**Hands on 1**

Mandatory Hands-on

Sparshak Ghosh

**Spring Data JPA**

**• Spring Data JPA - Quick Example**

**ANSWER**

**Dependencies**

Spring boot dev tools, Spring data JPA, MySql driver

**Main.java**

package com.cognizant.orm\_learn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class OrmLearnApplication {

private static final Logger LOGGER =

LoggerFactory.getLogger(OrmLearnApplication.class);

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

LOGGER.info("Inside main");

}

}

**Country table**

mysql> use sparshakDB;

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

mysql> insert into country values -> ('IN', 'India'), -> ('US', 'United States of America');

**Country.java**

package com.cognizant.orm\_learn.model;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

Mandatory Hands-on

Sparshak Ghosh

@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 + "]";

}

}

**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

Mandatory Hands-on

Sparshak Ghosh

public interface CountryRepository extends JpaRepository<Country, String>{

}

CountryService.java

package com.cognizant.orm\_learn.service;

import java.util.List;

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;

import jakarta.transaction.Transactional;

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

@Transactional

public List<Country> getAllCountries(){

return countryRepository.findAll();

}

}

**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;

Sparshak Ghosh

Mandatory Hands-on

@SpringBootApplication

public class OrmLearnApplication {

private static final Logger LOGGER =

LoggerFactory.getLogger(OrmLearnApplication.class);

private static CountryService countryService;

private static void testGetAllCountries() {

LOGGER.info("Start");

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

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

LOGGER.info("End");

}

public static void main(String[] args) {

ApplicationContext context = SpringApplication.run(OrmLearnApplication.class,

args);

countryService = context.getBean(CountryService.class);

testGetAllCountries();

}

}

**Hands on 4**

**• Difference between JPA, Hibernate and Spring Data JPA**

**i>Hibernate implementation:**

hibernate.config.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE hibernate-configuration PUBLIC

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

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

Sparshak Ghosh

Mandatory Hands-on

<hibernate-configuration>

<session-factory>

<property

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

<property name="hibernate.connection.password">\*\*\*\*\*\*\*\*</property>

<property

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

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

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

<property name="current\_session\_context\_class">thread</property>

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

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

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

<mapping class="com.shom.employee\_management.model.Employee" />

</session-factory>

</hibernate-configuration>

**Employee.java**

package com.shom.employee\_management.model;

import javax.persistence.Entity;

import javax.persistence.Id;

@Entity

public class Employee {

@Id

private Integer id;

private String name;

private String dept;

public Integer getId() {

return id;

}

public void setId(Integer id) {

Sparshak Ghosh

Mandatory Hands-on

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getDept() {

return dept;

}

public void setDept(String dept) {

this.dept = dept;

}

}

**EmployeeServiceHibernate.java**

package com.shom.employee\_management.service;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import org.hibernate.cfg.Configuration;

import com.shom.employee\_management.model.Employee;

public class EmployeeServiceHibernate {

private SessionFactory factory;

public EmployeeServiceHibernate() {

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

}

public void addEmployee(Employee employee) {

Mandatory Hands-on

Sparshak Ghosh

Session session = factory.openSession();

Transaction tx = null;

try {

tx = session.beginTransaction();

session.save(employee);

tx.commit();

} catch (Exception e) {

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

e.printStackTrace();

} finally {

session.close();

};

}

}

**EmployeeManagementApplication.java**

package com.shom.employee\_management;

import com.shom.employee\_management.model.Employee;

import com.shom.employee\_management.service.EmployeeServiceHibernate;

public class EmployeeManagementApplication2 {

public static void main(String[] args) {

EmployeeServiceHibernate employeeServiceHibernate = new

EmployeeServiceHibernate();

Employee employee = new Employee();

employee.setId(1);

employee.setName("Shom");

employee.setDept("Dev");

employeeServiceHibernate.addEmployee(employee);

System.out.println("Employee created");

Sparshak Ghosh

Mandatory Hands-on

}

}

**ii> Spring data jpa implementation:**

application.properties

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=password

spring.h2.console.enabled=true

spring.jpa.hibernate.ddl-auto=create-drop

spring.jpa.show-sql=true

**Employee.java**

package com.shom.employee\_management.model;

import javax.persistence.Entity;

import javax.persistence.Id;

@Entity

public class Employee {

@Id

private Integer id;

private String name;

private String dept;

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getName() {

return name;

Sparshak Ghosh

Mandatory Hands-on

}

public void setName(String name) {

this.name = name;

}

public String getDept() {

return dept;

}

public void setDept(String dept) {

this.dept = dept;

}

}

**EmployeeRepository.java**

package com.shom.employee\_management.repository;

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

import com.shom.employee\_management.model.Employee;

public interface EmployeeRepository extends JpaRepository<Employee, Integer>{

}

**EmployeeService.java**

package com.shom.employee\_management.service;

import javax.transaction.Transactional;

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

import org.springframework.stereotype.Service;

import com.shom.employee\_management.model.Employee;

import com.shom.employee\_management.repository.EmployeeRepository;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmp(Employee emp) {

employeeRepository.save(emp);

Sparshak Ghosh

Mandatory Hands-on

}

}

EmployeeController.java

package com.shom.employee\_management.controller;

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

import org.springframework.web.bind.annotation.\*;

import com.shom.employee\_management.model.Employee;

import com.shom.employee\_management.service.EmployeeService;

@RestController

@RequestMapping("/employees")

public class EmployeeController {

@Autowired

private EmployeeService employeeService;

@PostMapping

public String addEmployee(@RequestBody Employee employee) {

employeeService.addEmp(employee);

return "Employee added successfully!";

}

}

**EmployeeManagement.java**

package com.shom.employee\_management;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication(scanBasePackages = "com.shom.employee\_management")

public class EmployeeManagementApplication {

public static void main(String[] args) {

SpringApplication.run(EmployeeManagementApplication.class, args);

}

}