

Travaux pratiques : Java Persistence Api

1. Introduction

Java Standard Edition intègre à présent une API de persistance : Java Persistence Api (JPA). JPA permet de réaliser un Mapping Objet Relationnel en assurant la synchronisation d'un graphe d'objets avec des tables dans une base de données. Le but étant de masquer le plus possible la base de données sous-jacente. Il est par exemple possible de générer automatiquement les tables de la base à partir du code Java. De plus, le code, bien qu'à la norme Java Standard Edition, est compatible avec la norme Enterprise Java Bean (EJB 3.0).

Technologies utilisées :

- Eclipse
- Hibernate 3
- Spring 3

2. Un premier exemple

Commençons par étudier un exemple d'utilisation de JPA :

```
EntityManagerFactory emf = ...
EntityManager em = emf.createEntityManager(); // 4

EntityTransaction tx = em.getTransaction(); // 3
tx.begin();

Voiture voiture = new Voiture(); // 1

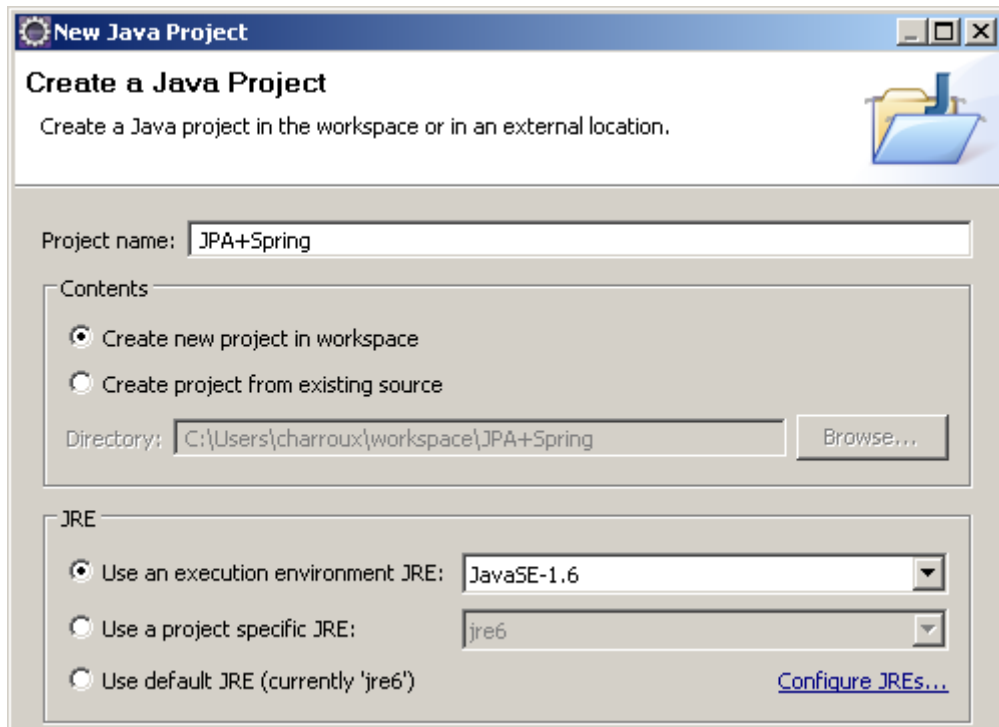
em.persist(voiture); // 2

tx.commit();
```

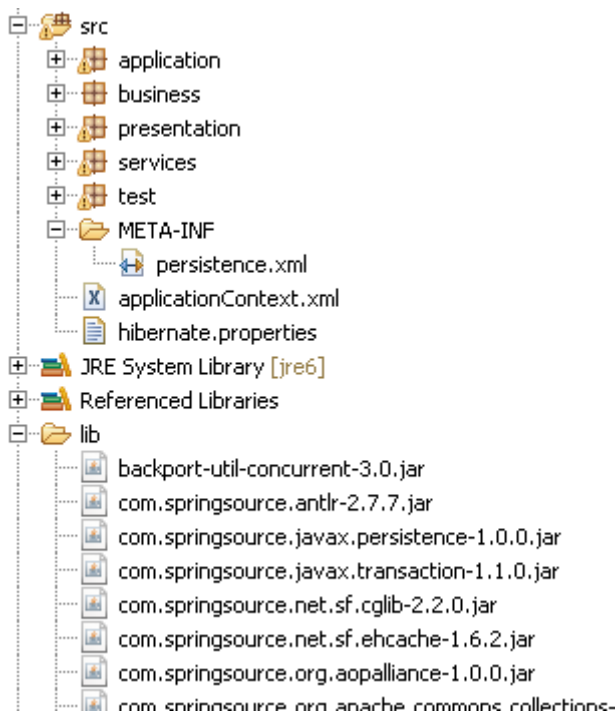
La partie essentielle sont les lignes 1 et 2 où on crée un objet de type `Voiture`, mais ce pourrait être n'importe quel objet Java, puis on demande sa sauvegarde dans la base de données via la méthode `em.persist(voiture)`. Le reste du code, notamment la ligne 3, montre qu'il est possible de réaliser des transactions. L'`EntityManager` (ligne 4) est le point d'entrée principal de JPA car c'est via un `EntityManager` qu'on accède aux méthodes de sauvegarde, de recherche, de suppression... dans la base de données.

3. Utilisation d' Eclipse

Après avoir lancé Eclipse, il faut commencer par créer un projet Java (File -> New -> Java Project) :



Voici un exemple de la structure d'un projet :



On y voit :

- tous les packages correspondant aux répertoires contenant le code de l'application (application, business, presentation...).
- le répertoire META-INF qui contient le fichier persistence.xml de déclaration des classes persistantes.
- le fichier de déclaration de Spring `applicationContext.xml` permettant de créer un EntityManager.

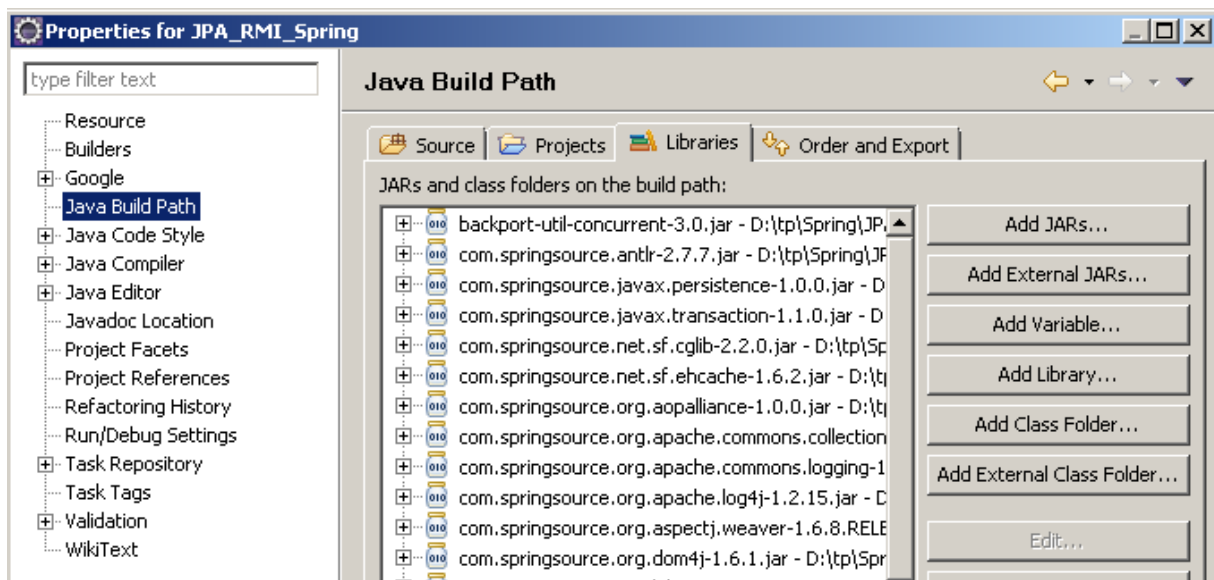
- Le fichier hibernate.properties qui permet de configurer finement l'accès à la base de données
- les libraries .jar (contenant entre autres la base de données).

A l'adresse :

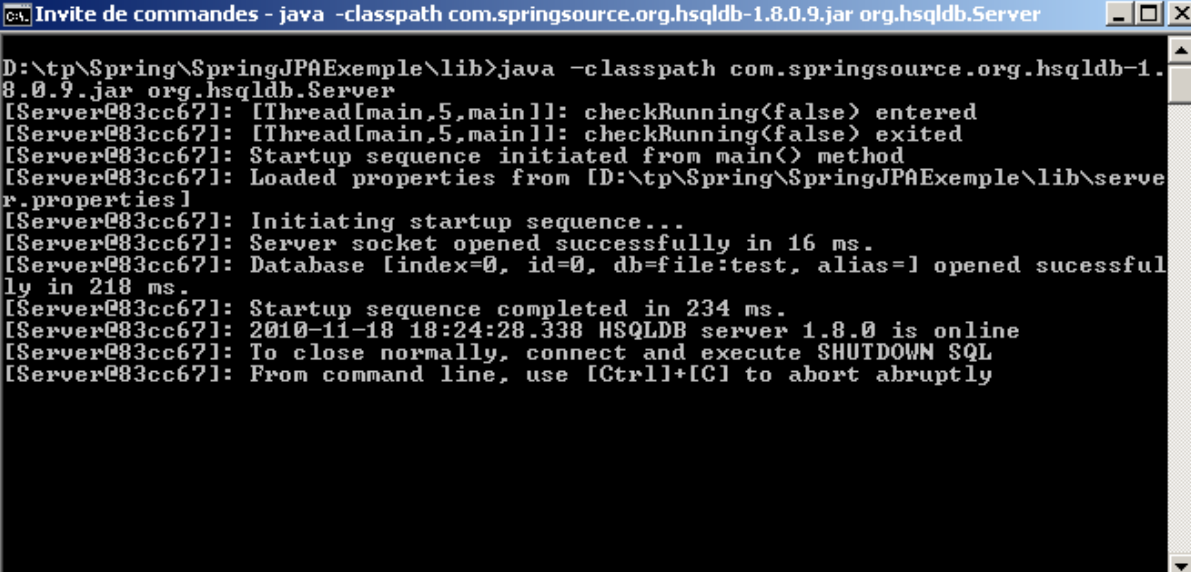
<http://perso.efrei.fr/~charroux/JPA+Spring/>

Vous disposer d'un exemple des fichiers applicationContext.xml, hibernate.properties et persistence.xml, d'une ébauche d'un programme de test, ainsi que toutes les librairies utiles au projet. Il faut décompresser le fichier téléchargé dans le répertoire du projet Eclipse, et faire F5 sur le projet sous Eclipse afin de voir les modifications apparaître.

Enfin, pour indiquer à Eclipse d'inclure les librairies dans le projet, il faut faire un clic droit sur le projet, aller dans propriétés -> Java build path et ajouter toutes les librairies téléchargées en tant que External JARs :



La base de données est incluse dans les librairies du projet. Pour démarrer le serveur de base de données il faut saisir la commande suivante :



```
Invite de commandes - java -classpath com.springsource.org.hsqldb-1.8.0.9.jar org.hsqldb.Server
D:\tp\Spring\SpringJPAExemple\lib>java -classpath com.springsource.org.hsqldb-1.8.0.9.jar org.hsqldb.Server
[Server@83cc67]: [Thread[main,5,main]]: checkRunning(false) entered
[Server@83cc67]: [Thread[main,5,main]]: checkRunning(false) exited
[Server@83cc67]: Startup sequence initiated from main() method
[Server@83cc67]: Loaded properties from [D:\tp\Spring\SpringJPAExemple\lib\server.properties]
[Server@83cc67]: Initiating startup sequence...
[Server@83cc67]: Server socket opened successfully in 16 ms.
[Server@83cc67]: Database [index=0, id=0, db=file:test, alias=] opened successfully in 218 ms.
[Server@83cc67]: Startup sequence completed in 234 ms.
[Server@83cc67]: 2010-11-18 18:24:28.338 HSQLDB server 1.8.0 is online
[Server@83cc67]: To close normally, connect and execute SHUTDOWN SQL
[Server@83cc67]: From command line, use [Ctrl]+[C] to abort abruptly
```

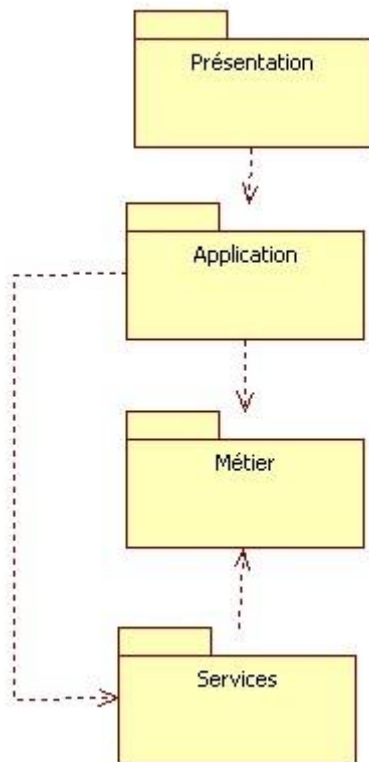
4. Premier essai

Il n'est pas évident de faire fonctionner une application du premier coup car, en plus des problèmes de programmation, peuvent venir se greffer des problèmes liés à la base de données ou à sa configuration.

Il est donc conseillé de commencer à développer une classe persistante toute simple, puis d'écrire un programme de test.

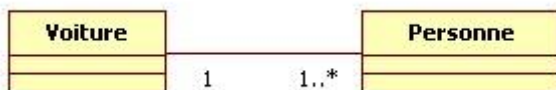
5. L'application à développer

L'application à programmer doit permettre à des particuliers de louer des voitures. Cette application doit être composée de couches afin de faciliter la réutilisabilité du code comme le montre le diagramme suivant.



6. La couche métier (les classes persistantes)

La couche Métier est composée au moins de 2 classes, Voiture et Personne associées entre elles par une association de 1 vers plusieurs.



Ce sont ces 2 classes qui vont être mappées en base de données avec 2 tables. Programmez ces classes en respectant la norme JPA.

Vous pouvez ajouter d'autres classes en utilisant l'héritage par exemple.

La couche présentation peut être composée d'un simple mais ou disposer d'une interface utilisateur basique comme l'exemple ci-dessous dans lequel on saisit le nom d'une personne désirant louer une voiture définit par son numéro d'immatriculation :

Nom personne :

Numero immatriculation :

OK

7. La couche Application

L'interface graphique accède à la couche Application via une interface :

```
package application;
```

```
public interface Location {
```

```
    public void louer(String nomPersonne, String immatriculation) throws  
    VoitureException;
```

```
    public void ajouterVoiture(String immatriculation) throws  
    VoitureException;
```

```
    /**  
     *  
     * @param nomPersonne  
     * @return Le numero d'immatriculation de la voiture louee par  
     nomPersonne ou null si la personne n'a pas louee de voiture  
     * @throws PersonneException si il n'y a pas de personne ayant le nom  
     nomPersonne  
     */  
    public String rechercheVoiture(String nomPersonne) throws  
    PersonneException;  
}
```

Votre travail consiste à coder une implantation de cette interface en vous appuyant sur une couche métier présentée précédemment.