

Blu-Ray STORE

Introduction

Le gérant d'un magasin de vente de Blu-Ray souhaite développer une application permettant de gérer (à distance) son stock de Blu-Ray.

Pour développer son application, nous allons commencer par créer notre première entité métier, notre premier service métier et notre premier repository.

Expression générale du besoin

Cette application disposera d'une console Web d'administration qui permettra à un administrateur de gérer :

- le stock de films
- les informations concernant chaque film proposés

Un film étant représenté par un JavaBean ayant pour propriétés :

- Un titre
- Un nombre d'exemplaires
- Un genre
- Un acteur principal du film
- (Éventuellement les autres acteurs du film)

Configuration du projet

Avant de réaliser notre application, nous allons expérimenter l'architecture cible autour de l'entité métier « Movie ».

Vous allez créer un projet Java / Maven à l'aide de IntelliJ Community et Java 17. Choisissez l'archetype quickstart comme dans vu en cours.

Le triplet Maven identifiant le projet sera le suivant :

- artifactId : core
- groupId : com.mycompany.bluraystore
- version : 1.0-SNAPSHOT

Assurez-vous que le pom.xml Maven référence bien la version 17 de Java et que votre projet sous IntelliJ soit également configuré pour utiliser Java 17.

Première entité métier

Avant de réaliser concrètement notre application, nous allons expérimenter l'architecture cible autour de l'entité métier « Movie ».

Créez tout d'abord un sous-package entity où se trouveront les classes entité.

L'entité métier « Movie » sera matérialisée par une classe java com.mycompany.bluraystore.entity.Movie au format JavaBean ayant pour l'instant des propriétés simples :

- title (String)
- genre (String)

Blu-Ray STORE

Premier service métier

Nous allons ensuite créer dans ce même projet un Service Métier autour du concept de film. Créer un nouveau sous-package nommé service et créez la classe `com.mycompany.bluraystore.service.MovieService`.

Ajoutez à cette classe une méthode permettant l'enregistrement d'un nouveau film dans le magasin :

- `public void registerMovie(Movie movie)` (Laissez la méthode vide pour l'instant)

Premier repository

Ajoutez maintenant un sous-package repository dans lequel nous allons créer notre premier repository autour du concept de film via la classe `com.mycompany.bluray.repository.MovieRepository`.

Ajoutez à la classe une méthode `public void add (Movie movie)` qui consistera à ajouter ce film dans une liste statique de films. (En s'inspirant de ce que nous avons fait ensemble durant le cours)

Vous pouvez maintenant ajouter dans le service métier `MovieService` une propriété de type `MovieRepository` et compléter sa méthode `registerMovie` en faisant appel à la méthode `add` de `MovieRepository`.

N'oubliez pas d'instancier `MovieRepository`.

Exécution

Pour tester l'ensemble nous allons modifier la classe `App` qui devrait déjà être présente dans votre projet. Elle doit disposer d'une méthode `main`.

Dans cette méthode `main` on effectuera les opérations suivantes :

1. Demander à l'utilisateur de saisir le titre et le genre du film à ajouter
2. Instancier un film en lui affectant le titre et le genre
3. Instancier `MovieService`
4. Exécuter la méthode `registerMovie`