

Projet ENSTA

Des questions pourront être ajoutées ou enlevées par la suite suivant l'avancement global du groupe.

Pour lancer le projet utiliser la commande : `mvn spring-boot:run`

Le projet a été créé grâce à Spring-Boot : <https://start.spring.io/>

On a choisi ceci pour pouvoir voir facilement les résultats de vos efforts avant de travailler sur le Front (qui sera en Angular).

On utilise une base en H2, elle a l'avantage d'être intégrée, donc vous n'avez rien à installer dans votre projet. Vous pouvez modifier l'initialisation de la base dans `src/main/resources/data.sql`

Une fois votre projet lancé (avec `mvn spring-boot:run`) vous pouvez voir l'évolution de votre base de données sur <http://localhost:8080/h2-console/> avec comme JDBC url : `jdbc:h2:mem:film`, comme utilisateur `sa` et aucun mot de passe (ce n'est évidemment pas sécurisé, on peut le modifier avec le fichier `application.properties`, mais pour notre projet cela sera suffisant)

La configuration de cette base de données se fait avec le fichier `src/main/resources/application.properties`

On a aussi mis en place l'outil Swagger qui vous permet de tester vos webservices sans avoir fait le front. Vous pouvez y accéder sur localhost:8080/swagger-ui.html

N'oubliez pas de faire la javadoc (pas nécessaire sur les getter setter)

Question 0 :

Comprendre le code qu'on vous a donné, faire la javadoc.

Question 1 :

Faire le delete, getAll, create, getById, update des films.

Question 2 :

Ajouter la colonne Réalisateur dans la base de données qui contient un Nom, un Prénom et une date de naissance. Un film doit donc avoir une nouvelle colonne réalisateur. Initialisez la table Réalisateur avec deux – trois exemples.

Commencez ensuite par modifier les fonctions de la question 1 pour prendre en compte cette nouvelle colonne.

Ensuite vous pourrez créer les fonctions pour faire le delete, getAll, create, getById, update des réalisateurs.

Question 3 :

On va faire une pagination pour les clients et les réalisateurs. C'est à dire qu'on va créer un objet `Page<T>` qui contient les attribues

```
private int number;
```

```
private int size;
```

```
private long total;
```

```
private List<T> data;
```

Les webservices renveront donc cet objet plutôt qu'une liste.

Question 4 :

On va faire la partie Front qui correspond à notre Back.

Question 5 :

On va mettre en place un filtre sur le nom du film, et sur le nom d'un réalisateur.

Question 6 :

On va mettre en place un order sur l'Id (par default), sur le nom d'un film, sur sa durée.

Question Bonus 1 :

Rajoutez une colonne genre dans la table des films. Créez une table Genre qui possède un nom. Initialisez avec des genres comme : action, thriller, policier, horreur, comédie, SF, drame, biopic. Les genres étant supposé être fixe (si modification cela sera fait par un admin), on n'a pas besoin de faire de create et de delete.

Faire les fonction getAll, getById pour les genres. Mettre à jours les fonctions des films pour faire apparaître le genre.

Question Bonus 2 :

On va mettre en place Sonar, un outil permettant de vérifier le clean de votre code. Il vous chiffre notamment votre dette technique.