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

Hibernate Implementation

<dependencies>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.6.15.Final</version>

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<version>2.1.214</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-simple</artifactId>

<version>1.7.36</version>

</dependency>

</dependencies>

**Employee.java**

package com.example;

import javax.persistence.\*;

@Entity

public class Employee {

@Id

@GeneratedValue

private int id;

private String name;

private String role;

public int getId() { return id; }

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

public String getName() { return name; }

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

public String getRole() { return role; }

public void setRole(String role) { this.role = role; }

}

**Main.java**

package com.example;

import org.hibernate.\*;

import org.hibernate.cfg.Configuration;

public class Main {

public static void main(String[] args) {

SessionFactory factory = new Configuration().configure().buildSessionFactory();

Session session = factory.openSession();

Transaction tx = null;

try {

tx = session.beginTransaction();

Employee emp = new Employee();

emp.setName("Alice");

emp.setRole("Developer");

session.save(emp);

tx.commit();

System.out.println("Employee saved with ID: " + emp.getId());

} catch (Exception e) {

if (tx != null) tx.rollback();

e.printStackTrace();

} finally {

session.close();

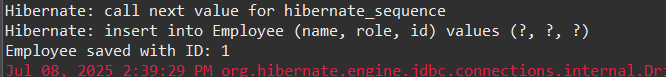
factory.close();

}

}

}

OUTPUT :



**Spring Data JPA Implementation**

<dependencies>

<dependency>

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

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

<version>2.7.13</version>

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

</dependencies>

**Employee.java**

package com.example.entity;

import javax.persistence.\*;

@Entity

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Integer id;

private String name;

private String role;

// Getters and setters

public Integer getId() { return id; }

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

public String getName() { return name; }

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

public String getRole() { return role; }

public void setRole(String role) { this.role = role; }

}

**EmployeeRepository.java**

package com.example.repository;

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

import com.example.entity.Employee;

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**EmployeeService.java**

package com.example.service;

import com.example.entity.Employee;

import com.example.repository.EmployeeRepository;

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

import org.springframework.stereotype.Service;

import javax.transaction.Transactional;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee);

System.out.println("Employee saved with Spring Data JPA");

}

}

**SpringDataJpaApp.java**

package com.example;

import com.example.entity.Employee;

import com.example.service.EmployeeService;

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

import org.springframework.boot.\*;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringDataJpaApp implements CommandLineRunner {

@Autowired

private EmployeeService service;

public static void main(String[] args) {

SpringApplication.run(SpringDataJpaApp.class, args);

}

@Override

public void run(String... args) throws Exception {

Employee emp = new Employee();

emp.setName("Bob");

emp.setRole("Tester");

service.addEmployee(emp);

}

}

OUTPUT :

