

Débugger une application web avec Maven et Jetty sous Eclipse

par Mohamed KARAMI (<http://karamimed.developpez.com/>)

Date de publication : 17 Avril 2009

Dernière mise à jour : 07 Mai 2009




Ce tutoriel a pour objectif, de montrer comment on peut débbuguer une application web déployée en utilisant Maven et Jetty sous Eclipse, ceci facilitera largement la vie des développeurs qui s'amuse à utiliser Maven et Jetty en ligne de commande.

I - Remerciements.....	3
II - Outils utilisés.....	3
III - Installation du plugin Maven sous Eclipse.....	3
IV - Configuration du plugin Jetty.....	3
V - Création du raccourcis " Run on Jetty ".....	3
VI - Création du raccourcis " Debug on Jetty ".....	5

I - Remerciements

Merci à **ram-0000** pour sa relecture, à **bbil** pour son aide concernant la génération de la version pdf, et à **Nono40** réalisateur de l'éditeur xml.


II - Outils utilisés

- 1 Eclipse Ganymede 3.4.2 : téléchargeable sur le site officiel :  Eclipse.org
- 2 Maven 2.0.9 et le plugin Maven d'Eclipse : depuis l'outil de mise à jour d'Eclipse  M2eclipse.sonatype.org
- 3 Serveur web Jetty 6.1.10 : téléchargeable sur le site :  Jetty.mortbay.org

III - Installation du plugin Maven sous Eclipse

Pour installer le plugin Maven sous Eclipse, utiliser le menu "Help => Software Update => Find and install => Search for new features to install"

Ajouter un nouveau site via " New remote site " et saisir l'url : <http://m2eclipse.sonatype.org/update/> avec comme nom Maven2, et suivre les instructions d'installation. Ignorer les packages optionnel (ils posent souvent des problèmes de compatibilité avec Eclipse).

Pour plus d'information l'installation de Maven, voir le lien suivant :  [Installation de Maven](#)

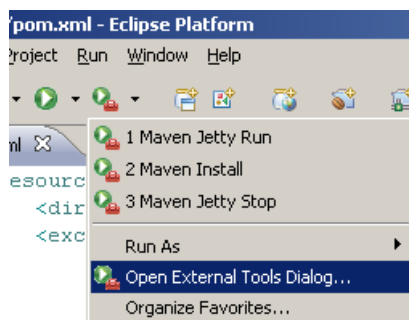
IV - Configuration du plugin Jetty

Il faut renseigner jetty comme plugin dans le fichier "pom.xml" du projet :

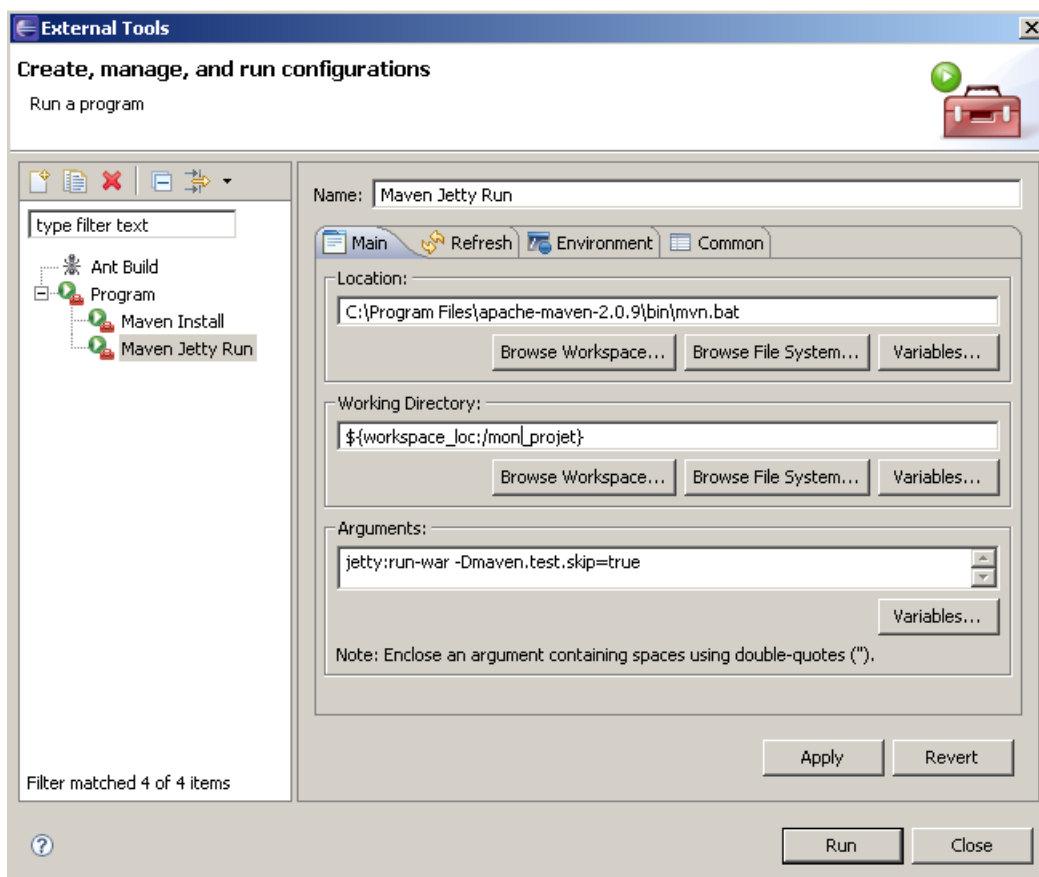
```
<plugins>
<plugin>
<groupId>org.mortbay.jetty</groupId>
<artifactId>maven-jetty-plugin</artifactId>
<version>6.1.10</version>
<configuration>
<scanIntervalSeconds>10</scanIntervalSeconds>
</configuration>
</plugin>
</plugins>
```

V - Création du raccourcis " Run on Jetty "

A l'aide du menu "Open External Tools Dialog", on peut ajouter un nouveau menu qui permet de lancer une application externe à Eclipse.



On sélectionne l'onglet " Main ", et on renseigne les informations de l'outil à ajouter :




Dans cet onglet on spécifie le dossier d'installation de Maven (le fichier mvn.bat), la racine de l'application à débbuguer, et la commande de lancement du serveur Jetty.

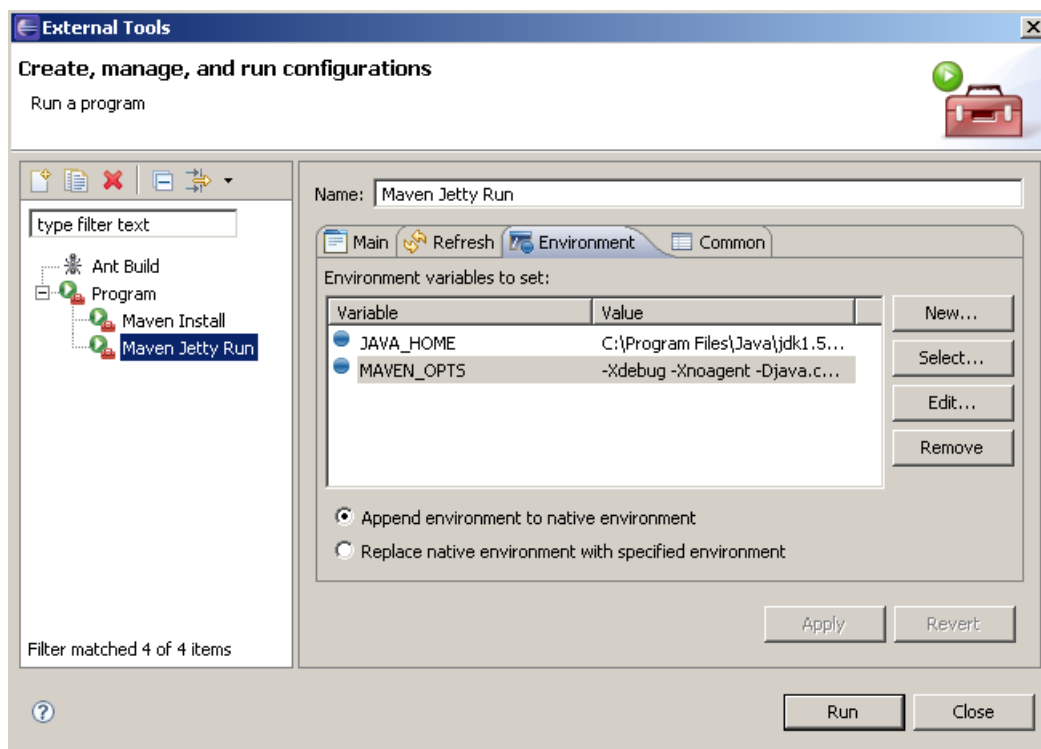
Configuration de l'onglet Main :

```

Location = C:\Program Files\apache-maven-2.0.9\bin\mvn.bat
Working directory = ${workspace_loc:/mon_projet}
Arguments = jetty:run-war -Dmaven.test.skip=true
    
```

 *L'option "-Dmaven.test.skip=true" permet d'ignorer les tests unitaires.*

Dans cet onglet, il faut déclarer les deux variables d'environnement, JAVA_HOME pour le dossier d'installation du JDK et les options de débbugage.



Configuration de l'onglet Environment :

```

JAVA_HOME = C:\Program Files\Java\jdk1.5.0_16
MAVEN_OPTS = -Xdebug -Xnoagent -Djava.compiler=NONE -
Xrunjwp:transport=dt_socket,server=y,suspend=y,address=8000
    
```

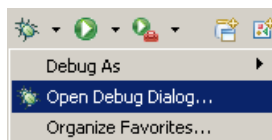
On peut lancer le serveur, à l'aide du bouton "Run", maintenant le serveur est en écoute, mais ceci n'est pas suffisant, il faut l'attacher un débogueur.



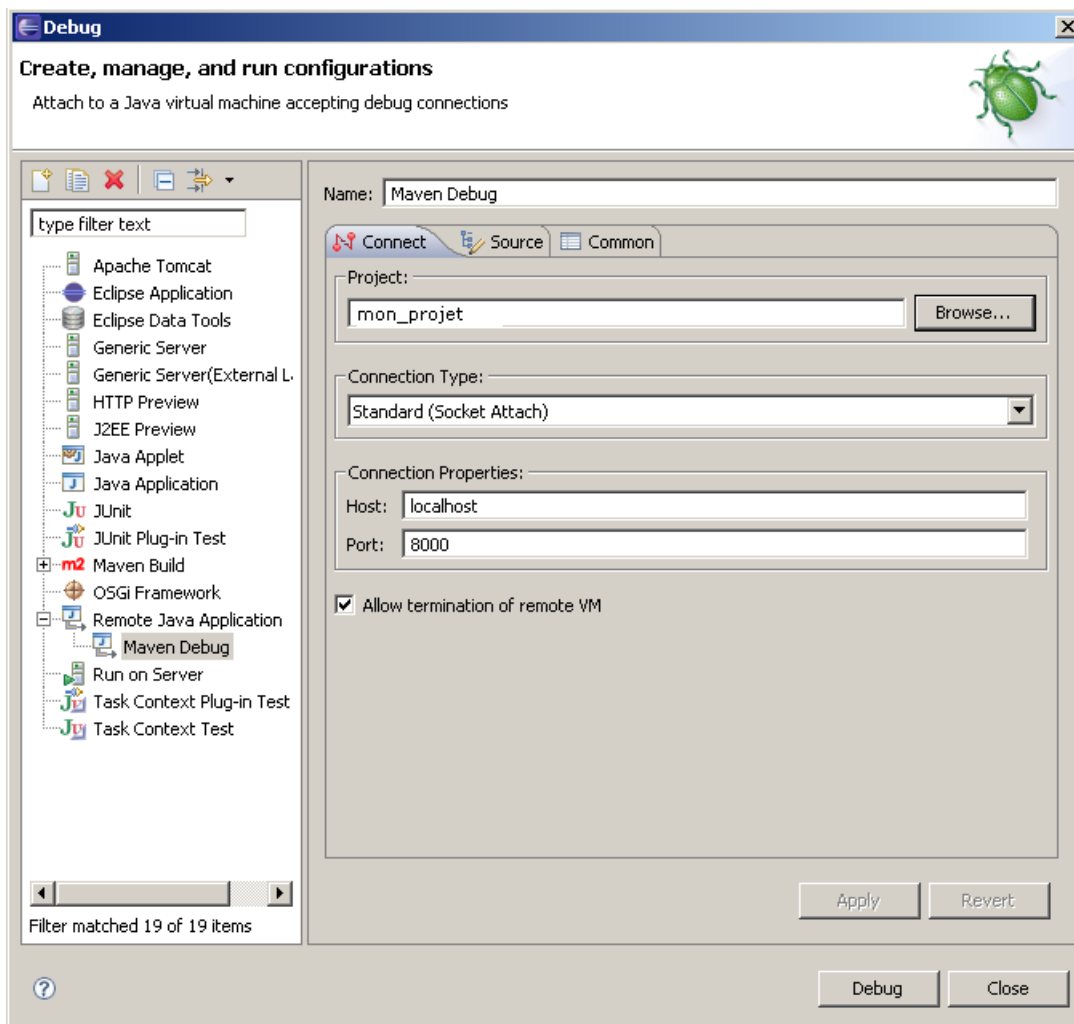
Le serveur est maintenant en écoute sur le port 8000.

VI - Création du raccourcis " Debug on Jetty "

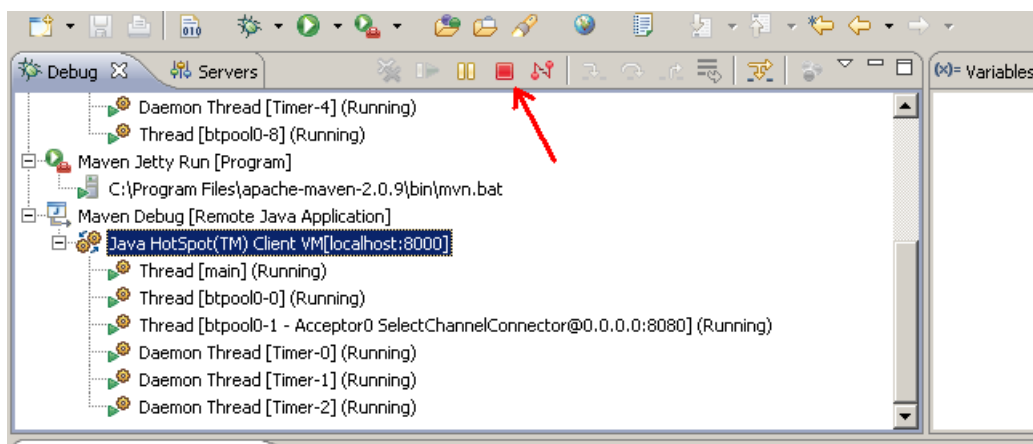
Sur le menu " Debug ", on ouvre l'écran "Debug Dialog/Configuration" :



On fait un click droit sur le nœud "Remote Java application", et on crée un nouveau nœud "Maven Debug", on spécifie le nom du projet à déboguer, et le port d'écoute 8000, afin de pouvoir s'attacher au projet qu'on a déjà démarré sur ce port.



Vous pouvez maintenant lancer le débogage en cliquant sur le bouton "Debug".



Une fois le débogage terminé, vous pouvez l'arrêter via le bouton en Rouge indiqué sur l'image ci-dessus.

⚠ Quand vous arrêtez le débogueur, il se peut que le serveur Jetty reste démarré, pour l'arrêter, il faut utiliser le gestionnaire des tâches et supprimer le processus "java.exe", et si vous essayez de relancer le serveur alors que le processus n'est pas arrêté vous aurez le message suivant :

```
Console [X]
<terminated> Maven Jetty Run [Program] C:\Program Files\apache-maven-2.0.9\bin\mvn.bat
FATAL ERROR in native method: JDWP No transports initialized, jvmtiError=JVMTI_ERROR_INTERNAL(113)
ERROR: transport error 202: bind failed: Address already in use ["transport.c",L41]
ERROR: JDWP Transport dt_socket failed to initialize, TRANSPORT_INIT(510) ["debugInit.c",L500]
JDWP exit error JVMTI_ERROR_INTERNAL(113): No transports initialized
```