**Hands on 1**

**Write queries on country table using Query Methods**

**Demonstrate implementation of Query Methods feature of Spring Data JPA**

**application.properties**

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

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

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

logging.level.org.hibernate.SQL=debug

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

**Country.java**

package com.example.countryapp.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

@Entity

@Table(name = "country")

public class Country {

@Id

private String code;

private String name;

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.example.countryapp.repository;

import com.example.countryapp.model.Country;

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

import org.springframework.stereotype.Repository;

import java.util.List;

@Repository

public interface CountryRepository extends JpaRepository<Country, String> {

List<Country> findByNameContainingIgnoreCase(String name);

List<Country> findByNameContainingIgnoreCaseOrderByNameAsc(String name);

List<Country> findByNameStartingWith(String prefix);

}

**CountryService.java**

package com.example.countryapp.service;

import com.example.countryapp.model.Country;

import java.util.List;

public interface CountryService {

List<Country> searchCountriesByName(String name);

List<Country> searchCountriesByNameSorted(String name);

List<Country> searchCountriesByStartingLetter(String prefix);

}

**CountryServiceImp.java**

package com.example.countryapp.service;

import com.example.countryapp.model.Country;

import com.example.countryapp.repository.CountryRepository;

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

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class CountryServiceImpl implements CountryService {

@Autowired

private CountryRepository countryRepository;

@Override

public List<Country> searchCountriesByName(String name) {

return countryRepository.findByNameContainingIgnoreCase(name);

}

@Override

public List<Country> searchCountriesByNameSorted(String name) {

return countryRepository.findByNameContainingIgnoreCaseOrderByNameAsc(name);

}

@Override

public List<Country> searchCountriesByStartingLetter(String prefix) {

return countryRepository.findByNameStartingWith(prefix);

}

}

**CountryAppApplication.java**

package com.example.countryapp;

import com.example.countryapp.model.Country;

import com.example.countryapp.service.CountryService;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import java.util.List;

@SpringBootApplication

public class CountryAppApplication implements CommandLineRunner {

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

@Autowired

private CountryService countryService;

public static void main(String[] args) {

SpringApplication.run(CountryAppApplication.class, args);

}

@Override

public void run(String... args) {

LOGGER.info("Search by name 'ou'");

List<Country> countries1 = countryService.searchCountriesByName("ou");

countries1.forEach(country -> System.out.println(country.getCode() + " " + country.getName()));

LOGGER.info("Search by name 'ou' (sorted)");

List<Country> countries2 = countryService.searchCountriesByNameSorted("ou");

countries2.forEach(country -> System.out.println(country.getCode() + " " + country.getName()));

LOGGER.info("Search by name starting with 'Z'");

List<Country> countries3 = countryService.searchCountriesByStartingLetter("Z");

countries3.forEach(country -> System.out.println(country.getCode() + " " + country.getName()));

}

}



