**Compte Rendu Projet JEE et Middleware**

**Web et Mobile : JEE - Spring-Angular**

Objective :

 Concevoir et Créer une application web et mobile qui permet de gérer des cinémas sur différentes villes.

  Les exigences fonctionnelles de l’application sont :

-          Gestion des Cinémas (Consultations, Saisie, Ajout, Edition, Mise à jour et suppression) .

-          Gestion des Salles et des Places.

-          Gestion Films.

-          Gestion Projection.

-          Gestion des ventes des Tickets.

   L’application se compose de 2 Parties :

 La partie backend est basée sur Spring Boot et se compose des couches DAO, Service (Métier) et Web.

-          La couche DAO est basée sur Spring Data, JPA, Hibernate

-          La couche Métier est définie par une interface et une implémentation quelques spécifications fonctionnelles qui nécessite des calculs ou des traitements particuliers

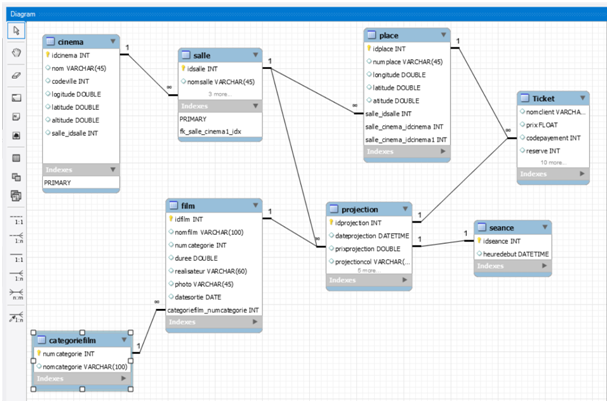
-          La couche Web est basée sur des API Restful basée sur Spring Data Rest ou des RestController

 La partie Frontend est basée sur le Framework Angular.

 La sécurité est basée sur Spring Security et Json Web Token

1 - Conception  de diagramme représentant les données

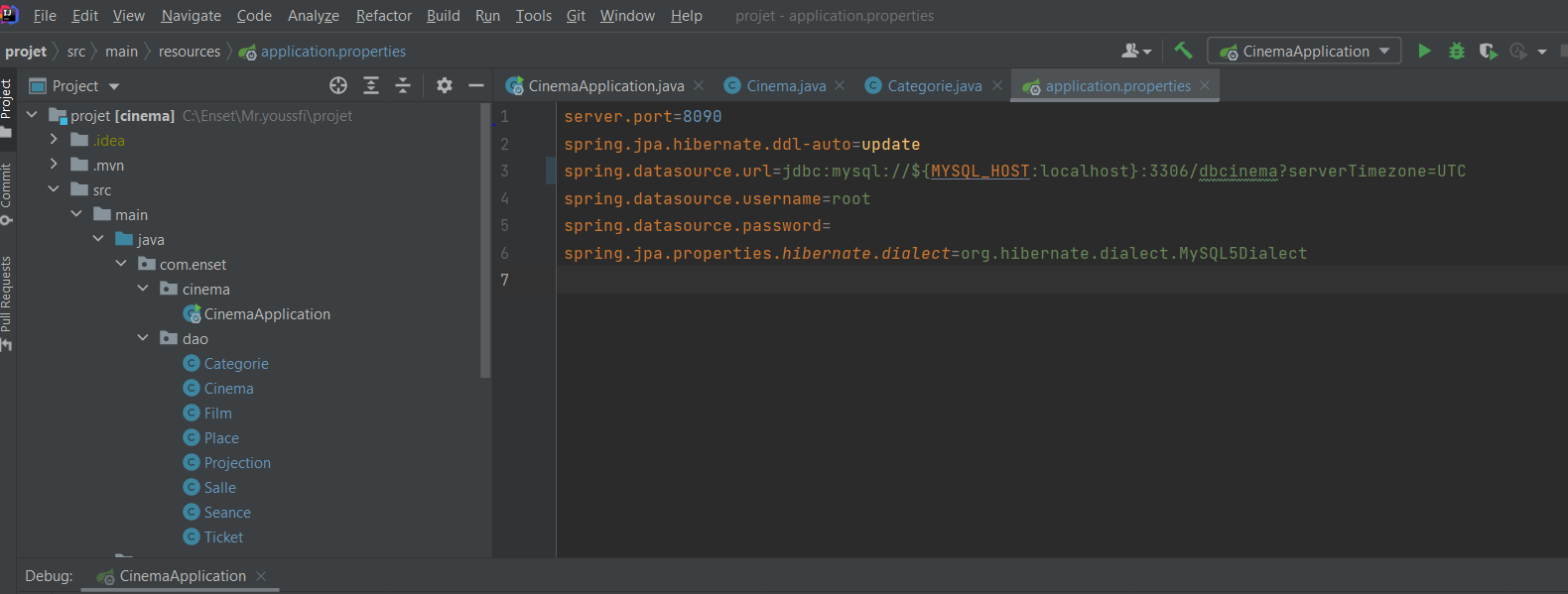
La conception des entités nécessaire pour la construction de la  base de données du projet gestion cinéma.



2- Partie Backend

* **La Couche DAO**

Nous utilisons Spring data JPA afin de créer notre base de données dbcinema avec MySql, ci-après notre fichier source application.properties contient les propriétés nécessaire de la création de la base de données  .



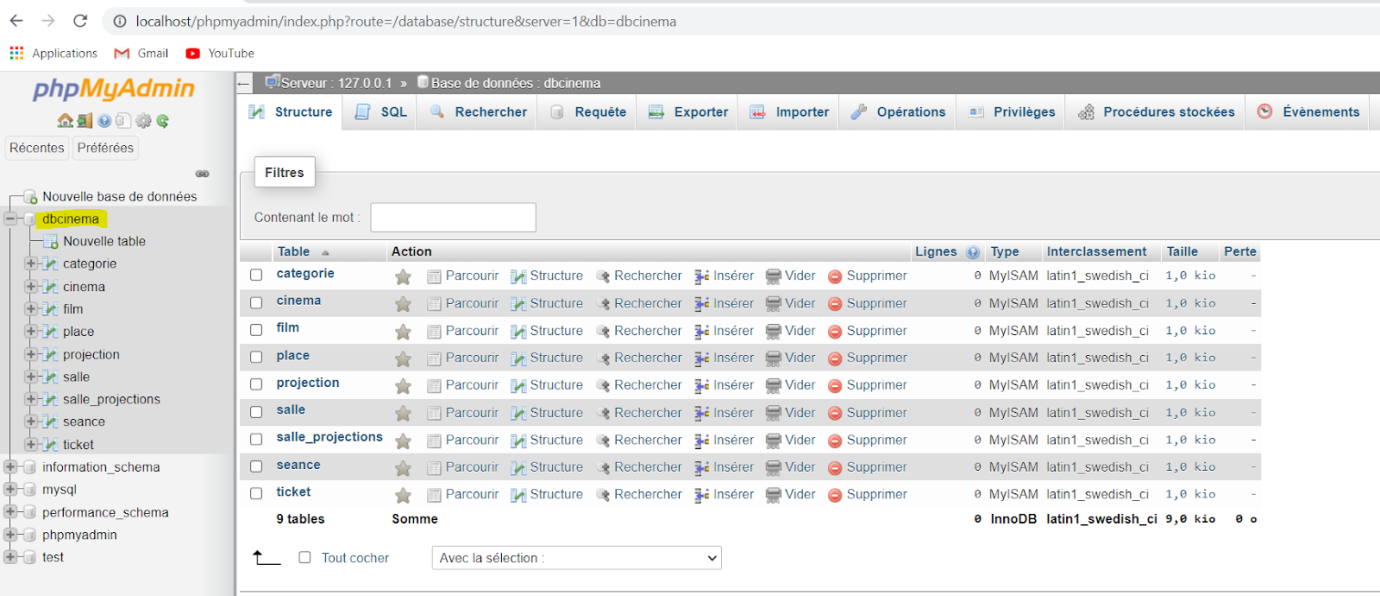
Après validation des entités par Spring data JPA voila les tables de notre base de données dbcinema  générer sur phpmyadmin.

Nb: la validation des entités par Spring data JPA se fait d’une manière automatique après la précision des propriétés datasource dans le fichier source application.properties . Spring analyse uniquement les classes  sous le package de classe principale main() ,mais dans le cas où les classes se trouvaient à l'extérieur du package principale ,dans ce cas la , il faut spécifier le chemin de ce package afin que spring peut les identifier.

Ajouter les notations suivante au programme main()

@EnableConfigurationProperties

@EntityScan(basePackages = {"com.enset.entites"})

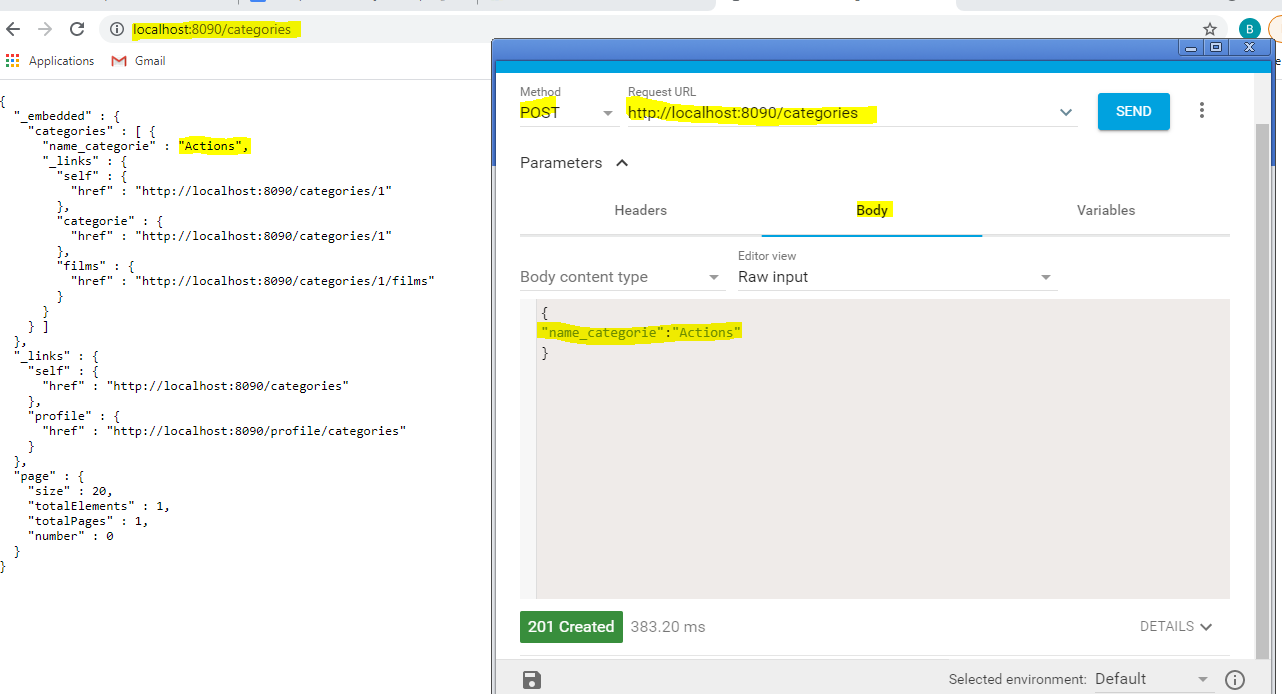


Interfaces JPA Repositories basées sur Spring Data

La première des choses en va créer une API REST pour chaque entité, en utilisant spring data Rest, en ajoutant la notation @RepositoryRestResource  à chaque interface, afin d’avoir accès aux méthodes via l’API REST.

L'interface Spring Data JpaRepository étend CrudRepository et fournit des méthodes de recherche prêtes à l'emploi.

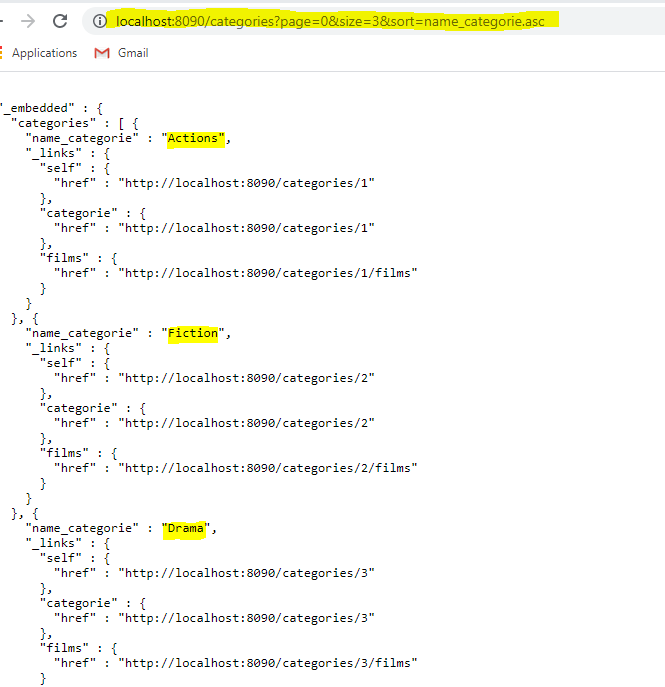
Nous utilisons la notation @EnableJpaRepositories(basePackages = "com.enset.dao") afin de permettre au spring d’analyser le package de base et tous ses sous-packages pour toutes les interfaces, spring va créer un lien pour gérer toutes les appels des méthodes de ces interfaces.



L’enregistrement des trois catégories dans la table categorie:



utilisation des filtres sur notre api



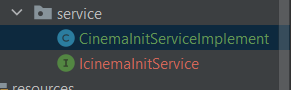
* **La couche Service (Métier)**

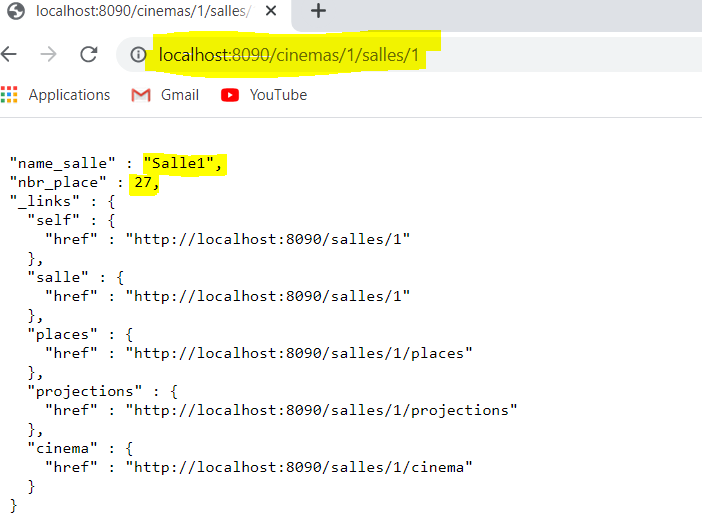
▪ Création de la classe CinemaInitServiceImplement

▪ Création de l’interface IcinemaInitService

▪ Créer une implémentation de cette interface

▪ Tester la couche métier





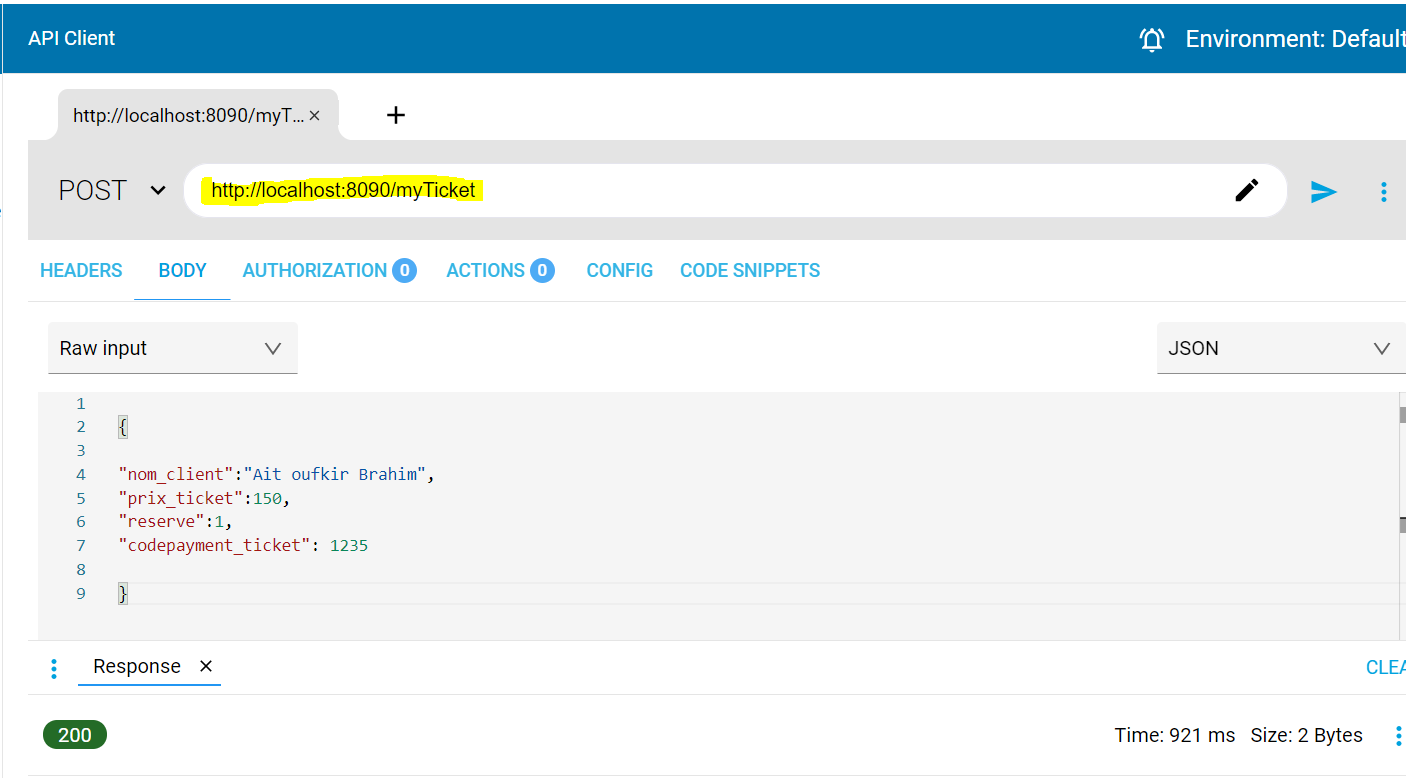
* **La couche Web**

▪ Création des Web services Restful basés sur Spring Data Rest

▪ Création des Web services Restful basés sur Spring RestController



▪ Tester la couche Web avec un client Rest ARC (Advanced Rest Client)



3- Partie Frontend :

Avant de commencer nous devons installer **Node.js** afin d’installer **Angular**

>\_ la commande ci-dessous install angular d’une manière global.

**https://lh3.googleusercontent.com/EJAd8irt85rHPyxIIgSI0B4_RGdUBaNKrDfJ55QyUKTvRI6a4gMG3joNnajtavdV93d9KBMlrundoQvfwHqDt5gLrYo_U7Q2Oxc1j-Ih2NiP32g4pTEqAi59ah-_NX-mCfcJV4Dc**

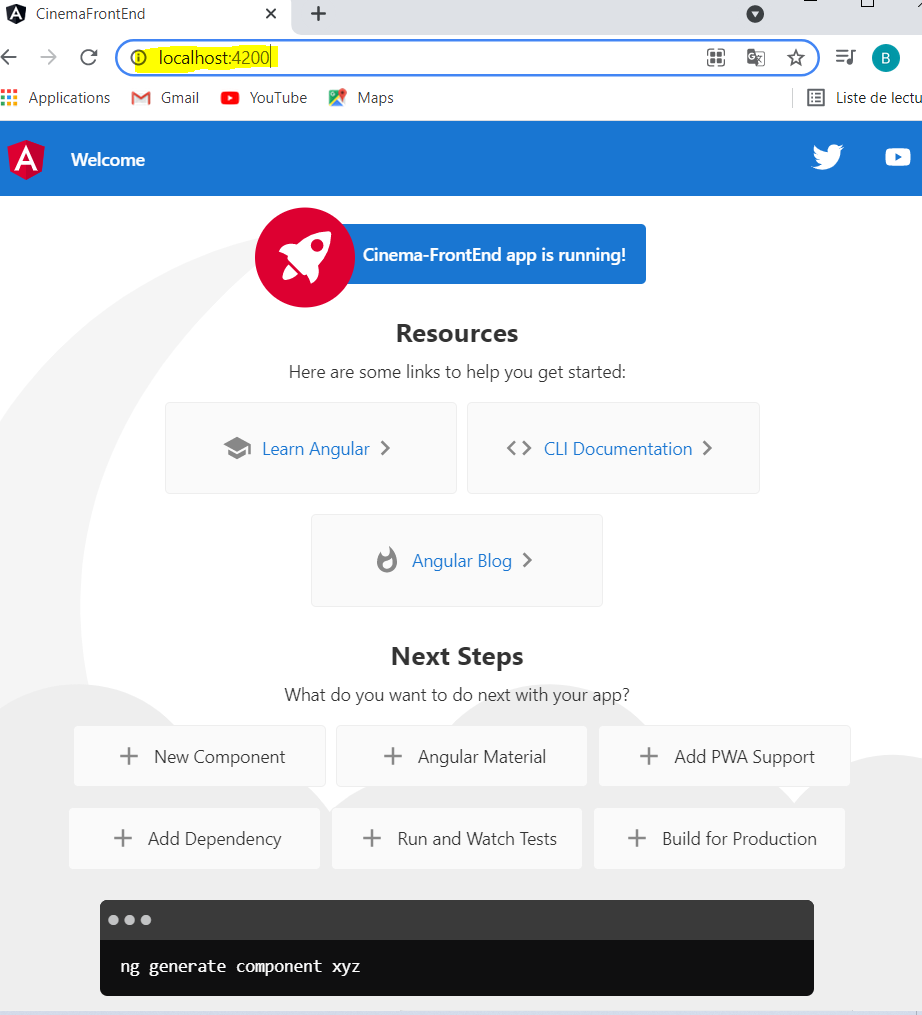
>\_ la commande ci-après créer un projet Angular.

**https://lh5.googleusercontent.com/G5x_kijBu-2ODJFHfY2Btn4-iVytFHu11lG_nF2QlhruFP6u0QxNSMRAigW58t9RNF2Y0WUmU0JozOsAQw08PiV-kYMyiiJHmOGrxix4KVrirjgM4cSCat2ogw710Fp6wQNgfmW5**

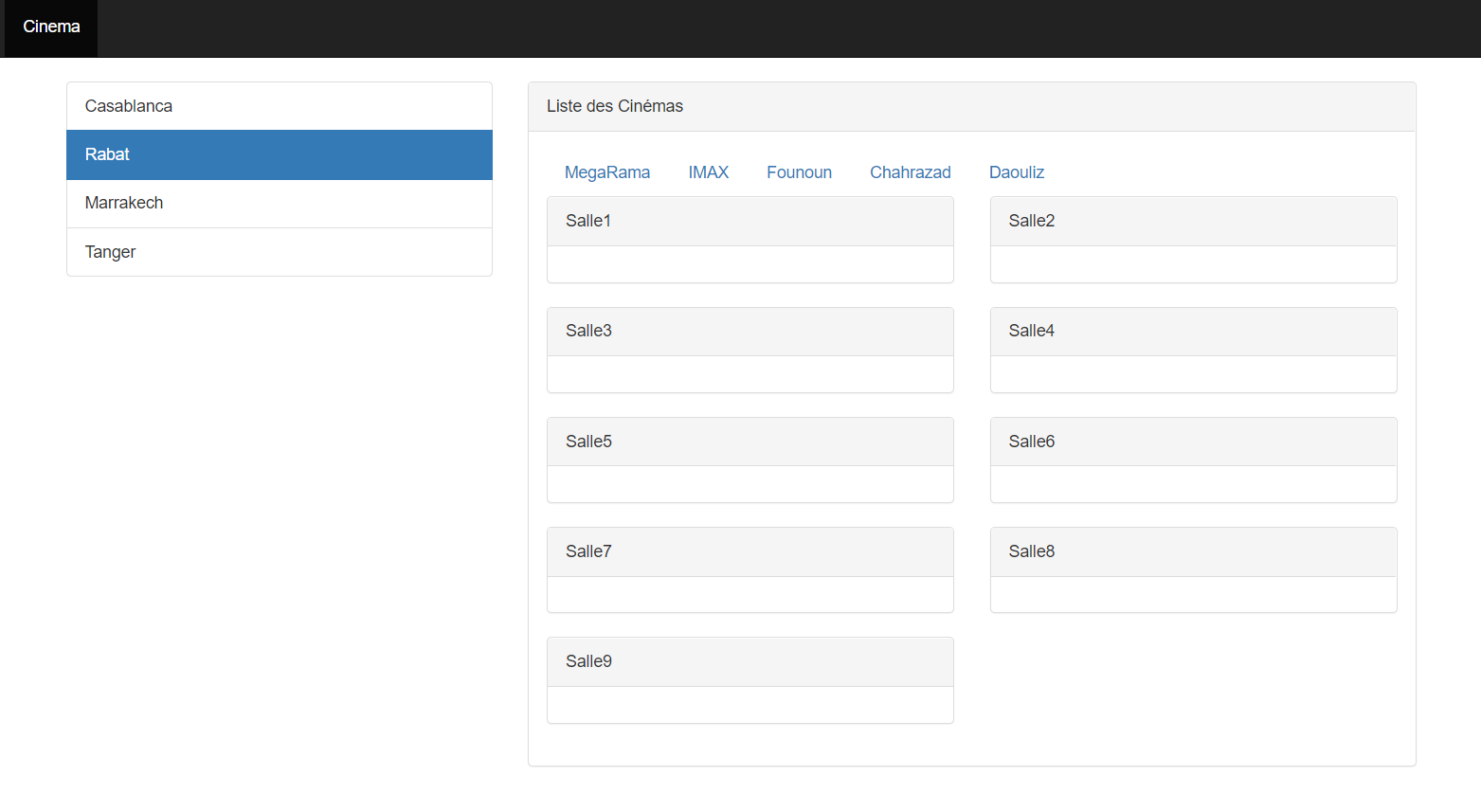
>\_ nous démarrons le serveur Angular avec cette commande .

**https://lh6.googleusercontent.com/lt7E0ig24wTTXxOaHOM_TfznZn8N4AZG2lOE1gr4IbGssdvpkKjMEQWoWSPCC_efEoZ6inbAq3nXT1I1qF-que8N8aRW7xeT7MX9-SzdmCHYD9duhZvhlUwWA69PLSnb4V3dGedp**

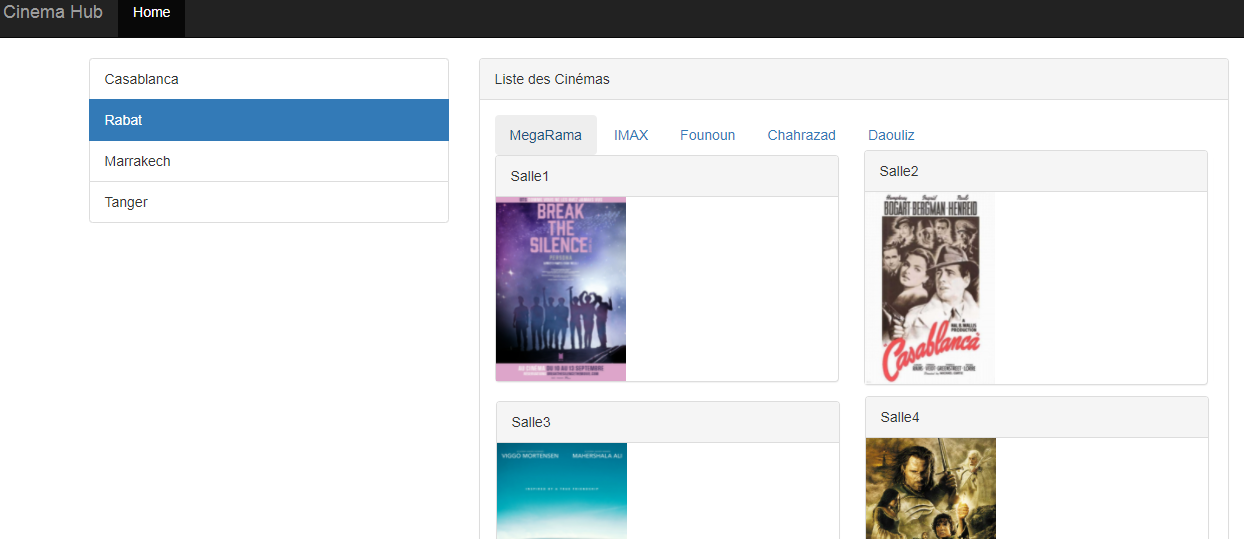
tapez sur l’adresse du  browser **http://localhost:4200/**

****

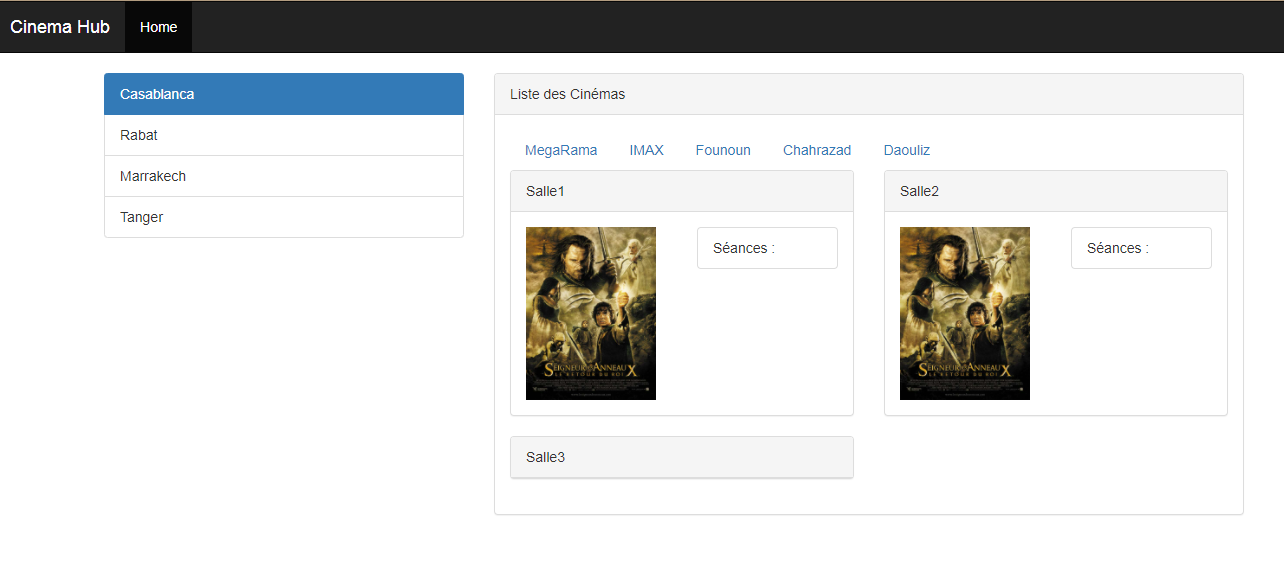
▪ Partie Front Office de Consultation des données du cinéma et réservation de tickets en utilisant Angular



j’ai afficher la maquet ci-dessous d’une manière aléatoire avec du bricolage dans le code



mais suivant les vidéos me donne ca



**merci d’avance afin que je puis terminer les tâches**

**voici le lien de code source sur mon dépôt GitHub**