Ecole Normale Supérieure de l'Enseignement Technique Mohammedia

Université Hassan II de Casablanca

ENSET

ا لمدرسة العليا لأساتذة التعليم التقني المحمدية

جامعة الحسن الثاني بالدار البيضاء

Département Mathématiques et Informatique

Cycle Ingénieur

« Glsid2 »

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

Réalisé par: Tarmoun Amal

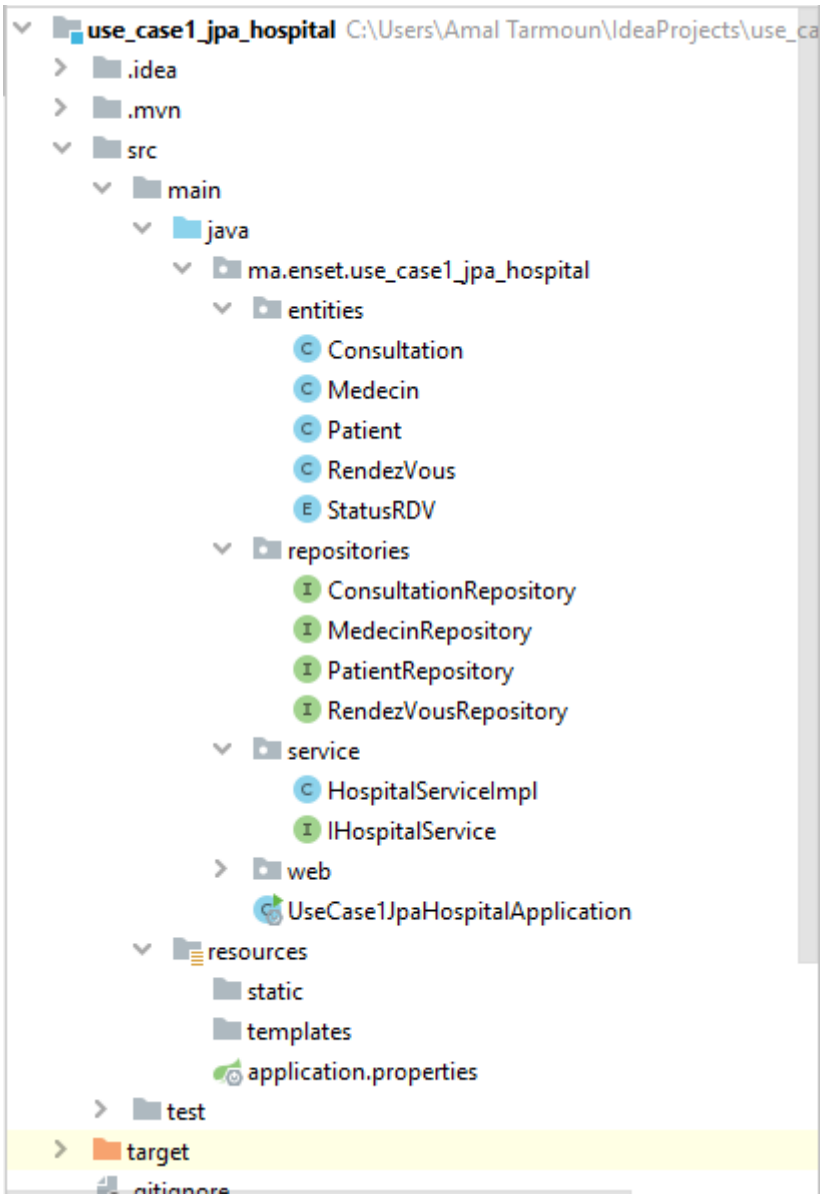
Classe : Glsid2

Encadré par:
Mr. Mohammed EL YOUSSEFI

Année universitaire : 2021 / 2022

Mapping objet relationnel avec JPA, Hibernate et Spring Data

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



Le package Entities :

La classe Patient

```java
package ma.enset.use_case1_jpa_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" )
    private Collection<RendezVous> rendezVous ;
}
```

La classe Medecin

```java
package ma.enset.use_case1_jpa_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")
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
    private Collection<RendezVous> rendezVous;
}
```

La classe Consultation

```java
package ma.enset.use_case1_jpa_hospital.entities;

import ...
@Entity
@Data @NoArgsConstructor @AllArgsConstructor
public class Consultation {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private Date dateConsultation ;
    private String rapport;
    @OneToOne
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
    private RendezVous rendezVous ;
}
```

La classe Rendez-vous

```java
import javax.persistence.*;
import java.util.Date;
@Entity
@Data @NoArgsConstructor @AllArgsConstructor
public class RendezVous {
    @Id
    private String id ;
    @Temporal(TemporalType.DATE)
    private Date date;
    @Enumerated(EnumType.STRING)
    private StatusRDV status;
    @ManyToOne
    // pas de jpa : le mapping objet json : prendre en consideration cette attribut
    // uniquement au  niveau de l'ajout ,
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
    private Patient patient;
    @ManyToOne
    private Medecin medecin;
    @OneToOne(mappedBy = "rendezVous")
    private Consultation consultation;
}

```

Le package Repository

L'interface PatienRepository

```java
package ma.enset.use_case1_jpa_hospital.repositories;

import ma.enset.use_case1_jpa_hospital.entities.Patient;
import org.springframework.data.jpa.repository.JpaRepository;

public interface PatientRepository extends JpaRepository<Patient,Long> {
    Patient findByNom(String name);
}
```

L'interface MedecinRepository

```java
package ma.enset.use_case1_jpa_hospital.repositories;

import ma.enset.use_case1_jpa_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.use_case1_jpa_hospital.repositories;

import ma.enset.use_case1_jpa_hospital.entities.Consultation;
import org.springframework.data.jpa.repository.JpaRepository;

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

L'interface RendezVousRepository

```java
package ma.enset.use_case1_jpa_hospital.repositories;

import ma.enset.use_case1_jpa_hospital.entities.RendezVous;
import org.springframework.data.jpa.repository.JpaRepository;

public interface RendezVousRepository extends JpaRepository<RendezVous,String> {
}
```

Le package Service

L'interface IHospitalService :

```java
package ma.enset.use_case1_jpa_hospital.service;

import ma.enset.use_case1_jpa_hospital.entities.Consultation;
import ma.enset.use_case1_jpa_hospital.entities.Medecin;
import ma.enset.use_case1_jpa_hospital.entities.Patient;
import ma.enset.use_case1_jpa_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
@Service

@Transactional
public class HospitalServiceImpl implements  IHospitalService{

    private PatientRepository patientRepository;
    private MedecinRepository medecinRepository;
    private RendezVousRepository rendezVousRepository;
    private ConsultationRepository consultationRepository;

    public HospitalServiceImpl(PatientRepository patientRepository, MedecinRepository medecinRepository,
                               RendezVousRepository rendezVousRepository, ConsultationRepository consultationRepository) {
        this.patientRepository = patientRepository;
        this.medecinRepository = medecinRepository;
        this.rendezVousRepository = rendezVousRepository;
        this.consultationRepository = consultationRepository;
    }

    @Override
    public Patient savePatient(Patient patient) { return patientRepository.save(patient); }

    @Override
    public Medecin saveMedecin(Medecin medecin) { return medecinRepository.save(medecin); }

    @Override
    public RendezVous saveRDV(RendezVous rendezVous) {
        // generer un string unique .  il depend de la date du systeme
        rendezVous.setId(UUID.randomUUID().toString());
        return rendezVousRepository.save(rendezVous);
    }

    @Override
    public Consultation saveConsultation(Consultation consultation) {
        return consultationRepository.save(consultation);
    }
}
```

Le package web

La classe HospitalResController

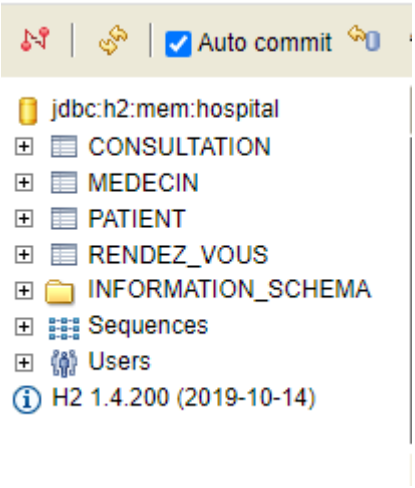
```
1   package ma.enset.use_case1_jpa_hospital.web;
2
3   import ...
10
11  @RestController
12  public class PatientRestController {
13
14      @Autowired
15      private PatientRepository patientRepository ;
16      @GetMapping("/patients")
17      public List<Patient> patientList() { return patientRepository.findAll(); }
20  }
21
```

La classe HospitalApplication :

```java
package ma.enset.use_case1_jpa_hospital;

import ...

@SpringBootApplication
public class UseCase1JpaHospitalApplication {

    public static void main(String[] args) { SpringApplication.run(UseCase1JpaHospitalApplication.class, args); }
    // au demmarage executer cette methode , et il va retourner un objet que spring va le mettre dans son context

    @Bean
    CommandLineRunner start(IHospitalService hospitalService ,
                            PatientRepository patientRepository,RendezVousRepository rendezVousRepository,
                            ConsultationRepository consultationRepository,MedecinRepository medecinRepository){
        return args->{
            Stream.of("amal","kawtar","ikram")
                    .forEach(name->{
                        Patient patient =new Patient();
                        patient.setNom(name);
                        patient.setDateNaissance(new Date());
                        patient.setMalade(false);
                        hospitalService.savePatient(patient);
                    });

            Stream.of("aymane","hanane","ilham")
                    .forEach(name->{
                        Medecin medecin =new Medecin();
                        medecin.setNom(name);
                        medecin.setEmail(name+"@gmail.com");
                        medecin.setSpecialite(Math.random()>0.5?"Cardio":"Dentiste");

                        hospitalService.saveMedecin(medecin);
                    });
            Patient patient =patientRepository.findById(1L).orElse( other: null);
            Patient patient1 =patientRepository.findByNom( name: "amal");
            Medecin medecin =medecinRepository.findByNom("ilham");

            RendezVous rendezVous  = new RendezVous();
            rendezVous.setDate(new Date());
            rendezVous.setStatus(StatusRDV.PENDING);
            rendezVous.setPatient(patient);
            rendezVous.setMedecin(medecin);

            hospitalService.saveRDV(rendezVous);

            RendezVous rendezVous1 = rendezVousRepository.findAll().get(0);

            Consultation consultation = new Consultation();
            consultation.setDateConsultation(new Date());

            consultation.setRendezVous(rendezVous1);
            consultation.setRapport(" Rapport dee la consultation  .........");
            hospitalService.saveConsultation(consultation);

        };
    }

}
```

Résultat :

La base de données H2 :



Run | Run Selected | Auto complete | Clear | SQL statement:

```
SELECT * FROM CONSULTATION
```

SELECT * FROM CONSULTATION;

ID	DATE_CONSULTATION	RAPPORT	RENDEZ_VOUS_ID
1	2022-06-04 20:37:37.983	Rapport dee la consultation	481dce11-ecc6-4a57-b9e3-31ed4be54768

(1 row, 16 ms)

Edit

Run | Run Selected | Auto complete | Clear | SQL statement:

```
SELECT * FROM MEDECIN
```

SELECT * FROM MEDECIN;

ID	EMAIL	NOM	SPECIALITE
1	aymane@gmail.com	aymane	Dentiste
2	hanane@gmail.com	hanane	Cardio
3	ilham@gmail.com	ilham	Dentiste

(3 rows, 8 ms)

Run | Run Selected | Auto complete | Clear | SQL statement:

```
SELECT * FROM PATIENT
```

SELECT * FROM PATIENT;

ID	DATE_NAISSANCE	MALADE	NOM
1	2022-06-04	FALSE	amal
2	2022-06-04	FALSE	kawtar
3	2022-06-04	FALSE	ikram

(3 rows, 0 ms)

Edit

Run | Run Selected | Auto complete | Clear | SQL statement:

```
SELECT * FROM RENDEZ_VOUS
```

SELECT * FROM RENDEZ_VOUS;

ID	DATE	STATUS	MEDECIN_ID	PATIENT_ID
481dce11-ecc6-4a57-b9e3-31ed4be54768	2022-06-04	PENDING	3	1

(1 row, 0 ms)

Edit

La partie Web :

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 :

Run | Run Selected | Auto complete | Clear | SQL statement:

```
SELECT * FROM ROLE
```

SELECT * FROM ROLE;

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

(3 rows, 3 ms)

Edit

Run | Run Selected | Auto complete | Clear | SQL statement:

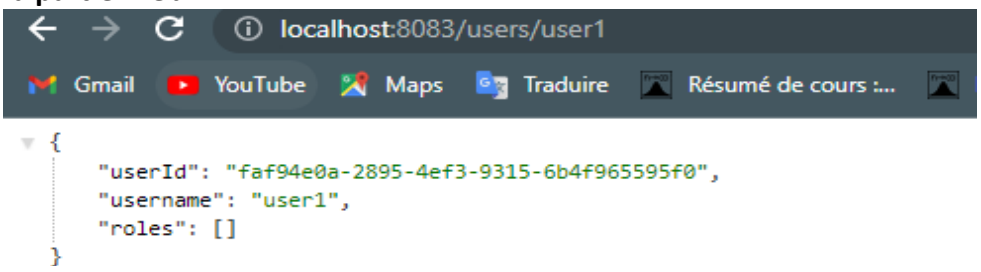
```
SELECT * FROM ROLE_USERS
```

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": []
}
```