

JEE – TP1 : l'accès aux données et Maven

L'ensemble des ressources nécessaires au TP est disponible dans dépôt GitHub suivant :

<https://github.com/jdassonvil/epsi>

Le corrigé est également disponible :

<https://github.com/jdassonvil/epsi-j2ee-tp1>

1.1 Création de la base de données

- Exécutez le script de création de base de données sur votre base MySQL

1.2 Prise en main de JDBC

- Lancez Eclipse et créez un nouveau projet JAVA
- Importez le Driver JDBC dans votre projet
- Ecrivez un programme qui se connecte à votre la base de données « musicstore » et affiche dans la console la liste des chansons (contenu de la table track)
- Modifiez ce programme pour n'afficher que la liste des chansons de l'artiste « Franck Sinatra ».
- Modifiez ce programme pour afficher la liste des chansons écrite par des artistes originaires de New York

1.3 Les POJO

- Dans un nouveau package model écrivez les classes correspondant aux POJO track, album et artist
- Utilisez le code précédent pour instancier les POJO correspondant aux éléments présents en base de données.
- Utilisez la méthode toString() pour afficher le contenu d'un POJO

1.4 Les DAO

- Dans un nouveau package dao écrivez les classes correspondant aux DAO des classes Track, Album et Artist. Pour chaque classe implémentez les méthodes CRUD (create, retrieve, update delete).
- Testez vos DAO en remplaçant le code écrit à l'exercice précédent par un appel aux DAO.
- Pourrai-t-on écrire un code plus générique ?

1.5 JPA/Hibernate

- Importez les jars présents dans le dossier hibernate du repository dans votre projet
- Créez un fichier persistence.xml dans le répertoire src/main/resource/META-INF et configurez-le. Assurez-vous de la bonne présence du fichier dans le buildpath.
- Utilisez les annotations pour décrire le mapping entre vos POJO et la base de données. Utilisez notamment l'annotation @JoinColumn pour lier les entités entre elles.
- Mettez à jour les DAO pour utiliser un entity manager
- Testez la différence de comportement entre LAZY et EAGER

1.6 Maven

- Ajoutez un fichier pom.xml à votre projet
 - Définissez dans le pom le groupId, l'artifactId ainsi que la version de votre projet
- Sur votre projet appliquez l'action « convert to maven project »
- Dans Eclipse, affichez la vue « Effective POM », que remarquez-vous ?
- Configurez le pom avec les dépendances précédemment importées à la main (Driver MySQL et Hibernate)