

**UE ENSICU60**  
**Architectures réparties - JEE**  
**Examen**

**Aix-Marseille Université**  
**Master Informatique**  
**Spécialité SIR**  
**Gaël Guibon, Omar Boucelma**  
2017-2018

# Chapitre 1

## Examen : Ajouter des fonctionnalités à une plateforme de produits

### 1.1 Résumé

Lors de cet examen vous devez ajouter des fonctionnalités à une application J2EE spring-boot côté serveur. La vue (HTML / JS) est déjà mise en place, il vous incombe donc de compléter les fonctionnalités des modèles, des contrôleurs, et de l'ORM. Il n'y a normalement pas besoin de modifier la vue pour cet examen.

Les fichiers \*.js vous permettent normalement de comprendre sans explications ce que la VUE attend du serveur, mais une partie des exercices de cet examen vous le décrivent.

Pour résumer il vous faut dans un premier temps :

- Récupérer le squelette de l'application à l'adresse suivante : <https://github.com/gguibon/CoursesJavaEE>, dossier examen

```
git clone URL
```

- Bien regarder les fichiers HTML et JS pour comprendre les attentes de la vue.
- Pouvoir interroger votre application sans interface :

```
curl http://localhost:8090  
# ou directement dans le navigateur
```

- Le package util vous fournit plusieurs classes et méthodes pratiques pour vous faire gagner du temps.
- A la fin de l'examen vous devez fournir : votre dossier maven (src + pom.xml) avec, si possible, votre application packagée en .jar exécutable ("mvn package")

**Après chaque fonctionnalité réussie je vous invite à conserver une version de votre projet !**

### 1.2 Gestion des utilisateurs

Le client souhaite contrôler l'accès à l'application.

- Un utilisateur doit être connecté pour pouvoir effectuer des actions (ajouter un produit/commentaire, etc) à l'application (dont le mapping principal est "/"). Un utilisateur non connecté peut visiter l'application.
- Il doit exister un compte unique d'administrateur permettant certaines actions supplémentaires : suppression d'un produit, suppression de n'importe quel commentaire

### 1.3 Gestion des produits

Le client veut pouvoir voir et gérer les produits à partir de `http://localhost:port`, c'est-à-dire :

1. Ajouter un produit
2. Permettre uniquement à l'administrateur de supprimer les produits
3. Voir le nombre de commentaires par produit
4. Voir le rang moyen attribué à un produit (utiliser la méthode disponible dans le squelette)
5. Filtrer les produits par categorie grâce aux boutons situés sur la gauche.

## 1.4 Gestion des commentaires

Chaque produit peut posséder des commentaires qui lui sont liés. Quand un produit est supprimé par l'administrateur, ses commentaires le sont aussi. Un utilisateur peut :

1. Ajouter un commentaire
2. Modifier uniquement un de ses commentaires
3. Supprimer uniquement un de ses commentaires

## 1.5 Architecture de l'application

Sur papier ou fichier .txt, fournir une brève description de l'organisation de l'application en décrivant brièvement :

1. La communication serveur-client de l'application
2. Les interactions ORM entre produits et commentaires
3. Vos choix d'implémentation le cas échéant

## 1.6 Fonctions supplémentaires

### 1.6.1 Importer un produit

Il doit être possible d'importer un produit. Importer un des fichiers json fournis doit permettre d'automatiquement intégrer son contenu en tant que nouveau produit avec commentaires dans la base de données.

L'action s'effectue par le bouton "upload" disponible dans la vue.

### 1.6.2 Fermeture de l'application

Le bouton "quit" doit permettre de fermer l'application. Utile pour une utilisation embarquée dans un jar exécutable.