



École Polytechnique de Montréal
Département Génie Informatique et Génie Logiciel
INF3710 – Fichiers et Bases de données

Travail Pratique N° 5

Interface SQL et persistance
(Relationnel et non relationnel)

I. Informations générales

Session	Fall 2017
Public cible	Étudiants de 1er cycle
Date de remise	Groupe2: 13/11/2017 Groupe1: 20/11/2017
Équipe de	2 étudiants
Pondération	6 %
Directives particulières	1. <i>Tout retard</i> dans la remise du compte-rendu <i>entraîne automatique-ment</i> une pénalité aux étudiants concernés. 2. Aucun compte-rendu ne sera corrigé, s'il est soumis par une équipe dont la taille est différente de deux (2) étudiants sans l'approbation préalable du chargé de laboratoire. La note de zéro sur vingt (0/20) sera attribuée aux étudiants concernés. 3. Soumission du compte rendu (au format PDF ou Word) par <i>Moodle</i> uniquement (https://moodle.polymtl.ca). 4. Aucune soumission "hors <i>Moodle</i> " ne sera corrigée. La note de zéro sur vingt (0/20) sera attribuée aux étudiants concernés.

II. Objectifs du laboratoire

Le but de ce TP est de permettre à l'étudiant(e) de se familiariser avec JPA et les patrons de persistance ***Repository***.

III. Travaux demandés

Question 1

En vous basant sur le projet Java fourni (le dossier est présent dans le matériel du TP sous Moodle), suivez les étapes suivantes pour pourvoir l'importer dans votre environnement Eclipse :

- 1- En utilisant le command prompt (cmd), accéder au dossier déjà copié et lancer la commande Gradlew.bat
 - 2- Une fois le build est réussi, vous lancer la commande Gradlew.bat eclipse
 - 3- Ouvrir Eclipse et importer le projet
- On vous demande de lire attentivement le code fourni et de comprendre l'intérêt de l'utilisation du JPA. Analysez spécialement les lignes suivantes: □

Ligne 11 et 13 de la classe CustomerRepository.

Ligne 10, 12 et 13 de la classe ProductRepository.

Ligne 8, 11 et 12 de la classe Customer.

Ligne 25, 28, 67 et 68 de la classe Application.

- Formulez une petite analyse de quelques lignes sur l'objectif de ces lignes de code, Quels sont les Repository utilisés et comment fonctionnent-ils ? Quels sont les différentes méthodes de persistances disponibles ?

- Démarrer le serveur Spring (clique sur la class 'Application' et fait 'Run/Java application').
Après que le server démarre, aller sur le browser et taper '<http://localhost:8080/product>' et '<http://localhost:8080/customer>', observer les résultats.
D'où viennent les données (Quelle base de donnée ? Relationnel ou non relationnel et pourquoi ?) PS : Utilisez google chrome sur les machines du lab svp.

Question 2

Une fois que vous avez compris le code, on vous demande d'apporter des modifications pour bien comprendre le lien et l'enchaînement entre les différentes classes Java.

Dans ce projet, on utilise le Framework Spring, qui vous permettra de manipuler les différents Repository par exemple (MongoRepository, JpaRepository etc.)

- Apporter la modification suivante sur le patron Repository de la classe ProductRepository

```
//public interface ProductRepository extends JpaRepository<Product, Long> {  
public interface ProductRepository extends MongoRepository<Product, Long> {
```

- Editer le fichier application.properties en ajouter la valeur de host (c'est l'adresse IP de votre VM, cela pour assurer la connexion en remote) et le port pour se connecter :

Exemple :

```
spring.data.mongodb.host=192.168.126.129  
spring.data.mongodb.port=27017
```

- Lancer le serveur *Mongod* (présent sur votre VM), arrêter le serveur spring depuis la console sur eclipse et réexécuter le projet Java de nouveau. Une fois c'est fait, lancer le client mongo *Mongo* (sur la VM).

Que remarquez-vous sur le Shell mongo et sur le localhost (<http://localhost:8080/product>) ? Quels sont les bases de données utilisées ? Relationnel ou non relationnel ?

Quel est l'impact de la modification du Repository ? Formuler une petite analyse sur le résultat obtenu et ajouter les captures d'écrans nécessaires dans votre rapport.

Question 3

- En utilisant le client mongo, formulez et exécuter la requête suivante : Insérer un nouveau document dans la collection Product tout en remplissant tous les champs existants et en ajoutant le nouveau champ 'Price' rempli avec une valeur.

- Afin de voir le lien entre le service REST et la base de données, vérifier votre insertion sur le browser. Que remarquez-vous ? Expliquez.

Utiliser la méthode `findByLabel` pour afficher sur le browser le document ajouté dans product et utiliser `findByLastName` pour afficher le customer 'Bauer'. Les URL sont les suivants :

<http://localhost:8080/product/search/findByLabel?label=XXX>

<http://localhost:8080/customer/search/findByLastName?name=XXX>

- Sur Eclipse, ajouter l'attribut 'Price' avec ses getters et setters, ré-exécuter le projet et vérifier la sortie sur le localhost. Que remarquez-vous ? Expliquez le résultat.

NB : Fournissez dans votre rapport des captures d'écrans des requêtes exécutées sur le mongo Shell, sur le localhost et les résultats obtenus.

VI. Livrable

Le livrable à soumettre est une archive .zip ou .rar dont le nom est formé des numéros de matricules des membres de l'équipe, séparé par un trait de soulignement (_). Il doit comporter les éléments suivants:

- a) Le projet java.
- b) Un rapport contenant les captures d'écrans demandées et toute justification nécessaire.

Vous pouvez également inscrire dans le rapport, toute autre information que vous désirez porter à la connaissance de la personne en charge de la correction.

V. Évaluation

Rubrique	Points
Questions	19
Presentation du rapport	1
Total	20

