Formation Apache Maven

TP 03 - Profils

Table of Contents

Etape 1 - Configuration fichier	1
Etape 2 - Filtering	2
Etape 3 - Profils	4

Etape 1 - Configuration fichier

• Créer un fichier application.properties dans la partie src/main/resources du projet :

- ① : Nouveau fichier à créer dans le répertoire *src/main/resources* (convention Maven).
 - Compléter

```
titre=Super App ! ①
```

- ① : représente le titre de l'application.
 - Modifier la classe *dev.App* comme suit :

```
package dev;
import java.util.ResourceBundle;
import com.github.lalyos.jfiglet.FigletFont;

/**
    * Hello world!
    *
    */
public class App {

    public static void main(String[] args) {
        String titre = ResourceBundle.getBundle("application").getString("titre"); ①
        String asciiArt = FigletFont.convertOneLine(titre);
        System.out.println(asciiArt);
    }
}
```

- 1 récupération du titre depuis le fichier de propriété.
 - Exécuter la classe *dev.App* et vérifier le résultat :



Etape 2 - Filtering

• Nous allons à présent ajouter un nouveau paramètre "mode" dans le fichier application.properties :

```
titre=Super App !
environnement=${mode}
```

• Activer le remplacement des variables dans le fichier *pom.xml*

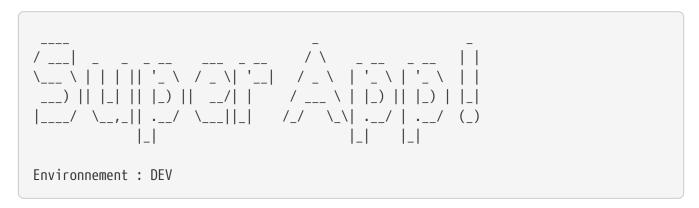
```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
   <modelVersion>4.0.0</modelVersion>
   cproperties>
      <junit.version>3.8.1</junit.version>
      <mode>DEV</mode> ①
   </properties>
   <build>
      <finalName>${project.artifactId}</finalName>
      <!-- Activation du remplacement des variables via Maven --> ②
      <resources>
         <resource>
             <directory>src/main/resources</directory>
             <filtering>true</filtering>
          </resource>
      </resources>
      <plugins>
      </plugins>
   </build>
</project>
```

- ① : Création d'une propriété *mode* avec la valeur *DEV*.
- ②: Configuration du filtering.
 - Compléter la classe dev.App :

```
public class App {
    public static void main(String[] args) {
        String titre = ResourceBundle.getBundle("application").getString("titre");
        String asciiArt = FigletFont.convertOneLine(titre);
        System.out.println(asciiArt);

    // affichage de l'environnement
        String environnement = ResourceBundle.getBundle("application").getString(
"environnement");
        System.out.println("Environnement : " + environnement);
}
```

• Exécuter l'application et vérifier que vous avez l'affichage suivant :



Etape 3 - Profils

- ullet Créer deux profils dev et prod qui remplissent différemment les informations du fichier application.properties :
 - Environnement *DEV* avec le profil dev
 - Environnement *PROD* pour la production

Pour cela modifier le fichier *pom.xml* :

```
"http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
   <modelVersion>4.0.0</modelVersion>
   cprofiles>
       cprofile>
          <id>dev</id>
          <activation>
              <activeByDefault>true</activeByDefault>
          </activation>
          cproperties>
              <mode>DEV</mode>
          </properties>
       </profile>
       cprofile>
          <id>prod</id>
          properties>
              <mode>PROD</mode>
          </properties>
       </profile>
   </profiles>
</project>
```

- Test du mode *dev* par défaut :
 - Assembler le projet via *mvn clean package*.
 - Exécuter l'application via java -jar target/premier-projet-maven.jar
 - Avez-vous le résultat attendu ?
- Test du mode prod :
 - Assembler le projet via mvn clean package -Pprod.
 - Exécuter l'application via java -jar target/premier-projet-maven.jar
 - Avez-vous le résultat attendu ?
- PAUSE GIT : Sauvegarder dans Github.