**Hands-on 1: Spring Data JPA - Quick Example**

**Software Pre-requisites**

* MySQL Server 8.0
* MySQL Workbench 8
* Eclipse IDE for Enterprise Java Developers 2019-03 R
* Maven 3.6.2

**Create a Spring Boot Project using Spring Initializr**

1. Go to: <https://start.spring.io>
2. Set the following:
   * **Group**: com.cognizant
   * **Artifact**: orm-learn
   * **Description**: Demo project for Spring Data JPA and Hibernate
3. Choose the following dependencies:
   * Spring Boot DevTools
   * Spring Data JPA
   * MySQL Driver
4. Click **Generate**, download the ZIP file.
5. Extract the project to Eclipse Workspace.
6. Import project in Eclipse:
   * File > Import > Maven > Existing Maven Projects > Browse to extracted folder > Finish
7. Open MySQL client and create a new schema:

mysql -u root -p

mysql> create schema ormlearn;

**Update application.properties (src/main/resources)**

# 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=root

# Hibernate configuration

spring.jpa.hibernate.ddl-auto=validate

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

**Build the Project**

Use the following command to build with proxy settings:

mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456

**Add Logs in OrmLearnApplication.java**

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

LOGGER.info("Inside main");

}

**Project Structure Walkthrough**

1. src/main/java: Contains main application source code
2. src/main/resources: Contains configuration files (like application.properties)
3. src/test/java: Contains test classes
4. OrmLearnApplication.java: Main method to launch the Spring Boot app
5. @SpringBootApplication: Indicates a Spring Boot app with auto-configuration
6. pom.xml: Maven build configuration file
   * View Dependency Hierarchy tab in Eclipse to explore dependencies

**Create Country Table in MySQL**

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');

**Create Persistence Class - Country.java**

**Location**: com.cognizant.ormlearn.model

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;

// Getters and Setters

@Override

public String toString() {

return "Country [code=" + code + ", name=" + name + "]";

}

}

**Create Repository Interface - CountryRepository.java**

**Location**: com.cognizant.ormlearn.repository

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> {

}

**Create Service Class - CountryService.java**

**Location**: com.cognizant.ormlearn.service

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();

}

}

**Update OrmLearnApplication.java to Test Service**

Import org.springframework.context.ApplicationContext;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import com.cognizant.ormlearn.service.CountryService;

import com.cognizant.ormlearn.model.Country;

import java.util.List;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@SpringBootApplication

public class OrmLearnApplication {

private static CountryService countryService;

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

public static void main(String[] args) {

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

countryService = context.getBean(CountryService.class);

testGetAllCountries();

}

private static void testGetAllCountries() {

LOGGER.info("Start");

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

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

LOGGER.info("End");

}

}

**Run the Application**

* Run OrmLearnApplication.java
* Check the log output to verify:
  + Inside main is logged
  + Countries from MySQL ormlearn.country table are fetched and printed