**WEEK 3:** **1. spring-data-jpa-handson**

**Hands on 1:-**

**Spring Data JPA - Quick Example**

**Create a Eclipse Project using Spring Initializer**

**Code:**

**application.properties**

logging.level.root=OFF

logging.level.org.hibernate.SQL=OFF

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

logging.level.org.springframework=OFF

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

spring.jpa.hibernate.ddl-auto=validate

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

**Country.java**

package com.cognizant.ormlearn.model;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "country")

public class Country {

@Id

@Column(name = "code")

private String code;

@Column(name = "name")

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.cognizant.ormlearn.repository;

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

import org.springframework.stereotype.Repository;

import com.cognizant.ormlearn.model.Country;

@Repository

public interface CountryRepository extends JpaRepository<Country, String> {

}

**CountryService.java**

package com.cognizant.ormlearn.service;

import java.util.List;

import javax.transaction.Transactional;

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

import org.springframework.stereotype.Service;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.repository.CountryRepository;

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

@Transactional

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

}

**OrmLearnApplication.java**

package com.cognizant.ormlearn;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.service.CountryService;

import java.util.List;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

@SpringBootApplication

public class OrmLearnApplication {

private static CountryService countryService;

public static void main(String[] args) {

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

countryService = context.getBean(CountryService.class);

testGetAllCountries();

}

private static void testGetAllCountries() {

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

System.out.println("Countries: " + countries);

}

}

**MySQL Schema**

CREATE SCHEMA ormlearn;

USE ormlearn;

CREATE TABLE country (

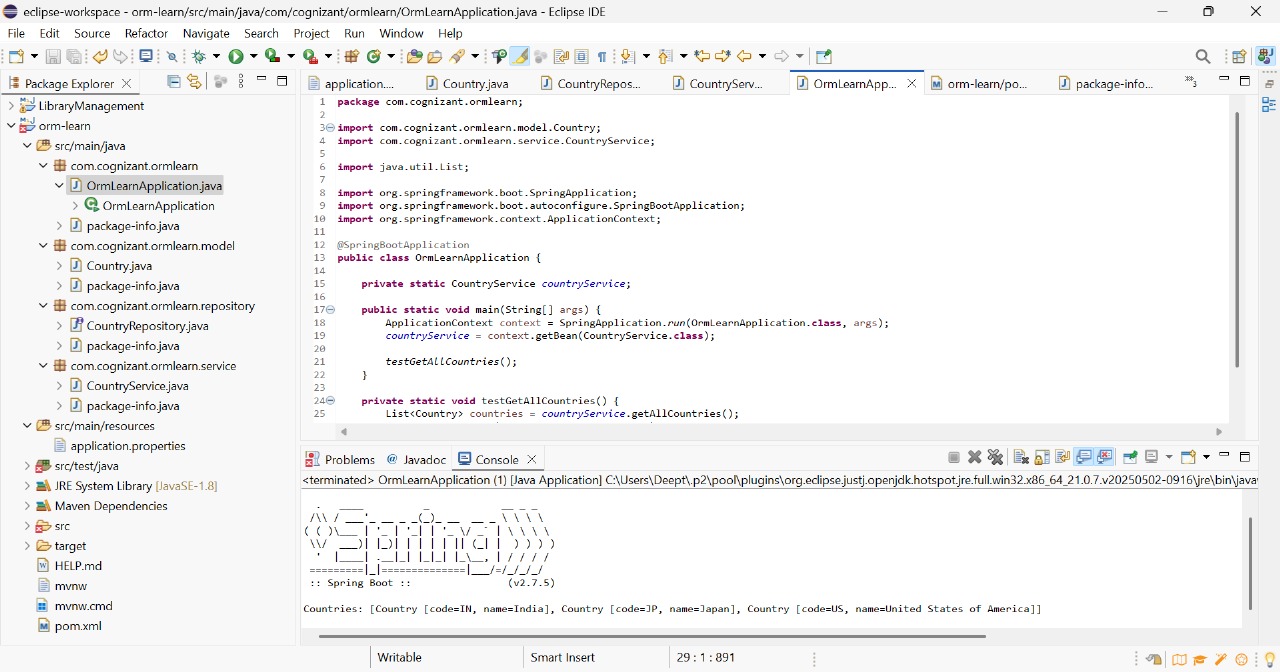
code VARCHAR(2) PRIMARY KEY,

name VARCHAR(50)

);

INSERT INTO country VALUES ('IN', 'India'), ('JP', 'Japan'), ('US', 'United States of America');

**Output:**



**Hands on 4**

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

**Code:**

**Employee.java**

package com.example.employeedemo.model;

import jakarta.persistence.\*;

@Entity

@Table(name = "employee")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String department;

private double salary;

public Employee() {}

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

this.name = name;

this.department = department;

this.salary = salary;

}

public Long getId() {

return id;

}

public void setId(Long 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.employeedemo.repository;

import com.example.employeedemo.model.Employee;

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

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

}

**EmployeeService.java**

package com.example.employeedemo.service;

import com.example.employeedemo.model.Employee;

import com.example.employeedemo.repository.EmployeeRepository;

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

import org.springframework.stereotype.Service;

import java.util.List;

import java.util.Optional;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

public List<Employee> getAllEmployees() {

return employeeRepository.findAll();

}

public Optional<Employee> getEmployeeById(Long id) {

return employeeRepository.findById(id);

}

public Employee addEmployee(Employee employee) {

return employeeRepository.save(employee);

}

public void deleteEmployee(Long id) {

employeeRepository.deleteById(id);

}

}

**EmployeeController.java**

package com.example.employeedemo.controller;

import com.example.employeedemo.model.Employee;

import com.example.employeedemo.service.EmployeeService;

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

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

import java.util.List;

import java.util.Optional;

@RestController

@RequestMapping("/employees")

public class EmployeeController {

@Autowired

private EmployeeService employeeService;

@GetMapping

public List<Employee> getAllEmployees() {

return employeeService.getAllEmployees();

}

@GetMapping("/{id}")

public Optional<Employee> getEmployeeById(@PathVariable Long id) {

return employeeService.getEmployeeById(id);

}

@PostMapping

public Employee addEmployee(@RequestBody Employee employee) {

return employeeService.addEmployee(employee);

}

@DeleteMapping("/{id}")

public void deleteEmployee(@PathVariable Long id) {

employeeService.deleteEmployee(id);

}

}

**EmployeeDemoApplication.java**

package com.example.employeedemo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class EmployeeDemoApplication {

public static void main(String[] args) {

SpringApplication.run(EmployeeDemoApplication.class, args);

}

}

**application.properties**

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

spring.datasource.username=root

spring.datasource.password=root

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

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

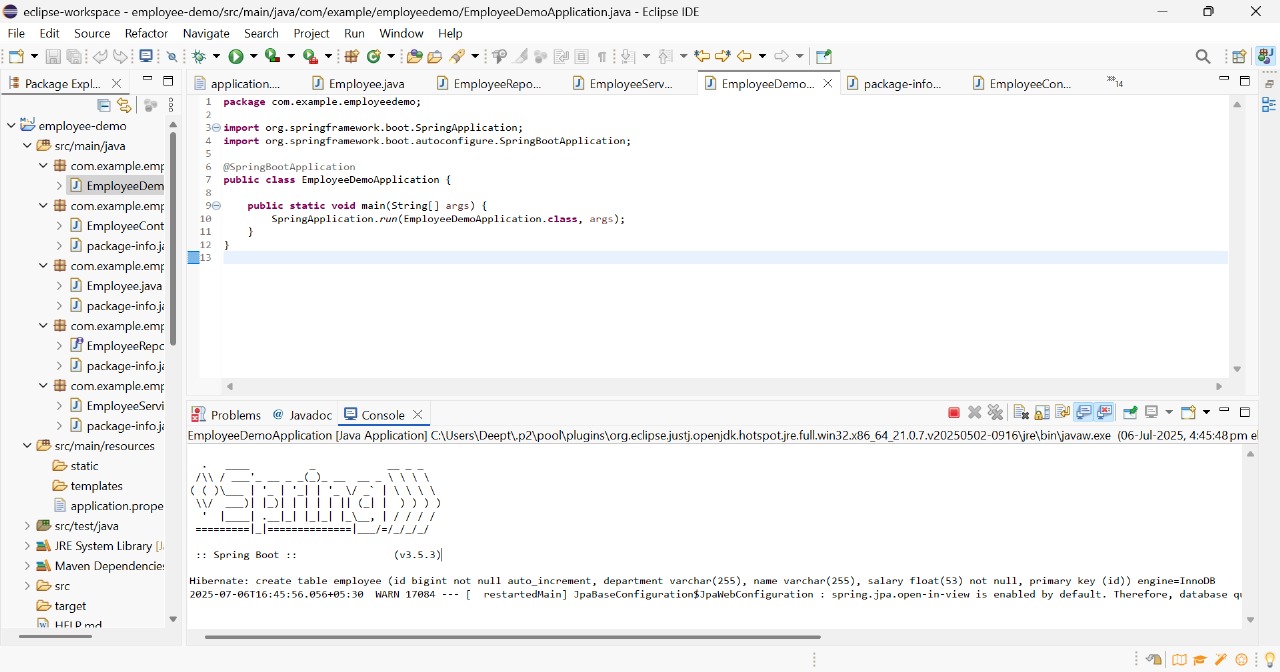
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

logging.level.root=WARN

logging.level.org.springframework.web=INFO

logging.level.org.hibernate=ERROR

**Output:**



**Hands on 5**

**Implement services for managing Country   
Code:**

**Country.java**

package com.example.countrydemo.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

@Entity

public class Country {

@Id

private String code;

private String name;

public Country() {}

public Country(String code, String name) {

this.code = code;

this.name = 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;

}

}

**CountryRepository.java**

package com.example.countrydemo.repository;

import java.util.List;

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

import org.springframework.stereotype.Repository;

import com.example.countrydemo.model.Country;

@Repository

public interface CountryRepository extends JpaRepository<Country, String> {

List<Country> findByNameContainingIgnoreCase(String name);

}

**CountryService.java**

package com.example.countrydemo.service;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import com.example.countrydemo.model.Country;

import com.example.countrydemo.repository.CountryRepository;

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

public Country addCountry(Country country) {

return countryRepository.save(country);

}

public Optional<Country> getCountryByCode(String code) {

return countryRepository.findById(code);

}

public void deleteCountry(String code) {

countryRepository.deleteById(code);

}

public List<Country> searchCountriesByName(String name) {

return countryRepository.findByNameContainingIgnoreCase(name);

}

}

**CommandLineAppStartupRunner.java**

package com.example.countrydemo;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.stereotype.Component;

import com.example.countrydemo.model.Country;

import com.example.countrydemo.service.CountryService;

@Component

public class CommandLineAppStartupRunner implements CommandLineRunner {

@Autowired

private CountryService countryService;

@Override

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

countryService.addCountry(new Country("IN", "India"));

countryService.addCountry(new Country("US", "United States"));

System.out.println("All countries:");

countryService.getAllCountries().forEach(c ->

System.out.println(c.getCode() + " - " + c.getName())

);

System.out.println("Search by name containing 'ind':");

countryService.searchCountriesByName("ind").forEach(c ->

System.out.println(c.getCode() + " - " + c.getName())

);

}

}

**CountryDemoApplication.java**

package com.example.countrydemo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class CountryDemoApplication {

public static void main(String[] args) {

SpringApplication.run(CountryDemoApplication.class, args);

}

}

**application.properties**

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

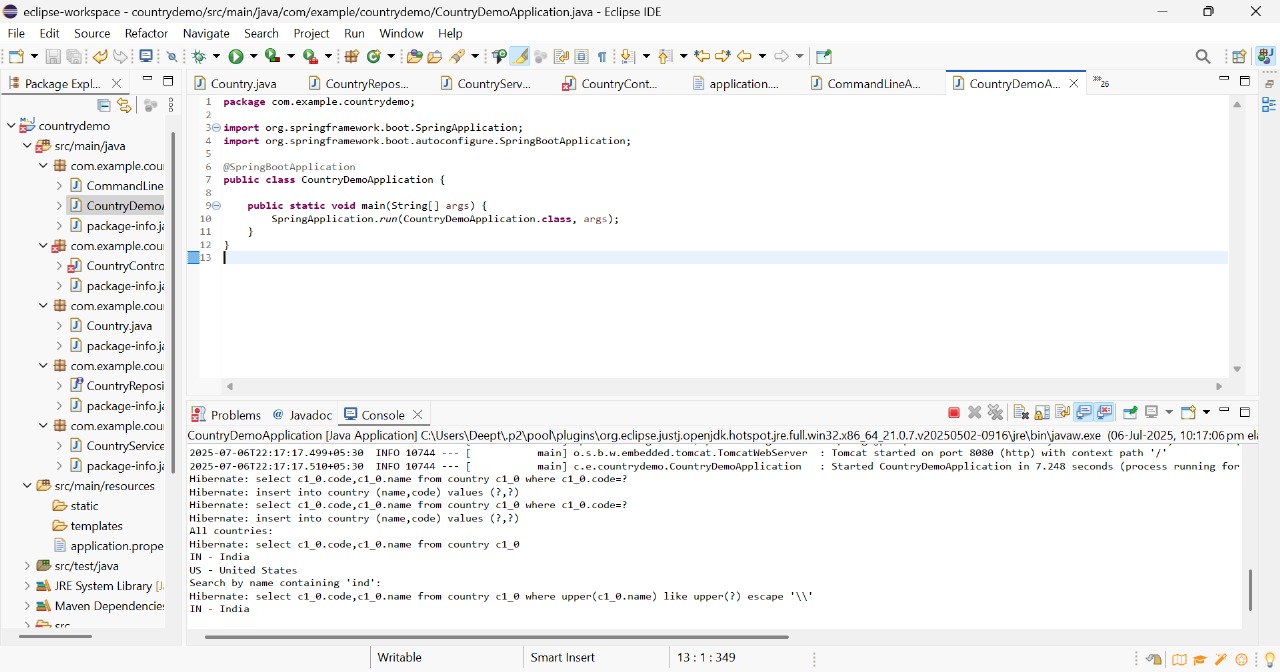
spring.datasource.username=root

spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

**Output:**



**Hands on 6**

**Find a country based on country code**

**Code:**

**Country.java**

package com.cognizant.springlearn.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.cognizant.springlearn.repository;

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

import com.cognizant.springlearn.model.Country;

public interface CountryRepository extends JpaRepository<Country, String> {

}

**CountryNotFoundException.java**

package com.cognizant.springlearn.service.exception;

public class CountryNotFoundException extends Exception {

public CountryNotFoundException(String message) {

super(message);

}

}

**CountryService.java**

package com.cognizant.springlearn.service;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.cognizant.springlearn.model.Country;

import com.cognizant.springlearn.repository.CountryRepository;

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

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

@Transactional

public Country findCountryByCode(String countryCode) throws CountryNotFoundException {

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

if (!result.isPresent()) {

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

}

return result.get();

}

}

**OrmLearnApplication.java**

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.autoconfigure.domain.EntityScan;

import org.springframework.context.ApplicationContext;

import com.cognizant.springlearn.model.Country;

import com.cognizant.springlearn.service.CountryService;

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

@SpringBootApplication

@EntityScan("com.cognizant.springlearn.model")

public class OrmLearnApplication {

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

private static CountryService countryService;

public static void main(String[] args) {

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

countryService = context.getBean(CountryService.class);

try {

getCountryByCodeTest();

} catch (CountryNotFoundException e) {

LOGGER.error("Exception: {}", e.getMessage());

}

}

private static void getCountryByCodeTest() throws CountryNotFoundException {

LOGGER.info("Start");

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

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

LOGGER.info("End");

}

}

**application.properties**

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

spring.datasource.username=root

spring.datasource.password=root

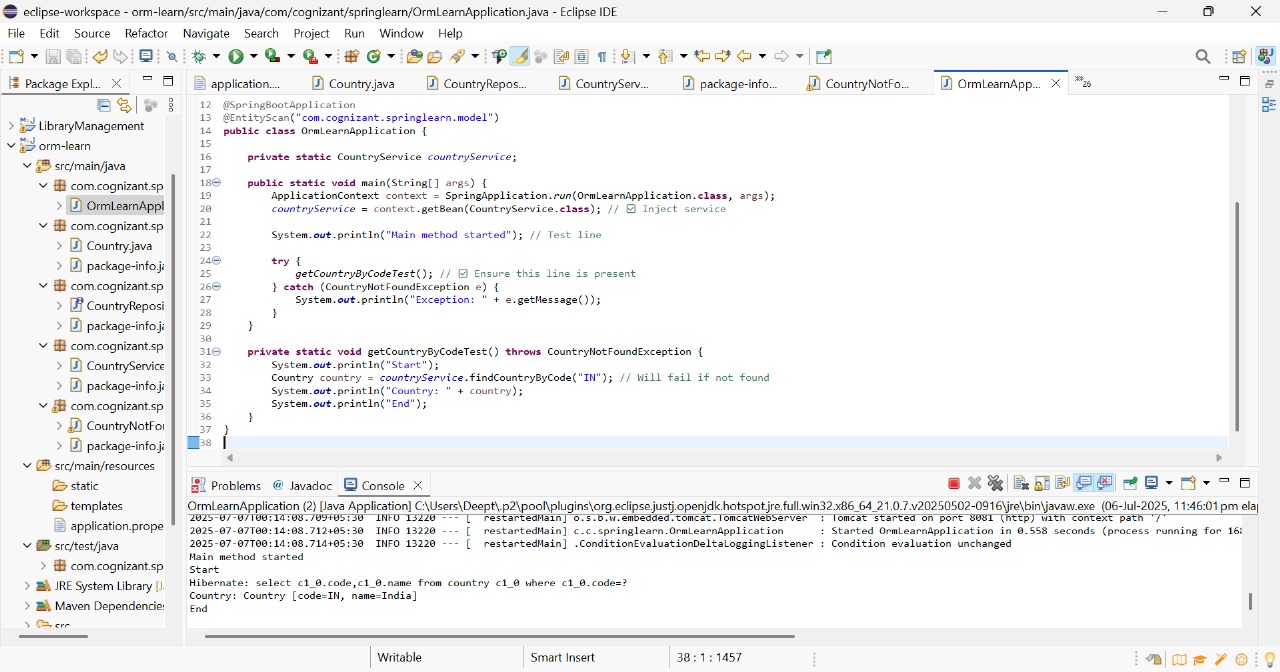
spring.jpa.hibernate.ddl-auto=validate

spring.jpa.show-sql=true

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

logging.level.com.cognizant=DEBUG

**Output:**



**Hands on 7**

**Add a new country**

**Code:**

**CountryService.java**

package com.cognizant.springlearn.service;

import com.cognizant.springlearn.model.Country;

import com.cognizant.springlearn.repository.CountryRepository;

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

import jakarta.transaction.Transactional;

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

import org.springframework.stereotype.Service;

import java.util.Optional;

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

public Country findCountryByCode(String code) throws CountryNotFoundException {

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

if (country.isEmpty()) {

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

}

return country.get();

}

@Transactional

public void addCountry(Country country) {

countryRepository.save(country);

}

}

**OrmLearnApplication.java**

package com.cognizant.springlearn;

import com.cognizant.springlearn.model.Country;

import com.cognizant.springlearn.service.CountryService;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.Bean;

@SpringBootApplication

public class OrmLearnApplication {

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Bean

public CommandLineRunner run(ApplicationContext context) {

return args -> {

System.out.println("Start");

CountryService countryService = context.getBean(CountryService.class);

testAddCountry(countryService);

System.out.println("End");

};

}

private void testAddCountry(CountryService countryService) {

Country country = new Country();

country.setCode("ZZ");

country.setName("Zootopia");

countryService.addCountry(country);

try {

Country retrieved = countryService.findCountryByCode("ZZ");

System.out.println("Added Country: " + retrieved);

} catch (CountryNotFoundException e) {

System.out.println("Country not found after adding: " + e.getMessage());

}

}

}

**Output:**

