TP3 : Configuration cycle de vie, profiles

Objectif

Ce TP permet d'aborder les points suivants :

- Application de filtres sur le répertoire ressources
- Les profiles
- Assemblage pour la création d'une distribution

Gestion des ressources

Créer un répertoire *src/main/velocity* et déplacer le fichier *output.vm*
Ajouter ce nouveau répertoire ressources dans la configuration

Externaliser les informations du niveau de trace présent dans *log4j.properties* dans un autre fichier *src/main/dev.properties*
Définir ensuite le filtre nécessaire dans la configuration.

Configuration du compilateur

Configurer le plugin compiler afin que la compilation soit compatible avec java 5 et que les classes produites contiennent des informations pour le debugger.

Création d'un profile

Configurer un profile « production » qui surcharge la configuration par défaut.

En particulier, il appliquera un nouveau filtre (*src/main/prod.properties*) au fichier de ressources et modifiera la configuration du compilateur :

- Ne plus inclure les informations de debug dans les classes
- Demandeur au compilateur d'optimiser le code
- Afficher les avertissements lors de la compilation
- Être en mode verbeux

Création d'une distribution

Lors de l'activation du profile « *production* », nous voulons associer la création d'une distribution à la phase package.

Pour cela, configurer le plugin *Assembly*.

Exécuter *mvn clean package* et vérifier la génération de la distribution.

La distribution n'est cependant pas exécutable, nous devons fournir les informations nécessaires au

plugin pour générer un fichier *MANIFEST.MF* contenant la référence à la classe principale.

Ajouter les balises :

```
<archive>
  <manifest>
    <mainClass>org.maven.myapp.Main</mainClass>
  </manifest>
</archive>
```

Tester l'exécutable généré avec la commande

java -jar target/simple-weather-child-1.0-jar-with-dependencies.jar