

**DEPARTEMENT MATHÉMATIQUES ET INFORMATIQUE**

# Compte Rendu

**Filière :**  
**« Génie du Logiciel et des Systèmes Informatiques Distribués »**  
**GLSID**

## Activité Pratique N°2 - JPA, Hibernate et Spring Data

**le 20/3/2022**

Réalisé par :

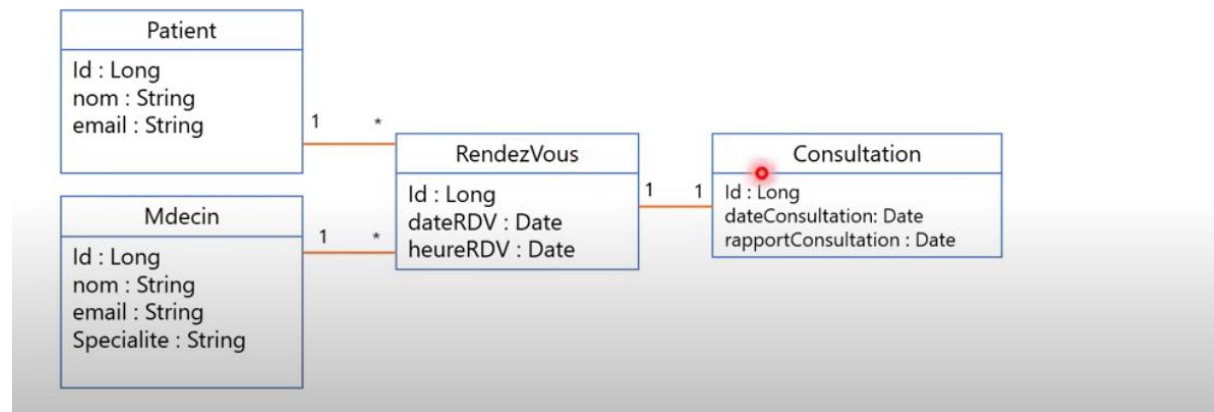
Hassan ELMAKHLOUFI

Encadré par :

M. Mohamed YOUSFI

**Année Universitaire : 2021-2022**

Dans ce Tp Je vais implémenter le diagramme de classe suivant en utilisant Spring boot (spring data):



## Entités

Entités patientes :

```
1 package ma.enset.hospital.entities;
2
3 import com.fasterxml.jackson.annotation.JsonProperty;
4 import lombok.Data;
5 import org.springframework.boot.autoconfigure.web.WebProperties;
6
7 import javax.persistence.*;
8 import java.util.Collection;
9 import java.util.Date;
10
11 @Data
12 @Entity
13 public class Patient {
14     @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private Long id ;
16     private String nom ;
17     private String email;
18     @Temporal(TemporalType.DATE)
19     private Date dateNaissance ;
20     private boolean malade;
21     @OneToMany(mappedBy = "patient", fetch = FetchType.LAZY)
22     @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
23     Collection<RendezVous> rendezVous ;
24
25 }
26
```

## Entités Médecin :

```
C Patient.java x C Medecin.java x I IHospital.java x C Hospitalimp.java x I Rendezvous
4 import lombok.AllArgsConstructor;
5 import lombok.Data;
6 import lombok.NoArgsConstructor;
7
8 import javax.persistence.*;
9 import java.util.Collection;
10
11 @Entity
12 @Data
13 @NoArgsConstructor
14 @AllArgsConstructor
15 public class Medecin {
16     @Id
17     @GeneratedValue(strategy = GenerationType.IDENTITY)
18     private Long id ;
19     private String nom ;
20     private String email;
21     private String Specialite;
22     @OneToMany(mappedBy = "medecin", fetch = FetchType.LAZY)
23     @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
24     private Collection<Rendezvous> rendezvous;
25
26 }
27
```

## Entités Consultation :

```
C Consultation.java x
1 package ma.enset.hospital.entities;
2
3 import lombok.AllArgsConstructor;
4 import lombok.Data;
5 import lombok.NoArgsConstructor;
6
7 import javax.persistence.*;
8 import java.util.Date;
9
10 @Entity @Data @AllArgsConstructor @NoArgsConstructor
11 public class Consultation {
12
13     @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
14     private Long id ;
15     private Date dateConsultation ;
16     private String rapport;
17     @OneToOne
18     private Rendezvous rendezvous;
19
20 }
```

## Entités Rendez-vous :

```
Consultation.java x RendezVous.java x
4 import lombok.Data;
5 import lombok.NoArgsConstructor;
6
7 import javax.persistence.*;
8 import java.util.Date;
9
10 @Data
11 @Entity @AllArgsConstructor @NoArgsConstructor
12 public class RendezVous {
13
14     @Id
15     private String id ;
16     private Date date ;
17     @Enumerated(EnumType.STRING)
18     private StatusRDV statusRDV ;
19     @ManyToOne
20     private Patient patient ;
21     @ManyToOne
22     private Medecin medecin ;
23     @OneToOne(mappedBy = "rendezVous")
24     private Consultation consultation;
25
26 }
27
```

## Repositories :

### Repository Patient :

```
Consultation.java x RendezVous.java x ConsultationRepository.java x PatientRepository.java x
1 package ma.enset.hospital.repositories;
2
3 import ma.enset.hospital.entities.Patient;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface PatientRepository extends JpaRepository<Patient,Long> {
7     Patient findByNom(String nom);
8 }
```

## Repository Médecin :

```
1 package ma.enset.hospital.repositories;
2
3 import ma.enset.hospital.entities.Medecin;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface MedecinRepository extends JpaRepository<Medecin, Long> {
7 }
8
```

## Repository Consultation :

```
1 package ma.enset.hospital.repositories;
2
3 import ma.enset.hospital.entities.Consultation;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface ConsultationRepository extends JpaRepository<Consultation, Long> {
7 }
8
```

## Repository Rendez-vous :

```
1 package ma.enset.hospital.repositories;
2
3 import ma.enset.hospital.entities.RendezVous;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface RendezvousRepository extends JpaRepository<RendezVous, String> {
7 }
8
```

## Métiers :

### L'implémentation métier de l'application :

```
IHospital.java x
1 package ma.enset.hospital.service;
2
3 import ma.enset.hospital.entities.Consultation;
4 import ma.enset.hospital.entities.Medecin;
5 import ma.enset.hospital.entities.Patient;
6 import ma.enset.hospital.entities.RendezVous;
7
8 public interface IHospital {
9
10     Patient savePatient(Patient patient);
11     Medecin saveMedecin(Medecin medecin);
12     RendezVous saveRendezvous(RendezVous rendezVous);
13     Consultation saveConsultation(Consultation consultation);
14 }
```

### Une implémentation de l'interface métier :

```
IHospital.java x Hospitalimp.java x
15 import java.util.UUID;
16
17 @Service
18 @Transactional
19 @AllArgsConstructor
20 public class Hospitalimp implements IHospital{
21     private PatientRepository patientRepository;
22     private MedecinRepository medecinRepository;
23     private RendezvousRepository rendezvousRepository;
24     private ConsultationRepository consultationRepository;
25
26     @Override
27     public Patient savePatient(Patient patient) { return patientRepository.save(patient); }
28
29
30     @Override
31     public Medecin saveMedecin(Medecin medecin) { return medecinRepository.save(medecin); }
32
33
34     @Override
35     public RendezVous saveRendezvous(RendezVous rendezVous) {
36         rendezVous.setId(UUID.randomUUID().toString());
37         return rendezvousRepository.save(rendezVous);
38     }
39
40
41     @Override
42     public Consultation saveConsultation(Consultation consultation) {
43         return consultationRepository.save(consultation);
44     }
45 }
```

Plugin supporting dependenc

## Application :

Dans cette classe j'ai testé l'application par l'instanciation des classes que j'ai créées puis je l'ai persistée dans la base de données H2

```
import java.util.Date;
import java.util.stream.Stream;

@SpringBootApplication
public class HospitalApplication {

    public static void main(String[] args) { SpringApplication.run(HospitalApplication.class, args); }

    @Bean
    CommandLineRunner start(HospitalImp hospitalImp, PatientRepository patientRepository, MedecinRepository medecinRepository) {
        return args -> {
            Stream.of("Mohammed", "hassan", "najat").forEach(name -> {
                Patient patient = new Patient();
                patient.setNom(name);
                patient.setDateNaissance(new Date());
                patient.setMalade(false);
                patientRepository.save(patient);
            });

            Stream.of("Ilyass", "Hanan", "Ilham").forEach(name -> {
                Medecin medecin = new Medecin();
                medecin.setNom(name);
                medecin.setEmail(name + "@gmail.com");
                medecin.setSpecialite(Math.random() > 0.5 ? "cardio" : "Dentiste");
            });
        };
    }
}
```