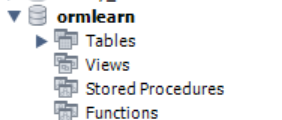
**Hands on 1**

**Spring Data JPA - Quick Example**

**ormlearn schema:**

create schema ormlearn;



**application.properties:**

# Spring Framework and application log

logging.level.org.springframework=info

logging.level.com.cognizant=debug

# Hibernate logs for displaying executed SQL, input and output

logging.level.org.hibernate.SQL=trace

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

# Log pattern

logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %-20.20thread %5p %-25.25logger**{25}** %25M %4L %m%n

# Database 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=rakesh

# Hibernate configuration

spring.jpa.hibernate.ddl-auto=validate

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

**OrmLearnApplication.java:**

package com.cognizant.orm\_learn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

*@SpringBootApplication*

public class OrmLearnApplication {

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

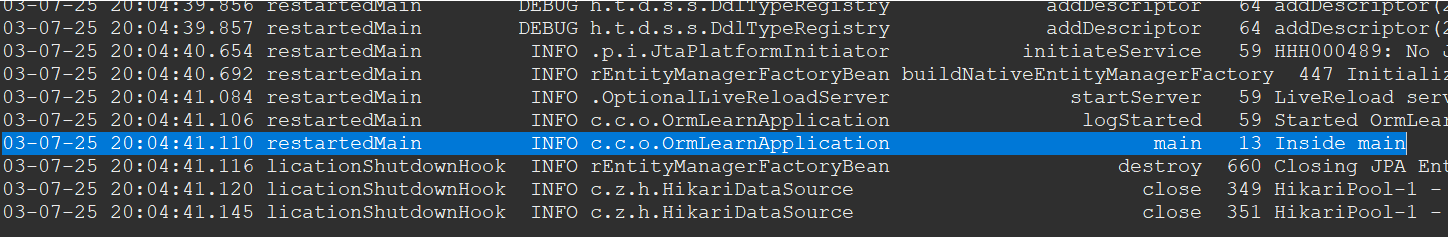
public static void main(String[] args) {

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

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

}

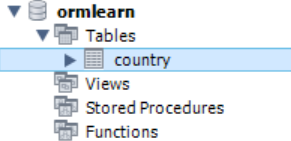
}



03-07-25 20:04:41.110 restartedMain INFO c.c.o.OrmLearnApplication main 13 Inside main

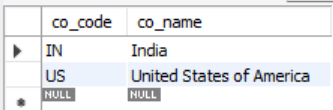
**Country Table Creation:**

create table country(co\_code varchar(2) primary key, co\_name varchar(50));



insert into country values ('IN', 'India');

insert into country values ('US', 'United States of America');



**Persistence Class - com.cognizant.orm-learn.model.Country**

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

public String getCode() {

return code;

}

public String getName() {

return name;

}

// Setters

public void setCode(String code) {

this.code = code;

}

public void setName(String name) {

this.name = name;

}

// toString

*@Override*

public String toString() {

return "Country{" +

"code='" + code + '\'' +

", name='" + name + '\'' +

'}';

}

}

**Repository Class - com.cognizant.orm-learn.CountryRepository**

package com.cognizant.orm\_learn.repository;

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> {

}

**Service Class - com.cognizant.orm-learn.service.CountryService**

package com.cognizant.orm\_learn.service;

import java.util.ArrayList;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

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

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

*@Service*

public class CountryService {

*@Autowired*

CountryRepository countryRepository;

*@Transactional*

public ArrayList<Country> getAllCountries(){

return (ArrayList<Country>) countryRepository.findAll();

}

}

**Testing in OrmLearnApplication.java**

package com.cognizant.orm\_learn;

import java.util.ArrayList;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

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

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

*@SpringBootApplication*

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);

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

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

*testGetAllCountries*();

}

private static void testGetAllCountries() {

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

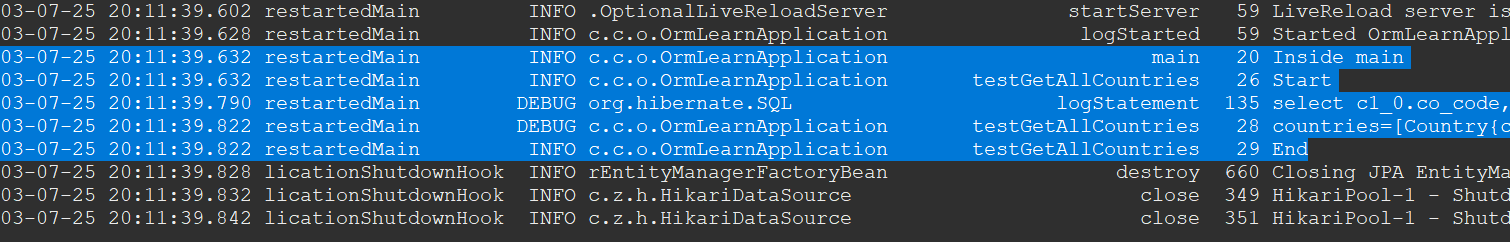
ArrayList<Country> countries = *countryService*.getAllCountries();

***LOGGER***.debug("countries={}", countries);

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

}

}



03-07-25 20:11:39.632 restartedMain INFO c.c.o.OrmLearnApplication main 20 Inside main

03-07-25 20:11:39.632 restartedMain INFO c.c.o.OrmLearnApplication testGetAllCountries 26 Start

03-07-25 20:11:39.790 restartedMain DEBUG org.hibernate.SQL logStatement 135 select c1\_0.co\_code,c1\_0.co\_name from country c1\_0

03-07-25 20:11:39.822 restartedMain DEBUG c.c.o.OrmLearnApplication testGetAllCountries 28 countries=[Country{code='IN', name='India'}, Country{code='US', name='United States of America'}]

03-07-25 20:11:39.822 restartedMain INFO c.c.o.OrmLearnApplication testGetAllCountries 29 End

**Hands on 4**

**Difference between JPA, Hibernate and Spring Data JPA**   
  
Java Persistence API (JPA)

* JSR 338 Specification for persisting, reading and managing data from Java objects
* Does not contain concrete implementation of the specification
* Hibernate is one of the implementation of JPA

Hibernate

* ORM Tool that implements JPA

Spring Data JPA

* Does not have JPA implementation, but reduces boiler plate code
* This is another level of abstraction over JPA implementation provider like Hibernate
* Manages transactions

**Refer code snippets below on how the code compares between Hibernate and Spring Data JPA  
Hibernate**

   /\* Method to CREATE an employee in the database \*/

   public Integer addEmployee(Employee employee){

      Session session = factory.openSession();

      Transaction tx = null;

      Integer employeeID = null;

      try {

         tx = session.beginTransaction();

         employeeID = (Integer) session.save(employee);

         tx.commit();

      } catch (HibernateException e) {

         if (tx != null) tx.rollback();

         e.printStackTrace();

      } finally {

         session.close();

      }

      return employeeID;

   }

**Spring Data JPA**  
EmployeeRespository.java

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

EmployeeService.java

@Autowire

  private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

  employeeRepository.save(employee);

  }

​​​​​​​

**Hands on 5**

**Implement services for managing Country**

**CountryRepository.java:**

package com.cognizant.orm\_learn.repository;

import java.util.ArrayList;

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> {

ArrayList<Country> findByCode(String code);

ArrayList<Country> findByNameContaining(String name);

void deleteByCode(String code);

}

**CountryService.java:**

package com.cognizant.orm\_learn.service;

import java.util.ArrayList;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

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

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

@Service

public class CountryService {

@Autowired

CountryRepository countryRepository;

@Transactional

public ArrayList<Country> getAllCountries(){

return (ArrayList<Country>) countryRepository.findAll();

}

@Transactional

public ArrayList<Country> findCountryByCode(String code) {

return countryRepository.findByCode(code);

}

@Transactional

public void addCountry(Country country) {

countryRepository.save(country);

}

@Transactional

public void updateCountry(String code,String name) {

if(!findCountryByCode(code).isEmpty()) {

countryRepository.save(new Country(code,name));

}

else {

System.out.println("Country with code "+code+" doesn't exist");

}

}

@Transactional

public void deleteCountry(String code) {

countryRepository.deleteByCode(code);

}

@Transactional

public ArrayList<Country> likeCountryName(String name) {

return countryRepository.findByNameContaining(name);

}

}

**OrmLearnApplication.java:**

package com.cognizant.orm\_learn;

import java.util.ArrayList;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

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

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

@SpringBootApplication

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);

LOGGER.info("Inside main");

countryService = context.getBean(CountryService.class);

testGetAllCountries();

testAllFeatures();

}

private static void testGetAllCountries() {

LOGGER.info("Start");

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

LOGGER.debug("countries={}", countries);

LOGGER.info("End");

}

private static void testAllFeatures() {

ArrayList<Country> country;

// Find Country by Country Code

country=countryService.findCountryByCode("IN");

System.out.println("Find Country by Country Code IN\n"+country);

// Add New Country

countryService.addCountry(new Country("ZZ","New Country"));

// Checking if New Country Added

country=countryService.findCountryByCode("ZZ");

System.out.println("Checking if Country "+new Country("ZZ","New Country")+" added: "+!country.isEmpty());

// Update Country

countryService.updateCountry("ZZ","Updated New Country");

// Checking if New Country Updated

country=countryService.findCountryByCode("ZZ");

System.out.println("Checking if updated "+country);

// Delete Country

countryService.deleteCountry("ZZ");

// Checking if deleted

if(countryService.findCountryByCode("ZZ").isEmpty()) {

System.out.println("Country Deleted");

}

else {

System.out.println("Country Not Deleted");

}

// Find list of countries matching a partial country name

country=countryService.likeCountryName("inc");

System.out.println("Countries matching inc");

for(Country c:country) {

System.out.print(c.getName()+", ");

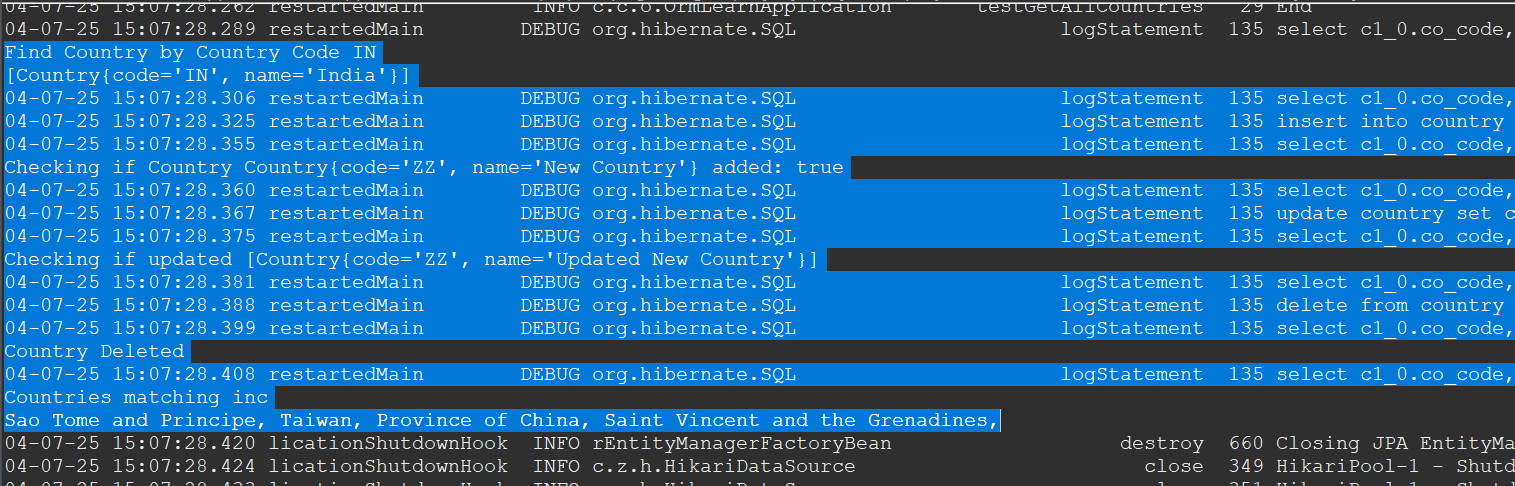
}

System.out.println();

}

}

**Output:**

****

Find Country by Country Code IN

[Country{code='IN', name='India'}]

04-07-25 15:07:28.306 restartedMain DEBUG org.hibernate.SQL logStatement 135 select c1\_0.co\_code,c1\_0.co\_name from country c1\_0 where c1\_0.co\_code=?

04-07-25 15:07:28.325 restartedMain DEBUG org.hibernate.SQL logStatement 135 insert into country (co\_name,co\_code) values (?,?)

04-07-25 15:07:28.355 restartedMain DEBUG org.hibernate.SQL logStatement 135 select c1\_0.co\_code,c1\_0.co\_name from country c1\_0 where c1\_0.co\_code=?

Checking if Country Country{code='ZZ', name='New Country'} added: true

04-07-25 15:07:28.360 restartedMain DEBUG org.hibernate.SQL logStatement 135 select c1\_0.co\_code,c1\_0.co\_name from country c1\_0 where c1\_0.co\_code=?

04-07-25 15:07:28.367 restartedMain DEBUG org.hibernate.SQL logStatement 135 update country set co\_name=? where co\_code=?

04-07-25 15:07:28.375 restartedMain DEBUG org.hibernate.SQL logStatement 135 select c1\_0.co\_code,c1\_0.co\_name from country c1\_0 where c1\_0.co\_code=?

Checking if updated [Country{code='ZZ', name='Updated New Country'}]

04-07-25 15:07:28.381 restartedMain DEBUG org.hibernate.SQL logStatement 135 select c1\_0.co\_code,c1\_0.co\_name from country c1\_0 where c1\_0.co\_code=?

04-07-25 15:07:28.388 restartedMain DEBUG org.hibernate.SQL logStatement 135 delete from country where co\_code=?

04-07-25 15:07:28.399 restartedMain DEBUG org.hibernate.SQL logStatement 135 select c1\_0.co\_code,c1\_0.co\_name from country c1\_0 where c1\_0.co\_code=?

Country Deleted

04-07-25 15:07:28.408 restartedMain DEBUG org.hibernate.SQL logStatement 135 select c1\_0.co\_code,c1\_0.co\_name from country c1\_0 where c1\_0.co\_name like ? escape '\\'

Countries matching inc

Sao Tome and Principe, Taiwan, Province of China, Saint Vincent and the Grenadines,

**Hands on 6**

**Find a country based on country code**

**CountryNotFoundException.java:**

package com.cognizant.orm\_learn.service.exception;

public class CountryNotFoundException extends Exception{

public CountryNotFoundException(String message) {

super(message);

}

}

**findCountryByCode():**

*@Transactional*

public Country findCountryByCode(String code) throws CountryNotFoundException{

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

if(country.isEmpty()) {

throw new CountryNotFoundException("Invalid Country Code Supplied");

}

return country.get(0);

}

**getAllCountriesTest():**

private static void getAllCountriesTest() throws CountryNotFoundException{

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

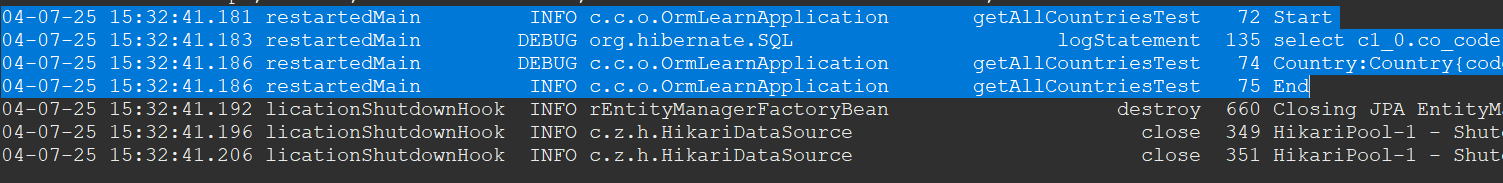
Country country = *countryService*.findCountryByCode("IN");

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

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

}

**Output:**



04-07-25 15:32:41.181 restartedMain INFO c.c.o.OrmLearnApplication getAllCountriesTest 72 Start

04-07-25 15:32:41.183 restartedMain DEBUG org.hibernate.SQL logStatement 135 select c1\_0.co\_code,c1\_0.co\_name from country c1\_0 where c1\_0.co\_code=?

04-07-25 15:32:41.186 restartedMain DEBUG c.c.o.OrmLearnApplication getAllCountriesTest 74 Country:Country{code='IN', name='India'}

04-07-25 15:32:41.186 restartedMain INFO c.c.o.OrmLearnApplication getAllCountriesTest 75 End

**Hands on 7**

**Add a new country**

**addCountry():**

*@Transactional*

public void addCountry(Country country) {

countryRepository.save(country);

}

**testAddCountry():**

private static void testAddCountry() throws CountryNotFoundException{

// Add New Country

*countryService*.addCountry(new Country("ZZ","New Country"));

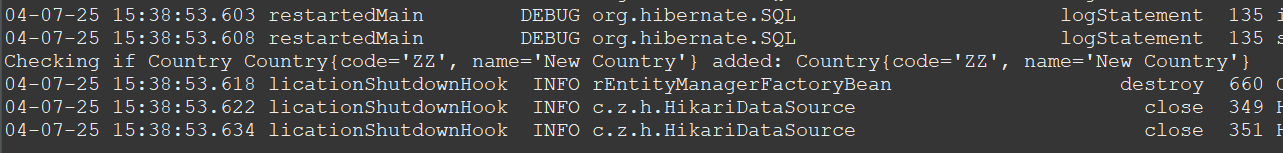
// Checking if New Country Added

Country c=*countryService*.findCountryByCode("ZZ");

System.***out***.println("Checking if Country "+new Country("ZZ","New Country")+" added: "+c);

}

**Output:**



**Hands on 8**

**Update a country based on code**

**updateCountry():**

*@Transactional*

public void updateCountry(String code,String name) throws CountryNotFoundException{

if(findCountryByCode(code)!=null) {

countryRepository.save(new Country(code,name));

}

else {

System.***out***.println("Country with code "+code+" doesn't exist");

}

}

**testUpdateCountry():**

private static void testUpdateCountry() throws CountryNotFoundException{

// Update Country

*countryService*.updateCountry("ZZ","Updated New Country");

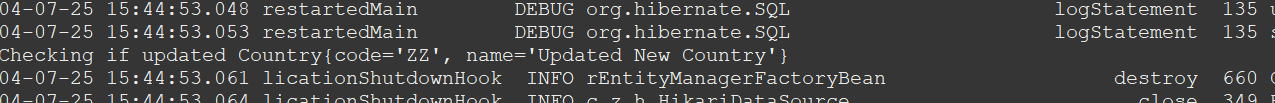
// Checking if New Country Updated

Country c=*countryService*.findCountryByCode("ZZ");

System.*out*.println("Checking if updated "+c);

}

**Output:**



**Hands on 9**

**Delete a country based on code**

**deleteCountry():**

*@Transactional*

public void deleteCountry(String code) {

countryRepository.deleteById(code);

}

**testDeleteCountry():**

private static void testDeleteCountry() throws CountryNotFoundException{

// Delete Country

*countryService*.deleteCountry("ZZ");

// Checking if deleted

try{

*countryService*.findCountryByCode("ZZ");

}

catch(CountryNotFoundException e) {

System.***out***.println("Country Deleted");

}

}

**Output:**

