Ecole Normale Supérieure de l'Enseignement Technique Mohammedia

Université Hassan II de Casablanca

ENSET

ا لمدرسة العليا لأساتذة التعليم التقنيالمحمدية

جامعة الحسن الثاني بالدار البيضاء

Département Mathématiques et Informatique

Cycle Ingénieur

« Ingénierie Informatique – Big Data et Cloud Computing »

COMPTE-RENDU:
Activité pratique N°2
Hibernate et Spring Data

Réalisé par: Asmaa ELASRI

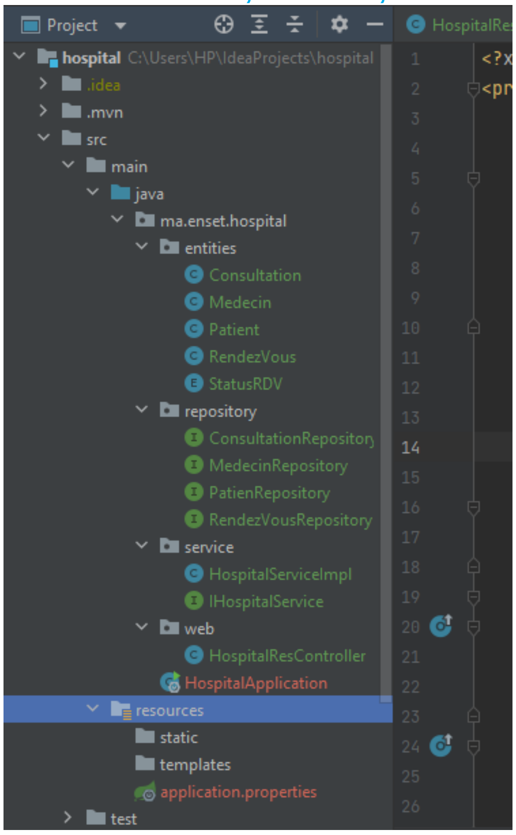
Encadré par: Mr. Mohammed EL YOUSSEFI

Classe : II-BDCC2

Année universitaire : 2021 / 2022

Mapping objet relationnel avec JPA, Hibernate et Spring Data

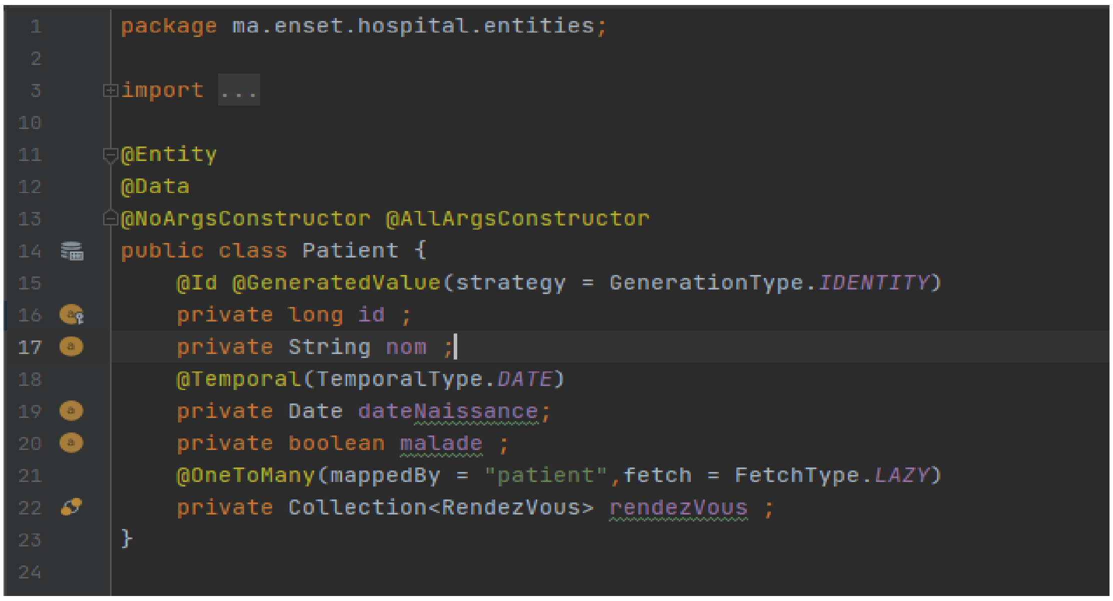
1- Cas de Patient, Medecin, Rendez-vous, Consultation



```
Project
hospital C:\Users\HP\IdeaProjects\hospital
  .idea
  .mvn
  src
    main
      java
        ma.enset.hospital
          entities
            Consultation
            Medecin
            Patient
            RendezVous
            StatusRDV
          repository
            ConsultationRepository
            MedecinRepository
            PatienRepository
            RendezVousRepository
          service
            HospitalServiceImpl
            IHospitalService
          web
            HospitalResController
          HospitalApplication
      resources
        static
        templates
        application.properties
    test
```

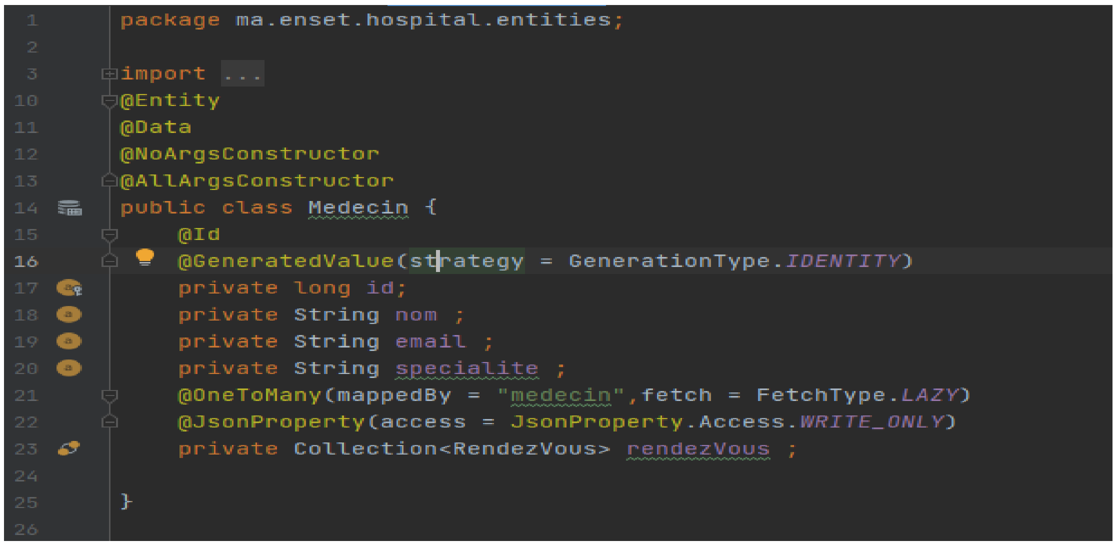
Le package Entities :

La classe Patient



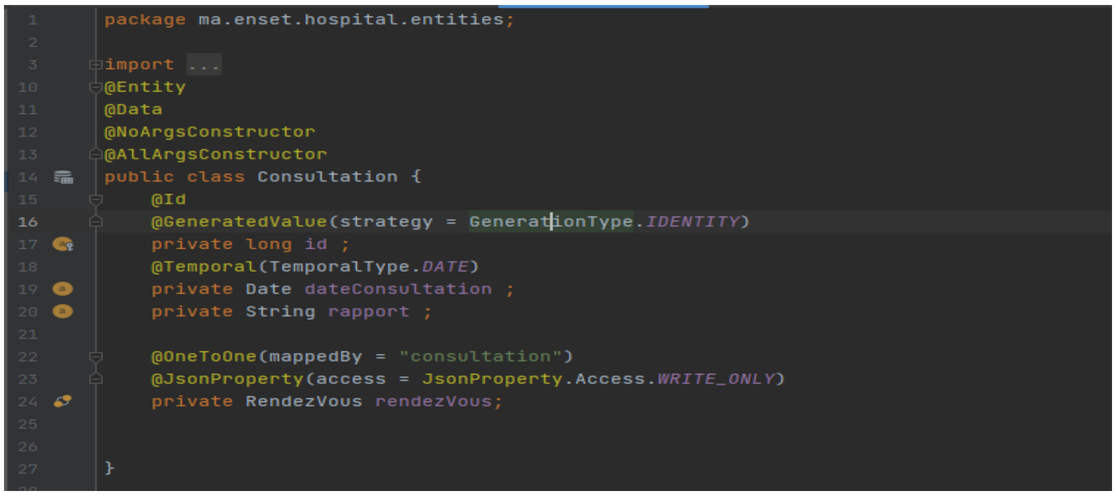
```java
package ma.enset.hospital.entities;

import ...

@Entity
@Data
@NoArgsConstructor @AllArgsConstructor
public class Patient {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id ;
    private String nom ;
    @Temporal(TemporalType.DATE)
    private Date dateNaissance;
    private boolean malade ;
    @OneToMany(mappedBy = "patient",fetch = FetchType.LAZY)
    private Collection<RendezVous> rendezVous ;
}
```

La classe Medecin



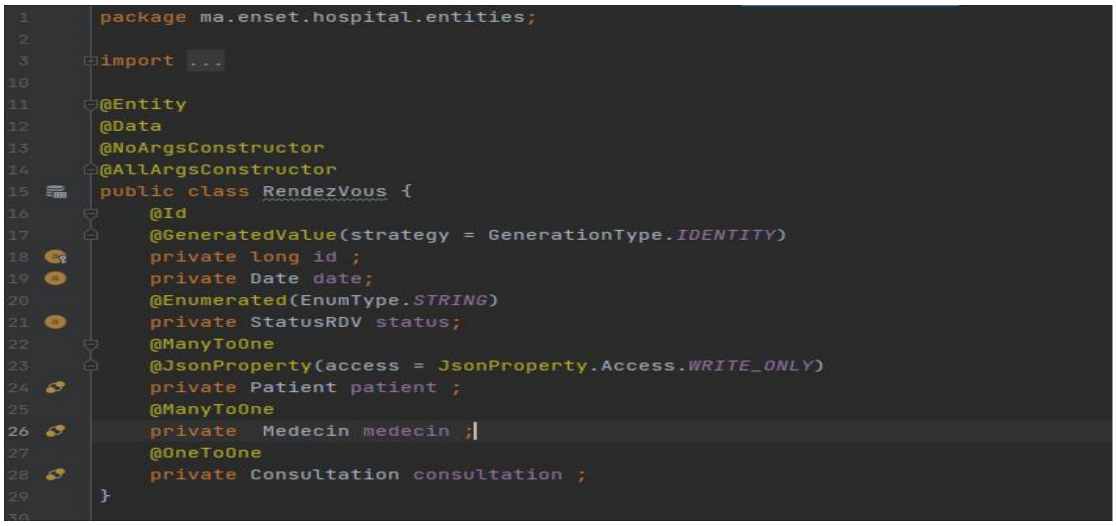
```java
package ma.enset.hospital.entities;

import ...
@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Medecin {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;
    private String nom ;
    private String email ;
    private String specialite ;
    @OneToMany(mappedBy = "medecin",fetch = FetchType.LAZY)
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
    private Collection<RendezVous> rendezVous ;

}
```

La classe Consultation



```java
package ma.enset.hospital.entities;

import ...
@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Consultation {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id ;
    @Temporal(TemporalType.DATE)
    private Date dateConsultation ;
    private String rapport ;

    @OneToOne(mappedBy = "consultation")
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
    private RendezVous rendezVous;


}
```

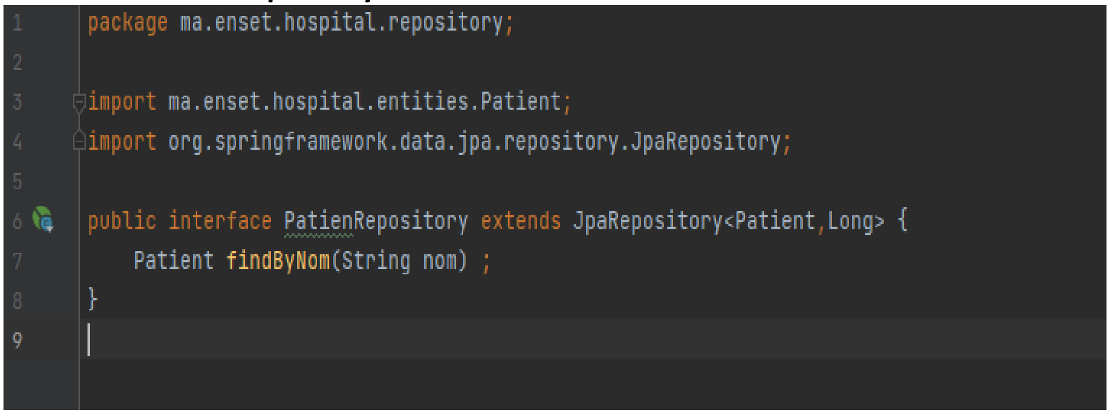
La classe Rendez-vous



```java
package ma.enset.hospital.entities;

import ...

@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
public class RendezVous {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id ;
    private Date date;
    @Enumerated(EnumType.STRING)
    private StatusRDV status;
    @ManyToOne
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
    private Patient patient ;
    @ManyToOne
    private  Medecin medecin ;
    @OneToOne
    private Consultation consultation ;
}
```

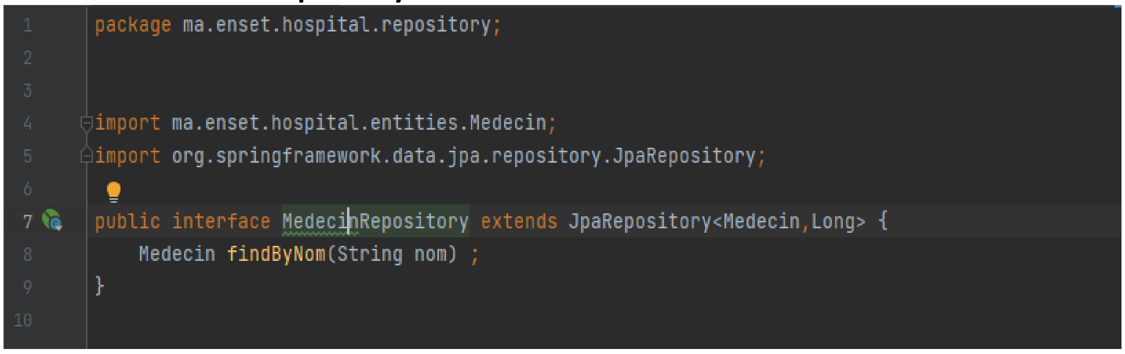
Le package Repository

L’interface PatienRepository



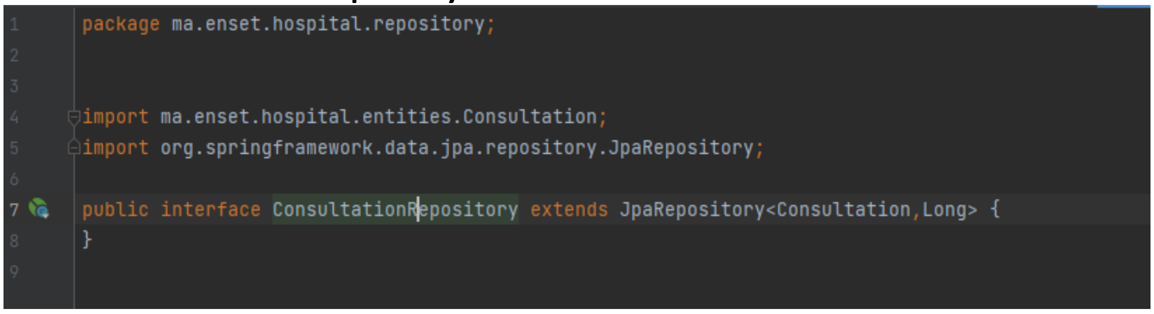
```java
package ma.enset.hospital.repository;

import ma.enset.hospital.entities.Patient;
import org.springframework.data.jpa.repository.JpaRepository;

public interface PatienRepository extends JpaRepository<Patient,Long> {
    Patient findByNom(String nom) ;
}

```

L’interface MedecinRepository



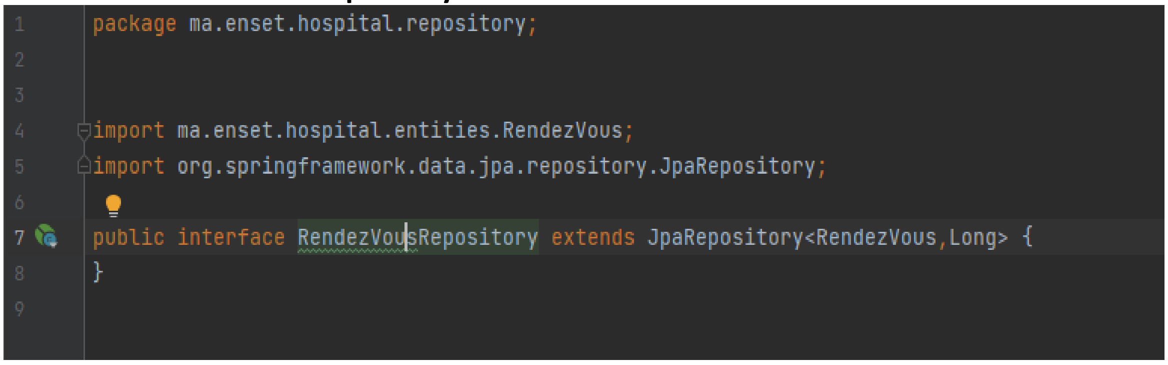
```java
package ma.enset.hospital.repository;


import ma.enset.hospital.entities.Medecin;
import org.springframework.data.jpa.repository.JpaRepository;

public interface MedecinRepository extends JpaRepository<Medecin,Long> {
    Medecin findByNom(String nom) ;
}

```

L’interface ConsultationRepository



```java
package ma.enset.hospital.repository;


import ma.enset.hospital.entities.Consultation;
import org.springframework.data.jpa.repository.JpaRepository;

public interface ConsultationRepository extends JpaRepository<Consultation,Long> {
}

```

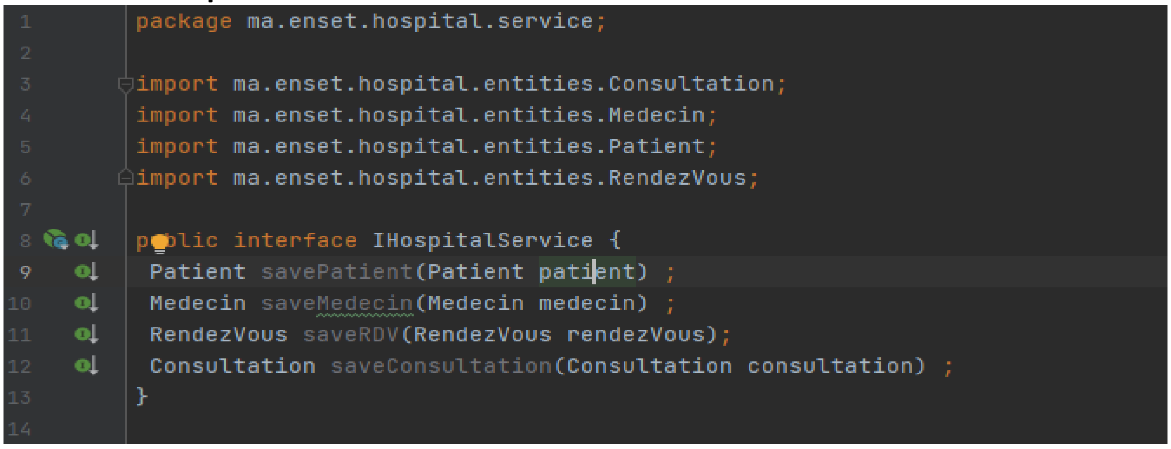
L’interface RendezVousRepository



```java
package ma.enset.hospital.repository;


import ma.enset.hospital.entities.RendezVous;
import org.springframework.data.jpa.repository.JpaRepository;

public interface RendezVousRepository extends JpaRepository<RendezVous,Long> {
}

```

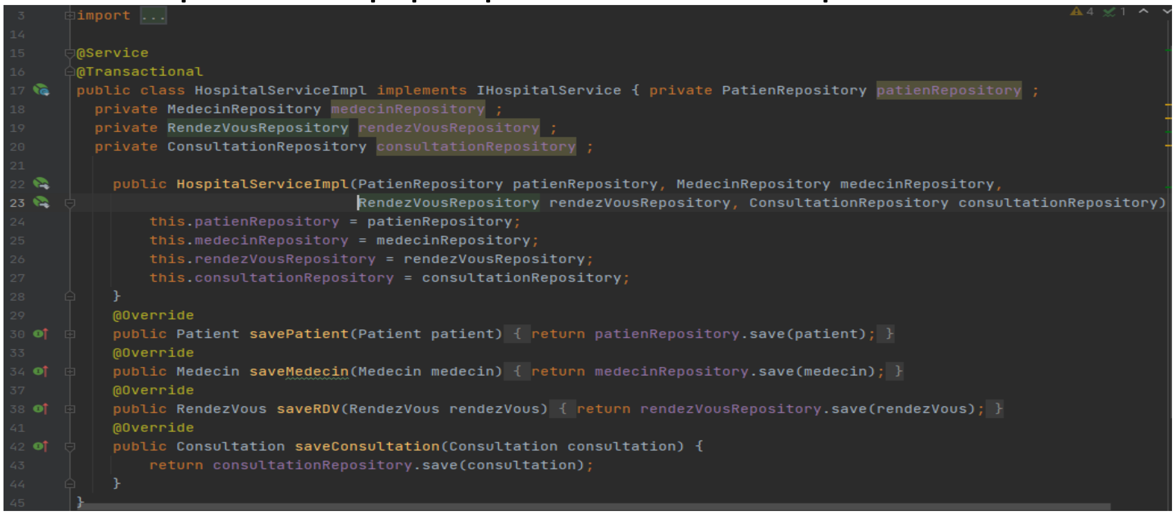
Le package Service

L'interface IHospitalService :



```java
package ma.enset.hospital.service;

import ma.enset.hospital.entities.Consultation;
import ma.enset.hospital.entities.Medecin;
import ma.enset.hospital.entities.Patient;
import ma.enset.hospital.entities.RendezVous;

public interface IHospitalService {
 Patient savePatient(Patient patient) ;
 Medecin saveMedecin(Medecin medecin) ;
 RendezVous saveRDV(RendezVous rendezVous);
 Consultation saveConsultation(Consultation consultation) ;
}
```

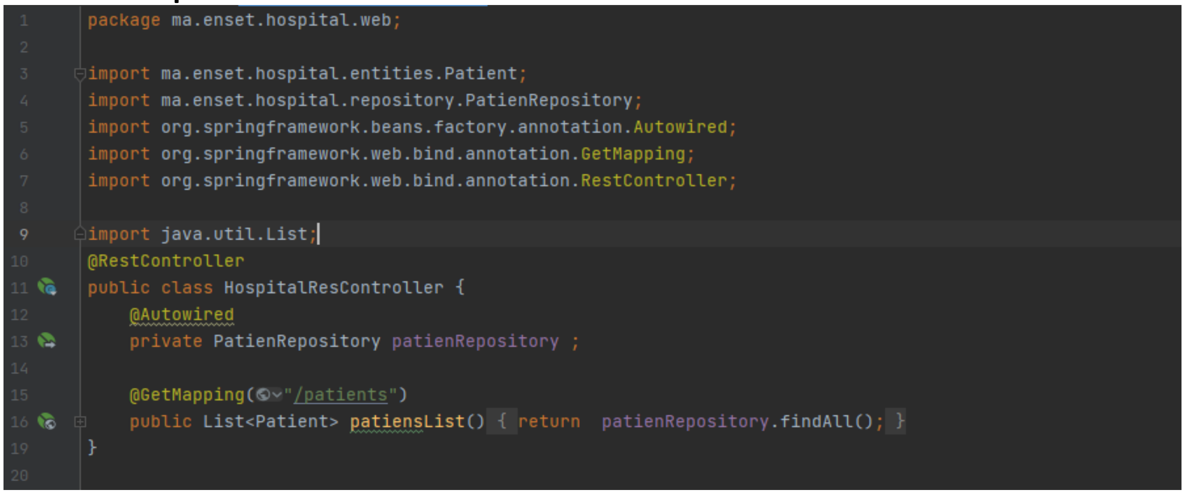
La classe HospitalServiceImpl qui implemente l'interface IHospitalService



```java
import ...

@Service
@Transactional
public class HospitalServiceImpl implements IHospitalService { private PatienRepository patienRepository ;
  private MedecinRepository medecinRepository ;
  private RendezVousRepository rendezVousRepository ;
  private ConsultationRepository consultationRepository ;

    public HospitalServiceImpl(PatienRepository patienRepository, MedecinRepository medecinRepository,
                               RendezVousRepository rendezVousRepository, ConsultationRepository consultationRepository)
        this.patienRepository = patienRepository;
        this.medecinRepository = medecinRepository;
        this.rendezVousRepository = rendezVousRepository;
        this.consultationRepository = consultationRepository;
    }
    @Override
    public Patient savePatient(Patient patient) { return patienRepository.save(patient); }
    @Override
    public Medecin saveMedecin(Medecin medecin) { return medecinRepository.save(medecin); }
    @Override
    public RendezVous saveRDV(RendezVous rendezVous) { return rendezVousRepository.save(rendezVous); }
    @Override
    public Consultation saveConsultation(Consultation consultation) {
        return consultationRepository.save(consultation);
    }
}
```

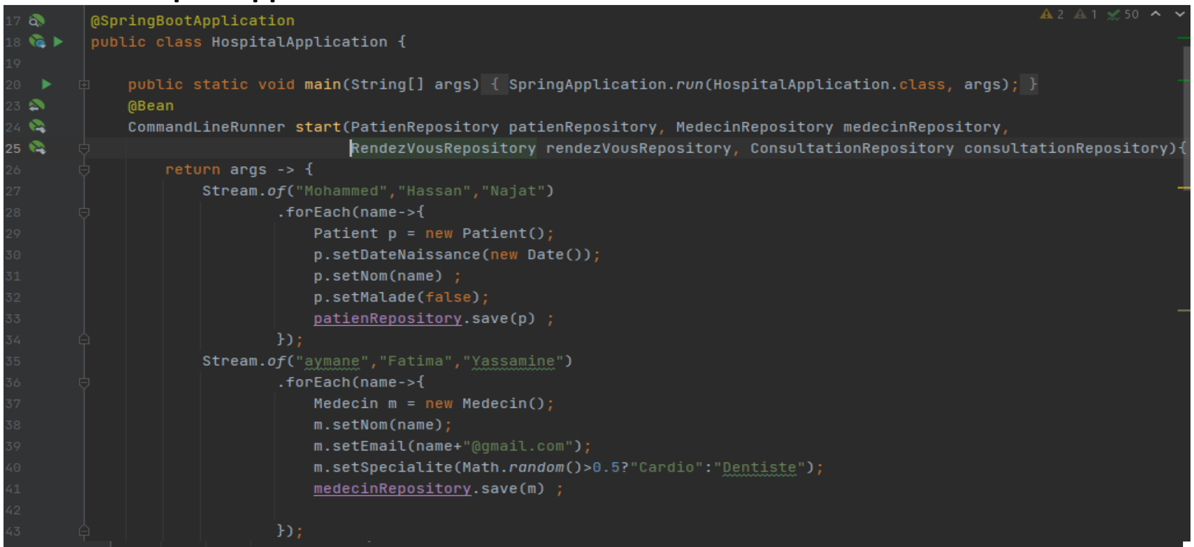
Le package web

La classe HospitalResController

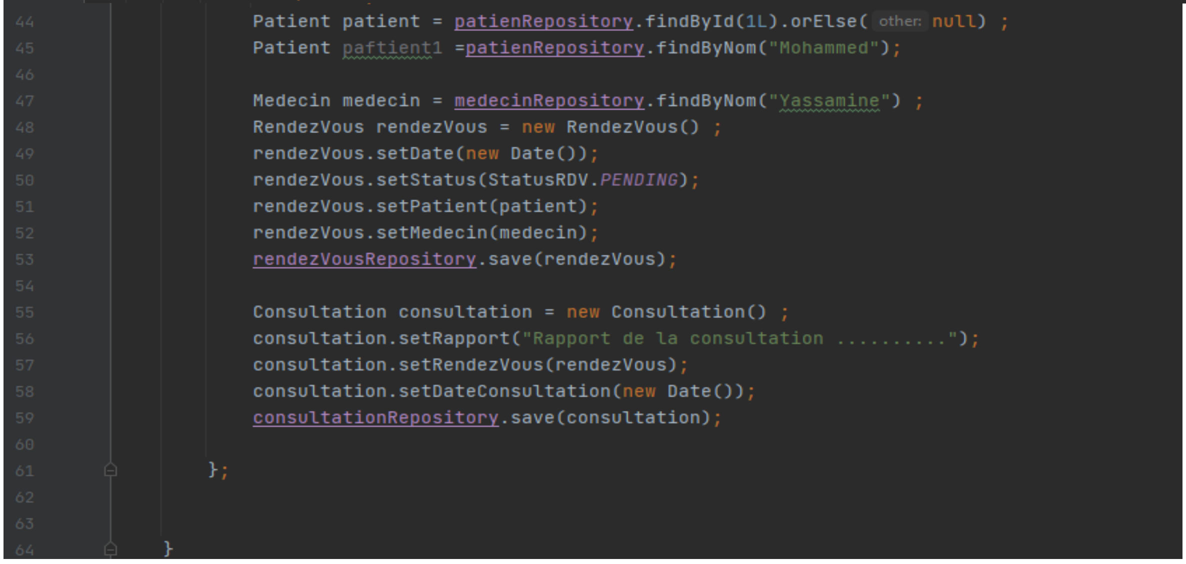


```java
package ma.enset.hospital.web;

import ma.enset.hospital.entities.Patient;
import ma.enset.hospital.repository.PatienRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;
@RestController
public class HospitalResController {
    @Autowired
    private PatienRepository patienRepository ;

    @GetMapping("/patients")
    public List<Patient> patiensList() { return  patienRepository.findAll(); }
}
```

La classe HospitalApplication :



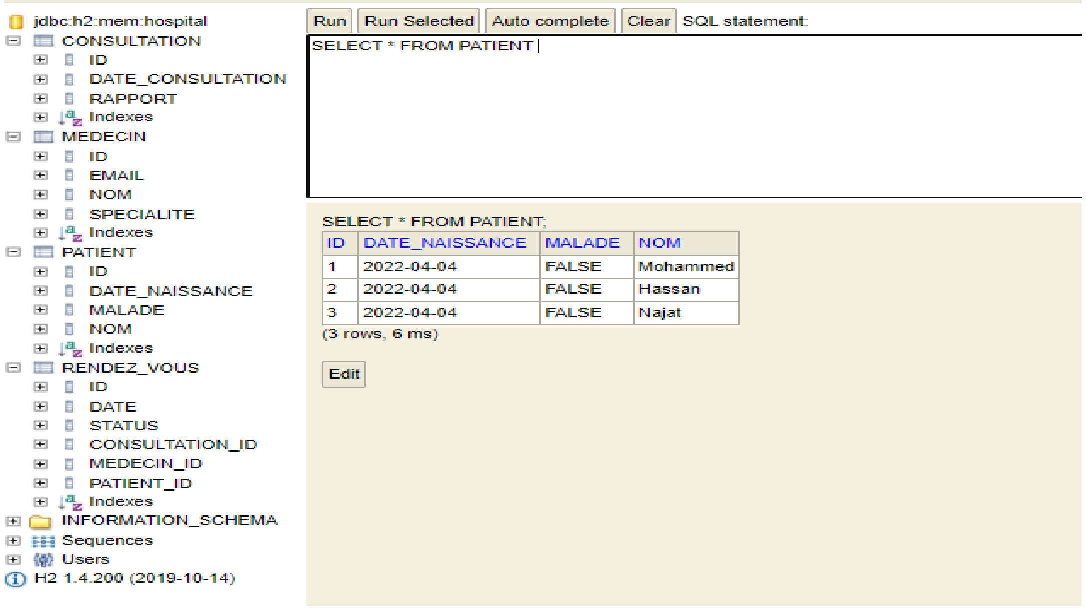
```java
@SpringBootApplication
public class HospitalApplication {

    public static void main(String[] args) { SpringApplication.run(HospitalApplication.class, args); }
    @Bean
    CommandLineRunner start(PatienRepository patienRepository, MedecinRepository medecinRepository,
                            RendezVousRepository rendezVousRepository, ConsultationRepository consultationRepository){
        return args -> {
            Stream.of("Mohammed","Hassan","Najat")
                    .forEach(name->{
                        Patient p = new Patient();
                        p.setDateNaissance(new Date());
                        p.setNom(name) ;
                        p.setMalade(false);
                        patienRepository.save(p) ;
                    });
            Stream.of("aymane","Fatima","Yassamine")
                    .forEach(name->{
                        Medecin m = new Medecin();
                        m.setNom(name);
                        m.setEmail(name+"@gmail.com");
                        m.setSpecialite(Math.random()>0.5?"Cardio":"Dentiste");
                        medecinRepository.save(m) ;

                    });
```



```java
            Patient patient = patienRepository.findById(1L).orElse( other: null) ;
            Patient paftient1 =patienRepository.findByNom("Mohammed");

            Medecin medecin = medecinRepository.findByNom("Yassamine") ;
            RendezVous rendezVous = new RendezVous() ;
            rendezVous.setDate(new Date());
            rendezVous.setStatus(StatusRDV.PENDING);
            rendezVous.setPatient(patient);
            rendezVous.setMedecin(medecin);
            rendezVousRepository.save(rendezVous);

            Consultation consultation = new Consultation() ;
            consultation.setRapport("Rapport de la consultation ..........");
            consultation.setRendezVous(rendezVous);
            consultation.setDateConsultation(new Date());
            consultationRepository.save(consultation);

        };


    }
```

Résultat :

La base de données H2 :

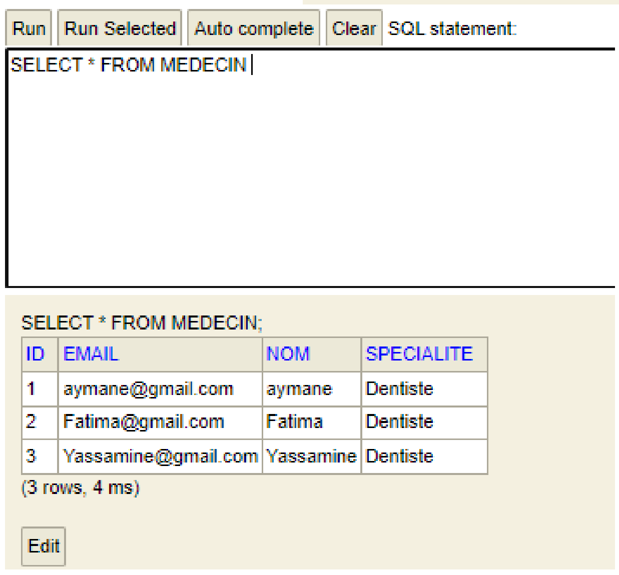


```
SELECT * FROM PATIENT
```

SELECT * FROM PATIENT;

ID	DATE_NAISSANCE	MALADE	NOM
1	2022-04-04	FALSE	Mohammed
2	2022-04-04	FALSE	Hassan
3	2022-04-04	FALSE	Najat

(3 rows, 6 ms)

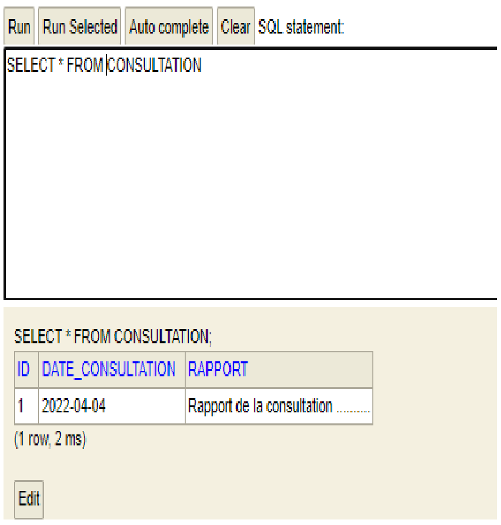


```
SELECT * FROM MEDECIN
```

SELECT * FROM MEDECIN;

ID	EMAIL	NOM	SPECIALITE
1	aymane@gmail.com	aymane	Dentiste
2	Fatima@gmail.com	Fatima	Dentiste
3	Yassamine@gmail.com	Yassamine	Dentiste

(3 rows, 4 ms)

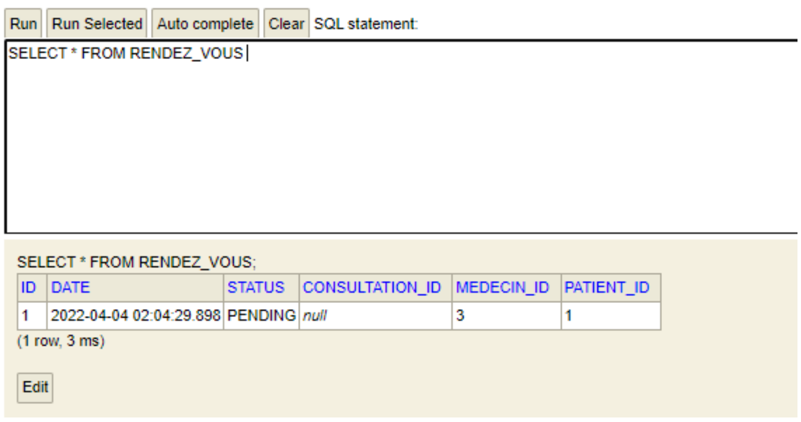


```
SELECT * FROM CONSULTATION
```

SELECT * FROM CONSULTATION;

ID	DATE_CONSULTATION	RAPPORT
1	2022-04-04	Rapport de la consultation ....

(1 row, 2 ms)



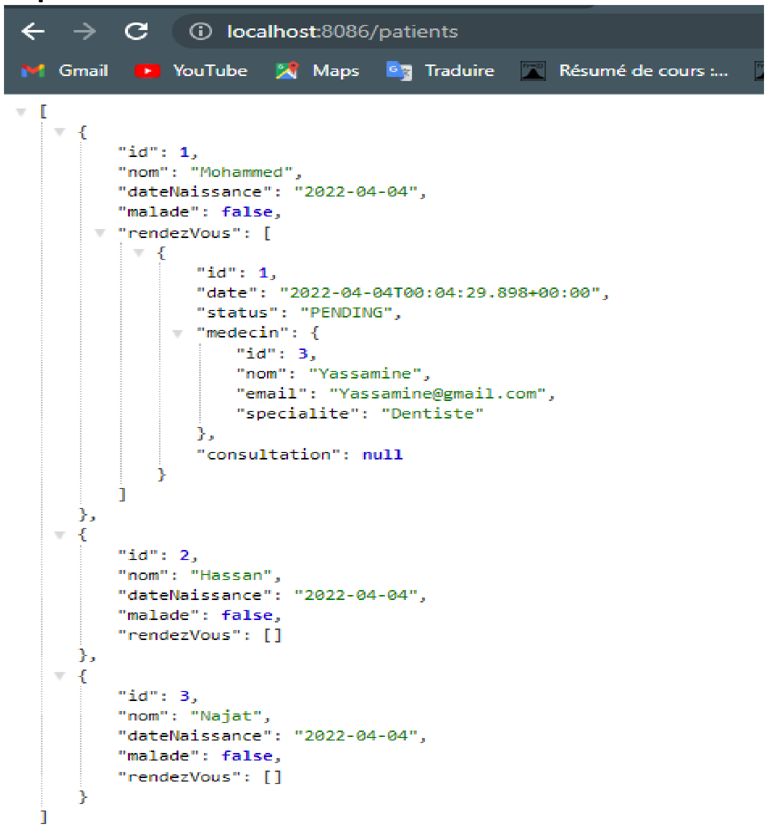
```
SELECT * FROM RENDEZ_VOUS
```

SELECT * FROM RENDEZ_VOUS;

ID	DATE	STATUS	CONSULTATION_ID	MEDECIN_ID	PATIENT_ID
1	2022-04-04 02:04:29.898	PENDING	*null*	3	1

(1 row, 3 ms)

La partie Web :

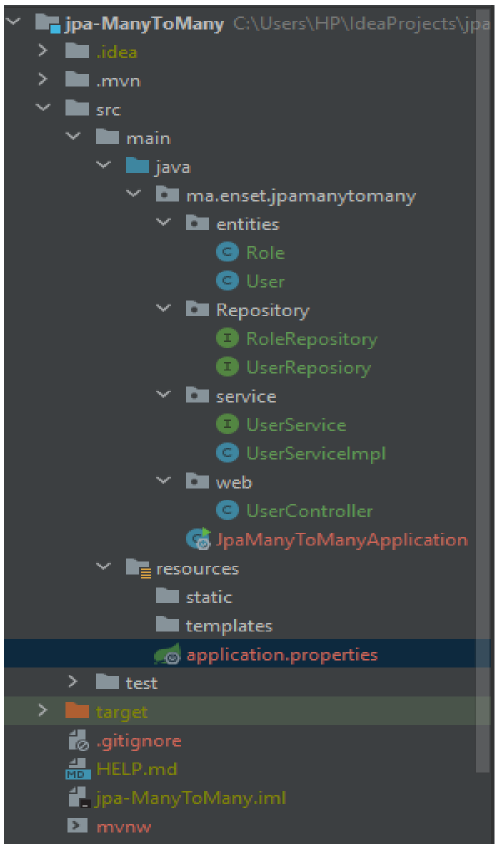


localhost:8086/patients

Gmail YouTube Maps Traduire Résumé de cours :...

```
[
  {
    "id": 1,
    "nom": "Mohammed",
    "dateNaissance": "2022-04-04",
    "malade": false,
    "rendezVous": [
      {
        "id": 1,
        "date": "2022-04-04T00:04:29.898+00:00",
        "status": "PENDING",
        "medecin": {
          "id": 3,
          "nom": "Yassamine",
          "email": "Yassamine@gmail.com",
          "specialite": "Dentiste"
        },
        "consultation": null
      }
    ]
  },
  {
    "id": 2,
    "nom": "Hassan",
    "dateNaissance": "2022-04-04",
    "malade": false,
    "rendezVous": []
  },
  {
    "id": 3,
    "nom": "Najat",
    "dateNaissance": "2022-04-04",
    "malade": false,
    "rendezVous": []
  }
]
```

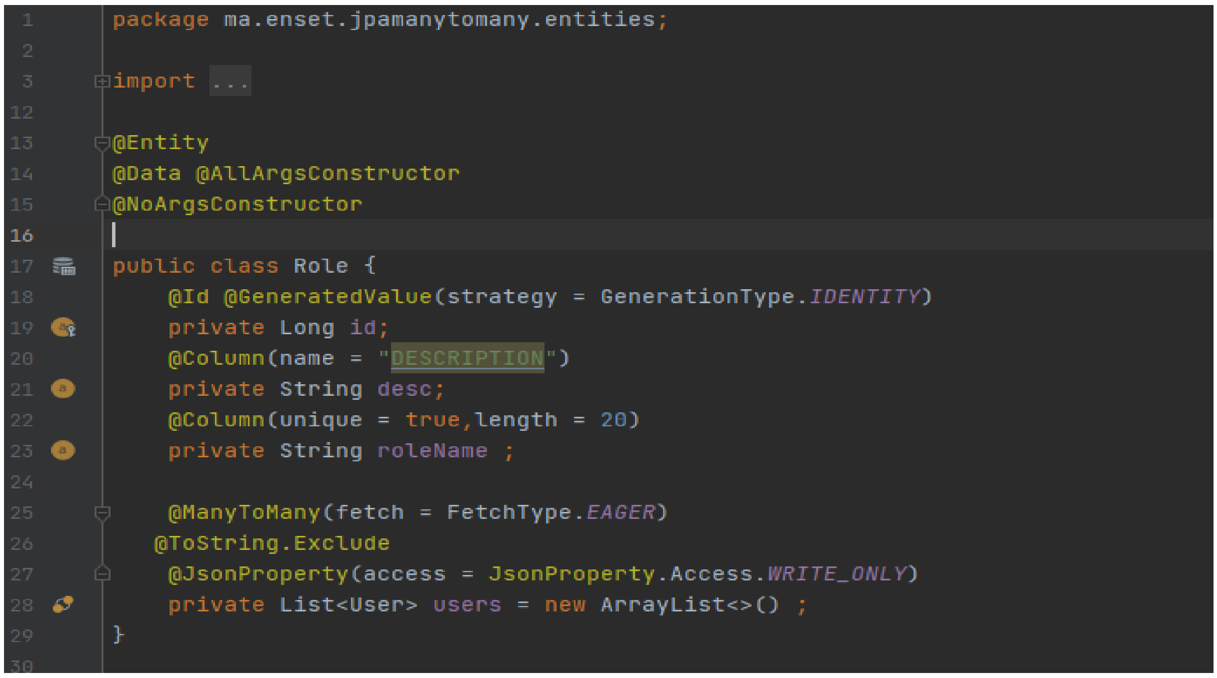
2- Cas de Users et Roles :



```
jpa-ManyToMany C:\Users\HP\IdeaProjects\jpa
  .idea
  .mvn
  src
    main
      java
        ma.enset.jpamanytomany
          entities
            Role
            User
          Repository
            RoleRepository
            UserReposiory
          service
            UserService
            UserServiceImpl
          web
            UserController
          JpaManyToManyApplication
      resources
        static
        templates
        application.properties
    test
  target
  .gitignore
  HELP.md
  jpa-ManyToMany.iml
  mvnw
```

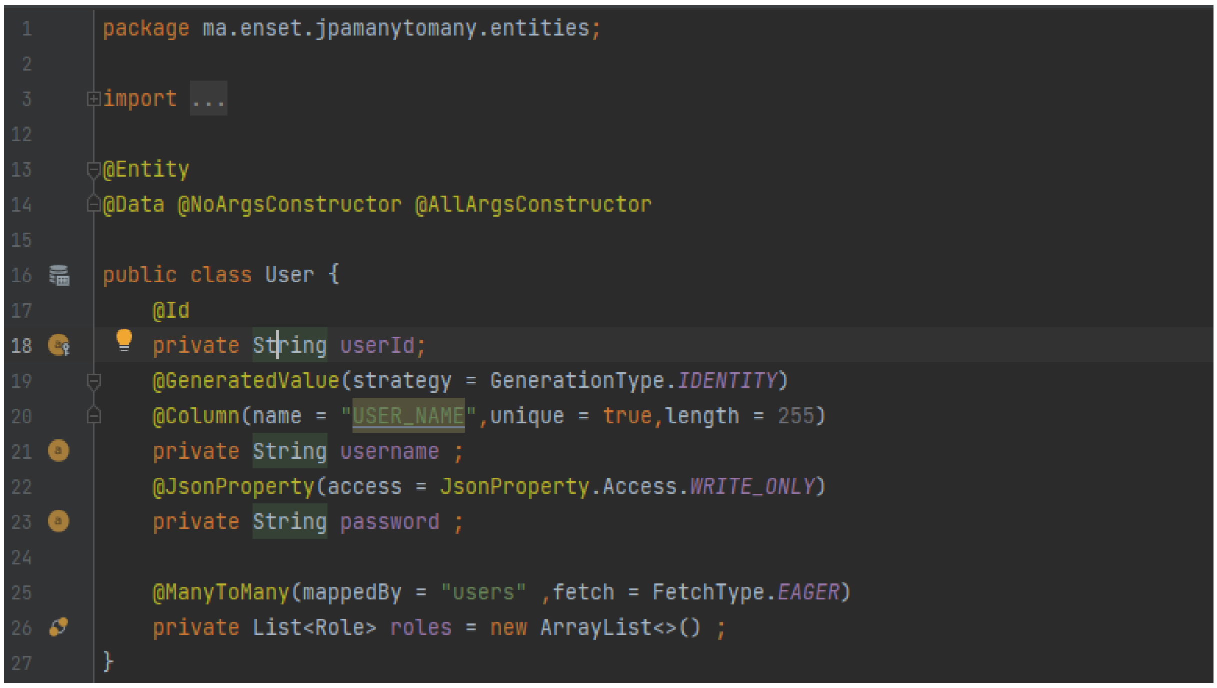
Le package Entities :

La classe Role



```java
package ma.enset.jpamanytomany.entities;

import ...

@Entity
@Data @AllArgsConstructor
@NoArgsConstructor

public class Role {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @Column(name = "DESCRIPTION")
    private String desc;
    @Column(unique = true,length = 20)
    private String roleName ;

    @ManyToMany(fetch = FetchType.EAGER)
   @ToString.Exclude
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
    private List<User> users = new ArrayList<>() ;
}
```

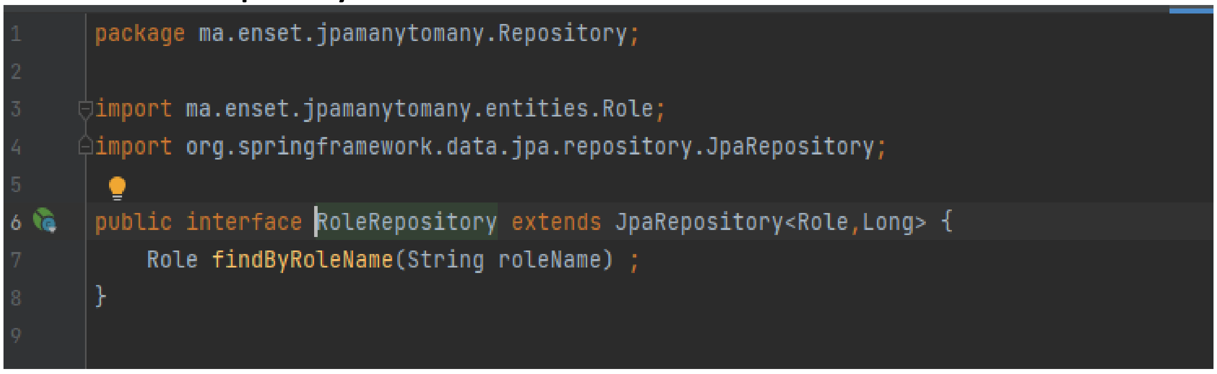
La classe User :



```java
package ma.enset.jpamanytomany.entities;

import ...

@Entity
@Data @NoArgsConstructor @AllArgsConstructor

public class User {
    @Id
    private String userId;
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "USER_NAME",unique = true,length = 255)
    private String username ;
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
    private String password ;

    @ManyToMany(mappedBy = "users" ,fetch = FetchType.EAGER)
    private List<Role> roles = new ArrayList<>() ;
}
```

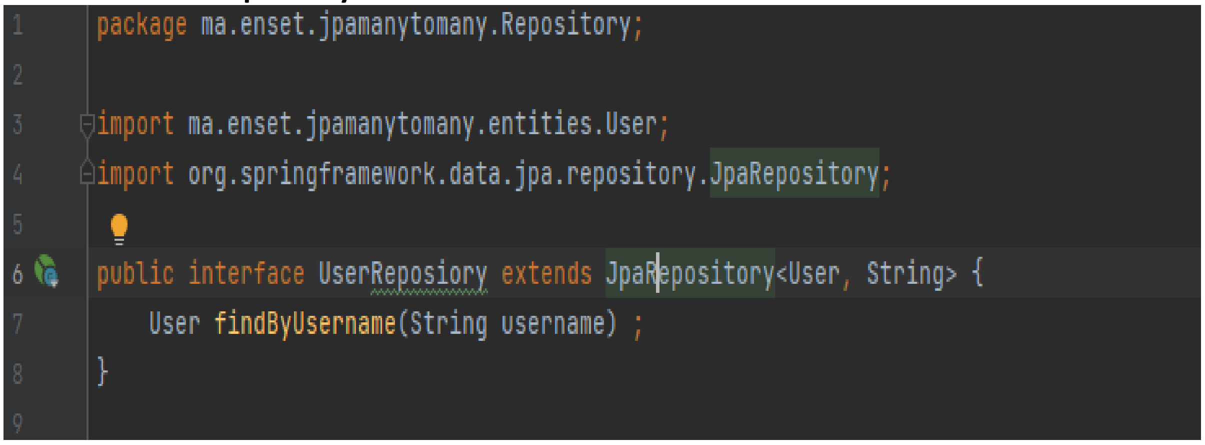
Le package Repository

L’interface RoleRepository



```java
package ma.enset.jpamanytomany.Repository;

import ma.enset.jpamanytomany.entities.Role;
import org.springframework.data.jpa.repository.JpaRepository;

public interface RoleRepository extends JpaRepository<Role,Long> {
    Role findByRoleName(String roleName) ;
}

```

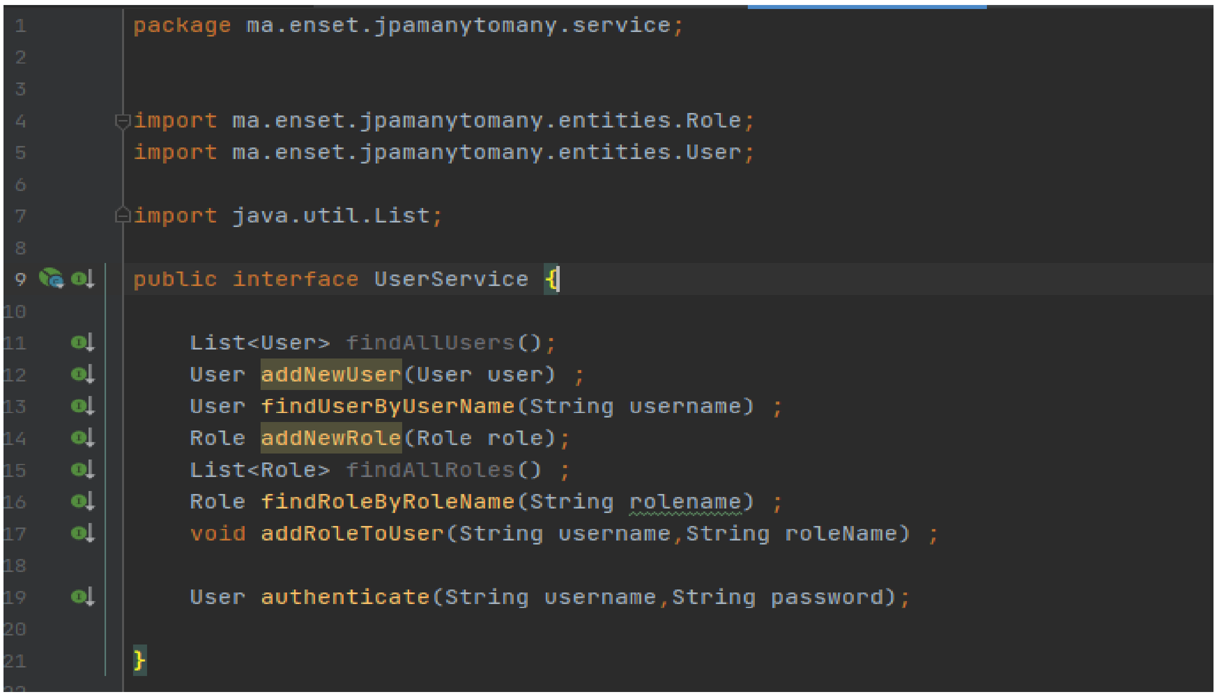
L’interface UserRepository



```java
package ma.enset.jpamanytomany.Repository;

import ma.enset.jpamanytomany.entities.User;
import org.springframework.data.jpa.repository.JpaRepository;

public interface UserReposiory extends JpaRepository<User, String> {
    User findByUsername(String username) ;
}

```

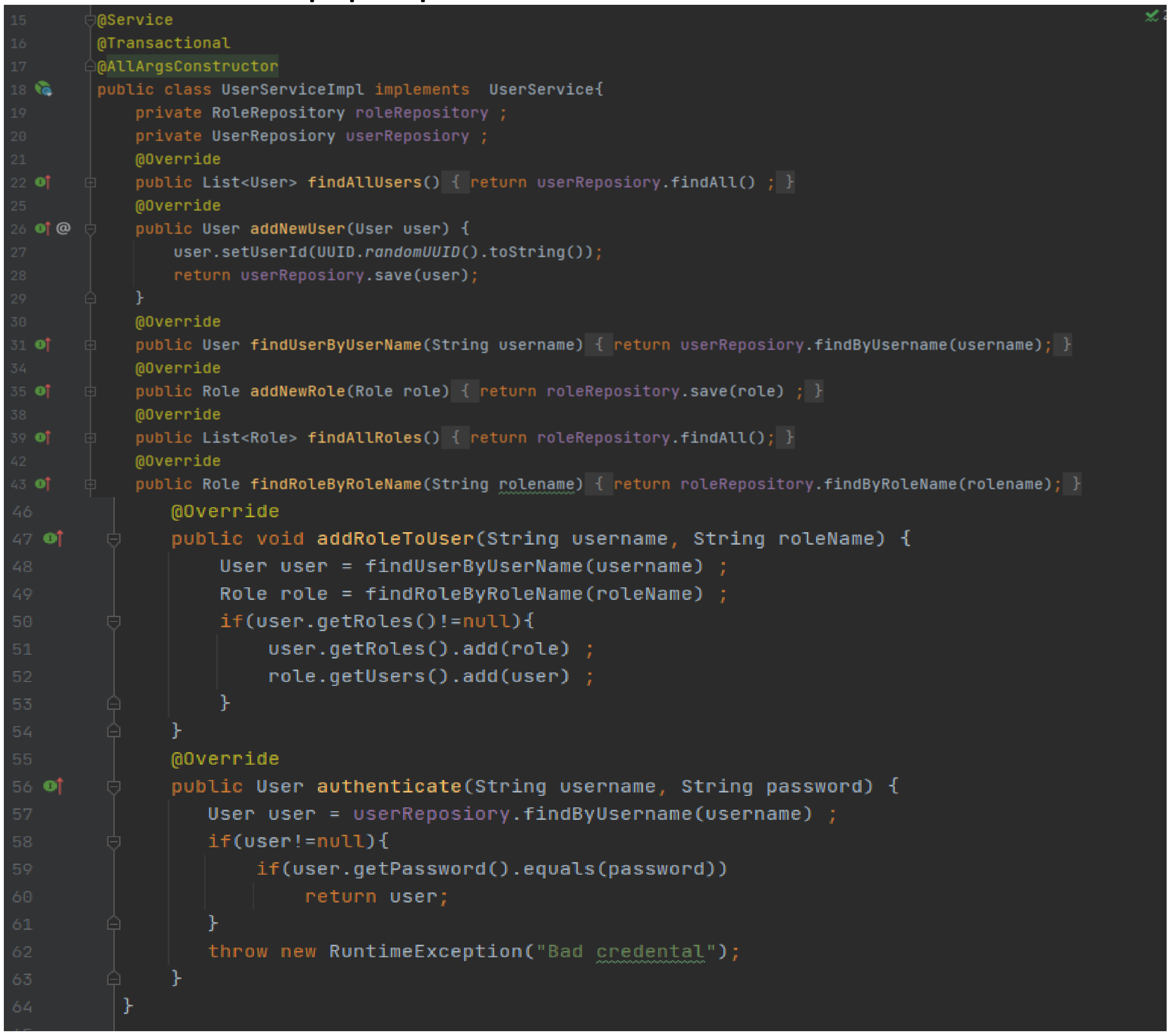
Le package Service

L’interface UserService



```java
package ma.enset.jpamanytomany.service;


import ma.enset.jpamanytomany.entities.Role;
import ma.enset.jpamanytomany.entities.User;

import java.util.List;

public interface UserService {

    List<User> findAllUsers();
    User addNewUser(User user) ;
    User findUserByUserName(String username) ;
    Role addNewRole(Role role);
    List<Role> findAllRoles() ;
    Role findRoleByRoleName(String rolename) ;
    void addRoleToUser(String username,String roleName) ;

    User authenticate(String username,String password);

}
```

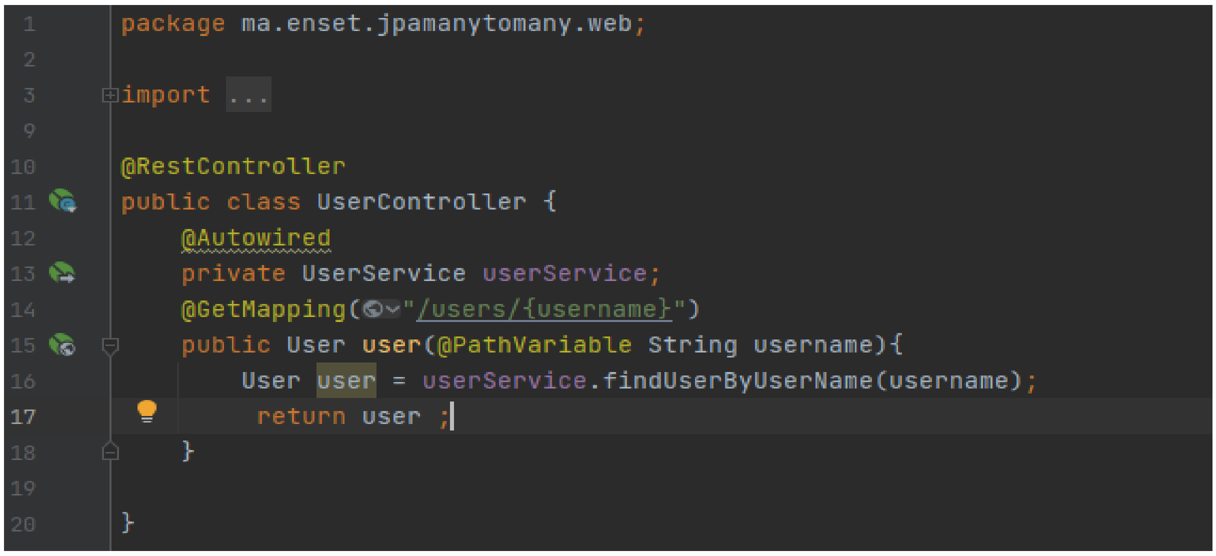
La classe UserServiceImpl qui implémente l’interface UserService



```java
@Service
@Transactional
@AllArgsConstructor
public class UserServiceImpl implements  UserService{
    private RoleRepository roleRepository ;
    private UserReposiory userReposiory ;
    @Override
    public List<User> findAllUsers() { return userReposiory.findAll() ; }
    @Override
    public User addNewUser(User user) {
        user.setUserId(UUID.randomUUID().toString());
        return userReposiory.save(user);
    }
    @Override
    public User findUserByUserName(String username) { return userReposiory.findByUsername(username); }
    @Override
    public Role addNewRole(Role role) { return roleRepository.save(role) ; }
    @Override
    public List<Role> findAllRoles() { return roleRepository.findAll(); }
    @Override
    public Role findRoleByRoleName(String rolename) { return roleRepository.findByRoleName(rolename); }
    @Override
    public void addRoleToUser(String username, String roleName) {
        User user = findUserByUserName(username) ;
        Role role = findRoleByRoleName(roleName) ;
        if(user.getRoles()!=null){
            user.getRoles().add(role) ;
            role.getUsers().add(user) ;
        }
    }
    @Override
    public User authenticate(String username, String password) {
       User user = userReposiory.findByUsername(username) ;
       if(user!=null){
           if(user.getPassword().equals(password))
               return user;
       }
       throw new RuntimeException("Bad credental");
    }
}
```

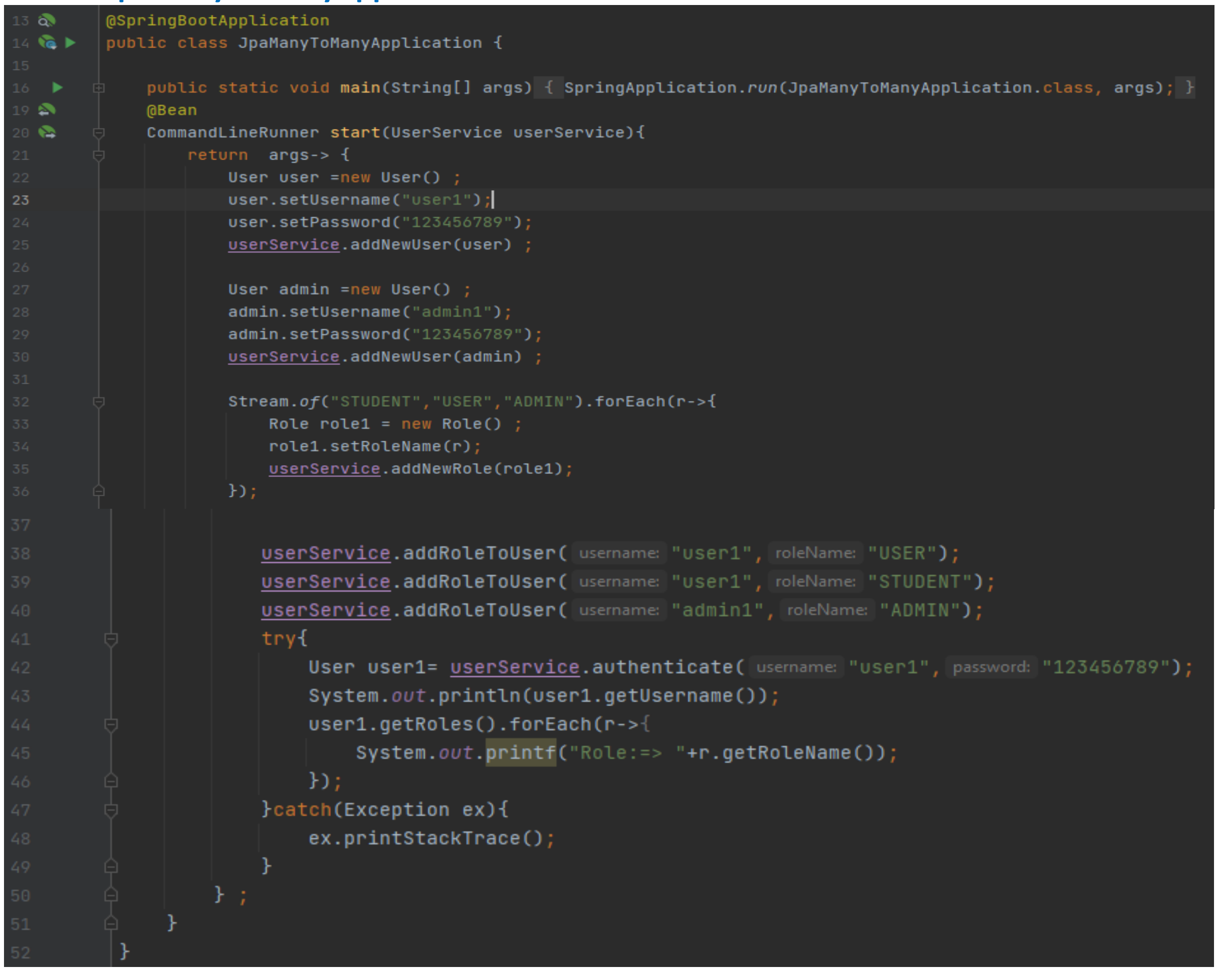
Le package web

UserController



```java
package ma.enset.jpamanytomany.web;

import ...

@RestController
public class UserController {
    @Autowired
    private UserService userService;
    @GetMapping("/users/{username}")
    public User user(@PathVariable String username){
        User user = userService.findUserByUserName(username);
         return user ;
    }

}
```

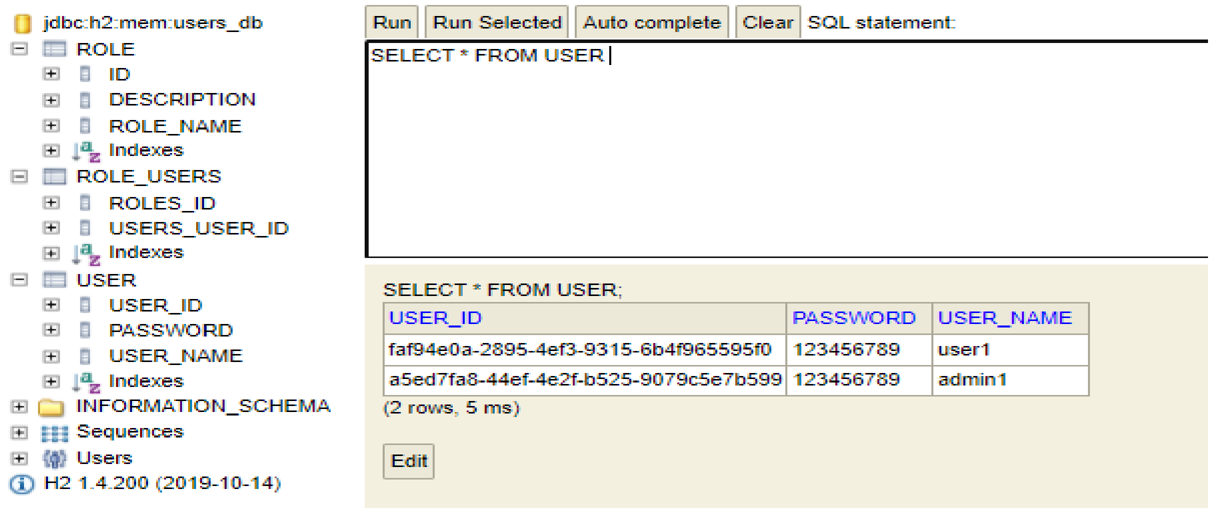
La class JpaManyToManyApplication



```java
@SpringBootApplication
public class JpaManyToManyApplication {

    public static void main(String[] args) { SpringApplication.run(JpaManyToManyApplication.class, args); }
    @Bean
    CommandLineRunner start(UserService userService){
        return  args-> {
            User user =new User() ;
            user.setUsername("user1");
            user.setPassword("123456789");
            userService.addNewUser(user) ;

            User admin =new User() ;
            admin.setUsername("admin1");
            admin.setPassword("123456789");
            userService.addNewUser(admin) ;

            Stream.of("STUDENT","USER","ADMIN").forEach(r->{
                Role role1 = new Role() ;
                role1.setRoleName(r);
                userService.addNewRole(role1);
            });

            userService.addRoleToUser( username: "user1", roleName: "USER");
            userService.addRoleToUser( username: "user1", roleName: "STUDENT");
            userService.addRoleToUser( username: "admin1", roleName: "ADMIN");
            try{
                User user1= userService.authenticate( username: "user1", password: "123456789");
                System.out.println(user1.getUsername());
                user1.getRoles().forEach(r->{
                    System.out.printf("Role:=> "+r.getRoleName());
                });
            }catch(Exception ex){
                ex.printStackTrace();
            }
        } ;
    }
}
```

Résultat

La base de données H2 :

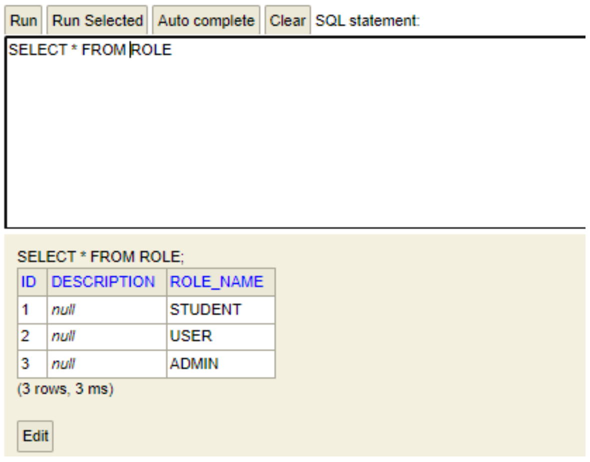


```
SELECT * FROM USER
```

SELECT * FROM USER;

USER_ID	PASSWORD	USER_NAME
faf94e0a-2895-4ef3-9315-6b4f965595f0	123456789	user1
a5ed7fa8-44ef-4e2f-b525-9079c5e7b599	123456789	admin1

(2 rows, 5 ms)

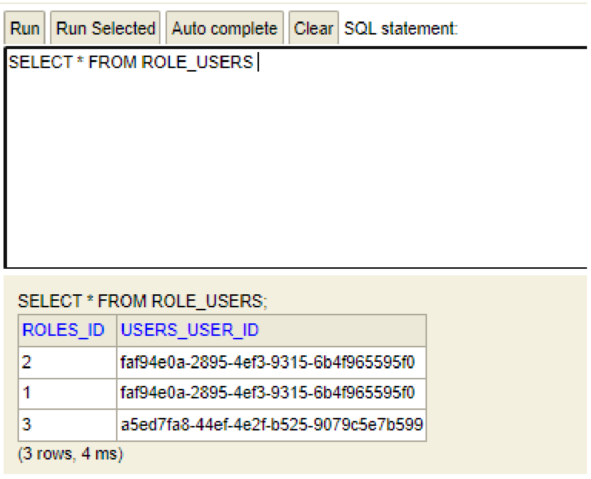


```
SELECT * FROM ROLE
```

SELECT * FROM ROLE;

ID	DESCRIPTION	ROLE_NAME
1	*null*	STUDENT
2	*null*	USER
3	*null*	ADMIN

(3 rows, 3 ms)



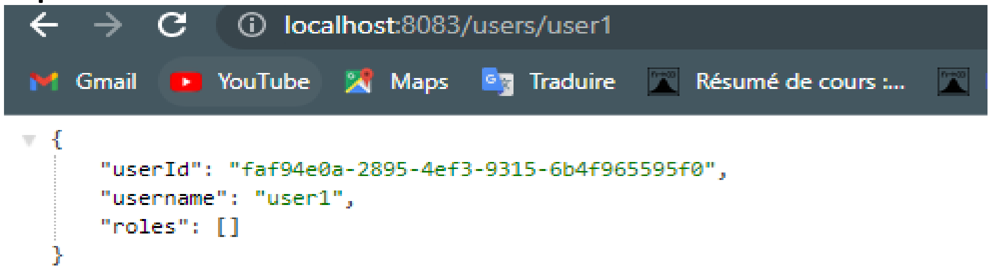
```
SELECT * FROM ROLE_USERS
```

SELECT * FROM ROLE_USERS;

ROLES_ID	USERS_USER_ID
2	faf94e0a-2895-4ef3-9315-6b4f965595f0
1	faf94e0a-2895-4ef3-9315-6b4f965595f0
3	a5ed7fa8-44ef-4e2f-b525-9079c5e7b599

(3 rows, 4 ms)

La partie Web :



```
{
    "userId": "faf94e0a-2895-4ef3-9315-6b4f965595f0",
    "username": "user1",
    "roles": []
}
```