

TP 3 - Accès JPA à une base de données via un patron DAO

Léa Brunschwig <lea.brunschwig@univ-pau.fr>

Dans ce TP, nous allons reprendre le code du premier TP et le compléter pour gérer un autre moyen de persistance des objets via JPA au lieu de JDBC.

Astuce

Sous Netbeans, il est possible d'obtenir automatiquement le code Java de POJO avec les annotations JPA à partir de la structure d'une base. Pour cela, il faut :

1. Créer, si elle n'existe pas, une connexion à la base : onglet *Service, Databases* -> *New Connection*
2. Dans le code source, via le menu contextuel : *New* -> *Entity Classes From Database*. Sélectionner la connexion à la base puis les tables.

⚠ Attention ⚠

Les POJO *Sport* et *Discipline* existent déjà dans le code du premier TP, dans le package `donnees`. Il faut que le code que vous allez rajouter pour l'implémentation JPA manipule ces POJO existant pour une compatibilité avec l'implémentation JDBC.

Travail à réaliser

Étape 1 : vérification du fonctionnement de JPA pour la base de données

1. Générez les POJO sous Netbeans via la méthode précisée ci-dessus. Afin de gérer le problème de l'existence des POJO à ne pas écraser, choisissez de les mettre dans un package de nom différent (`data` par exemple). Notez bien le nom de l'unité de persistance qui sera indispensable pour gérer les POJO. Celle-ci se retrouve et peut se modifier dans le fichier `persistence.xml` du répertoire `META-INF`.
2. Créez un petit programme Java qui réalise quelques manipulations (recherche, création, ...) d'objets sérialisés en BDD en adaptant par exemple le code suivant avec le bon nom d'unité de persistance :

```
import data.Sport;
import data.Discipline;

import javax.persistence.*;

public class Test {

    public static void main(String argv[]) {
        EntityManagerFactory emf =
Persistence.createEntityManagerFactory("SportsPU");
        EntityManager em = emf.createEntityManager();
        int cle = 1;
        Sport sport = em.find(Sport.class, new BigDecimal(cle));
        System.out.println("Le sport " + cle + " est " +
sport.getIntitule());
        for (Discipline disc : sport.getDisciplines()) {
            System.out.println(" -> " + disc.getIntitule());
        }
    }
}
```

Étape 2 : ajout du support JPA dans l'architecture en DAO et fabriques du TP2

1. Une fois que vous avez constaté le bon fonctionnement de la manipulation des POJO, reportez les annotations JPA des POJO générés vers les POJOS originaux du répertoire `donnees`. Il y a quelques différences comme la clé qui est un `int` au lieu d'un `BigDecimal` empêchant les POJO générés de remplacer directement les POJO originaux.
2. Modifiez le fichier de persistance pour que les classes des POJO soient maintenant celles du package `donnees`.
3. Modifiez votre programme de test pour utiliser les POJO du package `donnees`. Vérifiez que cela fonctionne comme avant.
4. Implémentez un DAO pour les POJO `Sport` et `Discipline` : deux classes `DAO_JPA_Sport` et `DAO_JPA_Discipline` héritant de la classe abstraite `DAO`.
5. Créez une fabrique pour instancier les DAO JPA : une classe `Sports_JPA_DAO_Factory` héritant de la classe abstraite `SportsDAOFactory`.
6. Modifiez la fabrique de fabriques `AbstractDAOFactory` pour gérer le cas de JPA.