**Spring Data JPA - Quick Start Guide**

**Overview**

This hands-on tutorial demonstrates how to create a Spring Boot application with Spring Data JPA and Hibernate for database operations using MySQL.

**Software Prerequisites**

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

**Project Setup**

**1. Create Eclipse Project using Spring Initializr**

1. Navigate to https://start.spring.io/
2. Configure project settings:
   * **Group**: com.cognizant
   * **Artifact Id**: orm-learn
   * **Description**: "Demo project for Spring Data JPA and Hibernate"
3. Select dependencies:
   * Spring Boot DevTools
   * Spring Data JPA
   * MySQL Driver
4. Click **Generate** and download the project as zip
5. Extract the zip to your Eclipse Workspace
6. Import the project in Eclipse:
   * File → Import → Maven → Existing Maven Projects
   * Browse and select extracted folder → Finish

**2. Database Schema Setup**

Create a new schema in MySQL database:

-- Connect to MySQL

mysql -u root -p

-- Create schema

mysql> create schema ormlearn;

**3. Application Configuration**

Open src/main/resources/application.properties and add the following configuration:

# 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

**4. Build the Project**

Execute the following Maven command:

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

**5. Add Logging to Main Method**

Update 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 Overview**

* **src/main/java**: Application source code
* **src/main/resources**: Application configuration files
* **src/test/java**: Test code
* **OrmLearnApplication.java**: Main application class with @SpringBootApplication
* **pom.xml**: Maven configuration with dependency management

**Database Table Creation**

Create the country table with sample data:

-- Create table

CREATE TABLE country(

co\_code VARCHAR(2) PRIMARY KEY,

co\_name VARCHAR(50)

);

-- Insert sample records

INSERT INTO country VALUES ('IN', 'India');

INSERT INTO country VALUES ('US', 'United States of America');

**Implementation**

**1. Entity Class - Country.java**

Create package: 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="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 and Setters

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;

}

// toString method

@Override

public String toString() {

return "Country{" +

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

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

'}';

}

}

**Key Annotations:**

* @Entity: Marks the class as a JPA entity
* @Table: Maps the entity to a database table
* @Id: Defines the primary key
* @Column: Maps fields to table columns

**2. Repository Interface - CountryRepository.java**

Create package: 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> {

// JpaRepository provides basic CRUD operations

// Additional custom methods can be defined here

}

**3. Service Class - CountryService.java**

Create package: com.cognizant.ormlearn.service

package com.cognizant.ormlearn.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

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();

}

}

**4. Testing in Main Application**

Update OrmLearnApplication.java:

package com.cognizant.ormlearn;

import java.util.List;

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.ormlearn.model.Country;

import com.cognizant.ormlearn.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);

countryService = context.getBean(CountryService.class);

LOGGER.info("Inside main");

testGetAllCountries();

}

private static void testGetAllCountries() {

LOGGER.info("Start");

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

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

LOGGER.info("End");

}

}

**Running the Application**

1. Ensure MySQL server is running
2. Verify that the ormlearn schema exists and contains the country table with sample data
3. Run the OrmLearnApplication class
4. Check the console logs to verify:
   * Database connection is established
   * SQL queries are executed (due to Hibernate logging)
   * Country data is retrieved and displayed

**Output**

