**Hands on 5 : Implement one to many relationship between Employee and Department**

**OrmLearnApplication.java :-**

package com.cognizant.orm\_learn;

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

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

import com.cognizant.orm\_learn.service.DepartmentService;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import jakarta.annotation.PostConstruct;

*@SpringBootApplication*

public class OrmLearnApplication {

private static final Logger ***LOGGER*** = LoggerFactory.*getLogger*(OrmLearnApplication.class);

*@Autowired*

private DepartmentService departmentService;

public static void main(String[] args) {

SpringApplication.*run*(OrmLearnApplication.class, args);

}

*@PostConstruct*

public void run() {

testGetDepartment();

}

private void testGetDepartment() {

***LOGGER***.info("Start - testGetDepartment");

Department dept = departmentService.get(1); // Ensure department with ID 1 exists and has employees

***LOGGER***.debug("Department: {}", dept);

***LOGGER***.debug("Employees in Department:");

for (Employee emp : dept.getEmployeeList()) {

***LOGGER***.debug("Employee: {}", emp);

}

***LOGGER***.info("End - testGetDepartment");

}

}

**DepartmentRepository.java :-**

package com.cognizant.orm\_learn.service;

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

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

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

import org.springframework.stereotype.Service;

*@Service*

public class DepartmentService {

*@Autowired*

private DepartmentRepository departmentRepository;

public Department get(int id) {

return departmentRepository.findById(id).orElse(null);

}

}

**EmployeeRepository.java :-**

package com.cognizant.orm\_learn.repository;

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

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

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**DepartmentService.java :-**

package com.cognizant.orm\_learn.service;

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

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

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

import org.springframework.stereotype.Service;

@Service

public class DepartmentService {

@Autowired

private DepartmentRepository departmentRepository;

public Department get(int id) {

return departmentRepository.findById(id).orElse(null);

}

}

**Department.java :-**

package com.cognizant.orm\_learn.model;

import jakarta.persistence.\*;

import java.util.Set;

@Entity

@Table(name = "department")

public class Department {

@Id

@Column(name = "dp\_id")

private int id;

@Column(name = "dp\_name")

private String name;

@OneToMany(mappedBy = "department", fetch = FetchType.EAGER) // EAGER fetch to avoid LazyInitializationException

private Set<Employee> employeeList;

public Department() {}

public Department(int id, String name) {

this.id = id;

this.name = name;

}

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 Set<Employee> getEmployeeList() { return employeeList; }

public void setEmployeeList(Set<Employee> employeeList) { this.employeeList = employeeList; }

@Override

public String toString() {

return "Department [id=" + id + ", name=" + name + "]";

}

}

package com.cognizant.orm\_learn.service;

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

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

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

import org.springframework.stereotype.Service;

import java.util.List;

import java.util.Optional;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository repository;

public Employee getEmployee(int id) {

Optional<Employee> result = repository.findById(id);

return result.orElse(null);

}

public List<Employee> getAllEmployees() {

return repository.findAll();

}

public Employee addEmployee(Employee employee) {

return repository.save(employee);

}

public Employee updateEmployee(Employee employee) {

if (repository.existsById(employee.getId())) {

return repository.save(employee);

}

return null;

}

public boolean deleteEmployee(int id) {

if (repository.existsById(id)) {

repository.deleteById(id);

return true;

}

return false;

} }

Application.properties :-

# Logging (Optional)

logging.level.org.springframework=info

logging.level.com.cognizant=debug

logging.level.org.hibernate.SQL=debug

# 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 Dialect

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

# DDL behavior: none, update, create, create-drop, validate

spring.jpa.hibernate.ddl-auto=validate

Output :-

