**SUPERSET ID : 6375627**

**WEEK 3 Spring Data JPA with Spring Boot, Hibernate**

**Implement services for managing Country**

**Country.java**  
package com.cognizant.ormlearn;

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;

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 + "]";

}

}

**CountryNotFoundException.java**  
package com.cognizant.ormlearn;

public class CountryNotFoundException extends Exception {

public CountryNotFoundException(String message) {

super(message);

}

}

**CountryService.java**  
package com.cognizant.ormlearn;

import java.util.List;

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

import org.springframework.stereotype.Service;

import jakarta.transaction.Transactional;

*@Service*

public class CountryService {

*@Autowired*

private CountryRepository countryRepository;

*@Transactional*

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

*@Transactional*

public Country findCountryByCode(String code) {

return countryRepository.findByCode(code);

}

*@Transactional*

public List<Country> findCountriesByPartialName(String partialName) {

return countryRepository.findByNameContaining(partialName);

}

*@Transactional*

public void addCountry(Country country) {

countryRepository.save(country);

}

*@Transactional*

public void updateCountry(Country country) {

countryRepository.save(country);

}

*@Transactional*

public void deleteCountry(String code) {

countryRepository.deleteById(code);

}

public void testMethod() {

System.***out***.println("CountryService method called");

}

}  
**CountryRepository.java**package com.cognizant.ormlearn;

import java.util.List;

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

import org.springframework.stereotype.Repository;

*@Repository*

public interface CountryRepository extends JpaRepository<Country, String> {

Country findByCode(String code);

List<Country> findByNameContaining(String partialName);

}  
**OrnLearnApplication.java**  
package com.cognizant.ormlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.context.ApplicationContext;

import org.springframework.boot.autoconfigure.SpringBootApplication;

*@SpringBootApplication*

public class OrmLearnApplication {

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

private static CountryService *countryService*;

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

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

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

***LOGGER***.info("Inside main");

*testGetAllCountries*();

*testFindCountryByCode*();

*testAddCountry*();

*testUpdateCountry*();

*testDeleteCountry*();

*testFindCountriesByPartialName*();

}

private static void testGetAllCountries() {

***LOGGER***.info("Start getAllCountries");

***LOGGER***.debug("Countries={}", *countryService*.getAllCountries());

***LOGGER***.info("End");

}

private static void testFindCountryByCode() throws CountryNotFoundException {

***LOGGER***.info("Start findCountryByCode");

***LOGGER***.debug("Country={}", *countryService*.findCountryByCode("IN"));

***LOGGER***.info("End");

}

private static void testAddCountry() {

***LOGGER***.info("Start addCountry");

Country newCountry = new Country();

newCountry.setCode("XY");

newCountry.setName("Testland");

*countryService*.addCountry(newCountry);

***LOGGER***.info("End");

}

private static void testUpdateCountry() throws CountryNotFoundException {

***LOGGER***.info("Start updateCountry");

Country updateCountry = new Country();

updateCountry.setCode("XY");

updateCountry.setName("Testlandia");

*countryService*.updateCountry(updateCountry);

***LOGGER***.info("End");

}

private static void testDeleteCountry() throws CountryNotFoundException {

***LOGGER***.info("Start deleteCountry");

*countryService*.deleteCountry("XY");

***LOGGER***.info("End");

}

private static void testFindCountriesByPartialName() {

***LOGGER***.info("Start findCountriesByPartialName");

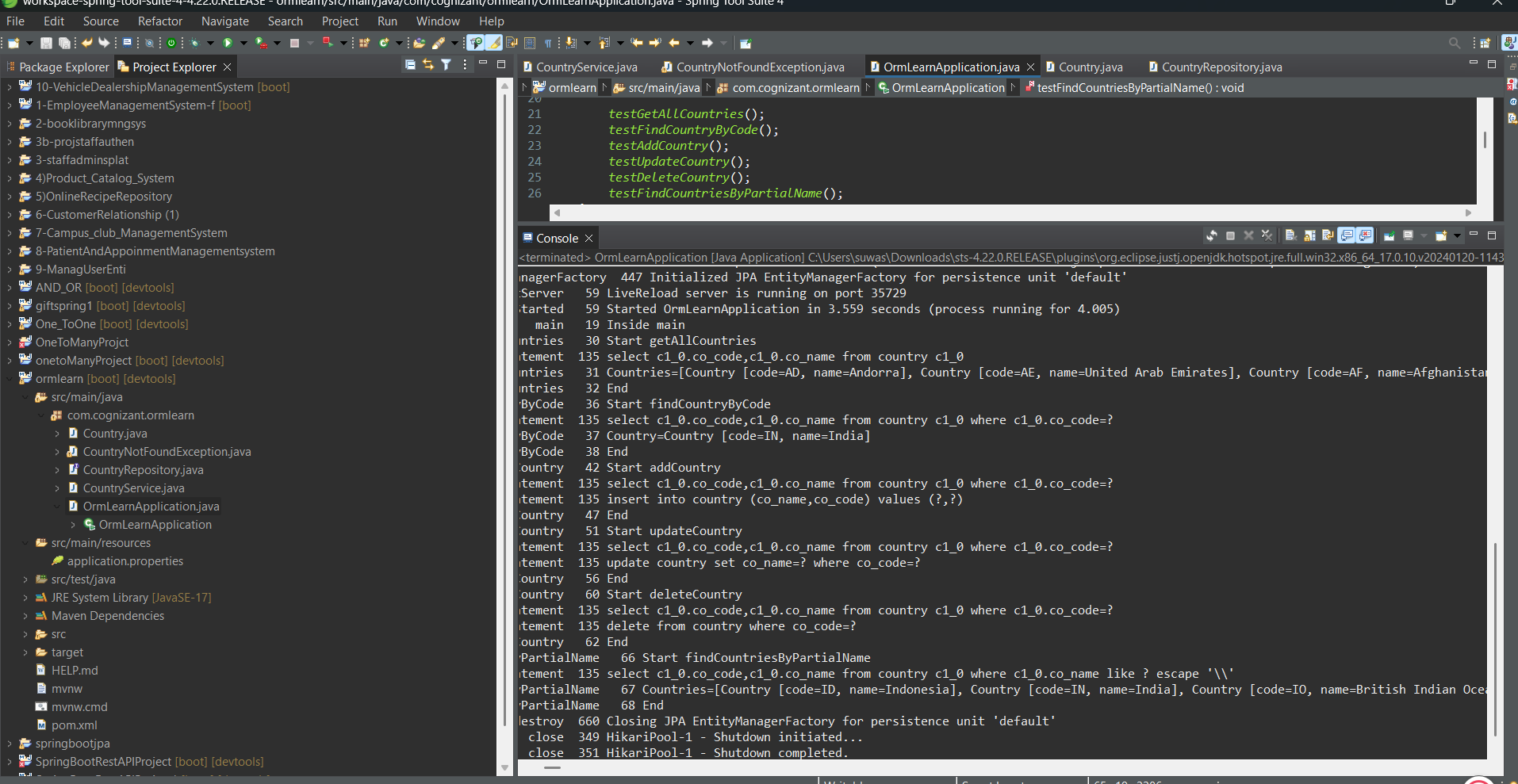
***LOGGER***.debug("Countries={}", *countryService*.findCountriesByPartialName("Ind"));

***LOGGER***.info("End");

}

}

**OUTPUT:**



**Find a country based on country code**

**Code:**

**CountryService.java**  
package com.cognizant.ormlearn;

import java.util.List;

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.ormlearn.Country;

*@Service*

public class CountryService {

*@Autowired*

private CountryRepository countryRepository;

*@Transactional*

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

*@Transactional*

public Country findCountryByCode(String countryCode) throws CountryNotFoundException {

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

if (!result.isPresent()) {

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

}

return result.get();

}

*@Transactional*

public List<Country> findCountriesByPartialName(String partialName) {

return countryRepository.findByNameContaining(partialName);

}

}

**OrmLearnApplication.java**

***testFindCountryByCode*();**

private static void testFindCountryByCode() {

***LOGGER***.info("Start testFindCountryByCode");

try {

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

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

} catch (CountryNotFoundException e) {

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

}

***LOGGER***.info("End testFindCountryByCode");

}

**CountryNotFoundException.java**  
package com.cognizant.ormlearn;

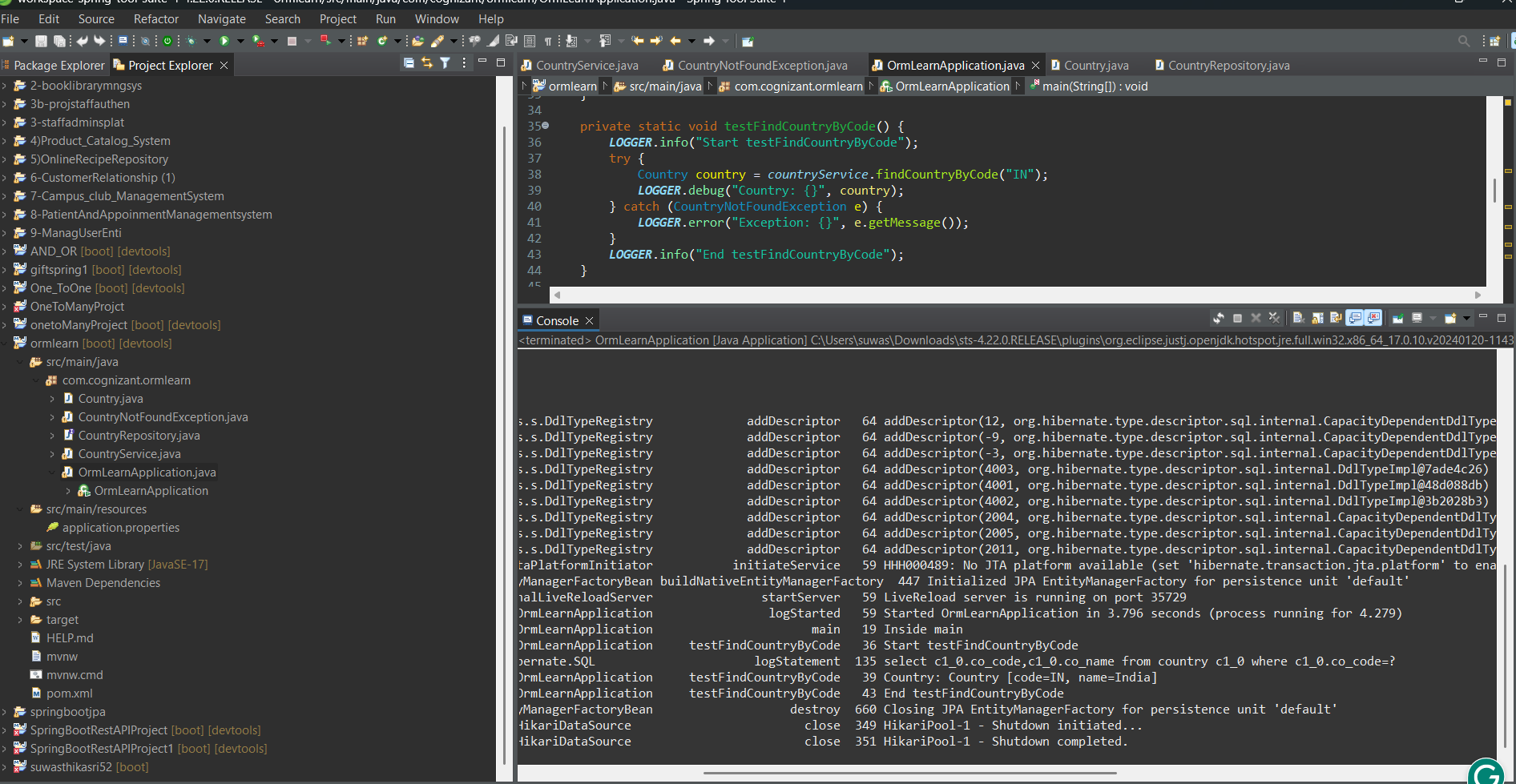
public class CountryNotFoundException extends Exception {

public CountryNotFoundException(String message) {

super(message);

}

}

OUTPUT:  


**Add a new country:**

**CountryService.java**

@Transactional

public void addCountry(Country country) {

countryRepository.save(country);

}

**OrmLearnApplication.java**

testAddCountry();

private static void testAddCountry() {

LOGGER.info("Start testAddCountry");

Country country = new Country();

country.setCode("SZ");

country.setName("SuvasthikaLand");

countryService.addCountry(country);

try {

Country retrievedCountry = countryService.findCountryByCode("SZ");

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

} catch (CountryNotFoundException e) {

LOGGER.error("Country not found: {}", e.getMessage());

}

LOGGER.info("End testAddCountry");

}

**OUTPUT:**

