Ecole Marocain Des Science De L'ingenieurie

Rapport du projet JEE

Developpement d'une application web JEE
Pour l'affichage des filmes
et gestion des cinemas

Realise par:

Waraki abdelbasset

Voici mon github pour consulter l'application web

https://github.com/warakiabdelbasset/cinemaJEE

Contents

Indroduction :

JEE (*Java Entreprise Edition*) est la version entreprise de la plate-forme "Java" qui se compose de l'environnement "JSE" ainsi que de nombreuses API et composants destinés à une utilisation "côté serveur" au sein du système d'information de l'entreprise. Il s'agit donc d'une évolution du Java.

Dès son origine, Java a révolutionné plusieurs domaines de l'informatique, que ça soit la téléphonie, l'internet ou les applications d'entreprise. Par la suite, Sun avait réorganisé son offre autour de trois briques :

- Java Micro Edition (JME) qui cible les terminaux portables.
- Java Standard Edition (JSE) qui vise les postes clients.
- Java Entreprise Edition (JEE) qui définit le cadre d'un serveur d'applications et d'intégration.

Maven project

Maven est un outil de construction de projets (build) open source développé par la fondation Apache, initialement pour les besoins du projet Jakarta Turbine. Il permet de faciliter et d'automatiser certaines tâches de la gestion d'un projet Java. Il permet notamment :

- d'automatiser certaines tâches : compilation, tests unitaires et déploiement des applications qui composent le projet
- de gérer des dépendances vis-à-vis des bibliothèques nécessaires au projet
- de générer des documentations concernant le projet

Au premier abord, il est facile de croire que Maven fait double emploi avec Ant. Ant et Maven sont tous les deux développés par le groupe Jakarta, ce qui prouve bien que leur utilité n'est pas aussi identique que cela. Ant, dont le but est d'automatiser certaines tâches répétitives, est plus ancien que Maven. Maven propose non seulement les fonctionnalités d'Ant mais en propose de nombreuses autres.

Pour gérer les dépendances du projet vis-à-vis de bibliothèques, Maven utilise un ou plusieurs dépôts qui peuvent être locaux ou distants.

Maven est extensible grâce à un mécanisme de plugins qui permettent d'ajouter des fonctionnalités.

Spring:

Spring est un framework open source pour construire et définir l'infrastructure d'une application Java, dont il facilite le développement et les tests. En 2004, Rod Johnson a écrit le livre Expert One-on-One J2EE Design and Development qui explique les raisons de la création de Spring

1. Spring boot

Spring Boot est un framework qui facilite le développement d'applications fondées sur Spring en offrant des outils permettant d'obtenir une application packagée en jar , totalement autonome. Ce qui nous intéresse particulièrement, puisque nous essayons de développer des Microservices

2. Spring security

Spring Security est un framework Java / Java EE qui fournit l'authentification, l'autorisation et d'autres fonctionnalités de sécurité pour les applications d'entreprise. Le projet a été lancé fin 2003 sous le nom de «Acegi Security» par Ben Alex, et il a été rendu public sous la licence Apache en mars 2004

Thymeleaf:

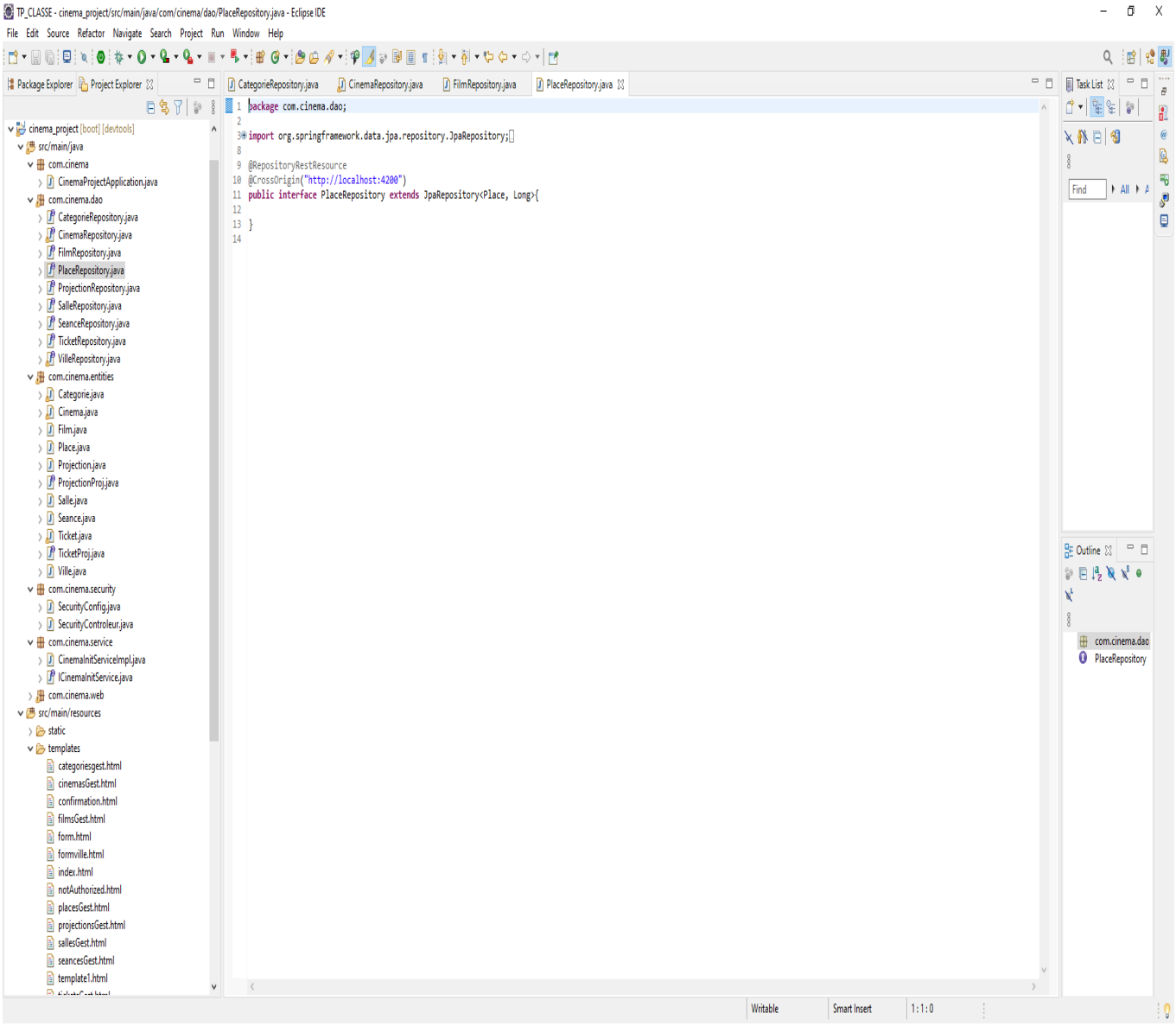
Angular:

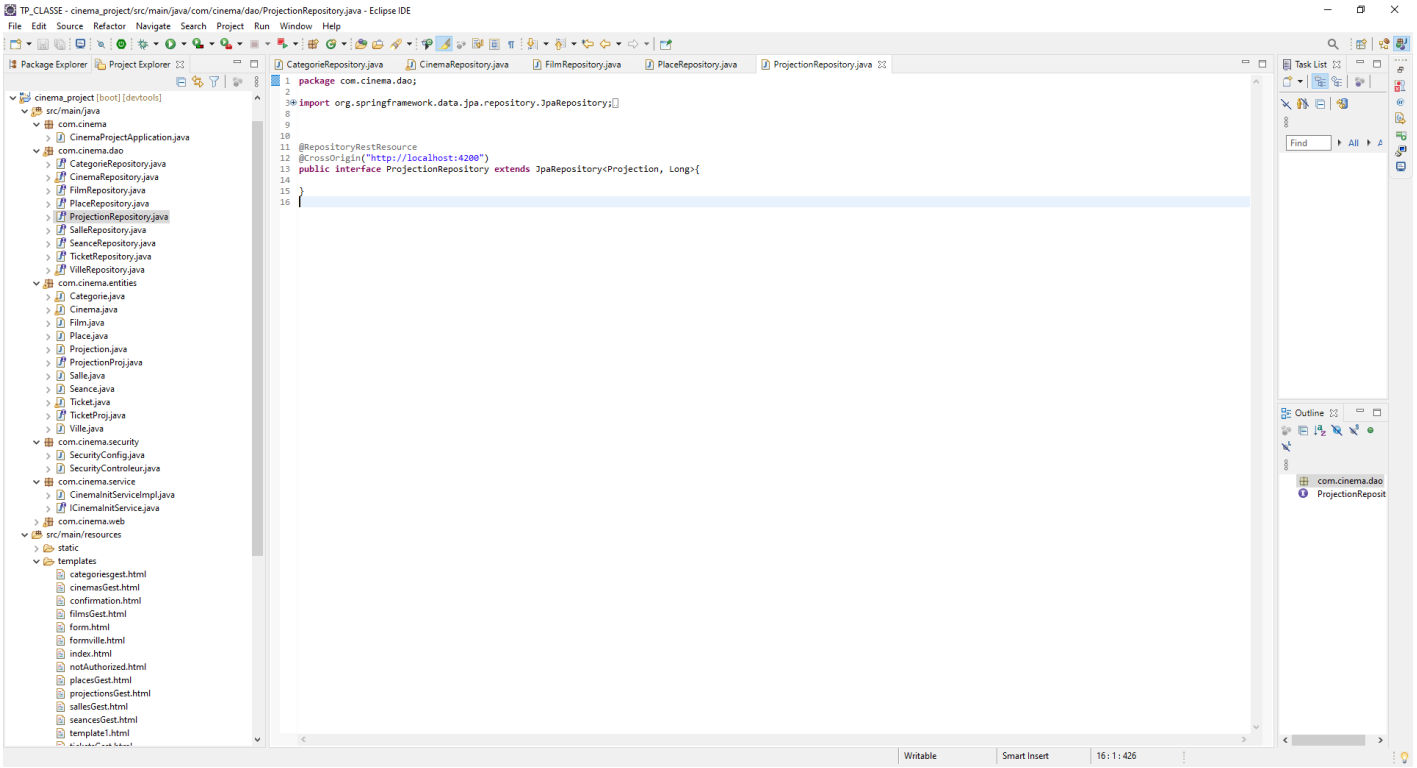
Capture d’ecran et commentaire:

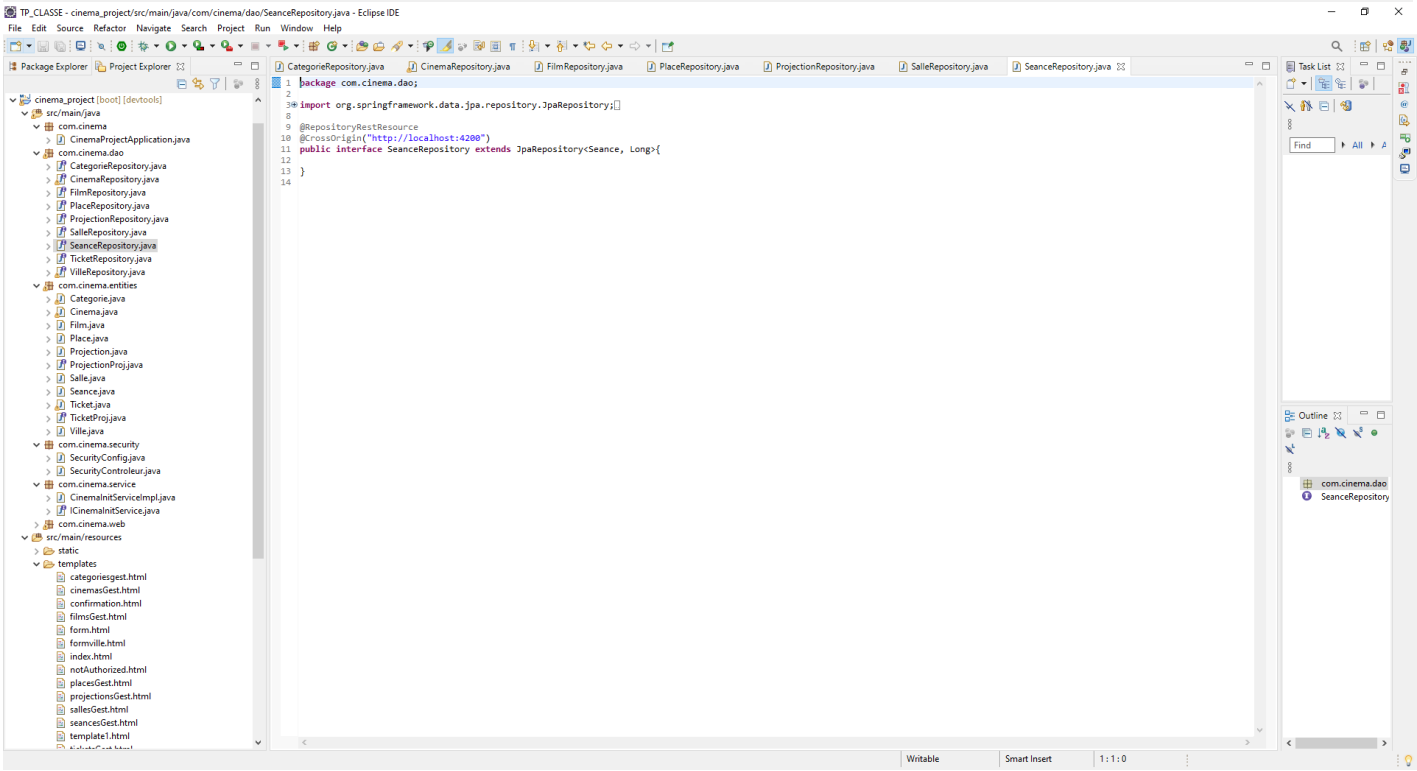
- DAO

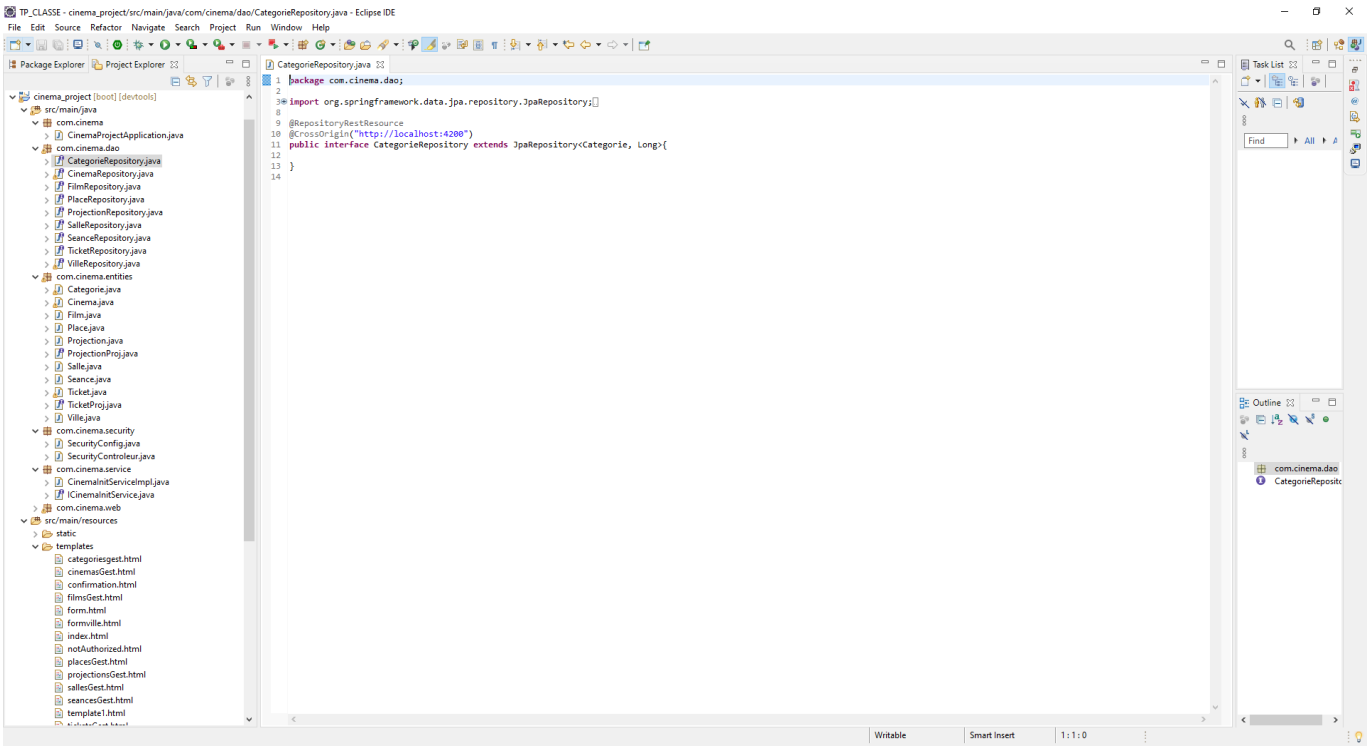
Tout d’abord on commencent par la partie interface .En va cree les interfaces avec les methodes quand va utilizer pour cree les classes entities,car nous etes dans l’heritage faible qui s’inpose sur la relation entre les interface et les class parceque si on veulent faire une maintenance il soit simple a l’integre.

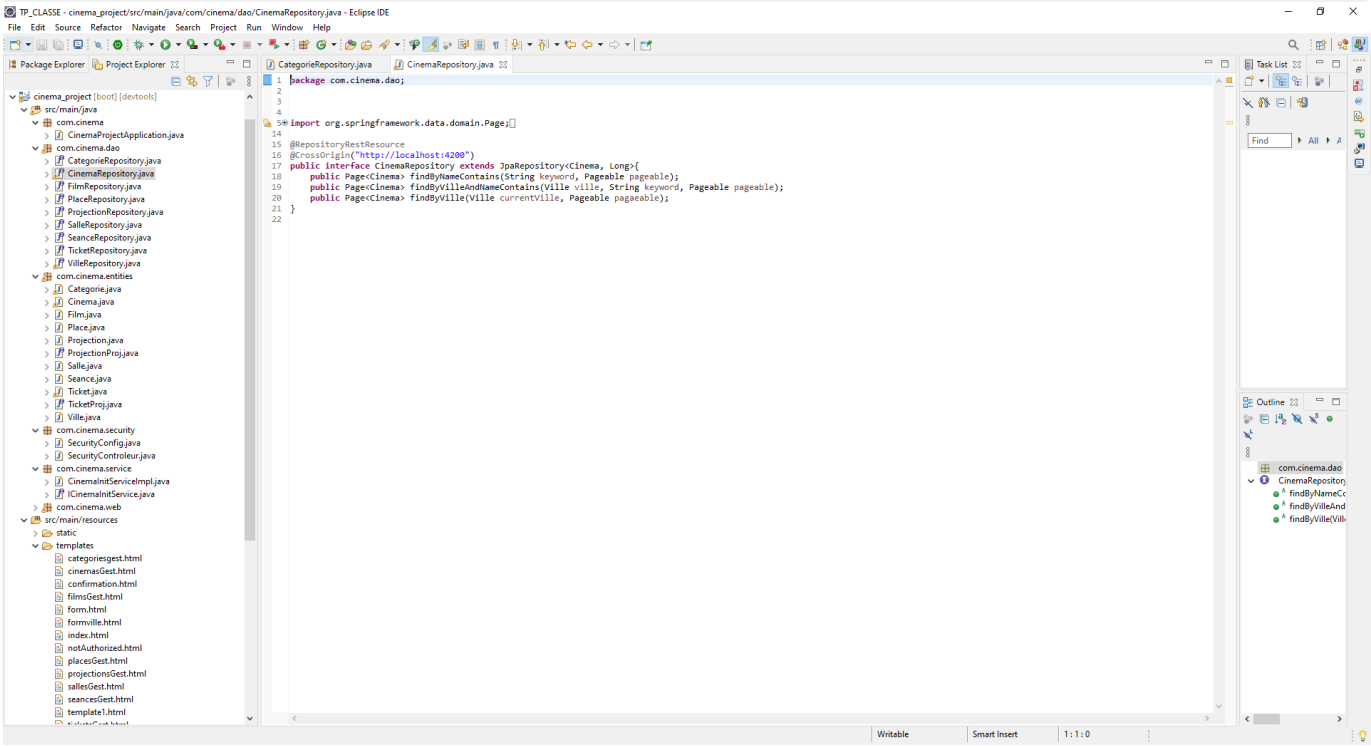
Voici quelque screenshote des interfaces:

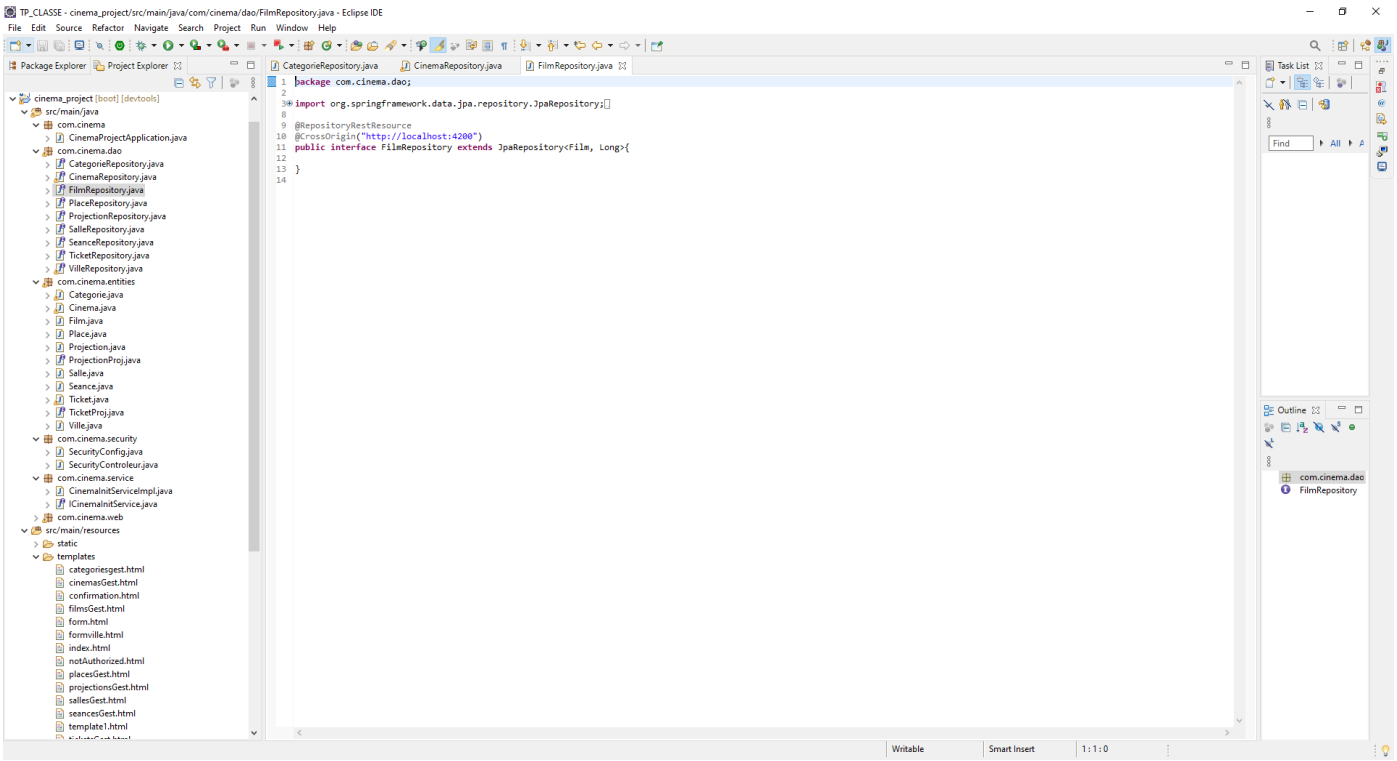

TP_CLASSE - cinema_project/src/main/java/com/cinema/dao/PlaceRepository.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
cinema_project [boot] [devtools]
src/main/java
com.cinema
CinemaProjectApplication.java
com.cinema.dao
CategorieRepository.java
CinemaRepository.java
FilmRepository.java
PlaceRepository.java
ProjectionRepository.java
SalleRepository.java
SeanceRepository.java
TicketRepository.java
VilleRepository.java
com.cinema.entities
Categorie.java
Cinema.java
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
CategorieRepository.java
CinemaRepository.java
FilmRepository.java
PlaceRepository.java
1 package com.cinema.dao;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
8
9 @RepositoryRestResource
10 @CrossOrigin("http://localhost:4200")
11 public interface PlaceRepository extends JpaRepository<Place, Long>{
12
13 }
14
Task List
Find
All
Outline
com.cinema.dao
PlaceRepository
Writable
Smart Insert
1 : 1 : 0


TP_CLASSE - cinema_project/src/main/java/com/cinema/dao/ProjectionRepository.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
cinema_project [boot] [devtools]
src/main/java
com.cinema
CinemaProjectApplication.java
com.cinema.dao
CategorieRepository.java
CinemaRepository.java
FilmRepository.java
PlaceRepository.java
ProjectionRepository.java
SalleRepository.java
SeanceRepository.java
TicketRepository.java
VilleRepository.java
com.cinema.entities
Categorie.java
Cinema.java
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
CategorieRepository.java
CinemaRepository.java
FilmRepository.java
PlaceRepository.java
ProjectionRepository.java
1 package com.cinema.dao;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
8
9
10
11 @RepositoryRestResource
12 @CrossOrigin("http://localhost:4200")
13 public interface ProjectionRepository extends JpaRepository<Projection, Long>{
14
15 }
16
Task List
Find
All
Outline
com.cinema.dao
ProjectionReposit
Writable
Smart Insert
16 : 1 : 426


TP_CLASSE - cinema_project/src/main/java/com/cinema/dao/SeanceRepository.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
cinema_project [boot] [devtools]
src/main/java
com.cinema
CinemaProjectApplication.java
com.cinema.dao
CategorieRepository.java
CinemaRepository.java
FilmRepository.java
PlaceRepository.java
ProjectionRepository.java
SalleRepository.java
SeanceRepository.java
TicketRepository.java
VilleRepository.java
com.cinema.entities
Categorie.java
Cinema.java
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
CategorieRepository.java
CinemaRepository.java
FilmRepository.java
PlaceRepository.java
ProjectionRepository.java
SalleRepository.java
SeanceRepository.java
1 package com.cinema.dao;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
8
9 @RepositoryRestResource
10 @CrossOrigin("http://localhost:4200")
11 public interface SeanceRepository extends JpaRepository<Seance, Long>{
12
13 }
14
Task List
Find
All
Outline
com.cinema.dao
SeanceRepository
Writable
Smart Insert
1 : 1 : 0


TP_CLASSE - cinema_project/src/main/java/com/cinema/dao/CategorieRepository.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
CategorieRepository.java
Task List
Outline
cinema_project [boot] [devtools]
src/main/java
com.cinema
CinemaProjectApplication.java
com.cinema.dao
CategorieRepository.java
CinemaRepository.java
FilmRepository.java
PlaceRepository.java
ProjectionRepository.java
SalleRepository.java
SeanceRepository.java
TicketRepository.java
VilleRepository.java
com.cinema.entities
Categorie.java
Cinema.java
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
package com.cinema.dao;
import org.springframework.data.jpa.repository.JpaRepository;
@RepositoryRestResource
@CrossOrigin("http://localhost:4200")
public interface CategorieRepository extends JpaRepository<Categorie, Long>{
}
Writable
Smart Insert
1 : 1 : 0

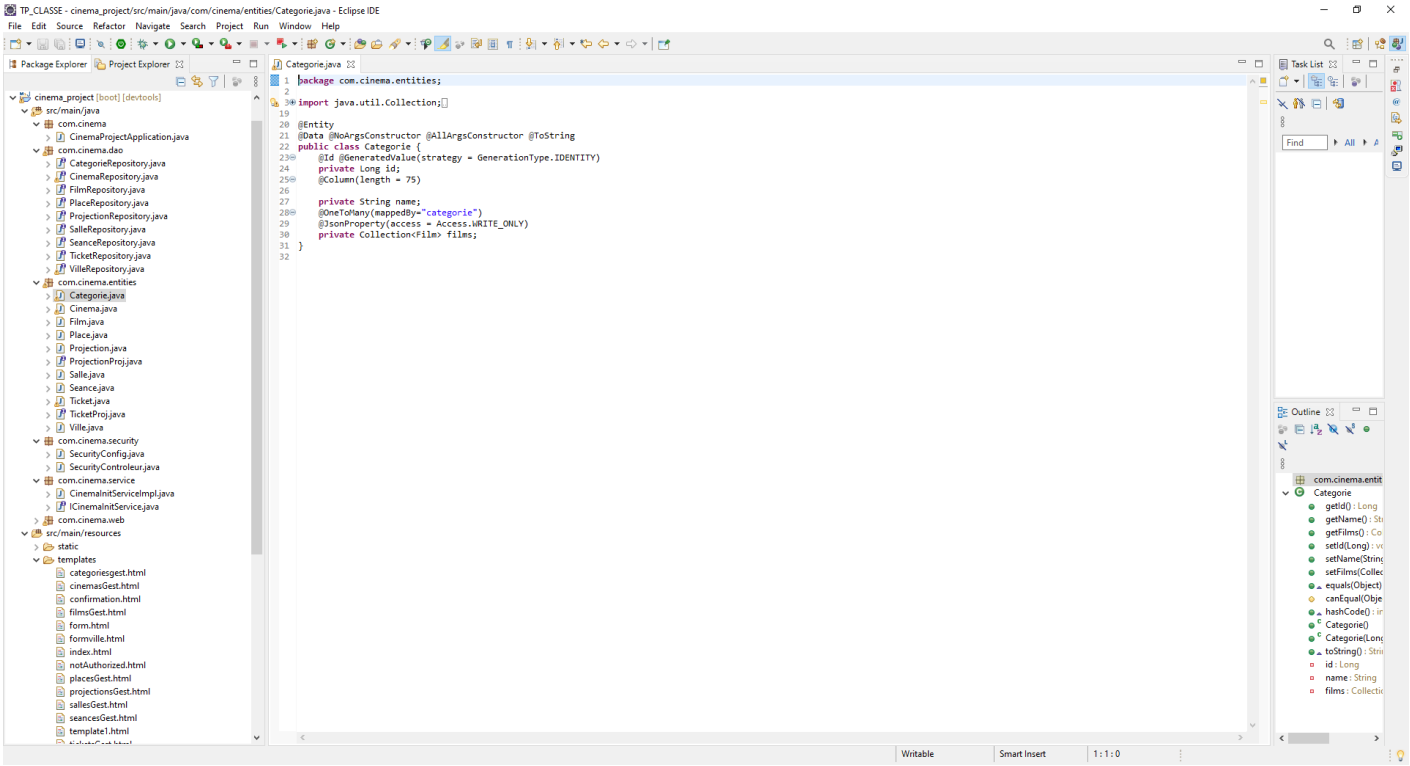

TP_CLASSE - cinema_project/src/main/java/com/cinema/dao/CinemaRepository.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
CategorieRepository.java
CinemaRepository.java
Task List
Outline
cinema_project [boot] [devtools]
src/main/java
com.cinema
CinemaProjectApplication.java
com.cinema.dao
CategorieRepository.java
CinemaRepository.java
FilmRepository.java
PlaceRepository.java
ProjectionRepository.java
SalleRepository.java
SeanceRepository.java
TicketRepository.java
VilleRepository.java
com.cinema.entities
Categorie.java
Cinema.java
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
package com.cinema.dao;
import org.springframework.data.domain.Page;
@RepositoryRestResource
@CrossOrigin("http://localhost:4200")
public interface CinemaRepository extends JpaRepository<Cinema, Long>{
public Page<Cinema> findByNameContains(String keyword, Pageable pageable);
public Page<Cinema> findByVilleAndNameContains(Ville ville, String keyword, Pageable pageable);
public Page<Cinema> findByVille(Ville currentVille, Pageable pageable);
}
Writable
Smart Insert
1 : 1 : 0

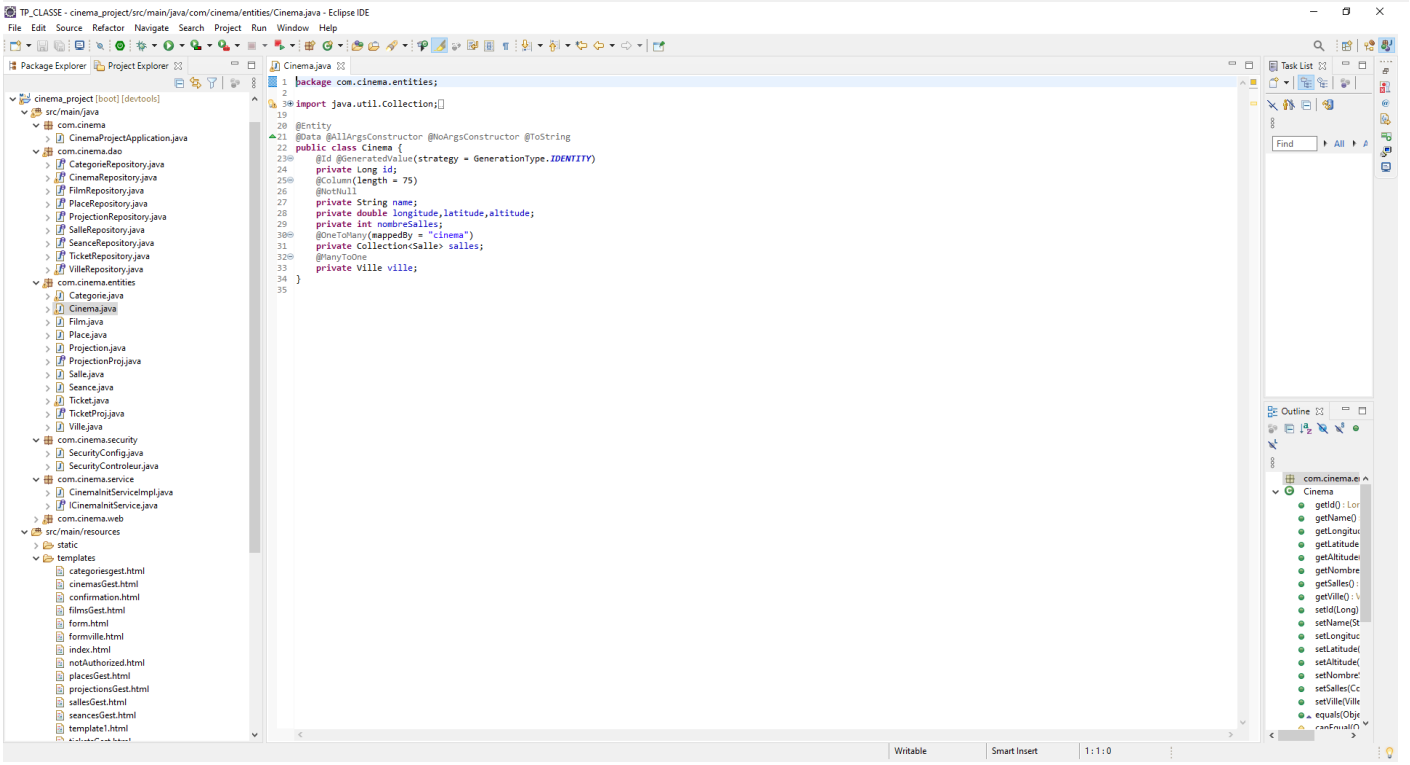


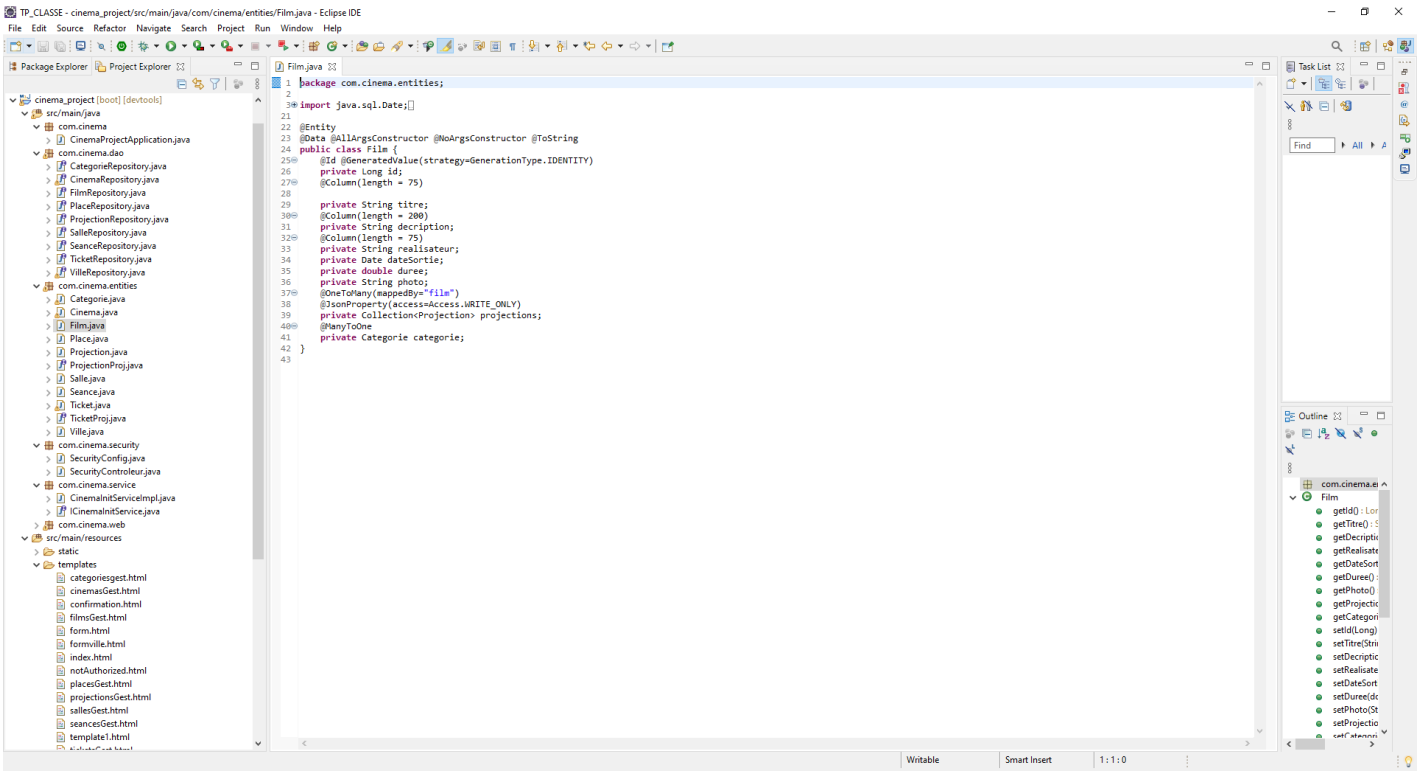
- Entities

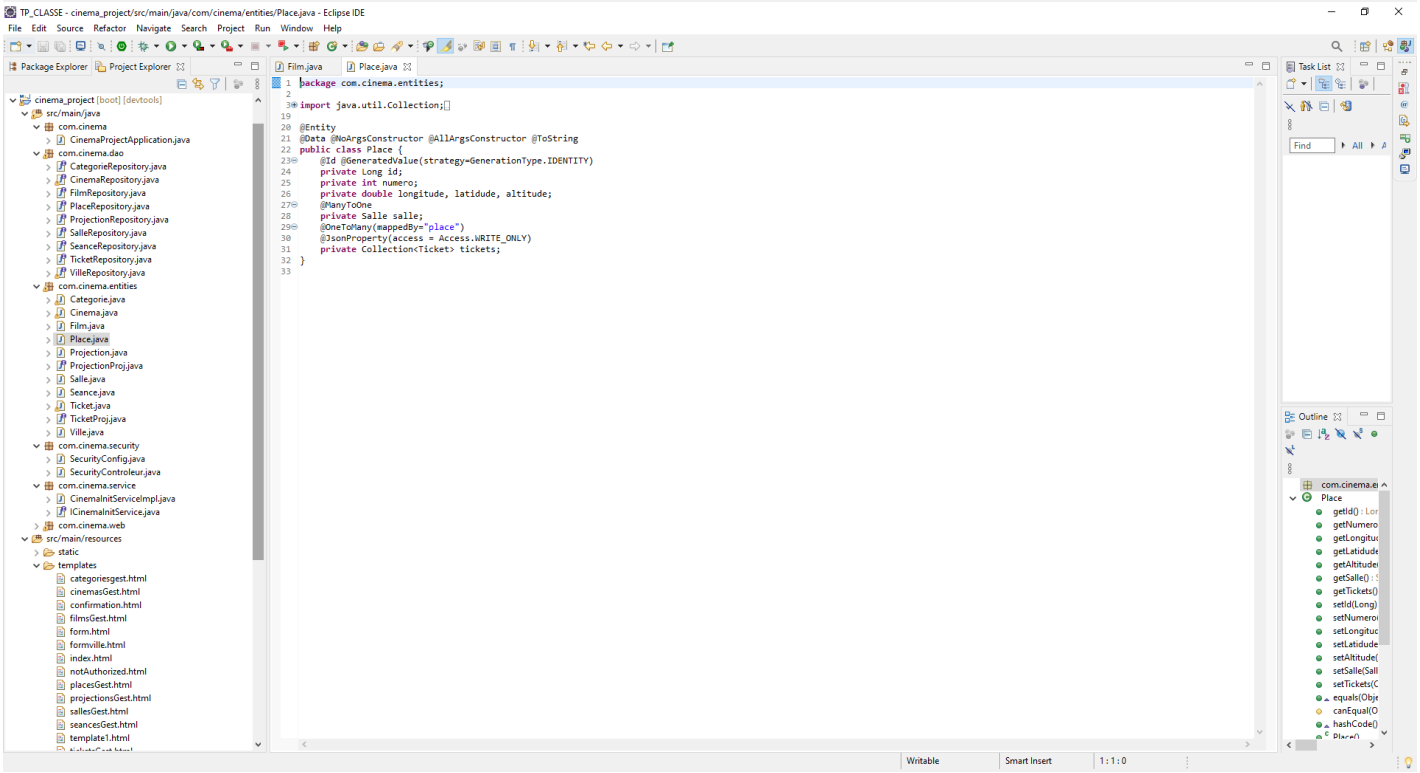
Puis en passe au deusieme package .En va cree les class qui corespondant aux interfaces et en va initializer et redefinir les method cree dans les interfaces. Et ce package (entities) va s'occupe de creation des tables on base de donne .

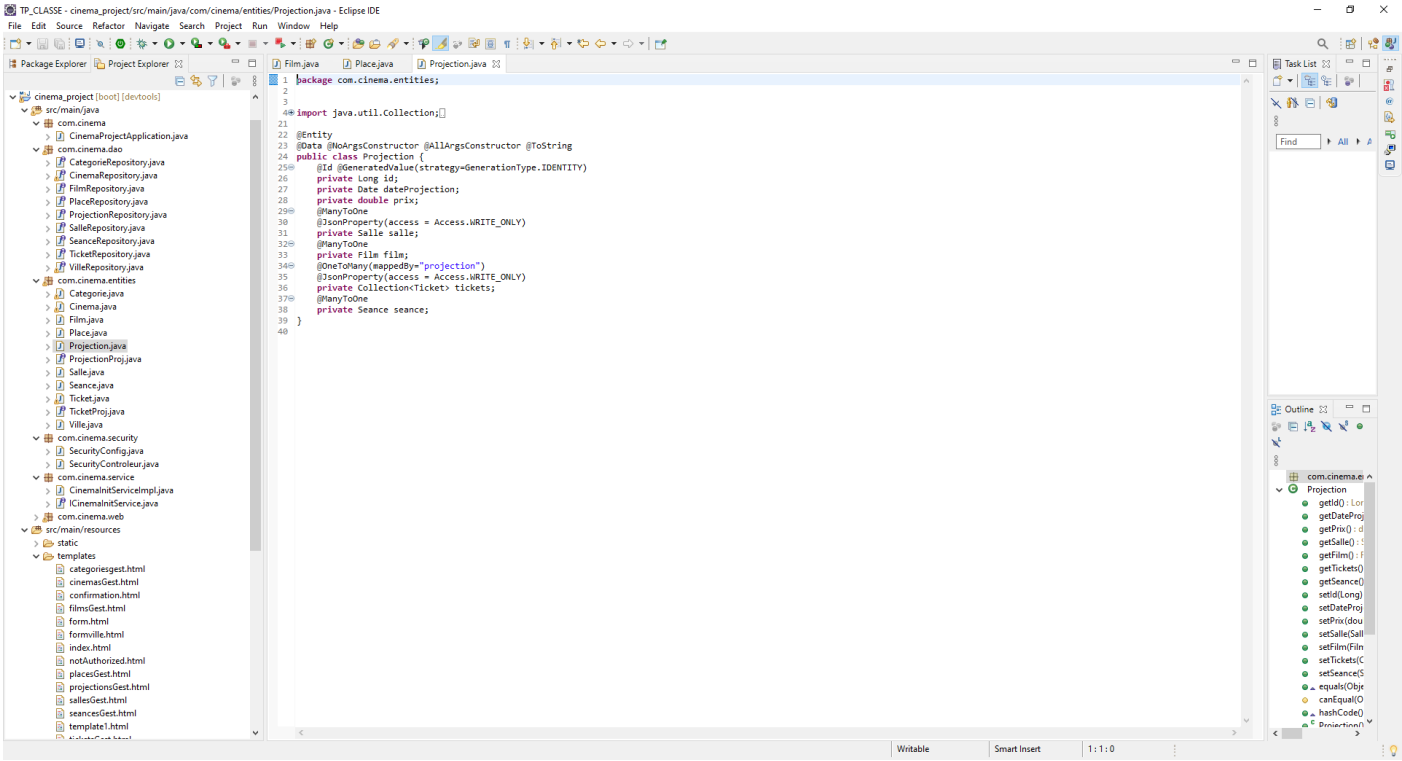
Voici quelque screen shoot de ces classes:

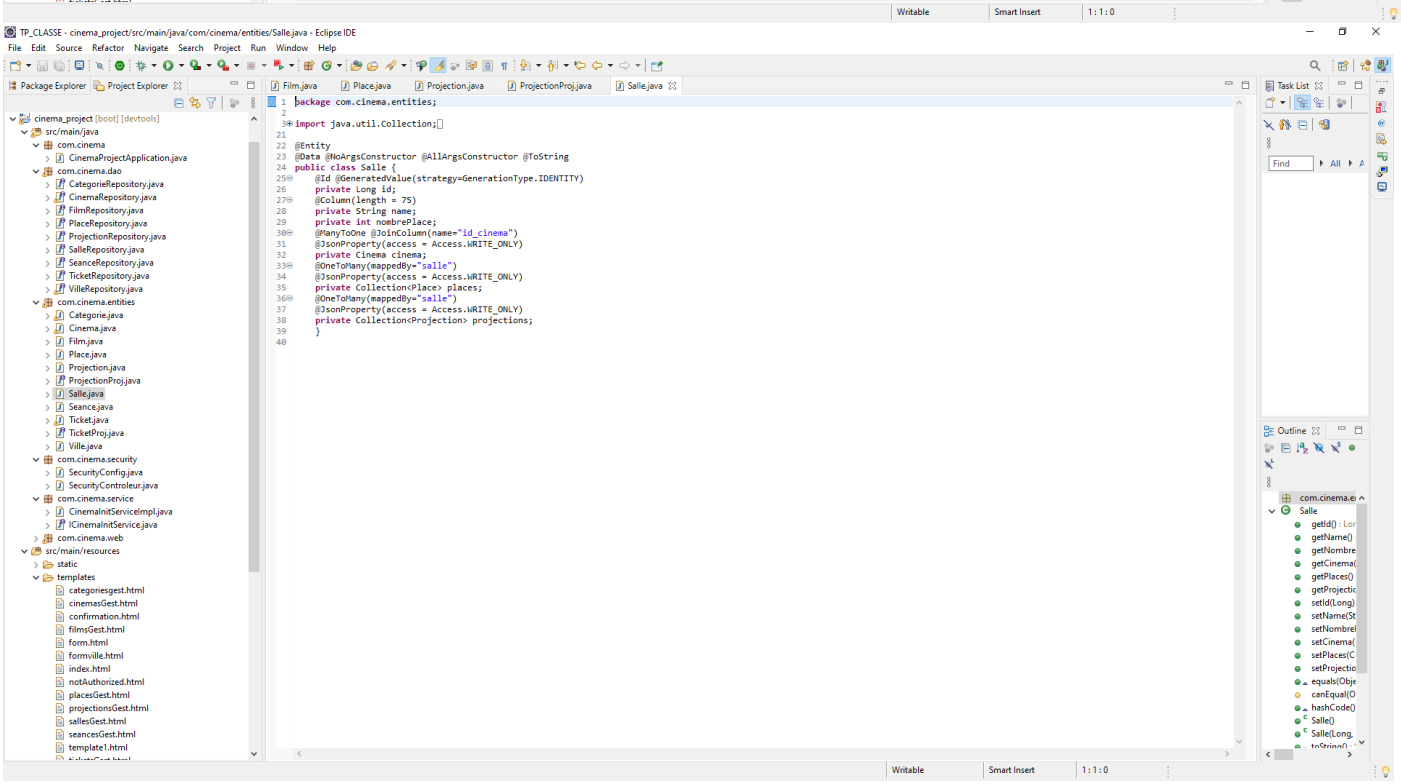

TP_CLASSE - cinema_project/src/main/java/com/cinema/entities/Categorie.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
Categorie.java
Task List
Outline
cinema_project [boot] [devtools]
src/main/java
com.cinema
CinemaProjectApplication.java
com.cinema.dao
CategorieRepository.java
CinemaRepository.java
FilmRepository.java
PlaceRepository.java
ProjectionRepository.java
SalleRepository.java
SeanceRepository.java
TicketRepository.java
VilleRepository.java
com.cinema.entities
Categorie.java
Cinema.java
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
package com.cinema.entities;
import java.util.Collection;
@Entity
@Data @NoArgsConstructor @AllArgsConstructor @ToString
public class Categorie {
@Id @GeneratedValue(strategy = GenerationType.IDENTITY)
private Long id;
@Column(length = 75)
private String name;
@OneToMany(mappedBy="categorie")
@JsonProperty(access = Access.WRITE_ONLY)
private Collection<Film> films;
}
Writable
Smart Insert
1 : 1 : 0

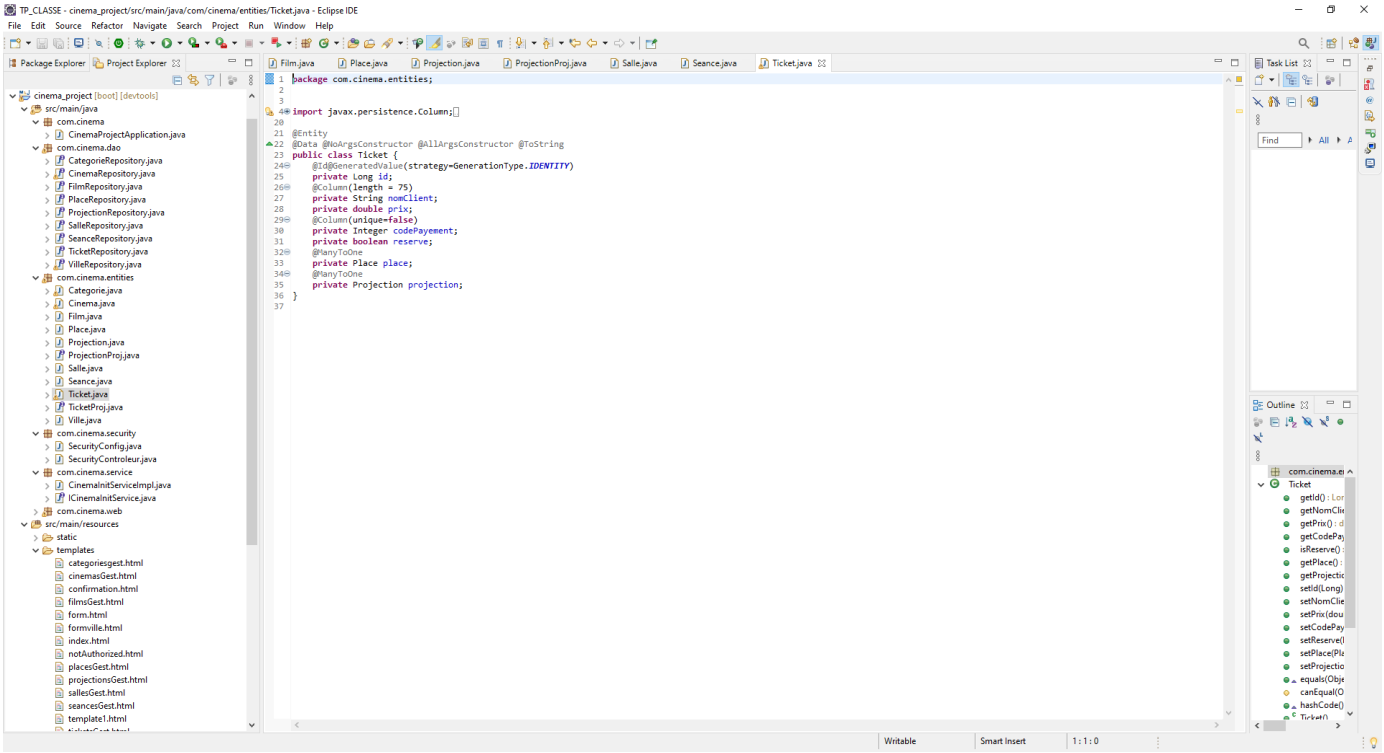

TP_CLASSE - cinema_project/src/main/java/com/cinema/entities/Cinema.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
Cinema.java
Task List
Outline
cinema_project [boot] [devtools]
src/main/java
com.cinema
CinemaProjectApplication.java
com.cinema.dao
CategorieRepository.java
CinemaRepository.java
FilmRepository.java
PlaceRepository.java
ProjectionRepository.java
SalleRepository.java
SeanceRepository.java
TicketRepository.java
VilleRepository.java
com.cinema.entities
Categorie.java
Cinema.java
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
package com.cinema.entities;
import java.util.Collection;
@Entity
@Data @AllArgsConstructor @NoArgsConstructor @ToString
public class Cinema {
@Id @GeneratedValue(strategy = GenerationType.IDENTITY)
private Long id;
@Column(length = 75)
@NotNull
private String name;
private double longitude,latitude,altitude;
private int nombreSalles;
@OneToMany(mappedBy = "cinema")
private Collection<Salle> salles;
@ManyToOne
private Ville ville;
}
Writable
Smart Insert
1 : 1 : 0

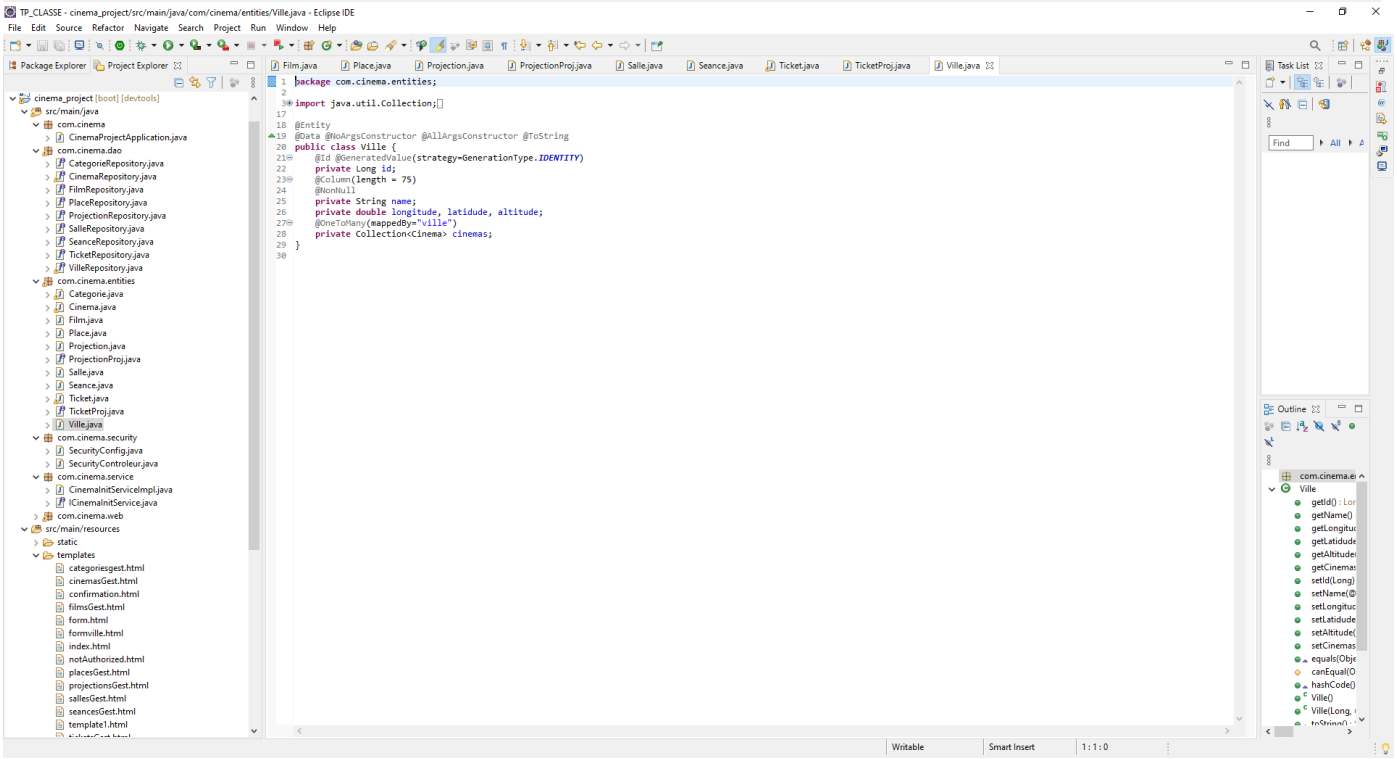

TP_CLASSE - cinema_project/src/main/java/com/cinema/entities/Film.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
cinema_project [boot] [devtools]
src/main/java
com.cinema
CinemaProjectApplication.java
com.cinema.dao
CategorieRepository.java
CinemaRepository.java
FilmRepository.java
PlaceRepository.java
ProjectionRepository.java
SalleRepository.java
SeanceRepository.java
TicketRepository.java
VilleRepository.java
com.cinema.entities
Categorie.java
Cinema.java
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
Film.java
1 package com.cinema.entities;
2
3 import java.sql.Date;
21
22 @Entity
23 @Data @AllArgsConstructor @NoArgsConstructor @ToString
24 public class Film {
25 @Id @GeneratedValue(strategy=GenerationType.IDENTITY)
26 private Long id;
27 @Column(length = 75)
28
29 private String titre;
30 @Column(length = 200)
31 private String decription;
32 @Column(length = 75)
33 private String realisateur;
34 private Date dateSortie;
35 private double duree;
36 private String photo;
37 @OneToMany(mappedBy="film")
38 @JsonProperty(access=Access.WRITE_ONLY)
39 private Collection<Projection> projections;
40 @ManyToOne
41 private Categorie categorie;
42 }
43
Task List
Outline
Writable
Smart Insert
1 : 1 : 0

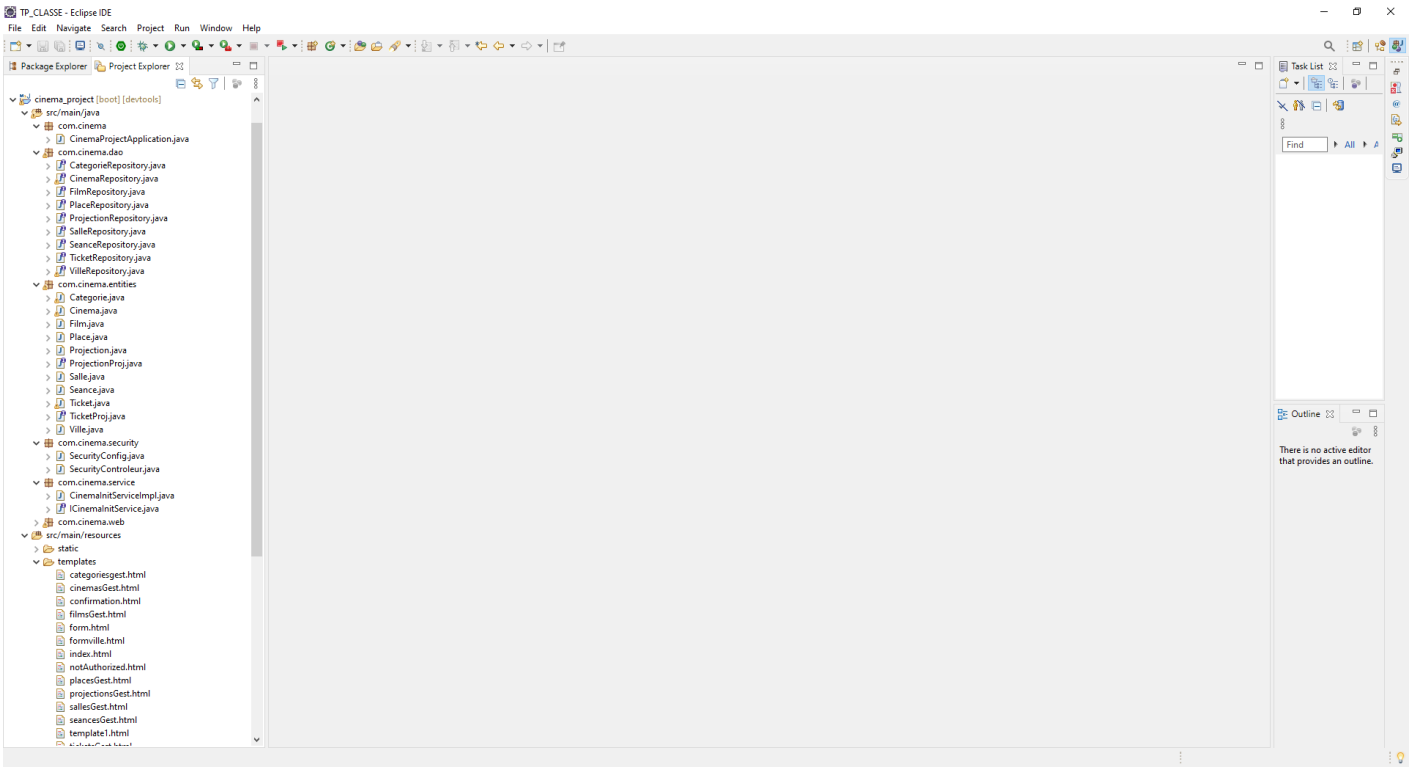

TP_CLASSE - cinema_project/src/main/java/com/cinema/entities/Place.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
cinema_project [boot] [devtools]
src/main/java
com.cinema
CinemaProjectApplication.java
com.cinema.dao
CategorieRepository.java
CinemaRepository.java
FilmRepository.java
PlaceRepository.java
ProjectionRepository.java
SalleRepository.java
SeanceRepository.java
TicketRepository.java
VilleRepository.java
com.cinema.entities
Categorie.java
Cinema.java
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
Film.java
Place.java
1 package com.cinema.entities;
2
3 import java.util.Collection;
19
20 @Entity
21 @Data @NoArgsConstructor @AllArgsConstructor @ToString
22 public class Place {
23 @Id @GeneratedValue(strategy=GenerationType.IDENTITY)
24 private Long id;
25 private int numero;
26 private double longitude, latidude, altitude;
27 @ManyToOne
28 private Salle salle;
29 @OneToMany(mappedBy="place")
30 @JsonProperty(access = Access.WRITE_ONLY)
31 private Collection<Ticket> tickets;
32 }
33
Task List
Outline
Writable
Smart Insert
1 : 1 : 0

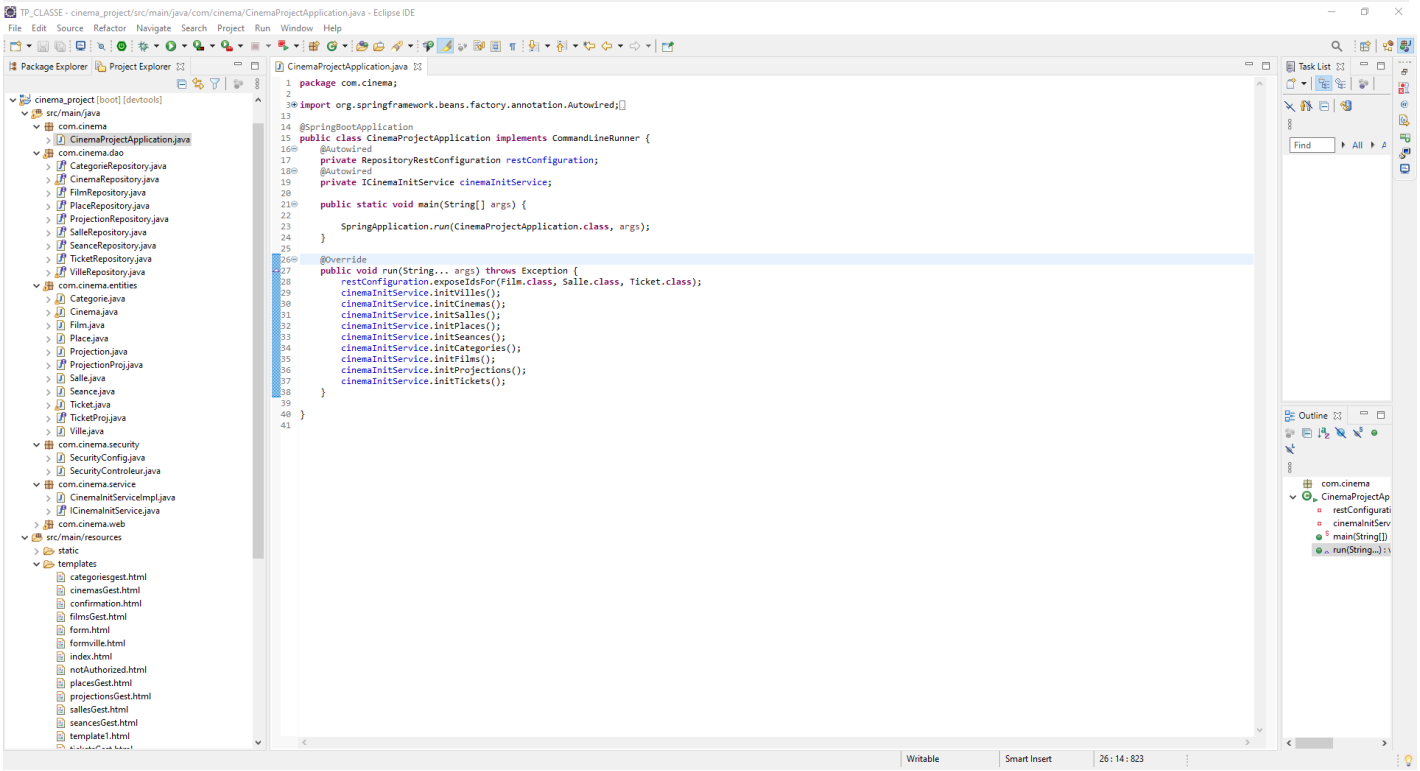

TP_CLASSE - cinema_project/src/main/java/com/cinema/entities/Projection.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
Film.java
Place.java
Projection.java
Task List
Outline
cinema_project [boot] [devtools]
src/main/java
com.cinema
CinemaProjectApplication.java
com.cinema.dao
CategorieRepository.java
CinemaRepository.java
FilmRepository.java
PlaceRepository.java
ProjectionRepository.java
SalleRepository.java
SeanceRepository.java
TicketRepository.java
VilleRepository.java
com.cinema.entities
Categorie.java
Cinema.java
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
1 package com.cinema.entities;
3 import java.util.Collection;
22 @Entity
23 @Data @NoArgsConstructor @AllArgsConstructor @ToString
24 public class Projection {
25 @Id @GeneratedValue(strategy=GenerationType.IDENTITY)
26 private Long id;
27 private Date dateProjection;
28 private double prix;
29 @ManyToOne
30 @JsonProperty(access = Access.WRITE_ONLY)
31 private Salle salle;
32 @ManyToOne
33 private Film film;
34 @OneToMany(mappedBy="projection")
35 @JsonProperty(access = Access.WRITE_ONLY)
36 private Collection<Ticket> tickets;
37 @ManyToOne
38 private Seance seance;
39 }
Writable
Smart Insert
1 : 1 : 0


TP_CLASSE - cinema_project/src/main/java/com/cinema/entities/Salle.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Task List
Outline
cinema_project [boot] [devtools]
src/main/java
com.cinema
CinemaProjectApplication.java
com.cinema.dao
CategorieRepository.java
CinemaRepository.java
FilmRepository.java
PlaceRepository.java
ProjectionRepository.java
SalleRepository.java
SeanceRepository.java
TicketRepository.java
VilleRepository.java
com.cinema.entities
Categorie.java
Cinema.java
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
1 package com.cinema.entities;
3 import java.util.Collection;
22 @Entity
23 @Data @NoArgsConstructor @AllArgsConstructor @ToString
24 public class Salle {
25 @Id @GeneratedValue(strategy=GenerationType.IDENTITY)
26 private Long id;
27 @Column(length = 75)
28 private String name;
29 private int nombrePlace;
30 @ManyToOne @JoinColumn(name="id_cinema")
31 @JsonProperty(access = Access.WRITE_ONLY)
32 private Cinema cinema;
33 @OneToMany(mappedBy="salle")
34 @JsonProperty(access = Access.WRITE_ONLY)
35 private Collection<Place> places;
36 @OneToMany(mappedBy="salle")
37 @JsonProperty(access = Access.WRITE_ONLY)
38 private Collection<Projection> projections;
39 }
Writable
Smart Insert
1 : 1 : 0


TP_CLASSE - cinema_project/src/main/java/com/cinema/entities/Ticket.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
Task List
Outline
cinema_project [boot] [devtools]
src/main/java
com.cinema
CinemaProjectApplication.java
com.cinema.dao
CategorieRepository.java
CinemaRepository.java
FilmRepository.java
PlaceRepository.java
ProjectionRepository.java
SalleRepository.java
SeanceRepository.java
TicketRepository.java
VilleRepository.java
com.cinema.entities
Categorie.java
Cinema.java
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
package com.cinema.entities;
import javax.persistence.Column;
@Entity
@Data @NoArgsConstructor @AllArgsConstructor @ToString
public class Ticket {
@Id@GeneratedValue(strategy=GenerationType.IDENTITY)
private Long id;
@Column(length = 75)
private String nomClient;
private double prix;
@Column(unique=false)
private Integer codePayement;
private boolean reserve;
@ManyToOne
private Place place;
@ManyToOne
private Projection projection;
}
Writable
Smart Insert
1 : 1 : 0


TP_CLASSE - cinema_project/src/main/java/com/cinema/entities/Ville.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
TicketProj.java
Ville.java
Task List
Outline
cinema_project [boot] [devtools]
src/main/java
com.cinema
CinemaProjectApplication.java
com.cinema.dao
CategorieRepository.java
CinemaRepository.java
FilmRepository.java
PlaceRepository.java
ProjectionRepository.java
SalleRepository.java
SeanceRepository.java
TicketRepository.java
VilleRepository.java
com.cinema.entities
Categorie.java
Cinema.java
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
package com.cinema.entities;
import java.util.Collection;
@Entity
@Data @NoArgsConstructor @AllArgsConstructor @ToString
public class Ville {
@Id @GeneratedValue(strategy=GenerationType.IDENTITY)
private Long id;
@Column(length = 75)
@NonNull
private String name;
private double longitude, latidude, altitude;
@OneToMany(mappedBy="ville")
private Collection<Cinema> cinemas;
}
Writable
Smart Insert
1 : 1 : 0

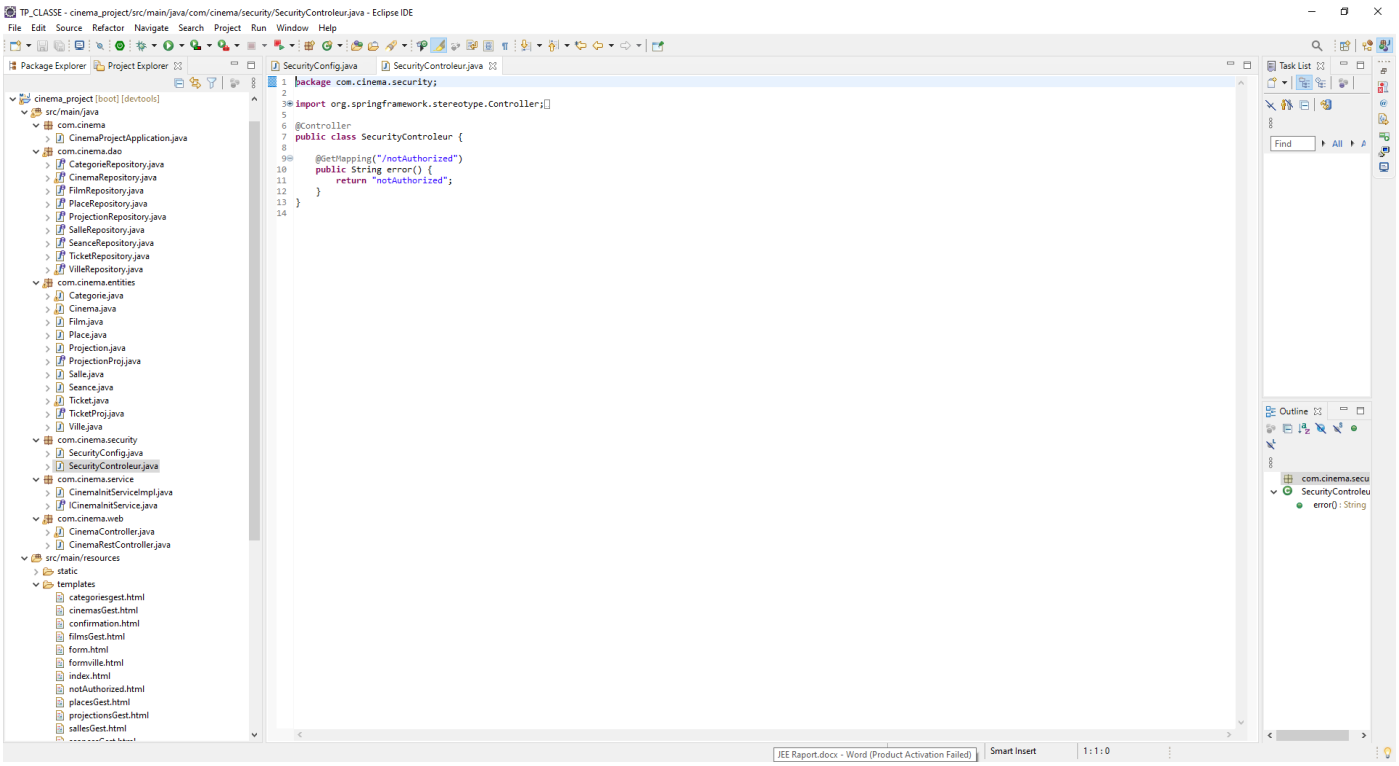

TP_CLASSE - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Package Explorer
Project Explorer
cinema_project [boot] [devtools]
src/main/java
com.cinema
CinemaProjectApplication.java
com.cinema.dao
CategorieRepository.java
CinemaRepository.java
FilmRepository.java
PlaceRepository.java
ProjectionRepository.java
SalleRepository.java
SeanceRepository.java
TicketRepository.java
VilleRepository.java
com.cinema.entities
Categorie.java
Cinema.java
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
Task List
Find
All
Outline
There is no active editor that provides an outline.

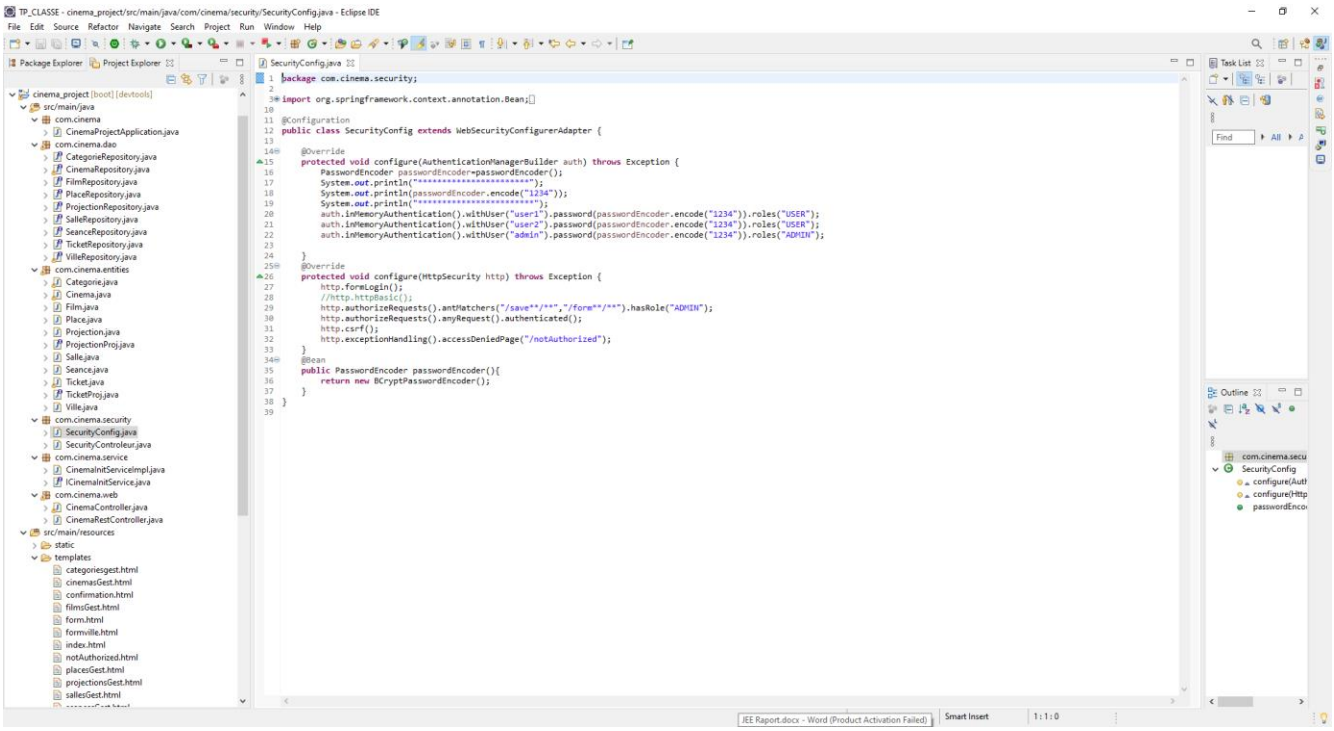

TP_CLASSE - cinema_project/src/main/java/com/cinema/CinemaProjectApplication.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Project Explorer
CinemaProjectApplication.java
1 package com.cinema;
2
3 import org.springframework.beans.factory.annotation.Autowired;
13
14 @SpringBootApplication
15 public class CinemaProjectApplication implements CommandLineRunner {
16 @Autowired
17 private RepositoryRestConfiguration restConfiguration;
18 @Autowired
19 private ICinemaInitService cinemaInitService;
20
21 public static void main(String[] args) {
22
23 SpringApplication.run(CinemaProjectApplication.class, args);
24 }
25
26 @Override
27 public void run(String... args) throws Exception {
28 restConfiguration.exposeIdsFor(Film.class, Salle.class, Ticket.class);
29 cinemaInitService.initVilles();
30 cinemaInitService.initCinemas();
31 cinemaInitService.initSalles();
32 cinemaInitService.initPlaces();
33 cinemaInitService.initSeances();
34 cinemaInitService.initCategories();
35 cinemaInitService.initFilms();
36 cinemaInitService.initProjections();
37 cinemaInitService.initTickets();
38 }
39
40 }
41
Task List
Outline
com.cinema
CinemaProjectAp
restConfigurati
cinemaInitServ
main(String[])
run(String...)
Writable
Smart Insert
26 : 14 : 823

- Security

Dans cette partie on est concentre sur spring security qui nous donne la possibilite de cree et gerer les compte pour l'application web .il nous offre une page d'authentification qu'il est securise par spring security comme ont peux modifier l'etat d'authentification et la page themelete.

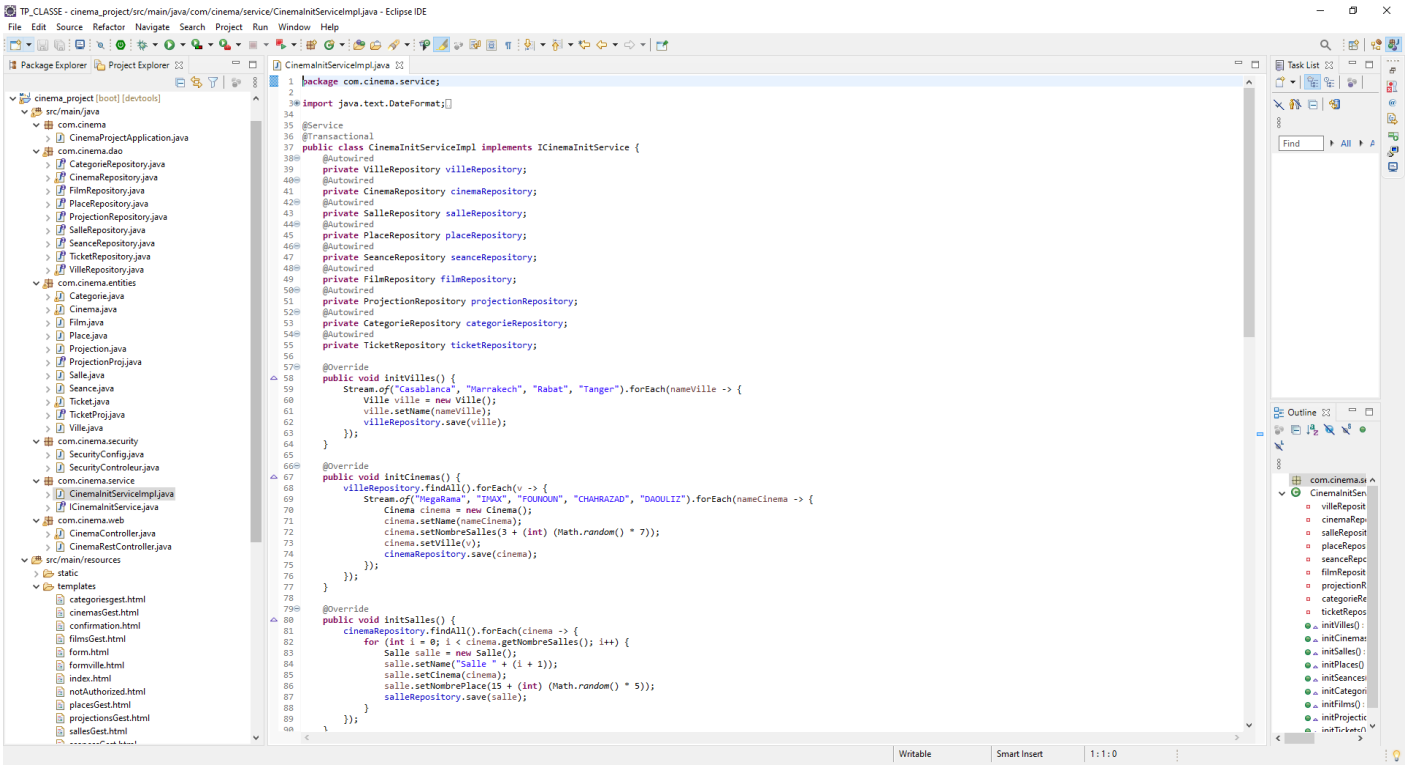
Voici quelque screen shoot :

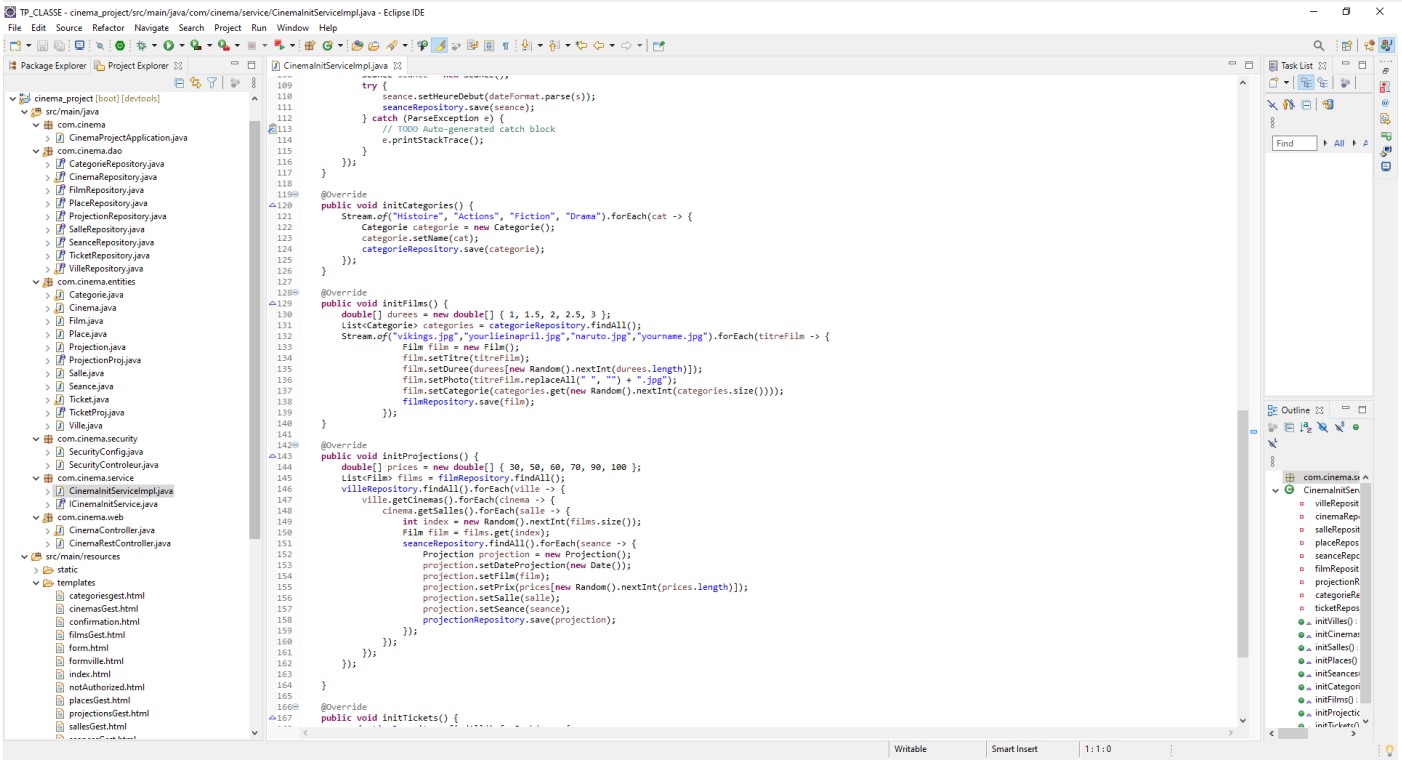




- Service

Voici quelque screen shoot pour la partie service :

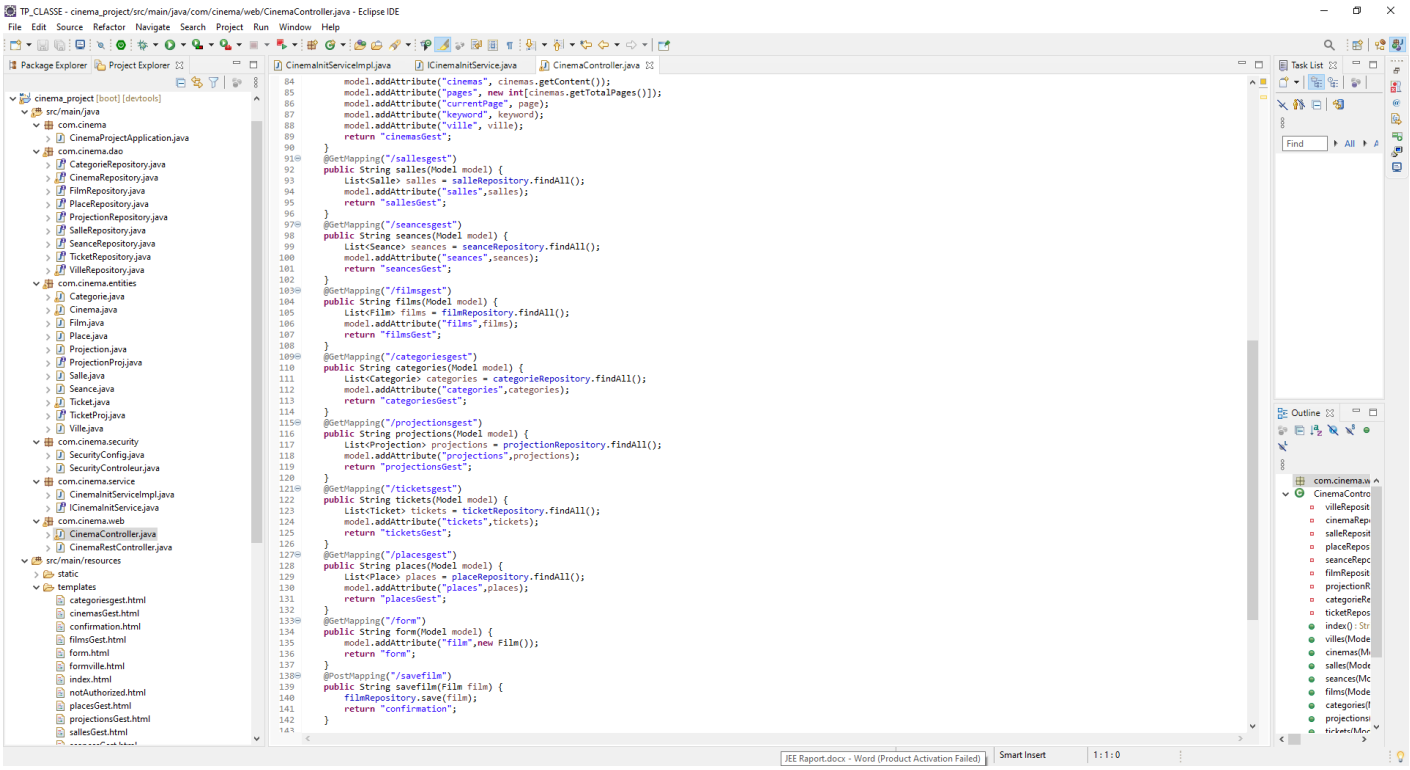


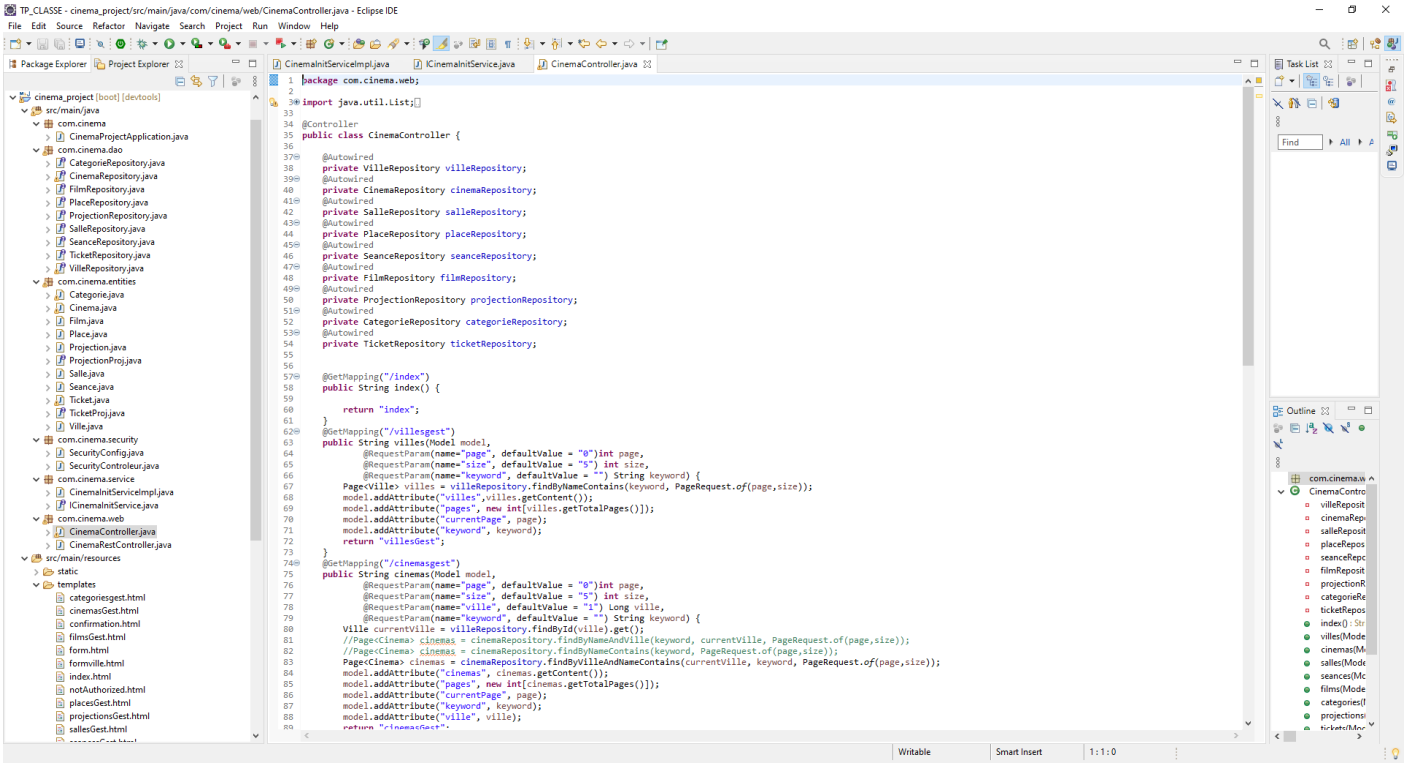


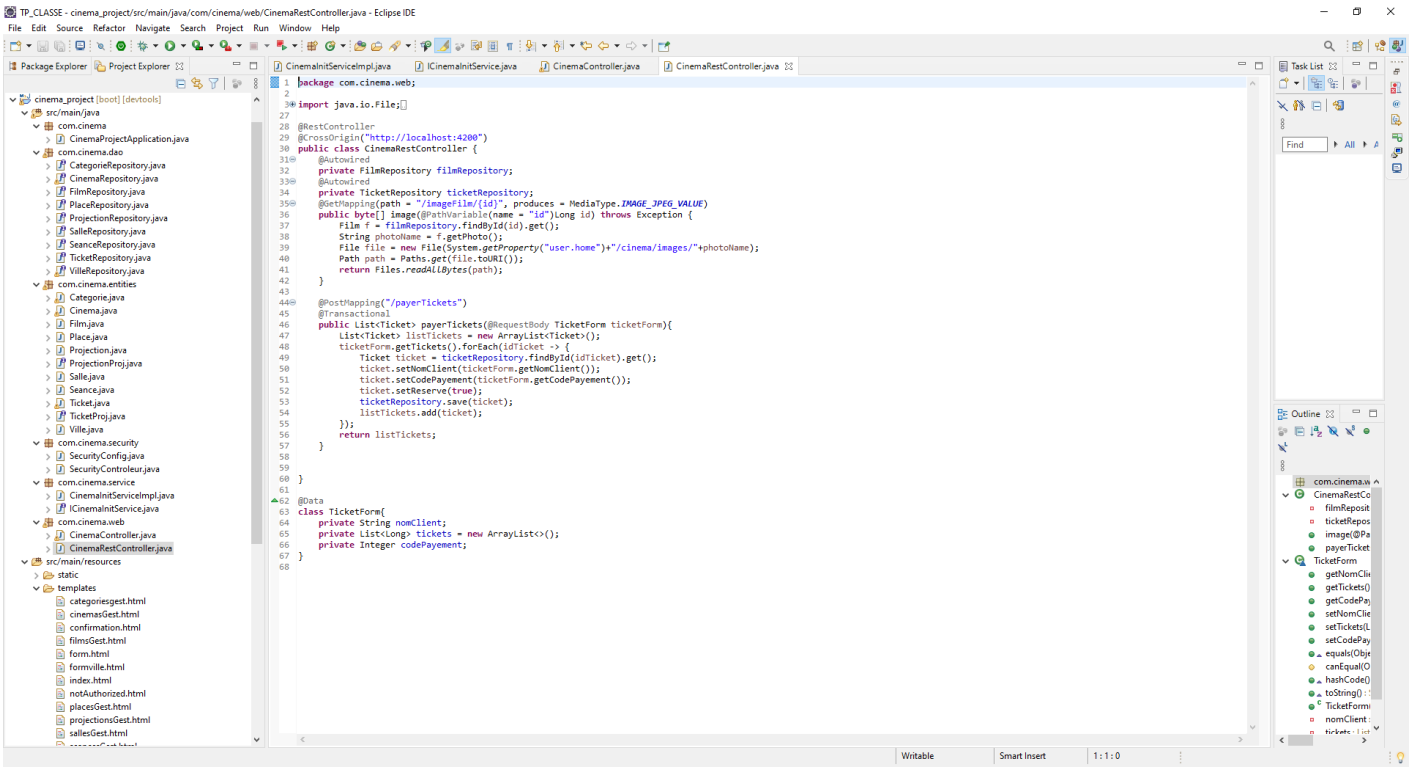
- Web

La partie WEB c’est la partie ou on trouvent le controleur qui nous aide a gerer les class les les method

Voici quelque screen shoot :





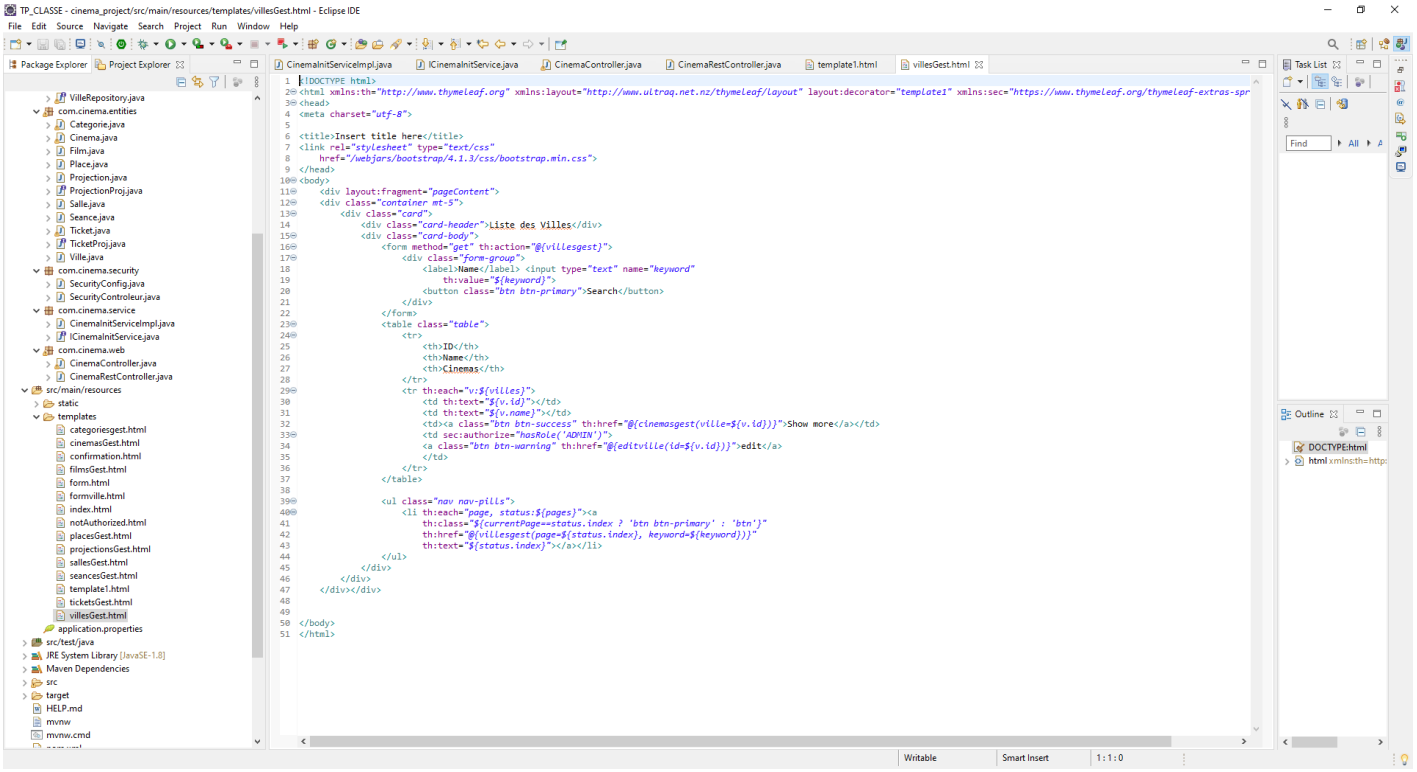



```java
package com.cinema.web;

import java.io.File;

@RestController
@CrossOrigin("http://localhost:4200")
public class CinemaRestController {
    @Autowired
    private FilmRepository filmRepository;
    @Autowired
    private TicketRepository ticketRepository;
    @GetMapping(path = "/imageFilm/{id}", produces = MediaType.IMAGE_JPEG_VALUE)
    public byte[] image(@PathVariable(name = "id")Long id) throws Exception {
        Film f = filmRepository.findById(id).get();
        String photoName = f.getPhoto();
        File file = new File(System.getProperty("user.home")+"/cinema/images/"+photoName);
        Path path = Paths.get(file.toURI());
        return Files.readAllBytes(path);
    }

    @PostMapping("/payerTickets")
    @Transactional
    public List<Ticket> payerTickets(@RequestBody TicketForm ticketForm){
        List<Ticket> listTickets = new ArrayList<Ticket>();
        ticketForm.getTickets().forEach(idTicket -> {
            Ticket ticket = ticketRepository.findById(idTicket).get();
            ticket.setNomClient(ticketForm.getNomClient());
            ticket.setCodePayement(ticketForm.getCodePayement());
            ticket.setReserve(true);
            ticketRepository.save(ticket);
            listTickets.add(ticket);
        });
        return listTickets;
    }

}

@Data
class TicketForm{
    private String nomClient;
    private List<Long> tickets = new ArrayList<>();
    private Integer codePayement;
}
```

- Thymeleaf

Thymeleaf est un moteur de modèle Java côté serveur moderne pour les environnements Web et autonomes.L'objectif principal de Thymeleaf est d'apporter des modèles naturels élégants à votre flux de travail de développement - du HTML qui peut être correctement affiché dans les navigateurs et également fonctionner comme des prototypes statiques, permettant une collaboration plus étroite dans les équipes de développement.Avec des modules pour Spring Framework, une multitude d'intégrations avec vos outils préférés et la possibilité de brancher vos propres fonctionnalités, Thymeleaf est idéal pour le développement Web JVM HTML5 moderne - bien qu'il puisse faire beaucoup plus.

Voici quelque exemple :


TP_CLASSE - cinema_project/src/main/resources/templates/villesGest.html - Eclipse IDE
File Edit Source Navigate Search Project Run Window Help
Package Explorer
Project Explorer
VilleRepository.java
com.cinema.entities
Categorie.java
Cinema.java
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
CinemaController.java
CinemaRestController.java
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
ticketsGest.html
villesGest.html
application.properties
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
CinemaInitServiceImpl.java
ICinemaInitService.java
CinemaController.java
CinemaRestController.java
template1.html
villesGest.html
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org" xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout" layout:decorator="template1" xmlns:sec="https://www.thymeleaf.org/thymeleaf-extras-spr
<head>
<meta charset="utf-8">
<title>Insert title here</title>
<link rel="stylesheet" type="text/css"
href="/webjars/bootstrap/4.1.3/css/bootstrap.min.css">
</head>
<body>

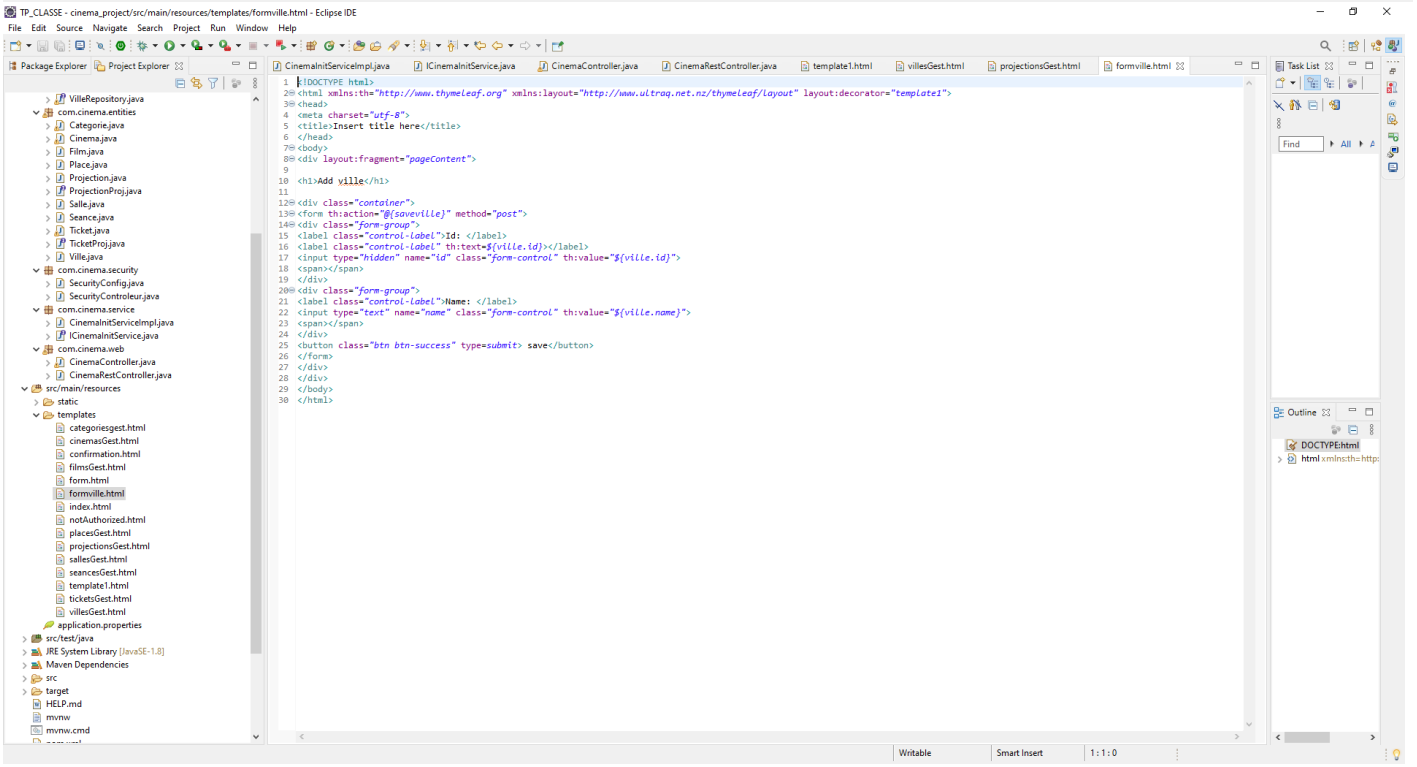
Liste des Villes

<form method="get" th:action="@{villesgest}">

<label>Name</label> <input type="text" name="keyword"
th:value="${keyword}">
<button class="btn btn-primary">Search</button>

</form>
<table class="table">
<tr>
<th>ID</th>
<th>Name</th>
<th>Cinemas</th>
</tr>
<tr th:each="v:${villes}">
<td th:text="${v.id}"></td>
<td th:text="${v.name}"></td>
<td>Show more</td>
<td sec:authorize="hasRole('ADMIN')">
edit
</td>
</tr>
</table>
<ul class="nav nav-pills">
<li th:each="page, status:${pages}"><a
th:class="${currentPage==status.index ? 'btn btn-primary' : 'btn'}"
th:href="@{villesgest(page=${status.index}, keyword=${keyword})}"
th:text="${status.index}">

</body>
</html>
Task List
Outline
DOCTYPE:html
html xmlns:th=http:
Writable
Smart Insert
1 : 1 : 0


TP_CLASSE - cinema_project/src/main/resources/templates/formville.html - Eclipse IDE
File Edit Source Navigate Search Project Run Window Help
Package Explorer
Project Explorer
VilleRepository.java
com.cinema.entities
Categorie.java
Cinema.java
Film.java
Place.java
Projection.java
ProjectionProj.java
Salle.java
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
CinemaController.java
CinemaRestController.java
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
ticketsGest.html
villesGest.html
application.properties
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
CinemaInitServiceImpl.java
ICinemaInitService.java
CinemaController.java
CinemaRestController.java
template1.html
villesGest.html
projectionsGest.html
formville.html
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org" xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout" layout:decorator="template1">
<head>
<meta charset="utf-8">
<title>Insert title here</title>
</head>
<body>

<h1>Add ville</h1>

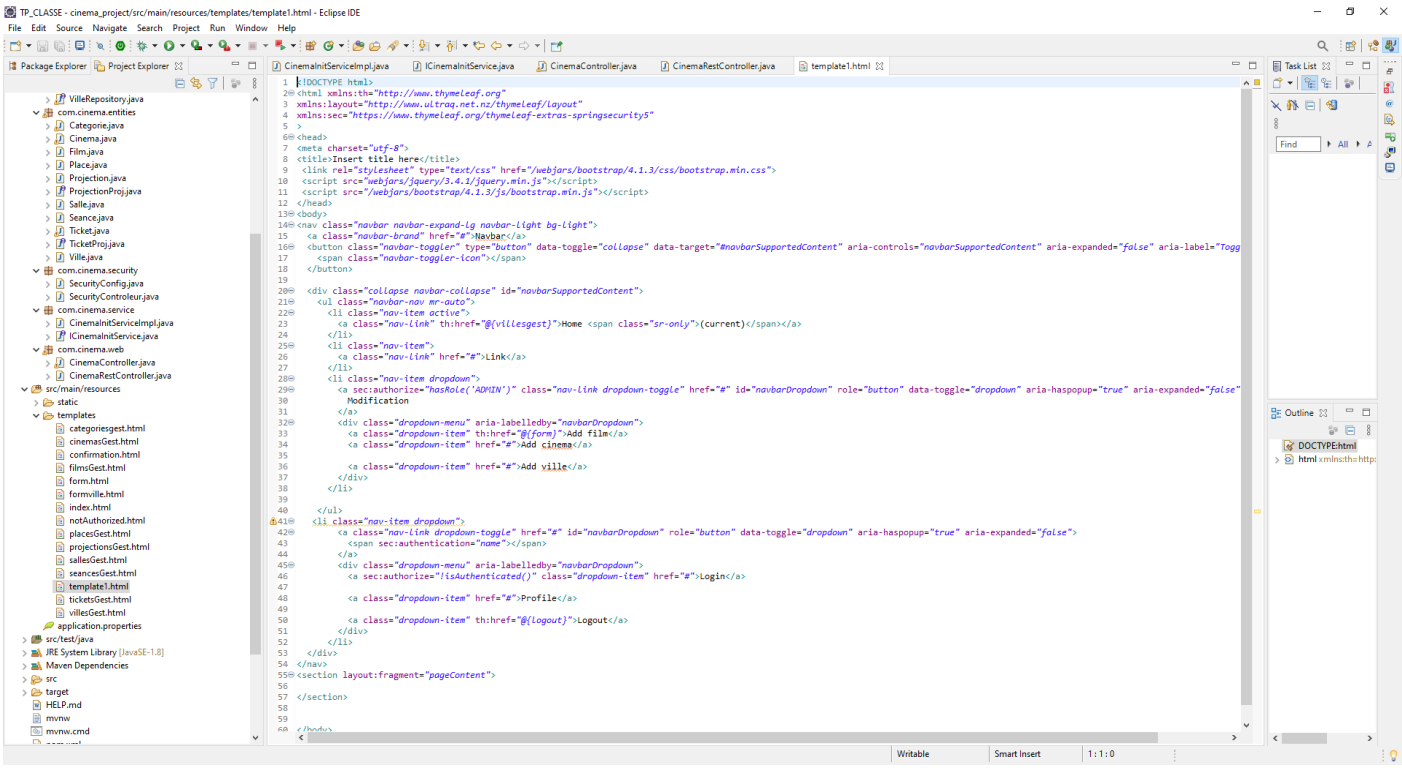
<form th:action="@{saveville}" method="post">

<label class="control-label">Id: </label>
<label class="control-label" th:text=${vville.id}></label>
<input type="hidden" name="id" class="form-control" th:value="${ville.id}">

<label class="control-label">Name: </label>
<input type="text" name="name" class="form-control" th:value="${ville.name}">

<button class="btn btn-success" type=submit> save</button>
</form>

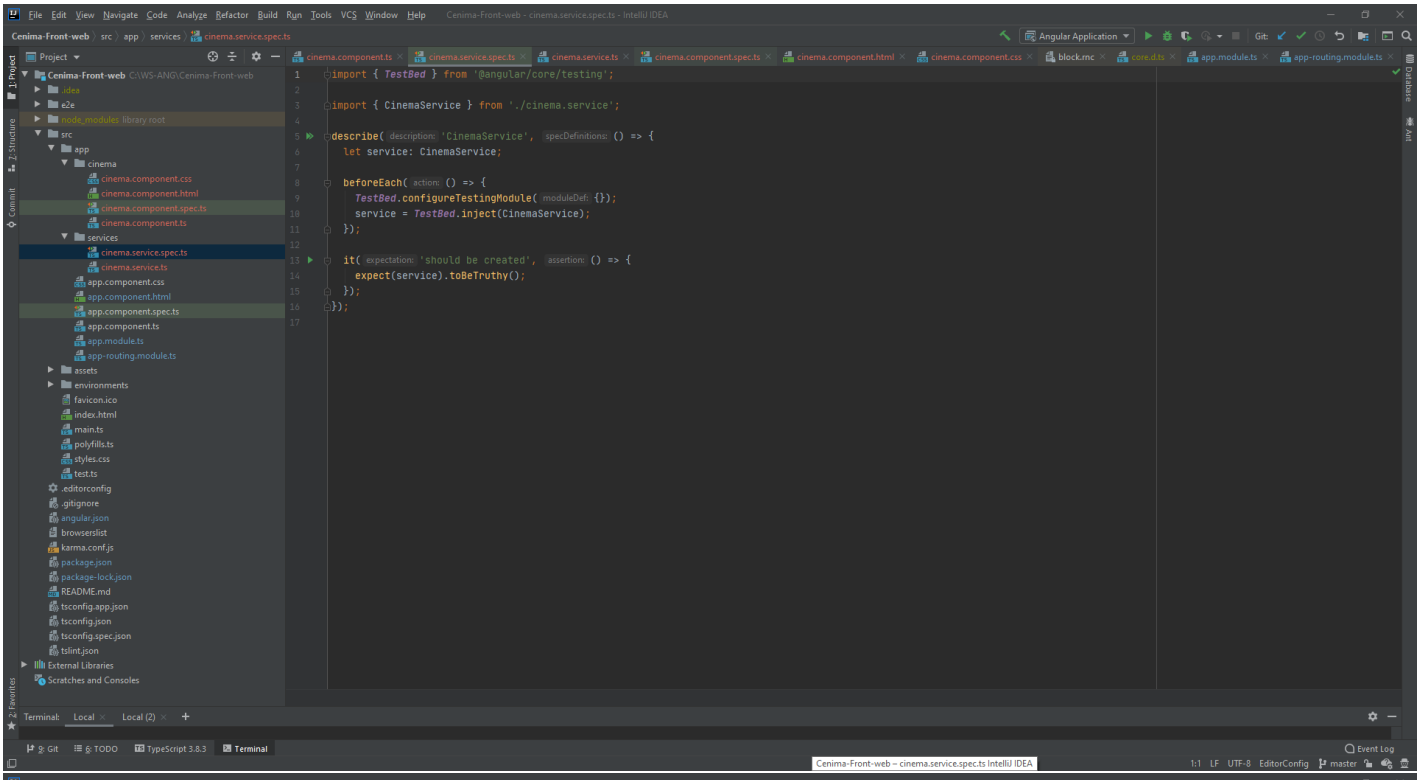
</body>
</html>
Task List
Outline
DOCTYPE:html
html xmlns:th=http:
Writable
Smart Insert
1 : 1 : 0

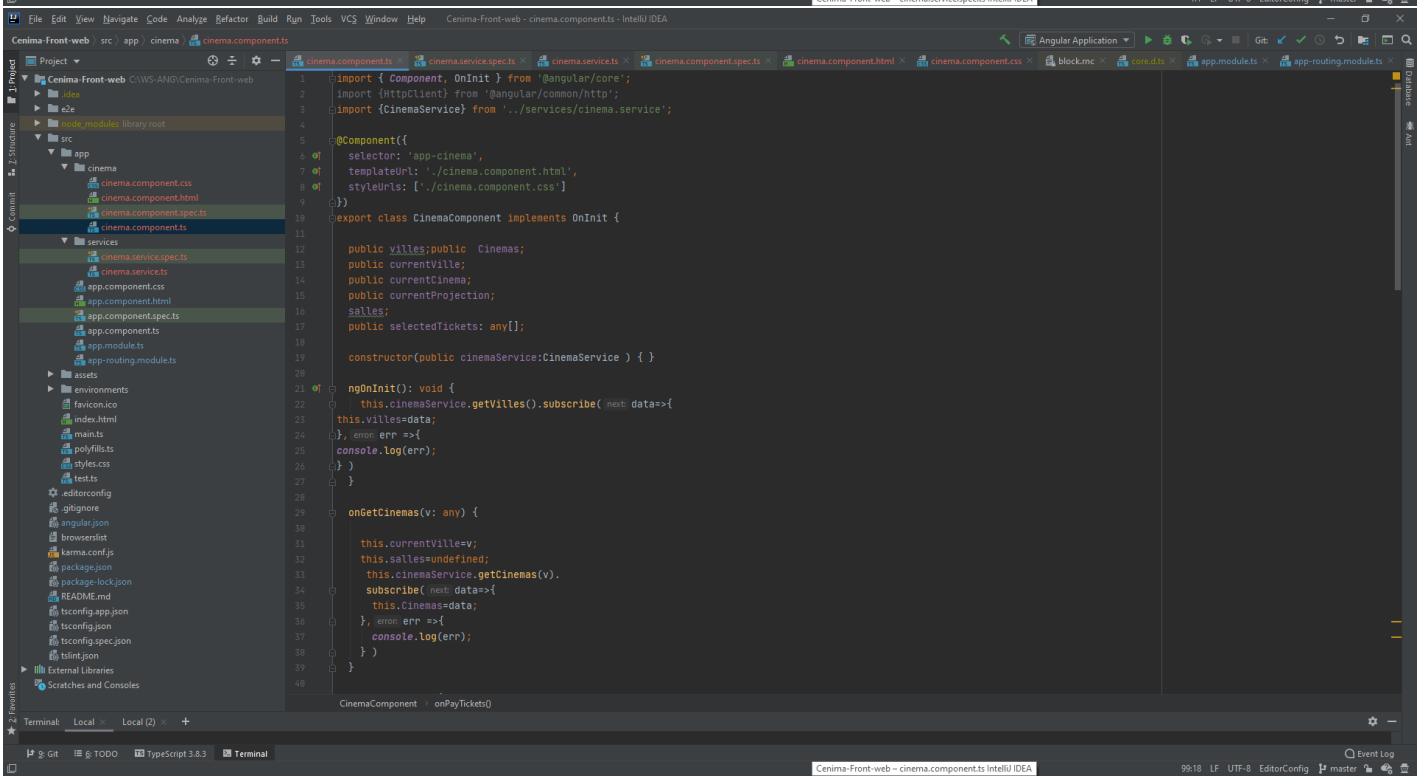


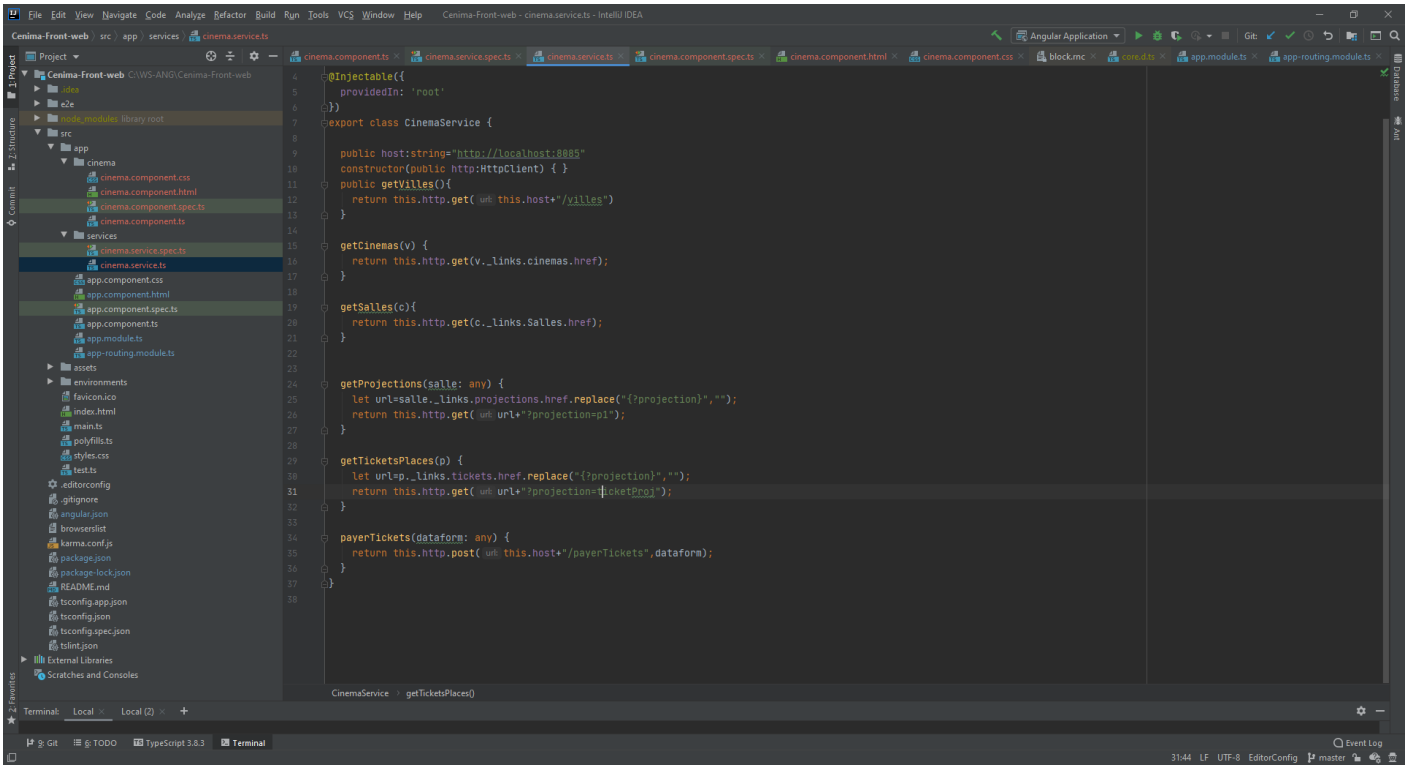
- Angular

Angular (communément appelé "Angular 2+" ou "Angular v2 et plus")[2,3] est un cadriciel (framework) côté client, open source, basé sur TypeScript, et co-dirigé par l'équipe du projet « Angular » à Google et par une communauté de particuliers et de sociétés. Angular est une réécriture complète de AngularJS, cadriciel construit par la même équipe.on la utilizer pour la partie front end.

Voicie quelque screen shoot :


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Cenima-Front-web - cinema.service.spec.ts - IntelliJ IDEA
Cenima-Front-web › src › app › services › cinema.service.spec.ts
Angular Application
Git:
Project
Cenima-Front-web C:\WS-ANG\Cenima-Front-web
.idea
e2e
node_modules library root
src
app
cinema
cinema.component.css
cinema.component.html
cinema.component.spec.ts
cinema.component.ts
services
cinema.service.spec.ts
cinema.service.ts
app.component.css
app.component.html
app.component.spec.ts
app.component.ts
app.module.ts
app-routing.module.ts
assets
environments
favicon.ico
index.html
main.ts
polyfills.ts
styles.css
test.ts
.editorconfig
.gitignore
angular.json
browserslist
karma.conf.js
package.json
package-lock.json
README.md
tsconfig.app.json
tsconfig.json
tsconfig.spec.json
tslint.json
External Libraries
Scratches and Consoles
cinema.component.ts
cinema.service.spec.ts
cinema.service.ts
cinema.component.spec.ts
cinema.component.html
cinema.component.css
block.rnc
core.d.ts
app.module.ts
app-routing.module.ts
import { TestBed } from '@angular/core/testing';
import { CinemaService } from './cinema.service';
describe(description: 'CinemaService', specDefinitions: () => {
let service: CinemaService;
beforeEach(action: () => {
TestBed.configureTestingModule(moduleDef: {});
service = TestBed.inject(CinemaService);
});
it(expectation: 'should be created', assertion: () => {
expect(service).toBeTruthy();
});
});
Terminal: Local Local (2)
Git TODO TypeScript 3.8.3 Terminal
Event Log
Cenima-Front-web – cinema.service.spec.ts IntelliJ IDEA
1:1 LF UTF-8 EditorConfig master

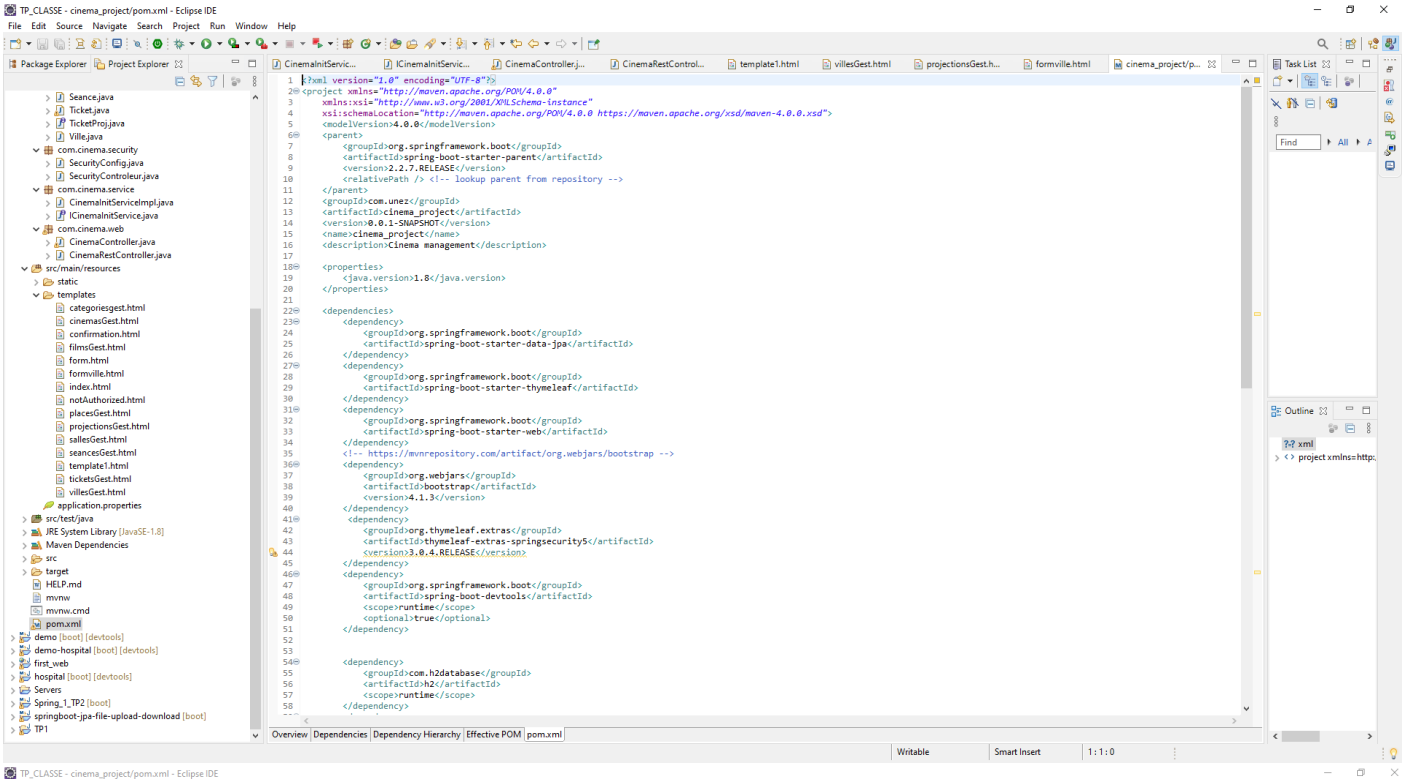

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Cenima-Front-web - cinema.component.ts - IntelliJ IDEA
Cenima-Front-web › src › app › cinema › cinema.component.ts
Angular Application
Git:
Project
Cenima-Front-web C:\WS-ANG\Cenima-Front-web
.idea
e2e
node_modules library root
src
app
cinema
cinema.component.css
cinema.component.html
cinema.component.spec.ts
cinema.component.ts
services
cinema.service.spec.ts
cinema.service.ts
app.component.css
app.component.html
app.component.spec.ts
app.component.ts
app.module.ts
app-routing.module.ts
assets
environments
favicon.ico
index.html
main.ts
polyfills.ts
styles.css
test.ts
.editorconfig
.gitignore
angular.json
browserslist
karma.conf.js
package.json
package-lock.json
README.md
tsconfig.app.json
tsconfig.json
tsconfig.spec.json
tslint.json
External Libraries
Scratches and Consoles
cinema.component.ts
cinema.service.spec.ts
cinema.service.ts
cinema.component.spec.ts
cinema.component.html
cinema.component.css
block.rnc
core.d.ts
app.module.ts
app-routing.module.ts
import { Component, OnInit } from '@angular/core';
import {HttpClient} from '@angular/common/http';
import {CinemaService} from '../services/cinema.service';
@Component({
selector: 'app-cinema',
templateUrl: './cinema.component.html',
styleUrls: ['./cinema.component.css']
})
export class CinemaComponent implements OnInit {
public villes;public Cinemas;
public currentVille;
public currentCinema;
public currentProjection;
salles;
public selectedTickets: any[];
constructor(public cinemaService:CinemaService) { }
ngOnInit(): void {
this.cinemaService.getVilles().subscribe(next: data=>{
this.villes=data;
}, error: err =>{
console.log(err);
})
}
onGetCinemas(v: any) {
this.currentVille=v;
this.salles=undefined;
this.cinemaService.getCinemas(v).
subscribe(next: data=>{
this.Cinemas=data;
}, error: err =>{
console.log(err);
})
}
CinemaComponent › onPayTickets()
Terminal: Local Local (2)
Git TODO TypeScript 3.8.3 Terminal
Event Log
Cenima-Front-web – cinema.component.ts IntelliJ IDEA
99:18 LF UTF-8 EditorConfig master

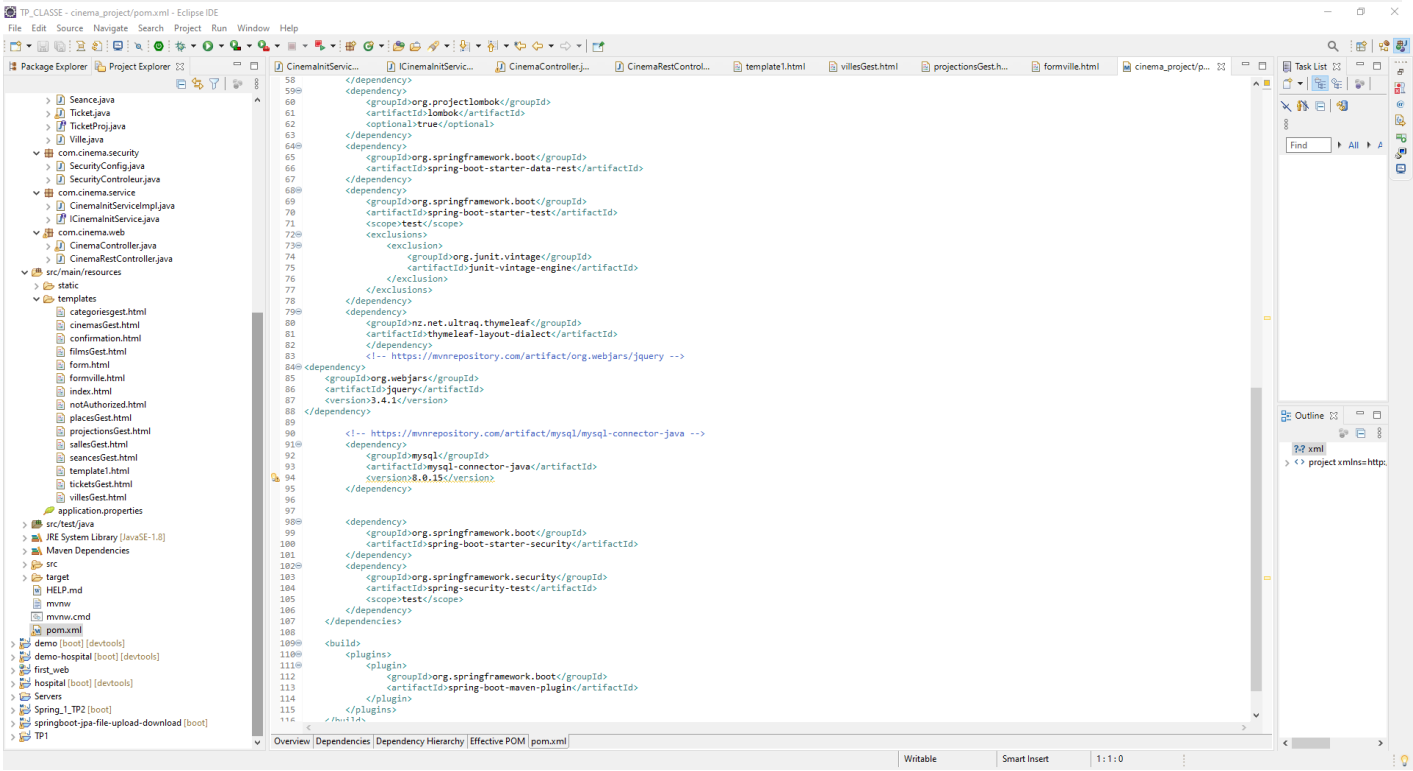


Pom.xml

Un modèle d'objet de projet ou POM est l'unité de travail fondamentale de Maven. Il s'agit d'un fichier XML qui contient des informations sur le projet et les détails de configuration utilisés par Maven pour générer le projet. Il contient des valeurs par défaut pour la plupart des projets. Par exemple, le répertoire de construction, qui est la cible; le répertoire source, qui est src / main / java; le répertoire source de test, qui est src / test / java; etc. Lors de l'exécution d'une tâche ou d'un objectif, Maven recherche le POM dans le répertoire actuel. Il lit le POM, obtient les informations de configuration nécessaires, puis exécute l'objectif.

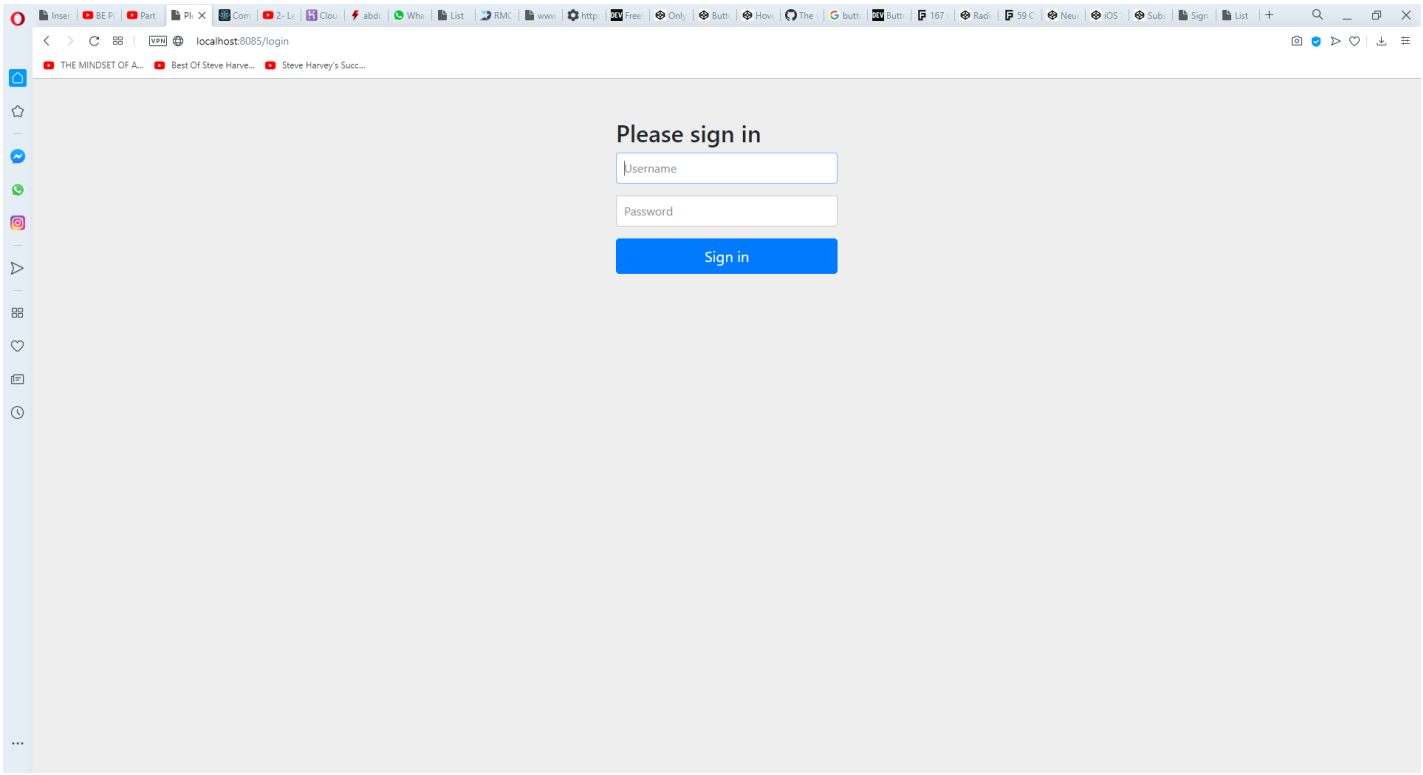
Voici quelque screen shoot ce fichier:

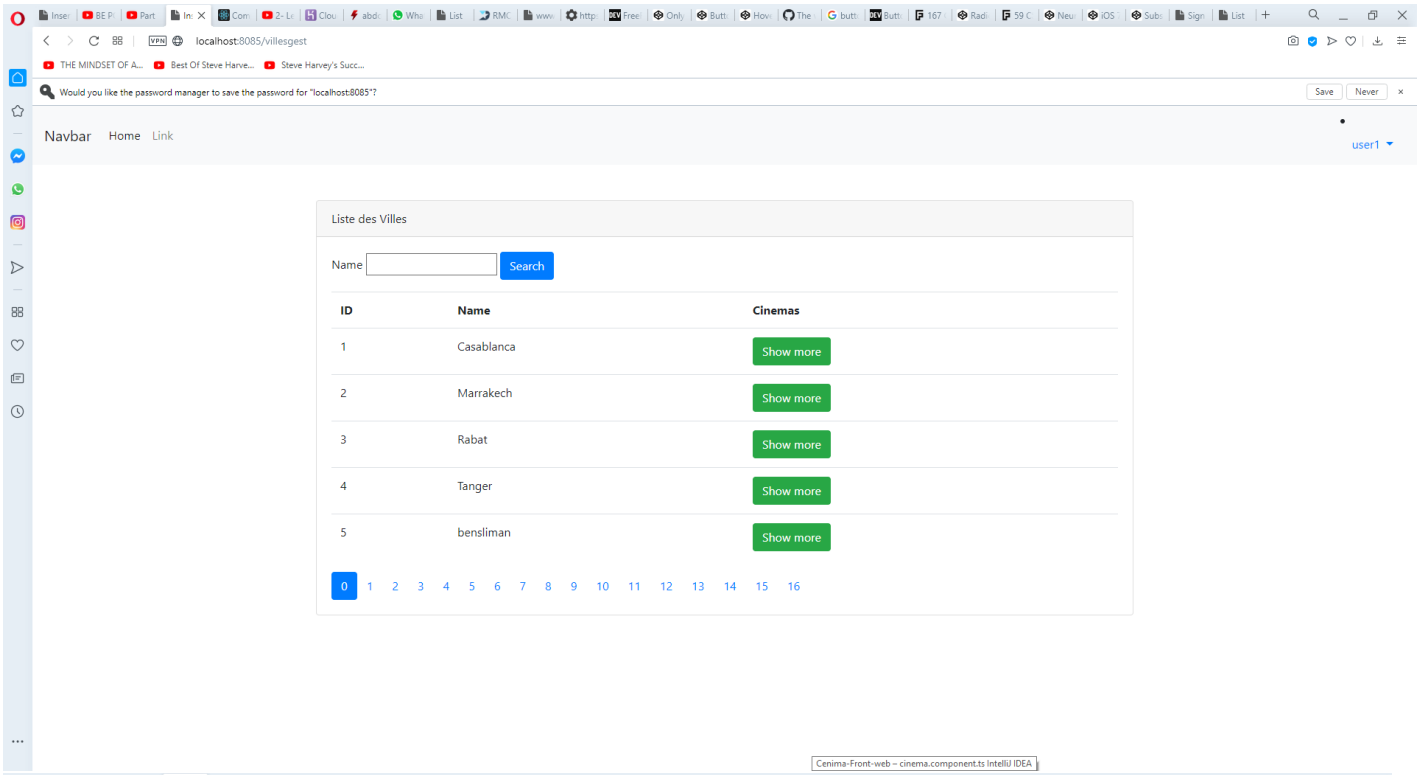

TP_CLASSE - cinema_project/pom.xml - Eclipse IDE
File Edit Source Navigate Search Project Run Window Help
Package Explorer
Project Explorer
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
CinemaController.java
CinemaRestController.java
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
ticketsGest.html
villesGest.html
application.properties
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
demo [boot] [devtools]
demo-hospital [boot] [devtools]
first_web
hospital [boot] [devtools]
Servers
Spring_1_TP2 [boot]
springboot-jpa-file-upload-download [boot]
TP1
CinemaInitServic...
ICinemaInitServic...
CinemaController.j...
CinemaRestControl...
template1.html
villesGest.html
projectionsGest.h...
formville.html
cinema_project/p...
Task List
Find
All
Outline
project xmlns=http:
Overview
Dependencies
Dependency Hierarchy
Effective POM
pom.xml
Writable
Smart Insert
1 : 1 : 0

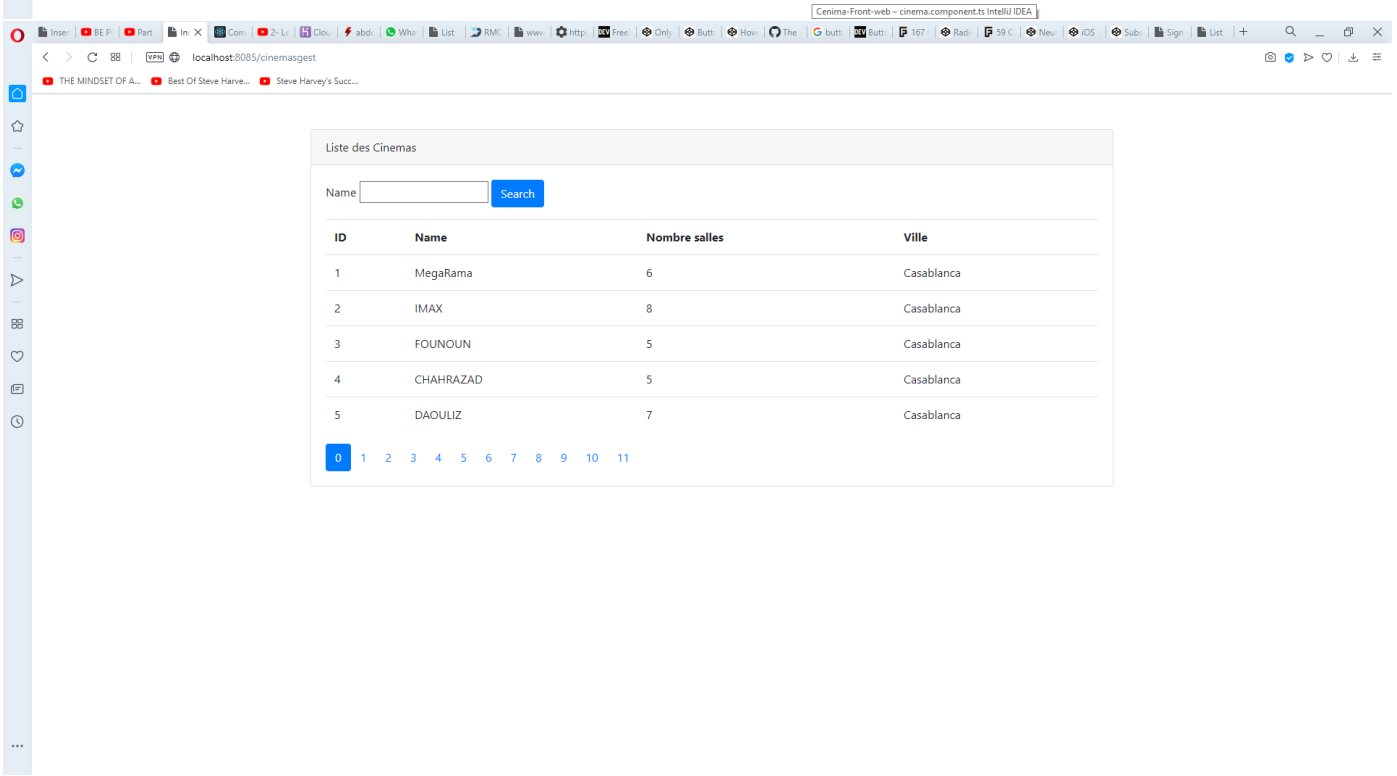

TP_CLASSE - cinema_project/pom.xml - Eclipse IDE
File Edit Source Navigate Search Project Run Window Help
Package Explorer
Project Explorer
Seance.java
Ticket.java
TicketProj.java
Ville.java
com.cinema.security
SecurityConfig.java
SecurityControleur.java
com.cinema.service
CinemaInitServiceImpl.java
ICinemaInitService.java
com.cinema.web
CinemaController.java
CinemaRestController.java
src/main/resources
static
templates
categoriesgest.html
cinemasGest.html
confirmation.html
filmsGest.html
form.html
formville.html
index.html
notAuthorized.html
placesGest.html
projectionsGest.html
sallesGest.html
seancesGest.html
template1.html
ticketsGest.html
villesGest.html
application.properties
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
demo [boot] [devtools]
demo-hospital [boot] [devtools]
first_web
hospital [boot] [devtools]
Servers
Spring_1_TP2 [boot]
springboot-jpa-file-upload-download [boot]
TP1
CinemaInitServic...
ICinemaInitServic...
CinemaController.j...
CinemaRestControl...
template1.html
villesGest.html
projectionsGest.h...
formville.html
cinema_project/p...
Task List
Find
All
Outline
project xmlns=http:
Overview
Dependencies
Dependency Hierarchy
Effective POM
pom.xml
Writable
Smart Insert
1 : 1 : 0

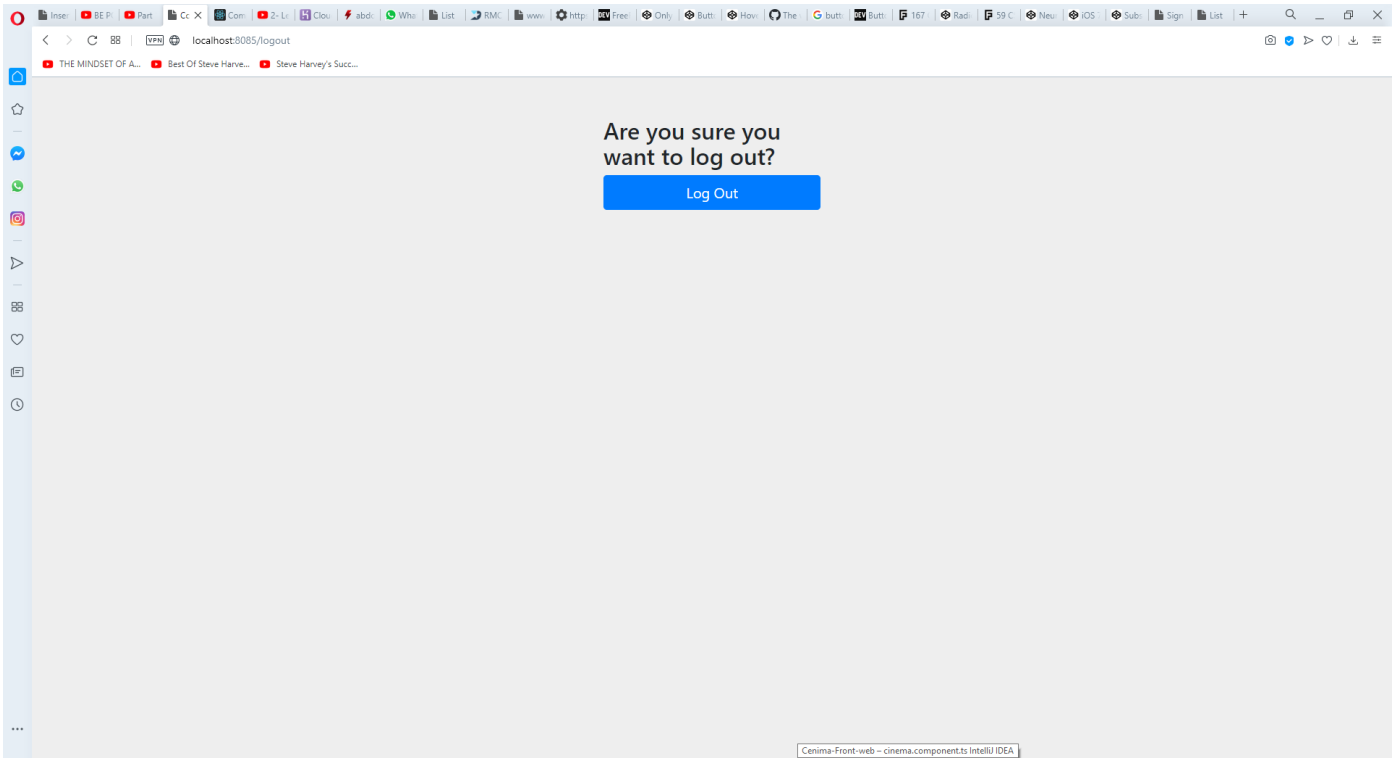
Screen de site web:

- User

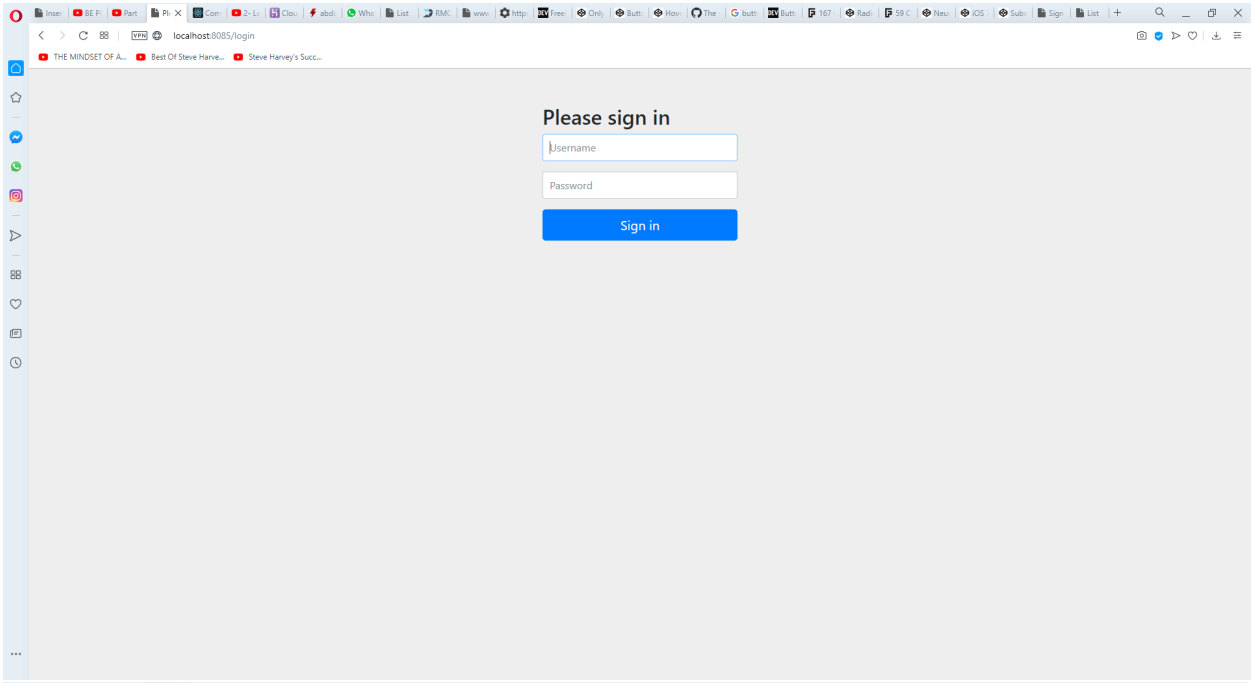


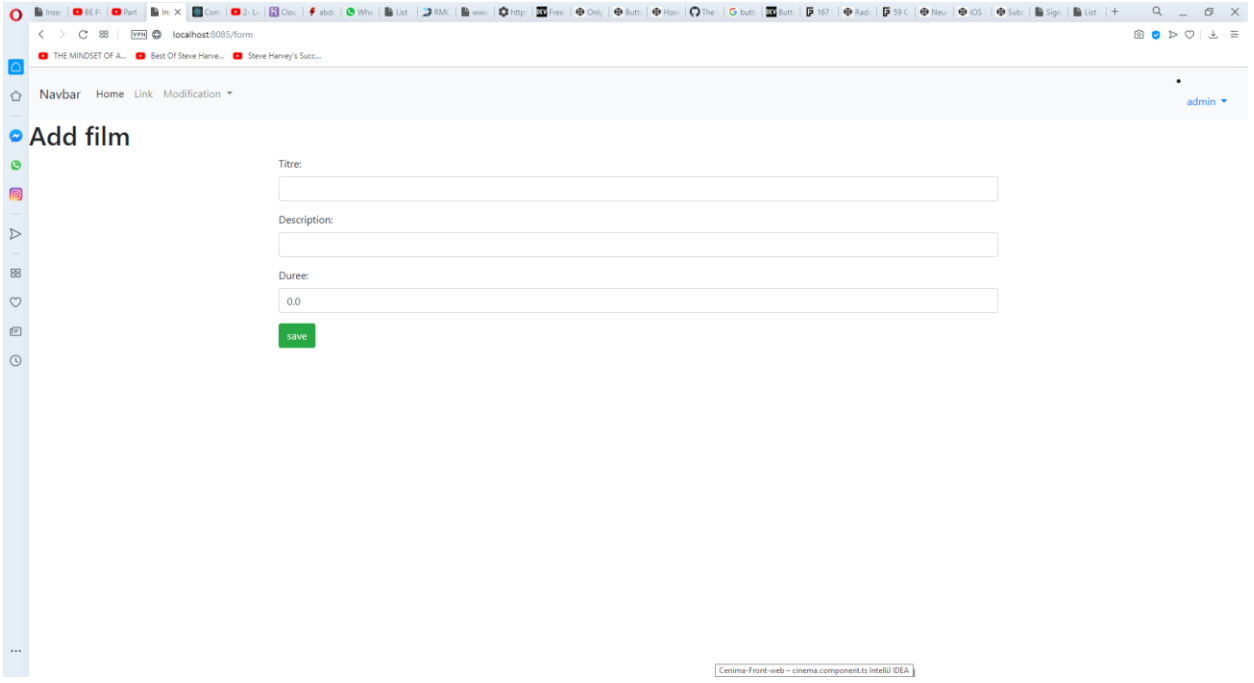

localhost:8085/villesgest
Would you like the password manager to save the password for "localhost:8085"?
Save
Never
Navbar
Home
Link
user1
Liste des Villes
Name
Search
ID
Name
Cinemas
1
Casablanca
Show more
2
Marrakech
Show more
3
Rabat
Show more
4
Tanger
Show more
5
bensliman
Show more
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

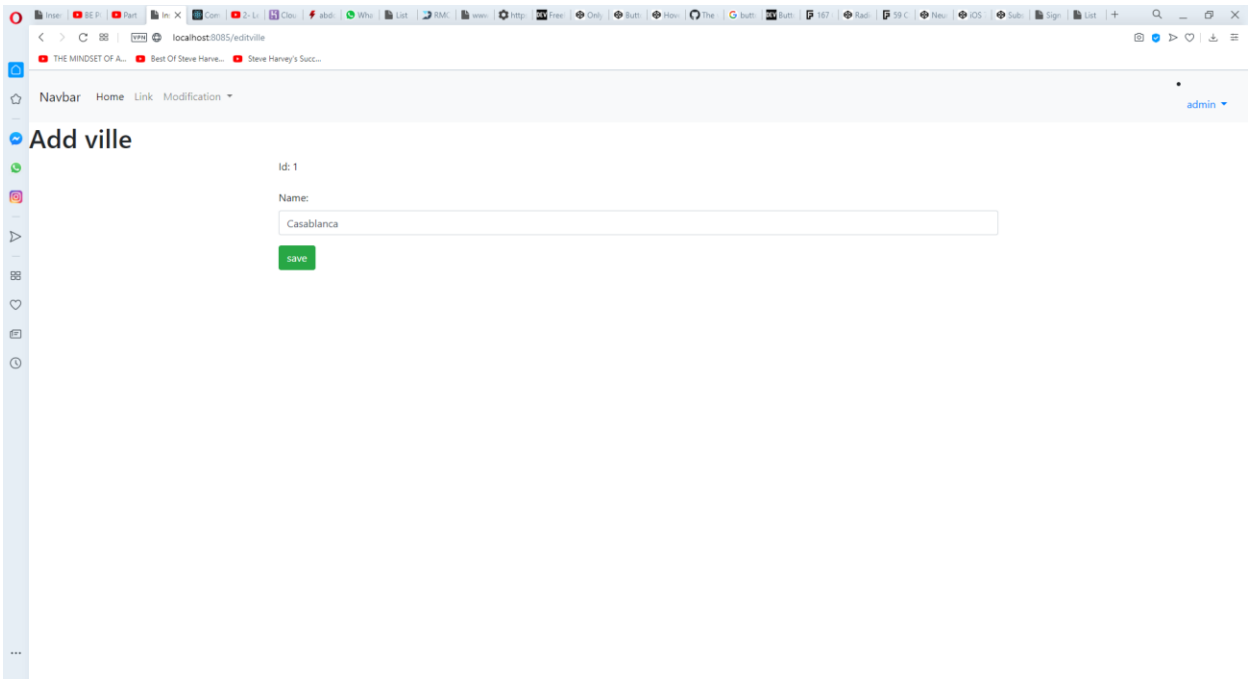

Cenima-Front-web – cinema.component.ts IntelliJ IDEA
localhost:8085/cinemasgest
Liste des Cinemas
Name
Search
ID
Name
Nombre salles
Ville
1
MegaRama
6
Casablanca
2
IMAX
8
Casablanca
3
FOUNOUN
5
Casablanca
4
CHAHRAZAD
5
Casablanca
5
DAOULIZ
7
Casablanca
0 1 2 3 4 5 6 7 8 9 10 11

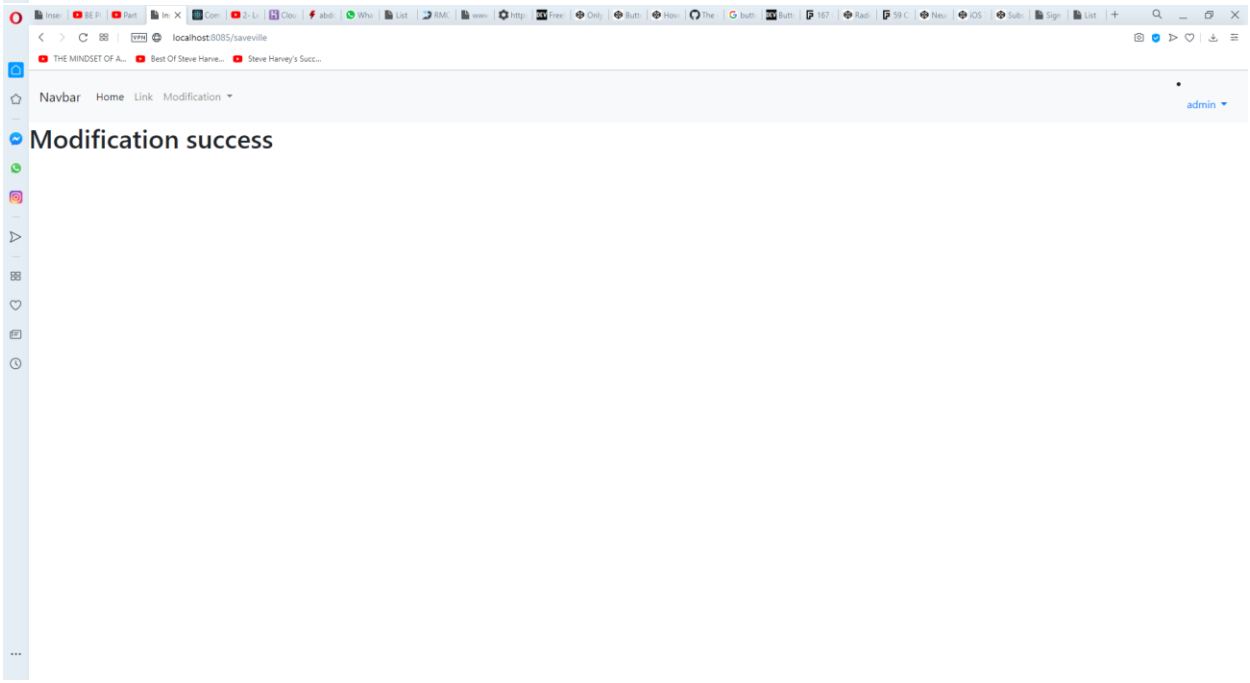

localhost:8085/logout
THE MINDSET OF A...
Best Of Steve Harve...
Steve Harvey's Succ...
Are you sure you want to log out?
Log Out
Cenima-Front-web – cinema.component.ts IntelliJ IDEA

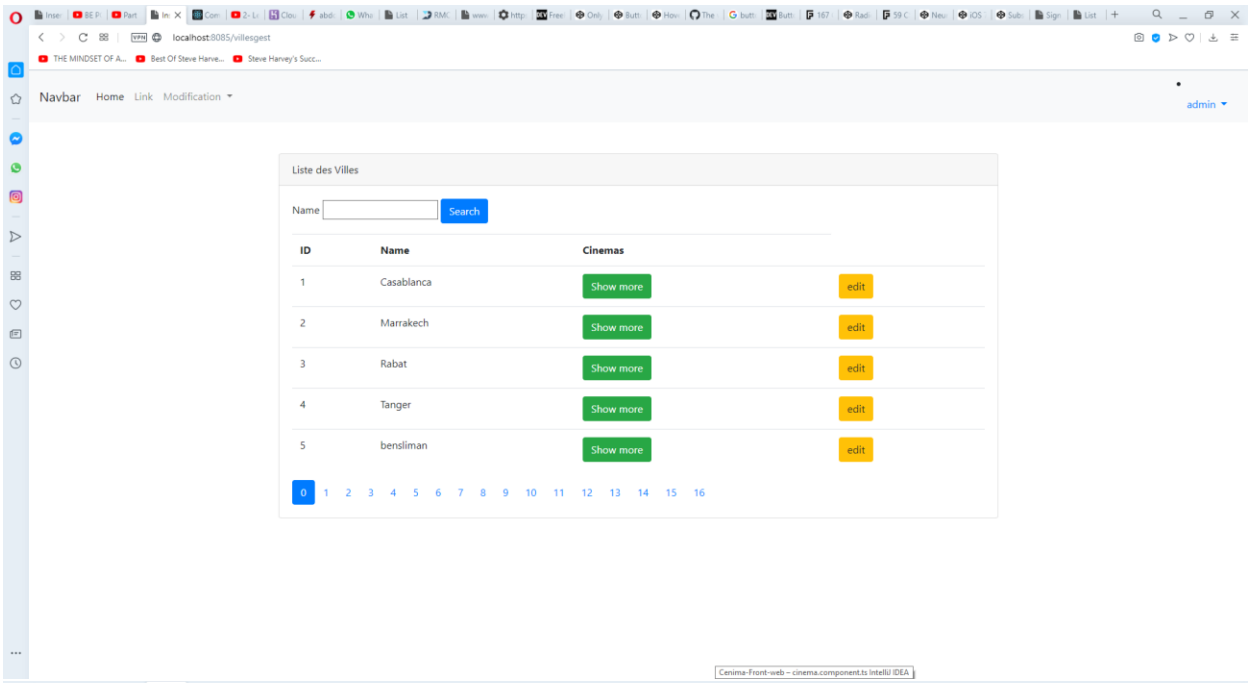
- Admin

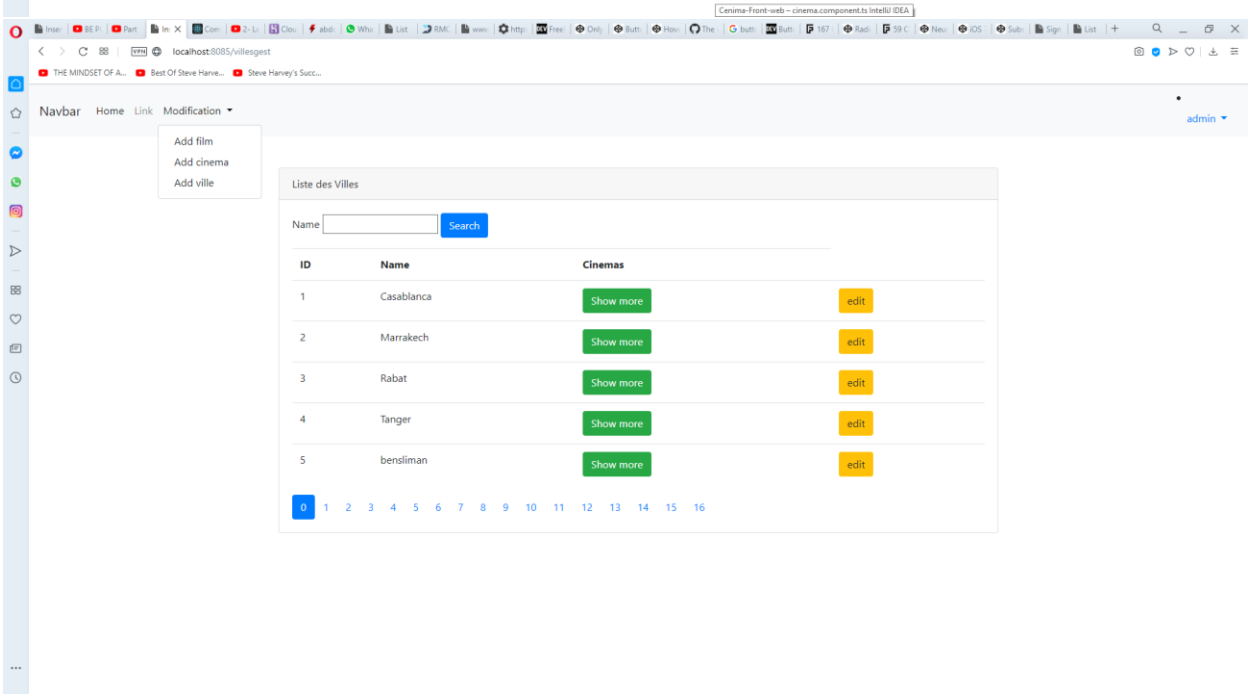

localhost:8085/login
Please sign in
Username
Password
Sign in

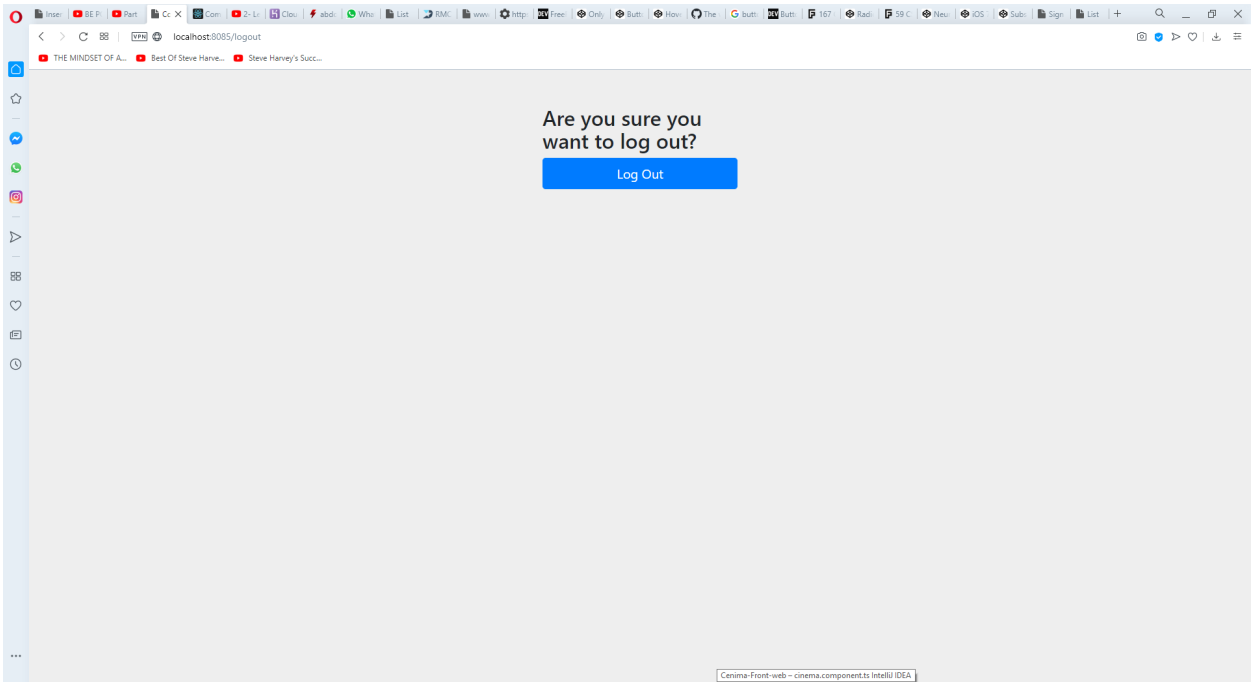

localhost:8085/form
Navbar
Home
Link
Modification
admin
Add film
Titre:
Description:
Duree:
0.0
save


localhost:8085/editville
Navbar
Home
Link
Modification
admin
Add ville
Id: 1
Name:
Casablanca
save


localhost:8085/saveville
Navbar
Home
Link
Modification
admin
Modification success


localhost:8085/villesgest
Navbar
Home
Link
Modification
admin
Liste des Villes
Name
Search
ID
Name
Cinemas
1
Casablanca
Show more
edit
2
Marrakech
Show more
edit
3
Rabat
Show more
edit
4
Tanger
Show more
edit
5
bensliman
Show more
edit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16


Cenima-Front-web – cinema.component.ts IntelliJ IDEA
localhost:8085/villesgest
Navbar
Home
Link
Modification
admin
Add film
Add cinema
Add ville
Liste des Villes
Name
Search
ID
Name
Cinemas
1
Casablanca
Show more
edit
2
Marrakech
Show more
edit
3
Rabat
Show more
edit
4
Tanger
Show more
edit
5
bensliman
Show more
edit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16


localhost:8085/logout
THE MINDSET OF A...
Best Of Steve Harve...
Steve Harvey's Succ...
Are you sure you want to log out?
Log Out
Cenima-Front-web – cinema.component.ts IntelliJ IDEA