**HandsOn 1.Spring Data JPA - Quick Example**

**SQL:**

create table country(co\_code varchar(2) primary key, co\_name varchar(50));

insert into country values ('IN', 'India');

insert into country values ('US', 'United States of America');

**1.Persistence Class - com.cognizant.orm-learn.model.Country:**

package com.cognizant.ormlearn.model.Country;

import jakarta.persistence.\*;

@Entity

@Table(name="country")

public class Country {

@Id

@Column(name = "co\_code")

private String code;

@Column(name = "co\_name")

private String name;

// Getters & Setters

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

@Override

public String toString() {

return "Country{" + "code='" + code + "', name='" + name + "'}";

}

}

**2.Repository Class - com.cognizant.orm-learn.CountryRepository:**

package com.cognizant.ormlearn.CountryRepository;

import com.cognizant.ormlearn.model.Country.Country;

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

import org.springframework.stereotype.Repository;

@Repository

public interface CountryRepository extends JpaRepository<Country, String> {

}

**3.Service Class - com.cognizant.orm-learn.service.CountryService:**

package com.cognizant.ormlearn.Service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.cognizant.ormlearn.CountryRepository.CountryRepository;

import com.cognizant.ormlearn.model.Country.Country;

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

public void displayAll() {

List<Country> conlist = countryRepository.findAll();

conlist.forEach(System.out::println);

}

}

**4.Testing in OrmLearnApplication.java:**

package com.cognizant.ormlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

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

import org.springframework.context.ApplicationContext;

import com.cognizant.ormlearn.Service.CountryService;

@SpringBootApplication

public class OrmLearnApplication {

@Autowired

private static CountryService serve;

private static final Logger LOGGER = LoggerFactory.getLogger(OrmLearnApplication.class);

public static void main(String[] args) {

ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);

serve = context.getBean(CountryService.class);

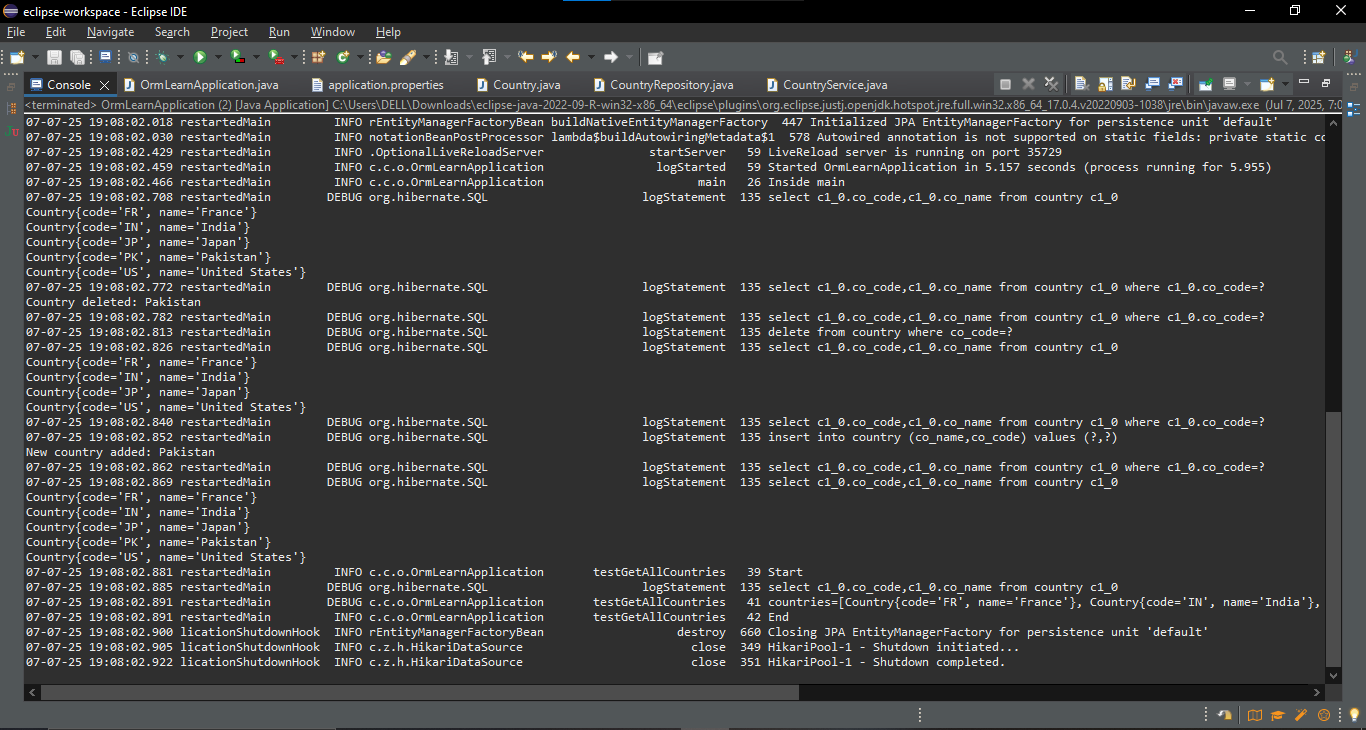
LOGGER.info("Inside main");

serve.displayAll();

}

}

**OUTPUT:**

****

**HandsOn 2. Difference between JPA, Hibernate and Spring Data JPA**

**1.Pom.xml:**

<dependencies>

<!-- Spring Boot Starter Data JPA -->

<dependency>

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

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

</dependency>

<!-- Hibernate (comes transitively via Spring Data JPA) -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

</dependency>

<!-- H2 Database for testing -->

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

<!-- Spring Boot Starter Web -->

<dependency>

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

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

</dependency>

</dependencies>

**2.Employee.java:**

import jakarta.persistence.\*;

@Entity

@Table(name = "employees")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Integer id;

private String name;

private String department;

}

**3.HibernateUtil.java:**

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

public class HibernateUtil {

private static final SessionFactory sessionFactory;

static {

try {

sessionFactory = new Configuration().configure("hibernate.cfg.xml").addAnnotatedClass(Employee.class).buildSessionFactory(); } catch (Throwable ex) {

throw new ExceptionInInitializerError(ex);

}

}

public static SessionFactory getSessionFactory() {

return sessionFactory; } }

**4.EmployeeDAO.java:**

import org.hibernate.\*;

public class EmployeeDAO {

public Integer addEmployee(Employee employee) {

Session session = HibernateUtil.getSessionFactory().openSession();

Transaction tx = null;

Integer employeeID = null;

try {

tx = session.beginTransaction();

employeeID = (Integer) session.save(employee);

tx.commit();

} catch (HibernateException e) {

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

e.printStackTrace();

} finally {

session.close();

}

return employeeID; } }

**5.EmployeeRepository.java:**

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

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

EmployeeService.java

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee);

}

}

**6.EmployeeController.java:**

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

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

@RestController

@RequestMapping("/api/employees")

public class EmployeeController {

@Autowired

private EmployeeService employeeService;

@PostMapping

public String createEmployee(@RequestBody Employee employee) {

employeeService.addEmployee(employee);

return "Employee saved successfully!";

}

}

**7.application.properties:**

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

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=root

spring.datasource.password=root

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

spring.jpa.hibernate.ddl-auto=update

spring.h2.console.enabled=true

**OUTPUT:**

