Aida Magou DIOP

M1 GLSI

Cours Inf3522 - Développement d'Applications JEE

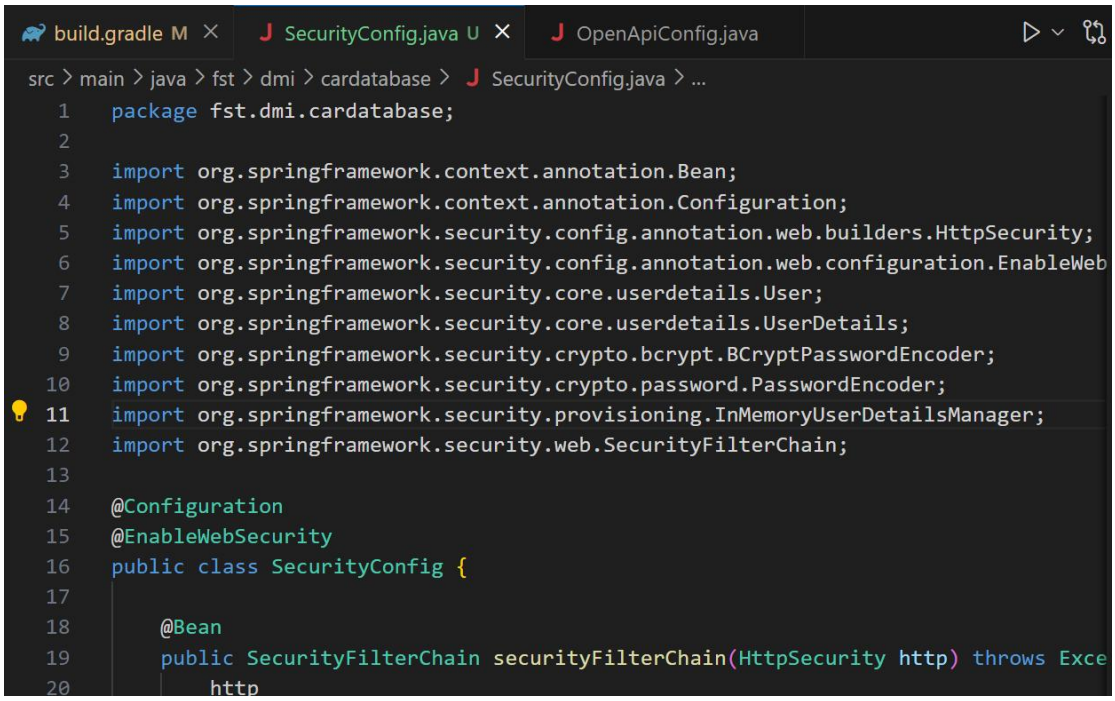
Lab_5 : Sécurisation

Étape 1: Configuration de Base de Spring Security

1. Ajoutons les dépendances nécessaires

Dans build.gradle :

```
dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-web'
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
    implementation 'org.springframework.boot:spring-boot-starter-data-rest'
    implementation group: 'org.springdoc', name: 'springdoc-openapi-starter-w
    developmentOnly 'org.springframework.boot:spring-boot-devtools'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'
    testImplementation 'org.springframework.boot:spring-boot-starter-security
    testImplementation 'org.springframework.security:spring-security-test'
    runtimeOnly 'org.mariadb.jdbc:mariadb-java-client'
}
```



Étape 2: Authentification avec Base de Données

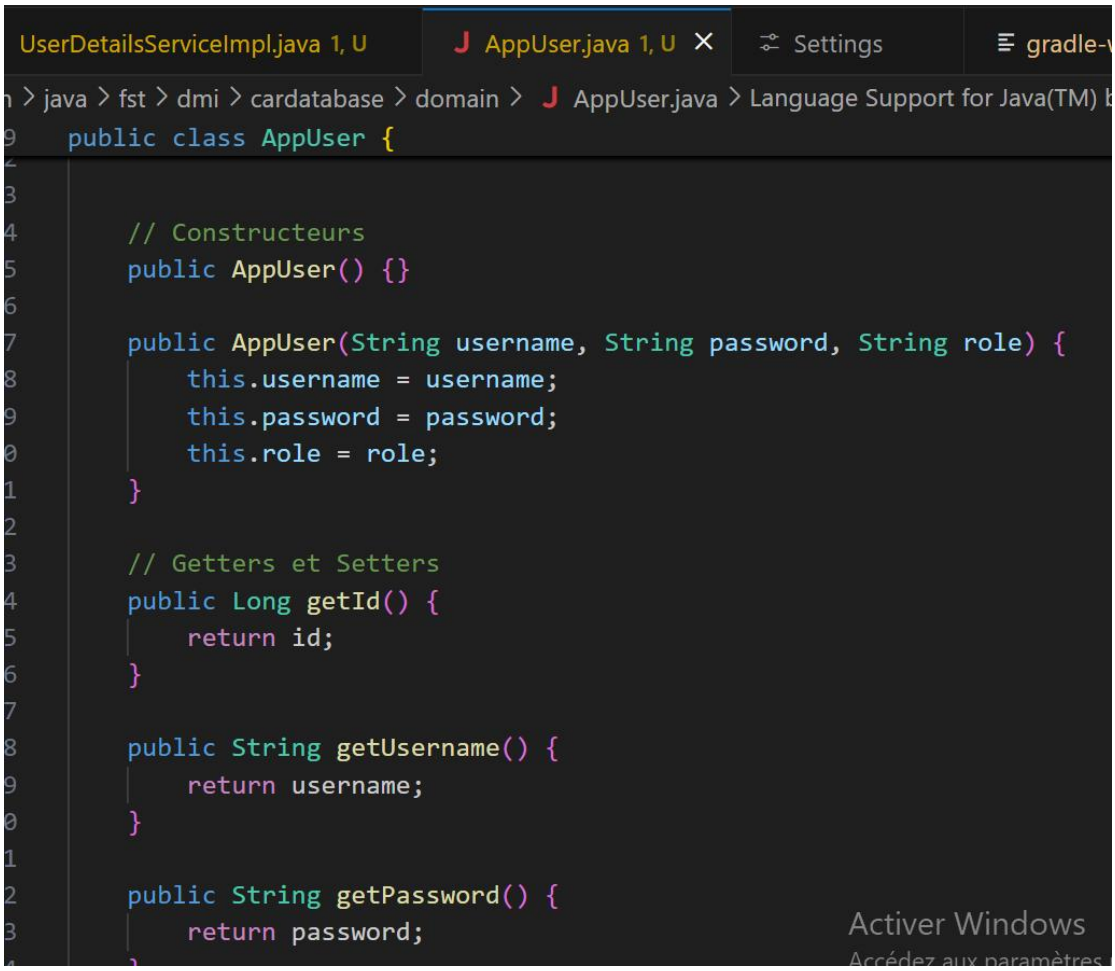
1. Créons l'entité AppUser

2. Créons le Repository

Ajoutons la dependance lombok dans build.gradle :

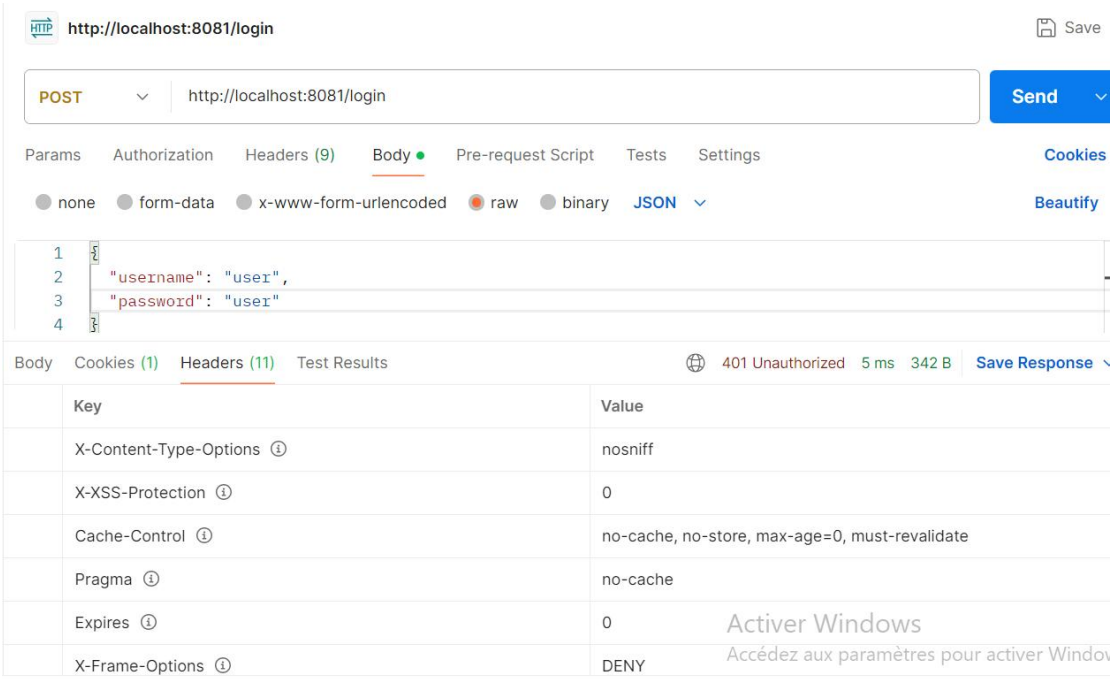
```
dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-web'
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
    implementation 'org.springframework.boot:spring-boot-starter-data-rest'
    implementation group: 'org.springdoc', name: 'springdoc-openapi-starter-webmv
    developmentOnly 'org.springframework.boot:spring-boot-devtools'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'
    testImplementation 'org.springframework.boot:spring-boot-starter-security'
    testImplementation 'org.springframework.security:spring-security-test'
    runtimeOnly 'org.mariadb.jdbc:mariadb-java-client'

    compileOnly 'org.projectlombok:lombok:1.18.30'
    annotationProcessor 'org.projectlombok:lombok:1.18.30'
}
```

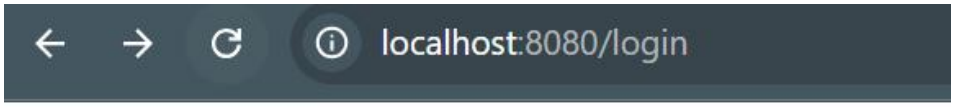
Ajoutons les constructeurs et les getters setters dans AppUser :



```java
public String getRole() {
    return role;
}

public void setId(Long id) {
    this.id = id;
}

public void setUsername(String username)
    this.username = username;
}

public void setPassword(String password)
    this.password = password;
}

public void setRole(String role) {
    this.role = role;
}
```

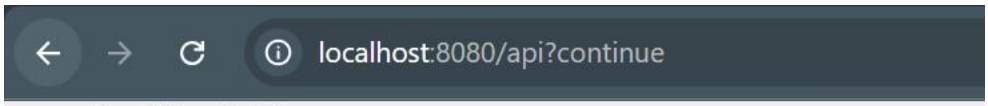
Testons :





Connexion

Nom d'utilisateur : user
Mot de passe : ••••
Se connecter



Impression élégante ☐


```
{
  "_links" : {
    "appUsers" : {
      "href" : "http://localhost:8080/api/appUsers"
    },
    "cars" : {
      "href" : "http://localhost:8080/api/cars"
    },
    "owners" : {
      "href" : "http://localhost:8080/api/owners"
    },
    "profile" : {
      "href" : "http://localhost:8080/api/profile"
    }
  }
}
```


```
MariaDB [cardb]> show tables;
+-----------------+
| Tables_in_cardb |
+-----------------+
| app_user        |
| app_user_seq    |
| car             |
| car_seq         |
| owner           |
| owner_seq       |
+-----------------+
6 rows in set (0.002 sec)

MariaDB [cardb]> select * from app_user;
+----+--------------------------------------------------------------+-------+----------+
| id | password                                                     | role  | username |
+----+--------------------------------------------------------------+-------+----------+
|  1 | $2a$10$NVM0n8ElaRgg7zWO1CxUdei7vWoPg91Lz2aYavh9.f9q0e4bRadue | USER  | user     |
|  2 | $2a$10$8cjz47bjbR4Mn8GMg9IZx.vyjhLXR/SKKMSZ9.mP9vpMu0ssKi8GW | ADMIN | admin    |
+----+--------------------------------------------------------------+-------+----------+
2 rows in set (0.000 sec)

MariaDB [cardb]>
```