**Hands on 1**

**Spring Data JPA - Quick Example**

**SOURCE CODE:**

**OrmLearnApplication.java**

package com.cognizant.orm\_learn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import com.cognizant.orm\_learn.model.Country;

import com.cognizant.orm\_learn.service.CountryServices;

import java.util.List;

*@SpringBootApplication*

public class OrmLearnApplication {

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

private static CountryServices *countryService*;

public static void main(String[] args) throws Exception {

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

*countryService* = context.getBean(CountryServices.class);

*testGetAllCountries*();

*testFindCountryByCode*();

*testAddCountry*();

*testUpdateCountry*();

*testDeleteCountry*();

}

private static void testGetAllCountries() {

*LOGGER*.info("Start");

List<Country> countries = *countryService*.getAllCountries();

*LOGGER*.debug("Countries: {}", countries);

*LOGGER*.info("End");

}

private static void testFindCountryByCode() throws Exception {

Country country = *countryService*.findCountryByCode("IN");

*LOGGER*.debug("Found: {}", country);

}

private static void testAddCountry() throws Exception {

Country newCountry = new Country();

newCountry.setCode("ZZ");

newCountry.setName("TestLand");

*countryService*.addCountry(newCountry);

*LOGGER*.debug("Added: {}", *countryService*.findCountryByCode("ZZ"));

}

private static void testUpdateCountry() throws Exception {

*countryService*.updateCountry("ZZ", "UpdatedLand");

*LOGGER*.debug("Updated: {}", *countryService*.findCountryByCode("ZZ"));

}

private static void testDeleteCountry() throws Exception {

*countryService*.deleteCountry("ZZ");

*LOGGER*.debug("Deleted ZZ country");

}

}

**Country.java:**

package com.cognizant.orm\_learn.model;

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; }

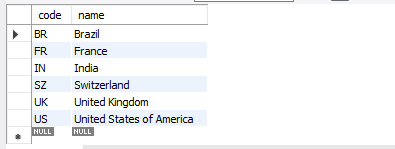
*@Override*

public String toString() {

return "Country [code=" + code + ", name=" + name + "]";

}

}



**CountryRepository.java:**

package com.cognizant.orm\_learn.repository;

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

import org.springframework.stereotype.Repository;

import com.cognizant.orm\_learn.model.Country;

*@Repository*

public interface CountryRepository extends JpaRepository<Country, String> {

}

**CountryServices.java:**

package com.cognizant.orm\_learn.service;

import java.util.List;

import java.util.Optional;

import org.springframework.transaction.annotation.Transactional;

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

import org.springframework.stereotype.Service;

import com.cognizant.orm\_learn.model.Country;

import com.cognizant.orm\_learn.repository.CountryRepository;

*@Service*

public class CountryServices {

*@Autowired*

private CountryRepository countryRepository;

*@Transactional*

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

*@Transactional*

public Country findCountryByCode(String code) throws Exception {

Optional<Country> result = countryRepository.findById(code);

if (!result.isPresent()) {

throw new Exception("Country not found");

}

return result.get();

}

*@Transactional*

public void addCountry(Country country) {

countryRepository.save(country);

}

*@Transactional*

public void updateCountry(String code, String name) throws Exception {

Country country = findCountryByCode(code);

country.setName(name);

countryRepository.save(country);

}

*@Transactional*

public void deleteCountry(String code) {

countryRepository.deleteById(code);

}

}

**application.properties.java:**

# DB config

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn

spring.datasource.username=root

spring.datasource.password=Shriya43@!

# Hibernate config

spring.jpa.hibernate.ddl-auto=validate

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect

# Logging

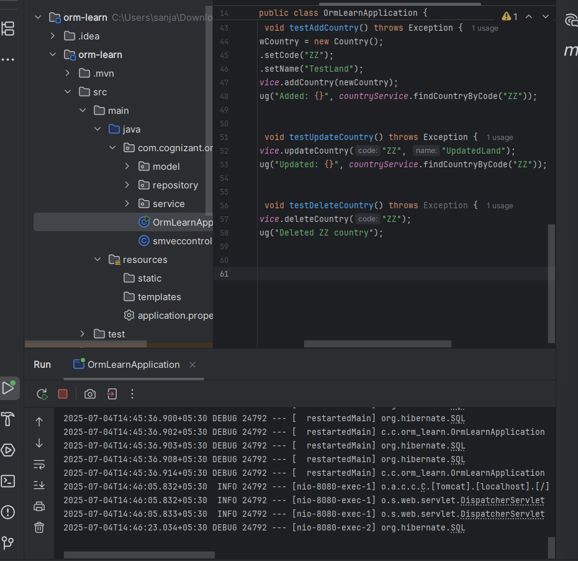
logging.level.org.springframework=info

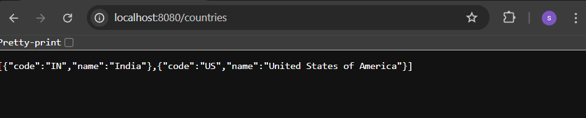
logging.level.com.cognizant=debug

logging.level.org.hibernate.SQL=trace

logging.level.org.hibernate.type.descriptor.sql=trace

**OUTPUT:**

****

****

**Hands on 4**

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

**SOURCE CODE:**

**Employee.java**

package com.cognizant.orm\_learn.model;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

*@Entity*

*@Table*(name = "employee")

public class Employee {

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.*IDENTITY*)

private Integer id;

private String name;

private String department;

private Double salary;

// Constructors

public Employee() {

}

public Employee(String name, String department, Double salary) {

this.name = name;

this.department = department;

this.salary = salary;

}

// 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 getDepartment() {

return department;

}

public void setDepartment(String department) {

this.department = department;

}

public Double getSalary() {

return salary;

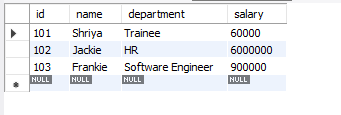
}

public void setSalary(Double salary) {

this.salary = salary;

}

}

****

**EmployeeRepository.java:**

package com.example.employee\_app.repository;

import com.example.employee\_app.model.Employee;

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

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**EmployeeService.java:**

package com.example.employee\_app.service;

import com.example.employee\_app.model.Employee;

import com.example.employee\_app.repository.EmployeeRepository;

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);

    }

}

**EmployeeAppApplication.java:**

package com.example.employee\_app;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class EmployeeAppApplication {

    public static void main(String[] args) {

        SpringApplication.run(EmployeeAppApplication.class, args);

    }

}

**Application.properties:**

spring.datasource.url=jdbc:mysql://localhost:3306/employee\_db

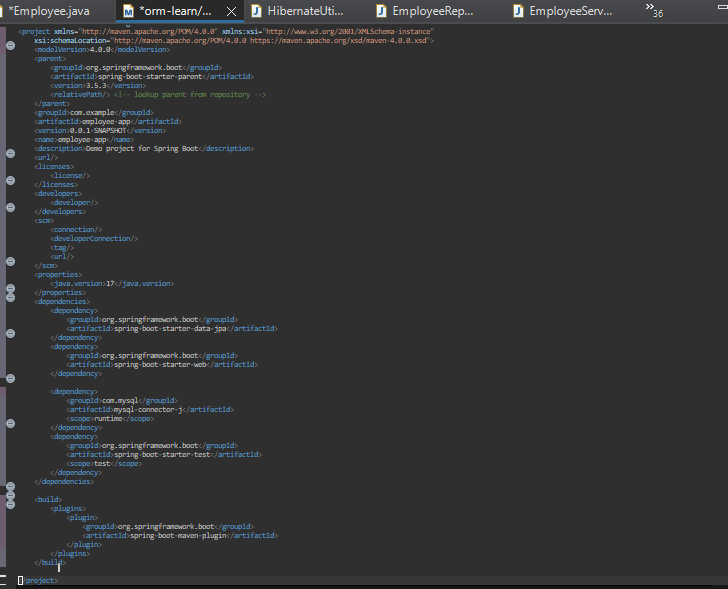
spring.datasource.username=root

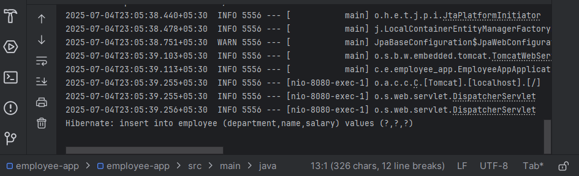
spring.datasource.password=Shriya43@!

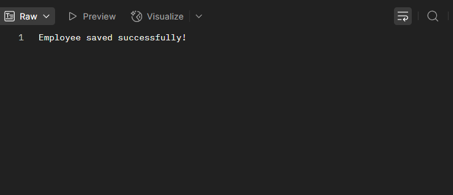
spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect



****

****