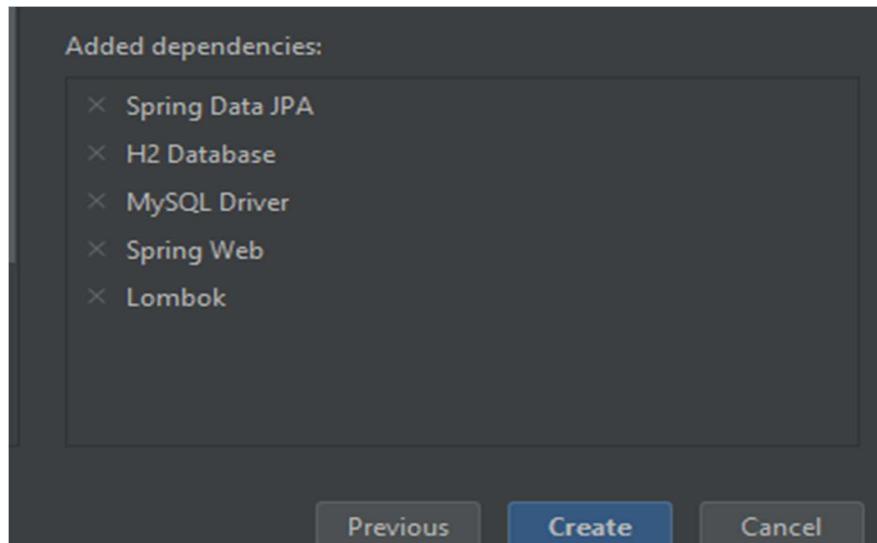


Compte rendu du tp

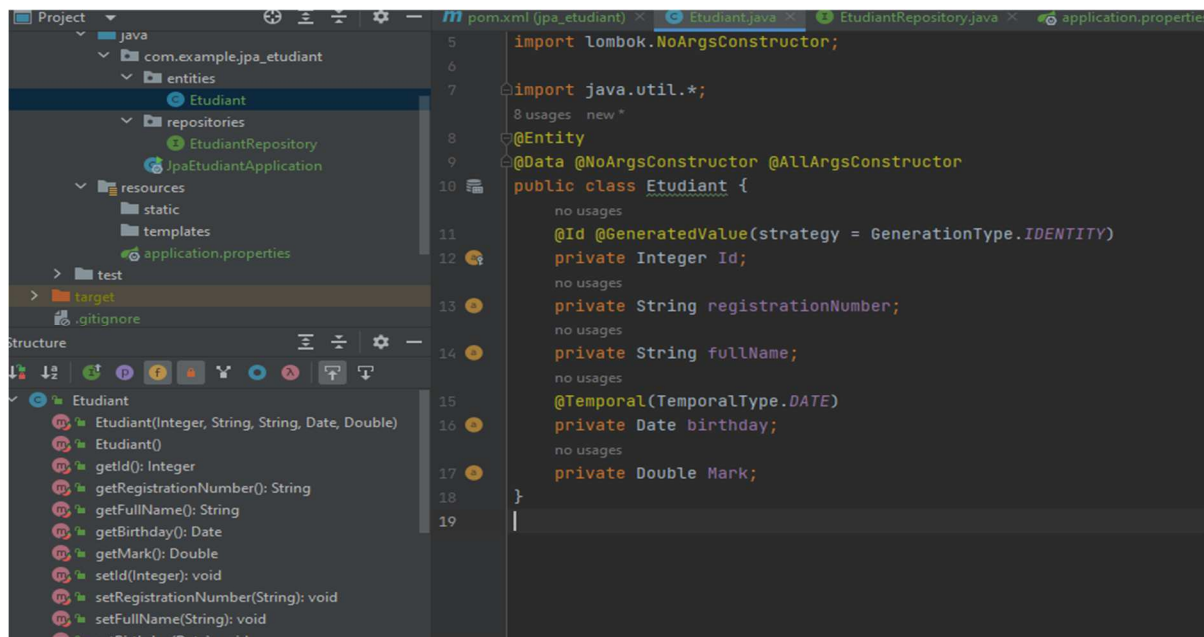
Réalisé par ELHANNIOUI CHAYMAE

En 4ème année Miage G5

1/ Tout d'abord, on a créé notre projet spring boot jpa_etudiant avec les dépendances : spring data jpa, lombok, spring web, et h2 database



2/ Après on a créé l'entité Jpa Etudiant



3/ Ensuite, on a configuré la data source(application properties):

```
spring.datasource.url=jdbc:h2:mem:etudiant-db
spring.h2.console.enabled=true
server.port=4455
```

4/ Création de l'interface Etudiant Repository basé sur spring data.

```
package com.example.jpa_etudiant.repositories;

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

public interface EtudiantRepository extends JpaRepository<Etudiant, Integer> {
}
```

5/ Connexion à la base de données etudiant-db

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:etudiant-db

User Name: sa

Password:

Connect Test Connection

← → ↻ ⓘ localhost:4455/h2-console/login.do?jsessionId=31cd3106bf7b64e725ef71a1171d1bd9

🔥 | 🛠️ | ☒ Auto commit | 🔄 | 📄 | Max rows: 1000 | 🏃 | 🛑 | 📄 | Auto complete Off | Auto select On | ⓘ

📁 jdbc:h2:mem:etudiant-db

- 📁 ETUDIANT
 - ID
 - MARK
 - BIRTHDAY
 - FULL_NAME
 - REGISTRATION_NUMBE
 - Indexes
- 📁 INFORMATION_SCHEMA
- 👤 Users
- 📄 H2 2.1.214 (2022-06-13)

Run Run Selected Auto complete Clear SQL statement:

Important Commands

?	Displays this Help Page
📄	Shows the Command History

6/ Test de l'application:

L'ajout des étudiants:

```
package com.example.jpa_etudiant;

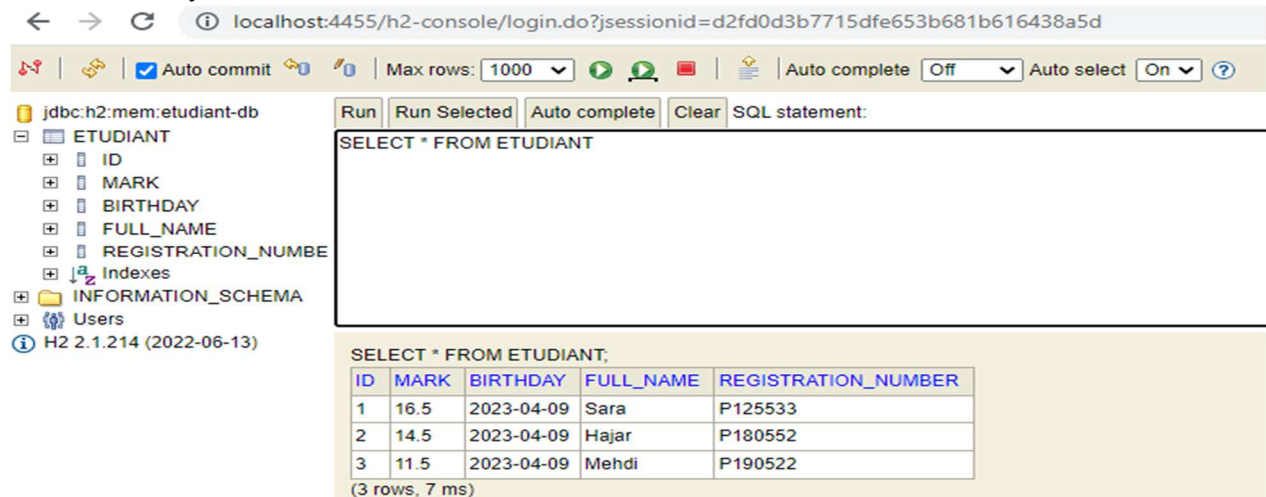
import com.example.jpa_etudiant.entities.Etudiant;
import com.example.jpa_etudiant.repositories.EtudiantRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

import java.util.*;

1 usage new *
@SpringBootApplication
public class JpaEtudiantApplication implements CommandLineRunner {
    4 usages
    @Autowired
    private EtudiantRepository etudiantRepository;
    no usages new *
    public static void main(String[] args) {
        SpringApplication.run(JpaEtudiantApplication.class, args);
    }
}
```

```
new *
public void run(String... args) throws Exception {
    etudiantRepository.save(
        new Etudiant( id: null, registrationNumber: "P125533", fullName: "Sara", new Date(), Mark: 16.50));
    etudiantRepository.save(
        new Etudiant( id: null, registrationNumber: "P180552", fullName: "Hajar", new Date(), Mark: 14.50));
    etudiantRepository.save(
        new Etudiant( id: null, registrationNumber: "P190522", fullName: "Mehdi", new Date(), Mark: 11.50));
}
```

On constate l'ajout des étudiants dans la base de données:



The screenshot shows a web application interface for a database. The URL bar indicates a connection to localhost:4455/h2-console/login.do. The interface includes a sidebar with a tree view of the database structure, showing a connection to 'jdbc:h2:mem:etudiant-db' and a table named 'ETUDIANT'. The table has columns: ID, MARK, BIRTHDAY, FULL_NAME, and REGISTRATION_NUMBE. The main area displays the SQL statement 'SELECT * FROM ETUDIANT' and its results in a table with 3 rows. The results are:

ID	MARK	BIRTHDAY	FULL_NAME	REGISTRATION_NUMBER
1	16.5	2023-04-09	Sara	P125533
2	14.5	2023-04-09	Hajar	P180552
3	11.5	2023-04-09	Mehdi	P190522

Below the table, it indicates '(3 rows, 7 ms)'.

L'affichage de la liste des étudiants

```
List<Etudiant> etudiants= etudiantRepository.findAll();
etudiants.forEach(e->{
    System.out.println("=====");
    System.out.println(e.getId());
    System.out.println(e.getRegistrationNumber());
    System.out.println(e.getFullName());
    System.out.println(e.getBirthday());
    System.out.println(e.getMark());
});
System.out.println("*****");
```

La modification de la note de l'étudiant de l'id 1

```
Etudiant etudiant = etudiantRepository.findById(1).orElse( other: null);
if(etudiant!=null){
    System.out.println(etudiant.getRegistrationNumber());
    System.out.println(etudiant.getFullName());
    System.out.println(etudiant.getBirthday());
}
etudiant.setMark(08.75);
etudiantRepository.save(etudiant);
```

On constate que la note de l'étudiant a été bien modifiée

localhost:4455/h2-console/login.do?jsessionId=d2fd0d3b7715dfe653b681b616438a5d

Auto commit ☒ Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:mem:etudiant-db

- ETUDIANT
 - ID
 - MARK
 - BIRTHDAY
 - FULL_NAME
 - REGISTRATION_NUMBE
 - Indexes
- INFORMATION_SCHEMA
- Users
- H2 2.1.214 (2022-06-13)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM ETUDIANT

ID	MARK	BIRTHDAY	FULL_NAME	REGISTRATION_NUMBER
1	16.5	2023-04-09	Sara	P125533
2	14.5	2023-04-09	Hajar	P180552
3	11.5	2023-04-09	Mehdi	P190522

(3 rows, 7 ms)

SELECT * FROM ETUDIANT;

ID	MARK	BIRTHDAY	FULL_NAME	REGISTRATION_NUMBER
1	8.75	2023-04-09	Sara	P125533

La suppression de l'étudiant d'id 3

```
etudiantRepository.deleteById(3);
```

L'étudiant a été supprimé automatiquement de la base de données

SELECT * FROM ETUDIANT;

ID	MARK	BIRTHDAY	FULL_NAME	REGISTRATION_NUMBER
1	8.75	2023-04-09	Sara	P125533
2	14.5	2023-04-09	Hajar	P180552

(2 rows, 5 ms)

Ajout des entités Professeur, Course et Session

```
Professeur.java × Course.java × Session.java × ProfessorRepository.java ×
1 package com.example.jpa_etudiant.entities;
2
3 import jakarta.persistence.*;
4 import lombok.AllArgsConstructor;
5 import lombok.Data;
6 import lombok.NoArgsConstructor;
7 import org.hibernate.annotations.CreationTimestamp;
8 import java.util.*;
9
10 @Entity
11 @Data
12 @NoArgsConstructor
13 @AllArgsConstructor
14 public class Professeur {
15
16     no usages
17     @Id
18     @GeneratedValue(strategy = GenerationType.IDENTITY)
19     private int id;
20
21     no usages
22     @Column(length = 30, nullable = false)
23     private String fullName;
24
25     no usages
26     @Temporal(TemporalType.DATE) @CreationTimestamp
27     private Date assignmentDate;
28 }
```



```
Professeur.java × Course.java × Session.java × ProfessorRepository.java ×
1 package com.example.jpa_etudiant.entities;
2
3 import jakarta.persistence.*;
4 import lombok.AllArgsConstructor;
5 import lombok.Data;
6 import lombok.NoArgsConstructor;
7
8 2 usages new *
9 @Entity
10 @Data
11 @NoArgsConstructor
12 @AllArgsConstructor
13 public class Course {
14     no usages
15     @Id
16     @GeneratedValue(strategy = GenerationType.IDENTITY)
17     private int id;
18     no usages
19     @Column(length=25, nullable=false, unique=false)
20     private String title;
21     no usages
22     private String description;
23     no usages
24     private int timing;
25 }
```

```
Professeur.java x Course.java x Session.java x ProfessorRepository.java x Cour
package com.example.jpa_etudiant.entities;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import java.util.*;

@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Session {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    @Temporal(TemporalType.DATE)
    private Date date;
    @Temporal(TemporalType.TIME)
    private Date start_time;
    @Temporal(TemporalType.TIME)
    private Date end_time;
}
```

Ajout des repositories

```
package com.example.jpa_etudiant.repositories;
import com.example.jpa_etudiant.entities.Course;
import org.springframework.data.jpa.repository.JpaRepository;

public interface CourseRepository extends JpaRepository<Course, Integer>{

}
```

```
package com.example.jpa_etudiant.repositories;
import com.example.jpa_etudiant.entities.Professeur;
import org.springframework.data.jpa.repository.JpaRepository;

public interface ProfessorRepository extends JpaRepository<Professeur, Integer> {

}
```



```
package com.example.jpa_etudiant.repositories;

import org.springframework.data.jpa.repository.JpaRepository;
import com.example.jpa_etudiant.entities.Session;

@Usagees new *
public interface SessionRepository extends JpaRepository<Session, Integer> {

}
```