ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR
Membre de HONORIS UNITED UNIVERSITIES

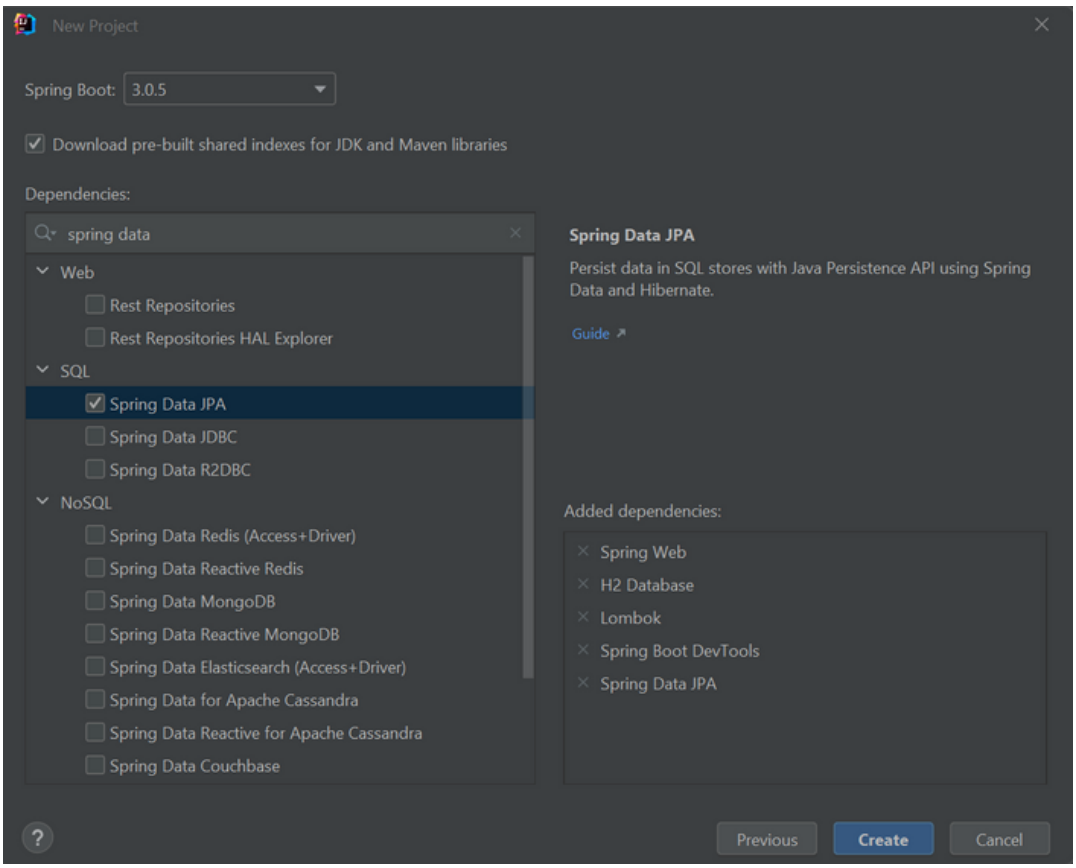
Compte Rendu TP2: Spring Data JPA

RÉALISÉ PAR:
EL YOUSFI SAFAA

2023

Objectif :

- L'objectif de ce tp est de créer un projet **Spring Boot** avec les dépondances suivantes:



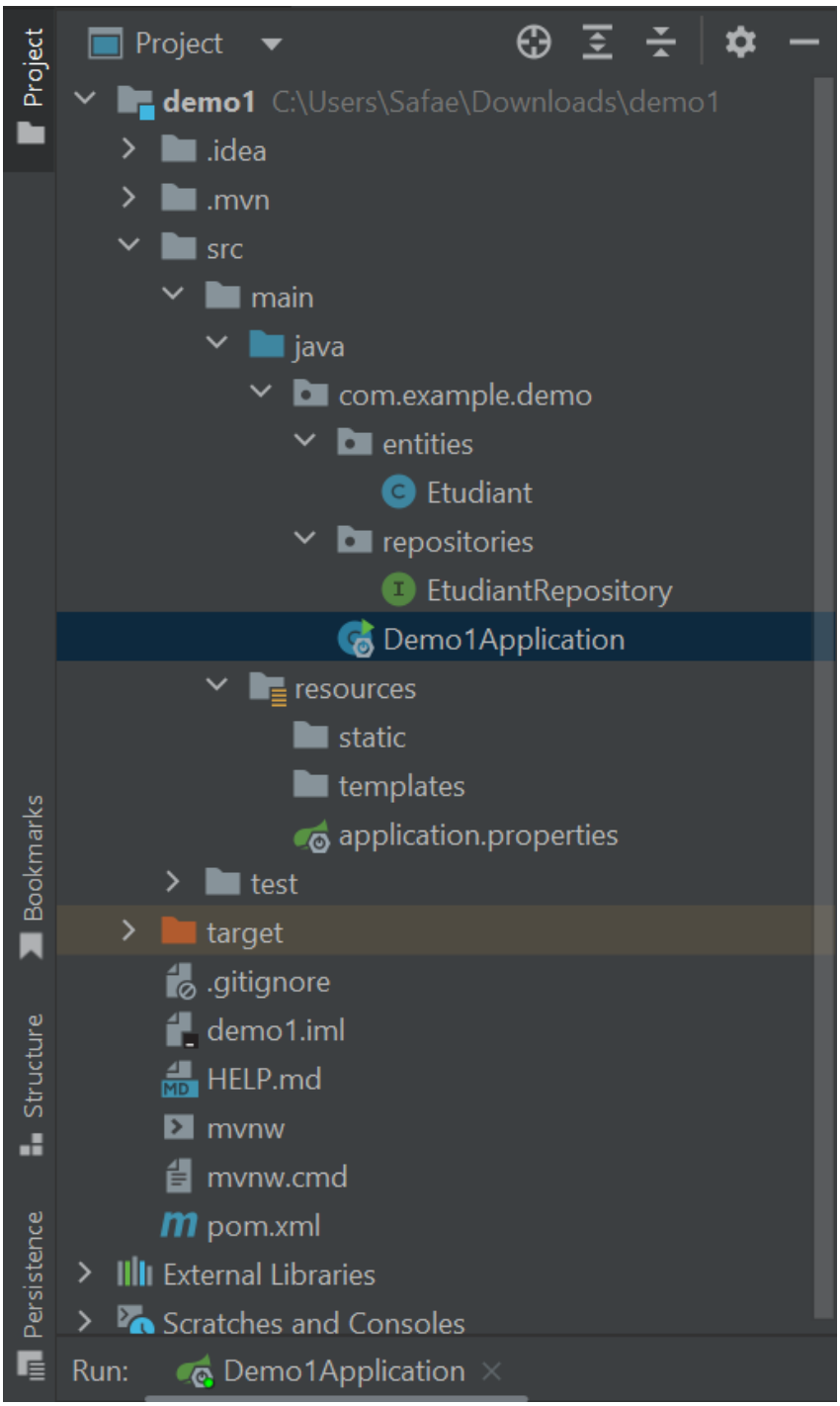
Pour la base de données, on a utilisé deux méthodes:

1- h2 Database
2- mysql Database

Frameworks ORM : Hibernate

- Hibernate est un **framework open source** gérant la **persistance** des objets en **base de données relationnelle.**

Les composants du projet



entities:

- Dossier contenant la classe Java **Etudiant**.

repositories:

- Dossier contenant l'interface **EtudiantRepository.**

Demo1Application:

- Fichier contenant la classe **Main**.

ressources:

- Dossier contenant le fichier **application.properties** => fichier de configuration d'une application **Spring Boot**.

Partie 1 :h2

Classe Etudiant

```java
package com.example.demo.entities;
import jakarta.persistence.*;
import lombok.*; /* Lombok:C'est un pluggin: à chaque enregistrement du fichier,le compilateur
Java compile avec lombok */
import java.util.Date;
7 usages
@Entity //entité jpa
@Table(name="Etudiant") @Data @NoArgsConstructor @AllArgsConstructor
/* @Data:Pour ajouter les getters et les setters
   @NoArgsConstructor: Constructeur sans paramètre
   @AllArgsConstructor:Constructeur avec paramètre
   @Id:Champ clé primaire
*/
@Setter @Getter
public class Etudiant {
    no usages
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    no usages
    @Column(name="Registration_number",unique = true)
    private String RegistrationNumber;
```

```java
    no usages
    @Column(name = "Name",length = 30,nullable = false)
    private String fullName;
    no usages
    @Temporal(TemporalType.DATE)
    private Date birthday;
    no usages
    private Boolean stillActive;
}
```

Interface EtudiantRepository

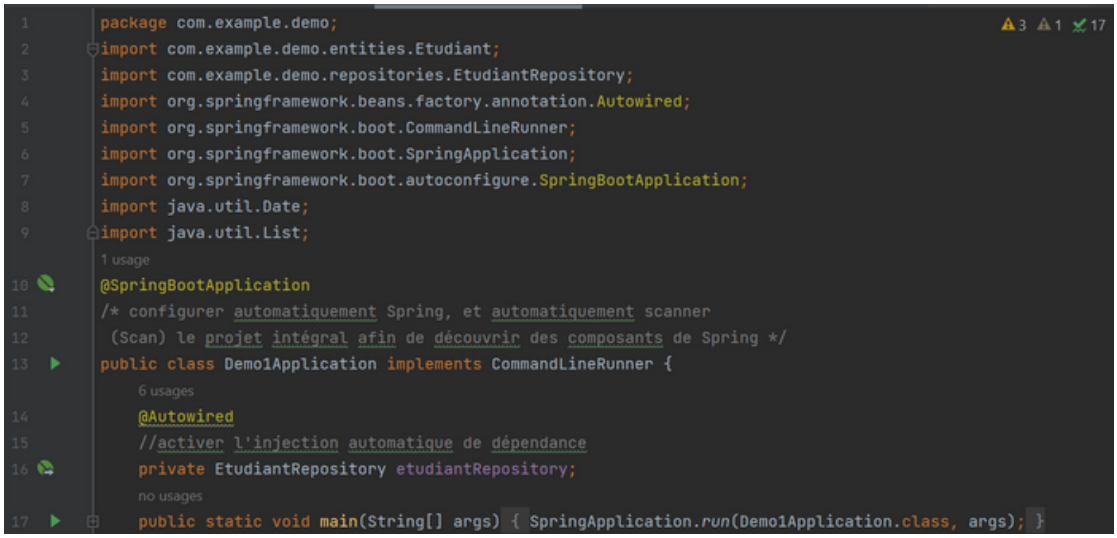
```java
package com.example.demo.repositories;

import com.example.demo.entities.Etudiant;
import org.springframework.data.jpa.repository.JpaRepository;

2 usages
public interface EtudiantRepository extends JpaRepository<Etudiant,Integer> {
    // spring data : hériter d'une interface génériwue JpaRepositor<Entité,Type d'id>

}
```

Classe Demo1Application



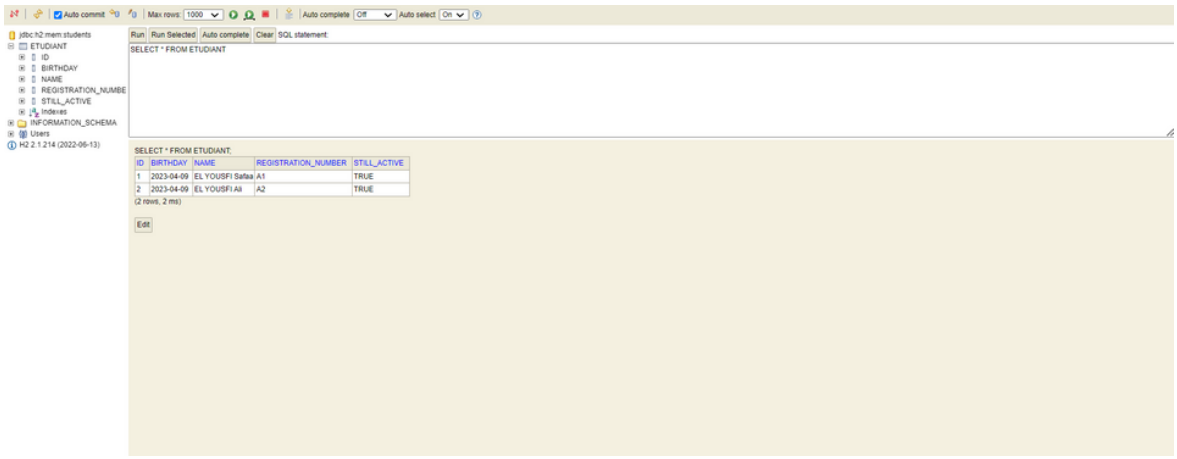
```java
package com.example.demo;
import com.example.demo.entities.Etudiant;
import com.example.demo.repositories.EtudiantRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import java.util.Date;
import java.util.List;
1 usage
@SpringBootApplication
/* configurer automatiquement Spring, et automatiquement scanner
 (Scan) le projet intégral afin de découvrir des composants de Spring */
public class Demo1Application implements CommandLineRunner {
    6 usages
    @Autowired
    //activer l'injection automatique de dépendance
    private EtudiantRepository etudiantRepository;
    no usages
    public static void main(String[] args) { SpringApplication.run(Demo1Application.class, args); }
```

```java
    @Override
    public void run(String... args) throws Exception {
        //Create (Insertion)
        /*L'id est null car elle s'incrémente automatiquement */
        System.out.println("****************** Insert **********************");
        etudiantRepository.save(new Etudiant( id: null, RegistrationNumber: "A1", fullName: "EL YOUSFI Safaa",new Date(),
        etudiantRepository.save(new Etudiant( id: null, RegistrationNumber: "A2", fullName: "EL YOUSFI Ali",new Date(), sti
        System.out.println("*****************************************");
        System.out.println("Count:"+etudiantRepository.count());
        List<Etudiant> etudiantList=etudiantRepository.findAll();
        etudiantList.forEach(e-> {
                    System.out.println("==================================");
                    System.out.println(e.getId());
                    System.out.println(e.getRegistrationNumber());
                    System.out.println(e.getFullName());
        });
```

```java
            System.out.println("********************** Read ************************");
            Etudiant etudiant=etudiantRepository.findById(1).orElse( other: null);
            System.out.println(etudiant.toString());
            System.out.println("********************** Update ************************");
            etudiant.setRegistrationNumber("S2");
            etudiantRepository.save(etudiant);
            System.out.println(etudiant.toString());
            System.out.println("********************** Delete ************************");
            etudiantRepository.deleteById(1);
            System.out.println("Count:" +etudiantRepository.count());

    }
}
```

Résultat d'excécution




```
************************************************
Count:2
====================================
1
A1
EL YOUSFI Safaa
====================================
2
A2
EL YOUSFI Ali
********************** Read ************************
Etudiant(id=1, RegistrationNumber=A1, fullName=EL YOUSFI Safaa, birthday=2023-04-09, stillActive=true)
********************** Update ************************
Etudiant(id=1, RegistrationNumber=S2, fullName=EL YOUSFI Safaa, birthday=2023-04-09, stillActive=true)
********************** Delete ************************
Count:1
```

Partie 2 :mysql

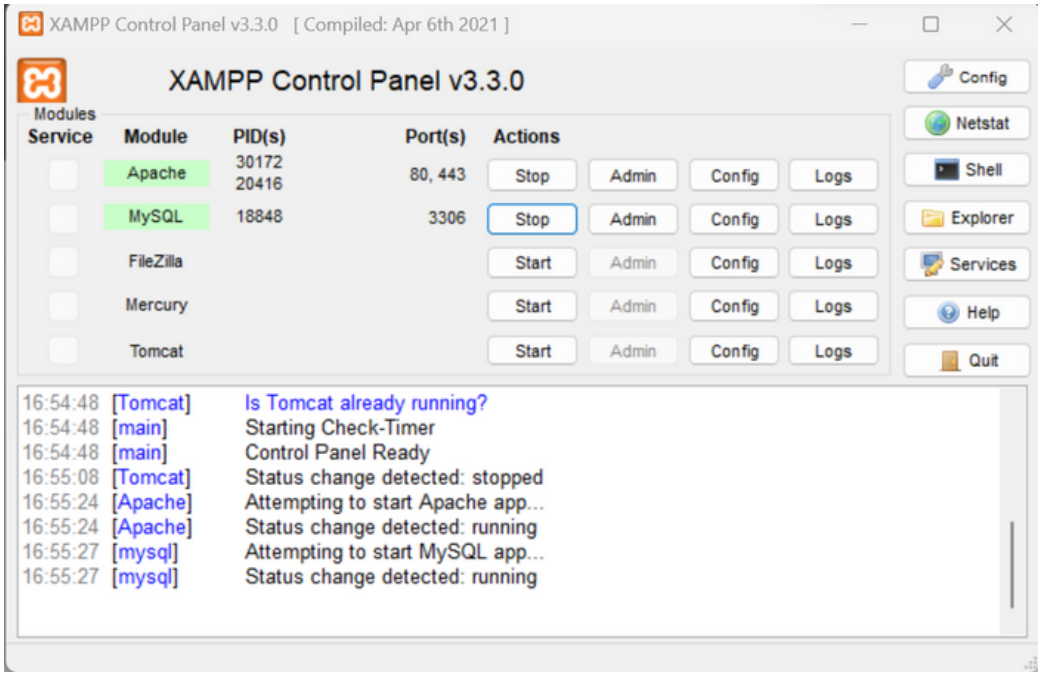
On met en block de commantaire la dépondence de h2 et on ajoute la dépondence de mysql

```
<!--<dependency>
    <groupId>com.h2database</groupId>
    <artifactId>h2</artifactId>
    <scope>runtime</scope>
</dependency>-->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.25</version>
</dependency>
```

On change les paramètres du fichier de configuration application.properties

```
spring.datasource.url=jdbc:mysql://localhost:3306/students?createDatabaseIfNotExist=true
spring.datasource.username=root
spring.datasource.password=
#spring.h2.console.enabled=true
server.port=8080
spring.jpa.hibernate.ddl-auto=update
spring.jpa.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect
spring.jpa.show-sql=true
```

On active Apache et MySQL



Résultat d'excécution

Base de donnée

Table