

Manuel de configuration et d'installation des outils pour le cours de JEE

Table des matières

Propos.....	3
Java.....	3
Tomcat.....	3
Eclipse.....	3
Quelle version ?.....	3
Configurer l'environnement Java8 sous Eclipse.....	3
Référencer le JDK.....	4
Lier le JDK comme version par défaut.....	7
Configurer le niveau de compilation par défaut.....	7
Configurer Tomcat8.....	9
Configuration générale.....	9
Configuration de la vue « server ».....	11
Configurer un projet pour travailler.....	14
Configurer le java build path.....	15
Vérification du compilateur.....	16
Configurer les options de développements.....	17
Formatage du code.....	17
Formatage automatique.....	17
Les options d'éditations.....	19
Supprimer la limite de la taille de la sortie de la console.....	19
Choix du navigateur.....	20
Chrome et Chrome Developer.....	20
Firefox et Firebug.....	20
Internet Explorer 11.....	20
Ne pas utiliser le navigateur d'Eclipse.....	20

Propos

Ce document regroupe les configurations intéressantes à connaître pour travailler sur le cours de JEE.

Les sections sont découpées en fonction de l'outil concerné.

Java

Pour le cours nous utiliserons JSE8, téléchargeable ici :

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>.

Téléchargez l'installateur et suivez les instructions si celui-ci ne se trouve pas sur votre poste de travail.

Notez qu'il s'agit d'un JDK et pas une JRE seule.

Tomcat

Le serveur applicatif, ou plus exactement container Web, utilisé pour le cours sera Tomcat 8.

Disponible ici : <http://tomcat.apache.org/download-80.cgi>.

Il implémente des JSR de JEE7 qui nous intéressent pour le cours, comme vous pouvez le voir sur le lien suivant :

<http://tomcat.apache.org/whichversion.html>

Eclipse

Eclipse est un IDE. Toutefois la configuration par défaut n'est pas forcément pleinement fonctionnelle.

Point important choisissez bien votre Workspace pour éviter d'avoir à exporter vos configuration en cas de changement.

Quelle version ?

STS, *Spring Tool Suite*, est un outil basé sur Eclipse mais qui a l'avantage de contenir de nombreux plugins en plus de ceux de JEE classique. J'ai une préférence pour celui-ci. Le lien est disponible ici <http://spring.io/tools/sts/all>.

Configurer l'environnement Java8 sous Eclipse

Pour pouvoir réaliser cette étape de configuration, il est nécessaire d'avoir effectué toutes les étapes précédentes : installation de java 8, de Tomcat 8 de STS.

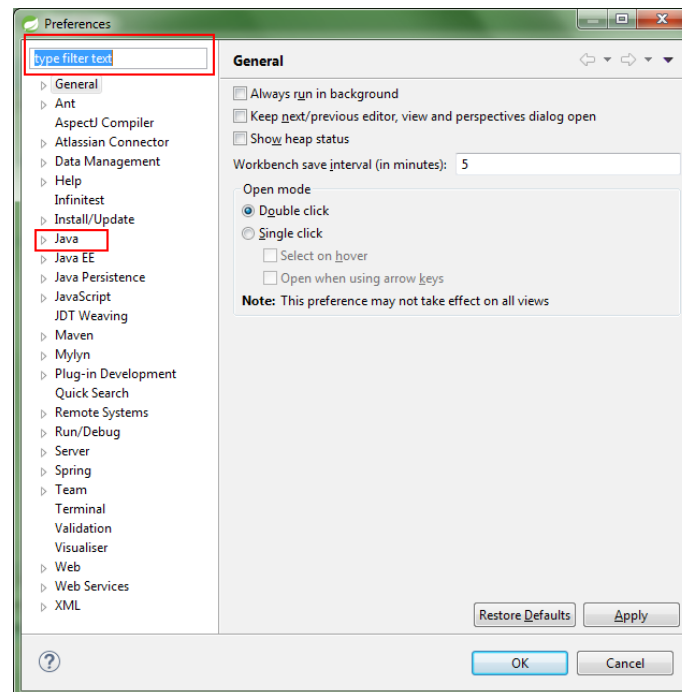
Dans STS nous avons deux niveaux de configuration : les préférences générales et la configuration propre au projet. En configurant correctement les préférences générales, nous diminuons considérablement le besoin de spécifier des configurations pour les projets.

Suivre ce tutoriel vous fera gagner du temps.

La première étape consiste à référencer un JDK

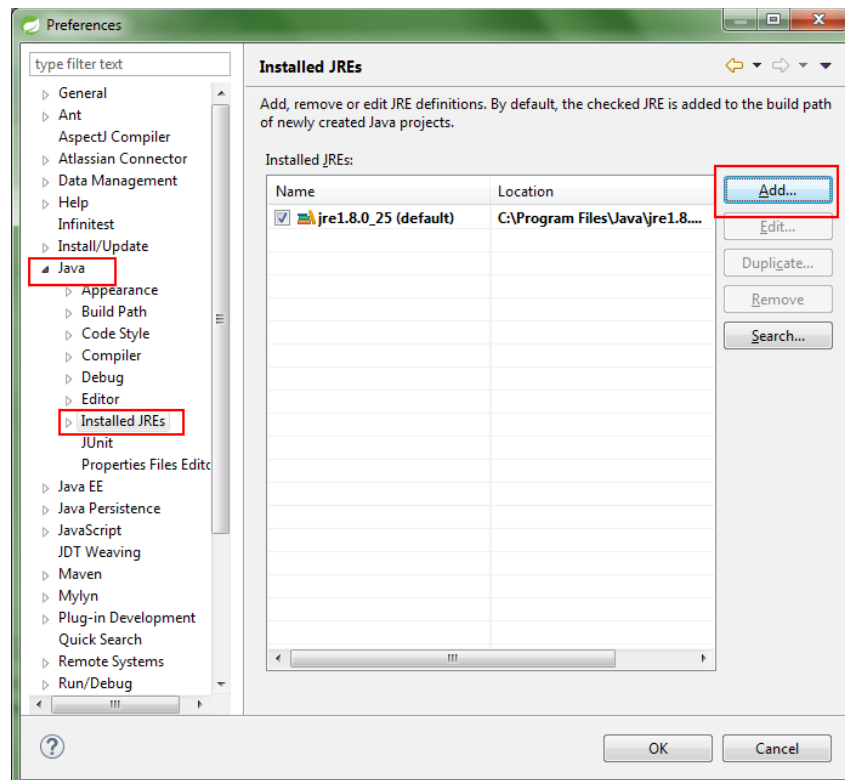
Référencer le JDK

Pour référencer le JDK il faut se rendre dans les préférences d'Eclipse dans Windows->Preferences. La capture suivante montre l'écran des préférences générales :

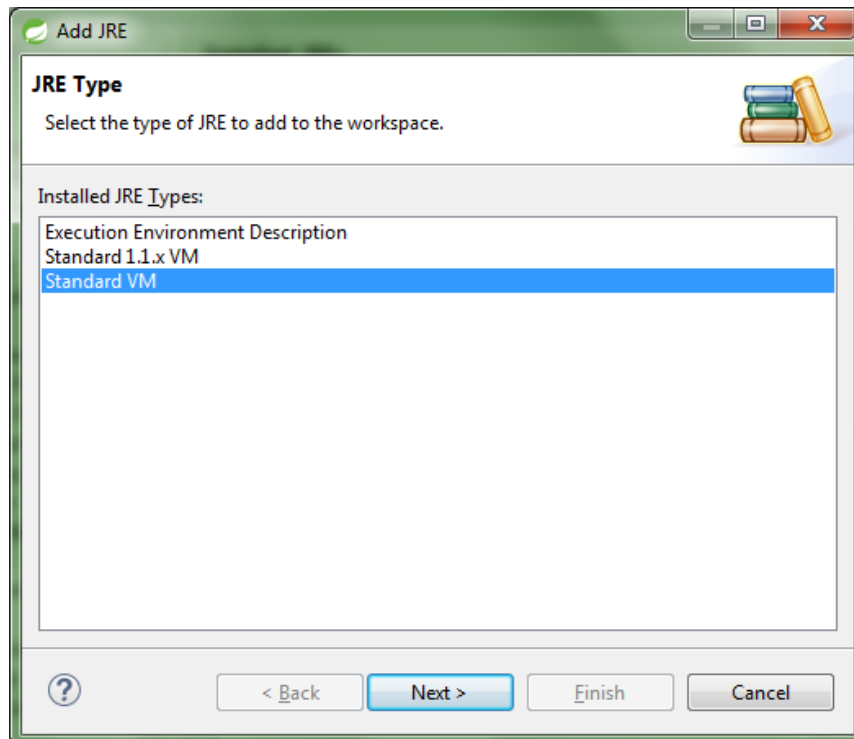


En haut à gauche, un champ de recherche, pratique, car le nombre d'option est très important dans STS.

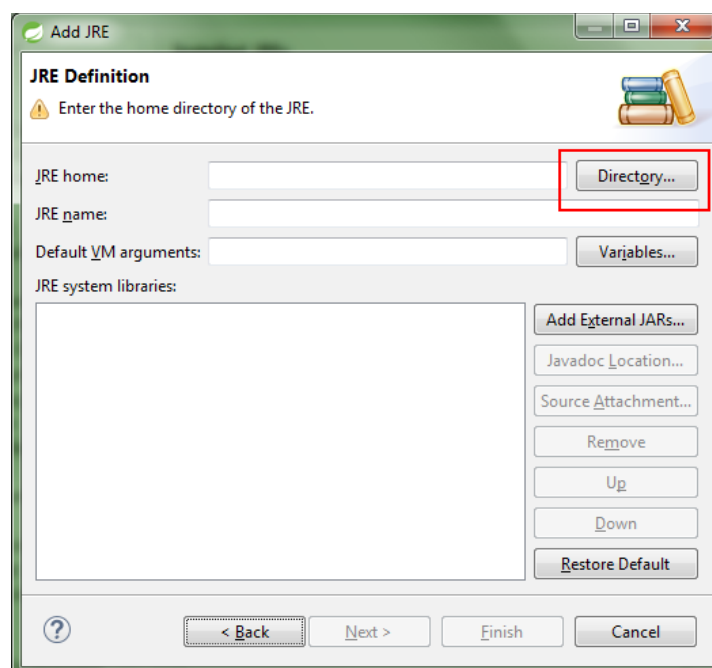
Nous allons nous rendre dans les options de Java pour notre configuration : Java->installed JRE's. Comme vous pouvez le voir sur la capture seul un JRE8 est installé et nous voulons un JDK.



En cliquant sur le bouton « add », vous obtenez un écran de choix des JRE à utiliser. Laissez l'option « standard VM », sélectionnée par défaut :

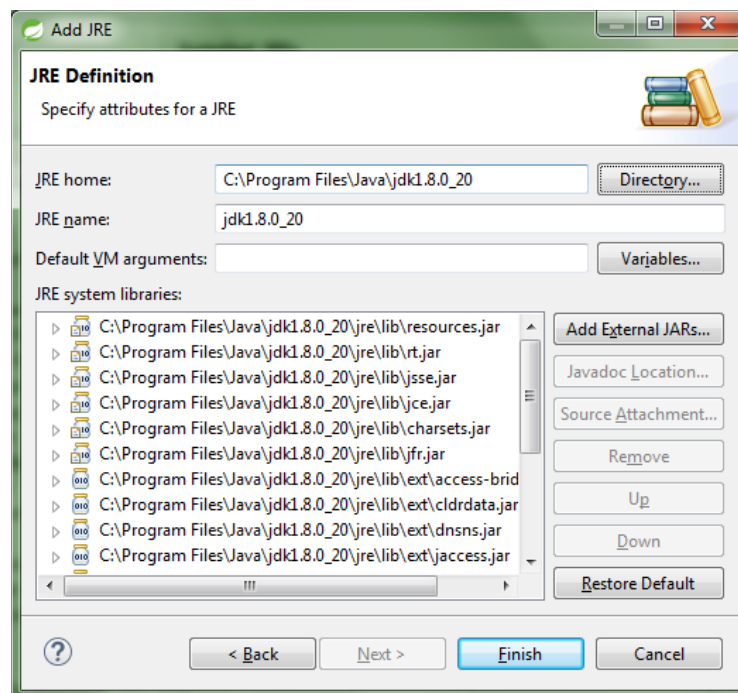


En cliquant sur next vous obtenez l'écran suivant :



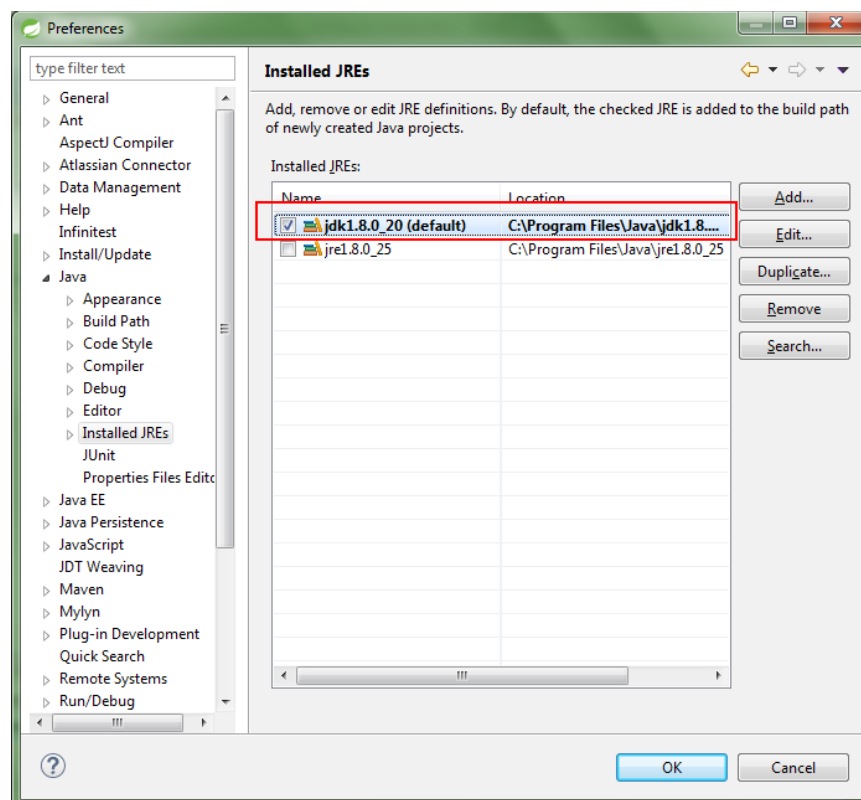
Cliquez sur le bouton « directory » et sélectionnez le répertoire d'installation de votre JDK.

Après avoir choisi le répertoire du JDK, vous devez obtenir l'écran suivant :



Il est possible que vous ayez une version plus récente au moment du cours, ce ne sera pas gênant pour l'utilisation que nous aurons de Java.

Cliquez sur « finish » pour terminer et sélectionnez votre JDK comme suit pour la passer comme runtime par défaut.

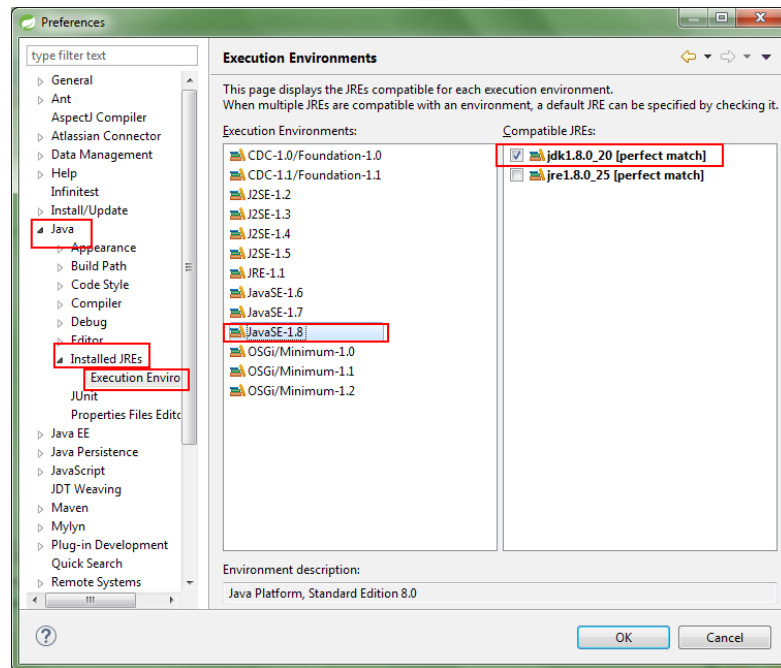


Cliquez sur OK pour fermer les préférences avant de passer à l'étape suivante.

Lier le JDK comme version par défaut

Pour passer étape il est nécessaire de repasser par Windows->Preferences pour lier votre JDK 8 au java par défaut (JSE8). Cette étape si elle n'est pas obligatoire permet de se prémunir contre le changement de version mineur du JDK.

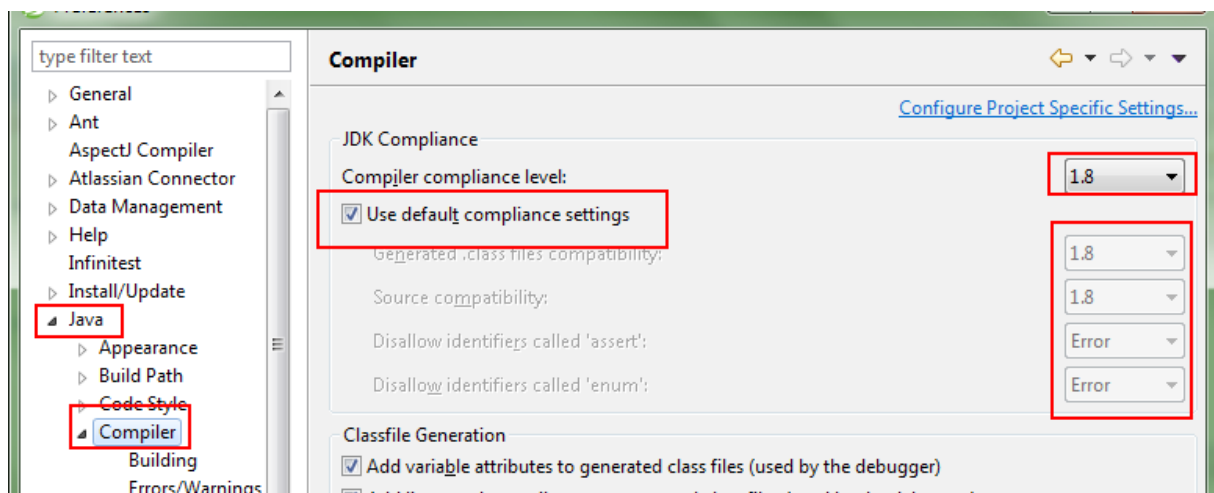
Rendez-vous dans Java->Installeds JRE's->Execution Environnement. Sélectionnez JavaSE-1.8 et dans la liste des JRE compatible et sélectionnez le JDK préalablement installé.



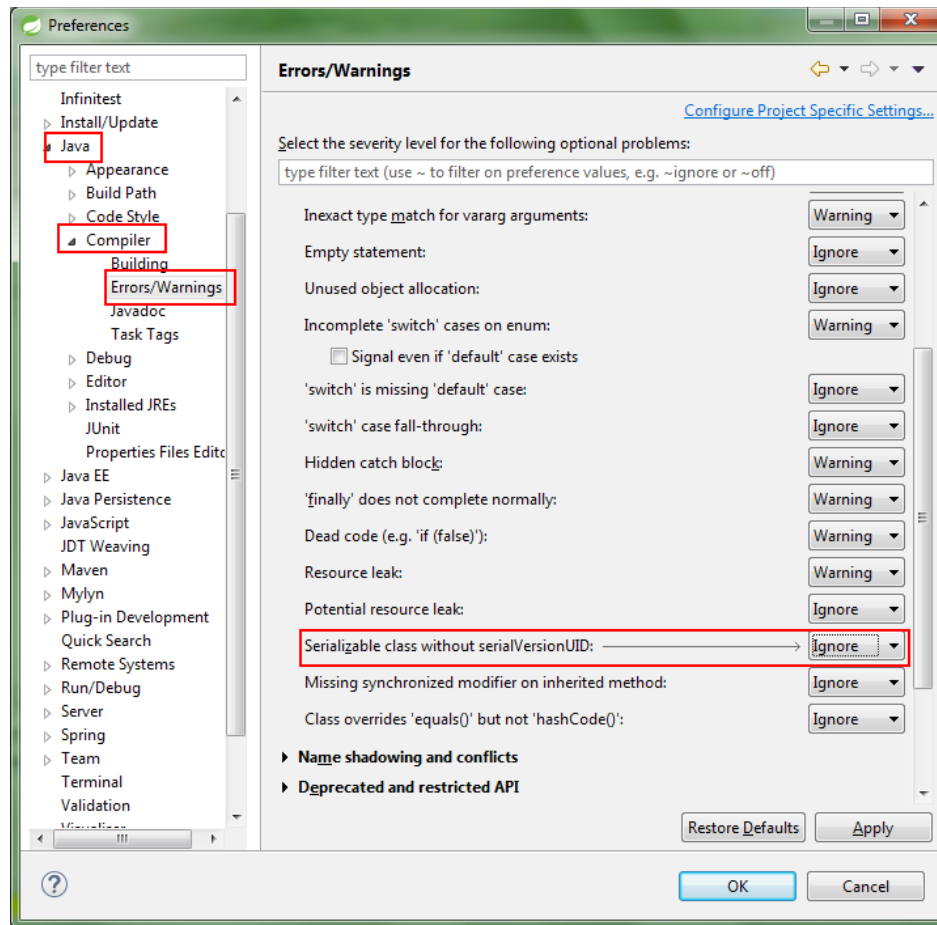
Configurer le niveau de compilation par défaut

Configurer le niveau de compilation par défaut permet de s'assurer que les projets soient toujours compilés en Java8 plutôt que de reconfigurer chaque projet.

Pour cela allez dans Java->Compiler pour afficher l'écran suivant, qui doit être comme suit :



En même temps, profitons-en pour modifier une option de compilation qui est assez frustrante et sans grand intérêt dans nos cas, la désactivation des warning pour les id de sérialisation manquants (qui crée un warning pour classe implémentant Serializable ce qui est le cas des HttpServlet par exemple). Il se situe dans la catégorie « Potential programming problems » :

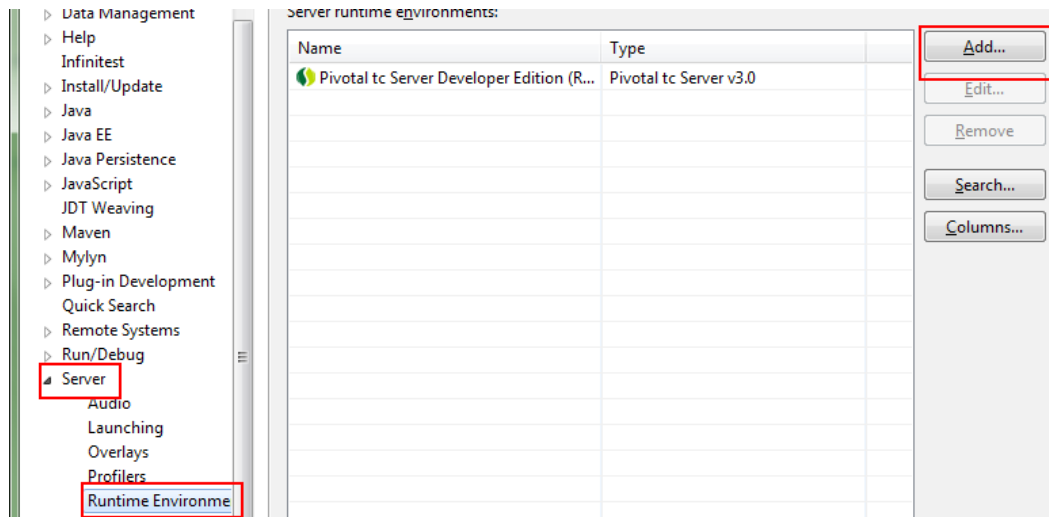


Nous avons maintenant une configuration de Java8 correct dans Eclipse. Il est temps configurer le serveur sur lequel nous allons déployer les projets : Tomcat8.

Configurer Tomcat8

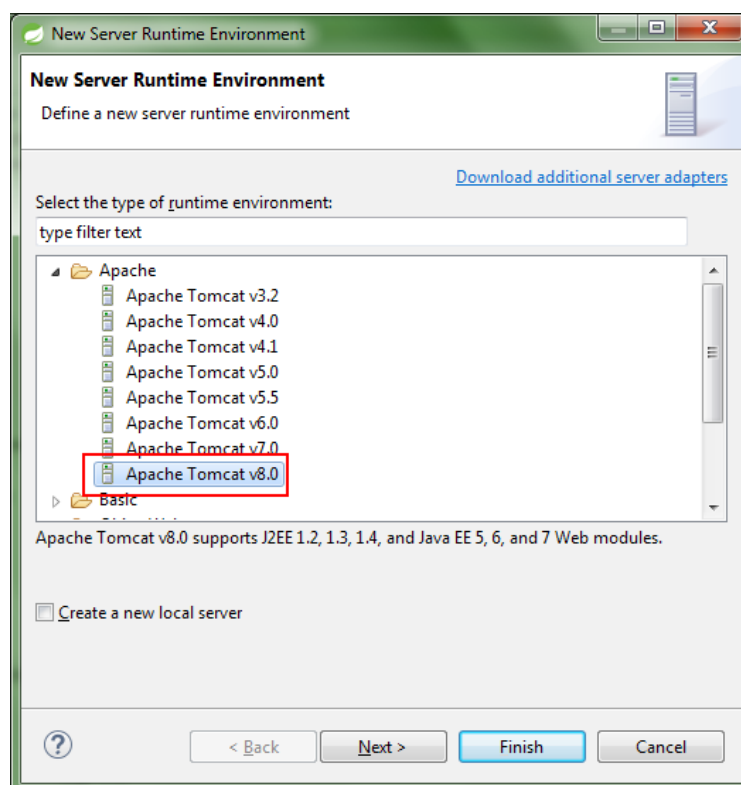
Configuration générale

Dans la première section nous avons téléchargé Tomcat8 qui doit être dézippé dans un répertoire du poste. Il faut maintenant le référencer sur Eclipse : De nouveau utilisez Windows->Preferences puis Server->RuntimeEnvironment



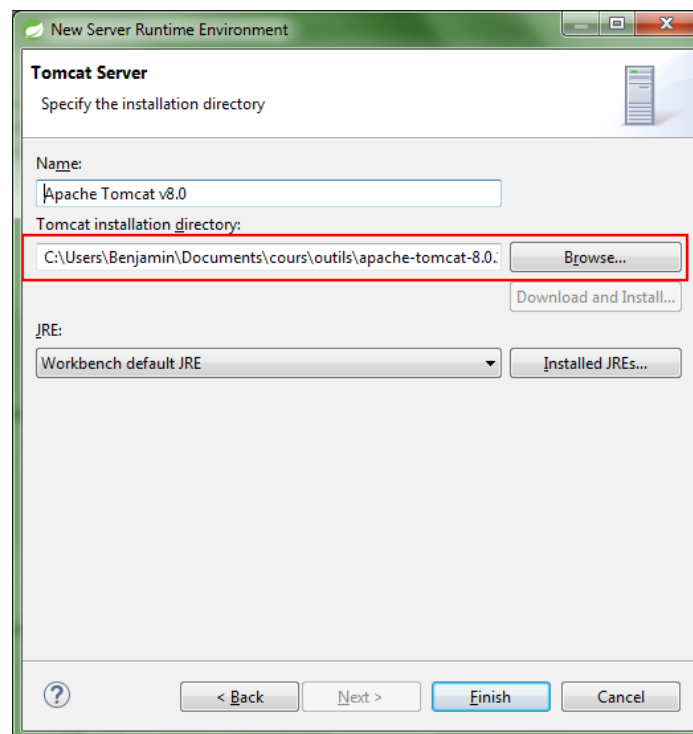
STS embarque un serveur par défaut que vous pouvez voir dans la liste.

Cliquer sur add pour afficher l'écran suivant :



Après avoir choisi le connecteur « Apache Tomcat v8.0 » cliquez sur « next » pour pouvoir aller le chercher sur son emplacement sur le disque.

Un clic sur le bouton finish permettra de fermer la fenêtre une fois que vous aurez sélectionné le chemin :



Un clic sur finish permet de valider définitivement l'installation du serveur dans les préférences générales.

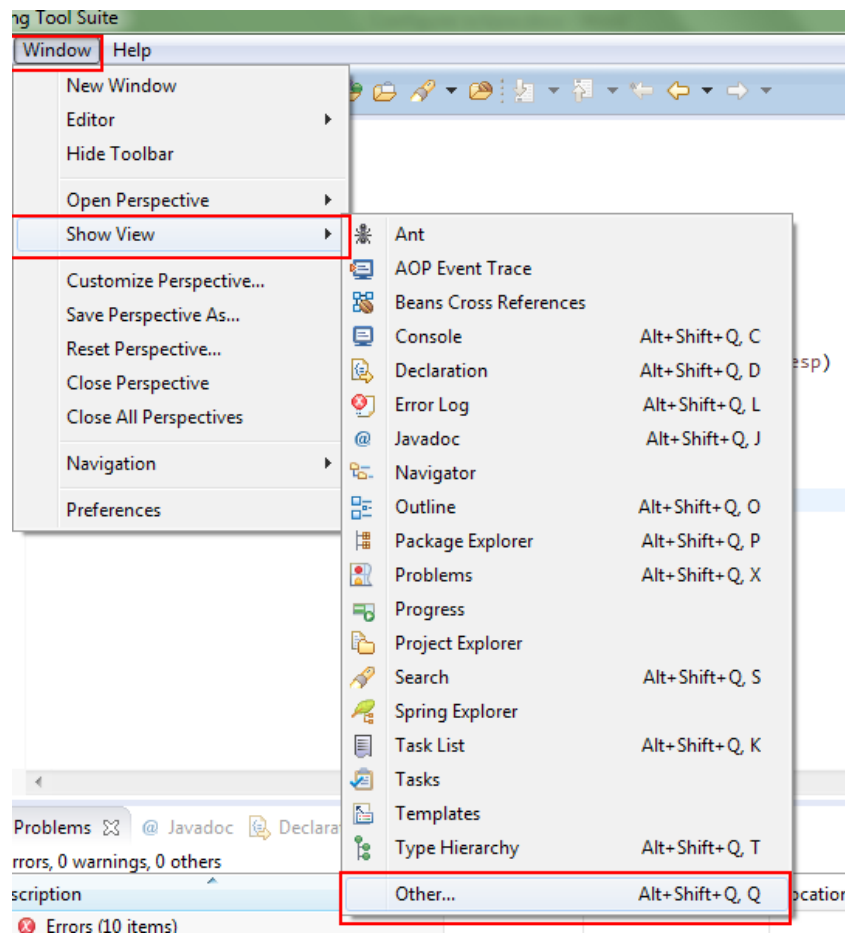
Configuration de la vue « server »

Eclipse propose un système de perspective « Java », « Jee », « Debug » avec chacun ses propres vues.

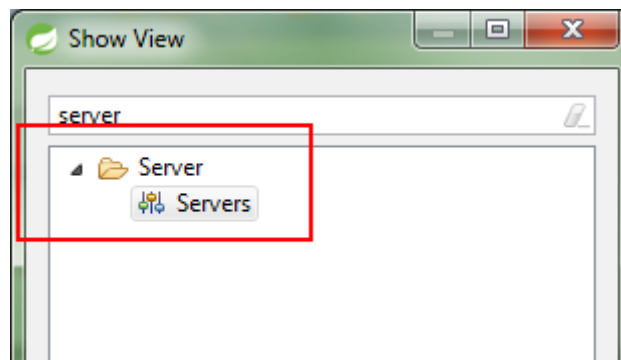
Je vous conseille d'utiliser la perspective Java pour vos développements elle est plus agréable que la perspective JEE – le changement de perspective peut se faire en haut à droite juste en dessous de la croix de fermeture de l'IDE.

Toutefois, la perspective Java ne possède pas par défaut la vue server qu'il faut ajouter manuellement. La vue server permet de piloter vos serveurs applicatifs depuis Eclipse et déployer vos applications pour les tester. Ce point est évoqué dans le premier cours :

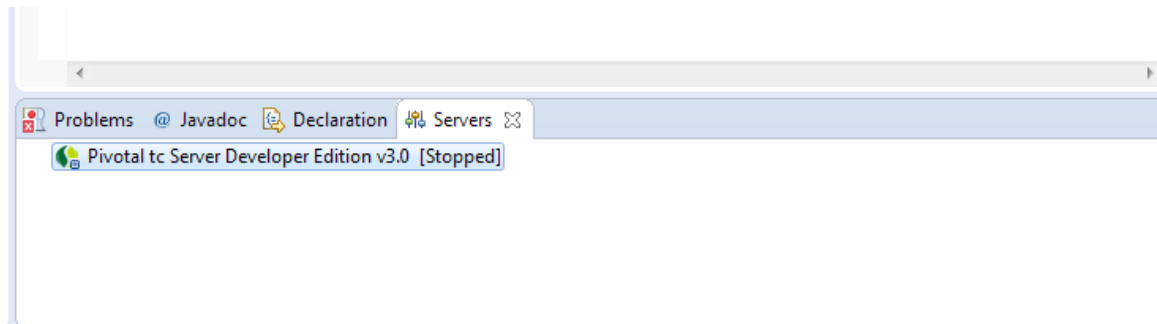
Pour pouvoir ajouter la vue allez dans window->show view->other comme dans la capture suivante



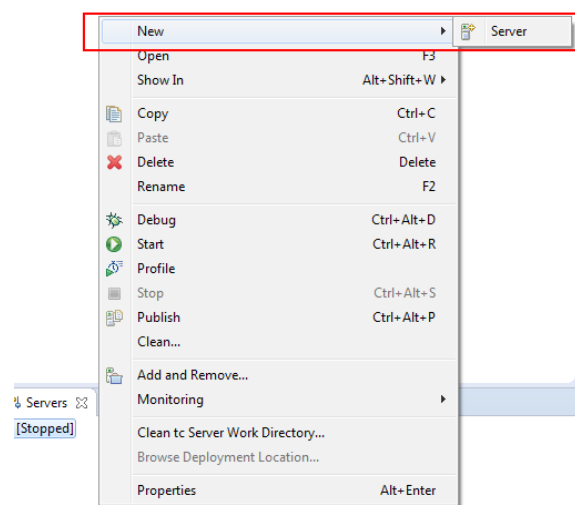
S'affiche alors une fenêtre où vous pouvez saisir « server » pour retrouver la vue en question :



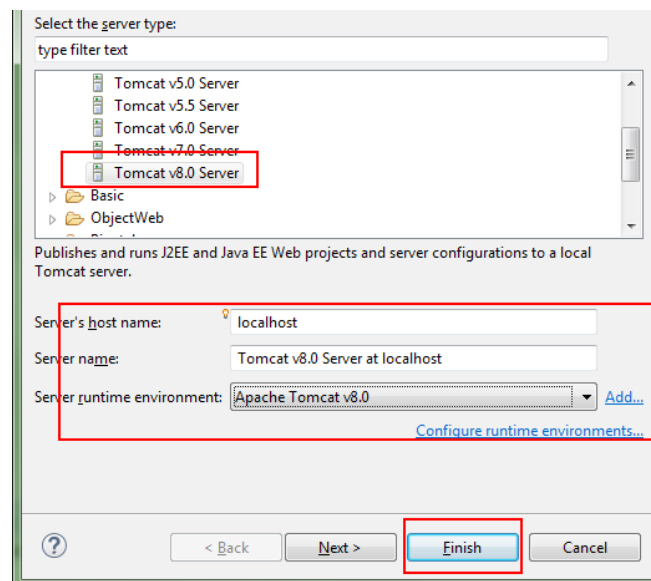
Il ne nous reste plus qu'à ajouter notre serveur Tomcat (sous STS il en existe déjà un) :



Faites un clic droit dans l'espace blanc pour ajouter le serveur



S'ouvre alors l'écran suivant qui permet d'ajouter le serveur tomcat 8 :

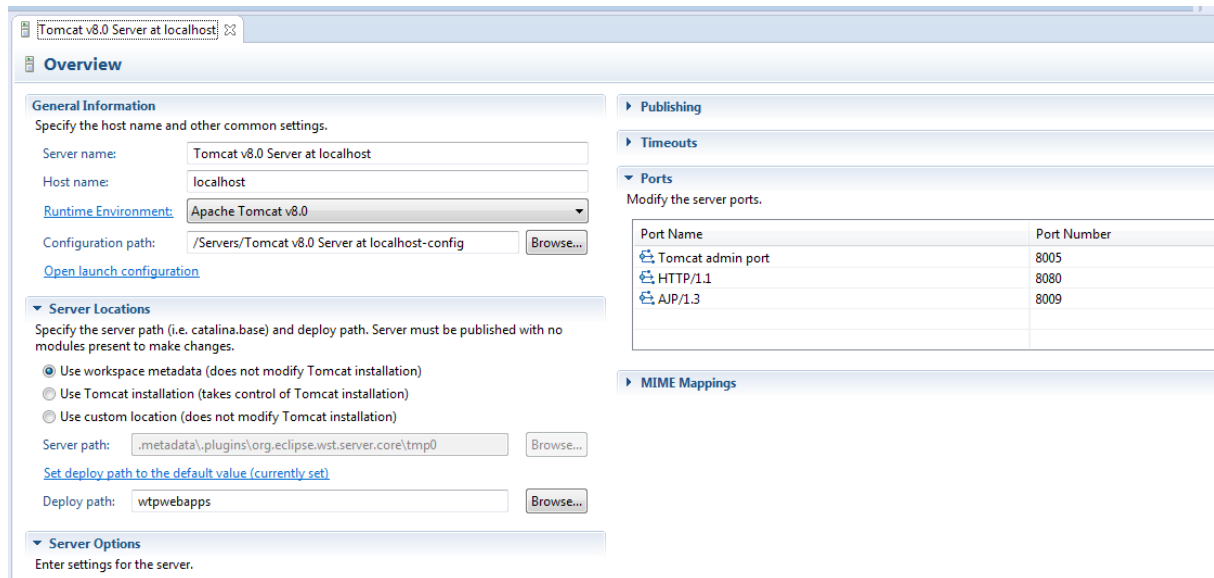


Un clic sur finish et celui-ci apparait dans la vue server.

Comme vous pouvez le voir, il y'a trois boutons en haut à droite « debug », « run » et « profile ».
Nous n'utilisons pas ce dernier durant le cours.

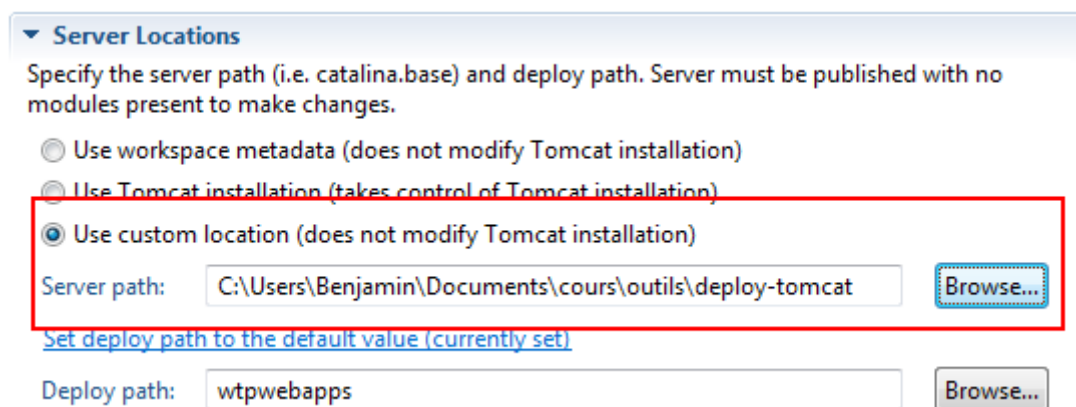


Un double clic sur le serveur Tomcat permet d'afficher des préférences d'utilisation :



Il nous reste un dernier élément à configurer il s'agit du « server location » que nous allons mettre à custom. Cela nous permettra de voir plus facilement les ressources déployés par Eclipse pour lancer nos applications.

Il suffit de choisir un emplacement :



Deux points importants :

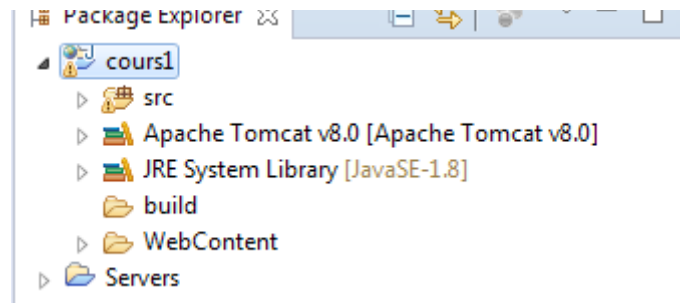
- d'une part il faut que le serveur n'ait aucune application attachée sinon configurer l'emplacement est impossible
- dans toute la durée du cours, il ne faut pas faire d'autre modification de configuration du tomcat (à l'exception des numéros de port en cas de problème avec les ports par défaut).

Note vous pouvez copier les .metadata dans votre workspace pour importer les informations.

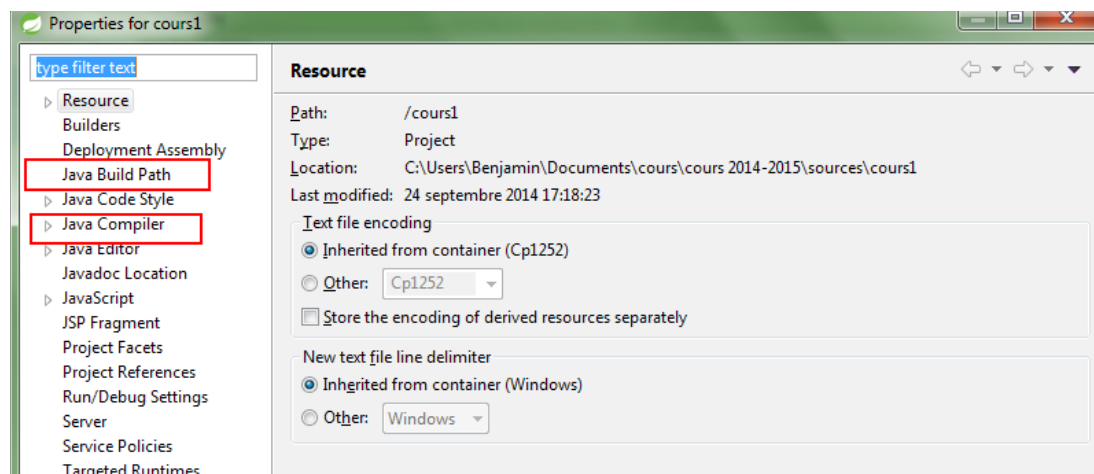
Configurer un projet pour travailler

Après avoir configuré tous les éléments précédents vous pouvez enfin créer votre nouveau projet web dynamique tels qu'il est montré dans le cours. Si vous avez déjà créé un projet avant, supprimez-le.

Le projet utilisé pour les exemples suivant sera le projet cours1 qui contient notre servlet HelloWorld.



En appuyant sur alt+entrée (ou clic droit properties), on affiche les préférences du projet. Deux options doivent retenir votre attention :



Java Build Path qui doit contenir les références du projet et le Java Compiler qui permet de définir le niveau de compilation du projet.

Java compiler qui contient les options de compilation que nous spécifierons toujours de manière globale.

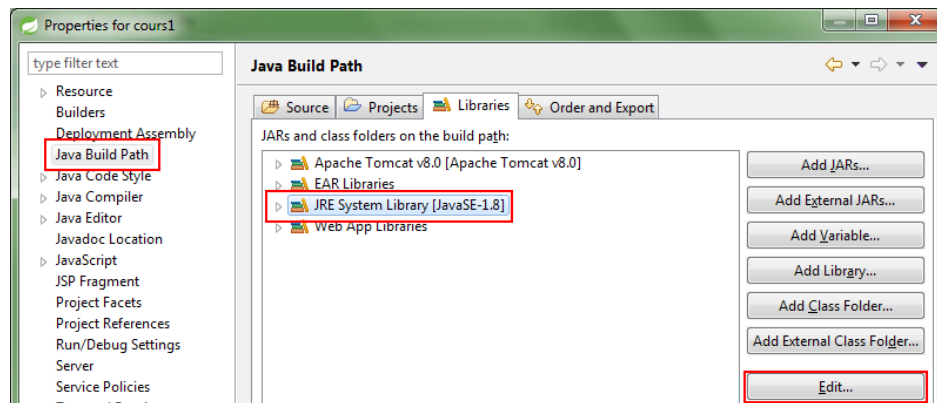
Configurer le java build path

Le java build path définit toutes les bibliothèques, projets et sources dont dépend votre projet.

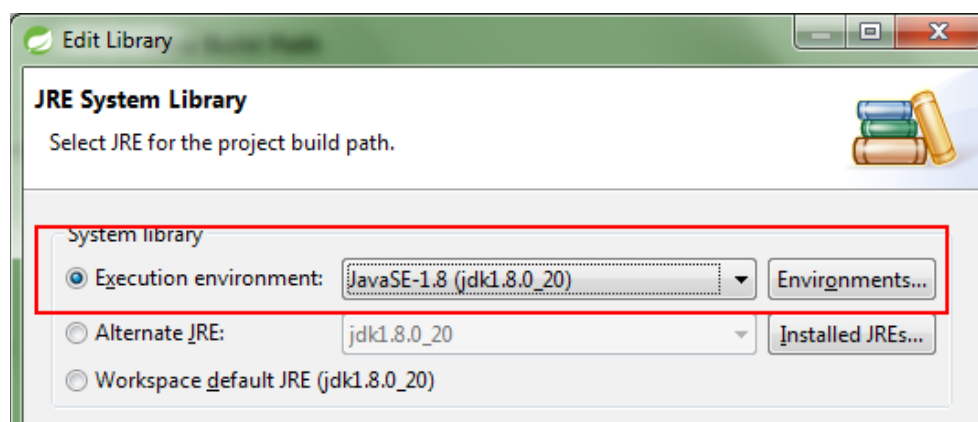
Dans l'onglet « libraries » se trouvent des entrées dont vous devez comprendre la signification.

L'entrée « JRE System Library »

Cette entrée permet de définir quelle version de l'API Java sera utilisée pour le programme. Ce n'est pas la version du compilateur mais il est préférable de mettre la même version que celle du compilateur en l'occurrence Java 8. Il serait possible d'utiliser une version précédente mais cela nous priverait simplement des nouvelles fonctions (l'API Java étant rétrocompatible ce n'est pas gênant).

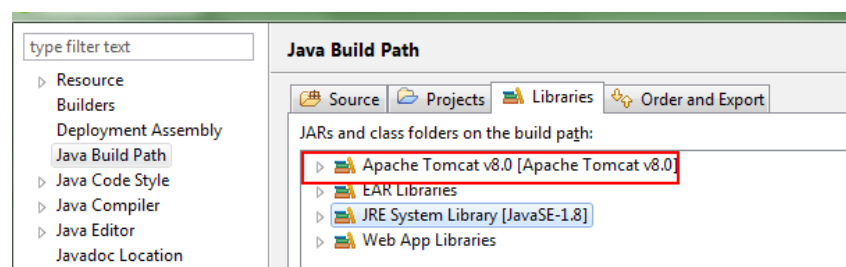


Si l'entrée n'est pas JavaSE-1.8, comme dans la capture cliquez sur le bouton « edit » pour ouvrir l'écran suivant en sélectionnant bien l'environnement qui convient :



L'entrée « Apache Tomcat v8.0 »

Cette entrée est obligatoire pour faire du JEE. Rappelons-nous JEE, est une spécification mais aussi un framework et il faut posséder une implémentation pour pouvoir travailler. Chaque serveur applicatif fournit une implémentation. Nous avons décidé de déployer sur Tomcat, nous allons donc utiliser l'implémentation de Tomcat :

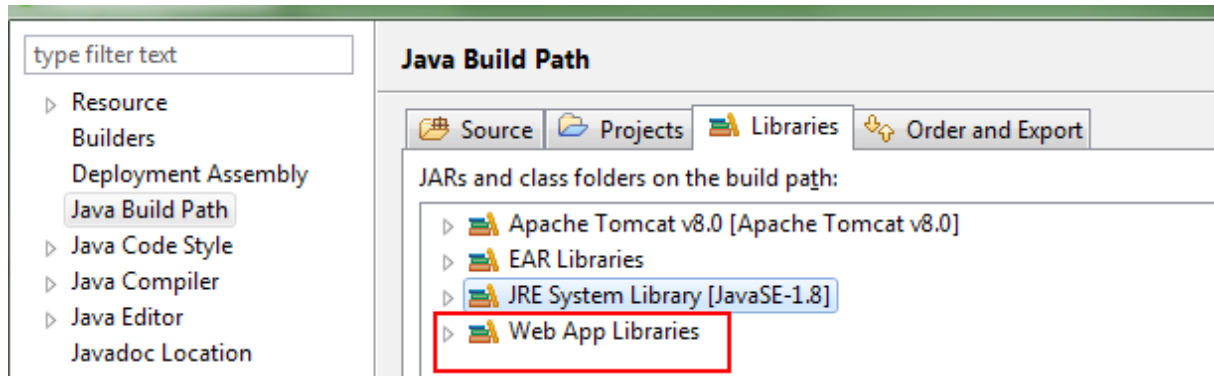


L'entrée « EAR Libraries »

Cette entrée ne sert à rien dans notre contexte. Vous pouvez la supprimer si vous le souhaitez.

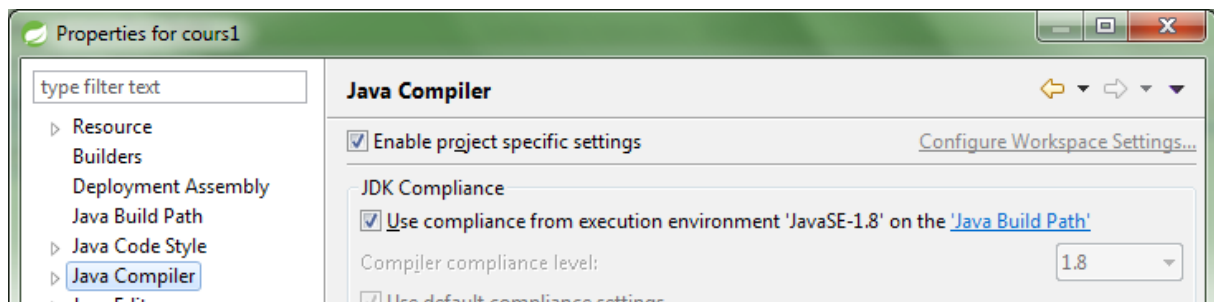
L'entrée « Web App Libraries »

Les applications Java utilisent le JDK et l'API JEE. Les applications sont souvent enrichies de Framework extérieur (sous forme de Jar). Les applications Web ont un répertoire (WebContent/WEB-INF/lib) pour stocker ces bibliothèques. Cette entrée permet de référencer tous les jars que vous aurez positionnés dans ce répertoire.



Vérification du compilateur

Vérifions également que le projet ait bien la compilation par défaut, sinon changez les options pour obtenir l'écran comme suit :



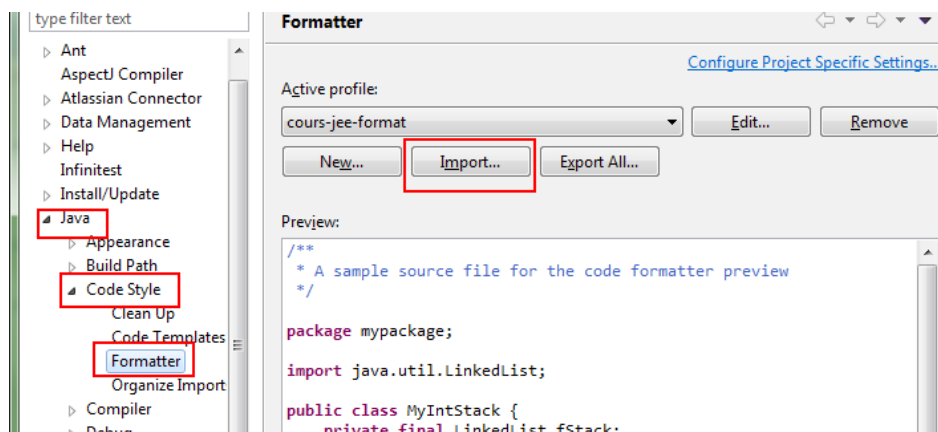
Configurer les options de développements

En plus des configurations que nous avons vues plus haut, il existe quelques options qui rendent plus confortable le développement. Avoir un format de code identique entre tous les développeurs permet d'éviter des problèmes lors des comparaisons de fichiers sources par exemple.

Formatage du code

Avec ce cours, vous pouvez accéder à un formatage unique. Cela évite des écarts qui nuisent à la lisibilité des comparaisons. Utilisez le fichier en pièce jointe pour le configurer.

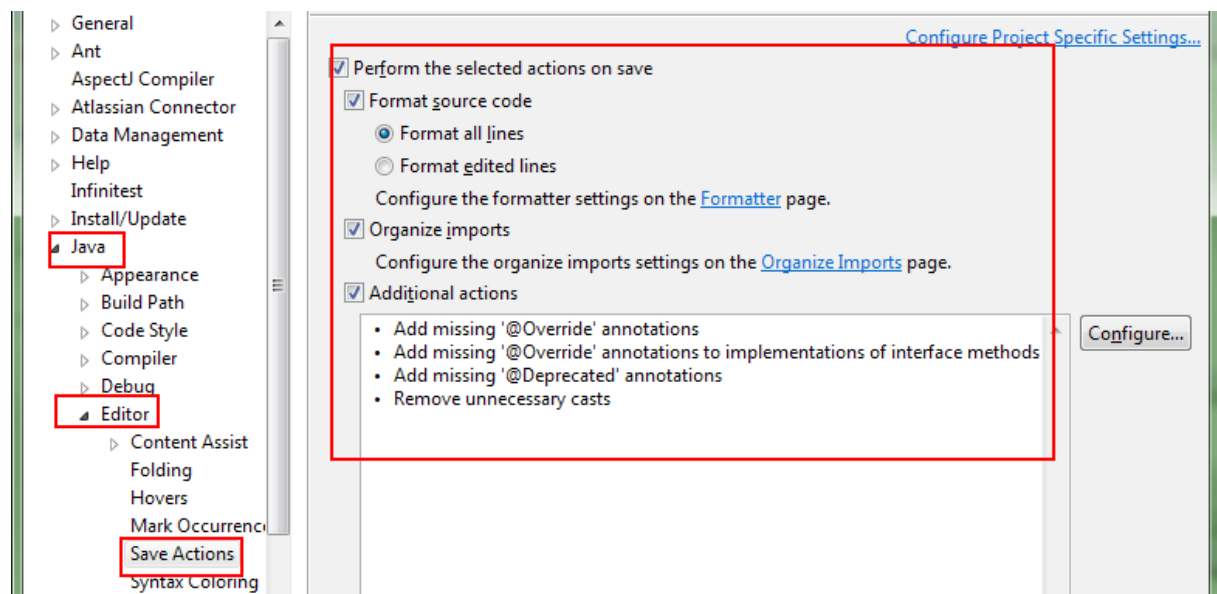
Allez dans windows->preferences->java->code style. En cliquant sur import vous pouvez choisir d'utiliser un fichier de format existant. Télécharger le sur <https://github.com/bkuchcik/cours-jee-isima/blob/master/documentations/conf-eclipse/format-cours-jee.xml>



Formatage automatique

Une option permet de faire des actions au moment de sauvegarder les fichiers. Impossible d'oublier des commandes répétitives.

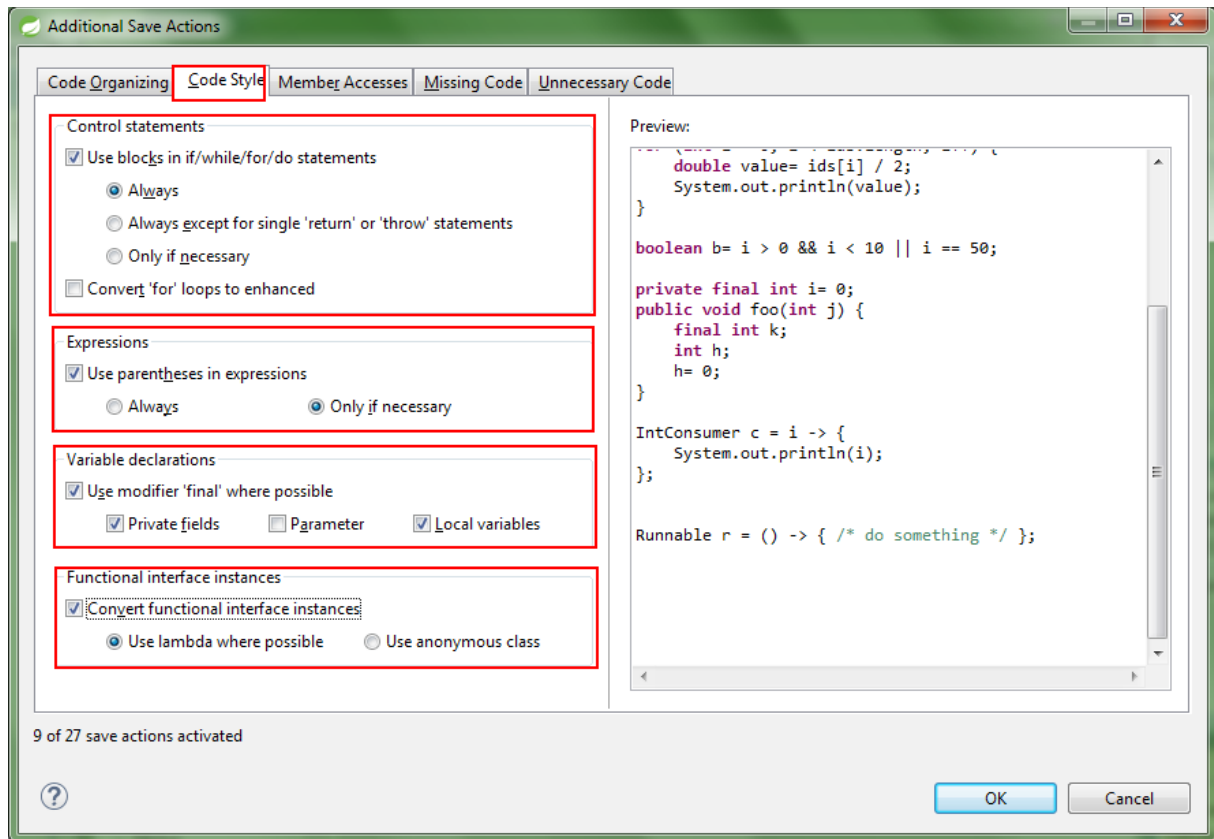
Allez dans windows->preference->Java -> editor -> Save Actions :



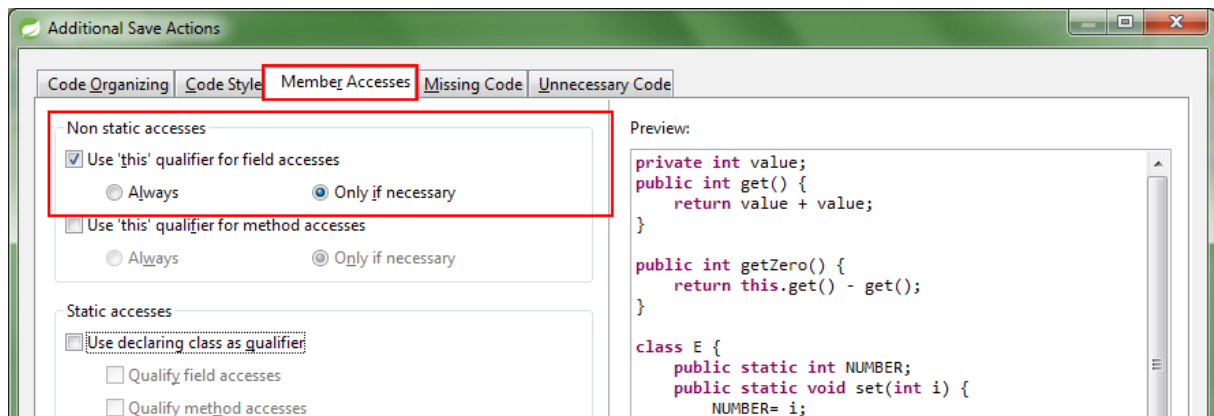
Sélectionnez tous les éléments que vous voyez dans la fenêtre.

Cliquez ensuite sur configurer d'autres éléments :

Sur l'onglet « code style », activez les options comme suit :



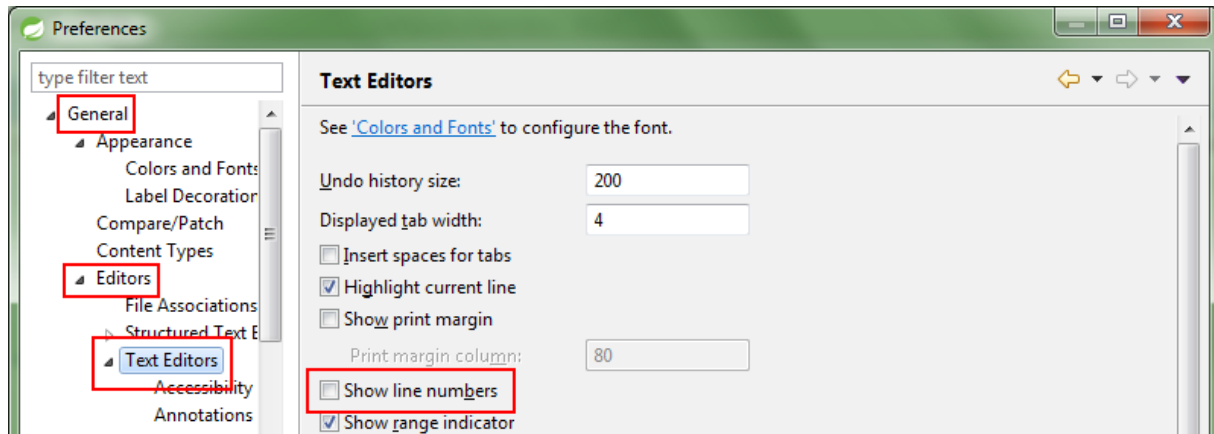
Sur l'onglet member access :



Les options d'éditeurs

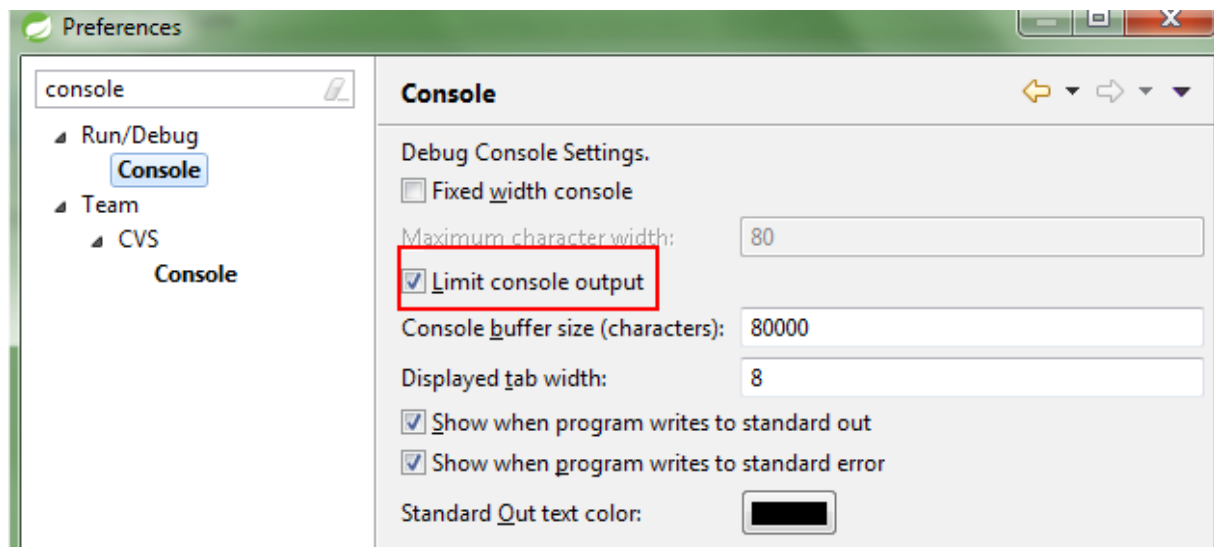
Par défaut Eclipse n'affiche pas dans les éditeurs le numéro des lignes ce qui est frustrant, surtout quand vous êtes amené à lire un stacktrace.

Corrigeons ce problème en cochant l'option « show line numbers » que l'on trouve dans Windows->Preferences puis General->Editors->Text editors



Supprimer la limite de la taille de la sortie de la console

La console permet d'afficher les traces du serveur, ou des programmes exécutés. Par défaut Eclipse en limite la taille ce qui dans certains tests peut-être ennuyeux puisque nous pouvons perdre des infos sur l'exécution. Tapez « console » dans le filtre et décochez la case suivante en rouge :



Choix du navigateur

Les applications Web sont accessibles depuis des navigateurs. Depuis plusieurs années, des outils de développement ce sont multipliés. Toutefois pour le cours, je vous demanderais de suivre les recommandations évoquées dans les chapitres suivants.

Chrome et Chrome Developer

Vous pouvez utiliser Chrome couplé à Chrome Developer. C'est n'est pas l'outil que je préconise en priorité car il manque quelques fonctions par rapport à Firebug.

Firefox et Firebug

Le couple Firefox/Firebug a ma préférence car l'outil est pratique notamment grâce à sa fonction de recherche que je trouve très efficace.

Internet Explorer 11

IE traîne une réputation, justifiée jusqu'à peu, de navigateur à problème. Toutefois des efforts considérables ont été apportés sur les dernières versions (dont la 11). Vous pouvez également utiliser l'outil de développement embarqué à ce navigateur.

Ne pas utiliser le navigateur d'Eclipse

Eclipse permet d'afficher les pages Web en utilisant le moteur de rendu défini dans les préférences.

Toutefois, pendant le cours je ne souhaite pas que vous utilisiez cette solution mais un navigateur externe.