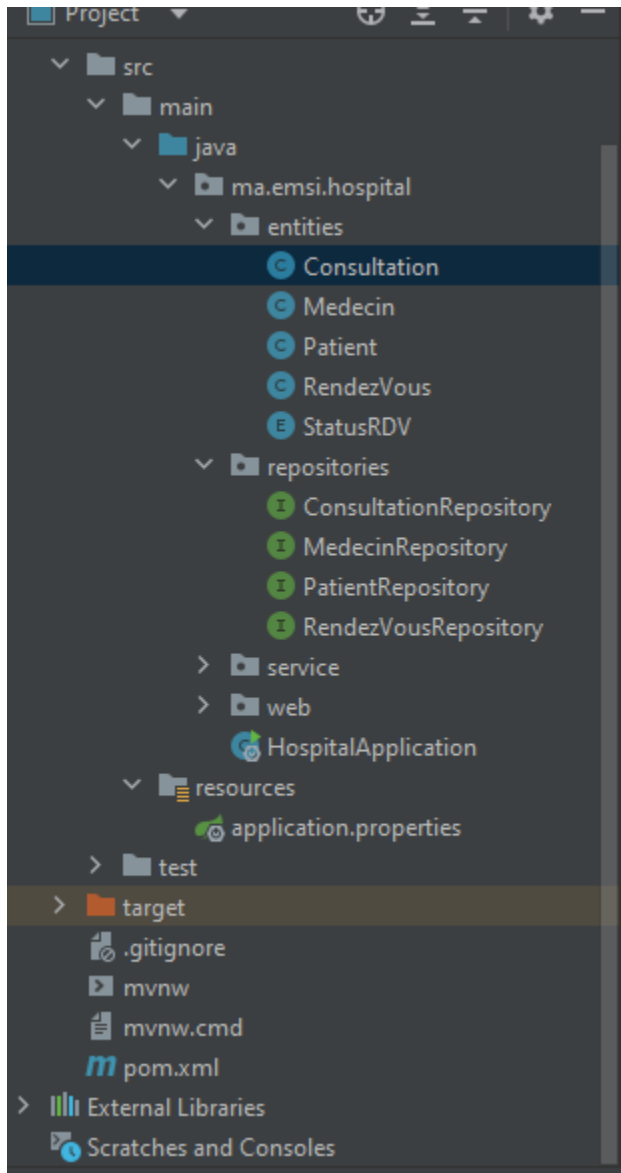


## Acitivite 2 JEE/ Spring Boot - ORM avec Spring Data JPA

Hicham Yassine 4IIR G4

Partie 2 APP hospital



```
spring.datasource.url=jdbc:h2:mem:hospital
spring.h2.console.enabled=true
server.port=8086
```

## Les class

```
Medecin.java x Patient.java x RendezVous.java x application.properties x
1 package ma.emsi.hospital.entities;
2 import ...
   13 usages
8 @Entity
9 @Data @NoArgsConstructor @AllArgsConstructor
10 public class Medecin {
11     @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
12     private Long id;
13     private String nom;
14     private String email;
15     private String specialite;
16     @OneToMany(mappedBy = "medecin", fetch = FetchType.LAZY)
17     @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
18     private Collection<RendezVous> rendezVous;
19 }
20
```

```
tion.properties x HospitalApplication.java x StatusRDV.java x Consultation.java x
1 package ma.emsi.hospital.entities;
2 import ...
   11 usages
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 }
```

```
Patient.java x RendezVous.java x application.properties x HospitalApplication. v
5 import lombok.Data;
6 import lombok.NoArgsConstructor;
7
8 import java.util.Collection;
9 import java.util.Date;
10
11 16 usages
12 @Entity
13 @Data @NoArgsConstructor @AllArgsConstructor
14 public class Patient {
15     @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
16     private Long id;
17     private String nom;
18     @Temporal(TemporalType.DATE)
19     private Date dateNaissance;
20     private boolean malade;
21
22     @OneToMany(mappedBy = "patient", fetch = FetchType.LAZY)
23     private Collection<RendezVous> rendezVous;
24 }
```

```
Patient.java x RendezVous.java x application.properties x HospitalApplication. v
1 package ma.emsi.hospital.entities;
2 import ...
3
4 15 usages
5 @Entity
6 @Data @NoArgsConstructor @AllArgsConstructor
7 public class RendezVous {
8     @Id // @GeneratedValue(strategy = GenerationType.IDENTITY)
9     private String id;
10     private Date date;
11     @Enumerated(EnumType.STRING)
12     private StatusRDV status;
13
14     @ManyToOne
15     @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
16     private Patient patient;
17
18     @ManyToOne
19     private Medecin medecin;
20
21     @OneToOne(mappedBy = "rendezVous")
22     private Consultation consultation;
23
24 }
```

```
1 package ma.emsi.hospital.entities;
2
3 public enum StatusRDV {
4     PENDING,
5     CANCELED,
6     DONE
7 }
8
```

Les repository

```
1 package ma.emsi.hospital.repositories;
2
3 import ma.emsi.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 }
9
```

Main

```

HospitalApplication.java x Patient.java x StatusRDV.java x ConsultationRepository.java x PatientRepository.java x RendezVous.java x application.pr
10 import org.springframework.boot.autoconfigure.SpringBootApplication;
11 import org.springframework.context.annotation.Bean;
12
13 import java.util.Date;
14 import java.util.stream.Stream;
15 @SpringBootApplication
16 public class HospitalApplication {
17     public static void main(String[] args) { SpringApplication.run(HospitalApplication.class, args); }
18
19     @Bean
20     CommandLineRunner start(IHospitalService hospitalService,
21                             PatientRepository patientRepository,
22                             RendezVousRepository rendezVousRepository,
23                             MedecinRepository medecinRepository){
24
25         return args -> {
26             Stream.of(...values: "yassine", "ismail", "amine")
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
35             Stream.of(...values: "walid", "aya", "fahd")
36                     .forEach(name ->{
37                         Medecin medecin = new Medecin();
38                         medecin.setNom(name);
39                         medecin.setSpecialite(Math.random()>0.5?"gero":"Dentiste");
40                         medecin.setEmail(name+"@gmail.com");
41                         hospitalService.saveMedecin(medecin);
42                     });
43         };
44     }
45 }

```

## H2 database tests

☒ Auto commit
 ☐ Max rows: 1000
 ☐ Auto complete Off

jdbc:h2:mem:hospital
 

+ CONSULTATION
 + MEDECIN
 - PATIENT
 + ID
 + DATE\_NAISSANCE
 + MALADE
 + NOM
 + Indexes
 + RENDEZ\_VOUS
 + INFORMATION\_SCHEMA
 + Users
 + H2 2.1.214 (2022-06-13)

Run Run Selected Auto complete Clear SQL statement:

SELECT \* FROM PATIENT PATIENT

SELECT \* FROM PATIENT PATIENT;
 

ID	DATE_NAISSANCE	MALADE	NOM
1	2023-04-17	FALSE	yassine
2	2023-04-17	FALSE	ismail
3	2023-04-17	FALSE	amine

 (3 rows, 5 ms)
 

Edit

jdbc:h2:mem:hospital

- CONSULTATION
- MEDECIN
  - ID
  - EMAIL
  - NOM
  - SPECIALITE
  - Indexes
- PATIENT
  - ID
  - DATE\_NAISSANCE
  - MALADE
  - NOM
  - Indexes
- RENDEZ\_VOUS
- INFORMATION\_SCHEMA
- Users
- H2 2.1.214 (2022-06-13)

Run Run Selected Auto complete Clear SQL statement:

SELECT \* FROM MEDECIN |

SELECT \* FROM MEDECIN;

ID	EMAIL	NOM	SPECIALITE
1	walid@gmail.com	walid	Dentiste
2	aya@gmail.com	aya	Dentiste
3	fahd@gmail.com	fahd	Dentiste

(3 rows, 2 ms)

Edit

```

1 package ma.emsi.hospital.service;
2
3 import ...
4
5
6
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 }

```

```
IHospitalServiceImpl.java x RendezVous.java x application.properties x IHospitalService.java v
private PatientRepository patientRepository;
2 usages
private MedecinRepository medecinRepository;
2 usages
private RendezVousRepository rendezVousRepository;
2 usages
private ConsultationRepository consultationRepository;
public IHospitalServiceImpl(PatientRepository patientRepository,
                           MedecinRepository medecinRepository,
                           RendezVousRepository rendezVousRepository,
                           ConsultationRepository consultationRepository) {
    this.patientRepository = patientRepository;
    this.medecinRepository = medecinRepository;
    this.rendezVousRepository = rendezVousRepository;
    this.consultationRepository = consultationRepository;
}
1 usage
@Override
public Patient savePatient(Patient patient) {
    return patientRepository.save(patient);
}
1 usage
@Override
public Medecin saveMedecin(Medecin medecin) {
    return medecinRepository.save(medecin);
}
1 usage
@Override
public RendezVous saveRDV(RendezVous rendezVous) {
    rendezVous.setId(UUID.randomUUID().toString());
    return rendezVousRepository.save(rendezVous);
}
1 usage
@Override
public Consultation saveConsultation(Consultation consultation) {
    return consultationRepository.save(consultation);
}
```