

TP5 : Projets multi-modules

Dans ce TP, nous allons mettre en place un projet Maven multi-modules qui produit deux applications :

- Un outil utilisable via une commande en ligne permettant de questionner la météo Yahoo
- Une application web qui propose une interface web pour faire la même chose.

Les deux applications stockent les résultats dans une base de données et utilisent la même logique applicative et de persistance.

Le projet multi-modules est donc composé de cinq modules :

- ***module-model*** : Un module modèle décrit les objets métiers des applications. Il utilise les annotations Hibernate pour automatiser la persistance. Le module est référencé par les deux applications
- ***module-service*** : Un module service contient toute la logique pour récupérer les données de Yahoo et de parser le XML fourni. Il a une dépendance sur le projet *model* et il est référencé par les deux applications.
- ***module-persist*** : Ce module contient l'interface d'accès aux données de persistances (*Data Access Objects – DAO*). Les deux applications utilisent ce module et ce module a des dépendances sur le modèle, *Hibernate* et le framework *Spring*.
- ***module-web*** : L'application Web contient de contrôleur MVC Spring qui utilise le module service et le module DAO. Il a donc des dépendances directes sur les modules *module-service* et *module-persist* et une dépendance transitive sur *module-model*.
- ***module-command*** : La commande en ligne comme le module web a des dépendances directes sur les modules *module-service* et *module-persist* et une dépendance transitive sur *module-model*.

Objectifs

- Mettre en place une architecture multi-modules
- Découvrir le plugin Hibernate3

Etapes

Projet multi-module

Création d'un projet multi-modules nommé « *multi-module* ». Les informations complémentaires habituelles peuvent être reprise du projet parent des Tps précédents.

Module modèle

Création du module modèle : « *module-model* » .

1. Référence au parent
 2. Ajout des dépendances vers Hibernate : *hibernate-annotations* et *hibernate-common-annotations*
 3. Exclusion de *javax.transaction/jta* car non utilisé
- Copier les sources fournis et Exécuter ***mvn install***

Module service

Création du module service : « *module-service* » . Reprendre le *pom.xml* du TP2 et ajouter une

référence au projet parent ainsi qu'une dépendance vers le *module-model*.
Copier les sources fournis et Exécuter ***mvn install***

Module persist

Création du module de persistance « *module-persist* » .

1. Référence au parent
2. Ajout des dépendances vers Hibernate : *hibernate.3.2.5.ga*
3. Exclusion de *javax.transaction/jta* car non utilisé et pas disponible dans le repository central
4. Ajout de dépendance vers Spring : *org.springframework/spring_2.0.7*

Module web

Création du module web : « *module-web* » .

1. Référence au parent
2. Ajout des dépendances vers les modules *service* et *persist*
3. Ajout de dépendances vers *Spring* et *velocity_1.5*
4. Ensuite nous définissons deux plugins (Jetty et Hibernate3) comme suit :

```
<build>
  <finalName>simple-webapp</finalName>
  <plugins>
    <plugin>
      <groupId>org.mortbay.jetty</groupId>
      <artifactId>maven-jetty-plugin</artifactId>
      <dependencies>
        <dependency>
          <groupId>hsqldb</groupId>
          <artifactId>hsqldb</artifactId>
          <version>1.8.0.7</version>
        </dependency>
      </dependencies>
    </plugin>
    <plugin>
      <groupId>org.codehaus.mojo</groupId>
      <artifactId>hibernate3-maven-plugin</artifactId>
      <version>2.0</version>
      <configuration>
        <components>
          <component>
            <name>hbm2ddl</name>
            <implementation>annotationconfiguration</implementation>
          </component>
        </components>
      </configuration>
      <dependencies>
        <dependency>
          <groupId>hsqldb</groupId>
          <artifactId>hsqldb</artifactId>
          <version>1.8.0.7</version>
        </dependency>
      </dependencies>
    </plugin>
  </plugins>
</build>
```

Copier les sources fournis et exécuter ***mvn install***

2. Génération de la base de données. Pour cela, nous utilisons le plugin Hibernate3. Dans le répertoire du projet webapp, exécuter : ***mvn hibernate3:hbm2ddl***
3. Vérifier la création de la base de données dans le répertoire *\${basedir}/data*

4. Exécution de Jetty et test de l'installation à l'URL
http://localhost:8080/module-web/weather.x?zip=60202

Module commande

Création du module de commande en ligne : « *module-command* » .

1. Référence au parent
2. Ajout de dépendances *Spring2.0.7*, *Velocity1.5*, *hsqldb_1.8.0.7* et les modules *service* et *persist*
3. Création de la distribution et exécution via la commande :
java -cp target/module-command-jar-with-dependencies.jar org.formation.weather.Main
4. Exécuter dorénavant *mvn install* au niveau du projet *multi-module*
5. Refaire éventuellement un refactoring des POM pour gérer les numéros de versions dans les POM parents
6. Exécuter *mvn dependency:analyze* et éliminer si besoin les références directes non déclarés