

# Comment développer pour le projet IceScrum<sup>2</sup>

<b>1) Création des comptes SVN .....</b>	<b>0</b>
<b>2) Téléchargement et Installation de l'environnement de production.....</b>	<b>0</b>
<b>3) Téléchargement et Installation de l'IDE .....</b>	<b>0</b>
<b>4) Récupérer le projet IceScrum .....</b>	<b>0</b>
<b>4.1) Sous Eclipse.....</b>	<b>0</b>
<b>4.2) Sous NetBeans.....</b>	<b>0</b>
<b>5) Utiliser le projet.....</b>	<b>0</b>
5.1) Updater (récupérer depuis le dépôt SVN).....	0
5.2) Commiter (envoyer vos modifications sur le dépôt SVN) .....	0
5.3) Quand faire un "Maven install" .....	0
<b>6) Reporter des bugs .....</b>	<b>0</b>
<b>7) Problèmes connus lors de l'utilisation du projet .....</b>	<b>0</b>
7.1 ) Performances d'éclipse .....	0
7.2) Maven 1 .....	0
7.3) Maven 2.....	0
7.4) Wamp/EasyPHP et Skype .....	0

## **1) Création des comptes SVN**

- SourceForge (pour commit sur le svn) : <http://sourceforge.net/> (il faut ensuite que Stéphane vous ajoute au projet IceScrum)
- Forum : <http://forum.icescrum.org/>

## **2) Téléchargement et Installation de l'environnement de production**

- Installer le JDK6
- Télécharger Tomcat v6 : <http://tomcat.apache.org/download-60.cgi> (core) et dézippez dans un coin de votre disque dur.
- Pour la BD MySQL, installer Wamp 2 (alternative à EasyPHP) <http://www.wampserver.com/download.php> (lien direct en bas de la page)
- Lancer Wamp, et créer grâce à phpMyAdmin deux nouvelles bases « **icescrum2** ». C'est Hibernate qui s'occupera de créer les tables.

**Uniquement pour Eclipse (Maven étant inclus par défaut dans NetBeans) :**

- Télécharger et installer Maven 2.0.9 (<http://maven.apache.org/download.html>). Suivre les instructions au bas de la page pour modifier les variables d'environnement.

## **3) Téléchargement et Installation de l'IDE**

### **3.1) IDE Eclipse**

Sous Eclipse, après avoir installé la version EE :

- Rajouter le plugin SVN via le menu "help/software updates", tuto : <http://subclipse.tigris.org/install.html>

- Télécharger et installer ce plugin eclipse :  
<http://m2eclipse.sonatype.org/update/>

- Window/Preferences (globalement pour Eclipse) :  
Java/Installed JREs : avoir un jdk au moins 1.5, (par exemple jdk1.6.0\_11). Le cocher comme jdk par défaut.  
Java/Compiler : JDK Compliance = 1.5, Use default compliance settings = true

La configuration d'Eclipse n'est pas stockée dans le SVN. Les fichiers de type ".project", ".settings", etc. spécifiques à l'IDE Eclipse sont ignorés (.svnignore) par le client SVN. Ces fichiers ne DOIVENT PAS être commités manuellement ou de quelque façon que ce soit.

### **3.2) IDE NetBeans**

Sous NetBeans :

- Installer SVN :  
Versioning -> Subversion -> checkout -> ok

- Vérifier que Maven est installé :  
Tools -> Plugins -> Available Plugins  
Si "Maven" n'est pas coché, le sélectionner, puis "Install".

- Vérifier que Maven est activé :  
Tools -> Plugins -> Installed  
Si "Maven" n'est pas activé, cliquer dessus, puis "Activate"

La configuration de NetBeans n'est pas stockée dans le SVN. Les fichiers de type "nbaction.xml", etc. spécifiques à l'IDE NetBeans sont ignorés (.svnignore) par le client SVN. Ces fichiers ne DOIVENT PAS être commités manuellement ou de quelque façon que ce soit.

## **4) Récupérer le projet IceScrum**

### **4.1) Sous Eclipse**

#### **Checkout du projet**

- Créez un nouveau projet dans Eclipse depuis un repository SVN.
  - URL : <https://icescrum.svn.sourceforge.net/svnroot/icescrum>
  - Login/mdp SourceForge (seulement au moment de commit)
- Sélectionnez la branche "trunk/icescrum2".

- Faites un import (Menu File/Import) de type "General/Maven projects" et sélectionnez le dossier icescrum2 dans votre workspace. 4 sous-projets doivent apparaître. Sélectionnez-les et faites finish. Cette étape nécessite un temps de traitement assez long (de 15 à 20 minutes)
- Faites "Window => Préférences => Maven" et décochez "Download repository index updates on startup"
- A vérifier : il peut être nécessaire de refaire une compilation :
  - . soit sur le projet père "icescrum2" : [Run As/Maven clean] puis [Run As/Maven install]
  - . soit dans l'ordre des projets : dao, services, presentation, web
- Créer un nouveau serveur de type Tomcat v6 dans Eclipse (onglet server) et associer le serveur à icescrum2-web.

(Ca ne doit plus arriver mais) Si le projet vous indique des commits à effectuer, faites un clic droit sur « **icescrum2** » et choisissez "Replace with => Latest from repository"

- Enfin, aller dans les Properties (clic-droit) de chaque projet : icescrum2, icescrum-dao, icescrum-presentation, icescrum-services, icescrum-web :
  - Java Compiler : décocher "Enable project specific settings"
  - Project Facets : Java - Version = 5.0
- Refaire un RunAs/Maven install.

## Configuration de la BD locale pour dev :

Par défaut, IceScrum utilise une base de données embarquée HSQLDB, qui est située dans "<HOME> (Mes documents sous windows)/icescrum/hsqldb".

Si vous souhaitez modifier cette configuration pour utiliser, par exemple, MySQL, vous pouvez le faire soit par modification du fichier de configuration, dans le projet "icescrum-web", dans le fichier :  
`/src/main/webapp/META-INF/icescrum2/config/configuration.properties` (fichier vide par défaut)

Vous pouvez alors surcharger les paramètres par défaut (HSQLDB) pour une BD MySQL avec :

### Paramètres pour utiliser une base de données MySQL :

```
database.driverClassName = org.gjt.mm.mysql.Driver
database.url = jdbc:mysql://127.0.0.1/icescrum2
database.dblogin = root
database.dbpwd =
database.dialect = org.hibernate.dialect.MySQLInnoDBDialect
```

## Restrictions d'accès à JAXB :

Cause = restrictions d'accès à la librairie JAXB (javax.xml.bin). Equinox provoque les erreurs de compil.

### - Avec un JDK 5 : pas d'erreurs

- Avec un JDK 6 : Il y a des erreurs du type *Access restriction: The type XmlElement is not accessible due to restriction on required library C:\Program Files\Java\jdk1.6.0\_12\jre\lib\rt.jar*

Pour les enlever : Window/preference => Java/Compiler/"Errors/Warnings"=>Deprecated and restricted API => Forbidden reference (access rules) => Warning

## Activation des assertions Java

- Avec un Tomcat lancé par WTP dans Eclipse :  
Vue "Servers", faire un "Open" sur Tomcat 6 Server at localhost  
Cliquer sur "Open launch configuration"  
Dans l'onglet "Arguments", ajouter à la fin des "VM arguments", le paramètre : "-enableassertions" ou "-ea" (pour activer les assertions pour tout le code)

Il est possible de ne les activer que pour certains paquetages java : à compléter.

### **4.2) Sous NetBeans**

#### Checkout

- Sous NetBeans, télécharger le projet Icescrum  
Versionning -> Subversion -> checkout  
Dans "Repository URL" tapez <https://icescrum.svn.sourceforge.net/svnroot/icescrum>  
Entrer le login et le mot de passe de Sourceforge.net  
Cliquez sur "Next"  
Dans "Repository folder" cliquer sur "Browse" et sélectionner Trunk -> Icescrum2  
Dans "Local Folder" sélectionner le dossier dans lequel sera importer IceScrum2  
Cliquez sur "Finish"  
Après le checkout, Eclipse demande si on veut associer un IDE : sélectionner close  
File -> Open Project : sélectionner icescrum2 puis cocher "Open as Main Project" et "Open Required Project"  
Sélectionner un serveur : clic droit sur WAR Module -> run -> GlassFish  
Sur le projet icescrum2(pom) faire un "clean and build". (Attention, le premier build est long, c'est normal).
- Lancer le projet IceScrum :  
Clic droit sur WAR Module -> run  
Remarque : Vous pouvez utiliser Tomcat à la place de Glassfish (Par défaut).

## 5) **Développer pour le projet**

Sur le répertoire icescrum2, vous pouvez faire "Run As => Maven clean" pour nettoyer et "Run As => Maven install" pour installer le projet et ses dépendances.  
Pour lancer l'application, lancer Wamp s'il n'est pas déjà lancé. Ensuite, lorsqu'il est bien démarré, lancer le serveur depuis l'onglet Server ou par clic droit sur icescrum-web et "Run As => run on Server".

L'application est visible à l'adresse <http://localhost:8080/icescrum2/>

### 5.1) Configuration de la BD locale pour dev :

Par défaut, IceScrum utilise une base de données embarquée HSQLDB, qui est située dans "<HOME> (Mes documents sous windows)/.icescrum/hsqldb".

Si vous souhaitez modifier cette configuration pour utiliser, par exemple, MySQL, vous pouvez le faire soit par IHM, soit par fichier de configuration :

par IHM :

vous connecter à IceScrum (voir la partie "Utiliser le projet") en tant que super administrateur (Login sa / Pass sa) et modifier les informations de connexion à la base de données. **Après avoir validé les nouveaux paramètres dans l'IHM + confirmer le re-démarrage, il faut attendre un peu (max 15 secondes) pour que l'appli re-démarre (si ça marche).**

par fichier de config :

si le changement de conf ne fonctionne pas par l'IHM, il faut modifier le fichier le fichier de configuration. Se reporter à la section correspondante à votre IDE (Eclipse ou Netbeans)

**Paramètres pour utiliser une base de données HSQL (configuration par défaut) :**

```
database.driverClassName = org.hsqldb.jdbcDriver
database.url = jdbc:hsqldb:file:${user.home}/.icescrum/hsqldb/icescrum2.tmp
database.dblogin = sa
database.dbpwd = sa
database.dialect = org.hibernate.dialect.HSQLDialect
```

**Paramètres pour utiliser une base de données MySQL :**

```
database.driverClassName = org.gjt.mm.mysql.Driver
database.url = jdbc:mysql://127.0.0.1/icescrum2
database.dblogin = root
database.dbpwd =
database.dialect = org.hibernate.dialect.MySQLInnoDBDialect
```

### 5.2) Updater (récupérer depuis le dépôt SVN)

Normalement, si votre application est bien configurée, vous pouvez faire un update (clic droit, Team=>Update) depuis le projet principal ou un des sous-projets. Si l'option n'apparaît pas, faites-le depuis le projet principal.

Ceci s'explique par le fait que les sous-projets ne sont qu'une **duplication** synchronisée (des **alias**) des sous-répertoires du projet principal.

### 5.3) Commiter (envoyer vos modifications sur le dépôt SVN)

Comme pour l'update, vous pouvez le faire de n'importe où.

Une popup s'ouvrira et vous demandera des identifiants. Entrez ceux que vous avez utilisé pour SourceForge. Vous pouvez les mémoriser si vous souhaitez ne pas avoir à les reentrer à chaque fois. Néanmoins, si quelqu'un commite depuis votre pc, ce sera avec vos identifiants.

### 5.4) Quand faire un "Maven install"

- A la première utilisation du projet
- Lorsqu'un commiteur vous indique qu'il a modifié une ou plusieurs librairies.

Faites un Maven clean auparavant si on souhaite faire un packaging. (F5 pour rafraichir, comme pour le Maven install).

Le Maven install (mis à part le premier) n'est pas nécessaire lors de vos tests avec le serveur de votre Eclipse.

## 5.5) Utiliser les assertions Java

La programmation par contrat et les assertions Java (depuis le jdk 1.4), avec des exemples : <http://manu40k.free.fr/articleProgrammationParContrat.pdf>

### Configurer les assertions à l'exécution :

Par défaut, au lancement de la JVM, les assertions ne sont pas actives. Pour les activer, il faut passer l'option "-ea" ou "-enableassertions", au lancement de la JVM (et donc, du serveur d'appli) :

- pour un serveur d'application lancé par WTP sous Eclipse : se reporter au chapitre Eclipse
- pour un serveur d'application lancé directement : ajouter l'option à la ligne de commande

Avec Glassfish (sous NetBeans par exemple), il faut se connecter à la console d'administrateur, une fois le serveur lancé depuis votre IDE : <http://localhost:4848/>

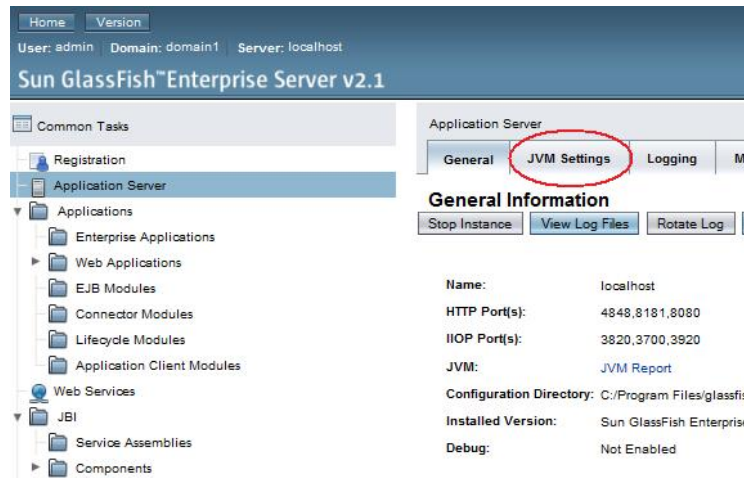


Les login et password sont : admin et adminadmin (par défaut).

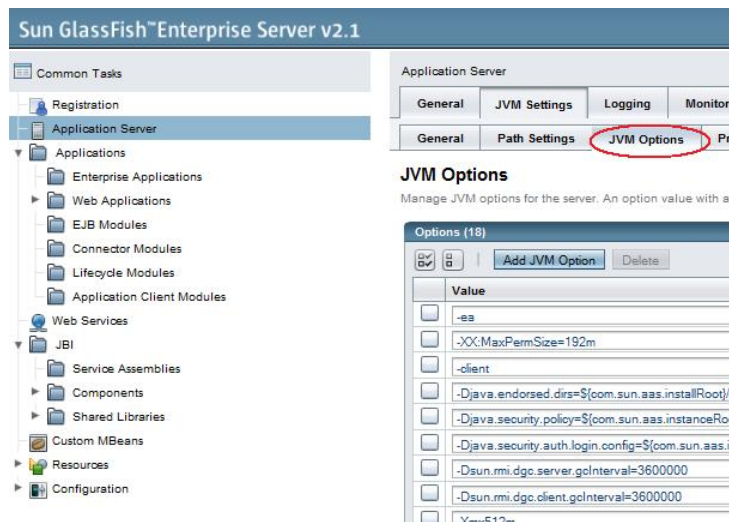
Une fois loggés, vous arrivez sur cette page :



Sélectionnez Application Serveur (entouré en rouge ci-dessus).



Choisissez alors JVM Settings, puis JVM Options et vous arrivez à l'écran ci-dessous.



Cliquez alors sur Add JVM Option et ajoutez l'option -ea. Une fois le serveur redémarré, il acceptera les assertions.

Exemple dans la console de NetBeans :

```
java.lang.AssertionError: DataModelError : the task creator must exist !!
```

## 6) Reporter des bugs

Lorsque vous détectez un bug dans l'application, 2 cas sont possibles :

### Story est en cours

vous corrigez le bug dès qu'il se présente ou rajoutez une tâche afin de corriger ce problème avant la fin du sprint.

## Story est finie

plusieurs cas se présentent :

- **Défaut critique**

Il doit être traité de manière prioritaire. On ajoute une tâche d'une heure dans le backlog de sprint afin d'étudier le défaut, identifier sa cause et corriger le défaut, en **mettant à jour les tests automatisés**. Si, au bout d'une heure, le bug n'est pas corrigé ou si on considère qu'il faudra plus d'une heure pour régler le problème, il faudra alors identifier les tâches nécessaires pour la résolution et les ajouter dans le backlog de sprint. Il est probable que cela aura un impact sur le périmètre du sprint. Il convient de ne déclarer comme critique ce qui est vraiment catastrophique pour l'usage du produit, sans solution de contournement.

- **Défaut majeur**

Il se règle de la même manière. La différence vient du fait que, si le bug n'est pas corrigé au bout d'une heure, on le place dans le backlog de produit afin de le régler lors d'une future itération.

- **Défaut mineur**

Il est directement placé dans le backlog de produit (à moins qu'il puisse être corrigé immédiatement).

voir : <http://www.aubryconseil.com/post/La-gestion-des-bugs-dans-le-projet-IceScrum>

## 7) Problèmes connus lors de l'utilisation du projet

### 7.1 ) Performances d'éclipse

Il faut rajouter les arguments suivants dans le raccourci de lancement d'Eclipse :

```
"-vmargs -Xms768m -Xmx768m -XX:PermSize=256m -XX:MaxPermSize=256m -XX:+CMSPermGenSweepingEnabled -XX:+CMSClassUnloadingEnabled"
```

source : <http://jvillage.wordpress.com/2008/02/09/optimisation-de-la-memoire-de-la-jvm-pour-eclipse/>

### 7.2) Maven 1

Parfois, le jdk n'est pas sélectionné par défaut. Allez dans Window => Préférences => Java => Installed JREs. Si le jdk n'apparaît pas, ajoutez le chemin vers le dossier. S'il n'est pas sélectionné, sélectionnez-le.

### 7.3) Maven 2

Pour régler l'erreur suivante :

*"The maven integration requires that eclipse be running in a JDK, because a number of Maven core plugins are using jars from the JDK."*

- Il faut en plus spécifier les variables de lancement d'Eclipse :
- Soit dans le fichier "**eclipse.ini**" en ajoutant : **-vm C:\<chemin\_vers\_JDK>\bin\javaw** (ne semble pas fonctionner pour les raccourcis Windows, cf. ci-dessous)
  - Soit quand on a un **raccourci** Windows, modifier le chemin vers la cible / target ainsi :  
**C:\<chemin\_vers\_eclipse>\eclipse.exe -vm "C:\<chemin\_vers\_JDK>\bin\javaw"**

ou `C:\<chemin_vers_eclipse>\eclipse.exe -vm "C:\<chemin_vers_JDK>\bin\"`

A tester ! C'est du bidouillage malheureusement nécessaire à ce jour.

#### 7.4) Wamp/EasyPHP et Skype

Ces logiciels utilisent le même port. Si vous souhaitez utiliser les 2 en même temps, lancez **d'abord** Wamp/EasyPHP et **ensuite** Skype (une fois que le premier est correctement démarré). Si l'icône de WAMP est en jaune lors du lancement, il risque de se produire un problème.

Nous cherchons une autre solution à ce problème.