

Eclipse et Tomcat

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille
Chercheur en Programmation par contrainte (IA)
Ingénieur en Génie logiciel

`elmouelhi.achref@gmail.com`

Plan

- 1 Auto-complétion
- 2 Auto-save
- 3 Désactiver la vérification d'orthographe
- 4 Génération du code de classe
- 5 Génération du `.JAR`
- 6 Bug d'Eclipse avec les projets JEE
- 7 Bug d'Eclipse avec les projets JEE créés avec Maven
- 8 Bug d'Eclipse avec les projets Web Spring
- 9 Problème de port avec Apache Tomcat

Auto-complétion

Comment activer l'Auto-complétion sous Eclipse

- Dans le menu `Window`, choisir `Preferences`
- Aller dans `Java` et étendre la section `Editor`
- Cliquer sur `Content Assist`
- Dans la rubrique `Auto Activation` , remplacer le contenu de la zone texte correspondante à `Auto activation triggers for java` par
`abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ.`
- Valider en cliquant sur `Apply and Close`

Auto-save

Comment activer l'enregistrement automatique

- Dans le menu `Window`, choisir `Preferences`
- Aller dans `General` et étendre la section `Editors`
- Cliquer sur `Autosave` et aller cocher la case `Enable autosave for dirty editors`
- Saisir la durée
- Valider en cliquant sur `Apply and Close`

Auto-save

Comment désactiver la vérification d'orthographe

- Dans le menu `Window`, choisir `Preferences`
- Aller dans `General` et étendre la section `Editors`
- Étendre `Text Editors` puis cliquer sur `Spelling`
- Décocher la case `Enable spell checking`
- Valider en cliquant sur `Apply and Close`

Génération du code de classe

Après avoir créé une classe et déclarer ses attributs, **Eclipse** nous offre la possibilité de générer :

- un ou plusieurs constructeurs avec et/ou sans paramètres
- les getters et/ou setters pour tous (ou quelques) attributs
- la méthode `toString()` en fonction des attributs sélectionnés
- les méthodes déléguées (sur les attributs de type `Collection`)
- ...

Génération du code de classe

Après avoir créé une classe et déclarer ses attributs, **Eclipse** nous offre la possibilité de générer :

- un ou plusieurs constructeurs avec et/ou sans paramètres
- les getters et/ou setters pour tous (ou quelques) attributs
- la méthode `toString()` en fonction des attributs sélectionnés
- les méthodes déléguées (sur les attributs de type `Collection`)
- ...

Pour cela

Aller dans le menu source et choisir la fonctionnalité désirée en cliquant sur `Generate...`

Génération du .JAR

.JAR : Java Archive

un fichier compressé (comme un exécutable Windows) utilisé pour partager un ensemble de classes Java.

Comment le générer ?

- Faire clic droit sur le nom du projet
- Choisir `Export`
- Chercher puis sélectionner `Jar file` et cliquer sur `Next`
- Choisir le nom et l'emplacement du fichier `.JAR` puis cliquer sur `Finish`

Bug d'Eclipse avec les projets JEE

Bug

Eclipse ne reconnaît plus les classes de base comme `String`, `List`, `Exception`... : tout est souligné en rouge

Solution

- Faire clic droit sur le nom du projet
- Choisir `Properties`
- Chercher puis sélectionner `Java Build Path`
- Dans `Libraries`, faire double-clic sur `JRE System Library`...
- Cocher `Alternate JRE` : et choisir la JRE installée sur la machine
- Cliquer sur `Finish` puis valider en cliquant sur `Apply and Close`

Bug d'Eclipse avec les projets JEE créés avec Maven

Bug

Eclipse signale une erreur dans le fichier `index.js` sans souligner une seule ligne

Solution

- Faire clic droit sur le nom du projet
- Choisir `Properties`
- Chercher puis sélectionner `Targeted Runtimes`
- Cocher la case `Apache Tomcat vX.X` puis cliquer sur `Apply and Close`

Bug d'Eclipse avec les projets JEE créés avec Maven

Bug

Il n'y a pas de `src/main/java` dans Java Resources

Solution

- Faire clic droit sur le nom du projet
- Aller dans Build Path > Configure Build Path...
- Cliquer sur Order and Export
- Cocher les trois case Maven Dependencies, Apache Tomcat vX.X et JRE System Library
- Cliquer sur Apply and Close

Bug d'Eclipse avec les projets Web Spring

Bug

Eclipse ne reconnaît plus

```
org.springframework.web.context.ContextLoaderListener
```

Solution

- Faire clic droit sur le nom du projet
- Choisir `Properties`
- Aller dans `Deployment Assembly`
- Cliquer sur `Add et sélectionner Java Build Path Entries`
- Cliquer sur `Next et choisir Maven Dependencies puis valider`
- Valider tout en cliquant sur `Apply and Close`

Bug d'Eclipse avec les projets JEE

Message d'erreur affiché

Several ports (8005, 8080, 8009) required by Tomcat Server at localhost are already in use

Solution

- Fermer Eclipse
- Avec le gestionnaire des tâches de Windows, chercher et finir la tâche de tous les processus `Java`
- Lancer `Cmder` puis aller dans le répertoire `bin` d'Apache
- Exécuter la commande `shutdown.sh`
- Redémarrer Eclipse