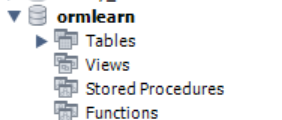
**Hands on 1**

**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);

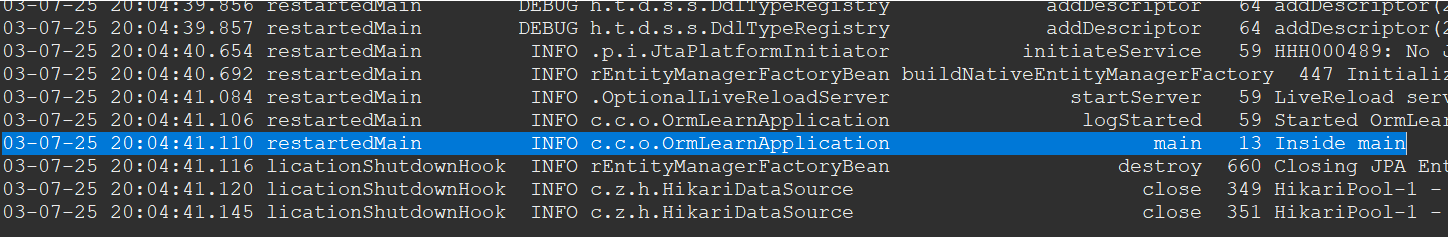
public static void main(String[] args) {

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

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

}

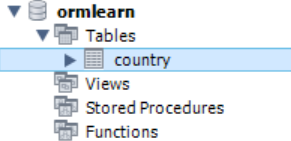
}



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

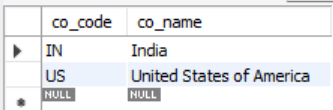
**Country Table Creation:**

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");

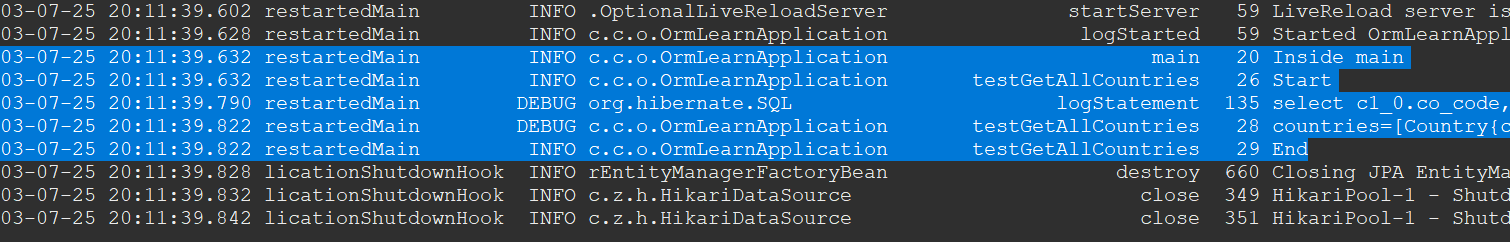
ArrayList<Country> countries = *countryService*.getAllCountries();

***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);

  }