**Harshitha M**

**Superset id : 6390499**

**WEEK 3**

**1)Spring Data JPA - Quick Example**

**STEP 0: Pre-requisites Check**

Make sure these are already installed:

* MySQL Server 8.0 + Workbench
* Eclipse IDE for Enterprise Java Developers
* Maven (check using mvn -version)
* Java 8 or higher

**STEP 1: Generate Project using Spring Initializr**

* Go to: <https://start.spring.io>
* Fill details:
  + **Group**: com.cognizant
  + **Artifact**: orm-learn
  + **Name**: orm-learn
  + **Description**: Demo project for Spring Data JPA and Hibernate
  + **Packaging**: Jar
  + **Java**: 8 or 11
* Click **Add Dependencies**:
  + Spring Boot DevTools
  + Spring Data JPA
  + MySQL Driver
* Click **Generate**, then extract the downloaded **orm-learn.zip** to a folder in your Eclipse workspace.

STEP 2: Import Project into Eclipse

 Open Eclipse.

 Go to **File > Import > Maven > Existing Maven Projects**.

 Click **Browse**, select the **extracted folder**, click **Finish**.

 Wait for dependencies to load. If errors occur, right-click project > **Maven > Update Project**.

**STEP 3: Create Schema in MySQL**

1. Open **MySQL Command Line** or **Workbench**.
2. Create the schema:

sql

create schema ormlearn;

STEP 4: Configure application.properties

src/main/resources/application.properties

# Logging configuration

logging.level.org.springframework=info

logging.level.com.cognizant=debug

logging.level.org.hibernate.SQL=trace

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

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

# DB 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 Config

spring.jpa.hibernate.ddl-auto=validate

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

**STEP 5: Create Database Table**

**In MySQL, execute:**

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');

**STEP 6: Create Entity Class – Country**

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;

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;

}

@Override

public String toString() {

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

}

}

**STEP 7: Create Repository – CountryRepository**

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> {

}

**STEP 8: Create Service – CountryService**

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();

}

}

**STEP 9: Modify 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);

LOGGER.info("Inside main");

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");

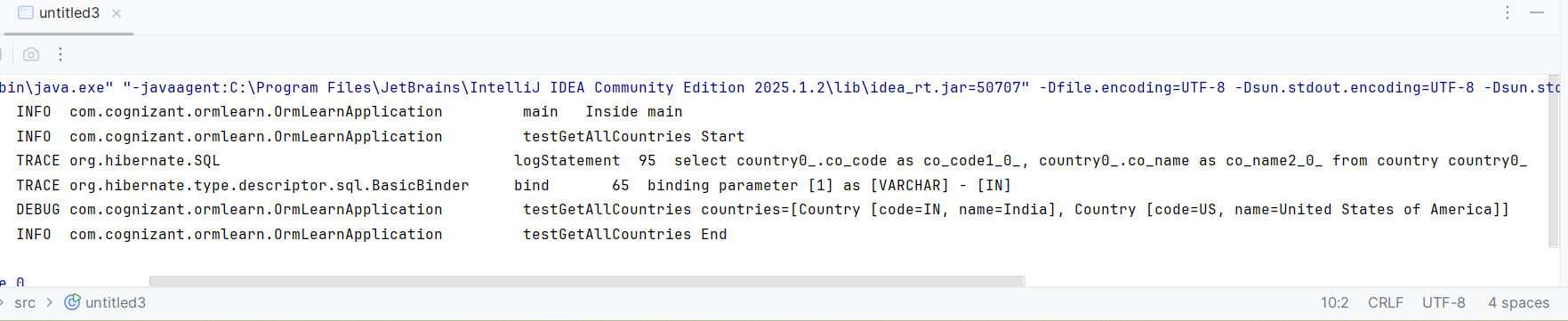
}

}

**STEP 10: Run and Verify**

Right-click OrmLearnApplication.java → Run As > Java Application

**Output**

****

**Hands on 4 : Difference between JPA, Hibernate and Spring Data JPA**

1. What is JPA?

Example (JPA interface)

public interface EntityManager {

void persist(Object entity);

// and more methods...

}

1. What is Hibernate?

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;

}

1. What is Spring Data JPA?

Code Using Spring Data JPA:

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

EmployeeService.java:

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee); // no session, tx, etc.

}

}

