**spring-data-jpa-handson**

**Spring Data JPA - Quick Example**

**CODING:**

**pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>studentapp</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>studentapp</name>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.1.1</version>

<relativePath/>

</parent>

<dependencies>

<!-- Spring Boot Starter Web -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<!-- Spring Data JPA -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<!-- H2 In-Memory Database -->

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**Creating main class**-StudentAppApplication.java

**StudentAppApplication.java**

package com.example.studentapp;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class StudentAppApplication implements CommandLineRunner {

@Autowired

private StudentRepository studentRepository;

public static void main(String[] args) {

SpringApplication.run(StudentAppApplication.class, args);

}

@Override

public void run(String... args) {

// Saving 2 students

Student s1 = new Student("Jerlin", "CSE");

Student s2 = new Student("Jenitta", "IT");

studentRepository.save(s1);

studentRepository.save(s2);

System.out.println("Students in DB:");

for (Student s : studentRepository.findAll()) {

System.out.println(s.getName() + " - " + s.getDepartment());

}

}

}

**Student.java**

package com.example.studentapp;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

@Entity

public class Student {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String department;

public Student() {}

public Student(String name, String department) {

this.name = name;

this.department = department;

}

public Long getId() {

return id;

}

public String getName() {

return name;

}

public String getDepartment() {

return department;

}

public void setName(String name) {

this.name = name;

}

public void setDepartment(String department) {

this.department = department;

}

}

**StudentRepository.java(Interface)**

package com.example.studentapp;

import org.springframework.data.jpa.repository.JpaRepository;

public interface StudentRepository extends JpaRepository<Student, Long> {

}

**Adding Application Configuration-src/resources/application.properties**

**application.properties**

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

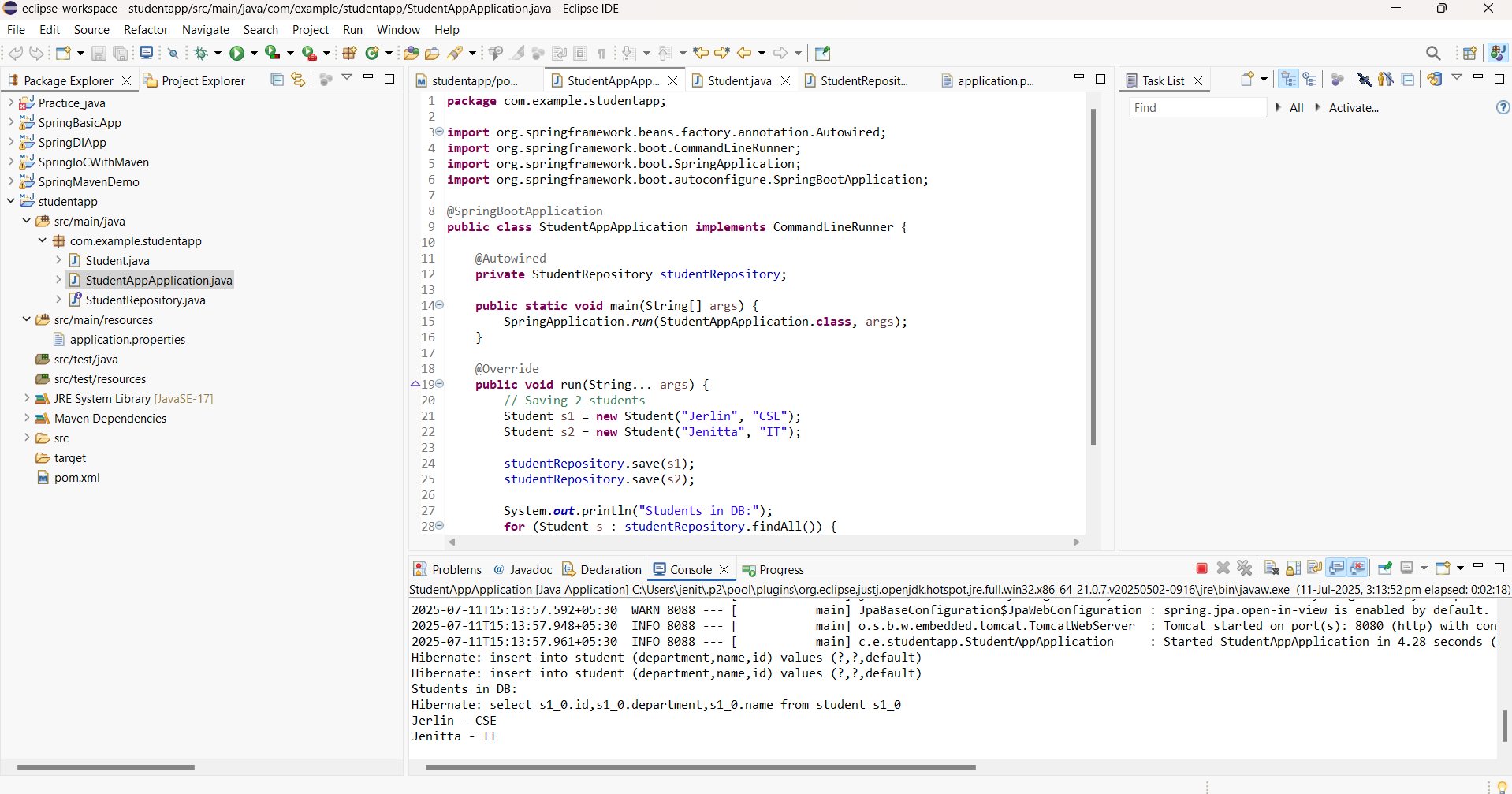
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.h2.console.enabled=true

**OUTPUT:**

****

**Difference between JPA, Hibernate and Spring Data JPA**