**Demonstrate implementation of Query Methods feature of Spring Data JPA & Demonstrate implementation of O/R Mapping**

**CountryController.java**

package com.cognizant.ormlearn.controller;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.service.CountryService;

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

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

import java.util.List;

*@RestController*

*@RequestMapping*("/countries")

public class CountryController {

*@Autowired*

private CountryService countryService;

*@GetMapping*("/{code}")

public Country getCountryByCode(*@PathVariable* String code) {

return countryService.findCountryByCode(code);

}

*@PostMapping*

public Country addCountry(*@RequestBody* Country country) {

return countryService.addCountry(country);

}

*@PutMapping*

public Country updateCountry(*@RequestBody* Country country) {

return countryService.updateCountry(country);

}

*@DeleteMapping*("/{code}")

public void deleteCountry(*@PathVariable* String code) {

countryService.deleteCountry(code);

}

*@GetMapping*

public List<Country> getAllCountries() {

return countryService.getAllCountries();

}

*@GetMapping*("/search/{name}")

public List<Country> searchByPartialName(*@PathVariable* String name) {

return countryService.findCountriesByPartialName(name);

}

*@GetMapping*("/search-hql/{name}")

public List<Country> searchByHql(*@PathVariable* String name) {

return countryService.searchByNameHql(name);

}

*@GetMapping*("/search-native/{name}")

public List<Country> searchByNative(*@PathVariable* String name) {

return countryService.searchByNameNative(name);

}

}

**CountryService.java**

package com.cognizant.ormlearn.service;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.service.exception.CountryNotFoundException;

import java.util.List;

public interface CountryService {

Country findCountryByCode(String code) throws CountryNotFoundException;

Country addCountry(Country country);

Country updateCountry(Country country);

void deleteCountry(String code);

List<Country> findCountriesByPartialName(String name);

List<Country> getAllCountries();

List<Country> searchByNameHql(String keyword);

List<Country> searchByNameNative(String keyword);

}

**CountryServiceImpl.java**

package com.cognizant.ormlearn.service;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.repository.CountryRepository;

import com.cognizant.ormlearn.service.exception.CountryNotFoundException;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import java.util.List;

*@Service*

public class CountryServiceImpl implements CountryService {

*@Autowired*

private CountryRepository countryRepository;

*@Override*

*@Transactional*(readOnly = true)

public Country findCountryByCode(String code) throws CountryNotFoundException {

return countryRepository.findById(code)

.orElseThrow(() -> new CountryNotFoundException("Country not found: " + code));

}

*@Override*

*@Transactional*

public Country addCountry(Country country) {

return countryRepository.save(country);

}

*@Override*

*@Transactional*

public Country updateCountry(Country country) throws CountryNotFoundException {

if (!countryRepository.existsById(country.getCode())) {

throw new CountryNotFoundException("Country not found: " + country.getCode());

}

return countryRepository.save(country);

}

*@Override*

*@Transactional*

public void deleteCountry(String code) throws CountryNotFoundException {

if (!countryRepository.existsById(code)) {

throw new CountryNotFoundException("Country not found: " + code);

}

countryRepository.deleteById(code);

}

*@Override*

*@Transactional*(readOnly = true)

public List<Country> findCountriesByPartialName(String name) {

return countryRepository.findByNameContainingIgnoreCase(name);

}

*@Override*

*@Transactional*(readOnly = true)

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

*@Override*

public List<Country> searchByNameHql(String keyword) {

return countryRepository.searchByNameHql(keyword);

}

*@Override*

public List<Country> searchByNameNative(String keyword) {

return countryRepository.searchByNameNative(keyword);

}

}

**CountryNotFoundException.java**

package com.cognizant.ormlearn.service.exception;

public class CountryNotFoundException extends RuntimeException {

public CountryNotFoundException(String message) {

super(message);

}

}

**Country.java**

package com.cognizant.ormlearn.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;

// 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 + "]";

}

}

**OrmLearnApplication.java**

package com.cognizant.ormlearn;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.service.CountryService;

import com.cognizant.ormlearn.service.exception.CountryNotFoundException;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import java.util.List;

*@SpringBootApplication*

public class OrmLearnApplication implements CommandLineRunner {

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

*@Autowired*

private CountryService countryService;

public static void main(String[] args) {

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

}

*@Override*

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

testFindByCode();

getAllCountriesTest();

testAddCountry();

testSearchByHql();

testSearchByNative();

}

private void testFindByCode() {

try {

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

***LOGGER***.info("Country: {}", country);

} catch (CountryNotFoundException e) {

***LOGGER***.error("Country not found: IN");

}

}

private void getAllCountriesTest() {

***LOGGER***.info("Fetching all countries...");

countryService.getAllCountries().forEach(c -> ***LOGGER***.info("Country: {}", c));

}

private void testAddCountry() {

Country newCountry = new Country();

newCountry.setCode("ZZ");

newCountry.setName("Zootopia");

countryService.addCountry(newCountry);

***LOGGER***.info("Added new country: {}", newCountry);

}

private void testSearchByHql() {

List<Country> results = countryService.searchByNameHql("land");

***LOGGER***.info("HQL Search Results:");

results.forEach(c -> ***LOGGER***.info("{}", c));

}

private void testSearchByNative() {

List<Country> results = countryService.searchByNameNative("land");

***LOGGER***.info("Native SQL Search Results:");

results.forEach(c -> ***LOGGER***.info("{}", c));

}

}

Output :



