**S.NO:1 SPRING DATA JPA – QUICK EXAMPLE**

**SOLUTION:**

**Create Student Entity**

package com.example.demo.entity;

import jakarta.persistence.\*;

public class Student {

private Long id;

private String name;

private String department;

public Long getId() { return id; }

public void setId(Long id) { this.id = id; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

public String getDepartment() { return department; }

public void setDepartment(String department) { this.department = department; }

}

**Create Student Repository Interface**

package com.example.demo.repository;

import com.example.demo.entity.Student;

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

public interface StudentRepository extends JpaRepository<Student, Long> {

}

**Create a Simple Controller**

package com.example.demo.controller;

import com.example.demo.entity.Student;

import com.example.demo.repository.StudentRepository;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

public class StudentController {

private final StudentRepository studentRepo;

public StudentController(StudentRepository studentRepo) {

this.studentRepo = studentRepo;

}

public Student createStudent(@RequestBody Student student) {

return studentRepo.save(student);

}

public List<Student> getAllStudents() {

return studentRepo.findAll();

}

}

**Configure application.properties**

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

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.hibernate.ddl-auto=update

spring.h2.console.enabled=true

**Run the App**

Run the main class DemoApplication.java, then:

{

"name": "Harshini",

"department": "CSE"

}

**S.NO:2 Difference between JPA, Hibernate and Spring Data JPA**

**SOLUTION:**

### 1. ****JPA (Java Persistence API)****

**It is a specification (interface)** – not an implementation.

* **Developed by**: Oracle as part of Java EE.
* **Goal:** To provide a **standard way** to map Java objects to relational databases (ORM).
* **Contains**: Interfaces like EntityManager, @Entity, @Table, @Id, etc.
* **Cannot be used alone** – needs an implementation (like Hibernate).

### 2. ****Hibernate****

**It is an implementation of JPA Developed by:** Red Hat.

* **Implements**: The JPA specification (and more).
* **Can be used without JPA** using its native API (Session, Query, etc).
* **Handles**: Object-relational mapping, caching, transaction management.

### 3. ****Spring Data JPA****

**It is a Spring-based abstraction** over JPA (and typically Hibernate).

* **Part of:** Spring Data project.
* **Goal**: Simplify JPA-based data access using **repository interfaces** – no need to write boilerplate code.
* **Automatically implements** methods like save(), findAll(), findById() just by extending JpaRepository.