**Hands on 1**

**Spring Data JPA - Quick Example**   
  
**Software Pre-requisites**

* MySQL Server 8.0
* MySQL Workbench 8
* Intellij JavaEE Edition
* Maven 3.6.2

**Create a Intellij Project using Spring Initializr**

* Go to <https://start.spring.io/>
* Change Group as “com.cognizant”
* Change Artifact Id as “orm-learn”
* In Options > Description enter "Demo project for Spring Data JPA and Hibernate"
* Click on menu and select "Spring Boot DevTools", "Spring Data JPA" and "MySQL Driver"
* Click Generate and download the project as zip
* Extract the zip in root folder to Intellij Workspace
* Import the project in Intellij "File > Import > Maven > Existing Maven Projects > Click Browse and select extracted folder > Finish"
* Create a new schema "ormlearn" in MySQL database. Execute the following commands to open MySQL client and create schema.

> mysql -u root -p

mysql> create schema ormlearn;

* In orm-learn Intellij project, open src/main/resources/application.properties and include the below database

# 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=password

# Hibernate configuration

spring.jpa.hibernate.ddl-auto=validate

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

SME to walk through the following aspects related to the project created:

1. src/main/java - Folder with application code
2. src/main/resources - Folder for application configuration
3. OrmLearnApplication.java - Walkthrough the main() method.
4. Purpose of @SpringBootApplication annotation
5. pom.xml
   1. Walkthrough all the configuration defined in XML file
   2. Open 'Dependency Hierarchy' and show the dependency tree.

**Country table creation**

* Create a new table country with columns for code and name. For sample, let us insert one country with values 'IN' and 'India' in this table.

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

* Insert couple of records into the table

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

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

insert into country values ('PK', 'Pakistan');

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

* Open Intellij with orm-learn project
* Create new package com.cognizant.orm-learn.model
* Create Country.java, then generate getters, setters and toString() methods.
* Include @Entity and @Table at class level
* Include @Column annotations in each getter method specifying the column name.
* package com.cognizant.orm\_learn.model;  
    
  import jakarta.persistence.Entity;  
  import jakarta.persistence.Column;  
  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;  
    
   // 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()  
    
   @Override  
   public String toString() {  
   return "Country{" +  
   "code='" + code + '\'' +  
   ", name='" + name + '\'' +  
   '}';  
   }  
  }

*Notes:*

* @Entity is an indicator to Spring Data JPA that it is an entity class for the application
* @Table helps in defining the mapping database table
* @Id helps is defining the primary key
* @Column helps in defining the mapping table column

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

* Create new package com.cognizant.orm-learn.repository
* Create new interface named CountryRepository that extends JpaRepository<Country, String>
* Define @Repository annotation at class level
* 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.countroller.CountryController**

* Create new package com.cognizant.orm-learn.controller
* Create new class CountryController
* Include new method getAllCountries() method that returns a list of countries.
* In getAllCountries() method invoke countryRepository.findAll() method and return the result
* package com.cognizant.orm\_learn.controller;  
    
  import com.cognizant.orm\_learn.model.Country;  
  import com.cognizant.orm\_learn.repository.CountryRepository;  
  import org.springframework.beans.factory.annotation.Autowired;  
  import org.springframework.web.bind.annotation.GetMapping;  
  import org.springframework.web.bind.annotation.RestController;  
    
  import java.util.List;  
    
  @RestController  
  public class CountryController {  
   @Autowired  
   private CountryRepository countryRepository;  
    
   @GetMapping("/")  
   public List<Country> getAllCountries(){  
   return countryRepository.findAll();  
   }  
  }

Output : Visit localhost:8080 home page

