**Hands on 5 : Implement services for managing Country** 

**OrmLearnApplication.java :-**

package com.cognizant.orm\_learn;

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

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

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 OrmLearnApplication implements CommandLineRunner {

*@Autowired*

private CountryService countryService;

public static void main(String[] args) {

SpringApplication.*run*(OrmLearnApplication.class, args);

}

*@Override*

public void run(String... args) {

// Add a country

Country india = new Country("IN", "India");

countryService.addCountry(india);

// Find by code

System.***out***.println(" Find IN: " + countryService.findByCode("IN"));

// Update

india.setName("Bharat");

countryService.updateCountry(india);

System.***out***.println(" Updated IN: " + countryService.findByCode("IN"));

// Search

System.***out***.println(" Search 'Bha': " + countryService.searchByPartialName("Bha"));

// Delete

countryService.deleteCountry("IN");

System.***out***.println(" Deleted IN: " + countryService.findByCode("IN"));

}

}

**Country.java :-**

package com.cognizant.orm\_learn.model;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

*@Entity*

*@Table*(name = "country")

public class Country {

*@Id*

*@Column*(name = "co\_code")

private String code;

*@Column*(name = "co\_name")

private String name;

// Constructors

public Country() {}

public Country(String code, String name) {

this.code = code;

this.name = name;

}

// Getters and 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 java.util.List;

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> {

List<Country> findByNameContainingIgnoreCase(String partialName);

}

**CountryService.java :-**

package com.cognizant.orm\_learn.service;

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

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

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

import org.springframework.stereotype.Service;

import java.util.List;

*@Service*

public class CountryService {

*@Autowired*

private CountryRepository repository;

public Country findByCode(String code) {

return repository.findById(code).orElse(null);

}

public Country addCountry(Country country) {

return repository.save(country);

}

public Country updateCountry(Country country) {

if (repository.existsById(country.getCode())) {

return repository.save(country);

}

return null;

}

public boolean deleteCountry(String code) {

if (repository.existsById(code)) {

repository.deleteById(code);

return true;

}

return false;

}

public List<Country> searchByPartialName(String partialName) {

return repository.findByNameContainingIgnoreCase(partialName);

}

}

**Application.properties :-**

logging.level.org.springframework=info

logging.level.com.cognizant=debug

logging.level.org.hibernate.SQL=debug

# DB Configuration

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=root

# Hibernate Dialect

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

# DDL behavior: none, update, create, create-drop, validate

spring.jpa.hibernate.ddl-auto=validate

Output :-



