que le **3ème Java Barcamp aura lieu samedi 31 janvier** toute la journée, cette fois-ci organisé par Eric Lefevre Ardant et Anthony Dahanne de Valtech. Plus d'information : <http://barcamp.org/JavaCampParis3>. Le BarCamp se déroulera le samedi chez SUN au Sun Customer Briefing Center, 42 avenue de Iéna, 75016 Paris. Superbes locaux où j'ai travaillé deux mois pour une mission (soupir...). Moi je suis déjà pris, donc ce sera sans moi.

## eXo Platform : rencontre avec Benjamin Mestrallet

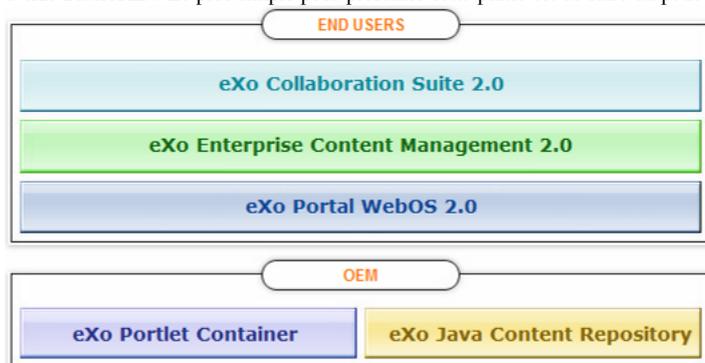


*J'ai eu l'occasion d'interviewer Benjamin Mestrallet, PDG de la*

société [eXo Platform](#) grâce à Dimitri Baeli. Editeur de logiciel français, eXo Platform est le leader des solutions de portails basées sur la JSR 168 et JSR 286. Les solutions disponibles en open-source, sont aussi livrées sous la forme de produits avec licences. eXo est un acteur très actif de la communauté Java. Au delà du socle technique dont nous allons parler, il est intéressant de comprendre la démarche et la vision d'eXo. Nous avons discuté pendant deux heures. J'ai vraiment apprécié notre rencontre, Benjamin ayant une vision très solide du futur, et surtout très réaliste. C'est ce que nous allons voir dans cet entretien. **Pour commencer, l'actualité d'eXo Platform a été riche en cette fin d'années. Beaucoup d'actualités, de voyages comme lu sur ton feed twitter (@benjmestrallet). En quelques mots peux-tu me parler de l'entreprise et de tes équipes ?** En effet, 2008 a été une année riche. L'année 2009 sera l'année de la stabilisation. 2008 a été l'année du lancement de beaucoup de produits et nous n'annonçons pas pour l'instant de nouveaux produits pour 2009. Concernant l'entreprise, en effet nous avons aussi recruté beaucoup de personnes. La société se compose de plusieurs centres de développement : le Viet-nam, l'Ukraine, la Tunisie et la France. eXo Platform représente environ 95 personnes aujourd'hui. Si tout va bien nous devrions démarrer aussi aux USA avec un bureau là-bas. **A propos du nom "eXo Platform", j'ai entendu plusieurs histoires, mais donc il semble que cela vienne du nom de ton projet de fin d'étude, tout simplement appelé "exo" comme exercice, et que finalement tu aurais gardé. Vrai ou pas vrai ? Légende urbaine ou pas ?** C'est vrai. **Comment l'aventure a commencé ?** Nous avons publié un article sur The Server Side à propos des portails. Cet article a été repéré par un membre du Department of Defense en 2003, l'équivalent de la DGA en France, au moment de la guerre en Irak. De fil en aiguille, en effet notre premier client a été le DoD. J'avoue que la première fois qu'un email avec "@jfc.com" arrive dans ta boîte aux lettres, tu n'y crois pas... Le code source a été placé sur SourceForge, c'était l'implémentation de la JSR 168, basée sur Pico Container, un des premiers frameworks d'inversion de contrôle. Nous avons été contacté afin d'écrire plusieurs articles sur les portails collaboratifs, et c'est de cette façon que notre premier client a démarré l'aventure eXo Platform, courant 2004. **La société a été créée à Vannes ?** En fait j'ai commencé à Paris, en banlieue parisienne. **... dans un garage ? Important le garage !** \*rires\*, oui le garage en effet. Nous avons commencé ensuite à Vannes, cela nous a donné la possibilité de démarrer rapidement, entre autre grâce à François Moron, qui est le directeur des opérations aujourd'hui chez eXo. **Comment se passe la répartition des équipes entre les différents centres de R&D ?** Nous sommes répartis par couche technique. L'implémentation de spécification se fait plutôt en Ukraine, le Viet-Nam se concentrant sur l'intégration et la réalisation des produits. La Tunisie démarre afin d'offrir un support francophone et surtout, le support des écritures right-to-left comme l'arabe, l'hébreu et le persan. Nous souhaitons démarrer très rapidement de nouveaux projets en Afrique du Nord et au Moyen-Orient. Le support des utilisateurs francophones en Tunisie nous permet aussi de renforcer nos équipes et d'offrir un service de qualité à nos clients en Europe. A propos nous recrutons en Tunisie, et nous cherchons des personnes motivées afin de renforcer nos équipes à Tunis (voir le site <http://www.exoplatform.com>). Si des personnes sont intéressées, elles peuvent postuler via le site Web d'eXo. **eXo Platform fait partie du JCP (Java Community Process) et vous intervenez sur la JSR 286, Portlet 2.0. Qui est en charge chez eXo de la JSR ?** Au départ c'était moi. A cette occasion j'ai fait la connaissance de Julien Viet. Il était à l'époque chef de projet chez RedHat de JBoss Portal. De fil en aiguille, Julien nous a rejoint et il est en charge aujourd'hui du produit [eXo Portal 2.5](#). **Par exemple, où travaillait Tugdual Grall avant de rejoindre eXo ?** Chez Oracle à San Francisco, pendant une dizaine d'années. Il est aujourd'hui VP of Business and Product Strategy chez nous. **Il y a pas mal de personnes connues de la communauté Java dans les membres d'eXo Platform non ?** Dimitri Baeli, OSSGTP, anciennement chez Prima Solutions nous a rejoint récemment. Il a mis en place Scrum et il travaille sur l'industrialisation de nos produits. Ensuite Brice Revenant dirige eXo Platform SEA (South East Asia). Il est basé au Viet-Nam, il travaillait anciennement chez Bull. Patrice Lamarque est Product Manager, il est en charge [d'eXo Collaboration Suite](#), notre suite collaborative avec email, agenda, carnet d'adresse. Il y aura d'autres personnes prochainement... **eXo Platform fait partie du consortium OW2, peux-tu expliquer ce que cela représente et comment vous avez rejoins ce groupe ?** OW2 est au départ une initiative de France Telecom, de Bull et de l'INRIA, visant à promouvoir les projets open-sources. Cependant aujourd'hui c'est un consortium bien plus important, l'Asie et le reste de l'Europe sont fortement représentés. Au départ, l'objectif est de fédérer des projets autour du middleware, plutôt en licence LGPL. Lorsque nous avons commencé à travailler avec OW2, nous avons proposé eXo qui est en double licence : GPL et commerciale. D'autre part nous sommes un socle applicatif. Aujourd'hui les projets regroupés autour d'OW2 sont plus vastes. On retrouve par exemple d'autres projets comme [xWiki](#), initié par Ludovic Dubost ou [elastic Grid](#) initié par Jérôme Bernard. **A propos des évolutions à venir d'eXo Platform, j'ai lu que tu parles d'un mode hébergé (SaaS) et d'un mode déconnecté. Qu'en est-il aujourd'hui ?** Oui à propos du mode hébergé, nous sommes utilisateurs d'Amazon S3 et EC2. Pour l'instant nous n'avons pas encore finalisé de produits. Notre idée est de proposer ce que l'on appelle "Paas" : Platform as a Service. Nous travaillons aujourd'hui avec des intégrateurs qui réalisent des projets chez des grands comptes. Notre souhait est d'offrir directement aux PME une solution locative et donc d'héberger leurs applications sur notre plate-forme. Ainsi, chacun se concentre sur son cœur de métier et ne prend pas en charge la gestion d'un data center par exemple. Sur ce modèle, Salesforce.com propose ce service, nous souhaitons aussi travailler dans ce sens. C'est un mouvement fort pour les années à venir. A la base notre offre vise à simplifier chez les grands comptes l'usage des familles de produits sur lesquels nous sommes positionnés. Là où il y a un IBM Websphere Portal, notre offre est plus compétitive. Nous avons fait un gros travail pour supprimer la segmentation du marché (Websphere Portal, FileNet, Lotus...) et d'offrir un socle commun plus puissant. Amazon SimpleDB, S3, EC2, le Cloud computing sont des sujets sur lesquels nous travaillons. Faire fonctionner une application J2EE en cluster demande un travail très intéressant. Il n'y a pas de multi-cast dans le cloud computing. Ce sont vraiment des sujets intéressants. **Je me souviens que lors de la précédente crise dans les années 2000, Linux a littéralement explosé. Face aux réductions de budgets, les DSI ont propulsé Linux. Surtout sur les marchés publics. Que penses-tu par rapport à cette crise ?** On sent un mouvement global, la crise va-t-elle privilégier l'open-source ? et d'autre part, est-ce que les choix techniques que nous avons fait sont les meilleurs ? Sur la première question, ce qui est intéressant c'est que les gens se posent plus

rapidement la question de "est-ce que c'est moins cher ?". C'est la première question. Ce n'est pas gratuit car c'est open-source, ce que les décideurs ont intégré. Alors quel est le prix ? On a vu que les couches basses ont percé dans le monde de l'open-source ces dernières années. Tout le monde est d'accord pour s'en servir. Je pense que l'on va remonter de ces couches basses vers les plateformes d'hébergement comme la nôtre, qui proposent des services à valeur ajoutée, comme Alfresco ou Liferay. Le pipe en ce moment est plus important, nous voyons que la crise provoque en ce moment un mouvement. Finalement en situation de danger, les décisions sont plus radicales, et nous sommes mieux positionnés. **Entre la France et l'export, comment est réparti l'activité ? le chiffre d'affaire ? as-tu des repères à nous donner ?** 35% du CA génère en 2008 s'est fait à l'export. En 2007 c'était 50/50. Notre chiffre d'affaire en 2008 a presque quadruplé en un an.

**Quelles sont vos sources de revenu ?** C'est de la souscription, et un pourcentage sur le support. Nous avons aussi des experts et des prestations de service. Finalement surtout de la formation, et des développements noyaux. Nos clients sont sponsors. **Quel est votre concurrent numéro 1 ? si nous étions dans la rue, est-ce que ce serait une petite épicerie ? un supermarché ? quel est sa taille par rapport à eXo ?** Cela dépend du marché, de la partie dans le monde dont nous parlons. Sur les appels pour marché public nous voyons souvent en concurrence des solutions proposées autour de [Liferay](#), [Alfresco](#), parfois du [Jahia](#). Sinon aussi IBM Websphere Portal et Oracle Portal. Nous voyons beaucoup de projets proposés sur Microsoft Office Sharepoint Server, surtout en Afrique du Nord. **J'ai vu dans la Finance aux USA que les solutions basées sur Microsoft avaient le vent en poupe, car ce sont des solutions qui partent d'une interface agréable, là où les solutions Java remontent des serveurs d'application vers l'interface utilisateur. JavaFX vient juste de sortir... On parlait RIA, GWT, Flex l'an passé comme d'une révolution... Pour revenir à eXo Platform. WebOS montre peut-être une piste pour le futur. Finalement, tu ne penses pas que Google sera ton plus gros compétiteur demain ?** Non nous ne sommes pas sur la même segmentation. Google est un partenaire avec lequel nous travaillons. Ils ont lancé le modèle "Gadget" et "OpenSocial" qui fédère beaucoup de sites (LinkedIn, MySpaces.com...). On a aussi un engouement justifié sur le développeur chez Google. Ils sont aussi fort en proposant des solutions qui ne sont pas uniquement basées sur Java mais sur Python, javascript, Groovy ou autre... Google Gadget est une solution agnostique de la partie serveur. Du côté du client, du html, du javascript. Et tu peux alors mettre ce que tu veux du côté du serveur, ce qui est vraiment intéressant car les perspectives sont plus larges. eXo Platform supporte l'intégration des Google Gadgets, ce qui nous ouvre aujourd'hui plus de 1000 composants que chacun peut placer sur son portail. On retrouve un mode client-serveur, le client communique avec le serveur avec du JSON. J'ai fait le tour de l'Asie avec Google en novembre dernier. Lors des conférences, une page pas très belle présentée par Google : résultat, 500 personnes dans la salle. Tu passes derrière et tu proposes une présentation très sympa ? 100 personnes ! C'est l'effet Google ! Le traiteur le nôtre pour eXo : 100 personnes. Des bières et des chips chez Google : 500 personnes... C'est Google \*pires\* **Pourtant tout ne marche pas si bien. Google Chrome est utilisé autant que Safari d'après des chiffres vus sur le site du NewYork Times je crois** Les gens qui utilisent un Mac utilisent Safari, il marche bien. Les iPhones, cela fait aussi un grand nombre de navigateur "Safari". Sur Google Android je crois que c'est Web Toolkit. Apple a vendu 17 millions d'iPhone... C'est énorme. **A propos donc de téléphone mobile, avez-vous des projets ou des prototypes ?** Il y a 3 usages différents : deux qui sont classiques, naviguer sur internet pour le premier, et faire de la synchronisation pour le deuxième. On étudie SyncML pour voir comment nous pourrions proposer la synchro entre notre suite collaborative et ton PDA. Le 3ème usage auquel nous pensons, c'est de proposer un "Application Store" pour proposer des gadgets et des outils. Le WebOS de Palm est très puissant et pourrait afficher le contenu de notre plate-forme. Goojet (<http://www.goojet.com/index.html>) est une société Toulousaine qui propose par exemple un système de gadgets et d'application mais pour téléphone portable. Présenter le portail ou le WebOS sur un petit écran, je ne sais pas. Par contre je pense que présenter une partie des gadgets de nos portails, cela serait possible... **Sur un autre sujet, tu as annoncé l'ouverture d'une filiale en Tunisie. Où en sommes-nous ?** Le marché de l'Afrique du Nord se développe à vitesse grand V. Nous avons la chance de parler français, nous sommes proches géographiquement. J'ai contacté un ami rencontré pendant mes études, Oualid Chaker. Ensemble nous avons monté ce projet et débuté fin juillet une activité de commercialisation et de support basée à Tunis, en charge des contacts avec les pays d'Afrique et du Moyen-Orient. **Peux-tu présenter les produits et le socle technique d'eXo Platform ?** Le plus simple pour présenter cette partie est de faire un petit schéma.



Au niveau du socle technique : 1) eXo Portlet Container est une implémentation des JSR 168 et JSR 286 (Portlet 2.0). Ce module gère le cycle de vie et les instances des composants de portlet. L'implémentation a été optimisée et fournit également des capacités de monitoring qui sont indispensables dans un environnement de production. 2) Les WebServices sont supportés avec une implémentation de la spécification JSR 311(JAX-RS) l'API pour les webservices de type REST. JSR 181 est une JSR pour les WebServices Metadata, qui te permet d'exposer via des annotations des web services type WSDL 3) eXo Java Content Repository. Il s'agit d'une implémentation de la JSR 170 et JSR 283 qui vise à rendre abstrait et à normaliser le stockage des données. Les dépôts de données sont

composés d'espaces de travail qui peuvent contenir du contenu dans une structure hiérarchique. Notre implémentation fournit des services pour gérer les versions, le verrouillage, la sécurité (LDAP), et les recherches. L'implémentation eXo permet également des connexions optimisées avec toutes les bases de données telles que Oracle, DB2, MySQL... On a donc 3 points à retenir. Exo Platform offre : - la portabilité des applications avec portlet - la portabilité des données avec le JCR - la portabilité de la logique métier avec JAX-RS et WS Metadata. **Wow.....** Le point essentiel est ensuite qu'on ajoute ensuite la notion de "Gadget serveur" à la Google. On peut soit les exposer pour nous, soit vers l'extérieur. On a là le socle portail d'eXo Platform. Nous stockons les préférences et les pages dans le JCR. Et on pense maintenant proposer un IDE pour que tu puisses développer en Groovy des applications hébergées dans la plate-forme... **... wow... wow..... et ensuite ?** On a ensuite travaillé sur la représentation et l'interface utilisateur. Tu as d'une part la vue classique, orienté Web avec le portail. Mais nous avons aussi la vue type "WebOS" qui t'offre un bureau avec ses applications, le tout déployé sur le même serveur. **... ah... wow...** Nos produits phares sont donc développés avec notre socle. - eXo ECM 2.2 (Enterprise Content Management) est notre offre de gestion de document avancée. ([voir sur le site pour plus de détails](#)) - eXo WCM 1.0 (Web Content Management) est un produit très intéressant de gestion de sites Internet. Il permet aux collaborateurs d'une entreprise d'avoir une vue synthétique pour gérer le contenu et les mises en ligne de sites Internet. - eXo CS 1.0 (Collaboration Suite) est un portail d'entreprise : agenda, courrier électronique, carnet d'adresse... nous proposons de regrouper dans un navigateur les informations de l'entreprise afin d'offrir un socle de collaboration et de partage très puissant. Il manque enfin le module transverse : une implémentation d'open-social, qui nous permet d'offrir une intégration avec les applications Web 2.0. Il sera possible d'exposer par exemple ton réseau d'ami... Selon les personnes avec lesquels on discute, nous présentons eXo Platform d'une manière ou d'une autre. En tant que technicien, tu as vu le socle technique, mais aussi notre chemin d'intégration, ce qui nous permet de travailler ensuite avec les clients et de réaliser des produits finis. **En tant que développeur, comment puis-je ensuite travailler avec vous afin de développer une solution sur votre socle ?** La notion de plate-forme te permet de rajouter tes propres composants. Tu as aussi un éditeur de gadget en ligne, ce qui ouvre le système et permet à chacun de contribuer à la plate-forme. **Mais est-ce suffisant pour développer une application ?** La partie serveur sera exposée avec REST, tu retrouves ton modèle classique, avec une application Java et celle-ci discutera via JSON par exemple, avec le gadget. **Les visiteurs sont aujourd'hui souvent sur des sites comme Facebook, MySpaces.com. Comment les faire venir ?** Justement, tu utiliseras la plate-forme d'hébergement eXo, tu pourras aussi gérer finement le cycle de vie de ton gadget, nous offrirons un service qui n'existe pas encore... Tes visiteurs seront sur leur page iGoogle avec une application exposée via OpenSocial par eXo Platform. Tu pourras donc par exemple accéder au gadget hébergé sur la plate-forme eXo... Nous offrons une version un peu plus entreprise pour gérer ce cycle de vie. C'est adapté aux entreprises qui cherchent à ouvrir leurs services. 600 millions de personnes sur OpenSocial, 180 millions chez Facebook... **Pour revenir à l'offre hébergée, je te présente du code normalisé pour qu'il fonctionne sur la plate-forme. Est-ce que je peux venir louer une partie de ta plate-forme ?** Pour tout ce qui est gadget, je pense que nous offrirons un point d'entrée et que tu pourras proposer tes gadgets. Les portlets par contre, sont du côté de ce que nos clients hébergent avec nos produits, et les utilisateurs de nos clients pourraient ajouter des gadgets afin d'ajouter de nouveaux services sur leur application. **Qu'en est-il du Java content repository ?** Du côté du JCR nous avons aussi des chemins encore à explorer. Aujourd'hui nous stockons les données soit dans une base relationnelle, soit sur le disque lorsque celles-ci sont trop grandes. Nous travaillons afin de remplacer cette base relationnelle par la solution type [Amazon SimpleDB](#). Cependant faire travailler ce type d'application dans un Cloud demande un travail assez important. Passionnant mais cela demande du travail. La version 1.11 du JCR devrait proposer en beta ces fonctionnalités. Donc à terme, Amazon EC2 pour la partie JVM, et Amazon DB à la place de clusters de MySQL. Nous serons alors prêt à segmenter pour que chaque société puisse alors utiliser une partie de nos produits. Cela à l'objectif des deux années à venir. **Il faut avoir aussi les investissements et les moyens. Par les temps qui courent, ne crains-tu pas que cela soit plus difficile ?** eXo Platform s'autofinance par son revenu, ce qui nous donne une liberté et donc une réactivité. Nos produits sont matures et sont sur les rails. Nos axes de dépense seront plutôt les efforts commerciaux, le travail que nous réalisons en ce moment pour aller à la rencontre de nos clients. Nous travaillons pour offrir une plate-forme de services et donc proposer à chaque client la même offre de support. Notre offre "Enterprise" répond à la gestion de versions, au cycle de vie de nos produits. **Dimitri m'a parlé aussi des méthodes de développement et de vos outils. Sans trop en dévoiler, vous êtes vraiment bien outillé et industrialisé. J'ai eu le sentiment en effet que les projets sont sur les rails, que les équipes produisent et travaillent avec une synergie dont on rêve parfois. Il manque encore des produits ?** Nous sommes open-source. Dès le départ nous avons eu besoin d'outils comme des wikis, les outils d'Atlassian, la gestion des bugs. Nous travaillons sur les outils de tests d'intégration, comme Selenium par exemple. Cependant, comme nous étions dès le départ open-source, les efforts sont logiques. Cela me fait aussi penser qu'il y a vraiment un potentiel pour, par exemple, proposer une plate-forme d'hébergement à des développeurs. J'imagine en effet que nous pourrions proposer d'héberger des projets, de les compiler, de les versionner... **Collabnet le fait déjà. Chez Reuters c'est la solution retenue pour les 3200 développeurs dans le monde. Il faudrait offrir un service SVN non ?** En fait nous avons travaillé avec le repository maven2 "Archiva". SVN fonctionne par dessus le protocole http, et notre JCR peut s'exposer avec des modules WebDAV. Il sera donc possible de le faire je pense. J'imagine mettre dans le JCR le code source d'une part, et les artifacts compilés d'autre part. A la fois un repo SVN et un repo Maven2 par exemple. On peut alors utiliser qu'un seul serveur au lieu de deux ! **...et je vais abonder dans ton sens, car aujourd'hui il est difficile de demander à des équipes MiddleWare d'héberger et de maintenir le repository SVN et le proxy Maven2. Cela pose des soucis de budget !** Ce n'est pas pour l'instant notre but, mais je pense que ce serait un produit que j'utiliserai. A réfléchir. Pour que cela soit intéressant il faut que l'automatisation soit poussée. **Pour boucler avec les questions du début, quels sont les clients en France qui utilisent eXo Platform ?** Je ne vais pas les citer tous, mais par exemple nous travaillons avec M6. Le dernier résultat dont je suis très fier c'est le support du texte "right-to-left" pour l'arabe, l'hébreu et le persan. C'est l'un des travaux de Julien Viet depuis qu'il nous a rejoint. **Pour terminer sur une note plus légère, quel usage de Twitter**

**fais-tu ?** Deux choses : où je suis, car je voyage pas mal. Et ce que je bois ;-) Julien Viet est un spécialiste des sushis par exemple. Twitter c'est un moyen déconnecté de rester informé de ce que chacun fait. J'utilisais plus twitter lorsqu'il y avait la notification par SMS. Je dois passer par le web. Cependant en Asie je me souviens que lorsque nous twitions sur #devfest en Asie, nous avons eu des journées énormes avec beaucoup de messages échangés. Twitter est déroutant. **Le mot de la fin ?** J'invite tout le monde à venir tester eXo, [des articles](#) sur internet vous donne un guide complet pour commencer rapidement à créer vos applications. eXo Platform peut très bien faire fonctionner par exemple une application en Flex, hébergée sous la forme d'une Portlet 2.0. A part un premier souci de z-index, nous n'avons pas eu de soucis. Il est possible de faire fonctionner des technologies comme GWT ou Flex avec eXo. **Merci Benjamin !** Merci Nicolas !

## SpringFuse beta 2 disponible

Jaxio annonce la sortie de la version beta de SpringFuse, la version hébergée du logiciel Celerio. SpringFuse est un générateur de projet qui vous fait gagner du temps. En partant de ce que nous avons tous dans nos application, une base de données, il vous permet en quelques minutes de générer du code. Jusqu'ici vous me dites "...comme AppFuse". Là où SpringFuse innove je trouve, c'est sur la qualité de ce qui est généré. Basé sur Spring MVC, Hibernate, SiteMesh, [la liste complète](#) de ce que vous recevez quelques minutes après avoir lancé la génération est intéressante. C'est surtout sur la qualité du code généré, qui permet de gagner du temps et de reprendre des parties de votre logiciel rapidement, en se concentrant ensuite sur le code métier. Il est possible de tester gratuitement le service, ce que j'ai fait par curiosité sur une base chez nous. J'ai pris une table avec 60 000 entrées, 14 colonnes, des données financières. Après 10 minutes en effet j'avais mon serveur jetty démarré, mes données affichées, et un code avec tests, documentation, configuration, et des exemples de pages pour continuer tout seul. Du côté des performances, le code généré par SpringFuse et déployé sur Jetty affiche plus vite les données dans un tableau simple que mon client TOAD ! Derrière c'est la qualité du code, la configuration du cache de deuxième niveau d'Hibernate et surtout un modèle de programmation qui reprend les principes vus dans les meilleurs livres. J'ai ainsi appris de nouvelles astuces en lisant simplement le code généré. Pour les vieux lecteurs du Touilleur (pas ceux qui n'ont plus de cheveux, ceux qui me lisent depuis un an) vous savez sans doute que je connais bien Florent Ramière et Nicolas Romanetti de Jaxio dont [je vous ai déjà parlé](#) sur le Touilleur. C'est donc en toute subjectivité (et pas objectivité) que j'ai testé l'outil. Je vous donne mon avis : pensez au temps gagné en utilisant un outil qui génère en 1 minute ce que vous pouvez écrire en une bonne semaine de travail... Testez-le, faites-vous votre avis, pensez-y pour faire du refactoring de votre logiciel. Vous avez une base de données ? vous êtes curieux et prêt à vous remettre en question ? Alors rendez-vous sur <http://www.springfuse.com>

## Paris JUG : déjà un an, venez le 10 février

**Événement de la communauté Java à ne pas manquer ce mois-ci sur Paris : le 1er anniversaire du Paris JUG.** Pour ceux qui ne connaissent pas le Paris JUG, le principe est de se retrouver gratuitement chaque deuxième mardi du mois afin d'assister à des présentations autour de Java. Depuis sa création l'an dernier par [Antonio Goncalves](#), [David Dewalle](#) et [Zouheir Cadi](#), le Paris JUG s'est fortement développé. C'est plus de 200 personnes par mois, des développeurs, des chefs de projets, des éditeurs, des chercheurs de têtes pour recruter, des journalistes, des geeks, des gens qui ne font PAS de Java mais qui viennent par curiosité... des lecteurs du Touilleur (environ 199 sur 200)... Le Paris JUG est une association qui fonctionne grâce aux sponsors, ce qui permet de préparer des événements, de participer à des manifestations comme Devox ou Jazon, et donc de faire connaître la communauté des développeurs Java de la région Ile-de-France. Depuis l'an dernier, le phénomène a pris de l'ampleur, des JUG se sont créés un peu partout en France : - Tours JUG, Christophe Jollivet - Bordeaux JUG, Xavier Hanin - Riviera JUG Nice Sofia Antipolis Nicolas Leroux et Stephane Epardaud - BreizhJug région Bretagne, Nicolas de Loof - Nantes JUG, Sébastien Roul - Lorraine JUG, Christophe Meyer, JM Doudoux et Xavier Roy - JUG Lyon, Julien Ripault et Cédric Exbrayat - Toulouse JUG, Gaël Blondelle La soirée anniversaire permettra de réunir les JUG leaders français, nos voisins du Luxembourg, des amis belges et suisses, ce qui promet un panel très sympa de têtes connues que j'ai vu à Devox. Ensuite Tarik Filali Ansary, un des habitués du Paris JUG, effectuera une présentation en 15 minutes du framework Apache Wicket. Nous aurons ensuite une présentation de la prochaine conférence en Suisse sur Java, Jazon'09 qui se déroule à Zurich. Elle se tient du 22 au 25 juin 2009, voir aussi [le site officiel](#). Cette présentation sera effectuée par Christian Frei. Les 3 sponsors buffet de cet anniversaire exceptionnel : - [Xebia](#), cabinet de conseil spécialisé Java - [Le Monde en tique](#), librairie bien connue des passionnés du monde informatique - [Jazon](#) la conférence internationale sur Java En deuxième partie de soirée, nous aurons la chance d'avoir la visite de Stephan Janssen, le fondateur et organisateur du salon Devox (ex Javapolis). Avec 3200 visiteurs (dont moi) en décembre dernier, Devox est l'un des plus gros événements Java en Europe qui se déroule chaque année depuis 8 ans. Stephan vient nous présenter son produit et sa société, Parleys.com dont je vous ai déjà parlé il y a longtemps sur le Touilleur. [Parleys.com](#) est un site réalisé avec les technologies d'Adobe qui vous propose de retrouver les meilleurs vidéos des conférences Devox, SpringOne et BeJUG. Grâce à la plateforme de Stephan, vous pouvez revoir tranquillement les présentations les plus marquantes de ces conférences. Parleys.com est une plateforme d'enseignement riche de nouvelle génération, l'exemple réussi d'une application Web 2.0. La soirée se poursuivra avec la présentation de **Java SE 7 Dolphin** et des nouveautés à venir comme le système modulaire, les superpackages, les génériques, le support du XML, l'api joda time, etc. La présentation est faite par Thomas Chamas, développeur Java à la Casden Banque populaire. Enfin pour terminer, Julien Gouesse proposera une présentation de la plateforme Java vue du point de vue de la création de jeux 3D basés sur OpenGL. L'inscription se fait via le site du Paris JUG : <http://www.parisjug.org> Je vous dis à mardi prochain, à la FIAP, non loin du RER Denfert-Rochereau.

## Scrum c'est fini ou le sujet People de janvier

Le sport ce mois-ci dans la blogosphère, c'est de d'annoncer l'échec ou la mort prochaine de Scrum. Chacun y va de son article, soit vraiment inspiré et intéressant, soit complètement à côté de la plaque. C'est un mouvement amorcé sur les blogs anglophones qui va sans doute arriver dans les semaines qui viennent chez nous. Je vous propose un résumé des questions soulevées ces derniers temps afin de faire avancer le débat. En novembre dernier, James Shore a publié un article intitulé "[The Decline and Fall of Agile](#)". En substance, James explique que les clients qu'il rencontre et qui disent avoir mis de l'Agilité dans leur gestion de projet sont souvent en souffrance. En passant d'une conception où le design et l'architecture sont fait en amont à un processus itératif comme Scrum, ses clients emmagasinent une énorme dette technique. Certes les fonctions clés du logiciel sont livrées mais il regrette que ses clients ne prennent pas en compte aussi les bonnes pratiques de la conception Agile proposées par l'eXtreme Programming (XP). Scrum est un formidable bâton de dynamite. Pour peu que votre projet soit un peu bancal, mettez en place Scrum, laissez exploser cette dynamite et les poissons morts viendront à la surface. C'est un catalyseur qui mettra en avant l'ensemble des points faibles d'un projet ou d'une équipe, chose que l'on ne voit pas aussi vite avec un projet au forfait classique. Alors que faire pour éviter cela ou se corriger ? La question est : comment durer ? James explique que l'introspection est l'un des principes les plus importants souvent oublié ou négligé par les personnes qui débutent avec Scrum. Le souci numéro un, et je le vois aussi aujourd'hui dans ma mission, c'est que l'on est tenté de ne choisir qu'une partie des règles de Scrum. Imaginez jouer au foot mais en ajoutant que vous pouvez ramasser la balle avec la main et que la durée des mi-temps n'est pas fixée à 45mn mais à votre bon vouloir... Premier principe : suivre dans un premier temps le processus. Agile ne veut pas dire "je peux faire n'importe quoi". Une critique faite à Scrum est aussi de ne pas avoir de pratique d'ingénierie. Oui en effet, Scrum n'est pas XP (eXtreme Programming). Scrum est un framework léger de gestion de projet. Ce n'est pas votre mère. Si vous souhaitez coder comme un cochon, libre à vous de faire ce que vous voulez. Jeff Sutherland explique qu'il voit que les équipes Scrum s'équipent en complément de méthodes XP, mais qu'il ne faut pas tomber dans le dogmatisme à outrance. J'ai fait l'expérience de faire du Scrum avec des pratiques de l'extreme programming (intégration continue, tests unitaires) sans non plus appliquer les 13 Principes d'XP car mon équipe marchait très bien sans. Et je ne suis pas le seul à [le dire](#). Regardons nos méthodes de développement : il est acquis en 2009 que l'industrialisation, l'intégration continue, les tests unitaires et le travail en binôme sont importants. Je me souviens de l'excitation autour d'XP que les gens ont un peu oublié. En 2002 [un livre écrit par des français](#) comment Laurent Bossavit plutôt bien écrit m'a fait découvrir XP. Aujourd'hui nous sommes tous d'accord pour dire que les tests unitaires, la pratique du TDD, la refactorisation sont des pratiques importantes, indispensables et professionnelles. Donc pour moi, je considère comme acquis dans notre conscience ce besoin, et qu'il n'est pas forcément nécessaire de cristalliser les personnes qui font du Scrum en leur disant "ET FAITES AUSSI DU XP NOM DE DIEU...". La majorité des équipes qui appliquent Scrum utilisent déjà tout ou partie des pratiques XP. Un développement au forfait fonctionne aussi bien avec des bonnes pratiques proposées par XP. Revenons à Scrum : l'importance de l'environnement. Je travaille depuis quelques mois dans une nouvelle équipe. Scrum ne fonctionne pas comme je l'ai vu fonctionner avant. Pour l'instant je dis que cela ne fonctionne pas encore. Je m'interroge et je me demande si c'est de ma faute, si c'est la faute des gens, des managers... Pourquoi cette recette de cuisine ne semble-t-elle pas fonctionner pour l'instant ? Parce que justement ce n'est pas une recette. Je pense maintenant que Scrum donne un cadre et des pratiques simples. En modifiant ce que faisait l'équipe auparavant, le premier effet est de voir surgir les problèmes que nous n'avions pas identifiés. Merci à Scrum. Ensuite je pense que bien que la recette soit définie, je dois m'autoriser à adapter une partie de Scrum à la vraie vie, ce que je fais. En faisant cela, j'ai bien conscience de prendre le risque de rendre les choses encore plus terribles... ce sera la faute de Scrum à la fin non ? Alors j'ai regardé mon environnement : les équipes, le produit, les managers. Et c'est là qu'il faut travailler, redoubler d'efforts pour que la mise en place de Scrum ne soit pas partielle mais complète. Expliquer, rassurer, accompagner, coacher, écouter, parler, dessiner, dialoguer, diriger... mais ne pas faire du Scrum light. Je l'ai dit au départ à un stand-up un matin : il n'y a pas de "on fait un peu de Scrum, il faut s'engager complètement et suivre toutes les pratiques". Si vous pensez à Scrum pour gérer votre projet, je vous conseille d'y aller complètement et de ne pas suivre qu'une partie du framework Scrum. Ne prenez pas que le dessert. Prenez l'entrée, le plat de résistance et le dessert. Avant de critiquer un sport, assurez-vous de bien en connaître toutes les règles, de l'avoir pratiqué et d'avoir aussi accepté toutes les contraintes. Nous en venons maintenant à ce que je dirai devant pas mal de monde en avril : scrum est un pétard qui a deux effets : euphorisant d'une part, explosif d'autre part. Dans les équipes qui travaillent encore avec des méthodes classiques de développement en cascade, le passage à une méthode Agile est un vrai explosif. Imaginez un chef de projet qui travaille 4 semaines sur une spécification technique. Il écrit noir sur blanc toutes les idées du client. Ensuite il rencontre les développeurs, qui ne sont pas d'accord ou qui objectent une partie du cahier des charges... Il réécrit alors ce cahier... Et ainsi de suite. Basculez ce projet à Scrum et la vie sera plus rose, le client sera impliqué et acteur de ses décisions. Nous livrerons moins mais mieux et plus souvent... Scrum dans ce cas dynamite un peu l'organisation. Les cycles sont plus courts, la vélocité de production augmente et la satisfaction des clients augmente... C'est le premier effet Scrum. Dans une équipe complètement déstructurée, chaotique, qui travaille dans l'urgence avec la méthode [LaRACHE](#), la valeur de chacun est estimée sur le temps qu'il passe à réaliser son travail. Souvent le principe est de répondre aux demandes du client afin de s'assurer que celui-ci soit satisfait. Je pense dans ce cas que Scrum apporte un effet apaisant en structurant le dialogue entre les clients externes (les chickens) et les chefs de projets (les pigs). Dans une équipe chaotique pour ne pas utiliser un autre mot, le client peut venir voir le développeur et demander une modification au nom du principe édicté par une personne qui n'a jamais codé : "le client est roi". Ce principe consomme beaucoup d'énergie, produit de la valeur sur le court terme mais n'est pas viable à terme. Imaginez une équipe de foot où les spectateurs pourraient venir sur le terrain pour expliquer au défenseur comment jouer... L'apport de Scrum dans ce type d'équipe est donc une structuration relativement légère qui permet un recadrage en douceur. Regardons un peu plus loin : au delà de Scrum, honnêtement je n'imagine pas travailler, diriger une équipe ou

même demain une entreprise sans au moins réfléchir aux pratiques, aux valeurs et aux techniques de cette méthode. Et parce que c'est une méthode simple, je me dis qu'il n'y pas de coûts excessifs à l'utiliser. Il reste cependant important d'intégrer des pratiques industrielles et des pratiques XP mais je pense que cela fait maintenant partie de nos habitudes. Il faut encore attendre et laisser mûrir Scrum. Pour terminer, imaginez que vous êtes skipper, que je vous confie la mission de traverser l'Atlantique en bateau. Pour cela, pensez comment cela se passerait dans le cadre d'un projet au forfait classique puis ensuite sous la forme d'un projet Agile utilisant Scrum... Moi je dis que le bonhomme qui chaque jour corrige son cap fait de l'itération car il sait qu'il n'y a pas qu'un seul cap pour traverser l'Atlantique. Sa carte marine est son burndown-chart : il a dessiné la route idéale avant de partir, et il reporte chaque jour le trajet réalisé afin de savoir le "reste à faire"... Sur ce, bonne journée. **Références:** Hervé Lourdin OCTO Technologies <http://blog.octo.com/index.php/2009/02/05/233-la-faute-a-scrum> Martin Fowler, sur XP et Scrum <http://martinfowler.com/bliki/FlaccidScrum.html> James Shore Scrum met en avant des problèmes existants, attention à sa fausse simplicité <http://jamesshore.com/Blog/The-Divide-and-Fall-of-Agile.html> Le Chaos et l'ordre, Scrum vient déplacer votre écosystème (merci à Gabriel Kastenbaum) <http://www.noop.nl/2009/02/the-divide-and-fall-of-agilists.html> Scrum ou XP ? Scrum ET XP Guillaume Bodet, Xebia <http://blog.xebia.fr/2008/01/10/scrum-ou-xp-scrum-et-xp/> Pourquoi les projets Agile ne peuvent pas être vraiment menés au forfait <http://blog.xebia.fr/2009/02/04/pourquoi-les-projets-agiles-ne-peuvent-pas-vraiment-etre-menes-au-forfait/>

## Service perturbé

Pas de long debriefing ce matin de la soirée JUG car je vous avoue qu'avec quelques heures de sommeil, j'ai un peu de mal ce matin. Hier soir : soirée anniversaire du Paris JUG organisée à la FIAP. C'était une très bonne soirée, avec des invités surprises et de nombreux cadeaux, la visite de 10 JUG francophone, un concours organisé par Jazoon, des présentations de 15 Minutes d'Apache Wicket, Parleys.com, Java 7 et Java 3D. J'ai pris quelques photos, discuté avec Thierry et Christian Frei de l'organisation de Jazoon, discuté Scrum avec Xebia et OCTO Technologies, parlé SpringFuse avec Nicolas et Florent, motivé Stéphane pour s'inscrire en tant que Rookie à Jazoon, parlé dans la rue sur l'organisation des soirées du JUG, discuté avec des JUG leaders de toute la France, parlé Adobe avec François le Droff, parlé vie en Suisse avec Jean-michel, Thomas et Thierry, etc. Paris JUG = réseau = discussion sympa avec gens sympathiques. Bref le mieux c'est que l'on reprenne rendez-vous ensemble. Je prépare un compte-rendu digne du Touilleur et vous repassez le chercher demain ? On fait comme ça ? Merci à David, Zouer et Antonio pour la soirée du [Paris JUG](#), il faut vraiment que les derniers sponsors viennent en aide financièrement pour que le Paris JUG continue de fonctionner. 259 visiteurs hier soir, 38 personnes au restaurant jusqu'à 02h00, c'était top. A bientôt

## Compte-rendu de la soirée anniversaire du Paris JUG

Mardi 11 février à Paris se tenait la soirée anniversaire du Paris JUG. Il s'agit d'un groupe de passionnés qui se retrouvent chaque deuxième mardi à Paris, gratuitement. Oui je sais vous le savez déjà. Mais c'est pour les derniers Java-istes parisiens (les pauvres) qui ne connaissent pas le Paris JUG... Démarrage dans une salle chauffée à blanc, avec foule en délire, masseuses (je vous JURE qu'il y avait des masseuses) et une bonne partie d'évanouissement tellement l'ambiance était dingue. Les sponsors (Xebia, le Monde en tique, Java Blackbelt et Jazoon) ont assuré la logistique et le buffet pour environ 250 personnes selon le comptage de la police, 300 selon le comptage du Touilleur Express et -1 selon Microsoft. Premiers à monter sur scène : l'équipe de [Java Black Belt](#). Sur simple demande via la mailing list du Paris JUG, vous pouvez gagner 15 points pour participer à la communauté des contributeurs du site Java Black Belt. Il y avait par ailleurs ensuite des enregistrements fait par John et il mettra les vidéos en ligne rapidement. Antonio, Zouheir et David les 3 JUG leaders du Paris JUG sont ensuite montés sur scène afin de raconter l'aventure depuis le début [il y a un an](#) jusqu'à ce soir, devant 300 personnes, comptage du Touilleur. Le site du Paris JUG a reçu la visite de 15 000 visiteurs différents ces 7 derniers mois. Sur la mailing-liste des membres, environ 187 inscrits. Chaque mois c'est 1600 visiteurs différents. C'est aussi un moyen pour les sponsors de se faire connaître. Des cabinets de conseils ont gagné une grosse visibilité en quelques mois. Je connais 3 personnes qui ont rejoint Zenika, Xebia et SFEIR via le bouche à oreille. D'ailleurs dommage qu'OCTO Technologies ne complète pas le board des sponsors. C'est peut-être prévu ? **Apache Wicket** Ensuite après un passage rapide des JUG leaders de France (j'y reviens) finalement c'est Tarik Filali Ansary, de la SSII SFEIR qui a pris la parole afin de présenter en 15 minutes le framework Web Apache Wicket. Et bien c'était très réussi. Après avoir présenté les concepts fondamentaux, Tarik a montré comment écrire une WebPage en Java et sa page HTML attachée. Ensuite il a montré comment coder un composant personnalisé. Au final, comme une bonne bande annonce, cela donne envie de tester Wicket. J'en profite pour vous glisser une publicité pour un article écrit par bibi : "[Apache Wicket et Apache Camel](#)" **Les JUG leader de France et du Luxembourg étaient tous là** Après cette présentation, nous avons eu la chance et le plaisir de voir l'ensemble des JUG leader de France et du Luxembourg sur scène : 10 Java User Group (JUG) réunis !!! Chaque JUG a pris la parole quelques instants, si vous lisez ces lignes et que vous êtes en province je vous demande d'aller voir un soir une réunion d'un JUG pour deux raisons : ces gens sont bénévoles et ensuite ils sont très sympas. Alors que vous soyez étudiant ou vieux routard, tentez le coup un soir. Dites que vous venez de la part du Touilleur ! Premier JUG : [le Breizh JUG de Rennes](#). Présenté par l'excellent Nicolas De Loof, ce jug a commencé il y a 6 mois. Avec environ 60 participants par session, le JUG se réunit chaque premier lundi du mois. Nicolas a bien mis l'ambiance en expliquant que l'objectif dans un premier temps est d'envahir la Bretagne, ensuite arriver jusqu'à Paris et enfin faire en sorte que Devoox soit organisé l'an prochain



3 minutes il réalise un montage parfait d'une présentation sur Spring... Vraiment sympa. **Java 7** Thomas Chamas nous a proposé ensuite une session de 15 minutes sur les nouveautés de Java 7, Dolphin. Débuté en 2006, le chantier Java 7 avance et la sortie est prévue pour 2010. Il est intéressant de voir les JSR retenues, ainsi que celles qui sont passées à la trappe depuis. Les closures ne feront finalement pas parties de la prochaine release de Java. Je vous propose de retrouver dans [un article sur devvox](#) une partie des sujets présentés par Thomas. **Réaliser un jeu en Java par Julien Gouesse** Pour terminer, Julien Gouesse a présenté aussi en 15 minutes les techniques disponibles pour réaliser un jeu en 3D en Java. Du côté graphisme, il nous présente une application simple utilisant le moteur de binding OpenGL de Java. Il présente ensuite le support du son et de la musique avec JOGG et JORBIS. La présentation s'est terminée par un jeu en 3D appelé "TUER" qui m'a plongé des années en arrière. Julien je ne sais pas du tout quel âge tu as, je dirai 25 ans à vue de nez. En 1996 je jouais à Wolfstein 3D, et je codais à l'époque en C un jeu en 3D avec beaucoup de difficultés. Le support d'OpenGL n'existait pas, mais pendant que tu te baladais dans le jeu afin de nous montrer les effets 3D, je me revoyais quelques années en arrière. Au final il faut retenir qu'il est possible et intéressant de regarder le support de la 3D pour réaliser des jeux en Java. Le téléphone mobile est l'une des premières plate-formes visées de jeu, des éditeurs proposent des jeux à emporter et ce marché est en pleine explosion... comme ce robot orange ! **Et enfin la 3ème mi-temps** Nous étions 38 motivés pour ensuite aller manger au restaurant. Ambiance sympa, encore l'occasion de rencontrer de nouvelles personnes, des gens qui disent "...ah c'est toi le touilleur !" en souriant, de parler de tous les sujets, et d'être vraiment content de partager ce moment grâce à David, Antonio et Zouheir.

## Présentation de Scrum en 10 minutes

Cette vidéo trouvée sur [AboutScrum.com](#) présente Scrum en 10 minutes. Dites-moi ce que vous en

pensez

## Un JUG à Lille

[Cyril Lakech](#) m'a signalé qu'un JUG dans le Nord serait en préparation. Avis à toutes les personnes habitant autour de la métropole Lilloise, vous pouvez contacter cyril via son blog <http://cyrillakech.blogspot.com/>. Bonne chance cyril et tiens-nous au courant.

## Les origines de Scrum

**Un peu d'histoire et quelques réflexions autour de Scrum. Vous savez sans doute que Scrum en anglais, se traduit par mêlée comme au Rugby. Mais savez-vous pourquoi ? Nous parlerons ensuite de la différence entre les méthodes Lean et les méthodes Agile, et comment il est possible de rejoindre ces deux mondes.** La première apparition du mot Scrum date d'un article de janvier 1986 de douze pages que vous pouvez trouver sur internet : "The New New Product Development Game" publié par Hiroataka Takeuchi et Ikujiro Nonaka. Il n'est pas question dans l'article de définir une méthodologie, simplement de rendre compte d'observations sur le terrain effectuées par les 2 professeurs de l'université Hitotsubashi au Japon. Scrum et le rugby sont cités afin d'expliquer la différence entre une méthode séquentielle et une méthode itérative par recouvrement. L'image pour expliquer le mieux possible la méthode séquentielle est par exemple le relais 4x100m aux Jeux Olympiques. La réussite de l'équipe dépend de l'enchaînement de séquences d'efforts individuelles. Par analogie avec le développement informatique, les 2 chercheurs expliquent donc qu'il y a un risque lors du passage de témoin et que par ailleurs, pendant qu'une équipe (un coureur) réalise un effort, les autres ne peuvent quasiment rien faire. Cette méthode séquentielle est utilisée depuis le début des années 70. Elle est fortement influencée par les travaux de la NASA et le développement PPP (Phase Program Planning). L'article traite de l'observation de nouvelles pratiques de développement de produits, à aucun moment il n'est fait mention de développement logiciel, nous sommes en janvier 1986... Le mot Scrum est utilisé afin d'expliquer comment travaille une équipe de Rugby sur le terrain : l'effort collaboratif est global, l'avancement se fait par petites itérations, les spécialités des joueurs sont mélangées et ils doivent s'accorder pour réussir ensemble dans un intervalle de temps limité. Ceux qui font déjà du Scrum voient où je veux en venir. Les 2 chercheurs identifient (en 1986) 6 facteurs clés novateurs pour l'époque : - l'instabilité intrinsèque - des équipes auto-organisées - le recouvrement des différentes

phases du cycle de développement - l'apprentissage global et multiple - le subtil contrôle - le transfert de la connaissance dans l'organisation L'instabilité intrinsèque signifie qu'il faut tout d'abord challenger une équipe avec un objectif précis, réaliste mais qui donne un fort challenge à l'équipe. Il s'agit de bousculer le développement en fixant un objectif. Les 2 auteurs citent ainsi que la créativité est décuplée s'il y a un challenge. Il s'agit de pousser l'équipe. Ils citent le cas d'Honda : soit faire un véhicule pour les jeunes, soit faire une voiture pas cher. Finalement les décideurs ont demandé les deux, de faire une voiture pour les jeunes pas chère... L'importance d'une équipe auto-organisée est surtout novateur pour l'époque. Il s'agit de sortir d'un cadre stricte où l'initiative individuelle n'a pas lieu d'être, où le manager omniscient sait tout et décide tout, pour déléguer à l'équipe plusieurs rôles. Aujourd'hui on voit encore des chefs de projet qui estiment le temps nécessaire pour coder une nouvelle fonction, ce qui est une erreur à mon avis. Il est plus logique de demander à son équipe, à ses développeurs de réaliser une estimation. Tout d'abord un développeur ne donne pas le même engagement si c'est lui qui a estimé la tâche, et par ailleurs le fait de demander à plusieurs personnes permet aussi de s'assurer que l'estimation, et donc la tâche à réaliser, a bien été comprise par le développeur et le chef de projet. Une équipe auto-organisée met en place des processus de communications que le monde extérieur ne va pas percevoir. Sur un terrain de foot, dans un régiment de commandos, la coordination et la communication sont important pour la bonne réussite du match ou de la mission. A votre avis pourquoi cela devrait être différent parce que nous sommes développeurs ? Enfin il y a ce que l'on appelle simplement l'esprit d'équipe qui permet de réussir. C'est l'idée qu'un groupe de personnes est plus fort que la somme de chacune des personnes, ce que l'on appelle la vision [holistique](#). On verra tout à l'heure qu'il est important de regarder Scrum dans son ensemble et pas en prenant chacun de ses éléments un par un. Le recouvrement des différentes phases du cycle de développement est cité par les 2 professeurs comme l'un des premiers éléments novateurs de leurs observations. En fait c'est l'idée d'itération. Ils constatent que des compagnies comme Xerox ou HP qui dès les années 1980 ont fait le choix d'utiliser des itérations, ont obtenu de meilleurs résultats, chiffre à l'appui. Ce qui me frappe c'est de lire ce texte qui a 23 ans et de voir à quel point il est d'actualité aujourd'hui. Fuji-Xerox a raccourci le développement d'une imprimante la FX-3500 de 38 mois à 28 mois en basculant d'une méthode séquentielle classique type waterfall vers une méthode itérative. Cependant il n'y a pas autant d'extrême que ce que l'on voit dans Scrum aujourd'hui. Je m'explique : les auteurs expliquent que le développement conserve ses phases globales qui sont par exemple : prototype, pré-maquette, maquette et commercialisation. Il y a donc bien un investissement initial dans de la recherche, il n'y a pas une mise en marche dès le départ dans l'idée de sortir au bout de deux sprints une version d'un logiciel qui marche... Hé bien oui, c'est peut-être une critique de Scrum, c'est que parfois pour de nouveaux développements on axe trop l'énergie afin de livrer du code rapidement qui répond à la demande du client, sans prendre le recul nécessaire ou le temps de valider de A à Z que la solution sera pérenne. La méthode PUMA de Jean-Pierre Vickoff met en avant justement l'importance de l'itératif tout en conservant une approche [en 4 phases](#). Pour défendre aussi Scrum, je pense qu'il appartient ensuite à chacun selon le contexte de son projet et le type de développement, d'envisager s'il est nécessaire ou non de travailler par phase. Enfin pour être certain d'être bien compris : je pense qu'il est possible de faire du Scrum en intégrant des phases et qu'il faut appliquer ce découpage selon le type de projet et de développement. L'apprentissage global et multiple est l'une des valeurs de Lean qui rappelle l'importance d'améliorer le niveau de ses équipes en permanence. Hier par exemple, en production j'ai rencontré un souci. C'est grâce à l'aide d'un senior sur le projet que j'ai appris comment résoudre le problème en regardant comment il travaille. Dans l'article un passage m'a frappé car il a cassé le mythe du Google Friday. En quelques mots, le vendredi chez Google chacun peut consacrer une partie de son temps à faire autre chose que son travail habituel. Voici ce que l'on peut lire dans l'article de Takeuchi et Nonaka page 7 : [...] *Learning at the individual level takes place in a number of ways. 3M, for example, encourages engineers to devote 15% of their company time to pursuing their "dream"*. En clair : dès 1984 une compagnie comme 3M prend conscience qu'en donnant 15% de temps libre passé sur le lieu de travail, les salariés pourraient se consacrer à autre chose que leur mission principale... Le contrôle subtil est ensuite exposé. Bien qu'une équipe soit autonome, elle est en fait encadrée et pilotée par des règles et des repères qui permettent aux managers de contrôler l'activité. C'est tout simplement les règles du Rugby qui en font un sport subtil plutôt qu'un pugilat, et c'est bien les règles qui assurent, avec la présence d'arbitres, que chaque match se déroule dans une ambiance où l'on respecte son adversaire. Dans l'article on retrouve là aussi une valeur de Lean, à savoir l'importance du casting d'une équipe. En sélectionnant les bonnes personnes pour constituer une équipe, en essayant d'obtenir un équilibre où chaque personne a sa place, les chances de réussir un projet sont plus grandes. Il est important d'encourager ensuite l'équipe à aller sur le terrain afin de rencontrer les clients, d'utiliser un système d'évaluation global afin de noter non pas l'individu mais le groupe dans son ensemble. Lors d'un match de foot, l'équipe gagne ou perd. On ne parle pas d'individualité. Il faut aussi anticiper et accepter les erreurs, ce qui est plus difficile dans la culture anglo-saxonne qu'au Japon. Là où un américain se congratule d'avoir lancé son logiciel, un japonais s'inquiète de ce qui n'a pas été parfait et pense comment améliorer la prochaine fois ce qui n'a pas été parfait... Et le français est en RTT, en grève, entrain de prendre un café et se bat avec sa AMOA/MOE/MERD dont lui seul a la connaissance... Le transfert de connaissance enfin est le dernier point abordé dans l'article. L'osmose est la première source de diffusion des bonnes pratiques. C'est aussi le choix d'une technologie ou d'un produit, qui fait qu'une autre équipe vient vous voir et se demande ce qu'il se passe. La propagation des bonnes pratiques est donc un autre facteur de réussite. Enfin, l'article se termine par une liste de limitations et de cas limites où il ne sera pas possible d'appliquer ces principes. Je suis tenté de traduire mot à mot ce que j'ai lu afin de ne pas en déformer les propos. Comme tout bon médicament, les 2 auteurs ne recommandent pas ce qu'ils ont observé à tout le monde. L'approche holistique (l'ensemble est plus fort que la somme de chacun des bonhommes) du développement a cependant un certain nombre de limitations et d'effets secondaires : - dans un premier temps, ils constatent que cela demande un effort considérable de l'équipe qui se manifeste par un nombre d'heures de travail beaucoup plus important. Cela va à l'encontre de Scrum qui préconise de ne pas épuiser ses équipes, mais on explique là clairement que la mise en place de ces pratiques va s'accompagner d'une surcharge de travail. - ces principes ne peuvent peut-être pas s'appliquer dans certains domaines qui demandent trop d'innovation ou de recherche comme la

chimie ou la biologie. - il ne s'appliquerait pas à des projets mamouths comme la conquête spatiale où les engagements ne peuvent pas se résoudre au sein d'une équipe, où les décideurs externes doivent faire partie de l'équipe - ces principes ne s'appliqueraient pas à une organisation dont le développement d'un produit se repose sur la connaissance d'un génie, d'une personne particulièrement forte qui créé le projet, l'invente, et qui donc fonctionne en mode monarchique. Enfin l'article se termine sur une réflexion de l'importance des managers, des encadrants et de la hiérarchie d'une entreprise. On peut lire par exemple que les top-managers ne doivent pas intervenir dans le développement, qu'ils doivent accepter une erreur en partant plutôt qu'en licenciant une équipe par exemple. Encore un principe Lean au passage. **Conclusion** Cet article de janvier 1986, qui repose sur de l'observation, fait le constat que les entreprises ayant adapté le processus de développement de leurs produits ont obtenu de meilleurs résultats. Tout comme Lean qui a une approche industrielle, cet article ne parle pas de développements logiciels. On parle bien de développement de produits, ce qui a son importance. Le dernier point qui me semble important, c'est l'importance de regarder **Scrum** dans son ensemble et pas sur certaines de ses caractéristiques. Si vous prenez le product backlog, c'est une perte de temps que de lister 250 histoires, si vous savez que votre équipe ne fait que 20 histoires par développement. Pourtant c'est une perte de temps encore plus grande que celle de ne **pas** lister du tout par **priorité** les fonctions à développer. Les critiques que l'on peut lire sur Scrum en ce moment sont normales, mais elles sont souvent construites sur la démonstration qu'un des points de Scrum ne fonctionne pas. La vision holistique (cela fait 3 fois que je vous bassine avec ce mot) veut tout simplement dire qu'il faut regarder l'ensemble d'une méthode, qu'il faut lire, l'expérimenter, ne pas tomber dans le piège facile de la critique de la partie commerciale de Scrum, et simplement réfléchir à ce que la méthode peut changer dans votre développement logiciel. Si demain je me permets de critiquer le foot au nom de la règle du hors-jeu, en ne regardant pas son apport dans le jeu, en n'essayant pas de regarder un match pour apprendre, en restant sur mes positions en pensant qu'il y a mieux, franchement je ne serai pas crédible. Je vous recommande d'aller lire l'article des 2 chercheurs, car il y a l'essence même de Scrum et des méthodes Agile. **Références:** [The New New Product Development Game Controlled Chaos Software Development](#) [http://www.entreprise-agile.com/méthode PUMA Article Wikipedia sur Scrum Le French Scrum User Group](http://www.entreprise-agile.com/méthode/PUMA_Article_Wikipedia_sur_Scrum_Le_French_Scrum_User_Group) est gratuit, ouvert à tous, dont je fais parti, et vous propose de parler de Scrum et des méthodes Agile sur Paris.

## Introduction à Spring Integration

Spring Integration est la solution d'intégration proposée par la communauté Spring. Cousin très proche de Mule, Spring Integration vous permet de réduire le code d'infrastructure technique de votre application. En quelques lignes vous pouvez ainsi lire un fichier sur le disque, vérifier chaque ligne puis ensuite enregistrer vers une base les données chargées. La version 1.0 a été annoncé en décembre dernier, Mark Fisher propose de vous montrer en 10 minutes comment écrire un service simple et faire fonctionner Spring Integration : <http://blog.springsource.com/2009/02/13/982/>. Dans son post dont j'ai repris le code ci-dessous, nous déclarons un bean Spring et nous allons montrer comment envoyer un message via Spring Integration. La configuration Spring importe le namespace de Spring Integration et permet de déclarer le bean Shouter :

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:si="http://www.springframework.org/schema/integration"
xsi:schemaLocation="
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/integration
http://www.springframework.org/schema/integration/spring-integration-1.0.xsd">

<si:channel id="input"/>

<si:channel id="output">
<si:queue capacity="10"/>
</si:channel>

<si:service-activator input-channel="input" output-channel="output" ref="shouter" method="shout"/>

<bean id="shouter" class="blog.Shouter"/>

</beans>
```

Voyons le bean Java qui va recevoir notre message et y ajouter des points d'exclamation:

```
public class Shouter {
    public String shout(String s) {
        return s.toUpperCase().concat("!!!");
    }
}
```

Le code Java pour tester le tout est simple :

```
public static void main(String[] args) {
    ApplicationContext context = new ClassPathXmlApplicationContext("blog/config.xml");
    MessageChannel input = (MessageChannel) context.getBean("input");
    PollableChannel output = (PollableChannel) context.getBean("output");

    input.send(new StringMessage("Le Touilleur Express"));
    Message< ?> reply = output.receive();

    System.out.println("received: " + reply);
}
```

**A quoi sert Spring Integration ?** Spring Integration permet de réduire le code technique de votre application. Vous lisez un message sur une queue JMS, vous voulez envoyer un email, vous souhaitez lire un fichier et le convertir vers un autre format, faire un traitement par batch, télécharger via FTP un fichier ou SSH, alors Spring Integration pourra vous intéresser. **Peut-on comparer Spring Integration à Mule ?** Pour qui connaît un peu [Mule 2](#), la ressemblance est frappante. Ross Mason, l'auteur de Mule, a tenté un moment de se rapprocher de la communauté Spring, mais celle-ci a préféré offrir une solution basée exclusivement sur Spring. Il n'y a pas de comparaison possible au sens "l'un est mieux que l'autre". Mule a cependant un peu plus de recul, [dispose de beaucoup plus de connecteurs](#) que Spring Integration pour l'instant. Sa communauté est assez active. Si vous souhaitez utiliser par exemple Tibco Rendezvous ou faire du SSH, il faudra utiliser Mule. Je pense qu'il est intéressant de connaître les deux projets et de ne pas s'aventurer à faire un choix, s'il faut en faire un. **Est-ce que Spring Integration est un ESB ?** Non, je dirai que c'est un framework léger d'intégration comme Apache Camel, mais pas un ESB. Le produit OpenESB basé sur le serveur d'application Glassfish est un ESB, car il est composé d'une API d'une part et d'un moteur d'autre part. Mule est un ESB léger, il propose son propre moteur ou il peut fonctionner dans une application Web. Spring Integration est par contre une brique peut-être plus facile à mettre en oeuvre si vos besoins sont avant tout techniques. **Pour aller plus loin** Le site de la communauté muleForge : <http://www.muleforge.org/> Le site de Mule 2 : <http://www.mulesource.org> [Article de Cyrille Le Clerc](#) sur le blog de Xebia en 2007 [Article de Sébastien Letélieu](#) d'Improve sur les ESB : [Java Camp II DB and ESB](#)

## Combien vaut votre site ?

A la fin de cet article vous auriez perdu 4mn de votre précieux temps, voir plus si vous lancez dans l'évaluation de votre site, celui du voisin, celui de la SSII d'en face et j'en passe... Alors allons perdre un peu de temps ensemble avec Stimator, dont le nom me fait penser à l'ensemble des emails de mon dossier Spam sur Gmail, si vous voyez ce que je veux dire... Le site [Stimator](#) calcule une valeur relative en dollars selon le nombre de pages indexées sur les moteurs de recherche, le nombre de lien intra-site, de liens venant de l'extérieur, de liens sur des sites comme Del.icio.us ou Facebook et vous donne une valeur de votre site. Premier réflexe, je regarde l'estimation du Touilleur Express : 1526 \$... Afin de donner une idée des chiffres calculés par Stimator, voici au 8 février une liste de sites sur Java : .....4 086 \$ : Blog [Application-Servers](#) .....216 783 \$ : Blog [Guillaume Laforge](#) (Groovy) .....1 652 \$ : Blog [SpringSource](#) (EN) .....5 739 \$ : Blog de [Kirk Pepperdine](#) (EN) .....343 \$ : Blog [OCTO Technologies](#) FR .....377 \$ : Blog [OCTO Technologies](#) domaine .com(EN) .....1 445 \$ : Blog [Xebia](#) France .....135 060 \$ : Site [InfoQ](#) (EN) .....681 039 \$ : Site [The Server Side](#) (EN) .....266 373 \$ : Site [L'Equipe](#) un intrus dans la liste .....761 107 \$ : Blog [The Aquarium](#) (Java) (EN) .....1 722 \$ : Blog [Paris Java User Group](#) 11 523 607 \$ : Site <http://java.sun.com> Et enfin le petit dernier de la classe : <http://www.google.com> est estimé à 1 156 988 215 \$ La valeur est estimée par Stimator sur la base du nombre d'articles, de liens entrants, de référencement, de mots clés et très certainement sur la valeur de l'indice Google rank. L'idée des créateurs du site est intéressante, elle pourrait faire tâche d'huile, Google a déjà proposé le Google Rank mais cet indicateur est trop axé sur le calcul des liens entrants vers votre site. Après ces 4 minutes de culture web offert par le Touilleur, je vous laisser reprendre le cours d'une activité normale !

## Hibernate : gérer le chargement des associations efficacement

### Hibernate en quelques mots

Hibernate est un moteur de mapping objet-relationnel (ORM) qui permet de charger des données venant d'une base de données, vers le monde Java en réduisant la quantité de code Java à écrire. Vous définissez une classe Java, ensuite un fichier de mapping au format XML, et Hibernate offre alors un moyen pour charger, modifier et effacer vos données. Il dispose aussi d'un cache de requête afin d'optimiser les appels et d'éviter des aller-retours inutiles avec la base de données. Un cache de second niveau offre un moyen très puissant mais délicat à maîtriser pour partager vos données et réduire le coût de la base de données dans une application. Il est possible d'utiliser soit des annotations sur la classe Java, soit des fichiers XML. Sachez que si vous utilisez des annotations, un fichier de configuration XML peut vous permettre de surcharger le mapping des annotations, et donc d'éviter d'être bloqué par du code déjà compilé sur lequel vous n'avez pas le contrôle.

### Objectifs de l'article

La première partie de l'article est assez dictatoire pour ceux qui ne connaissent pas Hibernate 3. La seconde partie sera plus intéressante pour les utilisateurs qui souhaitent voir quelques options avancées d'Hibernate. Nous verrons comment gérer efficacement les relations entre les objets, les stratégies de chargement et les différentes types d'option de chargement tardif (Lazy instanciation).

### Code source de l'article

Pour les besoins de cet article j'ai préparé un projet simple avec maven2, que vous pouvez télécharger en fin d'article. Le code est distribué sous licence Creative Commons v2 avec quelques restrictions sur la non modification et l'interdiction d'utiliser le code à des fins commerciales sans mon accord. Si vous souhaitez tester quelques fonctionnalités en lisant l'article, vous aurez besoin de Java 5 et de Maven 2.0.6 minimum pour compiler et tester le code. J'utilise un test unitaire

JUnit 3.8 simple pour lancer les différents tests, un simple "mvn test" en ligne de commande vous permet de compiler le code et de lancer la base de données embarquée HSQLDB.

## Un peu de relationnel

J'ai repris un exemple simple pour passer en revue quelques caractéristiques d'Hibernate plus ou moins connues. Prenons tout d'abord la classe Item, correspondant à un article. Un Item est caractérisé par son nom, son prix et une clé d'identité. Un constructeur vide sans argument est requis pour qu'Hibernate puisse décorer cette classe avec un proxy avec la librairie CGLIB, sans quoi une exception org.hibernate.InstantiationException sera levée par Hibernate. De même il est recommandé de ne pas déclarer de classe finale comme entité si vous souhaitez qu'Hibernate puisse utiliser un proxy. Si malgré tout vous êtes obligés de déclarer une méthode finale, il faudra alors désactiver le chargement tardif (lazy=false) dans la configuration de l'entité [1]

```
public class Item {
    private Integer id;
    private String name;
    private float price;
    private Set<Bid> bids=new HashSet<Bid>();

    public Item() {
    }

    public Item(String name, float price){
        //...
    }
    // Getter, setter, equals, hashCode et toString
}
```

La classe Bid (enchère en anglais) est une offre émise par un acheteur sur notre Item. Les attributs sont l'heure de l'enchère et son montant.

```
public class Bid {
    private Integer id;
    private float amount;
    private Date date;
    private Item item;

    public Bid(){
    }

    // getter, setter, hashCode, equals
}
```

Le fichier de mapping Hibernate Item.hbm.xml permet à Hibernate d'effectuer le mapping entre la classe Item et notre base de données. La relation un-vers-plusieurs (un Item contient plusieurs Bids) est définie avec un set. Nous utiliserons une table de jointure ITEM\_BIDS, le chargement tardif sera désactivé (lazy=false), nous indiquons à Hibernate que la relation inverse est déclarée dans Bid (inverse=true).

```
<hibernate-mapping>

<class name="org.touilleur.hibernate.v1.Item" table="ITEM">
    <id name="id" column="ITEM_ID">
        <generator class="native"/>
    </id>
    <property name="name"/>
    <property name="price"/>
    <set name="bids" table="ITEM_BIDS"
        lazy="false"
        inverse="true"
        fetch="select"
        cascade="all">
        <key column="ITEM_ID" not-null="true"/>
        <one-to-many class="org.touilleur.hibernate.v1.Bid"/>
    </set>
</class>

</hibernate-mapping>
```

Hibernate recommande d'utiliser des associations symétriques. Dans notre application, une enchère sans article n'a pas de sens. Nous configurons donc notre association many-to-one avec l'attribut not-null à true dans le fichier Bid.hbm.xml:

```
<hibernate-mapping>
<class name="org.touilleur.hibernate.v1.Bid" table="BID">
    <id name="id" column="BID_ID">
        <generator class="native"/>
    </id>
    <property name="amount"/>
    <property name="date"/>
    <many-to-one name="item"
        column="ITEM_ID"
        class="org.touilleur.hibernate.v1.Item"
        not-null="true"/>
</class>

</hibernate-mapping>
```

## Stratégie de chargement de l'association

Hibernate évite au maximum de faire appel à la base de données, et tend à limiter le nombre de requête, afin d'alléger le traitement. Par défaut la stratégie de chargement tardive qui permet d'optimiser les performances, peut provoquer quelques moments difficile lorsque surgit l'exception *LazyInitializationException*. Une exception de type *LazyInitializationException* sera renvoyée par Hibernate si une collection ou un proxy non initialisé est accédé en dehors de la portée de la Session, e.g. lorsque l'entité à laquelle appartient la collection ou qui a une référence vers le proxy est dans l'état "détachée". Cela se produit dans les architectures 3-tiers, où le tiers de présentation Web est séparé physiquement de l'ejbtiers pour des raisons de sécurité. Ce type d'architecture souffre de plusieurs soucis : si vous interdisez l'accès à la base de données à partir du web-tiers, il faudra donc que la stratégie de chargement tardive d'Hibernate soit désactivée (*lazy=false*). Dans le cas où nous souhaitons charger un Item, toutes les enchères associées seront alors chargées. Imaginez que pour seulement afficher le nom de l'Item, nous allons aussi charger 1500 Bids par exemple... Il est alors important de configurer Hibernate finement et d'utiliser des jointures afin d'éviter trop de requêtes. Dans un premier temps, nous allons regarder les différentes possibilités de chargement d'une association, en essayant différentes valeurs pour l'attribut *fetch* de l'attribut *set*. La valeur par défaut lorsqu'elle n'est pas précisée est "select". Les 3 valeurs possibles sont *select*, *join* et *subselect*. Nous désactivons le chargement tardif en ajoutant le mot clé "*lazy=false*" dans le fichier *Item.hbm.xml*.

## Fetch strategy par défaut en mode *lazy=false*

Pour ce test, dans le test unitaire *SelectTest*, nous allons voir comment il est possible de changer la stratégie de chargement directement dans le code. Avant cela, regardons comment travaille Hibernate par défaut avant de tester d'autres modes de chargement. Par défaut lorsque *FetchMode* est à *DEFAULT* comme ci-dessous, Hibernate se repose sur ce que vous avez déclaré dans votre fichier de mapping, *Item.hbm.xml* pour nous.

```
// voir org.touilleur.hibernate.SelectTest
Item item1 = (Item) session.createCriteria(Item.class)
    .setFetchMode("bids", FetchMode.DEFAULT)
    .add(Restrictions.idEq(id))
    .uniqueResult();
```

La configuration du chargement de l'association s'effectue dans le fichier de configuration *hibernate.cfg.xml* à l'aide de l'attribut *fetch*. La valeur par défaut est *select*, les 2 autres valeurs possibles sont *join* et *subselect*. Voyons d'abord par défaut le comportement d'Hibernate :

```
<set name="bids" table="ITEM_BIDS" lazy="false" inverse="false"
    fetch="select" cascade="all">
```

Pour continuer, exécutez la méthode *testSelectOneItemWithDefault*. La console affiche les requêtes SQL suivantes: ... INFO 2009-02-27 22:24:21,788 [Demo Hibernate] : Starting *selectOneItemWithDefault* Hibernate: /\* criteria query \*/ select this.ITEM\_ID as ITEM1\_0\_0\_, this.name as name0\_0\_, this.price as price0\_0\_ from ITEM this where this.ITEM\_ID = ? Hibernate: /\* load one-to-many org.touilleur.hibernate.vl.Item.bids \*/ select bids0.ITEM\_ID as ITEM4\_1\_, bids0.BID\_ID as BID1\_1\_, bids0.BID\_ID as BID1\_1\_0\_, bids0.amount as amount1\_0\_, bids0.date as date1\_0\_, bids0.ITEM\_ID as ITEM4\_1\_0\_ from BID bids0 where bids0.ITEM\_ID=? INFO 2009-02-27 22:24:21,791 [Demo Hibernate] : Loaded item1 with a FetchMode set to DEFAULT... 2 requêtes sont nécessaires pour effectuer le chargement de l'unique Item. **Chargement de l'ensemble des Items** Le test unitaire *testSelectAllItemsWithDefaultFetchMode* sélectionne les 2 Items de notre base et affiche ensuite les 2 items. La méthode *toString* de la class *Item* force le chargement de la liste des enchères. Nous avons désactivé le chargement tardif (*lazy=false*) et il est donc normal de voir Hibernate charger les Bids. Que se passe-t-il lorsque nous exécutons ce code ? INFO 2009-03-03 10:18:30,670 [Demo Hibernate] : testSelectAllItemsWithDefaultFetchMode - begin Hibernate: /\* criteria query \*/ select this.ITEM\_ID as ITEM1\_0\_0\_, this.name as name0\_0\_, this.price as price0\_0\_ from ITEM this Hibernate: /\* load one-to-many org.touilleur.hibernate.vl.Item.bids \*/ select bids0.ITEM\_ID as ITEM4\_1\_, bids0.BID\_ID as BID1\_1\_, bids0.BID\_ID as BID1\_1\_0\_, bids0.amount as amount1\_0\_, bids0.date as date1\_0\_, bids0.ITEM\_ID as ITEM4\_1\_0\_ from BID bids0 where bids0.ITEM\_ID=? Hibernate: /\* load one-to-many org.touilleur.hibernate.vl.Item.bids \*/ select bids0.ITEM\_ID as ITEM4\_1\_, bids0.BID\_ID as BID1\_1\_, bids0.BID\_ID as BID1\_1\_0\_, bids0.amount as amount1\_0\_, bids0.date as date1\_0\_, bids0.ITEM\_ID as ITEM4\_1\_0\_ from BID bids0 where bids0.ITEM\_ID=? INFO 2009-03-03 10:18:30,704 [Demo Hibernate] : Item #1 : Item{id=1, bids=[Bid{id=7, amount=107.0, date=2009-03-03 10:18:30.637}, Bid{id=10, amount=110.0, date=2009-03-03 10:18:30.643}, Bid{id=6, amount=106.0, date=2009-03-03 10:18:30.632}, Bid{id=8, amount=108.0, date=2009-03-03 10:18:30.638}, Bid{id=2, amount=102.0, date=2009-03-03 10:18:30.626}, Bid{id=4, amount=104.0, date=2009-03-03 10:18:30.629}, Bid{id=1, amount=101.0, date=2009-03-03 10:18:30.625}, Bid{id=9, amount=109.0, date=2009-03-03 10:18:30.639}, Bid{id=5, amount=105.0, date=2009-03-03 10:18:30.63}, Bid{id=3, amount=103.0, date=2009-03-03 10:18:30.628}], name='TV LCD', price=340.0} INFO 2009-03-03 10:18:30,706 [Demo Hibernate] : Item #2 : Item{id=2, bids=[Bid{id=20, amount=144.0, date=2009-03-03 10:18:30.657}, Bid{id=15, amount=125.0, date=2009-03-03 10:18:30.651}, Bid{id=13, amount=43.0, date=2009-03-03 10:18:30.647}, Bid{id=16, amount=116.0, date=2009-03-03 10:18:30.652}, Bid{id=14, amount=84.0, date=2009-03-03 10:18:30.648}, Bid{id=11, amount=10.0, date=2009-03-03 10:18:30.645}, Bid{id=17, amount=137.0, date=2009-03-03 10:18:30.654}, Bid{id=19, amount=139.0, date=2009-03-03 10:18:30.656}, Bid{id=12, amount=22.0, date=2009-03-03 10:18:30.646}, Bid{id=18, amount=138.0, date=2009-03-03 10:18:30.655}], name='Plasma', price=1230.0} INFO 2009-03-03 10:18:30,709 [Demo Hibernate] : testSelectAllItemsWithDefaultFetchMode - end Hibernate utilise 3 requêtes pour effectuer le chargement. C'est un problème important. Imaginez que votre base contient 1000 items qui eux-mêmes référencent 10 Bids différents... Nous n'optimisons pas vraiment notre chargement, cela nous coûte cher en nombre de requêtes SQL. De plus, n'étant pas en mode de chargement tardif, nous voyons

bien qu'Hibernate effectue les 3 requêtes avant d'attaquer l'affichage de notre résultat. **Que se passe-t-il lorsque le mode lazy est actif ?** Rappelons que par défaut Hibernate étant bien fait, ce mode est implicite et il est le mode de fonctionnement par défaut d'Hibernate. Pour ce test, je me contente de changer dans le fichier Item.hbm.xml la configuration du chargement tardif et je relance le même test afin de voir comment Hibernate charge la collection:

```
<hibernate-mapping>

  <class name="org.touilleur.hibernate.v1.Item" table="ITEM">
    <id name="id" column="ITEM_ID">
      <generator class="native"/>
    </id>
    <property name="name"/>
    <!-- The fetch join strategy uses a select join query to load with one transact the set of bids -->
    <property name="price"/>
    <!-- LAZY est à TRUE -->
    <set name="bids" table="ITEM_BIDS" lazy="true" inverse="true" fetch="select" cascade="all">
      <key column="ITEM_ID" not-null="true"/>
      <one-to-many class="org.touilleur.hibernate.v1.Bid"/>
    </set>
  </class>

</hibernate-mapping>
```

```
Je relance le même test unitaire : INFO 2009-03-03 10:26:38,910 [Demo Hibernate]
: testSelectAllItemsWithDefaultFetchMode - begin Hibernate: /* criteria query */
select this_.ITEM_ID as ITEM1_0_, this_.name as name0_0_, this_.price as
price0_0_ from ITEM this_ Hibernate: /* load one-to-many
org.touilleur.hibernate.v1.Item.bids */ select bids0_.ITEM_ID as ITEM4_1_,
bids0_.BID_ID as BID1_1_, bids0_.BID_ID as BID1_1_0_, bids0_.amount as
amount1_0_, bids0_.date as date1_0_, bids0_.ITEM_ID as ITEM4_1_0_ from BID
bids0_ where bids0_.ITEM_ID=? INFO 2009-03-03 10:26:38,937 [Demo Hibernate] :
Item #1 : Item{id=1, bids=[Bid{id=1, amount=101.0, date=2009-03-03
10:26:38.864}, Bid{id=10, amount=110.0, date=2009-03-03 10:26:38.883}, Bid{id=8,
amount=108.0, date=2009-03-03 10:26:38.878}, Bid{id=3, amount=103.0, date=2009-
03-03 10:26:38.867}, Bid{id=6, amount=106.0, date=2009-03-03 10:26:38.871},
Bid{id=2, amount=102.0, date=2009-03-03 10:26:38.866}, Bid{id=9, amount=109.0,
date=2009-03-03 10:26:38.879}, Bid{id=5, amount=105.0, date=2009-03-03
10:26:38.87}, Bid{id=7, amount=107.0, date=2009-03-03 10:26:38.876}, Bid{id=4,
amount=104.0, date=2009-03-03 10:26:38.868}], name='TV LCD', price=340.0}
Hibernate: /* load one-to-many org.touilleur.hibernate.v1.Item.bids */ select
bids0_.ITEM_ID as ITEM4_1_, bids0_.BID_ID as BID1_1_, bids0_.BID_ID as
BID1_1_0_, bids0_.amount as amount1_0_, bids0_.date as date1_0_, bids0_.ITEM_ID
as ITEM4_1_0_ from BID bids0_ where bids0_.ITEM_ID=? INFO 2009-03-03
10:26:38,944 [Demo Hibernate] : Item #2 : Item{id=2, bids=[Bid{id=18,
amount=138.0, date=2009-03-03 10:26:38.896}, Bid{id=11, amount=10.0, date=2009-
03-03 10:26:38.885}, Bid{id=20, amount=144.0, date=2009-03-03 10:26:38.898},
Bid{id=16, amount=116.0, date=2009-03-03 10:26:38.893}, Bid{id=14, amount=84.0,
date=2009-03-03 10:26:38.888}, Bid{id=19, amount=139.0, date=2009-03-03
10:26:38.897}, Bid{id=15, amount=125.0, date=2009-03-03 10:26:38.892},
Bid{id=12, amount=22.0, date=2009-03-03 10:26:38.886}, Bid{id=13, amount=43.0,
date=2009-03-03 10:26:38.887}, Bid{id=17, amount=137.0, date=2009-03-03
10:26:38.895}], name='Plasma', price=1230.0} INFO 2009-03-03 10:26:38,947 [Demo
Hibernate] : testSelectAllItemsWithDefaultFetchMode - end Cette fois-ci nous
constatons qu'Hibernate effectue une requête que lorsque cela devient nécessaire, ce qui est adapté
dans la majorité des cas. Lorsque le mode lazy est désactivé, Hibernate effectue n+1 requêtes sans
jointure. Cela peut entraîner un très grand nombre de requêtes vers la base. Nous allons donc voir
comment optimiser dans ce cas précis le chargement des associations.
```

## Fetch strategy à join et lazy=false

Si nous ne pouvons pas utiliser le chargement tardif, il est alors intéressant d'utiliser une requête et d'effectuer une jointure entre la table Item et la table Bid. Pour cela il suffit de changer de stratégie lors de la requête. Nous passons maintenant le mode de récupération (FetchMode) à JOIN :

```
// voir la class org.touilleur.hibernate.SelectTest
Item item1 = (Item) session.createCriteria(Item.class)
    .setFetchMode("bids", FetchMode.JOIN)
    .add(Restrictions.idEq(id))
    .uniqueResult();
```

Si vous savez qu'il sera toujours plus intéressant d'utiliser une requête par jointure vous pouvez aussi changer la stratégie dans le fichier Item.hbm.xml : <set name="bids" table="ITEM\_BIDS" lazy="false" inverse="true" fetch="join" cascade="all"> Le test selectOneItemWithJoin nous montre alors qu'une seule requête est exécutée, et qu'une jointure externe gauche entre la table Item et la table Bid permet de retrouver un Item avec 0 ou plusieurs Bid. INFO 2009-02-27 22:24:21,772 [Demo Hibernate] : Starting selectOneItemWithJoin Hibernate: /\* criteria query \*/ select this\_.ITEM\_ID as ITEM1\_0\_1\_, this\_.name as name0\_1\_, this\_.price as price0\_1\_, bids2\_.ITEM\_ID as ITEM4\_3\_, bids2\_.BID\_ID as BID1\_3\_, bids2\_.BID\_ID as BID1\_1\_0\_, bids2\_.amount as amount1\_0\_, bids2\_.date as date1\_0\_, bids2\_.ITEM\_ID as ITEM4\_1\_0\_ from ITEM this\_ left outer join BID bids2\_ on this\_.ITEM\_ID=bids2\_.ITEM\_ID where this\_.ITEM\_ID = ? INFO 2009-02-27 22:24:21,775 [Demo Hibernate] : Loaded item1 with a FetchMode set to JOIN Notez qu'Hibernate utilise une jointure externe de la table Item vers la table Bid, cela nous donne alors en simplifiant le code SQL la requête suivante : select i.\*,b\* from ITEM i outer join BID b on i.ITEM\_ID=b.ITEM\_ID Par défaut il n'y a donc pas de critères de distinction, et nous nous retrouvons alors avec 20 résultats, ce qui sera pratique pour remplir un tableau mais qui n'est pas forcément souhaitable pour votre code. INFO 2009-03-03 10:30:53,041 [Demo Hibernate] : testSelectAllItemsWithJoinFetchMode - begin Hibernate: /\* criteria query \*/ select this\_.ITEM\_ID as ITEM1\_0\_1\_, this\_.name as name0\_1\_, this\_.price as price0\_1\_, bids2\_.ITEM\_ID as ITEM4\_3\_, bids2\_.BID\_ID

```

as BID1_3_, bids2_.BID_ID as BID1_1_0_, bids2_.amount as amount1_0_, bids2_.date
as date1_0_, bids2_.ITEM_ID as ITEM4_1_0_ from ITEM this_ left outer join BID
bids2_ on this_.ITEM_ID=bids2_.ITEM_ID INFO 2009-03-03 10:30:53,072 [Demo
Hibernate] : Item #1 : Item{id=1, bids=[Bid{id=3, amount=103.0, date=2009-03-03
10:30:52.999}], Bid{id=8, amount=108.0, date=2009-03-03 10:30:53.01}, Bid{id=6,
amount=106.0, date=2009-03-03 10:30:53.003}, Bid{id=10, amount=110.0, date=2009-
03-03 10:30:53.014}, Bid{id=5, amount=105.0, date=2009-03-03 10:30:53.002},
Bid{id=2, amount=102.0, date=2009-03-03 10:30:52.998}, Bid{id=7, amount=107.0,
date=2009-03-03 10:30:53.009}, Bid{id=4, amount=104.0, date=2009-03-03
10:30:53.001}, Bid{id=9, amount=109.0, date=2009-03-03 10:30:53.011}, Bid{id=1,
amount=101.0, date=2009-03-03 10:30:52.997}], name='TV LCD', price=340.0} INFO
2009-03-03 10:30:53,074 [Demo Hibernate] : Item #1 : Item{id=1, bids=[Bid{id=3,
amount=103.0, date=2009-03-03 10:30:52.999}, Bid{id=8, amount=108.0, date=2009-
03-03 10:30:53.01}, Bid{id=6, amount=106.0, date=2009-03-03 10:30:53.003},
Bid{id=10, amount=110.0, date=2009-03-03 10:30:53.014}, Bid{id=5, amount=105.0,
date=2009-03-03 10:30:53.002}, Bid{id=2, amount=102.0, date=2009-03-03
10:30:52.998}, Bid{id=7, amount=107.0, date=2009-03-03 10:30:53.009}, Bid{id=4,
amount=104.0, date=2009-03-03 10:30:53.001}, Bid{id=9, amount=109.0, date=2009-
03-03 10:30:53.011}, Bid{id=1, amount=101.0, date=2009-03-03 10:30:52.997}],
name='TV LCD', price=340.0} ... Message répété 20 fois ... INFO 2009-03-03
10:30:53,125 [Demo Hibernate] : testSelectAllItemsWithJoinFetchMode - end Pour
ajouter une clause DISTINCT, nous pouvons modifier notre critère et utiliser un ResultTransformer
:

```

```

List l = session.createCriteria(Item.class)
    .setFetchMode("bids", FetchMode.JOIN)
    .setResultTransformer(Criteria.DISTINCT_ROOT_ENTITY)
    .list();

```

En exécutant à nouveau notre test voici le résultat : INFO 2009-03-03 10:43:31,196 [Demo Hibernate] : testSelectAllItemsWithJoinFetchMode - begin Hibernate: /\* criteria query \*/ select this\_.ITEM\_ID as ITEM1\_0\_1\_, this\_.name as name0\_1\_, this\_.price as price0\_1\_, bids2\_.ITEM\_ID as ITEM4\_3\_, bids2\_.BID\_ID as BID1\_3\_, bids2\_.BID\_ID as BID1\_1\_0\_, bids2\_.amount as amount1\_0\_, bids2\_.date as date1\_0\_, bids2\_.ITEM\_ID as ITEM4\_1\_0\_ from ITEM this\_ left outer join BID bids2\_ on this\_.ITEM\_ID=bids2\_.ITEM\_ID INFO 2009-03-03 10:43:31,229 [Demo Hibernate] : Item #1 : Item{id=1, bids=[Bid{id=8, amount=108.0, date=2009-03-03 10:43:31.162}, Bid{id=7, amount=107.0, date=2009-03-03 10:43:31.16}, Bid{id=2, amount=102.0, date=2009-03-03 10:43:31.149}, Bid{id=6, amount=106.0, date=2009-03-03 10:43:31.154}, Bid{id=10, amount=110.0, date=2009-03-03 10:43:31.167}, Bid{id=9, amount=109.0, date=2009-03-03 10:43:31.163}, Bid{id=3, amount=103.0, date=2009-03-03 10:43:31.15}, Bid{id=5, amount=105.0, date=2009-03-03 10:43:31.153}, Bid{id=1, amount=101.0, date=2009-03-03 10:43:31.147}, Bid{id=4, amount=104.0, date=2009-03-03 10:43:31.151}], name='TV LCD', price=340.0} INFO 2009-03-03 10:43:31,232 [Demo Hibernate] : Item #2 : Item{id=2, bids=[Bid{id=19, amount=139.0, date=2009-03-03 10:43:31.181}, Bid{id=17, amount=137.0, date=2009-03-03 10:43:31.179}, Bid{id=15, amount=125.0, date=2009-03-03 10:43:31.176}, Bid{id=13, amount=43.0, date=2009-03-03 10:43:31.171}, Bid{id=11, amount=10.0, date=2009-03-03 10:43:31.169}, Bid{id=12, amount=22.0, date=2009-03-03 10:43:31.17}, Bid{id=14, amount=84.0, date=2009-03-03 10:43:31.172}, Bid{id=20, amount=144.0, date=2009-03-03 10:43:31.182}, Bid{id=18, amount=138.0, date=2009-03-03 10:43:31.18}, Bid{id=16, amount=116.0, date=2009-03-03 10:43:31.177}], name='Plasma', price=1230.0} INFO 2009-03-03 10:43:31,235 [Demo Hibernate] : testSelectAllItemsWithJoinFetchMode - end Voir la FAQ Hibernate [[Hibernate does not return distinct results for a query with outer join fetching enabled for a collection \(even if I use the distinct keyword\)](#)] La sélection par jointure est donc une option possible. Nous sommes passés de 3 requêtes à une seule requête, ce qui est une optimisation permettant de réduire le nombre de requête SQL. Encore une fois, c'est à vous et à votre DBA de décider de la meilleure stratégie, il ne faut donc pas appliquer à l'aveugle les paramètres. Les jointures sont efficaces si vos colonnes de jointure sont correctement indexées, il est donc important de s'assurer que votre administrateur de base de données a aussi validé vos changements.

## Fetch mode par subselect

Hibernate propose un mode de sélection moins connu et pourtant très efficace sur les associations, le mode subselect. L'idée est simplement d'utiliser le résultat d'une première requête comme critère de la sélection de la deuxième requête pour effectuer une requête plus efficace. Ce que nous avons vu en premier, en mode lazy=false et fetch=select, c'est qu'Hibernate effectue n+1 requêtes pour charger la liste des Bids pour chacun des Items.

```

/* Retrouve la liste des Items */
select * from ITEM

/* Retrouve la list des Bids pour chaque Item */
select * from BID where ITEM_ID = 12
select * from BID where ITEM_ID = 34
select * from BID where ITEM_ID = 37
select * from BID where ITEM_ID = 39

```

Il y a deux façons d'optimiser ce type de requête : soit utiliser le paramètre batch-size dans l'élément set du fichier Item.hbm.xml, soit utiliser une requête de sous sélection. Je ne sais pas si ce mode est supporté par toutes les bases de données, mais voici le principe : - Sélectionner les Items - Sélectionner les Bids en utilisant les ID des items précédemment trouvé

```

/* Retrouve la liste des Items */
select * from ITEM

/* Retrouve la list des Bids pour chaque Item */
select * from BID where ITEM_ID IN( 12, 34, 37, 39)

```

Pour cela il n'est pas possible de changer le mode de sélection dans le code, il faut éditer le fichier Item.hbm.xml et préciser que le fetch type est **subselect**

```

<hibernate-mapping>
  <class name="org.touilleur.hibernate.v1.Item" table="ITEM">
    <id name="id" column="ITEM_ID">
      <generator class="native"/>
    </id>
    <property name="name"/>
    <!-- Nous testons subselect -->
    <property name="price"/>
    <set name="bids" table="ITEM_BIDS" lazy="false"
      inverse="true"
      fetch="subselect"
      cascade="all">
      <key column="ITEM_ID" not-null="true"/>
      <one-to-many class="org.touilleur.hibernate.v1.Bid"/>
    </set>
  </class>
</hibernate-mapping>

```

Après avoir édité le fichier Item.hbm.xml, le test unitaire

```

testSelectAllItemsWithSubSelectFetchMode nous montre que deux requêtes sont exécutées au lieu de n+1 dans le cas du mode "select" ou une seule requête dans le cas du mode "join" : INFO 2009-03-03 11:06:36,678 [Demo Hibernate] : testSelectAllItemsWithSubSelectFetchMode - begin Hibernate: /* criteria query */ select this_.ITEM_ID as ITEM1_0_0_, this_.name as name0_0_, this_.price as price0_0_ from ITEM this_ Hibernate: /* load one-to-many org.touilleur.hibernate.v1.Item.bids */ select bids0_.ITEM_ID as ITEM4_1_, bids0_.BID_ID as BID1_1_, bids0_.BID_ID as BID1_1_0_, bids0_.amount as amount1_0_, bids0_.date as datel_0_, bids0_.ITEM_ID as ITEM4_1_0_ from BID bids0_ where bids0_.ITEM_ID in (select this_.ITEM_ID from ITEM this_) INFO 2009-03-03 11:06:36,711 [Demo Hibernate] : Item #1 : Item{id=1, bids=[Bid{id=7, amount=107.0, date=2009-03-03 11:06:36.646}, Bid{id=3, amount=103.0, date=2009-03-03 11:06:36.636}, Bid{id=8, amount=108.0, date=2009-03-03 11:06:36.647}, Bid{id=2, amount=102.0, date=2009-03-03 11:06:36.635}, Bid{id=6, amount=106.0, date=2009-03-03 11:06:36.64}, Bid{id=5, amount=105.0, date=2009-03-03 11:06:36.639}, Bid{id=1, amount=101.0, date=2009-03-03 11:06:36.633}, Bid{id=9, amount=109.0, date=2009-03-03 11:06:36.648}, Bid{id=4, amount=104.0, date=2009-03-03 11:06:36.637}, Bid{id=10, amount=110.0, date=2009-03-03 11:06:36.652}], name='TV LCD', price=340.0) INFO 2009-03-03 11:06:36,714 [Demo Hibernate] : Item #2 : Item{id=2, bids=[Bid{id=14, amount=84.0, date=2009-03-03 11:06:36.657}, Bid{id=18, amount=138.0, date=2009-03-03 11:06:36.664}, Bid{id=17, amount=137.0, date=2009-03-03 11:06:36.663}, Bid{id=20, amount=144.0, date=2009-03-03 11:06:36.666}, Bid{id=15, amount=125.0, date=2009-03-03 11:06:36.66}, Bid{id=12, amount=22.0, date=2009-03-03 11:06:36.655}, Bid{id=13, amount=43.0, date=2009-03-03 11:06:36.656}, Bid{id=19, amount=139.0, date=2009-03-03 11:06:36.665}, Bid{id=16, amount=116.0, date=2009-03-03 11:06:36.661}, Bid{id=11, amount=10.0, date=2009-03-03 11:06:36.654}], name='Plasma', price=1230.0) INFO 2009-03-03 11:06:36,716 [Demo Hibernate] : testSelectAllItemsWithSubSelectFetchMode - end L'intérêt de ce mode de sélection est de récupérer en deux requêtes l'ensemble des Items et ensuite des Bids. Si vous faites enfin le test en mode "lazy=true" vous verrez par ailleurs qu'Hibernate n'effectue pas de chargement tardif mais qu'il charge toutes les données. Le mode de sélection en "SUBSELECT" limite à deux requêtes le nombre d'interrogation nécessaire, le tout sans jointure. C'est donc une option intéressante à tester selon la logique de votre application.

```

## Fetch mode par batch size

La 4ème technique pour améliorer les chargements par lot est d'activer l'option batch-size. Afin de montrer son fonctionnement nous allons d'abord limiter la taille de chargement à 1 pour le nombre d'Item et le nombre de collection de Bid :

```

<hibernate-mapping>
  <class name="org.touilleur.hibernate.v1.Item" table="ITEM" batch-size="1">
    <id name="id" column="ITEM_ID">
      <generator class="native"/>
    </id>
    <property name="name"/>
    <!-- The fetch join strategy uses a select join query to load with one transact the set of bids -->
    <property name="price"/>
    <set name="bids" table="ITEM_BIDS"
      lazy="false"
      inverse="true"
      fetch="select"
      cascade="all"
      batch-size="1">
      <key column="ITEM_ID" not-null="true"/>
      <one-to-many class="org.touilleur.hibernate.v1.Bid"/>
    </set>
  </class>
</hibernate-mapping>

```

Nous voyons qu'à l'exécution, Hibernate effectue une requête pour récupérer une liste de deux Items puis ensuite deux requêtes pour chacun des Items. En mode lazy, nous aurions simplement vu que la deuxième requête de chargement aurait été effectuée plus tardivement INFO 2009-03-03 13:19:19,261 [Demo Hibernate] : testSelectAllItemsWithSubSelectFetchMode - begin Hibernate: /\* criteria query \*/ select this\_.ITEM\_ID as ITEM1\_0\_0\_, this\_.name as name0\_0\_, this\_.price as price0\_0\_ from ITEM this\_ Hibernate: /\* load one-to-many org.touilleur.hibernate.v1.Item.bids \*/ select bids0\_.ITEM\_ID as ITEM4\_1\_, bids0\_.BID\_ID as BID1\_1\_, bids0\_.BID\_ID as BID1\_1\_0\_, bids0\_.amount as amount1\_0\_, bids0\_.date as datel\_0\_, bids0\_.ITEM\_ID as ITEM4\_1\_0\_ from BID bids0\_ where bids0\_.ITEM\_ID=? Hibernate: /\* load one-to-many org.touilleur.hibernate.v1.Item.bids \*/ select bids0\_.ITEM\_ID as ITEM4\_1\_,

```

bids0_.BID_ID as BID1_1_, bids0_.BID_ID as BID1_1_0_, bids0_.amount as
amount1_0_, bids0_.date as date1_0_, bids0_.ITEM_ID as ITEM4_1_0_ from BID
bids0_ where bids0_.ITEM_ID=? INFO 2009-03-03 13:19:19,294 [Demo Hibernate] :
Item #1 : Item{id=1, bids=[Bid{id=2, amount=102.0, date=2009-03-03
13:19:19.217}, Bid{id=5, amount=105.0, date=2009-03-03 13:19:19.221}, Bid{id=3,
amount=103.0, date=2009-03-03 13:19:19.218}, Bid{id=10, amount=110.0, date=2009-
03-03 13:19:19.233}, Bid{id=7, amount=107.0, date=2009-03-03 13:19:19.227},
Bid{id=4, amount=104.0, date=2009-03-03 13:19:19.219}, Bid{id=1, amount=101.0,
date=2009-03-03 13:19:19.215}, Bid{id=9, amount=109.0, date=2009-03-03
13:19:19.23}, Bid{id=8, amount=108.0, date=2009-03-03 13:19:19.229}, Bid{id=6,
amount=106.0, date=2009-03-03 13:19:19.222}], name='TV LCD', price=340.0} INFO
2009-03-03 13:19:19,297 [Demo Hibernate] : Item #2 : Item{id=2, bids=[Bid{id=19,
amount=139.0, date=2009-03-03 13:19:19.247}, Bid{id=16, amount=116.0, date=2009-
03-03 13:19:19.243}, Bid{id=11, amount=10.0, date=2009-03-03 13:19:19.236},
Bid{id=18, amount=138.0, date=2009-03-03 13:19:19.246}, Bid{id=13, amount=43.0,
date=2009-03-03 13:19:19.237}, Bid{id=20, amount=144.0, date=2009-03-03
13:19:19.248}, Bid{id=14, amount=84.0, date=2009-03-03 13:19:19.238}, Bid{id=15,
amount=125.0, date=2009-03-03 13:19:19.242}, Bid{id=17, amount=137.0, date=2009-
03-03 13:19:19.245}, Bid{id=12, amount=22.0, date=2009-03-03 13:19:19.236}],
name='Plasma', price=1230.0} INFO 2009-03-03 13:19:19,300 [Demo Hibernate] :
testSelectAllItemsWithSubselectFetchMode - end Le chargement par lot est une technique
efficace qui permet de précharger un nombre défini de proxy non initialisé si un premier proxy est
accédé. Il y a donc l'optimisation au niveau du chargement de l'entité (Item) et l'optimisation au
niveau du chargement de la collection (Bid). Le chargement par lot pour notre classe Item
fonctionne de la manière suivante : dans votre base imaginons que nous ayons 10 Items, chacun de
ces Items référencent un Acheteur avec une relation un-à-un. La relation est mappée avec un proxy
en mode lazy="true". Si par défaut vous chargez la liste des Items puis qu'ensuite vous appelez la
méthode getBuyer() pour retrouver l'acheteur, Hibernate effectuera alors 10 requêtes SQL (10
items). Le chargement par lot permet de spécifier à Hibernate la taille d'une fenêtre de chargement,
en utilisant les clés primaires ou les clés étrangères, qui ici seraient dans la table BUYER. <class
name="Item" batch-size="10">...</class> Hibernate exécutera non plus 10 requêtes SQL
mais une seule pour charger l'ensemble des acheteurs. Idéalement vous l'aurez deviné, ce batch-
size correspond au nombre d'éléments affichés sur une page Web, dans un tableau paginé par
exemple. Ensuite il est possible d'activer le chargement par lot sur les collections. Notre Item a une
collection de Bid. Imaginons que je charge mes 10 Items dans ma Session Hibernate. Par défaut,
chaque appel à getBids() entrainera alors une requête SQL, soit 10 requêtes pour charger chacune
des collections. En spécifiant une taille de batch comme ci-dessous il est possible d'optimiser le
nombre d'éléments préchargés et donc de réduire le nombre de requête SQL :

```

```

<class name="Item">
  <set name="Bid" batch-size="3">
    ...
  </set>
</class>

```

Pour 7 enchères, Hibernate chargera alors 3,3,1 en effectuant 3 requêtes au lieu de 7 requêtes par exemple. Cela permet de réduire encore une fois le nombre de requête, et cette valeur doit être guidée par le code de votre application et l'usage que vous en faites. Pour cette raison il est souvent plus souhaitable de définir la taille du batch au niveau du code comme dans ci-dessous:

```

List l = session.createCriteria(Item.class)
    .setFetchSize(10)
    .list();

```

Petite astuce au passage : si vous souhaitez limiter le nombre de résultat retourné, vous connaissez sans doute la commande setMaxResults(int i).

```

List l = session.createCriteria(Item.class)
    .setMaxResults(10)
    .list();

```

Par curiosité je me suis un peu demandé pourquoi Hibernate prend plus de temps que la même requête SQL exécutée sur ma base de test. En fouillant un peu je me suis aperçu qu'il est intéressant de préciser la taille du batch afin de ne récupérer qu'un seul lot :

```

List l = session.createCriteria(Item.class)
    .setMaxResults(10)
    .setFetchSize(10)
    .list();

```

Essayez sur votre code vous verrez la différence.

## Conclusion de la première partie

Lorsque le chargement tardif ne peut pas être utilisé, il est important de vérifier que la stratégie par défaut d'Hibernate n'entraîne pas un nombre de requêtes SQL trop importantes. Il est possible de limiter ce nombre en utilisant des jointures ou des selections imbriquées. A vous ensuite d'optimiser vos réglages selon votre application. Dans la deuxième partie nous allons voir les 3 paramètres différents de chargement des collections. J'espère que vous aurez apprécié. Si vous voulez aller un peu plus loin, il ne vous reste plus qu'à télécharger le code de l'article et à commencer à tester les différentes configurations possibles d'Hibernate. Pour les personnes à la recherche d'un tutorial sur Hibernate, la première partie vous montre un exemple simple de mapping pour commencer à apprendre Hibernate. Bonne lecture et à bientôt pour la deuxième partie ! [Code source de l'article ArticleTouilleur hibernate.tar.gz](#)

# Integrate Apache Camel with Apache Wicket

**This article is a translation of [an original article](#) I published a while ago about how to integrate Apache Camel with Apache Wicket.** In this article I propose that we have a look on a very simple usage of the lightweight integration framework Apache Camel with one of the most popular Java Web framework : Apache Wicket. The article's source code is available at the end of this article, ready to be compiled, so that you can test freely from home. If you'd like to reuse this article or its source code, please contact me before since it is not free for distribution. My email is displayed on the upper left logo. **What is Apache Wicket ?** [Wicket](#) is an open-source java framework that unlike a framework such as Struts 1, is component oriented. It lets you define your pages using plain Java, and then integrate them in static HTML source code with light tags, so that everything is displayed with a light application server such as Apache Tomcat for instance. When I was a projet leader at Reuters last year, my job was to be the person in charge for a proprietary framework, Karma, which is very close to Wicket. Thus it wasn't very hard to jump into the Wicket world. Component oriented frameworkd such as Wicket eases the development and is more close to what we've been doing for decade with Swing. In this article we will see how to create your first Wicket page, a custom component, and integrate Spring with Apache Camel to load our configuration. The most interesting concept about Wicket is that the event model is very close to Swing. You basically implement listeners and based on the Observer-Observable pattern, the Wicket engine notifies your logic depending on the user action on the web page. With the support of Ajax, it's a very interesting solution to learn. **What is Apache Camel ?** Apache Camel is an open-source framework based on Spring Core framework. It's an implementation of the populars EIP (Enterprise Integration Patterns) that offers either XML for configuration (like Mule) or a nice DSL based on Java, which I personally prefer. We will use the Java DSL in this article. Apache Camel is a good add-on for Mule. For me, Mule is a lightweight integration framework, not only an ESB. Camel is a toolbox that you can also integrate with Mule. In this architecture, Mule will be in charge for I/O and the Threading model. Camel would be in charge of routing, transform and mediation tasks. Here we will use only Camel, since my solution is pretty simple. **Content of the demonstration** I started to work on this article in november, with the aim to present this work at Devovx during my [quickie on ESB at Devovx 2008](#). I quickly realized that it wouldn't fit in the amount of time allocated for a quickie. I posted the french version in january and now you're reading the english translation. ... **and why didn't you take Mule/Spring Integration/Apache ServiceMix/OpenESB/AnyOtherIntegrationFramework ?** I really appreciate the hard work behind the other Integration framework but for the purpose of this article, Camel is enough. It would be very interesting to publish the same article based on Spring Integration for instance. Let me know if you plan to write such article so that we can reference it here ! **The sample application** Let's start with what you will see once completed. Remember : the source code is available at the end of this article. Unzip, open a xterm and simply type "mvn jetty:run". Open a web browser and here is what you'll see:

The screenshot displays a web application interface. At the top, there is a green header with the InnoTeria logo and navigation icons (home, search, mail). Below the header, there are two columns: 'New' and 'Forum'. The main content area is titled 'events' and shows '11th dec 2008'. To the right, there is a 'Devoxx Quickie demo' section with a welcome message and a form. The form has two input fields: 'Your name' and 'Your message', followed by a 'Send your request' button. Below the form, there is a 'Check incident mailbox...' link. The right sidebar shows a phone icon and the number '+33 6 63 20 48 50'.

As a customer I put my name in a form, and a little message to ask why my great PS3 wasn't delivered for Christmas. I click the submit button and a nice "Thanks" message is displayed. So

To send a request to the application, please fill the form and click the Submit button. Then click on the "Check mailbox" hyperlink at the bottom of this page.

Your name

Your message:

---

**Thanks!** 

[Check incident mailbox...](#)

far, so good.

**What's happening in the background ?** When I click the Send button, Wicket generates a message for Apache Camel. It transforms this message and store-it to a file, to demonstrate how to save a message on the local filesystem. I don't say this is what you should do. I just want to demonstrate some of Camel's features. So please don't blame me for what the application is doing, we're more interested by HOW it is doing it. We have another async process in Camel that polls this folder. For any incoming file, it reads it, transforms it using Velocity and then generate an email. We will use an outgoing silo so that Apache Camel send to an SMTP server the email. For my article I use MockMail, it's a fake SMTP server that's really easy to use, especially for JUnit test. **The administrator view** The admin page is a wicket page that reads the inbox with POP3 and display a list of customer's requests to the web site administrator. We will see how it is simple to implement our own Wicket component to iterate a collection of Messages. **First steps with Wicket, "Bonjour le monde"/"Hello world"** Download the source code and open InnoteriaWicketApplication java class with your favorite IDE (mine is IDEA IntelliJ). This class is our Wicket main application entry point. I use [Mock JavaMail](#) to declare a MailBox. The class ReportIncidentEndpoint is one of my Apache Camel component. It's the gate to the Camel world here for Wicket.

```
public class InnoteriaWicketApplication extends WebApplication {
    private ApplicationContext ctx;
    private ReportIncidentEndpoint endpoint;
    private Mailbox mailbox; // mock javamail

    private static ISpringContextLocator CTX_LOCATOR = new ISpringContextLocator() {
        public ApplicationContext getSpringContext() {
            return InnoteriaWicketApplication.get().ctx;
        }
    };

    public InnoteriaWicketApplication() {
    }

    public Class getHomePage() {
        return HomePage.class;
    }

    public static InnoteriaWicketApplication get() {
        return (InnoteriaWicketApplication) Application.get();
    }

    @Override
    protected void init() {
        ctx = new ClassPathXmlApplicationContext("applicationContext.xml");
        Address address= null;
        try {
            address = new InetAddress("incident@innoteria.fr");
        } catch (AddressException e) {
            e.printStackTrace();
        }
        mailbox = new Mailbox(address);
    }

    /**
     * This trick is used to decorate a specified template class with a SpringBeanLocator
     * so that I can use Spring to load my Apache Camel configuration.
     * @param clazz will be the endpoint to decorate
     * @param <T> template
     * @return a decorated class.
     */
    private <T> T createProxy(Class<T> clazz){
        return (T) LazyInitProxyFactory.createProxy(clazz, new SpringBeanLocator(clazz, CTX_LOCATOR));
    }

    public ReportIncidentEndpoint getReportIncidentEndpoint(){
        if(endpoint==null){
            endpoint=createProxy(ReportIncidentEndpoint.class);
        }
        return endpoint;
    }
}
```

```
}

```

It is now time to write your first page, the HomePage class. It extends the WebPage class. We see that a FeedbackPanel component has been declared with 2 text fields. We also added a Form. The onSubmit method will be triggered when the user submits the form from the web page. We will see later how an instance of MyReportIncident is used to hold details later. In a few lines, as you can see it's pretty simple to implement a Form and to trigger an action when the user clicks a button.

```
public class HomePage extends WebPage {

    private static final long serialVersionUID = 1L;

    public HomePage(final PageParameters parameters) {
        // un de mes composants
        FeedbackPanel feedbackPanel = new FeedbackPanel("feedback");
        add(feedbackPanel);

        final TextField yourName = new TextField("yourname", new Model());
        final TextField yourMessage = new TextField("yourmessage", new Model());
        yourMessage.setRequired(true);
        yourName.setRequired(true);

        Form form = new Form("formtest") {
            protected void onSubmit() {
                ReportIncidentEndpoint service = InnoteriaWicketApplication.get().getReportIncidentEndpoint();
                service.reportIncident(
                    new MyReportIncident(
                        yourName.getModelObject().toString(),
                        yourMessage.getModelObject().toString()
                    ));
                info("Thanks!");
            }
        };

        form.add(yourName);
        form.add(yourMessage);

        // Add a form with an onSubmit implementation that sets a message
        add(form);

        /* java code */
        add(new BookmarkablePageLink("link", CheckMailbox.class));
    }
}

```

Let's see on the HTML side how does it work. If you open HomePage.html we see:

```
...
```

Your name

Your message:

```
...
```

Wicket relies on the ID declared in the HTML to create a proxy between the HTML view and the Java side. Wicket really lets you play and design the HTML without the need to use complex components syntax. The HTML code is clean and you can even preview it with IDEA IntelliJ to validate that the layout is correct. Enough for my short introduction to Wicket, I let you have a look at the code while I prepare the next step. Browse the source code, try to compile and deploy with maven 2 (mvn jetty:run) and when you're ok to continue, come back here. **The Apache Camel side** Our entry point in Camel is ReportIncidentEndpoint. The reportIncident method is notified by Wicket when the user clicks the send button in the user interface. It publishes to an in-memory Queue handled by Apache Camel. ReportIncidentEndpoint is a Spring Bean which gets injected into InnoteriaWicketApplication.

```
/**
 * @author Nicolas Martignole http://touilleur-express.fr
 */
public class ReportIncidentEndpoint implements Serializable {
    private CamelContext context;
    private static final long serialVersionUID=2233L;

    public ReportIncidentEndpoint() throws Exception {
        // create the context
        context = new DefaultCamelContext();

        // append the routes to the context
        context.addRoutes(new ReportIncidentRoutes());
    }
}

```

```

    // at the end start the camel context
    context.start();
}

/**
 * Callback method executed from the HomePage when the user
 * clicks on submit.
 * @param message is a message created by Wicket
 * @return a status message that is shown on HomePage.
 */
public String reportIncident(MyReportIncident message) {
    // create the producer template to use for sending messages
    ProducerTemplate producer = context.createProducerTemplate();

    // Send the message to an in-memory queue and return
    // See GenerateEmailStoreToFilePollFolderAndSendFile
    // it will read the message from direct:start in-memory queue
    Object mailBody = producer.sendBody("direct:start", message);
    System.out.println("Body:" + mailBody);

    return "OK";
}
}

```

Let's see now how we integrate our application with Apache Camel. I stated explicitly 2 roads to show you one of Camel's principle. Open ReportIncidentRoutes in your editor. This class extends RouteBuilder, a Camel super class that lets you define the logical path within your application. Please note that XML is also an option. Note that it is also possible to use XML to define your route, but here I find elegant the java definition of your path, especially when your architecture will not change any further. The first road: - reads the in-memory queue "direct:start" - calls a Velocity template to generate a message - saves set the file on the file system. Please note that it would be very simple to replace the "save file on disk" operation with a "generate and send JMS Message" to a Queue. For the purpose of this article it's simpler to use an in-memory queue or the file system.

```

public class ReportIncidentRoutes extends RouteBuilder {
    // Any message posted on direct:start will be
    // handled by velocity template, then persisted on the file system
    public void configure() throws Exception {
        from("direct:start")
            .to("velocity:MailBody.vm")
            .setHeader(FileComponent.HEADER_FILE_NAME, "errorEmail.txt")
            .to("file:///./target/demoCamel?append=false");
        ...
    }
}

```

Let's add another Camel path/road. The second road : - polls the folder for any incoming file - adds an email subject to the message - connects to a real SMTP server and publish the message Do you remember that we created a MockMail Mailbox in InnoteriaWicketApplication ? Once this application is deployed, MockMail starts to listen for incoming SMTP requests on port 25. Try a telnet localhost 25, you should see that someone is listening ! So our application will really send an email. You can adapt the source code and use the GMail SMTPs server if you want to complete this article. Here, the user name is nicolas. We will use it later to retrieve the list of email with POP3.

```

from("file:///./target/demoCamel")
    .setHeader("subject", constant("New incident reported"))
    .to("smtp://nicolas@localhost?password=secret&to=incident@innoteria.fr");

```

Apache Camel is a framework that can simplify your source code. I picked another sample to demonstrate one of its nice feature : a simple rules engine that would be enough in most use cases : // sample 1 if the header key foo is equals to bar then forward to b queue  
from("queue:a").filter(header("foo").isEqualTo("bar")).to("queue:b"); // sample 2  
from("queue:c").choice().when(header("foo").isEqualTo("bar")).to("queue:d")  
.when(header("foo").isEqualTo("cheese")).to("queue:e")  
.otherwise().to("queue:errorQueue"); Apache Camel is an interesting solution. It aims to reduce the amount of technical code in your application, usually required to read a file, generate an email and send it to a SMTP server. The idea behind this framework is to ease integration with external systems such as JMS, SSH, FTP, File, SMTP or FTP. Don't hesitate to go on Apache Camel web site and have a look at the documentation, there's a lot of interesting use cases. **How can I read the mailbox content and display a list of emails ?** Let's go back to Wicket. We will see in a minut how to create your own component. The next user story is : how can I read the list of incoming customer's request from my administrator web page ? If you click on the "Check Incident Mailbox..." on the home page you'll see this screen :



CheckMailbox class is a Wicket web page that fetches the list of email sent to incident AT innoteria.fr. An Iterator of Message is used by our custom component to display each message. I had to add an Adapter for Message, cause javax.mail.Message is not Serializable. Wicket requires Serializable instance. This is an usage of Iterator and Adapter pattern by the way.

```
public class CheckMailbox extends WebPage {
    private static final long serialVersionUID = 1L;

    public CheckMailbox(final PageParameters parameters) {
        Address address = null;
        try {
            address = new InternetAddress("incident@innoteria.fr");
        } catch (AddressException e) {
            e.printStackTrace();
        }
        Mailbox mailbox = Mailbox.get(address);
        // javax.mail.message subclasses are not serializable
        // thus we need this little trick to adapt our collection.
        // I don't wan use collections-utils and folks, it's too simple.
        InnoteriaMessageIterator imi=new InnoteriaMessageIterator(mailbox.iterator());
        MessagePanel rv = new MessagePanel("repeatingPanel",imi);
        add(rv);
    }
}
```

**My first component** The class MessagePanel is a Wicket component, attached with its own subset of HTML. DisplayMessagePanel is a Panel included in MessagePanel. When I wrote my code I thought it would be useful to add some more components, but I didn't finish. Anyway... I really appreciate how Wicket describes components.

```
public class DisplayMessagePanel extends Panel {

    public DisplayMessagePanel(String id, final InnoteriaMessageIterator listOfMessages) {
        super(id);

        add(new RefreshingView("repeatingView") {

            @Override
            protected Iterator getItemModels() {
                return new ModelIteratorAdapter(listOfMessages.iterator()) {
                    @Override
                    protected IModel model(Object o) {
                        return new CompoundPropertyModel((InnoteriaMessage) o);
                    }
                };
            }

            @Override
            protected void populateItem(Item item) {
                // Those 3 properties are read directly on InnoteriaMessage POJO
                // See CompoundPropertyModel javadoc for more details
                item.add(new Label("subject"));
                item.add(new Label("from"));
                item.add(new Label("content"));
            }
        });
    }
}
```

**Conclusion** I hope you appreciate, I spent a lot of time to write this article. The idea was to demonstrate Apache Wicket and Apache Camel with a real use case. It would be interesting to write the Spring Integraton or the Mule version. Not a lot of work but I have already some article to complete... If you appreciate reading this article, please vote for it on [DZone](#) Source code: [touilleur-wicket-camel-1.0.tar.gz](http://touilleur-wicket-camel-1.0.tar.gz)

## Jazoon 2009

# JAZOON09

THE INTERNATIONAL CONFERENCE ON JAVA TECHNOLOGY  
JUNE 22–25, 2009 ZURICH

[Jazoon 2009](#)

est une conférence sur Java organisée fin juin à Zurich. L'an passé c'était 6 Keynotes, 76 conférences techniques et 9 tutoriaux, ce qui en fait une excellente occasion de se tenir au courant de l'actualité de Java, à 6 mois de Devovx. C'est le deuxième événement européen sur Java, accessible facilement pour tout le monde. Au programme cette année, une nouveauté que Christian Frei nous a présenté lors de la soirée du Paris JUG, les Jazoon Rookie. L'idée de ce programme est de proposer à de jeunes et talentueux développeurs de venir présenter un sujet de 20 minutes à Jazoon. Si vous 26 ans ou moins, vous pouvez tenter votre chance et ensuite gagner une invitation gratuite, tout frais payés, afin d'y participer cet été. Une présentation de 20 minutes, c'est un exercice vraiment sympa. J'ai tenté l'expérience à Devovx en décembre dernier, un grand moment de stress et de bonheur. Un peu comme découvrir une piste noir la première fois en ski. Les détails sont sur [la page Jazoon Rookie](#). Je participerai officiellement à Jazoon cette année en tant que blogueur : d'ici à juin je vous tiendrai informé de l'actualité des conférences, et sur place je couvrirai les différentes conférences pendant 3 jours. Encore un très bon programme (et du travail) en perspective.

## Soirée du Paris JUG et inauguration du French SUG

Deux événements sur mon agenda : mardi 10 mars soirée Web avec Apache Wicket et le Web sémantique [au Paris JUG](#). La première présentation organisée par Carl Azoury et Nicolas Andre de Zenika a pour thème Apache Wicket. La deuxième partie traite du Web sémantique par Alexandre Bertails d'Atos Origin. Plus de détails [sur cette page](#). La semaine suivante à Paris aura lieu l'inauguration du French Scrum User Group. Le French SUG est une nouvelle association libre et ouverte à tous. Elle a pour objectif la promotion des pratiques de Scrum en France. Jeudi 19 mars, la soirée inaugurale sera placée sous le parrainage exceptionnel de Jeff Sutherland, l'un des cofondateurs de Scrum. Les détails de la soirée sont sur [la page MeetUp](#). Il y a déjà plus de 130 personnes inscrites, ce qui promet un événement exceptionnel. Enfin avec 250 personnes inscrites sur le groupe MeetUp, c'est un vrai plébiscite. A mardi ou à jeudi prochain pour ceux qui lisent le Touilleur Express.

## Hibernate : le chargement tardif c'est extra

Voici la suite de [mon article précédent](#) sur Hibernate et le chargement des associations. Il existe 3 paramètres possible pour l'élément lazy lors de la définition d'une association dans une entité : - false - true - extra

...

...

Le test est simple : j'édite le fichier Item.hbm.xml et j'active le chargement tardif (lazy=true). Le code Java affiche le nom de l'item chargé puis ensuite le nombre d'enchères sur l'Item en utilisant la méthode size() sur le HashSet. **Code Java**

```
long start = System.currentTimeMillis();
Session session = HibernateUtil.getSessionFactory().getCurrentSession();
session.beginTransaction();
item1 = (Item) session.load(Item.class, new Integer(id));
logger.info("Item name is ["+item1.getName()+"]");
logger.info("Size of Bids Set : "+ item1.getBids().size()) ;
logger.info("Ellapsed: " + (System.currentTimeMillis() - start));
```

### Résultats de l'exécution

```
Hibernate: /* load org.touilleur.hibernate.v1.Item */
select item0_.ITEM_ID as ITEM1_0_0_, item0_.name as name0_0_, item0_.price as price0_0_
from ITEM item0_ where item0_.ITEM_ID=?
INFO 2009-03-09 22:12:20,234 [Demo Hibernate] : Item name is [Item 1]
Hibernate: /* load one-to-many org.touilleur.hibernate.v1.Item.bids */
select bids0_.ITEM_ID as ITEM4_1_, bids0_.BID_ID as BID1_1_,
bids0_.BID_ID as BID1_1_0_,
bids0_.amount as amount1_0_, bids0_.date as date1_0_,
bids0_.ITEM_ID as ITEM4_1_0_
from BID bids0_ where bids0_.ITEM_ID=?
INFO 2009-03-09 22:12:20,244 [Demo Hibernate] : Size of Bids Set : 10
INFO 2009-03-09 22:12:20,244 [Demo Hibernate] : Ellapsed: 23
```

On constate qu'Hibernate effectue tout d'abord une requête sur la table Item, affiche le nom de l'item puis ensuite effectue le chargement de l'ensemble des Bids en spécifiant l'id de l'Item. Je vois plusieurs soucis : le chargement de l'ensemble des colonnes de la table Bid, la deuxième

requête, et finalement la requête SQL n'est pas franchement optimisée : nous voulons afficher la taille de la collection, or Hibernate charge l'ensemble des éléments... Ce qui veut dire que 10 objets de type Bid seront créés et que la taille du hashSet de bids sera calculée en effectuant un appel à la méthode hashSet.size(). Je ne suis pas sûr d'être clair : pour retourner la taille de la collection, Hibernate charge toute la collection et retourne la taille de la liste... Vous imaginez la consommation en terme de mémoire ? Je serai presque tenté de dire qu'Hibernate n'est pas franchement écologique. Voyons ce qu'il se passe lorsque le chargement tardif est désactivé en mettant lazy=false dans l'attribut set, dans le fichier Item.hbm.xml

```
INFO 2009-03-09 22:17:20,756 [Demo Hibernate] : testLazyFalse
Hibernate: /* load org.touilleur.hibernate.v1.Item */
select item0_.ITEM_ID as ITEM1_0_, item0_.name as name0_0_,
       item0_.price as price0_0_ from ITEM item0_
where item0_.ITEM_ID=?
Hibernate: /* load one-to-many org.touilleur.hibernate.v1.Item.bids */
select bids0_.ITEM_ID as ITEM4_1_, bids0_.BID_ID as BID1_1_,
       bids0_.BID_ID as BID1_1_0_, bids0_.amount as amount1_0_,
       bids0_.date as date1_0_, bids0_.ITEM_ID as ITEM4_1_0_
from BID bids0_
where bids0_.ITEM_ID=?
INFO 2009-03-09 22:17:20,777 [Demo Hibernate] : Item name is [Item 1]
INFO 2009-03-09 22:17:20,778 [Demo Hibernate] : Size of Bids Set : 10
INFO 2009-03-09 22:17:20,778 [Demo Hibernate] : Ellapsed: 21
```

Nous constatons que la seule différence, c'est qu'Hibernate effectue toutes les requêtes avant l'exécution du code... Mais encore une fois, si notre code se contentait d'afficher par exemple le nombre d'enchères (ici 10) nous aurions chargé encore une fois l'arbre d'enchères pour rien. Bref pas d'optimisation en vue pour l'instant. Voyons pour terminer ce qu'Hibernate fait lorsque le paramètre lazy est mis à **extra** :

...

...

L'exécution nous donne le résultat suivant:

```
Hibernate: /* load org.touilleur.hibernate.v1.Item */
select item0_.ITEM_ID as ITEM1_0_, item0_.name as name0_0_,
       item0_.price as price0_0_
from ITEM item0_
where item0_.ITEM_ID=?
INFO 2009-03-09 22:25:10,702 [Demo Hibernate] : Item name is [Item 1]
Hibernate:
select count(BID_ID) from BID where ITEM_ID =?
INFO 2009-03-09 22:25:10,707 [Demo Hibernate] : Size of Bids Set : 10
INFO 2009-03-09 22:25:10,707 [Demo Hibernate] : Ellapsed: 17
```

Cette fois-ci Hibernate utilise un SELECT COUNT pour calculer la taille de la collection **sans** la charger, ce qui est d'une part plus rapide, et d'autre part plus économique en terme de données chargées et d'instance préparées. La valeur **lazy** est un moyen d'indiquer à Hibernate que l'objet Père (ici Item) contient une collection d'enfants (Bids) très grande. Si votre application calcule la taille du Set avec la méthode size() ou que vous testez par exemple si la collection est vide avec getBids().isEmpty(), Hibernate effectue une requête de type SELECT COUNT, ce qui revient donc à éviter de charger le contenu des enchères. C'est donc une optimisation possible dans une application afin d'éviter de charger un arbre d'objet trop important dans votre code. Si vous utilisez les annotations voici comment déclarer ce paramètre :

```
// Code de la classe Item
@OneToMany
@org.hibernate.annotations.LazyCollection(org.hibernate.annotations.LazyCollectionOption.EXTRA)
private Set bids = new HashSet();
```

Voilà pour la deuxième partie, j'ai encore quelques idées que je vous proposerai dans d'autres articles prochainement.

## Spring : keep-it under control

*Note à mes collègues en particulier : on en parle après le stand-up ?* Dans notre application cet après-midi j'ai vu un problème de déclaration de propriétés Spring au moment du chargement de mon application. Un fichier de Properties chargé par Spring lors du démarrage par un [PropertyPlaceholderConfiguration](#) n'était pas à jour, il me manquait 3 propriétés... soit. Quelque part, je me demande si le fait de retarder finalement tardivement cette vérification est une si bonne idée. J'ai dû attendre le démarrage de mon application sur Weblogic et ensuite l'exécution de mon premier batch pour le voir... C'est la vie. Je n'ai plus qu'à aller m'immoler pour ne pas avoir pensé à mettre à jour mon fichier primeweb.properties... ou pas. Que se passerait-il avec un simple EJB ? Que se passerait-il avec du code classique old-school qui serait capable de me dire poliment que 3 propriétés ne sont pas renseignées ? Là pour tout vous dire, j'ai pris une grande exception en pleine face. Je regarderai demain comment au moins mettre un peu de vérification afin de rendre plus polie mon application. Alors j'ai regardé le fichier application.xml... et il y en a du monde. Un peu comme un car de supporter de l'OM/du PSG/de l'OL de retour d'un match, c'était un sacré foutoir pour comprendre comment dans ce sac de nœud personne n'avait été blessé jusqu'à maintenant. Il y avait au début du fichier une organisation assez simple avec un DAO, deux Beans déclarés

normalement. Et plus je suis entré dans ce fichier de configuration, plus j'ai eu du mal à comprendre les relations entre Beans... Un peu comme les acteurs d'un sitcom, on ne sait plus très bien qui couche avec qui... Il faut alors configurer votre IDEA (IntelliJ supporte très bien Spring) et il est ensuite tout à fait possible de comprendre les déclarations. Ce n'est pas la faute de Spring, c'est notre faute à nous. On est généreux, nous avons mis un peu tout et n'importe quoi sous la forme de Bean comme un groupe de junkie très excité à l'idée de s'injecter des dépendances...

**L'inversion de contrôle est un concept franchement sympa tant qu'on garde le contrôle.**

*Inversion of Control is a nice concept if you keep some sort of control*, copyright nicolas martignole pour la version pour l'export... A vouloir trop retourner la responsabilité du code, est-ce que finalement nous n'avons pas mis dans un fichier XML un peu trop de choses ? Le prix à payer pour cette flexibilité très pratique lors des tests, est qu'une partie de la vérification qui était effectuée à la compilation, voir au démarrage de l'application, est maintenant pris en charge par Spring. Et ça, ça me dérange. Ce n'est pas la faute de Spring, c'est encore une fois notre faute à nous. Pour peu que quelqu'un propose un outil, nous sommes tentés de nous en servir et un peu comme un enfant avec un pistolet ou un bouledogue avec un bébé : aie cela fait mal. Les EJB3 ont un mérite : ils arrivent après la version actuelle de Spring 2.5. Et à ce titre, je commence vraiment à croire que c'est une option très sérieuse qui dans le cadre d'une application de gestion, est tout à fait convenable. J'ai hâte d'acheter et de lire [le livre d'Antonio Goncalves du Paris JUG](#) car je pense qu'il y aura du bon sens. Spring reste un très bon framework qui ne fait pas que de l'IOC (Inversion of Control) mais aussi beaucoup d'autres tâches barbant. La gestion de la partie JDBC, des templates JMS ou des Transactions : merci ! cela nous évite de devoir gérer une soupe technique peu intéressante. Cependant pour certains cas (que je devrai préciser si j'étais un gars bien) je continuerai à faire du BoViCoJa (Bon Vieux Code Java) au lieu d'utiliser Spring. Que l'on ne m'accuse pas de cracher dans la soupe, mais qu'en même temps les gens prennent conscience qu'il ne faut pas tout faire faire par Spring. Il sera parfois plus intéressant de faire du code classique et d'éviter de déclarer une Interface, puis une seule implémentation, pour ensuite l'injecter dans votre code. Je vous parlerai la prochaine fois de notre copain NoClassDefFoundError et d'AspectJ, le tout mijoté avec du Maven2 et des problèmes de dépendance.

## Compte-rendu de la présentation de Wicket et du Web Sémantique

Atterrissage en douceur hier soir sur la planète Wicket, avec aux commandes les 2 fondateurs de Zénika : Carl Azoury et Nicolas André, anciens de Sysdeo et maintenant aux commandes de Zénika. Le format choisi était ambitieux : réaliser en une heure une application CRUD de gestion de contacts. Quelques sueurs froides au départ, mais pour la suite franchement c'était vraiment bien piloté. De mémoire de Joueur, les derniers codeurs en direct étaient Jean-François Hélie et Christian Blavier d'OCTO Technologies en juin dernier, lors de [la soirée TDD](#) et Spring. Je trouve qu'ils ont vraiment bien assuré sur le contenu et sur la forme. Avant cela, Antonio rappelle l'importance des sponsors, le Paris JUG a besoin de l'aide et du support de sponsors afin de continuer à pouvoir accueillir tout le monde. Concernant l'offre Java BlackBelt évoquée le mois dernier, vous pouvez contacter l'équipe en vous abonnant à la liste de diffusion du Paris JUG [directement sur le site](#). Enfin chacun est venu avec son attestation d'inscription. Les places sont limitées pour des raisons de sécurité. Hier soir je pense que la salle était pleine avec ses 180 places. La présentation Wicket elle-même était sympa, je pense que mes voisins seront en mesure de couvrir en détail la soirée. Allez voir [Olivier](#) (de Zenika) je pense que son blog ne manquera pas de vous expliquer le contenu de la soirée. Carl a annoncé que le 29 avril prochain se tiendra une nouvelle soirée sur le thème de Wicket organisée comme [la présentation de Lean](#). Le présentateur sera l'un des co-auteur de Wicket, je crois qu'il s'agit de Jonathan Locke. A suivre donc. Encore une fois bravo à Zénika qui après Ross Mason, [Ari Zilka](#) et la présentation sur Lean, prépare un événement sympathique. Enfin pour terminer, Nicolas André et Carl Azoury ont montré qu'en 163 lignes de codes et 12 classes, il était possible de réaliser une application Web Ajaxisée avec le support du Drag-and-drop. Zenika propose un concours sur leur blog à ceux qui souhaitent coder la même application avec une autre technologie que Wicket. Le cahier des charges est assez précis, retrouvez l'article complet [sur le blog de Zenika](#). Après un buffer où j'ai croisé pas mal de monde, la deuxième partie de la soirée était une présentation du Web Sémantique par Alexandre Bertails d'Atos Origin Open Source Center. La présentation était très claire, alors que le sujet est assez compliqué à expliquer. Les notions de Triplet, d'URI, de Graphe ont peut-être fait peur à quelques-uns d'entre nous. Ce que j'ai retenu, c'est que l'objet du Web Sémantique est de séparer les Données des Applications qui les gèrent. Ainsi, à terme on imagine un moyen complet pour décrire de l'information. Cela ouvre la porte à la création d'applications capable de s'approprier l'information et d'y apporter de la valeur. Alexandre prend l'exemple de la construction d'une phrase : SUJET - VERBE - COMPLEMENT pour expliquer la notion de Triplet. Par exemple :

Le Touilleur est un blog

Chacun des éléments du Triplet sont donc le sujet, le verbe et le complément. Le tout est décrit avec 3 URI différentes pour chacun des tuples. Bien entendu différents formats comme RDF ou OWL permettent de formaliser en langage informatique ces prédicats. Je discute un moment avec [Sébastien Letélic](#), directeur technique d'Improve Santé. La formalisation du dossier médical avec le Web Sémantique permet de s'assurer que le format pour exprimer finalement nos données est ensuite utilisé au mieux. Il travaille je crois sur un projet avec le SAMU afin de proposer à terme un système médicalisé basé sur une ontologie médicale, où les médecins décrivent et mettent au point ces concepts. Alexandre présente un site que vous pouvez vous aussi tester si les concepts de Web Sémantique vous semblent un peu compliqué. Le site [dbpedia.org](#) est une extraction du contenu de Wikipedia, enregistré avec le format du web sémantique. Vous pouvez ainsi entrer des requêtes simples en utilisant par exemple [OpenLink iSPARQL](#). Un exemple de requête logique:

Barack Obama | est né | à un endroit

Un endroit | a | pour nom ?

Regardez par exemple [la page sur Java](#) de dbpedia avec les différents attributs (les tuplets). On constate que le contenu exprime avec des URI le type de contenu. La page web elle-même vous présente de l'information, mais démarque aussi le contenu avec des annotations afin de vous dire : "ici c'est le titre" "là c'est une image" "ici il s'agit d'un commentaire", ce qui permet alors à votre navigateur d'en déduire ce que vous voyez. Je sens que je rame. Imaginez le Touilleur Express : chaque article est balisé (de manière invisible pour vous). Lorsque je place une date dans un de mes articles, je précise qu'il s'agit d'un rendez-vous. Firefox qui affiche la page peut alors vous proposer par exemple de sauver ce rendez-vous dans votre agenda. C'est ce que l'on appelle les [microformats](#), sans doute l'avenir du contenu intelligent sur Internet. [Le site officiel](#) sur les microformats nous informe qu'un certain nombre de format, spécifié avec RFC, sont dans les cartons. Tout ceci est encore assez jeune, Alexandre explique que les derniers travaux datent de l'an passé. En conclusion, car j'ai du mal à vous retranscrire ici cette très bonne présentation, je pense que le Web sémantique est un concept qui sera mature d'ici quelques mois, voire quelques années. Avec la puissance des réseaux et ce besoin de libérer les données des applications, nous pourrons alors avoir une nouvelle forme de navigation sur Internet, un nouveau moyen d'échanger l'information. Le mois prochain la prochaine présentation du Paris JUG sera sur Scrum, par Eric Mignot et moi-même. Nous travaillons avec Eric sur un format et un contenu qui sera spécialement formaté pour le Paris JUG, avec pour moi l'envie de vous proposer une présentation aux petits oignons. Au mois prochain ! Références : [Blog de Sébastien Letélicé](#)

## Java BarCamp Paris 4 le 31 mars



<http://barcamp.org/JavaCampParis4> La 4ème édition du Java BarCamp se tiendra le mardi 31 mars prochain dans les locaux de Google. Le nombre de place est limitée à 70, l'inscription est donc obligatoire sur le site en créant un compte sur le wiki, puis en éditant la page afin d'ajouter votre nom directement sur le site. N'hésitez pas à contacter l'équipe si vous avez des difficultés à vous inscrire, ou même moi directement, je vous inscrirai sur le site. J'ai assisté aux deux premières soirées BarCamp, c'était l'occasion de croiser de nouvelles personnes par rapport au Paris JUG, et de ne pas parler QUE de Java. Car si vous reprenez le compte-rendu de [la première soirée](#) et de [la deuxième](#), nous avons parlé aussi de Scrum, de TDD, d'outils d'intégration, etc. A la dernière soirée j'ai bifurqué vers des sujets connus. Je me promets pour la prochaine séance d'aller à des sujets auxquels je ne connais rien. Cela dit, cher lecteur, si vous comptez venir voici un peu de travail pour toi afin d'amener de quoi passer une bonne soirée : - Prends un sujet technique, d'architecture ou de gestion de projet, que tu maîtrises à fond. Tu viens le jour J gonflé à bloc avec une belle présentation, de quoi nous montrer que tu es sur ce sujet et que tu adores. Exemple : *GWT, Wicket, Scrum, EasyMock, JBoss Portal, Ivy, Grails, Glassfish...* - Prends ensuite un sujet polémique pour ta deuxième idée : *GWT mieux que Wicket ? IDEA IntelliJ est mieux qu'Eclipse mais pourquoi ? Spring et Guice ne font-ils pas la même chose ? Groovy est-il moins rapide que Java ? Développer sur Google Android ou sur l'iPhone ? Pourquoi Python va remplacer Java d'ici 2 ans ? Pourquoi les vieux geeks Java bossent sur Mac et les jeunes sur Linux ?* - Prends enfin une phrase au hasard sur TheServerSide, InfoQ ou DZone afin de proposer un sujet hype et voir si quelqu'un d'autre connaît ce sujet. Exemple "[Bamboo 2.2 instant scaling using the Cloud](#)" pour faire de l'intégration continue ou encore "[Sortie de Nexus 1.3, qui connaît ?](#)". Il t'est interdit de prendre un article du Touilleur. Le jour J, tu peux ensuite te rendre sur place. Après ton arrivée, tu prends quelques post-it et un marqueur. Tu marques alors quelques mots clés comme "EasyMock !" ou "TDD !" ou encore "Grails ?". Une fois que tout le monde a mis sur post-it 3 ou 4 idées, nous ferons un regroupement des sujets similaires, afin de regrouper par salle les personnes ayant les mêmes interrogations pour qu'elles discutent ensemble. Il n'y a pas d'animateur, ni de présentateur. Il faut simplement que tu sois prêt à prendre la parole, soit pour poser pleins de questions, soit pour répondre aux questions des autres. Le tout dans une bonne ambiance, avec le sourire, et dans l'idée de s'amuser. L'inscription et les détails sont sur le site du Java Camp Paris : <http://barcamp.org/JavaCampParis4> Enfin pour terminer, voici une proposition

de logo que j'ai envoyé à l'équipe,  ou sinon aussi `[caption id="attachment_879" align="alignnone" width="300" caption="Logo Java BarCamp`

 Paris"] `[caption]` et un autre fait ce soir



<http://barcamp.org/JavaCampParis4> Lequel vous préférez ?

## CMMI : 608 pages pour changer

La traduction officielle en français de CMMI-DEV de quelques 608 pages est disponible sur le site du [SEI](#) (Software Engineering Institute). Bravo pour l'effort de traduction. Je n'ai pas résisté au plaisir de vous faire découvrir CMMI. Notez qu'à la fin de cet article vous serez convaincu que c'est un mal nécessaire pour structurer une entreprise, mais que ce n'est pas une solution à adopter sans recul et sans prendre en compte les dernières recherches en gestion du développement informatique. CMMI-DEV est un ensemble de pratiques destinées à améliorer les processus d'une entreprise dans le cadre du développement, de l'exploitation et du support d'un logiciel informatique. Basé sur une échelle de 1 à 5, 5 étant le niveau le plus avancé de l'utilisation des recommandations de CMMI (Capacity Model Maturity), c'est un système de qualité de plus en plus examiné par les responsables informatiques. Une entreprise certifiée CMMI 3 par exemple met en place un ensemble de structuration et de techniques afin de couvrir l'ensemble de la chaîne de production d'un logiciel. Cela va de la capture du besoin, de la définition de critères de qualité, aux méthodes de recettes, en passant par la production. CMMI n'est pas une méthode de développement de logiciel. Ce n'est pas non plus une méthode qui vous apporte des techniques de génie logiciel comme XP. C'est un cadre de pratiques en vue d'améliorer les processus métiers de votre service informatique, du développeur à la production en passant par l'équipe qualité. Si vous ne voyez pas la nuance, je vous propose de vous plonger dans [l'excellente FAQ de CMMI](#) en anglais. De plus, c'est un **modèle** vers lequel il faut tendre (d'où ces niveaux de 1 à 5) mais il faut bien entendu prendre en compte le contexte de votre métier. Les 5 niveaux du classement CMMI sont ([article de Wikipedia](#))

\* **Initial (niveau de maturité 1)** : Il n'y a pas de grand pilier directionnel, aucune façon de faire ou standard ne sont établis (ou bien ils sont documentés mais ne sont pas utilisés), tout doit être fait. Il n'y a pas de surveillance (monitoring), aucune évaluation de performance et la communication est absente. Les faiblesses ne sont pas identifiées et les employés ne sont pas au courant de leurs responsabilités de façon définie et absolue. Les réactions aux incidents se font en mode urgence, sans identification claire des priorités. À ce niveau les solutions ainsi que les projets sont décidés, développés et instaurés par un individu. Les compétences et les ressources propres de cet individu sont la raison du succès ou de l'échec du projet (par dérision, ce niveau est aussi nommé héroïque ou chaotique). Il n'y a pas de description du niveau de maturité 1 dans le modèle. \* **Discipliné (niveau de maturité 2)** : Une discipline est établie pour chaque projet et se matérialise essentiellement par des plans de projet (plan de développement, d'assurance qualité, de gestion de configuration ...). Le chef de projet a une forte responsabilité dans le niveau 2 : il doit définir, documenter, appliquer et maintenir à jour ses plans. D'un projet à l'autre, il capitalise et améliore ses pratiques de gestion de projet et d'ingénierie. \* **Ajusté (niveau de maturité 3)** : Ce niveau est caractérisé par une standardisation adéquate des pratiques, une capitalisation centralisée (en particulier sur les mesures réalisées dans les projets) et une maîtrise du référentiel interne (ou Système Qualité). Il existe des lignes directrices, un plan stratégique et une planification de l'amélioration de processus pour le futur, en ligne avec les objectifs d'affaire de l'organisation. Les employés sont formés et conscients de leurs responsabilités ainsi que de leurs devoirs. \* **Géré quantitativement (niveau de maturité 4)** : Les projets sont pilotés sur la base d'objectifs quantitatifs de qualité produit et processus. La capacité des activités (ou sous-processus) critiques est déterminée par l'organisation, ainsi que les modèles de performance et de prédiction associés. L'expression de la qualité demandée par le client est prise en compte pour quantifier les objectifs du projet et établir des plans selon la capacité des processus de l'organisation. \* **En optimisation (niveau de maturité 5)** : Les processus qui sont gérés quantitativement pour le pilotage de projet (niveau de maturité 4) sont en optimisation constante afin d'anticiper les évolutions prévues (besoins clients, nouvelles technologies...).

[Fin de la citation de l'article de Wikipedia] De ce que j'ai lu, CMMI n'est pas non plus une méthode de gestion de projet. C'est une démarche qui permet de faire monter en compétence une entreprise afin d'atteindre une maturité suffisante pour réaliser avec succès ses projets. Pour cela CMMI s'attache à proposer des moyens pour structurer tout un département. Il s'agit donc d'un road-book qui vous accompagne pour créer une équipe assurance qualité, une équipe support, une équipe de production ou une équipe de la gestion du risque. La structuration d'une entreprise ou d'un service avec des standards reconnus doit logiquement apporter une meilleure qualité ou réussite des projets. Bien que CMMI soit formalisé depuis presque 10 ans, il n'y a pas de mentions de cycle itératif ou de livraison par lot. Michel VOLLE qui a fait une lecture et un commentaire vraiment intéressant sur CMMI dit ainsi [sur son site](#):

On ne trouve pas dans CMMI la recommandation de découper la livraison en lots exploitables pour éviter l'« effet tunnel » et tirer parti, en cours de réalisation, des indications fournies par de premiers utilisateurs. CMMI ne met pas en garde non plus contre les difficultés logistiques que comporte l'organisation des réunions avec des parties prenantes qui ont d'autres priorités.

Au niveau de maturité 3, CMMI dans la partie perfectionnement de projet renforce l'idée que la spécification fonctionnelle, puis technique avant l'implémentation apporte une meilleure qualité. Quelle perte d'énergie. Qui croit encore qu'un document formalisé, qui plus est non contractuel, est une garantie de réussite ? Depuis quand un document Word dans un processus de développement **informatique** serait le Saint-Graal de la réussite des projets ?

[...]CMMI dit que les spécifications doivent être validées, mais il n'indique pas qui doit faire cette validation. S'il s'agit de s'assurer que les spécifications répondent bien

aux besoins des utilisateurs la responsabilité de leur validation appartiendra à la maîtrise d'ouvrage (à l'exception de la solution technique, qui relève de la maîtrise d'œuvre). L'expérience montre combien il est difficile de présenter les spécifications de telle sorte que l'on puisse obtenir, de la part de la maîtrise d'ouvrage, une validation authentique (c'est-à-dire une validation qui engage véritablement son expertise professionnelle et sa responsabilité). On ne voit pas apparaître dans CMMI les étapes de modélisation (MCD ou UML) pourtant nécessaire pour assurer la qualité sémantique des données et des référentiels ; les revues de pairs (peer reviews) sont, étrangement, mentionnées après la livraison du produit alors qu'elles sont surtout utiles lors de la conception de la solution.

(extrait du site de M.VOLLE) **Scrum et CMMI ?** *Scrum est un outil léger et simple qui permet de maximiser le canal de communication entre le responsable du produit et l'équipe de développement* (Eric Mignot). C'est donc une méthode de gestion de projet adaptée pour livrer par itération une quantité de travail (l'incrément) prêt à fonctionner et parfaitement fonctionnel. On reproche à Scrum de ne pas avoir de pratiques logicielles. En effet ce n'est pas une méthode de développement. Sur le terrain, les équipes de développement utilisent souvent l'eXtreme Programming (XP) pour renforcer la qualité et la communication **au sein de l'équipe**. Scrum est un outil qui permet de s'assurer que l'équipe développe et livre régulièrement les fonctions les plus importantes d'un projet. Cela permet au responsable produit d'intégrer régulièrement les demandes de changement. C'est donc une méthode de communication **entre l'équipe et le client**. Je placerais CMMI sur un plan plus large, au niveau de l'organisation d'une entreprise, voire d'un service. Les pratiques de CMMI sont des repères pour structurer dans le temps une organisation. Sachez qu'il faut entre 20 et 30 mois pour passer d'un niveau CMMI 2 à un niveau 3 par exemple ) (référence: [paragraphe Les cinq niveaux de CMMI du document de M.VOLLE](#)). Si quelqu'un vous dit un jour que "vous devriez faire du ci-aime-aime-aie au lieu de Scrum" je vous autorise à rigoler très fort et à lui faire suivre ce lien. On se marra à plusieurs. *Références:* [Article de M.Volle](#) analyse de CMMI [La FAO en anglais à propos de CMMI](#) <http://blog.xebia.fr/2007/09/17/revue-de-presse-xebia-23/#AMagicPotionforCodeWarriorsCMMI@orAgile:WhyNotEmbraceBoth!> une approche qui montre que l'Agile et CMMI sont amenés à travailler ensemble (10 pages)

## iPhone : les nouveautés pour juin 2009



C'est ce soir qu'Apple a dévoilé les détails sur la 3ème version du système d'exploitation de l'iphone. La nouvelle mise à jour sera disponible en juin prochain, gratuitement pour les iPhones. Le kit de développement propose enfin aux développeurs de nouvelles possibilités pour le multimédia, le streaming vidéo et audio, ce qui devrait nous apporter un nouveau lot d'applications très rapidement. Le support Bluetooth de l'iphone sera exploité afin de pouvoir s'échanger entre iPhone des fichiers, ou pour pouvoir jouer en réseau. Les échanges entre le serveur et l'iphone permettront aussi de pousser des données vers l'iphone. L'idée de faire fonctionner une application en tâche de fond a été abandonnée. Cela grève beaucoup trop l'autonomie de l'appareil. Un vrai support du mode **copier-coller** très sympa arrive sur l'iphone. Sur la vidéo on peut voir Scott double-taper rapidement sur du texte pour passer en mode copier-coller. Il sélectionne la zone et valide en tapant 2 fois. Si vous voulez annuler la sélection, il suffit de secouer le téléphone... j'adore. Ensuite la présentation nous montre le support du mode paysage sur le client SMS et le client email. Cela facilitera la saisie rapide de texte. Pas de possibilité de changer de clavier (anglais -> français) mais là c'est mon côté geek. L'ajout le plus intéressant est un vrai support des MMS dans le module de SMS qui se retrouve maintenant nommé "Messages". Il est possible d'envoyer des images et aussi du son enregistré avec l'iphone. Marrant... mais toujours pas de vidéo. Enfin la suite de la démonstration montre l'ajout d'une fonction de recherche, bien connue des utilisateurs de Mac OS X. Pas mal mais je ne suis pas franchement convaincu. A tester. Apple a annoncé que 17 millions d'iphone ont été vendus à ce jour ! Rendez-vous en juin pour mettre à jour votre "précieux"... Voir enfin le site d'Apple : <http://www.apple.com/iphone/preview-iphone-os/>

## Première soirée du French Scrum User Group

J'ai participé ce soir à la première soirée du French Scrum User Group organisée à Paris. Avec 282 membres inscrits, 130 personnes présentes ce soir malgré les perturbations dans les transports, et surtout la présence de Jeff Sutherland, le contrat est rempli. Comme vu avec Claude Aubry, je mets sous presse dès ce soir un billet sur la soirée afin que, vous lecteur, puissiez lire dès demain vendredi ce petit compte-rendu. Alors allons-y ! Luc Legardeur, le fondateur du French Scrum User Group, prend la parole afin de présenter dans un premier temps les objectifs du French SUG. C'est un groupe d'utilisateurs de Scrum ouvert à tous et gratuit, que vous pouvez rejoindre si vous vous intéressez à Scrum. Son rôle est de promouvoir la méthode Scrum en France, d'être le média de communication sur Scrum et l'Agilité. Luc rappelle qu'il s'agit d'une association loi 1901, et ce afin de pouvoir collecter des fonds de la part des sponsors afin de couvrir les frais de fonctionnement. Les sponsors de la soirée ce soir sont Borland, avec la présence de David Wilby, ainsi que les sponsors du SUG : [Xebia](#), [Zenika](#), [Microsoft](#) et [Valtech](#). Si vous souhaitez vous inscrire et participer au prochain événement, vous pouvez vous inscrire sur [le site MeetUp du French SUG](#). Jeff Sutherland prend ensuite la parole, afin de nous donner tout d'abord un aperçu de l'adoption de Scrum dans le monde et en France. Il existe à ce jour 18 Scrum user group aux USA, 18 en Europe, 8 dans la partie Asie, Afrique, Océanie et Moyen-Orient. Le groupe français est

semble-t-il le plus important en nombre de personnes inscrites. La présentation de Jeff est un moment très sympathique, il parle de cas d'utilisation de Scrum sur des projets classiques mais aussi dans des aventures moins classiques comme la conduite d'une Eglise avec Scrum par la Révérente Sutherland, son épouse présente dans la salle... Il parle ensuite du dernier événement qui s'est déroulé en Floride : [The Orlando Scrum Gathering 2009](#). Il s'agit d'un événement qui s'est déroulé début mars visant à réunir la crème de la communauté Agile et Scrum afin de discuter pendant 2 jours sur Scrum. J'ai noté un détail très important : la présence de Gregory Balestrero, le President & CEO de [PMI](#) (Project Management Institute). C'est une association à but non lucratif qui propose depuis 1969 une méthode de gestion de projet certifiante. C'est l'une des méthodes les plus connues, surtout de l'autre côté de l'Atlantique. Jeff Sutherland explique que la communauté Scrum travaille main dans la main avec la communauté PMI afin de réfléchir à l'utilisation de principes Agiles dans la communauté PMI. Plus tard dans la soirée, Eric Mignot explique aussi qu'il va aller à la rencontre de la communauté [PMI](#) (wikipedia) afin de travailler et voir comment Scrum et PMI peuvent fonctionner ensemble. Jeff dans la deuxième partie de sa présentation couvre aussi l'effet Scrum. Que se passe-t-il une fois que Scrum a été mis en place ? Il prend alors pour exemple le film Matrix, où Neo a le choix entre la pillule bleu et la pillule rouge. Jeff, pour la petite histoire tu m'as piqué l'idée. J'ai écrit [le 24 mars 2007](#) un article avec la pillule bleu et la pillule rouge ! La pillule bleu permet de se réveiller après avoir rêvé. Vous mettez Scrum en place, cela commence à fonctionner. Votre manager vous demande cependant de ne pas utiliser tout Scrum, de faire des Sprints à durée variable... Vous venez de prendre la pillule bleu et très bientôt votre projet va redevenir ce qu'il était avant Scrum. De l'autre côté vous avez la pillule rouge. Celle-ci si vous la prenez aura un effet hallucinogène sur votre projet. Pour ma part je disais que Scrum avait l'effet d'un pétard : il soulève brutalement votre client pour construire rapidement un produit, tout en apportant aussi un effet euphorisant à votre équipe. Le double effet pétard. Le message que Jeff Sutherland veut faire passer c'est que Scrum est simple, facile à comprendre. Sa mise en place par contre demande beaucoup de travail, de la volonté et de l'engagement. L'effet et le résultat de la gestion d'un projet avec Scrum sera visible très rapidement. Donnez un peu de responsabilité à l'équipe, organisez une liste triée par priorité de développement à effectuer. Ne changez pas le planning toutes les semaines. Acceptez que ce soit l'équipe qui vous donne une estimation pour réaliser une nouvelle fonction. Exigez de tester un produit terminé et fonctionnel. Revenez tout le temps observer si l'équipe va bien et si elle avance sans problèmes. Souriez, vous ne rêvez pas, c'est simple et c'est Scrum. La suite de la présentation de Diesler de Borland est un témoignage sur la mise en place de Scrum aux seins de 3 équipes de Borland. J'ai retenu qu'il faut pas faire des slides que les gens lisent, que Microsoft Visio 2007 ou KeyNote sur Mac auraient donné un résultat plus joli... etc. Le plus intéressant est que Scrum est un chemin vers l'amélioration. La route est droite, l'objectif est limpide... mais ça monte ! Ensuite nous avons eu la chance d'avoir la présence de [Claude Aubry](#). Claude fait parti comme moi du bureau du French SUG. C'est l'auteur du blog "[Scrum, Agilité et Rock'n roll](#)". Après une carrière dans l'industrie, le développement et le monde de l'aéronautique, il a créé sa structure de conseil sur Toulouse. Il est aussi professeur à l'Université Paul Sabatier de Toulouse. C'est l'une des grandes références de Scrum en France, et donc sa présence était aussi très importante. Assez de cirage de chaussures, passons à sa présentation. Exercice de ce soir : parler de la communauté Scrum en France. Comme il l'explique pour commencer, il a découvert Scrum en 1996 et s'y est progressivement intéressé jusqu'à créer une structure pour en faire son métier. A ce jour on recense entre 400 000 et 500 000 informaticiens en France. C'est amusant mais j'ai fait un exercice d'estimation. Je suis allé sur le site du Syntec et j'ai vu que 60% de ces informaticiens font du développement et de la gestion de projet. Les 40% restant font de la maintenance, de l'admin système, de l'exploitation, etc. Cela donne donc une fourchette haute dans les 275 000 informaticiens en France susceptible d'entre parler de Scrum un jour. Ensuite tout en bas de l'échelle, il y a eu 540 personnes certifiées Scrum Master en France depuis l'an dernier, dont 120 par Jeff Sutherland avec la formation de [Xebia](#). Imaginons que sur ces 500 personnes, 250 diffusent activement et 100 sont vraiment Scrum Master, d'équipe de 10 personnes... on arrive au chiffre de 1000 personnes ayant soit fait du Scrum, soit participé à Scrum. Dans la suite de la présentation, Claude Aubry rappelle un peu le fonctionnement de Scrum. Il cherche aussi à identifier très justement la particularité de la culture française. Je pense en effet qu'il a tout à fait raison de parler de l'écosystème où nous travaillons, de la structuration des projets, des missions "au forfait" qui sont un héritage lourd parfois. Il explique que les développements informatiques sont structurés comme le développement dans le bâtiment. Le vocabulaire MOA/MOE et la séparation entre les deux vient directement de nos voisins les bâtisseurs. Dans la culture française, nous avons une forte curiosité. Nous adorons remettre en question ce que l'on nous apprend, tout en conservant un aspect cartésien, Descartes est français après tout. Pour autant, Claude explique que Scrum fonctionne très bien et qu'il n'y a pas de changements de règles à utiliser. Il faut simplement s'adapter à son interlocuteur. J'ai apprécié sa présentation, on retrouve les sujets de fonds qu'il traite sur son blog. Très intéressant. Guillaume Bodet, directeur technique de Xebia France, prend enfin la parole pour la dernière partie de la soirée. Son objectif est de nous proposer des techniques pour vendre Scrum à nos managers, à nos clients et à nos équipes. Tout d'abord une distinction sur la notion de valeur et la notion de coût. Scrum est vendu d'abord pour la valeur qu'il crée : cycle itératif rapide, meilleure communication, meilleure qualité, moins d'échec de livraison, produit mieux testé et terminé, etc. On oublie aussi le coût : remise en question des rôles, gestion de la communication entre les personnes, conflit d'intérêt entre interlocuteurs, peur du changement, peur de perdre le contrôle en tant que manager. Il résume très justement le cycle d'adoption en effectuant une analogie avec le cycle d'une psycho-sociologue qui soigne des familles en souffrance. L'introduction d'une thérapie de groupe fait d'abord flancher et exploser la famille avant de la rendre plus forte. Scrum va tout d'abord déstabiliser une équipe et apporter du chaos. Pendant la phase où l'équipe pense que c'est pire qu'avant et que Scrum ne fonctionne pas, il faut s'accrocher. C'est ensuite au bout de quelques temps que l'équipe dépasse son seuil de production, et qu'elle commence à fonctionner à 400%. Comme toutes méthodes, il y a donc d'abord une phase de résistance de la part de l'équipe. Vient ensuite la résilience, la résonance et enfin pour terminer la renaissance. Guillaume Bodet rappelle aussi l'importance d'adapter notre discours à la cible. Un DSI n'a pas les mêmes préoccupations qu'une équipe de développement. Il faut donc prendre le temps d'écouter et de rassurer. Il nous montre ensuite différents Kits afin de répondre aux sceptiques, aux inquiets, à ceux qui disent

"vous savez des méthodes j'en ai vu avant"... C'était un très bon moment. J'oublie (faute d'avoir pris des notes cette fois-ci) une bonne partie de la présentation de Guillaume. Tout le monde a bien aimé sa présentation, à la fois riche et intéressante. En principe Guillaume devrait faire une présentation autour de Scrum le mois prochain avec nous au Paris Java User Group. C'est le mardi 14 avril, les inscriptions se feront sur [le site du Paris JUG](#) à partir du 5 avril je pense. Pour terminer un buffet royal offert par Borland m'a permis de faire un tour, de discuter avec beaucoup de nouvelles personnes, dont quelques unes ont déjà lu le blog. Juste le temps d'écouter Eric Mignot discuter avec Antonio Goncalves sur CMMI et Scrum qu'il était déjà temps de partir... Antonio termine l'écriture en anglais d'un livre de référence [sur Java EE6](#). Plus de 440 pages, ce sera l'un des livres phares sur Java EE 6. Sortie prévue pour Java One. Comme on dit dans ce cas là : chapeau ! A quand la prochaine soirée scrum ? Est-ce qu'un Café-Livre tente des gens en avril par exemple ? Une discussion sur un thème précis de Scrum ? Sur ce, Ctrl-Z je vais dormir. Je relirai/corrigerai demain.

## Jazon 2009 : GlassFish community events

La conférence Jazon aura lieu du 22 au 25 juin 2009 à Zurich. Pour la communauté open-source Java, c'est un événement à ne pas manquer. Les conférences débutent le 23 juin, mais dès la veille deux événements sont déjà annoncés : le 22 juin se déroulera toute la journée un événement sur Java EE 6 et Glassfish v3 avec la présence de Roberto Chinnici (un des spec leads de JEE 6) et de Jérôme Dochez (Architecte sur GlassFish). Peut-être aussi la présence d'Alexis Moussine-Pouchkine et d'Antonio Goncalves ? Durant cette journée, les speakers parleront aussi de GlassFish v2, des déploiements en cluster et le fonctionnement par module de GlassFish v3.

## Les Universités du SI vues par un geek

Les Universités du SI est un événement organisé par OCTO Technology en juillet prochain. L'idée est de réunir les Boss et les Geeks afin de communiquer et découvrir sur les 4 thèmes suivants : - Gouvernance - Technologies - Méthodologies - Usabilité

### Technologies

[Julien Dubois](#), [Didier Girard](#), [Guillaume Laforge](#) ou [Sami Jaber](#) cette année. [Antonio Goncalves](#) l'an passé. Vous reconnaissez ces noms si vous lisez souvent le Touilleur Express. C'est donc avec un oeil curieux que j'ai regardé le programme de l'an passé afin de voir le contenu des présentations. J'ai décidé d'y participer cette année, en tant que blogueur afin de couvrir quelques conférences et vous donner une vision "Geek". Je pense qu'il est intéressant pour nous de nous rendre compte des enjeux en ce moment dans le monde de l'informatique. 2009 sera une année riche même si votre DSI est pauvre. Au programme des Universités du SI cette année, Didier Girard parlera [des technologies de Google pour le monde de l'entreprise](#). Sami Jaber présente [une solution de portail d'entreprise virtuel basée sur GWT](#) (et je parlais de Liferay ou d'eXo Platform quelques lignes plus haut...). Guillaume fera une présentation sur [une mise en oeuvre de langage dynamique structuré \(DSL\)](#) avec Groovy. Guy Barette et Mario Cardinal de Microsoft nous parlerons du Cloud-Computing. Mario parlera des enjeux dans le domaine de l'architecture. Si vous souhaitez que votre application monte en charge, il est important de suivre des bonnes pratiques d'Architecture. Il couvrira aussi les architectures en couche et comment remplacer une couche dans un système d'information. J'ai vu tout cela sur [une petite vidéo fun](#) où nous les voyons avec des teeshirts de l'équipe de Hockey du Canada. Maintenant si nous allions plutôt regarder les autres conférences pour changer ?

### Gouvernance

Sur le thème de la Gouvernance, une conférence que j'irai voir avec beaucoup de curiosité traite de l'Agilité : "[Peut-on généraliser les bénéfices de l'Agile à tout le DSI ?](#)" d'Ismaël Héry d'OCTO Technology. J'irai écouter aussi le fondateur de Mozilla Europe dont je lis souvent l'excellent [blog](#), monsieur [Tristian Nitot](#) himself. Sa présentation est limpide : comment 10 000 développeurs arrivent à faire un seul produit : Firefox ? [La présentation de Pierre Piezzardi](#), directeur technique d'OCTO, parlera de la mise en place d'une informatique conviviale, basée sur l'amélioration et les valeurs Lean dans une entreprise virtuelle. J'attends de sa présentation une vision des prochaines années et de la position du SI dans l'entreprise. Je pense personnellement que nous terminons la phase "Informatique" avec barbus des années 90, la phase "Open-Source Everywhere" des années 2000 et que l'étape suivante sera plus éthique, plus écologique (green computing) et plus dure qu'elle n'a été. Le rachat de SUN par IBM par exemple est un signal fort que les choses sont entrain de basculer doucement. Je pense que les éditeurs de logiciels classiques qui n'ont pas encore vu que le monde change ne pourront plus continuer très longtemps à vendre cher des logiciels qui ne valent rien... A suivre donc.

### Méthodologies

Sur le thème de [la Méthodologie](#), on retrouve une présentation qui s'axera je pense sur la création de valeurs, la communication. Alain Buzzacaro présente [la boîte à outils du Manager IT](#). [Emmanuel Gaillot](#), organisateur des Dojos et de la conférence Agile Open France proposera de faire une séance de brustring. Personne ne connaît ce mot car je viens de l'inventer. Voici en quoi cela consiste : vous mettez des gens dans une pièce, vous secouez la pièce comme une boîte de Puzzle, ensuite vous regardez ce qu'il se passe. J'espère pouvoir y assister. [Eric Pantera](#) d'OCTO Technology présente un sujet sur (à mon avis) la nécessaire transformation et remise en question de la Maîtrise d'Ouvrage dans le cadre de projet piloté avec une méthode Agile (XP, Scrum, les deux mon capitaine).

## Usabilité

Enfin le dernier thème de l'USI traite de l'Usabilité et de l'ergonomie. J'ai marqué en rouge sur mon agenda [la conférence de Johan Adda sur l'iphone](#). Le logiciel de l'iphone est révolutionnaire dans sa simplicité et son ergonomie. Le sens du détail, le travail de l'interface et la simplicité sont les clés du succès de l'iphone. A voir absolument ! Deuxième personne à ne pas louper : [Frédéric Cavazza](#) dont [les messages sur Twitter](#) sont l'une de mes sources d'information sur le web et les business à découvrir. Si vous êtes branché Web 2.0 et que vous cherchez une idée pour votre future entreprise, Frédéric présentera [l'impact du Web 2.0 sur les standards de conception](#). Pour les geeks et pour les boss.

## Combien la conférence coûte-t-elle vous rapporte en terme d'informations ?

Parlons un peu maintenant du sujet délicat par les temps qui courent : le coût de ces 2 jours. 1500 EUR si vous vous enregistrez avant le 31 mars. Ne partez pas j'ai une bonne nouvelle pour vous. Non les lecteurs du Touilleur Express n'ont pas droit à une réduction mais il y a mieux. La bonne nouvelle cette année c'est que les 2 jours de conférences peuvent être pris en charge par votre organisme de formation. Je vous explique car c'est très important. L'an passé j'ai assisté à [la journée Valtech RIA](#) sur Flex, GWT et Silverlight. Je suis allé voir tout simplement la personne responsable de la formation chez Reuters, et le coût (790 EUR je crois) de cette formation a été payé par mon DIF (droit individuel à la formation). Si le contenu des 2 jours vous intéresse, mais que vous savez qu'en ce moment niveau budget c'est un peu tendu, posez 2 RTT, demandez à financer les 2 jours avec votre organisme de formation, ainsi tout le monde sera content. Est-ce qu'il s'agit d'une formation ? Oui complètement. Je rejoins sur ce point la vision de Didier Girard, le responsable technique de SFEIR. En 2008, 15 de ses consultants ont assisté au salon Devovx sur Java en Belgique. Nous en parlions ensemble, et il est certain que ce type d'événement est un moyen d'apporter aux consultants de l'information, de les motiver et de les intéresser afin qu'en retour ils soient encore plus performants avec vos clients (là je parle au monsieur technique de la SSII qui me lit). Donc voilà, j'ai envie d'y aller, avec un oeil de Geek pour la partie technique et un oeil de Boss pour la partie entreprise. J'y serai le 1er et 2 juillet, après Jazoon en Suisse... Encore pleins d'articles sur le Touilleur Express en perspective. Ah le site pour terminer : <http://www.universite-du-si.com/>

## The Agile Process Maturity Model APMM

*Dimitri Baeli [d'eXo Platform](#) m'a fait suivre un article publié sur le site IBM DeveloperWork à propos d'APMM, The Agile Process Maturity Model. Dans cet article, Scott Ambler propose une échelle de classement des processus Agile afin d'aider à identifier selon votre organisation la solution la plus adaptée. Son point de vue est "one does not fit all", une solution ne peut convenir à tout le monde, à l'ensemble des problèmes d'une organisation. L'article original en anglais est disponible [à cette adresse](#), la traduction reprend le contenu en essayant d'être le plus fidèle possible. Je vous invite à lire la version en anglais pour suivre le sens précis des propos de Scott Ambler. [Début de la traduction de [l'article original de Scott Ambler](#)]*

### L'Agile Process Maturity Model (APMM)

L'objectif de l'Agile Process Maturity Model (APMM) est de fournir un cadre de mesure pour la pléthore de méthodologies Agiles présentes aujourd'hui. L'APMM définit trois niveaux, dont chacun s'appuie sur les autres, pour des processus et méthodes Agiles. Chez IBM, nous avons constaté que de nombreux clients trouvent que le message Agile est confus, en partie en raison de la multitude de voix au sein de la communauté agile mais plus encore parce que la rhétorique agile semble souvent ignorer ou masquer de nombreuses questions importantes de nos clients qui font face au jour le jour à ces méthodes. De là vient le besoin de quelque chose comme APMM afin de fournir un cadre [nécessaire] pour les méthodes et pratiques agiles

### APMM Niveau 1: Agile Software Development

Le niveau 1 du processus de maturité Agile adresse une partie du système de cycle de développement ([SDLC](#)). Quelques exemples de processus agiles de niveau 1 : - **Scrum**. L'objectif de Scrum est la gestion du développement au sens projet d'un produit et un outil pour la gestion des exigences. Les promoteurs de Scrum affirment que c'est un processus cadré, mais si c'est le cas alors il est éparpillé (*sparse en anglais*) et, au mieux, incomplet. Une description plus précise [*de Scrum*] est que c'est un cycle de haut niveau pour la construction d'itérations (ce que Scrum appelle "sprints") et de plusieurs pratiques telles que la réunion quotidienne de l'équipe (stand-up "Scrum"), la mise en place d'un responsable de produit (Product Owner), d'un carnet de produits (Product Backlog), de réunions de planifications, et l'objectif de livrer à chaque fin de sprint un produit potentiellement prêt à fonctionner. - **Extreme Programming (XP)**. XP est une collection de pratiques logicielles, comme le refactoring, les tests avant l'écriture du code, la programmation en binôme, la présence sur site du client, la mise en place de l'intégration continue, la responsabilité collective de l'équipe et la mise en place d'itérations. - **Agile Modeling (AM)**. AM est un ensemble de pratiques pour faire de la modélisation légère et de la documentation, y compris la gestion des exigences, de spécifications exécutables, participation active des parties prenantes, gestion par ordre de priorité des exigences, et preuve du fonctionnement avec le code (*NDLR comme le canada dry, c'est du scrum sans l'alcool...*). - **Agile Data (AD)**. AD est un ensemble de pratiques Agile pour le développement de base de données. Cela comprend par exemple une modélisation Agile des données, mettre en place une base de test, de refactoring et un système d'intégration continue pour la mise à jour de la base.

## APMM Niveau 2: Disciplined Agile Software Delivery

Le niveau 2 du modèle de maturité étend le niveau 1 afin de se rapprocher du système de cycle de vie complet (SDLC). Comme [les critères de développement d'une équipe Agile et Disciplinée le suggèrent](#), le niveau 2 tend à pousser encore plus certaines pratiques Agile comme les tests, la mesure de performance et la mise en place d'indicateur de qualité afin d'assurer un audit continu du produit. (NDLR : ce que l'on retrouve dans CMMI par exemple). Des exemples de processus de niveau 2 : - **Rational Unified Process (RUP)**. RUP est une méthode de prise en charge du cycle de vie du développement d'un logiciel qui peut être mise en place et adaptée à votre cadre, que ce soit de manière très agile ou sous forme très traditionnelle. Quelques pratiques de RUP incluent par exemple l'évaluation du risque et de la valeur, le développement piloté par les tests (TDD), de la modélisation des processus métiers et l'intégration continue. - **Open Unified Process (OpenUP)** OpenUP, disponible en open source, regroupe et étend les pratiques de Scrum, XP, AM et RUP pour les équipes de plusieurs départements qui construisent des applications d'entreprise. Quelques pratiques d'OpenUP sont la réunion de l'équipe tous les jours, le développement par ordre de priorité, la gestion et l'évaluation dans le cycle de vie du risque, les TDD, la participation active des parties prenantes(stakeholder) et l'intégration continue. - **Dynamic System Development Method (DSDM)**. DSDM est un processus Agile de livraison initialement fondé sur Rapid Application Development (RAD), qui est souvent utilisé pour le développement intensif des interfaces utilisateurs d'une application. Quelques pratiques de DSDM comprennent le prototypage, le test dans l'ensemble du cycle de vie, la mise en place de changements réversibles, et l'étude de faisabilité.

## APMM Niveau 3: Agility à l'échelle

Au début de l'Agilité, les applications où le développement Agile était appliqué étaient plus petites et couvraient des objectifs simples et clairs. Aujourd'hui la situation a beaucoup changé et les entreprises souhaitent appliquer les méthodes Agile à un spectre plus large d'applications. Les méthodes Agile ont donc besoin de s'adapter afin de répondre à différents besoins métiers, différents types d'organisation, et différents types de problèmes techniques que les entreprises rencontrent aujourd'hui. C'est l'objectif de ce niveau 3 de cette échelle de maturité : adresser explicitement les complexités que les équipes disciplinées et agiles rencontrent dans le monde réel. Les facteurs de difficultés qu'une équipe Agile peut donc rencontrer à un niveau plus ou moins important sont par exemple : la taille de l'équipe, la répartition physique, la distribution décidée par l'entreprise, la conformité réglementaire à respecter, l'organisation et l'enracinement culturel inadapté, la complexité des SI existants, la discipline de l'entreprise (tels que respecter les choix d'Architecture NDLR ou la stratégie de réutilisation de licences de logiciels payés les yeux de la tête...). Chacun de ces facteurs apporte une échelle de complexités et de difficultés pour une équipe Agile, et donc chaque équipe rencontrant une combinaison différente, aura donc besoin d'un processus différent adapté à son cadre et à son entreprise afin de s'adapter précisément à son environnement. Malheureusement le terme de "maturité" est chargé d'un sens assez lourd dans le domaine des processus logiciels, et non des moindres, en raison du processus [CMMI](#) du Software Engineering Institute (SEI). Beaucoup de bon travail a été fait pour montrer que CMMI et les méthodes Agile peuvent être mis en oeuvre en même temps, et je me réjouis de voir que cette stratégie se concrétise. Cependant, l'objectif de CMMI est de fournir un cadre pour l'amélioration des processus métiers ; l'APMM est beaucoup plus modeste : - il ne cherche à définir un cadre qui peut être utilisé pour mettre un contexte ordonné aux myriades de processus Agile. A l'avenir sur ce blog je vous parlerai de comment CMMI et APMM peuvent fonctionner ensemble. " **[Fin de la traduction]**

## Analyse

L'article part du constat qu'il y a de plus en plus de mise en place de méthodes Agile au sein des entreprises, et que cette mise en place est difficile. La difficulté vient du cadre, du milieu et de l'existant. La proposition de ce modèle est donc de classer par niveau de maturité les différentes méthodes Agiles. Le but semble-t-il est d'expliquer que chacune de ces méthodes couvre tout ou partie des besoins. Scrum et XP n'adressent qu'une partie des besoins, là où RUP propose d'adresser un spectre plus large de problèmes, et donc de besoins. Après avoir lu cet article, je trouve que le classement est un peu faible. Cité dans l'article sur [le Processus Unifié](#) de Wikipedia, [Scott Ambler](#) est l'auteur ou le co-auteur de 19 ouvrages. C'est un partisan de RUP et des différents processus unifiés apparus ces dernières années. Je vois ces 3 niveaux comme un empilement successif de soins pour un même patient souffrant de différentes pathologies dont la gravité n'est peut-être pas forcément toujours bien évaluée. Entre un ongle cassé, une bonne bronchite ou une jambe cassée, il convient d'adapter le traitement. Un coach Agile dont la mission est d'aider une équipe, doit en premier lieu observer avant d'agir. Il y a certainement une partie dogmatique, que ce soit chez XP ou chez Scrum, où chacun des coaches Agile de ces 2 camps est certain de détenir le bon remède pour le patient. Je ne pense pas qu'un processus comme RUP réponde aux besoins récurrents et identifiés dans les entreprises de favoriser le canal de communication entre l'équipe de développement et les responsables produits. C'est le rôle de Scrum. De même je ne pense pas qu'une solution comme EUP proposé par Scott réponde au besoin de qualité et de transparence qu'XP apporte à une équipe. Il est donc important de prendre le temps de comprendre le fonctionnement de l'existant dans une équipe, un département et une entreprise avant de tenter d'insérer une méthode Agile, quelque soit son nom. Pour cette raison, toutes les méthodes Agile ne sont donc pas forcément adaptées. Je pense cependant que plus une méthode Agile tente de proposer des processus et une structuration, moins elle est Agile. RUP est une méthode Agile au sens itératif, incrémental, guidée par les besoins des utilisateurs et centrée sur l'architecture logicielle. Scrum est une méthode Agile qui est aussi itérative, incrémental et guidée par les besoins de l'utilisateur. Et le mot "Architecture" a été prononcé jeudi dernier aussi par Jeff Sutherland. Pourtant elle n'a rien à voir avec RUP ou XP... car elle ne répond pas au même besoin. Si vous voyez une offre d'emploi pour un Coach XP ou Scrum, imaginez une petite annonce d'un

malade qui recherche un gastroentérologue pour traiter la chute des cheveux... Le patient s'est auto-diagnostiqué et il cherche maintenant son médecin ! C'est fort non ? Il faut donc parler "Coach Agile" et ensuite rencontrer plusieurs personnes. Celle qui ne vous dit pas de quoi vous souffrez (si vous êtes en souffrance) n'est pas la bonne personne. Enfin pour terminer sur l'analogie avec la médecine, en Chine les bons médecins ne soignent jamais car leurs patients ne sont jamais malade. Pour cela, la thérapie chinoise s'applique à traiter de manière préventive plutôt que curative. Appliqué à notre monde d'informaticien, cela revient à mettre en place quand tout va bien des principes Agile comme l'intégration continue, la gestion des demandes du client par priorité, la modélisation et l'architecture agile, les tests unitaires, etc. Lorsque tout va mal il est plus difficile de mettre en place un processus large comme RUP, il est plus facile par contre de travailler pour mettre quelques pratiques XP dans une équipe. Il faut à mon avis d'abord travailler avec des outils et des méthodes simples avant d'envisager de mettre en place des processus plus complexes, même s'ils sont Agile. Plus le médicament est fort, plus les effets secondaires sont intenses.

## Références

Voici quelques articles pour aller plus loin : - <http://www.agilemodeling.com> : la page de l'Agile Modeling dont l'auteur est aussi Scott Ambler, est une présentation d'une méthode Agile pour modéliser et documenter un logiciel. Les principes d'itération, de livrer par incrément, de privilégier la communication, bref une méthode Agile. - <http://www.enterpriseunifiedprocess.com/> Scott (toujours lui) s'est ensuite attaqué à définir une méthode de gestion du cycle de vie de l'application dans le cycle IT. Là où RUP est axé uniquement sur le logiciel, EUP est un travail qui tente de définir le cycle de vie d'un logiciel. A cet instant précis j'essaie de comprendre l'intérêt de faire des courbes en couleur pour expliquer au lecteur que la conception d'un logiciel comprend une phase d'analyse, une phase de réalisation et une phase d'obsolescence... - [Présentation du Processus Unifié et de RUP](#) sur Wikipedia - [Scaling Scrum ?](#) par Ivar Jacobson. Son point de vue est que Scrum n'est pas adapté pour de larges architectures mais que c'est une très bonne solution pour des projets unitaires. Il utilise l'image suivante : imaginez que vous découvrez un processus chimique dans une éprouvette qui fonctionne à merveille. Pour le mettre en place à grande échelle, est-ce que vous prenez 1000 tubes à essai ou vous réfléchissez à un moyen de mettre en oeuvre votre découverte ? Il recommande de conserver Scrum comme un des moyens de résoudre des problèmes, et que c'est l'utilisation au cas-par-cas de différentes méthodes qui permettent de "monter en charge". Il explique aussi que la technique de "Scrum of Scrum" proposée par l'alliance scrum n'est pas une solution pour monter en charge. On ne peut pas faire du Scrum partout, c'est idiot et cela montre que l'on a rien d'autres à proposer.

## Jug Lyon

[Le Jug Lyon](#) (Java User Group) démarre le mardi 21 avril par une soirée sur GWT et Maven. Organisé par Cédric Exbrayat, Laurent Gayet et Julien Ripault, ce java user group propose chaque 3ème mardi du mois de se retrouver sur la région Lyonnaise afin de partager ensemble un moment sympa autour de la communauté Java. La soirée aura lieu dans les locaux de l'INSA dans le centre de Lyon près de l'université Claude Bernard.

## Rencontre avec les speakers de l'USI

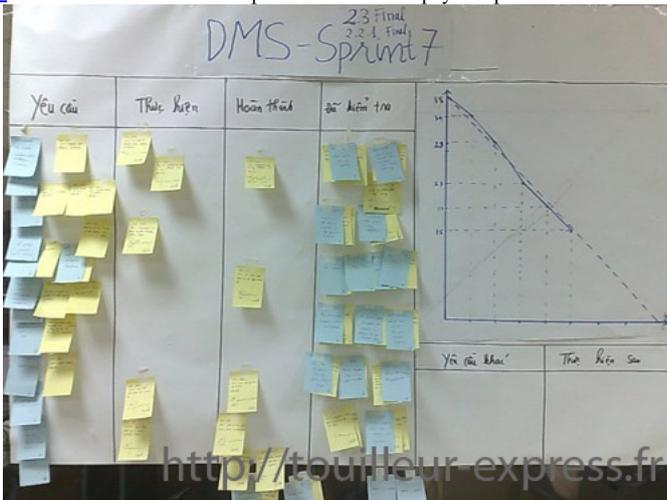
Suite à [un article sur l'Université du SI](#) sur mon blog, j'ai été contacté par OCTO Technology et je suis donc allé hier soir à la réunion de présentations des speakers chez à Paris. Après un introduction par les organisateurs, chacun des présentateurs est passé devant la caméra afin de faire un pitch de 2 minutes sur sa présentation. Le tout suivi d'un buffet, l'occasion pour moi de discuter avec Didier Girard [des domaines en .tel](#) et d'Android, de parler avec [Guillaume Duquesnay](#) de la mise en place de Scrum dans les entreprises. Les présentations de certains speakers comme [Alain Buzzacaro](#) étaient très sympas, avec un ton décalé qui donne envie de s'intéresser au contenu. Au final je pense que nous aurons vraiment un mix entre des présentations très "Boss" et des présentations très "Geeks". L'Agilité est un mot présent dans pas mal de présentation. J'ai eu le sentiment que le message était : la révolution est en marche, comment maintenant s'adapter et adapter son entreprise ? Pour terminer la partie la plus intéressante pour vous : **des cadeaux**. L'équipe des organisateurs de l'USI m'a donné une offre limitée dans le temps et en quantité pour les lecteurs du Touilleur Express : 50% de réduction, rien que ça ! Contactez-moi rapidement par email en indiquant "offre USI" dans le sujet de l'email à nicolas AT touilleur-express.fr, l'email affiché en haut à gauche du blog sous le logo. Premier arrivé, premier servi !

## Podcast épisode 2 sur Scrum avec Dimitri Baeli

J'ai interviewé Dimitri BAELI d'eXoPlatform cette semaine à propos de Scrum. L'interview écrite sera sur le site du [French SUG](#) la semaine prochaine. En attendant j'en ai profité pour mettre en ligne sous la forme d'un podcast l'interview réalisé mercredi dernier. L'interview est disponible sur la page [Podcast](#) du Touilleur. Dimitri nous explique comment l'éditeur français de logiciel open-source eXoPlatform a mis en place la méthode Scrum. La particularité d'eXo c'est qu'il y a 3 centres de développements : en Ukraine, au Viet-Nam et en Tunisie. Il nous explique son rôle chez eXo, les raisons du succès de Scrum parmi d'autres méthodes comme l'eXtreme Programming ou les valeurs du Lean. Le podcast est aussi disponible sur l'iTunes Store, gratuitement bien entendu.

### 3 exemples de tableaux Scrum

Suite à l'interview de la semaine passée, Dimitri Baeli d'eXoPlatform m'a envoyé un petit cadeau : 3 photos de chacun des Kanban Scrum pour le Viêt-Nam, la Tunisie et l'Ukraine. Réécoutez [le podcast](#) si vous souhaitez en savoir plus. A chacun des pays un petit merci : cảm ơn lắm <sup>سكراً</sup>



ДЯКУЮ



eXoPlatform est un

éditeur de logiciel open-source français qui utilise entre autre la méthode Scrum pour le développement de leurs logiciels dans 3 centres de développement. Le Kanban est un tableau d'information qui permet de représenter physiquement et facilement l'état d'un projet. L'utilisation de Post-It (c) permet facilement de mettre à jour la liste des activités à faire durant le Sprint. Le graphique à droite est un Burndown chart. Il représente la quantité de travail à faire. Chaque jour, l'équipe cumule le nombre de points restant à faire et met à jour ce graphique. C'est un indicateur d'avancement. Durant le sprint, si un événement non prévu survient, il est possible de le "parquer" sur une zone du tableau afin de le traiter au prochain Sprint. Si Scrum vous intéresse, [le site du French Scrum User Group](#) propose plusieurs articles afin d'expliquer le fonctionnement de Scrum.

### JavaCamp 4 demain soir

Rendez-vous demain soir au 4ème JavaCamp organisé chez Google, près d'Opéra à Paris. <http://barcamp.org/JavaCampParis4> Comme prévu, je vous proposerai un petit compte-rendu dans

la soirée ou mercredi au plus tard. A demain !

## Le Touilleur Express Manifesto

Cette page très sérieuse présente le "Touilleur Express Manifesto", le seul, le vrai et l'unique. Depuis le début du Blog, les centaines de lecteurs, que dis-je, les millions de lecteurs du Touilleur qui se reconnaissent dans chaque article ne veulent qu'une seule chose : un moyen de se reconnaître les uns les autres. D'où l'idée du Manifesto : un texte simple, que chaque lecteur du Touilleur Express s'engage à respecter afin de donner un avantage caché à ses collègues, à son chef, à un candidat qu'il reçoit en rendez-vous, à un vieux pote qui cherche du travail. Imaginez ces centaines, pardon, ces millions de lecteur qui comme une communauté secrète pourraient se reconnaître les uns-les autres.... Et moi qui traverserait les foules dans une petite voiture blanche blindée avec des gardes du corps... Oui il est temps de créer un lien unique entre nous tous, entre votre porte-monnaie et le mien, entre toi là, oui toi là aussi... Un Manifesto c'est un texte engagé, qui vous propose de se retrouver autour des mêmes valeurs, afin ensuite de créer des groupes, puis des foules, et enfin regrouper toute la communauté Java autour d'une seule bannière, le "Touilleur Express Manifesto". Si vous souhaitez briller en société, montrer un badge sur votre site Internet que personne il l'a, obtenir même pourquoi pas une Certification Touilleur Express, bref avoir des jantes alliages et des petites lumières qui clignotent sur votre site, c'est le Manifesto qu'il vous faut. Vous pouvez aussi imprimer des tes-shirts, des mugs, des nuisettes pour votre copine, des casquettes avec le logo du Touilleur Express, voir des casques de chantier, mais à part si vous faites de la MOA/MOE je vois pas trop l'utilité. En Laouasie du Sud, des jeunes vierges se jettent dans la mer en criant "toooooooooooooiiiiillleuuuuuuur" en signe d'acceptation du Manifesto. Ma mère s'est fait tatouer "Mom of the Touilleur Express" IBM rachètera SUN uniquement SI JE VEUX... Alors il est temps de prendre une grande respiration et de lire le texte suivant à voix haute

### Le Touilleur Express Manifesto

(à lire à voix haute, c'est à dire assez fort en général)

Moi, lecteur du blog le Touilleur Express, je prends les engagements simples, faciles et gratuits suivant. Pour cela je me lève avec ma tasse de café et je m'engage chaque jour sur les idées suivantes: - en tant que recruteur, si lors d'un entretien d'embauche un candidat place les mots "touilleur express", "sardine" et "roquefort" dans la conversation alors je m'engage officiellement à lui verser un bonus sur son salaire annuel. - en tant qu'architecte je m'engage à cacher dans le code de mon application une phrase cachée qui s'affichera le 24 septembre de chaque année. - en tant qu'inventeur d'un langage, d'un produit, d'une application, je m'engage à nommer une class TouiExpress.java avec un code trouvé sur le blog au hasard. - en tant que Scrum master je m'engage à offrir une tournée générale à la fin du premier Sprint afin de fêter cette bonne semaine où l'on s'est bien marré - en tant que Blonde qui fait un Jeu débile sur TF1 je m'engage à arrêter de se mettre devant les lettres à deviner, mince c'est vrai quoi, moi je joue sérieusement à la Roue de la Fortune - en tant que programmeur qui fait du C#, je m'engage... non laissez tomber les gars, c'est déjà assez dur. - en tant que programmeur Java sur Spring, Hibernate et GWT, je m'engage à ne pas prendre la grosse tête et à penser que dans 2 ans ce sera d'autres technos - en tant qu'utilisateur de IBM Websphere, je m'engage franchement à faire un effort, prendre Glassfish v3 et à arrêter de fumer mon tapis de souris - en tant que lecteur des AUTRES BLOGS que le Touilleur, je m'engage DISCRETEMENT à faire un lien vers le Touilleur Express dans les commentaires avec les mots "banane", "la pêche" ou "Java juste fait-le" - en tant que programmeur responsable, je m'engage à éteindre mon PC le week-end (eh ouais... mais si personne fait rien on aura de l'eau dans le salon) - en tant qu'utilisateur d'Eclipse je m'engage à installer IDEA IntelliJ, à effacer Eclipse et à dire tout haut "... ah ouais c'est bien quand même" - en tant qu'utilisateur de Flex je m'engage à ne pas dire de mal de GWT / Wicket. Et inversement pour GWT qui ne doit pas dire de mal sur Wicket. Faites vous des bisous - en tant que participant d'un Paris JUG ou d'un Barcamp, je m'engage à dire de temps en temps "Twix" dans la conversation afin d'identifier un autre lecteur du Touilleur Express. - en tant que client, j'accepte de comprendre que les plannings sur 6 mois vendus au forfait par des entreprises ne sont qu'une grosse fumisterie, autant prendre une carte de Marabout à Barbès ou appeler Madame Irma. - en tant que stagiaire dans une grande entreprise, je m'engage à remercier Mr T.Express dans ma soutenance de stage. Si vous faites une thèse ou si vous écrivez un livre, ça marche aussi, je prends. - en tant que français travaillant à l'Etranger je m'engage à expliquer que le Touilleur Express veut dire "projets qui réussis pilotés par des équipes qui se marrent bien" - en tant que Cyrille de la société X qui tient à garder l'anonymat je m'engage à ne pas poser de questions trop compliquées le 14 avril prochain au Paris JUG, juste à raconter une bonne blague - en tant que lecteur qui est arrivé jusqu'ici, je m'engage à comprendre que ceci n'est qu'une grosse poulade, que sinon dans la vie le Touilleur Express est une personne très sérieuse et qu'il faut bien rire de temps en temps ma bonne dame, hein, et en plus c'est la crise.

Après l'avoir lu, si vous partagez les valeurs du Touilleur Express, signez le manifesto (avec la zone commentaire) et gagnez le droit de mettre ce logo sur votre site Web :



*Le Touilleur Express*

J'ai signé le Touilleur Express Manifesto

<http://touilleur-express.fr>



*Le Touilleur Express*

J'ai signé le Touilleur Express Manifesto

<http://touilleur-express.fr>

Si vous êtes plusieurs, n'hésitez pas à dire ci-dessous dans votre commentaire une phrase comme "Eh ! on est 3 gars dans mon équipe et on va se faire tatouer Touilleur Express" ou une autre phrase si vous avez peur des aiguilles. A vous d'être original !

## Compte-rendu léger de la grosse soirée JavaCamp4

Version light ce matin : - arrivé à 18h30 chez Google, superbes locaux dans Paris place de l'Opéra - première séance sur le Cloud Computing, 20 personnes, discussions autour d'Amazon, de Google, du milieu de la santé. - deuxième séance sur "Quelle technologie choisir pour faire un site Web dynamique, tout en respectant des techniques d'industrialisation" proposé par Eric Lefevre. Discussions sur les technos RIA du moment, avec GWT, JQuery et Flex essentiellement. Didier Girard résume assez bien le choix : pour faire une application web, prendre GWT. Pour moderniser un site existant, faire un site internet dynamique simple, prendre JQuery. Ensuite personnellement je pense que Flex pour une application légère est la meilleure solution. Donc le besoin doit guider le choix de la solution. Ensuite petit buffet de 21h30 à 00h10, discussions sur l'Agilité, sur Python, sur le French JUG, sur la Finance, sur le Touilleur Express, sur le boulot d'écrire un livre, sur les podcasts... etc. Désolé pour cette version légère, je vous redonnerai une part de gâteau ce soir ou demain. Ci-dessous la photo du buffet, la seule que j'ai pris :



Merci à Philippe et à Luc pour l'organisation très pro du JavaCamp, avec distributions de tee-shirt et de clé USB Google à la fin de la soirée. Google nous a vraiment gâté, c'était une bonne soirée.

## Poitou-Charentes JUG, première soirée maven le 10 avril

Vu sur le blog de [Nicolas de Loof](#) (Jug leader breton et commiter Maven) ce matin : il participera à l'inauguration du Poitou-Charentes Java User Group qui aura lieu le 10 avril prochain, soit ce vendredi. Avis à toutes les personnes de la région, de Poitiers à la Rochelle, la soirée sur Maven aura lieu à 19h au Futuroscope dans des locaux prêtés par SERLI. Plus d'information sur [le blog de Nicolas](#) et <http://www.poitoucharentesjug.org/>

## Tests avec SpringFuse, la solution pour gagner du temps

Aujourd'hui au programme, découverte et test de [SpringFuse](#), l'économiseur de temps de Jaxio. Je n'utilise pas le mot générateur de code, car bien que SpringFuse génère en effet du code, c'est plus le temps gagné qui en fait sa vraie valeur. Le principe est le suivant : vous avez une base de données classique, Oracle, MySQL, PostgreSQL ou autre. Comment construire une application web en quelques minutes, avec l'ensemble de son code source ? Je pense à Hibernate Tools, à JBoss Seam, à Grails. Finalement c'est Seam qui se rapproche le plus, sans faire tout ce que fait

SpringFuse. SpringFuse génère le code source d'une application Java avec SpringMVC 2.5.6, Hibernate, Spring Security, bref les Usual Suspects que nous connaissons tous. Le projet généré est un projet Maven 2, avec ses tests unitaires, des services comme l'authentification, l'envoi d'email, la gestion des rôles etc. C'est ce que nous codons habituellement nous-même.

## A toi lecteur qui est sceptique

Petite discussion pour toi qui se dit "... un générateur de code, je n'en n'ai pas besoin". En fait je pense comme toi. Oui tu n'en n'as pas besoin. Nous vivons dans un monde parfait où chacun dispose d'un temps indéfini pour coder. Dans ce monde parfait, tu es un maître de Spring, tu maîtrises Hibernate et les subtilités du cache de second niveau. [Les articles du Touilleur Express](#) sur l'instanciation tardive ? Trop facile. Spring MVC ? Finger in the noise. Tu fais partie bien entendu d'une équipe de 5 développeurs qui tous, avec le même niveau d'expertise, maîtrisent à fond Spring. D'ailleurs Spring 4 ce sera toi, bref rien à faire d'un outil qui finalement... te remplace mon ami. Ecrire souvent la même chose, en faisant attention dans mon DAO à faire ceci, à me souvenir de la configuration du cache, à m'assurer des déclarations dans Spring MVC... au bout d'un moment on a déjà vu le film. L'idée du logiciel SpringFuse n'est pas, d'après ce que j'en ai compris et mes tests, de te remplacer, toi lecteur. C'est juste ton meilleur ami qui se met à côté de toi, qui tape un peu vite du code, et qui est un peu meilleur que toi. Ensuite le principe c'est de te laisser le code, prêt à tourner avec Jetty, mais que tu puisses continuer à travailler dessus et à mettre par exemple du JQuery afin d'améliorer les Widgets.

## Génération spontanée

Il est temps de voir en direct si toutes ses promesses sont tenues. Après m'être inscrit sur le site [SpringFuse.com](#) j'ai en fait suivi [le tutorial](#). Celui-ci marche très bien au passage sur Mac. J'ai la version 2.0.6 de maven par défaut sur Mac, ce qui n'a pas posé de problèmes. Après avoir téléchargé le projet d'exemple afin de créer une petite base [H2Database](#), j'ai lancé la génération du modèle SpringFuse.

```
macbook-pro-de-nicolas-martignole:Dev nicolas$ mkdir SpringFuseTest
macbook-pro-de-nicolas-martignole:Dev nicolas$ cd SpringFuseTest/
macbook-pro-de-nicolas-martignole:SpringFuseTest nicolas$ mvn -v
Maven version: 2.0.6
macbook-pro-de-nicolas-martignole:SpringFuseTest nicolas$ unzip springfuse-example.zip
Archive:  springfuse-example.zip
  inflating: pom.xml
  inflating: README.txt
   creating: src/
   creating: src/main/
   creating: src/main/sql/
   creating: src/main/sql/h2/
  inflating: src/main/sql/h2/comment.sql
  inflating: src/main/sql/h2/create.sql
  inflating: src/main/sql/h2/drop.sql
  inflating: src/main/sql/h2/init.sql
macbook-pro-de-nicolas-martignole:SpringFuseTest nicolas$ mvn initialize
[INFO] Scanning for projects...
Downloading: http://maven2.springfuse.com/com/h2database/h2/1.1.105/h2-1.1.105.pom
Downloading: http://repo1.maven.org/maven2/com/h2database/h2/1.1.105/h2-1.1.105.pom
654b downloaded
Downloading: http://maven2.springfuse.com/com/h2database/h2/1.1.105/h2-1.1.105.jar
Downloading: http://repo1.maven.org/maven2/com/h2database/h2/1.1.105/h2-1.1.105.jar
1095K downloaded
[INFO] -----
[INFO] Building springfuse-example
[INFO]   task-segment: [initialize]
[INFO] -----
Downloading: http://maven2.springfuse.com/org/codehaus/mojo/sql-maven-plugin/1.0/sql-maven-plugin-1.0.pom
Downloading: http://repo1.maven.org/maven2/org/codehaus/mojo/sql-maven-plugin/1.0/sql-maven-plugin-1.0.pom
...
[INFO] [springfuse:extract {execution: Extract Database Meta Data}]
[INFO] Ready to extract the database schema
[INFO] Database schema extracted
[INFO] File data-model.springfuse passed reverse conversion OK
[INFO] File data-model.springfuse created successfully
[INFO] You are now ready to upload data-model.springfuse to http://www.springfuse.com/ and generate your project!
[INFO] -----
[INFO] BUILD SUCCESSFUL
[INFO] -----
[INFO] Total time: 43 seconds
[INFO] Finished at: Thu Apr 02 09:08:52 CEST 2009
[INFO] Final Memory: 7M/17M
[INFO] -----
macbook-pro-de-nicolas-martignole:SpringFuseTest nicolas$ ls -l data-model.springfuse
-rw-r--r--  1 nicolas  staff  1764  2 avr 09:08 data-model.springfuse
macbook-pro-de-nicolas-martignole:SpringFuseTest nicolas$
```

L'exécution génère un fichier data-mode.springfuse à partir de la base de données. Le modèle est un fichier au format [Google ProtocolBuffer](#). De retour sur l'interface web de SpringFuse, après m'être authentifié je vais dans "My Account > Generate New Projet". Je conserve le nom conseillé par le tutorial "quickstartdb" et je génère ensuite mon code. Le générateur SpringFuse est une application hébergée sur une architecture de cloud-computing bien connue, Amazon EC2. La génération est vraiment rapide, quelques instants plus tard, je télécharge le code généré qui fait 330kb. Il est temps de lancer ~~Eclipse~~, pardon, IDEA IntelliJ 8.1. Je décide de créer un nouveau projet en important le code eclipse. En 10 secondes, IDEA IntelliJ me génère un projet, il détecte au passage Spring et un module de type Web. Wooow cela semble bon du premier coup.

## Premiers pas

Je tente un `mvn -Ph2 jetty:run` à la racine de mon projet quickstartdb... Spring 2.5.6 est récupéré via Maven2, le tout se passe en quelques minutes le temps de récupérer les librairies. Il est temps de lancer un navigateur et de regarder tout d'abord la partie vue, avant d'aller voir sous le capot et de regarder le moteur. Une fois le site lancé, je découvre la partie vue de l'application. Au premier abord cela semble un peu austère, mais la documentation donne tout de suite quelques pistes à tester. Je m'authentifie en tant qu'admin, et je commence à tester la modification des rôles, à télécharger un document vers le site, ensuite à tester les recherches avec les différentes fonctions de recherche. Le tout fonctionne vraiment rapidement. L'interface propose d'effectuer une recherche par Pattern type Query Hibernate, ensuite une recherche par l'exemple et enfin une recherche utilisant les NamedQuery d'Hibernate. Voyons maintenant le code généré. La structure du projet maven2 est vraiment claire. L'organisation du code est faite par domaine, par service. Aucune dépendances cycliques signalés par IDEA IntelliJ entre les packages. Les tests unitaires sont générés pour chacun des éléments. Avec la partie web, le projet complet fait 133 classes Java. Il y a aussi du code pratique qui fait gagner du temps comme un service d'envoi d'email templatisé avec Velocity, la gestion de l'authentification, un service pour vous retourner un mot de passe si vous l'avez oublié, etc. Le moteur d'authentification basé sur SpringSecurity est simple, j'ai l'impression de visiter un grand appartement type loft, très clair et bien agencé. Poursuivons la visite.

## La qualité du code

En regardant le modèle tout d'abord, je suis allé chercher quelques problèmes connus avec Hibernate, que nous voyons sur le terrain afin de voir comment SpringFuse les adresse. Je commence simplement par la gestion de l'identité des entités, avec les méthodes `equals()` et `hashCode()`. Comment sont surchargées les méthodes `equals()` et `hashCode()` des entités mappées par Hibernate ? Par défaut, Java fournit une implémentation pour les méthodes `hashCode()` et `equals()`, qui permet de définir l'identité d'un objet. Chaque nouvelle instance sera donc différente, ce qui est en principe pratique. Si vous souhaitez affiner votre gestion des objets en mémoire, en surchargeant `equals()` et `hashCode()` vous pouvez alors définir votre notion de l'égalité. Hibernate se charge de passer le contenu d'un objet en mémoire vers la base de données et inversement. Il est donc tentant de se reposer sur les clés primaires d'une base de données pour dire que deux objets sont égaux. Mais est-ce suffisant ? Chaque Session Hibernate a un cache. Lorsque vous créez un objet avec `new()`, puis qu'ensuite vous le sauvez, Hibernate sait ensuite sans problème le gérer. Les problèmes surviennent lorsque vous fermez ensuite votre Session. Si vous recherchez "le même objet" vous verrez qu'Hibernate vous retournera une autre instance au sens Java, si vous ne surchargez pas `equals/hashCode`. Car même si les objets sont identiques en terme de contenu, ce ne sont pas les mêmes. En principe cela ne pose pas de problèmes... tant que vous n'utilisez pas de Collections. Dans une classe avec une association one-to-many, un Set d'éléments utilisera donc par défaut le couple `hashCode>equals` afin de stocker vos objets. Pour peu que ce code ne soit pas surchargé, ou mal surchargé, les problèmes de gestion des collections surviennent... (voir <http://www.hibernate.org/109.html?cmd=prntdoc>) La documentation d'Hibernate dit clairement :

The general contract is: if you want to store an object in a List [1], Map [2] or a Set [3] then it is an requirement that equals [4] and hashCode [5] are implemented so they obey the standard contract as specified in the documentation.

La bonne nouvelle est que SpringFuse génère un code particulièrement "balèze" pour ces cas difficiles, qui sont loin d'être facile à maîtriser. Des tests unitaires poussés sont aussi générés afin de s'assurer que l'ensemble fonctionne comme un V8 de Mustang et pas comme un moteur de Solex. Regardons ensemble la gestion de l'égalité (`equals` et `hashCode`) sur la classe `AccountModel` généré par SpringFuse:

```
/**
 * Indicates whether some other object is "equal to" this one. For information on the
 * current implementation you can
 * read the discussion about object
 \* identity on hibernate.org
 *
 * @see java.lang.Object#equals(Object)
 * @see #setEqualsAndHashCodeStrategy()
 * @return true if the equals, false otherwise
 */
public boolean equals(Object account) {
    if (this == account) {
        return true;
    }

    if (account == null) {
        return false;
    }

    if (!(account instanceof AccountModel)) {
        return false;
    }

    AccountModel other = (AccountModel) account;

    setEqualsAndHashCodeStrategy();

    if (this._useUidInEquals != other._useUidInEquals && other._freezeUseUidInEquals) {
        if (logger.isErrorEnabled()) {
            logger.error("Limit case reached in equals strategy. Developer, fix me", new Exception("stack trace"));
        }
        throw new IllegalStateException("Limit case reached in equals strategy. Developer, fix me");
    }
}
```

```

    }

    if (_useUidInEquals) {
        boolean eq = _uidInEquals.equals(other._uidInEquals);

        // ensure (during dev) that equals contract is respected
        if (eq && hashCode() != other.hashCode() && logger.isErrorEnabled()) {
            logger.error("Limit case reached in equals strategy. Developper, fix me", new Exception("stack trace"));
        }

        return eq;
    } else {
        if (other.getAccountId() == null) {
            return false;
        }
        boolean eq = getAccountId().equals(other.getAccountId());

        // ensure (during dev) that equals contract is respected
        if (eq && hashCode() != other.hashCode() && logger.isErrorEnabled()) {
            logger.error("Limit case reached in equals strategy. Developper, fix me", new Exception("stack trace"));
        }

        return eq;
    }
}

private void setEqualsAndHashCodeStrategy() {
    if (!freezeUseUidInEquals) {
        freezeUseUidInEquals = true;
        _useUidInEquals = useUidInEquals();

        if (_useUidInEquals) {
            _uidInEquals = new java.rmi.dgc.VMID();
        }
    }
}

private boolean useUidInEquals() {
    return !hasPrimaryKey();
}

```

Un peu plus loin nous avons aussi un bout de code Java qui donne une idée de ce qu'il est possible de faire avec le code généré par SpringFuse :

```

if (_useUidInEquals) {
    _uidInEquals = new java.rmi.dgc.VMID();
}

```

Que ceux qui connaissent ce qu'est qu'un VMID lèvent la main... Ok deux personnes, Cyrille et qui d'autre ? Florent... Bon ça fait pas beaucoup de monde non ? VMID c'est un identifiant d'objet unique par Java Virtual machine, qui est surtout utilisé pour identifier dans un cluster votre objet. Jusque là vous buvez votre lait fraise et vous vous demandez où je veux en venir. Il se trouve que le code généré par SpringFuse fonctionne plutôt bien avec le cache de second niveau d'Hibernate. Et que ce cache de second niveau permet d'échanger entre deux java virtual machines des objets... Bref SpringFuse vous génère du code qui sera capable de fonctionner dans un cluster. Bon courage pour le faire vous même du premier coup ! La javadoc de SpringFuse vous propose de lire la page suivante : <http://www.hibernate.org/109.html>. Si vous regardez le code, vous avez ici un moyen vraiment intelligent d'aller plus loin et de comprendre mieux Hibernate. Et cela ne fait que commencer...

## Comment SpringFuse gère les relations one-to-many et many-to-many ?

Je suis ensuite allé regarder la gestion des associations one-to-many et many-to-many. Certains disent que les relations many-to-many sont plutôt un signe que le modèle est pourri. Ce que l'on voit sur le terrain, c'est que nos clients ont cette modélisation sur les bases de données de nos vraies applications de gestion. Et souvent il est impossible de remettre en cause le modèle relationnel, d'où l'importance alors d'avoir des best-practices. Il faut donc apprendre à gérer correctement ce type de relation. La class AccountModel déclare une relation one-to-many avec la class Document. Un utilisateur a un ou plusieurs Document dans l'application. Géré avec un HashSet dans le modèle, le set est déclaré dans le fichier de mapping comme suit:

```

<set name="documents" inverse="true" cascade="all"
    lazy="extra" batch-size="10" embed-xml="false"
    node="documents">
    <key column="ACCOUNT_ID"/>
    <one-to-many class="com.touilleur.domain.DocumentModel" />
    <filter name="myDocumentFilter"/>
</set>

```

Souvenez-vous dans un ancien article du Touilleur Express que vous avez lu en février, la gestion du chargement tardif avec la ligne "**lazy=extra**" permet de faire travailler la base de données correctement. Si vous souhaitez connaître le nombre de Document d'un utilisateur, Hibernate exécutera donc une simple requête select count SQL au lieu de créer la collection de Document en chargeant l'ensemble des Documents de la base, puis d'utiliser la méthode size() sur le Set... Encore une bonne pratique offerte sur un plateau. Je continue la visite.

## Le mapping Hibernate

Prenez ces quelques lignes au début du fichier AccountModel.hbm.xml qui montre les choix d'optimisation proposés par SpringFuse.

```

...
<hibernate-mapping>
  <class name="com.touilleur.domain.AccountModel"
    table="ACCOUNT"
    lazy="true"
    dynamic-update="true"
    optimistic-lock="version"
    batch-size="30"
    node="account">

```

L'attribut "dynamic-update" est ici à true. Il permet de générer une clause UPDATE en SQL ne contenant que le nom des colonnes modifiés, afin d'éviter de transmettre un graphe complet d'objet vers la base. La valeur du paramètre batch-size qui est de 1 par défaut, est aussi un moyen d'améliorer la stratégie de chargement par lot. Lorsque vous affichez une page avec un tableau listant le nombre d'utilisateurs, cela permet de récupérer par lot de 30 les Accounts de votre base. Encore une fois, des optimisations simples qui donneront de meilleurs résultats.

## Les tests

Les tests unitaires utilisent la nouvelle version de JUnit, basée sur des annotations. Pour certaines classes comme les DAO, SpringFuse utilise [easyMock](#). Au passage, si vous n'avez jamais "mocké" de code, c'est un excellent tutorial sur les bonnes pratiques de l'écriture des tests. Nous sommes donc en plein TDD (Test Driven Development) et la qualité industrielle du code est là.

## La partie Spring MVC

Du côté du framework web, c'est Spring MVC. A part la littérature, c'est un framework que je ne connais pas, sur lequel je n'ai pas d'expériences. C'est donc pour moi l'occasion de faire le tour du propriétaire afin de regarder comment est structurée l'application. La première bonne nouvelle c'est qu'il s'agit de la dernière version utilisant les annotations. Le code est donc vraiment simple et clair.

## Exemple d'astuce offerte par SpringFuse

Une astuce trouvée dans la classe NamedQueryUtilHibernate illustre le principe de SpringFuse, qui est de vous donner des solutions gratuitement. Pour la suite de cette partie, nous allons regarder les requêtes nommées dans Hibernate (named query). Le principe est de permettre de déclarer dans un fichier XML (named-query.xml) les requêtes au format HQL afin d'éviter de les définir dans le code Java. Cela permet ensuite de les maintenir plus facilement, entre autres. Par exemple voici comment SpringFuse propose de déclarer une requête

```

<query name="getAllAccountsQueryName" comment="enableDynamicOrderBySupport">>
  from com.touilleur.domain.AccountModel
</query>

```

Cela permet ensuite d'appeler la query `getAllAccountsQueryName` directement à partir du code:

```
session.getNamedQuery("getAllAccountsQueryName").list();
```

L'une des limitations des requêtes nommées c'est qu'il n'est pas possible de spécifier un ordre de tri. Imaginez un tableau avec 4 colonnes, l'utilisateur clique sur la colonne Ville et vous souhaitez trier le tableau, et aussi paginer le résultat... On voit dans le code XML ci-dessus que SpringFuse a déclaré un commentaire dans l'attribut "comment" avec pour valeur ce qui ressemble plus à un mot clé qu'un commentaire : "enableDynamicOrderBySupport". La réponse se trouve dans la classe NamedQueryUtilHibernate, regardez la méthode `recreateNamedQueryRecreationWithOrderClauseIfNeeded`, vous verrez que le code va recréer une nouvelle requête automatiquement avec un ordre de tri, si vous avez spécifié ce mot clé dans le champ commentaire ! Cela peut vous paraître obscur, vous n'imaginez pas la complexité pour expliquer en quelques lignes cette idée, mais elle est vraiment simple et astucieuse. Au final, tout est transparent puisque vous utilisez cette astuce via les manager générés par SpringFuse. Cela illustre un principe de SpringFuse : **ce n'est pas un logiciel open-source mais un logiciel open-minded**. Ce genre de code pourrait se retrouver au fin fond d'un JAR, sans que vous n'ayez la possibilité de regarder le code. Ici c'est le contraire, l'objectif c'est de vous donner des stéroïdes afin de renforcer vos abdominaux de développeur Java. J'apprécie la distribution de bonnes idées.

## Conclusion

SpringFuse est gratuit jusqu'à 10 Tables de base de données. C'est un moyen transparent et simple de vous laisser tester la solution avec votre application. J'imagine maintenant plusieurs usages pour SpringFuse: - commencer un projet rapidement - apprendre de nouvelles technologies comme Hibernate ou Spring MVC comme un livre dynamique - reprendre un projet avec une forte dette technique afin de le dynamiser rapidement - gagner du temps sur la partie technique pour se faire plaisir sur la partie métier J'écris ces lignes peut-être trop emballé. Il est aussi temps de parler sur des regrets ou des questions. Pourquoi ne pas avoir tel ou tel framework de rendu encore plus joli comme JQuery par exemple ? Je pense que finalement SpringFuse fait le choix intelligent de faire très bien ce qu'il y a de plus important dans une application : les fondations. Libre à vous ensuite je pense de placer une librairie de rendu en plus de ce que vous donne le framework Spring MVC. Le lien pour tester : <http://www.springfuse.com>

## Prochaine soirée Paris JUG sur Scrum et les

## méthodes Agiles

Au programme de la prochaine soirée du 14 avril du [Paris JUG](#) : Scrum. Pour une fois je ne serai pas dans la salle à prendre des notes, mais face à vous puisque c'est en compagnie d'Eric Mignot de [Pyxis Technologies](#) que je vous propose de découvrir Scrum durant la première partie. Mon objectif est de vous parler de Scrum pour vous qui êtes développeur, architecte ou chef de projet. Vous qui venez habituellement au Paris JUG pour voir des sujets plutôt techniques, ce soir l'idée c'est de vous expliquer simplement ce qu'est Scrum. Quel est son fonctionnement ? A quoi sert Scrum ? Qu'est-ce que les méthodes Agiles ? En quoi le sujet me concerne-t-il ? A quels problèmes cela répond-t-il ? Du côté pratique, j'enregistrerai la soirée afin d'en faire un Podcast, et je vous propose aussi d'utiliser Twitter. Pendant la soirée, vous pourrez envoyer vos questions à "[Letouilleur](#)", je lirai via mon iphone vos questions, vos remarques afin d'animer la soirée sans couper le rythme pour tenir nos 57 minutes de présentation. J'ai rencontré Eric lors du [premier barcamp](#) organisé par Luc Bizeul avec l'aide d'OCTO Technology en octobre dernier. J'ai apprécié le bonhomme, à l'aise pour expliquer Scrum, avec une bonne expérience de sujets connus et de questions que nous nous posons tous, entre autre sur la contractualisation. Eric a un autre point en commun avec moi, quoiqu'il s'en défende, c'est l'utilisation d'analogie pour expliquer simplement des choses compliquées. J'ai préparé mes fiches, je réserve quelques surprises à Eric afin que l'heure soit animée, que nous puissions couvrir notre partie sans aussi déborder chez le suivant : Guillaume ! La deuxième partie de la soirée sera consacrée à la mise en place de Scrum par [Guillaume Bodel](#) de Xebia. Quels sont les obstacles que l'on peut rencontrer ? Quels sont les effets de la mise en place de Scrum ? Basé sur son expérience du terrain, Guillaume nous présentera comment adapter la mise en place d'une méthode Agile à notre environnement, à notre [biotopie](#). Il est intervenu à plusieurs reprises sur [TV4IT](#), ainsi que sur le blog de Xebia France : "[Pourquoi les projets agiles ne peuvent pas \(vraiment\) être menés au forfait ?](#)". Les inscriptions à la soirée ne sont pas encore actives, je pense que l'équipe du Paris JUG s'en occupera bientôt, [rendez-vous sur le site du Paris JUG](#). A mardi soir prochain donc !

## Java et Groovy 1.6.1 supportés sur Google AppEngine

Ce matin la nouvelle s'est propagée via Twitter en quelques heures. [@DidierGirard](#) a annoncé ce matin vers 06h00 l'information, j'ai vu aussi que les JavaPosses avaient assisté à un événement ([@DickWall](#)). C'est surtout [Guillaume Laforge](#) qui est fier d'annoncer que Groovy sera supporté par [Google AppEngine](#). SpringSource [a publié ce matin un tutorial](#) en anglais pour écrire sa première Groovlet et la déployer sur GoogleAppEngine, qui est un serveur d'application déployé en cloud-computing. D'ici une semaine nous devrions avoir un Gtwitter, twitter écrit en Groovy au rythme où vont les choses ! Didier a déjà publié une news sur [son blog](#), je le laisse donc passer devant, le Touilleur Express vous donnera d'autres informations demain dans la soirée. Stay Tuned ! Il y a encore d'autres choses à venir. Voir : [L'annonce sur le blog SpringSource du support de Groovy](#) <http://googleappengine.blogspot.com/2009/04/seriously-this-time-new-language-on-app.html>

## Avez-vous une identité numérique ?

Dimanche matin je voyais sur Twitter quelques derniers messages, dont l'un de Luc Bizeul de retour du premier [E-Réputation BarCamp](#). C'est un barcamp sur la gestion de son identité sur Internet, sa e-reputation Petit test pour commencer: si vous tapez votre nom sur Google, que se passe-t-il ? Qu'est-ce que l'identité numérique ? Comment pouvons-nous la contrôler ? Quelle rôle joue-t-elle dans le processus de recrutement ? L'identité numérique est votre carte d'identité sur Internet. C'est la représentation de tous les forums où vous avez posté un message, de vos profils Facebook, LinkedIn ou Viadeo. C'est les sujets de votre blog, de votre page Twitter, des mailing-lists auxquelles vous avez participé... Bref c'est votre histoire personnelle racontée par Internet, par un moteur de recherche. Encore plus cruel ou direct, avez-vous tenté de taper votre prénom et nom sur un site comme [123People.fr](#) ? Ce site est assez bête et puissant, il se contente de vous donner ce que vous avez laissé traîner sur Internet. Si vous ne souhaitez donc pas y apparaître, il vous suffit d'aller faire le ménage. L'un de mes collègues de travail a vu ainsi sa liste d'achat Amazon.com apparaître sur ce site... Marrant ! Pour peu que votre voisin de bureau ait mis ses photos de mariage, les photos du dernier bébé, en quelques cliques de souris vous saurez tout sur lui. Comment contrôler son identité ? Il semble difficile voire impossible de la contrôler complètement. Certains fuient à tout prix le côté "je m'expose" d'Internet. Ecrire un blog ou même poster un message avec une vraie identité ? Inconcevable ! Je me demande à quel moment il sera devenu anormal de ne pas apparaître dans les pages de recherche Google... Dans la recherche de candidats, nous nous rendons compte dans notre milieu que l'usage des moteurs de recherche prend de plus en plus le pas sur la simple lecture du CV. En effet, il est beaucoup plus intéressant pour un recruteur de voir que l'an passé vous êtes allé au Tchad sur votre page Facebook pour aider des enfants, que de savoir que votre niveau d'allemand est "scolaire". Il sera peut-être amusé de voir aussi vos réponses enflammées sur telle liste de diffusion, ou sur tel blog, au moins ce candidat a quelque chose à dire. L'exercice de la maîtrise de son image n'est pas un sujet enseigné aux étudiants, alors qu'il est clair que cela fait partie maintenant du package pour vous aider à trouver un boulot. Un salarié d'une entreprise avec sa e-réputation apporte un capital d'affect à l'identité de son entreprise. Regardez LinkedIn et Viadeo, qui vous propose de vous regrouper avec vos "collègues". Qui a vraiment envie d'associer son identité avec le nom d'une grande entreprise qui licencie en ce moment ? Le prix à payer de cette transparence c'est d'accepter dès le départ de courir le risque que l'attachement à une entreprise puisse se retourner contre vous. Une fois votre identité numérique en place, pour peu que vous commenciez à être connu, il vous arrive des trucs

marrants, ou pas. Frederic Cavazza l'an passé [raconte ainsi](#) qu'un escroc se faisait passer pour lui sur Viadeo, afin d'attirer et de faire payer des conseils... Bref il s'est fait voler sa propre identité numérique... A suivre

## Recrutement en 2009

*Pierre est directeur des ressources humaines chez un grand éditeur. La crise a du bon, les candidats se montrent moins gourmands, il est temps de commencer à chercher de nouveaux candidats pour renforcer ses équipes. Mais par quoi commencer ? Comment trouver de bons candidats dans cet univers de personnes qui font du Java ? Voyons comment il va mener sa quête au candidat idéal... Pierre lit comme chaque matin le blog le Touilleur Express, une tasse de café à la main :*

Un rapide tour des différents acteurs du marché de notre secteur montre que les pages de recrutement sont encore plutôt bien remplies. OCTO Technology a publié un communiqué de presse [le 30 mars dernier](#) afin d'annoncer le désir de recruter jusqu'à 40 personnes (sur 100 collaborateurs actuellement) d'ici à fin 2009. La crise actuelle nous a pas fait oublier ce qu'il se passait il y a deux ans : les profils Java avec 2 à 4 ans d'expérience étaient rares et difficiles à intéresser. Quand le marché se tend, c'est aussi l'occasion de ré-équilibrer la relation candidat-employeur. Il y a deux ou trois ans, nous parlions de speed-dating, de soirée Poker afin de séduire des candidats. Aujourd'hui le vent a tourné.

Il est bien gentil, mais je ne trouve pas grand chose de neuf pour lancer ma chasse... Notre entreprise a décidé de se repositionner sur la plateforme Java. Nous souhaitons profiter de ce début d'année afin d'engager rapidement 3 développeurs Java. Je prends mon cahier et commence à lister mes pistes pour trouver quelques potentiels : - cooptation et réseau d'anciens - réseaux sociaux - annonce sur sites classiques comme Monster.fr - chasseur de tête - ... Ok, voyons d'abord la cooptation. Le plan chez nous est que chaque salarié touche jusqu'à 2000 EUR, si son coopté passe avec succès la période d'essai. L'an passé sur nos 550 collaborateurs nous avons ainsi recruté... attendez où est ce powerpoint ? ah c'est là !!! 4 personnes. Bon pas un franc succès... Que raconte le Touilleur là dessus ?

[...] Ce réseau quelque part je le trouve assez plat. Avez-vous envie de recruter 2 Paul Durand promo 97 ? pas forcément. Du côté des anciens collègues, le mercato entre SSII est "réglementé". Afin de suivre un système éthique, les acteurs des différents cabinets responsables s'accordent afin de ne pas se piquer les consultants les uns les autres. Il est donc parfois difficile de contacter d'anciens collègues [...]

Bref je peux faire passer un email en interne afin de relancer la machine, peut-être organiser un concours et faire gagner une participation à un salon professionnel... A réfléchir.... Je mets donc 4x2000 = 8000 EUR de budget sur la première ligne de mon cahier et je passe à la suivante. Les réseaux sociaux... Mes assistantes de recrutement sont toutes inscrites sur Viadeo, LinkedIn, MySpaces, Facebook... mais je trouve que le résultat ne va pas dans mon sens : RECRUTER ! L'an passé nous avons par contre décidé d'utiliser systématiquement Viadeo et LinkedIn pour "profilier" nos candidats, détecter et valider les profils. Je ne me suis pas privé de téléphoner à quelques personnes de mon réseau pour me renseigner sur mes candidats, ce qui m'a aussi permis de repêcher des profils mal vendus sur LinkedIn... Dingue comment cet outil peut être aussi tranchant lorsqu'il est mal utilisé. Le deuxième souci c'est que mon réseau ne voit pas ces candidats. Nous avons beaucoup de mal à nous connecter à eux. En effet, moi j'ai mis mes amis DRH du réseau "DRH User Group" mais à part eux, j'avoue que je n'ai pas de contacts intéressants... Bon je mets 0 candidat directement, par contre je prends un abonnement professionnel à Viadeo et LinkedIn pour toute l'équipe, hop, 5000 EUR. Voyons ce que dit le Touilleur ensuite sur les annonces classiques sur [Monster.fr](#) ou [Lesjeudi.com](#)...

[...] Les annonces sur les sites classiques me font penser à du chalutage en règle. Une campagne de 15 jours coûte 560 EUR pour une seule annonce sur Monster.fr en mars 2009. Pour ce prix, avec quelques mots clés bateaux comme J2EE et Struts, vous êtes certain en tant que recruteur de ramasser un peu tout ce qui traîne. Pas de soucis ensuite si vous pouvez recevoir 50 CV, peut-être plus aujourd'hui, et que vous avez moyen de trier rapidement les poissons...

Ah. Mais d'où il sort ces chiffres ? aaah je vois cela sur [la page Entreprise de Monster.fr](#) Bigre il a raison... Voyons si je mets 3 annonces pour 30 jours... 1515 EUR HT. Ok je note cela dans ma ligne, c'est le prix. Là où il a raison c'est le coût caché. Je me souviens l'an passé, nous avons publié une annonce mal rédigée par un chef de projet débordé. Il avait mis Sturts au lieu de Struts... Nous ne savions pas que c'était pas bon... Hop une annonce pour rien. Et cette autre annonce où nous avons reçu un nombre impressionnant de CV, c'est le travail de Sandrine à plein temps... Appeler les candidats qui ont trouvé entre temps, vérifier les CV, les références... Et au final je ne suis toujours pas certain de trouver les meilleurs candidats. Rien sur les chasseurs de têtes... Mais est-ce que je suis prêt à payer une chasse pour 3 profils de développeur Java ? Voyons combien m'avait coûté la chasse de ce chef d'équipe. Salaire annuel : 75 000 EUR avec primes, et nous avons versé 10% au chasseur, soit 7500. C'est un excellent élément mais je ne suis pas certain de pouvoir faire la même chose pour 3 candidats, avec une fourchette de salaire allant de 35 à 55 k EUR... J'abandonne cette idée. Bon que raconte le Touilleur Express ?

[...] les recrutements seront peut-être moins dynamiques, mais nous sortons d'une période qui était exceptionnelle. Il faut donc continuer à renforcer son CV et à se faire connaître des recruteurs en venant par exemple au Paris JUG. A mardi prochain !

Bon, j'avoue que je n'ai toujours pas trouvé de nouvelles idées pour recruter. Je vais essayer de

contacter Nicolas pour voir... - Salut Nicolas, c'est Pierre de GrosEditeurQueTuConnais, - Salut Pierre [longues discussions sur pleins de sujets] - Et sinon, tu as des idées pour trouver de nouveaux candidats ? des gens bons en Java, sympathiques, motivés ? - Et bien tu as la réponse à la fin de l'article, relis-bien - ... désolé je ne vois pas - Paris JUG - ... ? - C'est une association qui permet à des passionnés de Java de se retrouver chaque mois, qui fonctionne grâce à des sponsors. - Et en quoi cela va-t-il m'aider moi GrosEditeurQueToutLeMondeConnais ? - Tu peux te faire connaître de la communauté, avec un budget raisonnable tu peux avoir accès aux personnes les plus geeks de la communauté Java, qui elles-mêmes connaissent d'autres personnes - Bah si c'est pour payer de la bière de 10 personnes - Pas du tout, c'est très sérieux. Il y a plus de 180 personnes à chaque fois, les réunions sur Paris sont courues par le tout-paris, les geeks les plus connus de la communauté open-source Java. - Comment cela fonctionne-t-il ? - Tu peux devenir sponsor à l'année. J'ai en tête le cabinet Zenika qui s'est fait connaître au même niveau que Xebia ou Valtech par exemple, et qui organise aussi des événements - ... et si je souhaite simplement participer une fois ? - et bien tu peux devenir sponsor de la soirée seulement. Cela te permet déjà de communiquer. - Wow... et ça marche ? - Je ne dis pas que ça fait repousser les cheveux, mais tu as accès à une mine d'or, que peu de personnes connaissent... Ensuite à toi aussi de participer et de proposer à tes équipes de faire une présentation par exemple - Ok, je contacte qui maintenant ? - Tu envoies un email à team@parisjug.org de ma part ou sinon sur le site du [ParisJUG](http://ParisJUG) - Merci Nicolas ! - De rien ! :-)

## Soirée Wicket chez Zenika

Zenika organise le mercredi 29 avril à partir de 18h30 une soirée sur Wicket. Au programme : présentation de Wicket par [Martijn Dashorst](#), l'un des principaux committers de Wicket, et co-auteur du livre "[Wicket in Action](#)" avec Eelco Hillenius. L'inscription et les détails sont [sur le site de Zenika](#). La présentation se déroulera [comme lors de la dernière soirée sur Lean](#), dans l'Auditorium du Club Confair au 54, rue Laffitte, 75009 Paris.

## Google AppEngine : retour de soirée

 Google a annoncé le support de Java et de Groovy pour la plate-forme de CloudComputing, Google AppEngine. L'annonce finalement a été faite dans la nuit de mardi à mercredi, nous savions donc le contenu de la soirée mystère. Car il faut aussi vous raconter que tout ceci a commencé il y a un mois. Google a organisé discrètement ce rendez-vous avec 50 personnes. Pour mon égo-mètre qui ne demande qu'à partir au quart de tour, j'étais très content de faire partie des happy-few. Bon, en fait nous nous sommes retrouvés comme par hasard avec les Usual Suspects, à part quelques têtes plus rares, ou venues de loin comme Nicolas de Loof ou [Sami Jaber](#). Dans la première partie de la soirée Maxime TIRAN de Google France nous présente la démarche de Google. Le navigateur est le client du futur, l'effort d'un navigateur comme Chrome est de proposer un outil puissant, afin de pouvoir proposer de nouveaux services. Les dernières technologies Webs comme HTML5 basées sur des standards ouverts visent à rattraper Flash, Java FX et Silverlight. Google se positionne aussi sur l'objectif de faciliter l'accès aux données et d'offrir un socle de Partage. Maxime mentionne le projet [Shindig](#) hébergé par Apache, un conteneur OpenSocial qui est une implémentation de l'API OpenSocial. Google souhaite proposer à la communauté des développeurs des présentations, comme cela se fait aux Etats-Unis, afin de prendre en compte votre point de vue. Il rappelle que les projets sur code.google.com sont suivis par les équipes en interne chez Google. Dans la deuxième partie de la présentation, Didier Girard de SFEIR nous propose une présentation des nouveautés de cette version d'AppEngine. Didier explique en quelques mots le principe du Cloud-Computing et les 3 types de service. Il y a tout d'abord le cloud computing physique, où l'infrastructure est un service (IaaS). C'est l'hébergement classique comme chez Gandi.net qui vous permet de rajouter du CPU ou de la mémoire sur votre serveur. Vient ensuite l'hébergement de services, comme Salesforce.com. Le principe est d'héberger à l'extérieur un service comme la gestion client (CRM) ou vos emails (Google Mail). Enfin, et c'est le principe d'AppEngine, l'hébergement de type plateforme (Platform As A Service). Google AppEngine est un serveur d'application qui vous permet de faire tourner des applications écrites en python (depuis un an) ou en Java/Groovy (depuis mercredi dernier). Ses équipes ont développé deux applications qu'il nous montrera ensuite plus tard dans la soirée. Il a eu en effet la possibilité de tester il y a plus de deux mois, les premières versions bêta du support de Java, avant l'ouverture aux développeurs. Un exemple d'application hébergée sur Google AppEngine assez célèbre : <http://www.whitehouse.gov/openforquestions/>. Oui il s'agit bien de la maison blanche de Barack Obama. Il s'agit d'une application développée à l'occasion d'une séance de questions ouvertes posées par les Internauts. La plate-forme de Google a permis d'offrir des performances le soir de l'événement assez importante. Il y a aussi d'ailleurs du GWT. C'est une application en Python. Les 5 nouveautés de cette nouvelle version de [Google AppEngine](#) sont : 1) La possibilité d'acheter de la puissance supplémentaire 2) Le support de tâches type cron déclenchées par une URL 3) Support d'une base de données type BigTable 4) L'accès à travers votre firewall à vos applications d'entreprise (SDC) 5) Early look at Java Support -> premier regard au support de Java Concernant le point 1 : l'offre est gratuite jusqu'à 5 millions de pages vues par mois. Autant dire, on a le temps de voir venir. Je n'ai aucuns détails sur le pricing, nada, rien. Impossible de comparer avec Amazon EC2. Les offres ne sont de toutes les façons pas comparables. Le support des tâches de type cron : il suffit de placer un fichier XML spécial dans son WEB-INF, avec une URL et une syntaxe type Cron. Au déploiement, le moteur de Google AppEngine voit que vous souhaitez qu'il appelle l'url /mail/sendMailingList.do par exemple... Bref il faut que votre service soit exposé sous la forme d'une URL. Nada sur la sécurité... je pense cependant qu'il est faisable de protéger cette url. Peut-on ne pas la déclarer dans le web.xml ? non. Je dois creuser ce sujet. Concernant le support du stockage de données, Didier nous a montré sur une petite application le principe de l'utilisation de JDO. Basé sur DataNucleus, un projet en license Apache v2, le support JDO permet de vraiment écrire du code portable. C'est plutôt une bonne nouvelle. [Plus de détails sur cette page](#).

Vient ensuite un sujet intéressant. Si je code demain une application pour mon client, et que je la déploie sur Google AppEngine, se pose alors le problème de l'accès aux données du SI. Google propose le [Secure Data Connector](#) (SDC). SDC est un logiciel client déployé de l'autre côté de votre firewall qui communique via HTTPS avec la plate-forme Google AppEngine. Ce système permet de construire un pont entre le SI de l'entreprise et le monde de Google. Du côté de votre SI, il faut que vos services soient exposés sous la forme d'URL. J'ai donc compris qu'il faut que votre système d'information s'expose sous forme d'url HTTP, avec une architecture de type REST. Je vois déjà les administrateurs de sécurité avec un lance-roquette sur l'épaule... C'est pas gagné, même si la solution d'un point de vue technique est bonne. Le code du client SDC est libre, open-source. Tout le monde peut le regarder, ce qui permet aussi à tout le monde de trouver des failles de sécurité. Ce sera bien entendu le point le plus sensible, même s'il n'y a pas d'autres solutions pour l'instant. Enfin pour le 5ème point, le support de Java et de Groovy. J'ai discuté ensuite avec [Guillaume Laforge](#), qui m'a expliqué le travail réalisé avec les équipes de Google, situées à Atlanta. Avec l'aide des équipes de Google AppEngine, les développeurs de Groovy ont adapté la dernière version afin qu'elle fonctionne sur l'AppEngine. La difficulté, comme Guillaume l'explique, c'est que Groovy étant un langage dynamique, il a besoin de certains droits avec le SecurityManager afin d'instrumenter le code. Par contre il semble que le support de Grails ne sera pas possible, vu la complexité et les efforts à fournir. Je pense que Guillaume pourra expliquer mieux que moi les détails techniques. Didier a ensuite fait plusieurs démonstrations. Google AppEngine vous propose un plugin pour Eclipse, afin de démarrer rapidement l'écriture de sa première application. Je pense à Maven2 dans ma tête quand [Nicolas de Loof](#) précise qu'un plugin pour l'appEngine est dans son sac à dos et devrait être disponible d'ici quelques jours. Nicolas propose un petit plugin maven afin de faciliter le déploiement d'une application à partir de maven. Allez sur [son blog](#), hop, et revenez après j'ai pas terminé. J'ai vu que les équipes de SFEIR ont sacrément bossé sur un moteur de Blog écrit en Java, hébergé sur Google AppEngine. L'idée est de basculer les blogs de SFEIR vers cette plateforme si j'ai bien compris. Le projet est open-source, à quand un Touilleur Express sur l'AppEngine ? Si j'avais un peu de temps, mais pour l'instant je préfère attendre encore un peu. Peut-être d'ici 6 mois. **Est-ce que Google AppEngine est pour vous ?** Oui définitivement si votre service Web doit gérer une charge fluctuante de visites, si vous avez développé un Gadget pour Facebook ou une application Open-Social par exemple. C'est un moyen de déployer sur une grosse machine énorme votre application, sans vous soucier de devoir réinstaller un jour votre machine. Vous n'aurez plus à gérer la sécurité, la mise à jour, les patches, les sauvegardes... Zéro administration système. J'ai pour l'instant un peu de mal à imaginer une application d'entreprise type extranet, ou une application CRM. Comment s'interfacier avec un Siebel ou un SAP ? Nous avons ce connecteur SDC pour connecter notre SI à Google AppEngine. Mais avouez que pour convaincre l'équipe sécurité... C'est pas gagné. **Les services Google** La plateforme AppEngine offre des services tels que l'authentification ou l'envoi des emails. La page "[Services API](#)" donne la liste des services que nous pouvons utiliser dans nos applications Java déployées sur Google AppEngine: \* Memcache est un cache mémoire simple et partagé \* URL Fetch permet d'utiliser java.net.URLConnection pour parler avec d'autres applications, HTTP et HTTPS \* Mail permet d'envoyer des emails avec l'api Java Mail \* Images est un service pour manipuler des photos ou des images : retailler, améliorer, agrandir, basculer etc. \* Google Accounts permet d'authentifier les utilisateurs avec le système de Google. Cela vous décharge de la gestion des users, des mots de passes. Il y a un énorme écosystème autour de Google : Calendar, Blogs, News, Recherche, Traduction, Paiement, Photos... j'attends maintenant que Google nous offre des services plus intégrés. Certes nous pouvons tous prendre la doc de chacune de ces APIs puis commencer à coder. Mais rêvons un peu, et imaginons un peu de code qui permet de prendre rendez-vous chez votre Coiffeur en prépayant par carte bleu votre séance...

```
public class TouilleurHairCutServlet extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws IOException {

        GoogleCalendarService calendarService = GoogleRepository.getCalendarService();

        Rendezvous rd=RendezvousHelper.fetch(req);

        calendarService.createEvent(rd.getDate(),rd.getSubject());

        //.....

        GoogleCheckoutService checkoutService=GoogleRepository.getCheckoutService();
        try{
            checkoutService.bill(req);
        }catch(GoogleCheckoutServiceException e) {
            // cancel event in calendar and say bye
            .....
        }

        // Send an email with Google Calendar event, Google Map for address...
        ...
        ...

        // Fin
        resp.setContentType("text/plain");
        resp.getWriter().println("Votre rendez-vous est confirmé et payé... ");
    }
}
```

Rêvons un peu... une application Web où vous pourriez consommer facilement différents services Google. Je pense plus particulièrement à tout ce qui est OpenSocial afin de créer des gadgets pour HiFive, Salesforce.com, LinkedIn.com ou MySpaces... J'écris ce code (bien timbré il faut le reconnaître) et je repense à ce que disait Nicolas : où est Google Guice ? A priori rien ne nous empêche d'utiliser Guice, Spring ou autre. Il y a cependant un dernier point, et après je vous laisse : toutes les APIs de Java ne sont pas supportées. Et bien oui... pas possible de créer une Thread ou

de vous balader sur le filesystem... qui n'existe pas au sens classique. Il va donc y avoir un sacré effort pour lister les frameworks supportés et ceux non supportés, ce qui va alimenter la blogosphère rapidement. C'est donc un facteur important à prendre en compte : tout n'est pas supporté, car comme toute application hébergée sur une plateforme de type PaaS, vous êtes sur un système virtuel. Oubliez l'idée de créer des fichiers plats, des sockets. Utilisez URL Fetch ou BigTable. Il y a un sujet d'architecture intéressant. Si j'étais didier, je profiterai bien de l'[USI](#) pour parler un peu du type d'application que l'on peut faire avec Google AppEngine. Merci à Google et à SFEIR pour l'organisation de la soirée, et à bientôt.

## Compte-rendu de la soirée Scrum au Paris JUG



Avec 190 inscrits, encore une fois le Paris JUG a fait salle comble. Hier soir grâce à Eric "Bob" Mignot nous avons vu comment fonctionne Scrum. Je ne vais pas me lancer dans un résumé de [notre présentation](#), car Olivier Croisier a posté un très bon résumé ce matin sur son blog, [The Coder's Breakfast](#). La deuxième partie de la soirée a été animée par Guillaume Bodet, Directeur Technique de Xebia. L'objectif de la présentation est de répondre aux questions souvent posées par les clients que Guillaume rencontre, lors de la mise en place de Scrum. Le plan de la présentation : - Trouver le product owner - D'où vient le product backlog ? - Et mon planning ? - Mon projet est trop gros ! - Et l'architecture alors ? Dans un premier temps, la difficulté est de trouver le Product Owner. Rappel de la théorie : il est le propriétaire du produit, il doit porter la vision du produit, il est en charge du ROI, du budget et du planning de livraison. Guillaume présente le meilleur Product Owner au monde : **Steve Jobs** ! Mais comment faire lorsqu'on a pas un Steve Jobs sous la main ? Tout d'abord il n'est pas forcément seul. Il doit par contre être responsable de tout et avoir une vision de ce que le projet ou le produit doit être. Guillaume liste dans un slide une liste des rôles existants dans les structures qui font de bons product owner : - Chez les Editeurs de Logiciels -> Responsable Produit - Web -> Le Directeur Marketing - Banque -> Chef de projet utilisateurs - Industrie -> AMOA + users groups Guillaume Bodet aborde ensuite le Product Backlog. D'où vient ce PB ? Selon la littérature Agile, il émerge un peu par magie, sans finalement donner des repères assez solides pour commencer. Par son expérience, il explique que le démarrage d'un projet peut se faire avec un document d'une dizaine de pages, avec des critères d'estimation de la taille du projet, une présentation du ROI, l'évaluation du risque et la définition d'une équipe. Lorsqu'il explique cela, je pense alors à un Business Plan. C'est clairement un document synthétique qui donne la vision d'un projet d'entreprise, qui présente la partie budget, les équipes, sans chercher à définir en détails le produit. Il explique ensuite que les équipes de Xebia mettent en place des spécifications exécutables (TDD, TDR) afin d'aider le client à formaliser ses demandes. Contrairement à une idée reçue, Scrum ne supprime pas les documents de spécifications techniques ou fonctionnelles. Simplement, on cherche à être moins exhaustif sur les points qui ne seront pas développés en premier. Le planning est important. Nous devons rendre compte de l'avancement. Il nous explique tout d'abord ce que les chefs de projets font en principe avec Microsoft Project et les diagrammes de Gantt. L'objectif d'un [diagramme de Gantt](#) est d'effectuer un ordonnancement des tâches, afin de communiquer un planning. C'est ensuite un outil qui permet d'assurer le suivi de l'avancement. Il parle d'un dérapage souvent rencontré en conduite de projet : - vous définissez avec beaucoup d'efforts un planning type diagramme Gantt - chaque jour vous collectez le temps passé par chacun des développeurs. Problème : que se passe-t-il lorsque le temps estimé déborde ? vous devez alors réadapter le diagramme afin de faire "rentrer" cette mise à jour - vous perdez du temps pendant une heure à remettre à jour ce diagramme, pendant ce temps là vous ne faites rien de productif - ensuite vous êtes débordé, donc un assistant chef de projet se charge de mettre à jour ce machin. Bravo vous faites preuve de délégation. Bref le planning classique est parfois une souffrance, voire une grosse perte de temps. A cela, l'Agile répond : on ne fait pas de planning, ce qui n'est pas forcément mieux. Guillaume explique que scrum propose de faire des plannings de release, afin de s'assurer que le développement d'un produit ou d'un logiciel suit une roadmap, respecte la vision du



product owner. Guillaume parle aussi parfois de nous, praticiens de Scrum, qui tentent de vendre le Burndown Chart comme une prévision de

l'avenir... ce qui n'est pas le cas. Il explique que c'est un outil destiné avant tout à faciliter la communication entre le Product Owner et l'équipe, ni plus ni moins. Pour terminer, j'ai apprécié sa vision de la planification. Il y a 5 niveaux de planification : - au niveau quotidien, défini lors du stand-up meeting du matin - au niveau de l'itération (sprint planning) - au niveau de la release (release planning) - au niveau de la vision du produit (roadmap) - au niveau de la vision annuelle. Il rappelle qu'un plan fixe des objectifs, la planification permet ensuite d'adapter ce plan à la vraie vie. Pour preuves, il parle de la démarche d'[Eisenhower](#), général durant la 2ème guerre mondiale et 34ème président des Etats-unis qui a juste réussi le plus gros projet de notre histoire : [Opération Overlord](#). Les estimations et la planification : c'est un sujet qui me passionne. J'ai bien aimé l'évocation de Cocomo II mais bon, il y a d'autres méthodes moins hype, je vous prépare un billet sur ce que je suis entrain de faire avec mon équipe en ce moment. J'en profite pour vous proposer de relire un billet de décembre dernier : [Devoxx les estimations de temps](#). Ensuite Guillaume parle du cas des projets qui ne pourraient pas faire de Scrum car "ils sont trop gros". Il partage tout d'abord l'expérience du Projet ProRail PUB, la refonte du SI du système des chemins de fer hollandais. Avec un budget supérieur au million, 25 personnes ont réalisé en 8 mois une première version. La v1 était opérationnelle. De nouvelles versions continues à être mise en production tous les 3 mois. Les gros projets se divisent en plusieurs équipes. La mise en place de Scrum of Scrum permet de coordonner le développement d'une plateforme. C'est donc possible, en conservant une taille respectable d'équipe (<15 personnes). L'architecture ensuite est importante. Il casse le mythe des Agilistes qui vous disent que "l'architecture émerge" tel un iceberg... C'est faux. C'est au prix d'un travail d'analyse, de spécifications, d'études de faisabilité que les projets se construisent une architecture. Il a une pensée émue aussi pour les équipes d'Architecture transverse qui sont parfois déconnectées des équipes de terrain. Un architecte internet doit faire partie d'une équipe Scrum. L'architecture se pilote par le besoin, comme dit Scott Ambler. La modélisation doit se faire à plusieurs, il ne faut pas laisser à une seule personne le soin de décider d'une architecture : c'est bien trop dangereux. Enfin il rappelle l'importance de rester simple (et humble) et de penser qu'il est plus important de livrer un produit qui fonctionne, qu'une documentation exhaustive qui ne fait pas gagner d'argent. En conclusion, j'ai trouvé que la présentation de Guillaume complétait bien la première partie. Merci à l'équipe du Paris JUG de nous avoir donné l'occasion de faire 2 heures sur Scrum et les méthodes Agile ! **Annonces et informations** En début de soirée, Tanguy Bayard a fait une présentation de 10 minutes vraiment sympa. Il a présenté le principe du Paris JUG à l'équipe de direction de [SFEIR](#). Grâce à lui, SFEIR est devenu sponsor annuel du Paris JUG ! Franchement bravo et merci à SFEIR. [Oxiane](#) est aussi un nouveau sponsor, vous pouvez lire leur blog très sympa, merci à Gabriel Kastenbaum. Enfin Antonio a parlé du podcast "[Les Cast Codeurs](#)" dont je vous parlerai dans un autre billet. :-)

**The 3ème mi-temps** Enfin en 3ème partie de soirée, entouré de Florent Ramière, de David Gagot, de Jean-Michel Bea, de Sébastien Douche, de Cyrille Leclerc et de Thomas nous avons pas mal discuté de Google App Engine. Je pense qu'il y a beaucoup de restrictions et que de parler de Java alors qu'il y a tant de limitations, c'est un point important à ne pas négliger. David explique qu'en même temps nous parlons d'une version bêta de ce support. Qu'avec le temps d'ici 6 mois nous aurons aussi apprivoisé ce nouveau système. Je demande à Florent si pour SpringFuse il serait prêt à basculer d'Amazon EC2 à Google App Engine (GAE). A vue de nez oui, maintenant en perdant une base de données ou un système de fichier, cela fonctionnera en utilisant le moteur BigTable. Il va y réfléchir, même si ce serait plus pour tester que pour vraiment s'en servir. David et Florent parlent ensuite de Mockito versus EasyMock, après une discussion sur [Citcon](#) très intéressante. C'est sans doute la bonne bière qui m'a aussi convaincu de m'inscrire pour mi septembre ! Avec Thomas et Florent nous parlons aussi pas mal sur Google et le dernier Barcamp. Florent parle aussi des premiers retours des clients de SpringFuse, avec un démarrage très fort sur un gros projet entre autre. Enfin voilà, fin de la 14ème soirée du Paris JUG. L'occasion encore de discuter entre geeks et passionnés, de voir que presque 200 personnes sont venues nous écouter présenter Scrum avec Eric. Eric, merci à toi d'être venu. Départ le lendemain pour Bordeaux, tu étais la semaine dernière au Canada, je sais que tu bouges beaucoup mais merci d'être passé nous parler de Scrum. Je vous prépare un podcast avec l'enregistrement de la soirée avec mon histoire de machine à laver ;-)

## Les CastCodeurs : un podcast en français

Vive [Twitter](#). Mardi matin j'ai vu la nouvelle passer en quelques heures, la sortie d'un podcast en français sur Java : [les Cast Codeurs](#). Nos voisins anglophones ont le podcast [JavaPosse](#) qui est une petite perle à écouter le matin dans le RER, pour être tenu informé des dernières nouvelles. Les Javaposse's sont 4 geeks bien allumés et bien sympathiques, que j'ai rencontré à Devoxx en



décembre dernier: [Les CastCodeurs](#) c'est un podcast animé par Emmanuel Bernard, Guillaume Laforge, Vincent Massol et Antonio Goncalves. Avec au programme une revue de presse, puis une partie un peu plus technique sur Java appelée "les mains dans le cambouis", l'objectif est de parler en français de l'actualité de la plateforme Java et des langages apparentés comme Groovy par exemple. Pour avoir écouté deux fois le podcast, je le trouve franchement bien réalisé. Gros boulot de montage derrière, post-prod sympa, sujets brûlants et intéressants. Un petit bémol pour la partie code, qui est difficile à faire passer à la radio. Côté longueur : parfait ! une heure c'est très bien. Longue vie aux CastCodeurs, et on attend le prochain

épisode ! je sais que Guillaume est à Seattle cette semaine, qu'Antonio bosse sur son prochain livre Java EE 6 aussi, gros boulot en plus donc pour tout le monde. Le site: <http://lescastcodeurs.com/>

## The Normandy JUG



# NormandyJUG

Et hop,

15ème Java User Group à Rouen : [The Normandy JUG](#) ! Ils ont besoin de sponsors, et je connais déjà des normands qui pourront participer aux soirées comme Dimitri Baeli d'eXo Platform par exemple. Au delà de l'ouverture d'un Java User Group, il y a un phénomène impressionnant : la création de lien social entre les développeurs Java. Il existe depuis peut-être 10 ans d'excellents sites communautaires comme [Developpez.com](#) qui nous permettent déjà de communiquer. Mais depuis que j'ai découvert le Paris Java User Group, j'ai vraiment l'impression de faire partie d'une communauté d'utilisateurs, de développeurs, de passionnés. C'est donc à chaque fois avec plaisir que j'apprends la naissance d'un nouveau JUG en France. L'équipe du Paris JUG peut déjà réserver Bercy pour l'année prochaine pour le deuxième anniversaire du Paris JUG. Longue vie au NormandyJUG ! Si les JUG veulent un peu de pub, envoyez-moi un email et je ferai suivre sur le Touilleur Express avec plaisir.

## Jelly Party : partager son appart pour travailler à plusieurs

[caption id="attachment\_1152" width="300" caption="Official Jelly Logo"]



[/caption] Et si au lieu de travailler tout seul à la

**maison, vous partagiez votre appartement et votre connexion Wifi de temps en temps ? Nouveau phénomène de co-working, qui permet de partager son espace de travail, la Jelly Party est aussi le moyen de rompre la monotonie et de rencontrer de nouvelles personnes. En France, le phénomène se transforme même en JellyCamp, comme l'explique Philippe Antoine dans la deuxième partie.**

Lors du dernier Barcamp, Philippe Antoine m'a expliqué le principe des **Jelly Party**. Nouveau phénomène social, voici en quelques mots la définition et le principe : une Jelly est un regroupement de travailleurs afin de partager dans un appartement une journée de travail. C'est au départ donc pour les indépendants et les freelances, afin de travailler sur son projet chez un ami, le temps d'une journée. Le principe est le suivant : prenez un groupe de 5 à 10 personnes, travaillant avec un ordinateur portable. Vous les invitez chez vous, vous fournissez l'accès à Internet gratuitement, ainsi que quelques boissons et l'électricité. L'idée n'est pas de travailler forcément avec votre voisin, plutôt de passer une journée de travail tout en rencontrant d'autres travailleurs, d'autres entrepreneurs, des artistes, des écrivains, des traducteurs, des graphistes, etc. Bref toutes les professions liant l'informatique où en général vous travaillez tout seul à la maison. Lorsque l'on travaille seul, pour arrêter de parler au frigidaire de votre pire client, et s'aérer le cerveau, c'est un moyen sympa. Vous pouvez vous arrêter prendre un café dans un petit bistro, comme l'expliquait Antonio Goncalves, mais rien ne vaut finalement une journée de temps en temps avec des personnes que vous ne connaissez pas, mais qui travaillent comme vous, chez eux. L'idée de départ date de 2006, à New-York. [Amit Gupta](#) est l'un des inventeurs de ce principe tout simple. Son organisation est vraiment intéressante : la cuisine est l'endroit stratégique pour discuter et faire une pause. Dans le couloir, des post-its permettent à chacun de laisser des informations ou des petits mots. La chambre permet de passer des coups de fil privés. Le salon est organisé pour que chacun puisse poser son ordinateur portable et disposer d'une prise électrique.



En général, chacun vient avec une ou deux connaissances.

Tout le monde ne se connaît pas, ce qui permet aussi de rencontrer de nouvelles têtes. Sortir de chez soi permet aussi de travailler plus efficacement. Vous n'êtes pas tenté de jouer avec le chat, de piquer une petite sieste ou de faire du ménage. Vous vous sentirez sans doute plus motivé, en étant avec d'autres entrepreneurs. En France nous pouvons citer aussi l'exemple de [LaCantine](#), qui est un lieu adapté pour recevoir des travailleurs indépendants, à la journée ou plus. Philippe Antoine et Luc Bizeul ont organisé la première JellyParty à Paris [vendredi 17 avril](#). Cela a donné l'idée à Philippe de tenter l'aventure de la Jelly Party sous une forme un peu différente. Je l'ai interviewé afin qu'il me donne son point de vue sur le sujet. **Nicolas : Philippe, comment as-tu entendu**



**parler des Jelly Party ?** Philippe : Je cherchais un format en fait afin de proposer à plusieurs personnes de se regrouper pour travailler, mais surtout pour échanger et discuter sur des sujets techniques. **N: Vous avez fait une première Jelly début avril à Paris, comment cela s'est-il passé ?** P: Nous avons lancé l'invitation via Twitter et le bouche à oreille. J'ai tenu la première JellyParty chez moi. Les gens sont arrivés vers 18h00, nous avons passé une bonne soirée à discuter. Les échanges étaient vraiment intéressants ([photo de la soirée](#)). **N: Plutôt que la partie travail, c'était donc les échanges et les rencontres alors ?** P: Finalement oui. Est-ce que cela reste dans l'esprit de la Jelly ? Je pense que oui. Pour moi le premier principe c'est que l'événement se passe chez quelqu'un. Il y a la notion d'hôte qui reçoit. Ensuite nous n'avons pas fixé d'ordre du jour. L'idée comme un barcamp : venez et nous allons parler. De quoi ? je n'en sais rien. Mais venez, avec votre ordinateur, et nous passerons un bon moment. **N: Si je mets un peu d'esprit du Touilleur dans tes parties, il y a un côté réunion tupperware alors ?** P: Oui un peu de ça \*rires\*. C'est donc pour moi l'occasion de discuter sur des sujets qui nous passionnent, de créer des réseaux entre les gens, et d'échanger plus qu'à un BarCamp. **N: Côté organisation, que conseilles-tu et comment s'y prendre ?** P: D'un point de vue communication, il faut annoncer la soirée. Je trouve que Twitter est l'idéal : tu ne connais pas forcément tes followers, mais eux seraient intéressés pour discuter avec toi. En nombre, 12 personnes me semble bien. Après si tu as un grand appartement, tu fais comme tu veux. Ensuite nous pouvons demander à chacun d'amener de quoi manger, un ordinateur, et une idée de sujet. Enfin pour la partie Wifi j'avais laissé le mot de passe afin que chacun puisse se connecter facilement. **N: Pour terminer, quels sont les 3 tags pour te définir ?** P: Je dirai HTML 5, TDD et les Tribus **N: Merci philippe, rendez-vous au prochain JellyCamp / Jelly Party alors !** P: Merci Nicolas. A bientôt ! **Références:** [Philippe ANTOINE sur Twitter](#) [Article sur Wired Site officiel d'Amit Gupta Coworking at NYC](#) [Liste des Jelly dans le monde](#)

## Oracle a racheté SUN Microsystems

Oracle a annoncé le rachat de SUN Microsystems pour quelques 7 milliards de dollars. Après le loupé il y a quelques semaines entre IBM et SUN, cette fois-ci l'opération a réussi. Dès 2009 cette acquisition rajoutera 15 cents par action aux résultats du nouveau groupe, simplement par l'effet du chiffre d'affaire de SUN. La blogosphère, ma boîte Gmail, Twitter, tout le monde ne parle que de ça... Encore une fois Twitter a été pour moi le premier moyen d'être informé, suivi par la mailing list JavaPosse toujours au courant de ce genre de rumeurs. Bon mon avis sur la question: attendons quelques jours avant de crier au loup. Les avis sont partagés, certains prédisent la mort de MySQL, d'autres de la JVM de SUN qui ferait doublon avec "autre-jvm" d'oracle... Bien malin celui qui peut parler au nom d'Oracle... Projets secrets ? Rapprochement programmé ? A suivre

## Kalistick propose une solution d'analyse de la qualité du code en mode SaaS

J'ai été contacté par les fondateurs de [Kalistick](#) il y a quelques semaines. Comme vous le savez, j'aime bien tester et parler ensuite de nouveaux logiciels, comme [SpringFuse](#) récemment. Je prends le temps de vous en parler afin d'être transparent : je fais cela sur mon temps libre sans aucune contre-parties, à part une bière payée par Nicolas Romanetti de SpringFuse le 10 mars 2009 à 23h34 après la soirée du Paris JUG (!!!). Pour être tout à fait sérieux, je tenais à préciser tout cela, soyons plus concret et parlons maintenant de Kalistick. L'éditeur Kalistick est une société française basée à Lyon, qui édite un logiciel d'analyse de code pour Java et C# en mode hébergé (Software as a Service). L'intérêt de la solution est donc de nous permettre d'utiliser un moteur d'analyse déporté, toujours à jour, disponible, et capable aussi de construire à terme une base de "bonnes pratiques". En envoyant votre code, vous contribuez aussi à améliorer l'outil, en proposant de vrais cas, ce que je trouve vraiment intéressant et novateur. Le fonctionnement est simple : tout d'abord

nous préparons en local une archive avec le code source, les bibliothèques et le code compilé. Après avoir téléchargé vers Kalistick cet ensemble, l'analyse commence. Vous recevez alors par email un message vous informant que l'analyse est terminée. Les tableaux de bords, les agents d'analyse et les explications sur l'analyse permettent ensuite d'identifier rapidement les points noirs de l'application. Vous pouvez renouveler l'analyse, et suivre ainsi l'avancement des améliorations. Du côté de la concurrence, nous trouvons [Insite SaaS](#) de l'éditeur Metrixware. C'est l'outil utilisé par mon client actuel, BNP Paribas. Je n'ai pas eu l'occasion de m'en servir, mais il est intéressant de constater que les éditeurs suivent aussi la tendance du SaaS tout comme "[Application Intelligence Platform](#)" de Cast. Chacunes de ces solutions se différencient sur les langages supportés et sur le type de règles. Pour ma part c'est Java, nous allons voir ce que je peux apprendre. J'ai pris du vieux code de notre application, qui présente peu de dépendances. Pour tout vous dire, c'est du code qui date de 2003 ou 2004, dans lequel on trouve tout un tas de fonctions plus ou moins utiles. J'ai donc l'impression d'emmener un vieux chien de 14 ans chez le véto lorsque je commence l'analyse... La première partie consiste à préparer une archive au format tar.gz ou zip avec le code source, les bibliothèques et les jars compilés. La préparation de l'archive se fait à l'aide d'un Agent proposé avec Kalistick. Pour le code C#, [un Add-In pour Visual Studio](#) permet aussi de préparer son projet. J'ai fait analyser environ 31000 lignes de codes Java, soit 244 classes, de qualité franchement pas top afin de voir ce que cela allait donner. L'analyse terminée, vous recevez un email vous informant que le rapport est prêt. Je me connecte sur le portail de Kalistick. L'interface est vraiment très bien faite. Je reconnais Ext-JS, l'organisation est claire et simple. Franchement, on voit l'outil fait par de vrais développeurs qui ont travaillé en SSII sur le terrain : les informations les plus importantes sont affichées dès la page de synthèse. J'ai été assez bluffé par la qualité de l'analyse. Les metrics sont groupées par domaine, afin de vous aider à regarder les domaines les plus importants pour vous : Efficacité, Evolutivité, Fiabilité, Maintenabilité, Sécurité et Transferabilité. Pour moi sur ce code, c'est tout d'abord la fiabilité. 27% du code analysé présente des problèmes dans le domaine de la Fiabilité, ce qui revient à dire qu'un quart de ce que j'ai utilisé... n'est pas fiable. Nous allons voir un exemple trouvé par Kalistick, et tester au passage si vous êtes observateur ou pas :-)

Prenons ce bout de code, digne des années 2000 où l'on portait des pantalons en patte d'éléphant. A votre avis, que se passe-t-il à la ligne 96 ?

```

90 public static Properties filter(Properties source, ArrayList includingKeys) {
91     if (includingKeys != null) {
92         return new Properties(source);
93     } else {
94         final Properties properties = new Properties();
95
96         for (Iterator iterator = includingKeys.iterator(); iterator.hasNext();) {
97             Object o = iterator.next();
98
99             if (source.containsKey(o)) {
100                 properties.put(o, source.get(o));
101             }
102         }
103         return properties;
104     }
105 }

```

Je vous passe les mauvaises pratiques, on cherche un bug... Regardez la variable *includingKeys* et vous verrez le souci. Kalistick trouve aussi des problèmes plus compliqués, comme le problème de comparaison des Float lorsque l'on tombe sur un cas NaN (not a number). En effet, une division par 0 retourne un NaN en Java, qui n'est pas inférieur à 0, même pas comparable à quoique ce soit. Il faut donc utiliser `Double.isNaN()` pour déterminer si un Float est un NaN. A cet instant précis je sens un grand vide dans votre regard. Ce code n'est pas bon ([tiré de cet article](#)) si vous passez en argument NaN car NaN n'est pas inférieur à 0, il n'est même pas égal à lui-même pour un float.

```

public employee(float monthly_salary){
    if (monthly_salary < 0.0)
        throw new IllegalArgumentException("illegal monthly salary");
    this.yearly_salary = 12.0 * monthly_salary;
}

```

Celui-ci est correct, si je passe un salaire incorrect égal à NaN, une exception sera bien levée.

```

public employee(float monthly_salary)
{
    if (monthly_salary >= 0.0) {
        this.yearly_salary = 12.0 * monthly_salary;
    }
    else throw new IllegalArgumentException("illegal monthly salary");
}

```

Cet exemple n'est pas le plus intéressant. Comme Emmanuel Bernard d'ailleurs le rappelle dans le podast des [LesCastCodeurs](#), dès que vous travaillez avec des doubles ou des floats, il faut redoubler d'attention. D'où l'idée d'utiliser Kalistick pour éviter de se tromper... ce qui arrivera tôt ou tard. Jugez ce code par exemple (tiré de [cet article](#) de Kasper B. Graversen référencé sur [lesCastCodeurs](#)) :

```

double nan = 0.0 / 0.0, infinity = 1.0 / 0.0;
System.out.println( Double.MAX_VALUE != infinity );
System.out.println( Double.MAX_VALUE != nan );
System.out.println( infinity != nan );

>> true
>> true
>> true

```

D'autres metrics comme le problème de l'équilibre `hashCode()/equals()` sont expliqués très

clairement par Kalistick :

Si une des deux méthodes equals() ou hashCode() doit être surchargée, il est nécessaire de surcharger l'autre. Ces méthodes visent à comparer le contenu de deux objets. Les spécifications Java exigent que ces deux méthodes soient surchargées ensemble pour garantir le bon fonctionnement de mécanismes tels que les collections basées sur les hash (HashMap, Hashtable, ...). \* <http://www.javapractices.com/Topic17.cjp> \* <http://www.javapractices.com/Topic28.cjp> (extrait documentation en ligne Kalistick)

Parmi d'autres points notés par Kalistick : - Quelques soucis de création de Singleton qui ne sont pas Thread-Safe. Comme on le sait, la création en "lazy" des Singletons n'est pas thread-safe, ce qui peut donner de beaux bugs difficiles à trouver ([voir cet article sur OnJava.com](#)) - Problème sur les DateFormatter qui ne sont pas Thread-Safe, ce que l'on ne sait pas toujours ([voir cet article](#)) - et pleins d'autres... de quoi faire 3 pages sans problèmes En conclusion pour la partie analyse, le moteur est puissant et la qualité des explications, en français, fait gagner un temps précieux. J'ai aussi trouvé pratique une fonction toute bête : la possibilité de désactiver facilement certaines alertes. Par exemple concernant la Javadoc, sur cette partie de code je n'ai pas envie de voir les problèmes de documentation. D'un clic, je peux ajouter ou exclure les règles, ce qui est pratique pour ne conserver que ce qui est vraiment important. Là où un autre outil joue l'esbrouffe avec 724 440 metrics dont 80% ne servent pas, Kalistick est pragmatique, et donc bien plus efficace. Je ne vous cache pas qu'après avoir récupéré mon diagnostic, j'ai eu l'envie de commencer à corriger le code, et à me lancer dans du nettoyage en règle. Contrat rempli donc. La valeur ajoutée se trouve dans les cas que l'on ne peut pas détecter par l'analyse statique du code source. L'instrumentation du bytecode est un moyen de trouver plus de problèmes, ce que sait faire Kalistick. Son concurrent en terme de metrics est semble-t-il Findbugs. Mais le produit va plus loin, car n'oublions pas le Service. Kalistick propose ensuite une partie Plan d'Action qui vous aide à corriger et à reprendre le code. C'est donc le médecin qui détecte de quoi vous souffrez, mais qui vous propose ensuite de vous aider à corriger les problèmes. C'est une approche plus intéressante que l'approche des outils open-source, qui sont bons pour vous dire de quoi vous souffrez, pas forcément bon pour vous expliquer comment ensuite résoudre ces problèmes. Côté roadmap, la prochaine version est prévue pour le mois de juin : - gestion des tags pour lister, filtrer, rechercher... - mesure de l'effort fourni (charts sur le nombre de violations corrigées, nouvelles, ...) - nouvelle règle qui croise le couplage afférent et la complexité cyclomatique Puis un peu plus tard dans l'année: - nouvelles règles d'implémentation - nouvelles règles de structure (avec le couplage afférent) - règles de design (type étanchéité des packages) - antipatterns Voilà, test terminé pour moi. Je pourrai encore en parler pas mal, mais je vous laisse car je vais faire un test tordu : prendre le code généré par [SpringFuse](#) et le faire auditer par [Kalistick](#)... Hop à bientôt. **Références** Le site de Kalistick <http://www.kalistick.fr> Problème des Floats et NaN <http://www.concentric.net/~Twang/tech/javafloat.htm>

## Envoyer des messages vers Twitter avec Log4J

Petit délire de ce week-end : coder un Appender pour Log4J afin de publier vers votre compte Twitter des messages de log4j. Bref quelques heures plus tard, le temps de faire un projet propre, je vous ai mis le tout en ligne il y a quelques minutes. Vous trouverez en anglais des explications complètes [sur la page dédiée](#), le code source, les ressources et un exemple d'utilisation. Le principe est simple : après avoir configuré un nouvel appender, puis un logger générique, vous pouvez alors envoyer des messages de log4j avec du code classique de votre application:

```
// The TWITTER Logger is defined in the Log4j.xml
Logger twitterLogger=Logger.getLogger("TWITTER");

// This message won't be published cause we configured
// the appender to ERROR Level
twitterLogger.debug("This message should not be published");

// This message will be published
// Using standard Twitter syntax, you'll send me
// a small helloWorld !
twitterLogger.error("Hello @Letouilleur ! it works !");
```

Vous devez simplement configurer un appender "TwitterAppender" soit dans votre fichier log4j.xml, soit dans un fichier log4j.properties ou directement en Java. Dans cet exemple, nous configurons un appender avec un compte Twitter valide et un mot de passe : log4j.xml :

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "/Users/nicolas/Dev/TwitterAppender/src/test/resources/configuration/log4j.dtd">

<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/"
    debug="true">
    ...
    ...
    <appender name="TWITTER_APPENDER" class="com.letouilleur.express.TwitterAppender">
        <!-- true or false, I recommend true to avoid locking your app while we publish to twitter-->
        <param name="useAsynchronous" value="true"/>
        <!-- Set here a valid twitter userId screenname that we will use to authenticate -->
        <param name="screenId" value="YOUR TWITTER USERID HERE"/>
        <!-- Set here its password -->
        <param name="screenPassword" value="YOUR TWITTER PASSWORD HERE"/>
        <!-- If you want to test and you do not want to really publish messages, set this to true -->
        <param name="fakePublishing" value="false"/>
    </appender>
```

Nous configurons ensuite un logger "TWITTER. A noter que le nom n'a aucuns importances, vous pouvez même ajouter une référence à l'appender TwitterAppender à des loggers existants dans votre application, et la rendre ainsi capable de "twitter" !

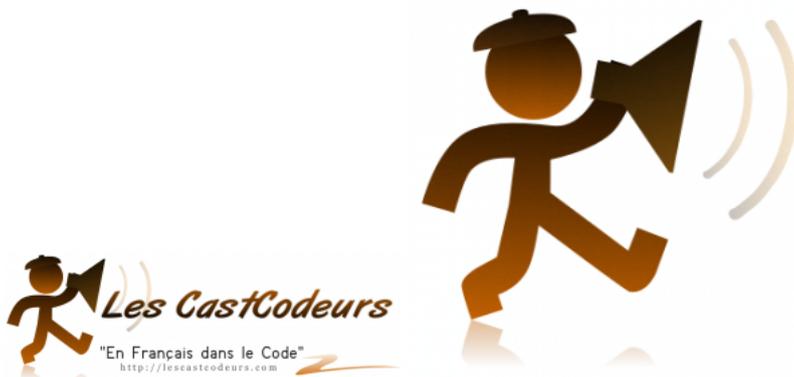
```
<!-- Configure a TWITTER logger that will be use from my sample class TwitterAppenderTestLog4jXML -->
<!-- Any message with level ERROR that is send to Logger "TWITTER" will be published -->
<logger name="TWITTER">
  <level value="ERROR"/>
  <appender-ref ref="TWITTER_APPENDER"/>
</logger>

...
... // other configuration params...
...
</log4j:configuration>
```

**Et à quoi cela sert-il ?** Ce système permet à votre application de publier son status. L'idée que j'ai serait de créer un compte dédié pour votre application. Appelons par exemple "primeweb" ce compte Twitter. Nous pourrions alors imaginer que notre système publie son status sur sa page Twitter. A 06h00 du matin "primeweb" génère des rapports financiers. Il pourrait ainsi annoncer fièrement "345 rapports générés, aucunes erreurs signalées" par exemple... J'imagine aussi qu'en retour nous pourrions répondre et twitter avec notre compte robot. Je pourrai par exemple envoyer un message direct à "primeweb" en écrivant "Salut mon gars, tu peux #relancer\_batch\_asie\_as\_of\_20090404 stp ? merci et ensuite #notifier\_Letouilleur\_si\_erreurs". Bref un système simple pour discuter avec votre logiciel... Je suis certain que ce bon vieux primeweb rêve de twitter comme un fou et de nous raconter tout ce qu'il fait. Primeweb c'est le projet sur lequel je travaille en ce moment à la GrandeBanqueFrançaiseQueJePeuxPasCiter... Rendez-vous sur la page "[TwitterAppender](#)" pour tester. Le code est en license BSD, libre, avec les sources. J'utilise l'excellente API [Twitter4J](#) pour envoyer mes twits.

## Logo les CastCodeurs

Voici les 2 logos que j'ai fait pour le podcast sur Java en Français "[Les CastCodeurs](#)" d'Emmanuel Bernard, Guillaume Laforge, Antonion Goncalves et Vincent Massol :



## Première soirée du Ch'ti JUG à Lille le 13 mai

J'ai le plaisir de vous faire suivre le message de l'équipe du Ch'ti JUG : "Nous avons le plaisir de vous annoncer que la première session du Ch'ti JUG est prévue le 13 mai. Elle aura lieu à 18h30 aux Salons du Pont Neuf à Lille ([cliquer ici pour un plan d'accès](#)) Cette session est rendue possible grâce à la société [ProxiAD Nord](#). Le thème de cette session sera : "**Java EE 6, qu'est-ce qui nous attend ?**" J2EE 1.4, Java EE 5 et bientôt Java EE 6. L'année 2009 coïncide avec la sortie de la nouvelle version de l'Enterprise Edition et de son 10e anniversaire. Mais le J2EE de l'époque ressemble-t-il encore au Java EE d'aujourd'hui? Dans cette session, Antonio Goncalves vous présentera les nouveautés Java EE 6 : un modèle de programmation centré sur les EJBs, un packaging simplifié, une couche de persistance enrichie, une nouvelle API pour les ressources REST, une simplification radicale de la couche web, et bien d'autres améliorations qui feront de Java EE 6 la plateforme idéale pour vos applications. Basé sur des standards, s'inspirant de frameworks open source, utilisant les conteneurs léger, Java EE 6 allie standards, robustesse et solutions éprouvées. Architecte senior, Antonio Goncalves intervient chez ses clients comme consultant et formateur. Ancien consultant Weblogic chez BEA Systems, il s'est spécialisé depuis 1998 dans l'architecture logicielle et les technologies Java. Antonio est l'auteur du livre [Java EE 5](#) aux éditions Eyrolles ainsi que le livre [Beginning Java EE 6 Platform with GlassFish 3](#) édité chez APress. Il est membre du JCP et intervient sur les JSR 316 (Java EE 6), JSR 317 (JPA 2.0) et JSR 318 (EJB 3.1). Il enseigne également la plate-forme Java EE au Conservatoire National des Arts et Métiers. Il est rédacteur technique pour les sites DevX et developpez.com. Antonio est co-fondateur et co-responsable du Paris Java User Group. Pour assister à cette session, [merci de vous inscrire rapidement](#) car les places sont limitées ! "

## JSR-299 vs JSR CrazyBob

Prenez le temps de lire [les quelques 68 commentaires échangés](#) entre Gavin King (auteur

d'Hibernate, spec lead de la JSR-299 WebBeans) et [Bob Lee "Crazy Bob"](#) (auteur de Google Guice, lead developer de la librairie Google Android). Je vous renvoie vers [le très bon billet](#) écrit par Guillaume Carré sur le blog de Xebia pour le fond du débat, car après avoir lu les échanges, je me suis demandé si quelqu'un s'était déjà posé la question suivante : à quoi servent les [JSR](#) (Java Specification Requests) ? Il y a plus de 10 ans, SUN Microsystems a eu l'excellente initiative de libérer le processus de définition du langage Java. L'objectif à l'époque était de prendre en compte les demandes grandissantes d'évolution des différents éditeurs du tout jeune langage Java. Rappelons que chaque JSR propose une implémentation de référence de la spécification, avec un code source accessible. Il faut aussi mettre au point un lot de tests afin de proposer un TCK (Technology Compatibility Kit). **Un match de boxe** Gavin King est spec leader de la JSR-299, spécification apparue début 2006 et qui est en phase finale, afin d'être livrée d'ici septembre 2009 dans la future version JEE6. Au début de cet année, la spécification initialement présentée comme WebBeans a été renommée "[Java Context and Dependency Injection](#)". Derrière tout cela, les acteurs de la communauté avait noté que la spécification avait en effet évolué vers l'injection de service, le terme WebBeans était alors trop réducteur par rapport à la portée de la spécification. Pourtant en relisant ne serait-ce que les premières lignes de la JSR, on voit que le scope initial, intimement lié à JBoss Seam, était la gestion du cycle de vie dans une application Web... La spécification a aujourd'hui bien évolué, et propose un standard d'injection de dépendances à Java EE 6. Preuve que le processus communautaire a du bon, il est clair que la future version entreprise de Java sera plus mature et plus flexible que les versions antérieures. L'effort du JCP, et je pense que les standards ont du bon, est que la spec JSR-299 offre un moyen **standard** donc portable d'apporter de l'injection de dépendance à une application d'entreprise. Je n'ai rien contre Spring, je pense qu'il est plutôt bon et raisonnable de conserver le système libre et d'offrir (tardivement) une spécification d'injection pour les services et les dépendances. **Un petit pavé** Crazy Bob jette un pavé dans la marre en annonçant [le 5 mai dernier](#) une proposition de JSR sponsorisée par Google, SpringSource, PicoContainer, Plexus, Tapestry IoC, et Simject en portant l'effort sur un scope plus réduit. Il souhaite proposer une spécification afin que l'injection de dépendance soit un standard dans Java SE 7 par exemple. Il pense même qu'il sera possible en 5 mois de finaliser cette JSR afin de l'inclure dans la future version de Java EE6. C'est tout simplement impossible, voir irréaliste. Cela entraîne bien entendu une réaction un peu épidermique des personnes de la spécification JSR-299. Vous suivez toujours ? Pourtant cette initiative n'est pas inutile car : - elle offrirait une gestion de l'injection native dans Java - elle ne nous forcerait pas à utiliser tout Java EE6 - elle reste pragmatique et ne se focalise que sur les points précis de l'IoC que tout le monde utilise (Guice, Spring...) - elle nous permettrait de rendre portable notre code d'injection (Spring vers Guice, Guice vers Spring, Spring vers Spring...) **JSR-299 une marieuse** Nous avons d'un côté la JSR-299 qui est pour moi une marieuse : faire se rencontrer les différents composants de votre application d'entreprise, vous aider à faire collaborer votre application avec la gestion des sources externes. A ce titre je pense qu'il est indispensable d'avoir cela sous la forme d'une spécification. Cela en fait certainement l'une des JSR les plus complexes. **Le petit nouveau** D'autre part, nous avons des développeurs utilisateurs de Spring et de Guice qui souhaitent rendre standard l'injection de dépendance dans une spécification. Cela revient aussi à dire que tout le monde recevrait le support de l'IoC... même ceux qui s'en passent très bien. Tout ceci ne vous intéressera que si vous faites appel à l'injection de dépendance, que vous le faites car vous souhaitez changer de framework d'IoC, que vous vous fichez du reste des excellents fonctionnalités dans Spring comme la gestion des abstractions (JDBC, JMS, Sécurité...), la programmation orientée Aspect, la gestion des Transactions, bref que vous ne vous intéressez QUE à l'injection de dépendance... Est-ce bien réaliste ? N'oublions pas aussi les "mauvais" effet de la surinjection de dépendance. Nous avons tous vu une fois une application avec des fichiers de mapping obèses, où le framework d'injection de dépendance est sur-utilisé. Rien ne remplace une architecture propre et une bonne utilisation des patterns. Je ne suis pas un fan du "tout-Spring" comme je n'étais pas fan du "tout-EJB". **Alors à quoi servent les JSR ?** Le principe des spécifications ouvertes a apporté à Java une bouffée d'oxygène. Dès 98 cela a été le moyen de sécuriser les investissements des clients finaux. Grâce à la normalisation et aux standards, un code écrit en Java sur Windows et compilé avec la JVM de SUN fonctionne sur votre serveur Solaris, avec votre serveur d'application Oracle Weblogic et JRockit. Standard est donc synonyme de sécurité, d'indépendance et de pérennité dans le temps. Les JSR sont donc un ensemble de règles appuyées par une implémentation de référence, avec accès libre au code source, ce qui nous sécurise et évite de devenir dépendant d'un éditeur de logiciel. **La dure vérité** Deuxième effet notoire : il y a toujours quelqu'un de plus intelligent que vous. Il y a toujours un petit gars quelque part dans le monde qui a déjà eu cette idée géniale. Il y aura toujours un étudiant prêt à faire 60 heures par semaine pour coder une API. Il y aura toujours un Doug Lea ou un Gavin King avec un oeil d'artiste sur le code. Java et donc les JSR vous apprennent une chose : vous êtes une m... et il va falloir faire avec. Non vous n'êtes pas génial à avoir pensé vous aussi à un framework d'inversion de contrôle. Le principe a été décrit en [1988](#) à ma connaissance. Non arrêtez d'écrire des classes comme "SynchronizedHashMap" et reprenez la lecture de la spécification Java. Les spécifications sont donc un moyen de s'assurer que les meilleurs principes se diffusent et que la plateforme Java évolue. C'est un canal de diffusion de la connaissance, qui permet aux éditeurs et aux acteurs de la communauté Open-Source de travailler main dans la main (ou poing dans figure aussi) mais de travailler ensemble. Quelque part, je vois cela comme les tests indépendants [EuroNCAP](#) de crash-test des voitures. Mine de rien, en 10 ans nous avons tous un ABS et bientôt l'ESP sur nos voitures, car un organisme indépendant a décidé de noter les constructeurs sur la sécurité de nos voitures. **Conclusion** Les standards ont du bon, merci à CrazyBob de penser qu'une JSR pour une version light de l'IoC est une bonne idée. Par contre j'ai franchement du mal à me projeter dans 2 ans, et à imaginer un standard de facto pour l'IoC. L'approbation de cette JSR par l'Expert Group, dont fait partie RedHat (Gavin King) pour sortir d'ici 5 mois me semble tout simplement impossible. Mais comme je vous le disais : nous sommes bêtes et nous devons donc attendre patiemment que les gourous du JCP s'organisent afin de s'accorder sur cette JSR. Messieurs de RedHat, de SpringSource et de SUN-Oracle : respirez un grand coup et revenez vous assoir afin de travailler ensemble, car le plus gros risque que court Java serait que ce processus soit remis en cause. Il faut que la communauté reste prudente et fasse la distinction entre ce qui est important pour faire évoluer Java, de ce qui est purement manoeuvre de noyautage. Il n'y a pas de

guerre froide entre les grands acteurs (Oracle/SUN, RedHat/JBoss, Google, IBM et SpringSource) et il serait dommage de commencer à casser ce système. Pour terminer allez lire les échanges entre Gavin King et Bob Lee :

<http://in.relation.to/Bloggers/CommentsOnAnnotationsForDependencyInjection> La page sur la proposition de JSR Light sans sucre d'injection de dépendance : <http://google-code-updates.blogspot.com/2009/05/javaxinjectinject.html>

## Première soirée JUG Toulousain

Avec un peu de retard pour cause de vacances, je vous informe que **ce soir** aura lieu la première soirée du JUG Toulouse, à l'Université Paul Sabatier. Au programme : GWT par Bruno Marchesson de Valtech et ensuite JavaME par Sylvain Wallez (CTO Goojet) Tous les détails ainsi que le plan d'accès corrigé sont sur le site du JUG Toulouse: <http://www.jugtoulouse.org/> Pour les Parisiens, rendez-vous ce soir pour la soirée GridComputing. Je vous livrerai un petit billet comme d'habitude à pas d'heure après la soirée. Stay Tuned !

## Compte-rendu de la soirée DataGrid et JavaRebel

En guise d'introduction : après être arrivé avec David assez tôt pour lui donner un coup de main, présentation des solutions de cache, de grid computing et de datagrid par [Erwan Alliaume](#) et [Cyrille Le Clerc](#) de Xebia ainsi que [Jean-Michel Bea](#) de FastConnect. Je reviendrai plus longuement ensuite dessus. En deuxième partie, présentation de JavaRebel par Toomas Römer de [ZeroTurnaround](#). Enfin en troisième partie, au bar le Falstaff j'ai interviewé Thierry Czarnyszka venu nous parler de Jazoon et enfin Toomas Römer sur JavaRebel, sur une proposition de Florent Ramière de Jaxio qui m'avait contacté pour cela cette semaine. En premier lieu, vous qui êtes venu vous avez peut-être fait connaissance avec de nouvelles têtes au sein de l'équipe du ParisJUG... dont moi-même ;-) Face au succès et au boulot que demande le ParisJUG, Antonio, David et Zouheir nous ont contacté afin de venir les aider et participer à l'animation du ParisJUG. Les 3 Jug-leaders restent les mêmes, nous sommes 4 nouvelles têtes pour aider l'équipe. Le rôle de chacun n'est pas spécialement défini, mais bien entendu selon nos profils et notre expérience, chacun apportera sa contribution. Les autres contributeurs de la "Crew" sont : - José Paumard - Tanguy Bayard - Charles-Alexandre Sabourdin et donc moi-même, Nicolas Martignole. Erwan débute la présentation avec tout d'abord une présentation des caches distribués. [Ehcache](#) ou [JBoss Cache](#) sont deux solutions populaires de la distribution de données. Erwan explique que le premier besoin est de décharger de la base de données le chargement récurent de certaines données. Un cache permet en effet d'éviter de multiples accès à la base pour ne conserver que les données les plus lues. La [JSR-107](#) est par ailleurs une JSR sur la gestion du cache pour ceux qui souhaitent y jeter un oeil. L'utilisation d'un cache comme il l'explique ensuite, s'accompagne cependant d'une contre-partie, à savoir l'utilisation de mémoire, d'espace disque, et d'échanges sur le réseau afin de maintenir la cohérence du cache. Afin de ne conserver que les données les plus utiles, différentes stratégies d'éviction permettent de faire le ménage : LIFO, FIFO, temps, taille, etc. La gestion des transactions par contre dégrade les performances, bien qu'il soit possible de s'en servir. L'objectif reste bien de décharger de la base de données et donc des utilisations comme le cache de second niveau d'Hibernate avec Ehcache sont des cas répandus. Erwan montre aussi un slide afin d'attirer notre attention sur les performances. Il attaque ensuite le sujet des Network Attached Memory (NAM) comme Terracotta. L'objectif ici est de répliquer au niveau de la JVM les données. La mise à jour n'est plus alors applicative mais native à la JVM. C'est donc bien la JVM qui se charge de se synchroniser avec d'autres JVM. L'application adresse un espace de stockage répliqué en cluster, ce qui permet à plusieurs JVM de partager un espace mémoire commun. Erwan attire fort justement notre attention sur le besoin d'un Virtual GC comme d'une Virtual Heap, ce qui a un coût. Il parle aussi de l'invocation de méthode distante, en nous rappelant que Terracotta par ses propres mécanismes d'optimisation n'a pas forcément la même donnée sur tous les noeuds d'un cluster. Jean-Michel Bea nous parle ensuite des Grilles de données. FastConnect dont il est l'un des consultants, est un intégrateur privilégié de Gigaspace. Il parle en connaissance de cause de ce sujet. Une grille de donnée vise à répartir et à diviser la donnée, afin de la rapprocher de l'utilisateur. Nous sommes là sur des problématiques d'architecture de haut niveau. Il présente les solutions de Grilles de données comme des "DB Shock Absorber" visant à atténuer l'utilisation des bases classiques. J'ai bien aimé aussi ses rappels sur les limites du réseau, le coût de la latence réseau et du débit limité. Nous parlons bien là de sites géographiques, d'infrastructure réseau distribuée. Cyrille prend la parole en expliquant qu'historiquement, lorsque le transport des données coûte trop cher en temps, la solution aujourd'hui est d'utiliser des procédures stockées afin d'effectuer le traitement en base. Nous savons tous aussi que le souci de ces solutions est qu'une partie de la logique métier se retrouve en base. Cyrille prend ensuite la parole et continue la démonstration sur la Grille de donnée. Comme il l'explique, le traitement de données massives comme la réservation d'un billet de train, s'effectue aujourd'hui sans soucis sur des Mainframes. Avec une capacité de 1,5 To de mémoire et 64 processeurs, pour un coût supérieur au million d'Euro, ce n'est pas une solution envisageable partout. Sa partie sera donc de nous expliquer comment nous pouvons nous en sortir avec les solutions du marché. Je vais passer moins en détail que la présentation de Cyrille, car son contenu était très complet. Il prend l'exemple d'un système de réservation de billet de train. Comme il l'explique, si vous devez travailler avec une grille de donnée, votre premier réflex doit être de revoir votre modèle Java. Là où une application de gestion est une pâle copie de la base relationnelle, vous devez revoir votre modèle pour utiliser efficacement une grille de données. Ses recommandations : créer tout d'abord une clé de partitionnement, constituée des données métiers, de l'association de plusieurs entités, de plusieurs classes. Ensuite, identifiez et adaptez ce modèle Java à votre problématique métier. Dans son exemple, pour déterminer si un siège de train est booké, la version SGBDR utilise une table

Booking. On imagine une requête et donc une jointure. Dans le modèle Grid, il le remplace par un boolean, calculé au moment de la création et de la mise-à-jour des données. Je comprends alors que l'utilisation correct d'une Grille doit nécessairement entraîner une refonte réfléchie de son modèle de données. Jean-Michel reprend la parole pour expliquer le cas classique d'un transfert d'argent entre 2 comptes. Il y a quelques années, nous avons essayé de résoudre ce problème par une transaction afin que le retrait du compte A, puis le crédit du compte B, soit effectué dans une transaction. Si vous passez ce modèle vers une grille, il faudra alors isoler les 2 Comptes. Pour résoudre le problème, on créera une entité CheckOut pour le compte A avec la liste des transferts sortant. Le compte B aura une entité CheckIn avec la liste des transferts arrivant. A noter que dans la finance c'est exactement ce que l'on fait du côté du BackOffice pour la gestion des Deals entre différents Brookers sur le marché. Jean-michel explique que la notion de transaction n'est pas nécessaire sur la forme que nous pensons. Ce qui doit être atomique, c'est l'ordre d'envoi d'argent pour le compte A, et l'ordre de réception pour le compte B, ce que font les applications classiques. Très intéressant. On se force à ce modèle afin que par la suite, l'application puisse être déployé sur une grille. Votre modèle sera alors réparti dans un espace distribué, ce qui pourra garantir une montée en charge tout à fait correcte. Cyrille parle de Websphere extremScale qui semble très intéressant, et Jean-Michel de Gigaspace, ce qui permet d'illustrer la discussion avec de vrais cas clients. Les questions de la salle portent sur la réelle nécessité d'utiliser une Grille. Cyrille répond avec le système de réservation de billet de train, je pense qu'il parle à cet instant de sa propre expérience chez Voyages-SNCF. Il explique qu'il existe certes des alternatives comme Oracle RAC, mais que ce sont des solutions techniques, là où eux trois viennent de nous parler de solution d'architecture, pour qu'ensuite l'application se comporte mieux et puisse monter en charge. Après un buffet où je vous rappelle que la règle pour bien manger, c'est de rester à côté d'Antonio, la deuxième heure reprend avec notre équipe de 3 speakers. Ils viennent ensuite sur le sujet des API. Cyrille présente le pattern Map Reduce, avec un exemple de recherche de place de train, où chaque gare est colocalisée sur différents serveurs. La colocalisation des données, c'est la répartition géographique intelligente de l'information. C'est tout simplement trouver une stratégie pour ranger les données correctement. Concernant les problématiques d'ACID, Jean-Michel explique que Gigaspace peut proposer ce type de service. Mais est-ce vraiment souhaitable ? Des solutions d'architecture et l'utilisation de patterns permettent de créer un modèle de donnée, différent du modèle relationnel, pour répondre à ces problèmes. Je retiens qu'il n'y a donc pas d'intérêt à utiliser une Grille si vous ne pouvez pas vous permettre une certaine forme d'abstraction par rapport à la base de données. Pour terminer, ils abordent la mise à jour d'une base de données classiques par rapport à une Grid et inversement, avant de conclure avec un slide reprenant les 4 types de grilles. Je vous mettrai les détails de ce slide demain puisqu'Erwan m'a envoyé les slides. En conclusion, une bonne présentation qui nous permet de prendre conscience que les solutions de Grid Computing s'accompagnent d'une nécessaire réflexion sur le modèle de données. La colocalisation, la déportation du traitement au plus près de la donnée, la mise en place d'outils de réplication avec la base de données pour le reste du modèle du SI sont donc les clés de la mise en place d'une solution de grille. Je vous proposerai dans un autre article le compte-rendu sur la présentation de JavaRebel, ainsi qu'un interview exclusif de Toomas Römer de ZeroTurnaround, l'un des développeurs de JavaRebel. Il présentera le même sujet le mercredi 13 mai au [Tours JUG](#) et ensuite le jeudi 14 mai au [Bordeaux JUG](#). Si vous lisez ces lignes et que vous êtes à Tours ou Bordeaux, dépêchez-vous de vous inscrire et passez le bonjour de ma part aux JUG leader de là-bas.

## JavaRebel: interview de Toomas Römer

Rencontre avec Toomas Römer à l'occasion de sa venue le 12 mai dernier [au Paris JUG](#). Sur une idée de Florent Ramière [de Jaxio](#), j'ai préparé quelques questions afin de vous expliquer tout d'abord le principe de JavaRebel, puis un peu l'histoire de la société [ZeroTurnaround](#). L'interview s'est déroulé au Falstaff, la fameuse troisième mi-temps qui a lieu après chaque soirée du Paris JUG. *The original interview was conducted in English. You can try to use [Google Translate](#) but the quality is not great. French to English is never perfect. Sorry for english readers.*



**Bonjour Toomas, peux-tu te présenter en quelques mots ?** Je m'appelle Toomas Römer, je viens d'Estonie. Je suis l'un des fondateurs de la société ZeroTurnaround avec Jevgeni Kabanov. Je contribue sur le produit JavaRebel, JspWeaver et j'ai aussi travaillé sur le framework web [Aranea](#). Mon blog : <http://dow.ngra.de> **A quoi sert JavaRebel ?** JavaRebel est un système qui permet de gagner du temps en évitant de devoir redéployer toute une application, en particulier avec les serveurs d'application J2EE. Pour cela, un agent s'installe dans la JVM. Il surveille un ensemble de répertoire que tu configures dans lequel se trouve tes classes, tes ressources, tes JSP, bref ton application. Lorsqu'une modification survient, l'agent se charge alors de remplacer ton code de manière transparente. Cela évite de devoir arrêter et relancer ton application. Là où le Hotswap en Java est limité, JavaRebel par un système sophistiqué d'enrichissement du byte-code permet d'accélérer d'environ 80% le temps de développement. Tu peux ainsi ajouter des méthodes à une classe, changer ta configuration Spring, puis tester et valider immédiatement si le tout fonctionne. Les 2 seuls cas qui obligent encore à arrêter et relancer l'application sont l'ajout ou le retrait d'interface sur une Classe ou le changement d'une Superclasse. Un tableau comparatif complet [à cette adresse](#) permet de comprendre ce qu'apporte JavaRebel. **D'où vient cette idée ?** Jevgeni Kabanov a eu l'idée en 2007, sur un besoin que chaque développeur J2EE connaît : comment déployer rapidement une application sans perdre de temps ? J'étais à l'époque encore à l'armée, nous avons commencé à travailler dessus. ZeroTurnaround est maintenant une petite entreprise de 5 développeurs, ainsi que d'autres personnes pour la partie commerciale et la gestion. Nous avons participé à DevOxx 2008, à QCon et nous serons à JavaOne cette année. Nos produits sont: - JavaRebel pour accélérer le développement Java - JSPWeaver supprime complètement la

compilation des JSP en la remplaçant par son interprétation à l'exécution - LiveRebel est notre solution de changement à chaud d'applications d'entreprise. Nous venons d'ouvrir la bêta.

**J'imagine que techniquement, vous avez dû rencontrer pas mal de difficultés non ?** Nos soucis ont surtout porté sur les vendeurs de JVM. Nous supportons la JVM HotSpot de SUN, Oracle JRockit, IBM J9 et l'Apple MRJ. Le support de la jvm d'IBM est celui qui nous a demandé le plus de travail. Lorsque l'application se plante, il était difficile, faute d'outils, d'avancer. Mais avec beaucoup d'efforts, nous pouvons aujourd'hui offrir le support de ces JVM. Du côté des serveurs d'application nous supportons les plus connus (ndlr: [liste complète ici](#)). Comme tu le sais, les ClassLoader des serveurs d'application ne sont pas standards. Entre JBoss ApplicationServer et IBM Websphere, cela nous aussi demandé pas mal de travail. Caucho Resin était aussi assez difficile à mettre en place. Au final nous avons 10 containers supportés. **Parlons licence, comment fonctionnez-vous ?** JavaRebel est un produit commercial, tu peux avoir des licences d'évaluation de 3 Mois, ce qui laisse le temps de se faire une idée je pense de ses qualités. Nous mettons surtout en avant le temps, et donc l'argent gagné grâce à cet outil. Tu as d'ailleurs un outil d'estimation du coût sur [la page d'accueil de notre site](#). Le prix de la licence est de 59\$ pour quelq'un de seul, 149\$ pour un salarié d'une entreprise et surtout, nous offrons gratuitement des licences [aux projets Open-source](#) ainsi qu'aux personnes avec une ceinture marron sur JavaBlackBelt ! (NDLR: 10 développeurs à 40 EUR/heure qui prennent 8 mn par cycle de compilation-développement-test et qui relancent ne serait-ce que 2 fois par heure l'application coûtent environ 158 000 EUR par an...) **Quelles méthodologies utilisez-vous pour ajouter le support d'un nouveau framework comme Spring ou Guice ?** Nous sommes nous même développeurs sur d'autres projets, nous suivons donc beaucoup l'actualité des frameworks que nous utilisons. La communauté des utilisateurs de JavaRebel via les forums sur ZeroTurnaround nous font part aussi des cas qu'ils rencontrent, ce qui permet de faire évoluer le produit et de continuer à sortir une version environ tous les 3 mois. **Vous vous concentrez sur la stabilité de l'outil ou le support de nouveaux frameworks ?** Plutôt maintenant sur la stabilité, le panel des containers étant assez large. **Du côté des éditeurs comme Eclipse que tu nous as montré ou IDEA IntelliJ, bref des outils, qu'offrez-vous ?** Nous avons tout d'abord une documentation complète et gratuite sur le site pour que chacun puisse configurer son IDE favori, que ce soit Eclipse, IDEA IntelliJ ou Netbeans, [avec qui j'ai eu un peu plus de mal](#). Nous avons un plugin dans IDEA IntelliJ, ainsi qu'un plugin Maven2 pour générer le fichier rebel.xml qui te permet de lister les répertoires de type target/classes utilisés par différents modules Maven2. **Une question : j'ai un serveur Weblogic sur Linux. Je travaille avec IDEA IntelliJ sur Windows. Comment faut-il configurer son environnement ?** JavaRebel fonctionne en surveillant tes classes, il n'y a pas d'accès réseau entre l'agent de la JVM et ton éditeur. Il faut donc que ton serveur Linux fonctionne avec une Webapp explosée, que ton répertoire sous Windows soit accessible à distance de Linux, avec un montage. Ou plus simplement, que tu aies par exemple un partage sur Linux afin de déposer tes classes à partir de Windows au moment de la compilation. **Là tu es d'accord que c'est pas non plus le plus facile non ?** Le cas le plus courant est quand même d'avoir tout sur sa propre machine non ? Serveur d'application et éditeur Java, afin de faciliter le développement. Nous avons beaucoup d'articles sur notre blog comme "[Developping in Exploded Format](#)" pour aider les développeurs à trouver la meilleur configuration possible. **Quels sont les autres fonctionnalités ?** JavaRebel est capable de recharger tes fichiers de configuration de Spring. Cela te permet d'ajouter un Bean, de le déclarer dans la configuration de spring, puis simplement de le tester tout de suite sans devoir arrêter et relancer ton serveur. Nous avons aussi le support de l'injection, avec AspectJ grâce à nos plugins, ainsi que Struts2, Google Guice, Tapestry4, Apache Wicket et WebObjects. ([liste complète](#)) **Le mot de la fin ?** Merci au Paris JUG, je vais aller ensuite à Tours puis Bordeaux. N'hésitez pas à télécharger et tester JavaRebel, contactez-nous si vous souhaitez une licence pour évaluer JavaRebel au delà des 30 jours de la période d'essai. Et à bientôt ! **Merci Toomas** Merci Nicolas ! **Références:** - La page [JavaRebel](#) sur le site de ZeroTurnaround - [Le blog](#) de Toomas et de Jevgeni - [Jevgeni](#) sur Twitter - [Toomas](#) sur Twitter - [Exemple d'article de Jevgeni](#) sur la lenteur de System.currentTimeMillis... **Rencontre avec Toomas Römer à l'occasion de sa venue le 12 mai dernier au Paris JUG.** Sur une idée de Florent Ramière [de Jaxio](#), j'ai préparé quelques questions afin de vous expliquer tout d'abord le principe de JavaRebel, puis un peu l'histoire de la société [ZeroTurnaround](#). L'interview s'est déroulé au Falstaff, la fameuse troisième mi-temps qui a lieu après chaque soirée du Paris JUG.

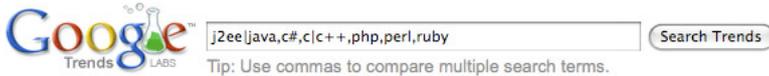
## James Gosling sera à Jazoon 09

Finalelement James Gosling, l'un des fondateurs du langage Java, sera là le mardi 23 juin prochain à [Jazoon](#). Pour rappel, Jazoon est une conférence qui aura lieu du 22 au 26 juin à Zurich en Suisse. Idéalement situé avant l'été, ce sera l'occasion d'assister à de bonnes conférences. J'y serai du mardi 23 au jeudi 25. L'équipe d'organisation de Jazoon a d'excellentes idées. La chasse à James a commencé il y a 2 semaines. Inscrit à la liste de diffusion, je reçois [un email m'invitant à demander au manager de James de le laisser venir](#) en Europe. Avec beaucoup d'humour, le mail tourné à la "Free James" a donc rempli son objectif : il sera bien là et se chargera de la Keynotes d'ouverture. Mardi dernier, Thierry Czarnyszka de l'équipe d'organisation de Jazoon est passé au Paris JUG. Il nous a montré la première page de la Tribune, flanquée d'une publicité pour Jazoon. Un peu intrigué j'ai mené mon enquête, et c'était en fait une bonne blague, Thierry avait découpé et collé une publicité de Jazoon sur le journal ! Démasqué ! [Le programme est en ligne](#), et j'ai aussi noté un événement qui nous changera de Devvix : l'organisation d'une soirée le mercredi soir. Je trouve que l'initiative est bonne, car les conférences sont aussi l'endroit privilégié pour faire des connaissances, prendre des contacts et étendre son réseau. La partie réseautage est peut-être aussi importante que les conférences elles-mêmes. Si vous pensez à changer de boulot, c'est le moment d'utiliser le DIF, d'aller faire un tour là-bas et de discuter avec des exposants, d'aller écouter les experts sur JEE6, Spring, les architectures REST... Bon pour terminer le plus intéressant. Si je vous parle de Jazoon c'est que j'ai aussi quelque chose pour vous. Bénéficiez d'une petite réduction en venant de la part du blog le Touilleur Express en envoyant un email à nicolas AT touilleur - express.fr avec dans le sujet "Jazoon". Grâce à cela, vous pourrez encore vous inscrire et bénéficier d'un prix un peu plus intéressant. Rendez-vous en juin pour un retour sur Jazoon en direct de là-

bas !

# Java est-il populaire ?

La grande question lorsque l'on cherche du travail par les temps qui courent, c'est de savoir si le vernis à ongle sur son CV intéressera toujours un employeur. Si vous êtes actuellement "à l'écoute du marché" comme on dit, doux euphémisme pour dire en recherche active d'un autre poste, voici quelques indicateurs pour vous expliquer l'état du marché. **Popularité de Java dans la communauté Open-Source** Tout d'abord un indicateur pour vous rassurer : Java est encore très populaire. Classé premier depuis plus d'un an par [l'indicateur TIOBE](#), il a regardé la dégringolade de son copain le C# qui est en perte de vitesse. Je pense qu'un indicateur aussi de la popularité de Java est de regarder les projets sur SourceForge.net. J'ai trouvé [un site](#) qui a listé 34210 projets Java, suivi de 27723 projets C++ et 22855 projets en C. Le langage C# arrive après le PHP et le Python avec 8825 projets. Je retiens que C et C++ représentent beaucoup plus de projets que Java cependant. Cela semble réaliste avec ce que l'on observe sur Internet. Je ne cherche pas à dénigrer bien entendu le C#, mais à expliquer par des chiffres ce que l'on voit sur Internet. Google Trends n'est pas un outil très fiable. Si vous tapez Java, sans surprises les résultats affichés parlent de l'île de Java. Pour obtenir quelques chiffres avec un peu plus de sens, il faut combiner des requêtes et il semble possible d'obtenir une courbe. Voyez par vous-même [ici](#). Google Trends n'est qu'un indicateur des recherches effectuées par les Internaute.



Les

**offres d'emploi** Un exercice amusant est de regarder les offres d'emploi sur quelques sites français. Voici la méthode utilisée : tapez un mot clé et comptez le nombre de résultats pour chaque mot clé. Le site de recherche d'emploi "[LesJeudis.com](#)" affiche sur la page d'accueil le nombre d'offres d'emploi : 687 offres Java, 360 offres PHP, 308 offres .NET. Pour le site Monster.fr j'ai tapé l'intitulé du poste, afin de comptabiliser le nombre de propositions. Pour le site lesjeudis.com j'ai utilisé [la sélection par technologie](#) afin d'affiner les résultats. J'ai bien conscience que cela ne représente qu'une photo instantanée du marché.

Nombre d'offres d'emploi par langage  
mars 2009 Le Touilleur Express

	Monster.fr	Lesjeudi.com
Développeur Java	117	176
Développeur J2EE	73	325
Développeur C#	64	263
Développeur C	64	435
Développeur C++	82	3
Développeur Python	11	40
Perl	60	78
PHP	320	340
Ruby	2	7
Cobol	43	64
Java (mot seul)	465	500
Javascript	154	136
Java (ds intitulé poste)	168	NA
Architecte Java	9	NA

Il est difficile de distinguer dans ces résultats un indicateur de bonne ou mauvaise santé du marché. Cependant comme nous allons voir plus loin, les personnes avec une expérience en Java sont plutôt bien placés sur le marché de l'emploi. **Combien de développeur Java en France ?** On estime à 500 000 personnes le nombre d'informaticien en France selon [le Syntec](#) et les chiffres de [la DARES](#) (Direction de l'Animation de la Recherche, des Etudes et des Statistiques). Selon la DARES, 60% sont des diplômés Bac+5, soit 300 000 personnes environ. Le nombre d'informaticien en dessous de 40 ans, selon [l'étude de 2002 de la DARES](#) est de 263 258 personnes. Si 10% de ces personnes sont des développeurs Java, du junior jusqu'à l'architecte senior ou l'expert, cela représente 26 325 personnes en France. Cela me semble un peu optimiste non ? ramené à 500 offres d'emploi sur le site Monster.fr, le ratio entre personne capable de taper quelques lignes de Java et la demande serait de 1.89 offres pour 100 postes. [Le baromètre pour les Freelances](#) du site Freelance-Info.fr est aussi un bon indicateur de la demande par secteur, en visant cependant le marché des indépendants, qui reste très minoritaire en France. La DARES estime à 1 le nombre d'informaticien indépendant en 2006, ce qui représente à

peine 5000 personnes... Java est le 3ème poste demandé. **Java est populaire et il y a des offres mais...** J'ai bien conscience que cette analyse de coiffeur fera rire doucement une partie des lecteurs. Faire une analogie entre la popularité d'un langage, les recherches sur Google Trends et des calculs d'alchimiste, je suis motivé... Mais au delà de cela, si nous regardons le marché de manière objective, la demande pour les profils Java reste assez élevée. Ce que les personnes qui ne connaissent pas la plateforme Java ne voient pas de premier abord, c'est l'explosion des solutions open-sources dans le monde Java. C'est très certainement aussi l'une des raisons de son succès, et donc de sa popularité. RedHat place la France en tête de 75 pays utilisant des solutions open-sources. Le succès de consortium comme O2Web, d'éditeurs comme eXoPlatform, la bonne représentation des français dans la communauté open-source (Guillaume Laforge pour Groovy, Emmanuel Bernard pour Hibernate, Antonio Goncalves pour JEE6...), pour moi la communauté se porte très bien depuis ces quelques dernières années. Sur ce plan, Java est populaire ! A propos de la communauté Java en France: les passionnés de la plateforme Java se retrouvent entre eux pour échanger sur Java, ce que l'on appelle des **Java User Group** ou JUG. [A Paris](#) c'est prêt de 200 personnes qui chaque 2ème mardi se retrouvent pour assister à des présentations. Dans toute la France, les JUG renforcent la cohésion et permettent à chacun de se rencontrer, entre passionné. A ce jour voici la liste et les informations sur les JUG : - [Le Tours JUG](#) - [Le Bordeaux JUG](#) - [Le Breizh JUG](#) (Bretagne, Rennes) - [Le Nantes JUG](#) - [Le Lyon JUG](#) - [Le Lorraine JUG](#) - [Le Toulouse JUG](#) - [Le Riviera JUG](#) basé à Sophia Antipolis - [Le Marseille JUG](#) - [Le ChtiJUG](#) - [Le Normandy JUG](#) - [Le Paris JUG](#) - [Le Poitou Charentes JUG](#) (ajouté le 18 mai) - [Le JUG virtuel de Developpez.com](#) - ... (m'en manque encore) Enfin un dernier mot sur les conférences. La conférence [Devoxx](#) est organisée par le BeJUG (Belgique). En décembre 2008 avec 3200 participants, c'était l'un des événements les plus importants. En juin prochain aura lieu aussi [Jazoon](#) avec environ 880 participants, 37 pays représentés. Dans quelques semaines, aura lieu enfin JavaOne, l'événement annuel sur Java à San Francisco. C'est donc encore un signe de bonne santé de Java. Je regrette simplement pour ma part l'absence d'événement en France. Je pense que l'organisation d'un événement ne peut se faire que par un JUG. Il faudra que la communauté des développeurs en France se réveille, sorte un peu des bureaux pour aller à une Conférence en France. Nous avons des présentateurs francophones excellents, pourquoi n'y-aurait-il pas un jour un événement de cet ampleur en France ? à Paris ? **Conclusion** La plateforme Java est populaire. Que ce soit du côté du marché de l'emploi, des différents projets open-source, des déploiements en entreprise ou encore des communautés en France comme dans le monde : il se passe toujours quelque chose à côté de chez vous. Alors si quelqu'un vous dit que "le marché est tendu" merci de lui faire lire [cet article](#) et aussi celui-ci...

## Grande enquête sur Scrum

Le [French SUG](#) (French Scrum User Group) a lancé [une enquête en ligne](#) sur Scrum et les méthodes Agiles. L'objectif de cette enquête est de collecter et de publier une étude sur les méthodes Agiles en France. Axé sur Scrum, mais pas seulement, c'est l'occasion de répondre, d'interroger votre client, une équipe qui a mis en place Scrum ou autre, bref de donner votre avis. En nom et prénom du parrain vous pouvez préciser mes coordonnées. A propos d'Agilité, préparez vos lecteurs RSS ! Je serai aux [XP Days](#) lundi et mardi prochain. J'espère vous faire partager ces 2 jours qui se passeront à Paris via Twitter tout d'abord, puis via le blog. Au programme regardez ce que j'ai sur le Menu. Tout d'abord je laisserai Eric "Bob" Mignot faire une présentation de Scrum avec Eric Segulier, j'irai peut-être voir "[Un projet extrêmement ambitieux et flou, utilisant les technos les plus récentes, distribué sur plusieurs continents... vous savez](#)" par Isabelle Therrien et Érik Lebel mais la session "[Blanche Neige](#)" me tente bien aussi... Les présentations sur Lean et Kanban m'intéressent beaucoup. A ce propos je vous recommande de lire "[Kanban vs Scrum](#)" d'Henrik Kniberg. Attention c'est [le PDF](#) qu'il faut lire, pas l'article du blog. On assistera aussi à la présentation "[Product Owner, qui es-tu ? que fais-tu ?](#)" de Guillaume Carré et de Guillaume Bodet de Xebia France. Mais l'une des participations à laquelle je serai sans fautes c'est la présentation d'Eric Lefevre-Ardant : "[Scrum est-il dangereux ?](#)". J'ai rencontré Eric à l'occasion du 4ème BarCamp, il m'a expliqué l'objectif de cette session qui sera de faire une autocritique sur la mise en place de Scrum. L'idée trouve ses sources à la dernière CITCON "[Is Scrum Evil?](#)" qui tire son contenu d'un article de James Shore très cité dans la communauté Agile: "[The Decline and fall of Agile](#)". Nous avons discuté ensemble sur la traduction car evil pouvant se traduire par "dangereux" ou "mauvais". Au delà du contenu qui sera animé par Eric, du côté des spectateurs je pense que nous aurons l'occasion de voir quelques stéréotypes : les Scrumistes aveugles qui pensent que Scrum se suffit à lui-même, les XPiens extrémistes qui pensent qu'XP peut répondre à l'ensemble des besoins, etc. Ca va être sport ! Pour terminer il y a un article sur le Touilleur dont je suis assez content, qui m'a demandé pas mal de boulot : "[Les origines de Scrum](#)". Si vous voulez le relire, il est encore en rayon dans la section "[Scrum](#)" du Touilleur Express. A propos enfin du bashing sur Scrum, à la page People j'ai aussi un bon numéro avec "[Scrum c'est fini ou le sujet People de janvier](#)". N'oubliez pas enfin de remplir [le questionnaire sur l'Agilité](#) sur le site du French SUG.

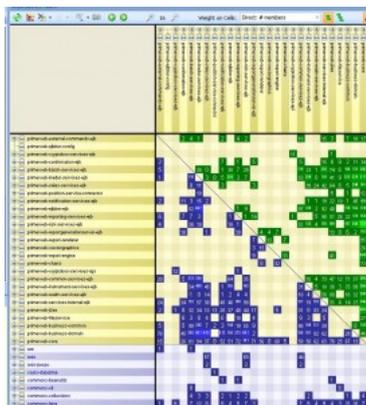
## Pour faire de l'archéologie dans du code Java

Séance émotion il y a deux jours. Avec un autre Nicolas (nous sommes 3 nicolas ici sur 16 personnes) nous nous attaquons à un chantier tout à fait intéressant qui fait partie de la vraie vie d'un Architecte : du dépolluage de code. Enfin quand je dis dépolluer, c'est pour rester poli. Notre objectif est de pouvoir rapidement tester deux solutions (Mule et TIBCO BusinessWorks) sur la plateforme de traitement financière de notre client, afin de réduire la dette technique et la quantité de code interne à maintenir. Bref nous voilà plongé dans des discussions pour essayer de comprendre comment découper ce *GrosMachin* en *PetitMachins*. L'idée derrière sera ensuite de jeter des *PetitsMachinsPourris* et de remplacer le tout par des *PetitsMachinsPropresEtNeufs*... Oui car quelque chose que l'on ne vous dit pas tellement c'est que les 2/3 de notre boulot à nous, les "architectes" c'est plutôt de faire du ravalement de façade que de construire un nouvel immeuble

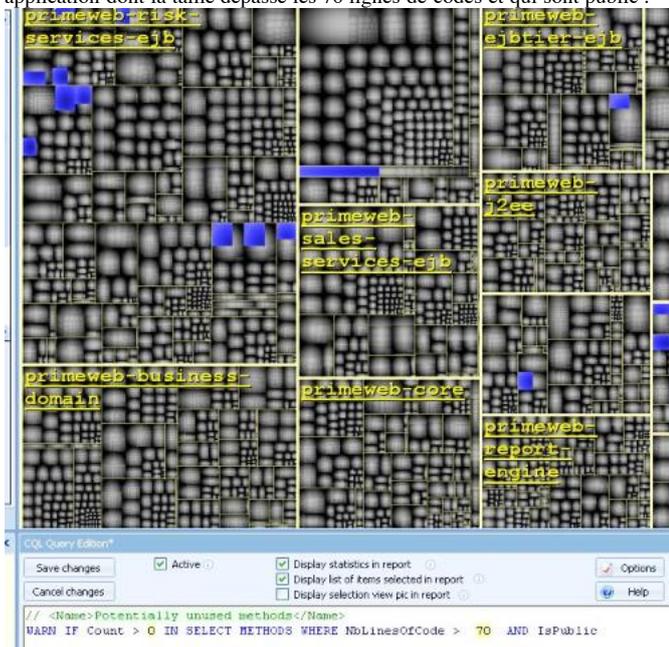
hyper design. Nous nous efforçons de trouver des solutions d'architecture afin d'assurer la montée en charge de la plateforme et de réduire les coûts de support. Or ces derniers mois, les chantiers que nous avons menés étaient plutôt axés sur la production, la compilation et l'intégration. Je me suis ensuite attaqué à la partie gestion d'équipe, en mettant en place Scrum avec succès. Notre prochaine mission consiste maintenant à revoir le traitement par Batch afin d'accélérer la plateforme. Le choix d'une architecture orientée Événement semble acquis afin d'effectuer le traitement de l'information au fil de l'eau. Il est temps maintenant de découper le système existant afin de venir greffer ces nouveaux composants. **Ce que vous allez croiser sur votre chemin (lorsque vous serez un maître Java)** Les logiciels créés sur des architectures J2EE EJB2.1 sont aujourd'hui matures. Ils constituent un bon nombre d'applications déployées chez les clients avec lesquels nous travaillons. Nous voyons sur le terrain des applications en production, avec bien souvent du Spring et de l'Hibernate ajoutés par la suite, afin de réduire le code technique. Mais force est de constater que cela rajoute parfois encore plus de travail, car la bonne maîtrise d'un outil de mapping relationnel ou d'un framework léger d'injection demande du temps. Prenez le temps de parler avec des consultants et des indépendants, une bonne partie de l'activité aujourd'hui consiste à moderniser l'existant et à revoir l'architecture d'une plateforme. Bref nous sommes en plein dedans... **La taille du sabre laser ne compte pas** Après avoir migré vers maven2 l'ensemble du projet, avoir travaillé sur la mise en place de Bamboo et d'Hudson, avoir même pris du temps pour que les développeurs puissent utiliser une VirtualMachine avec un serveur Weblogic pré-installé, nous nous sommes assis devant [le meilleur éditeur Java](#) du monde pour commencer à regarder comment reprendre ces 11000 fichiers Java, sans bien évidemment tout casser. A cet instant de la présentation il y a deux personnes : il y a un Nicolas conservateur et prudent, et il y a un Nicolas qui ferait bien tout tomber histoire de reprendre le code source. Bien entendu aucuns des deux n'a la bonne démarche, ce qu'il faut faire se situe quelque part entre les deux positions. Si nous croyons fort à l'homéopathie, moi personnellement je ne crois pas que changer "un peu de code" et revoir progressivement l'ensemble soit la bonne solution. A contrario, mon idée de reprendre à zéro les packages, de remonter les modules maven2 proprement plutôt que de subir l'existant n'est pas non plus la meilleure idée : risque de ne pas réussir, temps nécessaire pour immobiliser l'ensemble du code, temps perdu à identifier du code "mort" inutilement refactorisé, etc. Donc retenez ceci :

Ne sciez pas la branche où vous êtes assis mais ne prenez pas une pince à épiler pour retirer une poutre.

Alors quelque part entre ces 2 points, notre mission est d'extraire une partie du flot de traitement, afin de pouvoir le remplacer facilement et sans risques par une nouvelle architecture. Nous sommes dans un projet basé sur des EJB (des vieux) avec un habillage de Printemps, collection Spring. Le tout servi sur un serveur d'app, avec s'il-vous-plaît une couche Web et une couche Métier. Les gens de la Sécurité ont aussi demandé que la partie Web soit sur un réseau A et la partie Métier sur un réseau B, et que la base de données soit sur un réseau C... Et A n'a pas le droit d'aller discuter avec C. Vous commencez à visualiser un souci non ? Pour remplir notre mission nous avons tout d'abord tenté l'aventure "*on s'assoit et on réfléchit très fort*" devant [IDEA Intellij](#). Au bout de 2 heures avec quelques feuilles remplies de dessins, vu la complexité et de la sévérité du cas, nous n'avions pas tellement avancé. Notez que je joue la transparence en vous racontant que tout ne se passe pas bien dans la vraie vie. Premier objectif : regrouper par module maven les grandes parties du projet. S'assurer que les dépendances transitives sont correctes afin que les modules de base comme "common" soient compilés avant la partie du domain, la couche service, pour terminer sur la couche de présentation. Découper ensuite verticalement par fonction. Mettre à feu doux, remuer de temps en temps, prier très fort, livrer tous les modules réorganisés aux développeurs. Attendre 3 ou 4 semaines, s'apercevoir que votre chef d'oeuvre est à nouveau tout pourri car Mike a tiré une dépendance de la couche domain vers la couche web, craquer nerveusement, partir, fin de l'histoire... Ou sinon vous pouvez aussi demander à vos amis (si vous avez des bons amis) ce qu'ils en pensent. 2 mots arrivent dans la conversation : XDepend et SonarJ. Voilà de beaux couteaux pour découper proprement votre produit sans en mettre partout. Alors allons-y, leçon de cuisine avec le Touilleur Express. **XDepend, what else?** [XDepend](#) est littéralement l'équivalent d'un IRM pour vieux code fatigué. L'outil analyse votre structure de projet à partir des JAR de votre projet. C'est un outil destiné aux développeurs plutôt qu'au chef de projet. A ne pas confondre avec Sonar, ni avec SonarJ dont je vous parlerai plus loin. XDepend est le petit cousin de l'excellent outil NDepend, utilisé dans le monde .NET pour adresser le même besoin. NDepend est un outil créé par Patrick Smacchia. XDepend est commercialisé par [OCTO Technology](#). L'outil permet d'extraire la structure d'un produit, d'analyser le code avec des metrics, d'identifier la structure et les dépendances dans le code afin de le ré architecturer ensuite correctement. Les points forts sont l'outil graphique et le langage de requête, auquel j'ajouterai la performance global de l'outil. Pour effectuer mon test je suis parti d'un EAR, que j'ai extrait dans un répertoire de test. Après avoir téléchargé la version d'évaluation, j'ai commencé mon analyse en moins de 10 minutes. L'outil graphique facile à prendre en main propose différentes vues. L'affichage matriciel et la radarMap sont vraiment très pratiques. Tout d'abord une capture d'écran de l'affichage matriciel permet d'identifier les relations entre les différents JAR de l'EAR, afin de pouvoir ensuite factoriser le code et réduire les dépendances entre les différents modules. [caption id="attachment\_1287" align="alignnone" width="271" caption=""]

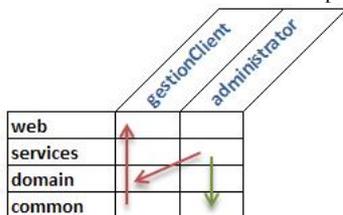


Chaque colonne et chaque ligne contient les JAR packagés dans mon EAR. Chaque cellule de la matrice indique le nombre de dépendance. Il est très facile de naviguer afin de "descendre" dans un JAR pour identifier à la ligne de code prêt la dépendance. Grâce à cet outil j'ai tout de suite vu qu'un module X n'utilise pas le module Y, mais qu'il dépendait à 54% d'un module Z. J'ai donc rassemblé des modules qui n'ont pas lieu d'être individuellement. Une originalité et une force du logiciel XDepend c'est aussi le langage CQL (Code Query Language). Grâce à ces requêtes il est possible de trouver facilement tout un ensemble de metrics, et de visualiser le résultat sur une cartographie de votre application. Dans l'exemple ci-dessous j'ai demandé à XDepend de m'afficher l'ensemble des méthodes de mon application dont la taille dépasse les 70 lignes de codes et qui sont public :



Un point plus

difficile à traiter avec XDepend est l'isolation d'architecture basée sur les noms des packages. C'est un concept peu utilisé mais bien pratique, qui revient à dire que le nom d'un package permet de définir un domaine fonctionnel, le nom d'un sous-package permet ensuite de donner la couche technique. Voici comment cela fonctionne : imaginons une application J2EE constituée d'une partie "GestionClient" et d'une partie "Administrateur" pour faire simple. Représentons verticalement ces tranches, et représentons horizontalement le découpage de l'application. Par exemple la couche Web, ensuite la partie service, le domaine et enfin une couche de fondation pour les classes



<http://touilleur-express.fr>

utilitaires. Voici ce que cela donnerait L'idée de la gestion de l'isolation par package consiste à dire que chaque cellule de la matrice correspond à un pattern de packages Java. Par exemple \*.dto pour le domaine, \*.web.\* pour la couche web, etc. Les flèches rouges représentent les interdictions. Il ne sera pas autorisé d'accéder à la couche Web à partir d'une classe du package common. Et ce, afin d'éviter bien entendu les dépendances cycliques. De même, on pourrait imaginer que les classes de la partie "Services" du domaine fonctionnel "Administrateur" ne doivent pas dépendre de classes de "Domain/Gestion client", etc. Avec Xdepend sous réserve de bien connaître les noms de vos packages, vous pouvez créer des requêtes CQL afin de vérifier ce genre de règles :

```
WARN IF Count > 0 IN SELECT PACKAGES
WHERE IsDirectlyUsing "web" AND NameIs "common"
```

Quelques bémols cependant : difficile de s'y retrouver lorsque les noms des packages sont "artistiques". De plus, si je prends 5 jours pour organiser le code, déplacer des classes, m'assurer du respect de mon schéma d'architecture, **je n'ai pas envie de refaire la même chose dans 3 mois**. Il faut donc une solution qui renforce l'architecture et le découpage au moment de l'écriture du code. Du côté d'IDEA Intellij, il n'y a pas de solutions de ce type. Seul le respect de bonnes pratiques, l'isolation dans l'usine de construction de code et l'analyse avec XDepend m'assure que le code restera propre. Bref des solutions a posteriori qui n'empêche pas un nouvel arrivant dans l'équipe d'importer une classe d'un lointain package par erreur... **SonarJ** Il est temps de vous (re)parler de SonarJ d'hello2morrow. J'avais rencontré l'équipe de SonarJ [il y a presque un an](#). L'outil est séduisant car il s'intègre à Eclipse afin de vous notifier des violations d'architecture au moment même où vous tentez d'importer une classe. Il travaille aussi sur un modèle matriciel. Le découpage est plus facile qu'avec XDepend car SonarJ vous laisse décider pour chaque slice et chaque layer des packages à utiliser. Un système d'expression régulière permet de ranger dans la couche "gestionClient/Web" l'ensemble du code existant sans être obligé de changer votre code. Or le produit sur lequel je travaille en ce moment est dans ce cas. Certaines parties sont correctement découpées, le rangement par domaine puis par couche technique a été mis en place. Pour des parties plus anciennes ce n'est pas le cas. SonarJ serait donc un complément, après avoir défriché le code avec XDepend. J'ai par contre trouvé qu'XDepend était plus puissant et plus rapide, et qu'il me permettait de diagnostiquer plus facilement l'existant, sans passer par une étape de classement un peu longue avec SonarJ. Bref les 2 outils sont complémentaires. A propos des metrics d'XDepend, des requêtes types permettent de gagner du temps. Par exemple, quelles sont les méthodes private qui ne sont jamais appelées dans le code ? // Potentially unused methods

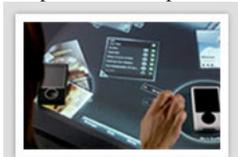
```
WARN IF Count > 0 IN SELECT TOP 10 METHODS WHERE MethodCa == 0 AND // Ca=0 -> No
Afférent Coupling -> The method is not used in the context of this application.
!IsPublic AND // Public methods might be used by client applications of your
jars. !IsClassConstructor // The IL code never explicitly calls class
constructors. Maven2 Un dernier mot aussi sur Maven2 qui propose 2 outils assez pratiques,
que j'utilise en complément des outils cités ci-dessus. Le plugin "dependency" permet d'afficher la
liste des dépendances directes et indirectes d'un module, et d'analyser l'usage ou l'oubli de
dépendances. Tout d'abord l'affichage de l'arbre des dépendances d'un module :
nicolas@macbookpro> mvn dependency:tree
C:\workarea\development\primeweb\projects\risk\services-ejb>mvn dependency:tree
[INFO] Scanning for projects... [INFO] -----
[INFO] Building org.touilleur.article [INFO]
task-segment: [dependency:tree] [INFO] -----
[INFO] [dependency:tree] [INFO]
org.touilleur.article:ejb:0-DEV [INFO] +- org.touilleur.module10:jar:0-
DEV:compile [INFO] | +- org.ostermiller:utils:jar:1.05.00:compile [INFO] | +-
commons-pool:commons-pool:jar:1.0.1:compile [INFO] | +-
jdbcappender:jdbcappender:jar:2.1.01:compile [INFO] | \-
org.springframework:spring:jar:2.5.5:compile [INFO] +-
org.touilleur.module20:jar:0-DEV:compile [INFO] | +- cglib:cglib-
nodep:jar:2.1.3:compile [INFO] | +- asm:asm:jar:2.2.3:compile [INFO] | +-
asm:asm-commons:jar:2.2.3:compile [INFO] | | \- asm:asm-tree:jar:2.2.3:compile
[INFO] | \- asm:asm-util:jar:2.2.3:compile ... [INFO] +-
hsqldb:hsqldb:jar:1.8.0.4:test [INFO] +- dbunit:dbunit:jar:2.1:test [INFO] +-
com.mockrunner:mockrunner:jar:0.3.1:test [INFO] +- j2ee:j2ee:jar:1.4:provided
[INFO] -----
[INFO] BUILD SUCCESSFUL [INFO] -----
[INFO] Total time: 30 seconds [INFO] Finished at: Thu
May 21 23:30:50 CEST 2009 [INFO] Final Memory: 16M/127M [INFO] -----
Mon module "article" dépend
du module10 et du module20. Ensuite l'analyse des dépendances se base sur le code compilé :
C:\workarea>mvn dependency:analyze [INFO] Scanning for projects... [INFO]
Searching repository for plugin with prefix: 'dependency'. [INFO] -----
[INFO] Building
MonGrosProjet-core [INFO] task-segment: [dependency:analyze] [INFO] -----
[INFO] Preparing
dependency:analyze [INFO] [resources:resources] [INFO] Using default encoding to
copy filtered resources. [INFO] [compiler:compile] [INFO] Nothing to compile -
all classes are up to date [INFO] [resources:testResources] [INFO] Using default
encoding to copy filtered resources. [INFO] [compiler:testCompile] [INFO]
Nothing to compile - all classes are up to date [INFO] [dependency:analyze]
[WARNING] Unused declared dependencies found: [WARNING]
org.subethamail:subethasmtplib:jar:1.2:test [WARNING]
org.subethamail:subethasmtplib-wiser:jar:1.2-javal.4:test [WARNING]
javax.mail:mail:jar:1.4:test [WARNING] com.sun.xml.bind:jaxb-
impl:jar:2.1.9:compile [INFO] -----
[INFO] BUILD SUCCESSFUL [INFO] -----
[INFO] Total time: 16 seconds
[INFO] Finished at: Fri May 22 10:50:07 CEST 2009 [INFO] Final Memory: 14M/127M
[INFO] -----
```

L'analyse permet d'identifier les dépendances non déclarées dans le pom.xml mais pourtant bien utilisés par le bytecode, ainsi que d'afficher les dépendances inutiles. Attention cependant à ne pas couper trop vite vos dépendances, j'ai vu des cas mal gérés par le plugin, ou aussi le souci des tests d'intégration déclarés dans d'autres répertoires. **Conclusion** Le mot de la fin : dans notre métier une partie de notre activité est purement de l'innovation. Un peu comme descendre une piste de ski de neige fraîche, c'est le bonheur. Mais il y a aussi un besoin de compétences pour trouver rapidement des solutions d'architecture afin de réduire la dette technique, la complexité du code et améliorer la qualité globale. Il est donc important d'investir un peu de son temps pour apprendre à utiliser ces nouveaux outils. La liste des outils s'agrandit de jour en jour, je vous parlais il y a quelques semaines de [Kalistick](#), Cédric Vidal un lecteur via Twitter m'a proposé de regarder aussi [Lattix DSM](#), n'hésitez pas à parler dans les commentaires ci-dessous des outils que vous utilisez sur le terrain. De mon côté j'hésite à acheter XDepend pour Innoteria afin de l'utiliser avec mon client, ou voir s'il serait intéressé pour l'acheter. M'est avis que j'aurai aussi vite fait de l'acheter... Mais c'est une autre histoire. Article sur XDepend, SonarJ, et mon retour sur expérience en ce

moment sur l'architecture

## USI (Université du SI) : des petits nouveaux et quelques nouvelles

Le programme complet des [Universités du SI](#) est en ligne. Les 4 thèmes couvriront [la Technologie](#) bien entendu, mais aussi [la Gouvernance](#) afin de maximiser le capital humain de l'entreprise, [la Méthodologie](#) avec l'émergence des méthodes Agiles et [l'Usabilité](#) qui remet le client et l'utilisateur final au premier plan. Parmi les acteurs connus de la communauté Java, [Tugdual Grall](#) d'eXoPlatform et [Vincent Massol](#) d'XWiki présenteront le positionnement des Content Management System par rapport aux Wikis. Les 2 outils font partis de l'écosystème de l'entreprise, sont-ils complémentaires ? Quelles sont les différences et les points communs des 2 solutions ? J'en profite aussi pour rappeler que Vincent fait parti du podcast francophone sur Java, [les Cast Codeurs](#) dont



j'ai fait le logo il y a quelques semaines. Autre nouveauté au programme : [les Espaces Usabilité](#). L'idée sera de présenter les technologies du futur comme la présentation de la table interactive Microsoft Surface, un système pour suivre l'œil de l'internaute entrain de visiter un site Internet, des démonstrations de Google StreetView, de Microsoft Silverlight et d'Adobe Flex. Je trouve que l'idée est vraiment intéressante, en complément des conférences. Voici une vidéo de la table Microsoft Surface trouvée sur YouTube :

Cela donne une idée assez précise de ce qui arrivera dans un futur proche. Quant à moi je vous donne rendez-vous le 1er et le 2 juillet, retransmission en direct sur le Touilleur Express, partenaire blogueur de l'USI.

## Agenda le Touilleur Express

Voilà une nouvelle idée : si vous souhaitez être informé des événements de la communauté Java en France, ainsi que des conférences comme Jazon ou l'USI, je vous propose de vous inscrire à mon agenda public. Vous serez ainsi tenu informé de la publication des articles du Touilleur Express, des soirées Java, des soirées Scrum, de l'actualité des JUG en France, bref de tout ce qui se passe dans la communauté Java et Agiles en France.

### Le Touilleur Express, Evenements Java

[Aujourd'hui](#) **dimanche, 14 février** Imprimer [Semaine](#) [Mois](#) [Mon plan](#)

Événements après le 14/2. [Rechercher des événements avant cette date](#)

Événements jusqu'au 15/3. [Rechercher des événements après cette date](#)

Événements affichés dans le fuseau horaire : Paris



L'agenda sera accessible sur sa page dédiée par la suite. La fonction d'Agenda de Google permet aussi de préciser la localisation de l'événement, de donner quelques détails, et aussi de vous tenir informé par SMS lorsqu'un événement est ajouté à l'agenda. Qu'en pensez-vous ? Quels types d'événements souhaiteriez-vous ? J'ai pensé à ajouter les codings dojo du lundi, il manque les Valtech Days, Devovx, les événements des JUG en région. Est-ce que l'agenda doit se concentrer uniquement sur les événements parisiens où je peux venir, ou faut-il couvrir l'ensemble des événements ? Nous

verrons à l'usage à quoi sert cet agenda, je souhaite en faire un endroit pour que chacun puisse venir proposer de l'information. Si vous souhaitez ajouter un événement, n'hésitez pas à envoyer un email. Je travaille sur une nouvelle solution qui permettra à chacun de proposer un événement sur Java ou les méthodes Agiles, afin que l'agenda soit le plus complet possible.

## XP Day France 2009 - journée 1

J'ai participé cette année à [XP Day France 2009](#). Organisé par Sandrine Olivencia, Yannick Ameur, Sébastien Douche et Thibaut Bouchette pendant presque un an, avec le support et l'aide de l'association Agile France, cet événement est l'occasion de réunir 250 personnes pendant 2 jours sur le thème de l'Agilité. Tout d'abord côté organisation, un grand bravo. Le cadre était un espace de conférence en plein bois de Vincennes, avec des repas au top, environ 3h par jour de temps libre pour discuter entre chaque conférence, bref l'occasion de lier connaissance, de partager son expérience et d'assister à de nombreuses conférences. Le nom XP Day tout d'abord, sera peut-être Agile Day l'an prochain. Car à part XP, nous avons parlé Scrum, Kanban, Lean (qui se défend d'être Agile si j'ai bien compris...) ainsi que valeurs humaines, hédonismes, jeux d'entreprises, etc. C'est donc le sujet de l'Agile en général qui était traité pendant ces 2 jours. **Première session : "Soigner sa schizophrénie, projet MOA/MOE voyage autour des exigences fonctionnelles exécutables"** Objectif de cette session: présenter des outils pour rendre moins flou la discussion entre la MOA et la MOE. Dresser un état de l'art dans le domaine des tests fonctionnels. Herve Lourdin d'ECTO Technologies, Rémy Sanlanville d'Orange Business, Emmanuel Hugonnet de SilverPeas axent la présentation sur la définition de l'environnement entre le métier d'une part et la réalisation d'autre part. MOA pour le métier, les idées et le contexte. Pour la MOE, notre vision est plutôt axée sur la réalisation, l'architecture, la définition du done se fait avec les tests unitaires. Comment rejoindre ces 2 hémisphères et améliorer la communication ? La clé est tout d'abord la communication. Pour cela, chacun doit s'attacher à définir la notion de "TERMINE" car c'est le contrat et l'engagement qui permettent la réalisation du développement. Emmanuel appuie la démonstration par un graphique montrant que presque 43% des fonctionnalités d'un logiciel ne sont pas utilisées. Nous sommes face à un réel problème. La communication entre le métier et la réalisation s'est axée aujourd'hui sur un contrat, les fameuses spécifications. Or un contrat ne doit pas être là pour remplacer une communication forcément nécessaire. Le développement par l'acceptation des tests (Acceptance Test Driven Development) est une technique qui permet de faire émerger un discours, un langage commun compris des développeurs et du métier. La création d'un mini DSL, l'écriture par le formalisme "Given... When... Then..." permet par exemple de se centrer sur la réalisation et l'écriture des tests d'acceptation. Faire de l'ATDD est donc un moyen de bien faire ce que je dois faire, les TDD étant un moyen de bien faire les choses. Quelques outils cités : [JBehave](#), [Fitness Framework](#) ou encore [RSpec](#). Présentation très sympa, j'ai pris ensuite plaisir à écouter Eric, Hervé, Rémy et Emmanuel durant le cocktail échanger leurs points de vues sur les outils, sur le métier du testeur, la perception des équipes d'assurance qualité dans l'entreprise, la phrase d'une dame présente qui lâche mollement une énormité comme "dans Lean les tests unitaires sont du gachi", bref je me suis bien amusé et j'ai compris la démarche, l'intérêt de travailler par la définition de l'acceptance, plutôt que sur la définition du contrat initial. Voici mon ROTI (Return On Time Invested) Qualité de la présentation : 4/5 Ce que j'ai appris : 4/5 **2) Le Planning Game(détails)** Premier atelier pratique pour moi. Dans une salle de 40 personnes, l'objectif était pour François Wauquier de SFEIR de présenter la cérémonie du Planning Game, de nous faire réaliser un exercice de planification avec un jeu de rôle, afin de nous faire prendre conscience de l'importance de cette cérémonie (XP ou Scrum). Le public présent dans mon groupe était constitué de gens de la MOA, peu au fait de ces techniques. Après la présentation détaillée de la séance de Planning, nous voilà découper en plusieurs groupes. Objectif pour notre équipe : jouer le rôle d'une agence de voyage qui doit organiser une colonie de vacances. L'un des acteurs joue le rôle du client. Au bout de quelques minutes je me rends compte que personne ne maîtrise franchement le sujet. Quelqu'un tente l'explication d'une User Story, un autre essaye de comprendre à quoi servent ces cartes (Planning Poker). Avec l'aide de Jean-Laurent de Morlhon de Vidal, nous voilà finalement entrain d'expliquer rapidement le jeu du planning poker. A cet instant précis je me dis que j'aurai dû partir. Mais finalement, la magie va opérer (j'en rajoute un peu là). Je pense à cet instant que je perds mon temps. Puis surgissent les questions des 6-8 personnes présentes. Tout d'un coup nous voilà entrain de trouver les mots pour expliquer le fonctionnement du jeu, pourquoi nous utilisons une unité arbitraire, etc. François vient dans notre groupe et recadre très professionnellement l'activité, tout en nous laissant faire notre propre expérimentation. Bref finalement j'ai appris de cette session à expliquer simplement le Planning Poker et les unités de planification, à des personnes du métier. Intéressant et enrichissant. Qualité de la présentation : 3/5 Ce que j'ai appris : 2/5 **Pause déjeuner** Rupture dans la journée, le moment du repas est l'occasion de lier connaissance. XP Day a une très bonne idée : organiser un repas assis à des tables rondes de 12 personnes. C'est donc le moyen de discuter librement sur la matinée. Je reconnais 2 têtes connues : David Gageot de [Tech4Quant](#) et Jean-Laurent de Morlhon de Vidal, tous les deux anciens collaborateurs de Valtech. Quelques instants après m'être assis, je me rends compte que je suis à une table de Valtech, actuels ou anciens salariés. C'est l'occasion de discuter, de faire connaissance avec Xavier Warzee de Microsoft, membre du bureau du French SUG comme moi. Là je vous sens sceptique, vous vous dites c'est quoi cette page People ? Ah mais mon bon monsieur sur le Touilleur Express on a aussi nos pages People afin de raconter un peu qui fait quoi, cela fait vendre plus de numéros ! Bref bon repas. **3) Travailler entre Adultes(détail)** Retour et envie d'aller écouter Raphaël Derbier. Tant pis je loupe le show de Régis Médina, mais je suis venu chercher des réponses à mes questions. Raphaël nous propose de nous faire prendre conscience de notre propre comportement dans l'entreprise. Une comparaison ludique entre notre monde d'adulte et le monde des enfants va nous permettre de nous rendre compte de notre propre comportement. Tout d'abord l'idée que l'entreprise est notre mère nourricière : "Ah mais moi je vais pas acheter un livre, c'est à l'entreprise de me l'acheter" ou encore "Moi je veux une formation pour être manager" et aussi le très pratique "On a pas le droit de le faire alors je le fais pas". Bref voici ce que je note sur mes feuilles : un enfant n'hésite pas à demander du feedback : "Papa, il est beau mon dessin ?" sans la peur d'être jugé. Un enfant utilise le jeu pour apprendre. Raphaël

explique que si des équipes d'architecture prennent le temps avec un jeu de représenter le SI, cela permet de mettre en avant les problèmes techniques très rapidement, sans perdre 3 mois pour se rendre compte que 2 équipes ne se parlent pas... J'évoque le jeu LEGO Serious Game qui est licencié officiellement et vendu pour gérer le conflit et améliorer la communication dans les équipes. D'ailleurs le lendemain j'aurai une discussion avec Isabelle de Pyxis sur l'utilisation des Legos pour faire un Burndown Chart. Elle ne connaissait pas cette idée. Raphaël explique tout d'abord qu'être adulte s'est "Prendre soin de soi". Et un enfant est par définition un être qui n'est pas capable de prendre soin de lui, au sens autonomie. Etre adulte c'est donc faire un pas vers l'autonomie. Par des jeux de 5mn il nous force à échanger avec nos voisins. "Quelle est la dernière occasion où tu as pris soin de toi ?". Intéressant, car cela fait émerger aussi le besoin fondamental d'être en groupe. Il parle d'intelligence collective, de l'importance d'être acteur et d'être présent. Au niveau des règles il explique aussi que les enfants utilisent les règles comme socle de départ mais s'autorisent la transgression ou la tricherie, dans le but de s'adapter à l'environnement. Si votre question commence par "Est-ce que je peux..." alors la réponse sera oui. Cela nous fait gagner du temps, s'autoriser à ne pas suivre la règle s'est faire travailler [son système limbique](#) en utilisant la palette de nos émotions : la peur, le plaisir, la colère, etc. Difficile de vous résumer l'excellente présentation de Raphaël, qui m'a éclairé sur quelques points, et qui m'a donné de nouveaux outils pour former des équipes à la méthode Scrum. Je constaterai le lendemain à la session "Scrum est-il dangereux" que certains d'entre nous sont de grands enfants, mais je réserve cela pour demain.

**ROTI Qualité de la présentation : 4,5/5 Ce que j'ai appris : 4/5 4) Push Pull réaliser sans itérations (détail)** Yan Picard de Muller nous propose d'expérimenter en petits groupes du Kanban. Lorsque le mode purement itératif ne répond pas à la demande, [il y a d'autres alternatives](#) dont j'ai déjà parlé sur le blog le Touilleur Express. La production pilotée par la consommation est un principe passionnant qui nous vient de Lean. Yan organise chaque groupe comme une Agence Web en charge de la réalisation d'un site Internet. Il explique le fonctionnement et nous propose d'expérimenter tout de suite le fonctionnement. J'ai senti qu'il connaissait son sujet, j'ai eu du mal à comprendre par contre ensuite comment nous devons faire pour réaliser le jeu. J'ai repensé à mon enfance, lorsqu'un cousin vous expliquait la règle du Monopoly, et qu'au départ vous aviez du mal à comprendre. Bref nous voilà dans la situation où comme des enfants, nous n'arrivons pas à tout saisir. Finalement, après quelques essais le jeu se met en place sans trop de soucis. Nous déroulons quelques cycles, avec des lancés de dés comme 1D6+4 qui nous permettent de jouer un rôle. Je dois jouer le rôle de l'architecte de l'information, marrant. L'exercice a été difficile car nous étions trop nombreux. A retester en petit groupe je pense, car sinon côté organisation c'était très bien. Je dis cela car lorsque vous verrez le ROTI, je ne souhaite pas blesser Yan, je veux donner mon sentiment très personnel tout en m'autorisant à dire "je n'ai pas trouvé cela top POUR MOI" uniquement sans porter un jugement de valeur ROTI Qualité de la présentation : 2,5/5 Ce que j'ai appris : 2/5 5) **La parabole du trafic urbain : l'Agilité expliquée autrement (détail)** Tout d'abord si vous lisez le blog souvent, vous savez que je suis un grand fan des paraboles, métaphores et autres images, qui me permettent d'expliquer des concepts avec des mots simples. François Bachmann est Suisse, Coach Agile, et un gars très sympa avec qui j'ai diné ensuite lors du grand diner XP Day. François voici tout d'abord [l'article sur la machine à laver du Touilleur Express](#), histoire que j'ai rejoué lors de la présentation de Scrum au Paris JUG avec Eric "Bob" Mignot. François nous propose de nous présenter quelques images simples afin d'établir un parallèle entre les méthodes Agiles et le trafic urbain. Son objectif est de nous donner quelques phrases, quelques repères pour expliquer à nos clients et à nos managers les grands principes de l'Agilité. Tout d'abord observons les différents trafics. Sur une photo nous voyons une rue de New Dehli encombrée de piétons, de voitures, une anarchie apparente pour nous avec notre oeil d'européen. Ensuite en Allemagne, à un péage, le flux de voiture qui se divise pour passer la barrière de péage. Au Japon, où des indicateurs sur la route donnent les temps de parcours. Bref en quelques slides nous voyons où il veut nous emmener. Il y a tout d'abord différents types de trafic : - Habituel - A valeur ajoutée - Urgence Le trafic habituel est constitué par les gens qui vont travailler par exemple. Le risque est faible, nous acceptons du retard et ce processus s'optimise en travaillant sur la disponibilité. Vient ensuite le trafic à valeur ajoutée : par exemple un bus ou un taxi dans sa voie de circulation. C'est la fiabilité qui est alors important, il ne faut pas perdre des clients en cours de route. Alors on met en place des routes spécialisées. Vient enfin les urgences, les pompiers par exemple. Sur l'autoroute, une voie de secours permet de faire passer devant les autres ces véhicules car là le risque vital est là, c'est la vitesse moyenne qu'il faut optimiser. Personnellement durant son explication je visualise mon SI chez mon client actuel plutôt que les méthodes Agiles. François nous fait ensuite prendre conscience des méthodes de gestion du trafic. On utilise traditionnellement le mode "Command and Control". Un feu rouge permet d'arrêter le flux de voitures. Dans les méthodes classiques de développement, la spécification figée est un feu rouge/feu vert qui permet de réguler le flux. Sur un aéroport, une tour de contrôle retire le contrôle aux avions et se charge d'aiguiller les avions, car la sécurité prime avant tout. Intéressant non ? Certains projets ont peut-être besoin d'une tour de contrôle centralisée, d'autres non. C'est ce que je déduis à cet instant de la présentation. Ce feu rouge justement. Comment se règle les temps d'allumage ? Par statistiques ? par observation des flux ? On est en plein dans la théorie des fluides. Le débit se calcule avec le diamètre du tuyau, mais aussi sa longueur... Intéressant. Je te fais remarquer cher lecteur que je suis en plein pédalage d'idées avec une métaphore à la seconde... Que se passe-t-il en cas d'accident ? Sur autoroute, une voie de secours permet de s'arrêter en sécurité et de quitter le véhicule en panne, sans pénaliser le trafic derrière nous. Encore une fois j'ai un peu de mal à voir le rapprochement avec Agile. Je pense que dans Scrum, cela s'apparente à la gestion d'un obstacle, identifié pendant le standup meeting du matin. On parque le problème afin de l'adresser sans arrêter la circulation pour autant. Oui il a raison... ça marche et j'ai trouvé tout seul ! François parle encore du feu rouge et des intersections. Il nous dit : regardez ce qu'est qu'un rond-point (un giratoire). C'est un système auto-organisé qui fait appel au courage du conducteur : "je dois prendre la bonne décision". Le giratoire est un système qui résoud l'encombrement avec des règles simples. C'est le passage d'un mode Command-Control vers un mode autorégulé... plus Agile ! La communication (le clignotant), la gestion de l'engagement (j'y vais à fond ou pas), l'absence de préselection des flux (priorité à ceux qui sont entrain de tourner) le respect de l'autre, la transparence... Oui tout y est ou presque. Cette séance sera l'occasion d'ajouter à mon panier de connaissance la parabole du trafic routier. Cependant j'ai plus de facilité à rapprocher cela de mes

besoins en architecture, que sur les méthodes Agile. Je préfère l'analogie avec le monde du Sport pour expliquer Scrum rapidement. Mais la présentation était très intéressante ROTI Qualité de la présentation : 4,5/5 Ce que j'ai appris : 3,5/5 **Cocktail et dîner, le moment Off** J'avais décidé de rester dîner, ce qui m'a donné l'occasion de présenter Scrum à Cédric Vidal de [ProxiAD](#) avec justement l'aide de François Bachmann de Sprint IT. Les moments forts d'XP Day sont donc les conférences mais aussi les moments off qui permettent de discuter avec chacun. On retrouve les habitués des soirées Java et des BarCamps, mais j'ai noté une bonne représentation des personnes du métier, de la MOA. Je me demande si un effet bénéfique de Scrum n'est pas d'apporter aux gens du métier des réponses plus pragmatiques qu'une autre méthode. Est-ce que regarder le produit et le placer au centre n'est pas plus simple pour eux que de parler de techniques d'ingénierie logicielle ? Qui a discuté avec des personnes de la MOA durant les XP Day ? Vous saurez tout en revenant lire plus tard la suite... Retour sur ma première journée à la conférence XP Day 2009 France.

## XP Day France 2009 - jour 2

Après une première journée riche en contenu, j'ai axé ma deuxième journée sur le jeu. Voici le programme de ma journée : - [Rétrospective : l'amélioration continue apprivoisée !](#) (Jacques Couvreur et François Bachmann) - [Scrum est-il dangereux ?](#) (Eric Lefevre-Ardant et Guillaume Tardif) - [Dans la peau du challengeur : remettre en question l'ordre établi](#) (Luc Bizeul et Emmanuel Gaillot) - [Le "Business Value Game" trois itérations dans la peau du Product Owner](#) (Pascal Van Cauwenberghe, Portia Tung et Laurent Morisseau (?)) **Rétrospective : l'amélioration continue apprivoisée !** La rétrospective est une technique qui peut être utilisée en dehors de Scrum ou d'XP. Par exemple après un incident, un "post-mortem" est une forme de rétrospective, la fin d'un projet est une rétrospective, etc. La présentation s'appuie entre autre sur le livre "[Agile Retrospective : making good teams great](#)" d'Esther Derby, Diana Larsen et Ken Schwaber ([voir aussi sur le site du French.SUG](#)). La rétrospective s'effectue en 5 temps : - planter le décor (qui fait quoi ? quel est votre rôle ? pourquoi êtes vous là ?) - collecter les données (nombre d'incidents, mail d'un client externe, logs, fiches bugs, etc) - réaliser un aperçu général afin d'extraire de la donnée utile - décider d'un plan d'action et d'amélioration - conclure et s'assurer que la rétrospective s'est bien terminée Pour nous apprendre à conduire une rétrospective pendant une heure, François et Jacques nous proposent un jeu de rôle disponible sur leur site internet [l'Alchimiste Agile](#). Notre mission sera de transformer le plomb en or, de transformer notre projet qui ne s'observe pas en une mine d'or pour s'améliorer. Je retrouve encore François et ses images, j'adore. Bref nous nous mettons par groupe de 10 personnes. Chacun prend une carte, notre mission sera de faire une rétrospective de Noël 2008. Je serai un nain, l'équivalent du développeur. Mon voisin sera un Elfe, en charge de la qualité. Ma voisine sera un parent, il y a aussi des rôles du Père Noël, des enfants capricieux, d'autres Nains, bref une sacrée ambiance. En main j'ai ma carte de rôle et 2 cartes de comportement. Noël s'est mal passé car j'ai donné avec beaucoup de retard ma commande de bois au service des achats des Elfes. J'ai eu aussi à gérer le retour d'un lot de poupée qui ont pris feu, car je les ai mal fabriqués. Chacun a donc un rôle et des incidents qui se sont déroulés. Un enfant très capricieux en face de moi râle car son camion de pompier... bleu n'est pas arrivé, joué par Charles Gay. Bonne ambiance générale. Le tout est animé par Portia Tung, notre père Noël venue de Londres, enfin plutôt madame Noël. Première phase de la rétrospective : planter le décor. Chacun parle à son tour en jouant le rôle. Il faut rendre compte de ses soucis, sans attaquer personnellement quelqu'un mais simplement en travaillant sur les faits. Pour réaliser cela, chacun marque sur un post-it un événement avec la date de cet événement. Nous regroupons les post-its sur une ligne de temps sur un tableau. Cela permet de montrer à tout le monde ce qui s'est passé. La deuxième phase d'évaluation des données doit nous permettre d'extraire ensemble de la donnée. Nous utilisons quelques techniques comme le "5 Why" où vous demandez 5 fois pourquoi ceci à quelqu'un qui a réussi, ou qui a eu un souci. Attention une rétrospective c'est aussi **pour parler de ce qui a réussi** afin de continuer à réussir avec les autres. Nous utilisons des votes pour que les points qui nous semblent les plus importants soient mis en avant. Au final 2 éléments sortent du tableau. Dans la phase suivante nous devons trouver une solution ou un moyen de continuer une bonne pratique. Chaque joueur doit proposer 2 actions à partir des 2 éléments que nous avons sélectionné au coup précédent. 10 personnes, 20 idées sur un tableau. Là encore nous votons, et nous conserverons 3 idées au final (je crois). L'intérêt aussi est de voir les idées des autres, et de créer de la communication. Nous décidons ensemble des solutions, nous nous focalisons sur la prochaine itération (noël prochain) Enfin nous terminons par un dernier tour de salle pour que chacun puisse partager son sentiment, ses émotions en utilisant un canevas pour s'exprimer. Noël 2009 devrait se passer mieux que Noël 2008, chacun est reparti avec des idées et des actions afin d'être plus performant et de continuer à s'amuser la prochaine fois. **Débriefing** Excellent exercice, qui permet de comprendre le principe de la Rétrospective. J'utiliserai ce moyen pour améliorer ma rétrospective la prochaine fois au boulot. Les outils cités vont aussi m'aider à améliorer la phase de rétrospective. Je retiens aussi l'idée de la gestion du post-mortem en cas de problèmes. Il est important de rappeler l'importance de réunir les gens. Mais si vous mettez dans une salle des gens avec des tensions, vous pouvez demander aux personnes présentes de voter de manière anonyme en répondant à cette question - Dans cette réunion rétrospective quel sera votre rôle selon vous ? [Explorer](#), [Shopper](#), [Vacationner](#) ou [Prisonner](#) ? L'Explorateur a envie de découvrir et de comprendre Le Shopper est là dans l'envie de repartir avec au moins une bonne idée Le Vacationner est content d'être là mais n'a pas trop envie de participer Le Prisonner a été forcé de venir contre son gré Réaliser un vote anonyme, cela vous donnera alors une idée de votre public pour conduire la rétrospective. En conclusion j'ai beaucoup appris, la séance de jeu a été vraiment bien menée par les 2 animateurs. ROTI : Qualité de la présentation : 4,5/5 Ce que j'ai appris : 5/5 Il est temps de continuer et d'aller parler Scrum maintenant... tout un programme ! Je fais une petite pause et je reviens après la publicité.

## XP Day France 2009, Scrum est-il dangereux ?



Crédit photo: Jean-Laurent de Morlhon

Suite de [mon billet précédent](#) à propos d'XP Day. J'ai assisté au débat ouvert "Scrum est-il dangereux" animé par [Eric Lefevre-Ardant](#) et Guillaume Tardif. L'objectif de l'heure sera de lister tous les points qui répondent à la question "Oui Scrum est dangereux" puis ensuite de lister tous les points "Non Scrum est une bonne chose" et enfin réaliser une synthèse pour confronter nos idées. J'ai aussi enregistré le tout, comme Raphaël de Pyxis. Vous pouvez écouter la version audio [sur le Podcast "Le Touilleur Express"](#) sur iTunes. Tout d'abord plantons le décor. Dans une salle de 50 personnes, nous nous asseyons en U, chacun a pris à l'entrée soit une carte verte "Non scrum est une bonne chose" soit une carte rouge "Oui Scrum est dangereux". Je décide de prendre une carte verte. Un paper-board permet de noter les idées et les points soulevés. Eric et Guillaume seront simplement là comme des arbitres afin que le passage de parole soit respecté. Dans l'assemblée je retrouve toutes les têtes connues de la communauté Agile avec tout d'abord une présence remarquée de Pyxis Technologies : [Eric Mignot](#), Vincent Tencé, Isabelle Therrien et Raphaël Pierquin. David Gageot de Tech4Quant, Jean-Laurent de Morlhon, [Emmanuel Gaillot](#) d'OCTO Technology qui dira en préambule "je pense que Scrum est un mot", Yannick Ameur et Sébastien Douche, co-organisateur d'XP Day, bref vous l'aurez compris : pratiquement que des personnes avec une bonne connaissance de Scrum. Voilà pour le décor. Eric explique que cette session est librement inspirée de la session "[Is Scrum Evil](#)" animé par Jeffrey Frerick. Elle sera rejouée en septembre à la conférence [CITCON 2009](#) à Paris le 18 et le 19 septembre, conférence déjà complète co-organisée par Eric. Il n'y aura pas de débat, nous commençons par exprimer une idée sur "Oui Scrum est dangereux parce que..." appuyé d'arguments afin que le débat soit construit. Quelques suggestions plus tard, Vincent de Pyxis a raison de demander si nous voulons parler du framework Scrum en tant que tel ou de son application. Or il apparaît rapidement que les points levés par les participants se focalisent d'abord sur l'effet Scrum, sur son application et ses effets. Il n'y a pas pour l'instant de critiques sur le contenu de Scrum en tant que tel. Quelqu'un se risque à dire "Scrum est dangereux car il n'y a pas de pratiques logicielles comme la méthode ChlingChling". Ce à quoi chacun remet Scrum sur son terrain, et on évacue rapidement la question XP vs Scrum. Voici au fil de l'eau comment s'est déroulée la discussion. Ce qui suit est transcrit en essayant d'être fidèle aux propos de chacun : DEBUT DES DEBATS

## PREMIERE PARTIE: Scrum est dangereux parce que...

Quelqu'un lance le débat en disant que "Scrum est une mauvaise chose car il fragilise les équipes qui le mettent en place par rapport à d'autres, vis à vis du management". "Scrum est dangereux car lorsque l'on met en place la méthode on a pas le droit à plusieurs essais". "Le dogmatisme de Scrum est dangereux". "Scrum est dangereux car le jargon de Scrum peut nous discrediter". "Scrum est dangereux car le jeu de carte peut nous marginaliser". "Scrum est dangereux lorsqu'il est appliqué seul sans méthodes logicielles". Emmanuel Gaillot demande pour qui Scrum peut-être dangereux ? Eric "Bob" Mignot enchaîne avec "Scrum est dangereux car il révèle les incompetents dans une équipe". Il est dangereux pour les gens qui s'en servent. Scrum est dangereux car il fait partir des gens, il peut casser des équipes. Heureusement que la documentation nous sauve si quelqu'un part... \*rires\*. L'introduction des méthodes Agile est dangereux, ou amène du danger. Quelqu'un dit que le changement apporte le danger. Toujours pas pour l'instant d'argument sur le contenu de Scrum. David Gageot explique que Scrum est dangereux car certaines équipes qui ont mis en place avec succès Scrum cherchent ensuite à changer les autres équipes. L'effet viral et l'autopromotion est dangereux, surtout s'il n'est pas accompagné finalement par un coach ou quelqu'un d'assez expérimenté. La radicalisation des équipes est dangereux. Emmanuel dit que Scrum est dangereux car le chef de projet n'existe plus. Alexandre explique que Scrum est dangereux car il supprimerait certains postes de l'entreprise, et à terme il mettrait en danger l'entreprise. En retour certains expliquent que la structuration MOA/MOE actuelle vient aussi des méthodes classiques. Scrum est dangereux car il empêche l'évolution hiérarchique dans l'entreprise. Scrum est dangereux car tout repose sur les épaules du Scrum Master et du Product Owner. Scrum est dangereux car les utilisateurs adaptent la méthode à leurs pratiques, ce qui peut entrainer un échec, qui sera attribué... à Scrum. Scrum est dangereux car la responsabilité est diluée, l'engagement à terme est moins clair... On est plus sûr de rien. Tout est flexible... Emmanuel ouvre la boîte de la certification. "Elle entraine une logique qu'il y a des gens qui savent mieux que d'autres ce qui est bien et ce qui est pas bien, ce qu'il faut faire, ce qu'il ne faut pas faire. Si on fait bien ce qui est bien on est certifié, si on fait pas bien ce qui n'est pas bien on est pas certifié". Les personnes certifiées sont un danger pour l'entreprise car elles se comportent comme celles qui ont le savoir, et c'est un danger pour elles-mêmes, elles risquent de se prendre un retour cinglant de la part de l'entreprise. L'autorité de certification est dérangeante. Dans un deuxième point il exprime que la certification payante, le recrutement, nous font penser que l'on tombe dans des arguments marketings, dans une démarche mercantile. Eric Mignot demande si c'est Scrum en tant que tel ou si c'est l'industrie autour de Scrum qui est dangereux ? Isabelle demande si c'est parce que la certification scrum master est trop facile à obtenir que ce serait un danger ? Eric Mignot

replante un jalon afin que le débat continue, en nous réexpliquant Scrum, en expliquant la complémentarité avec XP. Il tente un coup de force afin que quelqu'un explique Scrum pour que la suite des débats se concentre sur le contenu. Il demande si quelqu'un pense que le Product Backlog est dangereux ? Est-ce que le Sprint est dangereux ? Est-ce que le rôle de Product Owner est dangereux ? Amusant de voir que l'assemblée finalement continue à parler, mais sans suivre la piste proposée par Eric. David Gageot joue le jeu, il dit que le planning meeting est dangereux car le product owner ne vient qu'au début et que l'on ne le reverra qu'à la rétrospective... La rétro fait remonter les problèmes mais c'est dangereux car on ne fait rien après. La démo dans Scrum est dangereux car on ne fait que la démonstration de la fonction, sans montrer le reste du logiciel, en ne passant que sur la démonstration faite au client... A cet instant, Emmanuel apporte une nouvelle image pour nous résumer quelque part notre discussion : ce ne sont pas les armes qui tuent, ce sont les gens qui s'en servent. Il explique que certains défendent Scrum en disant que ce n'est pas l'outil mais sa mise en place qui est dangereuse. Je trouve l'image courageuse et intéressante à cet instant de la discussion. Je me demande si nous partons pas vers une discussion qui se terminera par [la Loi de Godwin](#). Il demande pourquoi certains peuvent en faire un jeu de pouvoir, peuvent le tourner à leur sauce, ce qui est vrai. Vincent de Pyxis répond en demandant si finalement, quelque soit les méthodes, celles-ci nous montrent que notre industrie est mauvaise. David reparle de l'effet Matrix, de la pillule bleue et de la pillule rouge, abordé lors de [l'inauguration de la soirée du French SUG](#) par Jeff Sutherland. Scrum marcherait pour des équipes de 2 personnes comme 100 personnes... CMMI et PMI arrivent, Scrum est la solution à tout... Bref ce qui est dangereux c'est peut-être plutôt cela d'après David. Je parle peut-être pour la première fois pour dire que derrière Scrum il y a une industrie, des enjeux, que des gens ont compris que l'on pouvait faire de l'argent. De même comme une arme, il est facile de s'en servir et de faire des dégâts. XP est pour moi un lance-roquette, là où Scrum serait plus facile à introduire. Scrum n'est pas dangereux en lui-même. Le danger c'est nous même résume Vincent. Il demande comment on peut améliorer notre industrie. Quelqu'un reprend la parole pour dire que Scrum est dangereux, car il est fait pour être facilement adopté, sans peut-être parler des effets secondaires... On a l'image d'un médicament avec ses effets secondaires. Jean-Laurent dit que Scrum est physiquement dangereux, car il n'y a pas de moments pour souffler comme dans un développement waterfall classique. C'est épuisant :-). Scrum peut être dangereux car si sa mise en place ne réussit pas, il décrédibilise l'Agilité. On parle de tueur d'Agilité. Une autre idée ensuite, est que le canvas léger mais strict de Scrum empêche l'Agilité, ne laisse peut-être pas la liberté à l'amélioration. Virgile Delécolle rappelle que les projets Scrum qui ont fonctionné ne font pas que du Scrum. Scrum est dangereux car il n'y a pas d'engagements sur le périmètre. Une mauvaise équipe peut livrer peu, la médiocrité ne serait pas punie ? Quelqu'un se demande comment une entreprise qui utilise Scrum peut être pérenne, s'en sortir financièrement, se battre face à d'autres entreprises sur des appels d'offre. Scrum est dangereux car il y aurait une difficulté pour la contractualisation. Quelqu'un dit qu'il serait plus difficile d'être innovant avec du Scrum. Que l'exploratoire serait plus difficile... Et la dernière : Scrum est dangereux, car il peut être mal compris ou utilisé incorrectement. Eric Mignot dit "pour moi l'un des seuls points faibles de Scrum, c'est que c'est **extrêmement difficile à faire**". Et quelqu'un pour conclure en rigolant "... c'est pour cela qu'il y a une certification". Scrum révèle beaucoup de choses.

## FIN DE LA PREMIERE PARTIE

*Après cette première partie, nous enchaînons maintenant sur une discussion afin de lister les bonnes choses dans Scrum*

## DEBUT DE LA DEUXIEME PARTIE : Scrum est une bonne chose car...

Scrum est une bonne chose car c'est un bon moyen pour commencer. Scrum est une bonne chose car il me permet rapidement de mieux connaître mes collègues. Scrum parle aux gens qui ne sont pas développeurs (pas forcément compris). Scrum aide à impliquer les gens. Scrum aide à faire émerger les problèmes, les boulets, les planqués dans une entreprise. Scrum est simple, on ne parle pas de développement logiciel. Quelqu'un dit "quand j'ai voulu former des non-développeurs à XP, cela a été plus difficile qu'avec Scrum. Le package Scrum est plus léger" (*euh quelqu'un pour signaler que les 2 n'ont rien à voir s'il vous plaît ? merci*). Isabelle propose que Scrum est une bonne chose car il donne des mesures précises sur l'avancement, et qu'il est facile de donner des mesures de cet avancement. Quelqu'un dit "Scrum est une bonne chose car le Waterfall est en recul". David dit : Scrum est une bonne chose car il a réussi peut-être un peu mieux que XP à faire parler de l'Agilité. Virgile enchaîne: Scrum est une bonne chose car c'est plus facile de vendre des présentations XP. C'est un pied dans la porte pour parler d'XP. Vincent de Pyxis aborde alors un point intéressant : Scrum est une bonne chose, car plus on aura des implémentations de Scrum qui vont échouer, plus le monde du développement logiciel va réaliser qu'il n'est pas bon. On peut espérer alors que l'on laissera ce job à des gens dont ce sera le métier, et que l'on ne laissera pas des amateurs mettre en place avec les conséquences que l'on sait, la méthode elle-même. Il parle d'amateurisme généralisé. Il faudrait que ce soit une profession... Quelqu'un dit : Scrum permet d'échouer plus vite. Scrum est une bonne chose car il donne un cadre méthodologique à XP. Eric Mignot est une bonne chose, il favorise les comportements adultes dans les organisations. Scrum permet de montrer à d'autres équipes le résultat du travail des développeurs, grâce à la rétrospective. Emmanuel dit que Scrum ne laisse pas indifférent, il amène les gens à réfléchir sur les autres. Scrum est une bonne chose car il donne du pouvoir au développeur. Scrum est une bonne chose par rapport à XP car on peut parler facilement des principes de l'Agilité à l'entreprise, aux développeurs. Scrum est une bonne chose car il permet de rendre légitime des pratiques comme les itérations ou les sprints. Scrum est une bonne chose car il force les développeurs à lever régulièrement la tête de leur clavier, lors des rétrospectives, des réunions, des sprints meetings. Scrum c'est bien car cela évite de coder des choses inutiles. Scrum c'est bien car cela augmente la motivation des équipes. Les gens sont plus heureux. Eric Mignot explique que des gens essayent le Scrum et reprennent du plaisir à bosser. Vincent dit c'est une bonne chose car ce n'est pas une méthodologie. Donc cela ne donne pas l'illusion qu'il existerait une bonne recette pour développer du logiciel. On parle tout le temps de "méthodologie" pour Scrum, ce qui n'est pas vrai. Cela va

plutôt nous permettre de nous tourner vers nous même pour trouver la solution, plutôt que de se baser sur une méthode pour résoudre un problème. Scrum fournit des outils de communication, il permet de donner de la transparence et de l'honnêteté. Xavier dit "on jouera carte sur table". David Gageot: Scrum donne du rythme. Quelqu'un dit : Scrum est plus humain. Isabelle : Scrum donne le contrôle sur le projet au client. David explique que la notion de focus sur le "produit" est important. On est là pour développer quelque chose. Scrum est une bonne chose grâce au rôle du Product Owner. La notion de Product Backlog, de Product Owner est plus fort. Emmanuel rappelle dans XP la notion de Client qui existe déjà. Mais la notion de "responsable de produit" est plus forte dans Scrum... Scrum est une bonne chose car il permet d'exploiter mieux l'humain. Les tableaux et la transparence permettent de voir l'avancement du projet. Quelqu'un explique que Scrum est une bonne chose car il est facile de trouver de la documentation et des ressources. Les retours sur expérience. Eric "Bob" dit que Scrum est une bonne chose car il permet de retrouver le plaisir qu'il avait à coder lorsqu'il était jeune. Scrum est une bonne chose car il remet le bon sens au premier rang. Scrum est une bonne chose car il permet de créer des communautés. Yannick explique que Scrum est une bonne chose car on parle d'heure idéale, pas de 8h bête par jour. C'est une bonne chose encore une fois car cela force à nous réfléchir

## FIN DE LA DEUXIEME PARTIE

*Passons à la synthèse : Eric demande si les personnes avec une carte rouge en main veulent prendre une carte verte, ou inversement. Les débats durent depuis 1h11...*

### La synthèse

Vincent explique que Scrum est une voie pour lui. Il se demande s'il y a du dogmatisme ? il ne sait pas. David a changé, il a bien aimé entendre que les gens ont conscience que le danger de Scrum est que ce n'est pas qu'un mot, qu'il n'est pas une recette magique, qu'il est difficile, bref que les gens sont réalistes sur Scrum. Scrum n'est pas qu'une partie de l'Agilité, Jean-Laurent explique que pour l'instant il n'a pas trouvé mieux pour son besoin, dans son expérience professionnelle. En attendant peut-être de trouver mieux demain. Michel fait une petite synthèse : Scrum peut être dangereux par la radicalisation de nos ressources humaines machines outils. Sur la chaîne de production on se retrouve très exposé. C'est une méthode qui demande à la machine outil salarié de travailler en transparence, en binôme, à faire remonter toutes ses difficultés, avec une organisation derrière qui reste très traditionnelle, exploitation, rentabilité... *Note de Nicolas: soit c'est de la provocation intelligente, soit c'est de l'extrémisme qui montre une profonde méconnaissance de notre métier... Je suis effaré des propos, comme si la recherche de l'amélioration pouvait être critiquable, comme si la recherche de la performance, du plaisir de travailler intelligemment était critiquable... Monsieur Virgile je ne vous ai pas compris.* David explique qu'il y a beaucoup de business autour de Scrum. Vouloir embrasser toutes les équipes, c'est dangereux. Il y a toujours une phase exploratoire, mais nous devrions peut-être continuer à chercher, à ne pas s'arrêter à Scrum comme la solution. Il y aura sans doute d'autres méthodes, d'autres approches plus tard. Emmanuel enfin s'interroge sur la traduction de la réunion "Is scrum evil?" en anglais se traduit par diabolique, mais Eric Lefevre explique que la démarche était de construire une discussion, pas d'apporter une démarche de valeur. Il faut accepter que d'autres personnes soient différentes, qu'elles préfèrent XP, qu'elles préfèrent du Waterfall, il faut avoir une tolérance sur les autres. L'essentiel est de se sentir bien. Et pour terminer je me lance: Scrum est une bonne chose car cela va nous faire prendre conscience que dans la vie de l'entreprise, et dans le développement logiciel, notre oeil d'ingénieur qui pense avoir la vérité vraie, je pense à XP, je pense que l'on s'est planté. Et Scrum pour moi c'est un peu l'étincelle qui est entrain de nous faire prendre conscience que notre métier a besoin de personnes qui travaillent avec de la valeur humaine, de l'humanisme car Scrum c'est bcp sur la communication. C'est entrain de nous dire que ce que vous faites depuis 15-20 ans c'est quelque chose qui a été structuré historiquement, et Scrum réussit à faire ce que d'autres méthodes jusqu'à preuve du contraire, n'ont pas réussi : faire parler de l'Agilité, nous faire prendre conscience que l'industrie logicielle n'est pas parfaite. Vincent explique que l'on a relativement peu confiance en l'humain. Il n'y a jamais eu pourtant de tentatives de comparer XP à Scrum, et il est marrant de constater comment le débat s'est en partie organisé face à XP. Quelqu'un dit que même pas 25% des entreprises réussiront avec Scrum. Eric Mignot dit enfin qu'il pense qu'il y a un fantasme par rapport à la formation certifiante. Il demande "Pourquoi voulez-vous faire du Scrum ?" Les gens vont dire "pour régler tel et tel problème". Eric dit "mais si tu connais le problème... pourquoi as-tu besoin de Scrum alors pour le résoudre ?". Vincent de Pyxis explique que l'origine de la certification est de crédibiliser la démarche par rapport à du Waterfall. Que dans les faits, 85% de ce que les gens vont apprendre lors de la formation sera oublié s'ils ne travaillent pas par la suite simplement en lisant un livre ou deux de Ken Schwaber. Je disais une fois à quelqu'un : Scrum c'est pas ta mère qui te fait passer ton Bac mon coco. C'est ensuite pour que tu fasses des études et que tu bosses, si vraiment tu veux t'impliquer et mettre en place Scrum dans ton équipe. Eric Lefevre-Ardant propose pour terminer un livre : "Artisanal Retro-Futurism X Team-Scale Anarcho Syndicalism" Brian Marick **Ma petite conclusion et mon avis à moi que j'ai...** Tout d'abord sur les personnes présentes : les équipes de Pyxis par leur expérience ont conservé le débat sur le fond, chacun a exprimé ses points de vues, quelques remarques de personnes de la communauté purement XP pour pimenter, des interventions intéressantes de David qui avec sa carte rouge a amené Scrum sur la table d'examen, bref je resors de là avec de nouvelles idées et une prise de conscience. Vincent de Pyxis qui demande si ce n'est pas l'industrie qui est malade, Eric "Bob" Mignot assis à côté d'Emmanuel Gaillot, deux acteurs de théâtre c'était

marrant. Bravo à Eric et Guillaume qui ont piloté la séance, en laissant chacun s'exprimer tout en freinant les emballages de certains, c'était un moment agréable. Une réflexion sur la certification pour faire avancer le débat: Est-ce que la certification Spring est suffisante pour se déclarer prêt à coder une application d'entreprise dès le lendemain ? non. Elle est une garantie que vous avez reçu une formation par une personne habilitée et pas par un charlatan. Car il faut savoir que pour être CSP (certified scrum practionner) et formateur Scrum, c'est un sacré boulot, une reconnaissance par d'autres CSP, un ticket d'entrée à 7000 \$ par an, ce qui fait que les gens qui sont formateur Scrum sont des gens qui DOIVENT en faire à plein temps, et qui doivent aussi être sur le terrain, en faisant de l'accompagnement d'équipe. Avez-vous déjà vu un professeur de fac à vos côtés le premier jour de votre premier boulot pour vous aider ? non. Et pourtant vous avez eu ce sacré diplôme qui vous a rendu "ingénieur" ou autre... Le côté mercantile fait peur car aussi c'est un domaine que certains ne connaissent pas. Avez-vous déjà discuté avec des gens qui font des certifications sur des progiciels ? Pensez-vous que la certification Microsoft c'est bien ? Il y a des gens qui savent mieux que d'autres, qui travaillent pour être formateur, afin que d'autres gens qui ne savent pas puissent apprendre. Ensuite il y a des gens, de l'humain. Il y a des vendeurs d'armes, qui savent que l'entreprise va se blesser. Il y a des vendeurs de valeur, d'engagement et de plaisir, qui viennent dans des équipes pour reconstruire un capital humain. Prendre conscience que le système actuel peut être amélioré c'est une chose. Prendre ensuite les bons interlocuteurs pour progresser, c'est une autre affaire. Mon avis très personnel est que nous manquons un peu d'objectivité sur Scrum. Nous devrions parler des effets secondaires à nos clients. Pour autant, n'est-ce pas de notre responsabilité de tenter de changer ? Il y a des gens qui sont planqués, qui végètent et qui s'éclatent dans leurs coins. Et puis il y a d'autres gens qui ne supportent pas un système de développement basé sur des techniques vieilles de 15-20 ans, où personne n'a cherché à comprendre l'absurdité de nos comportements. Il y a ceux qui sont frustrés car c'est l'équipe de foot Scrum qui en ce moment se la joue plutôt que son équipe XP. A XP Day j'ai discuté avec beaucoup de personnes du métier ou des chefs de projets. Tous m'ont dit que sans Scrum ils n'auraient pas fait connaissance avec l'Agilité, puis le Lean, puis XP, bref le changement. Alors que Scrum soit le pont vers un autre monde, c'est quand même une bonne chose. Je vous donne [rendez-vous le jeudi 18 juin](#) à partir de 19h00 à l'EPITA à Paris pour participer à un WorldCafe sur Scrum, organisé par le French SUG. Si vous voulez rencontrer des gens qui font du Scrum, n'hésitez pas à venir en vous inscrivant sur le site MeetUp du French SUG. **Resources** [Eric a mit en ligne les photos de cette séance et de la journée](#) Merci à Eric et Guillaume. Enfin vous retrouverez l'enregistrement complet de la session sur le podcast, si vous voulez revivre les 1h22 de la séance, avec 45mn pour la première partie... Retrouvez la transcription de la séance "Scrum est-il dangereux" des XP Days France 2009.

## XP Day France 2009 - jour 2, Dans la peau du Challenger et Business Value Game

Suite de mon retour sur XP Day, après [la première journée](#), la deuxième sur [la rétrospective](#) et [le débat sur Scrum](#). Après un repas en compagnie d'une partie de l'association Agile France, je reprends par un jeu afin de comprendre les mécanismes du Leadership. **Atelier "Dans la peau du Challenger"** Emmanuel Gaillot [d'OCTO Technology](#) et Luc Bizeul, aux couleurs de [Pyxis Technologies](#) nous proposent de découvrir les challengers. Alors que les leaders et les chefs d'équipe sont mis en avant dans les entreprises, finalement comment émergent des challengers ? Un billet sur [le blog de Pyxis Technologies](#) de Martin Proulx apporte aussi un point de vue intéressant sur la page "l'Equipe" que l'on trouve sur les sites des entreprises, je vous conseille d'aller le lire. Nous voilà réunis dans une salle, environ 40 personnes. Emmanuel nous présente le cadre : un jeu politique adapté du livre "[Playing politics: The Nightmare continues](#)" de Michael Laver. Par groupe de 8 personnes, nous sélectionnons un leader. Pour notre groupe nous prenons le plus âgé (le plus sage aussi) à savoir Michel. L'exercice va constituer à jouer une campagne politique, afin de voir comment des Challengeurs émergent dans une organisation. Chaque équipe décide d'une politique budgétaire, ce qui fait qu'ensuite nous devons trouver quelle équipe a une politique assez proche, afin de nous allier avec elle. Luc joue le rôle de la Banque. Un ensemble de Loi permet de cadrer le jeu. Nous pouvons proposer des lois. Le premier élu a une orientation très communautaire et souhaite partager le revenu du plus riche avec le suivant... Mais il se fera exclure car une des Lois est de ne pas justement faire voter des lois pour forcer une équipe à donner de l'argent à une autre... Bref tout ceci pour vous raconter finalement ce qui s'est passé : on a passé un bon moment. L'exercice permet de regarder les mécanismes d'échanges, de négociations et de transactions avec les autres personnes. Chacun tente d'être moral, mais ce qui est amusant c'est qu'au final c'est l'équipe qui aura eu l'idée de jouer la montre de Laurent Morisseau qui gagnera. J'ai pas appris grand chose, mais j'ai bien aimé l'organisation et l'animation d'Emmanuel, qui a piloté avec brio sa démonstration. Il sera d'ailleurs Speaker à l'USI 2009 [en juillet prochain](#). ROTI: Qualité de la présentation : 4/5 Ce que j'ai appris : 2/5 **Dernier atelier : The Business Value Game** Je termine XP Day par un jeu, encore un jeu, organisé par [Pascal Van Cauwenberghe](#), Portia Tung et [Laurent Morisseau](#). Pascal est venu chez Zenika faire [une présentation sur Lean](#), je m'attendais à retrouver l'esprit Lean lors de ce jeu. L'exercice par table de 10 personnes était de jouer des itérations afin de livrer des fonctions, en utilisant la valeur métier pour diriger notre production. Cet exercice nous permet tout d'abord d'apprendre comment calculer ou estimer la valeur métier. Au début nous recevons un jeu de cartes spéciales avec par exemple "Château Fort coût: 1000 EUR complexité: 4". La valeur métier se calcule en divisant le coût par la complexité, ici 25 points de VM. Cela nous permet de constituer un product backlog ordonné par valeur métier. Un facteur supplémentaire s'ajoute ensuite : la satisfaction du client. A VM équivalent, certains artefacts rapportent plus de satisfaction au client Paul qu'au client Pierre. Si Pierre perd l'ensemble de sa satisfaction, nous devons retirer alors l'ensemble de ses commandes, le client furieux s'en va et nous ne pourrions donc pas produire... Bref le jeu va constituer par séquence à sélectionner l'ordre de production, à produire et ensuite comptabiliser la quantité de choses produites. L'exercice a été réalisé mais peut-être que nous étions trop nombreux. J'ai eu du mal à comprendre certains points, ce qui me fait penser que l'an prochain il faudra peut-être réduire le nombre de

participants... Pas facile de refuser du monde. Au jeu de l'élection politique, les organisateurs ont été strict en demandant aux personnes en surnombre d'être observateur. Je pense que le même principe devra être mis en place, afin d'éviter que 10 personnes essayent de tomber d'accord... Mais en dehors de cela, Laurent, Pascal et Portia ont très bien organisé et séquencé le jeu, exercice difficile avec beaucoup de monde. ROTI: Qualité de la présentation : 4/5 Ce que j'ai appris : 3/5

**Synthèse et l'heure du bilan** Ces journées XP Day étaient vraiment une bonne synthèse de l'Agilité, un moyen de confronter les points de vues de chacun (XP, Lean, Scrum), de réfléchir et d'échanger ensemble. Organisation top, peut-être trouver un moyen pour limiter le nombre de personnes par session ? Repas et buffet 5/5, temps consacré à discuter et échanger : 5/5. Du côté du contenu, j'aurai peut-être mieux fait de ne pas faire autant de session jeux le mardi, mais j'ai aimé ces moments qui permettent aussi de discuter avec pas mal de monde. Je conseille XP Day l'an prochain à ceux qui aiment s'aérer le cerveau, qui sont ouverts d'esprit, qui sont prêts à venir se faire challenger et à discuter. XP Day était aussi beaucoup de gens du métier, de la maîtrise d'ouvrage, qui cherchent de nouvelles techniques de gestion de projet, de gestion d'équipe. J'aurai aimé échanger avec les personnes venues du Canada, qui semblent avoir une sacrée avance par rapport à nous, que ce soit dans la culture ou dans la démarche. Mais nous nous réveillons petit à petit, la France est une terre à découvrir et à conquérir. Prochain rendez-vous, l'USI où nous verrons comment les coachs Agile expliquent aux Boss ce qu'est l'Agilité. Je retiens un mot de quelqu'un de Pyxis : quand comprendrons-nous que le passage vers des méthodes Agilite demande des gens qualifiés, sera créateur de nouveaux emplois, et devra être effectué par l'accompagnement du Changement ? Dernier article de la série sur XP Day, où je raconte ma participation au jeu "dans la peau du challenger" puis ensuite "Business Value Game".

## JavaOne la semaine prochaine, Devvix en novembre

Tel un oiseau migrateur, le Geek Java effectue 2 à 3 migrations par an afin de s'accoupler. La première a lieu au début de l'été, la seconde en général juste avant l'hiver. Cette année, ce sera JavaOne à partir du 2 juin jusqu'au 5 juin prochain à San Francisco. Entre la grippe Mexicaine, le rachat par Oracle de SUN Microsystems et la crise financière, à mon avis ça va swinguer pendant la semaine... JavaOne c'est plus de [1300 sujets proposés](#) par différents présentateurs, avec un comité de sélection de 35 personnes. Autant dire que le niveau est relevé, que pas mal de personnes se sont fait refuser leur présentation. [Les 4 thèmes](#) cette année : \* Rich Media Applications and Interactive Content \* Mobility \* Services including Cloud Computing \* Core Technologies – Java Platform Standard Edition (JavaSE), Java Platform Enterprise Edition (Java EE), Embedded, Real-Time and Java Card technologies, etc. JavaOne c'est tout d'abord des conférences toute la journée. Tôt le matin et tard le soir, se déroule aussi des mini-conférences plus détendues avec des sujets souvent funs comme par exemple cet année un sujet sur l'utilisation de la Wii et de JavaFX (BOF-4844 - Java and JavaFX Technologies and the Nintendo Wiimote: Just How Much Fun Can You Have? par Simon Ritter et Angela Caicedo). Pour suivre JavaOne, le mieux est d'utiliser la CB du Web, j'ai nommé : [Twitter](#) ! Alors Charlies-Papa-Bravo si tu veux suivre un peu ce qu'il se passe la semaine prochaine voici les personnes de la communauté francophone que tu pourras écouter : - Guillaume Laforge @glaforge, SpringSource/Groovy - Emmanuel Bernard @emmanuelbernard, JBoss Redhat/Hibernate - Alexis Moussine-Pouchkine @alexismp, SUN/Glassfish - Jérôme Bernard @jeromebernard, Elastic Grid - Ludovic Dubost @ldubost XWiki - Tugdual Grall @tgrall eXoPlatform - Benjamin Mestrallet @benjmestrallet

eXoPlatform



Si non à part [Jazoon](#) dans moins d'un mois maintenant, l'événement à suivre c'est Devvix 2009, du 16 au 20 novembre à Anvers en Belgique. [Les inscriptions sont ouvertes](#), [une petite vidéo sympathique](#) vous donne un aperçu de Devvix 2008. Et si non vous pouvez aussi commander de vieux numéros du Touilleur Express si vous souhaitez relire les articles sur Devvix cet été sur la page. Cela se passe [par ici](#). Donc cette semaine nous regarderons JavaOne, sans doutes quelques annonces autour de JavaFX. David Dewalle du Paris JUG est là-bas aussi, il nous racontera tout cela en revenant en France.

## World Café Scrum le 18 juin, une occasion de l'ouvrir



Avec le sponsor de Xebia France

Jeudi 18 juin à partir de 19h00 le French SUG organise un World Café autour de Scrum à Paris, à l'EPITA. Dans un premier temps, nous découvrirons [les résultats du sondage](#), dépouillé par l'équipe du French SUG. Dans un deuxième temps nous organiserons [un World Café](#). Voici le principe de fonctionnement :



- Planter le décor -

Créer un endroit convivial et sympathique - Rechercher et Explorer des questions qui ont du sens - Faire participer tout le monde - Connecter des perspectives différentes - Découvrir différents points de vue - Partager les découvertes collectives Ce type d'événement s'est déjà déroulé par exemple avec [l'association SigmaT](#) à Toulouse, autour de l'Agilité, animé entre autre par Claude Aubry. Le déroulement de la soirée n'est pas encore fixé, mais voici dans les grandes lignes les idées : - collecter des questions autour de Scrum - répartir une question par table - sélectionner un animateur par table pour faire la collecte et la synthèse - laisser chaque table débattre et marquer les réponses sur la nappe en papier - demander à tout le monde de changer de table toutes les 10mn par exemple - en fin de soirée, chaque animateur de table présente les résultats des discussions L'inscription se fait sur [le site Meetup du FrenchSUG](#). Si vous avez envie de venir débattre autour de Scrum, de l'Agilité, des gens qui font du Scrum, des réussites, des échecs, bref si vous avez envie de vous faire secouer un peu les neurones, rendez-vous le 18 juin prochain.

## JavaOne 09 comme si vous étiez là bas mais en fait non car c'est la crise

Quelques informations de JavaOne qui débute ce mardi. Lors de la conférence d'ouverture, quelques annonces et quelques présentations, avec la présence de Larry Ellison, PDG d'Oracle sur la scène. A noter l'annonce [d'un Java Store](#). Ce portail accessible actuellement en bêta privé proposera une liste de logiciels et de jeux réalisés avec la technologie Java et JavaFX, avec à priori un modèle économique identique à l'AppleStore. La conférence JavaOne 2009 peut être suivie avec Twitter ou via [le site JavaOne 2009](#) qui permet de suivre en direct les plus grosses conférences. D'autres blogs comme [eXoPlatform](#) proposent aussi de suivre le déroulement de JavaOne. **Bravo Groovy !** La nouvelle aussi importante de la journée, c'est que Groovy a remporté cette année le Script Bowl. Organisé par Robert Chinnici l'an passé, l'idée est d'élire le langage de script préféré. Les concurrents étaient représentés par les leaders de leurs communautés : Tom Enebo (JRuby), Rich Hickey (Clojure), Guillaume Laforge (Groovy), Martin Odersky (Scala) et Frank Wierzbicki (Jython). Chacun des représentants devaient montrer pourquoi son langage de script était le mieux, Guillaume a raconté sur Twitter, on devine que cela a dû demander un bon boulot de sa part. Alors bravo Guillaume ! [Alex Miller](#) raconte sur son blog le déroulement de la session. Voici une traduction libre en français. La session est organisée en 2 rounds. Un premier round où chaque gourou du langage présente les fonctionnalités clés, un deuxième round basé sur des fonctionnalités demandées par la Communauté. **Round 1 - les fonctions clés du langage** Franck Wierzbicki a fait une présentation rapide de Jython en mettant en avant la facilité de lecture et sa simplicité ("executable pseudo-code"). Il a aussi parlé de la facilité de l'usage du typage dynamique et de la réflexion. Guillaume Laforge présente quelques unes des nouvelles fonctions de Groovy 1.6 avec en particulier le meta-programming au moment de la compilation. Groovy a déjà un grand nombre de fonctionnalités de méta-programming et la version 1.6 permet de déclencher de la transformation du code à partir des annotations en autorisant la modification de l'AST ([Abstract Syntax Tree](#)) avant la génération du code. (note de nicolas : une très belle utilisation du pattern visitor par Groovy qui permet lors du parcours de l'arbre de surcharger et placer son code). Parmi les annotations citées par Guillaume : @Singleton, @Immutable, @Delegate, @Lazy et d'autres pour Swing comme @Bindable ou @Vetoable par exemple. Exemple de code:

```
@Singleton class Evil {
}
```

Rich Hickey ([Clojure](#)) a montré une version multi-threadée [du jeu de la vie](#), comment Clojure facilite grandement l'écriture de programmes multi-threadés en nous permettant de se focaliser sur la coordination des Threads plutôt que de s'engager sur la gestion des verrous et des états partagés. [Dick Wall](#) (un des podcasteurs de JavaPosse) présentait [Scala](#). Il a pris une approche intéressante en montrant comment certaines nouvelles fonctionnalités de Java 7 (Projet Coin) sont déjà supportées dans Scala. Il a montré par exemple l'utilisation des Strings dans les switch cases, la gestion des exceptions en Scala. Il a aussi parlé de [l'inférence de types](#) (note de nicolas : spéciale

dédicace à Sadek, y'en a dans F#, Haskell et Caml) et des blocks ARM qui sont supportés en Scala. Tom Enebo du projet JRuby a terminé le premier round avec une démonstration de la facilité à créer un DSL en utilisant [JMonkeyEngine](#) 3D Physics qui est un moteur pour créer des jeux impressionnants en Java. (note de dj nicolas : spéciale dédicace à Guillaume Pelletier, pour la partie 3D) **Round 2, l'épreuve de la communauté** Dans la deuxième partie, chaque langage doit parler de la communauté. Franck parle d'un IDE appelé Field utilisé par des artistes au MIT pour créer des projets artistiques, en utilisant Jython comme le langage de script. Guillaume a parlé de Grails et de Griffon et a montré une démonstration. Il dit d'ailleurs sur Twitter :

If you wanna learn abt the cool #swing #groovy #griffon demo that won the #javaone Script Bowl, go to the BOF-5189 on Wed 8:45pm, GW 104

Rich a parlé ensuite de l'utilisation du code comme données, Dick Wall pour Scala a fait une démonstration d'utilisation de CouchDB (une base Erlang) et de son utilisation à travers l'API Scala, écrite par Debashish Ghosh. Tom a montré l'essor de la communauté Ruby et JRuby en parlant du nombre de conférences, de livres, d'IDE et d'implémentations. Bref chacun a défendu son langage de script, j'imagine que cela devait être délicat. Un vote à la fin par les spectateurs a permis d'élire **Groovy comme le langage de script préféré à JavaOne 2009**. Encore bravo à Guillaume, qui n'a plus qu'à venir au Paris JUG afin de nous faire une petite présentation aux oignons. **Twitter** Twitter et le channel #javaone sont particulièrement actifs, plus de 500 messages, avec par exemple des messages importants. Voici une petite sélection avant d'aller dormir afin de vous faire bosser un peu avec Google: "jdk 7 = death of classpath #javaone" de @judahilm, @bienator "Ouch. RT @timbray: #javaone press room consensus: This is the last JavaOne." "Just heard that Groovy won the Script Bowl at #javaone , beating Scala, JRuby, Clojure and Jython! Well done Guillaume!!" (Rod Johnson le patron de SpringSource dit bravo à Guillaume Laforge) "Java Store is a confidential project. Please do not share your experience, thoughts, or opinions as well as any images - that SUCKS #javaone" de @dushyanth "I don't expect a lot of changes, just an expanded investment ... coming from Oracle". -- Larry Ellison at #JavaOne" de @webmink "In javaone technical keynote (the real keynote IMO). JB in the Kenai demo announcing "Hudson in the cloud". @alexismp Je vous conseille de vous prendre un compte Twitter, vous pourrez ainsi suivre mes blablas (@Letouilleur) ainsi que ceux des gens qui sont à San Francisco en regardant le channel #javaone.

## Réaction de Rod Johnson à l'annonce de Red Hat

Rod Johnson(CEO de SpringSource, co-auteur de Spring) [publie sur le blog de SpringSource](#) une réaction assez vive à l'annonce de la nouvelle stratégie de Red Hat Inc. appelée [JBoss Open Choice](#). Red Hat Inc. communique officiellement une offre ouverte visant à rassurer les clients finaux en proposant un package construit sur JBoss MicroContainer, capable de supporter Java EE, Spring Framework, Struts, bref un peu prêt tout ce dont on a besoin pour construire une application d'entreprise. L'objectif de l'offre JBoss Open Choice Strategy est d'ouvrir un peu plus l'architecture de JBoss vers d'autres frameworks comme GWT, Seam, Struts et Spring, avec je pense l'idée du support derrière... Rod Johnson objecte sur son blog que le choix reste dans les mains du client, qui n'est peut-être pas faux, et que Spring étant (d'après lui) un standard de facto, il considère cette annonce comme un mouvement défensif de Red Hat Inc. Bref cette manœuvre d'ouverture ne semble pas le rassurer. Red Hat Inc. explique l'intérêt de la nouvelle offre :

The JBoss Open Choice strategy aims **to provide a common application platform supporting various current and future programming styles** and enabling different deployment models. The technology behind the JBoss Open Choice strategy is designed to support models such as Plain Old Java Objects, Java EE, Spring Framework, OSGi and a variety of Rich Internet Application frameworks. The flexible platform is intended **to enable enterprises to separate core enterprise services** such as persistence, caching and clustering, making the operational footprint simpler and easing development and deployment. All of these capabilities are expected to be available through a common core which will streamline the management of these varied environments. ([source](#))

Mes idées sur le sujet (qui ne sont certainement pas toutes pertinentes) : c'est bien l'annonce d'un noyau modulaire et simple qui fait réagir Rod sur son blog, Red Hat propose de pérenniser le JBoss Microcontainer, l'un des points clés de l'architecture de JBoss, et qui dès le départ a été construit afin de piloter différents modules. J'ai présenté l'architecture du MicroKernel (pas forcément la même chose que le Microcontainer) lors de ma formation JBoss [en 2005 sur le Touilleur Express](#) à Berlin. Comme toujours, une réaction en plein JavaOne pour alimenter aussi le buzz autour de SpringSource, c'est toujours bon à prendre. On retiendra donc que Red Hat Inc. semble suivre la voie tracée par SpringSource comme l'explique [Matt Asay sur CNet](#):

[...]It also means it will spend all its time fending off open-source upstarts in open source's primary hunting ground: infrastructure. It's therefore not surprising that Red Hat's new JBoss Choice program appears to be a thinly veiled attempt to crimp SpringSource's style, given the ubiquitous open-source Spring Framework and the threat it poses to JBoss Application Server.

SpringSource est entrain de réussir à monétiser le marché de l'open-source, pour l'entreprise et à l'adresse des personnes de l'exploitation. Red Hat Inc. a donc logiquement répondu et semble indiquer qu'ils souhaitent suivre la même voie, ce qu'ils font d'ailleurs très bien avec succès dans d'autres domaines. A suivre... Note: je passe le micro à [Cyrille](#) qui doit avoir une idée sur la question...

## A toi, Chef de projet, qui veut passer à Scrum

Plus j'avance avec Scrum, plus je me rends compte que l'on ne comprend pas Scrum. Au delà d'un framework léger, qui permet de livrer rapidement et régulièrement, c'est tout un changement sur la Valeur qui constitue l'une des vraies forces de Scrum, et des méthodes Agiles. Le sujet est passionnant, je pense de plus en plus qu'un Scrum Master ne doit être ni un chef de projet, ni un développeur. Je rejoins l'idée de quelqu'un de Pyxis Technologies qui disait "...quand va-t-on réaliser que mettre en place Scrum est un nouveau métier ? qu'accompagner une équipe avec une méthode Agile est un nouveau métier ?". Et il n'a pas tort non ? Les premiers effets de Scrum sont des livraisons plus régulières, un meilleur moral pour l'équipe, une meilleure qualité, la possibilité d'intégrer le changement et une bonne communication avec le Client. C'est le premier effet. Cet effet finalement est assez salutaire: mettez de l'oxygène dans un aquarium et vos poissons seront contents. Le deuxième effet qui prend plus de temps, c'est la création de nouvelles Valeurs. Je pense à Transparence, Aider l'autre, Communiquer, Demander de l'aide, etc. La transparence ne vient pas magiquement avec Scrum. Mais les "cérémonies" facilitent ce changement, pour ceux qui le veulent. Le deuxième effet de Scrum ne se mettra en place qu'avec quelqu'un qui fait prendre conscience de ces Valeurs à l'équipe. Cela demande du temps et de l'accompagnement. Je vous le dis clairement : votre formation de Scrum vous aide à répondre aux questions "comment" mais pas aux questions "pourquoi". Il va falloir lire, vous documenter, vous faire aider d'un coach. Ensuite je me rends compte que pour être Scrum Master, vous ne devez pas être chef de projet. En tant que chef de projet, rien que ce mot "chef" qui est lourd de sens, devient "responsable du projet" lorsque vous passez à Scrum. Vous n'êtes plus chef de rien, votre travail est de faciliter la vie de votre équipe, de s'assurer que l'on suit le framework Scrum, et c'est tout. Le projet est piloté par l'équipe, l'avancement du projet est donné par le framework Scrum, c'est déstabilisant non ? Vous n'avez plus de rôle de responsable et donc c'est par le leadership que vous serez légitime. Être Scrum Master, c'est être un capitaine dans une équipe de foot : vous aussi vous allez jouer avec les autres, on vous reconnaît simplement pour votre leadership et votre expérience. Chez le client pour lequel je travaille aujourd'hui je travaille mon leadership, dans le sens d'aider, faciliter, mettre une bonne ambiance, être là pour aider à résoudre les conflits... ce qui pour moi représente vraiment un nouveau rôle, un nouveau métier. Et franchement, c'est bien plus sympa que de piloter des développeurs à coup de Microsoft Project, en s'excitant pour mettre à jour un diagramme de Gantt... Si vous envisagez d'utiliser Scrum, prenez le temps de former vos équipes, et prenez 10 fois plus de temps pour trouver un cabinet de conseil ou embaucher un collaborateur afin de piloter vos équipes. Ce métier s'appelle "Coach Agile" pour l'instant, et retenez que ce n'est pas un chef de projet, pas un architecte, c'est un coach. Si vous pensez que votre responsable ne vous suivra pas, envisagez sérieusement de changer d'équipe, de responsable ou de société. Il est difficile, mais pas impossible, de mettre en place Scrum sans le support de son supérieur hiérarchique. Rendez-vous le jeudi 18 juin au [Scrum World Cafe](#)



Avec le sponsor de Xebia France

## Liste des logiciels testés sur le Touilleur Express

J'ai eu l'occasion de tester différents logiciels autour de Java depuis un an, sur la proposition d'éditeurs. Cela permet de proposer une opinion relativement neutre et technique sur les logiciels. Voici la liste des articles publiés depuis un an: - [XDepend](#) d'OCTO Technology. Analyse de la structure du code Java. - [Kalistick](#), outil d'analyse de code en mode hébergé (Saas). - [SpringFuse](#) de Celerio, moteur de démarrage rapide des projets avec Spring, en mode hébergé (Saas). - [SonarJ](#) de Hello2morrow, pour analyser la complexité du code et construire une bonne architecture. - [Dependency Analyzer](#) outil open-source d'analyse des dépendances de maven2. Prochaine idée d'articles, que je n'ai pas encore commencé : [Structure 101](#). Outil d'analyse disponible à la fois sous la forme d'un plugin Eclipse/IDEA IntelliJ et d'une application classique. Paul Hickey de HeadwaySoftware.com m'a contacté suite à un article publié. Son outil est utilisé entre autre par [Kent Benck](#), je me suis dit qu'il serait intéressant de tester Structure101. A suivre très bientôt !

## Jazoon 2009 : Didier Girard de SFEIR interviewé

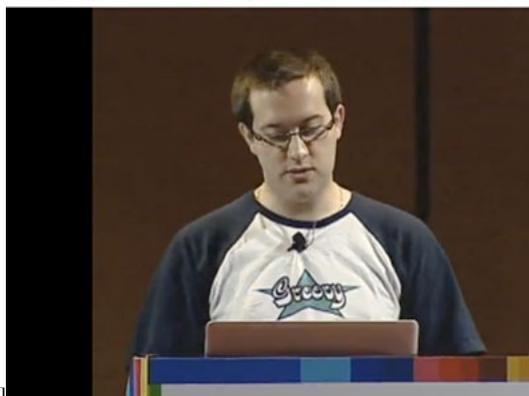
Voici l'interview réalisée par Thierry Czarnyszka, membre de l'équipe d'organisation [de Jazoon](#), de Didier Girard de [SFEIR](#). Jazoon est un salon sur Java qui aura lieu du 22 au 26 juin prochain à Zurich en Suisse. **Thierry Czarnyszka de l'équipe d'organisation de Jazoon: Bonjour Didier, tu as contribué en tant que membre du comité des programmes au choix des sujets abordés à Jazoon'09: conférence Java et Open Source qui se tiendra à Zürich du 22 au 25 juin prochain. Quels sujets ont particulièrement attiré ton attention pour cette édition 2009 ?** Didier Girard, Directeur Technique de SFEIR: Cette conférence est assurément orientée vers le

futur. Elle aborde tous les sujets que nous devons maîtriser: les technologies RIA, les langages dynamiques, REST, le Cloud Computing et entre autres la téléphonie mobile. **Jazoon: En participant à de tels événements, en l'occurrence comme Jazoon, quels bénéfices et intérêts les personnes impliquées dans les technologies Java et Open Source puissent en tirer ?** DG: Nous faisons un des métiers les plus atypiques: dans 5 ans beaucoup des connaissances qui nous sont nécessaires pour trouver un emploi actuellement ne seront plus valables. Ceci était vrai il y a 5 ans et le sera encore dans 5 ans. Il est donc nécessaire pour chacun d'entre nous de rester à l'écoute de l'orientation technologique prise par le marché de manière à garder une bonne employabilité. C'est ce que je fais, c'est ce que je conseille à tous les ingénieurs que je rencontre. Aller à Jazoon est le meilleur moyen d'atteindre cet objectif. **Jazoon: La France dispose du DIF - Droit Individuel à la Formation. C'est un dispositif intéressant comparée aux autres pays européens voisins mais il semble être méconnu. Partages-tu ce sentiment? Le DIF, ne pourrait-il pas être une opportunité et encourager tout à chacun à en bénéficier?** DG: Le fait que Jazoon soit éligible au DIF est une bonne chose. Le DIF est bon moyen pour monter en compétence. **Jazoon: La venue de James Gosling à Jazoon'09, le père et fondateur de Java, en Europe est rare: que pensez-tu de cette opportunité de le rencontrer en tant que membre de la communauté Java?** DG: Il n'est pas si facile de faire venir James Gosling en Europe. Bravo aux organisateurs de la conférence. Rencontrer le père du langage de programmation le plus populaire au monde est un must. Grâce à lui tout un écosystème est né. Rien que pour sa venue, le déplacement à Jazoon est intéressant. **Jazoon: Pourquoi iras-tu à Jazoon'09 ?** DG: Jazoon, c'est l'événement java Européen de ce milieu d'année. C'est une conférence où de grands speakers seront présents. La taille de l'événement fait qu'ils seront accessibles. De plus Jazoon sera le moment idéal pour discuter des annonces de Google I/O et de JavaOne. Autant de bonnes raisons pour faire le déplacement. **Jazoon: Merci Didier et au plaisir de te voir prochainement à Zürich à partir du 22 juin prochain.** Pour plus d'information sur la conférence visitez [jazoon.com](http://jazoon.com) Pour recevoir des informations complémentaires au sujet du DIF adressez votre mail à [info@jazoon.com](mailto:info@jazoon.com)

## Premiers pas avec Grails 1.1.1 et Google AppEngine

Ah là là... dire qu'à la fin de la lecture de ce billet j'aurai complètement changé votre vision du Web... Je comprends mieux maintenant Didier Girard de SFEIR, il y a un avant et un après Google AppEngine... Mais bon, soyons patient et voyons ce que nous pouvons faire aujourd'hui. Ce qui suit est inspiré, pour ne pas dire affreusement copié sur ce que Guillaume Laforge a présenté [le 28 mai dernier lors de la conférence Google I/O](#) à San Francisco. J'ai simplement adapté la présentation en français, je l'ai fait fonctionner, pour le reste tout le crédit va à Guillaume. [caption id="attachment\_1427" align="alignnone" width="391" caption="Guillaume Laforge Google IO

Google I/O 2009 - Groovy and Grails in App Engine



2009"] A la fin de cet article vous serez en mesure de déployer sur Google AppEngine une petite application simple. **Installer Grails** [Téléchargez](#) et installer Grails 1.1.1. Pour ma part, sur Mac OS j'ai ensuite édité mon fichier `.bash_profile` afin de déclarer le chemin d'installation de Grails, et fixer la version de Java à la version 5 pour éviter quelques soucis plus tard avec Google AppEngine et DataNucleus. Voici le contenu du fichier `$HOME/.bash_profile`

```
export M2_HOME=/usr/share/maven
export JAVA_HOME=/System/Library/Frameworks/JavaVM.framework/Versions/1.5.0/Home
export GRAILS_HOME=/Users/nicolas/Dev/grails/grails-1.1.1
export APPENGINE_HOME=/Users/nicolas/Dev/appengine-java-sdk-1.2.1
export PATH=$PATH:$GRAILS_HOME/bin
```

Vérifiez que Grails est correctement installé avant d'aller plus loin:

```
macbook-pro-de-nicolas-martignole:~ nicolas$ export GRAILS_HOME=/Users/nicolas/Dev/grails/grails-1.1.1
macbook-pro-de-nicolas-martignole:~ nicolas$ export PATH=$PATH:$GRAILS_HOME/bin
macbook-pro-de-nicolas-martignole:~ nicolas$ grails -version
Welcome to Grails 1.1.1 - http://grails.org/
Licensed under Apache Standard License 2.0
Grails home is set to: /Users/nicolas/Dev/grails/grails-1.1.1
```

```
Base Directory: /Users/nicolas
Running pre-compiled script
Script not found: Version
macbook-pro-de-nicolas-martignole:~ nicolas$
```

Les observateurs ont remarqué certainement la variable APPENGINE\_HOME. Il est temps de télécharger Google AppEngine, après avoir demandé un compte. Suivez attentivement les explications de la page principale de [Google AppEngine](#). Une fois le téléchargement effectué, j'ai simplement décompressé l'archive tar.gz dans un répertoire, puis ensuite j'ai configuré la variable APPENGINE\_HOME afin qu'elle pointe vers le répertoire de GAE (Google AppEngine). **Créer sa première application Grails** Grails vous permet de créer rapidement une application et de la tester. Allons-y, placez-vous dans un nouveau répertoire vide et créez votre première application grails, appelée ici gaetex:

```
grails create-app gaetex
```

Lorsque nous utilisons Google AppEngine, nous devons désinstaller le plugin JPA et désinstaller le plugin Hibernate. Le plugin JPA génère un fichier persistence.xml qui vient en conflit avec celui utilisé par le mode JPA du plugin Google AppEngine. Concernant Hibernate, nous n'en avons pas besoin ici. Pour cela il suffit d'exécuter les 2 commandes Grails suivantes en vous mettant d'abord dans le répertoire gaetex que nous venons de créer :

```
cd gaetex
grails uninstall-plugin hibernate
grails uninstall-plugin jpa
```

Notez qu'en principe, le plugin jpa n'est pas installé, j'ai ajouté cette commande au cas où vous l'auriez installé aussi. **Installer Google AppEngine** L'installation de Google AppEngine est simple. Assurez-vous simplement que la variable APPENGINE\_HOME pointe vers votre répertoire d'installation de GAE puis lancer la commande d'installation du plugin App-Engine :

```
grails install-plugin app-engine
```

Ce qui donne sur ma machine:

```
macbook-pro-de-nicolas-martignole:gaetex nicolas$ echo $APPENGINE_HOME
/Users/nicolas/Dev/appengine-java-sdk-1.2.1
macbook-pro-de-nicolas-martignole:gaetex nicolas$ grails install-plugin app-engine
...
...
```

**Tester le tout** Vérifions que tout fonctionne en démarrant le serveur Jetty intégré à Grails. Dans le répertoire gaetex, lancez grails:

```
grails app-engine
```

Connectez-vous à votre machine (<http://localhost:8080/>) et vous devriez voir votre première page Grails s'afficher. **Créer une première entité** Nous allons maintenant créer une entité Book, et générer ensuite la partie vue et contrôleur. Avec Google AppEngine, il n'est pas possible de placer ses Entités directement dans le répertoire "grails-app/domain". Il est nécessaire d'utiliser un package. J'ai donc fait comme Guillaume durant sa démonstration, j'ai créé une entité Book dans le package "demo" avec la commande suivante :

```
grails create-domain-class demo.Book
```

Il serait intéressant maintenant d'ajouter deux attributs "title" et "author" à notre Book. Editez le fichier gaetex/grails-app/domain/demo/Book.groovy avec un éditeur de texte, et ajoutez simplement 2 attributs pour le titre et l'auteur du livre :

```
package demo
```

```
import javax.persistence.*;
// import com.google.appengine.api.datastore.Key;

@Entity
class Book {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    Long id

    // Added by Nicolas Martignole
    String title
    String author

    static constraints = {
        id visible:false
    }
}
```

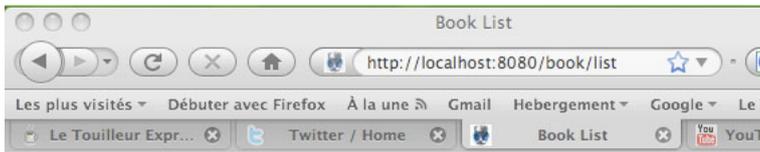
Avec ces deux attributs en place, nous allons maintenant générer un contrôleur et tester à nouveau notre application. Lancez la commande de génération du contrôleur et des vues:

```
grails generate-all demo.Book
```

Puis relancez ensuite le serveur Grails :

```
grails app-engine
```

Voici ce que cela donne après avoir créé quelques livres (au logo prêt) :



# Le Touilleur Express

Home New Book

## Book List

Id	Author	Title
1	Guillaume Laforge	Groovy in Action
2	Antonio Goncalves	Java EE6
3	Nicolas Martignole	Mon premier livre

Vous pouvez retrouver la même chose avec la voix de Guillaume à partir de la minute 50 sur la vidéo [YouTube](#) de la présentation. **Déployer son application Grails sur Google AppEngine** Voici la liste des étapes: 1) créer une application sur Google AppEngine 2) modifier le fichier grails-app/conf/Config.groovy et ajouter une ligne avec le nom de votre application 3) packager et préparer l'application 4) effacer un fichier parasite (probleme à résoudre) 5) utiliser le script appCfg.sh pour fournir votre email et votre mot-de-passe Google 6) ensuite simplement mettre à jour avec le plugin google-app-engine de Grails Voyons en détail chaque étape: 1) Rendez-vous sur "Google AppEngine" afin de faire une demande, puis ensuite de créer votre application. L'application id doit être unique, ce pourra être par exemple testgrails1234 2) Ouvrir le fichier Config.groovy se trouvant dans votre application et définissez l'application id que vous venez de mettre

```
google.appengine.application="touilleur"
```

3) Lancer la création du package avec les commandes suivantes: Il faut fixer la version à un nombre entier car GAE n'accepte pas les numéros de version type x.y.z

```
grails set-version 1
grails app-engine package
```

4) Effacer le fichier "datastore-indexes.xml" sans quoi voici l'erreur que vous rencontrerez:

```
macbook-pro-de-nicolas-martignole:gaetex nicolas$ /Users/nicolas/Dev/appengine-java-sdk-1.2.1/bin/appcfg.sh update target/war
Reading application configuration data...
Bad configuration: XML error validating target/war/WEB-INF/datastore-indexes.xml against /Users/nicolas/Dev/appengine-java-sdk-1.2.1/docs/
Caused by: cvc-elt.1: Cannot find the declaration of element 'datastore-indexes'.
Please see the logs [/tmp/appcfg16875.log] for further information.
```

**Pour résoudre ce problème, effacez le fichier datastore-indexes.xml** avant de relancer la commande appcfg.sh (voir étape suivante)

```
rm target/war/WEB-INF/datastore-indexes.xml
```

5) Lancer la première fois seulement la mise à jour avec l'utilitaire appcfg.sh de Google AppEngine comme suit:

```
$APPENGINE_HOME/bin/appcfg.sh update target/war
```

Renseigner votre adresse mail Google ainsi que votre mot de passe. Il suffit de mettre son login en fait (mdupont et pas mdupont@gmail.com). 6) Ensuite à chaque fois que vous souhaitez mettre à jour votre application vous pourrez utiliser le plugin de Gae dans Grails, plugin qui ne marche pas la première fois avec l'authentification en se plantant. C'est pour cette raison qu'il faut utiliser l'utilitaire appcfg.sh la première fois.

```
grails app-engine deploy
```

Si tout se déroule sans problèmes vous devriez alors mettre en ligne votre première application !!! L'url de votre application sera <http://<applicationid>.appspot.com> Si vous voulez tester un exemple d'application : [PetClinic with Grails](#) **Conclusion** Comptez 1h la première fois, puis ensuite 10 minutes pour refaire cet exercice. J'ai été vraiment séduit par la simplicité de Grails, et cette intégration avec Google App Engine. Celui-ci est gratuit jusqu'à un certain nombre de requêtes par jour, et croyez-moi vous avez largement de quoi tester votre code sans devoir payer, toujours dans l'esprit Google. Je crois vraiment à un nouveau type de développement rapide d'applications, à la monétisation de compétences en Groovy et en Grails. Pour ma part je travaille en ce moment sur une application pour le Touilleur Express, un mix de LinkedIn avec une touche de Google Calendar, un esprit de Geek et bien entendu, toujours la pâte Touilleur Express. Guillaume Laforge travaille aussi sur Gaelyk, un framework léger qui facilite l'intégration de Google App Engine si vous souhaitez n'écrire qu'en Groovy votre application Web, sans utiliser Grails. Il cite aussi [le framework iUI](#) qui est un framework Javascript destiné à faciliter l'écriture d'application pour l'iphone, hébergé sur Internet. Cela revient à ne pas passer par l'AppleStore, à simplement offrir un service type iPhone mais hébergé sur Internet directement. **Références** - [Page d'accueil Google App Engine](#) - [Page pour créer sa première application GAE](#) - [Plugin GAE Grails](#) le site à surveiller

- [Groovy Web Console](#) permet de taper du Groovy, application Google App Engine - [My Own Groovy](#) par Guillaume Laforge avec quelques applications pour tester - [Portail des applications Google App Engine](#) **Problèmes connus** Il y a un certain nombre de soucis pour l'instant sur le plugin Google App Engine pour Grails au moment où j'écris ces lignes (juin 2009). Ne vous lancez pas dans ce test si vous ne maîtrisez pas un minimum Grails. Je n'ai testé que la version JPA, je ne sais pas si la version JDO est plus ou moins stable. Pour terminer je vous recommande de regarder la présentation de Guillaume Laforge sur YouTube:

## Le portail d'eXoPlatform et JBoss Portal fusionnent

Exclusif le Touilleur Express : la solution [eXoPortal](#) de l'éditeur français eXoPlatform et le produit JBoss Portlet Container, qui implémentent chacun la spécification JSR-286 Portlet 2.0, ont annoncé aujourd'hui à 15h00 leur fusion. L'annonce a eu lieu lors du salon "[Gartner Portals, Content & Collaboration Summit](#)". La fusion se fait vers le projet open-source JBoss Portal. Julien Viet d'eXoPlatform est l'ancien spec-leader de JBoss Portal, même le fondateur du projet en 2004. Comme il l'explique sur [le blog d'eXoPlatform](#), les 2 produits fournissent le même service, avec deux approches différentes :

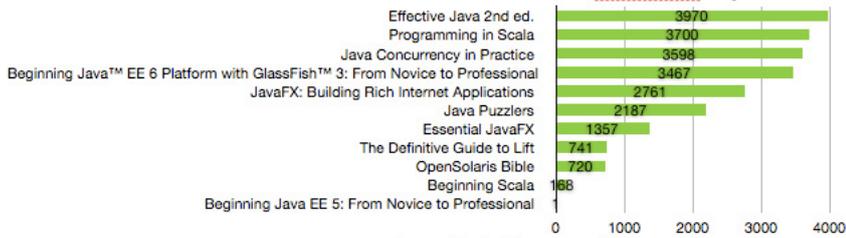
JBoss Portal and eXo Portal in essence do the same job: aggregating applications to the user desktop, however both focus on different strengths. The main difference between products comes from the nature of the companies.

C'est une excellente nouvelle pour la communauté d'une part, et pour eXoPlatform, éditeur de logiciel français. Un peu de cocorico par les temps qui courent, cela fait du bien non ? La nouvelle version sera distribuée sous licence LGPL, en fin d'année. Cette annonce va aussi redynamiser le projet JBoss Portal, la contribution d'eXoPlatform apporte un retour précieux des déploiements du produit eXo Portal à la version communautaire. Bravo à l'équipe d'eXo Platform, d'autres nouvelles à suivre pour mi-juin. **Références:** - [Détails et FAQ sur le site d'eXoPlatform](#) - [L'annonce officielle](#) - [Rencontre avec Benjamin Mestrallet](#) interview réalisée en janvier dernier. - [Interview de Dimitri Baeli](#) à propos de la mise en place de Scrum chez eXo platform.

## Les 10 livres les plus achetés à JavaOne, la bonne liste

TheServerSide a publié [la liste des 10 livres](#) les plus achetés à JavaOne 2009. Certainement pressé, une coquille s'est glissée dans le classement concernant le livre d'Antonio Goncalves sur Java EE 6. Le livre "*Beginning Java™ EE 6 Platform with GlassFish™ 3: From Novice to Professional*" est le 7ème livre le plus vendu cette année à JavaOne 2009 au Java Book Store ! En fouillant un peu plus, vous allez voir que le livre est même plutôt bien classé sur Amazon.com, mais avant cela voici la liste **corrigée**, la source originale est sur The Server Side. 1. [JavaFX: Building Rich Internet Applications](#) - Addison Wesley ISBN: 013701287X 2. *Essential JavaFX* - PTR (out June 11, 2009) ISBN: 0137042795 3. *Effective Java 2nd ed.* - PTR ISBN: 0321356683 4. *Java Puzzlers* - Addison Wesley ISBN: 032133678X 5. *Programming in Scala* - Artima ISBN: 0981531601 6. *Java Concurrency in Practice* - Addison Wesley ISBN:0321349601 7. [Beginning Java™ EE 6 Platform with GlassFish™ 3: From Novice to Professional ISBN: 1430219548](#) 8. *The Definitive Guide to Lift* - Apress ISBN: 1430224215 9. *Beginning Scala* - Apress ISBN: 1430219890 10. *OpenSolaris Bible* - Wiley ISBN: 0470385480 On note l'intérêt pour JavaFX des personnes présentes à JavaOne, intérêt qui n'est pas encore reflété dans les ventes sur Amazon. J'ai regardé le classement sur Amazon, classement que vous retrouvez sur chaque page. Je n'ai regardé que la version anglaise du site Amazon. J'ai tapé le code ISBN de chaque livre, puis j'ai relevé le rang de vente Amazon globale, j'ai appliqué un ratio pour que le livre le moins vendu soit sur une base 1. Voici les résultats classés par ordre de classement sur Amazon.com ( pas amazon.fr). Plus la barre horizontale est grande, plus le livre **se vend**. Le livre le plus demandé sur Amazon.com de la liste de JavaOne est en fait "Effective Java, 2nd Edition". Le livre d'Antonio a un classement tout à fait honorable pour un livre qui vient juste de sortir.

### Classement relatif des ventes de livres sur Java sur Amazon.com 10 juin 2009



Source: <http://touilleur-express.fr>

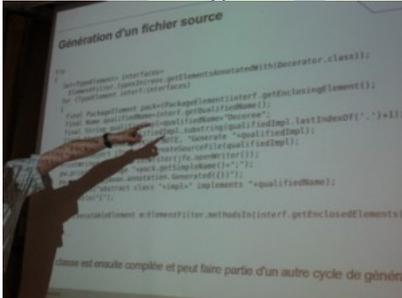
Ce qui me surprend c'est qu'il n'y a aucuns livres sur Spring. Est-ce que quelqu'un qui a vu le Java BookStore sur place peut nous dire quels éditeurs étaient représentés ? Je suis étonné de ne pas voir un livre comme "[Pro Spring 2.5](#)". On est bien d'accord aussi que JavaOne n'est pas complètement représentatif. Si vous souhaitez rencontrer Antonio Goncalves, il dédicace **le samedi 20 juin** dans l'après-midi son livre au magasin "[Le Monde en tique](#)" à Paris. C'est l'une des librairies les plus connues de Paris, c'est aussi un partenaire du Paris JUG. Pour ma part j'ai déjà commandé mon exemplaire... lecture pour cet été.

## Soirée Paris JUG sur Java Avancé

Annotations, Typages et Généricités, l'API Concurrent en Java, tel était le menu de [la soirée du 9 juin 2009](#) au Paris JUG. Avec 24h de retard, un petit retour qui sera plus court que d'habitude. La soirée s'est déroulée autour de 3 présentations. Philippe Prados d'Atos Origin, qui publie régulièrement dans différents magazines comme [PROgrammez.com](#) ou [GNU/Linux Magazine France](#) par exemple. Ensuite Denis Ah-Kang d'Atos Origin, Open Source Center. Et enfin Alexandre Bertails d'Atos Origin, qui a la chance de repasser ce soir après [la soirée Web Sémantique](#) du 11 mars dernier. **Les Annotations et la JSR-269**



Durant la première partie, Philippe Prados nous a présenté les Annotations en Java 5, puis ensuite la JSR-269 qui permet de créer des Annotations utilisées lors de la compilation du code source vers le bytecode. La JSR-269 "[Pluggable Annotation Processing API](#)" est donc une spécification d'annotation qui permet de déclarer des annotations au niveau de la source. Elle ne date pas d'aujourd'hui puisqu'elle est disponible depuis 2006. C'est peut-être inquiétant d'ailleurs de voir que les articles autour de cette spécification ne sont pas très frais comme sur [Developpez.com](#) ou [TheServerSide](#).



La présentation était très complète, quoiqu'un peu rapide dans le rythme. Ce que j'ai retenu : ce système d'annotation permet d'ajouter du code comme une interface à une classe existante, il permet en quelques sortes d'écrire des ressources comme un fichier Web.xml automatiquement. C'est donc intéressant dans un premier temps pour la génération de ressources. J'ai ensuite vu une démonstration de génération de code. Difficile de trouver un use-case, de l'aveu même du présentateur. Mais intéressant... pour se dire que cela ressemble un peu trop à de la science-fiction. Je suis resté plus dubitatif durant la partie sur le pattern Decorator. L'intérêt d'un Dynamix Proxy en Java est justement d'avoir une classe qui implémente des interfaces connues à l'exécution, principe utilisé depuis longtemps en Java... Il faudra que je rédige un truc là-dessus. A propos des patterns justement, l'un des 23 patterns du [Gang Of Four](#) s'appelle *Proxy* ou Procuration en Français, il appartient à la catégorie patterns structuraux. Le principe de ce pattern est d'envelopper un Objet dans un mandataire et d'utiliser ce mandataire à la place de l'objet original. **Deuxième présentation : java.util.concurrent**



Ensuite la deuxième présentation était sur l'api [java.util.concurrent](#). Présenté par Denis Ah-Kang d'Atos Origin. Dans un premier temps, après avoir rappelé les bases de la synchronisation et des problèmes connus en Java, Denis demande la signification de "volatile" à la salle. Grand silence, quelques mains courageuses se lèvent, les réponses arrivent péniblement... ce qui montre aussi que l'on parle d'un mot clé très peu utilisé en Java. Est-ce que "volatile" n'est pas un mot clé pour indiquer que cette variable est un poulet ? non ce n'est pas cela, dommage José ;-) Rien à voir avec la grippe aviaire. La bonne réponse : le mot clé volatile est un marqueur pour indiquer à la JVM que la variable sera accédée par plusieurs Threads. Par conséquence, la variable ne sera jamais caché dans le cache de chaque Thread, la valeur sera lue à chaque fois par les différentes Threads en mémoire. Et c'est là que l'on apprend que les Threads ont des caches de variables... Plus intéressant, lorsque vous mettez ce mot clé "volatile", l'accès à la variable sera effectué dans un bloc synchronized utilisant l'instance. Un simple "synchronized(this)" finalement, comme on le voit lorsque l'on décompile le code avec javap (et pas javac). Sinon quoi d'autre ? volatile s'utilise sur des objets mais aussi sur des variables simples type "int". Cela permet de rendre atomique par exemple l'incréméntation d'une variable de type "int i++", car sachez-le, par défaut un simple "i++" n'est pas atomique en Java... Denis montre ensuite justement ce cas précis avec un bout de code simple, où l'on constate que les accès concurrents ne s'effectuent pas correctement. Dans la suite de la présentation il aborde plus le côté API, la JSR-166 de Doug Lea entre autre. Je vous conseille de lire tranquillement sur internet, sur les différents blogs que vous rencontrez, il y a toujours quelques trucs à savoir. On oublie pas les 12 classes du package "[java.util.concurrent.atomic](#)" par exemple, qui vous éviterons des soucis de synchronisation par exemple. La documentation sur ce sujet se trouve [dans la javadoc de Java 5](#). Sinon pour la suite, je vous renvoie au meilleur article du monde entier cité par les plus grands de ce monde, soyons modeste, hop je suis comme cela : [Brillez en société avec votre BlockingDeque](#). Si Asterix et Obelix n'ont pas lu cet article, ils pourront optimiser leur



campagne de chasse. **La pause buffet** Le temps de

prendre un peu de quoi boire, je croise pas mal de monde. A chaque fois, ces buffets sont trop courts. En fait nous devrions faire des buffets de 3h et une pause avec une présentation de 30 minutes. Qu'en pensez-vous ? Et si par exemple j'avais envie de ne faire que la présentation #1 pour ensuite discuter et boire un coup avec d'autres personnes ? Ah l'idée ne vient pas de moi, et cela permettrait même de faire entrer les gens qui ne souhaitent assister QUE à la deuxième présentation... Oh là je mets le souk là. Dites, vous en pensez quoi vous ? Vous pouvez laisser un commentaire en dessous ? Je sais qu'il faut s'inscrire sur le Touilleur Express pour soumettre un commentaire. Je fais cela pour deux raisons : éviter le spam bien entendu, et ensuite revendre 3000\$ votre adresse à tout le monde. Certaines adresses ne valent pas un clou sois-dit en passant.

**Dernière présentation sur le Typage et la Généricité** J'ai du boulot... Alexandre a fait une grosse présentation, une qui demande 3 heures de travail pour restituer le tout. Je vais faire plus court. Alexandre part d'un message passé par Emmanuel Bernard sur Twitter, en galère avec les templates et la généricité. Exemple que je n'ai pas sous les yeux bien entendu. Petit rappel d'Alexandre, la différence entre le Typage Statique et le typage Dynamique, entre le Type fort et le typage faible, entre l'Heritage et le Sous-Typage, afin de suivre la suite. Je retiens tout d'abord que bien que Java soit plutôt "Typage Statique", le bytecode généré est du "Typage Dynamique". Lorsque les Génériques ont été ajouté à la version 5 de Java, le choix de conserver une compatibilité au niveau du bytecode entraîne des choses pas très belles dans le bytecode. Pour illustrer ce propos, je vais tenter de refaire l'exemple d'Alexandre (je sens que je suis encore parti pour écrire un billet de 10 pages) Voici tout d'abord un bout de code Java 1.4 simple

```
import java.util.*;

public class Test {
    public static void main(String args[]) throws Exception {

        List l=new ArrayList();
        l.add(new Integer(123));
        Integer i=(Integer)l.get(0); // notez ici le cast explicite
        System.out.println(i); // affiche 123

    }
}
```

Je compile puis je décompile ma classe :

```
javac Test.java
javap -c Test
```

Voici le byte code généré :

```
Compiled from "Test.java"
public class Test extends java.lang.Object{
public Test();
Code:
0:   aload_0
1:   invokespecial   #1; //Method java/lang/Object."<":()V
4:   return

public static void main(java.lang.String[]) throws java.lang.Exception;
Code:
0:   new           #2; //class java/util/ArrayList
3:   dup
4:   invokespecial   #3; //Method java/util/ArrayList."<":()V
7:   astore_1
8:   aload_1
9:   new           #4; //class java/lang/Integer
12:  dup
13:  bipush 123
15:  invokespecial   #5; //Method java/lang/Integer."<":(I)V
18:  invokeinterface #6,  2; //InterfaceMethod java/util/List.add:(Ljava/lang/Object;)Z
23:  pop
24:  aload_1
25:  iconst_0
26:  invokeinterface #7,  2; //InterfaceMethod java/util/List.get:(I)Ljava/lang/Object;
31:  checkcast      #4; //class java/lang/Integer
34:  astore_2
35:  getstatic      #8; //Field java/lang/System.out:Ljava/io/PrintStream;
38:  aload_2
39:  invokevirtual   #9; //Method java/io/PrintStream.println:(Ljava/lang/Object;)V
42:  return
}
```

On voit notre constructeur par défaut. Le plus intéressant (alexandre tu me corriges si je me trompe) c'est la ligne 18 lorsque l'on ajoute l'Integer dans la liste. L'ajout dans la liste de l'élément de type Integer s'effectue avec l'appel à l'interface de List, qui ici ne sait pas qu'il s'agit d'un Integer. Tout va bien. A la ligne 26 on dépile l'élément et à la ligne 31 on s'assure que le type retourné est bien un Integer. Voyons maintenant la version Java 5 du même code, avec les génériques. Je change la valeur stockée pour enregistrer 456 au lieu de 123, pour le reste vous connaissez. Le code Java tout d'abord

```
import java.util.*;

public class Test {
    public static void main(String args[]) throws Exception {

        List<Integer> l=new ArrayList<Integer>();
        l.add(new Integer(456));
        Integer i=l.get(0); // plus besoin de cast
        System.out.println(i);

    }
}
```

Voyons si vous suiviez... regardez ce code et dites-moi ce que vous en pensez :

```
Compiled from "Test.java"
public class Test extends java.lang.Object{
public Test();
Code:
0:   aload_0
1:   invokespecial   #1; //Method java/lang/Object."<":()V
4:   return

public static void main(java.lang.String[]) throws java.lang.Exception;
Code:
0:   new           #2; //class java/util/ArrayList
3:   dup
4:   invokespecial   #3; //Method java/util/ArrayList."<":()V
7:   astore_1
8:   aload_1
9:   new           #4; //class java/lang/Integer
12:  dup
13:  sipush 456
16:  invokespecial   #5; //Method java/lang/Integer."<":(I)V
19:  invokeinterface #6,  2; //InterfaceMethod java/util/List.add:(Ljava/lang/Object;)Z
24:  pop
25:  aload_1
26:  iconst_0
27:  invokeinterface #7,  2; //InterfaceMethod java/util/List.get:(I)Ljava/lang/Object;
32:  checkcast      #4; //class java/lang/Integer
35:  astore_2
36:  getstatic      #8; //Field java/lang/System.out:Ljava/io/PrintStream;
39:  aload_2
40:  invokevirtual   #9; //Method java/io/PrintStream.println:(Ljava/lang/Object;)V
43:  return
}
```

A part un sipush au lieu d'un bipush ligne 13... c'est la même chose. **Java 5 et 6 ayant gardé une compatibilité descendante au niveau du bytecode, les Génériques ne sont pas exprimés dans le bytecode.** Nous retrouvons simplement un "checkcast" ligne 32 afin de vérifier l'objet retourné. Et ce, parce que les types des arguments d'une méthode Java sont **invariants**. C'est pour cette raison qu'à la ligne 19 l'ajout dans la liste est générique, on ajoute un Object dans la liste. Ce que l'on retiendra donc : les génériques sont là pour nous aider lors de l'écriture du code, mais au final dans le bytecode, on retrouve les mécanismes de cast pour vérifier le type. Intéressant non ? Alexandre aborde ensuite des sujets comme la JSR-303 "[Bean Validation](#)" qui souhaite ajouter des validateurs sur les POJO. Par exemple, une annotation @NotNull. J'en ai parlé il y a un an exactement sur le Touilleur Express [dans cet article](#) si cela vous intéresse. Avec ce que vous avez lu plus haut, et je pense qu'Emmanuel Bernard nous expliquerait cela certainement mieux, comment sera implémenter ce @NotNull dans le bytecode ? Je pense simplement avec du code que nous aurions écrit autrement, avec un `if(objArg!=null)`... L'implémentation de ces annotations utilise-t-elle la JSR-269 que nous avons vu au début ? Ou utilise-t-elle un moteur d'enrichissement du bytecode comme ASM ou BCEL ? Lisez les articles très intéressants sur le blog d'Emmanuel Bernard : - [Constraint composition proposal in Bean Validation \(JSR 303\)](#) - [Bean Validation \(JSR 303\) hits proposed final draft](#) La conclusion d'Alexandre est que les quelques soucis de Typages dans Java font que des langages dynamiques comme Groovy (qui génère du bytecode java) ont le vent en poupe. Alexandre est un fan de Scala, où les problèmes de typages sont résolus dans le langage et où le typage statique est utilisé à sa juste valeur. Je m'arrête là car il est impossible de tout restituer ici sur le blog, vous verrez la présentation en vidéo dès que nous aurons terminé le montage. **La conclusion de la soirée** Lors de la troisième mi-temps au Falstaff, nous avons finalement pas mal discuté entre les personnes qui font parties de l'organisation du Paris JUG. Je vous laisse quelques photos et aussi quelques vidéos. Spéciale dédicace pour Denis et ses problèmes de "Queue". Pour terminer quelques photos et une petite vidéo faite à l'arrache :

A

bientôt PS: dédicace à ma mon épouse qui rentre de l'assemblée générale des copropriétaires pendant que je blogue...

## Dernière minute : une soirée spéciale lundi 15 juin au Paris JUG

Communiqué de l'équipe d'organisation du Paris JUG: [caption id="attachment\_1470"



align="alignnone" width="140" caption="Eric Evans"]

[/caption] **Lundi 15**

**juin prochain, soirée spéciale Domain-Driven-Design avec la venue en France d'Eric EVANS !!! Dépechez-vous de vous inscrire !** Le Paris Java User Group organise une soirée exceptionnelle dans quelques jours, lundi 15 juin, à Paris, en raison de la venue d'Eric Evans, l'un des fondateurs du Domain Driven Design (DDD) L'événement aura lieu ce lundi à l'EPITA, métro ligne 7 porte d'Italie, au 14-16 rue Voltaire, le Kremlin-Bicêtre. L'heure : 19h30. Le nombre de place sera limité à 180 places, l'inscription se fait sur [le site du Paris JUG](#) comme d'habitude, elle sera active dans la journée de jeudi. Nous vous conseillons de vous tenir informé en vous abonnant à [la liste de diffusion du Paris JUG](#). Eric est l'auteur du livre "Domain Driven Design: Tackling Complexity in Software" publié en 2004 chez Addison-Wesley. Depuis le début des années 1990, il a travaillé sur de nombreux projets de développement informatique, des grands systèmes d'entreprise en technologie objet, avec de nombreuses approches différentes et de nombreux résultats différents. Le livre est une synthèse de cette expérience. Il présente un système de modélisation et de conception pilotée par ce que l'on appelle "DDD" pour "Domain Driven Design". Le Paris JUG est un groupe d'utilisateurs Java ayant pour but de réunir les acteurs du langage (professionnels, communautés open source, institutions, étudiants, enseignants...) afin d'échanger des idées et de

discuter des avancées technologiques de la plateforme Java. Nous nous rencontrons le deuxième mardi du mois. Les réunions sont gratuites, ouvertes à tous et consistent en une présentation technique, une démonstration d'un produit, ou une table ronde. Après la réunion nous nous retrouvons généralement pour un échange informel autour d'un verre et d'un mini-buffet. Merci de faire suivre le message sur vos blogs car vu le peu de temps pour organiser l'événement, nous avons besoin de votre aide pour faire suivre l'information.

## Jazoon 2009 : l'accueil des bloggers

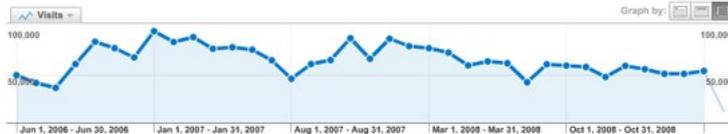
# JAZOON09

THE INTERNATIONAL CONFERENCE ON JAVA TECHNOLOGY  
JUNE 22 – 25, 2009 ZURICH

Plus que deux semaines avant d'aller à [Jazoon 2009](#) en Suisse ! L'équipe d'organisation fait un sacré travail : que ce soit l'inscription, le site ou les conférences, franchement on voit que c'est de l'horlogerie Suisse derrière. A propos de cela, j'ai été contacté comme d'autres bloggers comme [Olivier Croisier](#), par l'équipe de Jazoon afin de participer à la "**Jazoon Bloggers SQUAD**". Felipe Gaucho sera l'hôte des différents bloggers invités par l'équipe de Jazoon, l'idée sera de fédérer les personnes qui écrivent pas mal sur la blogosphère afin bien entendu de couvrir Jazoon. Ce sera l'occasion de partager avec d'autres bloggers nos points de vues, et d'échanger sur le dur métier de blogger dans le monde Java :-). Imaginez un peu la traduction du Touilleur Express dans différentes langues : - Angleterre "The Express Coffee Spoon" - Espagne "El cucharadita muy rapido" - Allemagne "Schnell Kaffee Löffel" Note : I apologize for my poor level of Spanish and German... Rendez-vous dans deux semaines pour vous faire vivre l'aventure de l'intérieur avec quelques surprises... Suivez les organisateurs de Jazoon sur Twitter : <http://twitter.com/Jazoon>

## Fred Cavazza : 50 000 visites cumulées par mois

[Fred Cavazza publie un billet](#) sur son blog à l'occasion de l'anniversaire de sa 6ème année sur la Blogosphère. En quelques chiffres, cela donne le tourmis : 50 000 visites ~~uniques par jour~~ cumulées par mois , 2000 billets publiés, 6 blogs, impressionnant. Voici par exemple ses chiffres Google



Analytics (Updated, merci patrice) Voici la même courbe des cumuls de visites par mois pour le Touilleur Express:

### Tableau de bord

Moyenne par mois des visites sur le Touilleur Express

1 janv. 2008 - 12 juin 2009



Le tourne autour des 8000 visites par mois depuis janvier, soit environ 37000 visites depuis le début

### Tableau de bord

1 janv. 2009 - 11 juin 2009



### Fréquentation du site

37 474 Visites	73,30 % Taux de rebond
63 788 Pages vues	00:01:55 Temps moyen passé sur le site
1,70 Pages par visite	56,39 % Nouvelles visites (en %)

Le Touilleur Express - Visites unique sdu 01 janvier au 11

de l'année. Visites par jour : où je rejoins ses propos c'est sur ce que rapporte le blog en terme de valeur humaine, de rencontres, d'échanges, de discussions... Et le ton parfois décalé du Touilleur Express me permet de me marrer, tout en continuant à me faire plaisir. Fred Cavazza fera une présentation cette année à l'USI sur le thème de "[l'impact du Web 2.0 sur les standards de conception](#)". A suivre donc.

Là

# Eric Evans, Domain Driven Design au Paris JUG

Cher lecteur bonjour. Lorsque tu liras ces quelques lignes, je serai entrain de dormir. En effet, prévu de longue date, ce mardi 16 juin est une journée off, ce qui m'arrange bien par ailleurs. Alors tu t'es inscrit pour la soirée Domain-Driven-Design organisée par le Paris JUG. Tu es arrivé à l'heure, c'est bien. Saches que tu étais 111 personnes ce soir. Qu'il y avait 168 inscrits. Et note cher lecteur que nous avons mobilisé ces 168 personnes en **4 jours**. Merci Twitter, merci aux blogs qui ont fait suivre l'information (SFEIR, Xebia, Zenika) et surtout un grand merci à l'EPITA qui nous a reçu. Antonio a eu vent de la venue d'Eric mardi dernier. Nous en avons discuté mardi soir entre l'équipe, le canal historique (Antonio, David et Zouher) et les dissidents, dit "la Crew" à savoir Tanguy, Charles, José et moi-même. Et motivé comme pas deux, nous voilà engagé pour trouver une salle pour 180 personnes, ce qui fut possible grâce à Tanguy Bayard de SFEIR et Nicolas le Coz de Xebia, ancien de l'EPITA. Arrivé tôt, me voilà déguisé en vendeur, pendant que Tanguy colle les affiches partout dans l'EPITA pour guider nos juges vers l'amphi A04. Quand je dis déguisé en vendeur, je me retrouve à t'accueillir, toi cher Jugeur qui est venu, pas comme les 58 qui n'ont pas pu venir. Cela me permet aussi de démontrer un certain talent à reconnaître les gens, avec un "Salut Laurent" ou un "Salut Claire, Salut Claude" du plus belle effet. C'est ça d'être un blogueur, à force on connaît beaucoup de monde, même si le monde ne vous connaît pas en retour. Le déguisement de vendeur sans doute... Bref passons. **Début** La séance commence. Soyons un peu plus sérieux pour parler contenu. Eric Evans nous présente ce soir [Domain Driven Design](#). L'objectif de la présentation sera de présenter la conception pilotée par le domaine. Aucuns frameworks, pas de mots techniques, du bon sens comme dit mon voisin du Coder's Breakfast sur Twitter. Mais du bon sens sacrément intelligent, ce qui vaut le coup de passer 2 heures avec Eric. Passons rapidement sur le style. C'est vrai qu'il a un côté un peu mou, ce qui rend un peu long la présentation. C'est le fond qui est intéressant. A l'issue de la présentation je suis partagé. Je ne peux pas dire que j'ai été complètement emballé, ce serait faux. J'ai entendu quelques concepts intéressants qui me donnent envie d'aller plus loin. Eric a écrit un livre de 560 pages appelé ["Domain Driven Design, Tackling Complexity in the Heart of Software"](#). La présentation dure 2 heures. L'article du Touilleur doit faire environ 1500 caractères... Bref tu l'auras compris cher lecteur, il sera difficile de te résumer ici tout ce que j'ai vu. Allez bonne soirée à demain. Non soyons sérieux... Voici ce que j'ai entendu et que je dois te raconter. La définition du "domaine" pour commencer, est la sphère de connaissance qui influence notre activité d'après Eric. Tout d'abord il explique qu'il faut se faire aider et encadrer d'expert métier pour comprendre le métier, et donc le modèle de votre logiciel. Il débute la présentation par l'exemple d'un Cargo, qui transporte des conteneurs de Hong-Kong vers San Francisco. Il nous demande : comment modéliser ce transport de marchandise ? La première approche classique, avec une Entité Cargo, est de stocker en base une ligne par parcours. Ce qui l'embête dans cette approche c'est que le Business n'a pas forcément envie d'entendre parler de base de données. Il dit même qu'en travaillant de la sorte, un jour votre MOA débarque avec un schéma de base de données au lieu de vous expliquer le besoin... pas faux. Soit un Cargo qui a une relation de n "Stop". Chaque stop a une date de chargement et de déchargement. Ok ce modèle fonctionne. Prenons ensuite un Cargo qui a une Leg, ou parcours. Ce parcours a un point de départ et un point d'arrivée. Cela marche aussi. Dès le départ, il explique que le modèle dans la conception Objet est une **projection** déformée de la réalité. Pour illustrer son propos, il prend l'exemple d'une carte du monde. On voit dans le premier slide une carte Chinoise. La Chine est représentée au milieu, le reste autour. Ensuite une projection de Mercator avec les USA en plein milieu. Le message passe : quelque soit le vrai domaine, le domaine objet est une projection qui apporte nécessairement une déformation. Pour lui, un modèle est donc un système d'abstractions qui peut être utilisé pour résoudre un problème. Ensuite vient l'utilisabilité. Un modèle n'a de sens que s'il est possible de l'utiliser. Il nous explique qu'il est inutile de tenter de créer un modèle exhaustif, qui sera forcément trop compliqué. Il nous conseille de travailler par itération, et la suite de la présentation va donc s'axer sur la technique de conception de ce modèle. Tout d'abord posez vous cette question **"Qu'est-ce qui est le plus utile ?"**. Prenez quelques scénarii réels de votre client pour piloter votre conception. C'est ce qu'il va faire dans la suite de sa démonstration. Alors vous avez conçu votre Cargo et vos routes, votre bateau est parti de Hong-Kong vers San Francisco. Maintenant prenons le cas où le client décide de changer de destination alors que le bateau est au milieu du Pacifique. Comment votre modèle va-t-il gérer cela ? Il s'avère que le modèle construit sur les Legs (les routes) sera le plus adapté au problème du reroutage là où un modèle conçu avec des "Stops" ne sera pas adapté au changement facile. Eric Evans en profite alors pour nous parler de ce qu'il appelle "ubiquitous language". Lors de la conception objet, il nous demande d'utiliser un langage qui sera compris par tout le monde. Pour illustrer cela, il dit : "si demain vous démarrez un projet avec des Indiens, vous êtes d'accord pour parler et écrire uniquement en anglais non ? alors faites de même pour le design de votre modèle : utilisez un langage universel". Après avoir vu différents diagrammes, il complexifie encore la chose en ajoutant une troisième demande du métier : "Je veux la route la moins chère ou la route la plus rapide pour mon bateau". Il reprend le modèle pour nous montrer que rapidement, ce modèle ne sera pas capable de résoudre cette demande. Cela lui permet alors de nous faire prendre conscience de quelque chose : un modèle n'a de sens qu'avec le Contexte qui l'utilise. Le calcul du "plus rapide" ou du "moins cher" se modélise avec des Arcs, des Routes, du calcul par graphe. Rien à voir avec le modèle du transport, un simple modèle père-enfants relativement simple. Il nous demande alors d'isoler nos modèles selon le contexte, afin de nous garantir de la surcomplexité. De sa propre expérience, la tentation est grande de n'utiliser qu'un seul modèle pour faire tout, ce qui conduit à une architecture pilotée par la technique, au lieu d'une architecture métier. Le prérequis pour un modèle objet complexe est donc la disponibilité d'experts métiers, une écriture par itération, une définition des frontières du modèle très claire et une définition du contexte pour l'utilisation de ce modèle. Lors de la séance des questions réponses, il dira qu'en effet le terme "Business Driven Design" est aussi une bonne représentation de ses idées.

**Conclusion** Nous restons un peu sur notre faim. D'un point de vue conception, je vous avoue qu'il

est très difficile de te retranscrire ici ce que j'ai entendu. A titre personnel j'ai appris ce soir à voir le modèle à sa juste place, avec une prise de conscience du contexte. Ne pas chercher à être exhaustif, à créer des cas d'utilisations qui n'existent pas. Attention à ne pas aussi faire de la modélisation pilotée par la technique. Vous devez être en mesure d'exprimer toutes vos contraintes avec le code. Il cite pour terminer le framework [Qi4J](#) dont Florent Ramière de [Jaxio](#) et [Sébastien Letélicé](#) m'ont déjà parlé. Quand 2 geeks vous parlent d'un sujet, il faut se renseigner. Voir à ce sujet les articles écrits par Sébastien comme [celui-ci](#) par exemple. **Off** La soirée se termine, tu rentres chez toi cher lecteur. Cela me permet de discuter avec des têtes connues ou pas, de rencontrer Patrice Lamarque d'eXo Platform et de discuter de la fusion JBoss Portal / eXo Portal, de parler avec Didier Girard d'une idée pour cette été, pour enfin terminer dans une Brasserie avec les Jugeurs, quelques têtes connues comme Eric "Bob" Mignot et Eric Evans. Il est temps de prendre mon RER A garé en double file, et d'aller dormir. A bientôt, à jeudi soir pour les [Scrumistes](#) et au 7 juillet pour les Jugeurs.

## Google App Engine, Grails et Spring MVC

Je travaille depuis quelques jours avec la version Java de Google App Engine. Je bosse sur une idée de projet pour d'une part tester l'app-engine, et d'autre part... me faire plaisir. Comme dit Eric "Bob" Mignot : faut se faire plaisir. [Google App Engine](#) est une plate-forme de Cloud Computing qui permet de déployer facilement une application Web Java. Voici comment cela fonctionne : vous codez votre application Java en local, vous la téléchargez ensuite vers le serveur de Google, puis quelques instants plus tard votre application est déployée. L'intérêt et la force de la solution de Cloud Computing de Google est d'offrir un serveur très puissant, dont la capacité en CPU, en mémoire, en bande passante, s'**ajuste automatiquement** à l'usage, aux nombres de visiteurs. La différence avec un serveur classique sous Unix comme celui sur lequel j'héberge le blog, c'est la virtualisation totale du serveur. Chez l'hébergeur du blog le Touilleur Express, j'utilise déjà une solution de virtualisation, qui me permet de rajouter ou de retirer des CPUs ou de la bande passante. Cependant c'est à moi de régler cette capacité. Or au début je n'avais aucunes idées du nombre de processeurs nécessaires... J'ai payé pour rien jusqu'à trouver les bons réglages. Et encore, c'est empirique, il n'y a pas de console m'indiquant le taux d'usage de ma machine. Avec la solution de Google App Engine, s'offre à vous un portail Web d'administration qui vous affiche l'utilisation de votre application. Par défaut, gae (Google App Engine) est gratuit. Si vous savez écrire une application Web simple en Java, vous pouvez donc vous inscrire et tester en quelques heures. **CanadaDry, CanadaGrails** Bref me voilà il y a quelques temps devant mon éditeur favori, IDEA IntelliJ, avec une idée de développement histoire de me faire la main. Je décide de tester [Grails](#), un framework en Groovy qui permet d'écrire très rapidement une application. Une fois Grails installé, je récupère ensuite le plugin pour Google AppEngine afin de commencer à créer ma première application. Je décide de tester la version JDO, car après avoir vu la doc de DataNucleus, j'ai le sentiment que JPA est encore un peu trop neuf. Si vous n'avez pas encore testé Grails, prenez 15 minutes pour suivre [le Quick-Start](#) puis ensuite lisez attentivement les différentes pages sur [le plugin Google App Engine](#). J'ai été vraiment emballé de pouvoir faire une application type Crud en quelques minutes. Ensuite je suis quelqu'un de curieux. Alors j'ai garé la voiture "Grails" sur le côté de la route et j'ai commencé à regarder le moteur. Premier besoin : je veux changer une des vues pour afficher un datePicker au lieu d'un champ date. Je teste avec un autre plugin d'interface utilisateur... pas très emballé. Grails vient avec un moteur de plugin qui permet d'ajouter et de retirer des fonctionnalités rapidement. Le souci est qu'il est plus difficile de trouver la raison d'une erreur, et que pour valider vos changements vous déployez beaucoup plus, par peur que ce que vous venez de faire ne marche pas. Je rencontre ensuite un souci de namespace. Voici la solution que vous devrez aussi faire : **Message d'erreur**: Cannot find the declaration of element 'datastore-indexes'. **Solution**: Editer le fichier \$APP\_HOME/target/war/WEB-INF/datastore-indexes.xml et effacer le namespace de la première ligne :

```
xmlns="http://appengine.google.com/ns/datastore-indexes/1.0"
```

Le déploiement vers l'appengine est relativement facile, il faut simplement utiliser l'utilitaire appCfg.sh de GAE la première fois afin de spécifier son mot de passe Google. Par la suite, le plugin de Grails est capable de déployer l'application. Jusqu'ici tout va bien. **Difficultés rencontrées** J'ai eu du mal avec Groovy et Grails. Malgré le support d'IDEA IntelliJ de Groovy, j'ai pas mal de soucis avec le côté dynamique du langage. Bref je patauge pas mal et dès la première difficulté, je peste. Un langage dynamique, cela entraîne en fait pas mal de phase de redéploiement pour s'assurer que tout fonctionne. Je trouve un peu poussif les performances, quoique cela soit purement subjectif. Je parle de Grails + plugin Google App Engine. Ensuite j'ai trouvé que la promesse de l'écriture rapide était bien remplie. Un create entity, un update view, oui Grails est vraiment bien fait, bravo à l'équipe de Grails. Mais la perte de certains repères, du typage fort, représente une difficulté à ne pas négliger. **Spring MVC et SpringFuse** Un peu inquiet de ne pas avancer aussi vite que cela avec Grails, je recrée un nouveau projet Java afin de tester du simple code Java. C'est ensuite en mettant en place Spring MVC que j'ai l'idée de reprendre un projet généré avec [SpringFuse](#). Et hop, en quelques heures j'ai transformé mon application en projet full-Java, quelques JSPs, deux controllers, un peu de JDO pour Google App Engine, bref j'avance beaucoup plus vite. Paradoxal non ? C'est grâce au code de SpringFuse, à la fois simple et en même temps bien documenté. J'ai viré pas mal de trucs pour le faire passer sur GAE, mais rien de bien difficile. J'attends le support de JPA prévu pour bientôt. Sinon Spring MVC 2.5 marche très bien sur Google App Engine, qu'on se le dise. Je n'ai pas utilisé les annotations, ni les transactions par contre. **Premier bilan** Lorsque l'on bosse seul sur son projet dans son garage, et que l'on vise une application massivement Web, paradoxalement je pense que l'on va plus vite en n'écrivant que ce qui est nécessaire. Grails vaut vraiment le coup d'être testé. Le plugin GAE n'est pas encore parfait, et il sera plus difficile de trouver de l'aide sur Google en cas de plantage. Soit vous êtes un aventurier, soit vous êtes plutôt conservateur. A vous de voir. Je ne parle pas sinon des tests pourtant intéressants de GTPL, une idée vraiment intéressante qui permet d'écrire des templates de page en Groovy. Cela mérite un article plus complet. Suite des aventures dans le prochain numéro. Nom de code du projet : GeekDay

## Jazoon 2009

Lundi soir prochain, direction Zürich en Suisse pour participer à [Jazoon 2009](#). En quelques mots, Jazoon est un salon et un ensemble de conférences sur Java qui regroupe idéalement avant l'été les acteurs de la communauté Open-Source Java. En terme de taille c'est le second événement européen, derrière Devovx. A noter la présence plus importante de SpringSource, peu représentée en décembre dernier en raison de l'événement SpringOne en Floride, qui avait lieu en même temps que Devovx. Au programme cette année : EJB3, JEE6, Spring 3, Groovy, Grails, GridGain, DDD, Qi4J, Ajax, JSF, Testing, Portlet 2.0, JPA et bien d'autres sujets. Avec la présence de quelques têtes connues de la communauté francophone comme Emmanuel Bernard (les CastCodeurs/[Hibernate Search](#)), [Jérôme Dochez](#) et [Ludovic Champenois](#) de SUN Microsystems, [Vincent Massol](#) de XWiki, Antonio Goncalves qui sera là aussi, etc. Manque juste Guillaume Laforge pour que [les 4 CastCodeurs](#) soient au complet ! Lundi aura lieu une journée spéciale sur GlassFish avec Alexis Moussine-Pouchkine, les conférences démarrent mardi jusqu'à jeudi, avec une soirée mercredi soir. De mon côté j'axe ma visite autour de JEE6, de Spring 3, de Groovy et du Cloud Computing. Je viens chercher de l'information sur les dernières évolutions, je ne manquerai pas de bloguer bien sûr, et de vous ramener quelques petites vidéos de la semaine. Rendez-vous chaque jour, en compagnie d'autres bloggeurs nous serons plusieurs à couvrir Jazoon, vous nous verrez avec nos teeshirts "Jazoon SQUAD" ou avec mon polo "Le Touilleur Express". Et puis bien entendu, rendez-vous sur Twitter, sur @LeTouilleur afin de suivre Jazoon en direct.

## Jazoon : résumé

C'est parti, après une nuit dans un hôtel tout à fait sympathique, la journée démarre fort. Jugez un peu : arrivé relativement tôt j'ai la chance de tomber sur James Gosling. Ni une, ni deux, hop, me voilà en photo avec Monsieur Java, très sympa et vraiment coool.



Je vous raconterai dans différents billets les conférences auxquelles j'ai assisté, mais c'était une grosse journée, et ce n'est pas terminé. Voici ce que j'ai vu - James Gosling - Spring 3.0 - 7 Patterns of Groovy usage - Graddle the next build system (la même présentation que celle de Zenika) - GridGain - IDEA IntelliJ and Groovy/Grails A suivre sur le Touilleur Express

## Jazoon: James Gosling Keynotes

Après l'introduction de Christian FREI, l'un des organisateurs de Jazoon, voici qu'arrive James Gosling. En quelques mots, James est l'un des fondateurs de Java, le voir et l'écouter en live c'est intéressant. Le bonhomme est fortement teinté Geek, mais son discours est intéressant comme vous allez le voir. Java tout d'abord, c'est 6 millions de développeurs dans le monde, c'est 6 millions de développeurs dans le monde, 10 milliards de JVM déployées, dont la moitié sur des périphériques mobiles. C'est encore 15 millions téléchargement du JRE par semaine. bref Java est "everywhere". James raconte une anecdote amusante : sa fille de 21 ans, étudiante, apprend en ce moment Java... Comme il dit "*imaginez-là encours... elle répond au prof : oui c'est mon père qui a écrit ce machin*" Il nous présente l'histoire de Java, ses origines qui sont proches de l'électronique embarqué, de l'idée de "Write Once, Run everywhere". Aujourd'hui le langage est plutôt "**Learn Once, Work Everywhere**" ce que je trouve plus actuel et correct. Il nous fait aussi prendre conscience que la plateforme Java dans son ensemble est aujourd'hui la vraie force de la communauté, au delà du langage en lui-même. Il est vrai que des langages comme Groovy ou Scala font aujourd'hui partis de la même famille. Il continue ensuite : le réseau a tout changé. Des applications massives sont déployées en Java comme le réseau de santé au Brésil avec le dossier médical. Java est partout, dans votre PS3, dans votre télé, dans votre téléphone portable, dans votre voiture, au dessus de votre tête dans des satellites... Et oui ! Il aborde ensuite les produits de SUN Microsystems, avec une présentation de Glassfish V3 Preview. La sortie de Glassfish V3 est prévue pour septembre, en même temps que Java EE 6 (dixit Alexis qui est à côté de moi). Afin de démontrer la facilité avec laquelle on peut maintenant travailler avec les EJB 3.1, il montre une page JSP à laquelle il ajoute un appel vers un EJB3 en quelques lignes. Il sauvegarde le code, on recharge la page, et hop cela fonctionne. Sans redéployer. Et sans relancer une quelconque factory de Bean non plus, soit-dis en passant. Il aborde ensuite une partie sur Netbeans 6.7 et le projet Kenai, ce qui semble très prometteur. Il parle ensuite de Java RealTime, une version adaptée aux contraintes du temps réel. Viens ensuite la fameuse Loi de Moore, qui depuis 3 ans se tasse. Il semble qu'un plafond ait été atteint pour tout ce qui concerne la vitesse des CPU. Aujourd'hui c'est le nombre de processeurs, de coeurs, qui est important. Et à cet instant, James Gosling a une réflexion intelligente : si vous n'aviez pas la JVM, comment feriez-vous pour écrire un programme capable de fonctionner sur des CPUs multi-cores en étant capable de les utiliser ? Petit tacle gentil à nos voisins du C/C++ Sa présentation, quoique parfois un peu décalée, nous donne une vision assez générale de la place de l'écosystème de Java. Il présente aussi JavaFX, j'ai un peu de mal à

sentir l'enthousiasme, mais pas beaucoup sur le Cloud Computing. Il parle cependant du projet Kenai qui viserait à offrir un moyen de déployer dans le Cloud une application Java EE. A quand un GlassFish V3 "cloud-computed" ? Allez il est temps de commencer la journée des sessions, je reprends le sac et on y va.

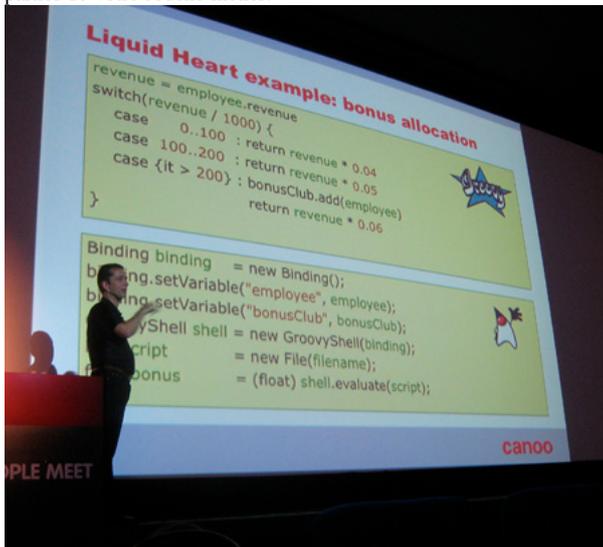
## Jazoon : Groovy seven usage patterns

Première conférence : [Groovy, Seven Usage Patterns](#) par Dierk König. L'objectif de cette présentation est de nous montrer 7 cas d'utilisations réels de Groovy dans le monde de l'entreprise. Voici une photo prise durant la séance :



#1 SuperGlue

Groovy permet de créer rapidement des applications simples en utilisant l'API de base. Dierk nous montre cela en écrivant en quelques lignes un lecteur de flux RSS qui affiche dans une fenêtre de type Swing la liste des derniers billets. Groovy est un moyen rapide de créer de l'assemblage. Il montre la lecture d'un flux XML et le traitement du DOM, puis ensuite la partie Swing. Vraiment intéressant. #2 Liquid Heart Ce pattern d'usage vise à nous montrer comment réutiliser la couche métier d'une application existante pour réutiliser les services avec du Groovy. La démonstration nous montre comment retirer du code métier écrit en dur pour le remplacer par du script Groovy. Chez mon client actuel nous nous en servons pour le calcul du risk par exemple. Ici Dierk nous montre comment externaliser le calcul de bonus d'un salarié dans un script Groovy chargé par Java à l'exécution, et évalué selon les paramètres. Cela vous permet de rendre très configurable certaines parties de votre couche métier.



#3 Keyhole Surgery Ce

pattern discutable vise à ouvrir une porte cachée dans l'application afin d'exécuter du Groovy. Il montre ainsi une application Web Java classique avec des utilisateurs. Une servlet sécurisée (hem...) permet d'exécuter du code Groovy directement via le navigateur. Dans l'exemple qu'il montre, il s'en sert pour remettre à zéro des paniers d'utilisateurs qui sont invalides. Groovy étant un langage de script, il est possible d'intervenir dans votre application Web à distance. #4 Smart Configuration Groovy est bien plus intelligent que des fichiers de configuration XML. L'un des cas d'usage qu'il nous montre ensuite est donc la configuration d'une application en script Groovy. L'injection et la configuration en devienne plus facile. #5 Unlimited Openness Plus qu'un pattern, Dierk parle maintenant de l'accès au code. Il nous recommande dans la mesure du possible de laisser le code ouvert, quoiqu'en Groovy je crois que de toutes les façons c'est le cas. Il nous montre l'exemple d'un moteur de blog écrit en Groovy, que sa société (Canoo) a porté facilement, car le code est accessible. Chaque besoin client étant spécifique, pourquoi ne pas laisser le code ? #6 House-Elf Scripts Pour la suite de sa présentation il s'appuie sur les Elfs de maison d'Harry Potter. Ce sont ces petits gars qui font le sale boulot pour vous, qui ne rechignent pas à la tâche. Groovy est parfait pour automatiser les tâches répétitives comme l'intégration, la construction, la gestion de release. Il nous encourage à aller regarder de ce côté. Cela tombe bien, je compte aller voir Graddle cet après-midi. #7 Prototype Enfin pour terminer, Groovy est un langage particulièrement adapté pour la création de prototype. L'interface Swing se fait rapidement avec

Griffon, une application Web se fait rapidement avec Grails, c'est l'essence même de Groovy. Au final en sortant j'ai appris encore de nouvelles choses sur Groovy. Lorsque Dierk tapait cependant son programme en Groovy, j'ai vu qu'il s'en sortait car il maîtrise son API à la perfection. L'aspect dynamique doit cependant vous faire prendre conscience que la maîtrise de Groovy demande un peu plus de temps.

## Jazoon : Spring 3.0 what's new ?

Après Groovy, j'attaque ma tournée par une conférence d'une personne absente de Devoxx 2008, Jürgen Holler de SpringSource, pour sa présentation sur [Spring 3.0](#). Il nous présentera une version actualisée par rapport à JavaOne, l'avantage d'être à Jazoon. [caption id="attachment\_1549" align="alignnone" width="500" caption="Jurgen Hoeler SpringSource Jazoon 2009"]



[/caption] Voici le

plan de sa présentation: - Java 5 Themes - REST - EL expression language - Annotated web controllers - RoadMap Dans un premier temps, Jürgen Hoeler, que j'avais interviewé [en novembre dernier](#), nous donne la vision de Spring. La version 3.0 sera tout d'abord le support de Java 5, avec l'utilisation massif des annotations. Il explique par exemple que dans Spring MVC, l'objectif est d'apporter les améliorations via les Annotations. Pour cette raison il nous encourage à utiliser celles-ci plutôt que le code classique pour les Controllers par exemple. Le support de l'expression language (EL) permet d'offrir des fichiers XML et des classes encore plus configurable. L'exemple qu'il a prit nous montre ainsi une DataSource dont le nom exact de la base n'est connu qu'à l'exécution. Si vos fichiers de configuration XML de Spring sont déjà configurables à l'aide de propriétés externes, cela vous semblera familier. Viens ensuite la présentation de REST. Dans l'architecture de Spring 3.0, le support de REST sera là, et vu le temps passé pour nous en parler, on pense à Java EE 6 et au profile Lite qui n'a justement pas... JAX-RS en standard. Le support de la JSR-303 (Bean Validation) aussi présent dans Java EE 6, sera offert par Spring. Comprendre par là, que les annotations seront reconnues et que la validation des beans sera effectuée par le framework Spring. Intéressant à noter. Les Portlets 2.0 retiennent mon attention, c'est l'un des sujets en ce moment chez mon client. Tout d'abord il explique que la spécification Portlet 1.0 étant assez incomplète, le support des Portlets 2.0 par Spring devrait permettre d'offrir un socle simple pour écrire des Portlets. Il cite JBoss Portal et eXo Portal. Concernant le support de Java EE 6, Spring 3.0 attend la sortie de la version finale de Java EE 6. Tout ce qui est JSF2.0 et JPA2.0 ne sont donc pas encore prévu dans la version qui sortira... courant juillet. Tout du moins pas officiellement, mais en early preview. Un retour sur Java 5 pour nous montrer ce qu'il y a de nouveau. Tout d'abord grâce aux génériques, le framework offrira un typage plus fort comme dans cet exemple:

```
BeanFactory returns typed bean instances as far as possible
T getBean(String name, Class requiredType)
Map getBeansOfType(Class type)
```

Concernant l'exécution de tâches asynchrones, Spring 3.0 proposera un TaskExecutor (ce n'est pas déjà le cas ?). La gestion des tâches asynchrones sera plus facile. Pas de mentions des Annotations. A propos de Spring WebFlow : la nouvelle version propose une version révisée de la gestion du binding et de la conversion de type. A propos ensuite (vous suivez ?) de la gestion des Annotations standards de Java, Spring offre le support des annotations du projet Java Config, la gestion de l'instanciation tardive des Beans avec le mot clé @Lazy. A propos de l'Expression Language j'ai pris un exemple de l'un de ses slides pour vous montrer comment il serait possible d'injecter une propriété à l'exécution :

```
@Repository
public class R {
    @Value("#{systemProps.dbname}")
    public void setDatabaseName(String dbName){
        ...
    }
}
```

N'importe quel de vos Beans Spring est accessible avec l'EL, ce qui permet aussi de récupérer la valeur d'une propriété d'un autre Bean pour l'injecter en paramètre... Je ne vous parle pas des

problèmes d'instanciations tardives, car il est tard. Concernant le support de REST avec Spring 3.0 : Spring MVC offre un support de REST accessible facilement et qui permet d'ajouter à une application Spring MVC le support de REST. Par contre si vous souhaitez utiliser JAX-RS... ce n'est pas possible. C'est l'un ou l'autre... "On attendra de voir ce que les clients veulent"

```
@RequestMapping(value = "/rewards/{id}",method = GET)
public Reward reward (@PathVariable("id") long id){
    return this.rewardsService.findReward(id);
}
```

Du côté du code, on note aussi l'apparition d'annotations supplémentaires (youpi déjà qu'on avait terminé tout l'autre stock, sympa d'en remettre encore) pour par exemple lire les paramètres des entêtes HTTP ou des Cookies, ce qui peut être utile pour la gestion de la localisation par exemple. Les 2 annotations sont @RequestHeader et @CookieValue. **Conclusion** Voilà pour la présentation de Spring. Jürgen est un bon speaker qui présente son sujet de manière synthétique. On sent la pression pour sortir en juillet la version, avant la sortie en septembre de son copain Java EE 6. L'un n'ayant d'ailleurs rien à voir avec l'autre. Next please ! PS: j'ai rectifié l'orthographe de Jürgen, il suffisait de regarder le premier slide au début de cet article.

## Jazoon : Gradle la présentation qui aura lieu jeudi soir...

Cher lecteur du Touilleur Express. Après avoir publié comme tu le sais de nombreuses fois à 2h du matin pour que tu aies ton numéro du Touilleur à lire au petit matin, sache que j'ai un nouveau concept à te présenter. J'ai assisté à la présentation qui va avoir lieu ce jeudi à Paris chez Zenika. Et donc tu peux soit attendre jeudi et te faire ta propre opinion, soit lire et voir tout le mal que je pense du speaker. Note que je viens de spoiler un peu... mince. Alors sur le fond tout d'abord, je garde la forme pour la suite. Hans Dockter est le responsable du projet [Gradle](#), un ancien commiter de JBoss IDE, et un formateur chez Skill Matters. La présentation sera animée, avec du code et un exemple intéressant à suivre. Tout d'abord la première question : pourquoi Gradle ? Ah il y a une autre question avant dans la salle : c'est quoi Gradle ? Gradle est un système de build écrit en Groovy qui permet de construire une application, de la packager, de gérer les releases, la version sur le repository du code source, bref de faire ce que fait Ant/Maven. Il utilise d'ailleurs les dépendances de Maven sans problèmes (qui a dit "pas comme maven" ?) Hans Dockter a présenté Gradle en axant dans sa première partie les défauts des autres outils comme Ant ou Maven. Là où il a tout à fait raison, c'est que Ant est bien trop simple. Il demande ainsi à la salle si nous avons une idée du nombre d'éléments de la DTD d'un script Ant ? la réponse est 5 : Tasks, Properties, et j'ai oublié les 3 autres... Bref cela fait que pour des tâches répétitives, il faut systématiquement définir ses tâches, comme JUnit. Bon ok. Gradle vise à rendre l'impossible, possible. A rendre le possible -> facile. Et le facile -> élégant. Il explique que le modèle des scripts Ant date des années 97-2000, que depuis nous avons besoin d'une solution plus souple. Alors ensuite Maven. Maven 2 offre beaucoup de fonctionnalités supplémentaires, voir peut-être trop d'après lui. Le support des plugins, des dépendances transitives, le build-by convention, bref tout ceci fait que Maven est plutôt un bon système. Il reproche simplement la verbosité du XML, le fait qu'il soit difficile de n'exécuter que certaines tâches, la complexité de certains plugins, le manque de consolidation dans la manière de gérer et configurer les plugins, bref trois fois rien... Alors arrive Gradle. A cet instant de la présentation cher lecteur, je suis un peu dubitatif. J'ai horreur des gens qui vendent un médicament en critiquant les autres médicaments au lieu de parler de la maladie. J'étais donc assez fatigué d'entendre depuis 15mn du vent sur "Ant il est tout pourri" et "Maven nous emmène jamais au heaven"... A ma gauche j'ai quand même [Vincent Massol](#) de XWiki, l'un des CastCodeurs, mais surtout l'un des auteurs de [Maven: A Developer's Notebook](#), livre à lire d'urgence. Et je sens que Vincent n'est pas traversé par le saint-esprit de Gradle. Je me trompe Vincent ? Hans Docket ouvre (enfin) son IDE favori (IDEA IntelliJ) sur sa machine favorite (Mac Book Pro) et il est temps de voir la tête d'un script Gradle.

```
task hello << {
    println 'Hello world!'
}
```

Après avoir défini une "task", comme il l'explique rapidement : un script Gradle est un script... Groovy. Cela donne accès à toute l'API, ce que je trouve bien. Disons que vous souhaitez afficher l'heure dans votre build, nul besoin d'écrire un quelconque plugin, un simple println ladate et c'est tout. La gestion des dépendances entre Task est limpide :

```
task hello << {
    println 'Hello world!'
}
task intro(dependsOn: hello) << {
    println "I'm Gradle from Jazoon"
}
```

Si j'exécute ce code voici ce que cela donne :

```
> gradle -q intro
Hello world!
I'm Gradle from Jazoon
```

Prenons ensuite un script build.gradle complet pour vous montrer ce que cela donne :

```
usePlugin 'java'

sourceCompatibility = 1.5
version = '1.0'
manifest.mainAttributes(
    'Implementation-Title': 'Gradle Quickstart',
```

```

    'Implementation-Version': version
  )
  repositories {
    mavenCentral()
  }
  dependencies {
    compile group: 'commons-collections', name: 'commons-collections', version: '3.2'
    testCompile group: 'junit', name: 'junit', version: '4.+
  }
  test {
    options.systemProperties['property'] = 'value'
  }
  uploadArchives {
    repositories {
      flatDir(dirs: file('repos'))
    }
  }
}

```

On est d'accord pour dire que c'est clairement moins verbeux qu'un script Ant ou Maven. Simple et efficace. Par la suite il explique que Gradle est construit par Gradle, que la documentation par exemple est testée. Les exemples de code qui sont dans le manuel de Gradle sont testés par Gradle ! La documentation est ainsi mise à jour sur le site Internet de Gradle automatiquement, la classe non ? **La critique, la critique, la critique** Sans langue de bois : mauvais speaker. Un peu fouillis, on sens que Gradle vaut le coup d'être regardé mais ce n'est pas la bonne personne pour en parler. Les exemples de scripts m'ont donné cependant envie d'y jeter un oeil, et il a de bons arguments intéressants... mais bon, voilà. Du côté de Gradle lui-même j'ai un truc qui ne m'a pas plus : la lenteur. Son script HelloWorld démarre poussivement en 2 secondes. Désolé mais mon système de build doit être rapide. Si je vois que ce système qui me fait remplacer 1 fichier pom.xml de 60 lignes par un script de 20 lignes multiplie par 3 le temps de compilation : poubelle. Allez voir jeudi soir la présentation chez Zenika, et prenez une autre opinion que la mienne, qui est très personnelle. On est sur un blog personnel, je peux largement me planter, mais je ne crois pas à Gradle pour l'instant.

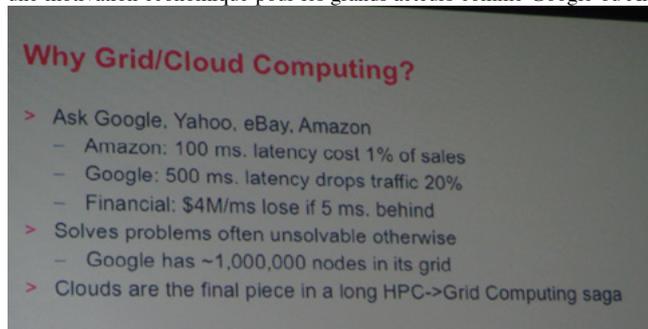
## Jazon : GridGain puis IDEA IntelliJ with Groovy and Grails

2 petites présentations plus courtes ensuite : GridGain par Nikita Ivanov de GridGain Systems. Puis ensuite le support de Groovy et de Grails dans IDEA IntelliJ. **GridGain** GridGain est une solution open-source de grille. La présentation sera construite avec 20% de slides et 80% de code. C'était une présentation vraiment intéressante, qui permet de voir la construction d'une Task exécutée en parallèle sur une grille. Nikita rappelle d'abord quelques définitions



Il nous demande

ensuite : mais après tout, pourquoi faire du Grid Computing ? Pour expliquer cela, il montre quelques chiffres des coûts lorsqu'un service comme eBay n'est pas disponible. Par exemple 100 millisecondes de latency sur Amazon.com et c'est 1% du chiffre d'affaire qui s'envole ! C'est donc une motivation économique pour les grands acteurs comme Google ou Amazon.



GridGain est l'un des

projets les plus utilisés, avec un exemple sur Amazon EC2 de 512 clusters déployés. Bien entendu le multicast n'étant pas disponible, GridGain utilise d'autres supports pour découvrir dans un réseau ses voisins. Je ne vais pas vous retranscrire ici l'exemple complet, que vous pouvez retrouver de toutes les façons sur le site de [GridGain](http://GridGain.com). Nikita a quelques phrases clés comme "If you cannot

partitionate your application, you simply cannot scale". L'exemple de traitement d'une tâche qu'il nous a montré se base sur des Annotations et des Aspects pour rendre "clusterisable" une partie du code. Un point important : lorsque vous mettez à jour le bytecode d'une application sur une machine A, la nouvelle version est repliquée sur toutes les machines du cluster automatiquement. Il nous demande "vous m'avez-vu faire un FTP ou une tâche Ant pour mettre à jour mon cluster ? non ! car c'est trop compliqué, c'est pour cette raison que GridGain s'en charge". A noter si vous avez quelques soucis avec le déploiement de vos versions en grille. Bonne présentation qui donne envie de tester GridGain rapidement. **IDEA IntelliJ et Groovy/Grails** Simplement "Awesome" et hallucinant. J'ai pris de Grails fonctionne, certes le présentateur est l'un des lead-développeurs d'IntelliJ... Mais quand lorsqu'un éditeur est bien utilisé, quels gains de productivité !!! Impressionné par le support de Groovy, les cas tordus liés au typage dynamique qu'IntelliJ résoud sans problèmes... Il écrase Eclipse sans problèmes. Du côté de Grails j'ai personnellement testé, et je peux vous dire qu'en effet le support de Grails fonctionne, mais qu'il y a encore quelques améliorations à faire pour que cela soit parfait. La prochaine version d'IDEA IntelliJ 9 offrira un peu tout ce qui est à la mode en ce moment comme le support de Google App Engine (aujourd'hui c'est un plugin dans la 8.1), le support de Gradle, d'autres trucs sympatiques et bien geeks. IDEA IntelliJ est gratuit pour ceux qui sont committers sur un projet Open-Source, il est aussi pas si cher comparé au temps gagné. Ainsi s'achève la journée de présentation.

## Jazoon : day 1 when all mics are off

Dernier billet pour cette première journée pour vous raconter un peu aussi ce qui se passe en dehors des conférences. Tout d'abord un mot pour ceux qui sont allés à Devvoxx, une page People où je raconte qui j'ai croisé, et enfin un avis sur la première journée. Comparé à Devvoxx ce que j'apprécie ici à Jazoon c'est l'accessibilité des Speakers. J'estime à 600 personnes finalement le nombre de personnes présentes, je n'ai pas de chiffres officiels. Cela facilite du coup les rencontres. Ce matin j'ai le plaisir de voir James Gosling, qui se demande ce que lui veut ce petit jeune avec son polo "*Le Twoueurre express, what is it ?*". C'est pas énorme une chîte photo du monsieur qui a inventé Java ? (celui de gauche)



Ensuite on croise quelques têtes connues comme Ludovic Champenois, Roberto Chinnici et Antonio Goncalves, qui parlent de Java EE 6



Pour vous donner une idée de ce que je veux dire par "les gens sont accessibles" j'ai dîné ce soir avec [Kirk Pepperdine](#), avec [Roberto Chinnici](#) de SUN Microsystems, spec lead de Java EE 6, avec Alexis Moussine-Pouchkine, avec Mike Keith d'Oracle, specification leader de JPA 2.0 et Antonio, bref une soirée sympathique. Concernant les repas, le wifi : rien à redire. Pour vous donner une idée de l'organisation Suisse, les premiers rangs dans les amphis sont avec des rallonges électriques afin de

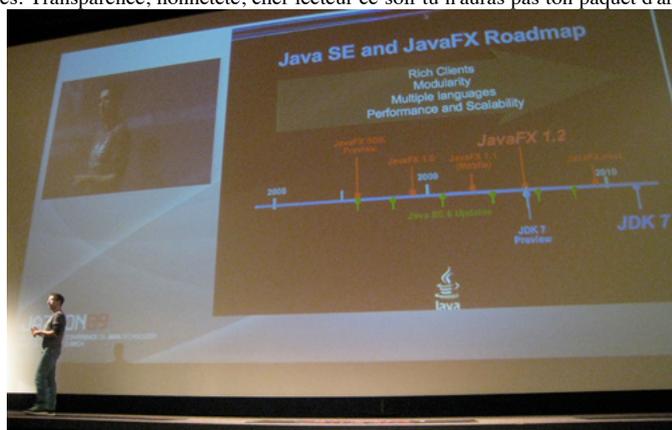
brancher son laptop. Il y a même des adaptateurs de prise pour le format suisse. Le repas est sympa, salade et saucisse en discutant avec Guillaume Mathias et Pablo de Xebia France. J'ai vu aussi ce matin Christophe Jollivet du [Tours JUG](#), ainsi que Jérôme Petit du [Poitou-Charentes JUG](#) qui organise d'ailleurs le 1er juillet une soirée Java EE 6. Première journée intéressante. Temps pourri par contre, mais cadre agréable. L'espace exposant est bien plus petit que Devovx, l'ambiance est détendue et un peu moins studieuse, pas de conférences casseroles où je pars au bout de 2mn, bref tout bon. Demain grosse journée, et demain soir il y a une soirée organisée par l'équipe de Jazoon.

## Jazoon day 2 : when you understand that you're at the good place



Jazoon journée 2.

Tombons tout d'abord directement le masque : il sera difficile ce soir de sortir 7 articles comme hier soir. Le responsable est blonde et s'appelle bière, le contenu de ce blog étant réalisé sans trucages je me rends compte qu'il va être TRES difficile de bloguer ce soir, surtout après quelques bières. Transparence, honnêteté, cher lecteur ce soir tu n'auras pas ton paquet d'articles. Ca c'est



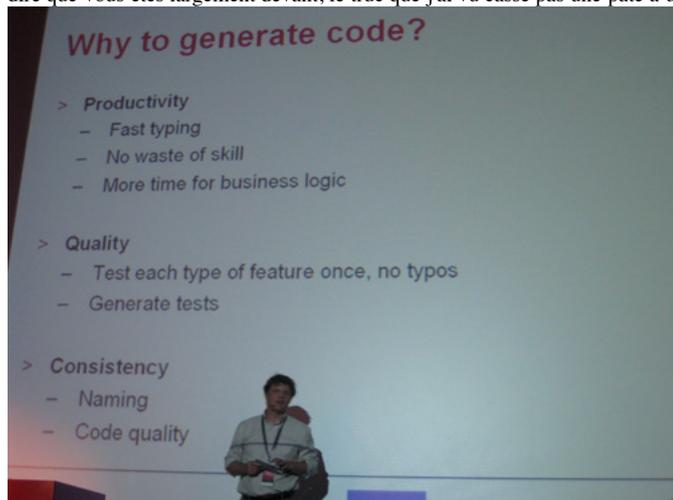
fait. La deuxième

journée a commencé par une KeyNotes de SUN Microsystems sur JDK 7.0, orienté autour des 5 nouveautés les plus importantes de Java, puis des 5 nouveautés de JavaFX les plus importantes. J'y reviendrai dans un article un peu plus tard. J'ai ensuite assisté dans l'ordre et sans trucages aux conférences suivantes: - *Refactoring of Large Software Systems revised* : conférence qui me parle très concrètement par rapport à ma mission actuelle. Comment faire pour reprendre un projet vieillissant, très intéressant. - *Next Generation Wikis : mixing content oriented application with Wikis* pas Vincent Massol. Intéressant de voir ce qu'il est possible de faire rapidement avec XWiki essentiellement. Photo de Vincent:



- *A memcached implementation in Java* par Bela Ban de JBoss/Red Hat. La meilleur conférence pour moi. Du

contenu très pragmatique et très intéressant. - *Practitioners approach to generate object-relational persistence and services* une conférence sur la génération de code, j'y suis allé pour comparer avec SpringFuse et pour en parler avec Nicolas et Florent de Jaxio. Et bien les gars vous pouvez vous dire que vous êtes largement devant, le truc que j'ai vu casse pas une pâte à un boiteux. Photo:



- Hibernate Envers,

*Entity Auditing* mini-conférence sur Envers, qui m'a donné une image intéressante et précise d'Envers. A creuser - *Gaming with JavaFX : developping the Next Generation of Casual Games* par Jose Maria Silveira Neto, contenu sympa mais un peu court. Son blog permet d'apprendre JavaFX en prenant l'axe "création de jeux du type Zelda". Intéressant pour comprendre la position de JavaFX. Ensuite nous nous sommes tous retrouvés pour la session Jazoon Rookie. En quelques mots : Jazoon Rookie est un concours qui offre la possibilité de venir à Jazoon tout frais payés si vous avez moins de 26 ans et que votre présentation est retenue. En retour, et c'est ce qui s'est passé durant l'heure à laquelle j'ai assisté, nous avons vu 3 présentations par 3 présentateurs plutôt juniors. Un jury constitué de personnalité comme Bela Ban ou James Gosling, a ensuite jugé la présentation de 20 minutes de chacun des rookies. Le public enfin a voté à l'applaudimètre pour la meilleur présentation. Le speaker utilisait une application avec son iPhone pour mesurer les décibels et donc l'applaudimètre. Marrant et très bonne ambiance.



Les 3 gagnants de

Java Rookie remportent une énorme cloche pour vache, humour Suisse bien sympa dans une ambiance conviviale, ainsi que quelques goodies. Ce moment était sympa, il nous a donné l'occasion de voir des petits jeunes en direct. Imaginez parler devant 500 personnes en anglais... Enfin, et j'y arrive car j'ai un truc à raconter, nous nous sommes tous retrouvés pour la soirée Jazoon. Avec quelques sandwiches et quelques bières, rapidement les francophones se regroupent. Essentiellement des JUG : Tours, Poitou-Charentes, Lyon, Lorraine. Quelques personnes comme [Anthony Dahanne de Valtech](#), Pablo Lopez et Guillaume Mathias de [Xebia France](#), Pierre-Antoine Gregoire d'AgilePartner et surtout un monsieur... dont je vais parler dans quelques instants. Nous voilà autour d'une table avec quelques français. Je me lance dans la discussion et je raconte une anecdote qui vient du Paris JUG. [Jean-Michel BEA](#) de FastConnect me racontait que les gens lui demandait souvent : "alors c'est toi Jean-Michel Doudoux ?", ce à quoi il répond à chaque fois, que non, ce n'est pas lui. Mais qui est ce Jean-Michel vous allez me dire ? C'est le monsieur qui a écrit un sacré nombre d'articles sur Java, dont un tutoriel plutôt connu, riche de plusieurs milliers d'articles sur Java appelé "[Développons en java](#)". Il y a pas mal de monde qui a débuté en Java en lisant ses articles, lorsque l'on voit la quantité de travail en 10 ans que cela a demandé, cela force le respect. Son site personnel d'ailleurs vous donnera une idée de la quantité et de la qualité du boulot, <http://www.jimdoudoux.fr>. Bref me voilà parti dans l'explication de l'anecdote, jusqu'à ce que la personne à ma gauche me tende poliment son badge Jazoon, afin que mes yeux lisent "J-E-A-N M-I-C-H-E-L DouDoux"... Ah mince, je me banane et la personne en question est juste à ma gauche, éclats de rires et petit moment sympa. Donc bref nous voilà avec Jean-Michel à discuter, car je connais son boulot et il lit aussi le *Touilleur Express*, photos de nous ensemble que je vous ferai suivre, très sympa.



Ensuite nous parlons

GWT. Je lui explique alors qu'à Paris je connais un petit gars qui s'appelle Didier Girard, gars très sympa, et je tartine en lui racontant qui est le bonhomme. Jean-Michel garde un silence poli, puis ensuite me retend son badge Jazon. Ok je relis "jean-michel Doudoux, SFEIR Benelux". Ah... donc il connaît aussi Didier... Re-grand moment de solitude où l'on se marre comme des baleines. Bref monsieur le Touilleur Express ce soir en a fait 2 à la suite bien marrante. Discussion ensuite sur Twitter, Jean-Michel n'est pas convaincu. Je lui explique que Twitter permet par exemple d'envoyer des petits mots doux à ceux rentrés tôt ce matin pour travailler comme un certain A.Goncalves. Hop, ni une, ni deux, un petit coucou à Antonio... qui retourne une réponse 2mn plus tard (mais tu dors jamais ???). Twitter is magique. Cette deuxième journée était axée pour moi autour des sujets de ma mission actuelle. La soirée et l'ambiance détendue de Jazon permettent de lier connaissance, de vous ridiculiser si vous parlez d'une personne alors qu'elle est à votre gauche et qu'elle ne vous reconnaît pas, de discuter avec les JUG leader. Spéciale dédicace (on se croirait sur NRJ...) à [Christophe Jollivet](#) du Tours JUG qui m'a sauvé la vie avec le transformateur du Mac Book Pro et les prises électriques Suisse. Prendre le chargeur de l'iPhone, déclipser la prise, la mettre sur le chargeur du MacBookPro, done. Il a aussi proposé l'idée suivante : contacter les universités et les écoles afin de proposer aux étudiants en stage de fin d'année de présenter leurs projets sur Java à des JUG, pour organiser un French JUG Rookie Contest, afin d'élire le meilleur Rookie. Nous avons aussi discuté de l'idée de monter un événement francophone type Devox ou Jazon en France, avec des conférences en français afin de se distinguer de nos 2 voisins. Je trouve l'idée vraiment intéressante, avoir des speakers en français comme Emmanuel Bernard, Guillaume Laforge, Vincent Massol, Antonio Goncalves... ça le fait non ? Sur ce, je suis désolé de ne pas sortir mes 15 billets habituels, je vais devoir rattraper cela demain. Jazon journée 3 à suivre. [Blog de Jean-Michel Doudoux](#) Voir aussi [le post de Jean-Michel sur la soirée](#)

## Soirée JavaFX au Tours JUG le 8 juillet prochain

Je fais suivre une annonce du Tours JUG de Christophe Jollivet: *Le mercredi 8 juillet aura lieu une soirée spéciale JavaFX. Les applications riches ont créé un nouveau standard en terme de sophistication et d'interactivité du contenu des applications. La demande pour ce type d'application est d'ailleurs grandissante sur un grand nombre de plateformes. JavaFX SDK et Java 6 ont rendu facile l'intégration des éléments d'applications riches dans les applications Java à destinations des plateformes telles que les ordinateurs de bureau, les téléphones ou les télévisions. Au cours de cette soirée, [Simon Ritter](#) évangéliste chez SUN Microsystems nous donnera un aperçu des capacités graphiques et multimédias de JavaFX mais aussi des outils disponibles pour créer ses applications. Mercredi 08 juillet à 19h00 nous vous attendons dans les locaux de SUPINFO, 15 place Michelet 37000 Tours. La soirée se terminera autour d'un buffet offert par Apside. Les événements des JUG sont gratuits et ouverts à tous alors n'hésitez pas à vous inscrire pour profiter de ces moments de convivialité et de partage sur le site du Tours Java User Group Venez nombreux et n'hésitez pas à relayer l'information dans votre entourage, une affiche est disponible en téléchargement sur notre site. L'équipe Tours JUG <http://www.toursjug.org> A noter que Simon présentera JavaFX au ParisJUG la veille, le mardi 7 juillet donc, pour les parisiens. Plus de détails sur le site du [ParisJUG](#).*

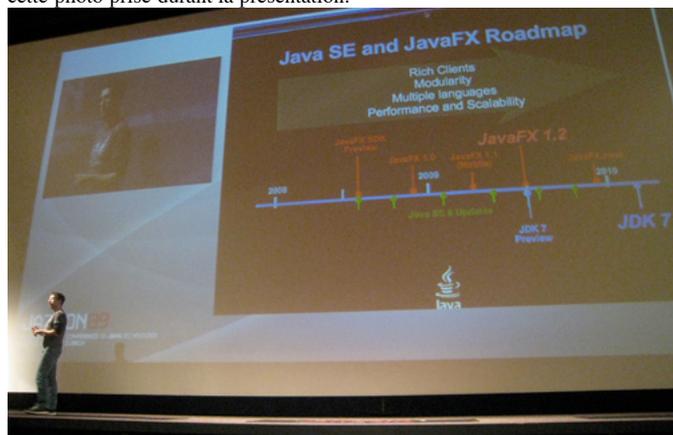
## Jazon day 2, Refactoring of Large Software Systems revised

L'objectif de cette présentation est nous faire partager un retour sur expérience d'un projet de refactoring d'une application de gestion classique en Java. Présenté par Sibylle Peter et Sven Ehrke de Canoo Engineering AG, la présentation était intéressante car très ancrée dans la réalité. Prenons tout d'abord un constat simple : dans la vraie vie, notre travail d'architecte senior et de consultant n'est pas toujours d'apporter de l'innovation. Nous sommes souvent mandatés par un client pour revoir une architecture, introduire de meilleurs pratiques, bref faire de la chirurgie esthétique, voir

parfois de la chirurgie lourde. La présentation nous a montré la démarche de l'équipe. Tout d'abord un point pour comprendre la démarche projet: cette équipe de consultants est venue de l'extérieure pour auditer puis refactorer une grosse application. Refactorer, en utilisant le verbe, signifie appliquer un ensemble d'opération visant à revoir le code d'une application dans le but de l'améliorer. Concernant le pourquoi de ce refactoring, les 2 speakers citent tout d'abord la dette technique. 8 ans de code, 140 entités, un énorme ensemble de code Java, un turn-over important dans l'équipe qui fait que personne ne sait plus pourquoi telle classe fait telle chose... bref tout ceci me parle. Une courbe nous montre le cycle de vie d'un logiciel. Il y a un point d'inflexion où le coût de maintenance excède le coût de développement d'une nouvelle fonctionnalité. Trop de code, cela fragilise forcément un produit. Introduire une nouvelle fonction devient de plus en plus risqué et complexe. Le projet étudié comprend 320 JSP, 1800 classes Java, très peu de documentation. A cause du turn-over des prestataires, peu de personne connaissent l'ensemble du produit. Les tests unitaires ne sont pas nombreux, et lorsqu'ils sont là, ils ne sont pas toujours écrits pour les parties les plus critiques. Bref un sacré projet. La démarche a constitué à prendre le code dans une branche séparée, à introduire des patterns de refactorisation, afin de découper le code. Cela a permis ensuite de retirer le framework Web propriétaire pour le remplacer par un framework moderne. Le découpage en couche, l'introduction de Facade et de DTOs, bref de la chirurgie plastique classique. L'ajout d'objet métier léger a aussi permis de remplacer des paquets de "String toto" par des objets légers intelligents, des BaseTypes Objects. Ces objets ont permis aux développeurs de faire de la validation légère au lieu de se trainer des String, relativement bêtes. Une classe simple déclarée permet par exemple de limiter la taille d'un champ à 10 caractères, de s'assurer que les 2 premiers caractères sont des chiffres. Pour cela un BaseType Object avec des expressions régulières permet de valider à la construction les objets. C'est du design orienté par le modèle. La couche Service sera structurée avec des points d'entrées et de sorties. Elle permet aussi d'agréger différents services. Je ne suis pas forcément d'accord avec la présentation, mais n'ayant pas le contexte complet il est difficile de juger de la pertinence de la solution présentée. Concernant la gestion de projet et d'équipe, les 2 speakers ont exposé quelques principes Agile, sans citer XP ou Scrum. Mais là encore, la remise en question du processus a permis de développer une nouvelle version plus rapidement. Une petite phrase : "Without changing it's observable behavior" Enfin les présentateurs parlent de l'industrialisation du projet. Installation d'un moteur d'intégration continue, mise en place de construction toutes les nuits, gestion des tests, déploiement chaque nuit, etc. En conclusion, pour réussir un projet de refactoring il faut un plan général, la confiance du client, un ingénieur connaissant l'ancienne version, une approche pragmatique et itérative, de bons outils, un peu de bol aussi, et du temps. J'ai bien aimé cette présentation qui nous parle, à nous les petits vieux appelés en mission chez des clients pour faire du bouche-à-bouche à du code de 95 ans. Refactorer une application demande un sacré paquet de compétences : honnêteté, vision, courage, esprit pragmatique, confiance, engagement...

## Jazon 2 : SUN KeyNotes, quoi de neuf sur JDK 7 et JavaFX ?

La journée de mercredi débute par une présentation d'une heure de Danny Coward de SUN Microsystems. L'objectif de sa présentation sera de parler des 5 points les plus innovants du JDK 7 puis des 5 points les plus innovants dans JavaFX 1.2. JDK 7 est prévu pour 2010 comme le montre cette photo prise durant la présentation:

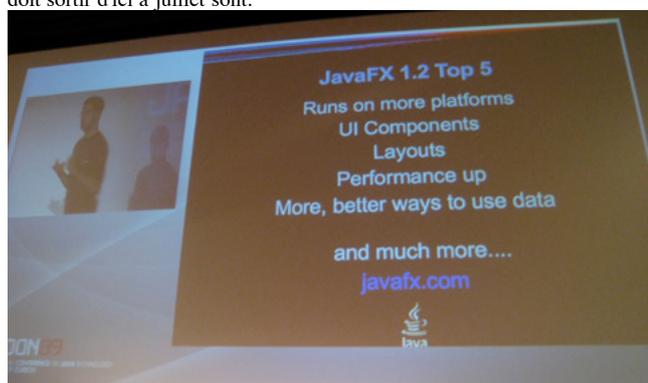


importants du JDK 7 :

Les 5 points les plus



La modularité est un sujet fleuve, qui à titre personnel ne m'intéresse pas pour l'instant est-il le bon ? C'est pas ma tasse de thé. Quelques ajouts dans le langage Java ([projet Coin](#)) devraient nous faciliter un peu la vie, la nouvelle API d'entrée/sortie est intéressante, le nouveau Garbage Collector G1 dont SUN souhaite rendre payant l'utilisation en production est aussi là, bref ce sujet étant largement couvert je ne m'y attarde pas plus que cela. La deuxième partie de la KeyNote m'intéresse un peu plus, puisque je vois JavaFX à nouveau après quelques mois. Les démonstrations vues à JavaOne nous donnent une idée de la puissance de JavaFX. Les 5 points les plus importants de Java FX 1.2 qui doit sortir d'ici à juillet sont:



La présentation donne en une heure un bon état de l'art, de l'avancement de JavaFX. Bien entendu nous attendons tous les outils et le support des périphériques mobiles comme Android pour vraiment regarder JavaFX. James Gosling disait la veille que la moitié des JVM dans le monde sont des téléphones portables, il serait dommage de ne pas travailler sur ce sujet dans les mois qui viennent je pense.

## Jazon day 3 : Adrian Coyle

Jazon troisième journée. Après une soirée animée, un réveil difficile où vous vous dites que vous n'avez plus 20 ans, retour ce matin dans la plus grande salle afin de participer à la KeyNotes de SpringSource. Sur scène, un très bon speaker en la personne d'Adrian Coyle, CTO de SpringSource. La présentation s'intitule : [The Changing Nature of Enterprise Java Application Development](#) La lumière s'éteint, début du show... Adrian Coyle : notre écosystème Java est comme la Forêt Vierge : une végétation dense, des petits arbres, de grands arbres, beaucoup d'animaux, qui se battent... On voit des strates, [la canopée](#) est l'étage en contact avec la lumière du soleil. Les arbres les plus grands, 20 mètres plus haut que les autres. En dessous de cela, la canopée, ceux qui bénéficient de la lumière. Pour un oiseau c'est cool. En dessous, pas top. La vie est difficile si vous êtes en dessous, 5% de la lumière du SUN (soleil) arrive jusqu'au sol. *Adrian prend le temps de laisser les images s'imprimer...* Le sol de la forêt ne reçoit que 2% de la lumière. Mais de temps en temps arrive une éclaircie dans la forêt. Dans ce cas, arrive une explosion de la vie, jusqu'à prendre peut-être la place des autres. Peut-être même une nouvelle canopée. Regardons les arbres géants : ce sont les grosses compagnies (IBM, SUN, Oracle...). La canopée est l'espace de vie pour les mediums-size players. Il constate qu'il est très dur pour une nouvelle idée, une nouvelle approche, de faire sa place. Oracle rachète SUN. Pour lui, un grand arbre est tombé. Cela crée un trou de lumière qui touche le sol, ce qui va entraîner du changement. Un ange passe.. Il cherche à nous faire prendre conscience qu'en ce moment, il se passe quelque chose. Où en sommes-nous dans notre forêt vierge ? Au couché du soleil, en anglais au [Sunset](#). SUN Microsystems se fait racheter par Oracle, nous sommes à la fin de quelque chose pour lui. Adrian montre la courbe de la quotation de SUN pour dire que la compagnie descend tout doucement. Après le couché du soleil, vient la période sombre (Dark Age) mais nous aurons ensuite une nouvelle aire, de l'innovation, un renouveau. *(A cet instant je vois presque Rod sur son cheval blanc... une larme coule... que c'est beau)* Slide suivant, les temps changent. Regardez, dit-il, la présentation de SUN la veille. Parmi les 5 premiers points les plus importants dans le JDK 7, nous voyons que le support des langages dynamiques est cité dans cette liste. C'est très important, vous devez comprendre que les alternatives à Java, qui devient parfois un peu lourd ou complexe pour certains sujets, se trouve dans les nouveaux langages dynamiques, pas dans une quelconque JSR pour les Closures en Java. En avons-nous besoin d'ailleurs ? Il continue ensuite en expliquant que l'explosion de nouveaux langages basés sur la JVM comme Groovy ou Scala doivent nous faire prendre conscience rapidement qu'un nouvel arbre est entrain de pousser dans la forêt. Adrian

affiche ensuite un slide avec différents langages: - Groovy, - Erlang, - Clojure, - Jython, - JRuby, - Ruby, - Scala. Il continue ensuite: Java est peut-être trop compliqué. L'introduction des Generics serait le début de cette complexité. D'autres disent que Java bouge trop lentement. D'autres pensent qu'il y a trop de fonctionnalités. Java est trop ceci, trop cela... Il nous demande à notre avis quel sera le langage qui aura le plus de chances de pousser dans cette nouvelle forêt ? 4 langages se détachent dans sa sélection Groovy JRuby Clojure et Scala Il revient ensuite sur [Groovy](#) qui a remporté le "Script Bowl" lors de JavaOne 2009 (voir cet article sur le Touilleur Express). Dans les slides qui suivent, il montre un exemple Groovy vs Java, largement discutable. Mais l'idée est de dire que l'on peut faire la même chose avec moins de code. Ok, à cet instant je suis d'accord sur la quantité. Mais il passe aussi sous silence le changement de paradigme. Programmer avec un langage dynamique est forcément différent par rapport à Java. Concernant la productivité, Groovy est plus concis, se concentrant sur l'efficacité. Adrian montre l'exemple du script HelloWorld, qu'il transforme en script Groovy. Il passe à un slide en Groovy avec 10 lignes au lieu de 30. Et il fait rire la salle en disant "et c'est la même taille de police". Adrian explique ensuite en quelques minutes les principes de Groovy.

```
class HelloWorld {
    def name
    def greet() { "Hello ${name}" }

    static main(args) {
        def helloWorld=new HelloWorld(name: "Groovy")
        println helloWorld.greet()
    }
}
```

Concernant ensuite la gestion des threads, Adrian montre un exemple en Scala.

```
class Pong extends Actor {
    def act() {
        var pongCount=0
        loop {
            react {
                case Ping =>
                    if(pongCount % 1000 ==0)
                        println("Pong: ping "+pongCount +" from + " sender)
                    sender ! Pong
                    pongCo = pongCount + 1
                case Stop =>
                    println("Pong : Stop")
                    exit('stop')
            }
        }
    }
}
```

Pour suivre un exemple avec Clojure, langage du domaine fonctionnel comme OCaml.

```
;; Hello

(use 'clojure.contrib.test-is')

(defn hello
  "return a greeting"
  ([] "Hello World Default")
  ([username] (str "Hello " username " !")))

(println (hello "World"))

(deftest test-hello
  (is (= "Hello World" (hello "World")))
  (is (= "Hello World" (hello "World")))
  (is (= "Hello World Default" (hello)))
  "Called with no arg, we default to World !")
)
```

Il pose ensuite une question à la salle: A votre avis, quel est le langage qui a **le plus de chances** d'être le plus populaire ? Il ne nous demande pas le meilleur langage, question à laquelle il n'y a pas de réponses. Il nous demande simplement quel serait selon-nous les chances de chacun de ses langages ? Clairement Groovy se détache du lot. Il conclue en expliquant de manière très pragmatique : "That's a reason why we bought G2One.Inc" SpringSource recherche la meilleur technologie, le meilleur framework, afin d'aider à faire émerger de nouveaux arbres dans la forêt vierge. Il nous demande : seriez-vous prêt à retravailler en assembleur plutôt qu'en Java ? Non bien entendu car vous êtes plus productif en Java. Il nous encourage ensuite à regarder Groovy comme un moyen d'améliorer encore cette productivité. Et un jour il espère nous demander : "seriez-vous prêt à retravailler en Java ?". Réaction plus retenue dans la salle, certains ont le sourire, d'autres moins. Mais l'objectif ce matin est de nous remuer et de nous faire prendre conscience que quelque chose est entrain de se passer. Adrian Colyer parle maintenant des frameworks. Nos besoins sont assez communs et connus, lors de la création d'une application Web. Si nous acceptons que les langages dynamiques peuvent nous faire gagner du temps, il est logique d'accepter que des frameworks nous aident à améliorer notre productivité. Un exemple de site : [Wired.com](#) 12 millions de lecteur cité sur le site de SpringSource utilise Grails. Aegon est une autre compagnie citée. Adrian montre ensuite un écran 3270 avec 2 couleurs. Il rappelle qu'il y a peu nous avions des applications développées en mode texte, avec des touches de fonctions pour naviguer. En 2009 nous sommes aujourd'hui complètement dans un autre univers, avec la présence du réseau. Il lance ensuite une charge sur la lourdeur des applications J2EE classiques, trop lourdes. Le déploiement d'une application écrite avec de vieilles technologies comme J2EE 1.4 est difficile. Alors vient OSGI et les modules. L'objectif est de livrer une application modulaire, afin de ne déployer qu'une

partie, et donc de faciliter le déploiement. SpringSource cherche avec Spring dm Server à proposer une autre approche. Tout d'abord il rappelle que le cycle de vie de déploiement ne s'arrête pas à la construction de votre War. Le déploiement et l'administration sont aussi très importants. D'où le besoin de rendre facilement administrable une application. *We live in an Interesting times...* s'affiche sur le slide suivant... C'est la fin de sa présentation, qui aura le mérite de lancer le débat sur l'avenir de Java et de la plateforme. La présentation s'est axée autour de la prise de conscience que les choses sont entrain de changer. Le Cloud Computing, les langages dynamiques, la modularité, on ne sait pas encore ce qu'il va se passer mais comme il le dit, nous vivons en ce moment des moments de changement intenses. Le futur arrive, que l'on aime cette idée ou pas, le Futur arrive à nous. **Analyse et Réflexions** Quoiqu'un peu polémiste, sa présentation vise tout d'abord à faire prendre conscience des changements qui s'opèrent dans notre industrie. Le rachat de SUN Microsystems par Oracle est un événement, dont la portée est encore mal estimée selon les analystes. SpringSource s'engouffre certainement sur une pente un peu glissante en se présentant comme le chevalier blanc. Quoique le framework soit excellent, le temps nous dira si en effet tout change, et donc le choix d'une solution plus légère, modulaire, basée sur des langages dynamiques, constitue le bon choix ou non. En tous les cas, il y a une guerre froide qui ne dit pas son nom, qui devra éclater pour que la communauté continue à diriger Java et son avenir. Je pense que la professionnalisation de l'open-source expose aujourd'hui les débats d'idées entre les grands contributeurs. Là où il y a quelques années, les guerres entre IBM et SUN Microsystems se déroulaient entre 2 commerciaux chez le client, négociant le cuir de leur Porsche de fonction, aujourd'hui un barbu de JBoss réglera son compte à un barbu de SpringSource autour d'une bière ou au board d'une JCR.

## Jazon Day 3 : Portlet 2.0 by Thomas Heute

Première présentation ce matin : "Portlet 2.0" par Thomas Heute de JBoss/Red Hat. La spécification des Portlets 2.0 est sortie en juin 2008. Thomas nous donne un retour un an après de la conférence. Lui-même contributeur depuis 2004 sur JBoss Portal, une année sur JBoss Seam, il est aujourd'hui le responsable du projet JBoss Portal chez RedHat. L'agenda de sa présentation sera :

- Quelles sont les nouvelles fonctionnalités de la version Portlet 2.0 - Adoption par les vendeurs -
- La portabilité - Portlets versus Gadgets Dans un premier temps, retour sur la JSR-286 Portlet 2.0 sortie l'an passé. Tout d'abord l'une des innovations les plus importantes est ce qu'il appelle "Resource Serving". L'objectif est de permettre l'envoi de données vers le navigateur qui ne font pas partie du portail, comme un fichier PDF ou Excel par exemple, tout en conservant les fonctionnalités d'une portlet. Cette spécification vise donc à offrir un accès au contexte afin d'assurer le support de la sécurité et de l'accès aux données par exemple. Il montre ensuite l'interaction entre les portlets avec une RemoteCommandPortlet sur une démonstration fonctionnant avec JBoss Portal. Une fenêtre HTML détachée permet d'envoyer des commandes à une autre portlet, la gestion de l'asynchrone est très simple. A ce propos, il explique que les améliorations du support d'Ajax dans la spécification Portlet 2.0 permettent à des frameworks comme ICEFaces ou RichFaces de fonctionner correctement dans des portlets. Les Portlets 2.0 peuvent maintenant partager leurs paramètres avec d'autres produits ce qui permet de faire communiquer les portlets entre elles. Thomas Heute montre l'exemple d'une portlet Google Map et d'une portlet Weather qui reçoivent toutes les deux en paramètre le code postal d'une ville américaine. Une portlet peut aussi déclencher un événement, les autres portlets peuvent être notifiées et se mettre à jour. Il conseille de ne s'en servir que pour de la mise à jour de l'interface. Sur le slide suivant il dresse un état de l'implémentation de la spécification par les différents acteurs du marché :
- La JSR-286 Portlet 2.0 est en version finale depuis le 12 juin 2008.
- JBoss Portlet Container 2.0 est sorti le 13 juin 2008 puis inclus dans JBoss Portal 2.7.0 en octobre
- eXo Portlet Container 2.0 est sorti le 18 juin 2008, mis dans eXo Portal le 1er juillet 2008
- Liferay portal a sorti sa version le 17 juillet 2008
- Apache Jetspeed assez tardivement le 27 mai 2009
- dernier - IBM Websphere portal 6.1 a été disponible sept 2008
- Oracle Portal (BEA) + Oracle (2) + Sun(1) ?

Concernant le support des frameworks WEB il existe 2 spécifications JSF Support :

- JSR-301 specification addresses JSR-166 - JSR-329 addressing JSR-286 (JBoss Portlet bridge on the way)

De ce qu'il a pu voir sur les forums, les pages : Wicket support -> mostly there with support for events Sruts 2 plugin for JSR-286 Spring MVC also support JSR-286 WebWork A propos de la portabilité : attention la spec ne dit pas comment on fait un portail mais une portlet. La notion de page n'est pas définie. La communication entre portlet: - UI Level - Le mécanisme événementiel n'est pas un remplacement par JMS, ne pas abuser de ce mécanisme. - Doit-on se coordonner à travers les pages ? - Portlets aren't a template mechanism per se Rester à une gestion des événements entre portlet qui sont sur la même page si vous ne voulez pas avoir de soucis. Slide suivant qui m'a fait sourire : **Portlet vs Widget vs Gadget (vs Toilet ?)** Thomas Heute explique tout d'abord en quoi consiste un Gadget, tel que l'on peut le voir sur Facebook. C'est un affichage déporté dans un autre site web d'une fonctionnalité hébergée sur un serveur primaire. Une portlet est plutôt la représentation locale d'un service distant, c'est une composition gérée par un portail, afin d'assurer des services communs (authentification, autorisation, communication, etc) dans un seul endroit. Il cite différentes initiatives autour des Gadgets : Google OpenSocial, OpenAjax Alliance (OBM, Google, Sun, Red Hat, Oracle, Microsoft) -> Widget proposal L'avantage des Portlets par rapport à un Gadget/Widget: -> Fait pour des applications plutôt complexes. -> Vous pouvez utiliser votre framework en général, -> L'état des portlets est gardé lorsque la page se rafraichie. -> propagation de l'identité, de la sécurité, interaction avec un portail, visuel. Les gadgets ont l'avantage d'être développable rapidement, comme pour consommer par exemple un flux RSS. Facile pour l'intégration du côté client, facile pour partager de l'information. On pense au Gadget Twitter que j'ai sur mon blog par exemple. Il termine par un slide sur ses idées pour l'avenir > We could use more annotations a la Servlet 3.0 or JSF 2.0 @Portlet (no portlet.xml) > We could all benefit from a PORTAL specification - more aware of the environment, communication with the portal itself - Portal object management - Federated search/index Enfin il revient sur l'annonce du partenariat entre JBoss Portal et eXo Portal. Il résume cela par "**JBoss builds the platform, eXo Platform builds the solution, We are hiring !**" **Réflexions et conclusion** Présentation intéressante, je crois que les Portlets 1.0 ne doivent plus être utilisées. Je

travaillerai cet été à l'étude de différents portails pour le client avec lequel je travaille, le sujet est donc d'actualité. Mon souci du moment : comment proposer l'étude d'alternative à la politique d'achat qui voudrait que l'on prenne Oracle Portal, sans savoir si celui-ci correspond à nos besoins ? Mais c'est une autre histoire...

## Jazon day 3 : Hibernate Search

J'ai assisté à la présentation d'Hibernate Search par Emmanuel Bernard, après une présentation sur les Portlets 2.0 par Thomas Heute. Tous les deux de JBoss/Red Hat. L'objectif est de nous présenter le fonctionnement d'Hibernate Search à travers des exemples de codes. Emmanuel assure la présentation avec un ordinateur de secours, son portable n'ayant pas supporté le voyage. Ouch ! Armé du package du bon speaker : un Mac + IDEA IntelliJ, il débute la présentation par une courte introduction afin de se présenter. Membre de l'équipe d'Hibernate, JCP Specification Leader de la JSR-303 "Bean Validation" et membre de l'expert group sur JPA 2.0. Il est l'auteur du livre "[Hibernate Search In Action](#)" chez Mannings. Enfin petite information, il revient vivre en France en octobre, ce qui permettra je l'espère de le voir lors des soirées des Java User Group en France. Son objectif sera de nous expliquer en quoi consiste la recherche "full-text", ce qu'elle peut nous apporter, la magie des analyseurs, les différents algorithmes pour la recherche approximative, bref comment Hibernate Search fonctionne. **La recherche** Que ce soit un formulaire sur un site web ou autre, tôt ou tard une application souhaite offrir une fonction de recherche dans les entités d'une base de données. Hibernate Search vise à offrir une solution efficace et puissante, basée sur Lucene, afin de nous assister lors de l'écriture de cette fonctionnalité clé dans les applications. Une recherche simple basée sur la syntaxe SQL est trop restrictive. Si vous tapez "car", comment extraire ce mot d'une colonne ? Certes il y a des requêtes comme "LIKE" mais celles-ci coûtent chères en terme de performance. Il explique que la base de données est le premier point de contention, qu'il est donc dommage de ne pas remonter cette fonction afin de la rendre plus puissante. Par ailleurs, si vous tapez "car", est-ce que le moteur de recherche sera capable de vous montrer les entrées avec "vehicle" ? Et si vous faites une faute de frappe, est-ce que le moteur sera capable de vous montrer un résultat ? La sélection par SQL est donc très limitée. Emmanuel enchaîne ensuite avec un argument : comment classer par pertinence les résultats ? De ces besoins est né l'idée d'Hibernate Search, offrir un moteur de recherche full-text à Hibernate. **Full Text Search** Pour répondre aux besoins exposés précédemment, la solution s'appelle "Full Text Search". Elle est relativement simple et puissante. Le principe est de créer un Index des mots clés, en supprimant les mots de liaisons, afin d'identifier les occurrences de chaque mot, et donc de créer de la pertinence. Ce principe est très performant puisque l'on ne fait pas d'accès à la base de données. Il existe déjà différentes solutions, soit embarquées dans la base, soit dans des boîtiers externes sous la forme d'application, mais qui manquent de flexibilité, puisqu'en tant que développeur il sera difficile d'affiner la stratégie d'indexation. **Exemples de problèmes** Voici la mission d'Hibernate Search : - Trouver le meilleur document, la meilleure entité selon son critère de choix. - Classer par pertinence - Utiliser un algorithme de similarité pour proposer à l'utilisateur des résultats similaires **Extraction** L'indexation s'effectue tout d'abord en découpant les phrases en mots, ce que l'on appelle l'extraction. A ce propos, la construction de l'index est gérée par Hibernate Search, elle s'effectue automatiquement. L'extraction consiste tout d'abord à découper les phrases et à filtrer les mots communs pour ne garder que le sens. **Approximation** Ce système permet d'assister l'utilisateur en lui proposant des résultats très proches lorsqu'une recherche exacte ne retourne pas de résultat par exemple. Il est possible de dire à Hibernate Search que l'approximation est moins importante que la recherche exacte, de sorte que la liste des résultats ne soit pas polluée et reste pertinente. Ce poids lors de la recherche est configuré par le développeur, ce qui nous permet de régler finement le comportement du moteur. Les réponses exactes apparaissent en premier, puis les approximations. La résolution d'approximation se fait avec des algos comme le calcul de distance Levenshtein. Tapez par exemple Hibernete au lieu d'Hibernate, et vous verrez que le moteur détecte les résultats proches, en calculant la distance entre les lettres. **L'approche n-gram** Emmanuel présente ensuite une fonction plus avancée d'indexation, appelée n-gram. Le principe consiste à découper des mots en paquets de lettre. Par exemple un tri-gram pour découper en paquet de 3 lettres un mot comme Hibernate donnerait : hib-ern-ate. Cela permet alors de créer des arbres d'indexation et de donner des plans de résultats encore plus pertinent. Emmanuel montre un exemple où il tape "poter". les livres sur Harry Potter s'affichent, mais aussi un "Capote" car le tri-gram "pot" a matché :-). **Hibernate Search** je mettrai des photos en ligne un peu plus tard avec des exemples de code Emmanuel prend ensuite l'exemple d'un Item pour nous présenter l'API. On voit une annotation @Indexed placée sur un Entity, ce qui permet à Hibernate Search de savoir que cette entité doit être indexée. Par ailleurs il explique qu'Hibernate Search sait que lorsque l'objet est changé, il doit mettre à jour l'index Lucène. Il nous montre enfin comment configurer l'indexation afin de ne pas découper en n-gram par exemple certains champs, afin de conserver des codes ISIN. La recherche s'effectue comme une requête HQL avec un entitymanager particulier, le FullTextEntityManager qui est une sous-classe de l'EntityManager d'Hibernate.

```
FullTextQuery q = entityManager.createFullTextQuery(luceneQuery, Item.class)
```

Pour construire sa requête Lucene, Hibernate Search laisse le soin au développeur de construire celle-ci, ce qui permet de régler finement le type de recherche. Il commence par faire une démonstration d'une recherche exacte, puis ensuite une démonstration d'une recherche avec un n-gram Analyzer. Il est aussi possible de déclarer par exemple 2 stratégies d'indexation différentes, une indexation exacte et une indexation pour une recherche approximative. Vraiment intéressant de voir la possibilité de réglage d'Hibernate Search. La quantité d'annotations peut faire peur, mais le code est vraiment simple à lire. Quelqu'un demande d'ailleurs s'il serait possible de déclarer dans un fichier XML ce qui est déclaré ici sous forme d'Annotations. **Approximation phonétique** Il existe différents algos pour indexer de manière phonétique des mots : - Soundex - Metaphone (JRSKP) - mostly for latin language Emmanuel explique cependant que ce n'est pas le cas le plus courant dans la vraie vie. **La recherche par synonyme** L'idée est de proposer à l'utilisateur des listes de résultats dont le domaine est proche sémantiquement, une voiture, une bagnole, un caisse

si vous voulez. Pour cela il faut un dictionnaire, que l'on indexe. Cela crée de gros indexes, mais il est possible de construire des dictionnaires par référence. Dans tous les cas, il nous conseille de construire nous-mêmes nos bases de synonymes. **What's the catch** Lucene étant relativement bas niveau, l'intégration avec un modèle, la création et la mise à jour de l'index sont donc des fonctions clés d'Hibernate Search. Celui-ci permet d'apporter facilement des fonctions de recherches très puissantes à une application. Et qui n'offrira pas finalement une fonctionnalité de recherche ? Bonne présentation, détendue, claire, et intéressante. Et en plus avec un Mac et IDEA IntelliJ !

## Jazon day 2 : XWiki and Wiki 2.0

Présentation suivante ce mercredi matin : *Next generation Wikis : Mixing Content-Oriented Applications with Wikis* by Vincent Massol Vincent Massol, de XWiki SAS, mais aussi l'un des CastCodeurs, nous propose ce matin une présentation des nouvelles fonctionnalités de l'application XWiki 2.0. Sa présentation commence par un rappel du concept de Wiki, qu'il appelle Wiki 1.0. Ensuite, l'approche qu'il va développer concernant XWiki 2.0, sera de nous montrer qu'il devient facile de créer des mini-applications hébergées dans un wiki. Il cite quelques exemples d'applications qui seraient faciles à développer avec la future version de XWiki : - Gestion de projet - Système de demande de congés - Notes de réunion - Factures ou notes de frais - Annuaire de contacts - Notes de frais Bref l'idée est de prendre en compte un fait : le Wiki est maintenant bien implanté dans l'entreprise, l'étape suivante est d'ouvrir le moteur afin d'offrir aux utilisateurs des applications, des formulaires, en conservant la philosophie du Wiki. En effet, tout ce qui va suivre sera codé directement dans l'interface de XWiki. Vincent Massol montre quelques exemples de sites comme "France 2025" construit autour de XWiki. La frontière entre un portail et un wiki est très fine. Comme il l'explique, les Wikis 1.0 présentent des données peu structurées. Les blogs ajoutent une journalisation, ce qui structure les données. Les portails d'entreprises ensuite ont structuré l'information par domaine. Ce qu'il va présenter est donc la dernière étape, la structuration des données personnalisés, la possibilité de développer son application. Un outil de développement léger en quelques sortes. L'idée des Wikis 2.0 est donc d'être l'Excel du Web. Un besoin simple ? Plutôt que de développer un gros projet, il explique que le moteur de Wiki doit fournir quelques services simples pour créer ses formulaires, et donc ajouter aux pages une application avec un coût de développement réduit. Il débute la démonstration. L'objet de la démo sera de créer une petite application de demande de congés. Pour comprendre la suite, il présente 2 concepts de XWiki. Une classe est une page, un template. Un Object est une instance de cette classe. Tout d'abord il commence par créer les méta-données associées à la page. Pour cela, un script dans une textarea permet de déclarer les attributs. La syntaxe utilisée est Velocity. Le support de Groovy est ~~aussi prévu~~ déjà présent, le support de Jython et de JRub est en cours (cf commentaire ci-dessous). Cette étape fait un petit peu peur au premier coup d'oeil, on pense à Microsoft Sharepoint qui permet de créer des applications. Si j'avais un bémol je dirai que j'ai eu un peu peur en regardant le code. Mais cependant la syntaxe est simple, et je pense qu'une fois l'outil maîtrisé, cela ne pose pas de soucis. De toutes les façons, ce qui suit est plutôt destiné à des administrateurs de portails, mais je n'ai pas assez de connaissances sur ce sujet. L'intérêt évident d'éditer en direct, c'est que le résultat est tout de suite disponible. Et en effet, je trouve le concept vraiment intéressant. Donc un template Velocity permet de décrire sa demande de congés, ce template est attaché à une page, il est temps ensuite de créer l'interface de saisie. Dans un deuxième temps, Vincent crée une ClassSheet, une feuille de style pour habiller la class qu'il a créé. La demande de congés est constitué de champs simples, de champs de type Date, et d'une liste déroulante d'utilisateurs, automatiquement peuplées par XWiki. Une fois la feuille de style prête, nous voyons alors la page de saisie s'afficher. Le tout en 10 minutes à peine. Le reste est bien de la syntaxe Wiki, vous pouvez alors intégrer des images, des feuilles de style, les méta-données sont stockées dans la version de la page (??). Un langage de requêtage, le xwql, permet de créer des éléments dynamiques, une liste des personnes ayant effectué une demande de congés. Vincent Massol explique ensuite que la version 2.0 de XWiki est polyglotte, elle supporte un nombre assez important de syntaxe Wiki. D'ailleurs comme il le fait remarquer, il est intéressant de remarquer que certains Wikis proposent par défaut un éditeur type Word, là où au départ le wiki a été créé pour qu'un langage simple de balisage soit utilisé ! La version 2.0 supporte aussi le "Roundtrip between XHTML and Wiki Syntax" et ce moteur est disponible comme librairie à part. Parmi les questions posées en fin de présentation, quelqu'un demande si un moteur de Workflow (pour valider la demande de vacances) serait prévu ? Au delà de la question, on comprend que les Wikis 2.0 se rapprochent de la frontière des applications d'entreprise. Est-ce que les utilisateurs accepteront quelques limitations ? Est-il envisageable de développer plus de fonctionnalités tout en gardant la simplicité et la puissance d'un Wiki ? Peut-être qu'un slide pour présenter les fonctionnalités des autres acteurs nous aurait aidé pour comprendre l'avantage d'utiliser XWiki. Bonne présentation qui montre en tous les cas que les Wikis sont devenus des outils incontournables dans les entreprises.

## Jazon day 2 : a memcached implementation in Java by Bela Ban, JBoss

Pause repas en compagnie de quelques français (Christophe Jollivet du ToursJUG et du site [Developpez.com](#), [Pierre-Alexandre Gregoire](#) d'Agile Partner, [Anthony Dahanne](#) de Valtech). Puis reprise en direction de la salle 9. Jazon est organisé dans un grand cinéma style UGC. Les conférences ont lieu dans des salles de cinéma, très confortables et correctement sonorisées. [Bela Ban](#) de JBoss/RedHat est d'origine Suisse. Après un PhD à Zurich, un post-doc à l'université de Cornell, il rejoint la Californie, et il démarre le projet JGroups, intègre JBoss RedHat par la suite, pour s'occuper du projet [JBossCache](#) et [JGroups](#). L'objectif de la présentation sera de nous présenter une implémentation de memcached réalisée en Java, des résultats sur les performances et une discussion sur les DataGrids. Bela Ban est un excellent orateur, sa présentation était très intéressante, animée par une démonstration en direct. Pour commencer, une phrase afin de marquer

les esprits: "**Memory is the new disk, disk is the new tape (Tim Bray)**" La mémoire est le nouvel espace de stockage, finalement le disque dur n'est-il pas devenu aujourd'hui l'équivalent des lecteurs à bande des années 80 ? La base de données est un point de contention, les allers-retours demandent des accès réseaux, ce qui dégrade les performances. Cependant, il n'est pas possible de stocker toutes les données en mémoire sur un seul serveur. Tout d'abord pour des raisons de place, mais aussi parce que la mémoire ne résiste pas à un crash. [memcached](#) est un serveur de cache écrit en C qui fournit un service de cache de type HashMap en mémoire. Le client du cache peut être écrit dans n'importe quel langage, puisque le protocole pour utiliser memcached est très simple. Il y a un Get, un Put, un Remove, c'est une simple HashMap. Des exemples d'applications Webs connues qui utilisent memcached : FaceBook par exemple. 800 serveurs, 28 TeraBytes de données, memcached est un cache massivement répliqué, qui évite des accès disques très couteux en architecture distribuée. C'est donc une grille de données distribuées. ([voir cet article de FaceBook](#)) Le [protocole](#) de memcached propose 6 opérations pour enregistrer ses données, 2 opérations pour retrouver une donnée unique ou un ensemble. Les échanges s'effectuent avec de l'Ascii par dessus du HTTP. Bela Ban explique que l'un des soucis de ce protocole est qu'il n'y a pas d'indicateur de la taille des buffers, ce qui fait qu'il est difficile avec Java NIO de préallouer efficacement des buffers. Il n'y a pas de cache de niveau 1 du côté du client, cela n'étant pas dans l'objet de memcached, bien que certaines implémentations du côté client soient capables de conserver des données. Les données sont stockées avec une date d'expiration, mais plusieurs serveurs memcached ne se connaissent pas entre eux, il n'y a pas de répllication entre les noeuds. Lorsqu'un serveur memcached (c'est un cache rappelons-le) s'arrête, ses données cachées disparaissent, et il faut alors prévoir un reash de l'ensemble de la grille pour redistribuer les données. Imaginez une partie de Poker, l'un des joueurs s'en va, il faut alors reprendre les jetons de chacun et les redistribuer de manière équitable. Cette maintenance et cette fragmentation du cache est coûteuse. **Implémentation d'un serveur memcached en Java** Bela Ban présente maintenant son implémentation d'un serveur compatible avec le protocole memcached, mais réalisée en Java. Tout d'abord dans son exemple de cache distribué, le client en Java dispose d'un cache L1 pour optimiser les performances. Du côté des grosses différences par rapport à l'implémentation en C, chacun des "ReplCaches" (son implémentation Java de memcached) utilise JGroups pour découvrir ses voisins. Les données sont donc automatiquement distribuées de manière efficace dans la grille, constituée par plusieurs serveurs "ReplCaches". Lorsqu'une donnée est placée dans le cache, selon la clé de hachage de cette donnée, la grille de données s'optimise automatiquement. Par ailleurs la répllication entre les serveurs memcached en Java se fait via un protocole binaire optimisé, ce qui permet de bien meilleures performances avec la version Java que la version C. Le protocole peut aussi être adapté, c'est le fonctionnement de JGroups. Si vous souhaitez ajouter de l'encryption, de la compression, il est possible de configurer la couche JGroups finement. Cela réduit l'usage du réseau. Bien entendu, TCP comme UDP sont supportés. **Démonstration** Le "ReplCache" est embarqué dans un conteneur de Servlet, dans le même espace d'adressage. Arrive maintenant le moment le plus intéressant je trouve de sa présentation. Il explique qu'il utilise ses "ReplCache" comme des disques durs physiques. Il est possible de sélectionner une stratégie de stockage pour chaque donnée, afin de faire... du RAID de données. Il est donc possible de définir si l'on souhaite qu'une donnée soit répliquée ou non plusieurs fois dans la grille. Les données qui peuvent être relues rapidement de la base de données pourront être réglées avec un compteur de répllication (repl-count) = 0 Les données critiques de travail de votre session (un panier d'achat) que vous ne devez pas perdre en cas de crashes, peuvent être répliquées dans toute la grille avec un compteur repl-count=-1. Sinon en général, l'usage est de ne stocker qu'une seule fois dans le cache cette donnée, la clé de hash permet de trouver rapidement quel serveur stocke la donnée. Il s'en sert comme d'un espace d'adressage, chaque client sait que par exemple les clés de A à H sont sur le serveur 1 et les clés de I à Z sont sur le serveur 2, ce qui permet vraiment d'optimiser les performances. **JBoss Infinispan** Bela Ban présente ensuite le projet Infinispan, une grille de données open-source basée sur l'implémentation en Java memcached, avec la possibilité de passer des objets de type Runnable, ce qui en fait aussi une grille de calcul. JBoss Infinispan est une implémentation de la JSR-107, la distribution du contenu caché est effectué selon la clé de hachage, afin d'assurer une distribution fine de celle-ci. Il est possible de créer un espace de stockage immense. 100 noeuds avec 2 Gb de mémoire, configuré en copie représentent 100 Gb d'espace de mémoire adressable. La mémoire est le nouveau disque dur comme il le rappelle. Infinispan est aussi compliant JTA et offre le support de JMX pour proposer une interface de gestion performante. Il est compatible avec le protocole ASCII classique de memcached, ce qui permet de brancher un serveur en PHP avec un cache Infinispan par exemple. Bien entendu le protocole binaire dont il a parlé est plus intéressant, et l'un des objectifs du projet sera de fournir rapidement des implémentations du côté client en C et en PHP. Du côté Java, l'API offre aussi une gestion asynchrone pour par exemple stocker de manière asynchrone une donnée :

```
Future putAsync(K key, V value)
```

**Perspectives** Bela Ban termine la présentation avec quelques idées pour la suite. Tout d'abord, réellement développer l'idée que la grille est un système de stockage de données, un file system. Pour cela, il imagine proposer une api type java.io pour utiliser la grille pour lire et écrire des données massivement répliquées. Encore une fois : *Memory is the new disk, why can't we implement JDBC on top of the grid ? use-it as a disk ? And what aboute Hibernate on a grid ?* La mémoire est le nouveau disque dur, celui-ci n'étant plus aujourd'hui que le vieux lecteur à bande destiné à faire une sauvegarde persistente. **Références** Démonstration Java WebStart de ReplCache : <http://www.jgroups.org/jnlp/replcache.jnlp> (lancez 2 ou 3 fois l'application en cliquant plusieurs fois sur le lien) ReplCache Flash demo: <http://www.jgroups.org/movies/ReplCache.swf> Infinispan : <http://www.infinispan.org>

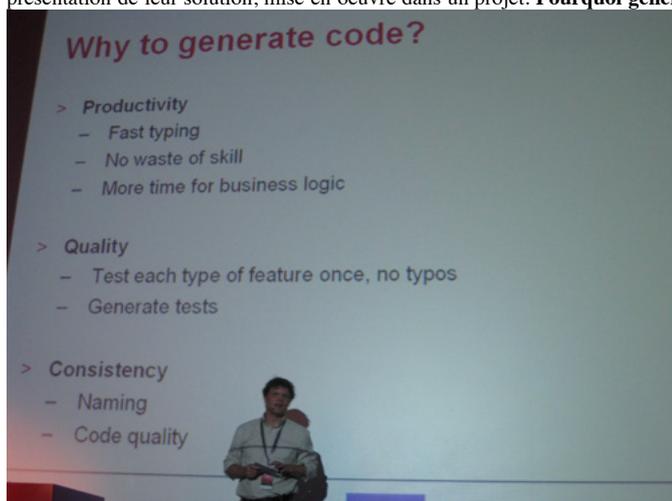
## Vidéo Jazon 2009

J'ai préparé une petite vidéo avec les quelques photos que j'ai pris à Jazon. Rien d'extraordinaire, merci iMovie 09 sur Mac qui permet de créer rapidement ce genre de clip. Au programme : James

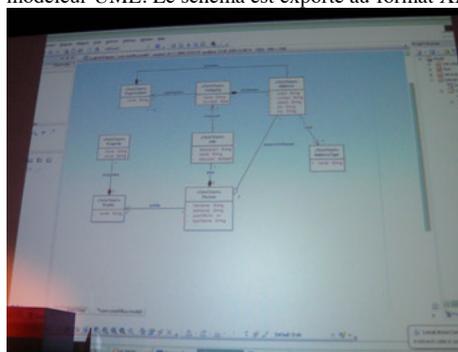
Gosling, Spring 3.0, quelques vidéos de la démonstration de Java Fx 1.2 et Jazoon Rookie. J'ai aussi changé le système de commentaires du Touilleur Express, il est possible de poster sans devoir s'inscrire au préalable. Cela devrait permettre à chacun de commenter plus facilement.

## Jazoon Day 2 : la génération de code

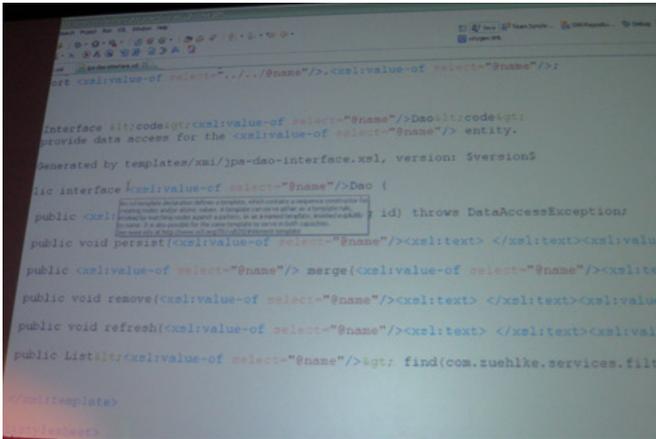
Présentation sur la génération de code. J'y suis allé pour comparer le sujet présenté à [SpringFuse](#), de Jaxio, la société de Nicolas Romanetti et de Florent Ramière, que vous connaissez si vous venez souvent au ParisJUG. La présentation est intitulée "[Practitioners approach to generate object-relational persistence and services](#)". Elle est présentée par Patrick Steger et Marc Baechinger de Zühlke Engineering AG. Tout d'abord, pourquoi générer du code, puis ensuite une présentation de leur solution, mise en oeuvre dans un projet. **Pourquoi générer du code ?**



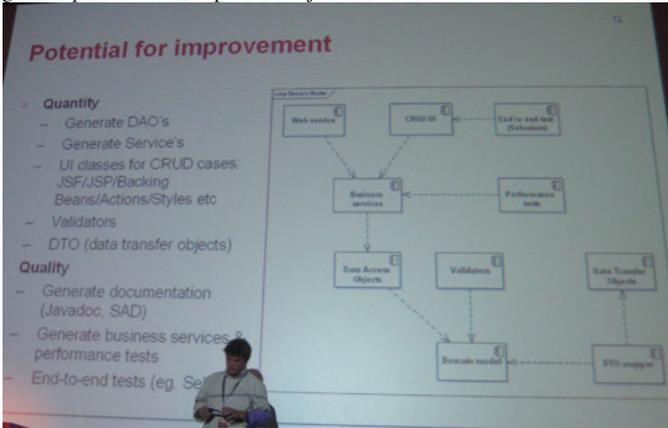
La génération de code permet de gagner du temps en générant rapidement le squelette de l'application, en s'attachant à générer avec une qualité constante la partie purement technique de l'application. Patrick explique aussi que la génération permet de réduire le nombre de bugs. En générant le coeur de l'application, il dégage du temps pour ensuite se concentrer sur la partie métier de l'application. C'est un facteur de motivation pour l'équipe. La solution présentée est basée tout d'abord sur l'utilisation d'un modèle UML. Le schéma est exporté au format XMI.



Des feuilles de style XSLT permettent de générer du code Java à partir du fichier XMI.



La solution est donc réalisable avec un peu d'effort. Parmi les améliorations proposées, les 2 présentateurs parlent de générer plus de code, hop attendez je vous retrouve le slide... voilà



Et c'est là que je me dis, mince quand même, si Jaxio était venu présenter [SpringFuse](#)... Parce que les petits gars d'en face, les DTOs, les Controllers, la vue, SpringFuse génère tout cela, avec une qualité tip top... Je suis allé voir les 2 présentateurs en fin de présentation. Les justifications et les idées sur la génération de code étaient intéressantes. Je pense qu'une partie de nous refuse de voir la génération comme un gain de temps. Certains pensent que ce n'est adapté qu'au [scaffolding](#) (échafaudage), alors que l'idée est de générer le coeur de l'application, de montrer les bonnes pratiques aux développeurs juniors, de s'assurer de la qualité, et de ne pas se planter lorsque l'on manque de temps.

## USI 2009 demain : Teeshirt Linux ou Costard-cravate ?

Rendez-vous demain et jeudi pour [l'USI 2009](#). Cet événement organisé par OCTO Technology vise à regrouper les boss et les geeks, afin de partager une vision métier et une vision technique sur le futur. Cette année avec environ 300 personnes, les 4 thèmes seront la Gouvernance, les Technologies, les Méthodologies et l'Usabilité. J'ai déjà un souci demain : soit Didier Girard avec les dernières technologies Google pour l'entreprise, soit Fred Cavazza avec le Web 2.0... A noter aussi la présence de Neal Ford, qui a fait l'une des meilleures KeyNotes à Jazoon la semaine dernière où j'ai eu le plaisir de l'écouter. Ne manquez pas sa présentation. Revenons un instant au superbe titre de ce post "Teeshirt Linux ou Costard Cravate". Le dilemme demain, c'est la tenue. A votre droite vous avez le kit costard-cravate que je mets de temps en temps, travaillant dans la Finance. A votre gauche vous avez un stock de teeshirt assez impressionnant ("JavaOne", "Debian Rulez", "Netscape Communicator for ever", "SpringOne"). Votre mission ce soir : sélectionner la bonne tenue pour demain. Soit vous partez équipé du kit Blackberry-cravate Hugo Boss, soit vous avez un HTC G1 hacké avec une coupe de cheveux à rendre jaloux un chauve... Demain j'hésite entre allez voir les copains parler, ou alors complètement m'engager pour allez voir des conférences moins geek et plus boss. D'un point de vue purement Le Touilleur Express, comme ici nous sommes complètement geek, le choix est vite fait. D'un point de vue personnel, les sujets "pas-geek" m'intéressent aussi... Alors on verra, surprise sur place ! Plus de détails sur [le programme](#). Vous pourrez suivre la journée sur Twitter, sur [@LeTouilleur](#) et grâce au mot clé [#USI2009](#). Pour terminer j'ai opté pour le polo indémodable "Le Touilleur Express" que j'ai reçu en 2 exemplaires de la part de [SpreadShirt](#). La suite demain...

## USI 2009, Keynote d'ouverture par Joël de Rosnay

L'ouverture de l'USI permet de prendre conscience de l'ampleur de l'événement. Avec 400 personnes, 50 sessions, l'agenda va être chargé. Côté organisation, rien à redire, manque juste un

accès Wifi pour les Geeks. Joël de Rosnay nous a offert une superbe première présentation. Est-ce qu'il vous est déjà arrivé lors d'une présentation de vraiment prendre votre pied en écoutant quelqu'un parler ? Je vais m'efforcer de vous retranscrire les points les plus marquants de sa présentation. Ne vous attendez pas à un résumé précis de son intervention de 50 minutes, la vidéo sera sur le site de l'USI 2009. Je prends plutôt l'axe homéopathique afin de vous proposer quelques points intéressants. La présentation s'axe sur une discussion sur les enjeux de la civilisation du numérique pour demain, de l'arrivée du virtuel et du lien avec le réel. Tout d'abord une phrase : «il y a de plus en plus de virtuel, pourtant les gens cherchent de plus en plus de contact réel» Joël de Rosnay nous brosse un portrait rapide de l'internet et il utilise une image parlante pour personnaliser l'Internet : celui-ci est un énorme cerveau et nous sommes chacun des neurones qui constituent ce cerveau. Il définit ensuite 3 types de clients Internet : l'internet classique où vous surfez avec votre navigateur, l'internet mobile avec votre iphone ou votre gphone et enfin l'internet des objets. Tout d'abord il y a plus de personnes qui utilisent Internet avec un appareil mobile que de personnes qui utilisent un accès classique. Ensuite, la courbe de progression de l'internet mobile dépasse l'internet classique. C'est un fait. Le troisième Internet, l'internet des objets, est la mise en relation quasi automatique des objets du vivant entre eux. Votre voiture qui envoie des diagnostics à un centre technique, votre frigidaire qui vous tweet son contenu... J'invente par rapport à ce qu'il a dit. Alors tout devient cliquable. Le monde des objets est cliquable, mais il est aussi un nouveau vecteur pour la virtualisation. Joël de Rosnay montre une vidéo de l'application Sekai Camera de TonchiDot Team. Imaginez que vous preniez votre iphone, vous mettez le mode caméra, et vous filmez les Champ-Élysées. Le moteur est capable alors d'ajouter des «Air Tags» dans le flux vidéo, de telle sorte que le monde réel se retrouve enrichi par le monde virtuel. Imaginez pointer une vitrine d'un restaurant avec votre caméra, et voir apparaître sur l'écran un tag avec une note de 1 à 5 d'appréciations données par les utilisateurs...

Cela nous amène ensuite à la Maison Intelligente. J'ai en tête à cet instant Marc Fleury, ancien CEO de JBoss, qui travaille maintenant dans ce secteur par exemple. Internet amène ensuite de nouveaux canaux de communication. Ces nouveaux canaux entrent en conflit avec les canaux traditionnels. La communication 1 à 1 : vous à un guichet de la Poste. La communication 1 à tous : la télé ou la radio. Internet amène de nouvelles formes qui se diffusent comme la communication de «tous vers tous» comme sur les forums, les chats ou les blogs... Et il amène aussi une communication de «tous vers un» lorsque vous lisez par exemple une page de Wikipedia écrite par plusieurs contributeurs, ou comme le site Agoravox. Ce site est constitué de 30 000 contributeurs. Avec 1.2 millions de pages vues par mois, c'est un gros média. Les médias classiques justement sont en compétition avec les nouveaux médias. Aux journaux papiers nous opposons les Blogs. A la radio, les podcats. A la télé, les télévisions webs. Aux opérateurs téléphoniques classiques nous pouvons opposer la VoIP avec Skype, etc. Bref il y a une nouvelle vague générée par Internet qui bouscule les médias classiques. Alors comment monétiser finalement tout cela ? Joel de Rosnay présente une courbe de puissance où l'on voit qu'un média comme TF1 touche 18 millions de personnes pendant 25mn au 20h, mais une vidéo sur YouTube sera peut-être vue 100 millions de fois pendant 12 mois sur YouTube. Au final c'est bien finalement ce qu'il appelle «la longue traîne» qui crée la valeur, donc le revenu. Pour gagner de l'argent, est-ce que publier du contenu accessible à une masse de personne ne fait pas plus gagner d'argent que de publier ponctuellement quelque chose ? Il parle ensuite de la dispersion de la quantité d'information. Imaginez un instant le nombre de petites vidéos sur chacun de nos ordinateurs, le nombre de photos, de vieux trucs que vous aimeriez vendre... Impossible pour vous de vendre cette valeur. Il explique alors la nécessité des Aggrégateurs. Google est un agrégateur d'information. eBay est un agrégateur de biens physiques, Wikipedia est un agrégateur de connaissance, Amazon est un agrégateur de livre... Bref l'émergence et la dispersion ne fonctionne que grâce à ces agrégateurs. Ces nouveaux courtiers en valeur d'ailleurs se payent en prenant un tout petit pourcentage sur votre achat. Mais prenez eBay et ses 250 millions d'articles vendus par mois... Les petits courtiers se payent donc avec le volume, sur des flux longs (pour les exemples des vidéos). Point suivant : l'émergence de la société de la recommandation. Aujourd'hui lorsque vous partez en vacances, le premier réflexe est de regarder la réputation de l'hôtel. Vous cherchez une petite amie sur Meetic ? Vous regardez son profil. Vous voulez acheter une voiture neuve ? encore Internet. Vous voulez changer d'entreprise ? bref la recommandation est partout. Cependant c'est une valeur différente de l'information. Pour cette raison il explique que pour lui nous sommes dans une société de la recommandation plutôt que de l'information. La suite des slides nous emmène vers les mondes virtuels, dont on parle moins, puis la réalité augmentée. La notion de bits vers atomes est représentée avec une imprimante 3D qui à partir d'un fichier (des bits) modélise dans de la résine une maquette (les atomes). La personnalisation est aussi un nouveau phénomène qui est rendu possible grâce à Internet. Vous pouvez par exemple créer votre tee-shirt personnel, votre paire de chaussures Nike (voir NikeiD),

vosre tasse de café avec la photo de votre chien... La personnalisation est aussi un nouveau vecteur important. Tout ceci apporte bien sûr un clash culturel. Le pouvoir bascule du côté du consommateur. Celui-ci devient aussi acteur. Prenons l'exemple du Tourisme Peer-to-Peer. Il est possible d'échanger son appartement ou de partir en week-end à Rome et de louer un appartement, sans passer par les centrales de réservation et devoir prendre un hôtel... Pensez à ce concept où vous pouvez offrir de la valeur aux autres sans passer par les grands acteurs. Moi qui blogue dans mon coin, je parle à beaucoup de monde. Toi qui te lance sur MyMusic et qui vend ensuite 100 000 CD, tu te passes des Majors. Il cite ensuite un exemple que je trouve très intéressant et qui va arriver en France : le micro-crédit. L'idée : vous avez besoin de 10 000 euros. Vous allez sur un site de micro crédit, vous donnez une garantie et hop vous avez ces 10 000 euros. Derrière, au lieu d'une banque, c'est 10 000 particuliers qui prêtent 1 euro chacun... Et qui en retour toucheront un petit quelque chose (quoiqu'avec 1 euro...). De manière plus intelligente, j'ai entendu sur le podcast JavaPosse cet exemple d'une entreprise qui propose aux particuliers la possibilité d'accorder des crédits aux pays en voie de développement. Bref cette révolution où une nouvelle forme de pouvoir nous est octroyée, est le découlement direct de l'émergence d'Internet. Dans la suite de la présentation, il aborde la gestion de la publicité et de la réputation. Il explique là encore qu'il est plus intéressant de dépenser un tout petit peu mais sur des points de passage massif (comme Google) plutôt que de manière ponctuelle sur un point où certes il y a du monde, mais où l'intervention est ponctuelle : le spot de publicité à 20h30. C'est ce qu'il présente sur un slide : «Flux + Buzz => Bizz» pour business Il termine par une prédiction, par sa vision sur les sujets qui devraient émerger rapidement. Tout d'abord la notion de P2P (Peer-to-Peer) appliquée à 5 secteurs : la Banque, l'Assurance, le secteur des Consultants, l'Éducation et l'Énergie. Pour la banque, ce sera la possibilité d'emprunter 500 EUR ou de prêter 10 EUR avec un intérêt pour ce prêt. Pour l'assurance, regardez les mutuelles. Pour l'éducation, on pense à ce site qui vous propose de faire vos devoirs. Pour l'énergie, demain j'espère pouvoir revendre via un site type eBay l'électricité produite avec mes éoliennes... Intéressant non ? En conclusion, il termine en expliquant aussi que cela crée de nouveaux risques, un besoin d'éthique, de régulation. La vie privée et Internet, l'eReputation, nous devons nous adapter à ce nouveau monde. Peut-être que demain quelqu'un prendra une photo de moi et me mettra dans une manifestation dans un montage vidéo... L'éthique et la sécurité devront donc aussi être présent dans le futur. Une très bonne introduction avant la journée qui m'attend.

## USI 2009 : Google pour l'Entreprise puis Amazon Architecture

**Présentation "Technologies Google pour l'entreprise"** J'ai finalement craqué pour les présentations plutôt techniques. Je commence ma journée par la présentation de Didier Girard : "Technologies Google pour l'entreprise". Axé sur la définition de la stratégie et la présentation des outils de Google, Didier termine par une démonstration convaincante de Google Wave. D'ailleurs, l'USI a prévu une autre présentation uniquement consacrée à Google Wave demain à 16h30. Il nous a présenté l'ensemble des outils de Google destinés à l'entreprise. La présentation de Google AppEngine est appuyée par l'exemple de l'application développée par Didier, pour Android. Il cite quelques chiffres qui donnent le ton : d'ici quelques années, 15% des CPUs seront achetés par Google, et l'un des plus gros consommateurs d'électricité en Californie sera bientôt Google. Plus tard dans la journée lors de la présentation d'Olivier Mallasi et André Nedelcoux d'OCTO, on apprendra aussi que Google possède à ce jour 39 Datacenters, que 400 serveurs par jour tombent en panne, et qu'ils consomment actuellement 2.5% de l'électricité aux USA. Chiffres difficiles à vérifier. Mais cela explique aussi pourquoi Google veut mettre ses machines au large sur des barges flottantes afin de réaliser des économies. J'ai apprécié de revoir les outils, mais j'ai surtout aimé la démonstration de Google Wave. Didier Girard a développé un Robot pour Wave capable d'envoyer des SMS, et d'en recevoir. En fin de présentation nous avons testé son application en envoyant des SMS et tout a bien fonctionné. Je m'arrête là pour ce soir car je vous reparlerai de Google Wave demain. **Architecturer pour le Cloud** J'enchaîne ensuite avec une présentation en Anglais de Simone Brunozzi d'Amazon : "Architecturer pour le Cloud". Il débute sa présentation par une "non-définition" du Cloud Computing, pour ensuite lister les bénéfices du cloud : - abstraction - disponible à la demande, - scalable, - paiement à l'utilisation Utiliser une plate-forme de Cloud Computing, c'est marcher dans des pas de géant tout en ne payant que ce que l'on utilise finalement. La scalabilité pour Amazon comme il l'explique, c'est tout d'abord un métier, dont ils sont spécialistes. Ensuite c'est un effort d'architecture de notre côté, car il ne suffit pas d'ajouter des machines sans réfléchir son SI dans le Cloud. Les points qu'il nous propose de regarder : - Design for failure - Loose coupling sets you free - Design for dynamism - Security is everywhere - Don't fear constraints - Many storage options - AWS ecosystem and community Tout d'abord comme le dit Werner Vogels le CTO d'Amazon : "Everything fails, all the time". Comme je dis à mes clients : "On est sûr d'une chose et on ne connaît pas l'autre. On sait que cela va planter mais on ne sait pas quand". Prendre en compte donc que l'application va planter. Amazon propose des solutions comme Elastic IP. Il prend l'exemple de la muraille de Chine, construite non pas pour empêcher les envahisseurs d'entre en Chine, mais pour les ralentir lorsqu'ils repartent, afin de les écraser lorsqu'ils étaient chargés de butin. La muraille a donc été dessinée dès le départ pour... ne pas marcher et laisser entrer les envahisseurs. Le deuxième point "Loosely coupled system" nous dit qu'il faut massivement découper notre application en boîte noire dans le Cloud afin d'en tirer parti. Pour cela on peut introduire de l'asynchrone avec Amazon Messaging Service, utiliser des Web Services comme Amazon, bref prévoir de faire des petits châteaux plutôt qu'un gros château. Le troisième point "Design for Dynamism" dans le Cloud est de dire que notre architecture d'application ne doit pas se reposer sur le serveur physique sur lequel elle tourne, elle doit être dynamique, elle doit être démarrable facilement avec un script de bootstrap sur Amazon, afin de pouvoir monter en charge. Concernant ensuite "Security is everywhere", tout d'abord Simone explique qu'Amazon nous offre la sécurité physique en répliquant nos données, en montant des datacenters certifiés (sismique/cambriolage/électricité...). La sécurité applicative est bien entendu à prendre en compte mais il n'y a pas de risques avec nos voisins sur Amazon. Le point suivant

"Don't fear constraints" nous fait comprendre qu'il faut s'adapter à la boîte à outil d'Amazon et ne pas avoir peur des contraintes. On vous demande plus de mémoire ? essayez le cache distribué (memcached). Vous voulez de meilleures performances avec votre base ? Regardez SimpleDB. Amazon a sa propre infrastructure sur le Cloud, au nom de quoi votre application ne pourrait-elle pas suivre les mêmes bonnes pratiques ? Concernant ensuite "Many Storage Options" ce slide présente simplement les différentes technologies d'Amazon : - Amazon S3 pour le stockage des données statiques - Cloudfront pour la distribution - SimpleDB pour une base simple et indexée - Amazon EC2 avec l'accès à un disque local physique pour stocker - Amazon EBS pour le stockage permanent Dernier point enfin "AWS Community and Ecosystem" avec l'exemple de la migration d'une application vers la plate-forme de Cloud Computing d'Amazon. Sa présentation est [en ligne](#) si vous souhaitez la regarder. Présentation intéressante, qui permet de prendre conscience de l'offre d'Amazon sur le sujet.

## USI 2009 Neal Ford de ThoughtWorks

La claque de la journée. Neal est à mon avis l'un des meilleurs speakers de cette première journée. Il est passé la semaine dernière à Zurich à [Jazoon](#), et j'avais beaucoup apprécié la fin de sa présentation, regrettant de ne pas l'avoir vue en entier. C'est donc avec plaisir que j'ai découvert qu'OCTO l'avait invité à l'USI cette année. Et bien je vais vous dire : ce gars est une bête. 161 slides en 55 minutes ! Je suis allé discuter avec lui et Guillaume Laforge rapidement après la présentation. Tout simplement un performer, qui a déjà fait plus de 100 présentations, une bête de scène. Le fond, la forme, tout était parfait. Attachez votre ceinture, on y va. La présentation s'intitule "*Les philosophes d'antan et la foire aux embrouilles*". Neal arrive sur scène. Avec son petit ventre, il passe derrière le micro et commence "*Neal Ford, ThoughtWorks, Américain comme mon ventre vous le montre (rires dans la salle), bonjour à tous*". Les lumières se baissent, le show commence. Tout d'abord un petit retour en arrière : "Those who cannot learn from history are doomed to repeat-it". Il débute son sujet par une introduction en nous expliquant comment les philosophes avaient déjà résolu certains de nos problèmes. L'un de nos gros défauts est de ne pas regarder le passé. Il montre ainsi ce livre de Kent Beck "[SmallTalk best practice](#)" de 1996 où avant l'arrivée de Java, des chapitres entiers louaient SmallTalk en pensant que ce serait le langage des années 2000. Avons-nous appris quelque chose du passé ? Comment devons-nous nous comporter et nous adapter afin de ne pas répéter les mêmes erreurs ? Platon avait déjà inventé le concept de Class et d'Instance. Nous mêmes dans notre vie quotidienne, nous voyons ce concept chaque jour sans nous en rendre compte. Prenez cette chaîne de restauration américaine. Lorsque vous voyez la tête de ce super hamburger sur l'affiche, une larme vous coule presque tellement cette photo est belle. Ensuite, une fois servi, la chose molle et fumante qui sort de cette petite boîte en carton, c'est pourtant bien ce que vous avez demandé non ? La Class est la photo, l'Instance est ce que vous obtenez. Il y a une différence entre ce que l'on spécifie et ce que l'on obtient. Platon 1 - la Salle 0 D'ailleurs Platon aurait détesté le JavaScript. Pour lui tout doit être objet, peut-être même à outrance. Mais son problème était qu'il lui manquait la logique. Aristote ensuite pose les bases de la logique moderne. Il définit 2 notions que nous pouvons appliquer chaque jour : «the essential property» et «the accidental property». Chaque objet a des propriétés fondamentales. Par extension il a des propriétés accidentelles, supplémentaires. Des effets de bords. On reviendra sur ce point dans quelques minutes. Galilée aurait été un excellent Architecture SOA. Il ne croyait personne, il souhaitait vérifier tout lui-même. Pour chaque cas, il mettait en place une expérience (un test) afin de valider son postulat. Neal explique ainsi qu'il aimait bien balancer toutes sortes de choses de la tour de Pise, quoique je ne pense pas que les 2 soient de la même époque. Mais bon, l'image est là pour nous expliquer que l'expérimentation est donc importante. Il explique ainsi que la contre-intuition conduit à la vérité. Nous devons donc apprendre de l'histoire que l'homme s'est souvent trompé. La Terre était plate avant que Galilée ne démontre le contraire, et nous devons donc ne pas répéter les mêmes erreurs que le passé. Neal présente ensuite le site [c2.com](#) qui liste des anti-patterns appliqués dans notre industrie. Il nous supplie de prendre le temps de lire ces anti-patterns qui démontrent que la croyance commune pour certains problèmes informatique a tort. Il passe ensuite en revue quelques patterns comme «Dead End» ou le cul-de-sac. Ce pattern s'applique à des projets qui vont se planter, où tout le monde presque le sait, mais où l'on continue en pensant qu'une hypothétique solution est possible. Aïe cela fait mal. Le pattern ensuite de la «Dart cible» : votre chef vient vous voir et il vous demande «Donne moi une estimation maintenant». Alors vous répondez avec une estimation en jours, en expliquant que celle-ci n'est pas certaine. Oui pas de problèmes dit votre chef, je sais que ce n'est qu'une estimation. L'anti-pattern s'active lorsque votre chef amnésique oublie que ce n'était QU'UNE ESTIMATION... Il aborde ensuite les projets «Mushroom» ou «Champignon de Paris». Il s'agit de projet où l'équipe de développement est coupé de la lumière et crache son produit sans jamais rencontrer un client final... Levez la main ceux qui sont concernés ? L'anti-pattern «[Standing of the Shoulders of Midgets](#)» est assez connue et dramatique. Voici comment je le résume «Vous devez utiliser Oracle Portayeu parce que la politique des achats c'est Oracle». C'est donc le pattern qui consiste à forcer le choix d'un produit avant même d'avoir parlé du besoin du client. Hop ça c'est offert. L'anti-pattern «[Net Negative Producing Programmer](#)» est le fantôme qu'en augmentant le nombre de développeur, la productivité va augmenter. Et bien pour certaines équipes, Neal explique qu'en virant 2 ou 3 personnes, parfois la productivité augmente de 30%. Radical mais pas faux. L'anti-pattern «[Boat Anchor](#)» est assez marrant. C'est ce bon vieux module que vous vous traînez, cette compatibilité avec le client qui n'a toujours pas mis à jour son produit, et qui comme une ancre de bateau, vous freine énormément. L'anti-pattern que je préfère et qui a bien fait rire la salle «[The Death by Planning Anti-Pattern](#)». C'est ce pauvre chef de projet qui passe sa vie devant Microsoft Project, et qui termine par une dépression car son outil ne gère pas l'imprévu, et qu'il passe son temps à remettre son planning à jour. Planifier c'est prévoir, pas prédire (ça c'est de moi, Nicolas Martignole) Il passe ensuite quelques autres patterns, j'ai noté «les Poulets élevés en batterie», «L'Homme des Cavernes Congelé» et un petit dernier pour rigoler «The Rubick's Cube». L'anti-pattern du Rubick's cube consiste à inventer des problèmes qui n'existent pas encore afin ensuite d'en proposer une solution, forcément la meilleure. Essayez, cela marche avec OSGi et SOA par exemple. Alors que faire ? Il continue en prenant l'exemple d'un client qu'il a rencontré. Les

départements de ce client échangent des messages dans des enveloppes avec des contrats papiers. Le problème est que les enveloppes coûtent beaucoup d'argent, et qu'il y a des mélanges des messages entre les départements. Alors arrive un gros cabinet, que l'on appellera «Accentué» et qui propose une solution SOA. Celle-ci vise à dématérialiser l'échange des documents et à tout faire au format électronique. 1 millions d'euros, 2 ans de projet prévu avant que la solution ne soit prête. La DSI voyant cela s'inquiète et demande conseil à Neal. Il propose après 3 jours d'audit de remplacer les enveloppes grises par des enveloppes de couleur pour chaque département.... Et le client était content, la solution de dématérialisation ne répondait pas à son besoin, qui était de réduire les erreurs de tris. Pensez-y la prochaine fois. Cela lui permet de revenir à ce qu'il appelle «l'accidental complexity». Celle-ci peut littéralement tuer une compagnie ou un département si nous ne sommes pas plus courageux et intelligents. Les responsables présents dans la salle écoutent poliment, le BlackBerry dans la main. J'espère que le message sera passé. Il parle ensuite du management qui a évolué, ce qui recoupe la présentation d'Alain Buzzacaro dont je vous parlerai un peu plus tard. Les nouveaux managers sont souvent des anciens Geeks, des anciens ITs. Et personne n'ose dire qu'ils ne sont pas bons. Cela provoque des problèmes humains, on a inventé l'Agilité pour que cela fonctionne mieux, mais il y a bien un problème derrière tout cela : de bons geeks ne font pas forcément de bons managers. Quelques slides plus loin, il explique le fonctionnement des éditeurs de logiciels. On pense à Oracle et IBM. Aujourd'hui vous, les gens qui font du Java, vous êtes comme les explorateurs et la ruée vers l'Or au 19ème siècle aux USA. Et les éditeurs ont bien compris ce qu'il se passait, ils sont les Levi Strauss des temps modernes. Je m'explique. Levi Strauss s'est installé à San Francisco. Il a vite compris que plutôt que de creuser et de se salir les mains pour trouver un hypothétique nugget, il est plus intéressant de fournir les meilleurs pantalons aux chercheurs d'or, et donc de se faire beaucoup d'argent. Les éditeurs ont donc tout intérêt à ce que vous soyez les mains dans le camboui pendant qu'ils lustrent la carrosserie de leur Mercedes... «With great complexity, came great power». Un système simple et transparent ne donne pas assez de pouvoir. Un bon gros système bien opaque et compliqué, donne beaucoup de pouvoir. Si vous ne voulez pas vous faire enfermer par un vendeur, prenez un système simple et ouvert. La complexité accidentelle ou volontaire génère donc de l'argent. Mais moi, en tant que développeur, comment puis-je changer le monde et proposer autre chose ? Il nous parle alors comme à des enfants : pensez-vous que le business va tolérer les caprices de la DSI encore très longtemps ? Vous ne pensez pas qu'un jour, la récréation sera terminée et que vous devrez peut-être ne plus faire ce qu'il vous plaît ? Prenez Toyota aux USA. Leurs voitures étaient moins chères et de meilleure qualité. Ils ont dépassé Ford ou GM. La qualité était meilleure, et maintenant le prix est même plus élevé que les constructeurs américains, car les consommateurs ont conscience d'acheter de la qualité... Il parle ensuite de la Chine et de l'Inde, ce qu'il appelle Chindia. Quelques chiffres font peur : 1% des Indiens c'est déjà 11,3 millions de personnes... 0,1% des indiens avec un niveau Bac+5, c'est 1 millions de personnes, soit 2 fois la population des informaticiens en France estimée à 500 000 selon le Sintec. Alors que l'on parle moins d'Inde et de Chine pour nous ne signifie pas que nous sommes en sécurité. Il y aura un changement, on ne sait pas quand il arrivera, mais il sera là demain. Il parle ensuite du Manifeste Agile, en expliquant que les personnes qui l'ont rédigé ont raison. Nous devons apprendre à changer rapidement, à revoir notre méthode de travail et à virer ce cycle en cascade qui doit mourir. Pour conclure il nous laisse quelques conseils pour la suite de notre carrière : Comprendre rapidement les lois de l'entreprise où vous travaillez, retirer les obstacles qui vous causent des soucis, arrêter les managers qui vous font faire n'importe quoi. Soyez responsable. Pensez à vous-même, soyez critique comme Galilée. Ne pensez pas que cette solution aujourd'hui sera toujours pérenne dans quelques années. Pensez à vous-même en investissant du temps pour apprendre, ne croyez pas que vous savez, vous ne savez rien. Lisez, discutez, rencontrez, faite de l'amélioration continue. Sinon vous êtes mort. Faites des choses qui... marchent. Lisez le manifeste Agile et regardez comment fonctionne une équipe Agile. Soyez critique et adaptez les idées à votre entreprise. Il termine en disant que le prix d'une chose est différent de la valeur. Ne pas facturer et venir se faire aérer les neurones à l'USI a un prix, je retiens surtout que la valeur de ce que j'ai entendu est bien plus importante, et je ne regrette pas cette matinée.

## USI 2009 Le Management OneOnOne

Alain Buzzacaro d'OCTO Technology présente la boîte à outils du Manager IT. Il débute la présentation par un constat : une majorité des managers sont d'anciens Geeks. Or de bons geeks ne font pas forcément de bons Boss. L'IT tout d'abord a mauvaise réputation. Nous livrons tardivement, nous coûtons cher. Le climat est à la méfiance. Quelles sont les origines de ce constat ? Les managers sont d'anciens Geeks, le contexte et l'environnement les soumettent à une forte pression. Devenir manager n'est pas dans le patrimoine génétique du Geek, il est souvent démunis et ses premiers moments sont délicats. Alain revient sur sa propre expérience. Il a mis en place une technique simple et pragmatique pour créer de la valeur, améliorer son management et travailler avec plaisir avec ses équipiers. L'activité du management se décompose en 5 activités : - La Gestion (valider vacances, note de frais) - L'Opérationnel (comité de pilotage, réunion) - La Promotion de l'équipe (pub interne) - Le Networking (recevoir prospect, contacter client) - Le Management effectif (créer le contexte de l'intelligence collective) Or souvent les managers se retrouvent englués dans de la Gestion ou par exemple l'arbitrage de Budget, ce qui ne lui laisse pas beaucoup de temps pour réellement manager son équipe. Il est souhaitable de réduire bien entendu ces obstacles comme explique Alain. Mais il nous explique ensuite comment être un meilleur manager avec une boîte à outils simple. Alain a mis en place la réunion "**OneOnOne**". Le principe est limpide : chaque semaine, il passe 30 minutes en tête à tête avec chacun de ses collaborateurs. Il voit donc physiquement durant un entretien informel et sécurisé chacun de ses collaborateurs. Pour cela la réunion se déroule durant 30 minutes, avec 3 séquences : - LUI parle pendant 10 mn et moi le Manager j'ECOUTE - MOI je parle par rapport à ce qu'IL m'a dit et j'ECRIS - Nous terminons par le FUTUR et les actions pour cette semaine Il est important d'instaurer un rituel, de voir chacun de ses collaborateurs à heure fixe, afin de ne pas se laisser déborder, et que cela fonctionne. Il faut créer un climat de confiance. Alain explique qu'il est actif, il s'assoit à côté de la personne et ils notent ensemble la discussion afin d'assurer un suivi de qualité. Il faut 3 composants

pour que cela fonctionne : - le Feedback - le Coaching - la Delegation Le Feedback : tout le monde en veut. Que ce soit votre fils de 5 ans qui vous demande "Regarde Papa/Maman" car il a besoin d'un feedback, ou votre collaborateur qui souhaite aussi entendre parfois un "Merci" ou "Tout va bien" plutôt qu'attendre l'entretien annuel pour découvrir ce qui cloche. Il explique qu'il est important de ne pas hésiter à utiliser des superlatifs. Lorsque quelque chose s'est mal passé, contextualisez et décrivez l'impact que l'action a eu sur vous, sur d'autres membres de l'équipe. La personne demande du feedback. Le Coaching : aider le collaborateur à atteindre ses objectifs. Comme le rôle du Scrum Master, s'assurer qu'aucuns obstacles ne bloquent votre collaborateur. Son deuxième écran ne marche pas ? Allez chercher le câble d'alimentation manquant par exemple. Etre coach c'est aussi supporter. Attention cependant en tant que manager à ne pas devenir une ressource, vous devez vous effacer pour laisser votre collaborateur travailler. La Délégation enfin, est le moyen d'aider votre collaborateur, s'il est motivé et compétent. Utilisez d'autres formes de management selon le profil de votre collaborateur. En conclusion, car je ne peux pas reprendre ici toute la présentation, Alain Buzzacaro explique très clairement les bénéfices, et le fonctionnement de ce moyen simple, pour recréer de la valeur. Il nous engage à tester l'initiative, ce qui motivera votre équipe. La communication améliore la productivité et la performance, un bon manager crée de l'intelligence collective.

## La vérité vraie sur mon activité de blogueur

A l'USI comme à Jazon j'ai eu l'occasion de rencontrer de nouvelles personnes, qui lisent le blog et qui étaient contentes de faire la connaissance avec "Le Touilleur" comme ils disent. Je me rends compte que mon blog me dépasse un peu et qu'il me donne une image qui m'embête un peu. Je tiens à remettre un peu les pendules à l'heure. Morceaux choisis : - " Ah c'est toi le Touilleur !" - " Oui c'est moi... le Touilleur c'est le blog. Moi je m'appelle Nicolas" -" Ah c'est super. Dis-donc je sais pas comment tu fais pour faire autant d'article" Alors à cet instant je regarde ma boîte à réponse pour sélectionner une réponse: a) je prends des drogues b) en fait on est plusieurs contributeurs c) je n'ai pas de métier, mon métier c'est bloguer. d) je suis forcément célibataire donc avec plein de temps pour bloguer. Une fille ça aime pas les Blogs. e) je suis un extraterrestre envoyé sur Terre pour apprendre la technologie des humains **Et bien aucune de ces réponses n'est correcte.** Voici la vérité 1) Je m'investis vraiment dans le blog depuis 1 an 1/2, même si cela fait plus de 5 ans que je blogue 2) En tant qu'indépendant je peux prendre ce temps et le financer pour participer à des événements comme Jazon ou l'USI 3) Il est clair que je suis passionné par ce que je fais et que je m'amuse 4) Je n'en tire aucune gloire, aucuns revenus 5) Je ne dis pas que je ferai cela encore très longtemps à ce niveau 6) J'ai une femme formidable et 2 petits en bas âge, qui me réveillent le dimanche matin tôt, ce qui me permet de bloguer (ce sont les petits qui me réveillent, je précise...) 7) Je n'aime pas le foot, les émissions où une fille veut se marier avec un fermier, les émissions où les jeunes chantent comme des casseroles 8) J'aime bien écrire 9) Je prends le RER, ce qui me laisse du temps pour réfléchir à ce que je vais bloguer 10) Je suis plutôt quelqu'un qui aime bien rencontrer du monde, ce qui fait de nouvelles idées d'articles, et voilà Le Touilleur Express c'est le blog, ce n'est pas moi. C'est 387 articles, et je ne suis pas peu fier de vous annoncer que le blog a dépassé les 10 000 visites par mois dimanche dernier. Forcément cela fait du monde, un peu de pression, mais tant qu'on se marre, on continue non ? Allez je vous laisse contempler cette magnifique courbe, on se croit presque sur TF1. Blague à part, merci à toi de me



lire.

## USI 2009 le Bilan

L'USI 2009 se termine. 2 jours très riches, et surtout très enrichissant. En revenant ce soir j'ai l'impression d'avoir reçu des injections d'intelligence, de vision et de bon sens... Heureusement qu'il y a un événement de ce niveau et de cette qualité en France, nous aurions tôt fait sinon de nous regarder le nombril, pendant que nos voisins font là où nous, nous pensons. Cette deuxième journée commence par une Keynote très humaniste par un généticien très connu, Albert Jacquard. Je reviendrai sur sa présentation dans un article, car le personnage et le discours a eu le mérite de nous cadrer la journée qui allait suivre, en s'axant sur la valeur humaine. J'ai débuté ma journée par SpringSource et la présentation de Julien Dubois, qui m'a donné une phrase qui me parle : la séparation physique entre la couche Web et la couche métier a été dictée par des vendeurs... de hardwares. Trop content de pouvoir vous vendre plus de serveur, ceux-ci ont oublié de vous dire que dans la vraie vie... on peut s'en passer. J'ai une pensée émue pour mon client actuel qui est massivement Web d'un côté, EjbTiers de l'autre. C'est dingue de se dire que l'on doit faire vivre

aujourd'hui ce type d'architecture, et que le WebTiers n'a pas accès à la base de données. Quel gachis. Je continue ma journée avec la présentation "L'informatique Conviviale" par Pierre Piezzardi et Laurent Avignon d'OCTO Technology. Cette session a été épanouissante, alors que je ne savais pas vraiment quoi venir chercher en arrivant. La gestion du changement, la rupture et la responsabilisation, très intéressant et je bloguerai dessus plus longuement. Je me suis ensuite accordé un break, j'ai zappé Philip Evans. J'en ai profité pour aller voir aussi l'espace exposant, avec une démonstration de la table tactile Microsoft Surface, une démonstration d'un écran qui enregistre où votre regard se pose, et une table musicale. Je vous posterai une vidéo où vous verrez un Nicolas devant l'écran qui se fait enregistrer alors qu'il surfe... marrant. Le déjeuner avalé, j'enchaîne avec une présentation pétillante, qui fait du bien : Amélie Boucher m'a expliqué l'ergonomie en 50 minutes, et mon cerveau de geek primaire a tout compris. Je vous proposerai un article complet, j'ai franchement bien aimé le ton léger et le sujet sérieux, c'était très sympa. J'ai hésité entre la présentation de Fabrice Robini et celle de Guillaume Laforge sur les DSLs et j'ai craqué pour Groovy. Lors de la présentation j'ai appris encore de nouvelles choses sur Groovy, note pour plus tard cher lecteur que je te posterai un petit article aussi. Ensuite j'avais prévu d'aller voir Guillaume pour comprendre "Vaincre le Plouf". Seulement en face dans une salle s'est organisé une présentation de Google Wave. Désolé les gars, mais impossible pour moi de manquer cette présentation organisée au pied levé par Guillaume Laforge et Didier Girard. Et bien je peux te dire que je ne regrette pas ces 50 minutes. Ok la démonstration a pas mal planté. Mais au delà de la démonstration, pour moi c'était un choc. J'ai eu conscience de voir quelque chose qui va devenir dans les mois qui viennent un GROS truc, et qu'en même temps je ne savais pas ce que les gens allaient en faire. La salle de 50 places était blindée, j'avais à ma gauche un journaliste du Monde Informatique, bref ce moment était magique, c'est ça aussi l'USI. Ensuite j'ai fait un break et je dois vous dire que j'ai zappé la restitution du DSI. Désolé. Un article publié plus tard, la journée s'est terminée par une note de musique, et quel moment... [Eric Lewis](#) est un pianiste de jazz et de rock, qui nous a offert une heure de concert. C'est tout d'abord une bête de scène. Il a joué à TED, et j'ai hâte que l'équipe de l'USI 2009 vous fasse partager cette vidéo. Il a joué/rejoué des standards du rock avec le piano. Cela donne du Nirvana, du Ray Charles, du Coldplay... Enorme A la fin de son show il demande si nous souhaitons qu'il joue quelque chose. Comme personne ne se lance, **je crie "Michael Jackson"**. Et bien il a envoyé Billy Jean sans vraiment savoir au début, et toute la salle a terminé debout. 3 standing ovation, 2 rappels, juste énorme. Il a grandi dans une maison de musique. Sa mère et sa grand-mère étaient des musiciens. Il a grandi entouré de piano, et ce qui l'intéresse c'est de proposer une autre approche du Piano, de nous jouer de l'ACDC ou de l'IronMaiden, et d'arriver à sortir de nouveaux sons jamais entendu avant. C'était une grosse performance physique, et il a fait passer une quantité d'énergie, c'était juste énorme. Les quelques 400 personnes dans la salle se sont senties habitées tout comme lui. Il a fait par exemple une reprise de "Smells like teen spirit" de Nirvana, bref c'était juste énorme. La vidéo ci-dessous n'est pas de l'USI 2009 mais elle vous donne une idée du personnage :

#### Conclusion L'USI se termine.

Organisation énorme, aucuns couacs, juste parfait. Le contenu ? et bien après toutes les conférences auxquelles je me suis baladé, j'ai l'impression d'avoir découvert l'Amérique. Le format, le ton, l'organisation, cet ensemble entre Geeks et Boss fonctionne très bien. Cet après-midi j'ai bien aimé cette présentation sur l'ergonomie, hier j'ai beaucoup aimé cette journée, l'organisation, le contenu, la qualité des speakers... L'USI a certes un prix, garant de l'indépendance et d'une bonne organisation, mais je retiens finalement plutôt la Valeur de ce que j'ai vu pendant ces 2 jours. Les Keynotes bien entendu m'ont scotché. J'ai un peu décroché sur la Keynote de Daniel Cohen. Son discours était sur une autre strate, et je n'ai pas pris plaisir à l'écouter. J'ai adoré Joel de Rosnay, j'étais ému en écoutant Albert Jacquard, j'ai jubilé en écoutant Neil Ford... Juste énorme. Côté technique, par rapport à un Jazoon je n'ai pas appris la toute dernière nouveauté dans JEE6. Mais à la limite on s'en fiche. J'ai plutôt eu l'occasion d'aérer mes neurones de vieux geek, d'écouter des gens câblés dans l'autre sens. La qualité du contenu, l'organisation, les gens présents avec lesquels j'ai discuté, bref deux jours très riches et très intenses. Je termine ces 2 jours avec une tonne d'articles à écrire. Merci à toutes les personnes d'OCTO qui nous ont offert 2 jours vraiment top, merci aux speakers, et je passe le bonjour à toutes les personnes que j'ai croisé pendant ces 2 jours. A suivre d'ici quelques jours / semaines -> un paquet d'article sur l'USI 2009. A l'année prochaine !

## Jazoon 2009 : Hibernate Envers

Adam Warski de la société *Level N Consulting* a présenté Hibernate Envers lors d'une présentation courte de 20 minutes à Jazoon. Petit retour sur Envers, vous allez tout comprendre. Hibernate

Envers est un système simple qui permet d'ajouter de l'Audit et la gestion de Version à des Entités Hibernate. L'audit est l'enregistrement automatique en base de données des différentes versions d'une entité. Envers permet aussi de loguer les modifications faites à une Entité Hibernate dans une base de données, pour stocker l'heure et le nom de la personne qui a changé une Entité par exemple. L'utilisation est simple. Pour cela, il suffit d'ajouter l'annotation @Audited et de déclarer des listeners dans le fichier persistence.xml. Pour chaque entité auditée, une nouvelle table est créée qui va conserver l'historique des changements, un peu comme Subversion si vous voulez. Cela permet aussi de retrouver facilement l'historique des différentes versions d'une entité. Imaginez une Facture, ce système vous permet d'enregistrer les différentes versions de la Facture, sans aucuns efforts. Sur cette photo prise durant la présentation, on voit l'annotation et la déclaration dans la configuration Hibernate.



Que peut-on

auditer? Les types simples définis par JPA (String, Integer, Date), les composants et les relations entre les composants. Le support des Collections et des types spécifiques à Hibernate est aussi possible. Voyons avec un peu de code comment cela fonctionne. Si vous souhaitez auditer l'ensemble d'une Entité :

```
@Entity
@Audited // that's the important part :)
public class Person {
    @Id
    @GeneratedValue
    private int id;

    private String name;

    private String surname;

    @ManyToOne
    private Address address;

    // add getters, setters, constructors, equals and hashCode here
}
```

La déclaration s'effectue de la façon suivante si vous ne souhaitez auditez que certaines propriétés :

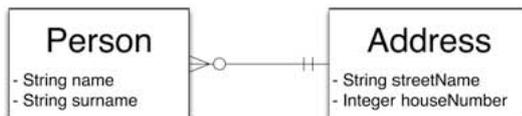
```
@Entity
public class Person {
    @Id
    @GeneratedValue
    private int id;

    @Audited
    private String name;

    @Audited
    @ManyToOne
    private Address address;

    // we don't care about this in Audit
    private String twitterName;
    private Date dateOfBirth;
}
```

Le développeur sélectionne les entités qu'il souhaite faire auditer. Pour chaque entité, une nouvelle entité auditée est créée dynamiquement. Par exemple Address aura une entité Address\_AUD pour l'Audit, et c'est bien cette deuxième entité qui est persisté en base... avec Hibernate. Pour chaque inser, update ou delete, une nouvelle révision est créée. Une Transaction est égale à une révision, si plusieurs entités sont embarquées dans une transaction. Le numéro de version est global si vous auditez plusieurs entités, comme le numéro SVN de votre code. Imaginez qu'une entité est un fichier Java que vous committez, c'est exactement pareil. Imaginons une entité Person et une entité



Address, toutes les 2 annotées:

Déclarons tout d'abord 2 adresses sur une personne John Doe :

```
// Revision 1
em.getTransaction().begin();
Address a1 = new Address("West st.", 10);
Address a2 = new Address("East st.", 15);

Person p = new Person("John", "Doe");
p.setAddress(a1);
entityManager.persist(a1);
entityManager.persist(a2);
entityManager.persist(p);
em.getTransaction().commit();
```

Si je modifie maintenant le nom de la personne et que j'utilise la deuxième adresse :

```
// Revision 2
em.getTransaction().begin();
p = entityManager.find(Person.class, id);
p.setName("Paul");
p.setAddress(a2); // declared in the other code part

em.getTransaction().commit();
```

Comme vous pouvez le constater, l'utilisation d'Envers est transparente pour le développeur, aucun bout de code supplémentaire n'est ajouté. Voyons maintenant comment retrouver nos données. Je recharge la révision 1 et je vérifie que le nom de la personne est bien John, puis ensuite comme on le voit, le chargement de l'arbre de relation vers Address fonctionne sans problèmes.

```
AuditReader ar = AuditReaderFactory.get(em);
```

```
// Reading the person at revision 1
old_p = ar.find(Person.class, id, 1);
assert "John".equals(old_p.getName());
assert a1.equals(old_p.getAddress());

// Reading the addresses at revision 1
old_a1 = ar.find(Address.class, a1_id, 1);
assert old_a1.getPersons().size() == 1;
assert old_a1.getPersons().contains(p);
old_a2 = ar.find(Address.class, a2_id, 1);
assert old_a2.getPersons().size() == 0;
```

La recherche est aussi très puissante et les systèmes de requêtes d'Hibernate permettent de déclarer par exemple:

```
auditReader.createQuery()
    .forRevisionsOfEntity(Person.class, false, true)
    .addProjection(AuditEntity.revisionNumber.count())
    .add(AuditEntity.id().eq(person.getId()))
    .getSingleResult()
```

Il est important de se souvenir que l'audit est effectué par une autre Entité Hibernate, et que la

Person:

Id	Name	Surname
123	John	Smith
857	Brad	Pitt
698	Mary	Doe
...		

Person\_AUD:

Rev number	Id	Name	Surname	Rev type
64	123	James	Smith	ADD
64	857	Brad	Pitt	ADD
85	123	Peter	Smith	MOD
90	501	Arnold	Schwarzenegger	DEL
90	123	John	Smith	MOD

table originale n'est jamais affectée.

**Comment écouter les changements sur une entité ?** Envers permet aussi de déclarer un Listener qui sera notifié lorsque l'entité est persistée vers la base de données. Je reprends le code directement des slides de la présentation, ne croyez pas que j'ai réussi à refaire cet article sans avoir les slides de la présentation ;-) Voyons tout d'abord une Entité de Révision. Elle est annotée avec @RevisionEntity, il faut aussi annoter un timestamp avec @RevisionTimeStamp et un numéro de révision, avec @RevisionNumber comme sur cet exemple:

```
@Entity
@RevisionEntity(ExampleListener.class)
public class ExampleRevEnt {
    @Id @GeneratedValue @RevisionNumber private Long id;
    @RevisionTimeStamp private Long timestamp;

    @ManyToOne private User modifiedBy;

    // Getters, setters, equals, hashCode ...
}
```

Vous pouvez ensuite déclarer un RevisionListener qui sera activé lorsque l'entité décorée est persistée, afin par exemple d'ajouter le nom de l'utilisateur qui a changé une Entité dans votre interface Web. Cela permet de créer une fonction de Log rapidement, sans alourdir le reste de votre application.

```
public class ExampleListener implements RevisionListener {
    public void newRevision(Object revEntity) {
        ExampleRevEnt revEnt = (ExampleRevEnt) revEntity;
        User currentUser = (User) Component.getInstance("currentUser");

        revEnt.setModifiedBy(currentUser);
    }
}
```

Pour plus de détails sur cette fonctionnalité : [voir la documentation d'Envers](#). **En conclusion** Envers 1.2.1 GA est sorti il y a quelques semaines, c'est un projet qui fonctionne avec Hibernate 3.3 et donc Java 5. Je retiendrai la facilité de mise en place, la clarté de l'API. Pour peu que votre application nécessite de l'audit, il serait dommage de s'en passer. Le site officiel : <http://www.jboss.org/envers/>. Crédit pour les extraits de code : Adam Warski de la société *Level N Consulting* ainsi que [la documentation d'Envers](#).

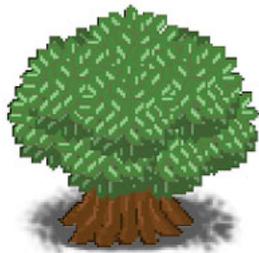
## Jazon 2009 : Gaming in JavaFX



Dernière présentation de cette deuxième journée, trop courte mais bien rafraîchissante : comment faire des jeux à la Zelda avec JavaFX ? Le speaker est bien connu de la communauté JavaFx : Jose Maria Silveira Neto, JUG Leader Brésilien du CEJUG. Allez voir [son blog](#) dans la section [game](#), c'est une mine d'or pour apprendre à écrire des jeux avec JavaFx. Si vous pensiez que j'étais une brute en écriture d'articles techniques, José Maria est une bête en écriture sur JavaFx et la création de jeu. Allez je vous dis regarder par exemple [cet article](#) mais revenez ensuite. Sa présentation de 20 minutes vise à nous montrer comment écrire rapidement des jeux classiques. Voici ce que cela donne en vidéo :

Je me dis après quelques slides qu'il ne reste plus qu'à offrir le support de JavaFx sur les téléphones portables. Sinon je ne vois pas l'intérêt de JavaFx pour l'instant, face à Adobe Flex ou même HTML 5 demain. Par contre, lorsque l'on sait qu'un grand nombre de téléphone embarquent une JVM, ou même Google Android, j'espère qu'il sera possible de jouer à ces petits jeux demain sur mobile. José Maria explique tout d'abord que le développement de jeu n'est pas facile. Le chargement des ressources, la gestion du réseau, des périphériques comme la manette de la Wii, tout ceci fait appel pour l'instant à des kits propriétaires de développement, ce qu'il regrette. JavaFx apporte une simplification de la gestion des ressources (images, vidéos, sons). Le support de l'accélération matériel est aussi un plus. Il cite le **Java Store** et il espère que demain, celui-ci sera le nouveau portail de Java pour trouver des applications sympathiques en JavaFx, et donc des jeux. Voici par exemple comment charger une image en JavaFx:

See <http://silveiraneto.net/> or <http://touilleur-express.fr>



```
var tree = ImageView {
    image: Image {
        url: "{__DIR__}tree.png"
    }
}
```

Vous pouvez aussi charger une URL

très facilement en JavaFx:

```
var bed = ImageView {
    image: Image {
        url: "http://example.com/bed.png"
    }
}
```

La suite de sa présentation nous montre comment animer des sprites, comment gérer la détection de collision, le chargement d'une carte (MapTiles) et le scrolling. Il cite le projet "[Project Darkstar](#)". Après enquête, il s'agit d'un projet de plate-forme ouverte de développement pour réaliser des jeux vidéos. C'est un moteur pour des jeux en ligne massivement multi-joueur, supporté par SUN Microsystems. Il permet de gérer finement les ressources en cluster, afin de répartir la charge des joueurs sur les serveurs. Il peut servir à réaliser un jeu social comme "2nd Life" par exemple. L'idée est de vous donner un moteur pour le chargement des zones de jeux, la gestion des joueurs, la sécurité et la gestion des problèmes de threads. Voici une vidéo dans laquelle Owen Kellet en septembre 2008 présente le projet.

### Conclusion En

conclusion c'est finalement sa présentation qui m'a le plus donné envie de regarder JavaFx pendant ces 3 jours. Le marché du casual games est énorme. Le nombre de téléphones mobiles aussi. La puissance et la simplicité de JavaFx me semblent plus adapté pour attaquer ce secteur, plutôt que le monde de la bonne vieille application de gestion. Rendez-vous mardi 7 juillet 2009 au Paris Java User Group pour une présentation sur Adobe Flex puis JavaFx.

## Jazon 2009 : Mockito



Voici enfin le dernier billet de ma visite à

[Jazon 2009](#). Je termine le troisième jour par une présentation de Mockito par l'excellent Szczepan Faber. Il bosse à Varsovie en Pologne, c'est un ancien de ThoughtWorks London, un ancien barman lorsqu'il était étudiant, et surtout ce genre de personne qui vous font passer une heure délirante en parlant d'un sujet sérieux. Allez soyons sérieux 30 secondes... Après une introduction, sa présentation débute par un slide avec la définition d'un Mock. Il pose la question : qu'est-ce qu'un mock ? Sa réponse : "**C'est un substitut d'un objet réel afin de faire des tests, en simulant les interactions**". Comme personne ne pipe mot, il appuie sur espace et une... poupée gonflable apparaît alors : "*vous voyez ? un mock permet de se passer de la vraie instance d'un objet, afin de pouvoir faire des tests tranquillement avec un substitut...*". La salle est morte de rires, je crois que

tout le monde se souviendra de la définition d'un Mock. Il continue ensuite, et puis au 4ème slide il démarre Eclipse, afin de nous montrer un peu comment marche Mockito. Hop quelques lignes, on note au passage sa maîtrise des raccourcis d'Eclipse, parfois j'ai même eu envie de passer à Eclipse et de laisser tomber IDEA IntelliJ. Mais vu que visiblement ce bonhomme n'est pas humain, ou qu'il y a un truc, je suis donc resté à IntelliJ. Le voilà qui code une classe, un mock. Mais il utilise... EasyMock. Est-ce que quelqu'un va lui dire qu'il s'est planté ? [Que nenni](#) ! C'est fait exprès.... Aaaah d'accord. Là je vais pas pouvoir continuer à bloguer cher lecteur, car le bonhomme en question a ensuite passé 32 minutes à coder, comme une bête. J'ai tout d'abord retenu une petite astuce toute simple à reprendre dans vos tests unitaires. Utilisez les expressions "given/when/then" pour exprimer vos tests. Cela donne par exemple :

```
@Test
public void shouldIncrementMyValue() throws Exception {
    // given

    // when

    // then
}
}
```

J'ai ensuite surtout vu la différence entre easyMock et Mockito. Respect à David Gageot qui nous l'a dit il y a 6 mois. Oui Mockito est mieux qu'easyMock. Maintenant il faut que je dise pourquoi. Ca c'est le plus dur, surtout sans notes. Durant sa présentation je n'ai pas eu le temps de prendre des notes. Ce bonhomme étant trop fort avec machin-eclipse, impossible de le suivre. Ensuite, sa présentation étant vraiment bien, et sans temps morts, j'ai pas pris de notes. Voilà désolé, hop là je sens un grand cri de frustration... allez je vais chercher des bouts de codes et je reviens. Mais je vous dis, sans notes, ça va être difficile. Regardons tout d'abord Mockito, le code ci-dessous n'est **pas du bon code** mais simplement quelques lignes afin de vous montrer le fonctionnement de Mockito.

```
import static org.mockito.Mockito.*;

import java.util.LinkedList;

/**
 * Simple Mockito test.
 * @author Nicolas Martignole
 */
public class Test {
    @SuppressWarnings("unchecked")
    public static void main(String[] args) {

        // Creation d'un mock
        LinkedList mockedList = mock(LinkedList.class);

        mockedList.add("Test Touilleur 1");
        mockedList.clear();

        // Faire autre chose
        // ....

        // Verifier que l'on a ajoute un element
        verify(mockedList).add("Test Touilleur 1");
        // Verifier que l'on a vide la liste
        verify(mockedList).clear();
    }
}
```

La différence avec EasyMock : pas de replay à effectuer. Le premier appel à un verify bascule le mock automatiquement. On voit aussi que la syntaxe est facilement lisible : "verifier que mockedList point clear a ete appelé" Mockito est aussi très intéressant lorsqu'un événement non attendu survient. Imaginons que je m'attende à ce que ma liste "mockée" soit manipulée, que l'on effectue un deuxième ajout. J'ajoute donc une étape de vérification qui va planter puisque je n'ai pas ajouté de deuxième élément à ma liste. Regardons comment le code réagit dans ce cas précis :

```
import static org.mockito.Mockito.*;

import java.util.LinkedList;

/**
 * Simple Mockito test.
 * @author Nicolas Martignole
 */
public class Test {
    @SuppressWarnings("unchecked")
    public static void main(String[] args) {

        // Creation d'un mock
        LinkedList mockedList = mock(LinkedList.class);

        mockedList.add("Test Touilleur 1");
        mockedList.clear();

        // Faire autre chose

        // Verifier que l'on a ajoute un element
        verify(mockedList).add("Test Touilleur 1");
        // Verifier que l'on a vide la liste
        verify(mockedList).clear();
    }
}
```

```

// Verifier que l'on a aussi ajouter un autre element
// ce qui n'est pas le cas, Mockito va donc
// nous le signaler
verify(mockedList).add("not added");
}
}

```

Voici ce que Mockito affiche lors de l'exécution :

```

Exception in thread "main" org.mockito.exceptions.verificaton.WantedButNotInvoked:
Wanted but not invoked:
linkedList.add("not added");
    at Test.main(Test.java:30)

```

En clair : vous avez demandé à vérifier que l'on ajoute la phrase "not added" mais ce n'est pas le cas. Mockito produit des exceptions très claires lorsqu'un cas non attendu survient. Il sépare clairement aussi la définition du stub de la partie vérification, très facilement. Je vous encourage à tester Mockito si vous utilisez easyMock, ou si les Mocks ne vous disent rien. Mockito permet de mocker des classes concrètes, certain préfèrent mettre des interfaces partout, et mocker ces interfaces. A la limite, faites ce qui vous arrange, la pratique des interfaces venant d'un livre de Martin Fowler qui disait, je crois, qu'il faut mocker des rôles, donc des interfaces. Quelques derniers mots sur les charmes de Mockito : - possibilité d'utiliser une vérification avec un ordre : inOrder() - atLeast(x), times(x), etc. - verifyNoMoreInteractions() - enchainement de commandes: thenReturn(x).thenThrow(y) - possibilité de matcher avec des arguments comme: anyObject(), etc. - utilise [hamcrest](#) comme jMock J'espère que nous aurons l'occasion de rencontrer Szczepan à Paris, au Paris JUG. Si vous le voyez à Devvix 2009, croyez-moi cela vaut le coup. **Références** - [Blog de Szczepan](#) - <http://mockito.org/> - [L'api de Mockito](#) Retrouvez l'ensemble des mes articles sur Jazoon 2009 [sur cette page](#)

## iPhone OS 3.1 pour bientôt, puis iPhone OS 4.0

Alors que votre "précieux" vient juste de recevoir sa petite mise à jour vers la version 3.0 du système d'exploitation, je peux vous annoncer qu'une version 3.1 est prévue pour très bientôt. Elle est déjà disponible pour les développeurs, afin de s'assurer de la compatibilité des applications. Au programme de cette nouvelle version ([voir l'article original en anglais](#)) - Meilleure édition des vidéos pour l'iPhone 3GS - l'iPhone vibre lorsque vous déplacez des icônes pour les ordonner. J'ai lu quelque part que l'on peut mettre 160 applications max sur un iphone. - un système anti-hameçonnage appelé "Fraud Protection" a été ajouté à Safari - l'iPhone démarre plus vite - Contrôle par la voix à travers le BlueTooth - API pour que les développeurs puissent accéder aux vidéos et les éditer. - Mise à jour du firmware du modem à la version 5.08.01 - Mise à jour de l'OpenGL et de Quartz. **iPhone 4.0 révélé en partie** Le site [unwired-view](#) a vu les dépôts de brevets d'Apple, et en a tiré un article avec quelques pistes pour la prochaine version de l'os de l'iphone. A la lecture de cet article, on se rend compte qu'Apple va prendre une sacrée avance sur Android et le reste du marché. Les points expliqués dans l'article : - Contrôle de l'iPhone selon l'environnement (géographique, température, environnement...) - Communications programmées et automatiques (envoi de SMS pour anniversaire) - Modes intelligents et avancés (pas simplement silencieux-normal, un truc plus intelligent) Le premier brevet protège l'idée suivante :

[...]n'importe quel événement basé sur la localisation (par exemple, le dispositif [l'iphone] entrant ou sortant d'un emplacement géographique spécifique, comme un pays ou un type de lieu, comme une salle de cinéma, etc), n'importe quel événement basé sur l'environnement (par exemple, le dispositif est soumis à une orientation physique, à un mouvement, à une température, du son, de la lumière, etc), tout événement de calendrier (par exemple, l'iphone atteint un moment de la journée, jour de la semaine, date, etc), tout événement basé sur l'utilisation particulière de l'appareil (par exemple, l'appareil est utilisé pour une fonction spécifique, pour une période de temps, la batterie de l'appareil a moins de la moitié de ses capacités restantes, etc), n'importe quel événement basé sur une nouvelle/un média (par exemple, l'appareil reçoit des informations sur un événement mondain, tel que la météo, les actualités rapport d'évaluation ou de sport, etc), et les combinaisons de ceux-ci.[...]

En clair : votre iPhone aura un système de stimulation, ce qui devrait le rendre capable de réagir à l'environnement. Cela ouvre bien entendu un monde à l'innovation. On imagine un iphone qui se met en mode silence au cinéma, qui n'essaye plus d'utiliser la 3G en permanence lorsque la batterie est presque plate, qui passe en mode bluetooth et GPS dans la voiture... Bref cela ouvre la porte à de nouvelles applications. Et là je vous parle d'un brevet.



### Communications programmés

Un deuxième brevet ([voir ici le brevet complet](#)) appelé "User-programmed automated communications" est un brevet qui protège l'idée que l'utilisateur peut configurer l'activation d'une fonction de communication de l'appareil selon différents stimuli comme une date, un endroit où se trouve l'appareil, un événement reçu, etc. L'appareil peut alors composer un numéro, envoyer un SMS, un

Fax, bref communiquer. L'idée protégée est donc la création de l'association d'un déclencheur et d'un actuateur comme l'envoi d'un message... Autant dire que cela ratisse large.



### Modes intelligents automatique [Le troisième](#)

[brevet](#), plus ancien, protège l'idée de ce que l'on appelle un "mode intelligent". Aujourd'hui votre smartphone est soit actif, soit en mode silencieux. L'idée de ce brevet est de permettre à l'appareil de s'adapter automatiquement selon l'environnement. Dans le brevet par exemple, on voit qu'il est prévu que l'appareil passe en mode "voicemail" et "mail" lorsqu'il détecte qu'il est dans un cinéma. Une autre idée serait de détecter que la personne est entrain de faire son jogging, comme avec les chaussures Nike+

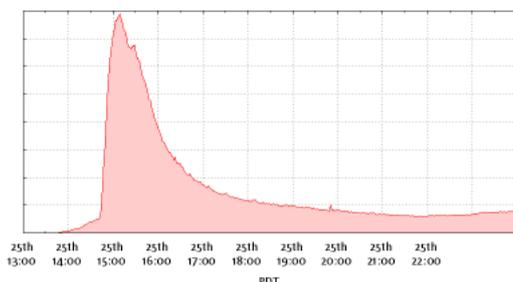
[0075]In some embodiments, if sensors (e.g., GPS and/or triangulation sensors or detectors) detect that a user is in a movie theater, the communications device can identify the preferred communication modes as voicemail and e-mail. In some embodiments, the communications device can connect to the Nike+ sensors installed on the user's shoes. After the communications device has connected to the Nike+ sensors, the communications device can identify that a user is jogging with the Nike+ workout feature. Based on this determination, the communications device can identify telephonic communications as the communication mode.

Ces 3 brevets ne sont que des idées protégées, et rien ne nous dit qu'elles seront implémentées en effet dans la prochaine version de l'iPhone. Mais je trouve intéressant l'idée d'avoir un appareil plus adapté à notre environnement, capable de prendre des décisions automatiquement. J'imagine mon iPhone basculer en Wifi tout seul et éteindre la 3G lorsque le GPS détecte qu'il est arrivé à la maison par exemple...

## USI 2009 Quelques idées issues des "grands" du web pour remettre en cause vos réflexes d'architectes

Suite de l'USI, [Olivier Mallassi](#) et [André Nedelcoux](#) d'OCTO Technology présentent les architectures des grands du Web comme Google, eBay et Amazon, afin de remettre en cause nos réflexes d'architecte. J'ai beaucoup aimé. La session débute avec quelques chiffres qui ont le mérite de rendre humble : Google c'est 39 centres de données dans le monde, 2.5 % de la consommation électrique aux USA. C'est 400 serveurs qui tombent en panne par jour. En cherchant quelques chiffres pour renforcer mon article, j'ai lu que YouTube coûte [2 millions de dollar par jour à Google](#) avec 75 milliard de vidéo pour 375 millions de visiteurs uniques en 2009. Le coût de la bande passante et du stockage est d'environ 2\$ par visiteur. La bande passante coûterait 1 million par jour, les datacenters seulement 36 000 dollar par jour selon Credit Suisse... Revenons à notre présentation. eBay c'est 250 millions d'objet en vente, 44 milliards de requêtes Web par jour, 6 millions de ligne de code. Forcément on se sent petit. Un slide suivante montre l'effet "Michael Jackson" sur le trafic de la première page de Google. D'ailleurs Google a cru un moment à une attaque. Le 25 juin à 15h00 c'est un tsunami qui s'est abattu sur le moteur de recherche.

Michael Jackson queries on 2009-06-25



Je me dis à cet instant, que Reuters

devrait surveiller les recherches Google pour détecter l'actualité avant qu'elle ne devienne de l'actualité. Tout d'abord ces grands sites ont quelques caractéristiques auxquelles nous devrions penser : leur croissance est très rapide et continue de s'accélérer. Amazon qui était un site Internet, a aujourd'hui plus de trafic sur sa plateforme Amazon de Cloud-Computing que sur la librairie. Olivier explique qu'ils sont passés de simple site internet à plateforme. Google de même avec Google App Engine. eBay n'a pas hésité au cours du temps à remettre son architecture en question plusieurs fois. Nous sommes passés du Perl au C++ puis au Java, plus facilement maintenable et **tout aussi performant**. Les bases de données sont structurées différemment des bases que nous voyons au quotidien dans nos applications de gestion. Ces bases sont en général faiblement relationnelles. La validation du modèle se fait maintenant dans le monde objet plutôt qu'au niveau de la base elle-même. Il n'y a pas de contraintes d'intégrité dans la base de données. Les données sont stockées sur différents serveurs tout d'abord verticalement. Pour faire simple : un serveur pour l'ensemble des CD, un serveur pour les DVD et un serveur pour les livres. Un partitionnement

horizontal des données accélère encore le stockage : les Livres dont le titre débute par "A" sont sur le serveur 1, les Livres "B" sur le serveur 2, etc. C'est donc une segmentation physique intelligente qui permet de monter en charge. La réplication des données est vitale, et elle fait partie intégrante des architectures de ces grands comme Amazon ou eBay. Différentes stratégies comme l'utilisation de Merkle Tree (autre nom pour Hash Tree) permettent de s'assurer de la validité des données. Pour maintenir l'information sur les serveurs en panne, des systèmes basés sur ce que l'on appelle "[Gossip protocol](#)" (gossip=ragô) permettent au réseau de s'auto-informer sur les machines en panne. Les architectures des grands du Web sont donc performantes, distribuées, résistantes à la panne et basées sur des plate-formes de Cloud Computing. Olivier rappelle qu'Amazon et eBay ont 6h d'indisponibilité par an... Cela force le respect. Leur synthèse nous propose ensuite de prendre un peu de recul. Ne pas partitionner de prime abord, cela vous crée des problèmes pour en résoudre d'autres que vous n'avez pas encore. Le mode relationnel n'est pas l'unique solution, pensez à diviser verticalement vos traitements dans votre application. La question de l'utilité de la base classique se pose, lorsque l'on voit que les grands du Web ne s'en servent pratiquement pas... La gestion des Transactions est complexe, prévoir des services de nettoyage qui vérifient l'intégrité des données à posteriori pour gérer les rares cas des transactions techniques abandonnées. La gestion des transactions est assez exemplaire. André explique le [Théorème de CAP](#) d'Eric Brewer, utile lorsque l'on parle de données massivement partagées : - Strong Consistency: all clients see the same view, even in presence of updates - High Availability: all clients can find some replica of the data, even in the presence of failures - Partition-tolerance: the system properties hold even when the system is partitioned Ce qui donne en français : - L'ensemble des clients voient la même vue, même lorsqu'il y a des mises-à-jour - Massivement disponible : tous les clients peuvent trouver de la donnée répliquée, même lorsqu'un crash survient - Partitionable : votre app-serveur et votre base de données peuvent être installés sur 2 machines, votre système est tolérant au partitionnement. Il s'avère que seul 2 des 3 postulats peuvent être respectés en même temps. Olivier explique qu'Amazon par exemple a fait une architecture en suivant les points 2 et 3. Le besoin en "High-Availability" et en "Partition-tolerance" sont plus importants que la Consistance. Cela crée des fenêtres d'inconsistance, mais avons-nous besoin absolument à la seconde près de la consistance des données ? A réfléchir ! Pour monter en charge, tout ce qui peut être traité avec de l'asynchronisme comme la préparation d'une facture par exemple, la validation d'une commande... doit être traité de cette façon. Cette architecture "Fire and Forget" est très intéressante. Tout d'abord Olivier explique que les données ne sont pas véhiculées, simplement l'endroit où elles se trouvent. L'architecture accède à la donnée le plus tardivement possible. Les systèmes sont fait de telle sorte que si un message est reçu plusieurs fois, cela n'a pas d'impact. L'ordre des messages ne doit pas être important. Pensez à votre lecteur de mail, lorsque vous prenez une conversation en cours, vous relisez les anciens messages commentés dans le message complet par exemple... La synchronisation tue l'architecture distribuée. **Chef on fait une mise en prod** J'ai franchement bien aimé cette partie, merci d'en avoir parlé. Après avoir parlé grosse architecture, les 2 présentateurs enfoncent le clou et parlent de la mise en production. Et là j'ai appris pas mal de choses. eBay est entièrement redéployé toutes les 2 semaines. Cette mise à jour peut prendre 3 jours. Le déploiement du code est différent de l'activation d'une nouvelle fonction. Chaque fonction d'Amazon par exemple est activable/désactivable. La mise en prod d'une fonction revient donc à mettre en marche du code... qui est peut-être en production depuis 3 semaines. Les fonctions sont cachées et activées par un switch ! En cas de panne sur une fonction, on peut même imaginer qu'un administrateur système "éteigne" la fonction en cause. Vive OSGi ! On recherche donc plutôt la robustesse applicative au lieu de la robustesse d'infrastructure. Ils savent que le système va planter, l'architecture est donc construite en fonction de ce postulat, afin que les systèmes reprennent automatiquement lorsqu'une autre partie du réseau repart. La présentation continue ensuite sur la partie humaine de ces projets. Comment font-ils ? Combien de développeurs ? Comment travaillent-ils ? Amazon c'est 1500 développeurs, 30 architectes. Pas beaucoup pour 6 millions de ligne de code. [Werner Vogels](#), CTO d'Amazon.com a une phrase simple : "You build it ? You run it !". Tu l'as fabriqué ? Et bien tu vas le faire marcher. Il demande aux équipes du support de réveiller à 3h du matin le chef de projet du module qui a planté afin que celui-ci se déplace. Rien de plus efficace pour s'assurer de la qualité des développements. Amazon c'est 55 millions de clients actifs. Entre 100 et 150 équipes collaborent pour créer une seule page d'Amazon.com. Chaque équipe est autonome, et elle ne dépasse pas 10 personnes. Werner dit "If you need more than 2 american pizzas to feed a team, split-it". Yahoo! est aussi l'un des premiers utilisateurs de Scrum, l'Agilité est un standard. L'esprit "on est en bêta tout le temps" permet aussi de recevoir du feedback des clients. Ces sites marchent car ils écoutent les clients. Amazon.com ouvre son système avec des Web Services, cela permet à des communautés de développeurs de travailler avec Amazon et d'accroître les ventes. Prenez Gmail, en bêta depuis 5 ans... Bref comme notre univers avance très vite, il est vital d'aller mettre en ligne le plus vite possible une version light de votre architecture, afin de commencer à prendre du recul et à écouter le client. Ne partez pas dans un effet tunnel, faites de l'incrémental, trouvez un moyen de mettre en ligne chaque mois par exemple. Le fantasme *spécification-codage-testage-mise en prod-merdouillage-debuggage-remise en prod-merde ça marche pas-debuggage* n'est pas du tout le modèle des grands du Web. **Conclusion** Présentation vraiment sympathique, pas du tout "*consultant-SOA-on se la raconte*" mais très concrète et très bien tournée. En sortant mes tags sur cette présentation sont : humilité / partager / petite équipe / 400 serveurs par jour se plantent chez Google. Il y a un monde entre l'univers des gens qui tartinent du Word et du PowerPoint avec une cravate et les autres, qui vont vraiment vous construire une architecture, mettre en place des outils, trouver une solution à l'arrache sans craquer, motiver et diriger une équipe. Le jour où un pingouin de 25 ans viendra vous parler de SOA, demandez-lui s'il sait comment fonctionne Amazon, eBay, Google... et par extension, d'autres sites comme SalesForce.com ou même... le Touilleur Express. Bien entendu lorsque l'on parle de ces grands sites, et que dans la vraie vie on est juste entrain de faire l'extranet d'un Assureur Suédois, il y a un monde... Mais prendre le temps de regarder ce qu'ils font nous donne des idées concrètes pour ne pas se planter, et continuer à se faire plaisir. Et entre nous, les Suédoises, ça scale horizontalement non ? **Références** <http://highscalability.com/amazon-architecture> à lire pour retrouver quelques idées et chiffres de la présentation. J'ai peut-être mélangé eBay et Amazon lorsque je cite certains chiffres, n'hésitez pas à corriger.

## USI 2009 Wiki contre CMS

Après [la présentation la semaine dernière à Jazon de XWiki 2.0](#), me voici de retour en face de Vincent Massol de [XWiki](#) et [Tugdual Grall](#) de la société [eXoPlatform](#). Honnêtement j'ai eu peur que l'un prenne le pas sur l'autre, mais ils ont réussi tous les deux à parler de leur domaine de compétence sans aller manger dans l'assiette du voisin. La présentation début par un passage en revue des différences entre un Wiki et un CMS. Ces 2 outils permettent de publier du contenu facilement. Le CMS (Content Management System) est orienté processus, le Wiki est orienté collaboration. La première différence se fait donc sur l'usage. Le CMS est axé sur la publication avec une validation, souvent un workflow, une gestion fine des auteurs, des catégories et des droits des différents utilisateurs. Souvent, le contenu du site publié via CMS est connu à l'avance, on retrouve des documents d'entreprise classiques. Le Wiki est adapté à un processus collaboratif, où la création et la publication sont mélangés. On a plus la notion d'itératif, le contenu s'auto-organise par les différents contributeurs. Souvent, le contenu exact n'est pas connu à l'avance. Chaque contributeur a les mêmes droits que les autres, et il peut défaire une modification. Basé sur cette première observation, on commence à comprendre que les 2 outils ne seront donc pas mis en place dans les mêmes structures, mais aussi que le contenu publié sera différent. Les structures hiérarchiques fortes, où la validation du contenu publié est importante, seront plus du côté du CMS, là où un Wiki sera adapté à des structures en réseau comme une communauté open-source. Tugdual et Vincent apportent une vision professionnelle sur le sujet. Les projets CMS sont mandatés souvent par la DSI ou l'AMOA, là où les Wikis sont poussés par des utilisateurs finaux, par des développeurs, par des installations locales dans des départements de l'entreprise. Petit jeu : pouvez-vous identifier le site réalisé avec un CMS et celui réalisé avec un Wiki ? J'ai refais 2 captures d'écran pour masquer les URLs Le premier site : [France 2025](#)



Le second site : [Canton](#)

de Genève. section Impôts



Le résultat du

sondage dans la salle à l'USI donne le CMS pour France 2025, alors que c'est l'inverse. Le premier site a été réalisé avec [XWiki](#), le second avec [eXo Portal](#). Vincent explique la différence entre les deux : le premier est un site où chaque internaute peut donner son avis. Le second est le site du Canton de Genève en Suisse, c'est la section des impôts qui publie des textes officiels. On comprend alors mieux l'usage de chacun des 2 outils. Se pose la question de l'évolution des 2 outils, de quoi sera fait le futur ? Vincent Massol parle tout d'abord des prochaines évolutions du Wiki, le Wiki 2.0. Là cher lecteur je te propose de relire [un ancien numéro du Touilleur Express](#), j'ai publié un article pas plus tard que la semaine dernière. Tugdual présente ensuite les évolutions des CMS, avec l'ajout d'un grand nombre de fonctions de collaboration qui viennent du monde du Wiki : possibilité de noter des articles, d'ajouter des commentaires ou des annotations. L'édition et la publication sont plus simples qu'au début des CMS, avec la prise en compte des retours d'expérience du monde des Wikis. Le CMS dépasse le cadre de la stricte publication de contenu, et propose aussi l'intégration de différentes applications, de différents Gadgets. Tugdual parle du projet [OpenSocial](#). Il explique la différence entre une Portlet qui tourne du côté serveur, très proche d'une servlet, et un Gadget ou Widget, qui tourne du côté du navigateur en consommant des services distants. Pour terminer la présentation, les 2 présentateurs essayent d'imaginer un monde où les rôles seraient inversés. Que se passerait-il si Wikipedia passait vers un CMS ? Que se passerait-il si le Journal Officiel passait en Wiki ? Chaque type de site a donc besoin d'un outil spécifique, il n'y a donc pas de Wiki contre CMS, plutôt une collaboration intelligente. Chacun dispose de son écosystème. Pour la collaboration, le Wiki est roi. Pour la publication de contenu, le CMS est roi. Et il ne faut pas tenter de forcer les outils dans un autre rôle non prévu, même si cela fonctionne techniquement. **Conclusion** Une bonne présentation qui permet de comprendre en une heure où chacun travaille, où se situent les différences entre un CMS et un Wiki. La salle était

comble, ce qui fait penser que le sujet intéresse. Les 2 speakers s'en sont bien sortis, j'ai trouvé la présentation très équilibrée. Chacun bien entendu a parlé de son produit, mais le fond et les réponses apportées étaient intéressantes. Et puis les deux font du Java, les deux sont en open-source, les deux sont français... Alors que demande le peuple ? La présentation est en ligne [sur le blog de Vincent Massol](#).

## USI 2009 : l'ergonomie entre forme et fonction, une recette par l'exemple

Amélie Boucher présente l'ergonomie, avec une présentation pétillante et un contenu vraiment intéressant. Comprendre l'ergonomie en une heure, contrat rempli. Elle est l'auteur du livre "[Ergonomie Web](#)", un livre de référence à lire par tous les concepteurs et les geeks. Elle est aussi la créatrice du site [ergolab.net](#). Une petite définition pour commencer :

"L'ensemble des connaissances relatives à l'homme et nécessaires pour concevoir des outils, des machines et des dispositifs qui puissent être utilisés avec un maximum de confort, de sécurité et d'efficacité." (Alain Wisner)

Comme elle dit, quand on a dit cela, on a tout dit et en fait on a rien dit. L'ergonomie c'est regarder une application pour qu'elle soit utilisable et que ce qu'elle propose soit utile. - être utile : est-ce que je répons ou non à un besoin ? - être utilisable : pour remplir une fonction, quelle est la facilité d'utilisation ? L'ergonomie est aussi une science pour "améliorer" et pas simplement constater comme elle va nous le montrer. On comprend donc que le travail d'ergonome est d'améliorer la communication entre la machine et la personne. Elle parle d'abord de l'**Utile**, de ce que nous mettons dans nos applications. Quel développeur n'a pas codé des fonctions sans vraiment penser à son utilité ? Une application doit servir à quelque chose, une fonction doit être utile, et il ne faut pas hésiter à faire des tests utilisateurs afin de se rendre compte de ce qui est utile dans un logiciel informatique. Concernant ensuite l'**Utilisabilité**, un logiciel ou un site informatique doit être facile à utiliser, facile à apprendre, instinctif. Avez-vous lu la notice de votre ipod par exemple ? L'utilisabilité c'est aussi proposer une interface qui n'induit pas en erreur, qui ne soit pas ambiguë. Pensez à Windows qui pour éteindre votre ordinateur... vous demande de cliquer sur "Démarrer"... Elle parle ensuite d'une notion intéressante : une application doit paraître simple, au risque sinon de décourager les utilisateurs. Combien d'utilisateurs n'osent pas utiliser un site parce qu'il "paraît compliqué" ? Pensez à votre grand-mère devant votre dernier téléphone portable... La question se pose : mais comment je fais ? Une application ne doit pas retirer de prime abord des fonctions. Il convient tout d'abord de masquer le trop de fonction, de sorte que seul l'indispensable soit visible. Elle démontre son argument avec une application qui présente un mode de saisie avancé, mode qui n'est pas actif par défaut. L'iPod Shuffle par exemple est un exemple de la simplification. L'écran LCD n'étant pas indispensable à la fonction de base qui est "j'écoute de la musique", celui-ci a été retiré. Masquer pour alléger l'interface oblige à hiérarchiser et à créer des workflows, ce qui simplifie le travail du développeur finalement. Le site Adobe Buzzword ou le traitement de texte Pages sur Mac par exemple, sont des exemples de logiciels qui masquent la complexité. Pour certaines fonctions peu utilisées, Amélie explique ensuite le principe de la fonction experte. Prenez par exemple la fonction qui permet de classer les icônes sur l'iPhone. Celle-ci s'active en gardant le doigt appuyé sur un des icônes, et permet de ranger ses applications. Elle est complètement invisible, seul un expert connaît cette fonction... et en parlera autour de lui. Elle continue la présentation en expliquant que le rôle de l'ergonome est aussi de trouver la bonne fonction, celle qui améliore l'efficacité. Pour cela elle nous montre un outil de traduction en ligne. Composé d'une zone de saisie, de deux listes déroulantes avec les langues et d'un bouton traduire. L'ergonome a ajouté un lien "Permuter" qui permet d'inverser les sélections entre les 2 listes, ce qui fait gagner du temps. Les fonctions doivent être optimisées, elles doivent être visibles et utilisables. 4 besoins fondamentaux sont nécessaires : - l'emplacement (caché dans la barre, en haut, dans un menu...) - la nature (zone de saisie, bouton) - le format (gros/petit/couleur) - la terminologie ("nouvel email" moins bien que "envoyer un message") Elle termine la présentation par une explication sur les techniques pour améliorer l'utilité et l'utilisabilité des sites, en parlant des tests à l'aveugle, focus groups, benchmark, questionnaire de satisfaction. Il y a aussi l'appel à l'expert, qui grâce à ses connaissances peut nous aider à concevoir un logiciel utile et utilisable. En conclusion, une bonne introduction sur l'ergonomie pour des Geeks comme pour des Boss. Les développeurs informatiques n'apprennent pas à bien concevoir. Ils apprennent à concevoir. Il y a un petit-chimiste chez nous qui fait que nous sommes très fort pour faire des choses pas utiles ou pas utilisables, même parfois les deux. **Bonus spécial pour toi lecteur** Je te propose pour terminer des exemples de sites pour te faire peur - [Gamequarium](#) a le mérite de se dire que finalement on est pas si mauvais en CSS - [CapstoneIdeas](#) avec un système de navigation très pratique, moi j'ai rien compris - [Horizontal Way](#) : va chercher le nonosse à droite de l'écran - [BowWowBooks](#) un site qui veut vendre des livres mais qui fait tout pour que tu ne comprennes pas qu'ils veulent vendre des livres... ou des croquettes pour chien - Piqué chez FredCavazza : [le pire site web de tous les temps](#), paraît que c'est fait exprès. Encore une bonne présentation de l'USI 2009.

## USI 2009 L'informatique conviviale (voyage au coeur des grandes organisations)

Cette session est présentée par [Pierre Pezziardi](#), Directeur Technique, Président, OCTO, Octopus Micro Finance ainsi que par [Laurent Avignon](#), Directeur Général Adjoint d'OCTO Technology. Sympa de voir 2 Boss mouiller le maillot et se lancer dans une présentation d'une heure, construite comme une pièce de théâtre. La session débute devant 200 personnes, par une présentation de

L'objectif : nous faire partager la conviction qu'un SI est aussi épanouissant à construire qu'à utiliser. Pour cela, chacun joue un rôle. Laurent sera le DSI et Pierre le journaliste, qui lui propose de remettre de la convivialité dans son SI. Laurent Avignon débute par quelques mots qui parlent aux DSI présent dans la salle : *"Mon DG me met la pression, la DSI est en danger, on parle de délocaliser en Inde une partie de mon activité... je n'ai pas le moral. En plus je vois les autres qui réussissent comme eBay, ou Firefox, mais comment faire ?"* Tout d'abord, comme explique Pierre, regardons les différences culturelles entre ces entreprises et la tienne. La gestion du changement fait partie de leur patrimoine génétique, avec des milliers puis des millions d'utilisateurs, ils se doivent d'innover. Le changement est perpétuel. Laurent : oui en effet, mais moi, avec mon SI et mon SLA à tenir, je dois tout prévoir. Je ne peux pas mettre du changement sans contrôle, je ne peux pas mettre en ligne une nouvelle version. Pour l'instant je me sens éloigné de ces grands du Web, que ce soit un eBay ou un Firefox. Pierre explique qu'en effet, changer un standard va te faire courir un risque. Cependant il existe des moyens pour redonner de la convivialité, car tu vas comprendre que le changement viendra de tes équipes, lorsque tu auras rendu plus convivial ton organisation et ton SI. Tout d'abord, il faut un peu de discipline, au niveau des moyens, d'un cahier des charges. Mais c'est aussi une mobilisation de tes équipes, avec un but à atteindre. Est-ce que tes équipes savent pourquoi elles travaillent ? Laurent répond que ce changement va apporter un risque de double culture. Certains vont s'essouffler, ceux qui font de l'Internet s'éclateront pendant que les autres seront sur le côté. C'est un risque trop important, qu'as-tu à répondre à cela ? Tout d'abord l'Entreprise est un organisme. Je te parle là de changer son ADN. Je vais te proposer quelques idées pour changer ton système. Tout d'abord, identifie un "Borat". Une personne prête à changer. Ensuite, si tu veux rendre convivial ton SI, mais que tu penses que le reste ne doit pas changer pour l'instant, je te propose de mettre en place des "Checkpoint Charlie". Ces portes d'entrées-sorties, très spécifiques, permettent de travailler avec le reste de l'entreprise. Les réformistes font peur car ils font changer les choses. Ce n'est pas l'innovation ou l'idée qui est innovante, mais la quantité des innovations proposées, bonnes ou mauvaise, comme le disait hier Luc de Brabandere. Tu dois identifier des leaders, capable de changer, et de mobiliser tes équipes. Laurent reprend : cela me semble encore un peu abstrait, tu peux m'en dire plus ? Pierre : je te propose de changer radicalement ton système d'information. Tu vas retrouver une infrastructure dynamique en changeant tes règles de fonctionnement. Regarde Wikipedia : 7ème site le plus visité dans le monde, 25 Web Masters, 30 000 contributeurs... et ça marche ! Est-ce que tu n'as pas dans ton SI des moyens d'améliorer la communication entre les équipes, en les laissant se contrôler à posteriori ? Pierre continue : Salesforce.com par exemple propose aux développeurs de se créer des outils. Ces développeurs se sont appropriés l'outil et ont développé eux-même des petites applications, sans validation au préalable. Sur le site de Bouygtel, quelqu'un a eu l'idée d'ajouter un forum afin que les clients posent leurs questions, et que d'autres clients répondent. On aurait pu avoir peur et mettre un modérateur... ce qui n'est pas le cas. Dans la vraie vie, il y a 99% de gens normaux qui ne font pas n'importe quoi. Le contrôle à posteriori permet d'effacer les messages qui ne respectent pas la charte du site. Et bien grâce à cette idée, Bouygtel offre un excellent service client qui ne lui coûte rien, avec des réponses très pointues... Et ça marche, c'est convivial. Pierre continue : le pari de la Confiance est opposé à l'idée de contrôle. Tu ne peux pas demander de la Confiance si tu conserves un Contrôle. Il faut donc que tu réfléchisses pour que ton Intranet, ton système d'achat, ton support réseau, tout soit plus convivial. Prenons par exemple l'outil SOCRATE de la SGCIB. Le front-desk avait un souci avec le backoffice. Plutôt que de tenter de concilier les deux mondes, les dirigeants ont laissé les équipes s'auto-organiser. Le système SOCRATE est ouvert, et il réconcilie la vue desk avec la vue compte, sans soucis. Laurent demande alors comment y aller ? Pierre résume ses idées. Tu veux un produit convivial et intégré ? Ce sera ton infrastructure et son interopérabilité qui feront un bon produit. Pour garder cependant de la bonne communication transverse, il faut des normes. Mais ce seront tes équipes qui les trouveront. Tu veux rendre plus ouvert ton SI ? met en place une Forge comme SourceForge.net, un appStore ou un Wiki par exemple. Enfin, ajouter plus de responsables ne rend pas une entreprise meilleure. Des grandes entreprises réussissent avec 3 niveaux hiérarchiques. Si le transverse fait le pari de la confiance, tu dynamises ton SI et tu le rends plus convivial. Le DSI facilite la communication et les échanges, demain je supprimerai ton poste et tu seras promu !

**Conclusion** La présentation était vraiment enrichissante en expliquant simplement que le changement vient avec de nouvelles méthodes, de nouveaux outils et une volonté de réformer. Moderniser son SI, c'est mettre en place un système de communication basé sur de nouveaux outils de l'entreprise 2.0. Chacun prend conscience que le DSI 2.0 a du pain sur la planche. Son travail n'est plus uniquement l'architecture du SI, mais la mise en place d'outils simples, ouverts et conviviaux, pour améliorer la confiance des utilisateurs. La session est difficile à retranscrire, j'ai tenté de vous redonner l'enchaînement, mais si celle-ci est disponible en vidéo sur le site de l'USI, je pense que vous prendrez plus de plaisir en les regardant qu'en me lisant. DSI du futur, convivialité, réformisme, des pointeurs qui nous donnent une idée du futur de demain.

## Revue du livre Spring par l'exemple de Gary Mak



J'ai reçu il y a quelques semaines le livre [Spring par l'exemple](#) de Gary Mak, traduit en français et publié chez Pearson. La traduction en français est d'excellente qualité. Bravo à Hervé Soulard pour la traduction et à Eric Hébert pour la relecture technique. Les quelques 510 pages couvrent le framework Spring en axant l'ouvrage sur la pratique. Chaque thème est construit autour du tryptique "Problème.Réponse.Exemple". Après avoir découvert la configuration des Beans, puis les fonctions élaborées du conteneur, les choses vraiment intéressantes commencent au chapitre sur les proxy dynamiques et l'AOP. Le sommaire du livre: 1. Inversion de contrôle et conteneurs 2. Introduction au framework Spring 3. Configuration des beans 4. Fonctions élaborées du conteneur Spring IoC 5. Proxy dynamique et Spring AOP classique 6. Spring 2.x AOP et prise en charge d'AspectJ 7. Prise en charge de JDBC 8. Gestion des transactions 9. Prise en charge de l'ORM 10. Framework Spring MVC 11. Intégration avec d'autres frameworks web 12. Prise en charge des tests Le chapitre sur la prise en charge de JDBC, suivi du chapitre sur les Transactions, avant d'attaquer la partie ORM, permet d'apprendre les bases d'une bonne utilisation de Spring 2.5 essentiellement. Le Framework MVC est survolé dans les derniers chapitres, l'intégration avec d'autres frameworks traite de Struts et de JSF. La prise en charge des tests permet enfin de terminer par quelques cas. Je trouve le livre plutôt bien fait. Pour un débutant sur Spring, c'est un bon départ. Il permet d'apprendre à utiliser Spring non pas en proposant de long discours, mais vraiment en vous mettant derrière les fourneaux. En tant que sénior, je reste un peu sur ma faim, mais j'ai appris quelques astuces supplémentaires. Le livre est proposé à 39.90 EUR, le site de Pearson vous offre 2 chapitres gratuitement. J'ai aussi parlé avec Julien Dubois de SpringSource lors de l'USI 2009, et il m'a indiqué que la nouvelle édition du livre qu'il a coécrit avec Arnaud Cogoluègues, Thierry Templier, et Jean-Philippe Retaillet est sortie le 9 juillet dernier. Voir [Amazon.fr](#). En attendant je commence à lire le livre d'Antonio Goncalves sur JEE6, qui a dit que l'on ne faisait rien l'été ?

## Paris JUG : retour sur la soirée RIA (Adobe Flex et JavaFx) partie 1

A défaut de son, au Paris JUG on a de l'humour. Je prends mon iPhone et je balance sur Twitter une [contrepéturie](#) pour résumer la galère dans laquelle nous sommes : "**Contrepéturie : panne de micro**". Avant tout, merci à François le Droff et à Simon Ritter. Assurer une présentation avec un micro bricolé avec un ampli, cela a certainement cassé leur présentation, et on est désolé... La prochaine session se déroulera au Zenith avec Gogo-danseuses, et j'ai proposé que l'on investisse dans un micro HF type casque afin que nos speakers puissent utiliser leur PC tout en parlant. Ne loupez pas la saison 3 du Paris JUG et ses nouveaux épisodes, ça sera mieux. **Adobe Flex, BlazeDS, Spring Flex par François le Droff** François le Droff a [un blog](#) que j'ai commencé à lire il y a presque 6 ans, il est d'ailleurs dans la blogoliste en bas à gauche. Il est architecte Java, travaille chez Adobe France en tant que consultant. La présentation de ce soir sera sur les dernières nouveautés de Flex, un tour de LiveCycle Data Services ES, une présentation de Spring Flex et enfin pour terminer une rapide démo de [Flash Catalyst](#). Après un rappel sur Flash, où l'on apprend que près de 700 millions de player flash sont installés, François explique que le moteur de Flash fonctionne avec une VM depuis la version 9, qui est une réécriture très importante du moteur de Flash. Flash 10 apporte encore plus de puissance, le support de fonctions de vidéos avancées, et on retient de cette partie que Flash est maintenant partout. Regardez YouTube! par exemple. Après une présentation de l'offre d'Adobe, un peu marketing mais il faut bien cela pour comprendre la suite, François axe ensuite sa présentation sur la partie serveur, afin de nous présenter Adobe LiveCycle Data Services. Cette suite d'application pour entreprise est l'offre d'Adobe pour créer des applications type Software as a Service, en offrant des services utilisant Flex, PDF mais aussi XML, HTML et Ajax, pour offrir aux clients vos données et des applications riches. L'effort d'Adobe est de présenter des solutions pour que vos données d'entreprise deviennent des services pour vos clients externes sur Internet. Côté client, 4 technologies : Adobe Acrobat Reader (PDF), Adobe Flash, HTML et Adobe AIR. Au niveau conteneur de service, le moteur d'Adobe offre différents connecteurs techniques, et donc différents protocoles. Gestion des données, publication de message, générateur de formulaires et de documents PDF, sécurité et signature numérique, bref le socle de service permet d'externaliser les données avec différents formats, en utilisant différents protocoles, plus ou moins optimisés. De l'HTTP, de l'AMF qui est un protocole ouvert et libre, ou encore le protocole RTMP (real time media protocol) qui sont tous [ouverts](#) et donc que vous pouvez implémenter librement. François explique les différences entre la version professionnelle et commerciale, Adobe LiveCycle DS, et le logiciel open-source [BlazeDS](#). BlazeDS est un composant Java déployé du côté serveur qui permet d'exposer à un client type Flex très facilement des données, en offrant une couche d'échanges de messages. Fonctionne avec Adobe Flash Player ou Adobe AIR. BlazeDS permet de pousser des données vers le navigateur ou le client Adobe AIR en utilisant différentes technologies comme [COMET](#) ou du simple pooling HTTP. BlazeDS est un

WAR que vous ajoutez à votre application pour facilement envoyer et recevoir des messages avec un client Flex. La version LiveCycle DS offre plus de fonctions d'entreprise, avec le support de la sécurité, moins de limitation sur le nombre de Threads, mais je n'ai pas noté toutes les différences. François aborde ensuite une partie pour nous, les geeks : Spring et Flex. Je ne vais pas pouvoir reprendre ici tout ce qu'il a montré, mais on retient au final que Flex s'intègre avec Spring de la manière suivante : nouvelles annotations qui vous permettent d'injecter le RemotingMessagingService de BlazeDS dans votre Bean Spring. Passage via la DispatcherServlet de Spring qui se charge de rerouter vers les composants serveurs de BlazeDS ce qui arrive. ~~Je vous recommande de lire par exemple "Intégration de Flex 3, BlazeDS et Spring sous Flex Builder 3" de Franek Ponchel sur Developpez.com pour comprendre le principe. A lire aussi, le tutorial de Sébastien Arbogast "Flex, Spring, BlazeDS full stack on Adobe Developer Connection"~~.

**Updated:** Les liens pointent vers des articles de l'ancienne version. L'intégration entre Spring et Adobe Flex va beaucoup plus loin, allez voir plutôt [l'article de François](#) qui précise et corrige mes propos. Merci et désolé François ! François nous explique aussi que le compilateur Flex étant open-source, il est très facile d'intégrer maven2 et le compilateur Flex afin d'utiliser par exemple de l'intégration continue pour construire son projet. Il nous montre enfin quelques slides sur [SpringFlex](#), le projet sponsorisé par SpringSource, et qui vous permet d'intégrer Adobe Flex très rapidement dans une application Spring. En conclusion, une présentation plutôt technique comme nous aimons, nous les geeks. Il y avait quelques parties un peu longues à mon goût, et vue les conditions offertes à François ce n'était pas facile pour lui, et il s'en est bien sorti. Donc si vous faites du Flex, allez regardez sur le site communautaire d'Adobe : [opensource.adobe.com](#). **Le buffet, le buffet, le buffet** Il y avait moins de monde ce soir au Paris JUG. 322 personnes selon les organisateurs au Falstaff à la troisième mi-temps, 152 personnes selon la police. Le buffet permet de croiser des têtes connues et de nouvelles personnes. Venir au Paris JUG c'est aussi avec une relation sociale avec des gens comme vous. Je parle de ceux qui portent des teeshirts "Glassfish and Spring Rulez" et boivent du coca light en trempant une chips dedans avant de vous serrer la main. Le vieux geek se reconnaît à sa belle chemise, son costard, un MacBook Pro et un air fatigué. Il est fatigué car il a codé jusqu'à 4h du matin. Il trempe toujours ses chips dans son coca, c'est comme cela que je les repère lorsqu'en me serrant la main, un mélange de sucre et de sel vient recouvrir mes phalanges (que c'est beau...). Le commercial a mis une cravate humoristique pour se fondre dans la masse. Il est super content d'être là, il y a une super ambiance, vous êtes super, dis-donc tu connais pas quelqu'un qui cherche une superbe boîte de super service pour super-pas-cher ? Super ! allez on s'appelle ! Le gars qui vous reconnaît car "VouZeteuLeTouyeur" ça c'est cool. J'aime bien. J'aime aussi la variante, plus marrante : le gars qui vous parle du Touilleur mais qui sait pas que c'est vous. C'est marrant. Y'a le gars qui pense que vous êtes une bête parce que vous avez parlé une fois de Spring AOP sur votre blog. C'est le bonhomme qui m'embarque dans une discussion sur l'AOP et qui parle comme cela : - *Dis-moi le Touilleur, toi qui t'y connais, lorsque je déclare dans mon fichier de conf de bean un auto-proxy, est-ce que je mets bean class 'égal' DefaultAdvisorAutoProxyCreator ou pas ?* - euh... attends je reprends du coca et des chips" - *Est-ce que le tissage avec AspectJ n'est pas uniquement automatique ?* - (moi obligé de répondre) *Spring par la pratique, page 141, je te cite de tête le chapitre : Avec AspectJ le tissage est uniquement automatique. Le seul contrôle dont dispose le développeur est de tisser ou non les aspects à base d'annotation via le tag aop:aspectj-autoproxy. [...] Le bean DefaultAdvisorAutoProxyCreator se déclare comme suit... <bean class="org.springframework.aop.framework.autoproxy.DefaultAdvisorAutoProxyCreator"/> - ...gné ?* - "tiens passe moi encore du coca s'il te plaît... et des chips.... Demande sinon à Antonio il doit savoir cela" Donc ne me posez pas des questions de geek, j'en sais rien. Je suis là pour tremper mes chips dans mon coca-light et ensuite allez bénir avec mes mains le maximum de personnes. Qu'on se le dise ! **JavaFx par Simon Ritter** Allez j'en garde pour la prochaine fois, je vous raconterai JavaFx un peu plus tard. Suite au prochain numéro. **[Update 10 juillet]** J'ai remplacé LiveCycle ES par DS, j'ai confondu les deux. Pour plus de détails, et pour voir les slides, François a résumé la présentation, et il a partagé ses slides [sur son blog](#). Merci.

## Twitter : des documents d'entreprise lâchés dans la nature

J'aurai bien titré cet article "Twitter : hacké" mais comme je trouve que c'est un peu exagéré, j'ai préféré vous donner un titre plutôt soft. Mais revenons aux faits... TechCrunch.com a publié [un article](#) le 14 juillet qui fait du bruit. Michael Arrington a reçu dans sa boîte email un zip avec 310 documents plus ou moins secrets de l'entreprise qui a créé [Twitter](#), le service de micro-blogging bien connu. Il semble qu'un pirate se soit introduit dans le système de Twitter... A ce moment précis ça sent le 22h35 sur TF1, genre Charles Villeneuve qui nous montre Kevin, 14 ans, hacker... En fait c'est tellement bête qu'on en rigolerait presque : le pirate a trouvé un mot de passe simple d'un des cofondateurs de Twitter et il est entré tranquillement dans l'espace Google Docs de Twitter. Avec ce passeport Google, il est ensuite allé sur Google Docs faire ses courses. Et il est revenu avec en effet des documents plus ou moins officiels, des notes de frais, des entretiens d'embauche, des documents de prospections, bref la vie de l'entreprise derrière Twitter. Le journaliste de TechCrunch [se demande](#) s'il doit ou non publier ces documents. Environ 410 commentaires en quelques heures sur le site de TechCrunch, il n'en faut pas plus pour secouer la blogosphère... et Twitter ! Il y a cependant un peu de moral dans l'histoire, l'auteur rassure tout le monde mais glisse cependant quelques détails croustillants comme "Some documents show floorplans and security passcodes to get into the Twitter offices. We're not going to post any of those documents...". Pour l'instant un article seulement [a été publié](#) par TechCrunch qui présente un projet de jeu vidéo "Final Tweet" dont le concept est franchement... pourri. Afin de relativiser un peu, il n'y a pas eu de craquage des systèmes de Twitter ou une attaque par force brute. Il y a eu un vrai travail de recherche de la part de la personne qui est entrée dans le système de Twitter et qui se fait appeler "Hacker Croll". En utilisant les faiblesses existantes des systèmes de Yahoo! puis ensuite de Google, il a simplement trouvé les accès pour entrer dans le système. Au départ, c'est donc une grosse négligence des utilisateurs de base, qui avec un mot de passe commun à Twitter et

à Yahoo! ont réussi à entrer dans le système. Plus grave, on imagine que ce genre d'aventure doit arriver relativement souvent mais que là, nous en entendons parler car il s'agit de Twitter. Le talent et le vrai travail du hacker Hacker Croll qui est français semble-t-il est expliqué dans cet article, où l'on peut lire (en anglais)

Hacker Croll claimed to have accessed Goldman's Twitter password by first gaining access to his Yahoo account. "One of the admins has a yahoo account, i've reset the password by answering to the secret question. Then, in the mailbox, i have found her [sic] twitter password," Hacker Croll said Wednesday in a posting to an online discussion forum. "I've used social engineering only, no exploit, no xss vulnerability, no backdoor, np sql injection."

Son travail aura eu au moins le mérite de rappeler l'importance des mesures de protection dans les entreprises. L'homme est souvent le premier maillon faible dans le système. Si votre entreprise a fait le choix de Google pour sa messagerie d'entreprise et ses documents, il serait peut-être pas mal d'aller faire un audit de votre infrastructure... Imaginez que l'on tombe sur les comptes, les messages privés, les propals commerciales, les CVs des consultants... Pensez au mot de passe du big boss, qui souvent préfère mettre alain1 ou motdepasse de peur d'oublier un vrai mot de passe comme %\$2Xww42Mdq3G4@... Bref Techcrunch fait encore le buzz. Dès qu'une entreprise devient notoire, et que ses documents sont sur le Cloud, elle est plus vulnérable qu'une petite PME au fin fond de l'Ardèche avec son armoire et ses classeurs à l'ancienne. Une bonne leçon pour nous tous...

## Google Wave : premiers tests

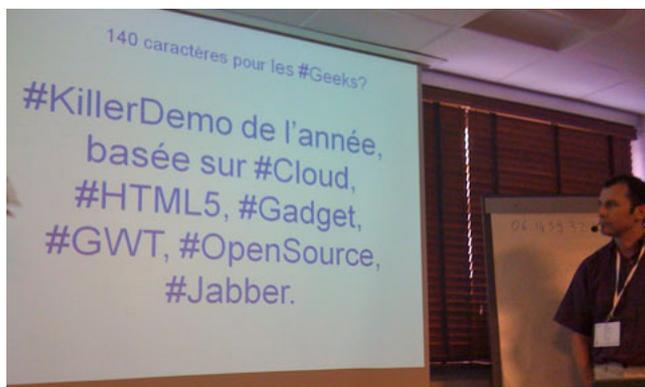
Ce matin un email dans ma boîte aux lettres : "Bonjour Nicolas Martignole, voici vos codes d'accès à Google Wave... Amusez-vous bien !". Si le mail n'était pas arrivé un dimanche matin à 00h03, j'aurais bien cru un moment que j'étais une personne très importante pour que l'on me donne un compte... En fait Google a annoncé [le 17 juillet dernier](#) que 5800 comptes Google Wave avaient été distribués lors de la conférence Google I/O et qu'au cours de l'été, 20 000 nouveaux comptes seront ouverts. Bref c'est donc une histoire de temps aussi pour vous si vous avez fait la demande. Premiers arrivés, premiers servis. Je m'étais inscrit quelques heures après l'annonce, grâce à l'information trouvée sur Twitter. Si vous souhaitez faire une demande de compte, rendez-vous [sur la page Google Wave](#). Bref avant de vous parler de mes premiers pas avec Google Wave, j'en profite pour revenir sur [I'USI 2009](#). Suite à sa présentation des outils Google pour l'entreprise, Didier Girard de SFEIR a proposé d'organiser une présentation de Google Wave avec Guillaume Laforge de SpringSource, chef de projet de Groovy. C'est donc dans une petite salle de 50 personnes que nous avons pu voir Google Wave pour l'une des premières fois en France.



Guillaume Laforge (à gauche) et Didier Girard (à droite) débutent la présentation par une diapo qui explique Google Wave en 140 caractères, à la Twitter. Notez cette idée pour vos prochaines présentations. Si vous n'êtes pas en mesure de résumer votre contenu en un message twitter, vous pouvez encore bosser. Excellente

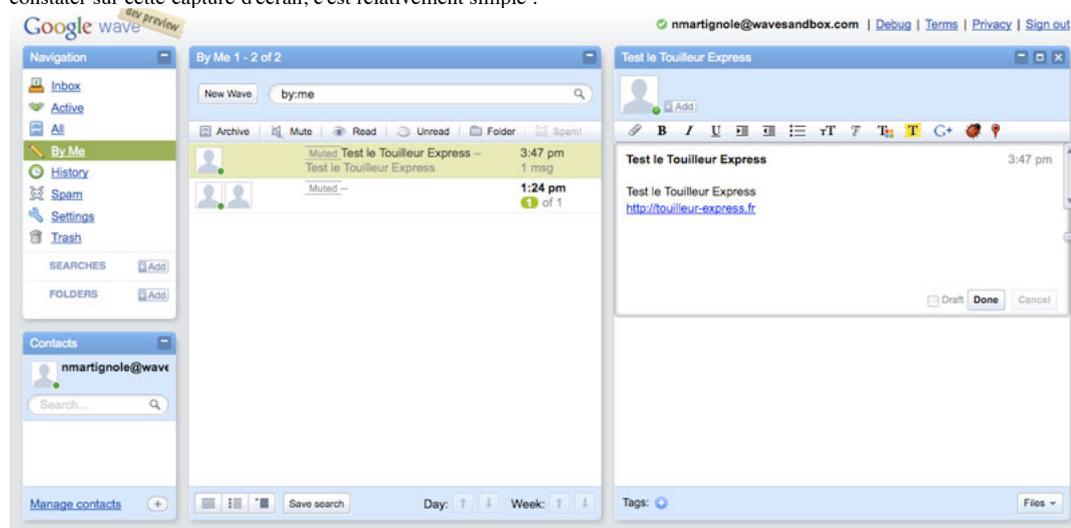


idée de Didier. J'aime bien aussi l'idée de Didier de se présenter non pas comme un Geek ou un Boss mais comme... un



Beek ou un Goss !

**Google Wave en quelques mots** Wave est le nouvel outil de communication propulsé par Google qui ressemble à un client de messagerie électronique, auquel des fonctions de messageries instantanées ont été ajoutées. Lorsque vous éditez un message pour répondre, et qu'un autre destinataire est en ligne, vous voyez littéralement l'autre personne "taper" son texte en direct sous vos yeux. C'est donc un mélange de twitter, de client mail et d'IRC/Chat. Google Wave c'est aussi le concept de Robot qui vient du monde IRC. Ces invités virtuels peuvent être ajoutés à votre conversation. Lorsqu'ils sont présents, le texte que vous tapez est analysé en temps réel afin de l'enrichir et de fournir de nouveaux services. Vous tapez un numéro de téléphone ? Un robot le reconnaît et ajoute une URL skype pour appeler le numéro. Vous tapez une adresse ? Un lien vers Google Maps est ajouté automatiquement. Il est possible aussi d'ajouter des robots capables de traduire à la volée votre texte. Dans la vidéo de la démonstration, on voit un francophone écrire un texte en français, et celui-ci est traduit en anglais à la volée. Il est donc possible d'imaginer de nouveaux outils, qui seront eux-même hébergés sur Google App Engine, la plateforme de Cloud Computing de Google. Didier Girard a implémenté un Robot capable de reconnaître un tag SMS. Lors de la démonstration, il tape "SMS:0600112233 ceci est un test", et quelques instants plus tard un message est affiché sur son téléphone portable. Un robot écoute la "Wave" et a reconnu le mot clé "SMS:". Après avoir extrait le numéro de téléphone, il se charge d'envoyer un vrai SMS au téléphone de Didier. La démonstration va plus loin puisque nous avons tous participé en envoyant un SMS qui s'est affiché... dans la Wave ! **A quoi pourra servir Google Wave ?** Après la démonstration, je discute un peu avec Didier. Nous sommes d'accord pour dire que nous venons de voir "quelque chose" mais qu'il est encore tôt pour se rendre compte de la portée de ce nouvel outil de communication. Quelque part j'essaye de me souvenir de la première fois que j'ai utilisé un client de messagerie en 94 à l'Ecole... De la première fois où j'ai envoyé un SMS à un ami, où j'ai commencé à Twitter... On ne se rend pas alors très bien compte si ce que l'on fait fera parti de notre quotidien dans quelques mois ou non. Je me souviens aussi de projets comme [ICQ](#) en 97, que vous connaissez peut-être. Mais qui se souvient en 1996 du projet Marimba, fondé par [Kim Polese](#) une ancienne dirigeante de SUN Microsystems ? A l'époque c'était THE application. Le principe consistait à pousser de l'information vers le client, un peu comme nos Widgets de météo et d'information sur Mac. Racheté par BMC Software en 2004, le produit n'existe plus... Kim Polese était la responsable produit de Java chez SUN en 1996... Je m'en souviens car j'ai travaillé avec SUN France à cet époque et que Marimba était le truc à la mode. Alors à quoi pourra bien servir Google Wave ? J' imagine de vrais usages, comme lorsque répondez pour la 6e fois dans un email, et que votre email ressemble à une conversation de Chat et que vous perdez un temps précieux avec vos correspondants. J' imagine aussi ce nouveau système de réunion, à la fois tableau blanc interactif, document partagé mais aussi plus proche du courrier électronique... De mon côté j'essayerai bien d'écrire un Robot pour Wave. Mon Geek Day Robot vous proposerait de vous tenir informé des prochaines soirées de Geek, de vous dire qui viendra et de vous donner quelques informations. Là expliqué comme cela, je sens pas un grand enthousiasme. Mais pas de soucis, je vous montrerai cela dès que possible. **L'interface de Google Wave** Comme vous pouvez le constater sur cette capture d'écran, c'est relativement simple :



La zone de Navigation sur la gauche rappelle un client de messagerie électronique classique. La zone Contact permet en bas à gauche de retrouver vos amis. La zone centrale marquée "By Me" sur la capture d'écran, est la zone qui liste les différentes Waves. A droite enfin, le contenu de ces

fameuses Wave. Par rapport à un email classique, les correspondants/contributeurs de la Wave sont listés en haut. Les Robots lorsqu'ils sont invités par le créateur de la Wave, apparaissent en haut.

**Bon alors, raconte...** Quelques minutes après m'être authentifié, je commence par lire les messages dans les plus grosses Waves. Mes premiers pas me font penser à un Wiki qui serait mis à jour sous mes yeux. Je vois ainsi un utilisateur entrain de compléter un des articles. Chacun de ses caractères, presque la vitesse à laquelle il tape, c'est assez bluffant. Google Docs propose un système similaire mais croyez-moi, il y a une différence entre les deux. J'y reviens après vous avoir laissé admirer une autre capture d'écran :

The screenshot shows a Google Wave inbox on the left and a 'Google-Wave Extensions List' on the right. The inbox contains various messages with timestamps and participant counts. The extension list on the right includes sections for 'Robots' and 'Text Conversion' with several entries and their descriptions.

Une Wave c'est donc un document partagé avec l'enregistrement de la conversation des différents contributeurs du document. Une ligne de temps avec des commandes "avance rapide", "retour au début" permet de rejouer la Wave et de voir l'avancement des discussions entre les membres de la conversation. Ou comment rejouer de l'instantané... Imaginez un instant l'espace de stockage et la puissance nécessaire pour réaliser ce boulot, qui n'a rien d'exceptionnel, à part le fait que...

personne d'autres ne pourrait le faire. Google s'appuie sur sa plateforme de Cloud Computing afin d'ajuster la quantité de puissance (et d'énergie électrique un jour) pour que le système fonctionne correctement. Chacun de vos caractères est envoyé au serveur, qui peut ainsi retourner aux autres contributeurs vos messages. Et l'édition à plusieurs fonctionne même très bien, bien mieux que sur Google Spreadsheets. Techniquement, Google Wave c'est du GWT du côté client ce qui rassure tout le monde quant à la pérennité de GWT. Cette fois-ci c'est bon, GWT est la technologie de rendu avec HTML 5 de Google. Côté serveur, une API permettra aussi d'intégrer ses propres Waves dans son site. Dès demain, nous pourrons donc avec une Wave attachée à mes billets sur le Touilleur Express, et chacun de vous pourra alors voir les autres lecteurs et discuter en temps réel à propos de l'article... *"Hé tu as vu comment il raconte encore n'importe quoi ? Waaah il est trop fort ce nicolas, je vais appeler mon chien Touilleur Express en hommage ! Tu sais que depuis que je lis le Touilleur Express j'ai les cheveux qui repoussent ?"*. Notez que je ne vous encourage pas à dire n'importe quoi... Comme toujours je suis très sérieux... Bon c'est pas tout, mais j'ai un joujou avec lequel m'amuser alors je vous laisse. Y'a la zone de commentaires froide en dessous où tu peux laisser un commentaire si tu veux. **Références** - Je vous recommande [l'article de François Goldgewicht](#) qui présente Wave en quelques mots. - [Faire une demande pour un compte Google Wave - La page de démarrage](#) de Google Wave SandBox, une fois que votre compte est actif - [L'article de Didier Girard](#) sur Application-Servers.com  
<http://francois.goldgewicht.com/2009/06/03/google-wave-en-quelques-mots/>

## Résultats de l'enquête sur l'adoption des méthodes Agiles en France

A l'initiative du [French Scrum User Group](#), association française loi 1901 destinée à promouvoir les méthodes Agiles en France, une enquête a été réalisée au deuxième trimestre 2009 auprès de 150 entreprises et d'un échantillon de 230 personnes. L'enquête a été aussi traduite par Luc Legardeur en anglais, Jeff Sutherland et James Cundiff ont aussi cosigné cette enquête. Je vous laisse la découvrir directement sur le site du French SUG : [en français](#) ou en [anglais](#). Bonne lecture et merci à l'ensemble des participants.

## Culture bolchevique

Ce matin, surprise en ouvrant mon navigateur chez le client où je travaille. Google Reader est bloqué par le proxy applicatif qui filtre le trafic et donc notre activité sur Internet. Je vous passe les noms d'oiseaux que j'ai utilisé, mais franchement, ras-le-bol. A cela s'ajoute des sites certainement très dangereux comme Amazon S3, ce qui nous empêche de récupérer des ressources Spring pour travailler. Par contre YouTube marche entre midi et deux, histoire de ne pas louper la dernière cascade limbale de SecretStory... Je comprends le responsable d'équipe, inquiet de voir une équipe de développeur entrain de surfer sur Internet, lire des blogs à 13h50 en mangeant un sandwich,

bref très inquiet à l'idée que ses poulets biologiques puissent décompresser 3mn avant de reprendre une activité de production aussi excitante qu'un string taille 62... Y'a de quoi déprimer. Alors oui, on devient mal poli et subversif. Bloquer l'accès à des sites de piratage et de peer-to-peer, bien entendu. Mais bloquer un système qui permet de lire d'un coup d'oeil les derniers messages sur les blogs, je trouve cela super dommage. Et je reste poli pour que ce message ne soit pas censuré par le moteur en question... Enfin heureusement, comme l'outil de filtrage est super bien configuré, je me suis aperçu que [www.google.com/reader](http://www.google.com/reader) était bloqué, mais pas d'autres urls comme [www.google.fr/reader](http://www.google.fr/reader). (gros soupir) Et vous ? est-ce que votre activité de surf est limitée en entreprise ?

## USI 2009 : Vaincre la Force du Plouf par Guillaume Duquesnay

L'équipe de l'USI 2009 a mis en ligne [les vidéos de la conférence USI 2009](#). Encore une fois : bravo les gars, c'est impressionnant. Nous sommes le 22 juillet et toutes les vidéos des grandes conférences sont en ligne. La qualité, le montage, la musique, très gros boulot de [Brainsonic](#) qui a réalisé l'ensemble du travail pour l'USI 2009. J'ai donc repris mon carnet de note, mon sac à dos, et je suis allé m'asseoir au fond de la salle de la présentation de [Guillaume Duquesnay](#) d'OCTO Technology. Et bien c'était vraiment intéressant. Si les mots [procrastination](#) et "Projet qui n'a jamais démarré" vous disent quelque chose, cette session est pour vous. La vidéo de la présentation de Guillaume est [en ligne](#), ainsi que les autres vidéos comme la session de Neal Ford, une des meilleures présentations de l'USI 2009. **Le Plouf** Guillaume se propose de nous présenter un animal dangereux : le Plouf. Le Plouf c'est cette force invisible et mystérieuse qui vous empêche de passer à l'Action lorsque vous avez pris une décision. Pensez à un petit travail de bricolage chez vous, à une étude que vous avez décidé de faire mais qui n'est pas commencée, à un blog que vous voulez mettre en ligne mais pour lequel il n'y a toujours rien... Le Plouf est silencieux, sournois. Le plan de la présentation sera le suivant : identifiez le plouf, mettez en place une tactique, une stratégie et passez à l'action pour le combattre. Guillaume commence par nous parler de ce projet que nous avons tous connu : vous avez vu ce projet qui disparaît un jour ? et qui n'en finissait pas de mourir depuis 1 an ? C'est le Plouf. Où vit le plouf ? partout autour de nous. C'est la patate chaude, quand on ne sait pas vraiment si dans un mois il faut mettre en ligne ou pas le service, quand on a pas vraiment encore un plan d'action, c'est le Plouf. Le Plouf vit donc dans l'ombre, dans le non-dit. Il chasse quand ? Plutôt lorsqu'une décision a été prise, mais que le passage à l'action ne se fait pas. C'est son territoire favori. On observe que dès que l'action débute, le Plouf est vaincu. Pour peu que quelqu'un commence à faire une première action, et le Plouf recule. En résumé, il chasse à l'aube des projets. Guillaume parle d'un projet où il est passé. 80 personnes en cadrage de projet depuis un an... et toujours pas d'actions. Le plouf attaque aussi les troupes... Alors est-ce que c'est pas une chose salutaire que les choses ne se fassent pas ? Quelles sont les conséquences ? Il explique qu'il a été chef de projet sur un projet qui pendant 2 ans attendait un go ou même un no-go. Difficile à vivre, on reste dans l'attente, on se démotive, et il a terminé ce projet en démissionnant. Il aurait aimé une action : le projet est annulé. Merci, au revoir... Mais là le Plouf s'est amusé. Guillaume s'est épuisé à essayer de faire des actions. Le Plouf est donc une espèce endémique, et ses effets sont très dangereux pour les équipes qui se démotivent. Il faut donc l'attaquer dès que possible. **La tactique pour le vaincre** L'action est l'ennemi du Plouf. Voyons les stratégies pour le vaincre. Guillaume dit : "il y a l'angoisse de la page blanche, pas l'angoisse de la page 4". Une fois lancé, vous savez que vous allez vaincre le Plouf. (C'est la Procrastination). La tactique proposée par Guillaume est la suivante : déclarer la première action, le petit a) de votre projet. Prenez quelque chose de très concret. Et tenez-vous y. Cette chose doit être facile à faire. Ce peut être un email que vous envoyez, une personne que vous allez rencontrer, un achat pour commencer vos travaux. Mais pensez à quelque chose de simple. Getting Things Done. Visualisez dans votre tête cette action, imaginez que vous allez réussir. Elle doit être possible et réalisable. Pensez que vous avez tout ce qu'il faut pour la réaliser, et lancez-vous. Le 2ème point : l'énergie. Vous vous souvenez de cette superbe réunion avec l'autre équipe ? et de la bonne ambiance ? et puis à la fin au lieu de commencer, vous vous dites : bon on va manger. Et puis plus rien. Il propose de débiter une première action dès la fin de la réunion, lorsque l'énergie est là et que les gens que vous avez eu du mal à réunir sont là. Ne les laissez pas partir, lancez Eclipse/IDEA IntelliJ et regardez ensemble ce que vous pouvez faire. Le plouf n'aime pas l'action. Si on commence à agir on a gagné. Pour cela il faut donc **décider et faire**. Sur une réunion de 2h, passez 1h30 pour décider et 30mn pour commencer. Hop c'est parti, le Plouf n'aura pas le temps de s'installer. Utilisez le même espace temps pour décider et puis pour faire. Guillaume passe ensuite à un sujet qui m'intéresse : le marketing. C'est l'action qui consiste à faire réagir le consommateur afin qu'il s'intéresse à votre produit. Pour vaincre le plouf, lorsque vous demandez quelque chose par email par exemple, **faites une demande clé en main**. Pensez à mettre un objet Doodle pour que vos destinataires puissent voter directement, pensez à mettre un sujet explicite pour votre email, à donner toutes les informations et à être très précis dans les engagements. Observez les emails de Marketing que vous recevez : ils sont clairs. Peut-on en dire autant de votre dernier email au support ? Guillaume revient sur une présentation qu'il a fait l'an passé avec Benoît Lafontaine : la méthode [PRACMA](#). Cette méthode enseignée à Harvard et à l'INSEAD est une méthode très compliquée avec une certification bien plus chère que la certification Scrum. C'est dire... Non en fait PRACMA c'est "Project Review At The Coffee Machine" copyright Guillaume Duquesnay et Benoît Lafontaine. Leur mission était de faire de l'architecture transverse, d'aller voir 31 équipes. Pour cela, au lieu de faire de la réunionnisme aigüe, ils ont eu l'idée "marketing" de proposer un café pour discuter avec chacune des équipes. Ils ont donc rempli leur mission, tout en séduisant les clients. Très astucieux ! Guillaume nous demande aussi de nous intéresser à nos sponsors. Qui est intéressé par votre projet ? Avez-vous un marché pour votre projet ? Des clients potentiels ? Si le projet ne démarre pas... y-a-t-il une demande réelle ? Est-ce qu'il y a un sponsor qui pousse pour que le projet démarre ou pas ? Allez chercher des sponsors, faites-vous connaître, et vous verrez que votre projet va se relancer tout seul. Ou pas. Il nous conseille aussi, sur la base des valeurs Agiles, de mettre à profit les vrais contacts avec les personnes plutôt que de se mettre

un préservatif appelé Lotus Notes pour communiquer avec vos sponsors, vos clients et vos chefs. **Et si le Plouf est là** Imaginons qu'après avoir lu mon charabia, et avoir vu la vidéo de Guillaume, vous vous rendez compte tout d'un coup que votre projet est entrain de se faire manger par le Plouf... Que faut-il faire ? Braquez la lumière sur le plouf. Votre projet est entrain de mourir ? Criez haut et fort que vous abandonnez le projet, et vous battez le Plouf. Cependant il y a des endroits où l'on ne dit pas non, où le politiquement correct, le diplomatique fait que le Plouf peut s'installer plus durablement. Et bien battez-vous. Guillaume dit "Faites passer votre message, risquez l'explicite plutôt que le non dit, mettez en lumière". Bref agissez. **Et si je suis tout seul ?** Le Plouf appliqué à soit-même, c'est aussi possible. Parlez-vous et pensez à la première petite action concrète pour lancer votre projet. Pensez "décider et faire" et faites-vous une offre clé en main. Par exemple, préparez cet email que vous devez envoyer, afin que dès demain vous puissiez passer à l'action. Essayez de vous motiver en pensant au résultat. Plutôt que de marquer "Payer ses impôts" dans votre agenda, vous pouvez marquer "Paye tes impôts si tu ne veux pas payer 10% de plus". Marketez-vous vos objectifs. **La Stratégie** Le Plouf aime l'espace. Si votre équipe est divisée en 2 open-spaces : pas bon. Faites le test suivant : dites à l'équipe "on part prendre un café". S'il faut plus de 5 mn pour que l'équipe se mobilise, il y a de l'espace pour le Plouf. Les gens doivent se rencontrer pour avancer, pour faciliter la communication. Votre camp de base doit donc favoriser la communication, votre équipe ne doit pas être trop grosse. En 2 minutes vous devez être déjà à la machine à café. Excellente idée de Guillaume. Vous décidez d'attaquer le plouf ? faites le plus au plus vite. Une date est décidée pour votre release ? ne la mettez pas à une autre date. Il vaut mieux échouer dans la lumière que de continuer dans l'ombre. Tenez vous à vos engagements. Ne laissez rien de "pas sûr" ou pas décidé. Moi-même dès demain avec Karim je vais parler de cette mise en production d'une nouvelle partie, et on va s'y tenir. Pour repérer ce qui va faire "Plouf", pensez que celui-ci est sournois, silencieux et invisible. Vous pensez que vos prochaines actions ne sont pas très claires ? Vous êtes dans l'habitat du Plouf ! Sortez vite et bougez vous ! Si vous sentez qu'il n'y a pas de réactions à vos emails, si vous sentez qu'il ne se passe rien, vous êtes entrain de vous faire manger par du Plouf. En tant que consultant, Guillaume propose aussi de laissez le plouf s'installer pour que les gens s'en rendent compte... Vous devez alors l'éclairer et dire ce qui s'est pas bien passé. Cela a foiré. Dites-le !!! Profitez de l'exemple d'un ratage pour agir et donc vaincre cette force invisible. Guillaume a alors une excellente phrase d'un philosophe célèbre :

"Si ça peut saigner, ça peut mourir" Arnold schwarzenegger in Predator

**Pourquoi le plouf s'installe ?** Nous sommes d'excellents Geeks, de très bons Boss. Pourtant dans tout ce qui vient d'être expliqué, nous avons tous un projet qui a fait Plouf. Alors pourquoi ? Guillaume explique alors que nous ne sommes pas armés pour faire du marketing et de la communication. Et là je suis vraiment d'accord. La culture française dans les Ecoles d'Ingénieur nous apprend que parce que ce que nous faisons est génial, nous serons reconnus et notre projet sera récompensé. C'est faux. J'ai travaillé dans une équipe américaine pendant 2 ans 1/2 chez Reuters. Et c'est là que j'ai appris à faire des présentations Powerpoint, des diagrammes Visio, des présentations percutantes. Car dans la culture américaine, le Show-Off et le marketing sont aussi importants que le contenu. Je vous rassure : si votre produit est une bouse, même avec de beaux slides vous ne le vendrez pas. Je parle bien de l'importance de marketer votre projet, de communiquer efficacement pour vaincre le Plouf. Alors soignez la communication visuelle, orale et pensez marketing. Votre génie on s'en fiche, il n'y a que vous qui êtes excité par Scala. Par contre vendez-moi votre projet. Le mot de la fin de Guillaume : **Nommez, convoquez et vous le battez** Guillaume merci pour cette présentation. J'aurai aimé la vivre en vrai, mais avec la présentation de Google Wave en face, difficile. Mais encore une fois, merci.

## Développeur après 31 ans ? Ridé et chauve tu seras

Aujourd'hui c'est facile, vous avez 31 ans, vous ne perdez pas trop vos cheveux ou les rides n'ont pas fait d'apparition sur votre visage. Au niveau Java, vous pouvez facilement calmer un petit jeune en expliquant comment fonctionne une [PriorityBlockingQueue](#). Vous avez commencé à regarder Groovy ou Scala. Les livres [JEE6](#) et Spring par la pratique ? Oui vous les avez achetés. Mais vous n'êtes encore certain de les lire. Mais vous les avez achetés. Cette année ChefSympa est muté à Londres et donc GrandChef est passé vous voir. Enfin disons qu'il est venu vous voir à la cafétaria alors que vous étiez entrain de discuter avec un chasseur de tête qui cherche un jeune de 23 ans avec 10 ans d'expérience en Scala. Bref GrandChef vous sert la pince : "*Bonjour Pierre, je n'irai pas par quatre chemins, est-ce que tu serais intéressé pour prendre le poste de ChefSympa?*". Et vous de répondre : "*Ah oui avec plaisir ! merci GrandChef !!! et je m'appelle Thomas, mais pas grave*". Alors vous êtes chef. Nouvelles cartes de visites, téléphone portable, accès plus important à Internet (vous pouvez aller sur Facebook). Vous avez aussi un nouveau bureau. Votre salaire sera réévalué en fin d'année. Et vous avez des gens à gérer. Wesh ! super, vous vous voyez enfin PetitChef et donc vous êtes super heureux. Notez que vos cheveux ne repoussent pas, vos rides non plus, mais on en parlera plus tard... Ce qui est dur depuis que vous êtes PetitChef c'est que vous n'avez plus le temps de lancer Néquipseu ou IDEA IntelliJ. Et le matin, comme maintenant vous devez dépiler 167 mails, vous n'avez plus le temps de lire des blogs techniques... Et par contre vous avez installé Microsoft Powerpoint, Visio 2007 avec une clé (merci Baztoune) et même Microsoft Project. Vous êtes PetitChef. Quelques jours passent... C'est sympa de gérer ses anciens amis, vous pouvez maintenant prendre des décisions. Vraiment trop génial. Bon vous en êtes à 320 mails par jour. Grâce aux filtres avancés de Lotus Notes, que vous avez trouvé tout seul, vous savez classer les emails: - URGENT - TRES URGENT - MEGA URGENT. C'est très pratique. La pression... non pas cette bière que vous buviez avec vos anciens amis le jeudi soir. Plutôt ce machin sournois où vous vous rendez compte que vous n'avez pas démarré Néclipseu ou IDEA IntelliJ depuis 3 semaines, que vous ne suivez plus les discussions techniques et que vos ex-amis-nouveaux-collaborateurs sont assez difficiles à gérer... Enfin vous avez fait des progrès avec

PowerPoint, et au niveau des diagrammes de Gantt vous êtes aussi plutôt fort. Vous êtes en fait *PetitChefDeProjet*. Ce qui vous donne un droit et un devoir de venir rassurer chaque mois les gens du métier pour leur montrer l'avancement du projet. On vous présente comme "*Pierre, non Thomas pardon, c'est l'ANCIEN architecte de Krousties et qui a pris la place de ChefSympa*". On parle de vous comme d'un ancien footballeur international qui serait maintenant en charge de l'équipe des Poussins du village. Vous voyez Laurent Fignon qui commente le Tour de France ? Pareil : rides et pas beaucoup de cheveux. Un peu dur de débarquer sur la planète GestionDeProjet-Marketing-Politique. Dans votre ancien référentiel, votre niveau en Java et votre connaissance vous donnait une assurance qui vous semble un peu loin ce matin devant 8 personnes, qui se demandent si vous vous appelez Pierre ou Thomas. Vous vous rendez compte à cet instant précis que vous commencez à regretter Eclipse (que je l'écrive une fois bien). **Mais comment en êtes vous arrivé là ?** Un jour, il y a un an, ChefSympa vous a dit gentiment : "*Tu sais Thomas, il faut penser à ta carrière, tu ne vas pas être développeur toute ta vie, tu as déjà 30 ans ! et tes cheveux ! c'est un signe !*". Donc première idée répandue dans les couches du management **français** c'est qu'il faut devenir manager de quelque chose à 30 ans, bref dès que vous mettez senior sur votre titre. Grosse connerie. Deuxième motivation pour devenir PetitChef : votre salaire. Architecte Senior Expert Maître Jedi Java Level 4 ça gagne très exactement entre 50 et 59 000 EUR par ans avec 8 années au compteur. Autant dire une belle somme. Mais vous avez bien compris qu'il faut devenir "PetitChef" pour passer le cap fatidique des 59k et enfin décrocher le super-jackpot pour vous payer une opération de chirurgie esthétique pour vos cheveux. A vous la route des 65 kEUR ! On vous fait comprendre que vous devez prendre des personnes sous votre responsabilité, ce qui aura un effet sur votre salaire, ce qui est vrai, mais à quel coût... Troisième effet : depuis que vous êtes avec Femme 1.0 (ou Mari 1.0 ou Copain 1.0), et bien vous ne pouvez plus lire [JEE6 for Beginner](#), même dans les toilettes (spéciale dédicace à Charles du Paris JUG). Bref votre niveau de geekitude descend de jour en jour, vous n'avez plus le temps de lire le Touilleur Express. Pour la peine, vous serez chauve. Bien fait. Votre conclusion c'est qu'il semble difficile de continuer à être Geek, alors autant devenir Boss. **Soyons sérieux** Tout d'abord, pour redevenir un peu plus concret, je pense qu'il est indispensable que des développeurs séniors de plus de 40 ans, motivés et heureux, continuent à travailler. Si nous regardons notre métier, il reste très artisanal. Ludovic Cinquin d'OCTO Technology disait en 2008 quelque chose comme ça : *[...] notre métier est artisanal, et il manque en fait la notion de [Compagnonnage](#). Dans d'autres métiers, les anciens apprennent aux juniors, et le savoir se transmet de génération en génération[...]*. Dans l'industrie informatique, le cycle étant plus court, il y a deux soucis. Tout d'abord il y a peu d'anciens, si je pense à Java. Ensuite, ces anciens sont souvent devenus des Managers, et il est alors difficile de transmettre son savoir. Je pense qu'il est donc urgent de défendre les personnes qui souhaitent continuer à développer, avec certainement une expertise à valoriser. Pourquoi ne pas proposer à ces personnes des moyens pour transmettre leur savoir ? Ce passage de connaissances doit aussi s'accompagner d'une reconnaissance financière. La voie du management doit être aussi revue et adaptée aux anciens Geeks. Un très bon ingénieur ne fait pas un très bon manager. Et bien souvent, les promus se retrouvent peu ou pas accompagnés. Il y a des exceptions comme chez Thomson-Reuters, où les chefs de projets reçoivent une formation poussée pour apprendre à gérer une équipe et à animer un projet. Il y a aussi la voie des passionnés. Regardez ceux qui continuent à 35 ans d'écrire des livres, avec des cheveux très longs, à s'occuper du Paris JUG... Ou ceux qui présentent une démo de Google Wave en disant très simplement "Je suis un Goss, je suis un Beek, mais pas un boss"... Et puis il y a vous. 31 ans c'est jeune. Dans 9 ans, vous serez expert Groovy-Scala. Java 7 sera peut-être sorti, mais en tous les cas, vous vous serez bien éclaté. Soit vous serez devenu manager, soit consultant, soit indépendant, soit encore mieux : Vous serez devenu un développeur chauve de 40 ans, heureux et fier de l'être ! Ou une développeuse ridée et heureuse ! **Et toi ?** Je suis heureux. Bientôt 34 ans, encore pas mal de cheveux. Indépendant c'est bien après 10 années d'expérience, dont 3 de PetitChef. Je serai peut-être manager dans un an, peut-être consultant indépendant, peut-être Architecte ou Développeur... Ridé et chauve. Comme Yoda en fait...

**Nouveau:** Si vous avez aimé cet article vous pouvez le partager sur Internet avec le lien suivant: [Partagez cet article](#)

## Java Bar Camp 5 à Paris dans le Jardin des Tuileries

Jeudi 27 août 2009 à partir de 18h30, rendez-vous dans le Jardin des Tuileries à Paris non loin du Musée du Louvres pour le 5ème Java Bar Camp. Après Paris Plage, voici donc le concept de Java Bar Camp en plein air, ce qui est une excellente idée. Les détails de l'événement sont sur [le site du BarCamp](#) ainsi [qu'une carte précise](#) pour être certain de vous retrouver avec les autres personnes intéressées. Comme toujours vous pouvez aussi retrouver cet événement sur [l'agenda du Touilleur Express](#), en attendant la sortie du projet Geek Day sur lequel je travaille en ce moment. De mon côté je vous prépare un numéro double spécial été du Touilleur Express avec les meilleurs articles des 6 derniers mois, afin de bronzer tout en vous cultivant. Rendez-vous la semaine prochaine pour ce numéro spécial.

## La JSR330 "Dependency Injection for Java" acceptée par l'executive commitee

[Quelques semaines en arrière](#) je vous parlais du débat entre la JSR-330 (@Inject) et la JSR-299 (Java Context and Dependency Injection). La JSR-330 est une spécification relativement légère qui vise à standardiser les annotations d'injection de dépendance. La JSR-299 est une spécification importante pour JEE6 et Java en général, qui permet de connecter différents beans facilement dans une application. L'histoire se termine bien puisque début juin, [l'executive commitee](#) JCP EE/ES a approuvé la JSR 330 avec 14 OUI, une abstention (Red Hat) et un non votant (Nortel). Sun,

Ericsson, IBM, Red Hat et Oracle ont demandé un effort de coordination entre les 2 expert-groups. La spécification sera dirigée par Rod Johnson (SpringSource) et Bob Lee dit "Crazy Bob" (Google Inc.) (et pas Doug Lea comme écrit ce matin, merci @alexismp). Quant à savoir si celle-ci fera partie finalement de la spécification de la JEE6, Antonio Goncalves sur le podcast "[Les CastCodeurs](#)" semble expliquer que ce sera possible. Affaire à suivre donc. Le sujet est important pour Java, cela revient à rendre presque standard l'injection de dépendance dans notre plateforme. **Update mercredi 5 août** Roberto Chinnici annonce sur [son blog](#) que finalement la spécification JSR-330 fera bien partie de JEE6. D'autre part, la spécification JSR-299 (ex-WebBeans) a été adapté afin d'utiliser la JSR-330, ce qui est donc une bonne nouvelle. Je ne croyais pas il y a quelques temps qu'il serait possible que la JSR-330 soit adoptée si vite, et finalement l'équipe du JCP a réussi à faire passer cette spécification en 74 jours. Cela retardera la sortie de JEE6 à novembre prochain, au bon moment pour Devoxx 2009. Encore une fois ce sera peut-être l'occasion de célébrer la sortie de la spécification JEE6. J'en profite pour vous parler rapidement du livre *'Beginning Java(tm) EE 6 Platform with Glassfish(tm) 3'* d'Antonio Goncalves : j'avance dans ma lecture et j'en suis au 3/4. Je vous ferai mon retour sans doute en septembre. C'est vraiment un très bon livre avec une qualité didactique, et un ton qui me plaît bien. **Références:** <http://jcp.org/en/jsr/results?id=4944> <http://www.touilleur-express.fr/2009/05/11/jsr-299-crazy-bob/> [Détails sur la JSR330](#)

## Numéro Double spécial été (avec des jeux)



Allez hop, je suis en vacances. Je vous laisse un numéro spécial avant de partir. Tout d'abord j'aimerais reparler des soirées du Paris JUG, puis un retour sur les événements auxquels j'ai participé et enfin un petit bilan perso de cette aventure. **Les soirées du Paris JUG** [Le Paris JUG](#) est une association qui organise chaque deuxième mardi du mois une soirée sur Paris, le tout gratuitement grâce aux sponsors. Lors de ces soirées, en général nous avons 2 présentations, entrecoupées d'un buffet. J'ai rejoins l'équipe d'organisation du Paris JUG il y a quelques mois. Nous sommes 7 personnes en tout. Antonio, David et Zouheir sont les 3 membres de l'association et nous sommes 4 personnes en plus pour les aider et les assister, étant donné l'organisation que cela demande. Il y a chaque mois 200 personnes, ce qui en fait un événement très populaire sur Paris. Par ailleurs il y a 14 autres Java User Group dans toute la France. Je vous encourage à consulter [sur ce billet](#) la liste que j'ai établie en février dernier. Voici les billets publiés depuis l'été dernier de la deuxième saison du Paris JUG: - [septembre 2008 soirée Groovy](#) - [octobre 2008 soirée OSGI](#) - [novembre 2008 Soirée GWT et Restlet](#) - [décembre 2008 soirée JBoss](#) - [janvier 2009 soirée JEE6 et Glassfish](#) - [janvier 2009 le tableau blanc](#) avec une comparaison des différents serveurs d'application. - [février 2009 soirée anniversaire du Paris JUG](#) déjà un an. - [mars 2009 soirée Wicket et Web Sémantique](#) - [avril 2009 soirée Scrum](#) - [mai 2009 soirée JavaRebel et DataGrid/CloudComputing](#) - [juin 2009 soirée Java Avancé](#) - [juillet 2009 Soirée RIA Adobe Flex et JavaFX](#) (billet 2 jamais terminé, désolé) **Devoxx 2008** En 2008 j'ai débuté ma "tournée" par Devoxx, le plus gros salon européen sur Java. 3200 personnes pendant 5 jours à Anvers en Belgique. Pour un budget de 600 Eur (train+hôtel+entrée du salon+bières pour les amis) vous pouvez venir rencontrer les meilleurs speakers sur Java en Europe. Cela fut aussi l'occasion de rencontrer beaucoup de monde. Si vous voulez me croiser, j'y retourne cette année en novembre 2009. - [Devoxx c'est parti](#) premier jour - [Présentation sur les méthodes d'estimation](#) - [Devoxx Romain Guy and Android](#) Devox: Flex, Flash and the Enterprise - [Retour sur ma journée de mercredi](#) et rencontre avec Antoine Borg (Mule Inc.) - [Présentation des services de CloudComputing d'Amazon et d'Elastic Grid](#) - [Devoxx: que contiendra Java 7 ?](#) - [Devoxx: le ralentissement de l'activité](#) interview des gens de Computer Futures. - [Conclusion de mon séjour à Devoxx](#) J'ai présenté aussi un sujet à Devoxx sur les ESB et le monde de la Finance. J'avais retranscrit ma présentation si cela vous intéresse [dans cet article](#). Je compte bien recommencer cette année et présenter un sujet. Voir aussi [tout les articles tagués devoxx](#) sur le Touilleur Express. **XP Day 2009** XP Day c'est la bouffée d'oxygène, le truc à faire si vous vous intéressez aux méthodes Agile et à Scrum. Pour un budget dans les 200 EUR franchement c'est deux journées très riches, très bien organisées par des bénévoles, avec l'occasion de rencontrer entre 300 et 400 personnes. C'était à Paris dans le bois de Vincennes. - [Première journée](#) - [Deuxième journée](#) - [Débat Scrum est-il dangereux ?](#) **Jazoon 2009** Je suis allé en juin 2009 à Zürich en Suisse, afin de participer à Jazoon. C'est un très bon souvenir. La différence par rapport à Devoxx c'est qu'il est plus facile de rencontrer des pointures du monde Java. Je garde un très bon souvenir d'une soirée avec Antonio Goncalves, Emmanuel Bernard, Alexis Moussine-Pouchkine, Roberto Chinnici, etc. A refaire l'an prochain j'espère ! Merci à Thierry pour l'organisation et pour l'accueil des bloggers européens. Voici les différents articles écrits sur Jazoon 2009: - [Jazoon 2009, l'accueil des bloggers](#) - [jazoon 2009 Journée 1](#) - [Journée 1 - Résumé de la présentation de James Gosling](#) - [Groovy, 7 usages patterns](#) - [Spring 3.0, what's new ?](#) - [Gradle - IDEA IntelliJ with Groovy and Grails](#) - [Day 1 when all mics are off](#) **Journée 2** - [When you understand that you're at the good place](#) (or the place to be) - [SUN KeyNotes on JDK 7 and JavaFX 1.2](#) - [Refactoring of Large Software System](#) par Sibylle Peter et Sven Ehrke de Canoo Engineering AG - [XWiki and Wikis 2.0](#) par Vincent Massol, XWiki SAS - [A memcached](#)

[implementation in Java](#) par Bela Ban, JBoss/Red Hat - [Practitioners approach to generate object-relational persistence and services](#) - [Hibernate Envers](#) - Entity Auditing, Quickie par Adam Warski de Level N Consulting - [Gaming with JavaFX](#): Developing the Next Generation of Casual Games par Jose Maria Silveira Neto de SUN Microsystem **Journée 3** - [KeyNotes d'ouverture par Adrian Colyer](#) - [Portlet 2.0](#) - [Hibernate Search](#) - [The Charm of Mockito](#): Test Spies in action par Szczepan Faber - OpenJDK par Dalibor Topic de Sun Microsystem Gmbh Enfin pour terminer un clip que j'ai réalisé avec quelques photos et vidéos de ces 3 jours. On y voit James Gosling, le cofondateur de Java, Juergen Hoeller de SpringSource, une démonstration de JavaFx 1.2, et la remise de prix

des Jazoon Rookie.

**USI 2009**

juillet 2009 OCTO Technology organise chaque année une rencontre pendant 2 jours entre les Geeks et les Boss. Ce fut 2 journées à la fois riches et enrichissantes. On ressort de ces deux jours avec une excellente photo du marché, des technologies, des tendances, des métiers, des réflexions en cours dans l'industrie, dans le SI, bref un événement très intéressant à ne pas manquer. Merci à Nelly pour l'organisation et l'accueil des bloggers. - [Keynote d'ouverture par Joël de Rosnay](#) - [Google pour l'Entreprise puis Amazon Architecture](#) - [Quelques idées issues des grands du web pour remettre en cause vos réflexes d'architecte](#) - [Neal Ford, la claque](#) - [Le management One on One](#) - [L'ergonomie entre forme et fonction, une recette par l'exemple](#) - [L'informatique conviviale, voyage au coeur des grandes organisations](#) - [Wiki contre CMS](#) - [Vaincre la force du plouf](#) - [Google Wave](#) - [Le bilan de ma visite](#) **Bilan de septembre 2008 à juillet 2009** La première saison a été très riche. Je ne sais pas ce que je ferai l'an prochain, j'aimerais retourner à chacun de ces événements car ils m'ont tous apporté un petit quelque chose. Mon status d'indépendant me permet de dégager du temps et de financer ce qui est pour moi de la formation. Et c'est aussi à chaque fois l'occasion de rencontrer du monde. Prenez le temps une fois par an d'aller à un événement sur Java ou sur les méthodes Agile. Si vous pouvez aller à Devvix ou Jazoon, foncez ! Et pensez que le DIFF permet de financer en partie ces événements. Ajoutez à cela les réductions que je vous ai proposé pour Jazoon 2009 et l'USI 2009, il faut y aller ! Sinon, regardez les Java User Group autour de vous. Il y en a 15 en France. A Paris c'est un petit mardi par mois. Et ce sera l'occasion qu'on se rencontre. Gandhi a dit cette très belle phrase : *Commencer par changer en vous, ce que vous voulez changer autour de vous*

## SpringSource racheté par VMware

L'éditeur VMware annonce son intention de racheter la société SpringSource pour un montant de 420 millions de dollars (362 millions de dollars, plus des actifs non levés d'une valeur de 58 millions de dollars), dans [un communiqué de presse](#) publié ce soir. SpringSource est présenté comme l'éditeur de logiciel de solutions "Lean" avec une suite d'outils et de frameworks visant à réduire le gaspillage en simplifiant le travail des développeurs. De la conception à l'exploitation, VMware vise le marché du PaaS (Platform as a Service) estimé à 15 milliards de dollars en 2016 selon une étude Forrester Research citée dans le communiqué de presse (Platform-As-A-Service Market Sizing, July 13, 2009). Ce rachat va surprendre les développeurs Spring de la communauté Java. L'an passé j'ai eu la chance d'interviewer Peter Cooper-Ellis VP Engineering de SpringSource avec Olivier Croisier ([The Coder's Breakfast](#)). Il nous avait présenté une stratégie de croissance externe sans penser peut-être à une sortie à un an après une deuxième levée de fond d'à peine 15 millions de dollars réalisée en 2008. Tout s'accélère, le rachat d'Hyperic en mai dernier par SpringSource, de [G2One Inc dont j'avais parlé l'an dernier](#), un partenariat avec Terracotta, bref ce qui est intéressant c'est que ce rachat offre à VMware un socle applicatif et logiciel solide pour construire la plateforme de cloud-computing de l'entreprise. En effet il y a deux marchés pour les 10 années à venir : d'une part le Cloud-Computing pour les masses (Google AppEngine, Microsoft Azure, Amazon) et d'autre part ce que j'appellerai le Closed-House-Enterprise Computing, ou comment utiliser ses machines mieux, pour moins cher, et plus facilement. VMware pour la virtualisation, SpringSource et Hyperic pour l'abstraction logicielle, l'ensemble se met en marche pour proposer d'ici quelques mois des solutions professionnelles pour la virtualisation de nos



applications.

Concernant SpringSource, Rod Johnson, CEO de SpringSource, rassure la communauté du monde open-source. L'équipe dirigeante de SpringSource ne devrait pas changer. Le nom de SpringSource devrait aussi continuer à exister. Pas de changements aussi je pense du côté du noyau open-source, de Groovy et de Grails. Ce rachat stratégique, le plus important pour la jeune société VMware qui n'existe que depuis 11 ans, est aussi je pense un très bon retour pour les investisseurs privés de SpringSource, qui avaient apportés 15 millions de \$ lors du deuxième tour de table [l'an passé](#). C'est bien le potentiel de l'entreprise qui a été valorisé, ses actifs et les différentes entreprises rachetées par SpringSource ces dernières années. [TechCrunch](#) d'ailleurs vient de publier un billet sur ce sujet il y a quelques minutes en anglais. Quand on sait que Benchmark.com, un des investisseurs dans SpringSource, a aussi misé sur Twitter ou FriendFeed, cela laisse rêveur... En conclusion, sacré coup de chapeau à l'équipe de SpringSource. Il n'y aura pas d'autres rachats de cet ampleur cette année dans notre communauté. A part un rachat de Red Hat par IBM, je ne vois plus trop ce qu'il peut nous arriver d'ici à la fin de l'année. **Références:**

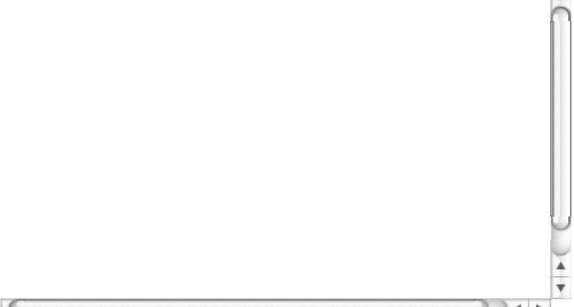
<http://www.vmware.com/company/news/releases/springsource.html>

<http://www.springsource.com/about/management> <http://www.bankingbonuses.com/article/627074/?k=j83s12y12h94s27k02> <http://www.reuters.com/article/mergersNews/idUSN1047900320090810?feedType=RSS&feedName=mergersNews>

<http://www.readwriteweb.com/enterprise/2009/08/vmware-acquires-springsource-for-420-million.php> <http://www.springsource.com/newsevents/springsource-announces-15-million-serie>

## "l'Apéro du Geek 1.0" sur Paris ce soir si ça vous dit

Si vous êtes sur Paris ce mercredi 12 août alors que Femme 1.0/Mari 1.0 sont en vacances, venez prendre un verre vers 19h30 autour du quartier Opéra de Châtelet en plein centre de Paris. On s'est dit : pourquoi ne pas faire comme ces gens qui organisent [les apéros du jeudi](#) ? Quand je dis "on" c'est surtout ma pomme pour l'instant. Pour les détails, suivez sur Twitter et sur [TinyThread](#) pour les détails de l'endroit exact du pot. Les détails suivront demain dans la journée. [Update 13h23] Rendez-vous au [Café Livre](#) au pied de la Tour Saint-Jacques, à Châtelet. En métro, c'est la ligne 4 station "Chatelet" et prendre la sortie 8 dans le métro, rue Saint-Denis. Le café livre est au **10 rue saint-martin** c'est une rue très longue, le café-resto est en face de la Tour Saint-Jacques. On peut y manger. Rendez-vous à partir de 19h15. En RER A c'est aussi faisable, il faudra marcher un petit peu mais c'est très rapide. Cherchez la sortie rue de Rivoli. Attention la rue Saint-Martin est tres longue. Voir aussi <http://www.cafelivre.com/> Sur cette photo de Google Map, le café est situé au niveau du panneau sens unique avec sa toile verte. C'est une rue RATP et il n'y a pas de photos

Google Street View. 

[Agrandir le plan](#)

## Petite retour sur notre premier Apéro de Geek 1.0

En plein mois d'août c'est toujours sympa de se croiser entre passionnés de Java (et de Python). Nous étions une petite dizaine au Café Livre en plein Paris (wifi gratuit, de la place pour discuter). Les sujets sont partis d'une discussion sur qui achète des livres d'informatique, puis un débat sur le livre électronique. Nous avons parlé ensuite de la gestion de la dette technique, comment le vieux

code doit être géré. C'était une bonne discussion, il y a certainement matière à en faire un article, je vous en reparlerai en septembre. Nous avons parlé aussi pas mal d'iPhone et de GPhone. Ainsi il paraît que le Samsung i7500 avec Android a quelques soucis, il reboot régulièrement. Information confirmée d'ailleurs par Korben [sur son blog](#) ce matin. Et vous avez aussi certainement entendu parler de cette histoire d'iPhone qui aurait explosé au visage d'un utilisateur... Le rachat de SpringSource par VMware a ensuite animé les débats. C'est la surprise qui est exprimé par pas mal de monde. On se rappelle du rachat de G2One par SpringSource fin 2008, et là SpringSource racheté par VMware : le signal est fort. Cloud Computing, mais est-ce que la partie Java est stratégique à long terme pour VMware ? Est-ce que le framework lui-même a un intérêt ? Non. C'est plutôt les outils de gestion, l'expertise Tomcat, et le pack d'outils qui intéresse VMware, ainsi que les possibilités offertes par Grails/Groovy. En tous les cas nous en avons bien discuté.



Florent Ramière nous rejoint ensuite, nous sommes maintenant pas mal, avec la présence de Luc Bizeul, David Gageot, Yannick Ameur, Sébastien Douche, Olivier Croisier, Patrick et Thomas Chamas, Jean-Michel Bea, Guillaume Tardif et François Ostyn, et enfin Cyrille Leclerc qui est arrivé un peu plus tard afin d'apporter le quota réglementaire de cravate ;-) Les discussions se sont animées autour des sujets du moment, du Paris JUG et du programme de la rentrée. Il y a eu un cours de géométrie sur la circonférence d'un boc de bière



intéressant par Florent On a parlé de la mission Apollo, saviez-vous qu'il y avait que 2 programmeurs ? que lors de l'alunissage, les chances de réussir n'étaient que de 50% ? Et qu'il restait 15 secondes de propergol pour se poser sur la Lune ? Florent qui a une sacrée culture explique que le radar d'atterrissage avait une mémoire saturée par les echos envoyés par la Terre. Cet ordinateur était vraiment très simple, il dispose d'une mémoire morte de 36 864 mots de 16 bits et d'une mémoire vive de 2 048 mots. Un peu plus tard, nous parlerons aussi de la conception des radars et d'une technique imaginée par les américains pour localiser tous les radars durant la guerre froide... Qui a dit que nous nous cultivions pas ? Pour revenir au Paris JUG, nous aimerions une présentation de qualité de Scala. En effet, c'est l'un des sujets qui revient souvent, et nous aimerions avoir un speaker francophone. Avis aux amateurs : vous pouvez nous contacter via le site du Paris JUG. Un des liens cités par florent : Guido van Rossum nous parle de Scala [sur son blog](#). David Gageot a lancé aussi plusieurs discussions sur Java, et des idées comme par exemple : imaginez Java sans interface, avec des mots "final". Il demande : quel mot clé de Java ne sert à rien ? Jean-Michel Bea cite un mot clé (que j'ai oublié), on parle de "volatile", et la discussion continue sur les Closures. Spring a fortement remis au premier plan le principe des Beans, et un débat entre ceux qui préfèrent l'injection par constructeur et les autres débute. De l'avis général, l'injection par constructeur est mieux. Petite discussion aussi sur Juergen Hoeller que nous sommes plusieurs à avoir eu la chance de rencontrer ([en 2008 pour moi](#)). Ce gars est vraiment un passionné, un bon programmeur et une personne qui fait la valeur du framework Spring. Et il est assez ouvert pour travailler avec Java EE 6, Antonio lui a offert son livre à Jazon d'ailleurs. Si vous avez l'occasion de discuter avec ce gars, vous verrez qu'il a un esprit de synthèse vraiment impressionnant. Un excellent programmeur. Je discute un moment aussi avec Cyrille, et d'ailleurs on est cité tous les deux dans [l'interview qu'Antonio Goncalves](#) a donné à Alexis MP sur le podcast de Glassfish. Alexis dit [à la fin](#) du podcast : "[...]I have to say that I attended a few of those and what I think is really nice about those is the people that come, both the quantity and the quality, there are some really good and hard questions sometimes. I'm sure you know who I'm talking about. Someone who asks really hard questions. And there's this other guy who actually writes blogs about everything that was said and shown during the conference, so it feels like a lively participating group.[...] " A propos de la gestion des fichiers de logs (spéciale dédicace à PrimeWeb) Florent a reparlé d'un outil sur lequel j'avais travaillé l'an dernier : [Splunk](#). C'est un outil d'analyse de fichiers de logs, qui permet de creuser graphiquement dans ses fichiers de logs afin de se créer des alertes personnalisées. Enfin voilà un résumé dans le désordre, je pense que la fois prochaine je ferai peut-être des interviews avec quelques questions préparées. C'était un petit moment sympa, à reprogrammer à l'occasion à la rentrée. En attendant je vous donne rendez-vous dans 2 semaines, il est temps pour moi de prendre des vacances. A bientôt.

## Paris JUG saison 2009 c'est reparti



[www.parisjug.org](http://www.parisjug.org) @parisjug

Le Paris JUG reprend comme chaque deuxième mardi du mois ses soirées. Nous vous proposons le 15 septembre une soirée sur le thème de la qualité logicielle et des outils de la gestion de la dette technique. Dans un premier temps, François le Droff et Romain Pelisse nous parlerons des points à regarder de près lors du développement d'une application Java. Si vous vous demandez ce que représente [la complexité cyclomatique](#), comment identifier les points à revoir dans votre logiciel, alors c'est pour vous. Ensuite, Olivier Gaudin de la société SonarSource nous présentera en 30 minutes le logiciel [Sonar](#). Cet outil est une véritable tour de contrôle de la qualité de votre logiciel. Il permet d'agrèger, formater et présenter des mesures de la qualité de votre programme via une interface Web. L'outil est open-source et la société [SonarSource](#) propose du support, du consulting et de la formation sur l'outil. En 3ème présentation nous aurons le plaisir d'assister à la présentation d'un outil que vous connaissez si vous lisez le Touilleur Express depuis 2 ans : SonarJ. A ne pas confondre avec Sonar donc. Je vous invite à lire [un article que j'ai écrit en juillet 2008](#) suite à un test poussé de SonarJ. La société hello2morrow a lancé une version Open-Source de SonarJ [en début d'année](#), ce qui en fait donc un produit éligible pour être présenté au Paris JUG. En effet, une des idées est plutôt de vous parler des produits open-sources, et de présenter les outils classiques lorsque les éditeurs proposent des cadeaux pour les Juggeurs. L'important étant de rester indépendant bien entendu. Enfin 4ème présentation de 30 minutes : le projet SQUALE (Software QUALity Enhancement) par Fabrice Bellingard, en LGPLv3. Cet outil de collecte et d'analyse de données est un projet initié avec Air France par Qualixo, une entreprise dynamique qui propose d'accompagner les entreprises afin de mettre en place une démarche qualité logicielle. J'en profite aussi pour vous signaler que le **Lyon JUG** organise le 15 septembre une soirée avec [Kalistick](#) et Sonar, donc si vous êtes dans la région, allez vous inscrire d'urgence sur [le site du Lyon JUG](#). Bref comme vous pouvez le voir, le menu de la rentrée est déjà bien complet. A propos de complet d'ailleurs, désolé mais il n'y a plus de places libres pour la soirée de mardi. Je vous raconterai comme d'habitude sur le Touilleur Express ce que vous avez loupé. J'ai envoyé sur le compte twitter du parisjug un message indiquant que les inscriptions étaient ouvertes, et Antonio a ensuite reposté un message quelques heures plus tard : les 195 places sont toutes parties en 2 jours. Et je rappelle que le Paris JUG c'est gratuit. Nous limitons la place pour des raisons de sécurité. Bref suivez le compte @parisjug si vous souhaitez être tenu informé de l'ouverture des inscriptions. Et pour ceux qui viennent, attention cette année vous devez être là à 19h25. Regardez bien les nouvelles règles [sur le site du Paris JUG](#) si vous ne voulez pas perdre votre place. Enfin pour terminer j'espère que vous avez remarqué le nouveau Juggy (la mascotte du Paris JUG) refait en vectoriel ce week-end. J'y ai passé quelques heures, afin de pouvoir aussi réaliser des posters afin de mettre en valeur nos sponsors, sans qui il serait difficile de réaliser ces soirées. Et j'espère que le nouveau Juggy 2009 vous plaît, on a gardé la mascotte originale dessiné par David, avec une petite touche un peu plus 3D. Rendez-vous mardi soir, ou mercredi matin sur le Touilleur Express pour lire le compte-rendu à pas d'heure comme d'habitude. (*Le Paris JUG c'est Antonio Goncalves, David Dewalle et Zouheir Cadi, les fondateurs que l'on appelle "le canal historique". C'est aussi depuis le début 2009, 4 assistants que l'on appelle la Team : Tanguy Bayard, José Paumard, Charles-Alexandre Sabourdin et donc moi-même.*)

## Portails et Portlets 2.0 pour votre architecture web.

**Les Portails d'intégration et plus particulièrement les Portlets sont des briques d'architecture qui offrent des services techniques, mais qui permettent aussi à plusieurs projets de se rejoindre et de travailler ensemble. Plutôt que de parler de SOA, nous verrons que le concept de WOPA (Web Oriented Portal Architecture) est une solution d'architecture pour les applications Webs. Un portail offre un grand nombre de services à une application web** Si vous avez déjà développé des applications webs en Java, vous serez d'accord sur le fait que les besoins fonctionnels et techniques des applications webs sont récurrents. Justement, les Portails d'intégration offrent un ensemble de fonctions standards. Voyons d'abord ce qu'il est possible de faire. Les services offerts par un Portail sont : - la gestion de l'authentification des utilisateurs - l'autorisation afin de déterminer les droits d'accès aux différentes parties de l'application - la personnalisation afin que l'application web s'adapte au profil du visiteur (P13N) - la possibilité de personnaliser son interface web facilement - la gestion de la navigation et sa définition (catégories, pages, sections) - création des pages de contenu (pages statiques, intégration de pages d'un autre serveur) - gestion de l'internationalisation (I18N) - la possibilité d'intégrer du contenu web d'une application distante - le support sans restriction des frameworks Web - la possibilité de mettre en ligne des services dans des Portlets rapidement - la capacité à offrir une version adaptée aux

téléphones portables - l'offre de composants standards Web 2.0 comme Wiki, Blog, Aggregateur, Mash-ups, etc - le support des Gadgets à la Google - la possibilité d'offrir des services de recherche, de gestion de contenu, de gestion électronique de document Un moteur de Portail open-source du monde Java comme [GateIn](#) (JBoss Portal + eXoPortal) ou [Liferay](#) est donc un moyen de gagner du temps en évitant de développer ou de devoir gérer des fonctions standards d'une application Web. L'installation est simple : un WAR déployé sur un serveur J2EE léger, et c'est tout. Si vous disposez d'une application Web Java, le déploiement s'effectue en quelques clics dans le portail. Pour aller plus loin, la création de quelques portlets permet en fait d'exposer les fonctions clés de son application web, afin que le Portail puisse gérer le cycle de vie et l'affichage de votre portlet. Une chose intéressante à noter : vous êtes libre d'utiliser n'importe quel framework Web : [GWT](#), [Wicket](#), [JSF](#) ou autre. De ce fait, la migration d'une application J2EE est donc relativement simple. **Qu'est-ce qu'une Portlet ?** Les [Portlets 2.0](#) ont été normalisées en juin 2008. C'est une spécification assez bas niveau, proche des Servlets. Une Portlet est une enveloppe pour votre application web. Elle dispose de différents états : minimisée, normal, maximisée. Il est aussi possible de basculer d'un mode vue à un mode édition pour la configuration. Du côté du code, en attendant que je vous propose d'autres articles, il y a une distinction entre la phase de rendu et la phase d'action. Le rendu est simplement l'action d'afficher la Portlet. La phase d'action permet de gérer les interactions entre l'utilisateur et l'application web. Enfin il est possible d'émettre des événements afin que d'autres Portlets soient notifiées et puissent interagir avec votre Portlet. C'est relativement simple, l'API Portlet 2.0 propose des annotations qui permettent de marquer vos méthodes. Il y a une petite gymnastique à apprendre afin d'adapter vos pages de rendus par rapport à la navigation. Du côté de la gestion des ressources (js,css,images) il est nécessaire que la Portlet gère le chargement de ces ressources, en raison de la gestion de la sécurité et du cache. Le contenu d'une Portlet enfin peut être caché par le Portail afin d'éviter le rechargement systématique du contenu. Les portlets offrent en fait un moyen standard et normaliser de déployer son application et d'interagir avec d'autres Portlets. Et il est tout à fait envisageable d'écrire des Portlets standards, sans tomber dans le piège d'utiliser une API propriétaire. **Le modèle de programmation des Portlets** (avec l'aide de Julien Viet d'eXo Platform) Le modèle de programmation natif d'une Portlet est similaire à celui des Servlets. Il se fait via le package javax.portlet. Le modèle de programmation est légèrement différent, comme expliqué au paragraphe précédent. La plupart des équipes de développement n'utiliseront pas directement l'API, pour la même raison que l'API Servlet en général n'est pas utilisée directement mais via un framework. En effet, la plupart des frameworks modernes proposent un adaptateur pour Portlet (Portlet Bridge). Avec un modèle de programmation bien pensé, l'exécution en mode Portlet ou directement dans une application web ne présente pas de différences pour le framework. Si vous débutez un projet basé sur des Portlets, il sera important de s'assurer de ce support dans votre framework Web avec un adaptateur pour Portlets. Les plus connus possèdent tous un bridge comme Java Server Faces ([via la JSR 301 Portlet Bridge Specification](#)), Wicket, Struts2 et Tapestry. D'autres framework web tel que Spring MVC ne proposent pas une telle portabilité mais néanmoins fournissent un support pour les Portlets en adaptant le framework à l'API Portlet. J'ai codé une Portlet avec Spring MVC sans soucis, grâce à SpringFuse l'outil de Celerio. Finalement pour de vieilles technologies telles que Struts 1, il existe des bridges mais qui sont gérés à un niveau en dessous avec une émulation de la couche Servlet via une Portlet. En général ce type de bridge a des limitations. **Les Portlets permettent de travailler à plusieurs équipes sur une même application web** Je prends le temps de vous parler de déploiement car le principe des Portlets permet de travailler par modules. Imaginez plusieurs équipes de développement qui travaillent sur un portail web d'entreprise. En se donnant rendez-vous techniquement sur un Portail, vous pouvez donc offrir un moyen pour chaque équipe de travailler de son côté, avec une installation locale de Liferay ou de GateIn, puis de livrer simplement leurs Portlets vers le serveur de production. Un administrateur se chargera alors d'ajouter et de mettre en place les Portlets du premier projet. J'y vois donc un moyen de faciliter le travail de plusieurs équipes. En effet, c'est bien plus souple que d'imaginer livrer un gros WAR constitué de plusieurs projets développés par plusieurs équipes. Imaginez la difficulté pour l'intégration et la mise en production. Un Portail d'intégration est donc un serveur mutualisé sur lequel plusieurs équipes déposent des Portlets, qui constituent au final une application Web complète. Le portail permet de découpler le cycle de vie du portail et des applications qui le composent. En effet vous pourrez facilement intégrer de nouvelles applications dans le portail à tout moment. Un autre intérêt de ce type d'architecture est l'intégration d'applications basées des technologies hétérogènes. Vous pourriez vouloir intégrer dans votre portail une application développée 2 ans auparavant par une filiale de l'entreprise par exemple. Chez le client chez lequel je travaille en ce moment, ce sera le moyen pour trois équipes à Paris, une à Londres et une à New-York de construire une plateforme de Prime Brokerage avec la même charte graphique, une seule authentification, la gestion de plusieurs langues, et surtout vous l'aurez compris, la possibilité pour chacune de ces équipes de travailler sur leurs Portlets et de les tester en local avec un Portail open-source avant le passage en production. Les Portlets ne sont pas révolutionnaires. Par contre, c'est l'utilisation dans votre architecture qui le sera. En effet, si vous avez aujourd'hui plusieurs équipes ayant réalisées chacune une application web, il est possible de repenser le système non pas avec une architecture SOA, mais une architecture Web Portal. Au lieu d'agréger des services différents dans un seul endroit pour construire une seule application web, vous offrez la possibilité à chaque équipe de livrer une tranche verticale qui va de la donnée à la couche web en passant par la couche service. Et quelque part, est-ce qu'une application Web dont l'architecture serait définie par l'interface graphique n'a pas plus de sens qu'une application web construite comme un puzzle de services ? **L'exemple d'Amazon.com** La page d'Amazon.com est construite par 34 équipes différentes. Chacune des parties de la page est une Portlet. C'est assez génial car la mise en production d'une nouvelle version n'est plus un problème : il n'y en a plus ! Terminé la mise en ligne d'une application Web complète. Alors comment font-ils ? Lorsqu'une équipe est confiante sur une nouvelle fonction, une nouvelle version de sa Portlet est activée sur le Portail. En cas de problèmes, l'administrateur peut aussi rapidement désactiver une Portlet, voire même recharger une version antérieure. Quand on pense à Scrum, à de l'incrémental, il paraît évident que ce Portail va vraiment devenir votre ami non ? **Une porte ouverte vers une architecture WoPA (Web Oriented Portal Architecture)** Je pense que les applications d'entreprises sont amenées à s'ouvrir de plus en plus. Nous avons pensé un

moment qu'une architecture orientée service serait la solution. Ces 5 dernières années en effet, nous sommes passés d'architectures client-serveur (J2EE 1.4 avec les EJB2.1, Corba, RPC) à des architectures où la Donnée a repris de l'importance avec les architectures REST. Personne ne semble vraiment savoir ce que représente l'architecture orientée service, à part les éditeurs de logiciels très friand d'acronyme dans leurs plaquettes publicitaires. Une architecture WoPA (copyright N.Martignole) avec un Portail d'intégration, est une application capable de combiner le travail de plusieurs équipes, de plusieurs projets, dans un seul endroit : le Portail Web. Cela remet en avant l'importance des données plutôt que de leur logiciel. Je pense surtout au monde de la Banque, où l'IT est important. Vous n'imaginez pas le nombre d'équipes et de projets chez le client où je travaille en ce moment. L'idée novatrice est aussi que l'interface Web n'est peut-être pas la partie la plus importante de votre application. A l'extrême, il est dommage de proposer un GUI Web et de ne rien proposer à vos clients afin qu'ils viennent chercher les Données directement. Bref ne pas faire de REST devrait être refusé par la DSI dans un schéma d'architecture. Pour expliquer cela, pensez à Twitter.com. Il s'avère que l'API de Twitter est utilisée 10 fois plus que le site web de Twitter. Pourquoi finalement votre application ne serait-elle pas utilisée non pas avec votre GUI Web que vous avez développé, mais via un Portail capable de venir chercher de la Donnée ? Est-ce que votre valeur ajoutée est de faire une application Web ou d'exposer de la Donnée de manière intelligente ? **Conclusion** C'est donc via une solution standard comme un Portail d'intégration et les Portlets qu'il semble plus logique de s'orienter, dès lors que plusieurs équipes exposent des données et des services aux utilisateurs. Je crois à une solution où l'intégration du travail de plusieurs équipes est fait dans un Portail dès lors qu'auparavant, chacune d'elles proposaient une interface Web différente. *Merci à [Julien Viet](#) de la société [eXo Platform](#) pour sa relecture et ses corrections.* **Ressources** [Liferay](#) est un portail open-source lancé en 2000 qui est compatible avec les standards Javas comme la JSR-168 (Portlet 1), JSR-286 (Portlet 2), JSR-170 (JCR) [GateIn](#) est un portail open-source constitué par la fusion de JBoss Portal et d'eXo Portal. Avec une interface très puissante, le support des standards (JSR-168,286,170), la possibilité d'ajouter des Gadgets facilement, c'est l'un des portails les plus en vogue en ce moment. <http://www.jboss.org/gatein> et <http://www.exoplatform.com> **Oracle Weblogic Portal 10g** est la version actuelle du portail d'Oracle. La version 11 devrait sortir à la fin de cette année avec le support de la JSR-286. [Présentation de Dion HINCHCLIFFE](#) sur Web Oriented Architecture [ProgrammableWeb](#) est un annuaire d'API et de Mashup [Démonstration](#) de la fonction Mashup d'Oracle Weblogic Portal La [JSR-301](#) est la Portlet Bridge Specifications for JSF, en cours de finalisation. La [JSR-286](#) est la spécification des Portlets 2.0 à laquelle eXo Platform, Liferay et Oracle ont participé La [JSR-170](#) est la spécification Java Content Repository, implémentée par le moteur d'eXo Portal et de Liferay. (Liens ci-dessous ajoutés suite à la remarque d'Alexis) IBM propose IBM Web Portal <http://www-01.ibm.com/software/fr/info1/websphere/> SUN Microsystems propose GlassFish Web Space Server basé sur Liferay <http://www.sun.com/software/products/webspace/index.xml>. Il y a une bonne présentation de 20 minutes très intéressante [ici](#) Vignette propose Vignette Portal 8 <http://www.vignette.com> J'ai oublié de citer l'implémentation de référence de la JSR-168 et de la JSR-286 : [Apache Pluto](#)

## Devoxx 2009 : plus que 76 jours

 Il y a le Salon de l'Automobile. C'est sympa, vous pouvez y aller avec votre Papa et rêver devant une Peugeot 407. Il y a le salon du mariage, c'est sympa aussi. Vous pouvez y aller avec votre futur(e) époux(se), rêver devant les robes, les buffets et les photographes de mariage. Sinon il y a le salon de l'Agriculture. C'est sympa, vous pouvez rêver devant Marguerite la Vache qui va terminer dans votre assiette, après 3 verres de Bergerac descendu au stand des vigneronns.... Mais franchement il y a mieux pour nous les geeks, c'est Devoxx. [Devoxx](#) c'est le salon de l'Agriculture pour le monde, le salon de l'Automobile pour les belles pépettes qui s'allongent sur les claviers Qwerty, et le salon du Mariage car il faut bien le dire, Java c'est un peu la deuxième femme de la maison. [Devoxx](#) c'est du 16 au 20 novembre 2009. 8ème édition du premier salon européen sur la plateforme Java. L'an passé avec 3200 participants, c'était déjà un gros événement. Inutile de vous dire qu'il n'est même pas envisageable de ne pas y retourner cette année. Et je vais en profiter pour vous donner quelques astuces pour ne pas trop vous retrouver dans un hôtel à 1h du centre des expositions. Côté budget, à 350 EUR pour les 3 jours de conférence, avec un billet de train et 2 nuits à l'hôtel, le budget reste raisonnable dans les 600-800 EUR HT environ. Et croyez-moi, en terme de retour sur investissement ce sera un excellent investissement pour votre entreprise. Voici un petit aperçu des speakers qui ont déjà répondu oui : # Alex Buckley - Computational Theologist # Arjen Poutsma - Spring 3.0 # Cameron Purdy - Oracle VP of Development # Chet Haase - Animation Lover # Don Brown - The web on OSGi # Gregor Hohpe - Enterprise Integration Patterns # Ivar Jacobson - father of components # Jason van Zyl - Founder of maven # Joseph D. Darcy - Project Coin lead # Kevin Nilson - Java Community addict # Mark Reinhold - Project Jigsaw (and much more) # Michael Ernst - JSR 305 EG # Michael Hüttermann - Agile author & JUG leader # Paul Brown - SOA Author # Roger Kitain - JSF co-spec lead # Ross Mason - Founder Mule # Timothy Perrett - Scala Lift committer Cette année Ross Mason fera le déplacement. Et je note que SpringSource sera aussi plus présent que l'an passé, où un événement Spring se déroulait en Floride en même temps que Devoxx 08. Il y aura aussi toutes les personnes de la communauté francophone de Java. En effet, Devoxx finalement c'est aussi l'endroit où il faut venir si vous souhaitez prendre une leçon de Baby-Foot sur le stand de RedHat avec un Emmanuel Bernard déchainé, voir un Vincent Massol entrain de prendre son petit-déjeuner dans un hôtel Ibis avec un portable sur la table, rencontrer enfin cet Antonio Goncalvès dont on parle si souvent, et j'en oublie... Rendez-vous donc dans 2 mois !

## Compte-rendu de la soirée Qualité du Logiciel au Paris JUG, le 15 septembre 2009

Début de la nouvelle saison du Paris JUG ce mardi 15 septembre avec une soirée sur la qualité du logiciel. La soirée était divisée en 4 sessions de 30 minutes. Après une présentation de la démarche qualité et des différents types d'outils open-sources, nous avons eu 3 présentations sur Sonar, SonarJ et Squala. Une superbe soirée avec 200 personnes dans une salle comble. **1)Contrôle Qualité avec des outils open-source** par François le Droff et Romain Pelisse Romain Pelisse et François le Droff débute par une bonne présentation afin passer en revue les outils open-sources. [Romain](#) est contributeur du projet [PMD](#) et leader du projet [XRadar](#). [François](#) est contributeur sur XRadar. Tout d'abord, quelques mots sur la qualité. Mesurer la qualité de son logiciel, c'est d'abord contrôler sa complexité. Lorsque votre logiciel ne fonctionne plus, vous êtes victime d'une tragédie, pas d'une statistique. J'en profite pour vous parler d'un document de 1995 qui ne fait que 8 pages et qui résume plus globalement le Chaos du développement logiciel : [The Chaos Report](#) par the Standish Group Report. Bref la qualité c'est fun, et l'on peut même parler de **Développement Piloté par la Qualité**. Qui est responsable de la qualité ? Tout le monde. Du développeur au Chef de projet en passant par l'Architecte. Quand ? Tout le temps. La Qualité n'est pas une option que l'on ajoute à posteriori. C'est une démarche qui doit faire partie du cycle de compilation et de déploiement de votre logiciel. Comment mettre en place des outils d'analyse de la qualité ? L'intégration continue est un bon endroit pour mettre en place ces outils. Ce qui est intéressant par contre dans un produit comme SonarJ que l'on verra ensuite, c'est qu'il est possible de mettre en place ces outils dans l'environnement de développement. Un des points forts d'IDEA Intellij c'est l'intégration d'un moteur de qualité qui est présent au moment de l'écriture de votre code. C'est bien souvent quelque chose que les développeurs découvrent, alors que finalement cela devrait être obligatoire dans un IDE en 2009. Il existe différents types d'outils pour l'analyse du code Java. Des outils d'analyse statique, du bytecode, des archives Java et d'autres comme l'analyse de la couverture de tests de votre code. [PMD](#) est un moteur d'analyse du code source Java, intégré avec un grand nombre d'IDE. Il permet d'identifier les bugs potentiels, le code mort, le code mal optimisé, le tout avec environ 250 règles. C'est un appareil de radiographie qui vous permet d'analyser du code de manière statique. Mon favori c'est aussi [FindBugs](#), un projet qui ne paye pas de mine mais qui génère des statistiques vraiment intéressantes pour trouver des bugs dans votre code. Les outils d'analyse du bytecode sont un peu moins précis mais permettent de détecter des cas que l'analyse de code source ne peut pas trouver. L'analyse du bytecode permet de générer des métriques de package sur le nombre de classes concrètes, abstraites et publiques. Il permet de détecter les dépendances cycliques, de calculer le degré d'abstraction ou d'instabilité du package. Je pense que le produit [XDepend](#) dont je vous ai parlé [il y a quelques mois](#), correspond à cette description. Il y a aussi des outils d'analyse des dépendances. Pratique afin de voir le sac de noeuds de votre projet Maven. Vous pouvez d'ailleurs retrouver une liste des outils d'analyse statique [sur Wikipedia](#) si vous cherchez un outil pour du C/C++/Java ou .NET. En conclusion pour cette première présentation, c'était un sans faute. Rythme parfait, et après 30 minutes nous étions en plein dans le sujet. La présentation complète est en ligne :

[Soirée Qualité Logicielle Paris JUG : Tour d'horizon des outils open source](#)

View more [documents](#) from [Francois Le Droff](#).

**2)Sonar** par Olivier Gaudin Olivier est l'un cofondateur de la société [SonarSource](#), éditeur du logiciel open-source [Sonar](#). Sa présentation était construite autour de la définition des 7 péchés capitaux du développeur, et de la réponse de l'outil Sonar. Sonar permet de collecter, d'analyser et de générer des rapports de la qualité pour la plateforme Java. Intégré à Maven, l'outil permet de mettre en place et de suivre l'évolution dans le temps d'indicateurs de qualité. Olivier commence [sa présentation](#) avec la phrase suivante : "A well-written program is a program where the cost of implementing a feature is constant throughout the program's lifetime." de Itay Maman. Ce sera d'ailleurs la même idée qui sera reprise ensuite avec la présentation de SonarJ. L'objectif de Sonar est de proposer un moyen de maîtriser l'érosion technique et de conserver un indicateur de la qualité, capable de remonter dans le temps. Les métriques d'analyse de la complexité d'un logiciel ne sont pas neuves. La complexité cyclomatique, les standards de programmation, les tests unitaires, en fait tout ceci a bientôt 20 ans. Sonar propose de mesurer, de visualiser, d'agir et de suivre les améliorations dans le temps. Avant de commencer avec Sonar, Olivier appuie le fait qu'il soit indispensable de mettre en place un gestionnaire de code source, un environnement d'intégration continue, un moteur de construction du logiciel maîtrisé comme Maven et un gestionnaire d'incident comme Jira. Quelque part, la gestion des incidents devrait faire partie de Sonar pour moi. Ce serait aussi le moyen de mesurer les progrès réalisés non ? [Les 7 péchés capitaux](#) s'appliquent très bien à ce que nous faisons, nous, les développeurs. Que ce soit l'avarice, l'envie ou la gourmandise, en effet on pense aux non-respect des standards de programmation, à la

duplication de code, au manque de tests unitaires, aux mauvais commentaires, aux bugs potentiels, à la complexité du code, bref à tout ce qu'un développeur peut produire. Pour répondre à cela, Sonar propose un tableau de bord avec une vue synthétique plutôt bien pensée. Un indicateur vous donne ainsi le montant de votre dette technique en dollars. Ce montant est calculé si votre équipe devait fixer l'ensemble des problèmes identifiés, avec un prix unitaire par élément. Cela donne une idée du prix de la dette technique de votre logiciel. Olivier termine par une démonstration de l'interface de Sonar. L'interface est plutôt bien pensée. Les indicateurs radars, les courbes d'évolution, tout ceci nous donne un sentiment de qualité sur le logiciel. Je pense que Sonar est un produit rapide et simple à mettre en oeuvre, et qu'il est dommage de s'en passer si votre projet est déjà en place avec Maven par exemple. La version 2.0 est prévue pour la fin de cette année

**3) SonarJ** Frédéric Brachfeld présente ensuite le produit SonarJ, dont [la version 5.0](#) vient justement de sortir. SonarJ est un produit dont j'ai déjà parlé plusieurs fois sur le Touilleur Express ([ici](#) et [surtout là](#)). C'est un produit qui se résume par une image : une ceinture de sécurité. Il permet de définir des règles d'architecture en se reposant sur le découpage par package Java, mais surtout via un plugin dans Eclipse, de s'assurer que ces règles sont respectées lors de l'édition du code et de la création de nouvelles classes. La démonstration s'est terminée un peu difficilement, ce que je regrette pour Frédéric. Moi j'ai vu ce produit, et j'ai vu une présentation de Frédéric chez mon ancien employeur qui m'avait bluffé. Vraiment l'effet "Wahouu" pendant la démonstration. Ce qu'il faut bien comprendre, c'est que c'est un moyen de rendre plus intelligent votre code source, pour un effort relativement simple. C'est aussi un outil qui vous permet de réorganiser votre code source. J'entendais des gens critiquer la séance de découpage des packages et dire "ouais dans la vraie vie les packages ne sont jamais si bien rangés"... ce qui est tout à fait vrai. Mais justement, cet outil permet de reprendre la main sur un sac de noeuds, il permet de reranger le code. Et surtout, il permet de s'assurer que le code ne redevient pas n'importe quoi en quelques semaines. J'en parle comme si j'avais des actions chez SonarJ, ce qui n'est pas le cas. J'en parle car j'étais déçu que la démonstration ne se passe pas comme prévu. J'aurais aimé cet effet "Wahouu...". Testez le produit, c'est le seul qui apporte de la Qualité non pas à posteriori mais à priori. Il vous aide au moment d'écrire votre code, pas une fois qu'il est commité dans Subversion. [Version open-source](#) de SonarJ.

**4) Squal**e Enfin pour terminer, et là j'ai trouvé que la présentation venait conclure parfaitement la soirée, nous avons eu une présentation de Squal (Software QUALity Enhancement) par [Fabrice Bellinard](#). La démarche du produit est de proposer une offre différente des produits existants. Squal analyse du code Java, du C, du C++ et du Cobol. Le point fort que Fabrice présente dès le début, c'est l'élaboration de modèles évolués. Je m'explique. Plutôt que de définir des métriques à effet de seuil, le moteur travaille avec des métriques linéaires. Là où un produit affiche un carré vert si la méthode doSomething fait 30 lignes, et un carré rouge si celle-ci fait 31 lignes, Squal propose une approche différente en lissant justement cet effet. Cela permet de rendre plus pertinent l'analyse et de donner des résultats plus précis afin de suivre la qualité de votre logiciel. [Squal](#) s'adresse au développeur afin qu'il identifie les zones à risque. Il s'adresse au Chef de projet qui souhaite par exemple contractualiser un niveau de qualité, et mettre en place des exigences en terme de tests afin de travailler avec un prestataire extérieur. Enfin il s'adresse aux responsables du produit, en offrant des indicateurs de plus haut niveau, afin de donner une idée de la complexité d'une application. Les métriques dans le développement logiciel sont certainement aussi nombreuses que le nombre de langage existant. Mais l'un des indicateurs qui marche bien, c'est le "WTF per minut". Lorsque vous lisez du code écrit par quelqu'un d'autre, combien de fois dites-vous à voix haute : "mais c'est quoi ce b... ?". Plus sérieusement, Fabrice explique par contre que la recherche a validé avec des modèles des axiomes comme "la probabilité d'avoir un bug dans une méthode est proportionnel à la complexité cyclomatique de celle-ci". Et c'est une des caractéristiques importantes de Squal. Squal est aussi développé en collaboration avec l'Inria-Futurs de Lille et LIASD-Paris 8. Les modèles de qualité sont donc validés par des chercheurs. De plus, l'entreprise Qualixo qui développe le produit, travaille avec PSA et Air France afin d'utiliser de réels projets pour affiner les modèles. C'est aussi une caractéristique importante du produit. Après cette présentation, Fabrice a effectué une démonstration du produit. L'approche de l'analyse s'effectue sur 4 niveaux. Tout d'abord la collecte des données brutes, ensuite l'analyse des pratiques (comme les règles de nommage), le respect de critères ensuite et enfin des facteurs qualitatifs comme l'estimation du coût de maintenance. L'interface Web de Squal permet en fait de naviguer dans son projet. Des indicateurs de ROI, en plus des indicateurs classiques de complexité du code, offrent aussi une vue plus pour les managers et les décideurs. Ainsi j'imagine qu'il sera possible de décider de ne pas refactorer un module de code, car sa dette technique est trop élevée. L'interface gagnerait à être un peu plus jolie, et un ergonome pourrait aussi aider à classer l'information et à proposer un modèle de navigation. Les données par contre et le principe des modèles est le point fort du logiciel [Squal](#). Une bonne présentation qui m'a intéressé, surtout lorsque l'on pense qu'il permet de supprimer cet effet de seuil "bien/pas bien" qui n'est pas très constructif sur les autres outils. **Conclusion** Pour terminer finalement je repense à la première présentation, où faire de la qualité a été expliqué très simplement. C'est pour votre bien, c'est bon et c'est indispensable. Mesurer permet de contrôler, et d'évaluer aussi la quantité de travail en plus de la qualité. Il faut faire de la qualité, mettre en place ces outils sur vos projets. **Post-Scriptum:**

1) Alexandre Boutin sur [son blog](#) pointe une conférence Agile 2009 sur le thème "[Tout d'abord, tuer toutes les métriques](#)" en anglais. Je cite :

"L'approche proposée est assez intéressante puisqu'il s'agit de regarder les métriques sous un nouvel angle, celui du comportement. Pour chaque métrique il est intéressant de se poser les 3 questions suivantes : En quoi cette métrique génère un comportement positif ? En quoi elle génère un comportement négatif ? Quel type de comportement permet de contourner la métrique ? Les orateurs ont également insisté sur le fait que le comportement ( »behaviour ») est émergent et changeant, donc ce qui est vrai à un instant ne l'est plus quelques mois après ... et donc il faut être prêt à changer de métrique. [...]"

2) Le [Spag](#) est une unité de mesure de la dette technique, présenté par James Shore. Encore merci au blog d'Alexandre qui est très complet.

## Devoxx : astuces et bons plans

Quelques informations si vous pouvez aller cette année à Devoxx, la plus grosse conférence sur la plate-forme Java en Europe. L'événement a lieu comme chaque année à Anvers en Belgique. A 2h de train de Paris, c'est une destination très bien desservie. L'exposition se déroule au nord de la ville au Metropolis, dans un grand multiplexe. En Taxi, compter 12-16 EUR de la Gare Centrale. Sinon le plus simple est de prendre le tram qui circule jusqu'à 22h30 environ le soir. Au delà, c'est Taxi obligatoirement. Pour aller le matin, prendre donc la ligne 6 direction Luchtbal. La ligne est sous-terre dans le centre-ville, comme le métro parisien, puis aérienne ensuite. Le Metropolis est le terminus au nord de cette ligne, vous ne pouvez donc pas vous tromper. Un passe de 3 jours ou un carnet de 10 tickets fera l'affaire. Sur place autour du Metropolis, il est possible le soir de dîner. Il y a quelques restaurants, une bonne pizzeria. Cependant l'intérêt de Devoxx étant d'en profiter un maximum, vous pouvez assister aux soirées jusqu'à 22h, et donc en profiter pour rencontrer pas mal de monde. L'an passé j'étais en compagnie de Florent Ramière de Jaxio. Nous sommes allés à une session de présentation de Mule, à 21h. Nous n'étions que 10 personnes, avec comme présentateur monsieur Antoine Borg en personne, l'un des contributeurs de Mule. Du coup cette présentation s'est transformée en discussion autour de Mule, des ESB, d'Apache Camel... jusque dans le Tram qui nous a ramené dans le centre ville. Alors voyons du côté des Hôtels. L'an passé j'étais à l'Ibis Antwerpen Centrum. J'y ai croisé l'équipe d'XWiki un matin, ainsi que les gens de RedHat. La ligne de bus 23 situé à 300 m au sud de l'hôtel permet de se rendre au Metropolis. L'an passé j'ai préféré prendre un taxi. Compter 25mn jusqu'au Metropolis. Evitez les hôtels situés à côté du Metropolis. C'est un peu comme prendre un Formule 1 à Rungis, la zone est assez pourrie le soir, pour en avoir discuté avec d'autres français qui étaient installés dans un drôle d'hôtel avec des dames, enfin des monsieurs, enfin vous voyez quoi. Vive le Brésil. Cette année j'ai réservé à l'hôtel Agora en centre ville, situé à côté de la Gare d'Anvers. Bon courage pour faire la réservation sur le site en flamand, mais on s'en sort avec Google Translate. 270 EUR pour 4 nuits. En effet je compte passer la semaine complète, les 2 jours d'universités le lundi et le mardi, puis les 3 jours de conférence ensuite le mercredi, jeudi et vendredi. Je ne vous raconte pas le nombre d'articles qu'il va falloir écrire pour vous chaque jour. Mais c'est fun :-). Regardez aussi la page "[Venue](#)" sur le site de Devoxx qui donne quelques informations intéressantes. Vous pouvez aussi suivre sur Twitter les discussions avec le tag #devoxx. Voici les francophones que vous connaissez qui vont à Devoxx : - Antonio Goncalves présentera [la plateforme JEE6](#) à l'Université. - Emmanuel Bernard présentera [Hibernate Search in Action](#). - Jean-Francois Arcand de SUN Microsystems présentera [Comet](#). Il devrait passer prochainement au Paris JUG. - Patric Chanazon de Google, cofondateur de l'OSSGTP, présentera [The Google Java App Engine](#). - Romain Guy de Google présentera [les dernières nouveautés](#) autour de la plate-forme Android. La liste des speakers est tout simplement impressionnante, avec les meilleurs présentateurs autour de la plate-forme Java. Alors ne vous posez pas trop de questions, prenez un billet de train, un sac de couchage, et on se retrouve là-bas !

## CITCON 2009 Paris

[CITCON](#) est une conférence en anglais sur l'intégration continue qui s'est déroulée à Paris ce week-end. Nous étions environ 120 personnes vendredi soir et samedi, avec une bonne représentation de nos voisins européens, peut-être 40% des participants. Vendredi soir la conférence a commencé par un long tour de salle afin que chacun puisse se présenter. Puis nous avons ensuite préparé le contenu de celle-ci. En effet, le format est de type open-space. Il n'y a pas de contenu préparé à l'avance. Chacun propose un sujet, qui est inscrit sur un post-it et placé sur un tableau blanc. Vous pouvez ensuite voter pour le sujet qui vous intéresse, afin d'en débattre. Avec une organisation vraiment top (merci à Eric Lefevre-Ardant, Jean-Michel Bea et un autre Jean-Michel) c'est vraiment un plaisir que d'être ici. J'ai retrouvé quelques habitués, mais aussi fait de nouvelles rencontres comme Marc Rambert de [Kalistick](#) ou François Goldgewicht d'[Aeon Consulting](#) (et auteur de Rss Reader sur iPhone), avec lequel j'ai pas mal discuté. Par contre la majorité de francophone a peut-être un peu exclu nos amis anglophones, comme lors des pauses et du déjeuner. Juste un sentiment. J'ai assisté à plusieurs sessions. Tout d'abord une séance sur "Done and Testing". L'objectif est de débattre sur la notion du "Terminé". Puis ensuite j'ai assisté à une conférence sur la mise en place d'un "pipeline" pour l'intégration continue. Une session qui ne m'a pas tellement plus, lorsque finalement celle-ci s'est orientée vers une présentation de Cruise de ThoughtWorks. Comme vous êtes libre d'entrer et de sortir des sessions, j'ai rejoins la session sur les Mocks, animée entre autre par David Gageot de Tech4Quant. Je reviendrai sur ces conférences un peu plus tard. Pendant l'heure du déjeuner, nous avons parlé Java, iPhone, comment travailler avec un apporteur d'affaire lorsque vous êtes indépendant. Nous avons eu aussi de bonnes discussions sur Java et nos habitudes. Par exemple, pensez-vous que les Interfaces en Java soient une chose à utiliser systématiquement ? Certains répondent oui, car cela permet de tester avec des frameworks. Je fais remarquer qu'il est dommage de construire un logiciel et de n'utiliser des Interfaces qu'en raison des outils de tests, si au final l'application ne s'en sert pas. Nous avons aussi combattu l'effet "*si jamais j'en ai besoin au cas où*". Avec cette philosophie, je prendrai des antibiotiques chaque jour "au cas où"... Bref il faut en faire un article, on vous en reparlera sur le Touilleur Express. Cet après-midi je vais me balader dans des sessions sur les tests d'acceptance, afin de découvrir un peu ce domaine. Et je vous raconterai tout cela un peu plus tard. Stay tuned

## CITCON 2009 Done and Testing

Première séance de [CITCON](#), je regarde le grand tableau blanc. Les postits de chacun sont maintenant tous cochés, ce qui permet de construire l'agenda de la journée.



(Crédit photo : [Cirilo en Natasha](#)) Je décide de commencer par un

débat sur "Testing and Done !" Je ne vais pas retranscrire en français toute la discussion, simplement vous donner quelques points discutés par l'ensemble de la salle pendant une heure. Tout d'abord, si votre responsable vous laisse coder sans limite, à quel moment décidez-vous de vous arrêter ? Comment décidez-vous que votre logiciel, ou votre développement est terminé ? Chacun propose sa définition du "done" (terminé). Un développeur explique que pour lui, c'est lorsque le code source est commité sur le gestionnaire de code source. Un autre répond alors que pour lui, c'est lorsque l'intégration continue a confirmé que son code compilait. Et un autre de répondre, que c'est lorsqu'en plus la version a été installé sur la machine de test... Bref chacun a une vision différente de la notion du Done. Le débat s'annonce passionnant. Une discussion sur la couverture des tests unitaires s'engage alors. Par expérience, il semble que 80% de couverture des tests unitaires soit un bon chiffre. Cela me rappelle mardi soir dernier, lors de la présentation de Sonar, qu'il y a quelque chose à creuser. Je me souviens d'un écran avec une liste des projets open-source audités par Sonar, et j'avais été étonné de voir que les 25 premiers projets audités se tenaient dans un mouchoir de poche. Après quelques recherches, j'ai trouvé ce que je cherchais. Sur le site [Nemo](#) de Sonar vous pouvez visiter des audits de différents projets open-source. Prenez le temps de regarder la couverture en tests unitaires des projets, et vous verrez qu'il est utopique de penser couvrir 100% de votre code (ligne ou branche de code). Bref 100% de couverture n'a rien à voir avec la notion du Done pour ce qui est purement tests unitaires. D'autres pensent qu'il est indispensable d'avoir des tests d'acceptance, qui lorsqu'ils passent tous, signent pour vous cette idée du "Done". Est-ce à dire que 100% de tests d'acceptance validés, font qu'une tâche est validée ? Comme j'ai assisté l'après-midi à une présentation sur les tests d'acceptance, je reviendrai sur ce sujet. Il y a donc des indicateurs techniques (Cobertura) et des indicateurs marketings (Fitness) mais finalement, tout ceci ne répond pas au problème. La notion de "shippable" est ensuite discutée, amenée je pense par quelques Scrumistes dans la salle. Un débat s'engage entre ceux qui doivent sortir un produit à date fixe, et qui n'ont donc pas le choix sur la date mais sur la liste des fonctions, et ceux qui doivent livrer un ensemble de fonction, sans que la date de sortie ne soit importante. Dire qu'un projet est terminé, c'est aussi avoir l'ordre de sortir ce produit sur le marché. Et tout le monde s'accorde à dire que parfois, des produits sortent alors que l'on sait qu'il y a des défauts... Mais tant pis ! Il semble donc très difficile d'avoir un produit réellement terminé, sans aucuns bugs. D'où l'existence de support dans notre industrie, et des patches correctifs. J'écoute tout le monde, et pendant la discussion me vient une idée. Est-ce que finalement, la notion de "Done" que l'on essaye de calquer sur une tâche ou un logiciel, ne devrait pas être pensée de la façon suivante : **"Done s'applique à la personne qui réalise, pas à ce qu'elle fait"**. Moi en tant que développeur, je peux me marquer comme "DONE" sur le Kanban car j'ai réalisé ma mission. Je passe la main ensuite au testeur, puis à l'intégrateur, et enfin l'équipe de production. Quelque part, j'imagine une chaîne de construction où chaque tâche est séquentielle. Et donc le "Done" dont nous parlons, le "My Task is Done" n'existe pas. Ce serait plutôt "My Job is over/done". Qu'en pensez-vous ? Ca sent le Lean, je sais. Nous réfléchissons ensuite sur l'utilité des tests. Eric "Bob" Mignot propose l'idée que les tests permettent de réduire la peur, l'inquiétude du Product Owner, en affichant des indicateurs de qualité mesurable dans le temps. Il est donc indispensable de travailler sur la mise en place de tests d'acceptance. Et il est important d'ajouter des tests que le chef de produit comprend. Pour améliorer la communication et décider à quel moment, une tâche est "Done". C'est le client qui décide que sa commande est comme il l'a souhaité. Un développeur s'insurge contre le marketing qui lui demande de sortir un produit, alors qu'il voudrait encore ajouter des tests. Il explique que pour lui, la notion du Done c'est lorsque le chef de produit décide de venir chercher le logiciel sur lequel il travaille pour le vendre, et qu'il ne décide pas vraiment. Quelque part, je pense qu'il ne faut pas être trop [manichéen](#), et qu'il est important d'avoir un consensus pour décider de la sortie d'un produit, ou de la fin d'un projet. Pour aller plus loin je vous laisse avec quelques pointeurs sur le sujet. Robert C.Martin (dit "Uncle Bob") a parlé de la Quintessence, le 5e élément du manifeste Agile lors de la conférence Agile 2008 à Toronto. Il propose de privilégier l'Artisanat par rapport à la Réalisation : Craftmanship over [Crap](#) qui est devenu ensuite Craftmanship over Execution dans [cet article](#). *Most software development teams execute, but they don't take care. We value execution, but we value craftmanship more.* Dans la salle, beaucoup de personnes comme [Antony Marciano](#) de Skills Matter, Gilles Mantel de Valtech, Eric Mignot de Pyxis, et surtout [Gojko Adzic](#) qui est passionnant, mais déroutant car lorsqu'il vous parle, on dirait que vous allez passer un mauvais quart d'heure. Je l'ai suivi dans une autre session l'après-midi, et ce gars est vraiment intéressant. Auteur de [2 livres](#) sur les tests d'acceptance, il connaît bien le sujet. Bon finalement je vous ai livré 960 mots et 1h30 de travail, à ce rythme vous aurez un article par jour, pas plus. A suivre !

## CITCON 2009 Mock mock and mock

Après [une session sur la notion de "Done"](#) sympathique, je suis allé écouter une discussion sur la

mise en place d'un Pipeline d'intégration continue. Finalement la discussion s'est transformée en présentation du logiciel "[Cruise](#)" par ThoughtWorks. Et comme je n'ai pas de temps à perdre à regarder une présentation commerciale, je suis allé rejoindre un débat sur les mocks dans une salle voisine. Sur scène, [David Gageot](#) de Tech4Quant et [Steve Freeman](#). David est bien connu de la communauté Java sur Paris, et c'est une étincelle qui a le mérite de faire avancer nos habitudes. Steve Freeman est anglais, c'est l'un des auteurs du livre "[Mock Objects and Test Driven Development](#)". C'est aussi un speaker reconnu qui a présenté 2 sujets à Agile 2009. Son expérience dans notre industrie est intéressante. Il explique ainsi que l'excitation récente sur TDD est paradoxale alors qu'en fait cette pratique existe depuis longtemps dans notre industrie... On en parle plus mais il faut bien comprendre que dans certaines industries comme l'électronique par exemple, la conception pilotée par les tests est le seul moyen de fonctionner. Je n'ai pas noté tout ce qui s'est dit pendant cette séance, mais il était intéressant de voir Steve plus ou moins découvrir le code de David, et l'utilisation de Mockito. Avant que je n'aille plus loin : [Mockito rocks](#) ! Prenez easyMock et Mockito, le premier donne mal au ventre, vous fait perdre vos cheveux. Le deuxième est plus digeste, simple et diablement plus intelligent que son copain. Sur le code présenté, nous essayons d'identifier des bonnes pratiques des Mocks. David recommande de ne pas utiliser de méthodes private dans des tests unitaires. Cela n'améliore pas la lisibilité. D'ailleurs si vous n'arrivez pas à lire ce que fait un test, c'est que la méthode testée est trop compliquée. David j'ai une idée : pourquoi ne pas développer un petit plugin pour Eclipse/IDEA IntelliJ qui afficherait la complexité cyclomatique dans la classe de test unitaire, au niveau de la méthode du test unitaire ? Cela donnerait un indicateur de ce que l'on est entrain de tester, et donc une idée de l'effort à fournir... Steve appuie aussi l'importance du nom de votre méthode de test. Même si l'anglais n'est pas votre langue native, il est important de trouver un certain phrasé sur les noms des méthodes des tests unitaires. Nous avons une discussion sur les pratiques à mettre en place pour mocker les DAO. [Gojko Adzic](#) rappelle qu'un pattern comme [Repository](#) présenté par Martin Fowler, permet de mettre une abstraction entre le moyen de stockage et les données. Ecrire un Repository c'est très simple. Prenez [cet article en C#](#) qui est assez complet, et vous aurez compris le principe. Voir aussi [cet article](#) inspiré par Eric Evans, le père du Domain Driven Design, qui est passé au Paris JUG l'été dernier, et [cet article](#) de Christian Bauer qui critique le pattern Repository si vous voulez avoir un autre son de cloche. Après pas mal de discussions animées, que je n'ai pas noté (pour une fois) nous avons fait une pause déjeuner. Je vous laisse regarder les articles, je me prépare pour l'article suivant sur les tests d'acceptance. Je reviens dans 5 minutes...

## CITCON 2009 User Acceptance Test

Je suis magiquement devenu plus intelligent. Pour cela j'ai passé une heure de brainstorming avec quelques brutes des tests d'acceptance. Le port du casque est obligatoire pendant la lecture de cet article, où je me suis laché. Alors présentons quelques acteurs tout d'abord. Celui qui prend la parole avec un accent russe, c'est Gojko Adzic. Un gars très brillant qui m'a donné envie de lire son livre : "[Bridging the Communication Gap, Specification by Example and Agile Acceptance Testing](#)" auquel Eric Lefèvre-Ardant a participé. A côté de lui, un accent purement londonien, c'est [Antony Marcano](#) de Skills Matter. Ces gars sont forts. Le reste des quelques 30 personnes est constitué de français, de quelques personnes de JetBrains et de Pulse, ainsi que d'autres personnes de ThoughtWorks. L'objet du débat est de discuter pourquoi les tests d'acceptance échouent, et quelles sont les choses que l'on peut améliorer afin d'éviter cela. Gojko liste quelques points qu'il a identifiés lorsqu'il travaillait sur son livre, qui font que les tests d'acceptance ne sont pas mis en place sur un projet. 1) No collaboration 2) Focusing on "how" and not on the "what" 3) Test unusable as live document 4) Expecting Acceptance Tests to be a full regression suite 5) Focusing on tools A cela, nous ajouterons après une discussion un point 6: 6) When acceptance tests are not considered as a value added activity. Par rapport au point 1, c'est assez facile à expliquer. Il est impossible de mettre en place des tests d'acceptance correctement, si votre client ou votre donneur d'ordre n'est pas impliqué dans l'écriture de ceux-ci. Pour rappel, pour ceux du fond qui ne suivent pas, un test d'acceptance est une histoire écrite sur un document électronique, que l'on peut transformer en test exécutable. Si par exemple j'écris: "En tant qu'administrateur je veux créer un mot clé Devxxx sur la section Catégorie en remplissant une zone de saisie avec Devxxx et en cliquant sur le bouton valider." Un outil comme [FitNesse](#) permet ainsi d'écrire ce type de test et de les exécuter facilement en quelques minutes. Gojko résume donc que les tests d'acceptance sont des spécifications. Si vous avez du mal à mettre en place une bonne collaboration, expliquez que vous travaillez sur des spécifications, c'est simple. Par rapport au point 2, nos discussions nous amènent à dire qu'il est important de travailler sur ce que l'on veut tester, pas sur comment. Pour cela, Gojko illustre son point avec l'exemple amusant de la fabrication du F-15 par l'armée américaine. Le militaire explique à un ingénieur les souhaits de l'armée américaine : - "Nous voulons un avion capable de voler à 2.5 Mach pour 10 millions de dollars". Les industriels débutent les recherches, mais il s'avère rapidement impossible de construire un tel avion à cette époque, capable d'aller à 2.5 Mach. Et là le projet aurait pu s'arrêter. Mais un gars plus malin que les autres demande alors - Pourquoi voulez-vous absolument un appareil capable de voler à 2.5 Mach ? - Et bien pour qu'il vole plus vite que le Mig, qui vole à mach 2 - Il faut 25mn pour passer de mach 1.5 à mach 2.5, est-ce c'est acceptable pour vous ? - Bien sûr que non ! Nous, ce que l'on veut, c'est un avion capable de bombarder, puis de repartir le plus vite possible sans se faire descendre par les Migs - Votre besoin est donc d'avoir un appareil capable de battre en dogfight ce Mig alors ? - Euh... oui... " Après cette discussion, l'ingénieur leur propose de réaliser un avion pour 9 millions de dollars, qui ne volera qu'à 1.8 Mach mais qui sera bien plus agile que le Mig de l'époque. Il pourra donc gagner les combats aériens. Et c'est sa compagnie qui a remporté le marché... Le besoin était d'éviter de se faire descendre, pas de voler le plus vite possible. Tout ceci pour vous dire que lorsque les clients vous demandent quelque chose, il faut absolument creuser leur besoin et il ne faut surtout pas commencer à coder sans avoir bien compris le problème à résoudre... Repensez au F15 la prochaine fois que votre BA vous demande quelque chose... Les tests d'acceptance ne font pas tout de toutes les façons. Il est important de pouvoir s'en servir comme d'une documentation vivante de votre projet. Antony Marcano insiste sur l'importance de l'expressivité des tests, sur l'importance de la maîtrise de l'écriture naturelle lorsque l'on rédige des

tests d'acceptance. La vraie documentation d'un projet, qui sera toujours à jours, c'est la liste des tests d'acceptance. Limpide, intelligent, j'aime. Un souci rencontré est l'utilisation de ces tests pour faire de la non-régression. Moi le premier sur notre projet, nous utilisons FitNesse pour comparer des rapports financiers. Or bien qu'il soit évident que cela soit possible, ce n'est pas l'objet des tests d'acceptance. On verra plus loin l'importance de la communication entre les humains. Vous savez, ce gars avec une cravate qui vient chaque mois et que l'on vénère comme "LE CLIENT". Et bien si votre chef de projet ne sait même pas comment il s'appelle, virez le chef de projet. C'est un incompétent. Pour le point 5, il est important de ne pas se faire embarquer par les outils de tests fonctionnels. Allez, qui est venu voir un jour son product owner, a lancé FitNesse en réunion, et a annoncé fièrement au client qu'il fallait maintenant qu'il utilise cet outil ? L'outil ne doit pas être imposé à votre client, à votre MOA. Le plus simple c'est un tableau blanc et des crayons. Si quelqu'un de votre équipe peut écrire quelques tests pendant la réunion, c'est génial. Dites au client que vous êtes entraîné de faire le compte-rendu de la réunion, ce qui est vrai, et écrivez correctement les différents points de validation sur votre page FitNesse. Vous ferez la plomberie plus tard. Si un outil d'acceptance test n'est pas bon, ou que vous êtes bloqué, changez d'outil ! Ne dites pas au client que "cette partie on peut pas la tester avec FitNesse". C'est simplement débile ! A vous de trouver d'autres moyens pour tester le logiciel et conserver le principe des tests d'acceptance. Gilles de Valtech apporte aussi un point dans la discussion intéressant : les développeurs pensent que les tests ne sont pas aussi important que le code. Si les tests unitaires ne compilent pas, ou ne passent pas, souvent ce n'est pas grave. Si les tests d'acceptance ne sont pas mis à jour, ce n'est pas grave. Et il explique alors qu'un des soucis de l'adoption des tests d'acceptance, est cette mauvaise image que nous leur donnons. Il est important de considérer que le code des tests unitaires est aussi important que le reste du logiciel. Un débat ensuite sur la perception des testeurs par le management. Et là il y a du travail... Les ingénieurs qualités sont mal perçus. Ils se retrouvent dans des équipes à part... "Ouais eux c'est les gens de la QA...". Et bien ce n'est pas bien. Si votre qualité est pourrie, et que vos spécialistes de la qualité sont dans un congélateur au lieu d'être sur les genoux de vos développeurs, vous pouvez toujours attendre un miracle. Craftmanship over Crap comme disait Uncle Bob... vous vous souvenez de ce matin ? Par rapport aux outils comme FitNesse, il est aussi important de se souvenir, que ceux-ci ne sont qu'une fraction de la demande du client. Prenez le jeu du "[Telephone Drawing Game](#)" qu'explique Antony. J'écris une phrase sur une feuille que je passe à mon voisin. Par exemple : "le chien chante à côté d'un Caddie". La personne suivante doit alors dessiner un petit dessin à partir de la phrase, plier la feuille pour masquer ma phrase et ne laisser que le dessin, puis ensuite faire passer la feuille à son voisin. Celui-ci doit écrire ce qu'il voit, plier la feuille pour cacher le dessin, et passer la feuille... et ainsi de suite. Vous suivez ? ... Bon allez voir [là](#) ... Et bien réfléchissez à ce qu'il se passe lorsqu'un client demande à un super fonctionnel qui demande à une MOA qui demande à un développeur de faire une superbe fonction... c'est juste impossible. Et pourtant c'est ce que l'on fait chaque jour non ? Il faut placer de la valeur sur une réunion, où le client et le développeur travaillent ensemble. Prenez un tableau blanc, des crayons, des boissons fraîches, et faites bossez ces gens ensemble. J'aime beaucoup l'idée de retirer dans un premier temps ces inséminateurs d'intelligence artificielle que sont les petits gars de la MOA. Si le client n'est pas disponible car trop important, trop rare, trop pas le temps de parler avec des développeurs, et bien tant pis. Antony explique qu'après tout, 40% des projets repartent en réécriture. Surtout lorsque les clients/décideurs changent... Je pose la question suivante : quel lien pouvons-nous faire entre les éléments d'un product backlog et les tests d'acceptance ? La réponse est la suivante : les User stories ne sont que le scope de ce qu'il faut faire. Et c'est souvent volontairement, que l'on laisse une part de vague dans leur définitions. Par contre, les tests d'acceptance que vous devez définir lorsque la tâche débute, sont eux des éléments de validation. Enfin il faut se souvenir qu'en Scrum, c'est le client au final qui accepte ou non la fonctionnalité. Les tests d'acceptance sont là pour aider, mais ce ne sont pas des tests de recette ! Gojko rappelle le principe des 3C : - Card, - Confirmation and - Conversation Une carte cartonnée qui sert de fiche navette, une confirmation au dernier moment avec le client, ce qui permet d'écrire les tests d'acceptance et enfin une Conversation avec lui pour bien définir ce qu'il faut faire et ce qu'il veut. Repensez aux F15. Finalement ce n'était pas un avion rapide que le client voulait, mais un avion capable de ne pas se faire descendre par le Mig de l'époque. Et la vitesse n'était pas la solution au problème. Pour terminer, nous avons rappelé l'importance du management dans ce processus. Il faut aligner tout le monde pour travailler ensemble. Une spécification ne fait pas tout, un analyste m'écrit une spécification et qui s'en lave les mains ensuite, c'est n'importe quoi. Un développeur qui commite son code et qui se fiche ensuite des tests ou de la mise en production, c'est n'importe quoi. Un chef de projet qui joue avec Microsoft Project mais qui ne sait pas comment s'appelle son client, c'est juste n'importe quoi. Enfin la conclusion a été qu'il ne faut pas sous-estimer les efforts nécessaires pour mettre en place des tests d'acceptance. Cela entraîne des changements parfois violent, et il faut donc accompagner le changement. N'essayez pas d'imposer un outil comme FitNesse à vos clients ou vos équipes d'analyste. Ce que vous voulez imposer c'est un principe de fonctionnement, des règles de jeu. Pas la couleur du maillot ou la taille des cheveux. En conclusion, une séance très intéressante qui donne envie de s'intéresser à ces fameux tests, pour améliorer la qualité de son travail. J'adopte ! **Références** [Blog d'antony](#) [Blog de Gojko Adzic](#)

## BarCamp Paris le 29 septembre à la Cantine

Vu ce matin sur Twitter, il y aura un BarCamp organisé par Google à LaCantine le mardi 29 septembre prochain de 14h00 à 21h00. On regrette peut-être le fait que cela se passe en partie en journée, mais allez sur [le site du BarCamp](#) pour plus de détails. Le 1er octobre prochain aura lieu en "[Fuse Community Day](#)" à Paris (inscription [ici](#)). Conférence gratuite, avec petit déjeuner et déjeuner autour des problématiques d'intégration. FUSE est un ESB basé sur Apache ServiceMix 4, si vous êtes intéressés par les ESB alors c'est pour vous. C'est une distribution professionnelle avec support. FUSE Mediation Router est aussi par exemple la version pro d'Apache Camel. Franchement je découvre ces produits, je n'en sais pas grand chose, que ce soit du côté de la qualité ou du support. Il est important cependant de noter que de plus en plus, les clients finaux cherchent un éditeur ou une entreprise pour offrir du support et de l'assistance sur les frameworks

open-source. Comme toujours vous pouvez retrouver ces événements sur [l'agenda Google Calendar du Touilleur Express](#) et bientôt sur GeekDays, le service pour les geeks qui devrait être lancé dans quelques semaines.

## L'état du marché informatique et quelques idées pour le futur

Le Syntec Informatique estime à environ 370 000 personnes le nombre d'informaticiens travaillant chez un éditeur, une SSII ou une société de conseil en technologie. Les SSII représentent 57% de l'activité, suivi par les Editeurs (29%) et enfin les Sociétés de Conseil en Technologies(14%) [1]. Ce qui revient donc à dire qu'environ 210 000 personnes travaillent au sein d'une SSII en France. Fin 2008 selon le Syntec près de 42% des offres des grandes SSII n'étaient pas pourvues, faute de candidats[2]. Pourtant, après une hausse de 5.5% en 2008, toujours selon le Syntec le secteur ne devrait pas procéder à de recrutement selon la présentation donnée par le Syntec Informatique [le 1er avril dernier](#). Selon les secteurs bien entendu, il y a de fortes variations. Le marché du conseil et des prestations informatiques en Ile-de-France devrait connaître un ralentissement plus important que l'infogérance applicative, bref si tout le monde s'accorde pour dire que le marché devient difficile, nous sommes encore sur un secteur très dynamique. Alors j'ai posé quelques questions autour de moi à des amis collaborateurs en SSII, des personnes qui travaillent chez des Editeurs, et des indépendants comme moi. Voici ce que j'ai entendu : en SSII cette année les augmentations sont difficiles à obtenir. Problème du marché, taux d'inter-contrat en hausse, il est demandé à chacun de faire un effort. L'ambiance n'est pas à la fête. Du côté des éditeurs, l'un d'eux qui offre habituellement jusqu'à 15% de variable en plus à un développeur, a complètement revu à la baisse ce variable, indexé sur les résultats de 2008. La "prime" ne sera donc "que" de 30% au lieu de 100% chaque année. L'année 2009 s'annonce comme encore plus difficile, on peut donc déjà penser que l'an prochain cette prime ne sera pas très élevée. Enfin les indépendants que je connais ont baissé leurs tarifs. En première ligne face au client, c'est eux qui offrent le plus de réactivité lorsque le client souhaite réduire le taux journalier facturé. Tout ceci pour vous dire : oui le marché un peu tendu, mais ce n'est pas non plus la fin du monde. Alors votre SSII vous annonce que c'est la crise, pas d'augmentation en vue... mince comment faire ? Tout d'abord regardez le secteur où vous travaillez. La finance prévoit 2% de croissance en 2009 car en raison des restructurations, des fusions (BNP Paribas rachète Fortis et l'activité Prime Brokerage de Bank Of America) et des nouvelles réglementations, il va y avoir beaucoup de travail cette année. [Le slide 15](#) de la présentation du Syntec informatique vous donnera d'ailleurs la même information. L'industrie par contre, nous pensons tous à l'automobile, devrait connaître un ralentissement, voire une réduction de 3% de l'activité. Si vous avez un doute sur votre valeur, publiez un CV et revenez voir votre responsable avec une offre ferme d'un concurrent... **Evolution du marché et des acteurs** Imaginons ensemble comment le marché des sociétés de service pourrait évoluer dans les années à venir. Je crois tout d'abord que l'éthique et l'humain vont revenir au premier plan. Les SSII qui embauchent par camion entier "de l'ingénieur" ont ensuite du mal à gérer le capital humain, voire à créer un esprit d'entreprise. Forcément, c'est une valeur qui ne s'achète pas, mais qui se construit. Ne soyons pas non plus trop subjectif, il y a d'excellents ingénieurs d'affaires et des responsables d'équipes qui recrutent au sein de leur "BU" (Business Unit) un esprit d'entreprise. Mais l'humain est finalement un capital plus important que la simple compétence, monnayée aux clients finaux des SSII. Le numéro 46 du magazine Newzzy parle ainsi de la renaissance de l'Humanisme, de l'importance de capitaliser sur l'Homme et pas sur une base de compétences. A quand un CV avec un intertitre comme "Auteur du site de micro-crédit pour le développement en Mauritanie des puits d'eau" plutôt que "Certifié Microsoft Level IV formation 4500 EUR" ? Pour cette raison, je crois aussi de plus en plus au recrutement par bouche à oreille. Curieux alors que le tout Internet semble être la voie royale, nous savons tous qu'il est plus intéressant d'être recruté par cooptation ou d'embaucher par recommandation. Cela permet de détecter ce qu'un CV sur Monster ne dit pas : êtes-vous un chic type ? (ou une nana sympa). L'humain serait donc la nouvelle pépite pour l'entreprise, plus que le CV ou l'expérience. Ce facteur humain prend toute son importance lorsque vous envisagez un projet piloté avec les méthodes Agile. Le principe de l'Agilité est très simple : améliorer la communication entre les différentes personnes d'un projet. "*Scrum sert à maximiser le canal de communication entre le responsable du produit d'une part et l'équipe d'autre part (E.Mignot, Pyxis Technologies)*". Le marché de l'informatique reste quoi qu'on en dise, relativement dynamique. Et il faut aussi bien se rendre compte que nous sortons de quelques années vraiment exceptionnelles. Les coups de fils et les mails sont revenus en septembre. J'ai des demandes de la part de mes amis pour devenir indépendant. Bref on sent que l'activité repart. Pour terminer je vous renvoie aussi à [un article publié en mai](#) sur les offres d'emploi autour de la plateforme Java. Références: [1] <http://pro.01net.com/editorial/399161/le-recrutement-sous-surveillance-en-2009/> [2] Le site du [Syntec informatique](#) propose de nombreuses études et des chiffres sur le marché de l'informatique Voir aussi <http://www.touilleur-express.fr/2009/04/08/recruter/> pour trouver le meilleur moyen pour recruter un ingénieur Java

## L'inter-contrat pour Geek, les matinées du Banc Parisien

Si vous êtes en inter-contrat en ce moment, j'ai une information sympa que m'a glissé Luc Bizeul de [Pyxis Technologies](#) , par ailleurs organisateur des premiers [BarCamp](#) Java à Paris. Chaque mercredi, si vous êtes libre en ce moment, vous avez la possibilité de vous retrouver de 10h00 à 12h00 à Paris. Un peu comme sur les coding-dojos du Lundi soir, vous pouvez occuper votre temps de manière utile en venant discuter, partager et coder avec d'autres geeks. Organisé par Laurent Cobos et Luc Bizeul de Pyxis Technologies, le prochain événement aura lieu le mercredi 30 septembre.

**Qui ?** Toute personne pouvant se libérer le mercredi matin. Le banc c'est plusieurs profils : développeur, accompagnateur, formateur... **Quoi ?** Le banc c'est un espace neutre permettant d'échanger, d'apprendre, d'enseigner, de profiter des compétences de chacun. **Où ?** A Paris, Au Père Tranquille 16 rue Pierre Lescot (M. Chatelet les Halles) **Quand ?** Chaque Mercredi de 10h à 12h30, possibilité de rester déjeuner sur place. **Comment ?** Le format du RDV du banc se rapproche d'un open space, plusieurs groupes peuvent se créer pour travailler/échanger sur un sujet particulier proposé par un des participants. Un exemple de sujet fut un coding dojo sur iPhone.

Le site [Le Banc Parisien](#) vous donnera plus d'informations. A tester au lieu de lire 01 Informatique ou de déprimer devant son écran tout seul en attendant le prochain rendez-vous. A quand les petites annonces sur le Touilleur Express ?

## Interruption de service

Bon, je m'essuie les mains et j'arrive... Je vais vous raconter un peu c'était pas beau à voir. Vers 19h45 je reçois un message urgent sur mon iphone : la charge CPU de votre serveur est passée au dessus de 20mn... Depuis quelques jours j'ai des soucis avec le blog. La charge CPU part à 100% et je dois le relancer. C'est arrivé début août et depuis je n'arrive pas à trouver ce qui se passe. Mais ce soir j'y crois. Ni une, ni deux je prends ma trousse de soin et je fonce au bloc voir l'état du patient. A cet instant j'ai deux options : soit un restart via la console de secours, soit essayer de se connecter via SSH mais il ne me reste que quelques minutes avant que le serveur ne meurt... J'arrive à me connecter. Je fonce vers le répertoire des fichiers logs. C'est là, sous mes yeux : un fichier access.log de 155Mo. Un petit tail me montre tout de suite que la raison du ramage du serveur Web est facile à trouver (j'ai changé l'adresse IP). On note que le User Agent est Java... ben dis-donc, c'est pas un méchant virus à vue de nez.

```
anyuser@touilleur-express.fr> tail -f access.log
123.999.170.30 - - [27/Sep/2009:23:01:29 +0200] "GET / HTTP/1.1" 200 40129 "-" "Java/1.5.0_14"
123.999.170.30 - - [27/Sep/2009:23:01:29 +0200] "GET / HTTP/1.1" 200 40129 "-" "Java/1.5.0_14"
123.999.170.30 - - [27/Sep/2009:23:01:17 +0200] "GET / HTTP/1.1" 200 40129 "-" "Java/1.5.0_14"
123.999.170.30 - - [27/Sep/2009:23:01:17 +0200] "GET / HTTP/1.1" 200 40129 "-" "Java/1.5.0_14"
123.999.170.30 - - [27/Sep/2009:23:01:32 +0200] "GET /feed/ HTTP/1.1" 200 77121 "-" "JT*****/1.0-alpha r27"
123.999.170.30 - - [27/Sep/2009:23:01:30 +0200] "GET / HTTP/1.1" 200 40129 "-" "Java/1.5.0_14"
123.999.170.30 - - [27/Sep/2009:23:01:34 +0200] "GET /feed/ HTTP/1.1" 200 77121 "-" "JT*****/1.0-alpha r27"
123.999.170.30 - - [27/Sep/2009:23:01:34 +0200] "GET / HTTP/1.1" 200 40129 "-" "Java/1.5.0_14"
123.999.170.30 - - [27/Sep/2009:23:01:34 +0200] "GET /feed/ HTTP/1.1" 200 77121 "-" "JT*****/1.0-alpha r27"
123.999.170.30 - - [27/Sep/2009:23:01:34 +0200] "GET /feed/ HTTP/1.1" 200 77121 "-" "JT*****/1.0-alpha r27"
```

Voyons un peu combien de fois cette adresse IP est passée ces dernières heures

```
anyuser@touilleur-express.fr> grep 123.999.170.30 | wc -l access.log
630348 access.log
```

Ouais super, c'est open-bar. 630 000 connexions, autant dire que c'est plus du flooding à ce niveau-là. Bon voyons qui est derrière cette adresse IP... Google est mon ami. Ah je suis sur kenai.net, et c'est un projet open source appelé JT\*\*\*\*\*. Un lecteur de feed RSS en ligne. C'est sans doute un petit bug, mais bon, le gars qui a codé cela a juste laissé un @#\$% de bug qui me coûte cher. Bon pas envie de jouer avec iptables, en fait je sais pas. Alors j'ai édité mon fichier de conf apache2 et j'ai ajouté une ligne sournoise, qui le vire et m'évite qu'il me pompe toutes mes ressources. **Stop au cannibalisme** La majorité des visiteurs du Touilleur Express passent via les feeds RSS. Ces feeds RSS permettent aussi facilement de récupérer le contenu d'un blog. Or le contenu du Touilleur Express n'est pas libre de droit. Il n'est pas possible d'agréger son contenu et de le poster sur un site. Et ce, parce que ces sites affichent ensuite des publicités, ce qui leur permet de vivre tranquillement en vivant sur votre dos. Depuis quelques temps sur Google Reader j'utilise [FairShare](#). Ce moteur très pratique me signale les articles copiés. Je peux alors simplement ignorer les citations ou les bouts recopiés, ou carrément lancer mes armées d'avocats sur le pauvre site en question... Bon ok, je n'ai pas d'avocats, mais le coeur y est... Et si vous pouviez vraiment ne pas utiliser ces services, vous me feriez très plaisir. Par exemple - bloglines.com - socialmedian.com - b-trail.com - dailynewsabout.com PS: j'ai changé l'IP et le nom du projet open-source bien entendu. Mais vivien, si tu lis mon blog, tu seras sympa de faire en sorte que j'ai pas une facture astronomique chez mon hébergeur ce mois-ci. Merci

## Naissance du MarsJUG, le Java User Group de Marseille

Oyé, oyé, Julien Viet est heureux de vous annoncer la naissance du petit [MarsJUG](#). Avec [le Riviera JUG](#) basé à Sophia, c'est donc le deuxième groupe d'utilisateurs de la plateforme Java qui voit le jour dans le sud de la France, et le 15e JUG en France si mes comptes sont bons. L'inauguration aura lieu le jeudi 15 octobre à partir de 18h30 avec la présence du parrain, Emmanuel Bernard (Hibernate Search, JBoss, les CastCodeurs). Au programme, une présentation d'Hibernate Search, qui permet de faire de la recherche full-text avec Hibernate. Bravo aux Marseillais, mais bon, Bordeaux est toujours en tête ;-) Pour plus de détails voir la page [événement](#) sur le site du MarsJUG. J'en profite aussi pour vous signaler un événement très sympa à Sophia-Antipolis : le 2 octobre le Riviera JUG organise la journée "[Sophia fait sa Java](#)". Avec la présence d'Antonio Goncalves, de Peter Muir, d'Alexis Moussine-Pouchkine et de Tom Baeyns, c'est un événement à ne pas manquer !

## Beginning Java EE 6 Platform with GlassFish

### 3

Voici une petite revue du livre "[Beginning Java EE 6 Platform with GlassFish 3](#)" édité chez APress, écrit par Antonio Goncalves, leader du Paris Java User Group, mais aussi expert sur les JSR-316 (Java EE 6), JSR-317 (JPA 2.0) et JSR-318 (EJB 3.1). 476 pages plus tard, après avoir pris mon temps pour le lire, j'ai été impressionné par la quantité et la qualité du travail. Expliquer un sujet comme la plate-forme Java EE 6 en 500 pages, ce n'est pas une mince affaire mais un exercice réussi. Ce livre s'adresse à un public assez large. Je pensais initialement qu'il faudrait avoir de bonnes connaissances au niveau de la plate-forme Java EE, mais le contenu étant très didactique, le livre s'adressera aussi sans problèmes aux débutants. A ma connaissance c'est le seul ouvrage en anglais sur ce sujet pour l'instant. Vous avez remarqué que je parle de "plate-forme Java EE" plutôt que de Java EE 6 ? Qui se souvient de la JSR-151 ? Il s'agit la spécification de Java J2EE 1.4 qui date de 2004. 5 ans, c'est beaucoup dans le monde Java. La spécification [JSR-244](#) spécifiant JEE5 date quant à elle de 2006. La spécification de la plate-forme Java EE6 sort à peine, puisque la spécification JSR-317 (JPA2.0) a été proposé en final draft le 22 septembre dernier. C'est donc un livre qui s'est terminé en même temps que le travail de spécification, avec certainement des précisions ou quelques changements l'année prochaine dans une réédition. Mais le contenu est vraiment complet. Le livre est construit autour de 15 chapitres : 1 - Java EE 6 at a Glance 2 - Java Persistence 3 - Object-Relational Mapping 4 - Managing Persistent Objects 5 - Callbacks and Listeners 6 - Enterprise Java Beans 7 - Session Beans and the Timer Service 8 - Callbacks and Interceptors 9 - Transactions and Security 10 - JavaServer Faces 11 - Pages and Components 12 - Processing and Navigation 13 - Sending Messages 14 - SOAP Web Services 15 - RESTful Web Services Moi qui suis sensible au style et à l'écriture, j'ai bien aimé le ton très didactique. Pour avoir lu pas mal de livres en Anglais et en Français, c'est vraiment bien écrit, les pages se tournent toutes seules. Le premier chapitre vous guide dans l'installation de votre plate-forme pour coder ensuite les exemples du livre. Mes premières impressions sont que JEE 6 est plus léger que son cousin JEE 5. Plus facile à mettre en marche aussi. Le chapitre 2 sur la persistance et JPA 2.0 nous guide dans l'écriture d'une petite application avec la gestion d'entité, l'écriture des tests avec Derby, les principes de requêtes, avec un résumé en fin de chapitre. C'est un chapitre orienté pratique. La théorie au chapitre 3 permet de comprendre en 60 pages les principes de l'Object-Relational Mapping. En expliquant en détail le principe du mapping avec annotation et XML, la gestion des relations de 1 à n, c'est plutôt théorique, mais bien illustré avec des schémas. Le chapitre 4 est dans la continuité du 3. Comment interroger des objets persistants ? Quel est le rôle de l'EntityManager, le langage JPQL, la création des Queries et enfin la gestion de la concurrence avec JPA 2. Le chapitre 5 relativement court n'est pas le plus important, on y parle des callbacks et des listeners afin d'être notifié du cycle de vie des Entities. C'est au chapitre 6 que l'on rentre dans le coeur de la spécification JSR-318 à mon avis. Avec tout d'abord un rappel des Entreprises Java Beans, réactualisé avec le recul dont nous bénéficions aujourd'hui. Le livre est construit en partant de la base de données, pour ensuite attaquer les EJBs et se termine par REST en passant par JSF. Tout au long du livre nous suivons la constructions d'un site marchand. J'en appelle tout d'abord aux sceptiques qui pensent que Spring peut sauver le monde. Franchement, allez voir un peu les dernières nouveautés, car la simplicité et la beauté des EJB 3.1 en font de sérieux candidats pour virer nos copains les petits Beans. Comment dire ? En fait c'est plus mature. L'anatomie d'un EJB coule de source. Je me souviens d'une présentation à DevOxx qui m'avait bluffé : l'écriture en 1h d'une application complète JEE6 avec au passage, de sérieux coups à notre bon vieux copain Spring. Il faut dire ce qu'il est : il serait dommage de ne vivre que sur la branche Spring et de ne pas regarder ce que Java propose désormais. Page 171 par exemple, on se plonge dans la description d'un conteneur EJB avec plaisir. Etre concis c'est facile. Etre concis et précis, c'est plus dur. Le conteneur embarqué permet ainsi de travailler simplement avec les EJBs :

```
EJBContainer ec=EJBContainer.createEJBContainer();
Context ctx=new Context();
BookEJB bookEJB=(BookEJB) ctx.lookup("java:global/BookEJB");
bookEJB.createBook(newBook);
```

Le chapitre 6 se termine par un atelier où le lecteur est amené à coder la partie service de son application. La compilation du projet avec Maven 2, révisée aussi par Arnaud Lheritier, vous permet vraiment d'être productif. Ce n'est pas un livre qui vous dit : "bon et bien vous lancez néclipsu et puis zou... ça compile...". Peut-être parce qu'Antonio est un fan d'IDEA IntelliJ ? Tant mieux ! Les chapitres 7,8 et 9 couvrent les services annexes de la plate-forme comme le Timer Service par exemple. Cela permet de créer des Beans qui s'activent de manière régulière, comme un cron. Très pratique. J'ai honnêtement survolé le chapitre 8, et j'ai regardé la gestion des Transactions et de la Sécurité. Le chapitre 9 débute par un rappel des principes des transactions (ACID) toujours bon à revoir. Bon l'exemple du transfert bancaire sans un peu la naphtaline mais comme le principe est d'expliquer l'Atomicité, cela fonctionne. Allez citez-moi les 6 attributs de démarcation des transactions gérées par le conteneur... Allez je vous attends : REQUIRED, REQUIRES\_NEW, SUPPORTS, MANDATORY, NOT\_SUPPORTED et NEVER. Un schéma simple résume les différences entre chacun de ces attributs, et du code vous montre ensuite sa mise en oeuvre dans un EJB. La sécurité est ensuite abordée. Les annotations bien entendu apportent une simplification, la possibilité de verrouiller le code de son application est bien expliquée. La fin du livre aborde la couche de présentation. JSF 2 tout d'abord, qui sera aussi présenté [le mardi 13 octobre](#) au Paris JUG. Si vous venez, pensez à vous inscrire sur la liste de diffusion du Paris JUG afin d'être notifié lorsque les inscriptions seront ouvertes. Revenons à JSF 2.0. Page 270, après un historique de JSF, nous entrons dans le vif du sujet. Les exemples de code me parlent, et là où il faut 3 interfaces, 2 DAO et je ne sais pas combien de machin pour afficher une liste, JSF 2.0 et les EJB 3.1 sont dramatiquement simples. Oui je sais, tu es fan de Wicket et de GWT. Non de Spring MVC pardon. Et tu te dis que je rame pour présenter le sujet... Pas du tout. Les gens ont longtemps pensé que la terre était plate, et lorsqu'un explorateur a prouvé le contraire, il s'est fait griller. Je m'attends donc à une levée de bois vert, où chacun avec son web framework et ses interfaces

s'empresera de crucifier JSF 2.0. Et bien non, en 2009 la spécification a vraiment évolué. Si vous sortez vos vieux numéros du Touilleur Express, regardez [en 2006](#) ce que je m'amusais à écrire pour l'époque. Ou cet article de [2005](#) qui est vraiment vieux, après un passage chez Valtech et une présentation des EJB3. Là où je préfère Wicket, c'est son intégration fine avec la page Web. JSF 2.0 n'est pas loin, mais il n'y a pas cette notion de balises standards décorées. Cela dit, la mise en page HTML s'effectue aussi sans problèmes. Le chapitre 11 parle de la mise en page, des styles, et aussi de JSP. Cependant comme expliqué à un point dans le livre, la plate-forme stratégique de rendu de JEE6 est JSF 2 plutôt que JSP. Pensez-y si vous souhaitez faire une formation de remise à niveau. Vers les pages 299-306 on redécouvre aussi (pour moi) que les taglibs peuvent rendre service pour des exemples de code très simple. La suite se concentre sur JSF 2, la définition de la navigation, les Validators et Ajax. Et j'ai trouvé ce chapitre vraiment complet, le sujet est abordé très précisément. La question des annotations versus la déclaration statique dans les fichiers XML est aussi expliquée simplement. Le chapitre 13 aborde la présentation du Messaging Layer de la plate-forme de Java EE6. De la définition d'un MDB en passant par une présentation détaillée de JMS, rien ne manque. L'injection de dépendance est au coeur de la plate-forme Java, avec l'injection des Ressources de type ConnectionFactory ou Queue. Rien à redire, c'est similaire à Spring, mais plus intuitif. L'arbre JNDI remplace ce qu'il manque je pense dans la sauce des descripteurs XML de Spring : un annuaire. Mince j'ai critiqué Spring, je vais encore me faire crucifier... Le chapitre 14 présente les Web Services. La mise en place de JAXB 2.0 et des annotations permet vraiment de rendre simple la création et l'appel des Web Services. C'est bien simple, comme un vendeur de Darty j'ai envie de vous dire : "et j'ai le même à la maison, et ça marche très bien...". Cependant quel est le recul et le retour sur cette partie ? Au niveau des performances ? De la simplicité de déploiement ? Enfin le dernier chapitre parle de REST et des Web Services. On sort un peu de la norme puisque REST est un principe d'architecture. Le chapitre présente JAX-RS 1.1 avec l'implémentation Jersey proposée par SUN Microsystems. Ce chapitre est vraiment intéressant à lire si l'architecture est un sujet qui vous intéresse. Il ouvre des portes vers de nouveaux services, une autre façon de concevoir nos applications. Le principe de REST date de 2000, mais est-ce à dire que nous en verrons que maintenant les premiers projets ?

```
@Path("/book")
public class BookResource {
    @GET
    @Produces("text/plain")
    public String getBookTitle() {
        return "GlassFish V3";
    }
}
```

J'imagine une architecture avec un Web tiers, composé de Portlet, mon dada en ce moment. Et ces portlets ou même des Gadgets comme chez eXo, pourraient consommer des services de type REST qui exposeraient du json par exemple. Cela résout un problème de rupture de protocole pour la sécurité, mais me semble plus élégant que des anciens EJB. **Conclusion** Le livre s'adresse donc à des débutants de la plate-forme JEE6, à des architectes qui souhaitent remettre leurs connaissances à jour. Honnêtement, ne faites pas la bêtise de vous enfermer dans une seule technologie comme Spring. Notre métier évolue très vite, et je ne dis pas que nous ferons tous du Java EE 6 l'an prochain, mais au moins vous aurez pris le temps de regarder et de comprendre. A quand aussi une version traduite en français, réactualisée l'année prochaine par exemple ?

## Revue de Presse

Petit tour des dernières informations passées cette semaine dans la blogosphère. **Déclarations de Larry Ellison à propos du Cloud Computing** Tout d'abord les déclarations de Larry Ellison, CEO d'Oracle à propos du mot "Cloud Computing". La semaine dernière au Churchill Club lors d'une interview en compagnie d'Ed Zander, CEO de Motorla, Larry Ellison a déclaré que le cloud computing était pour lui "nonsense and water vapor". Il ne s'insurge pas contre le cloud computing en tant que tel mais contre le Buzz qui lui semble exagéré. La vidéo dure 5mn. Larry Ellison explique que le Cloud Computing est pour lui, de la location de ressources distribués mises en réseau. Le Cloud c'est la mise en réseau d'ordinateurs tout simplement. Que cela soit un nouveau business model pour certains... pourquoi pas. Mais Salesforces.com après tout a débuté il y a 10 ans, et on parlait à l'époque simplement de "Hosted Service".

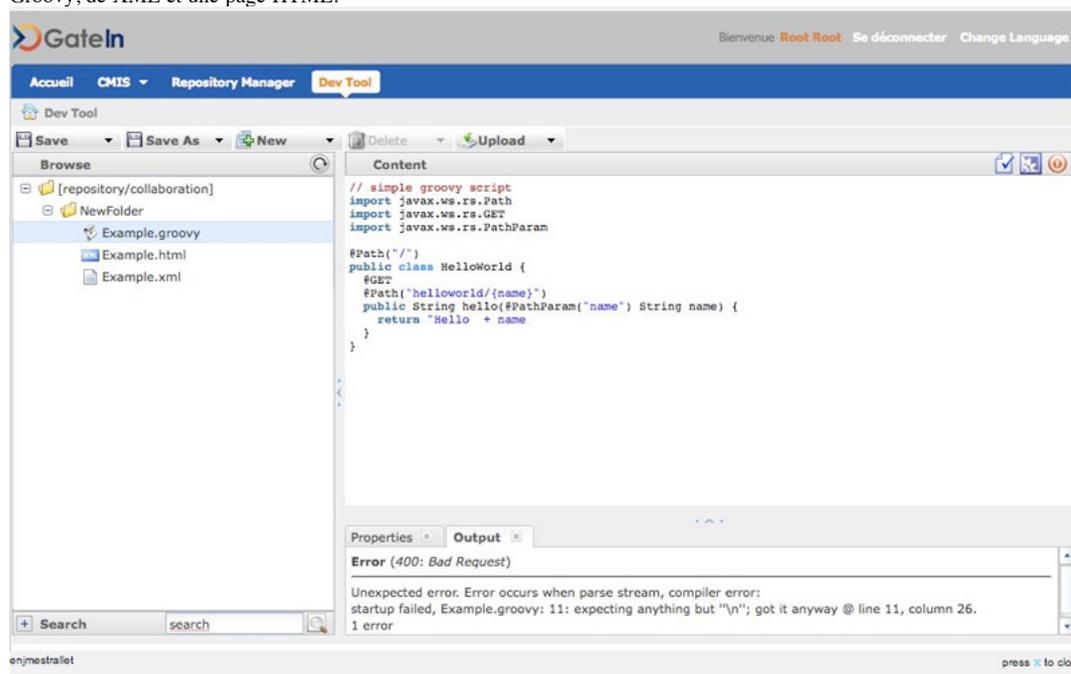
**100 000 invitations Google**

**Wave** La deuxième info de cette semaine, que vous avez vu sur Twitter, c'est que Google a lancé

100 000 invitations Google Wave supplémentaires. J'en ai eu 8 qui sont toutes parties comme des petits pains. L'info est sortie ce mercredi [sur le blog de Google](#). Ce n'est pas encore un lancement officiel, il manque un certain nombre de fonctions comme la possibilité de retirer une personne d'une Wave ou de créer des groupes. De même le mode Draft qui permet de taper tranquillement son texte puis de l'envoyer comme sur un Chat ne semble pas fonctionner. Bref ce n'est pas encore pour demain. Les add-ons pour Google Wave continuent à être développés par une communauté active sur Google App Engine. [Rabbit](#) par exemple permet d'ajouter la voix à une Wave. Cela transforme votre Wave en une salle de conférence en quelques sortes. [6 Rounds](#) ajoute la gestion de la vidéo, à la iChat. Bref tout ceci renforce l'intérêt de Google Wave, lorsque celui-ci sera disponible officiellement. **eXo Platform révèle quelques projets** Sur Twitter il ne fallait pas louper un échange entre @glaforge et @benjmestrallet vendredi dernier. Il semble que l'équipe d'eXo Platform réfléchit à un éditeur GWT avec coloration syntaxique pour Groovy, HTML, XML pour construire des applications REST en ligne.

```
@benjmestrallet said:
@glaforge you won :) it is the GWT eXo IDE with syntax coloring for Groovy ,
HTML, XML to build REST applications online ...
to come soon
```

Voici une capture d'écran de l'outil, où l'on voit en effet un éditeur de texte avec un peu de Groovy, de XML et une page HTML.



Source de l'image : <http://tweetphoto.com/ldt8appu> C'est amusant car l'idée de l'IDE simple et hébergé était une discussion que nous avions avec David Gageot il y a quelques mois. C'est plein de sens. Au lieu d'avoir un IDE sur sa machine, pourquoi ne pas penser et proposer un IDE en ligne ? Un moteur simple avec coloration syntaxique par exemple. Le modèle des Gadgets est de plus en plus intéressant. J'ai des tonnes de questions à poser à l'équipe d'eXo Platform maintenant, c'est malin ça. A ce propos, Ippon Technologies organise le jeudi 8 octobre un séminaire sur les Portails Opensource comme Liferay et eXo Portal. Plus de détails [sur le site d'Ippon](#). A bientôt !

## Enquête sur la Scrum Alliance

[La Scrum Alliance](#) est une association à but non lucratif, fondée par Ken Schwaber, Mike Cohn et Esther Derby. Sa mission est de promouvoir Scrum, de proposer des ressources pour les entreprises et les particuliers intéressés par Scrum, et d'aider les groupes d'utilisateurs de Scrum locaux. La démission de Ken Schwaber [le 15 septembre 2009](#) nous a donné envie d'enquêter sur l'association. Pour cela, le site internet de la Scrum Alliance propose de télécharger un bilan financier simplifié pour l'année 2007. Et les chiffres sont impressionnants... [Le formulaire F990](#) est un bilan financier simplifié qui est librement accessible sur le site de la Scrum Alliance. Le document est daté du 11/11/2008, sur l'exercice de l'année 2007. Vous pouvez en avoir une copie [sur cette page](#) en cliquant sur [ce lien](#). J'imagine que l'exercice 2008 sera aussi en ligne à la fin de cette année, et qu'il sera possible de retrouver ces chiffres sur Internet aussi. Avant toute chose, il est important de noter ceci: [...]*The Form 990, entitled "Return of Organization Exempt From Income Tax," is a report that must be filed each year with the Internal Revenue Service (IRS) by organizations exempt from Federal income taxes under section 501 of the Internal Revenue Code, and whose annual receipts are "normally" more than \$25,000 a year. It is an information return and not an income tax return since the organizations that file it do not pay income taxes (except, as explained below, in certain cases an organization may have to pay an "unrelated business income tax").* [1] L'exercice est clos avec un excédent brut de 899 000 USD. Les recettes sont de 1 494 921 USD pour l'année 2007. Ce chiffre se décompose en 1 053 395 USD de cotisations collectées au près des membres de la Scrum Alliance, 50 000 USD d'aide publique, 13 272 USD d'intérêts bancaires et 381 327 USD de bénéfice des 2 Scrum Gathering organisés en 2007. Le chiffre de 1 053 395 USD de cotisations, bien qu'il paraisse important, n'est pas extraordinaire, rapporté au 64 000 membres de l'association (voir [ici](#)). Les dépenses et les frais engagés s'élèvent à 595 099 USD,

dont 38 000 USD de frais de voyage, 261 356 de frais de contrats de services, et seulement 130 825 USD reversé pour aider les membres (Page 2, ligne 43-e, Membership support 130 825 USD). Intéressant non ? Ce qui est très important à noter, c'est qu'à la page 6, aucuns des membres de l'association ne touche un dollar pour ses heures de travail, précédemment listées à la page 5. Regardez la section V-A "Current Officers, Directors, Trustees and Key Employees", il est explicitement marqué qu'aucun ticket de présence n'est versé pour leurs heures. Cela ne veut pas dire que leurs frais ne sont pas remboursés, mais que cela ne leur rapporte rien directement. **Les Scrum Gathering ont généré un bénéfice de 381 000 \$ en 2007** A la page 13, le bilan financier du Portland Gathering et du London Gathering fait apparaître un bénéfice de 381 327 USD. Le Portland Gathering a reçu 206 800 USD en souscription, a dépensé 73 280, ce qui génère un profit net de 133 520 USD. Le Londong Gathering a reçu 527 066 USD, dépensé 279 259 USD et donc le revenu net est de 247 807 USD. Au total, pour 733 866 USD collectés, et 352 539 USD dépensé, le revenu net est de 381 327 USD. L'organisation des Scrum Gathering génère donc un revenu net d'environ 50% des frais d'inscriptions collectés. Evidemment tout ceci donne des trucs assez amusants, des débats et surtout un site qui m'a bien fait rire : [The Scrum Alliance Jet](#). C'est est un site parodique qui vous promet de réserver votre Jet privé dès lors que vous aurez passé la Précieuse Certification Scrum. **La Scrum Alliance et les groupes d'utilisateurs Scrum locaux** Mark Levison dans un article sur InfoQ s'inquiétait [au début de cette année](#) sur l'agreement que les Scrum User Group doivent signer pour avoir le droit de porter le nom officiel de SUG et l'usage des logos. Le SUG d'Orlando voulait fermer ses portes, [un article assez précis](#) relate pour quelles raisons, quelques SUG ne voulaient pas signer d'agreement sans contre-partie, à part le droit d'utiliser le terme Scrum User Group et le logo de l'alliance. **Un avis purement subjectif** Ce que m'inspire ces faits, et là je passe dans le subjectif, c'est que ces chiffres sont assez importants. On peut louer la transparence mais être aussi étonné par les montants collectés. Ce sont des sommes d'argent importantes. Et on ne parle que de 2007. [Alexandre Boutin](#) nous a aussi signalé sur la liste du French SUG une information importante : le 15 septembre dernier **Ken Schwaber** a annoncé qu'il démissionnait du board de la Scrum Alliance, ainsi que **Jim Cundiff** ([voir la news ici](#)) :

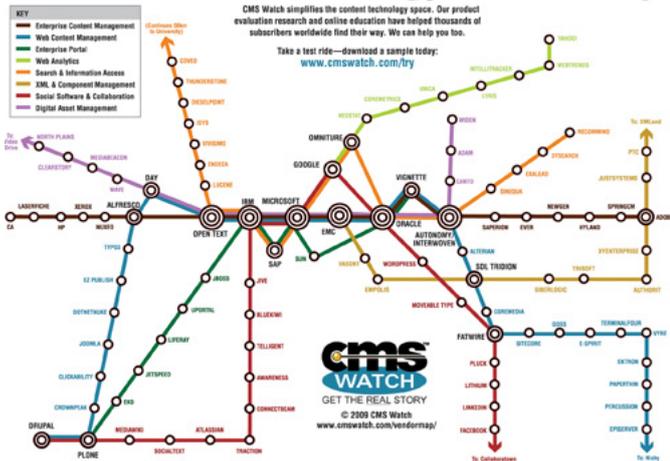
On September 15, Ken Schwaber resigned as President and Chair of the Board of Directors of the Scrum Alliance. Tom Mellor was elected President and Chair of the Board by the Board members. Ken expressed to the Board his intentions to remain active in the Scrum community. The Board expresses its appreciation to Ken for his service and leadership. Additionally, Jim Cundiff will no longer serve as the Managing Director of the Scrum Alliance effective immediately. Jim will support the transition of his duties to others in the organization and assist as needed until a new managing director is brought aboard. Jim has served the Scrum Alliance very effectively and capably and we thank him for his leadership. A search is underway for a new managing director. The Board anticipates filling the position in the very near future.  
[...]

Sur [le blog de Claude Aubry](#) j'ai vu aussi une référence au départ de Ken Schwaber. Le site [www.scrum.org](#) a été lancé par Ken Schwaber, [le nom de domaine appartient bien à Ken](#) et le site a été modifié il y a quelques semaines... Bref il sera important de garder un œil avisé sur la Scrum Alliance dans les mois qui viennent. **Références**  
[http://www.infoq.com/news/2009/05/scrum\\_alliance\\_change](http://www.infoq.com/news/2009/05/scrum_alliance_change)  
<http://www.infoq.com/news/2009/04/scrum-alliance-user-group>  
<http://www.aubryconseil.com/post/2007/06/06/241-la-scrum-alliance-me-congratule>  
<http://www.scrumalliancejets.com/> <http://www.scrumalliance.org/> [Le site de Ken Schwaber CSP](#)  
(Certified Scrum Practitioner)

## Séminaire sur les portails open-source chez Ippon Technologies

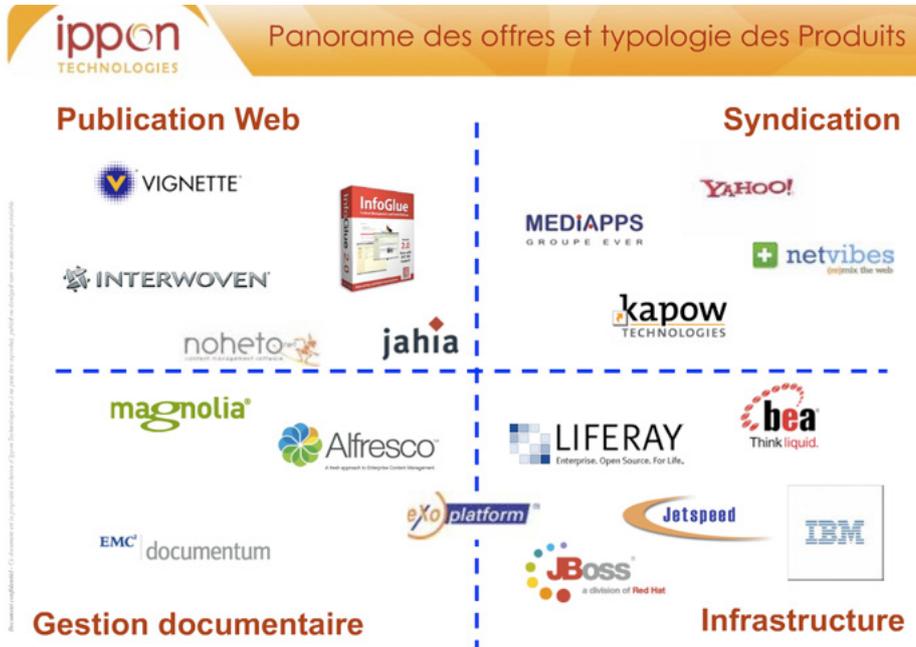
Jeudi matin j'ai assisté à un séminaire sur les Portails Java Open-Source organisé par [Ippon Technologies](#) dans les locaux de Sup Info à Paris. Après une présentation des Portails, de la spécification des Portlets et des différents types de projets rencontrés, nous avons eu le témoignage d'un client eXo Portal puis d'un client Liferay. Le tout s'est terminé avec une séance de questions/réponses. Bertrand Pinel d'Ippon Technologies a tout d'abord présenté un état de l'art des Portails Open-Source. La présentation débute par une explication du mot "Portail". Car chacun a une image en tête. Certains voient un portail à la iGoogle ou la NetVibes, d'autres voient une page comme Amazon.com, et d'autres voient un portail en fer forgé. Bon pour le groupe 3 : un Portail Java est un logiciel qui permet d'agréger du contenu, et d'agir comme un proxy vers l'information. Les Portlets sont en quelques sortes des applications légères qui, lorsqu'elles sont placées ensembles, créent un service sur une page. Comme tout ceci est un peu abstrait, je vous promets de vous raconter une vraie histoire d'ici à la fin de l'année sur le Touilleur Express. Un Portail est aussi un moyen de proposer des services communs comme l'authentification, la gestion des droits des utilisateurs, la possibilité de personnaliser son interface, ou de contribuer avec du contenu. Un portail est aussi un socle technique sur lequel vous pouvez faire fonctionner des services à valeur ajoutée comme un gestionnaire de contenu (CMS) ou un Wiki par exemple. Donc on retiendra surtout qu'un Portail est le chassis de la voiture, et que ce sont les services que l'on pose dessus qui en font ensuite "un portail de contenu" ou "un intranet" ou encore "une page d'accueil de notre site institutionnel". Vous suivez ? Tout le monde fait du portail... Mais tout le monde ne fait pas du Portail, avec un P majuscule. Les éditeurs traditionnels comme Oracle et IBM sont certes présents sur ce marché. Mais ce sont des constructeurs qui viennent du monde du middleware. A l'opposé, dans le monde de l'open-source il y a d'autres acteurs qui ne font que cela, et qui le font mieux d'après Bertrand Pinel. Avis que je partage de plus en plus. Cette carte de métro provient du site "CMS Watch". Elle présente l'ensemble des Portails, des CMS, des outils de publication de contenu. ([voir l'image en grande taille](#))

# 2009 Content Technology Vendor Map



Alfresco et Nuxeo

sont sur la branche "Enterprise Content Management" alors que les Portails d'Entreprise sont représentés par JBoss Portal, eXo Portal, Liferay ou uPortal par exemple. Je trouve ce schéma très intéressant pour comprendre comment chacun des acteurs du marché se positionne. Et finalement, est-ce qu'une approche modulaire à la eXo n'est pas plus intéressante qu'une approche tout ou rien à la Liferay ? Je crayonne sur mon bloc... "Modularité = Flexibilité = eXo". Nous parlons ensuite de licences, LifeRay comme eXo Portal sont open-sources. Puis ensuite quelque chose qui me parle en ce moment : est-il possible d'identifier les attentes des utilisateurs selon leur métier ? Un utilisateur final demande de la facilité d'utilisation. Une interface intuitive, une ergonomie, une simplicité pour qu'il soit autonome. Un administrateur de contenu demande des outils d'administration pratiques et simples. Un admin technique aimerait des Portlets d'administration afin de visualiser la santé de son portail. Un responsable marketing aimerait voir si cette portlet de simulation de crédit fonctionne bien ou non... Bref ce qui est très intéressant c'est qu'il faut penser aux différents types d'utilisateur. Liferay propose ainsi un grand nombre de portlets "out of the box", ce qui permet de commencer à s'en servir rapidement. Dans son Control Panel, un onglet permet de voir aussi la charge mémoire du serveur. Patrice Lamarque mentionne l'utilisation de JMX du côté de eXo Portal, mais je ne me souviens pas avoir vu pour l'instant de Portlets d'administration. Bertrand Pinel explique aussi que le choix d'un portail doit s'effectuer par rapport aux respects des standards, comme la JSR-168 et la JSR-286 pour les Portlets, mais aussi la JSR-170 pour tout ce qui est Java Content Repository. Au passage concernant les Portlets, il est important de comprendre qu'en terme de rendu, une Portlet ce n'est pas qu'une petite boîte avec un bouton "maximiser" et "fermer". Une Portlet peu être complètement transparente, le rendu ce n'est que du CSS avec un DIV, et le Portail vous permet bien entendu de ne pas afficher la petite boîte autour d'une Portlet. D'ailleurs je trouve dommage de voir par défaut un habillage de ce style dans Liferay, c'est trop réducteur. Donc retenez qu'une Portlet est un proxy du côté du serveur qui est capable de gérer des événements pour les faire passer à d'autres Portlets. Du côté de l'enveloppe c'est le Portail, donc son thème ou sa skin. Et à l'intérieur de la Portlet, c'est vous avec votre technologie Web : Wicket, JSF, GWT ou JSP par exemple. Bien entendu si vous voulez que votre Portlet ait le look de ses voisins, il faut alors utiliser les feuilles de style du Portail. Mais Liferay par exemple est assez bien fait car il est capable d'habiller votre Portlet si vous ne surchargez pas trop les styles. eXo Portal aussi, le résultat est similaire des deux côtés. Les slides suivants parlent d'architecture. Une Portlet est une mini-application Web embarquée qui ne peut vivre que dans un Portail. Ensuite, il est intéressant de voir que l'on peut soit faire une application complète avec accès à une base de données, soit une Portlet plus simple qui consommerait des services distants (EJB, REST), soit simplement une Portlet de réaffichage d'un contenu HTML distant (clipping, web proxy et WSRP). Quoi tu veux faire une Portlet avec une IFrame ? Oui monsieur, c'est possible. Imaginez un produit en fin de vie qui tourne sur un autre serveur Web. Avec ce système de clipping il est possible d'aller chercher des pages Webs distantes pour les afficher dans le Portail. Bien entendu ce n'est pas très élégant, mais ça marche. WSRP est une spécification qui permet à 2 portails de s'échanger du contenu. Un portail expose son contenu avec du WSRP et un autre portail consomme ce contenu. L'usage que je vois pour l'instant est le suivant : imaginons que votre architecture est découpée en plusieurs tiers. Une DMZ Internet avec les serveurs Webs et une deuxième DMZ avec les serveurs d'applications. Entre les deux, des EJB. Bref lorsque vous débarquez avec votre Portail, vous voilà bien embêté. Une idée serait d'installer un Portail dans la zone Internet, et un deuxième Portail dans la zone Service, et de faire communiquer les deux via WSRP. Je ne vous cache pas que cette architecture au final ne résout pas la sécurité, mais permet d'alimenter le portefeuille de l'éditeur de logiciel et du constructeur de machine. Mais bon, lorsque l'on ne peut pas réécrire d'anciens services à la sauce EJB 2.1, je dis : pourquoi pas ? Voyons maintenant le marché des Portails Open-Source :



La suite présente ensuite le projet [Jahia](#) qui est un gestionnaire de contenu très bien fait, à la Joomla en PHP par exemple. Je le testerai pour vous en reparler. JBoss Portal ensuite, est la solution de Portail de RedHat JBoss. Un peu austère, mais avec un moteur technique vraiment très puissant, il manquait à JBoss une couche applicative. eXo Portal par la société eXo Platform ensuite. Avec une centaine de collaborateurs, eXo est présent sur ce marché depuis de longues années. Membre du JCP sur la JSR-168 et la JSR-286, comme JBoss, les équipes d'eXo proposent en plus des solutions applicatives prêtes à l'emploi et modulaire. J'ai appris que [Sacha Labourey](#) (ex JBoss) avait rejoint le Board of Advisors de l'équipe d'eXo Platform ([voir le communiqué de presse ici](#)). Quand on sait qui est Sacha, je peux vous dire que c'est un peu comme si David Beckham venait jouer dans votre club de foot. C'est juste énorme. Il est certain que l'on va entendre parler d'eXo Platform dans les mois qui viennent. Bertrand Pinel parle ensuite du projet GateIn, qui est le rapprochement des équipes de JBoss Portal et des équipes d'eXo Portal. A priori d'après les rumeurs, la version Beta 2 de GateIn devrait sortir d'ici quelques semaines. Les équipes travaillent en ce moment à fond, et je crois que je le testerai pour vous en reparler un peu plus tard. Liferay ensuite est présenté. Avec 90 000 téléchargements par mois, c'est clairement un acteur très important, très populaire. Ippon Technologies a une très bonne expertise sur LifeRay, Geoffroy Gruel était la semaine dernière en Allemagne au Symposium Liferay. Installation facile, communauté importante, LifeRay manque un peu de présence en France par rapport à eXo Platform, mais se rattrape avec la taille de la communauté. Bref c'est une solution à étudier. Un point où j'émet une réserve, c'est le côté "tout-ou-rien" où l'on installe LifeRay, puis ensuite on découvre un peu les Portlets, mais on ne parle pas de produits. J'ai compris que chez eXo Platform, il y a de réels produits, avec des chefs de produits comme Patrice Lamarque qui était présent. C'est intéressant car cette approche produit est plus ouverte. Là où LifeRay c'est la suite Office, eXo semble proposer des briques selon vos besoins. Bon je jette un peu trop de fleurs sur eXo Platform alors je vais dire un truc négatif : je trouve que LifeRay est mieux terminé au niveau de l'interface utilisateur qu'eXo Portal 2.5. Plus utilisatif. Maintenant avec le nouveau GateIn, j'ai aussi l'impression que dans les mois qui viennent, les gens d'eXo vont repasser devant. Les points forts de Liferay ensuite : seul portail open-source cité en 2008 par le Gartner, sa licence MIT permet de changer le code, sans être tenu de le reverser dans le domaine public. eXo Portal est distribué en licence [Affero GPL](#). Vous pouvez changer le code, le partager, le distribuer, mais je crois que vous êtes tenu de le faire. Je ne sais pas vraiment la différence, mais Bertrand Pinel semble dire que Liferay vous permet plus de faire ce que vous voulez qu'eXo. **Les cas d'utilisation** Cette partie présente un retour sur expérience avec quelques exemples de projet type réalisé avec des Portails: - Intranet usine à sites - Bureau Virtuel - Internet de Mutualisation L'intranet usine à sites est un projet orienté contenu, qui vise à faciliter la communication dans l'entreprise, entre différents départements. Le Portail offre la possibilité d'agréger du contenu. Il propose aussi de travailler à plusieurs équipes, de construire du contenu commun. C'est donc un projet piloté par le contenu. Bertrand rappelle l'importance des comités de pilotage et des équipes d'encadrement. C'est le projet réalisé avec eXo Portal et eXo WCM par les équipes de GlobeCast que nous verrons ensuite. Le Bureau Virtuel est un projet où l'objectif est de construire un espace de travail pour le collaborateur. Chacun peut piocher des appliquettes dans une liste d'application pour les placer sur son bureau virtuel. Ici c'est la possibilité de personnaliser qui est mise en avant. C'est donc un projet piloté par les services que l'on souhaite offrir. J'aime beaucoup cette idée, où l'utilisateur final se construit son espace de travail. Dans la Finance prenons un opérateur du Front-Office qui traite sur le marché Européen le matin puis les US l'après-midi. Grâce au portail il peut se construire un espace dédié pour sa vue du matin, puis basculer vers un autre environnement l'après-midi. Je me souviens aussi d'une présentation d'Oracle qui, pour la Société Générale, a proposé de construire un bureau virtuel pour les conseillers en Agence. Lorsque vous viendrez en agence, une fois assis le conseiller aura une vue qui se sera personnalisée selon votre profil. Vous avez besoin d'un crédit à la consommation ? Une portlet s'affiche pour que le conseiller puisse vous faire une offre. Et c'est un crédit auto ? Une deuxième Portlet pour l'Assurance voiture apparaît alors... C'est donc un bureau qui s'adapte aussi à ce que vous êtes entrain de faire. J'adore l'idée d'avoir un logiciel qui s'adapte à l'usage. Oui c'est de l'ergonomie, mais c'est un truc qui coûte une fortune

avec une application Web classique, bien rigide, et bien incapable de s'adapter nativement à vous, l'utilisateur. Le troisième type de projet est l'intégration d'applications hétérogènes. Prenez un SI d'entreprise : un moteur Documentum, un annuaire LDAP sur Active Directory Server, un agenda sur Lotus Notes, le Portail est capable d'agrèger des contenus différents, provenant de sources diverses. Evidemment, ces connecteurs sont spécifiques à chaque Portail. Mais Documentum propose ses propres portlets standards par exemple. C'est donc un projet plus piloté par l'infrastructure pour le coup. L'exemple d'un site mutualisé qui regroupe différents départements est pertinent, c'est ce que je fais en ce moment. Comment regrouper sous une même bannière des applications Webs différentes ? Comment proposer plusieurs services à un seul utilisateur final ? Un Portail est donc aussi un moyen de présenter à un seul endroit les services de son SI ([voir aussi cet article](#)) Ce que je retiens c'est que certains projets de Portail permettent aussi de refondre le SI, en pilotant le développement de l'architecture par la couche de présentation. Oui je sais c'est bien capilo-tracté (tiré par les cheveux). Bon si vous voulez que l'on en parle vous m'appellez. Cela me donne envie de préparer une présentation pour l'USI pour 2010 par exemple. Va falloir décanter ces idées pour vous les présenter correctement. **Témoignages de GlobeCast et FranceBillet** Pour revenir au séminaire, nous avons ensuite assisté à 2 présentations de 2 clients d'Ippon Technologies. GlobeCast tout d'abord qui est une filiale de France Telecom, a présenté un projet d'Intranet réalisé avec eXo Portal et [eXo WCM](#). La deuxième présentation était un projet de portail LifeRay business-to-business par FranceBillet, la billetterie du groupe Fnac, qui a de grosses références comme le concert de U2 dernièrement par exemple. Nous avons terminé la matinée par une séance de questions/réponses. J'ai noté quelques questions autour de la sécurité. [LifeRay a des failles de sécurité](#) de type [XSS](#) connues et donc rapidement corrigées. C'est l'avantage de l'open-source. Mais donc attention avec la sécurité, un monsieur sécurité chez un éditeur ne serait pas une mauvaise idée je pense. Ensuite concernant GWT j'ai demandé si l'une des 2 équipes avait fait un projet avec du GWT. Bertrand Pinel cite avec LifeRay le projet [Vaadin](#). Après enquête en effet j'ai trouvé un exemple avec une application GWT qui propose un PortletContextListener afin de pouvoir faire suivre sous forme d'événements ce qui vient du monde GWT. Bon, allez je vous vois venir : je ferai un article là dessus un peu plus tard pour présenter ce principe. A vue de nez je pense même qu'il sera possible de faire aussi une version eXo Portal facilement. Voir [les articles sur le site de Vaadin](#) sur ce sujet. **Conclusion** Grâce à Ippon Technologies, nous avons eu encore une bonne présentation sur le thème des Portails et des Portlets. Les témoignages ont surtout mis en avant le besoin d'une organisation et d'une structuration pour développer un projet avec un Portail. Autant la première partie était technique, avec une présentation à la JUG, autant j'ai aussi bien aimé ensuite la seconde partie nous expliquant la démarche projet. J'ai rencontré au buffet d'autres personnes qui étudient aussi les portails. Nous avons eu aussi la chance de bien discuter avec Patrice Lamarque d'eXo Platform. La nouvelle version du moteur d'eXo s'appelle GateIn. Fruit du rapprochement entre JBoss et eXo Platform, il est très prometteur. Je pense que nous aurons une nouvelle génération de Portails d'ici quelques mois. Les Portails version 2010... Je vois bien une révolution où l'utilisateur est mis au centre de la nouvelle architecture de l'information. Pensez au développeur, à l'administrateur, au chef produit, à la responsable marketing, à l'utilisateur final... Le Web sera incontournable, la personnalisation aussi. Il ne manque plus que le Portail Mobile, léger et adapté au format de nos petits écrans, des outils avec une ergonomie très poussée pour que tout le monde puisse s'en servir. Le premier qui proposera un PortailOS aussi bien que mon iPhone aura juste gagné. Je pense à un OS qui marcherait bien entendu sur mon navigateur standard, pas spécifiquement sur mon mobile. Le premier qui proposera un site marchand avec des Portlets ou des Gadgets aura aussi gagné. Un annuaire des Gadgets et des Portlets en quelques sortes. Le premier qui aura des outils d'administration standards pour des administrateurs, des responsables de campagne marketing, des développeurs aussi pour construire des pages rapidement, celui-là aura aussi gagné. Le premier... on est pas loin je pense.

**Références** Les slides de la présentation sont [sur le blog d'Ippon Technologies](#).

[Seminaire Portail Open Source](#)

View more [documents](#) from [Ippon Technologies](#).

## Scrum ne fait rien par Tobias Mayer

Je vous propose la traduction en Français de l'article "Scrum doesn't do anything" de Tobias Mayer après avoir reçu son accord. L'article original pour ceux qui préfèrent la version en Anglais et pour suivre les commentaires se trouve [sur le blog de Tobias](#). Merci à Tobias de m'avoir autorisé à traduire son article et merci à Sébastien Douche pour m'avoir indiqué l'article

*original sur Twitter. This article is a French Translation of the original "Scrum doesn't do anything" wrote by Tobias Mayer. Published with his agreements, thanks again to Tobias for this very good article. Please see : <http://agileanarchy.wordpress.com/2009/10/11/scrum-doesnt-do-anything/> Tobias Mayer: "Faire du Scrum" est aussi vide de sens (et impossible) que de créer une instance d'une classe abstraite. Scrum est un cadre pour mettre en avant les dysfonctionnements d'une organisation. Ce n'est pas un processus et il n'est pas prescriptif. Le cadre de base de Scrum, tel que décrit (par exemple) [ici](#) et [ici](#), ne fait pas en effet grand chose. Il est, en quelque sorte, un contrat que nous mettons en place entre ceux qui recherchent de la Valeur et ceux qui la construisent. Mais un contrat ne produit rien. Une interface est passive. Il nous faut une implémentation. Scrum commence à fonctionner quand les gens le comprennent suffisamment pour en créer une mise en application concrète, pour ensuite faire émerger un processus qui leur est propre. Comprendre et respecter Scrum ainsi que l'ensemble des règles permet de s'aligner avec les autres et d'avoir un ensemble commun de valeurs et de principes à partir duquel travailler. Scrum lui-même peut être considéré comme analogue à un ensemble de règles de jeux. Pensez aux échecs, pensez au football. Les règles sont faciles à apprendre, et les connaître vous permet de jouer avec les autres. Les règles sont donc les liens qui nous unissent. Mais bien que vous connaissez les règles, ce n'est pas pour autant que vous tirez du plaisir de la partie [d'échec ou de foot]. Vous aurez besoin de jouer pour être impliqué. C'est donc jouer, et pas simplement connaître la règle qui apportera un résultat. D'ailleurs pour bien jouer, vous devez élaborer une stratégie, et dans les jeux d'équipe dont vous avez besoin de développer une relation de confiance et de partage avec les autres. Dans n'importe quel jeu, dès lors que vous commencez à ne plus suivre les règles, tout s'écroule, et plus personne ne voudra alors jouer avec vous. Et bien c'est exactement la même chose avec Scrum. Respecter le contrat, les règles, ou dans le langage des logiciels, respecter l'interface, vous permet de créer un processus qui convient à votre contexte - et votre contexte est beaucoup de choses: les gens, l'industrie, des entreprises, une place du marché, un produit, une localisation, une langue, l'environnement physique, la culture, les gens, etc. Il commence et se termine avec les gens. Scrum en fait ne fait rien, ce sont les gens qui font des choses. Le processus Scrum émerge à travers l'interaction entre les gens, et il sera différent pour chaque organisation et chaque équipe. Et c'est là que les gens trébuchent, trop nombreux à penser que la création de leur propre processus commence par briser les règles. C'est faux ! Faire cela, et vous aurez échoué avant de commencer. (Note de Nicolas : avant d'adapter Scrum, suivez simplement ses principes, votre équipe évoluera ensuite) La mise en place initiale de Scrum doit s'effectuer avec des paramètres convenus, elle doit être délimitée par des règles. Toutes les méthodes de la Classe abstraite doivent être implémentées, ou la classe ne se compilera pas et ne sera pas exécutée. Si vous essayez de déplacer votre Fou dans un jeu d'échec en ligne droite, alors votre adversaire quittera la table. Si vous voulez ramasser le ballon de football avec les mains sur le terrain et le jeter, vous prendrez un carton rouge. Le cadre de Scrum est un mécanisme puissant pour mettre en avant les dysfonctionnements d'une organisation et d'une équipe. C'est ce pouvoir même qui pousse les gens à essayer de contourner les règles et à prendre des raccourcis. Ils ne veulent souvent pas regarder. Mais ne pas chercher la cause d'un problème ne fait pas disparaître celui-ci, ou ne réparent pas ce qui est cassé, cela retarde simplement le moment où l'on échouera de manière inévitable. Nous nous en remettons pas pour rien à la valeur de Courage dans Scrum. Essayer de comparer Scrum à des pratiques d'implémentations comme XP ou de l'artisanat logiciel apporte peu de sens. Chacun de ces mouvements offre par ailleurs d'excellents outils pour mettre en place une pratique solide de Scrum, pour rendre concret l'abstrait. Scrum offre un cadre solide et un rythme implacable pour secouer la poussière des années [sur un projet]. Vous ne pouvez pas "faire du Scrum" mais vous pouvez certainement vous y embarquer, et le plus courageusement vous le faites, plus fort sera votre processus émergeant." FIN DE LA TRADUCTION*

## IDEA IntelliJ version Community : oui c'est gratuit et libre

[JetBrains](http://jetbrains.org/) l'éditeur d'IDEA IntelliJ propose désormais une version Community en version open-source : <http://jetbrains.org/>. Pour rappel, IDEA IntelliJ est un éditeur Java qui se distingue d'Eclipse grâce à l'intégration d'outils d'analyse du code, de refactoring, un éditeur rapide et simple, le support de Groovy, de Scala, de Google App Engine, d'Android, le tout avec des plugins pour les rares fonctions manquantes. La version Community Edition permet de découvrir le cœur d'IDEA IntelliJ simplement. La version Ultimate Edition est proposée pour 205\$ si vous êtes un développeur indépendant comme moi. Voici un tableau résumant les différences entre les 2 versions :

	- Supporté	- Non packagé mais dispo via un plugin gratuit	- Non supporté
	Community Edition		Ultimate Edition
<b>IDE Features</b>			
Local history	yes		yes
Maven	yes		yes
Ant	yes		yes
JUnit integration	yes		yes
TestNG integration	yes		yes
GUI builder	yes		yes
Code spell checker	yes		yes
Eclipse project interoperability	yes		yes
Java ME support	yes		yes

IDE talk	yes	yes
Task management	no	yes
UML class diagrams	no	yes
Code coverage	no	yes
Dependency structure matrix	no	yes
Android	no	yes
Code duplicates detection	no	yes
Structural search & replace	no	yes
Source level JSP debugging	no	yes
JavaScript debugger	no	yes
<b>Language Support (refactoring, error highlighting, code inspections, intention actions, quick-fixes, etc.)</b>		
Java	yes	yes
Groovy	yes	yes
XML	yes	yes
Regexp	yes	yes
Scala	yes	yes
Ruby/JRuby	no	yes
SQL	no	yes
HTML, XHTML, CSS	no	yes
XSL, XPath	no	yes
Freemarker/Velocity	no	yes
JavaScript/ActionScript	no	yes
PHP	no	yes
<b>Version control systems</b>		
CVS	yes	yes
Subversion	yes	yes
Git	yes	yes
Team Foundation Server	no	yes
ClearCase	no	yes
Perforce	no	yes
StarTeam	no	yes
Visual SourceSafe	no	yes
<b>Frameworks and Technologies</b>		
OSGi	yes	yes
Grails	no	yes
Rails	no	yes
JSP	no	yes
Servlets 3.0	no	yes
JavaServer Faces 1.0, 1.2	no	yes
EJB 3.0, 3.1	no	yes
Web Beans	no	yes
Bean validation	no	yes
Flex, Air	no	yes
GWT 1.X, 2.0	no	yes
Struts 1.2, 2	no	yes
Spring 2.5, 3.0	no	yes
JBoss Seam	no	yes
Hibernate	no	yes
JPA 1.0, 2.0	no	yes

JAX-WS, Apache AXIS	no	yes
REST WebServices	no	yes
Tapestry	no	yes
Google App Engine	no	yes
<b>Application Servers</b>		
Apache Tomcat	no	yes
GlassFish v2, v3	no	yes
JBoss	no	yes
WebLogic	no	yes
WebSphere	no	yes
Geronimo	no	yes
Resin	no	yes

Pour télécharger la version Community, rendez-vous sur le site <http://jetbrains.org/>

## Retour sur la soirée JSF 2.0 et Servlet 3.0

Mardi dernier [au Paris JUG](#) c'était une soirée placée sous le thème JEE6 avec une présentation de Java Server Faces 2.0 et de la spécification des Servlets 3.0. Damien Gouyette et François Petitit ont présenté les dernières évolutions de JSF 2. Remy Maucherat de JBoss RedHat a présenté la spécification des Servlets 3.0 **Retour express** Olivier Croisier de Zenika, auteur du blog "[The Coder's Breakfast](#)" a publié [un compte-rendu détaillé](#), ainsi que [Mathieu Petitdent](#) d'Object Direct. Du coup je ne vois pas trop l'utilité de vous resservir le même contenu, donc pas de retour cette fois-ci pour moi. **Emmanuel Bernard au Paris JUG** La soirée a été marquée cependant par la présence d'[Emmanuel BERNARD](#), de JBoss RedHat, pour la première fois. Auteur du livre [Hibernate Search in Action](#), et fondateur du Podcast "[Les CastCodeurs](#)", Emmanuel a surtout annoncé la sortie le 8 octobre dernier d'[Hibernate Validator 4.0](#). Il s'agit tout d'abord de l'implémentation de référence [de la JSR-303](#). Plutôt que de dupliquer du code de Validation au niveau de la vue, de la base de données ou des services métiers, l'idée de la JSR-303 est de proposer une validation au niveau du POJO. La JSR-314 (JSF 2.0) propose les Validators, mais plutôt que de dupliquer ce code de validation, l'idée de la JSR-303 est de proposer à JSF 2.0 un moteur de validation. De même avec JPA 2.0. A priori en décembre Emmanuel Bernard devrait nous en parler plus précisément, avec des exemples de code. Rendez-vous donc à la fin de cette année. **La troisième et la quatrième mi-temps** Changement de lieu, nous étions au Vavin. Service très pro, ambiance sympa et un peu bruyante. C'est toujours un plaisir de discuter avec tout le monde, de voir des têtes connues de la communauté parisienne, bref très sympa. Si vous voulez un teeshirt dédié d'Arnaud Héritier sur Maven2, une casquette JEE6 par Antonio Goncalves, ou un caleçon eXo Platform par Julien Viet, il faut venir au Paris JUG. A propos des enrouleurs et du kakimono. Mardi soir nous avions 3 grands enrouleurs pour mettre en valeur nos sponsors Platinumium. Même si j'ai réalisé les montages avec Adobe Illustrator, c'est grâce à l'aide de Karine VACCA [de l'agence 3-2-1 idCom](#) que nous avons eu de beaux enrouleurs, livrés à temps et de bonne qualité. Donc si vous cherchez une agence sympa pour faire de la communication, du marketing, organiser un événement comme une formation, ou un séminaire, contactez de ma part [3-2-1 idCom](#) Travail sérieux et professionnel, encore merci pour son aide. **Articles du blog le Touilleur Express connexes** Vous pouvez relire 2 articles sur Emmanuel Bernard : - [Présentation à Jazoon sur Hibernate Search](#) article de juin 2009. - [JSR-303, avez-vous validé votre Bean ?](#) article de juin 2008.

## Indépendant depuis 14 mois, le bilan

[caption id="attachment\_2111" align="alignnone" width="500" caption="Crédit photo: Guillaume Brialon licence Commons Creative"]



[caption] C'est passé vite... Vraiment. Voilà plus d'un an que je travaille en tant qu'indépendant. J'ai débuté mon activité

en septembre 2008, par la création d'une EURL (Innoteria), une mission longue chez un client et quelques missions de conseils ponctuelles en plus pendant l'année. J'ai pensé qu'il serait intéressant de vous parler de ce qui a vraiment changé, par rapport à ma vie de salarié. D'un point de vue charge de travail tout d'abord, j'ai travaillé 217 jours exactement depuis un an. Ce chiffre ne vous dit peut-être rien, mais cela correspond à votre temps d'activité si vous êtes salarié aux 35 heures et que votre entreprise fait partie du [SYNTEC](#). Ce doit être le cas si vous êtes en SSII. Bref je n'ai pas plus travaillé que lorsque j'étais salarié. **34 jours à ne pas travailler (donc facturer)** En réalité, c'est un peu différent. Cette année j'aurai pu ajouter 34 jours de plus chez mon client, soit 251 jours au total si je n'avais pris un seul jour de vacances. Dans les faits j'ai pris autant de vacances que l'an passé, auquel s'ajoute plus de temps pour participer à des conférences : - Devoux en Belgique en novembre 2008 ([voir les 10 articles](#)) - XPDays à Paris en mai 2009 ([article 1](#) et [2](#)) - Jazoon à Zürich en Suisse en juin 2009 ([voir la page complète](#)) - L'USI 2009 ([11 articles](#)) Merci le statut d'indépendant, je n'ai jamais autant participé à des conférences, ce qui m'a aussi donné pas mal de travail pour le blog par la suite. Je vois cela finalement comme à la fois de la formation continue, je peux être au courant des dernières nouveautés autour de la plate-forme Java, savoir qui est qui, rencontrer et interviewer des personnes passionnantes. **Etre indépendant c'est devenir dépendant des autres** Sans vraiment faire de la prospection active, je conseillerais à celui qui veut devenir indépendant de penser qu'il devra investir une journée par semaine pour sa communication. Commencez par faire des cartes de visite de qualité, construisez un site Internet si celui-ci peut vous apporter des contacts, essayez de vous trouver un nom sympa, pensez aussi à l'élévator pitch et enfin, participez à quelques soirées, quelques conférences. Il est important de se construire un réseau, basé sur la confiance et l'amitié. Faire des cartes de visite, facile vous savez faire. [L'élévator pitch](#) c'est votre manière de vous présenter à un inconnu, en 2 minutes, pas plus. La version zipée de votre CV. Cela donne quelque chose comme cela: *"Bonjour je m'appelle Nicolas. Après quelques années chez des éditeurs dans le monde de Java et de la Finance, je suis maintenant indépendant. Je termine une mission d'architecture et de gestion de projet en méthode Agile pour une grande banque. Je travaille sur une étude sur les portails open-source en ce moment. Je suis passionné par Java, assez jeune pour avoir encore envie de coder, assez vieux pour ne plus faire les mêmes bêtises qu'avant, et justement libre à partir de janvier pour venir travailler avec vous."*

**... Bah alors, tu viens plus aux soirées ? ... (Omar et Fred)** Ensuite si vous aimez bien coder dans votre coin, que sortir en semaine pour vous c'est une hérésie, et bien bon courage. Là vous allez vraiment être indépendant de chez indépendant : seul dans votre cuisine à tourner en rond... Etre indépendant c'est devenir très dépendant des autres. Et bien oui, plus de commercial qui vend votre petit nez à un client, c'est vous mon garçon qui allez devoir vous vendre. Et ça, c'est pas franchement un truc qui fait partie de notre OS de base, à nous les informaticiens. Si je n'avais pas connu le Paris JUG alors que j'étais encore salarié, je ne sais pas si je serai devenu indépendant. Car le Paris JUG (merci Antonio) c'est un peu le repère de la crème de la crème des bonnes brutes de Paris. C'est là qu'il est possible de rencontrer d'autres indépendants, et donc d'y trouver de bons conseils. C'est aussi sympa de se croiser ensuite dans notre milieu professionnel. **Redevenons sérieux** Bon comme je sens que vous avez pleins de questions, je vous ai préparé 2 billets en plus de celui-ci. Un article sur le statut juridique : lequel choisir ? (ouais super...) et un autre sur "Les revenus d'un informaticien en EURL" que je ne publierai que mercredi prochain. Je fais un bilan positif. La liberté gagnée lorsque l'on devient indépendant demande en retour un travail administratif et comptable. Il faut aussi penser à créer son réseau de partenaires et de clients, et à se tenir au fait des demandes du moment. Je pense qu'il est très important de bien communiquer. Et enfin il ne faut pas se faire griser par les gros chiffres que vous allez voir passer. Vous allez faire aussi de beaux chèques dans l'autre sens, croyez-moi. Donc si vous pensez devenir indépendant, ou si vous vous intéressez à ce sujet, j'ai 2 articles qui seront publiés automatiquement cette semaine après celui-ci. Je sors un peu de la thématique Java et Scrum pour parler des indépendants, mais je vous rassure, je vais commencer un gros boulot et écrire beaucoup, beaucoup... et pas forcément sur le Touilleur Express... (et là il y a une grosse musique de Suspens, vous sursautez et il y a marqué FIN) A demain pour le premier article sur les statuts juridiques, et après-demain pour un retour sur les revenus d'un indépendant en EURL. Dans cet article je fais un petit bilan de ma première année, pour donner aussi quelques conseils à ceux qui souhaitent se lancer.

## Les différents statuts juridiques pour les indépendants en informatique



Suite de la série d'article consacré au statut d'indépendant, [après un bilan hier](#). Devenir indépendant c'est tout d'abord faire le choix d'une structure juridique. J'ai opté pour l'EURL, je vais vous expliquer en quoi cela consiste, et mes motivations. Avant tout, quelques mots sur les différents statuts, car il y a pas mal de confusion. 1) **Le portage salarial** Depuis 2008 ce statut est intéressant, surtout si vous ne savez pas encore si vous comptez vous lancer vraiment ou non. La loi [2008-596](#) reconnaît le portage salarial, vous

permet donc de conserver un statut de salarié, tout en prospectant vous-même les clients. Vos revenus sont donc rangés dans la case "traitements et salaires". Pour 100 EUR facturé, il faut retirer les charges sociales, les charges patronales et les frais de la société de portage (autour de 10%). Au final [il vous reste 49 EUR environ](#). L'avantage de cette formule : vous pouvez démarrer dès lundi, aucunes créations d'entreprises, assez proche de l'intérim dans l'idée. L'inconvénient : est-ce que votre société de portage est référencée chez le client au niveau des achats ? Les grands comptes ne travaillent pas toujours avec les sociétés de portage. D'un point de vue fiscal, vous ne pouvez pas moduler votre revenu. Enfin rendez-vous compte que 100 000 EUR de chiffre d'affaire, c'est une facturation à 400 EUR HT et 251 jours d'activités. Important pour commencer à avoir quelques chiffres en tête. **2) L'Entreprise Individuelle** Aucun apport en capital obligatoire, il suffit pour les informaticiens qui font du conseil, de s'enregistrer à l'URSSAF. Aucune comptabilité exigée, vos revenus professionnels sont à déclarer dans la catégorie des bénéficiaires non commerciaux (BNC) sur votre feuille de déclaration de revenu. Du côté de votre protection sociale, vous êtes travailleur indépendant non salarié : pas d'assurance chômage, ni d'assurance accident du travail. Vous devez cotiser pour la retraite car en étant rattaché au régime de base, c'est très insuffisant. Je me demande aussi au niveau fiscal, s'il n'y a pas un effet massue. Imaginez inscrire 100 000 EUR dans la case BNC de votre déclaration de revenu... Le point sensible de ce statut est que vous êtes engagé financièrement de manière illimitée. Il n'y a pas de protection de vos biens propres par rapport à votre activité. Notez que dans notre secteur, l'informatique, je ne pense pas que cela soit trop grave. Mais bon, si vous avez un pépin financier, vos biens ne sont pas protégés. Il est possible cependant de les sécuriser, et il est conseillé de changer de régime matrimonial si vous êtes marié. Un peu compliqué. Côté revenu, on estime à 30% les charges, mais à 20% au moins l'impôt sur le revenu. Tout ceci est relatif. Donc en théorie (vérifiez par vous même) vous pouvez toucher 40 EUR pour 100 EUR facturé, le tout dépend de votre nombre de part par exemple. **3) Auto Entrepreneur** A mon avis, la FBI de l'année (Fausse Bonne Idée). Même statut juridique que l'EI (Entreprise Individuelle). Il faut que votre chiffre d'affaire ne dépasse pas 32 000 pour les services. Autant dire que ce statut n'est adapté que si vous pouvez continuer à être salarié dans votre SSII par exemple. 32 000 EUR de chiffre d'affaire, ce serait une facturation à 127 EUR HT par jour chez votre client. Laissez-moi vous dire que si vous êtes développeur avec 5 ans d'expérience, vous êtes actuellement dans une fourchette entre 300 et 450 EUR HT la journée. Donc à mon avis, ce statut vaut le coup si vous conservez une activité au 4/5ème par exemple et que vous voulez faire quelques missions courtes en plus. Attention par contre, votre contrat de travail peut vous interdire d'exercer une activité supplémentaire proche de votre activité actuelle. Parlez-en à votre DRH avant de vous lancer. **4) EURL** C'est une entreprise unipersonnelle à responsabilité limitée. La totalité des parts de cette entreprise est détenue par un Gérant: vous. C'est une vraie personne morale, enregistrée au RCS avec dépôt des statuts, une comptabilité et des comptes déposés chaque année. La responsabilité du gérant, sauf fautes graves, est limitée au montant des apports en capital. Je vais pouvoir vous en parler un peu plus, puisque c'est le statut juridique de mon entreprise. Tout d'abord : vous pouvez embaucher des salariés. Pensez au plombier ou à l'artisan maçon en EURL : il a des salariés. L'EURL permet donc d'embaucher normalement des personnes. Les bénéficiaires de l'entreprise sont soit soumis à l'IR, soit à l'IS. Vous pouvez passer de l'IR à l'IS, mais pas l'inverse. Tout le monde conseille de passer tout de suite à l'IS (Impôt sur les sociétés). Je vais vous expliquer un peu cela avec des chiffres pour mieux comprendre. A l'IS, l'associé ne déclare sur sa déclaration personnelle, que le revenu que lui a versé l'EURL pour sa gérance. Le dividende est ensuite ce qui reste à l'entreprise une fois toutes les charges payées, et une fois votre salaire de gérant majoritaire payé. Attention, vous n'êtes pas salarié. J'ai donc un numéro à l'URSSAF en tant qu'indépendant, gérant d'entreprise, avec un code SIREN. Mon entreprise d'autre part, a un autre numéro de SIREN, et elle me verse une indemnité de gestion, on appelle cela "le salaire de la gérance". En fin d'année, le comptable dresse le bilan financier de l'activité. S'il vous reste de l'argent (et il vous en restera) vous pouvez alors déclarer le versement d'un dividende. Pour moi, mes comptes sont clôturés le 31 décembre. J'ai effectué le paiement du dividende en août 2009, vous pouvez avancer bien entendu ce paiement, ce sont des détails qui se règlent facilement. Alors pourquoi je vous parle de dividende ? Et bien parce que cela vous permet de moduler votre fiscalité. Au lieu de toucher 100 EUR en salaire par exemple, vous pouvez vous verser 60 EUR de salaire, et garder 40 EUR que vous vous verserez plus tard sous la forme d'un dividende. Vous suivez ? En effet, le taux d'imposition des dividendes n'est pas le même que le taux d'imposition sur vos revenus, votre salaire. Et il est donc intéressant de ne pas se verser un salaire de 100 EUR. Si vous faites cela, votre entreprise va payer beaucoup plus de charges sociales, qui sont déjà assez élevées. [Le régime d'imposition est assez compliqué](#). L'avantage de cette solution est donc la possibilité d'ajuster ses revenus par rapport à sa fiscalité personnelle. Bien entendu cela prend du temps, il faut se faire aider par un fiscaliste. Enfin bien entendu, un dividende veut dire que votre entreprise a fait un bénéfice. Les autres régimes possibles sont la SARL, la SAS et la SASU. En SARL vous êtes associé avec d'autres personnes. La SAS est la société par action simplifiée ([voir cet article](#)). C'est un mix entre une Société Anonyme et une Société à responsabilité limitée. Cette structure est très en vogue depuis 10 ans, car elle offre une structure solide où le fonctionnement est réglé par les statuts. Ici un avocat est presque indispensable pour créer cette forme de structure. Concernant le régime fiscal, c'est similaire à une SARL ou une EURL. Enfin la SASU (SAS unipersonnelle). La SASU enfin est une SAS unipersonnelle, avec donc un seul actionnaire principal. Cette structure est intéressante pour les personnes qui montent un projet, un éditeur de logiciel ou une startup. En effet, il permet de faire entrer au capital facilement des tiers. La rédaction des statuts par contre doit être très rigoureuse, il faut faire appel à un professionnel. A réserver si vous pensez développer votre activité fortement. L'avantage de la SAS/SASU est que cela vous permet d'être référencé chez certains grands clients, qui n'acceptent pas les SARL/EURL. Cette information est à vérifier bien entendu. **Conclusion** Le portage salarial est simple et clair, surtout si vous avez une opportunité de mission à commencer rapidement. L'Entreprise Individuelle convient bien pour les freelances, les graphistes, les développeurs, qui cherchent une structure simple tout en proposant une vraie entreprise aux clients. Enfin l'EURL est le statut qui offre le plus de liberté, avec une complexité un peu plus importante. Il existe pour cela des experts-comptables qui peuvent vous aider et réaliser pour vous certaines démarches, ce qui peut vous aider si vous ne savez pas trop comment vous y prendre. De mon côté, j'ai fait toutes les démarches tout seul, mais le temps de rédiger les statuts, de faire les

démarches, comptez un à deux mois. Je vous avais d'ailleurs expliqué cela l'an dernier dans un article énormément lu "[Se mettre à son compte](#)". Pour terminer, renseignez-vous. Je ne peux pas vous garantir l'exactitude de toutes les informations. J'espère cependant vous avoir apporté un peu d'informations. Demain je vous présenterai une simulation financière d'un indépendant en EURL. Ce deuxième article d'une série sur les indépendants, vous propose une explication des différents statuts juridiques et fiscaux.

## Les revenus d'un informaticien indépendant en EURL

[caption id="attachment\_2123" align="alignnone" width="500" caption="source: Flickr, Crédit photo Guillaume Brialon licence Creative Commons"]



[caption] Voici le

3ème article après [un bilan de mes 14 mois d'indépendant](#) et [une présentation des différents statuts juridiques](#). Jeudi je terminerai cette série d'article par un sujet sur la prospection commerciale : comment trouver ses premiers clients ? La compta pour les nuls Allez, parlons un peu de chiffres. Lorsque je me suis lancé il y a un an, j'ai trouvé qu'il était difficile de calculer et d'estimer ses revenus à venir. La comptabilité tout d'abord, ce n'est pas un truc d'informaticien. Si vous êtes allergique à une feuille Excel, que vous ne faites pas les comptes à la maison, prenez un excellent comptable pour vous aider. C'est dur, c'est ennuyant, et on est nul. L'objectif de cet article est bien de parler argent. C'est tabou, mais comme personne n'en parle et qu'il me manquait des explications il y a un an et demie, je pense qu'il est utile d'expliquer tout cela. J'ai travaillé avec [l'excellent fichier Excel d'Olivier Deheurles](#) pour vous construire un cas que je considère comme réaliste. Je vais vous expliquer le cas de Bob qui a créé une EURL BobConsulting, avec des chiffres le plus réaliste possible. Bob est un développeur Java avec 6 ans d'expérience, qui est plutôt senior. Il facture 400 EUR par jour. Bien que cela vous semble énorme si vous découvrez ce chiffre, c'est un prix correct pour un profil de ce type. Le site Freelance-info propose [une base de tarif](#) complétée par les indépendants qui vous aidera à vous situer. Pour 2008, nous allons dire qu'il a travaillé 250 jours, ce qui est plutôt exceptionnel. Cela veut dire que Bob n'a pas pris un seul jour de congé et qu'il a travaillé sans interruption toute une année. Mais cela m'arrange car je tombe donc sur un chiffre d'affaire de 100 000 EUR. Je vais vous expliquer combien Bob "gagne vraiment" le plus précisément possible. Tout d'abord, Bob est gérant majoritaire de la SARL BobConsulting. Celle-ci lui verse chaque mois un salaire net de 3500 EUR. Dans la réalité, Bob décide de se verser ce salaire le 31 ou le 3, de moduler son montant d'un mois à l'autre, car il est le gérant unique. Bob se fait ses entretiens, il s'augmente, il réduit son salaire, parfois il se verse des primes car il a bien travaillé... Bon vous l'avez compris : Bob peut moduler son salaire mensuel, sachant que le comptable en fin d'année prépare la déclaration des revenus dans le bilan de votre entreprise. En général votre comptable préfère que le salaire versé soit fixe pour se simplifier la vie. Mais bon, Bob fait ce qu'il veut pendant l'année. Ce salaire de gérance est voté en assemblée générale une fois par an, par Bob qui est l'associé unique. Les cotisations sociales (maladie+retraite+urssaf) pour ce salaire de 42 000 EUR (12x3500) seront d'environ 7500 EUR. Au final, ce salaire ainsi que les charges, coûte à l'entreprise 49 500 EUR, soit la moitié du chiffre d'affaire de 100 000 EUR. Voici les fameux 50% dont on vous a parlé... Concernant les frais professionnels, Bob paye son téléphone portable, un accès à Internet, des frais de transports, des formations, un comptable, un ordinateur, une assurance professionnelle et des repas. Disons qu'il a 4500 EUR de frais par an. Ces frais sont déductibles dans une certaine mesure, du chiffre d'affaire pour calculer le bénéfice. Les règles étant assez compliquées, votre comptable s'en chargera. La SARL BobConsulting après avoir payé 49 500 EUR de charges (salaire de Bob+cotisations sociales), doit aussi payer 1000 EUR de taxe professionnelle, auquel je retire 4500 des frais professionnels, ce qui nous donne le bénéfice imposable de la SARL, soit 45 000 EUR : 100 000 - 49 500 - 4 500 - 1 000 = 45 000 de bénéfice. Le bénéfice est donc ce qui reste à l'entreprise une fois les frais, le salaire et les charges déduites. Que devient ce bénéfice ? Il est temps de payer maintenant l'impôt sur les sociétés, qui s'élève ici à environ 8 000 EUR. Il restera au final 37 000 EUR environ. Si notre SARL a un capital de 1000 EUR, vous devez donc laisser légalement 10% de ce capital sur le compte de l'entreprise, c'est la réserve légale. SARL BobConsulting peut maintenant verser le dividende de 36 000 EUR à Bob. Il y a cependant encore 12% de ce dividende que l'EURL doit payer à l'Etat (CSG+RDS+RSI) au moment du versement du dividende. Au final Bob recevra environ 32 000 EUR de dividende. Au final si je compte le salaire de Bob et le montant du dividende, la trésorerie de Bob sera de 42 000 EUR de salaire + 32 000 EUR de dividende, soit 74 000 EUR avant imposition. Prenez un couple marié avec 2 enfants, un impôt sur le revenu d'environ 6000 EUR, il reste réellement dans la poche de Bob la somme de 68 000 EUR.

Pour un chiffre d'affaire de 100 000 EUR, Bob aura donc réellement gagné 74 000 EUR avant de payer ses impôts sur le revenu. Le salaire versé en 2008 est imposé en 2009. Par contre, le dividende étant versé en 2009, il ne sera compté dans l'impôt sur le revenu de Bob qu'en 2010. D'après la feuille de calcul d'Olivier Deheurlès, un couple marié avec 2 enfants avec un deuxième salaire payera environ 6000 EUR d'impôts sur le revenu, il reste donc bien environ 68 000 EUR de trésorerie. Chaque mois Bob touche un salaire fixe de 3500 EUR, le dividende étant ensuite versé l'année suivante, comme une prime qui arriverait 4 ou 6 mois plus tard en quelques sortes. Dans la réalité, il manque pas mal de choses. Tout d'abord une mutuelle si votre épouse/époux ne peut pas vous couvrir. Comptez au moins 150 EUR par mois pour être couvert aussi bien que votre mutuelle de salarié. Ensuite ajoutez-y des compléments de retraites. En effet, le régime des indépendants n'est pas top si vous voulez une bonne retraite. Disons 300 EUR par mois sur un contrat Madelin ou sur un placement type Assurance Vie. Ensuite, il est très important de bien calculer la répartition salaire/dividende selon son profil fiscal. Si vous êtes célibataire, l'équilibrage ne sera pas le même qu'un couple marié avec 4 enfants, où madame travaille aussi par exemple. Au final donc, être indépendant rapporte donc certainement plus qu'être salarié, mais en retour vous devez vous occuper sérieusement de votre retraite, de vos investissements. Gagner plus pour pouvoir optimiser mieux. En tant que salarié, vous êtes pris en main et vous n'avez pas trop à vous soucier de tout cela. En tant qu'indépendant : vous vous débrouillez. Quelque part il n'y a rien de magique. Le système est fait de telle sorte que ceux qui peuvent prendre un peu plus de risque, gagnent plus. Il y a cependant un énorme risque qui est celui de l'inter-contrat. Là où un salarié conserve sa sécurité (un salaire), et donc délègue à un tiers la recherche d'une mission, un indépendant doit avoir des contacts et un réseau très important. Mais surtout : il ne rentre plus un euro. J'espère avoir expliqué le plus simplement possible un cas presque réel, afin de vous faire prendre conscience des ratios. Le chiffre d'affaire de mon exemple est plutôt élevé. Les indépendants dans la réalité ne travaillent pas 250 jours par an. Cela reviendrait à ne presque pas prendre de congés. En tant que salarié aux 35 heures, nous travaillons 218 jours par an, c'est un peu près la même chose en tant qu'indépendant (maladie, vacances, ponts, enfant malade...). Voilà vous savez tout. Plongez vous dans le fichier Excel d'Olivier si vous voulez comprendre précisément les calculs. **Bonus : un 4ème article demain** La suite demain sur la recherche des clients et je crois que j'aurai pas mal parlé des indépendants cette semaine. Bonne lecture !

## Trouver ses premiers clients



Photo par HHHH sur Flickr, licence Creative Commons libre de droits

**4e article de la série sur les indépendants, après "Les revenus d'un informaticien indépendant en EURL", "[Les différents statuts juridiques pour les indépendants en informatique](#)" et "[Indépendant depuis 14 mois](#)"** **Mon premier client**

Il y a une équation qui semble impossible à résoudre : trouver un client, démissionner, trouver du temps pour faire ses démarches de création, prendre rendez-vous avec la banque, trouver un comptable et rédiger ses statuts. J'ai fait un peu tout en même temps. La création à proprement parler m'a pris 3 semaines, alors que j'avais déjà commencé une première petite mission de consultant. J'ai donc eu le temps de faire les démarches, de prendre les rendez-vous, et de commencer à trouver ma première grosse mission. J'ai ensuite commencé par le site [Freelance-info.fr](#). Une fois mon profil complété, j'ai eu le lendemain un premier coup de fil, et au final j'ai fait 3 entretiens dans différentes SSII en une semaine. C'est plus tard que j'ai été contacté par Computer Futures avec qui je travaille depuis. Ce que j'ai aimé dans leur démarche, c'est qu'ils se positionnent comme un agent sportif. Mon contact s'est d'abord renseigné sur mon profil, il n'avait pas encore de mission à me proposer. Il m'a recontacté une semaine plus tard, j'ai passé des entretiens classiques, et ma mission à la BNP a débuté en septembre. Une mission de 12 mois pour commencer, une sécurité toute relative. En janvier, crise oblige, j'ai baissé ma facturation, et mon contrat s'est transformé en contrat renouvelable de 3 mois. Nous sommes en octobre, et comme vous le voyez, tout va bien. J'ai signé à chaque fin de trimestre un nouveau contrat commercial entre Computer Futures et mon EURL. La facturation est simple, hors-taxes s'agissant d'une entreprise anglaise. Ma société est payée à 30 jours. Je donne une facture le 31 janvier et je suis payé le 1er mars. Aucuns soucis de paiement, car Computer Futures est très sérieux. Voilà pour mon retour sur expérience. **Comment trouver des clients ?** Finalement, je me demande plutôt si vous ne devez pas tout faire pour que les clients vous trouvent... Que ce soit grâce à des contacts réguliers avec plusieurs SSII, en participant à des soirées ou des événements, vous devez investir du temps pour vous faire un nom. J'ai peur qu'il soit difficile de trouver réellement un client tout seul. Vous allez envoyer votre CV et votre plaquette à toutes les entreprises sur Paris ? Le plus sage est de faire appel aux entreprises qui sont référencées chez les grands comptes, et qui savent trouver des missions : les SSII. Toutes n'acceptent pas de travailler avec des indépendants. Et elles préfèrent placer leurs salariés en inter-contrat plutôt qu'un indépendant. Mais si votre profil est intéressant, et il l'est, vous pourrez trouver une mission avec eux. **Concernant les prix, qui négocie quoi ?** Attention, exercice difficile où vous devez être prêt. La question tombe très rapidement dans la conversation : quel est votre taux journalier ? A vous de répondre simplement et précisément. Votre interlocuteur connaît les prix du marché. En général, un intermédiaire comme une SSII ajoute entre 15% et 20% à votre taux pour facturer le client final. Si vous proposez 300 EUR, vous serez facturé facturé 345 EUR HT par jour au client final par exemple. Et en général, la discussion se fait à l'euro prêt. Préparez-vous à négocier et à ne pas vous laisser

impressionner par votre interlocuteur. Les commissions peuvent sembler élevées, mais si je me place du côté de la SSII, il y a le risque que l'indépendant décide de quitter la mission. C'est un argument un peu bancal à mon avis. Un consultant en SSII en mission qui s'ennuie ferme demandera aussi à quitter la mission. Bref notre marché est structuré comme cela aujourd'hui. On va en reparler dans quelques instants. Lorsque vous ne savez pas combien proposer comme taux journalier, demandez conseil à d'autres indépendants sur le type de la mission. Une mission de 2 jours est facturée plus chère qu'une mission longue qui offre un contrat ferme au consultant indépendant. Sachez baissez aussi votre facturation en cas de négociation délicate. Rappelez-vous : vous n'êtes plus entrain de négocier un salaire avec toute la hargne que vous aviez lorsque vous étiez salarié. Vous êtes entrain de négocier un tarif journalier dans le cadre d'un contrat commercial. Préparez-vous à l'exercice. **Les papiers** Une fois le premier client décroché, vous devez ensuite fournir un ensemble de justificatif à votre intermédiaire. Si vous êtes en EURL, cela pourra être par exemple votre dernier bilan, une attestation URSSAF, un KBis, vos statuts... Bref tout ce qui prouve que vous êtes une entreprise en règle et à jour du paiement de ses cotisations. **Etablir une facture** Demandez de l'aide à votre expert-comptable. Pour être valide, une facture doit comporter un certain nombre d'indications obligatoires. La TVA sera exigible sauf si vous facturez une entreprise non-français comme une boîte anglaise par exemple. Votre facture doit indiquer aussi clairement le délai de paiement, les indemnités. Bref un simple fichier Word ne fait pas l'affaire. Il faut faire cela très proprement, et votre comptable n'est pas là pour le faire à votre place. **Faire un Devis** Je ne peux pas expliquer en 4 lignes comment faire un Devis, juste vous donner quelques conseils. Tout d'abord, là aussi il y a un certain nombre de mentions obligatoires. La durée de validité du Devis, les modalités de paiement, le détail avec TVA, bref pas si simple la première fois. Votre devis doit détailler ce que vous comptez réaliser, et aussi ce que vous ne ferez pas. Si votre réalisation de site Internet n'inclus pas de support, marquez-le. Les indépendants sur le site Freelance-info conseillent d'être très précis, afin d'éviter de tomber face à un client un peu difficile. **Des clients, encore des clients** Je pense que la majorité d'entre vous (développeur Java avec un peu d'expérience) cherchera des missions longues. Il faut passer par une SSII si vous n'avez pas l'âme d'un commercial, ni le temps de rencontrer du monde, ou de blogger aussi. Les cabinets de conseils peuvent aussi vous donner un coup de main, notre marché étant assez bien organisé, je ne m'inquiète pas trop pour vous. Le client est là et il vous attend. A votre avis pourquoi travaillons-nous avec des intermédiaires et rarement avec les clients finaux ? Et bien aussi parce que les indépendants sont difficiles à trouver. Un acheteur qui est mandaté pour trouver un développeur Java avec 4 ans d'expérience n'a pas le temps de présélectionner des candidats, jouer les recruteurs ou les chasseurs de tête, subir nos humeurs, nos changements de dernière minute... J'ai un œil critique et je me dis qu'il serait trop simple de voir les SSII comme des intermédiaires qui se chargent sur votre dos. Il y a un travail en amont, c'est simplement ensuite les flux d'argent et l'opacité qui peuvent décourager un indépendant. Pour moi, la crise aura eu le mérite d'épurer ce marché, un peu comme les agences immobilières. Les intermédiaires peu sérieux doivent avoir plus de mal que les entreprises sérieuses je pense. Dans [cet article](#), une phrase résume bien l'état du marché entre les indépendants et les clients finaux : *Mais les indépendants-vendeurs semblent souffrir d'un grand déficit commercial. De leur côté, les entreprises-acheteuses sont dans l'impossibilité de localiser simplement les profils recherchés et quelque peu prisonnières de leurs habitudes.* Nous les indépendants nous ne sommes pas bon commercialement. A l'autre bout, les clients finaux ont du mal à nous trouver. Donc il y a de la place pour les intermédiaires : les SSII ou les cabinets spécialisés. Et ce marché ne cesse de grossir. Là tout bon informaticien se dit : "mais je vais créer mon site d'échange entre les indépendants et les acheteurs". Seulement, est-ce que cela fonctionne ? Passer par un brooker permet de se faire aider, un peu comme un agent sportif. Il existe donc plusieurs initiatives, des sites ou des places de marchés pour que les acheteurs et les freelances se rencontrent. Le site [IDirect](#) est intéressant, il propose de référencer les indépendants d'une part, les acheteurs d'autre part, afin de supprimer les intermédiaires. Le site [HitechPros.com](#) propose beaucoup de missions en informatique en général. Le portail [The IT JobBoard](#) est aussi pas mal chassé, vous pourrez y trouver des missions intéressantes. L'intermédiaire [Freelance-informatique.FR](#) est un courtier entre les indépendants et les SSII/clients finaux. Leur site donne un indicateur des profils les plus demandés, ce qui devrait vous rassurer car Java/J2ee se place 2ème derrière .NET. **Et la crise ?** Difficile de juger. En Grande-Bretagne c'était plutôt difficile il y a un an. Vous pouvez relire [un article que j'ai écrit l'an passé](#) sur ce sujet, après Devvxx 2008 pour vous faire une idée. Je pense que la demande en profil Java est là et bien là. Chacun est unique, vos compétences et votre expérience, c'est maintenant vous que vous vendez. En espérant que cet article vous aura apporté un peu d'informations, n'hésitez pas à commenter si vous souhaitez que je parle d'autres sujets sur le thème des indépendants. **Références** - Un témoignage intéressant à lire sur 01Net : "[Pas aussi facile d'assumer le statut de freelance](#)" - [Commercialisation des prestations informatiques, vers une nécessaire réforme](#)

## Evénement Spring à Paris le mercredi 4 novembre 2009

SFEIR et SpringSource organisent le mercredi 4 novembre [les Rencontres SpringSource 2009](#) à Paris à la Défense. [L'an dernier](#) cela avait été l'occasion d'interviewer Guillaume Laforge (chef de projet du langage Groovy), Peter Cooper-Ellis (Vice-president of Engineering and Product Management) et Jürgen Holler (Architecte en Chef du framework Spring). **Vu le programme cette année, je regrette de ne pas être disponible. Voici le communiqué de SpringSource et de SFEIR.** Voir : <http://www.rencontres-spring.com/> SpringSource lance une matinée de présentation le 4 novembre à la Défense (adresse plus bas). Avec l'émergence du cloud, la possibilité d'adapter les ressources système en fonction de ses besoins immédiats devient une réalité. L'intégration de SpringSource au sein de VMWare va permettre de combiner deux technologies en une seule pour révolutionner le cycle de vie complet d'une application dans le cloud. Pour cet événement, nous faisons venir : \* Adrian Colyer, le CTO de SpringSource, qui présentera l'ensemble des développements clés de notre roadmap. \* Javier Soltero, Chief Technology Officer for Management Products, qui montrera tous les avantages à utiliser notre

solution de monitoring Hyperic pour manager et déployer dans le cloud. \* Arjen Poutsma, Senior Enterprise Application Architect, qui dévoilera les fonctionnalités avancées de la nouvelle release Spring 3.0. Leurs présentations seront en Anglais. Le détail précis du programme ainsi que le formulaire de pré-inscription sont disponibles à cette adresse: <http://www.rencontres-spring.com/category/2009> L'événement aura lieu au centre d'affaires NCI ComSquare, en bas de l'esplanade de la Défense, toutes les indications précises sont disponibles à cette adresse: <http://www.groupenci.com/fr/ile-de-france/defense.com-square.html> Cette matinée va être la grande première en France pour dévoiler la stratégie entre SpringSource & VMware. Les sessions seront très interactives et vous donneront l'occasion de poser vos questions et d'obtenir toutes les explications sur le cloud de demain. En espérant vous compter parmi nous! J'ai rencontré Adrian Colyer en Suisse, c'est une personne vraiment intéressante et riche. Si vous voulez vous plonger dans sa présentation de Jazoon 2009, vous pouvez relire [cet article](#) que j'ai écrit en juin dernier. Si vous êtes plutôt intéressé par Spring 3.0, j'ai aussi suivi [la présentation de Jürgen Hoeller](#) toujours à Jazoon.

## Zenika



J'ai mis à jour la liste des blogs techniques sur Java affichée à gauche sur la page d'accueil du Touilleur Express. J'avais un peu oublié cette liste, qui n'a pas changé depuis 7 mois... Pour le coup, je vais vous présenter un peu [Zenika](#), qui est l'un des sponsors Platiniums [du Paris Java User Group](#). Zenika est un cabinet spécialisé dans les architectures Java/J2EE. Zenika s'est associé avec des partenaires privilégiés comme SpringSource et Terracotta, et l'activité est décomposée en une partie Architecture et Conseil, une partie Formation et une partie Réalisation. Cabinet fondé par des passionnés techniques, Zenika organise régulièrement des conférences, pas uniquement sur Java. L'agenda des prochaines présentations est [en ligne sur leur blog](#). Voici le menu pour la fin de cette année :

Toujours dans la logique de proposer des événements exceptionnels, voici la liste des prochaines conférences à venir. Le nombre de places est limité (maximum 40) n'hésitez pas à vous enregistrer en avance et surtout, en cas d'empêchement à libérer la place, nous allons gérer l'équivalent d'une liste d'attente. \* Le 29 Octobre 2009, [La programmation fonctionnelle avec F#](#) par Robert Pickering, reconnu comme l'un des experts mondiaux de F# et nommé Microsoft MVP (Most Valuable Professional) en octobre 2008. La deuxième édition de son livre « Beginning F# » devrait être publié en janvier 2010 par A-press. \* Le 4 novembre 2009, [Tester des services REST avec SoapUI](#) par Ole Lensmar, Co-fondateur de la société Eviware editrice de SoapUI, l'outil de test de Web Services le plus utilisé au monde, avec une base d'utilisateurs proche des 300 000. \* Le 12 novembre 2009, [Rétrospective de l'utilisation de BDD en entreprise](#) par Mauro Talevi, contributeur dans le développement du noyau du framework JBehave. \* Le 16 novembre 2009, [Command Query Responsibility Segregation](#), par Udi Dahan, expert de renommée internationale concernant l'architecture et la conception logicielle. Il est aussi le vainqueur, et ce 4 années consécutives, du très convoité prix MVP (Most Valuable Professional) décerné par Microsoft dans la catégorie Architecture. Membre du board de Microsoft concernant les nouvelles technologies WCF/WF/OSLO, L'initiative de Software Factories ainsi que du "Composite Application Library & Guidance". Il est l'un des 33 experts en Europe reconnu par l'association internationale .NET (INETA), un auteur et un formateur pour l'"International Association of Software Architects on Reliability, Availability, and Scalability". Enfin, il dirige le projet NServiceBUS, le plus populaire ESB Open Source en .NET. \* Le 07 décembre 2009, [Scrum et Kanban, Lean et Toyota Way](#) par Martine Devos, Scrum Trainer et coach d'équipes appliquant les principes et pratiques agiles, incluant Scrum, Extreme Programming et le développement Lean.

Si vous venez au Paris JUG vous connaissez certainement aussi Olivier Croisier l'auteur du Blog "[The Coder's Breakfast](#)" qui a rejoint Zenika. Enfin je garde un très bon souvenir de [la présentation de Wicket par Carl Azoury et Nicolas André](#) du mois de mars dernier. Bref voilà, si vous cherchez une équipe dynamique et compétente, ainsi que des conférences pointues, c'est chez [Zenika](#).

## Désendetteur Technique



Dimitri Baeli d'eXo

Platform a mis un nom sur notre métier, et sur ce que j'ai fait depuis un an : Désendetteur Technique. C'est l'histoire d'un message sur Twitter, qui est à l'origine de ce billet. J'ai trouvé cela génial, car cela correspond à une partie de notre activité : réduire la dette technique d'une application. A cela j'ajouterai, réduire la dette de gestion de projet en aidant une équipe à découvrir les méthodes Agiles. Notre rôle est d'identifier et de colmater rapidement les fuites dans votre projet. Tout d'abord les fuites techniques. Prenez un excellent outil comme [XDepend](#) par exemple. C'est un peu le scanner du Docteur House. Imaginez la scène un instant... Votre application émet des signes inquiétants. Votre projet semble s'arrêter petit à petit, et personne ne sait vraiment de quoi souffre celui-ci... Il est temps d'appeler un Désendetteur Technique... Quelques heures avec des outils comme XDepend ou Sonar, et le diagnostic tombe : votre produit souffre d'une bonne complexité cyclomatique ici, et il y a un problème maven là, ainsi qu'un souci d'organisation car vous n'avez pas d'intégration continue. Le patient: *Oui là ça fait mal docteur... Non je n'ai pas lu la documentation sur Hibernate... oui j'ai mis lazy=false car je ne sais pas comment gérer ces objets, non je ne m'excuse pas pour cela...* Bref comme 20% du code coûte 80% des performances, un Désendetteur Technique sera capable de vous aider à chercher du bon côté, à corriger le souci et surtout, à éviter qu'il ne se reproduise. Soigner et éviter de retomber malade, c'est la devise d'un Désendetteur Technique. Prenez un projet, passez le à maven2. Quelques mois plus tard c'est le foutoir. Pourquoi ? Parce qu'il faut aussi **apprendre** aux développeurs à se servir de Maven 2. Concernant les fuites dans la gestion de projet, je pense que trop de projets souffrent car ils ne sont pas gérés du tout. Et la gestion projet, c'est beaucoup d'écoute, un peu d'accompagnement et très peu de directif. Un mode de gestion où le développeur doit cocher une case pour dire ce qu'il a fait lundi, où il ne peut pas fermer un bug sans que son chef n'ait vu sa correction, où il est tenu d'écrire une spécification de 100 pages avant de coder, c'est un projet mort. L'Agilité c'est passer du Directif au Démerditif. J'explique aux gens que ça va faire mal, ils ne vont pas aimer, ça va être difficile et ça peut même échouer. Après cela, si je suis encore là, nous pouvons travailler. En janvier lors d'un stand-up devant 15 développeurs, j'ai un peu râlé en disant qu'il fallait arracher le pansement du petit bobo pour avancer et pas tirer doucement. Faire du Scrum, c'est faire un saut de 10 mètres dans le vide. Après soit vous avez pensé à prendre un gars pour vous attacher, soit vous vous écrasez 10 mètres plus bas. Passer à l'Agilité s'apparente plus à du saut à l'élastique qu'à un saut en parachute. Vous allez faire quelques aller-retours pas forcément agréable avant que cela ne fonctionne. Désendetteur Technique, j'adore. Franchement merci Dimitri de mettre un nom sur ce que les experts seniors font. Prenez [le livre blanc d'OCTO Technologies](#) sur la productivité. Ce livre est vraiment bien écrit. Il explique les différents points à mettre en place pour améliorer la productivité d'une équipe. C'est presque la Bible du Désendetteur Technique non ? Mais attendez... j'ai déjà vu cela quelque part. Je vous demande un peu de concentration. Ouvrez [cet article](#) dans une autre page et lisez tranquillement "The Joel Test". Comme moi vous voyez les 12 points que nous connaissons tous un peu près non ? Bon, et maintenant pouvez-vous s'il vous plaît, **regarder à quelle date cet article a été publié ?** En août 2000... Mince ça fait mal non ? Là vous vous dites juste que le bonhomme a écrit il y a 9 ans ce que l'on nous sert aujourd'hui comme de l'eau bénite ! Et c'est donc pour dire que les Désendetteurs Techniques sont des gars qui ont déjà lu tout cela, qui savent plus que vous ce qu'il faut faire. Ils sont capables d'identifier rapidement un gros souci dans votre projet. Que ce soit une architecture un peu pourrie, une librairie tordue, un réglage par défaut mal fait, c'est juste un gars qui sait ce que vous devriez savoir, et ce que vous saurez à terme. Ce bonhomme a juste un peu plus d'expérience et de culture que vous, il n'est pas plus intelligent que vous, il est juste plus instruit. Oui vous savez maintenant qu'il faut faire de l'intégration continue. Si vous m'aviez contacté [en février 2006](#) je vous aurai déjà parlé de CruiseControl car je m'amusais avec. Si vous me contactez maintenant, je vous parlerai d'Application Content Provider plutôt que de Portails et de CMS, car c'est un truc qui sera là dans quelques années. Notez la date de cet article, et on se revoit dans 4 ans. Suivez l'aventure d'eXo Platform, Benjamin Mestrallat parle d'Application Content Provider ([voir aussi cet article](#)). On en reparlera sur le blog. Enfin si vous n'avez pas de Désendetteur Technique et que vous en cherchez un, il faut contacter les cabinets spécialisés dont la liste des Blogs est là, juste à gauche de cet article. OCTO Technology, Xebia, Oxiane, Ippon Technologies, SFEIR, Zenika, je crois que je connais au moins 3 ou 4 personnes dans chacune de ces entreprises. Et là vous pouvez vous dire que c'est bon signe. Il y a des gars comme moi, passionné qui pourront vous donner un coup de main. Il y a des Docteur House, des équipes de SOS Fantômes, des Men in Blacks, bref il y a de quoi venir remettre d'équerre soit votre code, soit votre équipe ou sinon votre projet. Désendetteur Technique Merci Dimitri.

## Présentation de GateIn beta 2 et eXo Portal

## mardi 27 octobre

Je suis en formation chez eXo Platform aujourd'hui et demain sur les Portails et je vous informe qu'il y aura une grosse présentation demain matin à Paris sur GateIn beta 2, la vision d'eXo Platform, l'avenir des portails, bref du lourd. Le tout aura lieu à la Maison de l'Acquittaine, rue des Pyramides, station Pyramides L14, à quelques pas d'Opéra. J'y serai aussi demain.

eXo Platform a le plaisir de vous inviter à son séminaire gratuit intitulé "**eXo/JBoss : GateIn Le Portail Open Source de référence**" le mardi 27 Octobre à partir de 9h.

Dans le domaine des portails et de la gestion de contenus un grand nombre de solutions Open Source existent mais avec le rapprochement d'eXo Platform et de JBoss c'est une nouvelle génération de portail qui est lancée. eXo Platform édite une des solutions majeures de Portail d'entreprise adoptée aujourd'hui par un certain nombre de Grands Comptes comme le Ministère des Finances Belge, l'Etat de Genève, le Ministère de la Défense, le Groupe M6 ou bien des assurances, dont Generali, avec des volumétries de 2 500 à 30 000 utilisateurs. Avec ce séminaire, nous vous proposons une matinée à la découverte de la solution d'avenir en terme de collaboration et de gestion de contenus mais également la meilleure alternative aux acteurs traditionnels du marché. Voici le programme complet du séminaire : \* 09h00 – 09h30 : Enjeux et retours d'expérience sur la gestion de contenus et les portails open source. \* 09h30 – 10h30 : GateIn la nouvelle génération de portail d'intégration développée par JBoss et eXo \* 10h30 – 12h00 : Les modules collaboratifs intégrés au portail (GED, FAQ, Forum, Webmail..) et la gestion de contenus multi-sites \* 12h00 – 12h30 : Discussion libre avec nos "Product managers" Ce séminaire gratuit se déroulera à Paris, le mardi 27 octobre de 9h à 12h30 et débutera par un petit déjeuner d'accueil à partir de 8h45.

Pour vous inscrire, envoyez rapidement un email à [contact@exoplatform.com](mailto:contact@exoplatform.com) Dépêchez-vous c'est demain !

## Il faut écrire des specs itératives



L'Agilité peut faire croire que l'écriture de spécifications est dépassée, c'est faux. Soyons critique 2 secondes sur la façon dont les méthodes Agiles sont encore vendues à l'entreprise par des mauvais vendeurs: elles évitent l'effet tunnel, elles cassent l'effet Waterfall et donc, grâce à la livraison d'incrément de logiciels réguliers, elles promettent de meilleurs résultats. En effet, il existe des projets qui ne débutent pas avant l'écriture détaillée d'un document fonctionnel, puis d'un document technique. Ensuite ces documents se transforment magiquement en contrat entre le client et le prestataire. Oui là nous sommes en effet sur quelque chose de mauvais. Je peux écrire des heures dessus, mais vous le savez déjà. Une spécification qui ne bouge pas, c'est une photo qui vieillit sur votre écran. Une spécification doit vivre, on va y revenir. Lorsque j'ai commencé il y a 8 ans à écrire ma première spécification, j'ai eu la chance de travailler avec un bonhomme assez fort sur ce sujet (Fabien si tu es dans la salle, lève la main... ok merci...). Il m'a réellement appris à écrire. Le fonctionnel tout d'abord, qui se concentre sur le "quoi". Que veut le client ? Que fait-il devant son écran ? Que va faire votre logiciel ? La spécification technique ensuite, qui explique le "comment" et qui donne aussi les besoins techniques sur les aspects sécurité, déploiement, montée en charge. Bref Fabien lorsqu'il m'a demandé d'écrire ma première spécification, a beaucoup insisté sur la forme, ce qui était antinomique avec mes gènes d'informaticien, intéressé par le fond. Il insistait à l'époque pour que mes documents soient bien présentés. Bien écrire, bien mettre en page, faire des schémas soignés, aérer la mise en page, et surtout : être agréable à lire. Et j'ai compris assez tard que c'était le premier point : soigner la présentation. A votre avis pourquoi ? **Une spécification n'est pas une notice de médicament de contraception** Si la présentation de votre spécification ressemble à une notice de médicament, une recette de cuisine, ou un disclaimer comme lors de l'installation d'un jeu, ce n'est même pas la peine de prendre le temps d'écrire. Demandez à quelqu'un qui est capable d'écrire. Tout le monde n'est pas fait pour vulgariser, schématiser et délirer à mort dans un document de 305 pages sur *l'éplumage automatique des poulets avec le nouveau robot XVW8000-1*. **Une spécification doit vivre** Revenons un instant sur les vendeurs de lave-linge de l'Agilité qui vous expliquent que les spécifications créent un effet tunnel. Cette affirmation serait vraie si et seulement si plus personne n'avait le droit de modifier une spécification une fois celle-ci écrite. Est-ce que votre spécification est imprimée dans un livre broché avec couverture quadrichromique ? Non ! C'est un document Word. Et bien puisque vous parlez d'itération tout le temps autour de vous, vous devriez aussi appliquer ce principe à votre spécification technique. Il est vital d'écrire de nombreuses versions de votre spécification technique ou fonctionnel, et de conserver chaque version précieusement. **Quelques idées pour écrire** Le souci c'est que certains d'entre nous n'ont pas l'option "écrire de la spécification" dans leurs gènes. Prenons ceux qui aiment écrire et qui ont envie de mettre noir-sur-blanc quelques principes et quelques idées. Pour écrire, le plus simple est de penser à un artichaut. D'abord construisez un plan général et complet de votre document. Ensuite commencez par écrire

un résumé de 25 lignes au début du document, qui sera lu par 97% des gens qui n'ont pas envie de se plonger dans vos 305 pages sur *l'éplumage des poulets avec un robot avec un nom à la con..* On appelle cela un "Executive Summary". Ensuite prenez le soin de préciser à quelle audience s'adresse ce document : d'autres développeurs, la MOA, le gars qui fait le ménage le soir sur votre bureau, votre fils qui pourra dessiner au dos de vos pages, etc. Très important. Rien de plus frustrant que de prendre son lecteur par surprise en ne disant pas "*Hé sur ce blog on ne parle pas de .NET et d'élevage de fourmis*". Vous suivez ? Toute couverture de document doit avoir une image. Votre oeuvre de 305 pages doit avoir une image sur la couverture. Je vais vous expliquer pourquoi. En fait, personne ne se souvient d'un document qui s'appelle

"Spécifications Techniques Détaillées pour un outil robotisé d'éplumage des Gallus gallus domesticus, v 1.2 par Nicolas Martignole"

Par contre si vous mettez une grosse image sur la couverture de votre document, les lecteurs se souviendront de votre document et vous dirons "Où as-tu mis ton document avec [la photo du](#)



[poulet](#) ?" Pensez à un titre qui soit simple et qui ne soit pas ampoulé. Si vous n'avez pas d'humour, allez acheter des pilules d'humour dans une pharmacie. Vous en prenez 2 le matin et 2 le soir, et vous serez un gars/une fille super marrante dans quelques semaines. Ne pas confondre avec les pilules de l'amour, les bleus. Là vous serez simplement ridicule. Imaginez : "*Spécifications Techniques d'un Robot de découpage des poulets*" **Mais tu vas nous dire pourquoi il faudrait écrire ?** Enfin le morceau le plus intéressant. Pourquoi faut-il écrire un minimum avant de débiter un projet ? Tout d'abord parce que les spécifications sont un moyen relativement économique de faire fonctionner votre cortex de communicant. Ecrire une spécification c'est s'adresser à soit-même avant tout. C'est réfléchir de manière objective sur ce que vous allez faire dans quelques temps. C'est vous forcer à bien capturer les idées de votre client, à préparer un road-book pour les autres développeurs. Une spécification est toujours écrite par une seule personne. J'ai tenté l'écriture à 2, ça ne marche pas tellement. Il faut avoir des relecteurs qui vous corrigeront, mais isolez-vous pour écrire et mettre au propre les explications détaillées de ce que vous pensez réaliser. Une spécification permet certainement d'éviter un gâchis. Plutôt que d'ouvrir Eclipse et que de commencer à coder sur une idée, un travail en amont d'analyse permet de dégrossir le travail, et donc d'éviter ce temps perdu. Enfin une spécification n'a pas besoin d'être énorme. La taille ne compte pas. C'est le contenu qui compte. Peut-être que 10 pages avec 3 schémas seront suffisant, qui sait ? **Quel logiciel utiliser ?** Je termine rapidement, en conseillant d'écrire sur un logiciel comme Word ou Pages, afin de soigner la présentation et de préparer un document qui vit car il est imprimé. Oui monsieur, je te dis qu'il faut que tu l'imprimes, pour le lire dans le RER, aux petits-coins, dans la voiture... Bref pour t'en imprégner comme un sachet de thé dans de l'eau chaude. Oui bon, Google Docs je pense permet de gagner du temps et d'éviter d'abattre des arbres à Fontainebleau pour imprimer ton document. Oui en effet il offre des moyens de collaborations. Donc on peut utiliser Google Docs. Mais pas Emacs, pas LeX, pas un machin d'informaticien qui va produire une notice de médicament. Enfin pour terminer, prenez le temps de regarder [le site Microsoft Office Online](#) qui propose des templates et surtout des images pour illustrer vos documents. Voilà, qu'en pensez-vous ? **Crédits pour cet article** Cet article est inspiré de réflexions de Joel Splosky sur son blog "[Joel on Software](#)".

## Blog mis à jour

J'ai effectué quelques mises à jour aujourd'hui. Si vous constatez des problèmes sur le blog, faites-le moi savoir sur Twitter @LeTouilleur. Merci. [Update lundi matin] Le site était inaccessible entre 09h30 et 11h00 ce matin. La raison ? Vous tout simplement. Victime de son succès, le blog n'a pas tenu la charge, la base de données est tombée. Après avoir ajouté des parts sur mon interface Gandhi, j'ai relancé le tout. Désolé pour ces petits soucis. Pour vous donner une idée, la semaine passée j'ai eu un pic de visite à 955 visiteurs, et 6090 visiteurs différents (adresse IP). Cela représente 23670 hits en 6 jours, un hit étant une URL distincte vue. Chacun a donc regardé entre 3 et 4 pages, ce qui me fait plaisir par rapport au temps passé à écrire les articles. Je vais devoir trouver une nouvelle solution pour que le blog continue à fonctionner avec un budget raisonnable.

## GateIn et Liferay, les portails préparent l'entreprise de 2015

Si vous avez suivi les sujets du moment sur le Touilleur Express, vous savez qu'en ce moment je travaille sur les Portails Java Open-Source (Liferay, eXo Portal et Jahia). Après [quelques articles](#) sur les Portlets et les Portails, j'ai eu la chance de participer la semaine dernière à la formation eXo Portal organisée par Julien Viet. Il est temps de vous parler un peu de GateIn. Dans cet article j'ai essayé d'avoir une vision plus globale de ce que les Portails peuvent nous apporter. Nous reviendrons dans d'autres articles sur [GateIn](#) pour toi développeur. Là j'ai envie de parler à toi, le Client final ou le Visionnaire, histoire de vous parler un peu de ce qui nous attend. Tout d'abord, regardons un peu le marché des portails d'intégrations. L'étude "[Magic Quadrant for Horizontal Portals](#)" de 2009 est intéressante, car elle donne un état du marché, et quelques indications sur les acteurs du marché. 2 phrases retiennent mon attention : "*By 2011, Gartner expects at least 15% of new enterprise portal projects in Global 2000 firms to use open-source horizontal portal*

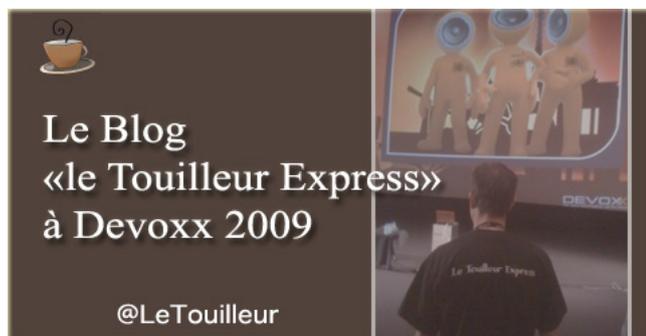
frameworks." et "By 2014, horizontal portal products based on portal containers will be used for no more than 60% of new enterprise portal projects.". En 2014, Gartner estime à 60% le nombre de nouveaux projets basés sur les portails d'intégration... C'est dans 5 ans... allo ? Le marché des portails est actuellement dominé par IBM, Oracle et Microsoft. Cependant, un effet bénéfique de la crise, c'est que les budgets d'investissement pour les portails internes ont connu un sacré ralentissement en 2009. Cela a pour effet de profiter directement à des acteurs comme Liferay, RedHat et eXo Platform. Le rapprochement entre eXo Platform SAS et Red Hat Inc. est certainement un signe fort qu'en 2010, ces deux là viendront perturber le marché avec un objectif à 5 ans très clair : devenir la première solution open-source java de portails d'entreprise. Liferay n'a qu'à bien se tenir. La semaine passée j'étais en formation à Paris avec Julien Viet, ancien de RedHat et maintenant Lead Developer chez eXo Platform, sur le Portail. La formation s'est déroulée sur 2 jours. Elle permet tout d'abord de comprendre la culture et le positionnement des Portails. Lorsque nous parlons de Portails, certains en ont une vision très Web. Ils imaginent des mashups, une page à la iGoogle. D'autres en ont une vision plutôt orientée middleware/service. Les portails sont des ponts entre différents services et une application Web. Et c'est cette culture qui doit, à mon avis, encore être travaillée. Liferay a quelque chose que son voisin n'a pas encore : une sacré interface prête à l'emploi avec ses quelques 60 petites applications. Son plus gros avantage et aussi son plus gros défaut. Petites applications, pas produits. Liferay descend du monde du Web, et s'adresse maintenant à l'infrastructure. Au contraire, JBoss Portal est un logiciel qui vient du monde du middleware, avec un socle robuste, un moteur d'exécution proche quelque part d'un serveur d'application. eXo Platform propose une vision plus produit, qui n'est ni du Web Portlet, ni du MiddleWare, mais ce que j'appellerais un moteur d'exécution pour applications webs. C'est peut-être le code génétique de [GateIn](#). [GateIn](#) est taillé pour être un moteur d'hébergement d'applications, comme JBoss Portal. Mais il est aussi structuré pour proposer de vrais produits plutôt qu'une boîte à Portlets comme chez Liferay. GateIn est le moteur de départ sur lequel nous allons déployer d'une part les produits eXo Platform, mais aussi votre propre application portail. Les produits eXo Platform permettent de gérer du contenu (eXo WCM), de gérer des documents (eXo DMS), de configurer finement la validation (eXo Workflow), de créer des forums et des FAQs intelligentes (eXo KS) et propose aussi une suite d'outils collaboratifs (eXo CS). Encore une fois, là où Liferay propose des Portlets, eXo Platform propose de réels produits, avec la documentation et les équipes produits, avec un service plus adapté à l'entreprise. Maintenant attention, il semble que la bonne pratique est de se baser sur des solutions plus complètes dès lors que les besoins de base ne sont pas couverts par le Portail. Liferay dès le guide de l'utilisateur, vous oriente vers [Alfresco](#) si vous souhaitez utiliser une solution de publication de contenu complète (CMS). Et eXo Portal fonctionne aussi d'ailleurs très bien avec [Alfresco](#). On comprend donc que l'énergie maintenant n'est plus dans le développement d'une application web avec son authentification, sa gestion de la mise en page, son cycle de vie... Non j'ai presque envie de croire que c'est terminé. Liferay et GateIn vous font regarder vers une nouvelle direction, qui me semble plus intelligente pour l'entreprise : l'intégration d'application dans un portail plutôt que l'intégration d'un service dans une application Web. Les produits d'eXo Platform seront tous bientôt compatibles avec [GateIn](#), qui se positionne comme un moteur d'exécution, une plate-forme, et la porte d'entrée vers le monde de l'entreprise. Pour revenir à l'intégration de solutions existants, prenez [SugarCRM](#) par exemple pour la gestion de la relation clientèle. Une fois intégré avec un moteur de publication de contenu, voilà votre site Internet d'entreprise avec son extranet. Si bien sûr votre métier est de développer des applications de gestion, peut-être que cela vous semble abstrait. Mais si votre métier est de développer des solutions webs d'entreprise, et bien cela devrait vous parler. Quelle entreprise en 2015 pourra se passer d'un site Internet, avec un extranet sécurisé pour ses employés, un outil de gestion de la clientèle, un Wiki, des Forums, et un moteur de publication pour les pages ? Dites-moi qui pourrait se passer de ces outils ? Rendez-vous maintenant pour construire ensemble pas-à-pas un nouvel écosystème d'applications dans GateIn. Première étape : installer GateIn, faire le tour du propriétaire, puis ensuite commencer à mettre ses premières applications. A dans 5 ans ? ou à demain ?

## Devoxx 2009... Préparez-vous



Plus que 10 jours avant le début de la plus grosse conférence sur Java en Europe, qui aura lieu à Anvers en Belgique. Lundi et mardi, Devoxx propose l'Université. La conférence à proprement parler débute le mercredi jusqu'au vendredi midi. Le format de l'Université est de proposer des séances longues en comité plus restreint pour travailler sur un sujet. De la formation, ni plus, ni moins. J'ai déjà repéré des sujets de fonds que je souhaite approfondir, afin de vous ramener quelques petites pépites de là-bas. Les conférences seront disponibles rapidement sur le site [Parleys.com](#), créé par le fondateur de Devoxx, le BeJUG. Si déjà vous voulez revoir quelques sessions de l'an passé, c'est un site très bien fait, sur lequel il est possible de se former à peu de coût aux dernières technologies autour de la plateforme Java. J'espère vous voir là-bas si vous avez l'occasion de venir. L'an passé, le Paris JUG avait organisé une sorte de FlashMob sur les marches de l'entrée, histoire de marquer le coup. On compte bien refaire quelque chose cette année, suivez-nous sur Twitter [@parisjug](#), ou sinon aussi [@letouilleur](#). RDV dans 10 jours !

## Devoxx 2009 : page spéciale



Bonjour à tous A partir

du lundi 16 novembre, rendez-vous chaque jour sur [la page spéciale](#) du blog "Le Touilleur Express" pour retrouver chaque jour un instantané de la journée. Au programme : des rencontres avec des acteurs de la communauté Java, des découvertes, des surprises, des bons gros articles écrits à 2h du matin qui vous font presque croire que vous y étiez et aussi des photos cette année, comme lors de [Jazoon 2009](#). Vous pouvez aussi me suivre via mon compte Twitter : [@LeTouilleur](#).

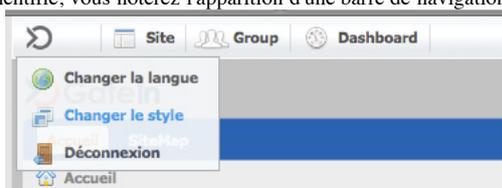
## GateIn : premiers pas

Dans ce premier article, je vous propose de découvrir [GateIn](#), le nouveau moteur fruit du rapprochement des équipes d'eXo Platform et de JBoss Portal. De l'installation aux premiers pas, l'idée est de vous présenter les nouveautés de GateIn, avec l'écriture d'une Portlet "HelloWorld", son déploiement et enfin une visite des coulisses pour bien comprendre ce que vous apporte GateIn. **Prérequis** Vous devez disposer de la version 6 de Java. La variable JAVA\_HOME doit être configurée et elle doit pointer vers votre répertoire d'installation de Java. **Installation** Commencez par télécharger la version Beta 2 de GateIn disponible sur [cette page](#). J'ai sélectionné la version [GateIn 3.0.0 Beta2 with Tomcat 6.0.20 \(tar.gz\)](#). Une fois l'archive décompressée, placez-vous dans le répertoire GateIn-3.0.0-Beta02/bin et démarrez gatein: `cd GateIn-3.0.0-Beta02/bin ./gatein.sh run` Vous pouvez soit lancer GateIn et conserver la main avec l'option `run`, soit lancer GateIn en tâche de fond avec l'option `start`. C'est le fonctionnement de Tomcat en standard. **Petit tour** Une fois le serveur démarré, ce qui ne prend pas plus de 20 secondes sur ma machine, vous pouvez alors ouvrir un navigateur sur la page <http://localhost:8080/portal> [caption id="attachment\_2270" align="alignnone" width="500" caption="GateIn startup screen"]



[caption] Premier

point que nous verrons plus loin, c'est la réactivité de l'interface. J'ai testé la version JBoss et il m'a semblé que la version Tomcat était légèrement plus rapide. Notez que l'interface est en Français par défaut, car ma version de Mac OS est en Français, ainsi que mes réglages dans Firefox. En haut à droite vous pouvez tester le changement de langue. Par défaut GateIn supporte déjà l'anglais, le français, l'espagnol, l'arabe, le vietnamien, le russe et l'ukrainien. GateIn est développé en partie par les équipes d'eXo Platform qui sont localisées en Ukraine et au Viet-Nam. Par ailleurs, eXo Platform a ouvert l'an passé un bureau à Tunis, et GateIn est l'un des rares portails à supporter l'écriture de droite à gauche, comme en arabe ou en hébreu par exemple. Nous allons débiter notre tour en nous authentifiant en tant qu'utilisateur **demo** avec pour mot de passe **gtn**. Une fois authentifié, vous noterez l'apparition d'une barre de navigation rapide, comme sur Jahia

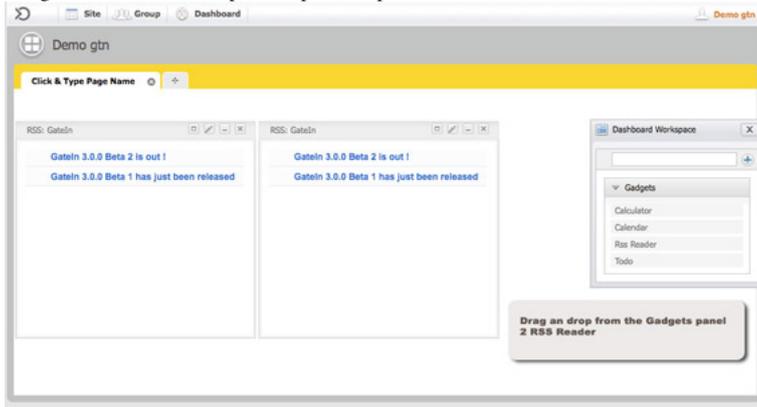


par exemple.

Cette barre est très pratique

car elle donne accès immédiatement à l'ensemble des espaces de GateIn. Le logo GateIn tout d'abord (comme sur la capture ci-dessus) permet de changer de style, de langue et de se déconnecter. L'onglet Site permet de basculer d'un portail à un autre. Il est en effet possible de créer différents espaces, et de régler finement les droits d'accès pour chaque groupe d'utilisateur. En parlant de groupe justement, l'onglet Groupe me donne accès à un ensemble de pages pour mon entité. Imaginons ici que c'est votre espace Jira, avec vos pages, vos outils, vos forums, votre wiki par exemple. **Le Dashboard** Cliquez maintenant sur le bouton Dashboard en haut à gauche de la page. Le Dashboard est votre espace personnalisé où il est possible de créer rapidement ses écrans

de travail. Bien entendu, par défaut le nombre de Gadget étant limité, cela va vous sembler un peu vide. Mais faites-moi confiance, je vais vous montrer comment utiliser des Gadgets additionnels externes plus loin. 1. Cliquez sur le lien "Add Gadget" à gauche, et effectuez un glisser-déposer du Gadget "RSS Reader". Répétez l'opération pour obtenir cet écran :

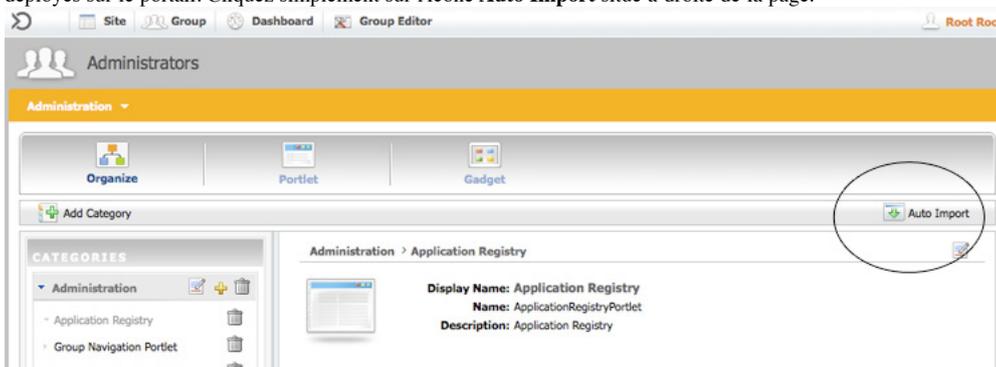


2. Cliquez maintenant sur le petit crayon situé sur l'un des 2 Gadgets que j'ai entouré sur cette capture d'écran. Saisissez l'url du flux rss du Touilleur Express (<http://www.touilleur-express.fr/feed/rss/>), entrez 5 messages et validez le tout. Vous devriez voir les 5 derniers articles publiés sur le blog, si vous



êtes connectés bien entendu à Internet.

Nous verrons plus loin comment créer ses propres Gadgets, passons maintenant du côté du développeur Java. Montre en main, vous avez dû mettre 5mn et 20 secondes, et encore c'est parce que je parle beaucoup... Passons derrière, suivez-moi **Votre première Portlet** Nous allons utiliser maven2 pour compiler notre premier projet pour GateIn. Téléchargez [le fichier zip](#) contenant le code source de cet article et décompressez le tout dans un nouveau répertoire. 1. Placez-vous dans le répertoire portlet et compilez le projet avec maven2: `mvn package ...` 2. Copiez ensuite le fichier war vers le répertoire webapps de votre installation de GateIn `cp target/texpress-portlet.war /Users/nicolas/Downloads/GateIn-3.0.0-Beta02/webapps/` 3. Lancez GateIn si vous l'aviez arrêté. Sinon vous devriez voir votre application se déployer sur GateIn, comme n'importe quelle application Java. 4. Connectez-vous sur GateIn (<http://localhost:8080/portal>) en tant qu'utilisateur **root** avec le mot de passe **gtn**. GateIn beta 2 n'importe pas automatiquement toutes les portlets distribuées avec le Portail. Il faut donc déclarer la première fois qu'une nouvelle Portlet "HelloWorld" est apparue. C'est simple et cela se fait en quelques clics. Pour simplifier, je vais vous donner les instructions de la version en Anglais. Basculez du Français à l'Anglais via le logo "GateIn" en haut à gauche et sélectionnez l'Anglais pour suivre la suite. 1. Après vous être authentifié en tant que root, allez dans le menu "Group" puis "Administration" et sélectionnez "Application Registry". Cet écran d'administration permet de contrôler les Portlets et les Gadgets déployés sur le portail. Cliquez simplement sur l'icône **Auto Import** situé à droite de la page:

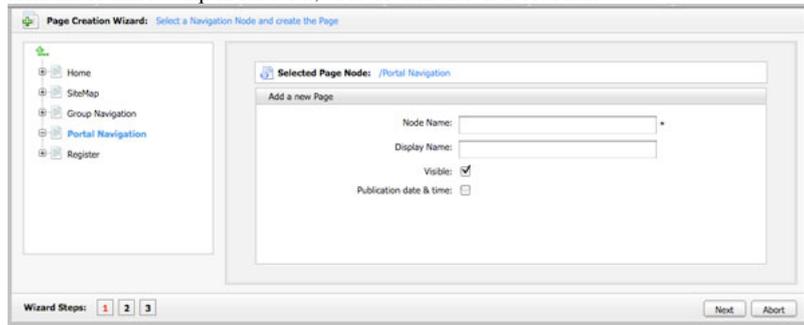


Une fois cette opération effectuée, vous verrez que votre Portlet "HelloWorld" est maintenant rangée dans la catégorie "Démonstration". Le rangement automatique est une astuce de Julien Viet. Elle se

base sur la définition des keywords dans le fichier portlet.xml Le fichier src/main/webapp/WEB-INF/portlet.xml est un fichier standard qui permet au conteneur de Portlet de démarrer et de connaître les caractéristiques de votre Portlet. Notre portlet supporte le mode view et le mode help. Les mots clés permettent de la classer automatiquement dans une catégorie de GateIn.

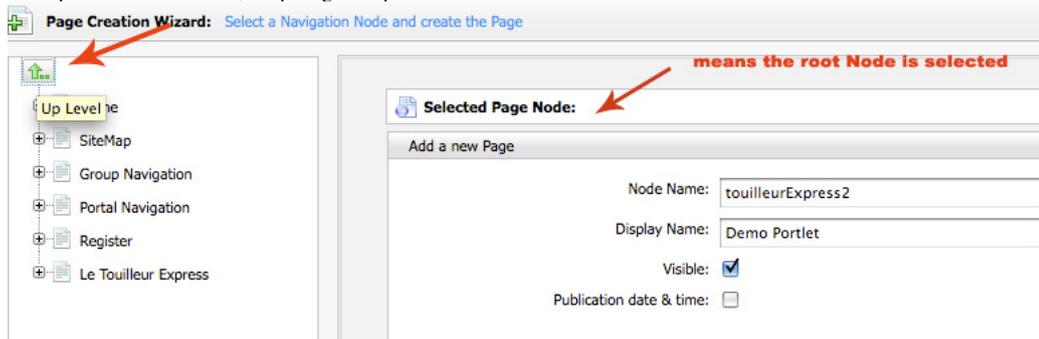
```
<?xml version="1.0" encoding="UTF-8"?>
<portlet-app xmlns="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd">
  <portlet>
    <description xml:lang="EN">Hello world JSP portlet</description>
    <portlet-name>HelloWorldJSP</portlet-name>
    <portlet-class>org.letouilleur.portlet.HelloWorldPortlet
  </portlet-class>
  <supports>
    <mime-type>text/html</mime-type>
    <portlet-mode>view</portlet-mode>
    <portlet-mode>help</portlet-mode>
  </supports>
  <portlet-info>
    <title>Hello World JSP</title>
    <keywords>demo, simple</keywords>
  </portlet-info>
</portlet>
</portlet-app>
```

Notez que cet auto import n'est à effectuer qu'une seule fois pour notre Portlet. Par la suite, nous pourrions écraser notre fichier WAR et travailler sans repasser par ce menu de gestion des Applications. Là je me permets de passer commande aux équipes d'eXo Platform et de JBoss : votre voisin Liferay a un répertoire "deploy" qui ajoute automatiquement la portlet dans le "Control Panel", l'équivalent de l'Application Registry. **Création d'une page** Continuons maintenant et terminons par la création d'une page. 1. Dans le menu Site Editor situé en haut, cliquez sur "Add New Page" ou "Ajouter une nouvelle Page". 2. Après avoir sélectionné "Add a New Page" un assistant se lance. Cliquez sur Next, vous devriez arriver à cet écran :



3.

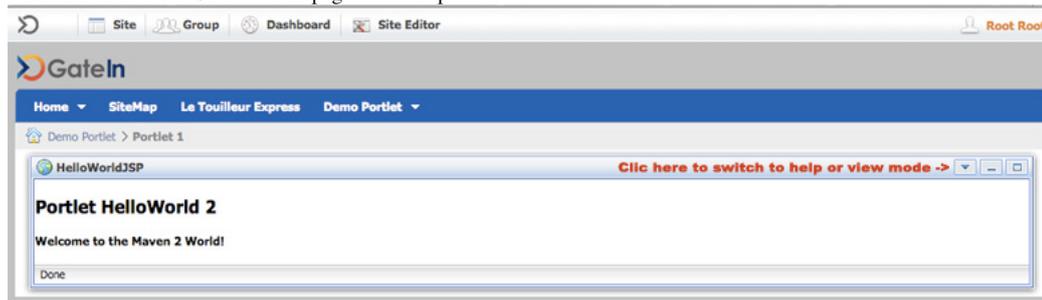
Sélectionnez tout d'abord à gauche le répertoire où la nouvelle Page sera ajoutée. Là je trouve qu'au niveau de l'ergonomie... c'est mal fait. Mais bon, je vais vous expliquer comment faire. Cliquez sur la grosse flèche verte à gauche dans l'arbre de navigation de façon à remonter à la racine des noeuds des catégories. Vous verrez alors que la zone "Selected page node" comme sur ma capture d'écran se vide, ce qui signifie que le noeud racine a été sélectionné.



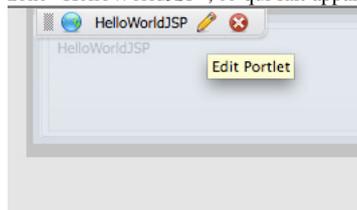
4. Donnez un nom à votre nouvelle catégorie, ici par exemple "touilleurExpress2" puis un nom en clair. Cliquez sur "Next" pour continuer, nous arrivons à la sélection du Layout. Un des points forts de GateIn c'est qu'il est facile de configurer la mise en page de chaque écran en un seul clic de souris. Je décide de conserver le style "Empty" et je continue en cliquant sur Next.



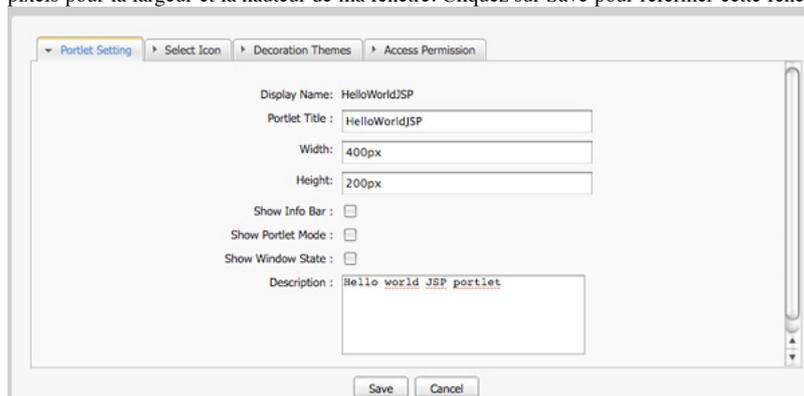
5. Il est temps de déposer notre Portlet. La page qui s'affiche maintenant permet de construire un écran en utilisant des Portlets. GateIn vient avec quelques Portlets en standard, que vous pouvez tester. Ouvrez la fenêtre **Page Editor** en haut à droite en cliquant sur la flèche. Aller dans la section "Demo" de la partie "Applications". Prenez et déposez votre portlet "HelloWorld" dans la zone centrale de la page. 6. Refermez la boîte "Page Editor" en cliquant sur la flèche, afin d'accéder au bouton "Save". Votre page est enfin prête !



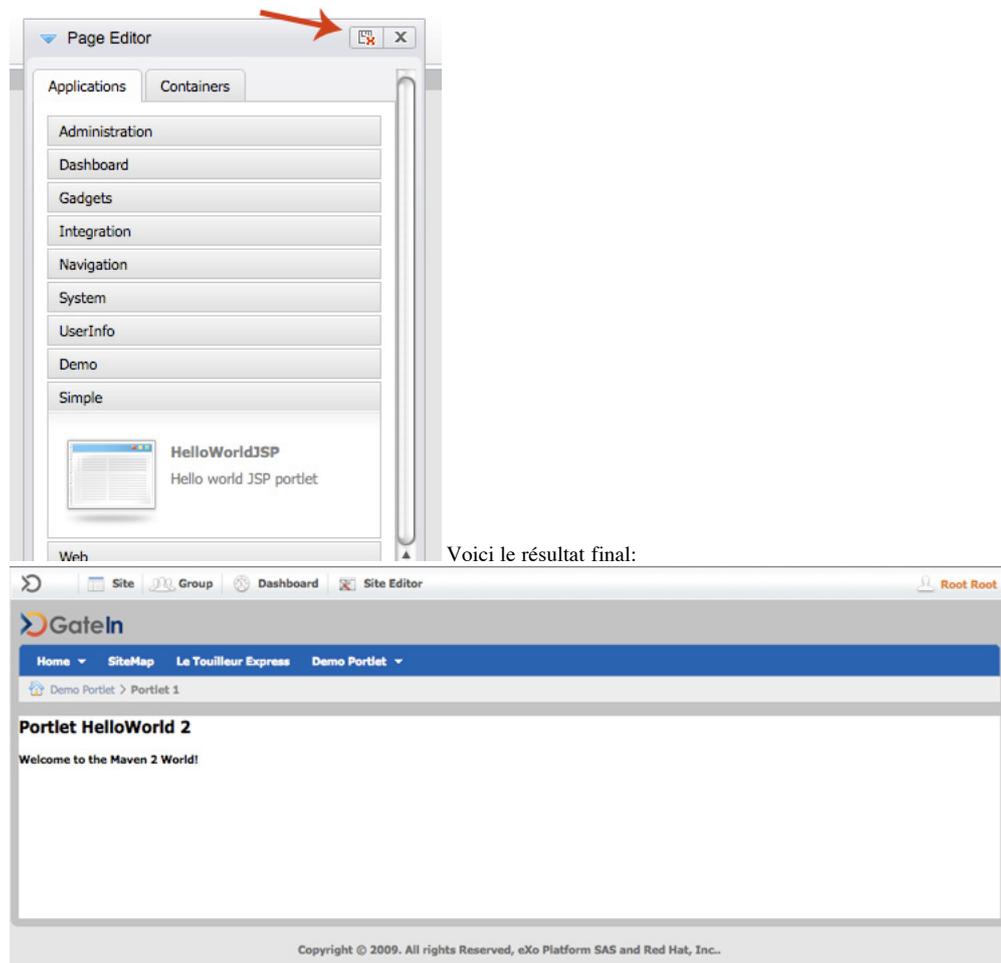
**Masquer l'aspect Portlet** Pour terminer nous allons voir comment masquer l'aspect Portlet, en désactivant l'affichage des contours de notre fenêtre. 1. En tant qu'utilisateur root toujours, cliquez sur "Site Editor>Edit Page" dans le menu en haut de la page. 2. Placez le curseur de la souris sur la zone "HelloWorldJSP", ce qui fait apparaître un menu contextuel



3. Cliquez sur le crayon jaune afin d'éditer les propriétés de la Portlet. Désactivez les 3 options comme sur cette capture d'écran. J'ai aussi fixé une taille en pixels pour la largeur et la hauteur de ma fenêtre. Cliquez sur Save pour refermer cette fenêtre.



4. Enfin pour sauver vos modifications et revenir au mode normal, cliquez sur la petite disquette située en haut et à droite de la fenêtre **Page Editor**



Là je passe une deuxième commande à eXo Platform. Liferay lorsque l'on retire les boutons de navigation, transforme les boutons de la portlet en lien HTML: "Maximize, Minimise". Là pour l'instant dans cette version de GateIn, il n'est plus possible de modifier l'état de la portlet. J'aimerais un lien "Configuration" pour passer à l'écran de configuration de la portlet directement sans passer par le mode édition. Rien de grave bien entendu, il s'agit encore d'une version beta de GateIn.

**Conclusion** Dans ce premier article nous avons vu comment installer GateIn, créer sa première page, déployer une portlet et modifier ensuite les pages du site. Je n'ai pas couvert la partie Java de ma Portlet, nous en parlerons dans un prochain article. Voilà j'espère que cet article vous permettra de commencer à utiliser GateIn.

## Devoxx journée 1: premiers retours

C'est parti, arrivé dans la matinée à Anvers au Metropolis, je retrouve déjà quelques têtes connues de la communauté Java francophone. Lundi et Mardi, Devoxx débute par la partie Université. Le format est plutôt original. Les présentations se déroulent sur 3 heures, avec une pause. C'est un format peu commun qui permet de plonger dans un sujet, et qui laisse du temps au présentateur.

**Les présentations de cette première journée** J'ai assisté à la présentation de Chris Richardson de SpringSource sur Amazon EC2 et CloudFoundry intitulée "[Architecting Robust Applications for Amazon EC2](#)" puis ensuite 2 petites présentations de 30 minutes pour terminer cette première journée. La première présentation "Introducing Scimpi" présente un outil qui s'interface avec le framework Naked Objects. Quoiqu'intéressante, je ne vous parlerai que du principe des Naked Objects qui a retenu mon attention. La deuxième présentation à laquelle j'ai assisté était sur le framework de test "[Spock](#)". Très intéressant, et cela mérite un article plus complet. **L'ambiance** Un peu moins de monde semble-t-il cette année, avec 2650 personnes, et 36 pays représentés. Les thèmes majeurs sont le Cloud Computing, JEE6, JavaFX et Adobe qui a effectué plusieurs présentations aujourd'hui. Autour de JEE6, une session sur JSF2 qui a fait salle comble. Des thèmes plus connexes comme SOA, ESB, l'architecture dans le Cloud sont aussi pas mal représentés. Il faut noter, en particulier vendredi prochain, la présence de plusieurs sessions sur JavaFX, dont une sur les jeux vidéos. Une bonne première journée, et la soirée n'est pas terminée, il n'est "que" 18h13 Ce soir j'aimerais vous livrer mon retour sur Amazon EC2, une très bonne présentation qui ouvre un peu les yeux pour ceux qui se lancent et qui veulent créer leur startup. C'est décidé, ma prochaine compagnie n'achètera pas de serveurs. A vous les studios ! PS: et le Wifi marche très bien cette année

## Eclipse DemoCamp à Paris

Mardi 17 novembre (demain) aura lieu le premier "Eclipse DemoCamp" à Paris à l'Epitech. L'Eclipse DemoCamp Paris 2009 accueillera des contributeurs et experts des technologies Eclipse

qui donneront des démonstrations de leurs projets. Les présentateurs, leurs projets : \* Olivier Moïses nous présentera Wazaabi 2, un framework Open Source d'interface graphique déclarative live. \* Jonathan Musset d'Obeo nous présentera Eclipse Acceleo 3.0 et Obeo Designer, un framework de construction de modèles graphiques par simple paramétrage \* Sebastian Zarnekow d'Itemis nous présentera Xtext, un framework permettant de développer des langages textuels spécialisés (Cette présentation sera en Anglais) \* Yves Yang de Soyatec nous présentera XWT, un framework d'interface graphique déclarative en XML \* Laurent Delvaux de Zenika nous présentera BIRT, une solution de reporting Open Source Cet évènement s'inscrit dans le cadre de la série des Eclipse DemoCamps promu par la fondation Eclipse et donnés partout en Europe tout au long du mois de Novembre :

[http://wiki.eclipse.org/Eclipse\\_DemoCamps\\_November\\_2009/Paris](http://wiki.eclipse.org/Eclipse_DemoCamps_November_2009/Paris) L'évènement aura lieu le mardi 17 Novembre 2009 de 18H30 à 22H00 dans les locaux de l'Epitech, au 24 rue Pasteur, 94270, Le Kremlin-Bicêtre : <http://eclipse-democamp-paris09.eventbrite.com/>

## Devoxx 2009 - Amazon Web Services

Chris Richardson de SpringSource a présenté l'offre de Cloud Computing d'Amazon appelée "[Amazon Web Service](#)" pendant 3 heures. Cette présentation fut très intéressante, et nous laisse vraiment avec pleins d'idées en tête, une grosse envie de sortir la carte bleu et de débiter dès ce soir quelques tests avec Amazon. Qui est Chris tout d'abord ? C'est un Java Champion comme Antonio Goncalves. Il a fondé la société [CloudFoundry](#) qui a été racheté par SpringSource... qui a été racheté par VMWare. Avec un talent certain pour présenter son sujet, il a animé sa présentation en expliquant aussi les limites de la solution d'Amazon Web Service, ce qui est appréciable. Je vous parlerai de [CloudFoundry.com](#) à la fin, car pour l'instant il commence la présentation et je ne veux pas vous faire loucher un instant. Les lumières s'éteignent, les premiers slides de Devoxx 2009 commencent à défiler... Introduction à la mode Touilleur Express

Commençons par quelques faits : regardons l'infrastructure IT du point de vue de l'Entreprise et de la Startup. Dans le monde de l'Entreprise, l'achat de serveurs pour héberger une application est un processus long. De même, l'installation de Weblogic, puis la configuration d'une base sur Oracle prend parfois plus de temps qu'envoyer un pigeon voyageur à un Corse. A croire que les logiciels qui nous font gagner du temps, demandent en retour un temps d'invocation énorme pour être installé par votre équipe IT. Alors de votre côté votre application Java est prête. Ce nouveau moteur de réservation des skis et des surfs pour la saison de février vous a demandé quelques semaines d'efforts. Par contre un souci : vous n'avez aucune idée de la puissance machine nécessaire, que ce soit le CPU ou le nombre de machine. C'est vrai qu'en juillet par contre, votre site de vente en ligne ne risque pas de servir à grand chose. A quoi bon alors commander 4 serveurs Solaris et 1 Oracle RAC pour un usage encore inconnu ? Enfin vous souhaitez tester votre application en charge. Une question : comment comptez-vous faire ? Est-ce que vous avez le luxe d'avoir une deuxième base clusterisée avec Oracle RAC pour les tests ? Certainement que non. Est-ce que de toutes les façons vous avez ne serait-ce qu'un serveur pour faire vos tests ? Non. Vous allez attendre le 1er février, quitte à faire un sacrifice rituel pour vous assurer les bons hospices d'un dieu magique... mais entre nous, vous ne savez même pas si votre application de vente de skis ne va pas se planter dans la poudreuse le 1er février au matin. A travers ces 3 paragraphes, nous avons déjà 3 motivations qui nous poussent à regarder le Cloud Computing en général, et le principe de l'infrastructure comme un service en particulier (IaaS).

**Réactivité** Tout d'abord le premier argument de la solution d'Amazon: la réactivité. Vous souhaitez 4 serveurs Tomcat+MySQL pour tester le mode cluster ? C'est possible en 10 minutes avec une Carte Bleue et un compte Amazon. Dans sa démonstration, Chris utilise le plugin "[Elastic Fox](#)" pour Firefox pour démarrer une instance sur la plateforme d'Amazon. **Même pas peur du 1er février** Faire héberger son application sur une plateforme comme Amazon EC2 c'est certainement s'assurer des nuits calmes. Pour peu que votre site fonctionne à mort le 1er février, vous serez en mesure d'ajouter des machines et de migrer votre application vers une machine plus puissante si nécessaire. Gardez le mot "migration" dans un espace mémoire, je vous en reparle tout à l'heure.

**Faire des tests** Un luxe mais un des points forts de la plateforme EC2 d'Amazon, c'est la possibilité de s'installer une ou plusieurs machines, le temps d'effectuer des tests. Pas plus tard qu'il y a quelques mois chez un client, cela me rappelle que j'avais demandé des images Linux sous VMWare pour travailler, chose que j'ai obtenu assez rapidement (10 jours) mais qui m'a demandé ensuite un certain temps pour l'installation et la configuration. L'intérêt de la solution d'Amazon EC2 c'est aussi de vous proposer un kit "pied sous la table". Par exemple une soudaine envie d'un Tomcat 6 sur Ubuntu pour tester votre application à 1h du matin ? Oui c'est possible. **Après les 35H, faut-il passer aux 0h dans l'IT ?** L'Entreprise ensuite lorsque son coeur de métier n'est pas l'informatique, doit-elle encore s'offrir le luxe d'avoir des machines dédiées ? Des personnes pour l'administrer ? La question mérite d'être posée. Selon Chris, les serveurs informatiques et les applications sont assez matures pour être maintenant externalisés et gérés comme un moyen de production externe au monde de l'entreprise. Il cite une étude où le premier critère qui freine l'adoption du Cloud Computing est la sécurité à 43% des réponses, suivi à 28% en raison de problèmes pour s'intégrer avec des applications existantes ([voir aussi cet article](#)). Et là, cher lecteur, je te demande un exercice de représentation. Est-ce que tu préfères déployer ton WAR sur Amazon EC2, un gars bien sous tout rapport, ou est-ce que tu préfères le donner à un prestataire externe recommandé par ton copain de chez IBM, prestataire qui aura peut-être disparu dans 6 mois, qui sera lui-même racheté par GrosPrestataire avec tout le bonheur et la qualité de service qui suit ? La sécurité du service, sa qualité, est-elle meilleure à votre avis chez Amazon ou dans votre département IT de 6 bonhommes ? Dur... mais vrai. Amazon EC2, sachez-le, concernant le problème de l'intégrité des données est certifié [SAS70](#) Type II. Ils sont aussi compatibles avec la norme [HIPAA](#) pour protéger la vie privée et le secret médical des personnes dans le monde de la Santé. Bon, je vous l'avoue : je ne connais pas ces normes. Mais à l'air sérieux de Chris, on sent que les petits gars d'Amazon pourraient héberger les plans de la nouvelle douche de l'Elysée. La sécurité par rapport au Cloud c'est avant tout un problème émotionnel. Moi j'ai plus confiance en Gmail qu'en mon Mac à la maison. On peut me voler mon Mac ou celui-ci peut tomber en panne. Je préfère que mes emails soient dans le cloud, comme c'est le cas avec Gmail. Alors arrêtons de

chipoter et passons à la suite. **Si tu veux faire une Startup** J'ai bien aimé cette partie, où il nous a amené à réfléchir à l'intérêt de la solution de Cloud Computing d'Amazon pour les startups. Avant tout, il explique qu'aux USA, certains fonds d'investissement ne regardent pas votre dossier si votre projet d'application Web est hébergé sur une plateforme classique. A votre avis pourquoi ? Et bien imaginez que votre service Web marche vraiment bien : vous êtes mort. En effet, le pire scénario pour une Startup qui réalise un service sur le Web serait... que celui-ci marche trop bien très rapidement. Imaginez la catastrophe : vous lancez votre startup, un article passe au journal de 20 heures, et boom votre serveur tombe, écroulé sous le nombre de requêtes. C'est ce qui est arrivé à l'IGN et à son [GéoPortal](#). Et même si le site fonctionne aujourd'hui très bien, que dire de son image ? Allo ! nous sommes en 2010, et un site avec 10 000 visites par mois c'est petit (c'est le *Touilleur Express*). Il y a beaucoup plus d'internautes au 21e siècle, pensez-y avant de vous lancer. Repensez aussi à une Startup: il y a 4 développeurs, et votre temps est le bien le plus précieux. Avez-vous vraiment envie de perdre votre temps à configurer un Linux avec le load-balancer d'Apache, puis le tuning de Tomcat avec MySQL là où des centaines de personnes l'ont déjà fait pour Amazon ? Et bien pas moi. Mon métier c'est développeur Java, pas ingénieur système au fait des dernières bidouilles sous Linux. Enfin au départ, une machine c'est largement suffisant. Pour peu qu'ensuite vous souhaitez tester une nouvelle version, que faut-il faire ? Et bien louer une deuxième machine sur Amazon, et ne pas toucher à l'ancienne version. Oui, tu vas gérer les upgrades de version en louant de nouvelles machines, car cela te coûte moins cher que de faire le kéké à upgrader à minuit ton unique et pauvre serveur chez Gandi.net. Il fallait aussi y penser. Une fois la version 2 stabilisée et testée, tu pourras sagement éteindre la version 1 pour économiser quelques dollars. Mais est-ce que tu te rends compte qu'il aurait été impossible de faire de la montée de version en passant d'une machine à l'autre sans le Cloud ? C'est la mort de la plateforme de "préprod" qui n'est jamais identique à la production, faute d'argent. Vive le Cloud ! Yes you can ! *Faisons une pause, la foule est en délire, et reprenons un peu de notre sérieux (légendaire) IaaS ?* C'est quoi IaaS ? La définition de l'Infrastructure as a Service: c'est une pile de serveurs hautement disponible gérée par un tiers, sur lequel vous déposez votre application. [Amazon EC2](#) est une infrastructure d'hébergement qui vous permet de louer des machines. Le nombre de machine ainsi que leur puissance peuvent être décidés à tout moment. La demande de machine s'effectue via une librairie de programmation (une API). A l'aide de simples messages, il est possible de créer et lancer une instance Amazon en quelques minutes. Les mots clés : flexibilité et location, puissance adaptable. Le monde de l'Entreprise est handicapé lorsqu'il doit décider du nombre et de la puissance des serveurs. N'importe quel architecte vous le dira : c'est du pifomètre. Combien de machines sont trop ou pas assez puissantes dans votre entreprise ? Il y a un gâchis de ressources et d'argent. Un autre effet positif du Cloud Computing : il vous permet de louer 5 serveurs par exemple le week-end, puis d'en rendre 4 pour n'en garder qu'un seul pour la semaine. L'exemple cité par Chris est un brasseur de bière américain qui a utilisé ce principe durant la saison du Super Bowl. Chaque week-end, des publicités sur les TV américaines faisaient venir beaucoup de visiteurs, mais la semaine, faute de matches, il n'était pas nécessaire de garder autant de serveurs. Les coûts sont réduits et les risques sont limités. En fin de présentation, Chris a parlé de CloudFoundry qui vous aide justement à ajuster ces paramètres, sans devoir passer votre certificat Amazon pour s'y retrouver dans l'API. Le premier prix de la plus petite machine d'Amazon est de \$0,085/heure. Pour ce prix, vous avez un serveur monoprocesseur 32 bits avec 1.7 Go de RAM, 160 Go d'espace disque, et un processeur d'1Ghz environ. C'est déjà trop pour des petits serveurs, il manque un bundle plus petit et plus économique. Cela représente environ 61 \$ par mois, ce qui peut sembler beaucoup par rapport à une solution dédiée... mais attendez de voir les avantages. L'offre d'Amazon c'est donc [Amazon Elastic Cloud Compute \(EC2\)](#), une plate-forme de virtualisation des serveurs sur laquelle on dépose des images systèmes. Il est aussi possible de prendre des packs préconfigurés par Amazon et aussi par la communauté avec un serveur linux + un tomcat + java + mysql par exemple. C'est aussi une offre de stockage avec Amazon S3, un moteur d'échanges de message à la JMS avec SQS, une base de données non relationnelle dans le Cloud avec SimpleDB, et un hébergement localisé pour les médias avec CloudFront. Chris a présenté pendant environ 1 heure la partie sur Amazon EC2, puis ensuite nous a montré comment développer une application d'hébergement de photos avec l'ensemble des services d'Amazon. Amazon EC2 est un service dans le cloud qui permet de démarrer une activité sur Internet, mais qui peut aussi être utilisé dans le cadre de l'entreprise. **Amazon EC2 bien expliqué** Amazon Elastic Cloud Computing (EC2) est une solution d'hébergement sur Internet qui fonctionne sur la base d'images systèmes virtuelles, une Amazon Image (AMI). Une AMI est une image d'une machine virtuelle, stockée sur Amazon S3 dont nous parlerons plus loin. Après avoir sélectionné le type d'image, Chris démarre ce que l'on appelle une instance. L'instance se voit allouée un certain nombre de ressources : cpu, mémoire et espace disque virtuel. Le contenu des disques est perdu en cas de crashes ou de redémarrage de l'instance. En fait une instance ne peut vivre qu'une fois, comme une Thread Java. Par contre, si vous faites rebooter le système d'exploitation qui tourne dans cette image, les disques ne sont pas perdus. Il existe plusieurs types d'instances, et il est possible de sélectionner la zone géographique où l'instance sera lancée. **Les outils** Pour commencer à utiliser Amazon EC2 il faut une Carte Bleu, un compte Amazon et c'est tout. Une API orientée service permet de créer une instance, de la contrôler, de l'arrêter. En fait, cette API est maintenant masquée par des outils comme Elastic Fox, ou des plugins dans Eclipse et IDEA IntelliJ. Vous n'avez pas besoin d'apprendre l'API. Par contre, il faut quelques connaissances sous Unix, sur SSH et sur le principe des clés publiques et privées. Rien de bien compliqué, mais c'est assez bas niveau. La machine la plus puissante que vous pouvez louer est un monstre 64 bits avec 128Go de mémoire. Là on parle de 560\$ par mois. Et le top du top, c'est que si vous voulez 10 machines de cette puissance, c'est possible ! En fait la limite du Cloud Computing c'est votre Carte Bleu, plutôt que le matériel lui-même. Tout le modèle de facturation est basé sur l'utilisation, avec [des prix à l'heure](#). L'idée est proche du domaine des téléphones portables. Vous téléphonez peu, vous prenez 2h. Vous êtes un gros consommateur, vous prenez 6H. Ici on parle plutôt de puissance machine, de la taille de la mémoire et de l'espace disque. Un slide dans la présentation montre un exemple avec 3 machines : 1 serveur web sur Apache, 2 serveurs sur Tomcat en cluster. Le tout pour 180\$ par mois environ. Et comme Chris l'explique : si vous avez envie de changer de machine, ne serait-ce que pour tester pendant une semaine, c'est aussi faisable. **Le Web tiers** Dans cette partie, Chris nous a montré des exemples de déploiement avec un serveur Apache. Difficile

ici de vous retranscrire les quelques 8 slides. En bref nous avons parlé d'adresse IP static, d'adresses privées pour le réseau Amazon et d'adresses publiques. Il a aussi montré comment créer une petite infrastructure avec un serveur Web rapidement. Concernant le Load-Balancing, il existe une solution proposée par Amazon. Il suffit de configurer l'adresse de l'Elastic Load-Balancer acheté avec votre compte en tant que CNAME sur la configuration DNS de votre domaine, afin de load-balancer les visiteurs sur plusieurs machines. Attention cependant, le SSL et les Sticky Sessions ne fonctionnent pas avec ce LoadBalancer. Mais c'est simple à installer. **Les serveurs d'applications** Dans cette partie, nous avons parlé Tomcat essentiellement. Depuis que la société de Chris s'est fait racheter d'abord par SpringSource puis ensuite VMWare, on sent la touche SpringSource avec quelques citations de ce Server. Mais sans plus. Il n'y a pas de Multicast dans l'offre EC2. Si votre serveur d'application utilise le multicast pour détecter ses voisins, comme avec JGroups, il faut alors soit configurer JGroups en mode TCP, soit utiliser Terracotta pour faire tourner Tomcat en le configurant correctement par exemple. Concernant la montée en charge, Amazon propose depuis mai 2009 un service appelé "Amazon Auto Scaling". Comme son nom l'indique, c'est un moteur capable de démarrer automatiquement de nouvelles instances lorsque la charge CPU ou mémoire devient trop importante sur un de vos serveurs. Vous pouvez limiter le nombre d'instances bien entendu. Ce système cependant présente quelques limites : tout d'abord Amazon regarde des metrics bas niveaux comme le CPU ou la mémoire, mais n'est pas capable d'utiliser JMX pour diagnostiquer l'application. Ensuite, même si le démarrage de nouvelles instances est automatique, il faut encore penser que la configuration de votre application (adresse IP de la base par exemple) n'est pas automatique. Cela demande un travail de configuration pour que tout fonctionne. **La base de données** Comment gérer une base de données lorsque l'on travaille avec Amazon EC2 ? Première mauvaise nouvelle : l'espace disque de la machine qui vous est alloué ne doit pas être considéré comme un espace qui sera toujours là. Si votre machine plante puis qu'elle redémarre, le système de fichier sera complètement initialisé. Cela pose un gros problème si vous souhaitez sauvegarder votre base en local sur la machine : c'est impossible. Pour répondre à ce problème, Amazon propose EBS (Elastic Block Storage). Sous ce nom digne d'une crème épilatoire, se cache en fait comme un disque réseau, qui sera monté sur l'image Amazon EC2 et qui sera sauvegardé. Similaire à un NAS en fait. EBS est automatiquement divisé sur plusieurs disques physiques. D'après Chris, c'est 10 fois plus solide qu'un disque physique classique. EBS coûte \$0.10 par mois, pour 1G. Il est possible de prendre des snapshots pour sauvegarder ce disque et de les placer sur S3. Par contre côté performance, d'après lui c'est largement en dessous d'une solution comme du RAID 10 avec 2Gb de cache, bref un NAS à 300 000 EUR. Mais c'est suffisant pour la majorité des cas. Concernant le monde des bases relationnelles, qui devrait disparaître bientôt, Amazon propose aussi "[Amazon Relational Database Service](#)" ou RDS. Il s'agit d'un serveur MySQL 5.1 configuré par Amazon pour vous. Ce service est administré par Amazon. Lors de la création, l'interface Web vous demande même de donner 4h dans la semaine pendant lesquelles Amazon peut éteindre ce service pour passer des patches de sécurité. L'interface demande aussi de donner un créneau de 2h dans la journée pour qu'Amazon effectue des sauvegardes de la base. Bref c'est un MySQL configuré, backupé, géré par Amazon, qui vous évite de vous ennuyer avec la configuration et la sauvegarde de votre base. Moi je dis : pourquoi pas ? **La sécurité** Bon j'espère que vous n'avez plus peur d'utiliser Amazon EC2 maintenant. Si c'est encore le cas, visualisez la tête de votre Chef de Prod. Chez mon ancien client, c'est marrant, mais le responsable de la production était tellement aimable qu'un hygiaphone de la Poste fait rêver à côté. Mon fils adore la grosse méchante dans *Monster*, vous voyez le poule ? Et bien c'est le chef de la prod. Bon j'en étais où moi ? Concernant la sécurité, il est possible de régler finement les droits d'accès à chacune de vos machines. SSH, HTTP ou FTP, n'importe quel port se règle afin de sécuriser votre réseau. **Le déploiement dans le monde** Il existe des projets pour lesquels il n'est pas possible légalement d'héberger aux USA des données. Pour cela, il existe depuis maintenant 2 ans des Datacenters Amazon en Europe. L'Asie est prévue pour 2010. Cela permet de rapprocher géographiquement vos serveurs de vos utilisateurs, et de rentrer aussi dans le cadre de la législation sur l'export du code source d'un logiciel. Si vous n'avez pas ce genre de soucis, Chris recommande de répartir sur chacune des 2 zones en Europe son Infrastructure. Cela rapproche la disponibilité de votre service des 99,95%, ce qui est tout à fait supérieur à beaucoup de services. **Critiques sur IaaS** Un souci qui saute aux yeux avec une infrastructure de type IaaS, c'est que cela demande encore un peu de temps pour tout ce qui est installation et configuration. Si cela vous freine, Chris parle alors de PaaS : Platform as a Service. Premier exemple dont je vous parlerai cette semaine : Google AppEngine. L'avantage ? Pour vous, plus d'installation d'images, de réglages de Linux ou autre : vous utilisez un méga serveur virtuel et vous ne payez que l'utilisation. Cependant Google AppEngine a quelques limitations : pas de base de données relationnelles, les transactions sont très limitées et surtout, l'API Java n'est pas complètement disponible. Chris a créé CloudFoundry il y a 2 ans, pour justement aider le développeur Java qui souhaite utiliser Amazon EC2 et ses différents services, sans avoir le temps ou l'envie de se plonger dans l'API d'Amazon. Je trouve son offre pleine de sens. Elle répond à un vrai besoin. De plus, Chris nous a montré des installations types proposées par son outil. Cela permet réellement de créer une infrastructure avec des serveurs configurés en cluster en quelques minutes. Pensez-y si vous avez de l'Apache, du Tomcat, du MySQL mais aussi d'autres serveurs. Ainsi se termine les 2/3 de la présentation. Nous allons maintenant passer à un résumé des différents produits de l'offre d'Amazon dans la deuxième partie.

## Devoxx 2009 - Amazon Web Services partie 2

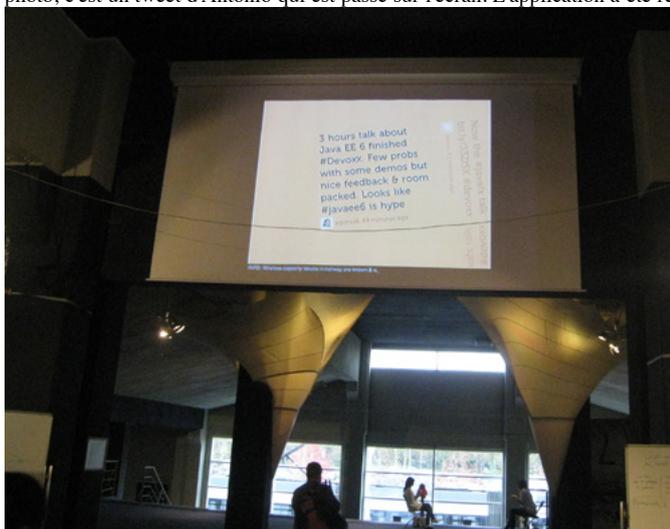
([suite de l'article sur Amazon Web Service](#)) La dernière heure de la présentation de Chris Richardson est consacrée à une revue des principaux services d'AWS, en présentant une application de stockage de photo à la Flickr. Une application Web Java sera déployée sur un serveur Tomcat sur une instance Amazon EC2. Le stockage des photos s'effectue sur Amazon S3, un espace de stockage que l'on peut consulter avec une URL. Des vignettes sont générées par un serveur de traitement des photos qui utilise Amazon SQS (Simple Queue System) pour échanger des messages asynchrones avec notre application web. La base de données de l'exemple sera Amazon Simple DB, qui n'est pas une base relationnelle. Il a consacré la dernière heure à passer en revue

chacune des technologies, c'était vraiment intéressant. **Un peu de théorie** Je vais vous donner quelques éléments de théorie. Cela vous permettra de briller en société, et votre ArchitecteEnChef aura le bec cloué. Tout d'abord parlons du théorème énoncé par le [Dr Eric Brewer de Berkley](#) : the CAP Theorem où CAP signifie Consistency, Availability et Partition tolerance. Il est établi qu'une application web ne peut pas à un instant donner réunir et couvrir les 3 points suivants : - Consistency : une donnée écrite dans le système peut être lue par l'ensemble des lecteurs. Une base relationnelle basée sur les principes ACID est l'exemple absolu de la consistance des informations. - Availability : le système reste disponible même lorsqu'une opération est en cours - Partition tolerance: le système fonctionne complètement malgré l'arrêt d'une partie du système Amazon couvre le A et le P du théorème : la donnée lue est garantie comme étant correcte et complète, et le système peut fonctionner et retourner vos données même si une partie est en panne. C'est donc la consistance de la donnée qui est sacrifiée. En clair voici comment cet effet sera observé sur Amazon S3 par exemple. D'un côté, vous déposez dans un espace une image. De l'autre, avec votre navigateur, vous chargez l'URL de l'image. Et bien il y aura certainement un décalage, l'image ne sera pas trouvée tout de suite de l'autre côté. En effet, Amazon va d'abord répliquer votre donnée, et va aussi s'assurer de son intégrité. Lors de l'envoi vers Amazon S3 d'une ressource, il est maintenant possible d'attacher un checksum MD5 dans le header HTTP d'un simple PUT utilisé lors de l'envoi de l'image. Cela permet à Amazon de vérifier que la donnée n'a pas été endommagée lors du transport. Chris affiche un slide explicite : "A GET might not set a POST". Concernant SQS, proche de JMS dans l'idée, un message posté ne va pas apparaître immédiatement du côté de la queue de lecture. Pour relativiser cela, il faut penser que dans bien des cas, ce n'est pas grave. Concernant notre application d'album photo, lors de l'envoi vers Amazon S3 il sera donc normal de ne pas voir apparaître tout de suite les photos. Il explique en fait que le système ne peut pas proposer de transaction ACID mais il parle plutôt de BASE : *Basically Available Soft-state Eventual consistency*. **Amazon S3** [Amazon S3](#) est la solution de stockage de la plateforme Amazon Web Services. Vous pouvez y mettre vos images, vos données statiques, les données sont stockées très simplement. L'intérêt de S3 c'est la haute disponibilité de vos données. Les chances de crashes et de perdre de la donnée sont presque nulles. Pour utiliser S3, il faut créer un Bucket, dans lequel sera rangé sous la forme de clé/valeur les données. Les données peuvent aller de quelques bytes à 5Go maximum. Un bucket est un nom unique qui permet ensuite de créer une url de la forme `http://s3.amazonaws.com/`. Chaque compte Amazon permet de créer jusqu'à 100 buckets, et il semble que le nombre d'objets stockés par Bucket ne soit pas limité. Les données sont ensuite stockées comme sur un système de fichier virtuel. Par exemple `http://letouilleurbucket.s3.amazonaws.com/logo.png`. La création des Buckets en Java est facile grâce à l'API JetS3t qui permet de manipuler facilement Amazon S3 directement à partir de votre code Java. Pour Grails il existe aussi un plugin. L'accès au bucket peut aussi être protégé afin d'en restreindre l'accès. Une API avec des ACLs permet de contrôler les droits de lecture et d'écriture sur les objets du Bucket. La facturation est calculée sur la base du volume stocké, que multiplie la quantité transféré. Un GET est par exemple 10 fois moins cher qu'un PUT ou un POST. L'opération DELETE est gratuite. Il existe aux USA une solution d'import où vous envoyez votre disque dur par FedEx. Amazon le charge dans un espace S3 et vous le retourne par transporteur. Cela permet de remplir son espace S3 très rapidement, sans payer des coûts prohibitifs. **CloudFront** [CloudFront](#) est un cache de données qui se géolocalise selon la demande des visiteurs. Lorsque les données sont stockées sur un serveur en Europe sur S3, les utilisateurs aux USA seront pénalisés par rapport aux performances. Pour répondre à cela, Amazon propose CloudFront. C'est un vrai "Content Delivery Network" capable d'agir de manière transparente, comme un proxy du côté de l'infrastructure d'Amazon. Si par exemple votre planche de Surf Jaune fait un tabac aux USA, l'image sera copiée vers un serveur S3 aux USA de manière transparente. Et l'image sera servie via cet emplacement pour l'Amérique du Nord, pendant 24h. Cela permet d'améliorer les performances de manière transparente. Ce service additionnel est aussi à régler à l'heure selon l'usage. **SimpleDB** [SimpleDB](#) est une base non relationnelle sans schéma. En bref : SGBDR tu peux mourir. L'intérêt ? La performance et la montée en charge mon bon monsieur. Voici comment s'organise SimpleDB. Tout d'abord les "Domaines" limités à 100 par instance de SimpleDB, représentent vos tables. Un Domain a un nom et contient des Items. Des Items ont un nom et des Attributes. Enfin les Attributes ont un nom et une ou plusieurs valeurs. Il n'y a pas de jointure, pas de transaction, et pas de locking possible... Forcément là vous avez un peu peur. Vous pouvez utiliser la syntaxe SQL de base, les données stockées dans SimpleDB sont automatiquement indexées. Mais comment faire une jointure alors ? Les relations de 1 à N par exemple seront représentés par ce que Chris appelle des tables dénormalisées. Il suffit de stocker la même donnée plusieurs fois dans différents Domaines. Le contrôle de l'intégrité sera effectué par l'application. Comment faire des requêtes du type : "Toutes les photos avec Ville=PARIS" ? Il suffit de créer des tables PhotosPriseAParis et d'y stocker toutes les photos prises à Paris. En fait, là où nos réflexes SQL nous permettraient de faire tout et n'importe quoi, ici il faut prévoir les types de requêtes que l'interface pourra envoyer, créer des structures de stockage et y ranger vos données. J'avoue que j'ai du mal à me voir créer autant de tables que par exemple des critères de tris dans un tableau... Mais c'est ce que fait tout les jours un site comme eBay pour vous montrer la liste des articles par exemple. L'idée du Cloud Computing est de dire que c'est au moment de la création d'une entité qu'il faut mettre à jour le système, et qu'il n'y a plus la magie de la base relationnelle. Bon cette partie mérite un article à elle toute seule, je vous mets cela sur la commande pour la prochaine fois. **Amazon Simple Queue Service** Pour ajouter de l'asynchronisme à une architecture, Amazon propose [SQS](#). Similaire quoique plus simple que JMS, ce système permet d'envoyer et de recevoir des messages entre machines dans votre propre réseau Amazon. Les messages sont limités en taille à 8kb et sont effacés après 4 jours s'ils ne sont pas lus. Si vous voulez envoyer une grosse photo... vous ne pouvez pas. Il faut la stocker sur S3, envoyer l'URL dans un message, la récupérer dans votre service de découpage des photos avant de réécrire dans un autre endroit sur S3 le fichier découpé par exemple. Chris montre l'API de SQS, qui me paraît plus simple que JMS. La redondance est le point fort de la solution, il est possible de lire plusieurs fois un message sur une Queue (principe du Topic). Il n'y a pas de garantie sur l'ordre d'arrivée des messages (principe CAP vu plus haut). **La GROSSE conclusion** Cette présentation de 3 heures est passée relativement vite. Pour moi, c'est clair, mon projet de Startup sera sur Amazon ou ne sera pas. Je n'imagine pas prétendre arriver à la cheville de ce qu'Amazon propose

aujourd'hui. J'ai aussi bien aimé la présentation rapide de CloudFoundry. Ce service permet de créer rapidement son réseau avec quelques machines, et de travailler sans perdre une semaine pour la configuration. Dans les idées discutées par Chris, l'une d'elle m'a complètement séduit : la gestion des nouvelles versions. Imaginons que votre site de vente de skis hébergé sur 5 machine fonctionne très bien, mais que vous souhaitez maintenant passer à 10 machines. Grâce à la souplesse du principe de l'IaaS, vous allez mettre en place 10 nouvelles machines, tester avec les adresses IP votre application, puis faire la bascule sans même éteindre les 5 anciens serveurs. Si au bout de quelques jours tout fonctionne, vous pourrez alors arrêter et effacer les 5 anciennes machines. Chaque montée de version peut donc s'effectuer sur une nouvelle machine toute neuve ! Les données statiques sont stockées sur S3, la base sur SimpleDB ou avec l'offre MySQL d'Amazon, bref à part la partie Web et votre serveur d'application, j'ai oublié quelque chose ? Très bonne présentation, je testerai Amazon EC2 rapidement et je vous en reparlerai.

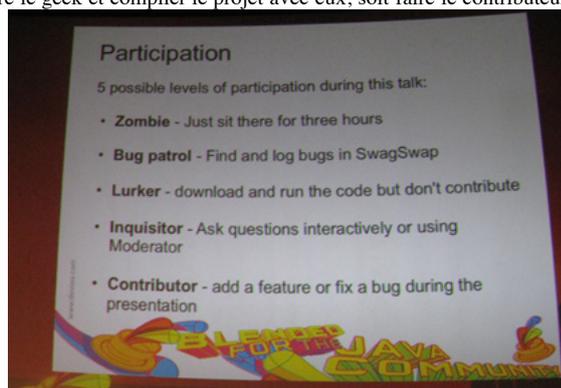
## Devoxx 2009 - journée 2, le résumé

**Résumé de cette journée :** J'ai débuté par 3h de présentation sur Google App Engine, un repas sympa, puis 3h écourté sur Pentaho Business Intelligence. Ensuite un tour des stands puis 2 sessions de 30 minutes : [JavaMonitor](#) et [JetBrains TeamCity](#). J'ai fait aussi un passage à la session sur JEE6 d'Antonio et d'Alexis ce matin, qui a fait salle comble. Rencontres aujourd'hui avec les gens de JetBrains, de JFrog Software, de RedHat, de SpringSource. Discussions aussi avec des personnes de la communauté comme Romain GUY (Google Android), les JUGs de France, bref une bonne deuxième journée. Il n'est que 18h20 et la soirée n'est pas encore commencé. Il reste des petites conférences jusqu'à 22h00. Qui a dit chargé ? Un écran géant est installé cette année dans le hall supérieur, avec les derniers messages Twitter tagués "#devoxx". Au moment où j'ai pris cette photo, c'est un tweet d'Antonio qui est passé sur l'écran. L'application a été réalisé en Flex.



Un autre truc

marrant, c'est ce slide ci-dessous présenté au début de la session Google App Engine. Les 2 présentateurs proposent plusieurs niveaux de participation à toute la salle. Soit faire la larve et écouter passivement, soit faire le geek et compiler le projet avec eux, soit faire le contributeur



actif, j'ai trouvé cela marrant.

Enfin j'ai mis [sur Flickr](#) toutes les photos et les vidéos de cette journée. Il y a entre autre une photo de l'espace d'exposition avec les stands des exposants. A toute à l'heure !

## Google App Engine à Devoxx 2009, journée 2

**Google App Engine for Java : a real voyage in the Cloud** Cette deuxième journée Devoxx Université débute par 3 heures sur Google App Engine. En fait cette présentation était une revue de code d'un projet développé sur Google App Engine par Sam Brodtkin et Scott Stevenson. Nous avons parcouru le code, tout en regardant le résultat de l'application développée en temps réel. A noter que la session a été très participative. L'ensemble de la salle a navigué sur le site, ce qui était plutôt vivant. L'application **SwagSwap** Scott débute la présentation en expliquant le but de son application, [SwagSwap](#). Tout d'abord un Swag, c'est un de ces petits cadeaux que l'on vous offre

lors des conférences. Tee-shirt, stylo, autocollant, poster, bref tous les goodies qui font le bonheur des Geeks. L'objectif de l'application SwagSwap est de permettre à chacun de lister les goodies de la conférence, puis de noter et de commenter les cadeaux offerts pas les sponsors. L'application est en ligne à l'adresse <http://swagswap.appspot.com> sur Google App Engine. Sujet plutôt sympa pour débiter la journée. Un slide me fait sourire, où Scott littéralement propose à l'audience 5 niveaux de participation : - Faire le Zombie : vous restez assis là pendant 3 heures sans rien faire - Bug Patrol : vous recherchez et vous loguez les bugs trouvés dans SwagSwap sur le site Google Code du projet - Lurker : vous téléchargez le code, vous le regardez mais vous n'y contribuez pas - Inquisiteur : vous posez des questions de manière interactive ou via le site Google Moderator qu'ils ont mis en place - Contributor : vous ajoutez une nouvelle fonction ou vous fixez un bug pendant les 3 heures. Scott commence ensuite sa démonstration de SwagSwap. L'interface sommaire à la Web 1.0 est développée dans la version qu'il sélectionne avec Spring MVC. En fait, Sam Brodtkin a codé une version Spring MVC, une version GWT. Scott Stevenson a codé une version JSF2. La suite de la présentation sera largement consacrée à une comparaison pointue des 3 solutions, ce qui est plutôt intéressant pour nous. **Quelques mots sur Google App Engine** A moins d'avoir fait un stage chez les Mormons, vous ne pouvez pas me dire que vous n'avez pas entendu parler de [Google App Engine](#). En quelques mots, c'est la solution PaaS (Platform as a Service) de Google pour faire tourner dans le Cloud une application Web Java. GAE (Google App Engine) utilise Jetty, et permet de faire tourner avec quelques limitations, une application java. Les avantages de la solution : plus besoin d'installer une infrastructure, même plus besoin de s'amuser avec des images comme avec Amazon EC2 ([voir mon article de lundi](#)). Google monte en puissance pour vous, de manière transparente. Vous ne payez que l'usage réel. Et chose très intéressante, Google ne vous fait payer le service que lorsque vous dépassez un seuil d'utilisation des ressources. Il est donc possible de démarrer un projet sur GAE et de ne rien payer. Google App Engine propose une console d'administration Web qui permet de suivre la consommation de son serveur. Il est aussi possible de voir les Entités de son domaine, de régler quelques paramètres de son application. Il y a cependant quelques limitations qui font qu'il sera peut-être impossible pour vous d'utiliser GAE. Tout d'abord le temps d'une requête Web est limité à 30 secondes. Au delà, Google coupe la connexion. Il n'est pas possible de faire tourner vous-même des tâches de fonds, ni des Threads sur le serveur. Vous n'avez en fait jamais accès à une machine, vous vous contentez de déposer votre WAR avec les outils de Google, c'est tout. Donc pas d'installation de logiciels possibles. Il n'y a pas de serveur Push, le système de fichier est considéré comme étant en lecture seul. Il n'est donc pas possible de sauver un fichier temporaire. Pour envoyer une image vers le serveur, il faut donc ruser et travailler en mémoire avant de stocker l'image. Enfin le point le plus important : vous n'avez pas accès à toutes les classes Java. Google donne une liste blanche des APIs autorisés. Par exemple il n'y a pas d'API Swing, donc il est impossible de retravailler des images sur le serveur. Pour la démo de SwagSwap, ils ont donc utilisé un service de Google pour créer des Thumbnails sur la page d'accueil. Bref tout ceci pour vous dire qu'il y a quelques sacrifices à faire pour que tout fonctionne. Le développeur travaille en local avec une émulation de Google App Engine. Des plugins pour IDEA IntelliJ et Eclipse permettent de packager et d'envoyer vers GAE votre archive, lorsque vous souhaitez faire une mise en production. **La base de données** Les données sont stockées avec BigTable. Le DataStore de GAE est un moteur basé sur un noyau hautement distribué, capable de monter en charge, transactionnel et sans schéma. En fait, BigTable c'est une énorme Hashtable dans le Cloud. GAE supporte JDO et JPA, mais les démonstrations ont été écrites avec JDO, qui semble mieux documenté d'après les speakers. **L'application de base** Le coeur de la partie Service est implémenté avec Spring 3.0, JDO, les transactions et un peu de RESTful avec Spring Web MVC. Ensuite, 3 GUIs différents ont été implémentés. La version Spring MVC a demandé 3 jours de travail. C'est une application Web 1.0 plutôt bien écrite de ce que j'ai vu. La version GWT 1.7 a demandé entre 3 et 4 semaines d'efforts. Il y a cependant un gros travail sur l'ergonomie. Sam a utilisé SmartGWT pour créer une interface plutôt sympathique. Enfin la version JSF2 a demandé 5 semaines, et toutes les fonctions n'ont pas été implémentées. Il a encore peu de projet avec JSF2 sur GAE, donc Scott a pas mal travaillé. Mais le résultat est sympa. Pour compiler et déployer le projet, ils ont utilisé Ant. Le support de Maven2 et de GAE d'après eux, ce n'est pas encore cela. Concernant la gestion du code source et les tickets d'incident, le tout est sur Google Code. **La sécurité** Les utilisateurs sont authentifiés avec les services de Google. Le code est très simple, et il est facile d'appeler les services de Google. Pour protéger les pages, Spring MVC et les filters permettent de déclarer les URLs qui demanderont de l'authentification. L'application a aussi un objet SwagUser avec par exemple les votes et vos commentaires. Cet objet est créé lorsque l'utilisateur s'authentifie la première fois sur l'application. Il est ensuite simplement rechargé si vous revenez. La clé primaire utilisée n'est pas l'email Google mais le Google Account Name. En fait, Sam a découvert que la version GWT envoyait vers le client la liste des emails des personnes ayant postées un Swag sur le site, plutôt gênant. **L'accès aux données** Concernant les données, un tour dans la classe ItemDAO nous montre que le code est plutôt simple. Ils utilisent Spring JDOTemplate, avec un persistenceManager. Les Transactions sont aussi utilisées, mais j'ai vu des méthodes synchronized (comme updateRating). Bizarre. Un souci de Google App Engine et du modèle de persistance : il n'est pas possible de faire des recherches sur plusieurs Entités à la fois. La recherche sur les champs textes est aussi forcément Case sensitive... Bref si vous voulez ajouter un champ "Search" sur votre application GAE, le mieux est de vous tourner vers un projet comme Compass. Celui-ci se place au dessus de Lucene. Il indexe les entités et permet de faire de la recherche full-text sur des Entités. Très pratique, je m'en suis servi sur <http://touilleur.appspot.com> (projet non terminé). Concernant l'authentification, si vous cliquez sur un lien dynamique (GET /view/item?order=toto) et que vous devez passer par la page d'authentification de GAE, la QueryString n'est pas conservée. Il y a un bug sur App Engine ([bug 2225](#)) à ce sujet. **Spring MVC** Des 3 frameworks utilisés, c'est le plus simple et le plus pratique. Nous avons passé un bon bout de temps dans le code. Cette partie était trop axée Spring, et plus tellement GAE. Mais bon, si vous ne connaissez pas encore Spring MVC, il faut vous y mettre. Les annotations et la gestion des RequestMapping par exemple, c'est fort. **Google Web Toolkit 1.7** La version GWT est clairement très Web 2.0. Mais ce que j'ai retenu c'est que trop de Web 2.0 c'est plutôt fatiguant. D'un point de vue applicatif, il n'y a pas grand chose de plus. Oui de l'Ajax, ok. Mais j'ai été un peu inquiet de voir la quantité de code pour un résultat qui n'est pas si énorme que cela. De plus avec 1000 fichiers dans cette petite application, je trouve que l'on se rapproche trop

de la limite des 3000 fichiers statiques par application Google App Engine. Du côté de GWT, le framework SmartGWT est vraiment propre. Rien à voir avec cette bouse de GWT-Ext qu'un client aux USA veut utiliser... **La gestion des images** L'application vous permet d'uploader vers le serveur une photo du Goodie, afin de l'afficher sur la page d'accueil en vignette, et en grand sur la page détaillée du Swag. Mine de rien, cette partie sur le traitement de l'image, pour GWT, JSF2 ou Spring MVC, représente un gros boulot par rapport à une application classique. Sam présente tout d'abord le service de Google qui traite les images, les retaille et vous les redonne. Plutôt bien fait je trouve. Comme il n'est pas possible de stocker les images dans un répertoire, il faut écrire une Servlet qui stream le contenu de l'image vers le navigateur. C'est pas top je trouve. Hier la démo d'Amazon avec le service Amazon S3 pour stocker les images m'a semblé bien plus pro que ce que j'ai vu là. Cela donne des URLs super sexy pour récupérer une image dans l'application. Jugez plutôt:

```

```

Bien entendu, cela pose aussi des soucis avec le cache du navigateur... Si l'image change, le navigateur ne rafraîchit pas l'image. Il semble qu'il ne soit pas possible de changer les headers HTTP et de faire expirer le cache... Je n'ai pas posé la question. **Qu'est-ce qui marche sur GAE ?** La liste des classes Java est limitée. En conséquence, tout les frameworks Java ne fonctionnent pas sur GAE. Par exemple Grails par défaut déclare un log4j avec un FileAppender... ce qui est interdit sur GAE. Regardez cette page pour vous donner une idée de ce qu'il est vraiment possible de faire sur GAE: [Will it play on Google App Engine ? JSF2.0](#) Là j'ai décroché... Ce que j'ai entendu c'est que c'est faisable, mais difficile. Les slides enchaînent les soucis et les limitations, de quoi déprimer. Mais le résultat est plutôt sympa concernant l'interface utilisateur. Le code source dans la partie Java est propre, comparé à la soupe de la version GWT+SmartGWT. Bon si vous voulez voir ce que donne JSF et Google App Engine, le mieux est de télécharger le code du projet et de regarder par vous-même **Autres trucs cités** Scott parle aussi du service d'envoi des emails de Google App Engine. Pratique et simple à utiliser, limité à 1000 mails par jour. Ils effectuent aussi une démonstration de la mise en production sur l'App Engine : simple et clair comme les autres projets Google. Concernant l'asynchronisme, il est possible de configurer des Jobs pour appeler une URL donnée avec une syntaxe à la Cron. Pas du JMS, mais pratique pour envoyer par exemple un email à minuit aux abonnés du site. Une question sur la gestion des backups : il y a un projet appelé GAE bar <http://aralbalkan.com/1784> **Conclusion** En sortant de la présentation, j'ai trouvé que la partie Spring du côté service était vraiment intéressante et adapté à Google App Engine. La version GWT+SmartGWT demande quelques efforts de programmation mais semble plutôt bien écrite. La version JSF2 m'a déprimé et me semble encore trop jeune par rapport à l'architecture de GAE pour l'instant. Google App Engine est très certainement une plate-forme intéressante. Attention cependant à bien se rendre compte que cette plate-forme demande un investissement assez important afin d'apprendre à écrire un modèle de données sans schémas. De nouveaux problèmes apparaissent comme ce cas de la gestion des vignettes du site. Ne pas pouvoir stocker un fichier sur un disque, cela vous force à chercher d'autres solutions. Nous manquons encore un peu de recul, mais ce projet est un très bon point de départ si vous cherchez un exemple complet d'application sur Google App Engine. J'aurai aimé une version Groovy pour comparer, avec Grails ou Gaelyk par exemple. Voilà je n'ai pas tout couvert, mais Romain de [Xebia](#) était aussi là, et donc [le blog de Xebia](#) vous donnera un autre angle sur ce sujet. **Références** [Page du projet SwagSwap sur Google](#) [Code Page d'accueil sur Google App Engine de SwagSwap](#) [La version Spring MVC](#) [La version JSF2](#) [La version GWT 1.7](#)

## Keynote d'ouverture - partie 1 - Devox 2009

Ce mercredi matin la journée débute par la grosse KeyNotes d'ouverture de la partie Conférence de Devox. 8ème édition, Stephan Janssen l'organisateur de Devox nous donne quelques chiffres pour commencer. Devox cette année, c'est 2500 personnes. Moins de monde que l'an passé, en raison de la crise. Il y a 737 sociétés représentés. Stephan met en avant des entreprises qui viennent à 20 personnes comme SFEIR, venu de Paris avec Didier Girard. Comme l'an passé, SFEIR fait un gros effort pour emmener ses consultants à Devox. Devox c'est aussi 217 étudiants, 132 sessions, 120 speakers. C'est aussi le support et la présence de 56 Java User Groups. 39 pays représentés, 19 exposants, cela vous donne une idée de l'ampleur et de l'importance de Devox pour la communauté Java. D'ailleurs si JavaOne venait à réduire le nombre de sessions, il est clair que Devox deviendra la première conférence autour de la plate-forme et de la communauté Java dans le monde. Stephan remercie les sponsors premiums, sans lesquels les frais d'inscription seraient plus élevés : IBM, Oracle, Red Hat et Adobe. Ce qui est marrant c'est que ces 4 compagnies ont du rouge sur leur logo. La suite de la présentation est consacrée à Parleys 3.0. Il s'agit du site sur lequel vous pourrez retrouver toutes les vidéos des conférences de Devox dès la semaine prochaine. Avant cela, Stephan revient sur les premières années de JavaPolis (l'ancien nom de Devox). Il y a 8 ans, les conférences étaient filmées. L'équipe de Stephan proposait ensuite d'acheter la version DVD pour 49 EUR. Le problème du DVD était que cela demandait beaucoup de temps, et que pas mal de DVD ont disparu durant le transport. D'après Stephan, les Postiers pensaient que les DVD étaient réservés à un public adulte... en raison des jaquettes flashys Bref Parleys.com est né. C'est un site qui héberge les vidéos et les slides des présentations. Le site donne accès à toutes les anciennes présentations de JavaPolis et Devox. Pour couvrir les frais d'hébergement, Parleys 3.0 propose une offre de souscription qui, pour 49 EUR pour 6 mois, vous donne accès à toutes les nouvelles vidéos de Devox 2009... dès la semaine prochaine ! Oui si vous voulez revoir ce dont je vous ai parlé cette semaine, rendez-vous sur <http://www.parleys.com>. Pour le prix, c'est une très bonne affaire. Tout sera sur <http://devox.parleys.com>. Les vidéos de cette année seront sinon disponibles gratuitement à partir de la fin de l'année 2010. L'accès payant aide à couvrir les frais, et permet de voir en avant-première les tendances de cette année. Les nouveautés de la version 3.0 de Parleys: - les Spaces peuvent contenir plusieurs Channels - les Spaces peuvent être privatisés pour les compagnies - Il est possible de louer une URL et de

customer son aspect. Par exemple <http://sap.parleys.com> - Il y a des Channels gratuits pour les JUG. Allez voir le channel du Paris JUG par exemple, sur lequel nous déposons nos vidéos. Depuis octobre et l'acquisition d'un matériel d'encodage, nous pouvons vous offrir les présentations du JUG rapidement. En conclusion, Stephan nous encourage à tester la nouvelle version sur le site [Beta](#) de Parleys [Cliquez-ici](#) pour la suite de la Keynotes d'ouverture par Oracle et SUN Microsystems PS: Vous pouvez même me voir en photo avec nos beaux teeshirts "Paris JUG Crew" [sur le site du Paris JUG](#)

## Keynotes : Oracle et SUN Microsystems font encore chambre à part

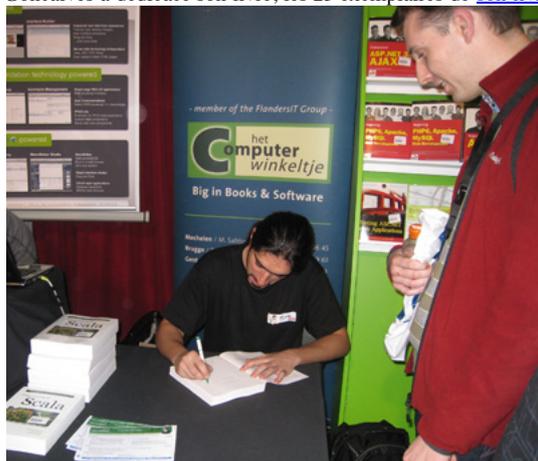
**La Keynote d'Oracle Disclaimer :** *les propos qui vont suivre n'engagent que ma propre responsabilité, et ne représente pas les opinions de mon employeur. En même temps comme je suis indépendant et que je suis mon propre employeur, on s'en fiche pas mal non ? aaah ahaaah ([suite de la partie 1](#))* Oracle monte sur scène ensuite. Steven G.Harris est SVP Oracle Corpo Steven G.Harris est Senior Vice President, Product Development, de la partie "Server Technologies" chez Oracle. Tout d'abord, nous sommes en novembre 2009 au moment où je vous écris ces lignes. L'acquisition de SUN Microsystems n'a pas été finalisé. Difficile pour Oracle dans ces conditions de nous donner une vision, si et seulement si, l'acquisition s'effectue. La présentation a donc été assez légère... Ce que je retiens : plutôt une bonne impression de la démarche d'Oracle. Peut-être que mes gênes de développeurs ont du mal, mais mes gênes d'entrepreneur sont plutôt rassurés. Concernant la stack, l'engagement pour Java reste intact, et sera même encore plus développé. En fait, tant que la plateforme Java fera gagner de l'argent à Oracle, pour quelles raisons Oracle voudrait y toucher ? Nous avons eu ensuite une démonstration de la modularité du moteur de Weblogic Dynamic Module. Ce micro-noyau permet avec de l'OSGi de démarrer des services, de faciliter l'administration. Bon je vais te le dire à toi mais ne le répète pas : OSGi je m'en fiche. Mais vraiment. Moi ce que je veux, c'est un système qui me permet de développer sans relancer mon serveur. Et j'ai été séduit par l'approche de Glassfish par rapport à OSGi. Pour moi, l'approche de SUN est plus intelligente. Je vois OSGi comme la caféine dans le café. C'est là, vous savez que cela vous donnera la patate, mais vous n'en faites pas une montagne. Je signe pour l'approche Glassfish. Et je me fiche de l'approche Glaçon d'Oracle. En fait, il y aura du mouvement dans les nouvelles équipes. Oracle vient de terminer la digestion de BEA. Est-ce que le morceau SUN et toutes ses équipes, ce n'est pas un peu trop gros ? Steven Harris nous donne quelques messages qui me semblent importants : Java en 2009 c'est une technologie, c'est une plate-forme et c'est une communauté. Lorsque vous dites "je fais du Java", vous ne vous rendez pas compte que vous dites aussi "...et je fais aussi partie de cette grosse communauté...". Et ça, croyez-moi, c'est plus important. Oubliez un instant votre éditeur favori. Java c'est votre diplôme. Vraiment. Oubliez votre école avec un nom genre ESIAEEEEP. Dites que vous faites du Java... Ah j'ai oublié de vous dire que la démo d'Oracle était écrite en Scala, avec le framework Lift. Certainement très intéressant... Mais bien éloigné de nos préoccupations de la vie quotidienne. Ce n'est pas parce que Scala rime avec Viagra que tous les vieux développeurs doivent s'y mettre. **La KeyNote de Sun Microsystems** Cette Keynote est animée par Roberto Chinnici et Ludovic Champenois de Sun Microsystems. Roberto est le spécificateur leader de JEE6. Je l'avais rencontré à Jazoon, mais si, [rappelez-vous](#) ! Ludovic est l'un des leaders techniques et architecte sur la plate-forme GlassFish et sur Netbeans. La présentation de 45mn, animée par une démonstration de codage en direct, était vraiment bien. Je vais vous dire un truc : faire du code et passer une démo lors d'une Keynote, c'est gonflé. 2 salles de 780 places remplies à bloc. Si à cette instant la démonstration foire... c'est la catastrophe. Bon, c'est arrivé pendant la démo Oracle lors de l'encodage de l'image. Dommage. La grosse nouvelle : JEE6 Final Release sera là le 10 décembre 2009. Roberto a couvert les nouveautés et les mises à jour de JEE6. Je vous invite à aller voir les blogs des voisins comme SFEIR ou Xebia, qui couvriront en détail les nouveautés de cette version. Il y a de nouvelles APIs, l'introduction du Web Profile, une meilleur "Pluggability" et surtout un moteur d'injection de dépendance. Est-ce que ce n'est pas l'arrivée de l'automne pour les moteurs d'injections de dépendances qui ne seront pas certifiés ? (*automne-printemps*, *printemps-Spring...* *blague... désolé*) Parmi les nouvelles APIs : JAX-RS RESTful web services API permet de mettre en place des services de type REST très facilement. C'est une API qui est déjà très utilisée, basée sur des annotations, avec un modèle de programmation intéressant lorsque c'est nécessaire. Le profil Web JEE6 est avant tout une bonne nouvelle pour simplifier le packaging des applications Webs. C'est aussi la possibilité d'introduire de la modularité, et de la souplesse. Le fichier web.xml devient optionnel. Une très bonne démonstration de Ludovic Champenois montre d'ailleurs que quelques annotations sont suffisantes pour créer une Servlet (@WebServlet, @WebFilter). Et aucuns fichiers XML dans les environs, l'injection de dépendance par un conteneur JEE6, c'est simple et rapide. La démonstration ensuite, qui a même reçu des applaudissements de toute la salle, était décomposée en 3 parties. Tout d'abord la création d'une Servlet, l'ajout d'un EJB3 et enfin l'appel d'un service branché via OSGi. Tout d'abord, Ludovic a montré qu'il n'était pas nécessaire de repackager toute l'application. Glassfish v3 permet de recharger votre Servlet à chaud sans perdre la session. Une Servlet dans le monde de JEE6 c'est très simple. Ensuite, nous avons vu la création d'un EJB3, et l'injection avec le moteur d'injection de dépendance dans la Servlet. Enfin Ludovic a terminé par une démonstration de l'intégration d'un service OSGi. Le tout en 5mn30, sans aucuns plantages. Ce qu'il faut retenir : Java EE 6 est là. Pour une application d'entreprise et même pour migrer un ancien projet, il faut absolument que vous regardiez cette nouvelle architecture. Simple, propre, mature, c'est aussi le résultat d'un effort de toute la communauté Java. La veille au soir j'ai aussi assisté à une Bird of a Father (BOF) à 20h00 avec des gens de SUN, des gens de RedHat, et le tout était très sympa. On peut dire que JEE6 est le résultat d'un effort commun. Enfin Glassfish, mais je l'ai déjà dit, est un très bon conteneur. Il démarre en quelques secondes. Les gens qui ont spécifié JEE6 ont appris à ne pas reproduire les erreurs des vieilles versions de Java (J2EE 1.4 surtout). Il faut comprendre que si cette spécification est arrivée en 2009, c'est qu'elle représente le fruit d'un gros travail de plusieurs entreprises et de la communauté. Je vous recommande de vous jeter sur JEE6 comme vous vous êtes jeté sur Spring il y a 3 ans. Les

choses changent, moi j'y crois.

## Devoxx 2009 : résumé de la première journée de Conférence



3ème jour pour moi ici mais premier jour de la partie "Conférences" de Devoxx. Vous avez été nombreux à lire les articles publiés cette semaine, merci beaucoup. Avant de publier une série d'articles sur le Touilleur Express, voici un résumé de ma journée de mercredi. Tout d'abord [les 3 grandes Keynotes](#) à 09h00 le matin avec Oracle, SUN Microsystems et Adobe. J'ai débuté les conférences avec une présentation de Scott Amblers. Bien connu des Agilistes, Scott a présenté les Mythes de l'Agilité. Je vous expliquerai dans un article en détail, mais en quelques mots : très bien. Grâce à différents sondages, effectués en 2008 et 2009, sur des sites comme Dr Dobb's Journal, Scott a enfoncé des mythes de l'Agilité. Il était temps que quelqu'un nous fasse redescendre sur Terre. J'ai beaucoup aimé. Antonio Goncalvès a dédié son livre, les 25 exemplaires de [son livre sur JEE6](#) sont partis



en 15 minutes.



Après une pause rapide, une photo de groupe avec les parisiens du Paris Java User Group, j'ai repris par une présentation sur la partie serveur d'HTML 5, les Web Sockets. J'ai ensuite suivi 2 présentations autour de Scala : une sur ScalaTest, très bien. Une autre sur Lift très mauvaise. Le contenu était intéressant mais la présentation était trop complexe. Ensuite un moment sympa : Frites et Bières Belges avec Jean-Laurent et Romain. J'ai ensuite rejoins les JUGs pour une session privée avec James Gosling, le père de Java. Nous avons discuté du rachat de Sun Microsystems par Oracle.



Concernant JavaOne 2010, le Moscone Center est réservé. Cependant pour l'instant, personne ne sait si JavaOne aura lieu. James note que le plus ennuyant finalement, c'est que justement, personne n'a encore pris de décisions... Pas top. Concernant les nouveaux langages, Scala est celui qui sort du lot. Cependant James dit que les bons développeurs Java ne deviennent pas forcément de bons développeurs Scala. Il y a même un problème : nous prenons Scala pour l'inférence de type, mais la partie Fonctionnelle est mal maîtrisée. Parfois elle n'est pas utilisée... Bref faire du scala, oui. Mais faire du Scala (avec un S majuscule) c'est dur. La syntaxe et les concepts s'adressent aux bons programmeurs. Et je partage plutôt cet avis pour l'instant. Une photo de Zouheir Cadi du Paris JUG



devant James Gosling, le père de Java: Nous nous sommes ensuite regroupé, une partie des Français est allée voir France-EIRE. Une autre est partie pour diner avec l'ensemble des JUGs du monde entier. Nous étions une trentaine de personnes. Et là, j'ai rencontré le créateur de Juggy, la mascotte du Paris JUG. Bruno est le JUG Leader du Bresil JUG. C'est un gars vraiment excellent. Je lui ai montré mon boulot sur la nouvelle version de Juggy. Et en retour il a proposé de nous faire fabriquer une peluche du Juggy Français !



On verra, mais ça sera sympa. Plus sérieusement, il y a quelques inquiétudes concernant le rapprochement Oracle-Sun. Aaron qui est le leader des JUGs et qui s'occupe des relations entre Sun et les JUGs, nous a aussi demandé quel était notre plus gros souci concernant l'avenir de Java. Pour ma part, j'ai répondu qu'il faut que toi, développeur, tu ne deviennes pas un vieux Coboliste de Java. Pour cela je t'encourage à regarder



d'autres langages comme Groovy ou Scala. Parmi les annonces de la journée : JDK7 est repoussé à la fin de l'année 2010. Par contre, il y aura enfin des closures. A priori ce serait l'implémentation FDM qui a été retenue... Ce débat ne m'intéresse pas. Je n'ai pas besoin de Closure en Java. Je préfère compléter mes connaissances en apprenant Groovy qui a déjà ces Closures. Groovy permet de répondre plus efficacement à certains besoins qu'en Java. Scala peut-être peut aussi compléter votre spectre de connaissance. Et n'oublions pas que ces 3 langages tournent sur la JVM, ce qui est le plus pertinent je pense. Pour cette raison, je ne suis pas intéressé par Ruby par exemple. Concernant JDK7, il y a un gros retard. Relisez [cet article de 2008](#), la majorité de ce qui a été dit hier était encore d'actualité 1 an plus tard. Il faut y voir aussi le signe que Java est devenu un langage mature. Est-il vraiment nécessaire de continuer à l'améliorer, tout en cherchant à garder une rétro-compatibilité au niveau du bytecode ? Aujourd'hui, Keynote, enregistrement des [CastCodeurs](#), et des conférences. La journée va être longue, je vous tiens au courant. Bonne journée (07h15 jeudi matin)

## Devoxx 2009 résumé de la journée de Jeudi



Ce matin réveil tôt, la soirée s'est terminée tard pour certains qui ont un peu fait la fête dans un pub Irlandais. De mon côté, après une soirée en compagnie des Jug leaders, j'ai débuté ce matin par la Keynote d'Ivar Jacobson. Pas mal, sans plus. J'ai ensuite loupé la meilleure conférence de Devoxx cette année par Uncle Bob. Mais c'était pour une excellente raison : participer à l'enregistrement d'un épisode avec



les Casts Codeurs.

Manque Alexis Moussine-Pouchkine pour la photo, **Infinispan**, **Data Grids par JBoss RedHat** Présentation ensuite pendant une heure d'**Infinispan**, par les équipes de RedHat JBoss. [Infinispan](#) est un Data Grid open-source qui permet de faire de la distribution de données en cluster. La présentation était plutôt générale, mais intéressante. Les équipes de JBoss sont parties de JBoss Cache, mais la vision d'**Infinispan** est plus produit là où JBoss Cache est une API d'après Manik Surtani, le présentateur. Les clés de hash permettent d'équilibrer et de répartir les données dans la grille facilement. Amazon EC2 ne donne pas par défaut de solution de Grid. Bien qu'Elastic Block Storage permette de créer une base persistante, en cas de crashes, cela vous demande un travail de restauration pour remonter votre backup. Une grille de données permet de réduire ce risque, avec plusieurs machines. Une bonne présentation qui donne envie de regarder le produit. **Pause déjeuner** Le déjeuner m'a offert un petit moment sympa comme il y en a parfois à Devoxx. J'ai

déjeuné en compagnie de [Roman Ströbol](#) (JetBrains) et [Bruno F. Souza](#) (Brazil JUG Leader), très sympa. Nos discussions autour du rachat de Sun par Oracle, les inquiétudes de la communauté, le futur de JetBrains, très sympa. **The not so dark art of performance tuning** J'ai repris ensuite par le show de [Kirk Pepperdine](#) et Dan Hardiker "The not so dark art of performance tuning". Cette présentation était une petite présentation de quelques outils comme JMeter par exemple. Mais l'essentiel du show était dans la salle. Sympa mais je n'ai pas appris grand chose. **Google App Engine et Gaelyk** Je continue ensuite avec une bonne grosse présentation sur Google App Engine et Gaelyk par Patrick Chanezon de Google, et Guillaume Laforge de Spring Source. Par rapport à la présentation de mardi, Patrick a parlé de la roadmap et du futur de GAE. Sur la roadmap des prochaines versions de Google App Engine : - Support/Hébergement des fichiers larges (à la S3) - Mapping operations across data sets - Curseur sur le Datastore - Alertes de notification lorsqu'une exception survient dans l'application - Datastore dump and restore Guillaume ensuite a fait une démonstration vraiment géniale de [Gaelyk](#). En quelques mots, Gaelyk est un nouveau framework Groovy qui facilite l'écriture des applications avec Google App Engine. Les slides parlent d'eux-même, vraiment intéressant. NOTE: je vais mettre le lien vers les slides de Guillaume sur SlideShare dans la soirée La démonstration en directe : Guillaume a écrit une application qui permet de recevoir par XMPP des demandes de recherche sur Twitter. Il suffit d'envoyer un message avec GTalk à une adresse, pour que celui-ci soit ensuite traité de manière asynchrone par Google App Engine. La page listant les résultats utilise 2 templates, similaire à des Taglibs de Grails ou une Snippet du framework Lift, mais en beaucoup plus simple. Vraiment une très bonne présentation sur la forme comme sur le contenu. **Funky Java, Objective Scala par Dick Wall** Je continue ma découverte de [Scala](#) avec ensuite une présentation de Dick Wall ([JavaPosse](#)) sur des astuces avancées de modélisation et de programmation en Java, ainsi qu'une introduction à Scala. Cette présentation aurait fait plaisir à David Gagot, c'est le type de présentation dont nous avons parlé cet été. Il a présenté les APIs [de Google Collections](#) comme Predicate et Function, avec des exemples de codes assez complet. **Maven 3** Enfin pour terminer la journée, j'ai tenté de regarder la présentation de Maven 3 de Jason van Zyl (fondateur de Maven). Quelques trucs sympas, Jason s'excuse pour Maven 1, nous dit que Maven 2 était un "experiment" et que Maven 3 sera la vraie version qui corrigera un peu tout. Je n'ai pas apprécié tout d'abord les slides illisibles, qui ne prennent pas le template de Devovx. Jugez par vous même ce que j'ai vu:



et aussi



Lorsqu'il y a 500 personnes dans une salle, qui allouent une heure sur une grosse conférence pour venir voir un speaker, la moindre des choses c'est de faire un effort sur les slides. Vous ne pensez pas ? Je suis donc allé écouter Romain Guy sur Android, ce qui était bien plus intéressant. **Conclusion** Voilà, il est 19h00, et [comme hier](#) je vous ai mis en ligne les photos de la journée sur Flickr [à cette adresse](#). Durant l'enregistrement du podcast "[Les CastCodeurs](#)" nous vous avons expliqué pourquoi nous pensons qu'il est intéressant de venir à Devovx. Je sais, pas l'argent, pas le temps, pas envie, le petit fait ses dents, votre belle-mère vient ce week-end... Mais l'an prochain si vous pouvez venir, croyez-moi c'est une bonne expérience. Il me reste des tonnes d'articles à écrire. Je vous publierai dans les semaines qui viennent un article de temps à autre, afin de prendre le temps de bien écrire, et de laisser des contenus de qualité. Et sinon allez bosser un peu, j'ai explosé le nombre de visites hier sur le Touilleur Express :-) Merci les gars A+

## Devovx 2009 : dernière soirée, dernière journée

*Note: You can also read this article [in English](#)* La conférence Devovx 2009 est terminée. Au moment où j'écris ces lignes du Thalys, j'ai pleins de souvenirs et de rencontres en tête. Une grosse semaine, où j'ai un peu levé le nez du blog, pour participer aux événements de Devovx et aux soirées qui suivent. D'où le peu d'articles, mais bon, il faut bien aussi en profiter non ? **Une rencontre** Hier soir, un message sur Twitter donne rendez-vous à 20h30 à tous les Français pour un dernier restaurant. D'ailleurs à part nous, peu de monde restait le soir. Nous voilà avec nos sacs prêt à partir. Arrive un Français dans le Hall. Je lui demande s'il veut se joindre à nous : [Patrick Chanezon](#) de Google, bien connu de la communauté Java. Il a effectué une présentation l'après-midi même de Google App Engine en compagnie de Guillaume, qui a présenté [Gaelyk](#). Nous y reviendrons dans un article plus complet. Patrick a rejoint Google en 2005. Il a tout d'abord travaillé à évangéliser la communauté alors même que Google n'avait que quelques APIs. Il a

évangélisé sur l'API Google AdWords, puis [Google Checkout](#), en dirigeant des équipes importantes chez Google, à San Francisco. Depuis l'été 2009, il est maintenant en charge de l'équipe "Client and Cloud advocacy" (advocacy=plaider) afin de promouvoir l'initiative Openweb standard, Google App Engine, GWT et le petit dernier, [Go Programming language](#). D'ailleurs il recrute en ce moment, si vous voulez le rejoindre en Californie, c'est [par là](#). Une soirée très sympa, où il nous a fait partager son expérience et la vision de Google par rapport à Google App Engine. Parmi les pistes auxquelles réfléchissent les développeurs de Google et la communauté, l'ajout d'un espace de stockage pour les fichiers, adressable par URL, est envisagé (un Amazon S3 like). Avec la puissance de Google, la possibilité d'avoir du Map-Reduce aussi est discutée. Et il y a bien entendu l'intégration avec les autres APIs de Google. Nous apprenons un peu plus sur le fonctionnement de certains produits de Google, sur les technologies utilisées, le tout était passionnant. **4 fameux Java guys have met 25 Frenchies** Lors d'une escale technique pour ravitaillement au Bar pendant la soirée, je passe devant une table où 4 personnes discutent tranquillement autour d'un mini-PC. Tiens mais ce sont les JavaPosses ! et les 4 cette fois-ci. Souvenez-vous l'an passé j'avais déjà eu [une photo avec Dick Wall et Carl Quinn](#)... (cyrille : c'est mon côté People). Bref j'ai discuté avec eux, et ils sont ensuite venus rejoindre la vingtaine de Français, dont l'équipe du Paris JUG. Voyez plutôt le résultat : sacré Français !



De gauche à droite: Dick Wall, Tor Norbye, Joe Nuxell, Nicolas Martignole (Le Touilleur Express), Antonio Goncalves ([Paris JUG](#) and [Les Castcodeurs](#)), Stéphane Epardaud ([Riviera JUG de Nice](#)) et devant avec un tee-shirt vert, Nicolas de Loof ([Breizh JUG](#)), manque Carl Quinn sur la photo.



Tor Norbye, Dick Wall et Patrick Chanezon. **Vendredi matin** Après une nuit bien méritée, nous avons terminé Devovx par 12 conférences divisées en 4 salles. J'ai sélectionné ce matin des sujets plus reposants. - Gaming Java FX : comment coder un jeu avec Java FX, le moteur physique, les collisions - Apache Tapestry 5 par H.Lewis Ship : très bonne présentation du framework Tapestry 5 par son fondateur - Project Lombok : un outil pour Eclipse qui permet d'annoter ses Beans et de retirer le code inutile. Très sympa A priori d'après un sondage, la conférence sur "Pomodoro Technique" par Staffan Nöteberg était bien, ainsi que la conférence "The web on OSGI, here is how" par Don Brown. La première conférence, dont Eric Lefevre a déjà parlé sur [son blog](#), est une technique pour se concentrer sur une tâche. La présentation sur OSGI et le Web était un retour sur expérience d'un gars qui a réellement fait des projets avec OSGI. D'après ceux qui y sont allés, cela remet OSGI dans son contexte. Souvenez-vous que vous pourrez retrouver les conférences dès la semaine prochaine sur le site [Parleys.com](#) pour 49 EUR pour 6 mois. **La meilleure conférence ?** Enfin nous avons terminé par un dernier repas entre Français, et le sujet était de demander à chacun le nom de la conférence qu'il a préféré. Celle qui a marqué les esprits, c'est [la présentation](#) de Robert C.Martin dit "[Uncle Bob](#)". Ce gars est tellement célèbre qu'il a même [sa page sur Wikipedia](#). Ensuite pour moi les présentations qui m'ont le plus marquées, sans ordre particulier - Amazon EC2, 3h de présentation et de retour sur expérience, passionnant - Google App Engine et surtout Gaelyk, la claque tout simplement par G.Laforge - Agile Myths par Scott Ambler. Quelqu'un qui nous ramène sur Terre par rapport à l'Agilité - La Keynote d'ouverture avec la présentation de JEE6 par R.Chinnici et L.Champenois J'ai eu du mal avec les présentations Scala, c'est mon petit regret cette semaine. J'aurai aimé aussi assister à la présentation sur [Clojure](#), au show des JavaPosses, à d'autres présentations, mais je compte me rattraper sur Parleys.com. **Conclusion** Devovx 2009 est la conférence à laquelle les Français doivent aller. Le côté

communauté, les rencontres, les présentations, et le fait que le tout soit à 2h de Paris, 3h de [Tours](#), oui c'est faisable même pour 2 jours seulement. Il ne me reste plus qu'à vous écrire 6 ou 7 articles, mais là tu vois, je suis un poil fatigué, donc je vais prendre mon temps. A bientôt !

## GateIn : découverte d'une Portlet simple (article 2)

Suite de la série d'articles consacrée à GateIn et aux Portlets ([voir article 1](#)). Au menu aujourd'hui, une portlet simple capable d'afficher une liste de messages. L'objectif de cet article est de parler un peu plus du côté Java des Portlets 2.0, avant de revenir à GateIn plus tard. Une Portlet comme vous allez vous en rendre compte, n'a rien de très visuel. Lorsque l'on pense Portlet au départ, on imagine une petite fenêtre dans une page web, avec des boutons maximiser/réduire/configurer. Or c'est le portail qui donne cet habillage, pas la spécification des Portlets en elle-même. Nous allons utiliser encore ici ce qu'il y a de plus simple, à savoir des pages JSP sans outils supplémentaires. Nous verrons dans quelques articles comment écrire une Portlet avec JSF 1.2 et comment utiliser [le bridge JSF 2.0](#) de GateIn. Une portlet c'est avant tout une application web java avec un fichier web.xml. Liferay 5.2 part d'un fichier web.xml, et le décore lors du déploiement dans le conteneur d'application. Cela peut sembler séduisant au premier abord : un fichier web.xml vide complètement anonyme par rapport au portail. Mais il n'y a pas de magie, Liferay se contente de réécrire sa version du fichier web.xml dans le répertoire webapps du conteneur Tomcat. Concernant GateIn, notre fichier web.xml déclare un PortletApplicationListener et un ServletWrapper. Le fichier web.xml d'une portlet sera différent pour chaque portail : Apache Pluto, Jahia, Liferay et donc GateIn. C'est une des limites de la portabilité des portlets. Peut-être que la solution du fichier web.xml vide pour Liferay est la voie à suivre, je ne sais pas.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
  <display-name>texpress-portlet</display-name>

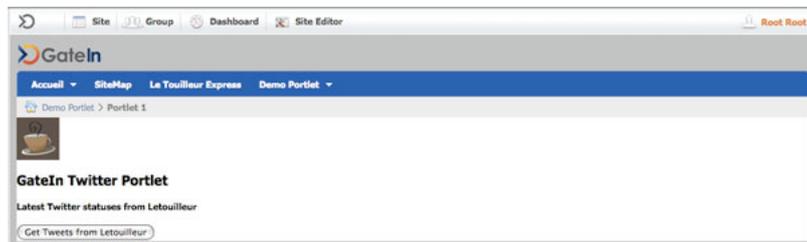
  <listener>
    <listener-class>org.exoplatform.services.portletcontainer.impl.servlet.PortletApplicationListener</listener-class>
  </listener>

  <servlet>
    <servlet-name>PortletWrapper</servlet-name>
    <servlet-class>org.exoplatform.services.portletcontainer.impl.servlet.ServletWrapper</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>PortletWrapper</servlet-name>
    <url-pattern>/PortletWrapper</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>PortletWrapper</servlet-name>
    <url-pattern>/PortletWrapper/*</url-pattern>
  </servlet-mapping>
</web-app>
```

Voyons maintenant ce qui est vraiment du domaine des Portlets, le fichier portlet.xml. Placé dans le répertoire WEB-INF, ce fichier permet de déclarer la liste des Portlets contenues dans l'archive. Pour notre exemple nous allons pour l'instant déclarer qu'une seule portlet Java :

```
<?xml version="1.0" encoding="UTF-8"?>
<portlet-app xmlns="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd">
  <portlet>
    <description xml:lang="EN">Twitter Portlet sample 01, this portlet demonstrates the processAction method</description>
    <portlet-name>TwitterPortlet01</portlet-name>
    <portlet-class>org.letouilleur.portlet.TwitterPortlet01</portlet-class>
    <supports>
      <mime-type>text/html</mime-type>
      <portlet-mode>view</portlet-mode>
    </supports>
    <portlet-info>
      <title>Twitter Portlet 01</title>
      <keywords>demo</keywords>
    </portlet-info>
  </portlet>
</portlet-app>
```

**La portlet TwitterPortlet01** J'aimerais écrire une Portlet capable d'afficher les 10 derniers tweets de mon compte Twitter. A terme je veux pouvoir configurer le nom de la personne que je regarde, et je veux aussi pouvoir informer d'autres Portlets lorsque je change le nom du compte Twitter. Dans un premier temps nous allons simplement compiler et déployer notre Portlet sur GateIn. Si vous ne l'avez pas encore lu, [lisez mon article précédent](#) sur GateIn afin d'apprendre à déployer une Portlet sur GateIn. Je vous ai préparé un petit projet maven avec une portlet toute simple pour cet article. Téléchargez le fichier zip [ici](#) et décompressez-le dans un répertoire. Compilez le projet avec Maven 2.2.1, et Java 6. Et enfin recopiez le fichier WAR vers le répertoire webapps de GateIn. Après avoir ajouté la Portlet dans une nouvelle page, voici ce que vous devez obtenir à l'écran :



Pour

ajouter la portlet, relisez [l'article de la semaine dernière](#). Notre portlet a un bouton Submit qui va récupérer mes derniers messages sur Twitter. Pour l'instant, nous allons simuler cette étape, et simplement afficher une liste envoyée par la Portlet. Ouvrez le code Java de la class TwitterPortlet01 et regardons ensemble la fin du fichier.

```
@Override
public void processAction(ActionRequest request, ActionResponse response) throws PortletException, IOException {
    java.util.List<String> tweets = new ArrayList<String>();
    tweets.add("Premier Tweet");
    tweets.add("Deuxième Tweet");
    tweets.add("Troisième Tweet");

    request.getPortletSession().setAttribute("tweets", tweets);
}
}
```

Là où une Servlet dispose d'une méthode doGet ou doPost, une Portlet s'abstrait du protocole et déclare 2 phases. Une phase de rendu et une phase d'action. La phase de rendu correspond simplement à un affichage de la portlet. La phase d'action est activée lorsque vous envoyez un formulaire par exemple à partir de la partie Web de l'application. Enfin une phase de rendu est toujours lancé après une phase d'action. Dans notre cas, la méthode doView est appelé lors de la phase "Render". Ensuite lorsque vous cliquez sur "Get tweets", la phase "Action" est déclenchée et nous passerons dans la méthode processAction. Celle-ci place dans la session de la portlet une liste de String, et la phase de "Render" est appelée, ce qui provoque l'affichage de la liste dans le fichier normal.jsp. Nous verrons plus loin comment déclarer des attributs de rendus publics, et comment aussi envoyer des événements entre Portlet. Pour l'instant, tout ce que nous faisons est relativement simple. Il est possible d'utiliser aussi des annotations avec les Portlets 2.0. Il est donc possible d'ajouter une annotation @ProcessAction(name="theActionName") afin de dispatcher automatiquement un appel entrant vers une méthode. Dans le cas où une méthode est annotée avec ProcessAction, il faut par contre retirer la méthode générique processAction de GenericPortlet, faute de quoi aucun appel ne sera effectué vers votre méthode annotée. **Du côté de la JSP** Ouvrez maintenant la page "normal.jsp" situé dans le répertoire web-apps. Tout d'abord on notera la présence d'une nouvelle taglib pour les Portlets 2.0, qui permet de déclarer automatiquement un certain nombre d'objets pratiques.

```
<%@ page import="java.util.List" %>
<%@ page pageEncoding="UTF-8" %>
<%@taglib uri="http://java.sun.com/portlet_2_0" prefix="portlet" %>

<portlet:defineObjects/>
<div id="content">
  
  <h2>GateIn Twitter Portlet</h2>
  <h4>Latest Twitter statuses from Letouilleur</h4>
  <%
    List<String> tweets = (List<String>) portletSession.getAttribute("tweets");
    if (tweets != null) {
      out.println("<ul>");
      for (String tweet : tweets) {
        out.print("<li>");
        out.print(tweet);
        out.println("</li>");
      }
      out.println("</ul>");
    }
  %>

  <portlet:actionURL var="myAction" portletMode="view"/>

  <form action="<%=myAction%>" method="post">
    <input type="submit" value="Get Tweets from Letouilleur"/>
  </form>
</div>
```

Le tag <portlet:defineObject> permet de déclarer automatiquement [un certain nombre d'objets](#) pour notre page JSP. Dans notre exemple cela permet de construire l'URL de l'image du logo du blog le Touilleur Express :

```

```

Nous pouvons aussi récupérer la session de la portlet afin de retrouver notre liste de messages

```
List<String> tweets = (List<String>) portletSession.getAttribute("tweets");
...
```

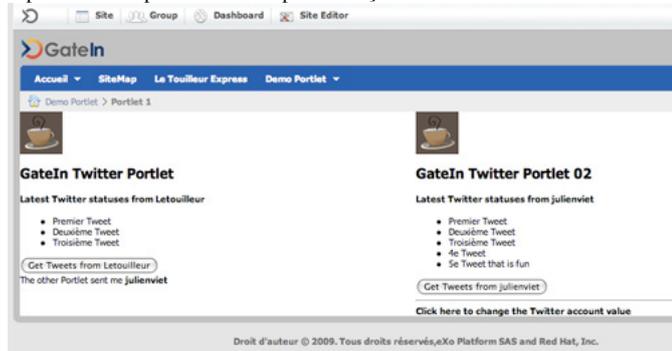
Pour construire l'URL de la FORM il suffit d'utiliser le tag actionURL comme dans cet exemple:

```
<portlet:actionURL var="myAction" portletMode="view" />
<form action="<%=myAction%>" method="post">
```

Enfin voici le résultat lorsque l'on clique sur le bouton Submit :



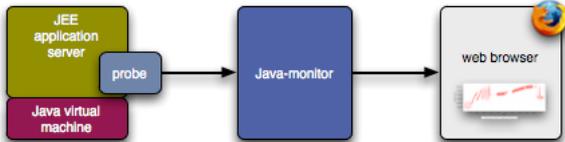
**Conclusion** J'ai présenté un exemple simple dans cet article en vous expliquant le principe des 2 phases des Portlets, ainsi que l'utilisation de la taglib des Portlets. Nous verrons dans un prochain article comment gérer encore plus de fonctionnalités des Portlets avec GateIn. Le prochain article sera disponible la semaine prochaine, voici une capture d'écran pour vous faire patienter: ]



Le prochain article sera publié le 27 novembre. A bientôt Le code source de l'article est disponible [à cette adresse](#).

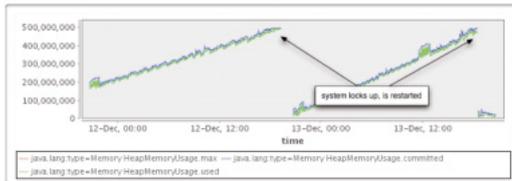
## Devoxx 2009 : Java-Monitor

Mardi dans l'après-midi j'ai assisté à une présentation de 30 minutes sur Java-Monitor par Kees Jan Koster. Java-Monitor est un outil gratuit open-source qui permet de monitorer votre JVM. La particularité de Java-Monitor, c'est que cette surveillance s'effectue directement sur le serveur de Java-Monitor. Une sonde est installée sous la forme d'un agent dans votre JVM et elle se connecte ensuite vers le serveur de Java-Monitor. Les résultats sont donc hébergés sur le site de Java-Monitor. Le tout est gratuit, et fonctionne avec aussi Amazon EC2.



Le fait de surveiller votre

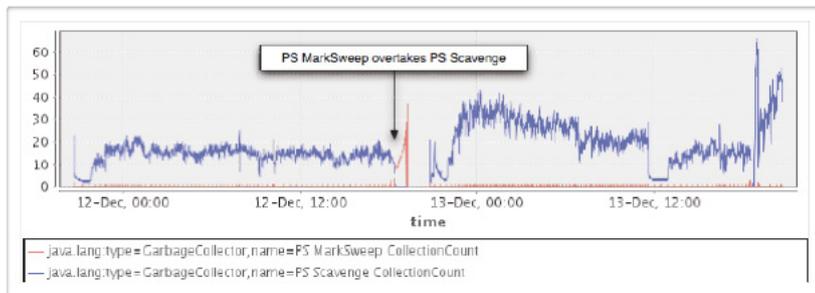
JVM avec ce service permet de partager avec une communauté vos soucis. Kees nous a présenté des cas typiques rencontrés sur des applications en production. Sa présentation très vivante nous a appris quelques informations intéressantes sur la JVM. Kees commence par nous montrer un graphique classique de l'occupation mémoire d'une application. Visiblement celle-ci souffre d'une fuite mémoire. Si vous n'avez jamais eu de fuite (et je vous le souhaite, vous êtes jeune) voici à



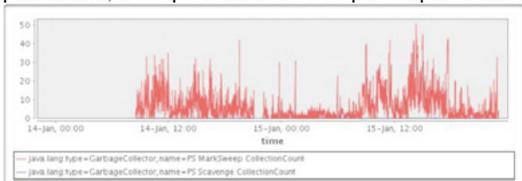
quoi cela ressemble:

Collectors de la JVM s'activent régulièrement comme le montre cette courbe:

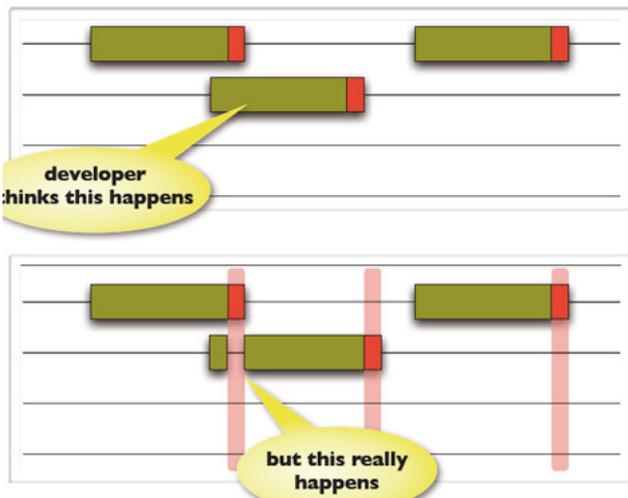
Les 2 Garbages



La solution est de trouver la fuite mémoire avec un outil comme JProfiler. Kees nous a ensuite montré un graphe qui montre que le système souffre d'un souci difficile à pointer au premier regard. Cette courbe montre l'activité du Garbage Collector. Comme vous pouvez le constater, celui-ci tourne en permanence, alors que la mémoire n'est pas remplie. Bizarre non ?



La raison de cette suractivité est un `System.gc()` placé par le développeur dans le code. Or comme il l'explique, c'est une très mauvaise idée. `System.gc()` risque de mettre en marche le Scavenge Collection Garbage Collector. Celui-ci met en pause les Threads le temps d'effectuer la collecte et de ranger les différents espaces mémoires. Il cite l'exemple d'un développeur qui avait ajouté `System.gc()` dans la méthode `destroy()` de sa Servlet, afin de libérer les ressources. Mauvaise idée, regardez l'implication pour les autres sessions dans le conteneur Web:



Si vous souhaitez en savoir plus, lisez les articles sur son site [à cette adresse](#). Pour trouver ces bouts de code, il conseille l'utilisation de [FindBugs](#). Il est aussi possible de désactiver les appels `System.gc()` en spécifiant cette option pour la JVM de Sun :

```
-XX:-DisableExplicitGC
```

FindBugs signale les bugs en analysant le bytecode des JARs. Quelques tests sur le driver JDBC de MySQL 5.1.5 montre qu'il y a ainsi 154 warnings, dont des soucis de `NullPointerException`, et des vulnérabilités à l'injection SQL. Le driver d'Oracle `ojdbc6.jar` de son côté souffre de presque 1028 warnings, avec par exemple des `System.exit()` dans le code du driver, ou des boucles infinies... Super non ? Il montre ensuite un problème des systèmes 32-bits:

```
java.lang.OutOfMemoryError: unable to create new native thread
at java.lang.Thread.start0(Native Method)
at java.lang.Thread.start(Thread.java:597)
at org.apache.tomcat.util.net.JIoEndpoint$Worker.start(JIoEndpoint.java:478)
at org.apache.tomcat.util.net.JIoEndpoint.newWorkerThread(JIoEndpoint.java:697)
at org.apache.tomcat.util.net.JIoEndpoint.createWorkerThread(JIoEndpoint.java:676)
at org.apache.tomcat.util.net.JIoEndpoint.getWorkerThread(JIoEndpoint.java:708)
at org.apache.tomcat.util.net.JIoEndpoint.processSocket(JIoEndpoint.java:743)
at org.apache.tomcat.util.net.JIoEndpoint$Acceptor.run(JIoEndpoint.java:320)
at java.lang.Thread.run(Thread.java:619)
```

Cette erreur est levée lorsque le système ne peut plus allouer de mémoire. Rappelons que l'espace adressable sur les systèmes 32-bits n'est que de 4Gb. En effet il n'y a que  $2^{32}$  adresses, ce qui correspond à presque 4Gb. Sachant que le système lui-même et les autres processus ont besoin de mémoire, il y a donc une limite rapidement atteinte sur les systèmes 32-bits. Il existe beaucoup d'articles sur ce sujet, je vous recommande de lire "[Tuning Garbage Collection with the JVM 5.0](#)". La solution lorsque votre application nécessite réellement autant de mémoire, c'est de passer sur un système 64-bits. Il explique ensuite un concept dont je n'avais jamais entendu parler : l'échec de

promotion d'un objet de la Young Generation vers le Survivor Spaces. Il faut vraiment lire ce [whitepaper](#) pour comprendre, c'est un truc compliqué que je vais tenter de vous expliquer en quelques lignes. Certaines options du GC font que l'espace mémoire se fragmente comme votre disque dur. C'est le cas lorsque vous utilisez le Concurrent Mark-Sweep GC. Imaginez que les promotions des objets de la Young-Generation vers la Old-Generation soit comparable à la recopie d'un disque A vers un disque B. Le CMSweep ne compacte pas le disque B lorsqu'il détruit des objets qui ne sont plus adressables. Le problème est lors de la recopie : Java n'arrive pas à déplacer des objets pour les stocker dans la Old Generation. Cela n'est pas très grave : votre application va exploser avec un OutOfMemory quand bien même il reste de la mémoire... Mais celle-ci est fragmentée. Pour voir ce problème il faut activer certaines options de la JVM:

```
-Xloggc:/var/log/gc.log -verbose:gc
-XX:+PrintGCDetails -XX:+PrintGCTimeStamps
```

Il faut ensuite regarder les logs du serveur, vous verrez que des messages "promotion failed" apparaissent alors

```
42536.572: [GC 42536.572: [ParNew: 353920K->2080K(353920K), 0.1246980 secs] 1450967K->1137843K(2520704K), 0.1249020 secs]
42546.733: [GC 42546.733: [ParNew: 316704K->3604K(353920K), 0.0599980 secs] 1452467K->1139368K(2520704K), 0.0602080 secs]
42555.527: [GC 42555.527: [ParNew: 318228K->5034K(353920K), 0.0609210 secs] 1453992K->1140797K(2520704K), 0.0611270 secs]
42564.154: [GC 42564.154: [ParNew: 319658K->5214K(353920K), 0.0780650 secs] 1455421K->1142411K(2520704K), 0.0782850 secs]
42571.444: [GC 42571.444: [ParNew: 319838K->5514K(353920K), 0.0649090 secs] 1457035K->1144161K(2520704K), 0.0651220 secs]
42576.714: [GC 42576.714: [ParNew (promotion failed): 320138K->320138K(353920K), 0.2365970 secs]42576.951: [CMS: 1139969K->1120688K(2166784K), 9.2214860 secs] 1458785K->1120688K(2520704K), 9.4584090 secs]
42587.017: [GC 42587.017: [ParNew: 314624K->34454K(353920K), 0.1340880 secs] 1435312K->1205264K(2520704K), 0.1344800 secs]
42588.700: [GC 42588.701: [ParNew: 349078K->23091K(353920K),
```

La solution est de régler la taille des différents espaces de la JVM ([voir ce document](#)). Il continue sa présentation en nous parlant des problèmes de File Descriptor sous Unix (Too Many Files open). Il suffit de changer la limite du nombre des fichiers pour l'utilisateur qui exécute votre serveur. A noter que les sockets sont comptés comme des fichiers sous Unix (je sais que ce que je dis n'est pas tout à fait exact, je fais court...) ce qui concerne donc les connexions vers la base, ou la partie web de votre serveur. Il rappelle aussi la grosse différence entre le mode serveur (-server) ou client pour la JVM de Sun avec quelques courbes. En conclusion, une petite présentation plutôt sympa de 30 minutes, qui me fait dire que si vous êtes en galère avec la JVM, il serait bon de passer un peu de temps sur les forums de Java-Monitor. **Références** [Les slides de la présentation Java-Monitor.com FindBugs](#), excellent outil pour analyser son bytecode.

## Devoxx 2009 : Agile Mythbusters par Scott Ambler

Première présentation ce mercredi après la Keynote d'ouverture par Scott Ambler sur les mythes de l'Agilité. J'ai adoré. Que ce soit sur la forme, comme sur le fond, une bonne présentation avec cependant parfois des chiffres interprétés qui m'ont dérangé. Mais le fond de la présentation est bon : oui il y a un décalage entre ce que les Agilistes disent faire et ce qu'ils font réellement. Et c'est le plus intéressant. Scott Ambler est bien connu de la communauté Agile. Il est officiellement "Practice Leader Agile Development" chez IBM. Auteur de [nombreux livres](#) et [d'articles](#), c'est aussi un bon speaker. Le thème de la présentation : il y a beaucoup de mythes sur le développement Agile dans le monde de l'Entreprise. Certains de ces mythes sont aujourd'hui utilisés par les promoteurs de l'Agilité, il est donc temps de regarder l'écart entre ce que l'on nous dit et la réalité du terrain. Pour appuyer sa démonstration et les nombreux chiffres qu'il cite, Scott a réalisé plusieurs sondages en 2008 et en 2009 sur son site personnel et surtout sur le site [Dr Dobb's Journal](#). La présentation ressemble à un mélange entre "Une Famille en Or" et l'émission "MythBusters" de Discovery Channel. Les gars de MythBusters sont assez marrants et parlent de sujets improbables comme "Est-ce qu'un implant mammaire peut exploser dans un avion ?" ou "Peut-on récupérer un disque dur crashé en le mettant au congélateur ?". Des trucs super important comme vous pouvez le constater. Mais chacune de ces émissions se terminent par un "Busted" ou "Confirmed", et c'est ce que Scott va utiliser. L'ensemble des sondages et des résultats est disponible sur la page <http://www.ambyssoft.com/surveys/>. J'ai trouvé aussi [les slides de la présentation](#) sur le site AgileTour 2009 Montreal si vous souhaitez lire par vous même. Les affirmations explorées:

- o The majority of organizations are now doing agile
- o The majority of project teams within those organizations are now doing agile
- o Agile is just for small, co-located teams building straightforward systems
- o The agile community is focused on development
- o Becoming a "certified master" is a good idea
- o Agile teams don't do up-front modeling
- o Agile teams test often and test early
- o Agile teams do not produce documentation
- o It's difficult to bring new people onto agile teams
- o Agile works better than traditional approaches

Je ne vais vous parler que de quelques éléments, les plus polémiques bien entendus. Tout d'abord l'affirmation selon laquelle la majorité des entreprises font de l'Agile ? 76% des personnes consultées sur "[State of the IT Union Survey de Juillet 2009](#)" ont répondu que l'Agilité était entrée dans leur entreprise. Ont donc répondu oui, les personnes qui font de l'Agilité mais aussi les

personnes qui connaissent une équipe qui fait de l'Agile... dans leur entreprise. Nous pouvons donc affirmer que OUI l'Agilité est entrée dans l'Entreprise, ce n'est pas un mythe. Le 6e slide de sa présentation par contre, montre que seulement 44% des gens qui affirment avoir de l'Agilité dans leur entreprise... en font eux-mêmes. La majorité des gens ne font donc pas de l'Agile pour l'instant. Un mythe aussi qui m'a plutôt étonné : l'Agilité ne marche qu'avec des petites équipes de 10 personnes ou moins. Et bien c'est faux d'après les personnes consultées. Mais je trouve que les graphiques étaient mal tournés. La majorité des équipes qui font de l'Agilité et qui réussissent ne dépasse pas 20 personnes. Et il l'a aussi expliqué ensuite. A votre avis, est-il possible de faire de l'Agilité lorsque le projet est dans un domaine réglementé ? Je pense aux marchés publics par exemple, aux marchés sur le domaine de la santé aux USA. Et bien contrairement à ce que l'on pense, 60% des personnes qui font de l'Agilité, le font sur un projet réglementé justement. Intéressant non ? CMMI et Agile ne vont pas ensemble : c'est faux d'après 78% des personnes impliquées. CMMI s'applique à la gouvernance de l'entreprise, là où l'Agilité répond à un besoin de communication et d'organisation projet. Peut-être que vous pensez que la majorité des projets Agiles ne marchent que sur de nouveaux projets... Vous vous dites : nous avons notre vieux projet de 4 ans, nous ne pouvons pas faire de l'Agile. A titre personnel je peux vous dire que c'est faux. C'est ce que je viens de faire pour une Banque Française. Le projet existe depuis 4 ans et nous l'avons basculé en Scrum sans soucis. Les gens ont répondu qu'à 78% leur projet contient du vieux code, qu'à 57% il doit s'intégrer avec des systèmes anciens et 45% travaillent avec des données historiques. Bref l'Agilité marche aussi sur les projets plus anciens. Mon préféré : Devenir un "Certified Master" en prenant un cours de 2 jours est une bonne idée. A votre avis qu'ont répondu les personnes interrogées ? **78% pensent que la certification de 2 jours n'a pas de sens**, seulement 10% respectent la certification et 12% s'en fiche complètement. Scott Ambler fait rire la salle en racontant cette histoire : un de ses amis lui a fait suivre une offre d'emploi pour un poste de chef de projet dans le secteur médical. Sur l'offre, il y avait écrit "PhD Physics, Certified Scrum Master". Comme si une certification peut avoir autant d'importance qu'un doctorat en médecine... on marche sur la tête. Son avis est qu'une certification se mérite et doit se gagner. Il est pour les certifications avancées de Scrum comme Certified Scrum Practitioner car celles-ci ont un sens. Un sujet bien polémique. Est-ce que la majorité des équipes Agiles sont co-localisées ? Scott définit cela par le fait que l'équipe au complet se trouve dans un seul et même open-space. Et bien en fait, 29% des équipes font de l'Agilité à distance, 13% avec temps de transport entre les équipiers, 17% dans le même bâtiment, et finalement 42% (la minorité donc) sont réellement co-localisés dans un même open-space. Ce mythe est donc Busted. Est-ce qu'une équipe Agile commence à coder dès le début d'un projet ? Et bien non. En fait le temps de démarrage moyen d'après le sondage est de 4 semaines environ. Il n'y a donc pas de miracles, mais 20% se mettent à coder dans la première semaine, ce qui est important je pense. La majorité des gens répondent 2 semaines. Un slide par contre montre que les équipes Agiles testent souvent et tôt. 71% des développeurs affirment travailler en TDD. Il explique aussi que cette pratique n'a pas besoin d'être systématique, et il nous dit de faire attention à ne pas être trop dogmatique. Une question intéressante vient en écho d'un article que je vous ai écrits [il y a quelques semaines](#). Le mythe : les équipes Agiles ne font pas de captures des demandes et de la modélisation dès le départ du projet. Elles commencent à coder sans savoir ce qu'il faut faire. A priori d'après les personnes qui répondent, ce n'est pas vrai. 52% affirment écrire dans un Document Word, 39% affirment utiliser un outil comme Jira pour lister les demandes, 26% utilisent un Wiki, bref la majorité des développeurs finalement capturent les demandes et les besoins avant de commencer à développer. J'imagine qu'un document Word doit s'apparenter à une spécification technique, et que pour ce qui est des Jiras, on doit être plus proche d'un Product Backlog. Une affirmation ensuite: un projet piloté en méthode Agile réussit mieux qu'un projet en méthode classique ou sans aucunes méthodes. Oui c'est vrai d'après les personnes ayant répondu au sondage. Les équipes Agiles produisent un travail de meilleure qualité, livrent plus de fonctionnalités, et respectent plus le budget qu'un projet classique (slide 25) Scott explique qu'il constate un décalage entre ce que les gens pensent faire et ce qu'ils font réellement. Ses sondages sont construits pour collecter justement ce type de différence, ce qui en fait un bon outil je pense. Je ne peux pas vous redonner ici l'ensemble de ses arguments, je vous encourage à regarder et à étudier les sondages sur son site. L'ensemble des chiffres est là, avec les questions. Cela vous permettra de vous faire votre propre idée. La communauté Agile doit encore faire un effort pour être plus critique, pour parler de ce qu'elle fait réellement. Cela rejoint aussi ce que nous avons entendu le jeudi matin : nous sommes dans une industrie de la mode. Ivar Jacobson explique que notre industrie est très friande de mode. En ce moment c'est Scrum, mais ce sera peut-être Lean demain, et TrucMachinChose après-demain... Il conseille d'adapter et de vous construire votre méthode de travail. Les principes de l'itératif et de l'incrémental, l'idée de livrer un logiciel qui marche, sont des bons principes. [Sa présentation au format PDF](#) [Le site et les différents sondages](#)

## Jahia: découverte d'un moteur d'intégration de contenu web



Dans la suite de mes aventures sur la découverte des portails, j'ai rencontré [Stéphane Monier](#) de la société [Jahia Solutions Group](#). Je vous propose une petite découverte du produit et de l'entreprise. Nous allons parler technique et aussi produit, je pense qu'il est intéressant de découvrir Jahia et les idées de ce produit. **Le produit en quelques mots** Jahia est un logiciel d'intégration de contenu web (Web Content Integration Software) basé sur un portail. Ecrit en Java, open-source (GPLv2) et basé sur des projets connus de la communauté, c'est un produit orienté publication de contenu sur le web et création de site d'entreprise. La force de Jahia est de proposer une solution qui permet de créer son site rapidement, à partir de templates. Complètement écrit en Java, c'est aussi un portail capable de faire

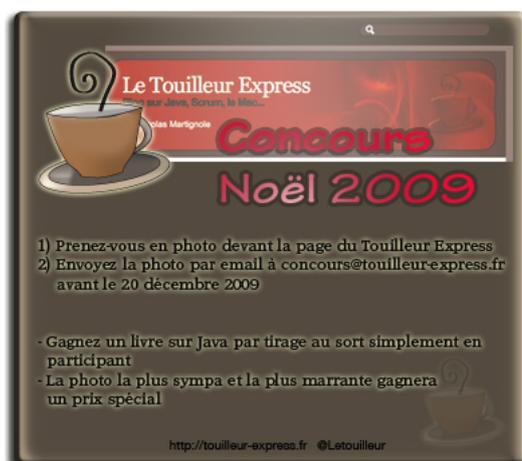
fonctionner des Portlets. Plutôt qu'un long discours, le mieux c'est que vous preniez quelques minutes pour le tester [sur le site de démonstration de Jahia](#). Avec quelques clics de souris, des glisser-déposer, zéro code à saisir, écrire un site sympa n'est plus de la Science-Fiction. J'ai testé et j'ai bien aimé. Avec un gestionnaire de contenu et un outil de création de site qui m'a paru simple, j'ai aussi bien aimé l'ergonomie globale du produit. La prise en main et la création de son premier site est rapide. Pour aller plus loin dans la mise en page, il faut ensuite apprendre le système de templates de Jahia. La documentation en ligne sur [Jahiapedia](#) est plutôt bien écrite je trouve. C'est important pour le développeur qui débute avec Jahia v6. **L'entreprise** Jahia Solutions Group est un éditeur Suisse. Avec une trentaine de personnes, il est représenté en Europe mais aussi en Amérique du Nord. Stéphane explique qu'un bureau au Canada s'est ouvert récemment. Piloté par Elie Auvray, on retrouve aussi des personnes comme Serge Huber, le CTO, qui avec quelques autres personnes travaillent pour proposer un logiciel sympa. En France Jahia travaille avec les intégrateurs comme Smile par exemple. Plusieurs références comme Michael Page, des références dans le monde de l'assurance ou de la Finance. La page "[Customers](#)" donne une idée des clients de Jahia, et ils sont plutôt nombreux. L'Ecole Polytechnique Fédérale de Lausanne (EPFL) est l'un des partenaires historiques de Jahia, avec un grand nombre d'installations de Jahia. Stéphane me confie aussi que les équipes de développement travaillent avec Scrum, en sprint de 3 semaines. **Le socle technique** Jahia a modernisé le socle technique de son moteur, avec la sortie de la version 6 récemment. Une migration vers le JCR [Apache Jackrabbit](#), l'utilisation d'[Apache Pluto 2.0](#) pour le moteur de Portlet 1.0 et 2.0, l'utilisation de GWT et de [GXT](#) pour l'interface d'administration, bref Jahia a fait un ravalement de façade complet cette année. La nouvelle version est plutôt aboutie, on imagine les mois de travail derrière pour mettre en place les briques techniques. **Amazon EC2** Jahia propose [une offre prête à l'emploi dans le Cloud](#), sur Amazon EC2. Pour un prix connu à l'avance, l'idée est de proposer un package avec une installation de Jahia configurée. Personnellement je crois à cette idée. J'ai en tête des clients potentiels qui n'ont pas l'infrastructure, et qui seraient prêts à sauter le pas. Mais est-ce que les coûts ne sont pas trop élevés pour l'instant ? **Modèle de développement** Nous discutons un moment sur le modèle de licence. Jahia propose une version communautaire, avec une licence GPL, et une version entreprise, avec plus de fonctionnalités, du support, des binaires certifiés, du [SLA](#), bref ce que proposent aussi les voisins que sont Liferay et eXo Platform. Ce modèle ouvert permet cependant de télécharger et de commencer à tester Jahia. Et prenez le temps de tester par vous-même, vous verrez que le logiciel est sympa, plutôt facile à mettre en marche et à configurer. Le modèle open-source de Jahia est intéressant. Appelé "Contribute or Pay", c'est la balance entre les contributeurs open-sources, qui peuvent devenir des committers Jahia, et les contributeurs payants, représentés par les clients. Clairement : soit tu contribues en tant que développeur, et donc ton travail rejoint le noyau de Jahia, soit tu sponsorises en tant que client le développement de nouvelles fonctionnalités, et celles-ci sont reversées dans le noyau de Jahia, pour faire avancer le logiciel. Les contributions financées par les entreprises seront donc aussi données à la communauté open-source, cela fonctionne dans les 2 sens. J'aime bien cette idée, où le client final qui paye pour avoir une fonctionnalité, accepte aussi que sa contribution soit mise dans la version libre et ouverte. C'est une forme de mécénat. Pour l'entreprise, c'est aussi la garantie qu'une fonction importante sera présente dans Jahia. Stéphane cite l'exemple d'un moteur de réécriture des URLs qui a été sponsorisé par un client. C'est plutôt simple et clair, tout est expliqué [sur cette page](#). **Les forces et les faiblesses de Jahia** La force de Jahia c'est tout d'abord son usabilité, le caractère prêt à l'emploi et la facilité pour construire son site. Chez un prospect chez qui j'ai travaillé en novembre (JL si tu lis ces lignes...) nous avons réalisé une maquette en une matinée. La construction du site et la gestion documentaire sont vraiment bien codés. Par rapport au voisin Liferay, Jahia offre moins de fonctionnalités "out-of-the-box" pour l'instant dans la version 6. Mais le développement d'outils communautaires est en cours, et viendra compléter la version 6.5 prévue l'année prochaine. La roadmap est aussi disponible sur le site de Jahia. Un des points intéressants que je n'ai pas testé, mais dont Stéphane a parlé, c'est ce qu'ils appellent le "Jahia United Content Hub". Ce moteur d'intégration permet d'interfacer son portail avec son SI, avec des connecteurs vers Alfresco, EMC Documentum, Microsoft Sharepoint, Lotus Notes, Oracle Stellent par exemple. Cette partie est en fait un travail en commun avec une autre entreprise Française, EntropySoft, dont d'ailleurs le site Web tourne avec Jahia ([allez voir ici](#)). C'est une passerelle vers le monde de l'entreprise, la GED et tous les documents qui doivent dormir dans les logiciels chez vous. Jahia est avant tout un moteur d'intégration de contenu pour le web, avec de la gestion documentaire et un moteur de validation. Liferay 5.2 est plus configurable, eXo ECM 1.0 est plus puissant techniquement. Je pense à Liferay et le mode de construction des pages, qui permet d'abord de mettre des Portlets en général, puis ensuite de mettre quelques portlets d'affichage du contenu Web. Mais il n'y a pas l'idée de templates de Jahia, qui permet elle de construire son site avec un cadre défini plus ou moins à l'avance. eXo ECM est plus proche de Jahia, avec aussi ce mode "édition en direct" dans eXo WCM comme Jahia, qui permet d'éditer sa page. Chez la version actuelle d'eXo ECM, il y a un effort plus important à fournir en terme d'intégration. On peut aussi dire que c'est un avantage, et que finalement vous n'utiliserez que ce dont vous avez besoin. En fait, eXo ECM est divisé en 3 produits : eXo WCM pour la gestion du contenu web, eXo Workflow pour la validation des modifications et eXo DMS pour la partie GED. **Le mot de la fin** En terme d'ergonomie, je préfère Jahia aux autres, si votre projet est vraiment avant tout orienté Internet. Je pense que pour de l'Intranet, dans la version v6 il manque quelques outils collaboratifs comme un Wiki ou des Forums. Par rapport à l'entreprise, Jahia est très cousin avec eXo Platform. Il y a des gènes que j'ai retrouvés chez les deux entreprises. Si j'étais un intégrateur du monde Java comme Xebia, Ippon Technologies ou Zenika, je regarderais Jahia. C'est du Java, de l'open-source, et c'est une boîte intéressante. **A propos de mon indépendance** Un mot pour vous dire aussi un point important concernant mon indépendance : je parle de GateIn sur le Touilleur Express car j'ai tout simplement fait une formation gratuitement avec eXo Platform en octobre à ma demande. En retour j'ai proposé d'écrire quelques articles. Bref un accord gagnant-gagnant, où j'ai appris un peu plus de détails sur le moteur d'eXo et où en retour, je vous en parle. C'est important peut-être de le dire, chose que je n'ai pas fait. Sinon je reçois aussi des livres de la part des éditeurs, on m'a invité à 2 conférences cette année, et les gens veulent absolument me donner des tee-shirts, ce qui énerve mon épouse car l'armoire est pleine. Vous voyez, je ne suis pas difficile à acheter :-). Blague à part, vous savez faire la part des choses, et un blog c'est forcément subjectif. Sinon il faut lire zero-un

nainformatik où d'ailleurs ils interviewent [des gars connus](#)... Bon allez, vous allez tester Jahia et on en reparle autour d'une bière au Paris JUG.

## Indépendance

J'ai envie de revenir sur plusieurs points pour vous informer un peu plus sur le blog le "Touilleur Express". Vous avez noté les articles sur GateIn ces dernières semaines. Et vous avez entendu mon enthousiasme sur JEE6 sur le podcast des "CastCodeurs". Moi je serai vous, je m'inquiéterai sur l'indépendance du "Touilleur Express" par rapport à notre industrie. Est-ce que Nicolas n'est pas entrain de se faire embarquer avec sa petite barque ? Concernant eXo Platform, j'ai appelé Benjamin Mestrallet il y a quelques semaines en lui proposant d'écrire une série d'articles sur GateIn. En retour j'ai demandé à participer à la formation eXo Portal avec Julien Viet. Et il a accepté. J'ai réellement envie de vous en parler, et de vous amener à comprendre qu'il y a de nouveaux concepts qui doivent vous intéresser pour 2010. Pour bien écrire sur le sujet et par curiosité, j'avais besoin de cette formation. Ensuite j'ai débuté par des articles sur les Portlets, plutôt bas niveau, car il faut expliquer les bases. Je pense écrire encore quelques articles, mais les équipes d'eXo Platform sont vraiment cools, elles ne mettent pas de pression. En fait je vais vous le dire : ce sont des amis dont j'ai envie de parler, ni plus, ni moins. Et eux ont des sujets bien plus importants en ce moment, croyez-moi. A propos de JEE6, mon enthousiasme par rapport à Devvxx sur les CastCodeurs doit certainement faire plaisir en effet aux gens de Sun. Mais encore une fois, c'était mon ressenti, ce que j'ai observé pendant 5 jours. Lorsque vous voyez une salle de 800 places remplies, que les gens se jettent sur le livre d'Antonio, que les séances de questions/réponses sont animées, et que la présentation de GlassFish est applaudie, ce n'est pas moi qui suis enthousiaste. C'était les gens présents à Devvxx. Le blog "Le Touilleur Express" mine de rien commence à me faire connaître un peu. Les éditeurs m'envoient leurs livres. J'ai reçu la nouvelle édition de "[Spring par la Pratique](#)", "[Android](#)" par Florent Garin ou encore "[GWT](#)" par Olivier Gérardin. Je compte lire ces livres mais comme je dis aux auteurs, ne vous attendez pas à un article dès la semaine prochaine. Je fais au mieux. Et une fois les livres lus, je compte les offrir au Paris JUG lors des séances Questions/Réponses pour les domaines sur lesquels je ne travaille pas par exemple. J'ai eu aussi des emails afin de me présenter des superbes produits pour révolutionner le développement... auxquels j'ai répondu que cela ne m'intéressait pas. J'ai parlé de produits qui m'intéressent comme [Kalistick](#) ou [SpringFuse](#). Le monde de l'open-source est déjà largement traité dans les autres blogs et je trouve intéressant de parler aussi un peu des outils qui peuvent nous faire gagner du temps. Qu'ils soient commerciaux ou en open-source. Enfin concernant les conférences, j'ai été invité à l'USI et à Jazoon en 2009. De mon côté cela me donne l'occasion de participer à ces événements et donc d'écrire sur mon blog. Du côté des organisateurs, cela fait partie du plan de communication pour faire connaître l'événement. Chacun est gagnant, même vous. Grâce à cela, vous avez des articles sur Jazoon et sur l'USI pour vous occuper cet hiver. Voilà, juste pour vous dire que je suis content du boulot réalisé pour le blog, et que j'arrive à maintenir un équilibre entre mon indépendance et les acteurs de notre communauté. Un blog c'est personnel donc forcément un peu subjectif. Voilà vous pouvez reprendre une activité normale (coder avec IDEA IntelliJ, aller chercher un café, éteindre le sèche-linge, commiter du code, prendre votre avion...)

## Concours spécial Noël 2009



(Update jeudi 26 : 2 photos pour vous inspirer en fin d'article) (Update 2 samedi : David à l'Apple Store à Paris) (Update 3 : ajout du premier prix pour le gagnant) Tiens ça serait sympa d'envoyer des photos de vous entrain de lire le blog "[Le Touilleur Express](#)" dans des endroits improbables, genre devant chez Microsoft ou chez Google... Pour cela, envoyez-moi une photo par email ([concours@touilleur-express.fr](mailto:concours@touilleur-express.fr)) de vous devant la page d'accueil du Touilleur Express, dans les endroits que vous voulez. Précisez votre nom, votre adresse postale et un petit mot sur l'endroit où vous avez pris la photo. Je ferai un tirage au sort afin de faire gagner un livre sur Java de votre choix sélectionné sur Amazon.fr parmi l'ensemble des participants. Il faut simplement que l'on voit la page d'accueil du Touilleur Express sur un écran, que ce soit votre PC, votre iPhone, votre mini-PC. Essayez de trouver un endroit original tant qu'à faire, évitez au bureau, osez les endroits plus sympas. Si vous prenez la photo à côté d'une star de la communauté Java, je vous offre un petit bonus. Faites du bruit sur Twitter, et envoyez-moi vos photos pour que l'on boucle cela avant le 20 décembre. La photo la plus sympa et

la plus marrante gagnera **un prix d'une valeur de 100 EUR**. Je verrai si je sélectionne la photo ou si je fais participer quelques lecteurs fidèles du Blog. Allez, hop, vous demandez à votre voisin de vous prendre en photo avec la page du Touilleur Express affichée sur votre iPhone et vous l'envoyez ! **Jeudi 26 novembre**: - Jonathan un lecteur assidu:



- Moi au restaurant Japonais, photo prise par David Chau de SFEIR [caption id="attachment\_2493" align="alignnone" width="225" caption="Nicolas



Martignole"] [caption] - Romaintaz et une photo avec une centaine de touillette à café [caption id="attachment\_2496" align="alignnone" width="300" caption="Photo par @romaintaz, des centaines de Touillettes à café"]



[/caption] - David Chau à l'Apple Store à Paris



@davidchau à l'AppleStore  
Concours <http://touilleur-express.fr>  
- Gilles Debunne m'a envoyé cette photo prise à Mountain View devant le building 43 en Californie, chez Google :



## GateIn: gestion des Préférences et communication entre Portlets (article 3)

Suite de notre série consacrée à GateIn et aux Portails. Aujourd'hui au menu, dernier article de la partie purement Portlet. Nous allons voir comment gérer les préférences d'une Portlet, puis comment envoyer un message pour que d'autres Portlets soient notifiées. **Installation** Comme dans l'article précédent, je vous ai préparé un projet maven avec le code et l'ensemble compilé si vous voulez tester rapidement. Téléchargez le fichier [portlet\\_article03.zip](#) et décompressez-le dans un répertoire différent. Vous pouvez ensuite soit lancer le projet IDEA IntelliJ, soit le project Eclipse (mvn eclipse:eclipse). Le fichier war doit être recopié dans le répertoire webapps de votre installation de GateIn. Vous devez avoir lu [le premier article](#) pour savoir comment ajouter une Portlet à GateIn la première fois. Pour le reste, c'est assez simple. **Les Préférences des Portlets** Voici ce que nous allons faire: nous allons déclarer une propriété de configuration afin que le nom du compte Twitter soit configurable. Lorsque la Portlet sera en mode EDIT, vous pourrez configurer le nom du compte Twitter à afficher. Enfin le mode configuration de GateIn vous montrera aussi votre propriété. La spécification des Portlets vous donne un objet [PortletPreferences](#) afin de stocker les paramètres de configuration de la Portlet. Par exemple l'adresse d'un serveur REST distant pour récupérer une liste de médicament (au hasard). Les propriétés d'une Portlet sont spécifiques à chaque utilisateur. Par ailleurs, elles sont sauvegardées lorsque vous arrêtez le serveur. Nous verrons cela tout à l'heure. **Comment utiliser une propriété d'une Portlet** Nous allons tout d'abord regarder ce que j'ai déclaré dans le fichier portlet.xml. La balise portlet-preferences permet de déclarer des couples clés/valeurs, comme un fichier de propriétés. J'ajoute ici une nouvelle propriété "defaultScreenName"

```
<portlet>
  <description>Twitter Portlet sample 02, this portlet demonstrates how to set and display
    a public portlet render parameter.
  </description>
  <portlet-name>TwitterPortlet02</portlet-name>
  <portlet-class>org.letouilleur.portlet.TwitterPortlet02</portlet-class>
  <supports>
    <mime-type>text/html</mime-type>
    <portlet-mode>view</portlet-mode>
    <portlet-mode>edit</portlet-mode>
  </supports>
  <portlet-info>
    <title>Twitter Portlet 02</title>
    <keywords>demo</keywords>
  </portlet-info>
  <portlet-preferences>
    <preference>
      <name>defaultScreenName</name>
      <value>Letouilleur</value>
    </preference>
  </portlet-preferences>
</portlet>
```

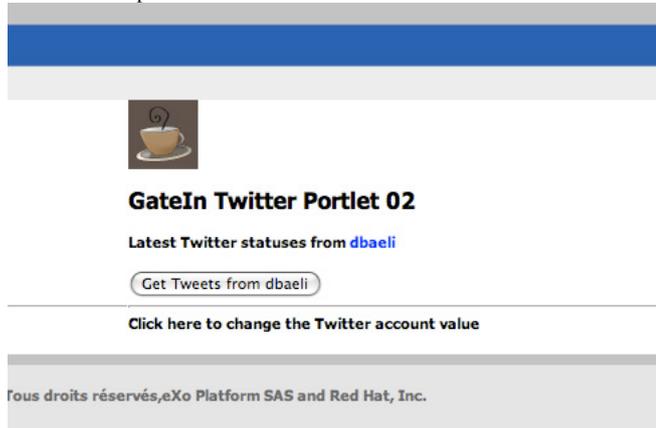
```
</portlet>
```

Cette propriété sera configurée à "Letouilleur" pour les utilisateurs n'ayant pas encore personnalisé la valeur. Regardez maintenant le fichier viewPortlet02.jsp. Cette page est la page du mode VIEW de la portlet TwitterPortlet02. Elle affichera le nom du compte. Voici comment je récupère les préférences afin d'afficher le nom du compte Twitter:

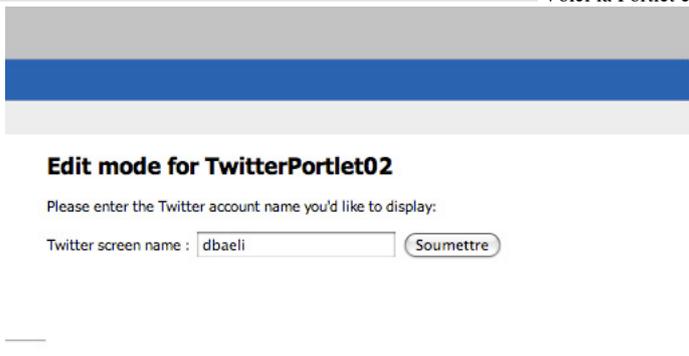
```
<%
    String tweeterAccount=renderRequest.getPreferences().getValue("defaultScreenName", "Anonymous");
%>

<h4>Latest Twitter statuses from <font color="blue"><%=tweeterAccount%></font></h4>
```

**Comment modifier la propriété ?** La portlet TwitterPortlet02 est déclarée avec 2 états dans le fichier portlet.xml : EDIT et VIEW. Lorsqu'elle bascule en mode EDIT, il est possible de changer le nom du compte Twitter. Voici la vue de la Portlet en mode VIEW :



Voici la Portlet en



mode EDIT : Que  
se passe-t-il lorsque l'on clique sur le bouton "Soumettre" ? Comme vu dans l'article précédent, cela déclenche la phase Action de la Portlet. Dans la class TwitterPortlet02, regardez le code de la méthode processAction:

```
@Override
public void processAction(ActionRequest request, ActionResponse response)
    throws PortletException, IOException {

    if (request.getPortletMode().equals(PortletMode.EDIT)) {
        sendActionEvent(request, response);
    }
    // ....
}
//....

private void sendActionEvent(ActionRequest request, ActionResponse response)
    throws PortletModeException, ReadOnlyException,
        IOException, ValidatorException {

    // Recupere le parametre envoye par le formulaire HTML
    String screenName = request.getParameter("screenName");

    // Change la valeur de la preference
    PortletPreferences prefs=request.getPreferences();
    prefs.setValue("defaultScreenName",screenName);
    prefs.store();

    // ...

    // back to view mode
    response.setPortletMode(PortletMode.VIEW);
}
```

Il est important de bien penser à appeler la méthode store() afin que la nouvelle valeur de la

propriété soit stockée par le Portail. Est-ce que vous vous rendez compte que c'est le Portail qui va stocker cette nouvelle valeur ? Si vous n'êtes pas convaincu, arrêtez et relancez GateIn, reconnectez-vous avec le même utilisateur, vous verrez que la valeur a été persisté par GateIn. Il y a quelques détails à connaître : tout d'abord le contenu du PortletPreferences ne peut être modifié que dans la méthode processAction(). Il y a aussi une méthode reset() que vous pouvez appeler pour remettre les préférences à un état stable avant vos changements. Pratique lorsqu'une exception survient. Il est temps maintenant d'informer les autres Portlets que vous venez de changer le nom du compte Twitter. **Envoyer un événement JSR-286 avec une Portlet** Voici ce que je veux faire : lorsque l'utilisateur change le nom du compte Twitter, j'aimerais que ma Portlet envoie aux autres Portlets la valeur de ce compte. Imaginons une autre Portlet qui par exemple recherche sur des flux RSS en se basant sur le nom d'un compte Twitter, ou d'un tag (comme #devoxx). Pour envoyer un événement, il faut tout d'abord le déclarer dans le fichier portlet.xml

(fichier portlet.xml)

```
<event-definition>
  <name>myEvent</name>
  <value-type>java.lang.String</value-type>
</event-definition>
```

Pour que le Portail surveille notre Portlet TwitterPortlet02, il convient maintenant de dire que cette Portlet peut émettre cet événement. Dans la balise portlet, j'ajoute à la fin un tag "supported-publishing-event"

```
<portlet>
  <description>Twitter Portlet sample 02, this portlet demonstrates how to set and display
    a public portlet render parameter.
  </description>
  <portlet-name>TwitterPortlet02</portlet-name>
  <portlet-class>org.letouilleur.portlet.TwitterPortlet02</portlet-class>
  <supports>
    <mime-type>text/html</mime-type>
    <portlet-mode>view</portlet-mode>
    <portlet-mode>edit</portlet-mode>
  </supports>
  <portlet-info>
    <title>Twitter Portlet 02</title>
    <keywords>demo</keywords>
  </portlet-info>
  <portlet-preferences>
    <preference>
      <name>defaultScreenName</name>
      <value>Letouilleur</value>
    </preference>
  </portlet-preferences>

  <supported-publishing-event>
    <name>myEvent</name>
  </supported-publishing-event>
</portlet>
```

Enfin, nous voulons que la portlet TwitterPortlet01 écoute cet événement. Pour cela, voici la définition de la portlet TwitterPortlet01. Nous ajoutons un tag "supported-processing-event" avec le nom du tag.

```
<portlet>
  <description>Twitter Portlet sample 01, this portlet demonstrates the processAction method.</description>
  <portlet-name>TwitterPortlet01</portlet-name>
  <portlet-class>org.letouilleur.portlet.TwitterPortlet01</portlet-class>
  <expiration-cache>0</expiration-cache>
  <supports>
    <mime-type>text/html</mime-type>
    <portlet-mode>view</portlet-mode>
  </supports>
  <supported-locale>fr</supported-locale>
  <supported-locale>en</supported-locale>

  <portlet-info>
    <title>Twitter Portlet 01</title>
    <keywords>demo</keywords>
  </portlet-info>

  <supported-processing-event>
    <name>myEvent</name>
  </supported-processing-event>
</portlet>
```

A savoir : si les 2 portlets sont distribués dans 2 WAR différents, il faut déclarer un element QName afin de qualifier le domaine de l'élément. Par ailleurs l'ordre des tags dans le fichier portlet.xml est déterminant pour GateIn. Si vous souhaitez déclarer qu'une portlet est capable d'émettre comme de recevoir un événement, attention à l'ordre de déclaration dans le fichier portal.xml.

```
<supported-publishing-event>
  <name>myEvent</name>
</supported-publishing-event>
```

```

<supported-processing-event>
  <name>anotherEvent</name>
</supported-processing-event>

<supported-processing-event>
  <name>andYetAnotherEvent</name>
</supported-processing-event>

```

**Envoyer un événement** L'envoi est très simple. Dans la phase d'Action d'une Portlet, il suffit d'appeler la méthode `setEvent` sur l'objet `ActionResponse`. Regardez le code de la méthode `sendActionEvent` de `TwitterPortlet02`, voici comment la valeur du nouveau compte Twitter est envoyé au Portlet

```

// Send an event
response.setEvent("myEvent", screenName);

```

**Recevoir un événement** La réception de l'événement par la Portlet `TwitterPortlet01` est effectué dans la méthode `processEvent`, héritée de `GenericPortlet`. La nouvelle valeur du compte Twitter est lue puis stockée comme nouvelle valeur de `myPublicScreenName`.

```

/**
 * This callback method is triggered when a "myEvent" JSR-286 event
 * is sent to the Portal.
 */
@Override
public void processEvent(EventRequest request, EventResponse response) {
    Event event = request.getEvent();
    if (event.getName().equals("myEvent")) {
        String payload = (String) event.getValue();
        //process payload here
        System.out.println("-----");
        System.out.println(getClass().getName() + " received an event: " + payload);
        System.out.println("-----");

        // Update the property, the Portal will then call the doView
        // to redisplay the page
        myPublicScreenName=payload;
    }
}

```

Après l'appel de la phase Action de cette Portlet, le portail appelle ensuite automatiquement une phase Render, ce qui pour a effet d'appeler la méthode `doView` de `TwitterPortlet01`. Je stocke alors la nouvelle valeur dans la session de la Portlet, c'est tout.

```

public void doView(RenderRequest request, RenderResponse response)
    throws PortletException, IOException {

    if (WindowState.MINIMIZED.equals(request.getWindowState())) {
        return;
    }

    // Set the last value of myPublicScreenName
    request.getPortletSession().setAttribute("myPublicScreenName",myPublicScreenName);

    if (WindowState.NORMAL.equals(request.getWindowState())) {
        normalView.include(request, response);
    } else {
        maximizedView.include(request, response);
    }
}

```

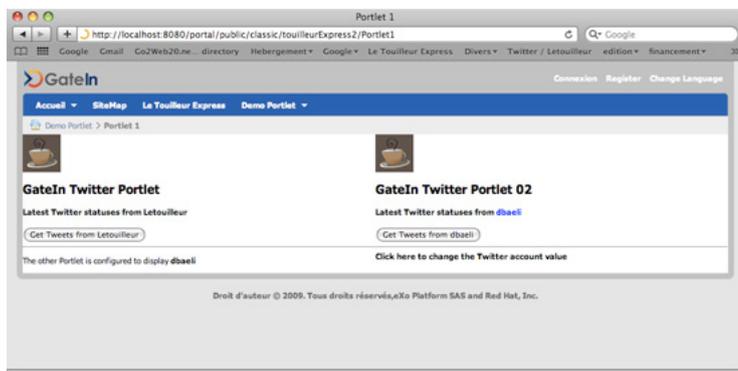
Enfin pour terminer, l'affichage de la nouvelle valeur s'effectue dans la page "normal.jsp" à la fin de la page:

```

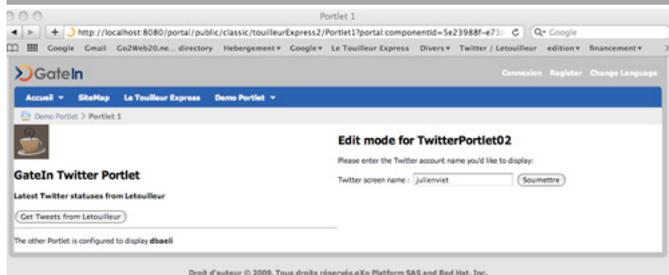
<% String publicParam =(String)portletSession.getAttribute("myPublicScreenName");%>
<% if (publicParam != null && !publicParam.equals("")) { %>
    The other Portlet is configured to display <strong><%=publicParam%></strong>
<% } %>

```

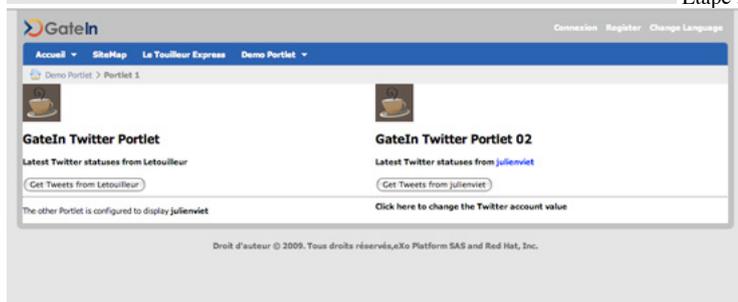
**Le résultat final** Voici une suite de captures d'écran, afin de vous montrer le résultat : Etape 1:



Etape 2:



Etape 3:

**Conclusion**

Dans ce troisième article nous avons vu comment lire et éditer les préférences d'une Portlet. Nous avons ensuite vu comment créer un message, l'envoyer et enfin le consommer dans une autre Portlet. Les prochains articles seront moins bas niveau que celui-ci mais j'ai pensé qu'il serait intéressant de vous montrer les bases des Portlets 2.0 Voir aussi les 2 anciens articles: - [Premiers pas avec GateIn](#), comment installer une Portlet - [GateIn, découverte d'une Portlet simple](#)

## Changement de look du Touilleur Express



Si vous venez habituellement avec un lecteur de flux RSS, dommage vous être entrain de louper quelque chose... Venez donc voir [le site en ligne](#), il a complètement changé d'habillage. Ce week-end j'ai fait un nettoyage et une réorganisation du thème. Moins d'images, plus de CSS, et surtout plus de Widgets WordPress. Cela vous donne un blog plus orienté blogzine, plus vivant. Et encore vous n'avez pas tout vu. Je continue à affiner encore quelques détails, comme tout ce qui concerne Google et AdWords. A noter que je vais retirer les grands bandeaux de publicités. Pour tout vous dire : cela ne me rapporte pas grand chose. Entre 5 et 9 euros par mois. Sinon si vous regardez en bas de la page, vous verrez que j'ai passé l'ensemble du contenu en licence [Creative Commons-Paternité-Pas d'Utilisation Commerciale-Partage des Conditions Initiales à l'Identique 2.0 France fr](#). Vous pouvez donc librement reproduire, distribuer et communiquer le contenu du blog au public, de modifier les articles, tant que vous citez que j'en suis l'auteur (Nicolas Martignole), que vous n'en faites aucunes exploitations commerciales (formation, plaquette d'entreprise, séminaire, livre, article presse). Cela vous donne assez de liberté, sans non plus que quelqu'un puisse faire un livre avec des passages de mon blog. Vous rigolez... mais c'est arrivé il y a quelques mois dans le domaine de l'Agilité, dans un livre en Français dont je ne ferai pas la publicité ici. J'ai utilisé le thème [Arthemis](#) créé par Michael Hutagalung, que j'ai traduit en Français, et adapté pour pouvoir créer une interface de navigation différente. J'ai fait les logos et une partie de la feuille de style, mais la mise en page est le fruit du travail de Michael Hutagalung. Merci à lui ! Concernant les commentaires, j'affiche maintenant les [Gravatars](#), c'est plus sympa. Vous pouvez vous créer le votre à partir de votre adresse email très facilement. Le blog "Le Touilleur Express" existe depuis [le 22 novembre 2003](#), les premiers articles étaient publiés sur [JRoller](#). Aujourd'hui il y a 462 articles et 858 commentaires. Et grâce aux nouveaux Widgets en bas de la page d'accueil, vous pouvez retrouver les articles les plus commentés. Pas les plus lus, puisque je viens d'activer le plugin qui affiche le nombre de vue d'un

article. Ne soyez pas étonner de voir "0 vue(s)" sur un article. Pour terminer, n'oubliez pas le ["Concours spécial Noël 2009"](#). Vous pouvez gagner un livre offert via Amazon par tirage au sort parmi les participants. Le grand vainqueur gagnera un cadeau d'une valeur de **100 EUR**. Donc envoyez-moi une photo par Twitter ou par email, avec votre tête à côté de la page d'accueil du Blog. Go go go !!!

## Les Closures enfin expliquées par un gars qui n'y connaît rien

**Les Closures...** Ce terme claque comme un vieux chewing-gum américain dans chacune de nos conversations de Geeks. C'était le mot à placer dans une conversation à Devoxx 2009 avec le JDK7. Alors voyons un peu de quoi il s'agit. J'en profite aussi pour présenter la version Groovy, avec un peu de code. Je me suis inspiré du livre [Groovy in Action](#) pour vous expliquer les Closures en Groovy, ce qui permet de comprendre leurs utilités en Java. Prenons un exemple simple en Java, pour tout d'abord expliquer quelques principes. Vous avez certainement déjà utilisé la méthode [sort](#) de la classe `java.util.Collections` non ? Cette méthode statique est capable d'effectuer un tri sur une liste en utilisant un comparateur. Imaginons un instant que je sois Georges. Il souhaite ranger ses capsules [Nespresso](#) par ordre alphabétique. Voyez ce que cela donne, avec un tri insensible à la casse :

```
public class ClosureTouilleurExpress {
    public static void main(String args[] ) {
        System.out.println("What else ?");

        List coffeeCaps = Arrays.asList( "Ristretto", "Arpeggio", "Roma", "Cosi", "Volluto", "Livanto", "Indriya");
        Collections.sort(coffeeCaps, new Comparator() {
            public int compare(String s1, String s2) {
                return s1.compareToIgnoreCase(s2);
            }
        });
        System.out.println("Mes capsules Nespresso par ordre alphabetique: " + coffeeCaps);
    }
}
```

L'exécution et le résultat:

```
What else ?
Mes capsules Nespresso par ordre alphabetique: [Arpeggio, Cosi, Indriya, Livanto, Ristretto, Roma, Volluto]
```

En Java il n'est pas possible de passer une référence vers une méthode, mais il est possible de définir une class anonyme, comme je viens de le faire ici. Il y a cependant une limitation: je ne peux pas passer plusieurs méthodes à ma méthode `sort`. Il est possible de s'en sortir en définissant une interface, implémentée par un objet distinct, et ensuite passer cette interface à la méthode `sort`... C'est un peu compliqué non ? Je vais vous raconter la fin du film. Une Closure en Java une fois compilée, sera en fait une classe avec une seule méthode "invoke" static. Cette classe sera générée par le compilateur Java. Ne soyez pas trop triste, c'est la vie. Java ne touche pas au bytecode de la JVM. C'est un artifice d'écriture. Groovy de même. Donc tout ce que vous voyez vise à simplifier notre travail d'écriture, mais ne fera pas aller plus vite votre application. Pensez-y lors de vos dîners en ville. Closure en Java = raccourci pour exprimer un besoin, celui de travail en fonctionnel. **Groovy** Groovy propose déjà les Closures, et je trouve cela plutôt simple. Reprenons notre exemple et transformons-le en Groovy:

```
List coffeeCaps =["Ristretto", "Arpeggio", "Roma", "Cosi", "Volluto", "Livanto", "Indriya"]
coffeeCaps.sort { String param1, String param2 -> param1.compareToIgnoreCase(param2) }
println "Mes capsules Nespress par ordre alphabetique: " + coffeeCaps
```

Plus simple que la version en Java non ? Si vous voulez tester ce script, vous pouvez même faire un copier-coller et lancer [la console Groovy](#), hébergée sur Google App Engine et développée par Guillaume Laforge. La méthode `sort` est implicitement disponible sur les Lists en Groovy. Lorsqu'une méthode est appelée sans arguments, il n'est pas nécessaire de préciser les parenthèses (`sort` au lieu de `sort()`). La construction d'une liste en Groovy aussi est simple, on faisait cela en Perl il y a 10 ans déjà. Enfin le plus intéressant c'est la fonction de tri. La voilà notre fameuse Closure ! La syntaxe est la suivante:

```
{ <arguments> -> <body de la closure> }
```

Les arguments peuvent être typés (`String p1, String p2`) ou non (`p1, p2`). Le corps d'une closure est du code classique, similaire à une méthode anonyme en fait. Lorsqu'une Closure n'a pas d'argument, un objet *it* est créé implicitement. Imaginons que vous souhaitez afficher le contenu de notre liste. Il suffit pour cela d'appeler la méthode `each()` et de passer une closure en argument. Ajoutons une ligne de Groovy dans la console:

```
List coffeeCaps =["Ristretto", "Arpeggio", "Roma", "Cosi", "Volluto", "Livanto", "Indriya"]
coffeeCaps.sort { param1, param2 -> param1.compareToIgnoreCase(param2) }

coffeeCaps.each{ println it } // Ici la closure n'a pas d'argument, le signe -> n'est pas indispensable
```

Et exécutons ensuite le tout:

```
Arpeggio
Cosi
Indriya
```

```
Livanto
Ristretto
Roma
Volluto
```

La méthode each() de la Liste appelle la méthode de la closure pour chacun des éléments de la liste. L'itération de la liste donne un curseur (it) que j'utilise pour afficher directement la liste. Simple non ? Une Closure en Groovy finalement c'est un objet, que l'on peut passer en argument d'une méthode, affecter à une instance, bref aussi facile à manipuler qu'un Object classique. Dans l'exemple ci-dessous, j'ai créé un objet Closure avec l'algo de tri par ordre alphabétique. Regardez comment la méthode sort() prend en argument cette Closure. Je vous laisse imaginer les possibilités que cela vous offre.

```
Closure rangeCapsule = { String p1, String p2 -> p1.compareToIgnoreCase(p2)}

List coffeeCaps =["Ristretto", "Arpeggio", "Roma", "Cosi", "Volluto", "Livanto", "Indriya"]
coffeeCaps.sort(rangeCapsule) //je passe la closure en argument de la methode sort()

coffeeCaps.each{ println it }
```

Est-ce que Groovy a réellement des Closures ? D'un point de vue technique, les Closures deviennent des Classes, qui seront ensuite compilées. Une Closure n'est jamais une méthode en Groovy. Mais le principe est vraiment pratique. Pour vous aider à comprendre par rapport à Java, imaginez en fait une class avec une méthode static, comme cet exemple [du site Groovy](#):

```
public class MyMath {
    public static int square(int numberToSquare){
        return numberToSquare * numberToSquare;
    }
}
...
...
int x, y;
x = 2;
y = MyMath.square(x); // y will equal 4.
```

La même chose en Groovy devient:

```
// Exemple complet
Closure test={ numberToSquare -> numberToSquare * numberToSquare }
println test(2) // affiche 4
```

**Et Java ?** Mark Reinhold's a publié un billet sur [son blog](#), et il semble qu'il n'y ait plus que 3 propositions pour ajouter le concept de Closure au langage Java dans le JDK7 prévue pour 2010 ou 2011. Les 3 propositions retenues sont CICE, FCM et BGGGA. Une série d'articles très bien écrits sur [Developez.com](#) vous donne une présentation détaillée de chacune des propositions. [CICE](#) (spec), par Bob Lee, Doug Lea et Josh Bloch. FCM (spec), par Stephen Colebourne et Stefan Schulz. [BGGGA](#) (spec), par Gilad Bracha, Neal Gafter, James Gosling et Peter von der Ahé. **BGGGA** La proposition BGGGA est la plus complète. Elle est parfois un peu complexe à comprendre, mais le principe est de déclarer d'une part les paramètres d'entrées/sorties, et ensuite de spécifier l'algo. L'exemple de la fonction "sum" pour réaliser une addition se décomposerait de cette façon:

```
{ int, int => int } additionne = { int x, int y => x+y };
```

L'usage de la closure type BGGGA serait alors:

```
int resultat=additionne.invoke(2,5);
// resultat vaut 7
```

La closure est définie en 3 zones. 1) Une zone où l'on spécifie le type de paramètres en entrées, le type retourné

```
{ int,int => int}
```

2) le nom de la closure

```
additionne
```

3) la définition de la closure

```
{ int x, int y => x+y }
```

La version FCM est différente, voici ce que cela donne:

```
 #(int(int,int)) plus = #(int x, int y) { return x + y; }
```

Allez lire [l'excellent article d'adiGuba](#) sur Developez.com **FCM** La plus grosse différence est l'utilisation du caractère # (dash) qui n'est pas utilisé en Java. Quelque part, cela pourrait nous aider à repérer une Closure rapidement, pourquoi pas ? **CICE** L'implémentation se base sur la simplicité. Assez différente des 2 autres, la proposition permet d'alléger du code classique. Voyez l'exemple de tri des éléments d'une liste par taille :

```
List ls = ... ;
Collections.sort(ls, new Comparator() {
    public int compare(String s1, String s2) {
        return s1.length() - s2.length();
    }
});
```

Avec la proposition CICE:

```
List ls = ... ;
Collections.sort(ls,
    Comparator(String s1, String s2){ return s1.length() - s2.length(); });
```

**Revenons à Groovy** Et juste pour le fun, voici l'implémentation en Groovy:

```
List coffeeCaps =["Ristretto", "Arpeggio", "Roma", "Cosi", "Volluto", "Livanto", "Indriya"]
coffeeCaps.sort { String p1, String p2 -> p1.length() <=> p2.length() }
println "Par ordre croissant: " + coffeeCaps
```

Ce qui donne:

```
Par ordre croissant: [Roma, Cosi, Volluto, Livanto, Indriya, Arpeggio, Ristretto]
```

**Conclusion** Je vous ai présenté le principe des Closures en me basant sur Groovy, dont la syntaxe est simple. J'avoue que j'ai un peu peur de ce que nous sommes entrain d'essayer de faire au langage Java en lui-même. Les Closures n'étaient pas nécessaires jusqu'à maintenant en Java. Simplement, les vieux développeurs s'ennuient. Et comme ils sont nombreux et qu'ils tiennent à ce que l'on se souvienne d'eux, ils tentent de proposer des innovations et des idées. Certaines sont bonnes, d'autres sont discutables. Je pense que les Closures sont une bonne idée, mais je suis plutôt [circonspect](#) quant à sa bonne utilisation par un junior devant son écran... J'ai peur. Réflexe de vieux. Lorsque le problème survient, avec son domaine d'application, je pense qu'il est plus intéressant d'apprendre un autre langage qui tourne sur la JVM comme Scala ou Groovy, plutôt que d'attendre un hypothétique consensus sur les Closures. Alors oui, il y aura des Closures en Java, la syntaxe sera simple ou non, mais je préfère maintenant investir mon temps à apprendre Scala et Groovy.

## Soucis sur le flux RSS du Touilleur Express

J'ai des soucis sur le flux RSS. Est-ce que quelques âmes charitables peuvent m'aider en m'indiquant: - l'url utilisé (exemple: <http://touilleur-express.fr/rss2>) - le logiciel de lecture de flux rss (ex: Google Reader) - l'url de l'article qui pose problème Merci pour votre aide

## Merci à l'équipe Primeweb



C'est fait, j'ai terminé ma mission de 14 mois chez BNPParibas, à l'Arbitrage, dans le département Prime Brokerage. Un pot de départ plus tard, et je suis revenu à la maison avec un [Nabaztag](#), offert par tout le monde. Un grand merci à tout le monde, j'ai baptisé le lapin "primeoueb" en souvenir de cette année passée avec vous. Bonne chance à tous pour la suite. Arrivé en septembre 2008, j'ai tout d'abord travaillé avec Nicolas Griso de [Xebia](#) pour remettre à plat toute l'usine logicielle. Passage de Maven 1 à Maven 2, amélioration des temps de compilation, et développements sur les parties un peu touchy, comme un cache par exemple. Zéro Scrum, pas de gestion de projet, rien. Ce n'était pas encore le moment. Quelques semaines plus tard, j'ai proposé de passer en Scrum complet l'équipe. Nous l'avons tout d'abord divisé en 3 petites équipes pour refléter les différences fonctionnelles dans le produit. Il est vrai qu'avant cela, nous faisons un stand-up à 14 personnes... qui durait trop longtemps. En janvier je me souviens être descendu aux services généraux pour récupérer des post-its, du scotch et des grandes feuilles de papier pour préparer des tableaux Scrum. Là grosse déception : post-it d'une seule couleur, pas de stylos pour écrire, et pas de feuilles. Après avoir piqué un rouleau dans une salle qui trainait, j'ai passé une commande sur JPG afin de se faire livrer dès le lendemain un gros stock de Post-Its de toutes les couleurs, du scotch, des feutres, bref de quoi bosser. Moment marrant : l'arrivée le lendemain au bureau. Me voilà entrain de dessiner sur une grande feuille 3 colonnes: "A faire", "En cours" et "Terminé". Je scote le tout sur un mur, devant la cafétaria. Environ 14mn et 23 secondes plus tard, GrandChef qui passe par là me demande ce que je fabrique. Je lui explique que je fais du Scrum, avec l'accord de ChefPrimeweb, un gars bien qui est curieux de voir ce que l'on va faire. C'est là qu'il faut être rodé pour expliquer Scrum en 3 minutes, sans vous planter. Grâce aussi à l'aide de Nicolas de Xebia, à deux nous lui proposons une petite présentation rapide au coin du comptoir de Scrum. On sent le scepticisme mais je détecte de la curiosité dans le regard... c'est gagné. Premier exercice pratique pour les développeurs un matin : je leur demande de prendre des post-its et de marquer ce qu'ils sont entrain de faire actuellement. Au bout de 10 minutes, la colonne "En cours" est tellement remplie que les discussions entre les développeurs ressemblent à une négociation pour le meilleur emplacement sur un marché dans un petit village. Et vas-y que je te pousse ton post-it pour mettre le mien, et que ça discute car en fait ils sont 2 à bosser sur la même chose (et ils ne le savaient pas), et que ça demande aussi s'il faut mettre les bugs... Oui allez-y, mettez les bugs, on mettra une deuxième feuille de 2m par 3 pour la suite... Bref j'exagère car en fait ce n'était pas autant le bazar. Mais l'intérêt de cet exercice était de montrer que chaque développeur a trop d'activités en cours. Que bien souvent, une tâche n'est pas terminée car elle attend une approbation, plus de détails. Ou parfois aussi, parce que finalement le développeur est bloqué. Alors il prend une autre tâche... Exercice simple qui permet de voir aussi la quantité de boulot que l'équipe est entrain d'effectuer. Nous travaillons ensuite sur la construction d'un Product Backlog. L'idée était de lister toutes les demandes, de les classer du plus important au moins important. Cela permet dès les premiers jours de ne se concentrer que sur les éléments les plus importants. Grâce à cela, Karim qui joue le rôle de Product Owner, peut aussi collecter les demandes des analystes financiers et les trier rapidement. Nous supprimons aussi rapidement les statuts utilisés jusqu'à maintenant. Avant, l'importance des demandes étaient classées de cette façon

: "Nice to have / Normal / Important / Très important". Le degré d'urgence : "Bas,Normal,Haut". Et enfin la criticité dans le cas d'un bug "Trivial / Non bloquant / Bloquant". Cette soupe de paramètre était interprétée différemment selon les personnes. Les développeurs se concentrent sur la criticité, là où quelqu'un du métier va plutôt hésiter entre "Important" et "Très important". Je demande un jour: "que fais-tu lorsque tu as 1 bug bloquant/Haut/Très important et qu'un deuxième bug avec ces mêmes caractéristiques entre dans ton système ? Tu mets une nouvelle catégorie comme *Très très important* ?". Bien entendu non. Nous avons donc basculé les JIRA avec un nouvel attribut "Scrum priority" afin de trier du plus important/urgent au moins important l'ensemble des bugs et des demandes d'amélioration. Chaque élément reçoit un numéro. Plus ce numéro est élevé, plus le JIRA est important. Un passage dans une Feuille Excel et voilà un Product Backlog priorisé. A l'époque les 5 premiers étaient numérotés 764 760 500 300 100 et 50. Aucune règles de calcul, mais nous aurions pû en mettre une pour dire que "cette tâche est 5 fois plus importante que cette autre tâche" par exemple. Ensuite nous avons tenté d'utiliser JIRA et [Greenhopper](#). Sympa, mais finalement j'allais plus vite avec un fichier Excel pour faire notre suivi de Sprint. Les Sprint Backlog et le Burndown chart étaient fait avec Excel. Je vous donne un conseil de vieux : ne vous excitez pas sur un outil comme Greenhopper ou autre. Pensez que l'outil doit être simple, efficace. Si une feuille de papier fait l'affaire... très bien. Si vous êtes à l'aise avec Greenhopper ou IceScrum, très bien. Mais je vous recommande par contre de faire vos tableaux de suivi de Sprint sur une feuille collée au mur. Je suis aussi passé chez Vidal et ils font exactement la même chose. Rien de mieux qu'un post-it que tu déplaces le matin en discutant, lors du standup meeting. Nous avons ensuite mis en place les cérémonies. J'ai trouvé que le rythme de mise en place était un peu lent. Un matin je me souviens avoir dit devant tout le monde : "*Bon les gars... Là le projet est entrain de souffrir. On essaye de retirer un pansement tout doucement, alors qu'il vaudrait mieux arracher un bon coup ce pansement et passer à la suite. Je propose que l'on fasse du Scrum à fond, et pas du Scrum light. C'est pire que tout.*". Et bien c'était le truc à faire. Ne pas jouer avec une partie de Scrum, c'est pire. Pour la rétrospective, j'ai un peu galéré au départ. Je ne savais pas comment faire parler les gars. Grand silence poli... certains mêmes bidouillaient leur iPhone pendant que je tentais d'animer la rétrospective... C'était gonflant. Finalement j'ai essayé quelques techniques. L'une des techniques qui a bien marché était la suivante. J'ai demandé aux développeurs de prendre les post-its des tâches livrées pendant le Sprint, et de les classer sur une ligne de temps sur un tableau. A gauche, l'élément qui a demandé le moins de temps. A droite, celui qui a demandé le plus de temps. Les développeurs ont donc dû discuter entre eux pour s'expliquer ce qu'ils avaient fait, afin de placer les post-its correctement. Pendant ce temps là je buvais un café tranquillement avec Karim, le Product Owner, qui du coup avait une rétrospective clé en main. Cela prend 5 minutes, et ça marche. Nous sommes ensuite venu voir les écarts entre les estimations/cotations données par l'équipe 2 semaines avant, et la ligne de temps. Marrant de voir un JIRA côté à 3 points complètement à droite d'un JIRA côté à 5. En fait, celui à 5 n'était pas si compliqué et long à faire. Aaaaah les jours hommes... Unité aussi inutile que bénie des Chefs de Projets... En fait c'est une unité qui vous donne une estimation assez précise du moment où vous allez vous planter. Car cette unité ne marche pas. J'en suis convaincu. J'ai une des équipes qui continue à s'en servir, je ne sais pas comment dire qu'il faut arrêter... Bon lorsque j'étais petit, à l'école nous parlions de voiture dans la cour. Et malheur à toi si ton père avait une voiture avec un moteur Diesel. Les enfants se moquaient de toi, boouuh ta voiture elle est toute pourrie. Et puis 20 ans plus tard, les mêmes enfants sont devenus des adultes. Avec une Golf TDI, une Seat Ibiza ou un Picasso HDI... C'est bien, vous avez compris. Et l'essence aujourd'hui, c'est old-school, limite hasbeen. J'attends le jour où ne pas avoir de véhicule hybride sera devenu ridicule... Les jours hommes c'est pareil. Cela serait bien que vous arrétiez. Vraiment. C'est bien trop réducteur. Alors lorsque quelqu'un demande "*Combien de temps pour faire ce développement ?*" vous devez mettre en marche votre cerveau afin de calculer un chiffre composé de la façon suivante: *temps estimé X risque X niveau\_de\_confiance X ce\_que\_vous\_avez\_mangé\_ce\_midi* Cela donne des chiffres rigolos qui ne veulent rien dire. Alors nous vous donnons un jeu de carte, estampillé Xebia, appelé "Planning Poker". Je vous en ai parlé [il y a bientôt 2 ans](#), ne faites pas ceux qui ne savent pas. Vous demandez à l'équipe de regarder la liste des tâches à faire. Quelle est la tâche que tout le monde connait le mieux, et qui a le moins de risque ? Pour nous, c'était faire une release du logiciel. Cette activité prend 2 jours calendaires, elle est subordonnée à quelques étapes importantes, des tests etc. Nous lui avons attribué la note de "3". Ensuite, j'ai demandé à chacun de classer les JIRA en utilisant cette échelle. Et nous avons mis 3 sprints pour nous créer une échelle de valeur. Un JIRA côté à 1 représente quelque chose qui ne prend qu'une demie journée, voir une journée, qui est facile et qui n'a pas de risque. Un JIRA côté à 3 peut représenter quelque chose de facile à faire mais qui est long (>2 jours) ou simplement une tâche compliquée mais qui ne devrait pas prendre plus de 2 heures (identifier la raison d'un bug en jouant un test fonctionnel). Bref nous avons mis du temps à nous créer cette échelle de valeur. Et vu de l'extérieur, impossible pour un nouveau de comprendre qu'un JIRA marqué à 20 représente en fait 4 jours ou 7 jours... Cette unité Scrum permet de mixer intelligemment votre confiance par rapport à la demande (est-ce que je sais faire ou pas), votre niveau d'expertise (est-ce que c'est mon domaine ?), votre peur (je le sens bien ce développement là) et aussi votre humeur (en ce moment je suis fatigué donc compte pas trop sur moi). L'exercice de vote était sympa à faire avec l'équipe. Cela permettait d'animer la réunion. Il fallait un peu aussi montrer comment cuisiner le Product Owner afin de cadrer ce qu'il y avait à faire et ce qu'il ne fallait pas faire. Grâce à cela, les engagements et le contenu était plutôt précis. J'ai regretté que les développeurs ne bossent pas à deux si souvent, mais c'était normal vu le contexte du projet et la quantité de travail à fournir. Après avoir lancé Scrum, j'ai fait aussi quelques développements, du tuning Hibernate, de l'amélioration de certains composants, des tests et de l'analyse des performances, de l'analyse de la dette technique avec des outils comme XDepend, un peu d'architecture... bref des trucs sympas. En août 2009 j'ai commencé une étude sur les Portails d'Intégration suivants : Liferay, eXo Portal et Oracle Weblogic Portal. Bonne expérience, qui m'a donné aussi l'occasion de rencontrer du monde chez Oracle, chez eXo Platform et chez Jahia. Au final il n'y a pas de vainqueurs, plus une recommandation par rapport aux besoins de l'équipe. Nos voisins américains ensuite sont venus travailler avec nous. Sympa, mais j'ai aussi mis les pieds dans le plat en expliquant que leur solution de portail basée sur GWT-Ext était morte (depuis [juin 2008](#)). Mon boulot est aussi de dire lorsque le client fait un mauvais choix. Depuis ils basculent vers GXT. C'est mieux. En partant je garde de bons souvenirs, où j'ai aussi

appris beaucoup du côté fonctionnel, grâce à des personnes qui aiment leur boulot. Bonne route à tous et merci pour cette expérience. Et le lapin s'appelle "primeoueb" donc je penserai à vous.

## Ne pas (trop) négocier les jours hommes



Si tu lis cet article c'est que quelqu'un qui te veut du bien a pensé qu'il serait intéressant que tu le lises. Ce matin les développeurs sont venus te voir. Comme chaque semaine, vous faites un point. Tu souhaites que l'équipe de développement code un questionnaire d'inscription pour ton site. Les écrans sont déjà prêts, il faut simplement maintenant implémenter le code derrière. La discussion commence. Dans ta tête, tu te dis "il y en a pour 2 jours". Mais ces développeurs sont de sacrés lascars. Et tu n'as pas confiance en eux. C'est dommage. On va travailler là dessus aussi. *Les amis, je veux lancer mon navigateur dans 2 jours, pouvoir remplir un questionnaire d'inscription et ensuite recevoir par email mon identifiant et un mot de passe.* Bravo tu viens de faire ta première User Story. Je me permet cependant de revenir sur un mot dans ta phrase qui m'agace : "...dans 2 jours". Au nom de quel pouvoir magique penses-tu que 2 jours c'est suffisant ? Et que se passe-t-il si nous pouvons le faire en 1 jour ? Nous irons boire des bières le deuxième jour ? L'un des premiers principes de Scrum, comme lorsque vous allez chez le médecin, est le suivant : tu t'assois, tu écoutes et tu ne négocies pas le traitement pour ta maladie. - *Euh docteur, je pense que j'ai une angine et qu'il me faut des antibiotiques - Euh cher patient, je pense que vous êtes un abruti et que si aviez fait 10 ans d'étude, vous seriez en mesure de me dire ce que je dois faire.* Vous devez faire confiance à l'équipe de développement. Et en retour, les développeurs doivent faire un effort de transparence, d'engagement et de communication. Car eux aussi ne sont pas parfaits, loin de là. **Il faut travailler sur le contenu, pas sur son prix** Lorsque vous devez travailler avec une équipe de développement, voici mes conseils : soyez précis dans votre demande. Mettez vous en situation dans quelques jours, et racontez sous la forme d'une histoire ce que vous voulez. Surtout, ne cédez pas à la tentation de proposer une solution technique pour l'instant. Exprimez votre besoin, et faites-le en direct devant les développeurs. Voici ce que cela donnerait si je reviens à votre demande de tout à l'heure : - *Les gars, en tant qu'utilisateur anonyme sur notre site, je souhaite remplir le questionnaire d'inscription et recevoir mes identifiants par email - Combien de questions y-a-t-il ? - Il y a 8 questions - Est-ce que tu veux que l'on divise cela sur plusieurs écrans ou est-ce que l'on place les 8 questions à la suite ? - Cela change quoi ? - Si tu divises sur plusieurs écrans, nous devons sans doute gérer le bouton "Back" du navigateur, cela prendra un peu plus de temps. - Et bien vu que les questions sont assez complètes, même si cela coûte plus cher je préfère en tant qu'utilisateur, utiliser plusieurs écrans successifs, et même revenir en arrière pour corriger ma saisie - Est-ce que l'identifiant peut être constitué d'un email seulement ? - Oui - Est-ce que tu veux un Captcha pour bloquer les bots qui spamment ? - Ah oui ! mais pas cette semaine. On est pas en ligne avant 2 mois - Que faisons-nous avec ce questionnaire ? - Je veux le stocker dans une base de données et ensuite regarder avec Excel son contenu - Pour cela, il faut 1 à 2 jours pour définir le modèle et t'installer la base. - En fait mon besoin pour l'instant c'est que les 30 bêta-testeurs puissent s'inscrire et tester - Donc une solution sans base de données, avec un fichier texte tabulé sur le serveur, avec un fichier par identifiant créé te suffira ? - Oui c'est parfait, et je vous recommanderai plus tard la base de données, car on est pas en ligne avant 2 mois - Ok merci. Allez les gars on vote !* Et l'équipe va vous donner le prix. C'est un devis, non négociable. Si le prix en jours/hommes vous fait peur, c'est que votre perception de la complexité n'est pas la même que celles des développeurs. Il faut absolument rediscuter tout de suite avec les développeurs sur le contenu. **Demandez un devis détaillé** Si vous sentez qu'il y a une zone de flou dans la réponse chiffrée, demandez à détailler les étapes. Essayez de voir s'il est possible de valider une première partie. Restez pragmatique, c'est vous le client. Ne laissez pas non plus les développeurs vous vendre des options dont vous n'avez pas besoin. **Ne vous focalisez pas trop sur les jours/hommes** Je vous pousse un peu plus loin. Attention à la notion de jour/homme. C'est une vision des années 90 de l'informatique. Ce serait penser que 3 développeurs vont plus vite qu'un développeur, ce qui est souvent le cas, mais pas toujours. Cette notion est dépassée. Ce qui est important aujourd'hui, c'est de travailler sur de nouvelles valeurs : l'engagement et la transparence. Et vous allez voir que cela fonctionne. Ce que vous demandez avec un chiffrage, c'est un engagement sur une date. Or le plus important, c'est le contenu. **Comment suivre les jours alors ? et s'assurer que les développeurs bossent ?** Lorsqu'une équipe vous donne un chiffrage en jours ou en nombre de patates, marquez-le et gardez-le sur une feuille de papier, avec les détails de la discussion. Si l'équipe s'est engagée, elle vous livrera ce que vous avez demandé, en temps et en heure. Si vous constatez un décalage, attendez la rétrospective si vous souhaitez en parler. Mais attention : ne revenez pas sur le passé. Une équipe s'engage pour développer la fonction expliquée ci-dessus en 4 jours. Finalement lors de la rétrospective, elle a travaillé sur ce sujet 2,5 jours. Après enquête, un des développeurs a trouvé une librairie et ils ont donc gagné un peu de temps. Et bien c'est très bien, laissez tranquille les développeurs. Ils ont livré à temps ce que vous aviez demandé, et au lieu de remplir 4 jours en ajoutant d'autres fonctions, ils se sont arrêtés à 2,5 jours. Et croyez-moi, un développeur a vite fait de vous rajouter des trucs qui ne servent à rien. Ce que je vous demande c'est de faire confiance à l'équipe. Lorsque je vais chez le garagiste, je lui fais confiance. Il me donne un devis, c'est trop cher, je ne vais pas faire réparer ma voiture. Lorsque je suis au restaurant, le menu affiche 19 EUR pour un steak-frite. Je ne vais pas négocier le prix avec le serveur en demandant à retirer 8 frites pour que le steak coûte 15 EUR. Lorsque vous êtes au rayon fruits et légumes, que vous remplissez votre sachet de tomate pour le faire peser, vous faites une estimation de la quantité qu'il vous faut. Est-ce que vous revenez prendre une autre tomate après avoir fait peser votre sac ? non. Alors laissez tomber cette mauvaise habitude de négocier le prix d'un développement. Négociez et discutez sur ce qu'il faut faire, et faites confiance aux développeurs. **Un mot pour nos amis les développeurs** Les gars, si vous voulez qu'on arrête de vous prendre pour des garagistes, soyez plus transparent. Il faut aussi tenir vos engagements. Si vous annoncez 2 jours, expliquez pourquoi. Il est important d'être honnête et de faire un effort pédagogique afin d'expliquer ce que vous allez

faire. Lorsqu'un sujet demande un peu de recherche et que vous n'êtes pas certain, dites-le. Il vaut mieux être clair dès le départ. Utilisez le mode interrogatif pour bien comprendre et écouter la demande du client. Par expérience, je me suis rendu compte qu'une discussion de 10 minutes évitait parfois des développements de plusieurs semaines. Un des gros soucis de notre formation d'ingénieur, c'est que l'on a été formé pour être indépendant. Or la vraie vie et le monde informatique vous demande exactement le contraire. On ne vous demande pas d'inventer à chaque fois une nouvelle méthode ou un nouveau framework. On vous demande juste de faire votre boulot, et d'être inventif pour que la solution soit industrialisable, facile à maintenir, bien codée et facile à utiliser. A vous aussi de vous remettre en question et de faire un effort pédagogique. Si vous vous comportez comme un garagiste, je ne suis pas étonné que votre client négocie avec vous les jours/hommes. Le secret est bien là : peu importe la méthode, c'est vous qui devez changer si vous souhaitez que votre client change. Et si le client ne change pas, changez de job. Tout le monde n'est pas prêt à changer. Je vous laisse, j'ai encore du boulot pour tenir mon engagement.

## Retour sur le livre "Android" par Florent Garin



J'ai reçu il y a quelques semaines le livre "[Android](#)" de Florent Garin, paru aux éditions Dunod. Je dépèle tout doucement la tonne de livre que je dois lire, et j'aime aussi prendre mon temps. D'où un retour un peu tardif par rapport au livre qui est sorti fin octobre. Avec 217 pages, le livre s'adresse aux développeurs qui souhaitent découvrir et commencer à programmer sur la plateforme Android. Le livre est très concret, et couvre les points les plus importants, tout en évitant de noyer le lecteur avec trop de détail. J'aime bien le style, à la fois simple mais didactique. Certaines parties sont un peu longues, mais elles serviront le lecteur à la recherche d'un guide de démarrage complet. Pour moi qui ne connaît rien à Android, j'ai refermé le livre après sa lecture, avec le sentiment d'avoir compris la spécificité du développement dans cet environnement. Les 4 composants clés sont bien expliqués : Activity, Service, BroadcastReceiver et ContentProvider. J'ai été un peu étonné en fait de l'API de Google. Il y a là aussi quelques restrictions, dues au moteur d'Android. **Ce que j'ai retenu** Florent Garin débute le livre après avoir expliqué l'alliance Open Handset par une présentation du coeur d'Android : basé sur un noyau Linux 2.6, nous apprenons que le coeur d'Android est constitué entre autre d'une implémentation d'OpenGL, d'un moteur de rendu web WebKit, ainsi que d'une base légère SQLite. Android tourne avec une machine virtuelle appelée [Dalvik](#). Ce n'est pas une machine virtuelle Java complète. Elle ne respecte pas les spécifications de SUN. Dalvik est une VM dite "[register-based](#)" là où une JVM classique est dite "[stacke-based](#)". L'utilitaire `dx` est utilisé pour convertir les `.class` d'une application Java en `.dex`, le format Dalvik. Chaque application Android tourne avec sa propre VM, ce qui permet aussi le multi-tâche dans un téléphone Android. Dalvik n'a pas de JIT, mais des optimisations permettent d'améliorer énormément les performances et surtout, de réduire la consommation énergétique des applications. Cela semble bête, mais la gestion de l'énergie est un gros challenge sur un téléphone embarqué. Romain Guy vous en parlerait mieux que moi. Le modèle de programmation est détaillé à partir du chapitre 3. Assez proche de nos habitudes, on se rend compte cependant rapidement que le développement est plutôt typé "C" en regardant les différents exemples que l'auteur propose. Il vous amène sans soucis devant Eclipse et vous aide à configurer votre environnement avec l'émulateur, c'est très pédagogique. Page 39 vous ferez la connaissance de la classe "R" comme Resources. Assez impressionnante, elle est générée par le plugin Eclipse et permet de regrouper sous la forme de constantes, toutes les ressources de votre application : boutons, message, nom de l'application, cases à cocher, etc. L'interface graphique ensuite est détaillée, avec une présentation de chacun des Widgets. Il est possible de construire son interface soit en la programmant à la Swing, soit en la déclarant en XML, un peu comme du MXML pour ceux qui connaissent Flex. Ce chapitre est intéressant mais si je devais faire une critique, je regrette qu'il n'y ait pas une présentation d'une petite application, afin de nous montrer un cas plus concret. C'est plus une énumération, plutôt bien faite car le code est détaillé. Le chapitre 5 est l'un des plus importants car il explique le modèle de composants d'Android. Les 4 composants sont : Activity, Service, BroadcastReceiver et ContentProvider. C'est la partie la plus intéressante et la plus compliquée du livre. Elle détaille avec un exemple les principes d'échanges de messages entre les applications. Comment une application peut-elle se réveiller lorsqu'un SMS arrive sur le téléphone ? La notion d'Intent, sorte de messages qui traversent le système est intéressante. En bref, une notion intéressante apparaît dans le chapitre : comme toutes les applications de votre téléphone peuvent traiter un type d'événement, celles-ci échangent en fait et collaborent pour faire des activités sur le téléphone, ce qui casse la notion d'application isolée. Cela évite aussi de devoir tout implémenter tout le temps. La suite du livre explique la persistance des données, comment sauver soit via le système de fichier, soit via SQLite. Ensuite une partie assez importante est consacrée à l'interface utilisateur. A la fin du chapitre vous serez en mesure d'utiliser des composants comme la barre de status, de jouer une animation en 2D et vous aurez survolé la partie 3D. Je pense aux créateurs de jeux et je pense que cette partie les laissera sur leur faim par contre. Le chapitre 8 est consacré au matériel. Comment envoyer et recevoir des SMS ? passer des appels vocaux ? Comment utiliser Google Maps et la géolocalisation ? Comment aussi utiliser l'appareil photo, le vibreur, l'accéléromètre et aussi l'écran tactile. A la page 154 un exemple simple vous montre comment afficher votre latitude et votre longitude, en quelques 40

lignes. L'API n'est pas forcément simple à programmer, mais l'auteur restitue cela plutôt bien, les exemples sont clairs. Le dernier chapitre est par contre le moins intéressant. Il est consacré aux échanges entre Android et le monde extérieur via le réseau. La partie sur XMPP est sympa, mais il y a quelques pages que j'ai laissées tomber comme la page 187 et sa tartine de code. En même temps ce sujet pourrait demander à lui tout seul une bonne cinquantaine de pages supplémentaires.

**Conclusion** Le livre "Android" est sympa à lire et il remplit bien son rôle : vous faire découvrir la programmation sur le moteur de Google. La maquette du livre est plutôt bien faite. A titre personnel j'aime moyennement le format. Par contre on apprécie l'entête et le résumé de chaque chapitre, ce qui permet de retrouver rapidement un passage. Si je devais laisser une critique par rapport au contenu, c'est qu'il manque une petite application complète par exemple, qui aurait été écrite au fil du livre. Cela laisse le champ libre pour quelqu'un qui souhaiterait écrire "Android par la pratique". A acheter si vous voulez programmer sur Android et si vous voulez comprendre simplement les possibilités de cette nouvelle plateforme. A 29 EUR en plus, le rapport qualité/prix est intéressant. Références: "[Android, Développer des applications mobiles pour les Google Phones](#)" par Florent Garin. Préface de Sylvain Wallez. 214 pages, éditeur Dunod. **Update:** le livre utilise la version 1.5 r2 (page 22). Il existe 3 versions du SDK au 6 décembre 2009 : - [Android SDK 1.5 r3](#) avril 2009 - [Android SDK 1.6 r2](#) décembre 2009 - [Android 2.0.1 r1](#) décembre 2009 A noter que la version 1.6 et la version 2.0.1 sont encore des versions "Béta". Florent si tu lis ces lignes, est-ce que tu peux donner ton avis et répondre à Guillaume ?

## Evénements des JUGs en décembre 2009

Voici une liste (qui sera mise à jour selon votre retour) avec la date et le nom du JUG (Java User Group) pour les soirées du mois de décembre: **Naissance du 14ème JUG Français, l'AlpesJUG basé à Grenoble** Loïc Descotte m'a informé que [l'AlpesJUG](#) ouvre ses portes en cette fin 2009. Organisé par Loïc Descotte, Emmanuel Hugonnet, Miguel Moquillon et Bruno Vernay. J'imagine que les speakers vont se bousculer cet hiver pour aller présenter leurs sujets là-bas et profiter ensuite des pistes de skis ! Bravo les gars, vous êtes les 14ème à rejoindre la famille des JUGs Français. **Lundi 7 décembre** Le Lorraine JUG organise une soirée spéciale [Google](#) avec Didier Girard de SFEIR France et Olivier Girardin de SFEIR Benelux, auteur d'un livre sur GWT que je vais lire. A partir de 18h30 à Villers-lès-Nancy Le BreizhJUG organise [une soirée Java EE6](#) avec la présence d'Antonio Goncalves, du Paris JUG, auteur du livre sur JEE6 dont je vous ai déjà pas mal parlé. **Mardi 8 décembre** Le Tours JUG organise une soirée "[Réussir vos projets](#)" avec Guillaume Dusquenay et Benoit Lafontaine d'OCTO. Vaincre la force du plouf est l'une de mes présentations favorites de l'USI 2009, à aller voir d'urgence, en espérant que les grèves de la SNCF ne perturbent pas ce programme. Le Paris JUG monte une soirée "[Java EE 6 et Spring 3.0](#)" avec la présence des stars du circuit. Les 240 places sont parties en quelques heures, c'est donc complet pour mardi soir. J'y serai, donc je vous ferai un rapport détaillé comme d'habitude. **Mercredi 9 décembre** Le RivieraJUG basé à Sophia-Antipolis vous propose une soirée Intégration continue. J'en profite pour faire un appel aux sponsors, le RivieraJUG a besoin d'un coup de main. Si vous voulez aider une structure sympa basée à Sophia, contactez rapidement [l'équipe du JUG](#). **Jeudi 10 décembre** Le MarsJUG (Marseille) organise une soirée [RESTful web services avec JAX-RS](#) présentée par Stéphane Epardaud. Le NantesJUG organise [une soirée sur l'open-source](#) avec la présentation du projet open-source Acceleo par Yvan Lussaud et ensuite une présentation du développement dans le monde du logiciel libre par Mikael Barbero, committer de la fondation Eclipse. Le JUGL (JUG Lausanne) propose [une soirée Grails/Groovy](#), avec une présentation de Cyril Picat d'Octo Technology. Le ChtiJUG organise [une soirée HTML5 chez Adeo](#) à Lezennes. A noter que le JUG utilise le moteur EventBrite pour gérer l'inscription des Juggeurs. Nous en discutons pour le Paris JUG, pour gérer entre autre les désistements. **Vendredi 11 décembre** 2ème conférence "[Innovations Web](#)" de 13h00 à 16h30. Là il ne s'agit pas d'un JUG mais l'événement aura lieu à Paris, et il est organisé par le [G-E-T](#) (Google EFREI Technologies). Avec la présence de Didier Girard, [Patrick Chanezon](#), et Ludovic Perrier, le vainqueur du concours Android/SFR. La soirée se poursuivra avec des ateliers d'initiations à Android organisés avec le Paris Google Technology User Group ([Paris GTUG](#)). **Samedi 12 décembre** Le Bordeaux JUG organise dans l'après-midi à l'ENSEIRB un Dojo Randori TDD en se basant sur le projet Greenfoot. Il s'agit d'un projet qui permet d'appréhender le paradigme objet au travers d'une scène "graphique" dans laquelle évoluent des acteurs. Rendez-vous sur [le Google Groups du Bordeaux JUG](#) pour plus de détails et [sur le site du Bordeaux JUG](#). **Mardi 15 décembre** [NormandyJUG](#) Google Android, Google Web Toolkit et découverte de JSF. Des goodies eXo Platform et JetBrains à gagner. [Soirée Wicket et Grails](#) au LyonJUG, avec Cédric Exbrayat pour Grails et Nicolas André/Carl Azoury pour Wicket. Le Toulouse JUG qui a le plus beau site de tous les JUGs Français organise [une soirée Glassfish et OW2](#). Alexis Moussine-Pouchkine présentera Glassfish v3 et JEE6, puis Gaël Blondelle présentera le consortium OW2, partenaire d'eXo Platform par exemple, et aidé par Bull en France. **Jeudi 17 décembre** Le Poitou-Charentes JUG à Niort propose [une soirée EasyAnt et Ivy](#) avec la présence de Xavier Hanin, contributeur Ivy et aussi JUG Leader du [Bordeaux JUG](#). **Conclusion** Avec autant de JUG en France, nous pouvons être fier après 2 ans du parcours réalisé. Chaque mois, c'est entre 30 et 250 personnes par soirée, qui se retrouvent et parlent de la plateforme Java. Je crois qu'il n'y a pas encore de JUG à Strasbourg, mais je sais qu'un JUG devrait se monter du côté de Perpignan. Pour l'instant ce ne sont que des rumeurs non confirmées. La preuve que les Francophones font du bruit : [la page JugEvents est trustée par les Frenchies](#) !!! Il manque encore quelques JUGs [sur la carte mondiale des JUGs](#) (Bordeaux, Lyon, Marseille) donc les gars, allez voir comment faire pour ajouter votre JUG sur la carte de France. Il manque des JUGs au Maroc, en Tunisie, n'hésitez pas à vous lancer. Je sais que pas mal de mes lecteurs francophones sont aussi là bas, j'en profite pour lancer l'idée. Avec 14 JUG (Java User Group) et 1 JUG virtuel ([java.developpez.com](#)) en cette fin d'année 2009, la communauté Java est bien représentée en France. Continuez les gars ! On se donne rendez-vous en février au Paris JUG pour fêter nos 2 ans, au plaisir de vous voir tous à Paris début février.



[View Larger Map](#)

## Soirée JEE6 et Spring 3.0 au Paris JUG

Soirée réussie hier soir au Paris JUG (Paris Java User Group) avec 240 personnes, des gens qui sont restés dans le couloir ou assis par terre, une soirée animée avec un débat JEE 6 ou Spring 3, bref du lourd. Un merci à Tanguy Bayard qui a assuré la sono.



La soirée débute avec la musique de Rocky, animée par Nicolas de Loof du Breizh JUG. Antonio Goncalvès pour JEE6 et Michael Isvy pour Spring 3.0 montent sur scène. Le ton est donné. Allons-nous assister à un combat de coq ? Et bien pas du tout. Dès le départ les règles sont posées : ce soir nous allons comparer les 2 univers, presque les 2 communautés, et arriver à la conclusion que tout est complémentaire. JEE6 sortira le 10 décembre prochain (demain). Spring 3.0 a un petit peu de retard mais devrait aussi sortir très bientôt. Antonio nous fait remonter dans le temps. Java EE a 10 ans. Il y a 10 ans, cette spécification était pilotée par 8 éditeurs et 0 personne de la communauté open-source. Il y a 10 ans, la spécification était décidée puis les implémentations servaient à essayer les plâtres. Aujourd'hui la manière de spécifier un tel travail a changé. Java EE 6 et ses 28 spécifications propose une simplification avec pour la première fois du "Pruning". Si vous vous posez la question JEE5 ou JEE6, ma réponse est très claire : prenez tout de suite JEE6. Pas de bullshit genre "on attend que la dernière version soit un peu validée...". Java EE6 c'est la plateforme des années qui viennent. En 2009 la spécification est pilotée par 15 sociétés, 10 individus et 1 université. Tout le monde ne vote pas cependant, comme le dira Antonio plus tard. Les spécifications aujourd'hui dans la plateforme Java EE 6 sont le fruit du travail de la communauté open-source. Prenez JCDI 1.0, qui était avant WebBeans, qui lui-même vient du travail sur JBoss Seam initié dès 2005. Prenez JPA bien entendu, fruit du travail d'Hibernate. D'ailleurs l'implémentation de référence de JPA2 c'est [Eclipse Toplink](#). Antonio montre quelques

innovations de Java EE6, pour ceux qui n'auraient pas encore acheté [son livre](#). Je vous glisse une photo où l'on voit Michael Isvy avec un teeshirt Glassfish et Antonio avec un teeshirt "I love Spring". Oui les Français sont des gars marrants qui dépassent les clivages et arrivent à parler



ensemble, ce qu'ils font très bien pendant cette première heure.

Antonio rappelle que toutes les implémentations de référence de Java EE 6 sont open-sources. Très important. Cela permet aux éditeurs de travailler avec les mêmes briques techniques. Par exemple prenez le projet Mojarra qui est l'implémentation de référence (RI) de Java Server Faces 2.0 (JSF 2.0). Ce projet porté par SUN est utilisé par JBoss dans ses produits. Ou prenez Jersey JAX-RS (SUN) il est utilisé dans Oracle Weblogic, etc. En conclusion je me dis que le standard Java EE 6 c'est aussi enfin une réunion de famille où les éditeurs vont utiliser les mêmes projets open-source dans leurs produits, grâce à la normalisation. Est-ce que cela ne veut pas dire que ce serait la fin de la bataille sur certains sujets comme les ORMs, l'inversion de contrôle, la persistance, la plateforme web orientée composants ? La spécification fixe une bonne fois pour toutes les règles, afin que chacun puisse en décliner son implémentation. On reparlera des standards avec Emmanuel Bernard tout à l'heure. J'espère que vous me suivez dans mes explications. **Spring 3.0** Michael Isvy prend la parole à son tour, afin de nous présenter les dernières nouveautés de Spring 3. Tout d'abord, pour la première fois il faudra passer votre JVM de la version 1.4 à la version 5 au minimum afin de faire tourner Spring 3. Cela permet d'alléger le code et de réduire la dette technique. Je trouve cela cohérent, après tout Java 1.4 n'est plus supporté depuis quelques temps déjà, et il serait temps de se bouger un peu aussi. Spring dans cette version propose le support des architectures REST avec des annotations particulières, la gestion de la config avec Java Config, un langage d'expression pour améliorer les possibilités de configuration et donc, un passage à Java 5 obligatoirement. Spring 3.0 c'est la simplification, et après la présentation j'ai aussi le sentiment suivant : voilà, on a un superbe moteur que la communauté a adopté. Maintenant on nettoie et on simplifie, on bosse afin d'ajouter quelques fonctions intéressantes, mais l'essentiel est là. Spring arrive à un plateau de maturité. C'est un framework comme le rappelle Michael, pas une spécification. **Alors Java EE 6 ou Spring 3.0 ?** Cette question est stupide. Les 2 mon capitaine. Je vous encourage à regarder les nouveautés de Java EE 6, qui est plus léger et modulaire qu'avant. Les avantages de Java EE 6 c'est que nous parlons tout d'abord d'une norme, avec du code portable. Oui ne criez pas au loup avec vos soucis de portage qui marchent pas... moi le premier j'attends de voir pour y croire. Java EE 6 c'est plusieurs compagnies qui travaillent de concert pour proposer à la communauté des implémentations de références qui sont toutes open-source, chose que j'ignorais. Contrairement à notre vieux pote J2ee 1.4, il y a dès demain des implémentations qui tournent, et qui sont validées depuis des mois par la communauté. Prenez Glassfish v3, qui devrait être là autour de la mi-décembre, nous avons un serveur d'application qui implémente les spécifications de Java EE 6 prêt pour vos développements et vos mises en production. Les critiques de Java EE 6 : oui il sera difficile d'étendre le spectre couvert par Java EE 6 ou de faire évoluer une des spécifications à l'intérieur des specs. Ce sera je pense, le point le plus délicat à gérer. Tous les JCPs n'ont pas de liste de diffusions publiques. Il y a donc des discussions de corridor entre éditeurs, où même Antonio n'a pas de droit de regards. Enfin une spécification de 28 spécifications a besoin du support de la communauté pour marcher, et là tout est à faire. Concernant Spring 3.0, Michael Isvy passe aussi en revue les forces et les faiblesses de Spring 3.0. Tout d'abord Spring marche sur tout type d'environnement et sur la plupart des JVMs. Par rapport à Java EE, Spring ne propose pas que des annotations. 50% des utilisateurs de Spring préfèrent le format XML pour la configuration. Dans ma tête je me dis que pour ce qui est des EJB 3.1, il reste possible de faire du XML si vous le voulez. C'est vous qui voyez comme dirait Laspalès. Les points faibles de Spring aujourd'hui : clairement, le gros soucis c'est que le coeur du Framework repose sur 5 personnes, employées chez SpringSource, qui portent le framework. C'est un risque opérationnel assez fort, où tous les oeufs sont dans le même panier. Certes ce sont des gens très forts et très intelligents. Mais ils ne vont pas faire "que" du Spring toute leur vie. Il existe aussi qu'une seule implémentation de Spring, puisque ce n'est pas une norme. Là où Spring continue à jouer le rôle de locomotive dans l'univers Java EE, c'est aussi grâce à sa capacité à innover. La programmation par aspect, la gestion de la sécurité, le moteur de Batches, la gestion des exceptions... il reste encore pas mal d'idées intéressantes que Spring pourrait reverser à la communauté. **Mon analyse en carton qui vaut ce qu'elle vaut** Est-ce Spring n'aurait pas dû sauter dans le train Java EE 6 ? Prenez Hibernate par exemple : JBoss RedHat a reversé ses efforts à la communauté, en permettant à des personnes comme Emmanuel Bernard de contribuer aux spécifications, en finançant son travail au sein de l'expert group sur Beans Validation (JSR-303). JBoss RedHat s'est associé avec Oracle, SUN et IBM afin de ramer dans le même sens, ce qui permet aujourd'hui de bénéficier des retombées marketings de Java EE 6 à peu de frais. Ce qui permet de signer chez des grands comptes où le DSI préfère prendre un standard... Si Spring était devenue une implémentation de référence de l'inversion de contrôle par exemple, cela aurait certainement offert à la communauté le meilleur framework sur ce sujet. Cela ne s'est pas fait car il y a des hommes. Des gens intelligents qu'il faut coordonner. Prenez 10 bonhommes avec un égo et une forte personnalité... difficile à gérer. Regardons aussi la partie financière, car c'est le nerf de la

guerre. SpringSource à ma connaissance ne gagne pas beaucoup d'argent avec le framework en lui-même. Qui va acheter du support pour le châssis de sa voiture ? Personne. Tu signes avec le concessionnaire, pas avec l'industriel qui a construit ta voiture. Les pépites chez SpringSource/VMWare s'appellent Groovy et Hyperic. Ne cherchez pas plus loin, pour l'instant il y a aussi CloudFoundry, mais le coeur du framework ne fait pas vivre SpringSource. Oui il y a les formations, mais ce n'est pas du revenu récurrent comme une licence ou du support. Du côté des produits packagés et certifiés, SpringSource fait un gros effort pour proposer des solutions certifiées aux Entreprises. Spring te Server et les outils d'administrations par exemple, mais est-ce que cela porte ses fruits ? **Conclusion de la première partie** Un bon démarrage, avec un peu trop de Java EE6 par rapport à Spring, avec aussi l'envie de défendre celui qui est entrain de se faire bousculer par les copains, avec aussi une envie de dire à un moment que pour l'instant, Java EE 6 doit faire ses preuves pour passer dans la classe supérieure. Mais avec aussi une envie de vous dire : allez voir le film, allez voir ce que propose Java EE 6. C'est intéressant. Je vais juste dormir, car il est un peu 04h20 du matin, et je me réserve quelques neurones pour vous parler de la présentation d'Emmanuel Bernard, qui était juste... énorme (j'en rajoute un peu car je veux un teeshirt JBoss Redhat). A+ *TinyURL de cet article si vous voulez en faire une publicité monstrueuse sur Twitter : <http://tinyurl.com/yhhqf3j>*

## Xebia France fête ses 5 ans

Xebia France a fêté ce soir ses 5 ans. Luc Legardeur a invité toute son équipe (35 personnes) sur un bateau-restaurant le temps d'une soirée sur la Seine. Il a aussi invité quelques personnes de la communauté Java, ce qui montre aussi l'attachement de Xebia à notre communauté. C'était une soirée sympa, simple, sans fioriture, avec le plaisir de discuter avec des gens content de travailler ensemble, et loin de l'image que l'on peut s'en faire parfois. Xebia c'est une équipe de passionnés, qui proposent une bouffée d'oxygène sur Paris pour ceux qui en ont marre de bosser avec Machin-Technologies. Je m'adresse aux clients comme à ceux qui souhaitent changer de boîte. Pour moi, Xebia c'est mon voisin. Je parle de Blog. Il y a quelques années lorsque j'ai commencé à écrire sur Java sur un Blog, je me suis bien entendu intéressé à mes voisins. Xebia n'existait pas. Je lisais quelques blogs en Français dont celui de Benoît Moussaud, aujourd'hui associé chez Xebia. C'est grâce à lui qu'un jour j'ai traversé la rue des Blogs pour aller visiter le blog de Xebia. Novembre 2006, premiers articles, le ton était donné. Ce voisin là allait faire du bruit, et semblait prêt à faire "différemment". **Le secret de Xebia c'est qu'ils ne font rien comme les autres.** Vraiment. Prenez l'idée du Xebia Poker, du calendrier de l'Avent humoristique, du sponsoring au JUG : à chaque fois ils innove avec des idées sympas. Leur [blog](#) est aussi l'un des points d'entrées, c'est une référence dans le monde Java Francophone. Une SSII en principe, une boîte de gars qui font du service différemment en pratique. Ce qui en fait son intérêt, c'est sa différence. Je suis attaché à Xebia car je suis voisin, que je partage les mêmes valeurs, et qu'en même temps nous sommes différents. Prenez le logo, regardez ce grand X. Je ne peux m'empêcher de penser que l'une des barres de ce X est un mur. Certainement l'establishment et le qu'en dira-t-on. L'approche de Xebia, c'est cette deuxième barre qui tape la première à 90°. Une autre vision, une autre approche, quelque chose que les clients n'ont pas vu. Lorsque Luc Legardeur a démarré son aventure, je pense qu'il n'a pas cherché à faire autrement. Parce que lui-même a une approche différente, son entreprise est différente. Le secret de Xebia qui se cache derrière ce X c'est de ne pas faire comme les autres. C'est un sport où les gens de Xebia sont plutôt forts : il est très important de ne pas faire comme les autres. Vous vous souvenez de cette publicité pour le Loto, où l'on voit des voitures dans un embouteillage ? Et il y a une autre voiture dans l'autre sens qui passe à fond en klaxonnant et en se



moquant gentiment des autres...

Xebia c'est pareil.

J'ai travaillé avec Nicolas pendant un an. Et bien là encore, un consultant de Xebia plongé dans 15 autres consultants ne décolore pas au lavage. Il garde son identité, sa poigne et on sens bien aussi qu'il propose quelque chose de différent. N'importe quel patron de SSII aurait du mal à comprendre le principe de vos XKE. Chaque premier jeudi du mois, tous les consultants de Xebia ne travaillent pas chez le client final et se retrouvent pour une journée à Paris. Cette journée est consacrée à faire des présentations, des ateliers, et surtout à discuter avec les autres personnes de sa boîte. Comprenez-bien que chaque mois, Xebia ne facture pas une journée de l'ensemble de ces développeurs au client final. Et dites-vous qu'ils l'ont fait pendant la crise... Cela ne rapporte rien à première vue. Mais en discutant avec les gens de Xebia, qui ne sont des gars comme vous et moi finalement, on se rend compte que cela crée de la valeur humaine. Une vision humaniste est donc compatible avec une boîte de services... Mon Dieu ils ne font pas "comme les autres" et "ils ne font pas ce qui marche".... Chaque année, Xebia prend l'ensemble des salariés et les emmène 3 jours, le temps d'un week-end, pour fêter l'année et décompresser un peu. On est loin du séminaire de VRP à Angoulême au Campanile avec Canal+ dans la chambre... Cette année ils sont allés au Maroc. L'an passé en Tunisie, et avant au Portugal... Et c'est comme cela que les gens chez Xebia sont simplement heureux et restent. En prenant le temps de les connaître on voit qu'ils ne sont pas élitistes ou pédants. En fait ils sont proches des gens qui viennent au JUG. Et croyez-moi, à part la cravate pour certains même à 04h00 du matin, les autres sont juste "normaux". C'est important de le dire. Quoi d'autre ? J'ai aussi appris qu'ils essayent de se retrouver le deuxième mercredi du mois

pour prendre un pot le soir, de manière plus informelle. C'est le conclave. Marrant, et c'est plus léger que les XKE qui sont des mini-Devoxx... chaque mois. Alors qui peut bosser chez Xebia ? Des gens motivés, passionnés, pas forcément expert de la JSR-487b dont on se fiche pas mal. Des juniors sans problèmes, des seniors aussi. J'ai entendu ce message : il faut que l'on se montre tel que l'on est, à savoir une boîte cool de passionnés. Et je vous le confirme, c'est le cas. Il n'y a pas tromperie sur la marchandise. Enfin il est important de parler de l'aide de Xebia à la communauté et plus particulièrement du sponsoring du Paris JUG. Au nom de l'ensemble des geeks de Paris, merci à Xebia. Que ce soit avec les présentations, les questions de Cyrille qui sont devenues incontournables, votre présence pour aider financièrement le JUG sans rien demander en retour, merci les gars. Et merci à Luc qui fait une confiance à ses équipes. Je souhaite encore de longues années à Xebia et j'espère que la communauté Java aura l'occasion de faire votre connaissance, de voir que vous êtes juste des gars biens où tout le monde peut venir tenter l'aventure avec un CV et sa personnalité. Merci à vous pour proposer quelque chose de différent sur la place de Paris. On se donne rendez-vous dans 5 ans, car je compte bien rester votre voisin encore un bon bout de temps.

## Présentation sur les Standards par Emmanuel Bernard au Paris JUG



Il y a des présentations qui m'ont marqué : vraiment bonne, où vous avez l'impression de passer la tête par la fenêtre en roulant sur une route au bord de l'océan. D'autres présentations m'ont laissé des souvenirs mitigés et m'ont même attiré des commentaires du speaker ([Gradle for ever](#)). Et bien mardi soir au Paris JUG c'était une très bonne présentation. L'indicateur de qualité c'est le nombre de pages que je gratte sur mon bloc, la taille de mon sourire à la fin et l'applaudimètre des quelques 240 personnes dans la salle. Ce soir sur une échelle de 1 à 10 nous étions à 12. Suivez le guide. L'objet de la présentation d'Emmanuel Bernard est limpide : "**Les Standards : caca ou pas caca ?**". Avec un sujet décalé, exercice imposé par Antonio, il s'en est très bien sorti. Emmanuel pour ceux qui auraient été dans une grotte ces derniers mois, travaille chez JBoss RedHat, s'occupe du Podcast "[Les CastCodeurs](#)", blog [de temps en temps](#) et trouve aussi le temps d'être membre du Java Community Process en participant à la spécification JPA 2.0 et en étant Spec Lead de la [JSR-303](#) (Bean Validation). Avec tout cela, il trouve le temps de venir au Paris JUG, c'était bien sympa. Le format de la présentation tout d'abord est original. Il a utilisé les services de Prezi.com ([voir ici des exemples](#)) ce qui a donné une présentation originale et animée. Sa présentation est en ligne [à cette adresse](#) si vous voulez vous en faire une idée.

**Standard : caca ou pas caca ?** Regardons les standards dans le monde Java. Ce n'est pas une dette technique mais une dette d'amour pour certaines spécifications. La légende veut que les spécifications désignées par des comités d'experts soient lourdes et inadaptées au marché. A l'opposé, le monde de l'open-source est un univers merveilleux où tout se passe bien, les gens s'éclatent et s'entendent super bien pour faire de magnifiques logiciels. Et donc il y aurait une épaisseur en latex entre ces deux mondes (ça c'est moi qui le dit, pas Emmanuel). Il est vrai que si l'on regarde certaines perles du Java Community Process comme les EJB 1.x, les Webs Services WS-\* ou CORBA, pour ne citer que les plus connues, on peut parler de Caca avec un C majuscule. A contrario, il y a des spécifications qui dans le monde Java, nous font juste travailler chaque jour : JAX-RS, JTA pour les Transactions, JDBC pour les accès à la base, les Annotations ou (petit coup de pub) les Beans Validations. Quelque part, merci d'avoir spécifié il y a quelques années JTA ou JDBC : cela nous permet d'avancer et de ne plus se regarder le nombril pour accéder à une base de données. Retenez bien cette idée d'Emmanuel : les standards nous font avancer. La standardisation en dehors du monde Java nous fait gagner du temps. HTTP ou TCP/IP : grâce à cela nous parlons de l'avenir et du Cloud Computing, et nous ne nous battons plus pour échanger des fichiers. Grâce aux standards. Prenez la JVM. Moi le premier je suis encore étonné que peu de gens soient au courant que la JVM est 30 langages différents (Java, Groovy, Scala, même PHP). Les gens pensent encore que Java est un langage... C'est mignon. Non c'est une communauté et c'est une plateforme. Le langage on s'en tape le coquillard, excusez du peu. La JVM est une idée piquée à d'autres langages, mais sa standardisation nous permet de ne pas nous prendre la tête lorsqu'un programme écrit sur Mac, testé sur Windows, est mis en production sur Solaris. C'est simplement la réalité du terrain aujourd'hui. Alors l'OpenSource serait trop bon. Bien meilleur que les standards validés par des comités. Regardons un peu cela avec ce que nous utilisons aujourd'hui : Spring, Hibernate, DOM4J, Ant par exemple sont des projets open-sources que tout le monde utilise. Mais comme pour les standards (EJB 1.x et Corba) il y a aussi de belles perles comme Maven 1.x, Struts 1 ou CGLibs. La communauté open-source Java c'est 230 000 projets sur SourceForge.net : dans le lot il y a Hibernate, mais il y a aussi des machins fumants qu'il ne faut pas utiliser. Et parfois il est trop tard pour se rendre compte qu'un projet open-source pose de gros problèmes (GWT-Ext par exemple). L'univers de la standardisation c'est Dallas d'après Emmanuel. Et vas-y que je me fache avec toi, et que je lance mon standard pour l'embêter, et que je ne vote pas pour ta spec, etc. Mais le monde de l'open-source : c'est pareil ! Emmanuel explique que c'est même encore plus compliqué que dans les comités de standardisations : faire passer une idée dans le monde open-source sur un projet, c'est parfois impossible. Un excellent développeur de base aura du mal à faire sa place dans un projet établi. Donc les 2 se valent. Emmanuel rappelle le phénomène "The ServerSide" il y a quelques années. C'est un peu la presse People du monde Java. Chacun est libre d'écrire son article et de le publier sur ce site. La modération est plutôt légère et cela entraîne parfois des débats passionnés mais **pas** passionnants, entre grosses têtes de notre communauté. Ecouter l'autre est dur, accepter son point de vue, encore plus. Les humains sont comme cela.

Emmanuel change de chapitre et attaque maintenant le versant Nord de l'Everest. Java EE ce n'est pas JBoss Application Server ou GlassFish. Ceux qui comparent Spring à Java EE sont stupides. Il faut comparer des produits. Il faut donc comparer [Glassfish](#) à [Spring dm Server](#). Emmanuel dit cela car un standard est bien mieux défini qu'un produit. En effet, il doit aller à fond dans les détails et doit donc tout spécifier. S'il reste une zone de flou, c'est un risque pour que chaque éditeur interprète cela à sa façon. C'est aussi pour cela qu'il faut proposer un kit de validation ([Technology Compatibility Kit](#)). Les standards peuvent aussi faire preuve d'innovation. C'est surtout finalement en terme d'aggrégation que les standards innovent, si l'on prend le cas de Java EE 6 et des 28 spécifications attachées. Mariage difficile mais pas impossible. Si les gens s'écoutent, c'est possible. Java EE 6 en est le meilleur exemple. Arrive la phrase qu'il fallait retenir de la présentation : "**Le standard est défini pour passer à autre chose**" Que vous inspire cette phrase mes amis ? Je me dis juste à cet instant précis : *dommage que Spring n'ait pas suivi quelques standards de Java EE 6, du coup ils vont mourir*. C'est méchant non ? Mais si SpringSource se sécurisait en effectuant un effort pour que quelques modules de leur architecture respectent les standards ? Cela permettrait alors à Spring de passer à autre chose, et de continuer à dynamiter la communauté avec leur formidable énergie. Lorsque l'on regarde de très bons produits comme Spring Batch ou Spring Security, on se dit qu'il serait intéressant de pousser ces projets afin qu'ils deviennent des standards dans Java EE 7. Vous ne pensez pas ? Car Java EE 6 c'est une standardisation entre autre de l'inversion de contrôle et de l'injection de dépendances : cela va tuer Spring en opposant une normalisation, reprise par les différents acteurs du marché. Un peu comme lorsque nous opposions X25 et TCP/IP. A un moment donné, le standard a bouffé son voisin. Je sais que c'est un peu exagéré, et que vous serez nombreux à commenter. J'essaye de voir avec vous si l'hypothèse suivante est plausible : en ne suivant pas les standards, SpringSource prend le risque qu'une partie de ses produits soient sur des niches un jour. A contrario, et si Java EE 6 faisait un bide ? Est-ce que le crédit confiance est assez élevé pour que la communauté adopte Java EE 6 ? Après tout, avec un Tomcat, du Spring et de l'Hibernate, on couvre aujourd'hui les besoins des applications Web simples. C'est pour cette raison que Java EE 6 propose un container EJB léger, embarqué. A un moment donné on a envie de dire : enfin ! **Les standards vus par le Développeur, le DSI et l'Editeur** Emmanuel nous propose maintenant de regarder les standards selon différents points de vue. Pour le développeur, un standard c'est une API validée, c'est donc un investissement intéressant s'il doit se former. C'est aussi le moyen d'ajouter une ligne sur son CV, et de se dire que Google est ton ami, lorsque tu seras bloqué sur un point délicat. Emmanuel évoque Rails et Ruby. Bien que ce soient de bons langages et de belles plateformes, la taille de la communauté n'est pas aussi importante que celle de Java. Du point de vue du DSI, celui-ci doit gérer des groupes de développeurs. Il a besoin de repères, et de s'assurer qu'il peut trouver des compétences demandées pour développer ses logiciels. En prenant un choix plus large, il sécurise ses investissements. Par ailleurs, savoir que l'on cherche un développeur Java/J2EE aujourd'hui, un développeur Java EE 6 demain, permet de réduire les coûts et d'éviter de payer trop cher ce qui serait trop rare (développeurs, consultants, indépendants). Les projets Open-Source sont de qualités inégales. Le DSI peut se poser la question de savoir s'il y a une société derrière, s'il peut avoir du support, si les développeurs sont aidés et s'il y a quelqu'un qui pilote les développements. Prenez Linux (je vais me faire pleins d'amis ce soir). Il y a tellement de gens qui veulent contribuer à ce projet, que finalement aujourd'hui il existe tout un tas de distributions, et que chacun veut tirer la couverture de son côté. Le DSI a été content de trouver sur son chemin un visage professionnel (là on pense un peu à RedHat) pour pouvoir taper dessus en cas de soucis, et surtout, sécuriser ses investissements. Certains projets open-source sont inquiétants, comme MySQL, aidé par SUN, qui lui-même a été racheté par Oracle... Oracle affirme qu'ils garderont MySQL, mais j'attends de voir avant d'y croire. Enfin Emmanuel passe au point de vue des Editeurs de logiciels. Il y a plusieurs approches, chaque éditeur a une politique différente. Ils peuvent - Réutilise les standards - Implémente les standards - Innover et défini de nouveaux standards. Certains éditeurs ne veulent pas des standards. Cela les protège, en ayant "de l'innovation". Quelque part, refuser d'utiliser des standards comme Spring ou Hibernate, c'est complètement **has-been** dans le monde Java. C'est mon opinion, pas ce que dit Emmanuel. Il passe rapidement ceux qui ignorent les standards afin de s'attacher aux suiveurs, ceux qui reprennent les standards sans réellement innover. Spring / les frameworks Webs / Hibernate : on se met par dessus les standards (jdbc pour hibernate) et on les réutilise sans trop forcer. Les frameworks Webs : on utilise JSP et Servlets mais on met un peu de rouge à lèvres pour vous exciter : Tapestry / Wicket. Ces projets open-source ont une raison d'exister tant que le standard n'existe pas. Hibernate aujourd'hui est masqué par JPA avec son implémentation de référence qui est Eclipse TopLink (Oracle a offert TopLink à la communauté Eclipse). Il y a ensuite les éditeurs qui innovent et implémentent les standards. JBoss a débuté en proposant une implémentation open-source de J2EE, ce qui en a fait son succès. Les éditeurs s'assurent une part du gâteau en implémentant un standard, afin de partager les retombées marketings et aussi afin de suivre les propositions normalisées. Enfin il y a les éditeurs qui innovent et tentent de définir les standards. C'est beaucoup plus cher, beaucoup plus risqué. L'objectif est de s'attaquer à un monopole, et l'exercice consiste à synchroniser son énergie pour sortir un standard, afin d'attaquer un monopole sur le marché. Est-ce qu'Emmanuel pense au framework Spring à cet instant ? Quoiqu'il en soit, pour l'éditeur le fait que son projet devienne un jour un standard le force à être encore meilleur. Hibernate n'étant plus le seul sur la place, normalisé par JPA2, il faut maintenant proposer d'autres innovations. **Et l'open-source ? Il y a des standards de fait après tout** Prenons le framework Spring, pour moi c'est un standard de fait. Les coûts de marketing aussi sont réduits. Puisque vous pouvez tester librement, c'est vous qui en faite la publicité. Concernant l'assurance qualité (QA) les projets open-sources sont très testés et corrigés aussi rapidement. Un projet open-source qui suit un standard normalisé, c'est certainement le duo gagnant je pense. **En conclusion** Pour terminer, Emmanuel explique qu'un bon standard doit être stable, extensible, ouvert sur le futur et surtout adopté par la communauté. L'objectif de Java EE 6 est de proposer un socle normalisé, afin que maintenant la communauté mette son énergie sur de nouveaux problèmes. Et là, mesdames et messieurs, il a raison. Bon sang quel temps perdu pendant 3 ans, nous devons maintenant passer à autre chose, et commencer à réfléchir à des serveurs de 4ème génération. Bon, Emmanuel, tu reviens quand au Paris JUG ?

## Je bosse dans une startup



Depuis quelques jours j'explique un peu ce que je fais. Après 15 mois en tant que consultant indépendant, je débute une nouvelle aventure depuis quelques jours. Il y a quelques mois, j'ai été contacté par des amis pour rejoindre un projet Web en tant que responsable technique. Quelques péripéties plus tard, je suis maintenant installé dans nos bureaux énormes de 560m<sup>2</sup> avec écran 23", télé 16/9, une méga Cafétaria, un restaurant gratuit, un MacBookPro pour chaque personne, une salle de sport, une carte Business, des tickets restaurants de 20 EUR, bref Google à côté, c'est des petits joueurs. D'ailleurs voici nos énormes locaux comme vous pouvez le



constater sur ces photos qui ne mentent pas:



(Nous sommes dans un incubateur et c'est très sympa en vrai) Bon sur ce que je fais exactement, je vais vous en parler un peu mais restez discret, je ne tiens pas à ce que tout le monde le sache (*j'adore écrire ce genre de phrase à tiroir*). C'est du Web grand public, un site avec des vrais utilisateurs, un truc assez sympa. Je m'arrête là pour la partie contenu puisque le reste est confidentiel pour l'instant. Parlons techno, car je vais travailler **avec vous** dans les semaines qui viennent. Au sommaire il y a du Java, du Grails, du Groovy, du Drupal (PHP) et du Cloud pour l'hébergement. Je me suis dit qu'il serait sympa de partager avec vous les quelques idées, les pistes, et voir un peu aussi comment vous réagissez. Nous sommes plusieurs sur le projet, mais je trouve qu'il serait intéressant d'écrire quelques articles et de vous faire partager nos idées, afin d'écouter vos commentaires. Et enfin comme je rassure aussi les personnes avec lesquelles je travaille, je serai muet comme une tombe sur la partie métier, qui est la plus stratégique. On va donc parler technologie, nous partagerons nos idées avec vous, et vous pourrez suivre l'aventure d'une jeune pousse, avec je l'espère pas mal d'articles à écrire. Voilà, à vous le micro maintenant.

## Vectorized version of Juggy, the Java User Group logo

I spent a few hours to vectorize the official Java User Group logo, Juggy. This is the Paris Java User Group logo, but it won't be too difficult to adapt-it to your local java user group. The original logo was created by Bruno Ferreira de Souza, President of SOUJava, the official Brazilian Java User Group. Here is a picture of Bruno and I at the last Devvoxx 2009, with the real Juggy :



This work is of course open-source, released under [the Creative Commons Attribution-Share Alike 2.0 France license](#). Here is the



Cette création est mise à disposition selon le Contrat Paternité-Partage des Conditions Initiales à l'identique 2.0 France disponible en ligne  
<http://creativecommons.org/licenses/by-sa/2.0/fr/>

Author: Nicolas Martignole  
 nicolas@martignole.net



original work:  
 here : [juggy.parisjug.version8.english.ai](http://juggy.parisjug.version8.english.ai) If you need another format, please contact-me. Let us know if you create your own logo based on Juggy. The Christmas version:



## Framework Play! à découvrir le jeudi 17 décembre



J'ai découvert le framework Play! à Devvix 2009 et j'avoue avoir été assez impressionné par les concepts et les idées. C'est du Java, du pur Java. C'est un framework Web, avec des idées novatrices dont je vais vous parler dans quelques minutes. L'objet de ce post c'est de vous faire part d'un événement **ce jeudi 17 décembre à Paris**. L'équipe des contributeurs du framework organise une soirée découverte de 18h30 à 21h00 chez Zenexity. Les détails et l'inscription sont [sur cette page](#). Play! est un framework Java créé par [Guillaume Bort](#), avec des références et des projets [en production](#), qui mérite le détour. La vidéo de la page d'accueil montre qu'il faut à peine 1 minute pour passer sa ceinture blanche sur Play! (le classique HelloWorld) et moins de 5 minutes pour comprendre les principes. Play! est semblable à Grails, mais c'est du pur Java. Clairement orienté efficacité, il plie en 4 les projets lourds. Apparenté à Rails dans l'idée, il y a bien entendu du redéploiement à chaud sans recompilation. Play! est intéressant aussi car il y a beaucoup d'idées vraiment nouvelles. C'est un framework orienté développeur web. Nicolas Leroux avait demandé à Devvix 2009 aux gens dans la salle s'ils se sentaient plutôt "Web Developer" ou "Enterprise Developer". Au départ, la majorité des gens lèvent la main et s'affirment "Enterprise Developer". Il explique ensuite que sa définition du Web Developer est plus "toute personne devant faire une application sans état du Web". Et là... presque tout le monde lève la main. C'est un fait : nous faisons tous du développement Web. Mais finalement, nous ne faisons pas tous du développement Entreprise. La définition des applications Webs est simple : proposer un service qui tourne sur un navigateur et laisser le soin aux clients riches de proposer une application. Play! comme Grails se place sur le plan de la création des applications Webs. Lancé par Guillaume Bort en 2007, la version 1.0 est disponible depuis le 19 octobre. Avec déjà des déploiements et un support pour ceux qui veulent être aidé, Play! est donc mature. je dis cela pour les frigidés qui attendent toujours la version 1.1 d'un produit avant de commencer à regarder. Ce framework Web fait partie des frameworks sans état comme Grails ou Django. Contrairement à Wicket, JSF ou GWT qui savent ce que vous voyez à un instant T en maintenant une session en mémoire sur le serveur. On est là plus sur du Web classique, et donc ceux qui recherchent un framework orienté composant seront déçus. Mais en fait je vais vous dire un truc : pour un site Web il vaut mieux faire le choix de ce type de framework qui suivent ce que nous propose pour l'instant le protocole HTTP : l'absence d'état. Play! est donc un framework clairement orienté REST, qui s'adapte aux standards du Web et qui fait ce que les autres font très bien : cookie, gestion du bouton back, authentification, etc. A titre personnel, puisque j'ai fait le choix de Grails pour mon projet, je vous

avoue que j'y crois plus pour faire des sites Webs à fort trafic, plutôt que JSF ou Wicket. Alors là vous vous dites : "ouais... bon et alors ?". Il n'y a pas de WAR. Lorsque vous développez, devinez sur quoi vous allez travailler ? Vos fichiers Java. Et il n'y a pas de compilation non plus. Ah ça c'est marrant non ? Et il n'y a donc pas de packaging. J'ai une pensée émue pour ceux qui compilent et partent prendre un café... Avec Play! vous n'aurez plus le temps. Complètement codé par des développeurs, il y a encore d'autres trucs assez marrant, bien plus puissant qu'avec Grails. En premier lieu je pense à la gestion des erreurs. Lorsque vous saisissez une erreur, et que vous tentez d'exécuter votre page, Play! vous expose des erreurs très claires, et vous propose même une solution :  Il y a beaucoup de fonctions : la sécurité, un moteur CRUD, la validation, la gestion des jobs asynchrones, de l'I18N, les Fixtures, des services REST, OpenId, l'envoi d'emails, des extensions à JPA, bref c'est assez complet. Play! propose plusieurs scopes comme Seam ou Grails : Application/Session/Flash et Request. Il y a aussi un moteur de binding simple qui permet de déclarer le nom d'une méthode à appeler selon une URL:

```
GET /clients/{id} Clients.show
```

Récupérer les paramètres de la Query String est aussi un jeu d'enfant. Si j'appelle GET /clients/222?date=08/01/08&page=2

```
public static void show(Date date, Integer page) {
    List articles = Articles.fromArchive(date, page);
    render(articles);
}
```

**Les points qui vont vous étonner** Si vous êtes habitués à Wicket, vous êtes déjà habitué à étendre certaines classes de Wicket pour que votre code fonctionne. Play! aussi vous offre une partie magique, si en retour vous étendez les classes standards. Grails se base sur le répertoire où vous stocker votre code, et sur le fait que votre class se termine par "-Controller" pour la transformer en Controller par exemple. J'ai bien aimé l'approche de Play! qui propose de casser l'encapsulation, mettre en public vos propriétés, et donc ne pas écrire des getter/setter qui ne servent à rien. Après tout, à quoi bon encapsuler sans réelle raison ? Comme Wicket, Play! propose d'ajouter à des pages HTML standards des tags afin de rendre "playable" votre page. On retrouve la même idée chez Grails. Concernant le modèle, Play! ajoute à JPA quelques options sympathiques. Il est possible mais pas obligatoire de faire hériter votre classe de base de la classe Model de Play! comme sur cet exemple ([tiré de la doc de Play!](#)):

```
package models;

import java.util.*;
import javax.persistence.*;

import play.db.jpa.*;

@Entity
public class User extends Model {

    public String email;
    public String password;
    public String fullname;
    public boolean isAdmin;

    public User(String email, String password, String fullname) {
        this.email = email;
        this.password = password;
        this.fullname = fullname;
    }
}
```

Cela simplifie la gestion de JPA. Est-ce que vous avez remarqué que les propriétés sont publiques ? Play! se chargera de générer les getters/setters pour vous, afin de conserver l'encapsulation plus tard. Mais à ce moment de l'écriture du code, c'est inutile. Il y a encore quelques concepts que je n'ai pas compris comme l'idée du "bazaar", un gestionnaire de code source où vous reversez vos modifications. Je ne vous ai pas aussi parlé de toute la partie test unitaire et test fonctionnel, très complète. Enfin pour vous laisser sur une grosse note positive : Play! est fait de telle manière qu'une modification dans votre code Java peut être testé tout de suite, sans recompiler, packager et déployer votre application. Comme Grails dont je suis amoureux, Play! est un outil méga-productif qui éclate sans soucis les frameworks à état que sont GWT, JSF et Wicket. Play comme Grails se base sur ce qui marche pour faire du Web : REST et l'absence d'état en mémoire. A part la session de l'utilisateur au sens HTTP, il n'y a pas de mastodonte en mémoire, qui représente ce que voit l'utilisateur. Bon je m'arrête là, j'en aurai pour une heure à vous détailler ces idées, mais c'est l'une des raisons pour lesquelles j'ai pris Grails pour mon projet. Bref si vous voulez découvrir un framework différent, rendez-vous [jeudi 17 décembre](#). Je vous conseille aussi de lire les articles de Guillaume Bort, le fondateur du framework Play! car ses articles sont vraiment intéressants ([essayez celui-ci](#))

## Grails en quelques mots



Je travaille depuis quelques mois avec le framework Web Grails. Cela fait quelques semaines que je m'en sers presque 8 heures par jour. Je vous propose un petit tour du propriétaire. [Grails](#) est un framework Web basé sur le langage Groovy clairement orienté productivité. J'ai ramené 4 livres de Devvix sur Groovy et Grails, et le meilleur ouvrage pour rentrer dedans reste "[Grails in Action](#)" de Glen Smith et Peter Ledbrook. 470 pages plus tard vous serez en mesure de construire une application industrielle, et il vous manquera un peu de détails sur Groovy, facilement couvert par exemple avec "[Groovy Recipes](#)" de Scott Davis. Grails s'installe en 5 minutes, se découvre en 30 minutes et se maîtrise en une semaine. Je me dis que si mon aventure de Startup ne marche pas, je pourrai toujours écrire un livre sur le sujet en Français. C'est vous dire si j'ai envie de vous en parler. **Introduction à Grails** Grails m'est tombé dessus car j'avais besoin d'une solution pour démarrer rapidement, valider rapidement (et éventuellement me planter rapidement). Je me surprends à tester des bouts de code, à écrire des choses comme `Books.findByAuthor('Guillaume Laforge')` et à voir que Grails génère le code de ce finder pour moi... Envie d'Ajax ? je transforme mes tags form en remoteForm et ça marche. Envie de faire un formulaire avec 4 écrans ? Spring Webflow est intégré, le flow est écrit en Groovy, rien de plus simple... Et c'est comme cela depuis 3 semaines... Je n'arrête pas de m'étonner de la maturité et de la puissance de ce framework Web. **Avec ou sans état ?** Grails comme Play! se classe dans la catégorie des frameworks Webs sans état. Il y a bien entendu une session, des scopes pour vous aider à travailler. Mais là où Wicket et JSF sont orientés composants, Grails est clairement orienté Web / HTTP. Il n'y a pas la notion d'Application et il n'y a pas la notion de composants comme en Wicket ou JSF2 par exemple. Nous pouvons donc dire que Grails est adapté pour la réalisation de projets plutôt Web, et il n'est pas forcément le plus adapté pour réaliser une application riche. Il y a pourtant des applications basées sur du Flex pour la couche de présentation qui travaillent avec une partie serveur en Grails pour les services. Cette alchimie fonctionne donc aussi. **Fan des conventions au lieu de la configuration** Grails tire ses origines du framework Rails du monde Ruby. Mais il utilise le langage Groovy qui tourne sur la JVM. Ses cousins ? JRuby on Rails, Clojure, le framework Lift du monde Scala, et Play! du monde Java. C'est un framework Web clairement orienté productivité et simplicité. Pour moi, il fait parti des frameworks webs de 4ème génération. Il y a d'abord eu les JSPs, les Servlets et les Applets. Puis ensuite Struts avec les premiers frameworks MVC de type 2. Ensuite l'apparition des RIAs ([Rich Internet Applications](#)) avec GWT, JSF, Wicket. Et donc maintenant je crois dur comme fer à ces nouveaux frameworks Webs, qui reviennent sur les bases du protocole HTTP pour offrir des performances et de la simplicité. Grails est sorti en 2006. Il y a des sites en production comme [Businessworld](#), [Sky.com](#) ou encore [Escapeer](#). La page [Testimonials](#) donne une idée de quelques projets réalisés avec Grails et Groovy. Groovy est le point fort de Grails. Je vous parlais de trouver des livres quelques lignes plus hauts. Vous pouvez aussi écrire des bouts de code comme par exemple: `Book.findAllByDatePublishedGreaterThanAndTitleLike(myDate, "Grails par le Touilleur Express")` Grails sera capable de générer le code Hibernate, Spring et Groovy afin de trouver tous les livres dont la date de publication est postérieure à une date, et dont le titre ressemble à ce que je précise... Lorsque je vous dis que vous allez gagner du temps, vous commencez à me croire ? **Grails utilise des standards** Grails utilise Groovy, et ensuite le couple Hibernate/Spring de base. Il est assez facile d'ajouter de nouvelles bibliothèques grâce à des plugins. Dans mon projet par exemple j'utilise Drools. Je n'ai eu besoin que de 10 minutes pour créer un contrôleur simple et une règle Drools pour tester une idée. **Les idées de Grails** Ce qui suit est tiré du livre "Grails in Action". Il y a 7 idées clés à comprendre pour adopter Grails: - Convention au lieu de la Configuration - Philosophie Agile - Des fondations solides - Scaffolding et moteur de template - Intégration avec Java - Une communauté et des plugins - La Productivité améliore la qualité et réduit le risque d'erreur Concernant **l'aspect convention au lieu de configuration**, cela revient à vous dire que pour gagner du temps, il faut suivre ce que dit le framework. J'ai juste un mot à dire aux gens qui font du Maven2, mais qui ne veulent pas ranger leurs fichiers dans `src/main/java`, qui désactivent les tests, qui veulent lancer des scripts Ant avec Maven : **arrêtez Maven**. Vous n'avez rien compris à sa philosophie, vous pleurez car "Mavenu il rameuu" alors que si vous aviez pris le soin de lire l'excellent livre "Apache Maven", vous ne seriez pas comme un gosse avec le pantalon mouillé. Si vous commencez à bidouiller l'outil, c'est que vous n'avez pas lu la documentation, que vous n'avez pas écouté ce que vous disent les développeurs. A ce titre, je trouve que les livres que j'ai lu m'ont énormément aidés par rapport à la doc de Grails, qui ne distille pas assez la philosophie, à mon sens. Donc si vous voulez me suivre : achetez des livres sur Grails. Grails est simple : une entité Survey est toujours stockée dans le répertoire domain. Cela permet à Grails de gérer l'objet via Hibernate pour vous. Un contrôleur est nommé SurveyController, et il est placé lui-aussi dans un endroit attendu par Grails. Si vous ajoutez une méthode "completeSurvey", celle-ci est alors accessible automatiquement via une url comme `/survey/completeSurvey`. Si vous créez une nouvelle entité "Answer", Grails se charge de créer la table en base de données. Vous n'avez pas généré de fichiers pour tout ce qui est CRUD ? Grails le fait tout seul. Pour l'affichage des pages, il suffit de créer un fichier `/views/survey/completeSurvey.gsp` et il sera alors automatiquement chargé... C'est bien de la Convention par dessus la Configuration et surtout pas de la Convention "au lieu de" la Configuration. Si vous voulez utiliser vos fichiers de mapping Hibernate : oui c'est possible. Mais par défaut : vous n'en n'avez pas besoin. Tout ce qu'il est possible de faire de manière répétitive et générique, c'est offert. Et comme vous aurez besoin de mettre le nez dedans, vous pouvez aussi le faire. J'en suis à 16 classes pour mon domaine, à 2700 lignes de codes, 19 contrôleurs, et en 3 semaines j'ai déjà une application qui tourne... Je vous assure que j'ai montré tout cela dans une vidéo très bientôt. **Grails est adapté au développement Agile**. En fait je pense même que l'on peut dire que c'est un framework Web Agile. Les tests unitaires comme les tests fonctionnels sont faciles à écrire. Le fait de pouvoir créer un War, c'est aussi très pratique. Je vous explique cela : j'ai

installé un Tomcat sur une machine, avec une base MySQL. Sur mon Mac, en ligne de commande il suffit que je tappe "grails war production" pour que cela me construise un fichier WAR, qu'il me suffit de déployer ensuite sur Tomcat pour la version en production. A aucuns moments je n'ai eu besoin de mettre le nez dans un quelconque fichier web.xml... Au quotidien je ne construis pas de WAR. Je tappe simplement "grails run-app" et un serveur Jetty démarre, intégré à Grails. Lorsque je code, je n'arrête presque pas le serveur. En fait, lorsque vous changez un bout de page GSP, celle-ci se rafraîchit tout en gardant votre session. Si vous ajoutez une nouvelle méthode dans votre contrôleur, l'url est disponible et vous pouvez continuer à coder. Il n'y a qu'en cas de plantages que je relance parfois le serveur, lorsque ma session est pourrie. Je trouve que la remontée d'erreur pourrait encore faire des progrès par rapport à Play! mais c'est déjà pas mal.

**Grails est basé sur des fondations solides.** Spring et Hibernate. Spring 3 depuis la toute dernière version 1.2.RC2 que je teste en ce moment. En fait vous ne voyez pas Hibernate et Spring, mais vous savez que vous pouvez aller voir si nécessaire. Le moteur de scheduling ? C'est Quartz. Le moteur de recherche ? Lucene et Compass pour les entités. Le moteur de construction ? SiteMesh... Bref il n'y a pas d'innovations. Nous parlons bien de techniques que vous connaissez déjà. Imaginez simplement ceci : écrire moins de code, travailler plus vite. **Le moteur de scaffolding et de templating.** J'adore Spring MVC. Je m'en suis servi l'été dernier pour mon projet sur Google App Engine (<http://touilleur.appspot.com>), projet que je ne finirai pas faute de temps. Grails est assez radical par rapport à Spring MVC. Là où vous auriez quelques beans annotés, des fichiers de configuration, un peu d'Hibernate et une configuration avec la base, Grails fait la même chose en quelques minutes. `grails create-app myapp ... grails run-app` Ouvrez votre navigateur : `http://localhost:8080/myapp`, c'est prêt. Ensuite si vous voulez ajouter un nouveau contrôleur : `grails create-controller MonContrôleur` C'est tout ! Lorsque vous créez une nouvelle entité, un nouvel objet du domaine, il est facile aussi pour Grails de vous générez automatiquement tout ce qui est CRUD. C'est ce que l'on appelle le scaffolding. Ce n'est pas destiné à remplacer votre application. En fait voici comment je travaille : je génère d'abord mes entités avec des idées à tester. Ensuite je crée un contrôleur pour cette entité en mode "scaffold". Cela me permet de tester mon objet, de valider les relations one-to-one, one-to-many, many-to-many et de m'assurer que tout marche. Enfin, je refais un vrai contrôleur pour mon application finale, et je ne touche pas à ce que Grails m'a généré. Je recopie quelques petits bouts de code pour gagner du temps, mais c'est là que je code vraiment mon application. Cela me fait gagner un temps fou. **L'intégration avec Java** est là pour rassurer les bons pères de famille. Il y a un répertoire Java. Vous pouvez y placer ce que vous voulez. Pour peu qu'une librairie de votre ERP soit en Java, et qu'il soit nécessaire d'écrire un peu de code orienté Service, Grails vous aménage un espace dédié pour le faire, et se débrouille pour injecter vos services Java dans les contrôleurs Grails... C'est très pratique. Je m'en suis servi pour une partie avec du Drools, que j'ai migré vers du Groovy car finalement, c'est plus simple. **Grails a une communauté.** C'est bête, mais lorsque le lundi matin vous êtes seul dans votre bureau de 7m2, croyez-moi vous êtes super content de voir qu'il y a pleins de gens qui se servent de Grails sur Internet. Google est ton ami. La documentation est bonne. Le produit vit, et la liste de diffusion users est très active. J'ai appris beaucoup en lisant les échanges ces dernières semaines. Enfin côté livre je vous le disais tout à l'heure, il y a de très bons livres en Anglais. Si j'avais le temps et le courage, je me lancerai bien dans l'écriture d'un livre en Français. Mais je suis trop bavard... Pas possible pour moi. Enfin pour terminer, le 7e point noté par les auteurs de Grails in Action, c'est l'aspect de productivité et la possibilité de valider ou non ses idées très rapidement. Pour notre projet qui sera en ligne l'été prochain, je me remercie chaque jour d'avoir pris Grails pour gagner du temps, faire des démonstrations qui marchent chaque semaine, jeter sans souci du code écrit en 3 jours au lieu de 2 semaines, bref je peux dormir tranquillement la nuit. Est-ce que tu comprends que je peux virer des idées foireuses car elles ne m'ont coûtés que 2 jours au lieu de 2 semaines ? Un vrai projet de Startup c'est beaucoup d'essais. Personne ne vous dira si ce que vous faites c'est bien ou pas. Il faut avoir cette possibilité de changer de cap très rapidement, encore une fois dans le cadre d'une Startup pour moi.

**Conclusion** Pour terminer, je vous laisse ici ce soir. Je vous montrerai un peu de Grails, du vrai code, pas un "Hello World", plutôt une application écrite en quelques heures, afin de vous faire comprendre ce qui a retenu mon attention. Je pense aussi que nous, **développeurs Javas**, devons nous remettre en question d'urgence. En voyant des projets sur PHP ou Ruby on Rails, j'ai un peu la gueule de bois avec mon Java. Je pense que pour le développement d'applications Webs, il faut revoir notre manière de penser, notre manière de travailler. J'ai même du mal à croire au succès de Struts hier, de Wicket/Tapestry/JSF en ce moment... Comment pouvons-nous rouler des mécaniques alors que d'autres langages sont bien meilleurs pour faire du développement webs ? Que d'autres frameworks sont bien mieux pensés et bien moins prétentieux que ce que nous vendons depuis 3-4 ans à nos clients ? Comment avons-nous pu vendre des développements avec du Struts, des JSPs à des clients finaux ? Comment ? Imaginez-vous entraîné d'expliquer aux clients que les développeurs sur certains projets doivent construire un WAR, arrêter et relancer leur serveur d'application avant de voir leur résultat... alors que c'est inutile avec Grails (et d'autres)... Et encore, je ne vous ai même pas encore dit que j'étais aussi tombé amoureux de Groovy... J'en garde pour la prochaine fois.

## Grails : l'application Zencontact de Zenika

[Le 10 mars 2009](#), Zenika a présenté Apache Wicket au Paris JUG. Carl Azoury et Nicolas André ont construit en une heure une application de gestion simple, avec des fonctionnalités avancées en Ajax. L'objet de l'article c'est de vous montrer, vidéo à l'appui, comment construire la même application avec Grails. Le gros plus : je vais réellement persister mes Contacts en base, sans efforts supplémentaires, là où la version Wicket s'arrête à la partie Web, sans la partie persistance de l'application. Le cahier des charges proposé par Zenika est très clair, vous pouvez retrouver [à cette adresse](#) la liste détaillée des fonctions à coder. **Le résultat final** Voici la liste du cahier des charges. Pour chaque élément j'ai mis en italique les points inutiles pour Grails et en rouge les fonctions que je n'ai pas développé. \* La navigation entre pages \* L'organisation d'une structure commune des pages (Type Tiles / Sitemesh) \* **La désactivation du lien correspondant à la page courante** \* L'édition d'un contact \* La création d'un contact \* Lister les contacts \* L'ajout rapide

d'un contact dans une liste \* Le refresh de la date courante via appels Ajax \* L'édition "in place" d'un libellé sans passer par un écran d'édition \* L'ajout de la validation (Nom et prénom obligatoires, contrôle du format date, contrôle du type email) \* Utilisation d'un date picker \* Synchronisation du format du DatePicker avec le format utilisé par le convertisseur \* Ajout des mêmes contrôles de validation côté client, en Javascript \* Gestion de la problématique du refresh afin d'éviter la double soumission \* **Non duplication du code du formulaire... le même composant doit être utilisé pour la page d'édition de contacts et de liste des contacts** \* Le drag & drop depuis la liste vers le formulaire d'édition (en fait même sur ma version Wicket cela ne marchait pas) \* L'affichage de message d'erreurs \* La réutilisation des mêmes messages d'erreurs en validation serveur et javascript \* Le tri de la liste des contacts par nom et par prénom **Le résultat final**

Avant tout, voici une vidéo du résultat final.

**Les fonctions non réalisées** Je n'ai pas codé la fonction qui désactive dans le menu du haut la page courante. Cela serait possible en codant un taglib dans Grails, mais ce n'est pas disponible par défaut. Pour des raisons de problèmes Javascript, dans cette version j'ai créé 2 pages distinctes pour implémenter d'une part la saisie rapide et d'autre part l'édition inPlace. Une page utilise Scriptaculous, une autre page utilise Prototype. Il faut voir pourquoi lorsque j'ai les deux sur la même page cela ne fonctionne pas. Enfin concernant la non-duplication du code, par défaut la pratique de Grails propose de créer une page gsp par action, ce que j'ai fait pour simplifier le code. Il est possible sinon de rappeler une vue. Sur ce point je n'ai pas bataillé. **Ce que fait cette application en plus** Tout d'abord, c'est une vraie application avec une base de données, Hibernate, Spring, bref tout ce que vous avez l'habitude d'utiliser. En version développeur, j'utilise une base en mémoire HSQLDB qui est effacé à chaque redémarrage. En mode production nous pouvons utiliser n'importe quelle base supportée par Hibernate. L'application est constituée d'une entité Contact associé à un contrôleur en charge de tout ce qui est opération CRUD (Create, Research, Update et Delete). Au total, voici le nombre de fichiers et de lignes de code de l'application :

Name	Files	LOC
Controllers	2	121
Domain Classes	1	12
Tag Libraries	1	26
Totals	4	159

Grails n'a besoin que de 4 classes : une classe Contact pour le domaine, un contrôleur et une Taglib Grails pour me simplifier la vie. Si je dresse un tableau de comparaison des 2 versions :

	Code	Pages	Fichiers de conf	JS	Lignes de code
<b>Wicket</b>	7 fichiers java	5 pages HTML	1 seul ds web.xml	Aucuns	237 lignes
<b>Grails</b>	4 fichiers groovy	13 fichiers GSP	6 fichiers	Aucuns	158 lignes

Grails est moins verbeux que la version Java. J'ai utilisé l'utilitaire [CLOC](#) en renommant les fichiers groovy en java, afin de regarder la quantité de code. Attention cependant : Grails génère

des fichiers groovy de travail dans un répertoire caché et le nombre total de lignes de code est de 787 lignes. Mais si l'on regarde le nombre de lignes que l'on manipule, c'est bien 158 lignes.

N'oubliez pas que l'application persiste réellement dans une base de données les Contacts.

**Comment créer son application en 8 minutes ?** Assez parlé, il est temps peut-être de voir comment créer cette application. Tout compris, j'ai passé presque 2h30 pour réaliser l'ensemble de l'application. J'ai perdu beaucoup de temps sur la partie Ajax, ne connaissant pas encore les plugins de Grails par coeur. Par contre, et c'est ce que je veux vous montrer, j'ai écrit le coeur de l'application en 8 minutes. Pré-requis: - Téléchargez [Grails 1.2-RC2](#) et décompressez l'archive dans un répertoire C:\grails par exemple - Ajoutez une variable d'environnement sous Windows appelé `GRAILS_HOME` qui pointe vers C:\grails. - Ajouter ensuite `%GRAILS_HOME%\bin` dans votre variable `PATH` sous windows afin que Grails se trouve dans le path. Ouvrez un terminal et vérifiez que tout fonctionne avec "grails -version" - Sous Unix, éditez votre fichier `.bash_profile` et ajoutez la variable `GRAILS_HOME`. Ajoutez aussi `GRAILS_HOME/bin` à votre `PATH`. Validez l'installation avec "grails -version" Ok c'était facile. On continue ? Dans un répertoire, nous allons créer une application zencontact :

```
grails create-app zencontactdemo
...
...
```

Ensuite déplacez vous dans le répertoire zencontact. Nous allons commencer par créer notre entité "Contact" avec la commande `create-domain-class` :

```
cd zencontactdemo
grails create-domain-class org.letouilleur.demo.Contact
...
```

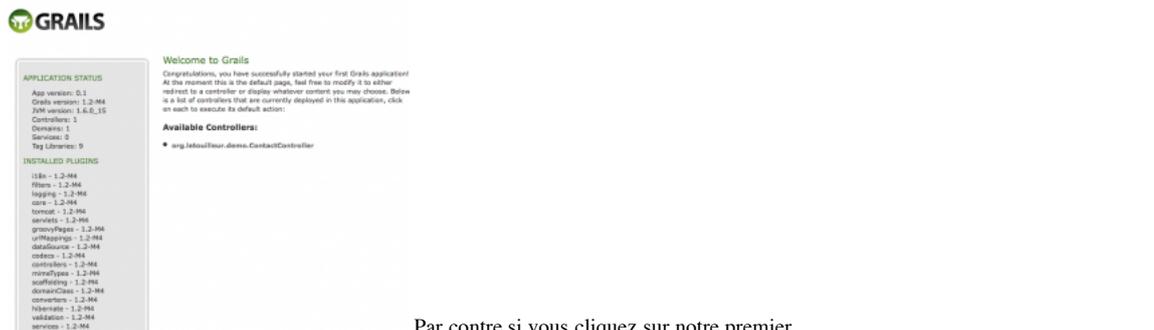
Maintenant, nous avons besoin d'un Contrôleur et de pages par défaut pour la partie CRUD. La commande `generate-all` va créer un Contrôleur pour afficher, éditer, supprimer et modifier les entités de type Contact. Il va aussi créer un ensemble de pages GSP pour chacune de ces actions. Le modèle est donc la class Contact, le contrôleur sera le fichier ContactController et enfin les vues seront l'ensemble des pages GSP (`edit.gsp`, `index.gsp`, `create.gsp`...) correspondant à chacune des actions du contrôleur Grails.

```
grails generate-all org.letouilleur.demo.Contact
```

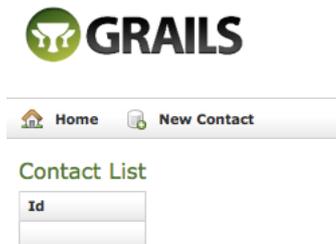
Je ne sais pas pour vous... mais comme cela fait déjà 6 minutes il faudrait peut-être tester non ? Alors nous allons lancer un serveur Jetty préconfiguré et déjà intégré à notre application. C'est le mode "développement" qui permet de travailler très rapidement sans devoir installer un serveur. La base de données est vraiment là, c'est une base en mémoire HSQLDB. Par défaut, en mode développement Grails efface et recrée la base et les tables à chaque redémarrage. Vous pouvez changer ce comportement bien sûr... En fait vous pouvez tout faire. Bref nous allons démarrer notre serveur et ensuite regarder la tête de notre application pour l'instant :

```
grails run-app
...
Environment set to development
[groovyc] Compiling 1 source file to /Users/nicolas/.grails/1.2-M4/projects/zencontactdemo/target/classes
Running Grails application..
Server running. Browse to http://localhost:8080/zencontactdemo
```

Si vous ouvrez maintenant votre navigateur favori vous devriez voir ceci :



Par contre si vous cliquez sur notre premier contrôleur, le moins que l'on puisse dire c'est que pour l'instant il ne se passe rien non ?



**Etape 2 : le domaine** Pour l'instant il ne se passe rien car Grails ne sait pas trop quoi faire. Pour l'aider, arrêtez le serveur pour l'instant et éditez le fichier `./grails-app/domain/org/letouilleur/demo/Contact.groovy` avec un éditeur de texte autre qu'Eclipse : En suivant le cahier des charges, nous allons déclarer les attributs `nom`, `prenom` comme

obligatoire. Remarquez aussi que le champ email est typé comme étant une adresse email. Groovy est expressif et Grails s'en sert simplement pour construire tout ce dont il a besoin. La validation, l'affichage des messages d'erreurs, les contraintes en base : la validation est déclarée ici une fois pour toute. Comme Beans Validator, Grails pense que le meilleur endroit pour déclarer les contraintes d'un modèle sont dans le modèle lui-même. Pas dans le GUI, pas dans la base. Cependant, la base de données qui est créé automatiquement par Grails reprendra ces contraintes afin de typer correctement les colonnes.

```
class Contact {
    String nom
    String prenom
    Date dateNaissance
    String email

    static constraints={
        prenom(blank:false)
        nom(blank:false)
        dateNaissance(max:new Date(), nullable:true)
        email(email:true, nullable:true, blank:true, unique:true)
    }
}
```

Après avoir changé autant notre domaine, nous allons simplement recréer toutes les vues et le contrôleur, avant de relancer le serveur. En ligne de commande, relancez la génération de cet ensemble et répondez "a" pour ALL lorsque Grails vous demande si vous voulez écraser vos anciens fichiers :

```
grails generate-all org.letouilleur.demo.Contact
...
File [...]zencontactdemo/grails-app/views/contact/create.gsp already exists. Overwrite? [y,n,a] a
```

Puis relancez le serveur (grails run-app) et revenons à notre page "contact". Vous pouvez maintenant créer un contact avec les champs que nous avons simplement déclaré dans notre POGO (Plain Old Groovy Object). Si vous cliquez sur "New Contact" vous pouvez maintenant vraiment créer des contacts et les stocker en base. Pas mal non au bout de 8 minutes ?



Home [New Contact](#)

### Contact List

Id	Prenom	Nom	Date Naissance	Email
1	Nicolas	Martignole	1975-09-24 00:00:00 CET	test@toto.fr

### Étape 3 : mettre en place le

**template** A cette étape nous avons une class ContactController.groovy qui nous donne toutes les fonctions du cahier des charges. Il suffit maintenant de travailler sur la vue, d'adapter un peu certaines parties, mais vous verrez que c'est plutôt simple. Tout d'abord, recopiez les images et la feuille CSS de Zenika dans le répertoire *web-app* de votre application. Nous allons débiter par modifier la page d'accueil. Il faut s'occuper du menu de navigation et de l'affichage de l'heure en temps réel. Dans le répertoire "grails-app/views/layouts/" nous allons d'abord nous attaquer au template. Grails utilise SiteMesh, et nous devons montrer qu'il est possible de définir le look and feel à un seul endroit. Pour cela je vais d'abord éditer le fichier main.gsp. Notez à la ligne 6 que ce tag permettra aux pages qui utiliseront ce template d'ajouter dans le HEADER du contenu. A la ligne 7 j'importe la librairie prototype. Un souci avec scriptaculous m'a obligé à déclarer un deuxième template "main2.gsp" pour certaines pages. A la ligne 12 j'ai déclaré les différentes URLs de mon application. Ici, en principe en Grails la bonne pratique veut que l'on crée une taglibs. Je l'ai fait un peu plus bas pour vous expliquer le principe. Pour nous, s'agissant d'un bout de code qui n'est pas dupliqué autre part, je peux très bien déclarer mes URLs en dur (principe DRY). La ligne 21 permet de déclarer la zone où Grails ajoutera le contenu des pages filles, que nous allons modifier dans 2 minutes.

```
<html>
<head>
  <title><g:layoutTitle default="zencontact" /></title>
  <link rel="stylesheet" href="{resource(dir: 'css', file: 'main.css')}" />
  <link rel="shortcut icon" href="{resource(dir: 'images', file: 'favicon.ico')}" type="image/x-icon" />
  <g:layoutHead />
  <g:javascript library="prototype" />
</head>
<body>
<div id="header">
  <g:link controller="contact" action="index">Accueil</g:link> |
  <g:link controller="contact" action="listWithIcons">Liste des contacts</g:link> |
  <g:link controller="contact" action="list">Liste v2</g:link> |
  <g:link controller="contact" action="create">Ajouter un contact</g:link>
</div>
<div id="content">
  <div id="contacts">

    <div class="contact">
      <g:layoutBody />
    </div>
  </div>
</div>
```

```

    </div>
  </div>
</div>
<div id="footer">
  <p id="legal"><a href="http://www.touilleurexpress.fr/">Le Touilleur Express</a> Copyright© 2009-2010</p>
</div>
</body>
</html>

```

Cr er un deuxi me fichier appel  main2.gsp dans le m me r pertoire, et changez simplement la ligne 7 en sp cifiant <g:javascript library="scriptaculous"/>. Voil  vous avez cr e 2 templates pour le site **Etape 4 : mettre en place la page d'accueil** La page d'accueil est simple. Elle affiche la navigation, et fait appel   une taglib pour afficher l'heure en temps r el. Cela va me permettre de vous expliquer comment cr er une taglib et ensuite comment ajouter un contr leur suppl mentaire. Vous allez m me faire de l'Ajax d s maintenant ! Voyons le code de cette page tout d'abord :

```

<html>
<head>
  <title>Zencontact avec Grails 1.2-M4</title>
  <meta name="layout" content="main2"/>
</head>
<body>
<table>
  <tbody><tr>
    <td width="300px"></td>
    <td width="300px"></td>
  </tr>
</tbody></table>
<br><br><br>
<!-- Appel d'un tag custom cree dans taglibs pr afficher la date -->
<div><g:showDateAndTime/></div>

<P>Voici la version Grails 1.2 de la d monstration Wicket de Zenika r alis e d but mars 2009 au Paris Java User Group.
  Nous retrouvons les fonctions du cahier des charges avec surtout un Controller complet capable de r ellement cr er,
  modifier et effacer un objet Contact.</P>
</body>
</html>

```

Explications : - la ligne 4 sp cifie que nous utiliserons le template "main2" pour cette page. En effet, notre taglib pour afficher l'heure utilise Scriptaculous. - la ligne 16 devrait vous intriguer... C'est une taglib, et pour l'instant elle n'existe pas Pour cr er la taglib vous devez ex cutez la commande suivante, relancez le serveur   la fin de la commande et laissez-le tourner.

```

grails create-tag-lib Tools
...
WARNING: You have not specified a package. It is good practise to place classes in packages (eg. mycompany.Book). Do you want to continue?
Y
...

```

```

grails run-app
...

```

Editez ensuite le fichier grails-app/taglib/ToolsTagLib.groovy et copiez-y ce bout de code:

```

class ToolsTagLib {
  def showDateAndTime = {attrs ->
    out << "<script type='text/javascript'">
    out << "new Ajax.PeriodicalUpdater('timerDiv', '/zencontactdemo/timer/getTime' , "
    out << "{method: 'get',frequency: 1,decay: 1});"
    out << "</script>"
    out << "<div id='timerDiv'></div>"
  }
}

```

Une taglib est automatiquement import e et devient disponible d s lors que vous la d clarez dans votre page. La ligne 16 de notre fichier index.gsp de tout   l'heure est presque pr te   fonctionner. Si vous regardez le code groovy de ce tag, vous verrez que j'utilise Scriptaculous et sa fonction Ajax.PeriodicalUpdater afin d'appeler l'url /zencontactdemo/timer/getTime. Cependant pour l'instant, il n'y a pas de contr leur associ . Nous allons donc cr er un contr leur et y ajouter la m thode getTime.

```

grails create-controller org.letouilleur.demo.Timer
...

```

Editez le fichier grails-app/controllers/org/letouilleur/demo/TimerController.groovy, voici comment celui-ci se pr sente par d faut :

```

package org.letouilleur.demo

class TimerController {

  def index = { }
}

```

En Grails, un Controller c'est une URL, et une m thode, c'est une sous-url de ce Controller. Ajoutez maintenant un fichier index.gsp dans le r pertoire grails-app/views/timer. Vous l'aurez compris : chaque Controller a son r pertoire, et une action correspond   un fichier de vue. Dans le fichier index.gsp, mettez "Coucou" et ensuite lancez le serveur si celui-ci est  teint. Vous verrez que l'url http://localhost:8080/zencontactdemo/timer fonctionne et affiche "Coucou". Par ailleurs, si vous  ditez le fichier index.gsp et que vous rechargez la page, **votre modification est prise en**

**compte sans relancer le serveur** comme avec Tapestry. Mais pas comme Wicket. Je vais faire mal là où cela fait mal. Avec Wicket il faut relancer le serveur dès lors que vous touchez à la partie Java. Avec Grails, vous êtes tranquille tant que vous n'ajoutez pas de nouveaux Controller ou que vous ne modifiez pas une classe de la partie Domain. A l'usage, on relance moins Grails que Wicket. J'ai travaillé avec les deux, croyez-moi. Pour terminer notre première page, il nous faut une méthode qui retourne la date et l'heure actuelle.

```
class TimerController {
    def index = { }
    def getTime = {
        def theTime = new java.text.SimpleDateFormat("HH:mm:ss dd MMM yyyy").format(new Date())
        [currentTime: theTime]
    }
}
```

Ajoutez aussi un fichier **getTime.gsp** dans le répertoire /grails-app/views/timer avec ce contenu:

```
Il est ${currentTime}
```

Pour tester le tout, ouvrez l'url <http://localhost:8080/zencontactdemo/timer/getTime> et vous devriez voir votre message s'afficher. Cette partie nous permet de voir comment Grails passe des paramètres du Modèle vers la Vue. La ligne 7 de la méthode getTime construit une Map avec pour clé "currentTime" et pour valeur le contenu de ma chaîne "theTime". Si je vous dis que derrière tout cela, c'est du Spring MVC, vous vous rendez compte de la simplicité de ce bout de code ? Notre vue utilise la syntaxe \${currentTime\$} pour afficher la valeur de l'objet passé en paramètre. C'est simple non ? Il est temps maintenant de charger notre page de démarrage et de vérifier que l'heure change toutes les secondes. Voilà pour la première page !

**Accueil | Liste des contacts | Liste v2 | Ajouter un contact**



Le Touilleur Express Copyright© 2009-2010

Je vous ai préparé un fichier zip avec l'ensemble du code de cette étape afin de vous aider si vous n'avez pas pu suivre toutes les explications. Cliquez-ici pour télécharger le fichier [grails\\_touilleur\\_etape1.tar.gz](#) **Conclusion étape 1** Je m'arrête là pour ce premier article. Si vous testez de votre côté vous verrez que la liste des contacts et l'ajout d'un nouveau contact fonctionne. A propos de validation, je ne sais pas si vous avez remarqué, mais l'email est vérifié. Si vous ne spécifiez pas de prénom ou de nom lors de la création, Grails vous affichera une erreur. Bref il ne reste plus qu'à implémenter les fonctions avancées de la liste de résultat, mais vous verrez que ce n'est pas beaucoup d'effort. Grails est un framework Web qui va de la couche présentation à la partie service et métier. Il vous surprendra par la rapidité à laquelle vous produisez un résultat parfait. Si lors de la lecture vous avez pensé parfois que je donnais trop de détails, c'est aussi pour vous montrer un **vrai** projet et pas un simple HelloWorld. C'est aussi pour cette raison que je trouve l'idée de Zenika intéressante : puisque l'on dit que c'est simple, autant montrer un exemple qui marche en une heure ! Wicket est certainement plus intégré et plus fort pour une application orientée vue. Et c'est un très bon framework. Je pense que Grails vous séduira plus par la partie Hibernate/CRUD que par la partie vue, qui reste simple. Du côté Ajax et Javascript j'ai un peu galéré, et là où Wicket propose des composants prêts à l'emploi, j'ai dû bidouiller pour couvrir le cahier des charges. J'exagère un peu car d'un autre côté, certaines fonctions sont très simples avec Grails. Vous souhaitez la liste de tous les contacts triés par nom ? En Grails c'est Contact.listAllByNom() que vous placez dans une list, et que vous itérez du côté de la vue... Nous verrons la prochaine fois que j'ai fait appel à des plugins pour compléter mon application. La vidéo vous montre le résultat terminé en moins de 5 minutes. La suite de cet article sera en ligne dans une semaine. A bientôt !

## Grails étape 2: itérer une collection et afficher

## le résultat

Les fêtes sont passées, voici la deuxième partie de mon article sur Grails consacrée à l'écriture de l'application Zencontact. Dans [le premier article](#) j'ai expliqué comment installer Grails et débiter son projet. Figurez-vous que depuis ce dernier article, j'ai trouvé comment utiliser les 2 librairies Prototype et Scriptaculous. Parfois il faut juste chercher un peu... Bref la version 2 est plus proche du cahier des charges de Zenika. Il ne manque presque rien je pense pour remplir [le cahier des charges](#). Aujourd'hui au programme : - comment mettre à jour sa version de Grails ? - Itérer une collection et afficher les résultats **Mettre à jour son application vers une nouvelle version de Grails** Pendant le break de la semaine de Noël, l'équipe de Grails a annoncé la sortie de [la version 1.2](#) du framework Grails. C'est parfait, cela va me permettre de vous montrer comment mettre à jour votre version de Grails, ainsi que notre projet zencontact. - Renommez votre ancien répertoire C:\grails en C:\grails-old par exemple - Téléchargez et décompressez Grails 1.2 dans le répertoire C:\grails - Téléchargez le projet de démarrage pour cet article, qui a été allégé depuis la version 1 via ce lien : [grails\\_zencontactdemo\\_debut\\_article2.tar.gz](#) - Placez-vous à la racine du projet zencontactdemo après avoir décompressé l'archive - Tapez simplement `grails upgrade` - Faites aussi le ménage avec `grails clean` avant de relancer le serveur avec `grails run-app` afin de tester la mise à jour Le script de mise à jour de Grails démarre. Nous pouvons constater que la version utilisée est bien la 1.2. Suivez les instructions, en principe cela ne devrait pas poser de problèmes.

```
macbook-pro-de-nicolas-martignole:zencontactdemo nicolas$ grails upgrade
Welcome to Grails 1.2.0 - http://grails.org/
Licensed under Apache Standard License 2.0
Grails home is set to: /Users/nicolas/Dev/grails/grails

Base Directory: /Users/nicolas/Dev/zenika/zencontactdemo/zencontactdemo
Resolving dependencies...
Dependencies resolved in 1502ms.
Running script /Users/nicolas/Dev/grails/grails/scripts/Upgrade.groovy
Environment set to development
NOTE: Your application currently expects grails version [1.2-M4], this target will upgrade it to Grails 1.2.0 ...

WARNING: This target will upgrade an older Grails application to 1.2.0.
However, tag libraries provided by earlier versions of Grails found in grails-app/taglib will be removed.
The target will not, however, delete tag libraries developed by yourself.
Are you sure you want to continue?
(y, n)
```

Lors du premier redémarrage, vous verrez que Grails met aussi à jour les Plugins, ce qui peut parfois poser des soucis avec les versions "beta". Mais ici, s'agissant de la version 1.2 il n'y aura aucuns soucis. **Les améliorations de la version 1.2 de Grails** La [release notes](#) de la version 1.2 est longue comme un guide d'installation de Windows. Cette version a été optimisée pour les performances. Spring 3.0 est utilisé par Grails et devient aussi utilisable pour votre code. N'importe quelle classe (java ou groovy) peut donc être annotée avec `@Component` afin d'être injectée dans une autre classe. Concernant la gestion des transactions, il est possible de définir un niveau de granularité plus bas pour les transactions, là où auparavant, l'ensemble d'un service Grails était transactionnel ou non. GORM est aussi configurable avec un ensemble de règle par défaut. J'aime beaucoup les nouveaux finders dynamiques qui traitent les Booleans. Enfin le support des requêtes nommées (named queries) la possibilité de faire des mappins "hasOne", la liste est longue. Attention dans la version 1.2, WebFlow est maintenant un plugin à part qu'il faut installer. Il n'est plus livré par défaut. Basé sur Spring WebFlow, ce plugin est très pratique pour faire des questionnaires, des wizards, des formulaires multi-écrans. Je pourrai écrire un article complet sur cette nouvelle version, mais croyez-moi, c'est une version très mature et très complète. Je dis cela pour les chagrineurs qui attendent que "*Grails soit stable*". **Scriptaculous et Prototype sur la même page** Avec la version 1.2 il est possible d'utiliser les 2 librairies Javascript Prototype et Scriptaculous sans soucis. Il suffit de les déclarer à la suite dans notre fichier main.gsp qui sert de template pour Sitemesh:

```
<html>
<head>
  <title><g:layoutTitle default="Zencontact"/></title>
  <link rel="stylesheet" href="${resource(dir: 'css', file: 'main.css')}" />
  <link rel="shortcut icon" href="${resource(dir: 'images', file: 'favicon.ico')}" type="image/x-icon" />
  <g:layoutHead />
<!-- ***** //-->
<g:javascript library="prototype" />
<g:javascript library="scriptaculous" />
<!-- ***** //-->
</head>
<body>
<div id="header">
  <g:link controller="contact" action="index">Accueil</g:link> |
  <g:link controller="contact" action="list">Liste des contacts</g:link> |
  <g:link controller="contact" action="create">Ajouter un contact</g:link>
</div>
<div id="content">
  <div id="contacts">
    <div class="contact">
      <g:layoutBody />
    </div>
  </div>
</div>
<div id="footer">
  <p id="legal"><a href="http://www.touilleurexpress.fr">Le Touilleur Express</a> Copyright© 2009-2010</p>
</div>
```

```
</body>
</html>
```

**La version de l'article 1** A la fin de l'article 1 il était possible de créer un contact, de le lister, et d'afficher l'heure sur la page d'accueil. Pour vous montrer maintenant une nouvelle version, j'ai préparé une archive du code source du début de l'étape 2. - Téléchargez le fichier [grails\\_zencontactdemo\\_debut\\_article2.tar.gz](#) - Décompressez-le dans un nouveau répertoire et lancez l'application. Comme je vous sens un peu fatigué après les fêtes je vous ai fait une vidéo avec ma belle voix que vous pouvez regarder en buvant une tisane (par exemple)

**Itérer une collection** Comment itérer une collection et afficher le résultat ? Plus clairement, nous devons lister les contacts et les afficher sur une page. Voyons comment cela fonctionne. Tout d'abord, souvenez-vous, nous utilisons une class Controller avec Grails. Tout se passe dans l'unique class ContactController.groovy. Pour récupérer la liste des contacts, nous utilisons un "finder". C'est une méthode statique qui est déclaré sur la classe Contact et qui permet à Grails de vous câbler tout cela sans aucuns efforts. Voici quelques exemples pour expliquer ce principe. Sachez qu'avec IDEA IntelliJ, celui-ci est capable de vous proposer via l'auto-complétion des méthodes, ce qui aide pas mal lorsque vous codez.

```
// Pour obtenir la liste des Contact
def myList = Contact.list()

// Pour obtenir la liste des utilisateurs dont le nom de famille est Martin
def myList = Contact.findAllByNom("Martin")

// Qui ressemble à Mar
def myList = Contact.findAllByNomLike("Ma%")

// Seulement le premier trouvé
def c1 = Contact.findByNameLike("Ma%")

// Trouve toutes les personnes dont la date de naissance
// est inférieure à la date d'aujourd'hui
def myList=Contact.findAllByDateNaissanceLessThan(new Date())

// Combine 2 attributs pr la recherche
def nicolas=Contact.findByNameAndPrenom("nicolas","martignole")
```

Que pensez-vous de tout cela ? Pas mal non ? Imaginez que le tout devient des requêtes avec Spring et Hibernate. Il est possible de combiner 2 attributs de la classe Contact au maximum comme sur le dernier exemple. J'adore Grails car **c'est précisément ce genre d'astuces qui font gagner du temps**. Pour vous expliquer comment construire la liste des contacts, nous allons casser un peu de code. Reprenez le fichier ContactController. Effacez pour l'instant tout ce qui se trouve dans la méthode list() et utilisez ce que vous venez de voir pour créer un objet myList. Comme vous pouvez le constater, on se fiche pas mal du type ici, à savoir un List. Pour retourner un objet à la vue, il suffit de le placer dans une Map, ce que je fais avec ces crochets à la fin de la méthode. Il n'y a pas de return car la dernière expression évaluée avec Grails est automatiquement retournée. La clé sera "contactInstanceList" et la valeur sera "myList".

```
package org.letouilleur.demo

class ContactController {

    static allowedMethods = [save: "POST", update: "POST", delete: "POST"]

    def index = {
        redirect(uri: '/')
    }

    def list = {
```

```
// Recupere la liste des contacts
def myList=Contact.list()

// place dans une Map cette liste avec la cle contactInstanceList
[contactInstanceList: myList]
}
...

```

Convention au lieu de configuration : pour afficher le résultat il suffit de créer un fichier list.gsp dans le répertoire views/contact. Ce fichier sera automatiquement appelé par Grails à la fin de l'exécution de la méthode list. Nous allons donc récupérer une liste de Contacts. Pour itérer cette collection du côté de la vue, nous utiliserons le tag <g:each>. Ce tag prend en argument une collection, et vous retourne une variable qu'il suffit d'utiliser. L'attribut status est un compteur très pratique lorsque vous souhaitez afficher une ligne sur deux dans un tableau avec une autre couleur. Il est possible d'utiliser le tag fieldValue ou directement le nom du curseur (ici contactInstanceList) et d'appeler l'attribut qui vous intéresse.

```
<g:each in="{contactInstanceList}"
    status="i"
    var="contactInstance">
    // version 1
    ${fieldValue(bean: contactInstance, field: "prenom")}
    // version 2 (voir explication)
    ${contactInstance?.nom}
    <br/>
</g:each>

```

Vous voyez ce point d'interrogation à la ligne 7 ? C'est un pattern de Groovy appelé [Safe Dereferencing](#) qui dit en quelque sorte : "...si tu as une valeur pour contactInstance alors appelle getNom() sinon ne fait rien". C'est un moyen de gérer les valeurs nulles sans if/then/else supplémentaire. Voilà comment afficher une collection : appeler l'un des finders, retourner la liste vers la vue et itérer la collection avec un curseur. Pour terminer nous allons afficher les noms des utilisateurs par ordre alphabétique. C'est très simple, il suffit d'utiliser le finder adéquat :

```
def list = {
    // Recupere la liste des contacts triee par Nom
    def myList=Contact.listOrderByNom()

    // place dans une Map cette liste avec la cle contactInstanceList
    [contactInstanceList: myList]
}

```

Voilà je vous laisse souffler et imaginez ce qu'il faudrait faire en Java avec du Spring et un peu d'Hibernate... Ce que j'aime dans Grails c'est que si vous voulez tester quelque chose rapidement, c'est faisable. **Paginer nos 102020 enregistrements sans pleurer** Tant qu'à faire, nous allons aussi préparer de quoi gérer la pagination. Imaginez que la base de données contienne beaucoup d'enregistrements. Il serait souhaitable de limiter le nombre d'enregistrement retournés et de paginer les résultats non ? Tout d'abord nous ajoutons le tag paginate dans le fichier list.gsp après le tag g:each. Ce tag va générer automatiquement les liens de navigation, le nombre de pages, bref toute l'artillerie nécessaire pour naviguer dans la liste des résultats. Si ce tag ne vous plaît pas, n'oubliez pas que vous pouvez créer le vôtre. Pour l'instant on note l'apparition d'un nouveau paramètre appelé contactInstanceTotal dans la vue :

```
<g:each in="{contactInstanceList}"
    status="i"
    var="contactInstance">
    // version 1
    ${fieldValue(bean: contactInstance, field: "prenom")}
    // version 2 (voir explication)
    ${contactInstance?.nom}
    <br/>
</g:each>
<g:paginate total="{contactInstanceTotal}"/>

```

Je modifie maintenant le code du Controller afin de gérer la pagination. Il est intéressant de vous expliquer en détail ce qui se passe dans ces 2 lignes car nous avons là plusieurs notions qui n'existent pas en Java, et qui viennent du monde de Groovy.

```
def list = {
    params.max = Math.min(params.max ? params.max.toInteger() : 10, 100)
    [contactInstanceList: Contact.listOrderByNom(params), contactInstanceTotal: Contact.count()]
}

```

Tout d'abord les paramètres de la requête HTTP (GET ou POST) sont automatiquement placés dans une Map appelé *params*. Vous pouvez écrire params.get("max") ou directement params.max afin de récupérer la valeur d'un paramètre HTTP.

```
params.max
```

Ensuite vous reconnaissez l'opérateur ternaire du monde java qui dit "*si le parametre max est spécifié alors prendre sa valeur en le transformant en un Integer, sinon prend 10*"

```
params.max ? params.max.toInteger() : 10
```

Cela me permet de vous donner un petit cours de Groovy et de vous parler de l'opérateur Elvis. L'opérateur Elvis comme en Java est un opérateur ternaire (If... then... else...) qui se lit de cette façon :

```
<condition> ? <expr1> : <expr2>
```

Si la condition est vraie alors l'opérateur 1 est évalué, sinon ce sera l'opérateur 2. Nous pourrions

par exemple écrire en Java ce bout de code qui affecte le nom de la personne :

```
String name = (person.getName() != null) ? person.getName() : "unknown"
```

Groovy étant capable de simplifier cela, un nouvel opérateur noté `?:` permet de raccourcir cette expression simplement :

```
// groovy
def name = person.getName() ?: "unknown"
```

L'opérateur Elvis en Groovy retourne le contenu de ce qu'il a sur la tête (à gauche) si cette expression est vraie, sinon ce qu'il a en dessous (à droite). Reprenons les quelques lignes de la méthode list de ContactController :

```
def list = {
    params.max = Math.min(params.max ? params.max.toInteger() : 10, 100)
    [contactInstanceList: Contact.listOrderByNom(params), contactInstanceTotal: Contact.count()]
}
```

La première ligne s'assure que le paramètre max existe et qu'il est dans une plage de valeur autorisée. La seconde ligne passe directement la map "params" à la méthode listOrderByNom. Grails utilise à tort ce principe de convention au lieu de la configuration. Il se trouve que cette méthode listXXX recherche automatiquement une clé "max" pour toute Map que vous lui passez en argument... Là je reconnais qu'il faut lire la documentation pour le savoir. Nous perdons en expressivité ce que l'on gagne en raccourci. Bref ce paramètre max va permettre de limiter le nombre d'enregistrements retournés. Enfin pour initialiser le nombre de contacts, nous utilisons la méthode count qui se charge de vous donner le nombre de contacts en base. C'est simple non ? **Faire un lien vers une autre page (méga super dur)** Bon quoi d'autre ? Il faut afficher une image devant chaque contact et créer un lien sur une image afin d'aller vers la page édition lorsque la personne clique dessus. Voici tout d'abord le code partiel de la page list.gsp :

```
// Iteration sur la liste des contacts tries par nom
<g:each in="${contactInstanceListByName}"
    status="i"
    var="contactInstance">
    <img src='<g:resource dir="images" file="people.png" />' alt="icon"/>
    ${contactInstance.prenom}
    ${contactInstance.nom}

    <g:link action="show" id="${contactInstance.id}">
    
    </g:link>
    <br/>
</g:each>
<div class="paginateButtons">
    <g:paginate total="${contactInstanceTotal}" />
</div>

<div class="contact">
    <g:include action="create" controller="contact" />
</div>
```

Ligne 5 : chargement d'une image Ligne 9 : utilisation du tag g:link. Nous allons appeler la méthode show du contrôleur ContactController. A noter que vous pouvez spécifier un autre contrôleur, que vous pouvez passer des paramètres, etc. Ce tag est assez puissant, [la documentation](#) vous donnera un aperçu de ce qu'il est possible de faire facilement. Ligne 14/15 et 16 nous retrouvons notre pagination avec une feuille de style pour l'affichage. Dans le cahier des charges de Zenika il est demandé aussi d'utiliser une autre page et de l'inclure dans la page qui liste les contacts. Le tag g:include permet de spécifier le nom d'un contrôleur et d'une action précise afin de charger la vue. Ligne 19 nous appelons l'action create de ContactController, et voilà c'est tout pour l'instant. **Reste à faire** Il reste maintenant 2 étapes compliquées : - mettre en place l'édition "in-place" du nom de famille - mettre en place un formulaire Ajax afin d'ajouter rapidement un nouvel utilisateur Nous verrons dans le troisième article comment écrire un tag personnalisé et ensuite comment mettre en place un formulaire Ajax en 3mn top chrono. A demain pour la suite !

## Concours Le Touilleur Express : gagnez 100 EUR avant le samedi 9 janvier



(dernière mise à jour dimanche 10 janvier) J'ai lancé il y a un mois un petit concours sur le Touilleur Express, et il est maintenant temps de relancer un peu le concours avant de le clôturer dans une semaine, le samedi 9 janvier à minuit. Le premier prix d'une valeur de **100 EUR** est un bon d'achat [Pimango](#). C'est un site de chèque cadeau virtuel qui permet d'acheter sur GrosBill, PriceMinister ou MisterGoodDeal. Et c'est offert de ma poche, je n'ai pas de deal avec eux :-). Je clôture le concours le samedi 9 janvier à minuit. Si par ailleurs vous souhaitez encore participer, vous pouvez m'envoyer votre photo à [concours@touilleur-express.fr](mailto:concours@touilleur-express.fr). Prenez une photo de votre écran avec le site du Touilleur Express clairement affiché. Si vous pouvez poser sur la photo c'est plus sympa. L'endroit ou la mise en situation doit être original. Une seule participation par adresse email. Si vous souhaitez modifier ou proposer une autre photo, envoyez-

moi une nouvelle photo et je la mettrai à jour. Pour rappel : - j'offre un livre sur Java au choix sur Amazon à l'un des participants tirés au sort. Le tirage au sort aura lieu le 10 janvier. - le gagnant sélectionné par vos votes durant la semaine du 11 gagne un bon d'achat [Pimengo](#) de 100 EUR à utiliser sur des sites comme GrosBill.com, PriceMinister ou autre. Toutes les photos sont sur [la galerie Flickr](#) où je vous encourage à laisser vos commentaires. - Jonathan un lecteur assidu:



- Romaintaz et une photo avec une centaine de touillette à café [caption id="attachment\_2496" align="alignnone" width="300" caption="Photo par



@romaintaz, des centaines de Touillettes à café"



[caption] - David Chau à l'Apple Store à Paris - Gilles Debunne : photo prise à Mountain View devant le building 43 en Californie, chez Google :



- Sylvain Ramousse :

regardez-moi ce menu Sushi spécial Touilleur Express avec un stock de touillette à café usagé

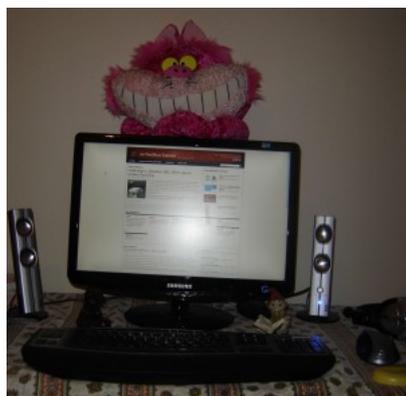


impressionnant !

Bureau de Yousri :



Photo de Christophe C. :



Credit photo : Gilles Debunne pour le concours sur <http://le-touilleur-express.fr>

Vous pouvez participer avec une photo prise avec votre iPhone, n'oubliez pas que simplement en participant,

vous pouvez gagner un livre sur Java sur Amazon. Allez les gars, je compte sur vous !

## Grails étape 3 : Bootstrap, XML, JSON, Ajax et création d'un GTag

3ème et dernier article sur [Grails](#) après [une prise en main](#) et [la réalisation d'un écran complet](#), nous allons voir comment créer un tag pour Grails et comment mettre en place Ajax. En bonus et pour débiter je vais vous montrer aussi quelques fonctionnalités de Grails : le Bootstrapping et les fonctions orientées REST pour afficher la liste des Contacts au format XML ou JSON en quelques minutes.

### Bootstrapping

Avant tout, vous pouvez télécharger [le code source de la partie 2](#) si vous ne l'avez pas déjà fait. Avez-vous remarqué qu'à chaque démarrage de l'application, il n'y a aucuns contacts ? C'est un peu gênant car il faut créer quelques Contacts pour voir la liste s'afficher. Grails en mode développement utilise une base de données en mémoire qui n'est pas persistée. A chaque démarrage il est donc normal que la liste des Contacts soit vide. Pour créer quelques enregistrements, ouvrez le fichier /grails-app/conf/BootStrap.groovy. Ce script est exécuté lors du démarrage. Nous allons créer quelques Contacts afin de tester. Rien de plus simple avec Grails, et vous verrez qu'il n'y a pas besoin de services, tout est livré sur l'entité Contact. Voici comment ajouter les contacts : dans la méthode init, créer un nouveau Contact :

```
class BootStrap {
    def init = { servletContext ->
        def nm = new Contact(prenom: "Nicolas", nom: "Martignole", dateNaissance: new Date()-120, email: "nicolas@test.fr")
        nm.save()
    }
    def destroy = {
    }
}
```

En Groovy il est possible d'utiliser ce que l'on appelle des arguments nommés. Franchement, vous ne trouvez pas cela plus clair que la version Java ? Là les paramètres sont clairement exprimés, sans ambiguïté. Notez aussi au passage qu'il est possible de retirer 120 jours à la date courante, car une Date en Groovy est facilement manipulable. C'est de la surcharge d'opérateur comme au bon vieux temps du C++. Si vous arrêtez et vous relancez le serveur, vous devriez voir maintenant que

[Accueil](#) | [Liste des contacts](#) | [Ajouter un contact](#)

Home <a href="#">New Contact</a>				
Contact List				
Id	Prenom	Nom	Date Naissance	Email
1	Nicolas	Martignole	2009-09-01 16:34:13 CEST	nicolas@test.fr

Le Touilleur Express Copyright© 2009-2010

vosre liste d'utilisateur s'est remplie :

Je vous laisse ajouter quelques utilisateurs, vous avez compris le principe. Sur votre version de production, il faudra ajouter un test afin de vérifier si l'application est en mode développement ou en mode production. Sans cela, ces entrées seront créées à chaque démarrage. Si vous voulez simplifier la saisie de la date de naissance, vous pouvez modifier la class Contact afin de rendre la date de naissance optionnel avec l'option nullable=true et retirer la contrainte max que j'avais placé dans la première version. Cela permettra de créer rapidement quelques enregistrements pour tester. La class Contact modifiée :

```
package org.letouilleur.demo

class Contact {
    String nom
    String prenom
    Date dateNaissance
    String email

    static constraints={
        prenom(blank:false)
        nom(blank:false)
        dateNaissance(nullable:true)
        email(email:true,nullable:true,blank:true,unique:true)
    }
}
```

Voici mon fichier Bootstrap final :

```
import org.letouilleur.demo.Contact
import javax.servlet.ServletContext

/**
 * Script de demarrage
 */
class BootStrap {

    def init = {ServletContext ctx ->
```

```

environments {
  production {

    ctx.setAttribute("env", "prod")
  }
  development {
    createTestEntries()
    ctx.setAttribute("env", "dev")
  }
}

}

def destroy = {
}

// Cette methode est appele uniquement en mode developpement afin de creer des Contacts

void createTestEntries() {
  def nm = new Contact(prenom: "Nicolas", nom: "Martignole", dateNaissance: new Date() - 120, email: "nicolas@test.fr")
  nm.save()

  // Permet de verifier que le Contact est bien stocke ou non
  def c = new Contact(prenom: "Paul", nom: "Gontran", dateNaissance: new Date(), email: "paul@toto.fr")
  c.save()
  if (c.hasErrors())
    println c.errors

  new Contact(prenom: "Georges", nom: "Lucas", dateNaissance: new Date() - 1, email: "lucas@star.fr").save(failOnError: true)
  new Contact(prenom: "Gregory", nom: "Madisson", dateNaissance: new Date() - 3, email: "mads@son.com").save()
  new Contact(prenom: "Jonathan", nom: "Lalou", dateNaissance: new Date() - 2, email: "john@lecowboy.fr").save()
  new Contact(prenom: "Katherine", nom: "Grah", dateNaissance: new Date() - 3, email: "kat@noop.fr").save()
  new Contact(prenom: "Amelie", nom: "Blou", dateNaissance: new Date() - 56, email: "amelie@noop.fr").save()
  new Contact(prenom: "Anatole", nom: "Ducib", dateNaissance: new Date() - 23, email: "am@noop.fr").save()
  new Contact(prenom: "Kim", nom: "ProdMan", dateNaissance: new Date() - 4, email: "kim@noop.fr").save()
  new Contact(prenom: "David", nom: "CeChau", dateNaissance: new Date() - 91, email: "david@noop.fr").save()
  new Contact(prenom: "Pierre", nom: "Henry", dateNaissance: new Date() - 3, email: "pierre@noop.fr").save()
  new Contact(prenom: "Andre", nom: "Magphone", email: "ant@noop.fr").save()
}
}

```

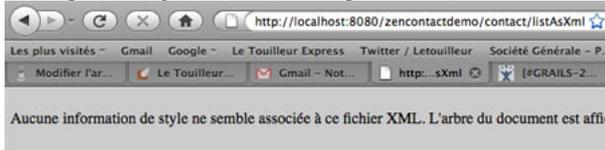
**Générer une liste au format XML ou JSON** Certaines fonctions de Grails sont vraiment sympas. Nous allons voir comment retourner la liste des contacts au format XML simple, au format avancé et enfin au format JSON. Pour cela, éditez `ContactController`. Vous savez maintenant que pour déclarer une nouvelle action, il suffit de créer une méthode. Grails rend accessible cette méthode sans même qu'il ne soit nécessaire de relancer le serveur. Essayez d'ajouter ces quelques lignes à la fin de votre fichier :

```

def listAsXml = {
  def newList = Contact.listOrderByNom()
  render newList as grails.converters.XML
}

```

Si vous regardez maintenant la page <http://localhost:8080/zencontactdemo/contact/listAsXml> vous verrez que Grails génère une liste triée par nom de famille des Contacts.



```

-<list>
- <contact id="7">
  <prenom>Amelie</prenom>
  <dateNaissance>2009-11-04 19:12:31.587 CET</dateNaissance>
  <email>amelie@noop.fr</email>
  <nom>Blou</nom>
</contact>
- <contact id="6">
  <prenom>Katherine</prenom>
  <dateNaissance>2009-12-27 19:12:31.579 CET</dateNaissance>
  <email>kat@noop.fr</email>
  <nom>Blou</nom>
</contact>
- <contact id="10">
  <prenom>David</prenom>
  <dateNaissance>2009-09-30 19:12:31.613 CEST</dateNaissance>
  <email>david@noop.fr</email>
  <nom>CeChau</nom>
</contact>
- <contact id="8">
  <prenom>Anatole</prenom>
  <dateNaissance>2009-12-07 19:12:31.596 CET</dateNaissance>
  <email>am@noop.fr</email>
  <nom>Ducib</nom>
</contact>

```

Grails propose les Converters. Il en existe 4 par défaut ([voir la doc](#) dans une autre fenêtre). Les converters "deep" gèrent les associations, les relations alors que les 2 autres Converters ne rendent pas les feuilles. Ici Contact n'ayant aucunes relations, j'utilise le Converter par défaut. Il est aussi possible de générer un fichier XML très facilement par exemple:

```

def converter = Contact.list() as XML

```

```
converter.render(new java.io.FileWriter("/path/to/my/file.xml"));
```

**Générer la liste des Contacts au format JSON** Le converter JSON par défaut gère en plus très bien l'encoding au format UTF-8 des entités, ce qui nous permettra de gérer sans soucis les accents par exemple :

```
import grails.converters.JSON
class ContactController {
  ..
  ...

  def listAsJson = {
    def newList = Contact.listOrderByNom()
    render newList as JSON
  }
}
```

Ce qui donne

```
[{"class":"org.letouilleur.demo.Contact","id":7,"prenom":"Amelie","dateNaissance":"2009-11-04T18:12:31Z","email":"amelie@noop.fr","nom":"B...
...
etc.
```

**Un seul enregistrement au format XML** Enfin si vous ne voulez retourner qu'un seul enregistrement, cela ne pose pas plus de problèmes. Voici une méthode showAsXml qui affichera un seul des enregistrements. Il suffit de passer l'id en argument.

```
def showAsXml = {
  def contactInstance = Contact.get(params.id)
  if (!contactInstance) {
    flash.message = "${message(code: 'default.not.found.message', args: [message(code: 'contact.label', default: 'Contact'), params.id])}
    redirect(action: "list")
  }
  else {
    render contactInstance as XML
  }
}
```

Et pour tester cette méthode (toujours sans relancer Grails) ouvrez par exemple cette URL : <http://localhost:8080/zencontactdemo/contact/showAsXml?id=7> Voilà, vous savez maintenant que Grails peut exposer facilement au format XML ou JSON des entités.

## Le tag editInLine

Là nous attaquons une partie un peu plus compliquée. Dans le cahier des charges de Zenika, il faut mettre en place une fonction d'édition du nom de famille en Ajax, en utilisant la fonction [Ajax.InPlaceEditor](#) de la librairie Javascript [Scriptaculous](#). J'ai un peu cherché mais il ne semble pas qu'il existe de tag Grails pour faire cela. Pas de plugins non plus. Enfin cela tombe bien, l'objectif est de vous montrer comment écrire ce tag. Mais avant cela, voici une petite vidéo du résultat final, de ce que vous allez coder dans un instant (désolé si vous n'avez pas Quicktime) :

Pour réaliser cela voici ce que nous allons faire: - créer un tag personnalisé dans la class ToolsTagLib - modifier le fichier list.gsp pour utiliser ce tag - ajouter une méthode dans ContactController pour sauvegarder le changement de nom - ajouter une page gsp pour afficher en gras

le nom édité Le tout sans relancer le serveur. Avant tout, vérifiez bien que le fichier grails-app/views/layout/main.gsp contient bien la déclaration de la librairie Prototype et de la librairie Scriptaculous.

```
<html>
<head>
  <title><g:layoutTitle default="Zencontact"/></title>
  <link rel="stylesheet" href="${resource(dir: 'css', file: 'main.css')}" />
  <link rel="shortcut icon" href="${resource(dir: 'images', file: 'favicon.ico')}" type="image/x-icon" />
  <g:layoutHead/>
  <g:javascript library="prototype" />
  <g:javascript library="scriptaculous" />
</head>
```

**Création d'un tag personnalisé** Lors du premier atelier nous avons créé une class ToolsTagLib.groovy dans le répertoire grails-app/taglib. Nous allons ajouter une nouvelle méthode dans cette classe afin de créer un tag editInPlace pour Grails. J'ai décidé que ce tag prendrait quelques arguments. Dans la page list.gsp, remplacez la ligne suivante utilisée lors de l'affichage du nom du contact par le code suivant :

Dans grails-app/views/contact/list.gsp

```
effacez la ligne
<td>${fieldValue(bean: contactInstance, field: "nom")}</td>
```

et remplacez par :

```
<td>
<g:editInPlace id="${contactInstance.id}" action="inlineEdit" controller="contact" paramName="nom" initialValue="${contactInstance.nom}" />
</td>
```

Notre tag prend plusieurs arguments. L'id du contact édité, l'action à appeler sur le contrôleur, le nom de celui-ci, le nom du paramètre initial et enfin la valeur à afficher lors du chargement de la page. Je vous donne le code du tag, avant de l'expliquer. C'est pas très simple, j'y ai passé une bonne demie heure le temps de le tester et de trouver comment récupérer les paramètres facilement :

```
def editInPlace = {attrs, body ->
  def id = attrs.remove('id')
  def initialVal = attrs.remove('initialValue')

  out << "<span id='${id}' class='editInline'>"
  out << initialVal
  out << "</span>"

  out << "<script type='text/javascript'>"
  out << "new Ajax.InPlaceEditor('${id}', '"

  // id has already been removed from attrs map
  out << createLink(action: attrs.action, id: id, controller: attrs.controller)

  out << "',{"

  if (attrs.paramName) {
    out << "callback: function(form,value) { return '${attrs.paramName}=' + encodeURIComponent(value) } "
  }
  out << "}); "
  out << "</script>"
}
```

Ligne 2 et 3 : je retire les paramètres passés en argument du tag dans la page GSP et je les stocke dans des variables Ligne 13 : j'utilise la fonction createLink de Grails qui est présente par défaut pour créer un lien Ligne 17 : je précise aussi le nom du paramètre Le tout repose sur la fonction Ajax.InPlaceEditor qui permet facilement de gérer l'édition. C'est cette même librairie qui est utilisée dans la version de Zenika. Exactement la même. La différence étant que là, au lieu d'avoir un composant, je dois coder un tag. **Modification de la page list.gsp** Editez et vérifiez que la page list.gsp utilise maintenant votre tag. Nous avons maintenant la version finale de cette page. Regardez attentivement le code, c'est relativement simple.

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <meta name="layout" content="main" />
  <g:set var="entityName" value="${message(code: 'contact.label', default: 'Contact')}" />
  <title><g:message code="default.list.label" args="[entityName]" /></title>
</head>
<body>
<div class="body">
  <g:if test="${flash.message}">
    <div class="message">${flash.message}</div>
  </g:if>

  <P>Sur cette page vous pouvez éditer une fiche, ajouter rapidement un nouvel utilisateur avec un formulaire Ajax. Nous montrons aussi comment il est possible d'inclure une autre vue dans la vue active sans ajouter une ligne de code au contrôleur.</P>
  <P>Vous pouvez éditer directement le nom de famille d'un utilisateur en cliquant sur son nom grâce à la librairie Scriptaculous.</P>

  <div id="updateMe">
    <g:each in="${contactInstanceList}" status="i" var="contactInstance">
      <img src='${resource dir="images" file="people.png"}' alt="icon" />
      ${contactInstance.prenom}
    </g:each>
  </div>
```

```

<!-- Exemple d'utilisation d'un tag custom -->
<g:editInPlace id="${contactInstance.id}" action="inlineEdit" controller="contact" paramName="nom" initialValue="${contactInstance.nom}" />

<g:link action="show" id="${contactInstance.id}">
  
</g:link>
<br/>
</g:each>
<div class="paginateButtons">
  <g:paginate total="${contactInstanceTotal}"/>
</div>

</div>

</div>

<div class="contact">
  <g:include action="create" controller="contact"/>
</div>
</body>
</html>

```

Nous retrouvons bien notre tag à la ligne 27, et aussi ce que nous avons préparé dans l'article précédent. Il ne manque plus que l'édition avec le formulaire Ajax et nous aurons terminé. **Ajout d'une méthode pour sauver le nouveau nom** Si vous testez le code à cet instant, vous verrez que cela marche pas. Le nom de famille n'est pas persisté. Il manque encore 2 choses : une méthode dans ContactController et une vue GSP. Le nom de la méthode appelée par le code Ajax.EditInPlace a été passé en argument : *inlineEdit*. Il faut donc créer cette méthode et ensuite une vue inlineEdit.gsp pour terminer cette partie. Voici la nouvelle méthode à ajouter à la fin de ContactController :

```

def inlineEdit = {
  def contactInstance = Contact.get(params.id)
  if (!contactInstance) {
    flash.message = "${message(code: 'default.not.found.message', args: [message(code: 'contact.label', default: 'Contact'), params.id])}"
    redirect(action: "list")
  }
  else {
    contactInstance.nom=params.nom
    contactInstance.save()
    return [contactInstance: contactInstance]
  }
}

```

Retenez que seul la ligne 8 est intéressante : nous modifions le nom de l'entité chargée et nous la sauvons ensuite. Il ne reste plus qu'à ajouter un fichier inlineEdit.gsp dans le répertoire grails-app/views/contact avec cette seule et unique ligne:

```

// Une seule ligne
<strong>${contactInstance.nom}</strong>

```

Une seule ligne car cette page est rendue... via Ajax. Il n'y a rien d'autres à ajouter. Voilà c'est terminé pour l'édition. Regardez la vidéo complète à la fin de cet article si vous souhaitez faire une pause avant la dernière partie. **Dernière modification : le formulaire de saisie rapide Ajax** Tout d'abord nous allons modifier la vue contact.gsp car nous n'avons pas encore activé de datePicker. Et comme la vue create est utilisée aussi dans la vue list, autant faire d'une pierre, deux coups. Pour le datePicker nous avons l'embarras du choix car il existe plusieurs plugins Grails. J'ai pris le plugin "[Calendar](#)". Allons-y, il y en a pour 3 minutes. Tout d'abord installons le plugin

```
grails install-plugin calendar
```

Voici la version finale de la page create.gsp avec la mise en page de la version Wicket de Zenika :

```

<%@ page import="org.letouilleur.demo.Contact" %>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <meta name="layout" content="main"/>
  <g:set var="entityName" value="${message(code: 'contact.label', default: 'Contact')}" />
  <title><g:message code="default.create.label" args="[entityName]" /></title>
  <calendar:resources />
</head>
<body>

<div class="body">
  <g:if test="${flash.message}">
    <div class="message">${flash.message}</div>
  </g:if>
  <g:hasErrors bean="${contactInstance}">
    <div class="errors">
      <g:renderErrors bean="${contactInstance}" as="list" />
    </div>
  </g:hasErrors>
  <g:form action="save" method="post">
    <div class="dialog">
      <table>
        <tbody>

          <tr class="prop">
            <td valign="top" class="name">
              <label for="prenom"><g:message code="contact.prenom.label" default="Prenom" /></label>
            </td>
            <td valign="top" class="value ${hasErrors(bean: contactInstance, field: 'prenom', 'errors')}>

```

```

        <g:textField name="prenom" value="{contactInstance?.prenom}"/>
    </td>
</tr>
<tr class="prop">
    <td valign="top" class="name">
        <label for="nom"><g:message code="contact.nom.label" default="Nom"/></label>
    </td>
    <td valign="top" class="value ${hasErrors(bean: contactInstance, field: 'nom', 'errors')}">
        <g:textField name="nom" value="{contactInstance?.nom}"/>
    </td>
</tr>
<tr class="prop">
    <td valign="top" class="name">
        <label for="dateNaissance"><g:message code="contact.dateNaissance.label" default="Date Naissance"/></label>
    </td>
    <td valign="top" class="value ${hasErrors(bean: contactInstance, field: 'dateNaissance', 'errors')}">
        <!-- utilisation du plugin calendar, a installer avec grails install-plugin calendar -->
        <calendar:datePicker name="dateNaissance" defaultValue="{contactInstance?.dateNaissance}" dateFormat="%d/%m/%Y"/>
    </td>
</tr>
<tr class="prop">
    <td valign="top" class="name">
        <label for="email"><g:message code="contact.email.label" default="Email"/></label>
    </td>
    <td valign="top" class="value ${hasErrors(bean: contactInstance, field: 'email', 'errors')}">
        <g:textField name="email" value="{contactInstance?.email}"/>
    </td>
</tr>
</tbody>
</table>
</div>
<div class="buttonbox">
    <g:submitButton name="create" class="button" value="{message(code: 'default.button.create.label', default: 'Create')}" />
</div>
</g:form>
</div>
</body>
</html>

```

Ligne 8 : cette déclaration permet de charger les ressources javascript nécessaires au plugin calendar. Ligne 48: utilisation du datePicker Si vous rechargez maintenant [la page de création](#) nous avons un datePicker en bonne et due forme. De même pour [la page qui liste les contacts](#) car celle-ci importe la page create.gsp. Passons maintenant à l'ajout du formulaire de saisie rapide à proprement parler. Grails fournit un support d'AJAX très puissant et simple à utiliser. La librairie des tags de base permet de réaliser rapidement des fonctions puissantes en Javascript, tout en assurant l'envers du décor avec les actions dans le Controller. En fait c'est tellement simple que vous n'aurez pas le temps de terminer votre café en lisant ce qui suit : Dans list.gsp, ajoutez un nouveau formulaire basé sur le tag formRemote:

```

<div id="updateMe">
    <g:each in="{contactInstanceList}" status="i" var="contactInstance">
        <img src='{resource dir="images" file="people.png"}' alt="icon"/>
        {contactInstance.prenom}

        <!-- Exemple d'utilisation d'un tag custom -->
        <g:editInPlace id="{contactInstance.id}" action="inlineEdit" controller="contact" paramName="nom" initialValue="{contactInstance.nom}" />

        <g:link action="show" id="{contactInstance.id}">
            
        </g:link>
        <br/>
    </g:each>
    <div class="paginateButtons">
        <g:paginate total="{contactInstanceTotal}" />
    </div>
</div>

<div>
    <g:formRemote name="quickSave"
                action="saveQuickInput"
                on404="alert('not found!')"
                update="updateMe"
                url="{[action:'saveQuickInput']}">
        Prénom : <g:textField name="prenom" value="{contactInstance?.prenom}"/>
        Nom : <g:textField name="nom" value="{contactInstance?.nom}"/>

        <g:submitButton name="create" class="button" value="Créer"/>

    </g:formRemote>
</div>

```

Ligne 1 : vous notez que j'ai nommé le DIV principal "updateMe" qui comme vous l'avez deviné... doit se rafraîchir dès qu'un contact est ajouté. Ligne 21 : nom de l'action à appeler sur ContactController Ligne 23 : nom du DIV à rappeler en cas de succès Le reste du formulaire est tout à fait normal, il n'y a rien d'exceptionnel. Il ne reste plus qu'à ajouter une nouvelle action dans ContactController. Je vous laisse faire maintenant, vous savez comment faire :

```

def saveQuickInput = {
    params.max = Math.min(params.max ? params.max.toInteger() : 10, 100)

    def contactInstance = new Contact(params)

    def existingUser = Contact.findByNomAndPrenom(params.nom, params.prenom)
    if (!existingUser) {

```

```

    contactInstance.save(flush: true)
    [contactInstanceList: Contact.listOrderByNom(params), user: contactInstance, contactInstanceTotal: Contact.count()]
  } else {
    [contactInstanceList: Contact.list(params), user: contactInstance, erreur: "Cet utilisateur existe déjà"]
  }
}
}

```

Et enfin comme toute action, il nous faut une vue. Je dis cela, ce n'est pas tout à fait vrai. Il est possible d'appeler une autre vue et vous n'êtes pas obligé de créer un fichier GSP par action dans votre Controller. Pour terminer voici le contenu de la page qui sera affichée lorsque l'utilisateur aura saisi un nouveau Contact :

```

<div class="errors">
  ${erreur} <br/>
</div>

<g:if test="${flash.message}">
  <div class="errors">${flash.message}</div>
</g:if>

<g:each in="${contactInstanceList}" status="i" var="contactInstance">
  <img src='<g:resource dir="images" file="people.png"/>' alt="icon"/>

  <g:if test="${contactInstance.nom == user.nom}">
    <strong>
  </g:if>
  ${contactInstance.prenom}

  <!-- Exemple d'utilisation d'un tag custom -->
  <g:editInPlace id="${contactInstance.id}" action="inlineEdit" controller="contact" paramName="nom" initialValue="${contactInstance.nom}" />

  <g:if test="${contactInstance.nom == user.nom}">
    </strong>
  </g:if>

  <g:link action="show" id="${contactInstance.id}">
    <img src='${resource(dir: 'images', file: 'edit.png')}' alt="edit" border="0"/>
  </g:link>
  <br/>
</g:each>
<div class="paginateButtons">
  <g:paginate total="${contactInstanceTotal}" controller="contact" action="list" next="Suivant" prev="Précédent" />
</div>

<div class="errors">
  ${erreur} <br/>
</div>

```

La pagination sur la page de résultat pose un souci : si vous créez un utilisateur appelé Bob Zorro par exemple, celui-ci sera affiché sur la 2ème page et vous ne le verrez pas tout de suite... Je pense que c'est un souci d'ergonomie plus que technique, donc nous n'en tiendrons pas compte ici. Voilà c'est terminé. Nous avons maintenant aussi le formulaire de création rapide. Si vous avez tout suivi jusqu'ici... chapeau. Cet article est énorme à lire, et il doit tenir sur 12 pages. Je vous remercie si vous êtes encore là, car tout ceci peut vous sembler un peu indigeste.

## Conclusion

Pour terminer voici une vidéo de l'application complète, avec une feuille de style adaptée, quelques améliorations sur la mise en page, mais le tout sans truccages, sans fil en nylon qui font bouger les tags sur la page et en essayant de coller à l'esprit du projet de Zenika au plus juste. Pour les observateurs attentifs il y a un bug à un moment donné... je vous laisse regarder :

**Alors Grails c'est mieux que Wicket ?** Il est temps de poser un peu le clavier et de porter une autocritique sur le projet. Là où j'ai clairement gagné du temps : sur tout ce qui est test, construction des actions et de la partie Controller. Là où j'ai bien passé 1 heure, c'est sur la création du tag personnalisé et sur la gestion de la partie Ajax. Pas compliqué, mais ne connaissant pas... j'ai un peu navigué dans la documentation de Grails pour boucler le tout. Je me demande combien de temps il me faudrait pour refaire le projet en partant de zéro, en connaissant ce qu'il y a à faire... Je dirai 1h00 je pense, en imaginant que je conserve le code du tag editInPlace quelque part. Le reste est faisable rapidement. Dans vos commentaires vous me demandez si Grails est mieux que GWT. Ils ne sont pas comparables sur le plan de la création d'application. GWT est fait pour la création d'application riche du côté du navigateur. Je préfère comparer GWT à Flex. GWT vous laisse écrire la partie serveur avec la persistance gérée par un ORM comme Hibernate, et l'injection de dépendance gérée par Spring par exemple. Grails ne vous laisse pas mettre le nez dans cette partie, ce qui je pense doit avoir ses limites. Vous êtes tributaire des mises à jour des versions de Spring et d'Hibernate puisque celles-ci font parties intégrantes de Grails... D'un autre côté, je vois cela comme un avantage : Grails vous prépare une version qui marche des différentes bibliothèques, et tant que vous n'avez pas un bug dans celle-ci, vous êtes tranquille. **Peut-on utiliser Grails en production ?** Oui. Souvenez-vous qu'à la fin, vous faites "grails war" et votre archive est prête à être déployée comme vous l'auriez construit à la main. SpringSource fournit aussi du support il me semble sur Grails [d'après cette page](#) mais difficile à dire. **Est-ce que Grails est réservé à des petites applications ou des maquettes ?** Je ne pense pas. Si je suis votre raisonnement, dès que votre application devient "sérieuse" il vous faut Java EE 6 ? C'est pas un peu bête comme raisonnement ? Je suis certain que vous pensez que PHP c'est "pour des petits sites" et Python "c'est pour des gars qui n'aiment pas Java". Excusez-moi de vous parler franchement : va falloir sortir un peu mon garçon. La majorité des projets Webs aujourd'hui tournent sur du PHP. Python a des Closures depuis 2004 je crois là où nous, dans le monde Java, nous n'aurons rien avant fin 2010... Coluche disait : "JE ME MARRE !" **Est-ce que Grails c'est vraiment de la balle et tu es prêt à mettre ton projet dessus ?** Un autre Nicolas m'a posé cette question. Est-ce que je suis prêt à risquer un projet, une entreprise en l'occurrence, sur ce framework ? Ma réponse est oui. Je limite le risque en prenant un outil qui me donne un feedback plus rapidement qu'avec Java. Cela me permet de valider rapidement ou non des idées. Et dans le cadre d'une start-up où vous devez trouver des idées, je suis vraiment content de ce choix. Je commence à voir les limites de Grails, mais aussi à voir les parties où il ne faut pas hésiter à se servir deux fois. **Et Groovy ? Et la suite ?** Depuis mi-2009 j'ai changé. Je dirai presque que j'ai muri : je sens de moins en moins l'option "Java seul contre le reste du monde". Je pense qu'il est intéressant pour vous de commencer à apprendre Scala ou Groovy en ce moment, bref n'importe quel langage qui tourne sur la JVM. **La force de Java ce n'est plus le langage. C'est la communauté, c'est les projets open-sources de qualité et c'est la machine virtuelle. Play! ou Grails ?** Ben j'en sais trop rien... Je dirai que je préfère Grails à cause de Groovy. Play! m'a semblé proche de Grails, et rejoint la famille des frameworks Webs Hyper-productifs qui écrasent gentiment les frameworks orientés composants comme JSF pour n'en citer qu'un. C'est mon avis très personnel, et souvenez-vous que je portais aux louanges JSF 1.2 il y a 2 ou 3 ans. A mon avis, le ticket d'entrée avec Play! doit être moins élevé. Souvenez-vous de la différence entre apprendre à faire du Ski Alpin et apprendre à faire du Snowboard. Grails c'est plutôt du Ski : tu vas bien galérer au début, là où Play! c'est ton pote Java avec donc le bénéfice de toutes les bibliothèques qui existent déjà... Peut-être que tu risques de te faire moins mal ? Il faudrait confronter les deux frameworks. Alors je vais lancer un appel solennel : **Si quelqu'un de l'équipe Play! lit cet article, ça ne vous dit pas de faire votre**

**version du projet de Zenika ?** Là on aura une idée plus précise je pense. Voilà j'espère que cette série d'articles vous aura donné une idée de ce que l'on peut faire avec Grails 1.2. Je vous proposerai d'autres articles sur les sujets que j'ai découvert ces derniers mois, il y a de quoi se chauffer cet hiver.

## Code du projet

Vous pouvez télécharger le code source de la version finale : [grails\\_final\\_touilleurexpress.tar.gz](http://grails_final_touilleurexpress.tar.gz)

## Université du SI 2010 : 1er et 2 juillet



La 3ème Université du SI organisée par [OCTO Technology](#) en partenariat avec [The Boston Consulting Group](#) aura lieu le 1er et le 2 juillet 2010 à Paris. Au programme cette année, Martin Fowler de ThougWorks après Neal Ford l'an passé [qui m'avait bluffé](#), Jean-Philippe Courtois de Microsoft International, des Keynotes de qualité je pense avec Bernard Stiegler, Philosophe ou Yves Morieux du BCG, bref certainement encore une bonne occasion pour s'aérer le cerveau. Le concept de l'USI est de mélanger pendant 2 jours des Boss et des Geeks. Cela donne une formule réussie qui vous permet d'observer comment 2 populations à priori difficiles à réunir, peuvent venir se nourrir intellectuellement. Un peu comme les fauves et les buffles en Afrique autour d'un même point d'eau. L'an passé, OCTO m'avait proposé de participer et d'en parler en tant que Blogger. L'idée d'inviter quelques geeks de la communauté Java, c'est de vous permettre de partager cette conférence, avec un œil plutôt technique, mais en essayant aussi de vous intéresser à l'autre côté du fleuve, aux besoins et aux inquiétudes de nos Boss. J'avais beaucoup apprécié les Keynotes et les rencontres lors des pauses. Quelque part le sentiment [d'un TED à la Française](#). Si vous voulez relire les articles de l'an dernier au coin du feu, en voici la liste: - [Vaincre la force du plouf](#) - [L'informatique conviviale \(voyage au coeur des grandes organisations\)](#) - [L'ergonomie : entre forme et fonction, une recette par l'exemple](#) - [Wiki contre CMS](#) avec XWiki présenté par Vincent Massol et eXo Portal présenté par Tugdual Grall d'eXo Platform - [Quelques idées issues des grands du Web pour remettre en cause vos réflexes d'architectes](#) - [Le Management One on One](#) - [Google pour l'Entreprise puis Amazon for Enterprise](#) - [Keynote d'ouverture par Joël de Rosnay](#) Enfin pour terminer [le bilan de l'USI 2009](#) Je crois que les inscriptions sont déjà ouvertes, rendez-vous en juillet prochain.

## Startup : la photo non contractuelle

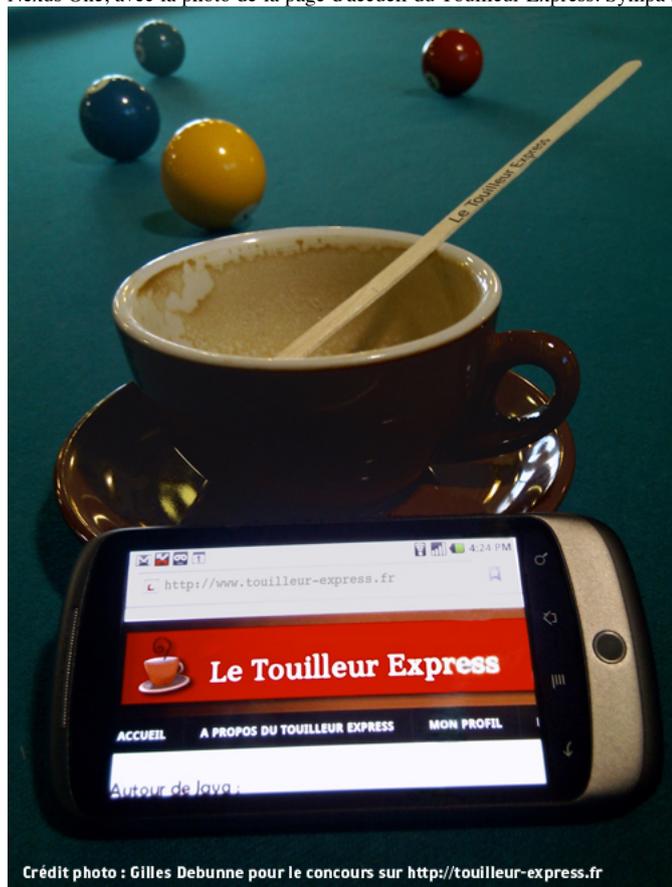


L'autre matin en arrivant à la Gare pour prendre le train, je vois des jeunes filles qui distribuent dans le froid des prospectus. Il s'agit d'un programme immobilier dont la construction va bientôt débiter en ville. Ces petites brochures montrent généralement quelques futurs appartements en situation. Un dessin idyllique trop beau pour être honnête représente votre futur immeuble dans une belle rue avec des arbres, aucuns immeubles aux alentours, des gens qui baladent un chien en chemisette, bref le cliché d'un monde parfait. Cela vous rassure, et vous tournez la page pour découvrir les différents appartements. Un crayonné un peu audacieux d'une blonde à forte poitrine entrain de nourrir un chat dans une cuisine, un plan de travail en marbre du Portugal, des volets électriques, vous vous voyez dans votre future cuisine... Toutes ces illustrations sont annotées d'un astérisque discret. Le petit texte en bas dit clairement "Photos non contractuelles, illustrations d'exemples, chiffres donnés à titre d'exemple". Dans la petite Startup où je travaille en ce moment, après un mois très Scrum, je me suis retrouvé à devoir créer un document d'architecture générale, suivi de spécifications techniques détaillées. Cela n'a pas été facile de se poser pendant une semaine, de sortir un plan détaillé, un chiffrage en [jours-hommes](#), un planning "prévisionnel". A tel point que si Luc qui travaille avec moi ne m'avait pas un peu forcé, j'aurais laissé tombé cette partie. Mon point de vue était de dire qu'il était inutile de perdre du temps à faire de la documentation... Grave erreur d'appréciation, caprice de consultant biberonné à la Scrum, je me rends compte que je n'étais plus objectif. Voici pourquoi. Tout d'abord avant de débiter n'importe quel projet, il est nécessaire de réfléchir un minimum à l'architecture globale. Celui qui vous vend que faire du Scrum dès le démarrage fonctionne, a tort. Il est nécessaire de réaliser une phase d'amorçage, durant laquelle l'Agilité peut être utilisée, mais durant laquelle il est encore trop tôt pour parler de méthodes de développements. Un bon tableau blanc, un appareil photo pour prendre des photos de vos dessins sur ce tableau, un stock de feuille, des feutres de vos enfants de toutes les couleurs, un bon café, des gens avec vous pour challenger vos idées... vous devez passer par cette phase d'accouchement. C'est un business plan technique. Réaliser un chiffrage... mon dieu. Moi qui bassine tout le monde en disant que les chiffres ne servent à rien, que le prédictif c'est bon pour Mr Marabout à la sortie du Métro Barbès... Je continue à croire cela. Je me rends compte aussi que pour mesurer les ressources nécessaires, communiquer nos besoins techniques, et ensuite pouvoir suivre le réalisé par rapport au prédictif, il faut chiffrer. Il ne faut pas tenter de chiffrer 234,4 jours mais simplement se rendre compte que, au vue du cahier des charges, 200 jours théoriques sont à prévoir. Ensuite nous avons découpé notre projet avec un chemin critique. Le travail que j'ai fait en décembre me permet de lever les points sur lesquels nous étions pas ou peu confiants. [Ce réseau de PERT](#) est un indicateur qui nous servira plus tard pour le séquençement de nos itérations. Nous allons faire du Scrum, mais nous avons aussi d'abord pensé à ce que nous voulions faire, en parlant exclusivement technique. Un peu comme un plan

pour un architecte. Viens ensuite la documentation. Celle-ci est un indicateur qui rassure les non-techniciens de l'équipe. Elle indique que nous avons marqué nos idées et nos discussions quelque part. Je pense que ce n'est pas complètement débile, que de capturer à un instant donné vos idées non ? Si vous faites des prototypes ou des maquettes en Java, soignez la présentation. Prenez 20 minutes pour chercher une feuille de style et quelques icônes afin que votre prototype soit sympa. L'aspect compte énormément pour les personnes qui ne sont pas techniques. Elles regardent vos écrans, vous expliquez votre superbe intégration de Drools avec Grails, mais elle, elle ne voit que cette page HTML moche qui ressemble à un kleenex sale plutôt qu'à son futur site Internet. Je vais vous parler de vous. Lorsque j'avais ce matin cette petite brochure sur ces futurs appartements dans ma ville, en fait j'ai été rassuré de voir la qualité de la cuisine, l'environnement, la belle façade du bâtiment. Du coup je ne me suis pas interrogé si le plombier a posé du PER ou du cuivre dans la salle de bain. Je ne me suis pas demandé s'il y aura des prises RJ45 dans le salon, si finalement les fenêtres sont en double ou triple vitrage. J'ai été rassuré par l'illustration. Je n'y connais rien en construction de bâtiment, je vous fais confiance. Par contre si votre immeuble de haute-qualité environnemental est moche, je ne l'achèterai pas. Et bien c'est peut-être la même chose entre vous, développeur, et votre client qui n'est pas développeur. Pensez-y. Lorsque vous rédigez de la documentation, vous rassurez vos partenaires car vous montrez que vous avez pris le temps de capturer ce qui fera le coeur de la Startup. Cette documentation peut vous faire réfléchir sur le projet. Elle peut vous permettre de vous dire qu'il vous faut 5 développeurs. Je me rends compte que je suis entrain d'écrire un "Business Plan Technique" afin d'aller le présenter aux responsables de la Startup. Moi aussi je suis à la recherche de crédits-développeurs et de crédits-achats de machine. Et c'est finalement ces quelques documents écrits en une semaine qui vont me servir pour faire tout cela. Alors oui nous ferons du Scrum, mais lorsque nous aurons une équipe. Et maintenant nous pouvons justifier qu'il faut 4 développeurs par exemple... Suite de mes aventures au prochain numéro.

## Concours le Touilleur Express : le Nexus One en photo

Je boucle ce week-end [le concours photo](#) lancé avant les vacances, avec un prix de 100 EUR pour le vainqueur et un livre sur Java par tirage au sort pour tous les participants. Gilles qui travaille chez Google en Californie m'a envoyé une photo du dernier téléphone portable de Google, le Nexus One, avec la photo de la page d'accueil du Touilleur Express. Sympa non ?



Crédit photo : Gilles Debunne pour le concours sur <http://touilleur-express.fr>

Bref je vous fais un autre billet demain pour boucler et que l'on décide du vainqueur, merci à tous ceux qui ont participé.

## Concours le Touilleur Express Noël 2009

[Mise à jour] Vote terminé, le vainqueur est Gilles avec la photo #8 Merci à tous, le concours est terminé. J'ai regroupé sur cette page les photos de chacun des participants avec leur pseudo. J'ai

demandé à une main innocente, ma fille, de piocher le nom de l'un de vous dans un chapeau et le gagnant d'un livre sur Java sur Amazon est Romainaz. Voici les photos et pour chacune d'elle, le pseudo de la personne. Votez simplement en postant un commentaire. Je départagerai les ex-aequo par tirage au sort. **Votez avant mercredi soir !!!** Le gagant remportera un bon d'achat de 100 EUR. Si par ailleurs vous voulez continuer à m'envoyer vos photos, n'hésitez pas ! **Photo 01**



**Photo 2** [caption id="attachment\_2496" align="alignnone" width="300" caption="Photo par @romaintaz, des centaines de Touillettes à



café"] [/caption] **Photo 3**



**Photo 4**



Photo 5



Photo 6



Photo 7

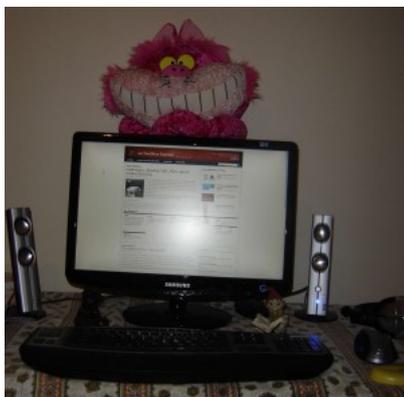


Photo 8



Crédit photo : Gilles Dubreton pour le concours sur <http://touilleur-express.fr>

Indiquez

le nom de la photo en commentaire, je clôture les votes mercredi soir.

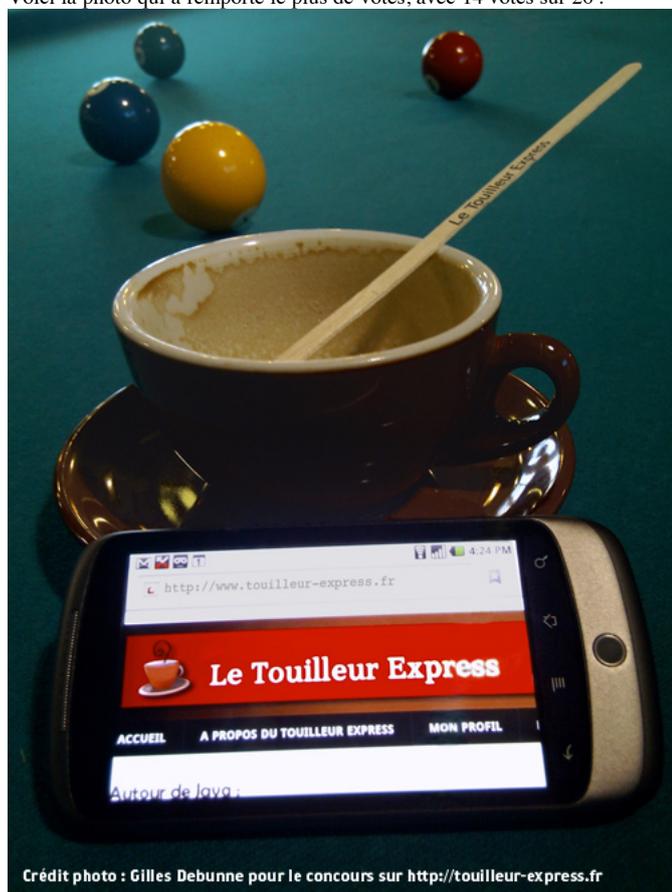
## Soirée Performance au Paris JUG lundi 18 janvier



L'année 2010 repart et nous vous proposons une première soirée dans une nouvelle salle le lundi 18 janvier. Nous recevons Kirk Pepperdine, que nous avons déjà reçu [en avril 2008](#) dans une toute petite salle à Paris. Kirk est un Java Champion tout comme Antonio Goncalvès. Il est souvent présent aux conférences en Europe. J'ai eu la chance de dîner avec lui à Jazoon 2009 à Zurich, c'est un sacré gaillard. Rendez-vous donc le lundi 18 janvier, de 19h15 à 22h30 [dans les locaux de la Sorbonne Paris IV](#), métro Ligne 3 Malesherbes. N'oubliez pas de vous inscrire sur le site du Paris JUG. Sinon le mardi 9 février aura lieu **le deuxième anniversaire du Paris JUG** avec une soirée exceptionnelle. Nous avons réservé une grande salle et un hall d'exposition, afin d'accueillir tout le monde. Comme vous le savez, les soirées du Paris JUG sont souvent complètes quelques jours avant celles-ci. Cette fois-ci c'est une salle de 500 places, un espace où nous vous offrirons un buffet, et très certainement quelques stands pour des exposants. Marquez-bien cette date dans votre agenda : mardi 9 février, à la Sorbonne aussi.

## Résultat du Concours Noël 2009

Voici la photo qui a remporté le plus de votes, avec 14 votes sur 26 :



Crédit photo : Gilles Debunne pour le concours sur <http://touilleur-express.fr>

Bravo à Gilles qui va recevoir un bon d'achat de 100 EUR. Bravo aussi à Romain qui a remporté par tirage au sort un livre et qui a demandé à recevoir [le livre "Beginning with Java EE 6 Platform with Glassfish 3"](#). Rendez-vous bientôt pour d'autres concours sur le Touilleur Express.

## NoSQL : un mouvement qui prend de l'ampleur



Il y a un nouveau mouvement dont je voulais vous parler depuis quelques temps qui commence à faire du bruit : [NoSQL](#) (Not only SQL). Lancé [au mois de juin 2009](#) à l'initiative de quelques architectes de grands sites communautaire comme Facebook ou LinkedIn, ce mouvement prend de l'ampleur. La première soirée du groupe NoSQL France a eu lieu en janvier à Paris [à l'initiative d'Octo Technology](#). L'objectif de ce mouvement est de parler des solutions de bases de données non relationnelles majoritairement utilisées par les grands sites communautaires. Les sites à fort trafic utilisent tous des solutions non relationnelles pour leurs bases de données. En fait au rythme où vont les choses, le modèle relationnel est remis en question dans toutes les architectures où les principes ACID ne peuvent pas être respectés. Je vous avais déjà parlé [du Théorème de CAP](#) qui montre qu'une donnée ne peut pas être à la fois consistante, accessible à tout moment par tous, et partitionnée de manière sûre :

[...] le Théorème de CAP d'Eric Brewer, utile lorsque l'on parle de données massivement partagées : - Strong Consistency: all clients see the same view, even in presence of updates - High Availability: all clients can find some replica of the data, even in the presence of failures - Partition-tolerance: the system properties hold even when the system is partitioned Ce qui donne en français : - L'ensemble des clients voient la même vue, même lorsqu'il y a des mises-à-jour - Massivement disponible : tous les clients peuvent trouver de la donnée répliquée, même lorsqu'un crash survient - Partitionable : votre app-serveur et votre base de données peuvent être installé sur 2 machines, votre système est tolérant au partitionnement. Il s'avère que seul 2 des 3 postulats peuvent être respectés en même temps

C'est donc à l'initiative des architectes des solutions de Cloud Computing et des sites communautaires, que ce mouvement NoSQL a débuté. Lors des premières réunions aux USA, il y a eu ainsi des présentations des projets suivants : - Le Projet [Voldemort](#) par Jay Kreps, LinkedIn (slides [pdf](#) [ppt](#), [video1](#), [video2](#)) - [Le projet Apache Cassandra](#) - Avinash Lakshman, Facebook (slides [pdf](#) [ppt](#), [video](#)) - Le projet [Dynomite](#) par Cliff Moon, Powerset ([slides](#), [video](#)) - Le projet Hadoop [HBase](#) - Ryan Rawson, Stumbleupon ([slides](#), [video](#)) - [Hypertable](#) - Doug Judd, Zvents (slides [pdf](#) [ppt](#), [video1](#), [video2](#)) - Le projet [Apache CouchDB](#) par Chris Anderson, couch.io ([slides](#), [video1](#), [video2](#)) Le projet Voldemort par exemple, [utilisé par les équipes d'Algo Deal](#) en France, est une énorme Map distribuée, capable de redistribuer la donnée entre plusieurs serveurs, en divisant celle-ci et en proposant une architecture transparente d'accès à un espace clé-valeur. LinkedIn s'en sert aussi en partie sur leurs serveurs. Avec 4 opérations de bases, c'est une API très simple (GET,PUT,GET\_ALL et DELETE) qui rappelle le concept de JavaSpace il y a 10 ans. Jay Kreps dans sa présentation parle d'un environnement avec 12 serveurs, 300 millions d'opérations par jour, 4 milliards d'objets stockés dans 10 datastores pour son équipe seulement... Apache Cassandra est une base de données distribuées, caractérisée par une structure de données par colonnes et super-colonnes, là où Voldemort est une simple hashmap. En termes de performances, un exemple sur une base MySQL de 50Gb montre des temps de lecture moyen de 350 ms, et de 1.5 ms pour Cassandra. Pour réaliser cette prouesse, Cassandra est une base de données distribuées, qui fonctionne donc avec plusieurs serveurs physiques. Vous vous en êtes certainement déjà servi si vous avez un compte sur Facebook (ce qui n'est pas mon cas) car cette technologie a été créé par les équipes de Facebook. L'accès à un nœud Cassandra s'effectue via l'API Apache Thrift. Un guide complet pour Mac OS X est dispo [ici](#). C'est encore délicat à mettre en oeuvre par rapport à d'autres projets, mais les concepts sont intéressants. Je passe sur Dynomite, un projet en Python encore trop en Alpha, pour parler de HBase, la base de données du projet Hadoop. C'est un projet qui vise à proposer une solution lorsque vous devez stocker une quantité énorme de données, et qu'une répartition physique de celles-ci devient obligatoire. Pensez à Google, c'est un candidat pour HBase. Thrift là encore peut être utilisé pour l'accès aux données. HBase est un projet construit sur Java, et la programmation est clairement plus facile que d'autres projets. On retrouve les concepts de Quorums, de finalement cohérent ([Eventually Consistent](#), lisez cet article !) bref ce qui caractérise un système de données massivement distribué. Hypertable est un projet porté par les équipes de ZVents qui est sponsorisé par [Baidu](#), l'un des plus gros moteurs de recherche en Chine. Peut-être plus proche de l'approche classique par Table, Hypertable n'est pas très bien documenté et il est difficile de connaître ses points forts par rapport aux autres. Il dispose lui aussi d'une interface pour Apache Thrift. Enfin Apache CouchDB est assez connu maintenant dans la communauté Java. C'est une base orientée document, capable d'utiliser l'algo Map-Reduce pour effectuer de la recherche sur les données. Ecrit avec le langage [Erlang](#), l'accès s'effectue avec une API Javascript/JSON orientée REST ce qui ouvre l'accès de la base à un très grand nombre de clients. CouchDB n'est pas une base relationnel, il n'y a pas de schémas. L'indexation des documents avec un B-Tree construit sur le nom du document et un numéro de séquence permet de gérer les versions des documents tout en assurant des performances intéressantes. Pour un moteur de gestion de documents (GED) ou pour un moteur de gestion de pages webs (WCM) je pense que CouchDB doit être intéressant à regarder. Je pense aussi à un JCR (Java Content Repository) où le serveur serait CouchDB par exemple... (*ça c'est pour Julien V. qui râle sur Tuan sur Twitter*) ([voir aussi ce comparatif entre JCR et CouchDB](#)) **Et un standard ?** Il nous manque quelque chose pour l'instant pour que ces solutions soient exploitables dans le cadre d'une solution d'entreprise : un standard. Merci au langage SQL et aux drivers JDBC. Grâce à eux nous pouvons coder avec une base HSQL en développement, puis une base MySQL en production. Il manque encore un peu de recul, de bonnes pratiques, et de personnes capables de factoriser ces concepts pour créer un nouveau standard. Il y a une niche en ce moment pour qu'un éditeur passe devant les autres et

propose une solution. Mais le marché aujourd'hui est piloté par les grands sites comme MySpace, Facebook ou Google, les seuls avec ces problématiques, les seuls avec les finances suffisantes pour explorer et développer ce type de solution. **Et toi ? Tu comptes t'en servir ?** Dans notre projet j'y ai pensé. [Neo4J](#) est une base qui travaille sur la théorie des Graphes pour modéliser les tables sous la forme de Nodes, avec des liens entrants, des liens sortants, et qui permet s'abstraire des contraintes des tables relationnelles, donc de monter en charge plus facilement. Grails a [un plugin Neo4J](#) qui offre un mapping limité, afin d'utiliser directement les objets d'un domaine Grails avec Neo4J. Il y a aussi un plugin Google App Engine, qui permet de travailler avec la base de données type BigTable de GAE et une application Grails. **Les 4 types de bases NoSQL** Nous venons de voir qu'il existe 4 catégories de solution dans le monde du "Not Only SQL". - les solutions clés-valeurs comme Voldemort, Dynamite ou le projet Tokyo - les solutions basées sur le concept de BigTable de Google comme HBase ou Hypertable - les solutions orientées document avec la recherche et l'indexation en tête, inspiré par Lotus Notes, comme CouchDB - les solutions basées sur la théorie des graphes (Euler) comme Neo4J, AllegroGraph ou VertexDB Les solutions les plus simples comme Voldemort permettent de gérer facilement un grand nombre de données. A l'opposé, les solutions comme Neo4J permettent de gérer des graphes d'objet où les relations entre les données sont très importantes par rapport à un modèle plat. Ceci a un coût en complexité et en performance. Un modèle graphique ressemble à un modèle clé-valeur où il serait possible de définir des relations entre les objets. Le sens des relations est important, l'exemple que j'ai vu était [ce dessin de super-vilains de Marvel](#). Une structure en graphe enfin peut être traversée par un Visitor, ce qui permet de réaliser un véritable crawler et d'extraire des données de votre base... Neo4J est une solution de persistance pour des applications développées aussi avec des choses plus exotiques comme Qi4J dont nous avons parlé [il y a quelques temps](#). Enfin Google a sorti [un papier intéressant en juin 2009](#) sur la théorie des Graphes autour de leur projet "Google Pregel", si vous cherchez un article pour bien dormir ce soir, je vous le conseille chaudement. Je vous tiendrai informé de la prochaine réunion du groupe NoSQL sur Paris, car c'est aussi un sujet sur lequel je travaille. **Références** - [Le Débriefing officiel de la première rencontre NoSQL](#) en juin 2009 - [Le Google Group NoSQL](#) si vous voulez "en être" et briller aux diners du JUG - [Article de Denis Dollfus](#) sur NoSQL - [Neo4J et Graph SQL](#) une présentation intéressante. Le volume des données s'accroît rapidement, et les solutions relationnelles vont/doivent mourir. Sympa pour Oracle. - [Google Pregel](#) avec des pointeurs intéressants. - [Devoxx 2009, présentation HBase](#) article très intéressante de Michael Figuiere de [Xebia](#) - [Amazon SimpleDB, le Harry Potter de Voldemort ?](#) et [Des Alternatives aux Bases de données relationnelles](#), très bonne introduction au monde du NoSQL par Olivier Mallassi d'[OCTO Technology](#).

## Rumeurs Apple de janvier : la iTablet ?



Le 27 janvier

prochain se tiendra "Come See Our Latest Creation", un événement exceptionnel à San Francisco organisé par Apple. La rumeur court qu'Apple annoncera la sortie d'une tablette. Au format Letter US, cette tablette sera plus légère qu'un MacBook Air, utilisera le système d'exploitation de l'iPhone, et visera avant tout à prendre le marché du livre numérique. C'est donc plus un iBook, avec en ligne de mire les appareils comme [l'Amazon Kindle](#) qu'un tueur des mini-PC comme [l'ASUS eeePC](#). Les dernières rumeurs annoncent qu'une caméra sera disponible, afin de proposer la possibilité de faire des vidéos conférences, fonctionnalité absente de l'iPhone pour l'instant. Comme l'ensemble des appareils Apple, l'appareil sera assez dépouillé au niveau des boutons physiques. L'essentiel de l'interface, et donc un clavier virtuel, reprend l'idée du clavier virtuel de l'iPhone. Ceci permet de penser que le support de l'écriture de droite à gauche (Hébreu, Arabe) et que le support d'un grand nombre de langues dès le départ, permettra à Apple de proposer sa Tablette à tout le monde, très rapidement. Le marché visé par cette nouvelle tablette est tout simplement énorme. Apple ambitionne de prendre le marché du livre éducatif en main. Ce n'est ni plus ni moins qu'un appareil destiné à remplacer les livres et les cahiers dans les écoles. A ce titre, des fonctions de reconnaissance de l'écriture, technologie bien maîtrisée de nos jours, seraient aussi prévues. Je pense qu'il faudra voir cet appareil comme le remplaçant des tablettes en cire de nos ancêtres Grecs, comme le remplaçant du cahier à spirale et des livres de cours, volumineux et coûteux. HarperCollins, McGraw Hill sont des éditeurs ayant dors et déjà fait état [de négociations et de contacts](#) avec Apple. L'appareil sera certainement connecté en permanence à Internet, via Wifi et 3G. Le support du Wifi semble évident, l'appareil étant destiné à être utilisé à la maison, à l'école et dans le monde de l'entreprise. Le support de la 3G semble moins évident, car l'usage nomade n'apparaît pas comme étant quelque chose de prévu. La vision d'Apple semble plutôt de dire que, chaque matin l'appareil se synchronisera et téléchargera les derniers journaux au format électronique, mais qu'une fois dans les transports, l'appareil ne sera pas connecté. Un article sur

iLounge évoque aussi [en fin d'article](#) la possibilité que la fonction de partage réseau via Bluetooth de l'iPhone soit utilisé, afin de se servir de son iPhone comme "clé 3G" et télécharger ainsi l'iTablet de la partie 3G. Il semble aussi acquis qu'Apple pense utiliser quelques bonnes idées de produits concurrents, comme la prise de note accompagnée de l'enregistrement de la voix. Imaginez que vous êtes un Geek comme moi, dans une conférence Java. Alors que vous prenez des notes sur la table, avec vos schémas, celle-ci enregistre la voix du présentateur. Plus tard, lorsque vous relisez vos notes, en pointant une zone sur l'écran, vous entendrez le passage audio associé à la note. C'est une idée qui existe déjà et dont je vous avais parlé [en juin 2008](#), lorsque Dimitri Baeli d'eXo Platform m'avait présenté [le LiveScribe](#). En terme de prix, nous pouvons nous attendre à un prix assez élevé. Je pense à quelque chose entre 300 et 400 EUR, la nouveauté va se payer au prix cher. Je ne crois pas que l'iTablet sera vendue par les opérateurs téléphoniques. Je vois plus cet appareil comme un nouveau périphérique informatique, utilisé quelques heures à la maison pour lire ses emails par exemple. Au travail, je vois très bien cet appareil remplacer le bon vieux cahier de prises de notes. Ce sera peut-être aussi un appareil capable de communiquer via Bluetooth avec d'autres iTablet, afin de proposer des réunions animées. Est-ce qu'une fonction de caméra et une intégration d'iChat permettront aussi de proposer de réelles fonctionnalités de conférence ? Il y a aussi un exercice où Apple est très fort, c'est la désinformation. Il n'est pas exclu que dans 4 jours, aucune tablette ne soit annoncée. Nous aurons peut-être droit à l'annonce d'un mini-Mac, destiné à s'attaquer au marché des microPC par exemple. Personne n'a vu venir le MacBook Air. Je me souviens d'un soir de janvier 2008 où je vous avais proposé de suivre en direct la retranscription de l'événement. Lorsque Steve Jobs a sorti le MacBook Air de son enveloppe de papier, il y eut un



grand moment dans la salle: Je vous donne donc rendez-vous à nouveau mercredi 27 janvier 2010 à 19h00 sur le Touilleur Express. L'événement aura lieu à 10h00 du matin à Yerba Buena Gardens à San Francisco. Tous les fans d'Apple suivront cet événement je l'espère. Bonne semaine et à mercredi !

## Gym avec Groovy, exercice 1



Merci de participer ce matin à notre petite série d'exercice sur [Groovy](#).

Après un échauffement, nous allons travailler les abdos-fixtures, faire quelques moulinets de Closures, s'étirer avec des astuces sur les fichiers, bref que du bon code pour débiter. Je vous prends en main cette semaine, avec chaque jour un petit exercice simple afin de vous réhabituer à un bon langage, léger. Après avoir pris votre tapis... de souris, venez vous assoir devant moi, et écoutez attentivement. Les exercices sont adaptés du livre "[Groovy Recipes](#)" par Scott Davis pour ceux qui veulent pratiquer de leur côté.

### La surcharge pondérale d'opérateur

Grâce à Java, je sais que ce petit problème de surcharge pondérale n'était plus trop d'actualité. Pourtant nous allons voir ensemble qu'avec [Groovy](#), la surcharge d'opérateur est une chose très pratique. Pour cela, nous allons travailler avec la date. Notre série d'exercices se déroulera dans une salle toute neuve hébergée sur Google App Engine (prononcez eng-i-neu) développée avec [Gaelyk](#) par [Guillaume Laforge](#). Ouvrez donc [la Groovy Web Console](#) et débutons par quelques exercices simples pour travailler la surcharge d'opérateur. Tapez simplement le texte ci-dessous puis appuyez sur "Execute script". Je note avec le symbole "=>" le résultat de l'opération.

```
def today=new Date()
=> Mon Jan 25 07:57:43 UTC 2010
```

La surcharge d'opérateur est une fonction très pratique en Groovy. Imaginons que vous souhaitez connaître la date dans 7 jours. L'opérateur '+' étant surchargé dans le cas d'une Date, il est alors possible d'écrire ce code:

```
def today=new Date()
println today
=> Tue Jan 26 08:28:33 UTC 2010

def tomorrow=today.next()
println tomorrow
=> Wed Jan 27 08:28:33 UTC 2010

def nextWeek=tomorrow+7
```

```
println nextWeek
=> Wed Feb 03 08:28:33 UTC 2010

def yesterday=today.previous()
println yesterday
=> Mon Jan 25 08:28:33 UTC 2010
```

Derrière l'opérateur plus, Groovy appelle la fonction next(). Voici une liste des opérateurs et des méthodes correspondants à ces opérateurs en Groovy :

Opérateur	Méthode
a==b or a!=b	a.equals(b)
a + b	a.plus(b)
a - b	a.minus(b)
a * b	a.multiply(b)
a / b	a.div(b)
a % b	a.mod(b)
a++ ou ++a	a.next()
a-- ou --a	a.previous()
a & b	a.and(b)
a   b	a.or(b)

Les opérateurs suivants sont plus communs, surtout lors de la manipulation des Listes :

Opérateur	Méthode
a[b]	a.getAt(b)
a[b]=c	a.putAt(b,c)
a << b	a.leftShift(b)
a >> b	a.rightShift(b)
a < b ou a > b	a.compareTo(b)
a <= b / a >= b	a.compareTo(b)

La manipulation des listes est encore plus simple en Groovy qu'en Java. Il est donc possible d'écrire par exemple ce bout de code (tiré du livre Groovy Recipes par Scott Davis).

```
def today=new Date()

(1..7).each{
println today++
}

=>
Mon Jan 25 09:31:41 CET 2010
Tue Jan 26 09:31:41 CET 2010
Wed Jan 27 09:31:41 CET 2010
Thu Jan 28 09:31:41 CET 2010
Fri Jan 29 09:31:41 CET 2010
Sat Jan 30 09:31:41 CET 2010
Sun Jan 31 09:31:41 CET 2010
```

*Note: Ce code ne marchera que sur votre machine en local en utilisant groovyConsole.*

## Passage sur le dos, parlons des listes

Avez-vous noté la notation (1..7) dans l'exercice précédent ? C'est ce que l'on appelle une *Range* en Groovy. Pratique non ? Passons à un exercice plus intéressant, et plus proche de ce que vous allez pratiquer avec Groovy d'ici quelques semaines : la manipulation des listes, des Maps et des Sets en Groovy.

### Création d'une liste

La syntaxe est simple. Une liste est créée en utilisant les crochets. C'est une ArrayList simple, ni plus, ni moins.

```
def tueilleurThemes=["Java","Apple","Scrum","Groovy"]

println tueilleurThemes
=> [Java, Groovy, Scrum, Apple]

println tueilleurThemes.size()
=> 4

tueilleurThemes << "Web 2.0"
println tueilleurThemes
=> [Java, Groovy, Scrum, Apple, Web 2.0]

// Récupérer le premier élément
println tueilleurThemes.first()
=> Java
println tueilleurThemes[0]
=> Java
```

### Itérer une liste en Groovy

Les closures sont utilisées lors de l'itération d'une liste. Voici quelques exemples d'itérations de la liste :

```
// Itération simple
def tueilleurThemes=["Java","Groovy","Scrum","Apple"]
tueilleurThemes.each{nom->
println nom
```

```

}
=>Java
Groovy
Scrum
Apple

// Version simplifié, it est une variable implicite
// Itération encore plus simple
def touilleurThemes=[ "Java", "Groovy", "Scrum", "Apple" ]
touilleurThemes.each{
  println it
}

```

Il reste possible d'itérer la liste en utilisant la syntaxe de Java 5 si vous préférez celle-ci :

```

def touilleurThemes=[ "Java", "Groovy", "Scrum", "Apple" ]
for(String item : touilleurThemes) {
  println item
}

=>Java
Groovy
Scrum
Apple

```

## Trier, ranger, manipuler une liste

Les listes sont des objets facilement manipulables. Il est possible de les trier, de les découper, de dépiler un élément comme sur une Queue, de rechercher un pattern, bref de faire un grand nombre d'opérations très pratique. Plutôt qu'un long discours, regardez attentivement ces exemples de code. Vous me ferrez 2 séries de 10, avec une grande respiration entre chaque exercice.

```

// Trier une liste avec sort
// Le tri est permanent, la liste originale est triée
def languages=[ "Java", "Groovy", "Scala", "Lisp", "Clojure", "Perl", "Python" ]
languages.sort()
languages.each{ println it }

=>Clojure
Groovy
Java
Lisp
Perl
Python
Scala

// Retourner une liste, cette fois-ci la liste originale
// ne sera pas modifiée.
def revertedList=languages.reverse()
revertedList.each{ println it }

=>
Scala
Python
Perl
Lisp
Java
Groovy
Clojure

// 3 Dépiler le dernier élément
def languages=[ "Java", "Groovy", "Scala", "Lisp", "Clojure", "Perl", "Python" ]
println languages.pop()
=> Python

// 4 Concatenation
def languages=[ "Java", "Groovy", "Scala", "Lisp", "Clojure", "Perl", "Python" ]
def newLanguages=[ "Ruby", "Goo", "Javascript" ]

languages+=newLanguages
println languages
=> [Java, Groovy, Scala, Lisp, Clojure, Perl, Python, Ruby, Goo, Javascript]

languages-=newLanguages
println languages
=> [Java, Groovy, Scala, Lisp, Clojure, Perl, Python]

Il est possible aussi de joindre une liste en une seule String, ou même de spécifier un séparateur
par exemple. Très pratique lorsque vous souhaitez écrire une ligne au format CSV à partir d'une
liste de résultats :

def languages=[ "Java", "Groovy", "Scala", "Lisp", "Clojure", "Perl", "Python" ]

println languages.join()
//JavaGroovyScalaLispClojurePerlPython

println languages.join(",")
//Java,Groovy,Scala,Lisp,Clojure,Perl,Python

```

Les closures peuvent aussi être utilisées pour rechercher un élément rapidement dans une liste. Imaginons que vous souhaitez retrouver l'ensemble des noms commençant par la lettre J :

```
def languages=["Java","Groovy","Scala","Lisp","Clojure","JRuby","Jython"]
def resultList = languages.findAll {it.startsWith("J")}
resultList.each{println it}
```

```
=>Java
JRuby
Jython
```

Les listes de chiffres sont faciles à manipuler, il est possible d'additionner la série, de rechercher la valeur maximum ou minimum :

```
def ages=[23,34,45,12,46,21,42,56,18]

println ages.max()
=> 56
println ages.min()
=> 12
println ages.sum()
=> 297
println "Moyenne: " + ages.sum()/ages.size()
=>33
```

La méthode **collect** permet de créer une nouvelle liste en concaténant un Objet à chacun des éléments de la liste initiale.

```
def ages=[23,34,45]
def resultList=ages.collect{ it += " ans"}
resultList.each{println it}

=>23 ans
34 ans
45 ans
```

### L'opérateur Spread (\*)

L'opérateur Spread (marqué avec une astérisque \*) est un opérateur courant utilisé pour exécuter une méthode sur chacun des éléments d'une liste, ou pour aussi distribuer les éléments d'une liste à une méthode. Il y a 2 variantes. L'opérateur spread seul (\*) et l'opérateur spread-dot (noté \*.). L'opérateur spread seul peut être utilisé pour passer une liste d'argument à une méthode. Il va itérer chacun des éléments et les passer un à un. C'est peu commun donc je n'en parlerai pas plus ici. L'opérateur spread-dot est plus courant. Dans l'exemple ci-dessous, le caractère \*. signifie : prend chacun des éléments de la liste et effectue un appel à la méthode toUpperCase, stocke le tout dans une nouvelle liste.

```
def languages=["Java","Groovy","Goo"]

// L'opérateur Spread dot
def resultList=languages*.toUpperCase()

// Affiche la nouvelle liste
resultList.each{println it}

=> JAVA,
GROOVY,
GOO
```

## Conclusion

Voilà c'est tout pour ce matin. Nous verrons la suite demain. Passez une bonne journée !

## Conférence Apple en direct : sortie de l'iPad

Suivez sur cette page les informations sur la conférence du 27 janvier 2010. ----- 18h24  
: nous sommes là, prêt à vous faire partager la conférence en direct 18h27 : une image du site

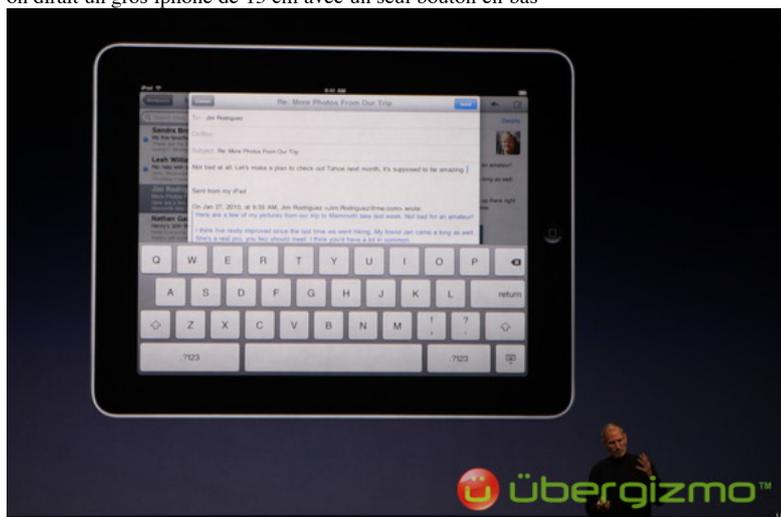


[Gizmodo](#) 18h52 : si vous voulez suivre le flux audio en temps réel : <http://www.blogsdna.com/6485/watch-live-coverage-of-apple-event-on-27th-january-2010-online.htm> 18h57 : c'est fait, je suis entré "virtuellement" [dans la salle](#) et je suis connecté au flux audio 19h00 : c'est parti ! Al Gore est là 19h01 : Entrée sur scène du principal acteur : Steve Jobs. 19h03 : 250 millions d'iPod vendus depuis sa sortie... plus de 1500 applications sur l'AppStore, 3 milliards de téléchargements d'application sur l'AppStore 19h05 : Je pense qu'il va vendre un Laptop avec une partie téléphonie... Tu iras acheter ton prochain portable avec son abonnement téléphonique. 19h08 : Steve explique qu'Apple est une compagnie qui tire beaucoup de ses revenus du mobile (iPod, iPhone) mais il manque quelque chose entre le portable et l'iphone. Pour jouer, pour lire, pour travailler... 19h10 : il taille gentiment les eeePC... 19h11 : voici la bête



19h12 :

on dirait un gros iphone de 15 cm avec un seul bouton en bas



19h15 :

démonstration du nouvel iTunes XL, de Safari, du clavier, des menus de navigation, c'est différent par rapport à l'iPhone. La salle applaudit...



19h17 :

Pas de support pour Adobe Flash sur cette version de l'iPad 19h18 : démonstration du lecteur de Mail, avec une interface complètement différente. Le Clavier virtuel promet le support et une sortie très rapide sur le marché. 19h19 : oh mais on dirait une carte du métro de Paris. C'est un PDF.



19h21 :

on entend du vieux Jazz de la Nouvelle Orléans. A priori il écoute de la musique... je ne vois pas ce qu'il fait. 19h25 : démonstration de quelques applications, l'iPad et l'application Google Maps montrent la Tour Eiffel. Ce qui est très novateur c'est cette interface, 1/3 et 2/3 où le panneau de gauche sert à montrer de la navigation, la zone centrale montre du contenu. Difficile à décrire, mais on sent que des ergonomes ont travaillé sur ce sujet. 19h29 : 700 grammes et une épaisseur de 1.3 cm !!! l'écran fait 10cm 19h30 : accéléromètre, 10 heures d'autonomie, micro, wifi, bluetooth 2.1, processeur 1Ghz Apple A4, 16Gb de mémoire et 64Gb de mémoire flash 19h33 : Scott Fortsall monte sur scène et présente l'intégration avec l'AppStore 19h33 : A propos de l'autonomie, Steve dit "I can take a flight from SF to Tokyo and watch video the whole way" (Steve Jobs), and it stays alive for 1 month on stand by mode." 19h36 : Scott joue, l'écran de l'iPad est assez grand pour jouer. Je repense à "Dungeon Hunter" développé par GameLof, un clone de Diablo, il serait parfait sur cet écran. Les jeux iPhone et iPod Touch sont 100% compatible avec l'iPad. Bonne nouvelle pour les entreprises qui ont fait ce choix pour les jeux vidéos et les applications. 19h37 Mark Hickey de Gameloft est sur scène et montre un FPS.



19h42 :

on attend le prix maintenant... La démo du jeu est sympa. 19h44 : un gars du New York Times est sur scène, présentation de la partie "ebook" de l'iPad qui ne me semble pas révolutionnaire. L'Amazon Kindle est encore tranquille. 19h45 : je n'ai pas vu de caméra par contre. 19h46 : un gars d'Electronic Arts est sur scène. Un jeu de voiture est montré. Ca bouge vite ! C'est Need for Speed Shift. Ils ont sorti une console portable ou quoi ? Je me demande s'il faut autant insister sur la partie jeu vidéo. Est-ce que l'iPad est la nouvelle console de jeu d'Apple ? Bah au bout de 10mn c'est le message... 3 éditeurs de jeux... 19h50 : un gars de MLB (Major League Baseball) présente une appli de Baseball. Une sorte de jeu de foot manager je pense... 19h53 : Steve Jobs revient et parle maintenant d'iBook et annonce la sortie d'un iBook store 19h55 : Apple attaque le marché de l'Amazon Kindle avec la sortie d'une application de lecture de livres électroniques et aussi un site marchand intégré. Mais est-ce que l'écran de l'iPad sera adapté à la lecture ?



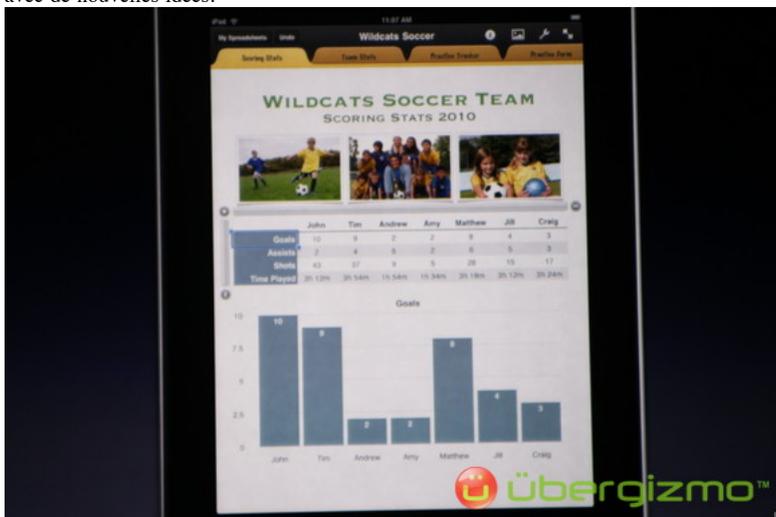
19h58 :

Steve parle maintenant d'une version d'iWork pour iPad, avec donc un clone d'Excel, de Word et de Publisher. J'ai hâte de faire mes présentations avec Keynote au Paris JUG ! 19h59 : le prix des livres n'est pas donné, dans les 15\$. Certes moins cher qu'un vrai livre, mais pas donné. 20h00 : le logiciel iBook pour lire les livres numériques permet de passer d'un affichage couleur à un affichage monochrome, et l'interface fait très "vrai livre". 20h01 : avec une version d'iWork, je pense que l'iPad va gagner en entreprise des parts de marchés très rapidement. Comme un portable, en plus pratique, en plus simple. C'est vraiment ce qui est le plus intéressant d'un point de vue business/positionnement. Demain, vous pourrez faire vos présentations avec l'iPad, travailler sur vos documents commerciaux, c'est plus léger qu'un portable, et peut-être adapté aux usages nomades de l'entreprise, là où l'iPhone montre ses limites.



20h05 :

c'est fait, ça a marché, j'ai envie de l'acheter 20h08 : Microsoft est mal. C'est un appareil qui va créer un nouveau marché, entre le full-nomade et le PC portable. Là encore Apple est innovant avec de nouvelles idées.

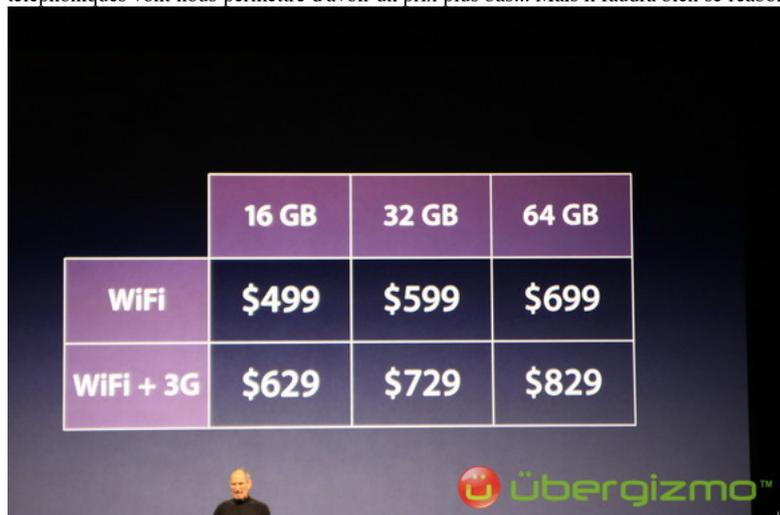


20h09 :

une rumeur parle de 100 USD, cela me semble très bas. 20h10 : Nous devrions parler du prix 20h12 : Steve Jobs parle maintenant de la partie Synchronisation entre l'iPad et le reste de vos iTrucs. 20h13 : Oui il y a de la 3G. Les iPads ne seront pas lockés à un opérateur. Pour les prix

c'est un peu compliqué. 20h18 : **Les Prix** sont assez simples, il existe plusieurs versions de l'iPad. La version Wifi seul + 16Gb est à \$499, la version la plus chère à 64Gb+3G est à \$849. C'est cher je trouve, mais c'est presque le prix de l'iPhone à sa sortie. J'espère que les opérateurs téléphoniques vont nous permettre d'avoir un prix plus bas... Mais il faudra bien se réabonner non ?

	16 GB	32 GB	64 GB
WiFi	\$499	\$599	\$699
WiFi + 3G	\$629	\$729	\$829



20h24

Steve Jobs montre un accessoire pratique, un vrai clavier qui se branche en dessous.



20h25

La version Wifi de l'iPad sera disponible dans 60 jours, vers début mars donc. Il faudra attendre un peu plus pour la version 3G. Je pense que la version 3G n'est pas très intéressante. Et mince, pourquoi Apple ne veut pas de tethering pour partager le modem de l'iPhone avec l'iPad ?



20h25 :

Présentation d'un film qui sera sans doute en ligne [sur le site d'Apple](#) dans quelques heures.



20h33 :

Steve Jobs termine la présentation qui fut assez impressionnante. *Apple has 125M account with credit cards on file. Users have downloaded over 12 billion products on these stores.* En fait il y a peu de chances que l'iPad ne marche pas. C'est cher, mais ce qui est bien est cher. Donc préparez environ 450 EUR pour avoir un appareil et s'en servir... 20h39 : **Fin de la présentation.** Je rends le micro et je pars en acheter un.... A bientôt

## 2ème Anniversaire du Paris JUG le 9 février 2010



Mardi 9 février à 19h00, nous organisons l'anniversaire du Paris JUG avec toute la "Crew" (7 personnes) du Paris JUG. Cette année, toute l'équipe a travaillé pour vous proposer un événement très sympa. La soirée sera placée sur le thème de l'Open-Source. Avec en première partie, une Keynote exceptionnelle par Sacha Labourey, puis ensuite des présentations courtes sur différents projets Open-Source, sur différents acteurs de la communauté Java. La traditionnelle 3ème mi-temps cette année a demandé aussi une organisation particulière. Nous avons réussi à privatiser un restaurant, et nous avons mis en place un système afin de régler au plus tôt le restaurateur, ce qui nous permettra de recevoir tout le monde, et ce qui vous permettra de profiter de la soirée. Attention, le nombre de places est limité, que ce soit à la soirée elle-même, ou ensuite le buffet dinatoire. Les inscriptions sont ouvertes depuis samedi 16h00, vous pouvez aussi nous suivre via [le compte Twitter du parisjug](#). Pour la première fois, il y aura aussi un espace pour quelques exposants. L'idée est aussi de vous proposer d'aller discuter et de rencontrer les geeks qui font les soirées du Paris JUG avec vous, et pourquoi pas, d'aller travailler avec eux. Il y aura aussi des concours et des lots à gagner, chacun de nos partenaires s'est investi pour que cela reste "geek" dans l'esprit du Paris JUG. Grâce à leur aide et à leur support financier au Paris JUG, cela permet de louer cette année une grande salle de plus de 500 places. Rendez-vous mardi 9 février. S'il n'y a qu'une soirée à faire en février, c'est celle-ci. Nous comptons sur vous pour en parler autour de vous cette semaine ! **Pour en savoir plus :** Le programme détaillé de la soirée : <http://www.parisjug.org/xwiki/bin/view/Meeting/20100209> L'inscription et le paiement pour le repas sont expliqués [sur cette page](#). **Si vous souhaitez**



Anniversaire Paris JUG

réutiliser des logos pour vos blogs, servez-vous :



## Fin de ma mission en Startup, 2 mois passionnants à vivre



J'ai terminé ma mission dans la Startup pour laquelle je travaillais depuis début décembre 2009. Je reprends mon activité d'indépendant après 2 mois intenses, consacrés à démarrer la partie technique de cette jeune entreprise. Je n'ai finalement pas voulu continuer en tant que salarié. Je revivrai cette aventure un jour, car je reste intéressé par les projets d'entrepreneurs et par l'envie de créer quelque chose. J'ai travaillé 2 mois, sans compter mes heures, ce qui m'a donné le temps de lancer la partie technique du projet, de mettre en place une architecture en collaboration avec une autre personne et de préparer un plan pour l'année. En partant l'autre jour, j'ai laissé un prototype avancé codé avec [Grails, un système de gestionnaire de code sources et de tickets](#), des serveurs configurés et installés, bref de quoi permettre à ceux qui reprennent le projet de travailler. Je pense que les jeunes entreprises ont besoin de personnes comme nous : **un Amorceur de Projet** ou peut-être "**Technical Project Leader**". C'est quelqu'un qui vient surtout avec son expérience et ses connaissances générales, plus que grâce à son profil technique. Votre mission, si vous l'acceptez, c'est de construire rapidement les bases d'une architecture, d'arbitrer entre différentes solutions en codant des prototypes et de faire passer le maximum de connaissances à l'équipe. J'ai bien aimé cette expérience, et je serai prêt à recommencer. Avec ou sans Scrum, peu importe. L'essentiel est que le ratio temps-passé/quantité d'informations soit au max. J'ai préféré ne pas continuer en tant que salarié. Manque de confiance de la part des dirigeants, écart culturel entre une vision scrum-startup-java-fun et une vision plus classique du développement logiciel. Je respecte complètement l'expérience des autres, ce qu'ils ont fait. Mais ce n'est pas ce que j'ai envie de faire. Ce n'était pas la méthode, bref il était temps d'arrêter avant d'aller plus loin. En attendant, je profite de février pour suivre une formation, travailler sur un projet annexe personnel, et rencontrer du monde. Je vous parlerai de mon projet professionnel sur le blog plus tard. Voilà, pas besoin d'en dire plus. Je recommande l'expérience aux Consultants Java +10 ans d'expérience en mal de sensations fortes. Vous en sortirez plus humble, plus fort, moins consultant, plus fatigué et enfin avec des tonnes d'histoires à raconter. La Startup c'est le Botox du vieux développeur en quelques sortes. Bonne continuation à l'équipe LC, c'était sympa.

## Et si le Cloud n'était qu'un nuage ?



Ce matin en prenant le RER, je suis encore tombé sur un article sur le "*Cloud Computing*". Je feuillette les pages, toujours les mêmes explications, le même emballement, bref à la lecture de ces quelques pages, on sent qu'il faut y aller. L'article parle d'Amazon EC2, et de Microsoft Azure. Et tout d'un coup, je me suis demandé si cet emballement n'était pas disproportionné, si cette excitation n'allait pas nous entraîner dans le mur. Je me suis demandé si nous n'étions pas à l'aube d'une sorte de crise du Cloud Computing... Voyons tout d'abord Amazon avant de lister les autres points d'inquiétude. **AWS ne représente que 4,28% du chiffre d'affaire 2009 d'Amazon** Amazon a annoncé dans ses résultats 2009 que le Cloud Computing via son service avait réalisé un chiffre d'affaire de 550 Millions de dollars sur l'année.

Cela peut sembler beaucoup mais ce n'est que 4,28% de 12 828 millions de dollars pour l'année 2009. Et encore je parle de la section "Other" [dans le rapport financier complet disponible sur Yahoo Finance](#) pour l'Amérique du Nord. 4% du chiffre d'affaire d'Amazon pour ce qui n'est pas vente de médias ou d'appareils électroniques pour 2009. La contribution au chiffre d'affaire est vraiment minoritaire, même si nous parlons de quelques 195 millions de dollars, uniquement pour les 3 derniers mois de 2009. Ces chiffres peuvent être vus de 2 façons. L'optimiste dira qu'Amazon est entrain de rentabiliser ses infrastructures. Le chiffre de 550 M\$ a été réalisé à 40% sur les 3 derniers mois de l'année 2009. Cette croissance du chiffre d'affaire est peut-être un indicateur que l'activité AWS dégage du chiffre. Le pessimiste relativise la contribution de l'activité Cloud Computing en regard du CA total sur 2009. Il ira même jusqu'à se demander, si ce ratio faible, qui n'est pas le coeur de l'activité d'Amazon, vaut le coup et les efforts mis en oeuvre. Pourtant à 200 millions de dollars par trimestre, il serait dommage de se priver de ce revenu. **Les systèmes sont opaques** Il y a quelque chose de flagrant, que ce soit Google App Engine ou Azure, ces systèmes sont plutôt opaques. Il y a une part de mystère, de choses que l'on ne vous dit pas techniquement. C'est un mal nécessaire pour assurer la montée en charge, mais c'est inquiétant. Je pense qu'il y a un manque de transparence sur les performances. Nous avons entendu des présentateurs nous parler de spécifications techniques. Mais nous n'avons pas entendu de comparaison de sites Internet à fort trafic, version AWS et version classiques. Et je trouve cela inquiétant. Ne pas savoir c'est accepter de prendre un risque. **La sécurité est-elle vraiment au rendez-vous ?** Imaginons un scénario catastrophe. Un jeune hacker trouve une faille dans l'API qui contrôle le Cloud Computing. Comme votre site est hébergé sur le cloud, c'est tout votre système d'information qui s'en trouve alors fragilisé. Je pense que finalement, votre réseau hébergé chez un prestataire est moins exposé que votre application sur Google App Engine. Lorsque vous décidez de basculer sur le Cloud une partie de votre infrastructure, vous allez aussi automatiquement devenir plus vulnérable aux failles de la plateforme elle-même. C'est un peu comme faire le choix de l'open-source. Vous êtes plus exposé mais en retour, les corrections de bugs sont plus rapides. Mais je ne pense pas que ce paradigme s'applique au Cloud, où vous êtes dépendant de l'infrastructure et des équipes de développement des fournisseurs du Cloud. **Tout a un prix** Lorsque l'on parle de Cloud Computing, le mot "économie" arrive rapidement dans la conversation. Or les premiers prix d'Amazon EC2 ne sont pas compétitifs. A 61\$ par mois, c'est cher. Pour 20 EUR par mois, vous avez une machine dédiée chez OVH, tout à fait correcte. En janvier pour héberger 2 serveurs, nous avons pris Gandi et son offre virtualisée. J'avoue avoir été déçu par les performances par rapport au prix, pour une machine sur laquelle j'avais installé Apache, MySQL, Tomcat et une appli Grails. Avec 6 parts, j'ai trouvé l'ensemble poussif. De plus, Tomcat avait un comportement bizarre sur cette machine virtualisée. Et là, aucuns moyens de comprendre ce qui se passe vraiment. Le Cloud Computing est plus cher qu'une solution classique d'hébergement dédié. La flexibilité n'est peut-être pas une option indispensable à votre projet. **Aucunes garanties de performance, des SLAs au rabais** Continuons notre réflexion. Vous payez plus cher un service dont les performances ne sont pas garanties. Ne pensez-vous pas que c'est inquiétant ? Pour aller plus loin, le SLA (Service Level Agreement) [d'Amazon EC2](#) s'engage à 99,95% et ne vous rembourse que de 10% si le service est en dessous. Le pire est que les temps de maintenance de la plateforme AWS sont exclus de ce SLA, ce qui revient à dire que vous acceptez que votre site tombe le temps d'une mise à jour d'Amazon EC2. Cela me dérange, je n'ai pas de visibilité, c'est inquiétant. **Coder moins pour gagner plus** Google App Engine est un outil magnifique, qui vous demande seulement de coder avec 7 doigts au lieu de 10 en faisant une croix sur un certain nombre d'API Java. Regardez [la liste des APIs que vous pouvez utiliser](#). Plutôt impressionnant non ? Dans la pratique, il y a pas mal de librairies qui ne marchent pas sur Google App Engine. Ce qui revient à dire que vous allez devoir adapter votre solution et votre code pour que celui-ci fonctionne. Encore un nuage de béatitude qui s'évapore, sitôt la mariée allongée dans le lit. Oui tu vas avoir de grand moments de solitude lorsque tu ne comprendras pas pourquoi le framework XXX ne fonctionne pas sur Google App Engine. **Votre application démarre lentement** Le même Google App Engine souffre d'un mal sournois. Lorsque vous visitez une application comme [celle-ci](#) qui n'a pas beaucoup de visiteurs, vous notez un chargement assez long la première fois. Il faut savoir que GAE démarre et alloue votre application que lorsqu'il y a un visiteur. Et qu'en l'absence de ces visiteurs, votre application est éteinte. Cela pose problème, et des discussions intéressantes sur la liste de diffusion de Gaelyk par exemple, montre que le problème est connu chez Google. En attendant, moi en tant que client je trouve cela nul. Et en tant que développeur je suis frustré. Et on a beau s'exciter devant GAE, il faut aussi parler de ce qui n'est pas top. **Vous allez faire du NoSQL** Ceux qui vendent cette idée sont des consultants comme moi, des gars bien qui veulent faire leur métier. Soyons méchant 3 minutes : après avoir tartiné tout le monde avec SOA, il semble que le nouveau Mojo s'appelle Cloud Computing. Et je pense à NoSQL. La majorité de ceux qui s'y intéresse, moi le premier, sont des jeunes de moins de 40 ans qui n'ont connu que le Web et que la base relationnel. Les anciens rigolent gentiment en nous voyant nous emballer pour une techno (le non-relationnel) qui existe déjà depuis 25 ans. Alors oui je serai là le mardi 16 février [à la 2e réunion du NoSQL User Group](#) car je suis emballé par les idées et ces nouvelles architectures, mais je suis inquiet à l'idée que certains vont en vendre à leurs clients, en n'ayant finalement aucunes expériences sur le sujet. Ce n'est pas méchant, c'est un fait. Il est plus facile de vendre quelque chose de nouveau mal connu que quelque chose de vieux très connu. SOA c'est hasbeen, passez au Cloud, camarade. **Conclusion qui scale** Avons-nous tous besoin du Cloud ? Oui pour mes emails sur Gmail. Certainement oui pour mon infrastructure Web. Peut-être pas pour mes applications d'entreprise. J'essaye de voir si les grandes idées du Cloud Computing ne cachent pas de futurs gros soucis, que ce soit sur la montée en charge, les promesses de la performance, le réseau qui tient la charge ou la sécurité. Peut-être qu'Amazon va simplement exploser dans 4 mois et que beaucoup de sites mettront la clé sous la porte. Ou peut-être que Kevin, 14 ans, arrivera à faire casser votre clé de cryptage, et que tous vos dossiers clients seront sur Internet jusqu'à la fin des temps. Ou peut-être que je me suis endormi dans le RER et que tout ceci n'était qu'un cauchemar. **Références** [Rapport financier d'Amazon 2009 noSql User Group](#) de Paris, prochaine soirée mardi 16 février chez OCTO Technology [Larry Ellison critique le Cloud Computing en septembre 2008](#)

## Ajout du Flux RSS des commentaires

J'ai ajouté un lien vers les flux RSS des commentaires du blog en bas de la page du Touilleur Express. L'url directe est <http://www.touilleur-express.fr/comments/feed/> pour ceux qui lisent ce message à travers Google Reader. Cela vous permet de suivre les commentaires des lecteurs du Touilleur Express. D'ailleurs je trouve que parfois vos commentaires sont plus intéressants que mon article. J'envisage donc de ne plus trop écrire et de vous laisser bosser un peu. Allez, au travail les gars, vous avez quelques 700 personnes abonnées au flux RSS comme le prouve cette image réalisée sans trucage sous contrôle d'huissier :

Google outils pour les webmasters

Flux	Abonnés Google	Envoi d'un flux
<a href="http://www.touilleur-express.fr/feed/">http://www.touilleur-express.fr/feed/</a>	716	Oui
<a href="http://www.touilleur-express.fr/feed/atom/">http://www.touilleur-express.fr/feed/atom/</a>	95	Oui, s
<a href="http://www.touilleur-express.fr/wp-rss2.php">http://www.touilleur-express.fr/wp-rss2.php</a>	60	Oui, s
<a href="http://www.touilleur-express.fr/podcast/podcasts_mp3.xml">http://www.touilleur-express.fr/podcast/podcasts_mp3.xml</a>	10	Oui, s
<a href="http://www.touilleur-express.fr/feed/rss/">http://www.touilleur-express.fr/feed/rss/</a>	3	Oui, s

Voilà, continuez, vous avez dépassé aussi les 1000 commentaires depuis longtemps, et moi j'arrive péniblement à mon 500e billet, prévu vers avril 2010 au rythme actuel.

## Groovy++, une version typée statiquement de Groovy

Mi-décembre, Alex Tkachman et Eugene Vigdorichik ont lancé un projet appelé Groovy++ (prononcez Groovy Booster). L'idée de ce projet est de créer une version typée statiquement de Groovy afin d'obtenir de meilleures performances et la vérification à la compilation de votre programme, tout en gardant l'expressivité et la souplesse de Groovy. Cela donne aussi un boost en terme de performances, sachant que l'idée est aussi de générer du code optimisé capable d'utiliser finement les bibliothèques Java existantes. Les auteurs parlent même de surpasser Java, ce qui semble un peu surprenant au premier abord. Alex dit :

To make a long story short "statically typed Groovy" (aka Static Groovy, aka Groovy Booster, aka Groovy++) is an attempt to combine performance and compile-time checks with the expressiveness of the Groovy programming language and to develop high-performant runtime libraries utilizing this combination.

Il ne s'agit pas d'un nouveau langage, mais d'une option capable d'accélérer les applications Groovy comme Grails d'ailleurs. Pour cela il suffit d'ajouter un JAR et d'annoter ses classes Groovy avec @Typed pour bénéficier de l'effet Groovy++. Voici un script Groovy écrit par Alex qui parcourt une arborescence de fichier, ouvre chacun des articles, compte le nombre d'occurrence de différents mots et stocke le tout dans 2 fichiers triés par ordre croissant et décroissant. Il y a en fait 2 tâches : la collecte des mots d'une part, et le rangement d'autre part.

```
package org.mbte.groovypp.examples.wordcount

for (i in 0..<10) {
    t1 = System.currentTimeMillis()
    counts = [:]
    new File("./20_newsgroups").eachFileRecurse{ f ->
        if (!f.isDirectory() && !f.path.contains(".svn")) {
            f.text.toLowerCase().eachMatch(/\w+/) { w ->
                def c = counts[w] ?: 0
                counts[w] = c + 1 } } }
    new File("counts-desccreasing-groovy").withWriter { out ->
        counts.sort { a, b -> b.value <=> a.value }.each { k, v -> out << "$k\t$v\n" } }
    new File("counts-alphabetical-groovy").withWriter { out ->
        counts.sort { it.key }.each { k, v -> out << "$k\t$v\n" } }
    println "Finished in ${System.currentTimeMillis() - t1} millis"
}
```

D'après les tests d'Alex sur un MacBookPro, ce script tourne en moyenne en 55 secondes avec un volume de 25 Mb. Les détails et le code sont disponibles [ici](#) si vous voulez creuser un peu plus. Avec Groovy il est possible de traiter en parallèle la partie collecte des mots, et la partie comptage-reporting. Voici la version avec l'annotation @Type de Groovy++ qui s'exécute un peu plus rapidement, mais pas comme en 5.5 secondes comme dit Alex sur son blog. Je ne suis pas d'accord avec ses chiffres.

```
@Typed package org.mbte.groovypp.examples.wordcount

import java.util.concurrent.*
import groovy.util.concurrent.*

for (i in 0..<10) {

    def t1 = System.currentTimeMillis()
    def pool = Executors.newFixedThreadPool(30)
    def counts = new AtomicIntegerMap ()
```

```

new File("./20_newsgroups").recurseFileIterator().filter{ file ->
    !file.directory && !file.path.contains(".svn")
}.each(pool,4) { file ->
    file.charSequence.eachMatch(/\w+/) { String w -> w = w.toLowerCase()
        counts[w].incrementAndGet ()
    }
}.whenBound {
    pool.execute {
        new File("counts-decreasing-groovy").withWriter { Writer out ->
            counts.asList().sort { a, b -> b.get() < => a.get() }.each { e -> out << "$e.key\t${e.get()}\n" }
        }
    } {
        new File("counts-alphabetical-groovy").withWriter { Writer out ->
            counts.asList().sort { a, b -> b.key < => a.key }.each { e -> out << "$e.key\t${e.get()}\n" }
        }
    }
    pool.shutdown()
}

pool.awaitTermination(30,TimeUnit.SECONDS)
println "Finished in ${System.currentTimeMillis() - t1} millis"
}

```

Notez l'utilisation de la class [Executors](#) et de la mise en parallèle du traitement. Une seule Thread principale démarre et itère chaque fichier. Une fois le traitement de celui-ci terminé, après compté le nombre de mot, la génération des rapports est déléguée à 4 threads en parallèle. Sympa non ? Voici enfin pour terminer la version Java:

```

public class JavaWordCount {
    public static void main(String[] args) throws IOException {

        for (int i = 0; i < 10; ++i) {
            Comparator<Map.Entry<String, Integer>> sortByCount = new Comparator<Map.Entry<String, Integer>>() {
                public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2) {
                    return o2.getValue() - o1.getValue();
                }
            };
            Comparator<Map.Entry<String, Integer>> sortByElement = new Comparator<Map.Entry<String, Integer>>() {
                public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2) {
                    return o1.getKey().compareTo(o2.getKey());
                }
            };
            Long timeStart = System.currentTimeMillis();
            Pattern wordPattern = Pattern.compile("\\w+");
            File rootDir = new File("./20_newsgroups");
            CountingSet counter = new CountingSet();
            for (File groupDirectory : rootDir.listFiles()) {
                if (groupDirectory.isDirectory() && !groupDirectory.getPath().contains(".svn")) {
                    for (File f : groupDirectory.listFiles()) {
                        if (f.isFile() && !f.getPath().contains(".svn")) {
                            BufferedReader reader = new BufferedReader(new FileReader(f));
                            String line;
                            while ((line = reader.readLine()) != null) {
                                Matcher matcher = wordPattern.matcher(line);
                                while (matcher.find()) {
                                    counter.add(matcher.group().toLowerCase());
                                }
                            }
                            reader.close();
                        }
                    }
                }
            }

            Writer pw = new BufferedWriter(new PrintWriter("./counts-alphabetical-java.txt"));
            sortAndDisplay(counter.entrySet(), sortByElement, pw);
            pw.close();

            pw = new BufferedWriter(new PrintWriter("./counts-decreasing-java.txt"));
            sortAndDisplay(counter.entrySet(), sortByCount, pw);
            pw.close();

            System.out.println("Finished in " + (System.currentTimeMillis() - timeStart) + " ms");
        }

        private static void sortAndDisplay(Set<Map.Entry<String, Integer>> set, Comparator<Map.Entry<String, Integer>> comp, Writer writer) {
            List<Map.Entry<String, Integer>> list = new ArrayList<Map.Entry<String, Integer>>(set);
            Collections.sort(list, comp);
            display(list, writer);
        }

        private static void display(Iterable<java.util.Map.Entry<String, Integer>> list, Writer writer) throws IOException {
            for (Map.Entry<String, Integer> entry : list) {
                writer.write(entry.getKey() + " : " + entry.getValue() + "\n");
            }
        }

        private static class CountingSet extends LinkedHashMap<String, Integer> {
            void add(String s) {
                Integer i = get(s);
                put(s, (i == null) ? Integer.valueOf(1) : Integer.valueOf(i + 1));
            }
        }
    }
}

```

Le code Java n'est pas particulièrement optimisé, mais il a le mérite de montrer plusieurs choses. Tout d'abord l'utilisation des APIs Java, ensuite le fait que **Java n'est pas adapté à ce type de traitement** lorsque l'on regarde le code source. Oui cela fonctionne, mais ce n'est pas simple à lire. **Mes propres tests** Du côté des performances voici mes propres résultats avec une arborescence de fichiers Groovy d'un projet assez important. Comptons tout d'abord sous Unix le nombre de fichiers :

```
nicolas@macbookpro> find . -type f | wc -l
2467
```

Cela représente un volume de 38 Mo de fichiers Java, Groovy, de pages JSP, bref un projet complet, tel qu'un outil de compilation comme Maven ou Graddle pourrait devoir traiter par exemple. J'ai fait 3 tirs de la version Java. Chaque tir est une série de 10 tests. Le premier test est assez lent car le système fait réellement un accès disque pour charger les fichiers, le test s'exécute en 190ms en moyenne pour la version Java. J'ai donc supprimé à chaque fois pour l'ensemble de mes tests la valeur la plus lente et la valeur la plus rapide, afin de lisser les échantillons de tests. Voici le script de la version Groovy++ que j'ai légèrement modifié :

```
@Typed(TypePolicy.MIXED)
package test
class Boosted {
    static main(args){
        for (i in 0..<10) {
            def t1 = System.currentTimeMillis()
            def counts = [:]
            new File(".").eachFileRecurse
            {
                f ->
                if (!f.isDirectory() && !f.path.contains(".svn"))
                {
                    f.text.toLowerCase().eachMatch(/\w+/) { w ->
                        def c = counts[w] ?: 0
                        counts[w] = c + 1
                    }
                }
            }
            new File("counts-decreasing-groovy").withWriter { out ->
                counts.sort { a, b -> b.value < => a.value }.each { k, v -> out << "$k\t$v\n" }
            }
            new File("counts-alphabetical-groovy").withWriter { out ->
                counts.sort { it.key }.each { k, v -> out << "$k\t$v\n" }
            }
            println "${System.currentTimeMillis() - t1}"
        }
    }
}
```

Groovy++ est distribué sous la forme d'un ZIP. Il suffit de le décompresser, de remplacer sa variable GROOVY\_HOME pour qu'elle pointe vers cette version, de mettre à jour son PATH et c'est tout. J'ai eu à augmenter les options Java pour résoudre un problème de OutOfMemory (export JAVA\_OPTS="-Xmx512m") mais à part cela, c'est du Groovy classique. Voici la moyenne des résultats ma machine (MacBookPro 2.4Ghz Intel Core 2 Duo, 4GB) pour la version Java, la version Groovy et la version Groovy++ Java 1.6.0.17 64-bits : 58,71 ms Groovy 1.6.0 : 5839,58 ms Groovy ++ 0.1.12: 3858,75 ms Mes propres tests montrent que **la version Groovy++ est plus rapide de 34% par rapport à la version Groovy**. Java reste largement plus rapide que la version Groovy et la version Groovy++. Tout ceci est fait avec un micro-test, ce qui n'est pas suffisant pour dire que l'un est plus rapide que l'autre. Mais cela me refait penser à Graddle, un outil de build Groovy, très sympa à utiliser, mais pour lequel j'avais une réserve à priori, sans avoir effectué de tests. Voyons enfin ce que cela donne avec la version 4 Threads d'Alex. Le script est modifié afin d'effectuer la partie reporting en parallèle.

```
@Typed(TypePolicy.MIXED)
package test

import java.util.concurrent.*
import groovy.util.concurrent.*

class BoostedOptimized {
    static main(args){
        for (i in 0..<10) {

            def t1 = System.currentTimeMillis()
            def pool = Executors.newFixedThreadPool(30)
            def counts = new AtomicIntegerMap ()

            new File(".").recurseFileIterator().filter{ file ->
                !file.directory && !file.path.contains(".svn")
            }.each(pool,4) { file ->
                file.charSequence.eachMatch(/\w+/) { String w -> w = w.toLowerCase()
                    counts[w].incrementAndGet ()
                }
            }.whenBound {
                pool.execute {
                    new File("counts-decreasing-groovy").withWriter { Writer out ->
                        counts.asList().sort { a, b -> b.get() <=> a.get() }.each { e -> out << "$e.key\t${e.get()}\n" }
                    }
                } {
                    new File("counts-alphabetical-groovy").withWriter { Writer out ->
                        counts.asList().sort { a, b -> b.key <=> a.key }.each { e -> out << "$e.key\t${e.get()}\n" }
                    }
                }
                pool.shutdown()
            }
        }
    }
}
```

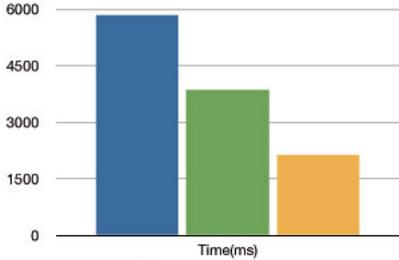
```

        pool.awaitTermination(30,TimeUnit.SECONDS)
        println "${System.currentTimeMillis() - t1}"
    }
}

```

J'ai effectué 3 séries de 10 tests, voici la synthèse: Java 1.6.0.17 64-bits : 58,71 ms Groovy 1.6.0 : 5839,58 ms Groovy ++ 0.1.12 : 3858,75 ms Groovy++ 4 Threads : 2129,17ms

	Time(ms)	% plus rapide
Groovy	5839,58	100
Groovy++	3858,75	33,92
Groovy++ 4Threads	2129,17	63,54



La version 4 Threads est donc 63,54% plus rapide que la version Groovy de base. Cependant je pense que c'est une erreur de les comparer, l'algo étant différent. Nous devons simplement retenir l'écart entre la version Groovy standard et la version exécutée avec Groovy Booster qui est de 33%. C'est déjà beaucoup. Groovy++ est un projet intéressant, qui montre qu'il est possible d'accélérer Groovy. Je serai intéressé de savoir le retour de Guillaume Laforge sur le sujet. Il faut aussi noter qu'il y a quelques bémols, l'utilisation de Groovy++ empêche d'utiliser quelques mots clés comme l'opérateur de transtypage "as" ("-1" as Integer par exemple). Enfin pour terminer, d'après les billets que j'ai vu, la version Scala est exactement aussi rapide que la version Java, il n'y a aucunes différences. C'est aussi l'avantage d'un langage avec typage fort, nous reparlerons de Scala bientôt. Les tableaux de résultat :

Java tir 1	Java tir 2	Java tir 3	Groovy tir 1	Groovy tir 2	Groovy tir 3	Groovy++ sans @Typed
128	106	172	6118	6131	6193	6628
99	129	54	5947	5959	6024	6488
69	56	28	5906	5822	5809	6364
40	27	56	5742	6039	5928	6490
56	36	46	5738	5689	5701	6336
26	29	28	5747	5676	5698	6263
30	30	58	5714	5649	5740	6260
49	40	17	5831	5651	5698	6268
			62,125	56,625	57,375	58,71
			5842,875	5827	5848,875	6387,125
					<b>5839,58</b>	<b>6387,125</b>
						4160,4166666667
						6141,25
						9941,2916666667
Groovy++ tir 1	Groovy++ tir 2	Groovy++ tir 3				
4258	4131	3922				
3845	4056	3935				
3739	3954	3781				
3790	4017	4023				
3718	3807	3814				
3736	3746	3756				
3711	3791	3709				
3723	3809	3839				
			3815	3913,875	3847,375	3858,75

Sources [Mail d'Alex à la liste Grails](#) [First preview of static compiler for Groovy is available](#) [How come that Groovy++ overperform Java?](#)

## Anniversaire du Paris JUG : petit mot à la pause

Je vous écris ces quelques lignes pendant la pause, la soirée se passe bien, avec quelques 400 personnes, des sponsors, des speakers et un invité surprise de dernière minute, en la personne de Marc Fleury, ex-CEO de JBoss. Voici le clip qui a été diffusé au début de la soirée : [Clip du deuxieme Anniversaire du Paris JUG](#)

Olivier

Croisier, blogger du Coder's Breakfast, couvre en "live" la conférence via Google Wave, directement sur son site. Allez-voir [les discussions sur son site](#), c'est très intéressant. Je vous laisse, il reste du boulot et la soirée n'est pas terminée !

## Retour sur le 2eme anniversaire du Paris JUG

La soirée anniversaire du Paris JUG était place sous le signe de l'Open Source en France. Après une présentation d'une heure, un buffet et ensuite quelques 8 présentations, nous nous sommes tous retrouvés dans un restaurant, puis une brasserie, pour terminer tranquillement à 05h00 ce matin. Je ne vous cache pas que taper un texte ce matin sur un clavier où, en plus les lettres ne sont pas classées par ordre alphabétique, relève de l'exploit. Mais je suis là. Après [une vidéo pour lancer la soirée](#), Sacha Labourey monte sur la scène d'un amphithéâtre de 500 places presque plein. Le thème de sa présentation sera l'open-source et son retour sur son expérience en tant que CTO chez JBoss-RedHat jusqu'en 2005. A la question "la révolution Open-Source a-t-elle eu lieu ?", la réponse est "Oui ! mais...". Il est clairement établi qu'Internet a joué un rôle de catalyseur, en permettant de développer à plusieurs, que l'on soit à San Francisco ou Beijing. C'est bien le développement logiciel qui a finalement révolutionné l'open-source, plus que les systèmes d'exploitations libres. Sacha explique ensuite que l'Open-Source apporte au monde de l'entreprise des méthodes de travail. La revue de code, les wikis, le pair-programming, sont des techniques issues du développement du monde de l'open-source. Il cite aussi l'exemple des gestionnaires de code source comme SVN qui n'utilisent pas de locks sur les fichiers, contrairement à Microsoft Visual Source Safe. Le développement libre a donc apporté à l'entreprise de nouvelles techniques de développement, ce dont je suis tout à fait d'accord. Nous pouvons extrapoler et penser que les gestionnaires de version décentralisés [comme Git](#) vont prendre de l'ampleur dans les années qui viennent par exemple. La méritocratie reconnaît les personnes sur la valeur de ce qu'elles produisent, pas sur leur intitulé de poste ou leurs diplômes. Et quelque part finalement, c'est bien ce qui fait la qualité des grands projets open-source. Ne pas utiliser aujourd'hui de librairies FOSS (Free and Open-Source Software) est une aberration. Nous pensions il y a 10 ans qu'en conservant son code fermé, nous serions plus compétitifs. C'est en fait l'inverse. En ouvrant son code, des contributeurs externes motivés par le projet peuvent venir le renforcer et vous aider. L'open-source a aussi permis de lutter contre les brevets logiciels. A partir du moment où du code est libre de droits, il n'y a plus de possibilités ou d'intérêts à le breveter. Rappelons qu'un Brevet Logiciel n'est ni plus ni moins qu'un péage pour se servir une rente viagère, ce qui est tout simplement merdique. Les projets FOSS permettent aussi aux développeurs de tout horizon de travailler ensemble. Sacha a parlé d'une personne que j'ai eu le plaisir de rencontrer en février 2005, il y a 4 ans : Mladen Turk. Venu de Croatie, développeur sur mod\_jk, c'était un personnage que j'avais croisé lors de ma formation JBoss (voir les vieux articles [ici](#) et [là](#)). Aujourd'hui eXo Platform applique un nouveau mode de recrutements, où les développeurs sont co-localisés, mais les chefs de produits, les évangélistes et les leads-développeurs sont partout dans le monde. C'est peut-être un nouveau modèle qui reflète aussi l'évolution des acteurs du monde de l'open-source ? Sacha explique que le développement open-source est un catalyseur de l'innovation, un vecteur de partage de la connaissance. Lorsque tu mets ton code sur Internet, et que tu te fais scanner par plusieurs personnes à l'autre bout de la Terre, il y a un moment assez curieux à passer. Dans le monde de l'entreprise, les revues de codes ne sont pas si fréquentes finalement, là où dans le monde de l'Open-Source, il y a toujours quelqu'un qui peut regarder votre code à 04h du matin. Lorsque l'on regarde le marché du développement logiciel propriétaire, nous pouvons constater une certaine inertie. Que ce soit sur la sortie de nouvelles versions, la correction de bugs, il y a un effort à fournir de la part de l'éditeur. Cet effort existe aussi dans le monde de l'open-source, mais tu es libre de proposer une correction à tout moment si tu désires avancer. Les projets FOSS font plus de versions, il y a des sorties de logiciel le 31 décembre comme un projet cité dans la soirée, bref la vie d'un projet FOSS est plus attractive qu'un produit fermé. Je ne pense pas que cela soit forcément mieux, et qu'un excès de versions n'est pas non plus un gage de qualité pour les utilisateurs finaux. Pour terminer, Sacha Labourey s'adresse au développeur, à l'indépendant et à l'entrepreneur. **Si je suis développeur salarié**, un projet open-source est un excellent moyen d'augmenter mes compétences en travaillant sur ce qui me plaît. Que ce soit en recevant du retour des autres, en se mettant en avant, en défendant ses idées, en argumentant, c'est un moyen d'acquérir une notoriété, de se faire plaisir et pourquoi pas, de faire du télétravail dans l'endroit de ses rêves. **Contribuer à un projet Open-source devrait être obligatoire !** Il nous invite à nous lancer, à contribuer à l'Open Source. En tant qu'indépendant l'Open Source est un bon moyen de proposer ses services dans le conseil et la formation. Il est important de contribuer à un projet d'envergure comme Apache Tomcat plutôt que Log4J par exemple. Enfin ne pas sur vendre le

produit que l'on défend. Il est important de rester objectif, afin d'apporter aussi un recul nécessaire pour aider les clients finaux. Enfin en tant qu'entrepreneur, créer aujourd'hui un projet Open Source et ensuite une entreprise est dur. Là où en 2000, il était aisé de lancer des projets, de se planter et de recommencer, les règles du jeu ont changées. A ce titre il est intéressant de voir l'aventure d'eXo Platform, de Talend ou Jahia Software aujourd'hui. J'aurai aimé avoir le retour de Sacha par rapport à 2010, et à la stratégie à adopter aujourd'hui. **L'invité surprise de la soirée** Enfin pour terminer sa présentation, pour la session questions-réponses, Sacha invite sur scène une personne qui a beaucoup compté pour moi professionnellement il y a 6 ans : **Marc Fleury**.



Copyright(c) Nicolas Martignole - Licence Creative Commons Paternité-Pas d'Utilisation Commerciale-Partage des Conditions Initiales à l'Identique 2.0 France

Fondateur et ancien patron de JBoss, il a cédé sa société en 2006 pour quelques 350 millions de dollar, dont 40% en cash. C'est une personne qui allie un hémisphère technique et un hémisphère business, ce qui donne quelqu'un avec une vision business capable de voir au delà de la ligne de code. Son parcours est vraiment intéressant et c'est un gars passionnant. A Antonio qui lui fait part de son passif de BEA, il raconte l'histoire suivante : lorsqu'il était à San Francisco en 1998, il voulait entrer chez Weblogic pour y travailler. La veille de signer son contrat, coup de fil : BEA rachète par Weblogic ! Déçu de ne pas être entré chez Weblogic, il lance JBoss en 1999, et l'aventure durera 7 années. Lors de son premier training à Londres, un jeune gars au deuxième rang pose beaucoup de questions et l'intrigue : c'est Sacha Labourey, qui rejoindra JBoss ensuite quelques temps plus tard. Marc a répondu à quelques questions, et il est resté avec nous jusqu'à 5h du matin. Aujourd'hui basé à Madrid en Espagne, il est retiré du monde Java. Passionné de musiques électroniques, il travaille sur des projets de Domotique et d'Appliance pour la maison. Nous avons eu une bonne discussion sur l'iPad. Cet appareil pour un ticket d'entrée de 400\$ révolutionnera aussi le marché de la domotique. Il est clair que dans quelques années, nous pourrons contrôler notre maison avec cet appareil, qui sera aussi capable de lire vos emails, jouer vos vidéos, se comporter comme un eBook. Il explique que pour lui, l'iPad vient de faire un Reset dans le monde de la Domotique. Bref un gars passionnant, que nous espérons revoir de temps en temps à Paris. **Conclusion sur la Keynote** Très bonne présentation. Sacha apporte un oeil éclairé et avisé sur la réalité du développement open-source. Dans les bémols, je pense qu'il y a une part de témoignage sur ce qu'il s'est passé, qui ne peut plus être d'actualité aujourd'hui. En 2010 les développeurs logiciels et le monde a évolué. Mais Sacha reste au fait de l'actualité puisqu'il est investisseur et conseiller spécialisé dans des entreprises comme eXo Platform par exemple. Nuls doutes que nous le croiserons à nouveau un jour. Merci à Sacha Labourey et à Marc Fleury au nom du Paris JUG.

## Lancement de Google Buzz, le tueur de Twitter ?

Google a activé Google Buzz il y a 24 heures comme annoncé [sur le blog officiel de Google](#). C'est un outil de micro-blogging similaire à Twitter, intégré à Gmail, qui permet de partager votre activité du moment avec vos amis. Là par exemple : "Je viens de publier un article sur le *Touilleur Express*". Google Buzz est en fait plus similaire au service [FriendFeed](#), car vos amis peuvent noter s'ils aiment une de vos remarques, un de vos messages. Il existe dès maintenant [une API Google](#) ce qui permettra de créer rapidement des applications intégrées à votre bureau comme Tweetie, ou

des clients pour téléphones mobiles. Comme expliqué sur le blog de Google, ce nouvel outil est une fonction supplémentaire intégrée à Gmail. Vos contacts Gmail deviennent automatiquement vos "followers". Contrairement à Twitter, il n'est pas possible de souscrire à un compte Google Buzz pour le suivre. Cet outil sera un sérieux concurrent à Google Wave pour les micro-discussions, les échanges type forum, que l'on voit sur Google Wave. C'est très similaire à une discussion lancée par une remarque de quelqu'un finalement. Parmi les limitations, Google Buzz ne marche pas avec les comptes Google Gmail pour l'entreprise, intégré à Google Application. Par contre, vous pouvez dépasser les 140 caractères par message, ce qui montre aussi que l'usage visé est différent de Twitter. Donc, non, Google Buzz ne sera pas un tueur de Twitter. **Pour aller plus loin:** [Annonce officielle de Google API Google Buzz](#)

## Version Play! de l'application ZenContact de Zenika



Après avoir codé [la version Grails](#) de l'application [Zencontact de Zenika](#), voyons aujourd'hui la version développée avec le framework Play! grâce à **Guillaume Bort**, créateur de Play! Mais avant, quelques mots sur Play!, un framework qui gagne à être connu. **Présentation du framework Play!** Play! est un framework Web Java open source ([license Apache 2.0](#)) développé par Guillaume Bort et une équipe de contributeurs. Play! a été écrit par des développeurs Webs, pour des développeurs Webs. Attention, je ne parle pas de développeur Web au sens css/html/php mais bien de vous, développeur Java, qui à un moment donné de votre vie avait dû coder cette @\$\* de page en JSP ou en JSF. Quelques caractéristiques de Play! : - En mode développement, il suffit d'éditer un fichier Java, de le sauvegarder et rechargez votre navigateur pour tester. Il n'y a pas de compilations de votre code Java ! Il n'est pas nécessaire de relancer le serveur. Cela fonctionne pour les pages comme pour les classes Java, ce qui est très pratique et très productif. Votre IDE assure même une première vérification, comme avec IDEA IntelliJ, ce qui permet de travailler plus facilement qu'avec Grails à mon avis. - Play est un framework sans état où la vue présentée au navigateur client n'est pas la représentation d'un état sur le serveur, contrairement à d'autres frameworks webs, ce qui permet de monter en charge sans risques. Cela respecte au passage l'architecture web, qui est sans état par nature. - Les pages webs utilisent un système de templates simples basés sur Groovy comme Expression Language. Il est possible de faire de l'héritage de pages, des inclusions et de définir des tags comme avec Grails. - Lorsqu'une erreur survient, Play s'efforce de présenter la source de l'erreur en vous montrant le code plutôt qu'une StackTrace assez indigeste. - Play est intégré avec Hibernate, OpenID, Memcached. - Un système de modules similaire aux plugins de Grails est en cours de construction. - Play n'est pas un framework de prototypage, c'est un outil mis en production avec de solides références - Play contrairement à Grails (que j'adore aussi) est complètement basé sur Java, ce qui permet de garder ses réflexes de "java-istes", son IDE favori (IDEA IntelliJ) et donc de travailler sans changer ses habitudes - Play démarre très rapidement en mode développement, bien plus vite que Grails - Enfin c'est fait par des Français, alors Cocorico, on va tester cela rapidement **Démonstration de l'application finale**

### Fonctions

**implémentées** Pour reprendre la liste des fonctions initiales, voici le périmètre couvert par la version Play! \* La navigation entre pages \* L'organisation d'une structure commune des pages (Type Tiles / Sitemesh) \* La désactivation du lien correspondant à la page courante \* L'édition d'un contact \* La création d'un contact \* Lister les contacts \* L'ajout rapide d'un contact dans une liste \* Le refresh de la date courante via appels Ajax \* L'édition « in place » d'un libellé sans passer par un écran d'édition \* L'ajout de la validation (Nom et prénom obligatoires, contrôle du format date, contrôle du type email) \* Utilisation d'un date picker \* Synchronisation du format du DatePicker avec le format utilisé par le convertisseur \* Gestion de la problématique du refresh afin d'éviter la double soumission \* L'affichage de message d'erreurs Les fonctions non implémentées: - le tri des colonnes n'est pas implémenté (order et orderBy). Play est tout à fait capable de le faire, le module CRUD par exemple a une méthode list avec un order et un orderBy. Cependant ce code

est assez verbeux. 3 autres points **non** couvert par la version Play : \* Ajout des mêmes contrôles de validation côté client, en Javascript \* Non duplication du code du formulaire... le même composant doit être utilisé pour la page d'édition de contacts et de liste des contacts \* La réutilisation des mêmes messages d'erreurs en validation serveur et javascript L'application utilise une base de données en mémoire, tout comme la version Grails. Une entité Contact est gérée par un unique contrôleur. Au premier coup d'oeil, j'apprécie de retrouver ce mode de convention au lieu de la configuration, comme pour Grails. Je ne pourrai pas comparer le nombre de lignes de codes, étant donné que la version Play! n'utilise pas le module CRUD. Mais la class Application ne fait que 30 lignes de code effectif pour remplir le cahier des charges exact de Zenika, le tout avec de la persistance. La class [CRUD.java](#) du module CRUD de Play! est assez verbeuse, mais très pratique. **L'entité de départ "No fluff, just stuff"** décrit bien la simplicité de la classe Contact. Ecrite en Java, elle étend une classe de base de Play! tout en restant très simple :

```
// (manque les imports

@Entity
public class Contact extends Model {
    @Required
    public String firstname;

    @Required
    public String name;

    @Required
    public Date birthdate;

    @Required
    @Email
    public String email;
}
```

L'annotation `@play.data.validation.Required` de Play permet de rendre obligatoire ce paramètre dans la vue, tout comme le tag `@Email`. Pour comparer, voici la version Grails de Contact:

```
package org.letouilleur.demo

class Contact {
    String nom
    String prenom
    Date dateNaissance
    String email

    static constraints={
        prenom(blank:false)
        nom(blank:false)
        dateNaissance(nullable:true)
        email(email:true,nullable:true,blank:true,unique:true)
    }
}
```

**Le Controller Application** Le Controller principal retourne la date et l'heure courante pour la page d'accueil. La méthode `list()` retourne une liste des Contacts, sans prendre en compte cependant deux critères de la vue (`order` et `orderBy`). La méthode `form()` est similaire à la méthode `show()` de la version Grails,

```
public class Application extends Controller {

    public static void index() {
        Date now = new Date();
        render(now);
    }

    public static void list() {
        List contacts = Contact.find("order by name, firstname").fetch();
        render(contacts);
    }

    public static void form(Long id) {
        if (id == null) {
            render();
        }
        Contact contact = Contact.findById(id);
        render(contact);
    }

    public static void save(@Valid Contact contact) {
        if (validation.hasErrors()) {
            if (request.isAjax()) error("Invalid value");
            render("@form", contact);
        }
        contact.save();
        list();
    }
}
```

Notez tout d'abord que dans la version Play, les méthodes du Controller ne retournent pas de Model, mais délèguent à un `render` l'opération de restitution du modèle. J'apprécie la légèreté et la simplicité du code, qui est facilement lisible. Certains d'entre vous doivent être entraînés de se demander pourquoi le Controller étend une classe de base, et même penser que c'est une limitation... ce que je ne partage pas. Cela ne me dérange pas ici de restreindre cette classe Java à

un rôle, en utilisant l'héritage pour la typer. Il y a plusieurs exemples intéressants dans Play! où l'approche est moins académique, et franchement, moi j'aime bien. Tenez, prenons la classe Contact. Avez-vous noté que les attributs sont publiques ? Qu'il n'y a pas d'encapsulation ? A priori, pour récupérer des champs simples, pourquoi s'embêter à les rendre private, pour ajouter un couple getter/setter simple ? A noter que si vous souhaitez écrire une méthode getName, cela reste possible.

```
public class Contact extends Model {
    @Required
    public String name;

    /** Met en majuscule le nom de famille */
    public String getName() {
        return (name!=null ? name.toUpperCase() : null);
    }
}
```

L'application est composée de 2 fichiers Java et de 3 pages HTML, ce qui est vraiment très léger. Bonne nouvelle : je n'ai pas croisé de fichiers XML pour faire marcher le tout. Là où j'ai eu quelques soucis pour comprendre, c'est la partie vue. Je vous donne un avis personnel : j'ai plus de mal avec la syntaxe de Play qu'avec celle de Grails pour les pages. Mais bon, c'est une histoire d'habitude je pense. Développons un peu cet argument. Voici la version Play de la page d'accueil. Comme dans Grails qui utilise Sitemesh, il est très facile de définir un template de page générique :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Zencontact, by zenexity ★ #{get 'title' /}</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <link rel="stylesheet" type="text/css" media="screen" href="@{'public/stylesheets/style.css'}" />
    <link rel="stylesheet" type="text/css" media="screen" href="@{'public/stylesheets/south-street/jquery-ui-1.7.2.custom.css'}" />
    <script src="@{'public/javascripts/jquery-1.4.min.js'}" type="text/javascript" charset="utf-8"></script>
    <script src="@{'public/javascripts/jquery-ui-1.7.2.custom.min.js'}" type="text/javascript" charset="utf-8"></script>
    <script src="@{'public/javascripts/jquery.editinplace.packed.js'}" type="text/javascript" charset="utf-8"></script>
  </head>
  <body>
    <div id="zencontact">
      <header>
        
        <h1>Zencontact <span>by zenexity</span></h1>
      </header>
      <nav>
        <a id="home" href="@{index()}" class="${request.action =~ /index/ ? 'selected' : ''}">Home</a>
        <a id="list" href="@{list()}" class="${request.action =~ /list/ ? 'selected' : ''}">List</a>
        <a id="new" href="@{form()}" class="${request.action =~ /form|save/ ? 'selected' : ''}">New</a>
      </nav>
      <section>
        #{doLayout /}
      </section>
      <footer>
        <a href="http://www.w3.org/TR/html5/">html5</a> -
        <a href="http://www.w3.org/TR/css3-roadmap/">css3</a> -
        <a href="http://www.playframework.org/">playframework</a>
      </footer>
    </div>
  </body>
</html>
```

La ligne 24 sera remplacée par le contenu des pages filles. La page d'accueil avec le compteur de temps étend la page main.html, définit le titre puis utilise jQuery pour récupérer l'heure. Ma version Grails était moins verbeuse car j'ai utilisé un GTag, ce qui a l'avantage d'alléger le code de cette page, mais ce qui n'est pas très réaliste pour un projet. Guillaume a prit l'option d'implémenter ici directement la logique et l'affichage, pourquoi pas ?

```
#{extends 'main.html' /}
#{set title:'Home' /}

${now.format("EEEE", ' MMMM dd', ' yyyy')} ${now.format("hh'h' MM'mn' ss's'")}
```

La page list.html est chargée d'itérer la collection des Contacts et de les afficher dans un tableau. Play a [un système de tag](#) comme Grails, ce qui permet par exemple d'utiliser le tag `list` pour itérer une collection:

```
<ul>
#{list items:products, as:'product'}
  <li>#{product}</li>
#{/list}
</ul>
```

La page list.html est donc simple. J'ai essayé sans succès de coder des liens sur les entêtes de colonnes afin d'effectuer un tri en passant un orderBy et un order, mais fautes de temps, je n'ai pas réussi à améliorer la version de Guillaume Bort.

```
#{extends 'main.html' /}
```

```

#{set title:'List' /}

<table>
  <thead>
    <tr>
      <th class="name">Name</th>
      <th class="firstname">First name</th>
      <th class="birthdate">Birth date</th>
      <th class="email">Email</th>
      <th class="edit"></th>
    </tr>
  </thead>
  <tbody>
    #{list contacts, as:'contact'}
    <tr class="contact" contactId="${contact.id}" draggable="true">
      <td id="name-${contact.id}">${contact.name}</td>
      <td id="firstname-${contact.id}">${contact.firstname}</td>
      <td id="birthdate-${contact.id}">${contact.birthdate?.format('yyyy-MM-dd')}</td>
      <td id="email-${contact.id}">${contact.email}</td>
      <td><a href="#{form(contact.id)}"></a></td>
    </tr>
  </list>
  <tr>
    #{form @save()}
    <td><input type="text" name="contact.name"></td>
    <td><input type="text" name="contact.firstname"></td>
    <td><input type="text" name="contact.birthdate"></td>
    <td><input type="text" name="contact.email"></td>
    <td><input type="submit" value="+"></td>
  </form>
  </tr>
</tbody>
</table>
// note : code Javascript pour l'Edit-In-Place et le Drag-and-drop effacé.

```

La page permet de lister une collection, d'éditer directement dans le tableau le nom d'une personne, et d'appeler la page form.html pour éditer une fiche. Notez dans la démonstration que la version de Guillaume affiche un nom en Chinois sans problèmes, merci l'UTF-8. Enfin la dernière page est le formulaire de création et d'édition d'une fiche.

```

#{extends 'main.html' /}
#{set title:'Form' /}

#{form @save()}
<input type="hidden" name="contact.id" value="${contact?.id}">

<p class="field">
  <label for="name">Name:</label>
  <input type="text" id="name" name="contact.name" value="${contact?.name}">
  <span class="error">${errors.forKey('contact.name')}</span>
</p>

<p class="field">
  <label for="firstname">First name:</label>
  <input type="text" id="firstname" name="contact.firstname" value="${contact?.firstname}">
  <span class="error">${errors.forKey('contact.firstname')}</span>
</p>

<p class="field">
  <label for="birthdate">Birth date:</label>
  <input type="text" id="birthdate" name="contact.birthdate" value="${contact?.birthdate?.format('yyyy-MM-dd')}>
  <span class="error">${errors.forKey('contact.birthdate')}</span>
</p>

<p class="field">
  <label for="email">Email:</label>
  <input type="text" id="email" name="contact.email" value="${contact?.email}">
  <span class="error">${errors.forKey('contact.email')}</span>
</p>

<p class="buttons">
  <a href="#{list()}">Cancel</a> or <input type="submit" value="Save this contact" id="saveContact">
</p>

<script type="text/javascript" charset="utf-8">
  $("#birthdate").datepicker({dateFormat:'yy-mm-dd', showAnim:'fadeIn'})
</script>
#{/form}

```

Par rapport à la version Grails, le format de la Date est bien mieux géré. J'aime bien aussi l'utilisation de Groovy, ce qui permet de retrouver les astuces et de gagner du temps, comme lorsque l'on s'assure qu'un paramètre n'est pas null avant de l'appeler :

```
<input type="text" id="birthdate" name="contact.birthdate" value="${contact?.birthdate?.format('yyyy-MM-dd')}>
```

**Bootstrap** Play permet de définir des entrées d'exemple dans un fichier externe, afin de remplir la base de tests au démarrage. Le format [YAML](#) est très simple à lire, et il est supporté <TROLL>par le meilleur éditeur du monde : [IDEA Intellij](#) </TROLL> maintenant aussi disponible en version open-source.

```

Contact(warry):
  name:      Dantec
  firstname: Maxime
  birthdate: 1985-11-12
  email:    hello-xxx@warry.fr

```

```
Contact(guillaume):
  name:      Bort
  firstname: Guillaume
  birthdate: 1980-12-21
  email:     xxx-bob@gmail.com
```

```
Contact(たの):
  name:      鈴木
  firstname: 太郎
  birthdate: 1970-05-13
  email:     contact@xxxjapan.com
```

La classe Bootstrap.java de Play permet de charger les Contacts, si la liste en base est vide :

```
@OnApplicationStart
public class Bootstrap extends Job {
    public void doJob() {
        if(Contact.count() == 0) {
            Fixtures.load("data.yml");
        }
    }
}
```

**Le résultat final** Le résultat final est très léger, et franchement, en terme de performances, le démarrage est moins poussif que Grails. La possibilité de débogger avec son éditeur, et le fait d'avoir des Stacktraces propres est aussi un gain par rapport à Grails. Je n'ai pas eu à relancer le serveur, en fait vous n'avez même pas besoin de compiler vos classes Java. Sur IDEA IntelliJ il n'y a pas de compilation comme dans Eclipse lorsque vous sauvegardez vos fichiers. En fait, vous ne sauvez jamais vos fichiers avec IDEA IntelliJ. Celui-ci le fait lorsque vous perdez le focus sur la fenêtre principal. Du coup, le couple IDEA + Play déchire en terme de vitesse de développement. J'ai cependant pas mal de bémols, très suggestifs et que vous ne devez pas prendre pour argent comptant. Tout d'abord du côté de la vue, j'ai tellement l'habitude des syntaxes verbeuses, à la JSF ou à la GSP pour Grails, que j'ai eu un peu de mal à me faire à la syntaxe de Play. Par ailleurs, parfois certains tags ne sont pas très consistants entre eux, et il n'est pas très évident de rentrer facilement dans cette partie. Un point positif de Play c'est la possibilité d'utiliser Selenium afin d'ajouter des tests d'intégrations très facilement. Il suffit de lancer l'application en mode test avec "play test zencontact" et ensuite d'ouvrir la page <http://localhost:9000/@tests> pour pouvoir lancer les tests Selenium. C'est une idée qui va encourager l'utilisation de TDD pour écrire une application Play, ce qui est vraiment bien. Concernant un point sur lequel j'ai eu des soucis avec Grails, c'est le support des bases "Legacy". Florent Ramière de [Jaxio](#) avec qui j'ai travaillé il y a une semaine, m'a montré les limitations de Grails dès qu'il s'agit de prendre une base existante. Oui cela fonctionne, c'est jouable, mais Grails perd de son charme lorsqu'il faut jouer à l'équilibriste avec des tables existantes. Je pense que Play! sera peut-être plus à l'aise grâce à l'intégration de JPA, mais nous devrions refaire le test sur la base de données de démarrage de [Celerio](#) pour voir le tout sur un cas d'entreprise. **Conclusion** Ce qui m'a séduit dans le code, c'est cette simplicité et ces idées novatrices qui sont proches de Grails et de Rails, mais le tout en Java. Avant tout pensé par des développeurs Webs, ce que je suis sans aucuns doutes, je trouve qu'il serait intéressant pour vous d'y jeter un oeil, et d'apprendre à vous en servir. Par rapport à Grails, je ne sais pas si la notion de Service et d'injection existe par exemple. Par contre, un moteur de Jobs permet d'exécuter périodiquement des actions, sans besoin d'avoir une requête HTTP. Des modules de sécurité, d'authentification, le support de GWT et de Spring, il y a de quoi démarrer une application Web rapidement. J'ai quelques réserves sur la partie vue, mais je pense que c'est par manque d'expérience. Je ne sais pas non plus si Play a la notion de Render pour afficher la liste des Contacts au format XML ou JSON très simplement, ou s'il faut passer par une API externe supplémentaire. Enfin on apprécie vraiment le fait de ne même pas devoir compiler son code Java. Vous pouvez dire adieu à Maven/Ant et même JRebel, là on parle de productivité. La mise en production est simple, Guillaume m'a confirmé qu'il est facile de créer un WAR final pour le déployer comme avec Grails:

```
play war zencontact -o zencontact.war
```

Play! par rapport à Grails manque de modules, mais ce n'est qu'une histoire de temps. Le code est propre, la documentation plutôt bien écrite, et l'ensemble est codé sur de solides fondations, basées sur une expérience et un certain talent pour écrire cela de manière moderne. J'ai apprécié, et je le range en plus sur mon étagère de framework à connaître pour travailler efficacement. Le code complet de l'application zencontact est disponible ici. 1. Téléchargez [une version récente de Play 2](#). Décompressez dans un répertoire, ajoutez à votre PATH ce répertoire 3. Ouvrez un terminal, vérifiez que Play démarre en tapant "play" 4. Téléchargez [l'application zencontact](#) 5. Décompressez-là dans un autre répertoire comme C:\demo pour avoir au final un C:\demo\zencontact par exemple 6. Dans le répertoire C:\demo, tapez "play run zencontact" et ouvrez votre navigateur sur <http://localhost:9000> Merci à Guillaume Bort de Zenexity qui a codé la version Play de l'application. Guillaume travaille pour [Zenexity](#), un cabinet de conseils et d'architecture qui propose une démarche d'architecture autour du Web. Allez voir aussi leurs différents billets car il y a beaucoup d'idées intéressantes et novatrices. Enfin ils organisent [des formations sur Play](#), ce qui pourra intéresser les équipes en phase de démarrage. Allez Grails, fait pas la tête, t'es encore mon préféré, mais tu as un sérieux concurrent ! A bientôt.

## Création du Domain Driven Design User Group



Domain Driven Design User Group  
<http://groups.google.fr/group/dddugparis>

Je vous annonce la création du Domain Driven Design Users' Group de Paris, un nouveau groupe d'utilisateurs consacrés au Domain Driven Design. Nous en avons parlé [il y a quelques mois](#) à l'occasion de la venue d'Eric Evans en France au Paris JUG, il y a maintenant un nouveau groupe à destination des passionnés de la conception logicielle par DDD. Les 2 fondateurs, Jérémie Grodziski et Grégory Weinbach, vous donnent rendez-vous [le mercredi 17 février](#) chez Zénika à l'occasion de la venue d'Eric Evans en France.  
**Le communiqué de presse:**

Le [Domain Driven Design](#) (DDD) est une approche de la conception de logiciel dont les principes fondateurs sont : - Le point d'attention principal doit être le domaine et sa logique, - La conception de domaine complexe doit être basée sur un modèle Le Domain Driven Design n'est pas une technologie ni une méthodologie. C'est une manière de penser et un ensemble de priorités, dont l'objectif est d'accélérer la réalisation des projets qui gèrent des domaines complexes. Le Domain Driven Design Users' Group de Paris a pour vocation de fédérer et mettre en relation la communauté DDD française. Les membres pourront y partager leurs expériences, leurs feedbacks et leur enthousiasme. Une rencontre sera organisée tous les deux mois pour permettre aux membres de se rencontrer et d'échanger sur cette approche. Eric Evans, créateur du concept, sera présent à Paris le 17 février prochain et animera une conférence sur le sujet "Modèles en action". Nous profiterons de cette occasion pour inaugurer le Users' Group en sa présence. Le nombre de participants étant réduit, il est nécessaire de s'inscrire au préalable pour y assister. Le groupe est organisé de manière participative puisque les membres pourront aussi proposer des sujets de conférence, des ateliers ou des débats. Le DDD Users Group Paris tiendra également informé ses membres des nouveautés et événements divers organisés autour du DDD en France, de manière à créer une réelle synergie autour de cette approche. Les inscriptions au DDD User Group de Paris se font directement via le groupe Google :

<http://groups.google.fr/group/dddugparis> Pour plus d'informations, contactez DDD User Group France : [info@domaindrivendesign.fr](mailto:info@domaindrivendesign.fr) Le DDD Users' Group de Paris est animé par Jérémie Grodziski et Grégory Weinbach. Jérémie est consultant indépendant. Il a une passion pour la conception de logiciels et de systèmes d'informations. La vision qu'il déploie pour ses clients depuis plus de dix ans vise à rapprocher le métier de l'outil informatique. Il associe état de l'art méthodologique et technologique, pragmatisme et expérience afin de répondre à trois enjeux majeurs : Comment maîtriser la complexité du métier et de la technologie ? Comment assurer l'efficacité de la réalisation et limiter les risques ? Comment valider la qualité et l'adéquation du système réalisé ? Son objectif est de réaliser des logiciels que les utilisateurs apprécient et sur lesquels les développeurs aiment travailler. [jeremie@grodziski.com](mailto:jeremie@grodziski.com), <http://www.grodziski.com/> Grégory est animateur du pôle Consulting et responsable de l'offre MDA d'[Objet Direct](#), société de services spécialisée sur les technologies objet. Pour lui, améliorer la productivité des développements passe par un recentrage sur le code à forte valeur ajoutée métier. Partie prenante de nombreux projets s'appuyant sur des techniques de génération de code, il collabore aussi au développement d'outils de génie logiciel : D.OM (Direct Object Modeler, outil de prototypage DDD) et Metadone (plateforme MDA) tous deux édités par [Objet Direct](#). [gweinbach@gmail.com](mailto:gweinbach@gmail.com), <http://mdblog.fr>



Bravo pour l'initiative et rendez-vous le 17 février à 19h00 chez Zenika.