

Brevet de Technicien Supérieur

Rapport TP3 :

**Activité Pratique N°3 : JPA Hibernate Spring Data**



**Réalisé par :**

**ZAKARIA ZIGHIGHI**

**Encadré par :**

**Pr. Mohammed Youssfi**

GLSID 2

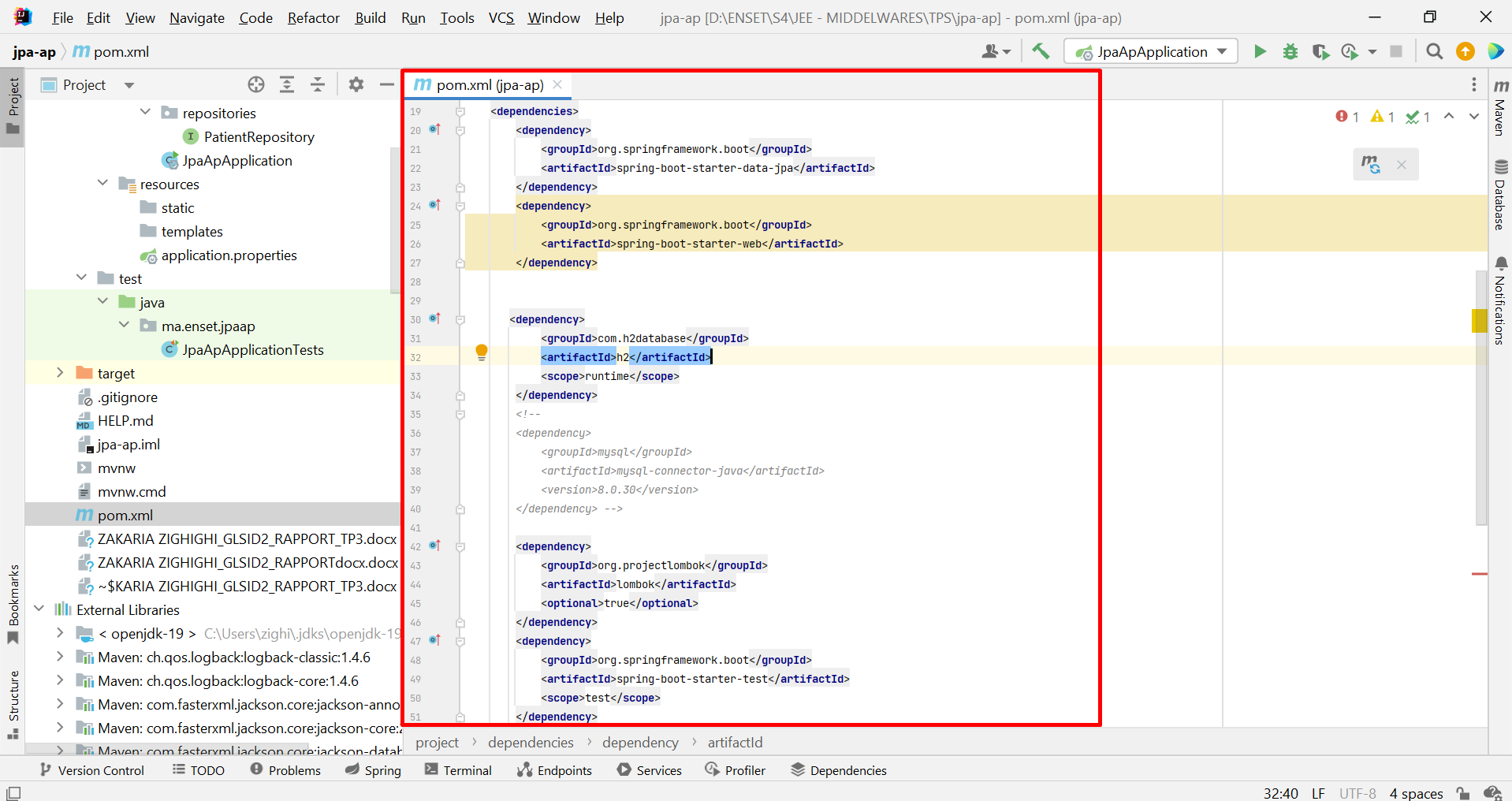
Année universitaire : 2022-23

# **Rapport TP3 :**

**LIEN REPO GITHUB :**

## **// 1- INSATLLER INTELLIJ**

**2. Créer un projet Spring Initializer avec les dépendances JPA, H2, Spring Web et Lombock**



**3. Créer l'entité JPA Patient ayant les attributs :**

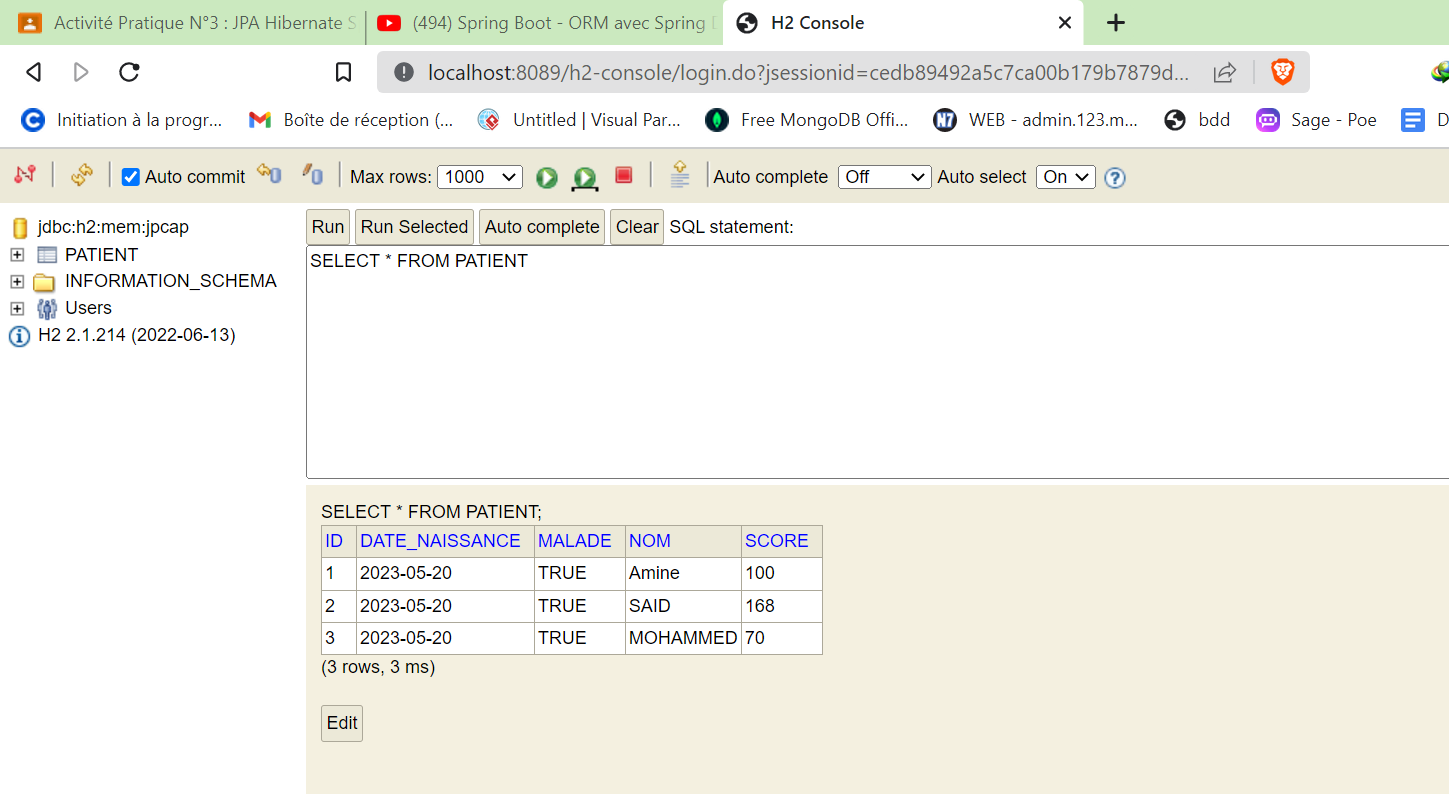
**package** ma.enset.jpaap.entities;  
  
**import** jakarta.persistence.\*;  
**import** lombok.AllArgsConstructor;  
**import** lombok.Data;  
**import** lombok.NoArgsConstructor;  
  
**import** java.util.Date;  
  
@Entity  
@Data @NoArgsConstructor @AllArgsConstructor  
**public class** Patient {  
 @Id @GeneratedValue(strategy = GenerationType.***IDENTITY***)  
 **private** Long **id**;  
 @Column(length = 50)  
 **private** String **nom**;  
 @Temporal(TemporalType.***DATE***)  
 **private** Date **dateNaissance**;  
 **private boolean malade**;  
 **private int score**;  
}

**4. Configurer l'unité de persistance dans le ficher application.properties**

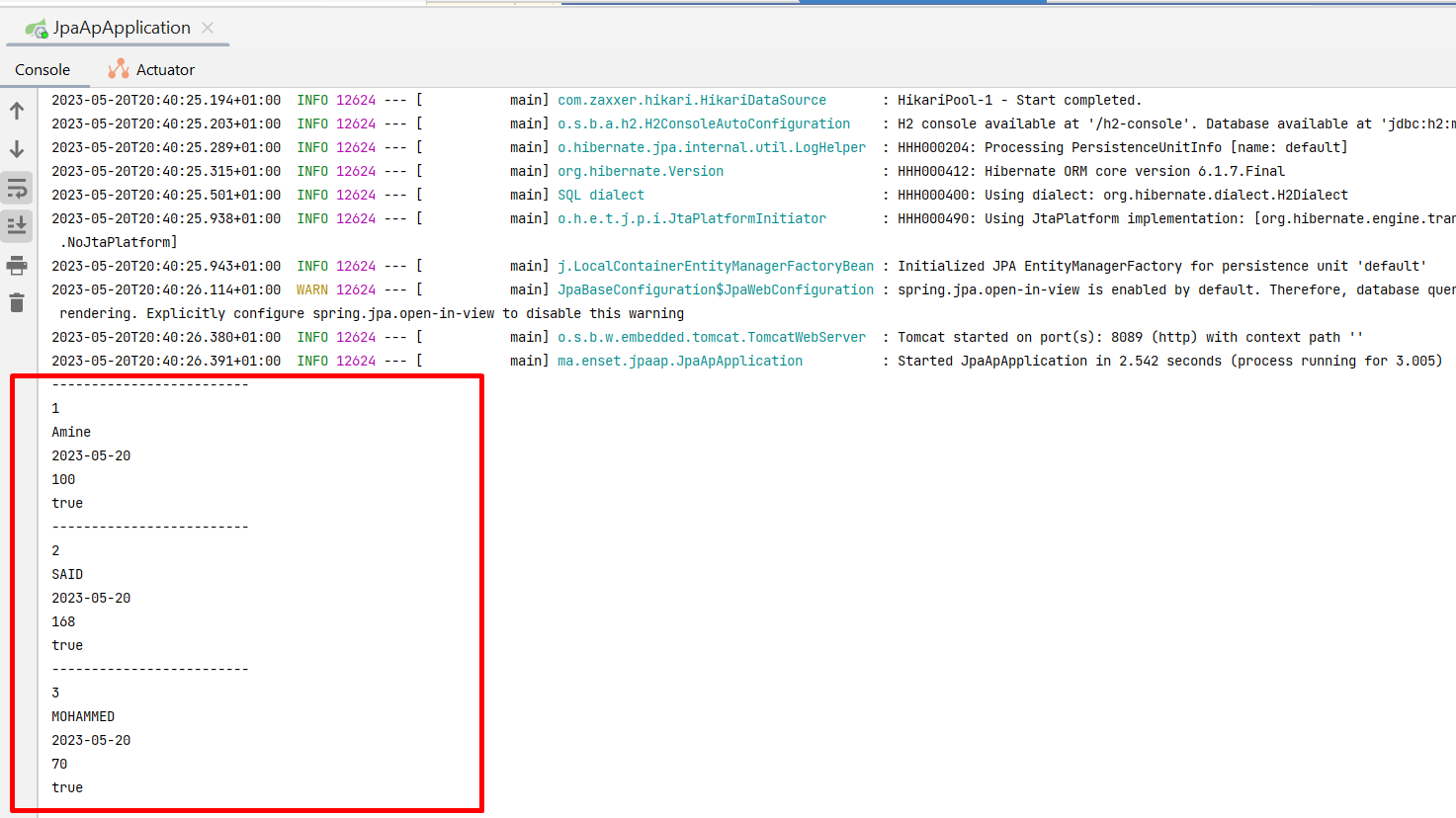
**spring.datasource.url**=**jdbc:h2:mem:jpcap  
spring.h2.console.enabled**=**true  
server.port**=**8089**

**5. Créer l'interface JPA Repository basée sur Spring data  
import** org.springframework.data.jpa.repository.JpaRepository;  
  
**public interface** PatientRepository **extends** JpaRepository<Patient, Long> {  
  
}

**6. Tester quelques opérations de gestion de patients :  
     - Ajouter des patients :**   
**package** ma.enset.jpaap;  
  
**import** ma.enset.jpaap.entities.Patient;  
**import** ma.enset.jpaap.repositories.PatientRepository;  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.boot.CommandLineRunner;  
**import** org.springframework.boot.SpringApplication;  
**import** org.springframework.boot.autoconfigure.SpringBootApplication;  
**import** org.springframework.data.domain.Page;  
**import** org.springframework.data.domain.PageRequest;  
  
**import** java.util.Date;  
**import** java.util.List;  
  
@SpringBootApplication  
**public class** JpaApApplication **implements** CommandLineRunner {  
 @Autowired  
 **private** PatientRepository **patientRepository**;  
  
  
 **public static void** main(String[] args) {  
 SpringApplication.*run*(JpaApApplication.**class**, args);  
 }  
  
 @Override  
 **public void** run(String... args) **throws** Exception {  
 **patientRepository**.save(**new** Patient(**null**,**"Amine"**,**new** Date(),**true**,100));  
 **patientRepository**.save(**new** Patient(**null**,**"SAID"**,**new** Date(),**true**,168));  
 **patientRepository**.save(**new** Patient(**null**,**"MOHAMMED"**,**new** Date(),**true**,70));  
  
 }

**  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
 - Consulter tous les patients**

List<Patient> patients = **patientRepository**.findAll();  
patients.forEach(p->{  
  
 System.***out***.println(**"-------------------------"**);  
 System.***out***.println(p.getId());  
 System.***out***.println(p.getNom());  
 System.***out***.println(p.getDateNaissance());  
 System.***out***.println(p.getScore());  
 System.***out***.println(p.isMalade());  
});

****

**- Consulter un patient by id**

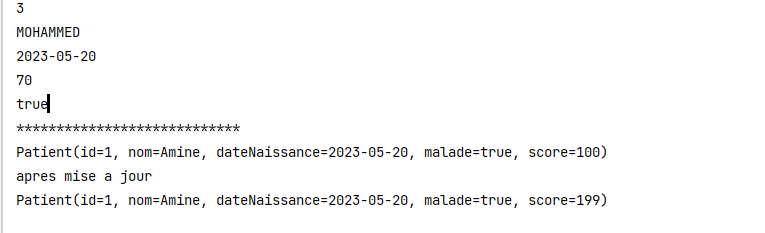
System.***out***.println(**"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**);  
Patient patient=**patientRepository**.findById(1L).orElse(**null**);  
System.***out***.println(patient);

A picture containing text, screenshot, font, line

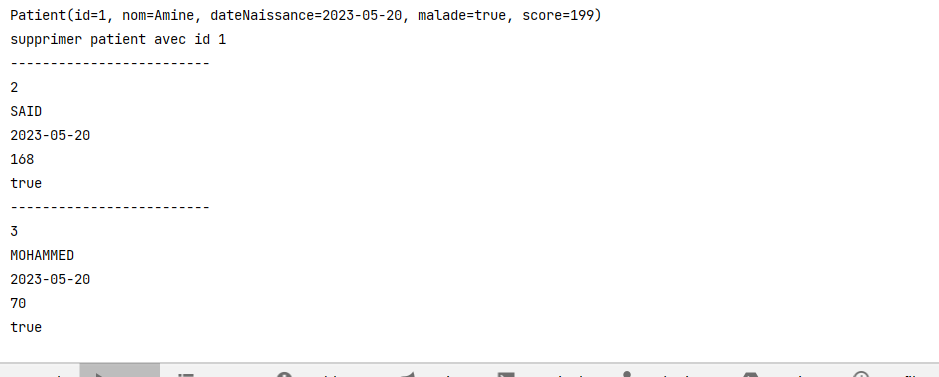
Description automatically generated

**- - Mettre à jour un patient**

System.***out***.println(**"apres mise a jour"**);  
patient.setScore(199);  
**patientRepository**.save(patient);  
System.***out***.println(patient);



**- supprimer un patient  
patientRepository**.deleteById(1L);

**  
  
  
  
  
  
  
  
  
7. Migrer de H2 Database vers MySQL  
spring.datasource.url**=**jdbc:mysql://localhost:3306/TP3DB?createDatabaseIfNotExist=true  
spring.datasource.username**=**root  
spring.datasource.password**=  
**spring.jpa.hibernate.ddl-auto**= **create  
spring.jpa.properties.hibernate.dialect**= **org.hibernate.dialect.MariaDBDialect  
server.port**=**8089  
spring.jpa.show-sql**=**true**

**Et dans pom.xml on ajoute la dépendance**

A screenshot of a computer

Description automatically generated with medium confidence<**dependency**>  
 <**groupId**>mysql</**groupId**>  
 <**artifactId**>mysql-connector-java</**artifactId**>  
 <**version**>8.0.30</**version**>  
</**dependency**>

**8. Reprendre les exemples du Patient, Médecin, rendez-vous, consultation, users et roles de la vidéo   
  
vidéo 1 : lien repo github :**

**Vidéo 2 : lien repo github :**