**SPRING DATA JPA**

**EXERCISE – 1 : SPRING DATA JPA – QUICK EXAMPLE**

**MySql - Database and table creation:**

Create a database named “ormlearn” and create a table named “countries” and insert the data into the table.

CREATE DATABASE ormlearn;

use ormlearn;

CREATE TABLE country(

co\_code VARCHAR(2) PRIMARY KEY,

co\_name VARCHAR(50)

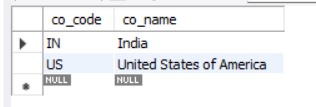
);

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

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

select \* from country;

**Table after inserting the data looks like:**



**Application.properties:**

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

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

spring.jpa.hibernate.ddl-auto=validate

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

**Country.java**

package com.cognizant.orm\_learn.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

import jakarta.persistence.Column;

@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+"]";

}

}

**CountryService.java**

package com.cognizant.orm\_learn.service;

import java.util.List;

import jakarta.transaction.Transactional;

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

import org.springframework.stereotype.Service;

import com.cognizant.orm\_learn.model.Country;

import com.cognizant.orm\_learn.repository.CountryRepository;

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

@Transactional

public List<Country> getAllCountries(){

return countryRepository.findAll();

}

}

**CountryRepository.java**

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>{

}

**OrmLearnApplication.java:**

package com.cognizant.orm\_learn;

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.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);

countryService=context.getBean(CountryService.class);

testGetAllCountries();

}

private static void testGetAllCountries() {

LOGGER.info("Start");

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

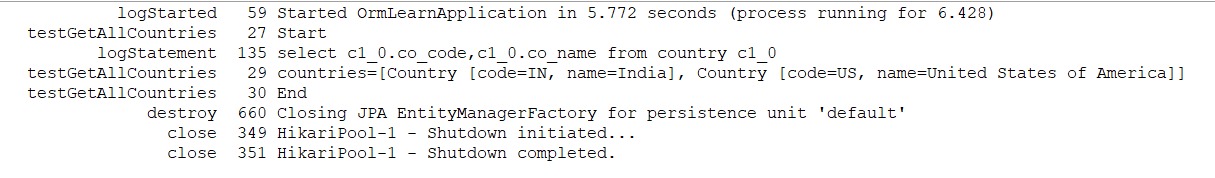
LOGGER.debug("countries={}", countries);

LOGGER.info("End");

}

}

**OUTPUT:**



**EXERCISE – 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

**MySql - Database and table creation:**

CREATE DATABASE Employee;

USE Employee;

CREATE TABLE emp\_table (

id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100),

designation VARCHAR(100),

salary DOUBLE,

phone varchar(10)

);

select \* from emp\_table;

**USING SPRING DATA JPA:**

**Application.properties**

spring.datasource.url=jdbc:mysql://localhost:3306/Employee

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

**EmployeeService.java**

package com.company.hrms.service;

import com.company.hrms.repository.EmployeeRepository;

import com.company.hrms.model.Employee;

import org.springframework.stereotype.Service;

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

import jakarta.transaction.Transactional;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository repo;

@Transactional

public void addEmployee(Employee emp) {

repo.save(emp);

}

}

**EmployeeRepository.java**

package com.company.hrms.repository;

import com.company.hrms.model.Employee;

import org.springframework.data.jpa.repository.JpaRepository;

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**Employee.java**

package com.company.hrms.model;

import jakarta.persistence.\*;

@Entity

@Table(name = "emp\_table")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private String designation;

private double salary;

private String phone;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getDesignation() {

return designation;

}

public void setDesignation(String designation) {

this.designation = designation;

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

this.salary = salary;

}

public String getPhone() {

return phone;

}

public void setPhone(String phone) {

this.phone = phone;

}

}

**JApplication.java**

package com.company.hrms;

import com.company.hrms.model.Employee;

import com.company.hrms.service.EmployeeService;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class JApplication implements CommandLineRunner {

@Autowired

private EmployeeService employeeService;

public static void main(String[] args) {

SpringApplication.run(JApplication.class, args);

}

@Override

public void run(String... args) {

Employee emp = new Employee();

emp.setName("Michael Johnson");

emp.setDesignation("Business Analyst");

emp.setSalary(68000);

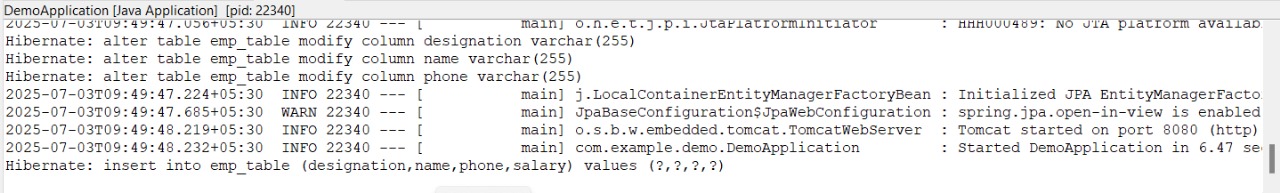
emp.setPhone("7654321098");

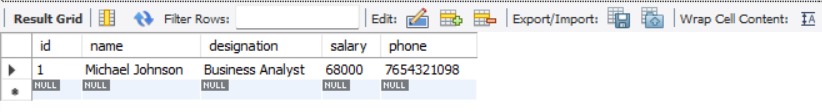
employeeService.addEmployee(emp);

}

}

**OUTPUT:**





**USING HIBERNATE:**

**hibernate.cfg.xml:**

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name="hibernate.connection.driver\_class">com.mysql.cj.jdbc.Driver</property>

<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/Employee</property>

<property name="hibernate.connection.username">root</property>

<property name="hibernate.connection.password">root</property>

<property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

<property name="hibernate.hbm2ddl.auto">update</property>

<property name="show\_sql">true</property>

<mapping class="com.example.Employee"/>

</session-factory>

</hibernate-configuration>

**Employee.java**

package com.example;

import jakarta.persistence.\*;

@Entity

@Table(name = "emp\_table")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private String designation;

private double salary;

private String phone;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getDesignation() {

return designation;

}

public void setDesignation(String designation) {

this.designation = designation;

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

this.salary = salary;

}

public String getPhone() {

return phone;

}

public void setPhone(String phone) {

this.phone = phone;

}

}

**HibernateMain.java:**

package com.example;

import org.hibernate.\*;

import org.hibernate.cfg.Configuration;

public class HibernateMain {

public static void main(String[] args) {

SessionFactory factory = new Configuration().configure().buildSessionFactory();

Session session = factory.openSession();

Transaction tx = null;

try {

tx = session.beginTransaction();

Employee emp = new Employee();

emp.setName("Emily Davis");

emp.setDesignation("HR Managert");

emp.setSalary(72000);

emp.setPhone("6543210987");

session.persist(emp);

tx.commit();

} catch (HibernateException e) {

if (tx != null) tx.rollback();

e.printStackTrace();

} finally {

session.close();

}

factory.close();

}

}

**OUTPUT:**

