

Rapport TP2 - JEE

Cas de Patient, Médecin, Rendez-vous, Consultation

1. Entities :

```
Consultation.java x
1  package ma.enset.tp2.entities;
2
3  import ...
10
11  @Entity
12  @Data @NoArgsConstructor @AllArgsConstructor
13  public class Consultation {
14      @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
15      private Long id;
16      private Date dateConsultation;
17      private String rapport;
18      @OneToOne
19      @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
20      private RendezVous rendezVous;
21  }
```

```
Medecin.java x
1  package ma.enset.tp2.entities;
2
3  import ...
10
11  @Entity
12  @Data @NoArgsConstructor @AllArgsConstructor
13  public class Medecin {
14      @Id
15      @GeneratedValue(strategy = GenerationType.IDENTITY)
16      private Long id;
17      private String nom;
18      private String email;
19      private String speciality;
20      @OneToMany(mappedBy = "medecin", fetch = FetchType.LAZY)
21      @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
22      private Collection<RendezVous> rendezVous;
23  }
```

```
Patient.java x
1 package ma.enset.tp2.entities;
2
3 import ...
10
11 @Entity
12 @Data @NoArgsConstructor @AllArgsConstructor
13 public class Patient {
14     @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private Long id;
16     @Column(length = 50)
17     private String nom;
18     @Temporal(TemporalType.DATE)
19     private Date dateNaissance;
20     private boolean malade;
21     @OneToMany(mappedBy = "patient", fetch = FetchType.LAZY)
22     private Collection<RendezVous> rendezVous;
23 }
```

```
RendezVous.java x
1 package ma.enset.tp2.entities;
2
3 import ...
10
11 @Entity
12 @Data @NoArgsConstructor @AllArgsConstructor
13 public class RendezVous {
14     @Id
15     private String id;
16     private Date date;
17     @Enumerated(EnumType.STRING)
18     private StatusRDV status;
19     @ManyToOne
20     @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
21     private Patient patient;
22     @ManyToOne
23     private Medecin medecin;
24     @OneToOne(mappedBy = "rendezVous")
25     private Consultation consultation;
26 }
```

```
StatusRDV.java x
1 package ma.enset.tp2.entities;
2
3 public enum StatusRDV {
4     PENDING,
5     CANCELED,
6     DONE
7 }
```

2. Repositories :

```
ConsultationRepository.java ×
1 package ma.enset.tp2.repositories;
2
3 import ma.enset.tp2.entities.Consultation;
4 import org.springframework.data.jpa.repository.JpaRepository;
5 import java.util.Date;
6
7 public interface ConsultationRepository extends JpaRepository<Consultation,Long> {
8     Consultation findConsultationByDateConsultation(Date date);
9 }
```

```
MedecinRepository.java ×
1 package ma.enset.tp2.repositories;
2
3 import ma.enset.tp2.entities.Medecin;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface MedecinRepository extends JpaRepository<Medecin,Long> {
7     Medecin findByName(String name);
8 }
```

```
PatientRepository.java ×
1 package ma.enset.tp2.repositories;
2
3 import ma.enset.tp2.entities.Patient;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface PatientRepository extends JpaRepository<Patient,Long> {
7     Patient findByName(String name);
8 }
```

```
RendezVousRepository.java ×
1 package ma.enset.tp2.repositories;
2
3 import ma.enset.tp2.entities.RendezVous;
4 import org.springframework.data.jpa.repository.JpaRepository;
5 import java.util.Date;
6
7 public interface RendezVousRepository extends JpaRepository<RendezVous,String> {
8     RendezVous findRendezVousByDate(Date date);
9 }
```

3. Service :

```
IHospitalService.java x
1 package ma.enset.tp2.service;
2
3 import ...
7
8 public interface IHospitalService {
9     Patient savePatient(Patient patient);
10    Medecin saveMedecin(Medecin medecin);
11    RendezVous saveRDV(RendezVous rendezVous);
12    Consultation saveConsultation(Consultation consultation);
13 }
```

```
HospitalServiceImpl.java x
15 @Service
16 @Transactional
17 public class HospitalServiceImpl implements IHospitalService {
18     private PatientRepository patientRepository;
19     private MedecinRepository medecinRepository;
20     private RendezVousRepository rendezVousRepository;
21     private ConsultationRepository consultationRepository;
22
23     public HospitalServiceImpl(PatientRepository patientRepository, MedecinRepository medecinRepository,
24                               RendezVousRepository rendezVousRepository, ConsultationRepository consultationRepository) {
25         this.patientRepository = patientRepository;
26         this.medecinRepository = medecinRepository;
27         this.rendezVousRepository = rendezVousRepository;
28         this.consultationRepository = consultationRepository;
29     }
30
31     @Override
32     public Patient savePatient(Patient patient) { return patientRepository.save(patient); }
35
36     @Override
37     public Medecin saveMedecin(Medecin medecin) { return medecinRepository.save(medecin); }
40
41     @Override
42     public RendezVous saveRDV(RendezVous rendezVous) {
43         rendezVous.setId(UUID.randomUUID().toString());
44         return rendezVousRepository.save(rendezVous);
45     }
46
47     @Override
48     public Consultation saveConsultation(Consultation consultation) {
49         return consultationRepository.save(consultation);
50     }
}
```

4. Web :

```
PatientRestController.java x
1 package ma.enset.tp2.web;
2
3 import ...
10
11 @RestController
12 public class PatientRestController {
13     @Autowired
14     private PatientRepository patientRepository;
15
16     @GetMapping("/patients")
17     public List<Patient> patientList() { return patientRepository.findAll(); }
18
19 }
```

5. Application :

```
HospitalApp.java x
1 package ma.enset.tp2;
2 import ...
13
14 @SpringBootApplication
15 public class HospitalApp {
16     public static void main(String[] args) {
17         SpringApplication.run(HospitalApp.class, args);
18     }
19
20 @Bean
21 CommandLineRunner start(IHospitalService hospitalService,
22     PatientRepository patientRepository,
23     MedecinRepository medecinRepository,
24     RendezVousRepository rendezVousRepository) {
25     return args -> {
26         Stream.of("Mohamed", "Amine", "Houssine")
27             .forEach(name->{
28                 Patient patient = new Patient();
29                 patient.setNom(name);
30                 patient.setDateNaissance(new Date());
31                 patient.setMalade(false);
32                 hospitalService.savePatient(patient);
33             });
34         Stream.of("Ahmed", "Ali", "Mustapha")
35             .forEach(name->{
36                 Medecin medecin = new Medecin();
37                 medecin.setNom(name);
38                 medecin.setEmail(name+"@hospital.com");
39                 medecin.setSpeciality(Math.random()>0.5?"Cardio":"Dentiste");
40                 hospitalService.saveMedecin(medecin);
41             });
42     }
```

```

43 //Patient p=patientRepository.findById(1L).orElse(null);
44 Patient p=patientRepository.findByNom( name: "Amine");
45 Medecin m=medecinRepository.findByNom( name: "Mustapha");
46
47 RendezVous rdv = new RendezVous();
48 rdv.setDate(new Date());
49 rdv.setStatus(StatusRDV.PENDING);
50 rdv.setMedecin(m);
51 rdv.setPatient(p);
52 RendezVous savedRdv = hospitalService.saveRDV(rdv);
53 System.out.println(savedRdv.getId());
54
55 RendezVous rendezVous = rendezVousRepository.findAll().get(0);
56 Consultation consultation = new Consultation();
57 consultation.setDateConsultation(new Date());
58 consultation.setRendezVous(rendezVous);
59 consultation.setRapport("Rapport de la consultation...");
60 hospitalService.saveConsultation(consultation);
61 };
62 }
63 }

```

6. Propriétés d'Application :

```

</> application.properties x
1 # Using H2 Database
2 spring.datasource.url=jdbc:h2:mem:hospitalDb
3 spring.h2.console.enabled=true
4 server.port=8086

```

7. Exécution de l'Application :

H2 Console interface showing a SQL query execution. The query is `SELECT * FROM CONSULTATION;`. The result set contains one row with columns ID, DATE_CONSULTATION, RAPPORT, and RENDEZ_VOUS_ID.

ID	DATE_CONSULTATION	RAPPORT	RENDEZ_VOUS_ID
1	2022-04-18 01:33:40.394	Rapport de la consultation...	3cb4d8ca-5402-4324-a47e-1a0aaf65eec3

H2 Console interface showing a SQL query execution. The query is `SELECT * FROM MEDECIN;`. The result set contains three rows with columns ID, EMAIL, NOM, and SPECIALITY.

ID	EMAIL	NOM	SPECIALITY
1	Ahmed@hospital.com	Ahmed	Cardio
2	Ali@hospital.com	Ali	Cardio
3	Mustapha@hospital.com	Mustapha	Dentiste

H2 Console interface showing a SQL query execution. The query is `SELECT * FROM PATIENT;`. The result set contains three rows with columns ID, DATE_NAISSANCE, MALADE, and NOM.

ID	DATE_NAISSANCE	MALADE	NOM
1	2022-04-18	FALSE	Mohamed
2	2022-04-18	FALSE	Amine
3	2022-04-18	FALSE	Houssine

H2 Console

localhost:8086/h2-console/login.do?sessionId=ebe5d2898fb00944df229e0aee9c9218

Auto commit Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:mem:hospitalDb

- CONSULTATION
- MEDECIN
- PATIENT
- RENDEZ_VOUS
- INFORMATION_SCHEMA
- Sequences
- Users
- H2 1.4.200 (2019-10-14)

SQL statement:

```
SELECT * FROM RENDEZ_VOUS
```

SELECT * FROM RENDEZ_VOUS;

ID	DATE	STATUS	MEDECIN_ID	PATIENT_ID
3cb4d8ca-5402-4324-a47e-1a0aaf65eec3	2022-04-18 01:33:40.363	PENDING	3	2

(1 row, 3 ms)

Edit

localhost:8086/patients

localhost:8086/patients

JSON Données brutes En-têtes

Enregistrer Copier Tout réduire Tout développer Filtrer le JSON

```

0:
  id: 1
  nom: "Mohamed"
  dateNaissance: "2022-04-18"
  malade: false
  rendezVous: []
1:
  id: 2
  nom: "Amine"
  dateNaissance: "2022-04-18"
  malade: false
  rendezVous:
    0:
      id: "46e1a3fc-f4c7-47eb-b719-04486d0eff35"
      date: "2022-04-18T04:17:08.931+00:00"
      status: "PENDING"
      medecin:
        id: 3
        nom: "Mustapha"
        email: "Mustapha@hospital.com"
        speciality: "Dentiste"
      consultation:
        id: 1
        dateConsultation: "2022-04-18T04:17:08.962+00:00"
        rapport: "Rapport de la consultation..."
2:
  id: 3
  nom: "Houssine"
  dateNaissance: "2022-04-18"
  malade: false
  rendezVous: []

```