# Evaluation Spring + Hibernate

A partir de l'application SpringMVC, créer une application de gestion en utilisant Spring MVC, Hibernate et une Bdd Hsqldb (ou autre SGBD si vous le souhaitez).

Le but est de mettre en place un CRUD classique.

**L'application web contient :**

* Une page listant les livres
* Une page pour ajouter un livre
* Une page pour modifier un livre. Lors de l'ouverture de cette page de modification, les champs doivent être pré chargés.
* Sur la page contenant la liste, avoir un bouton pour supprimer un livre.

**Les livrables à fournir sont :**

* Un zip de l'application
* Le script SQL de création de(s) table(s) (le mettre dans le répertoire src/main/resources)

**Critères de notation :**

* Le projet doit compiler
* Le serveur doit démarrer sans erreur
* On doit retrouver les 4 fonctionnalités principales
* Le code doit être propre et organisé (on respecte les normes java, pas .NET par exemple **voir aide**)

**Bonus**

* Si utilisation de constantes, de l'internationalisation, de commentaires ou si vous me surprenez positivement :)
* Si utilisation de CSS pour embellir l'application (rapide avec bootstrap)
  + c'est du bonus, l'important, c'est que la partie Spring/Hibernate fonctionne bien

**Adresse mail pour l’envoi :** [andre.aubin@viacesi.fr](mailto:andre.aubin@viacesi.fr)

Merci de m’indiquer dans le mail si c’est un projet Eclipse ou IntelliJ 😊

# Tableau de correspondance de ce qui doit être utilisé par étudiant

|  |  |
| --- | --- |
| **Type** | **Personne** |
| Bijoux | Quentin BERNICHON |
| Film | Florentin BOUCHEZ |
| Fruit | Rémy DETRE |
| Jeux | Paul KONCZAK |
| Livre | Nicolas MEINDL |
| Meuble | Edouard OLIVIER |
| Ordinateur | Edouard PENIN |
| Sac | Maxime PLAQUIN |
| Sandwich | Baptiste ROBACHE |
| Telephone | Cyril RUST |

Donc, si par exemple tu dois gérer les Bijoux, je veux voir :

* + BijouxController
  + BijouxService
  + BijouxDao
  + Une table BIJOUX
  + …

Contenu table :

Contenu de la table en BDD -> comme vous voulez, un nom, une description, un prix, une catégorie etc … Il faut que ce soit cohérent avec votre type d’objet

**Exemples de ce que vous pouvez utiliser** :

Bijoux -> nom, type, prix  
Film -> nom, genre, auteur, prix  
Fruit -> nom, famille, prix  
Jeux -> nom, editeur, prix  
Livre -> titre, auteur, prix  
Meuble -> nom, materiau, prix  
Ordinateur -> marque, modèle, prix  
Sac -> marque, taille, prix  
Sandwich -> nom, prix, estChaud  
Telephone -> marque, prix, tailleEcran, prix

# Aide

1 - Utiliser les mêmes versions de Spring et Hibernate, du coup, il ne devrait pas y avoir de changements à faire dans le pom.xml si vous avez suffisamment avancé lors du tuto.

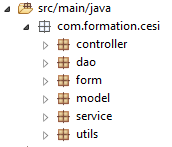
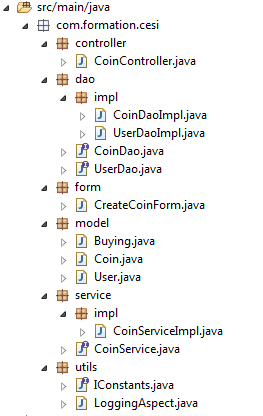
2 - Normes Java

* + Une classe Java commence toujours par une majuscule
  + Une variable commence toujours par une minuscule
  + Utilisation du camelCase
  + Le nom d’un package doit être tout en minuscule

3 - Il faut utiliser les annotations suivantes quand c’est nécessaire :

* + @Autowired pour injecter une dépendance
  + @Controller pour un Controller
  + @Repository pour un Dao
  + @Service pour un service

4 - Respecter (c’est toujours mieux) l’organisation des packages, par exemple de la manière suivante :

**C’est un exemple, l’important est que cela soit organisé**