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

Filière :

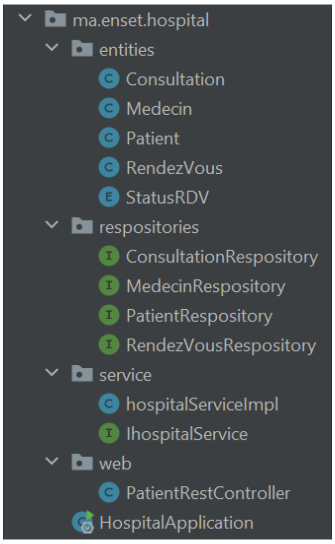
« Ingénierie Informatique : Big Data et Cloud Computing »

II-BDCC

TP 2: Jpa, Hibernate et Spring Data : Part 1

Réalisé Par: Laila Sad Elbouyoud
Classe: ii-BDCC 2

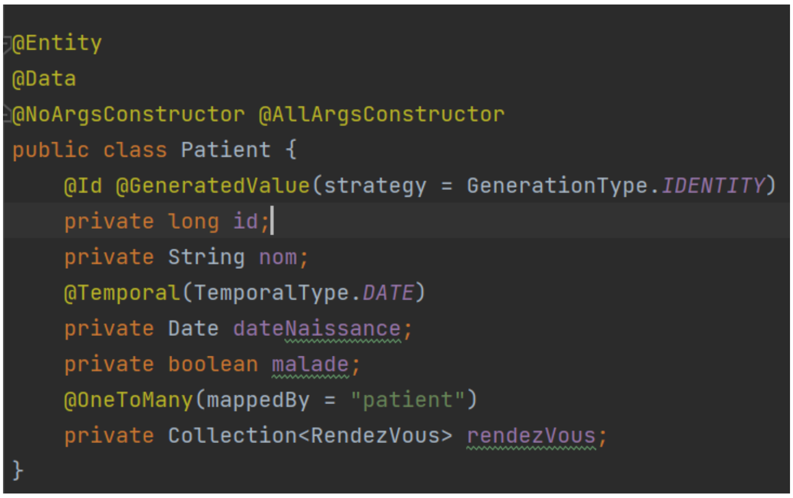
1. Hierarchie du projet :



```
ma.enset.hospital
    entities
        Consultation
        Medecin
        Patient
        RendezVous
        StatusRDV
    respositories
        ConsultationRespository
        MedecinRespository
        PatientRespository
        RendezVousRespository
    service
        hospitalServiceImpl
        IhospitalService
    web
        PatientRestController
    HospitalApplication
```

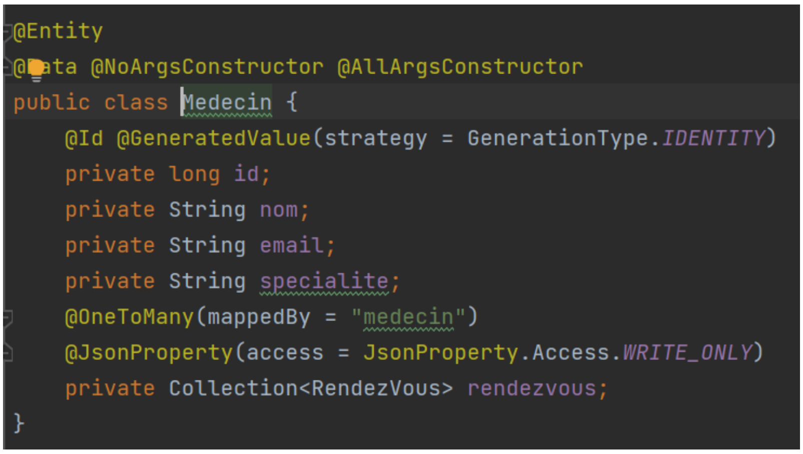
2. Les entités :

a. classe Patient :



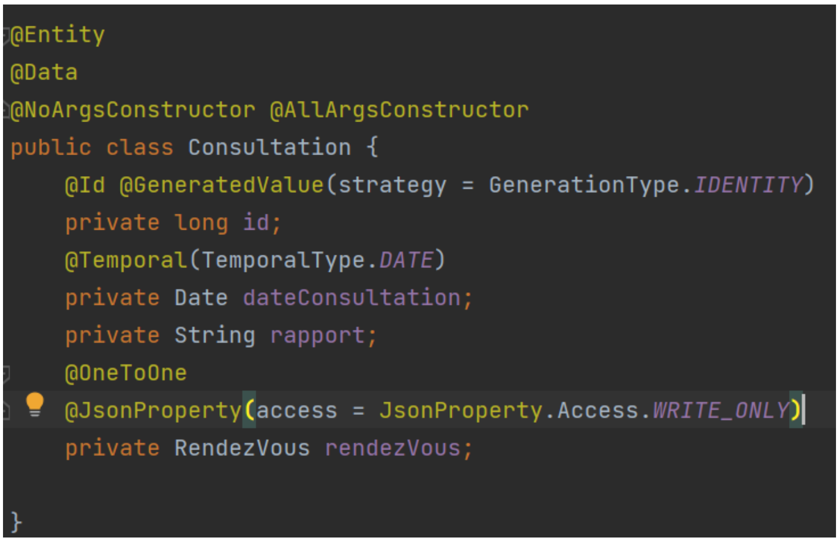
```java
@Entity
@Data
@NoArgsConstructor @AllArgsConstructor
public class Patient {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;
    private String nom;
    @Temporal(TemporalType.DATE)
    private Date dateNaissance;
    private boolean malade;
    @OneToMany(mappedBy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

b. classe Médecin :



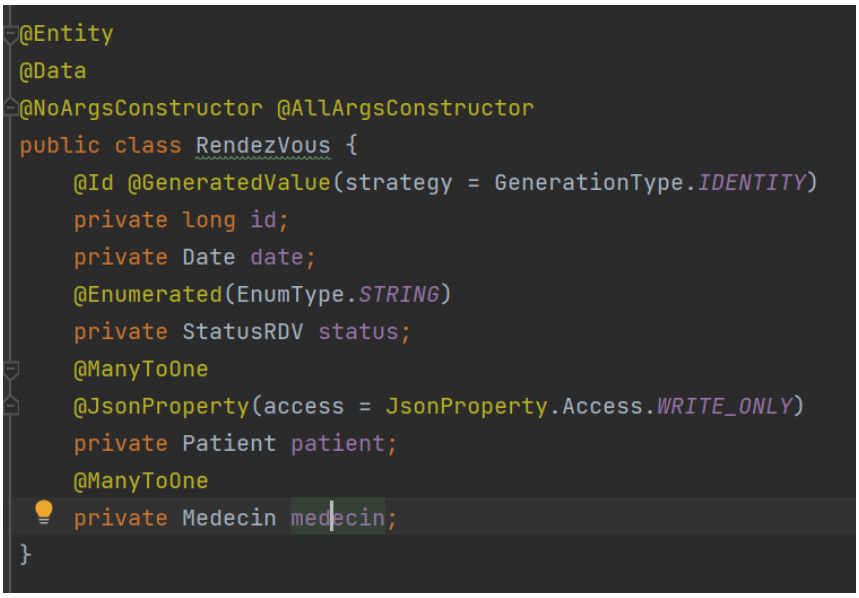
```java
@Entity
@Data @NoArgsConstructor @AllArgsConstructor
public class Medecin {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;
    private String nom;
    private String email;
    private String specialite;
    @OneToMany(mappedBy = "medecin")
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
    private Collection<RendezVous> rendezvous;
}
```

c. classe Consultation :



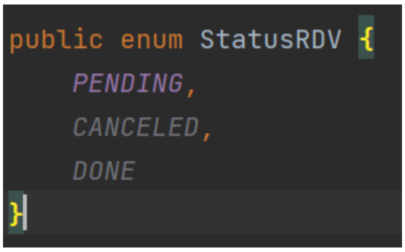
```java
@Entity
@Data
@NoArgsConstructor @AllArgsConstructor
public class Consultation {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;
    @Temporal(TemporalType.DATE)
    private Date dateConsultation;
    private String rapport;
    @OneToOne
    @JsonProperty(access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

}
```

d. classe Rendez Vous :



```java
@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;
}
```

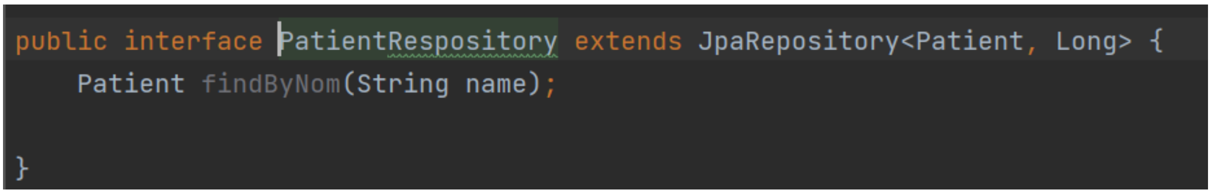
e. Enumération Status :



```java
public enum StatusRDV {
    PENDING,
    CANCELED,
    DONE
}
```

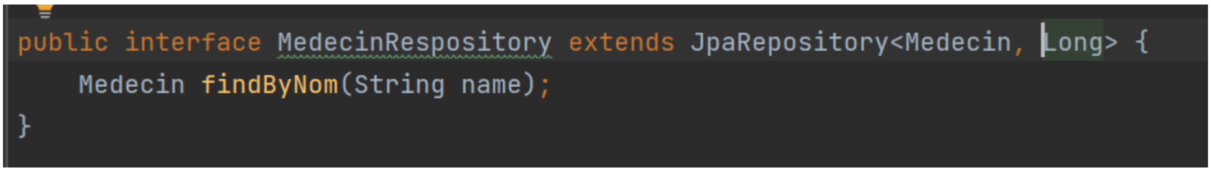
3. Respositories :

a. PatientRespository:



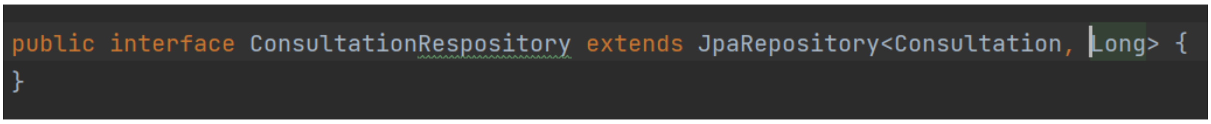
```
public interface PatientRespository extends JpaRepository<Patient, Long> {
    Patient findByNom(String name);

}
```

b. MedecinRespository:



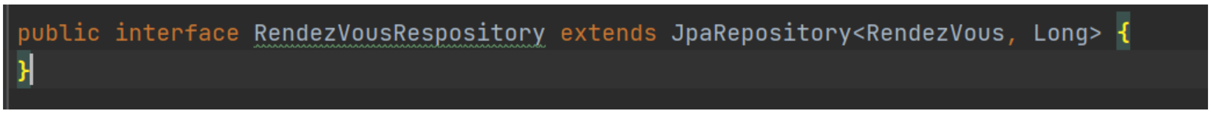
```
public interface MedecinRespository extends JpaRepository<Medecin, Long> {
    Medecin findByNom(String name);
}
```

c. ConsultationRespository:



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

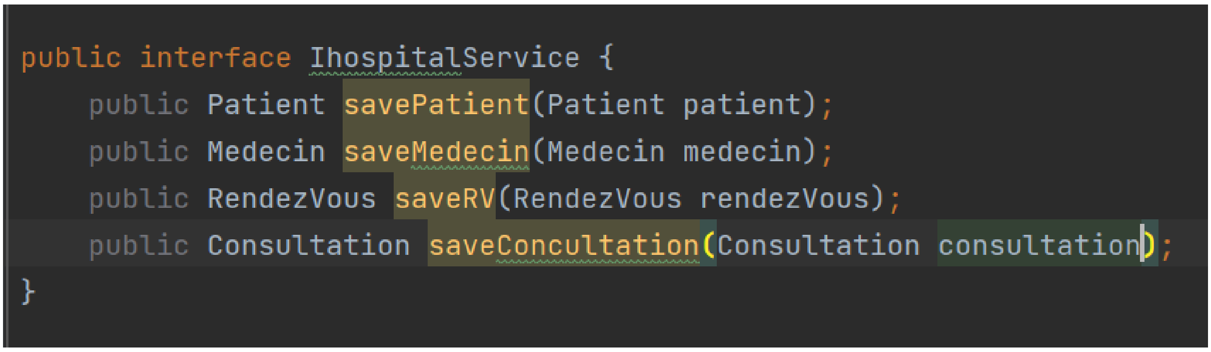
d. RendezVousRespository:



```
public interface RendezVousRespository extends JpaRepository<RendezVous, Long> {
}
```

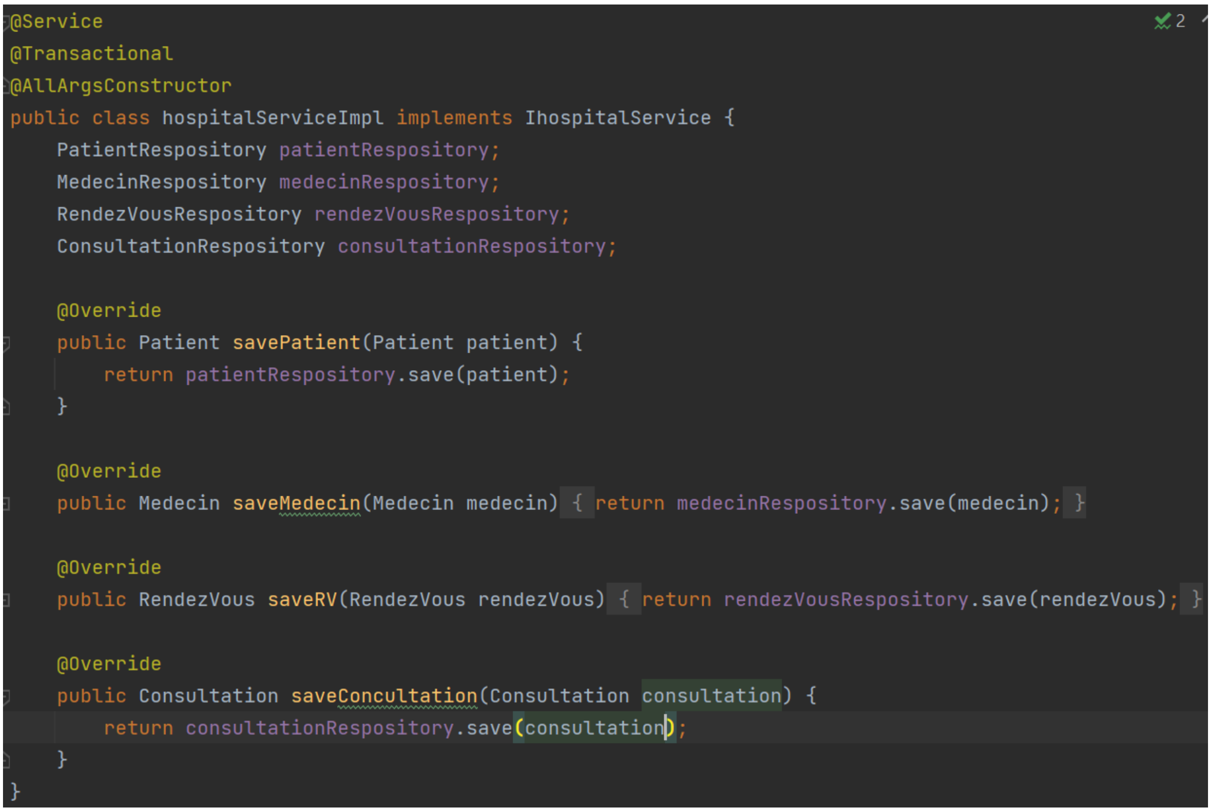
4. Couche Service:

a. Interface Service:



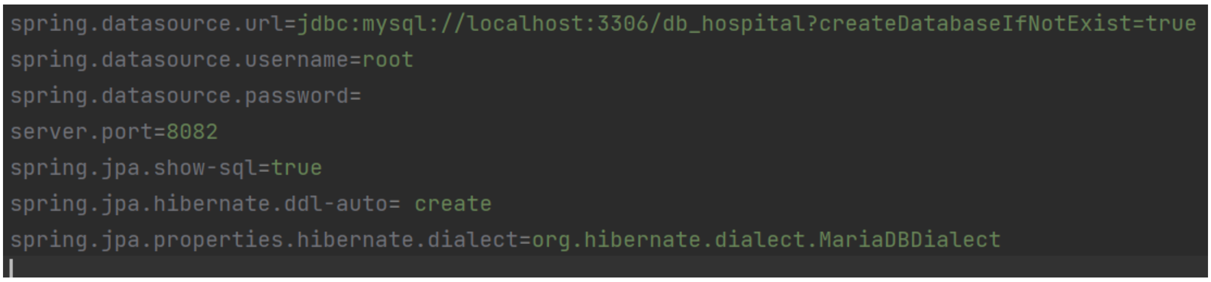
```
public interface IhospitalService {
    public Patient savePatient(Patient patient);
    public Medecin saveMedecin(Medecin medecin);
    public RendezVous saveRV(RendezVous rendezVous);
    public Consultation saveConcultation(Consultation consultation);
}
```

b. Implementation de l’interface Service :



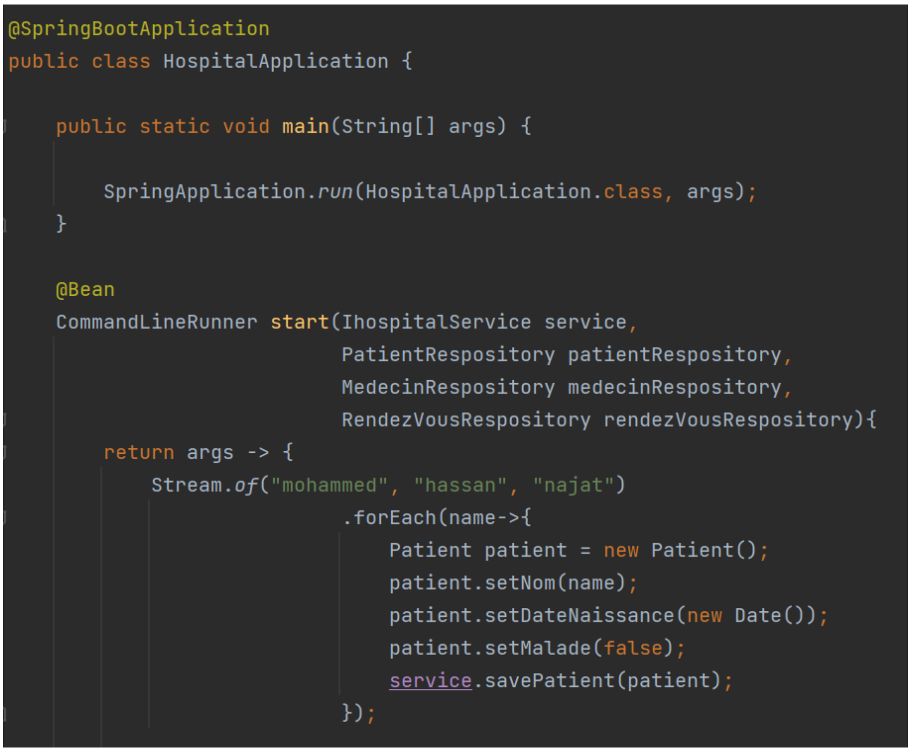
```
@Service
@Transactional
@AllArgsConstructor
public class hospitalServiceImpl implements IhospitalService {
    PatientRespository patientRespository;
    MedecinRespository medecinRespository;
    RendezVousRespository rendezVousRespository;
    ConsultationRespository consultationRespository;

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

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

    @Override
    public RendezVous saveRV(RendezVous rendezVous) { return rendezVousRespository.save(rendezVous); }

    @Override
    public Consultation saveConcultation(Consultation consultation) {
        return consultationRespository.save(consultation);
    }
}
```

5. Application Properties:

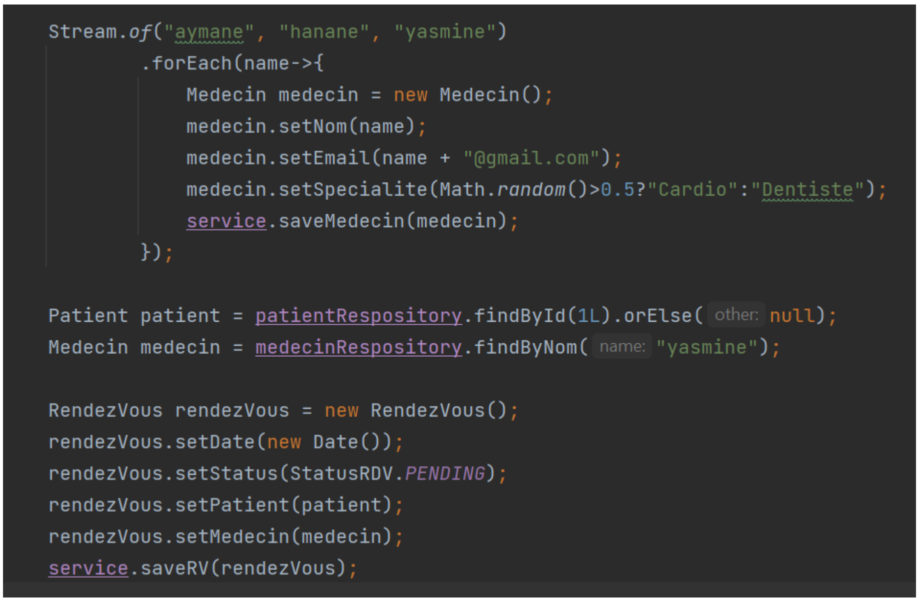


```
spring.datasource.url=jdbc:mysql://localhost:3306/db_hospital?createDatabaseIfNotExist=true
spring.datasource.username=root
spring.datasource.password=
server.port=8082
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto= create
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MariaDBDialect
```

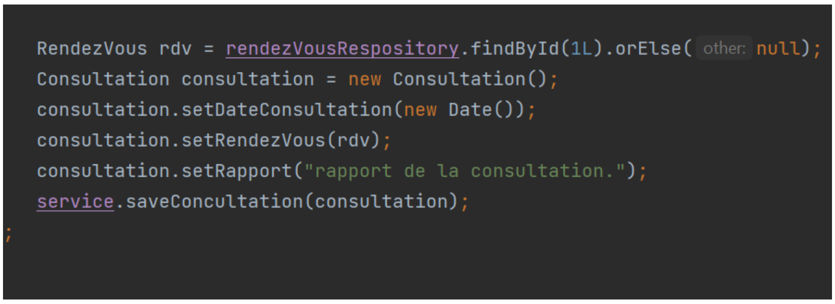
6. Application Main:



```
@SpringBootApplication
public class HospitalApplication {

    public static void main(String[] args) {

        SpringApplication.run(HospitalApplication.class, args);
    }

    @Bean
    CommandLineRunner start(IhospitalService service,
                            PatientRespository patientRespository,
                            MedecinRespository medecinRespository,
                            RendezVousRespository rendezVousRespository){
        return args -> {
            Stream.of("mohammed", "hassan", "najat")
                            .forEach(name->{
                                Patient patient = new Patient();
                                patient.setNom(name);
                                patient.setDateNaissance(new Date());
                                patient.setMalade(false);
                                service.savePatient(patient);
                            });
```

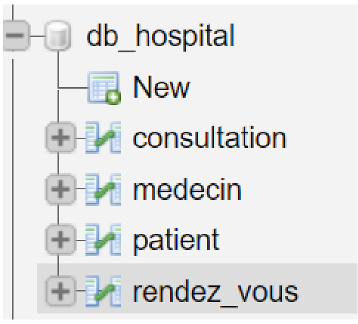


```
    Stream.of("aymane", "hanane", "yasmine")
            .forEach(name->{
                Medecin medecin = new Medecin();
                medecin.setNom(name);
                medecin.setEmail(name + "@gmail.com");
                medecin.setSpecialite(Math.random()>0.5?"Cardio":"Dentiste");
  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

    Patient patient = patientRespository.findById(1L).orElse( other: null);
    Medecin medecin = medecinRespository.findByNom( name: "yasmine");

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



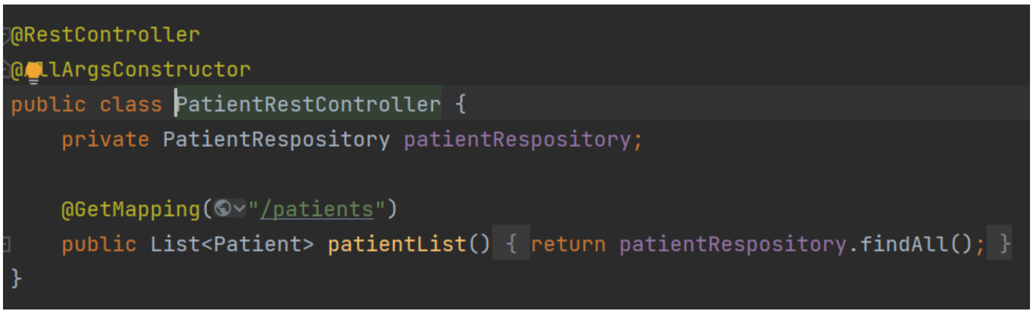
```
   RendezVous rdv = rendezVousRespository.findById(1L).orElse( other: null);
   Consultation consultation = new Consultation();
   consultation.setDateConsultation(new Date());
   consultation.setRendezVous(rdv);
   consultation.setRapport("rapport de la consultation.");
   service.saveConcultation(consultation);
;
```

7. Base de données:



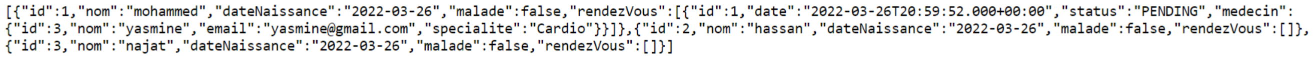
8. Couche Web:

a. Rest Controller:



```
@RestController
@AllArgsConstructor
public class PatientRestController {
    private PatientRespository patientRespository;

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

b. Resultat:



```
[{"id":1,"nom":"mohammed","dateNaissance":"2022-03-26","malade":false,"rendezVous":[{"id":1,"date":"2022-03-26T20:59:52.000+00:00","status":"PENDING","medecin":
{"id":3,"nom":"yasmine","email":"yasmine@gmail.com","specialite":"Cardio"}}]},{"id":2,"nom":"hassan","dateNaissance":"2022-03-26","malade":false,"rendezVous":[]},
{"id":3,"nom":"najat","dateNaissance":"2022-03-26","malade":false,"rendezVous":[]}]
```