

**2021-2022**

**Département Mathématiques et Informatique  
GLSID 2 – S4**

## **Architecture JEE et Middlewares**

**Rapport TP 3**

**Mise en œuvre JPA, Hibernate et Spring Data**

**Préparé par : Fatima Zahra HASBI  
Encadré par : Mr Mohamed YOUSSEFI**

# Introduction

Ce tp met le point sur les concepts de bases du mapping objet relationnel en utilisant JPA, Hibernate et Spring data.

## Réalisation

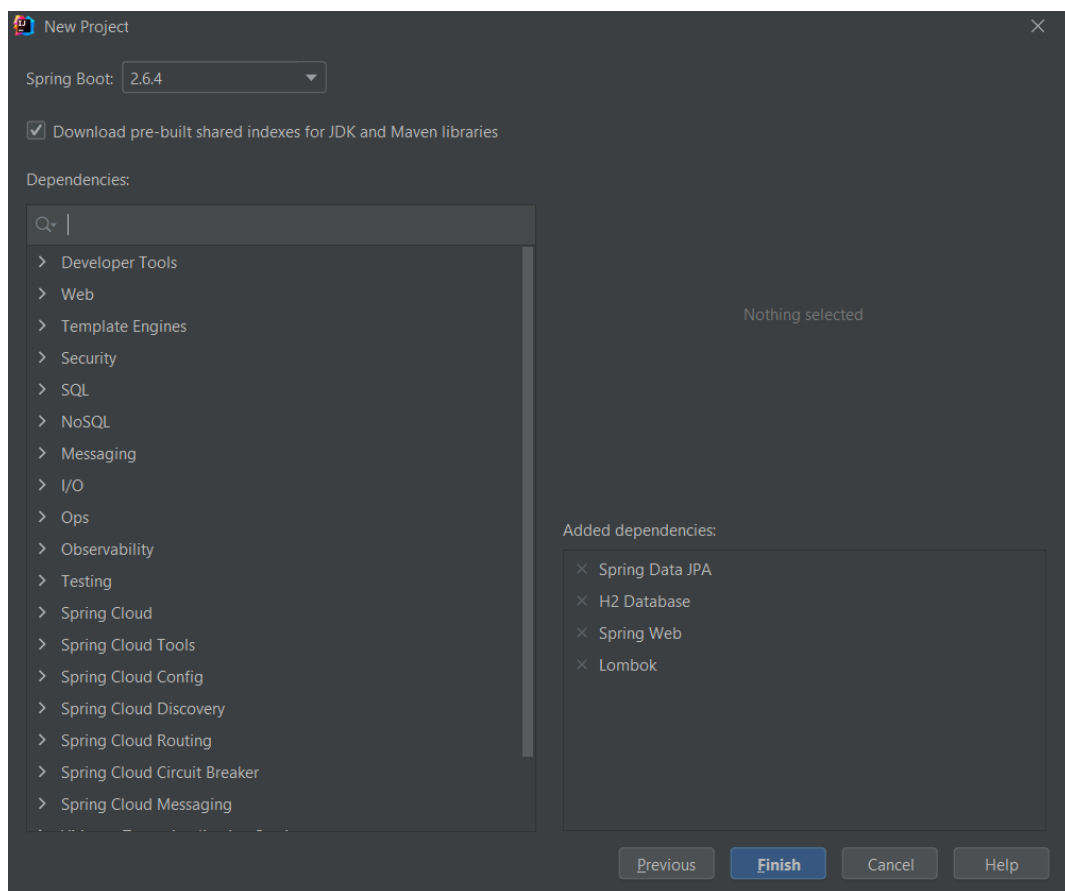
### 1- Initialisation d'un nouveau projet Spring initializr

The screenshot shows the 'New Project' dialog in IntelliJ IDEA. On the left, a list of project types is shown, with 'Spring Initializr' selected. On the right, various project configuration fields are visible:

- Server URL: `start.spring.io`
- Name: `tp3-jpa`
- Location: `~\OneDrive - enset-media.ac.ma\Hasbi\GLSID\S4\1- J2EE\tps\tp3-jpa`
- Language: `Java` (selected), `Kotlin`, `Groovy`
- Type: `Maven` (selected), `Gradle`
- Group: `ma.enset`
- Artifact: `tp3-jpa`
- Package name: `ma.enset.tp3jpa`
- Project SDK: `1.8 Oracle OpenJDK version 1.8.0_321`
- Java: `8` (selected)
- Packaging: `Jar` (selected), `War`

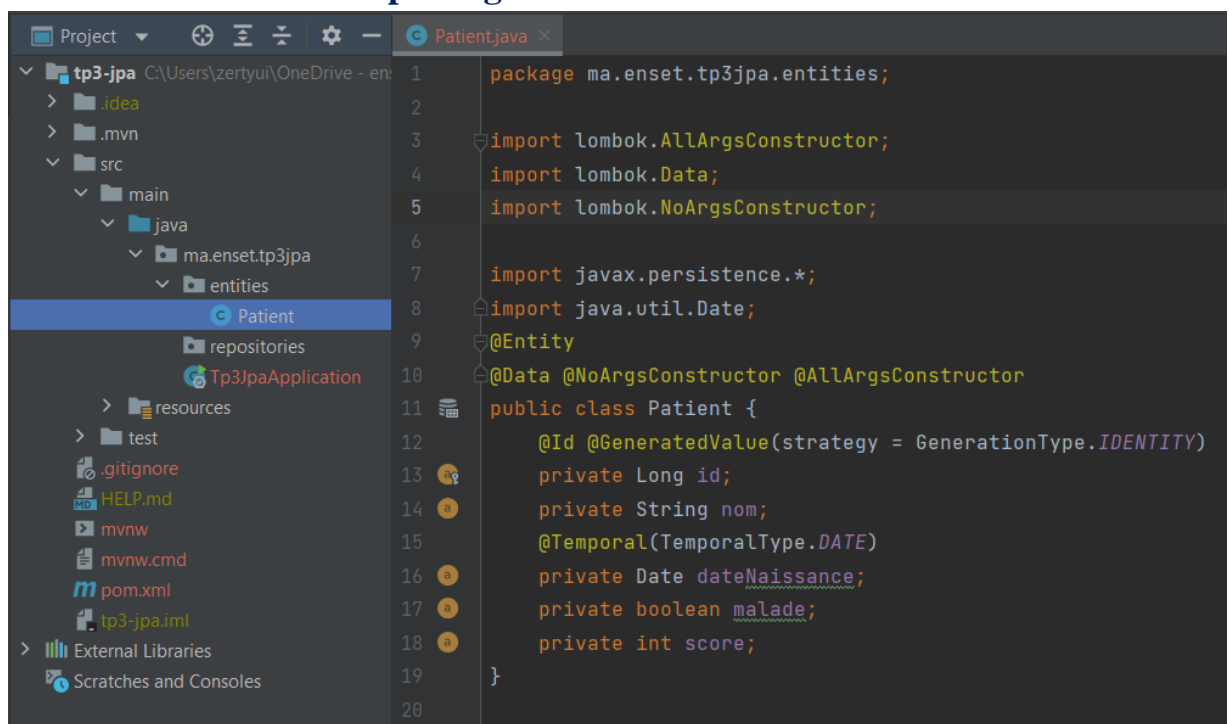
At the bottom right, there are buttons for 'Previous', 'Next' (highlighted in blue), 'Cancel', and 'Help'.

## 2- Ajouter les dépendances nécessaires pour notre projet



## 3- Création des classes de projet

### L'entité Patient dans le package entities



## L'interface PatientRepository dans le package repositories

```

1 package ma.enset.tp3jpa.repositories;
2
3 import ma.enset.tp3jpa.entities.Patient;
4 import org.springframework.data.domain.Page;
5 import org.springframework.data.domain.Pageable;
6 import org.springframework.data.jpa.repository.JpaRepository;
7 import org.springframework.data.jpa.repository.Query;
8 import org.springframework.data.repository.query.Param;
9
10 import java.util.Date;
11 import java.util.List;
12
13 public interface PatientRepository extends JpaRepository<Patient,Long> {
14     public List<Patient> findByMalade(Boolean m);
15     Page<Patient> findByMalade(boolean m , Pageable pageable);
16     List<Patient> findByMaladeAndScoreLessThan(boolean m , int score);
17     List<Patient> findByMaladeIsFalseAndScoreLessThan(int score);
18     @Query("select p from Patient p where p.dateNaissance between :x and :y or p.nom like :z")
19     public List<Patient> FindPatients(@Param("x") Date d1, @Param("y") Date d2, @Param("z") String nom);
20 }
21

```

## Configuration du fichier application.properties

```
application.properties x
1 spring.datasource.url=jdbc:h2:mem:patient-db
2 spring.h2.console.enabled=true
3 server.port=8082
4
5
```

### Tester l'application :

[illegible]

Ouvrant la base de donnée h2 sur localhost:8082/h2-console

English ▾ Preferences Tools Help

### Login

Saved Settings: Generic H2 (Embedded) ▾

Setting Name: Generic H2 (Embedded) Save Remove

---

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:mem:patient-db

User Name: sa

Password:

Connect Test Connection

< > ↺ ☐ | 🌐 localhost:8082/h2-console/login.do

🔗 | 💰 | ☒ Auto commit 🔗 | 🖋️ | Max rows: 1000 ▾ | ▶️ ▶️ ⏏️ | 📄

📁 jdbc:h2:mem:patient-db

- 📄 PATIENT
  - + 📄 ID
  - + 📄 DATE\_NAISSANCE
  - + 📄 MALADE
  - + 📄 NOM
  - + 📄 SCORE
  - + 📄 Indexes
- + 📁 INFORMATION\_SCHEMA
- + 📄 Sequences
- + 📄 Users

📘 H2 1.4.200 (2019-10-14)

Run Run Selected Auto complete Clear SQL s

```
SELECT * FROM PATIENT |
```

**Important Commands**

## Tp3JpaApplication.java

```
1 package ma.enset.tp3jpa;
2
3 import ma.enset.tp3jpa.entities.Patient;
4 import ma.enset.tp3jpa.repositories.PatientRepository;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.boot.CommandLineRunner;
7 import org.springframework.boot.SpringApplication;
8 import org.springframework.boot.autoconfigure.SpringBootApplication;
9
10 import java.util.Calendar;
11 import java.util.Date;
12 import java.util.List;
13
14 @SpringBootApplication
15 public class Tp3JpaApplication implements CommandLineRunner {
16     @Autowired
17     PatientRepository patientRepository;
18     public static void main(String[] args) {
19         SpringApplication.run(Tp3JpaApplication.class, args);
20     }
21
22     @Override
23     public void run(String... args) {
24         System.out.println("-----");
25
26         //Filling the database with random data
27         for (int i=0;i<100;i++){
28             patientRepository.save(new Patient( id: null, nom: "Fatima Zahra", new Date(), malade: Math.random() < 0.5, score: 44));
29         }
```

```
        // fetching all Patient data
        List<Patient> patients = patientRepository.findAll();
2
3        // displaying ALL
4        for (Patient p : patients) {
5            System.out.println(p.getNom());
6            System.out.println(p.getDateNaissance());
7            System.out.println(p.isMalade());
8            System.out.println(p.getScore());
9            System.out.println("#####");
10        }
11        // Fetching ALL MALADE = FALSE
12        List<Patient> byMalade = patientRepository.findByMalade( m: false);
13        System.out.println("----- Patients PAS Malades -----");
14        for (Patient p : byMalade) {
15            System.out.println(p.getNom());
16            System.out.println(p.getDateNaissance());
17            System.out.println(p.isMalade());
18            System.out.println(p.getScore());
19            System.out.println("#####");
20        }
21        // Fetching ALL MALADE = TRUE
22        byMalade = patientRepository.findByMalade( m: true);
23        System.out.println("----- Patients Malades -----");
24        for (Patient p : byMalade) {
25            System.out.println(p.getNom());
26            System.out.println(p.getDateNaissance());
27            System.out.println(p.isMalade());
28            System.out.println(p.getScore());
29            System.out.println("#####");
30        }
31    }
```

## Résultat

```
Console  🔥 Actuator
↑ #####
↓ Fatima Zahra
  2022-03-05
  false
  44
⌵ #####
🖨 Fatima Zahra
  2022-03-05
  false
  44
  #####
  Fatima Zahra
  2022-03-05
  true
  44
  #####
  Fatima Zahra
  2022-03-05
  true
  44
  #####
```

## Sur H2 database

jdbc:h2:mem:patient-db

PATIENT  
INFORMATION\_SCHEMA  
Sequences  
Users

H2 1.4.200 (2019-10-14)

Run Run Selected Auto complete Clear SQL statement:

```
SELECT * FROM PATIENT
```

```
SELECT * FROM PATIENT;
```

ID	DATE_NAISSANCE	MALADE	NOM	SCORE
1	2022-03-05	FALSE	Fatima Zahra	44
2	2022-03-05	FALSE	Fatima Zahra	44
3	2022-03-05	TRUE	Fatima Zahra	44
4	2022-03-05	FALSE	Fatima Zahra	44
5	2022-03-05	FALSE	Fatima Zahra	44
6	2022-03-05	FALSE	Fatima Zahra	44
7	2022-03-05	FALSE	Fatima Zahra	44
8	2022-03-05	TRUE	Fatima Zahra	44
9	2022-03-05	TRUE	Fatima Zahra	44
10	2022-03-05	FALSE	Fatima Zahra	44
11	2022-03-05	FALSE	Fatima Zahra	44
12	2022-03-05	FALSE	Fatima Zahra	44
13	2022-03-05	TRUE	Fatima Zahra	44
14	2022-03-05	TRUE	Fatima Zahra	44
15	2022-03-05	TRUE	Fatima Zahra	44
16	2022-03-05	FALSE	Fatima Zahra	44
17	2022-03-05	FALSE	Fatima Zahra	44

**FIN**