Cognizant Java FSE – (Deep Skilling) (WEEK-3)

# MODULE : Spring Data JPA with Hibernate

*Submitted by*

Name: S Subhashri

Email: [22102152@rmd.ac.in](mailto:22102152@rmd.ac.in)

College: RMD ENGINEERING COLLEGE

Batch: Java FSE – 2026 Superset ID: 6397078

# Exercise 1: Spring Data JPA - Quick Example Solution:

**Country.java**

package com.cognizant.orm\_learn.model; import jakarta.persistence.Column; import jakarta.persistence.Entity;

import jakarta.persistence.Id; import jakarta.persistence.Table; *@Entity*

*@Table*(name = "country") public class Country {

*@Id*

*@Column*(name = "code") private String code;

*@Column*(name = "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 java.util.Optional;

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();

}

*@Transactional*

public void addCountry(Country country) {

// Only add if country doesn't already exist

if (!countryRepository.existsById(country.getCode())) { countryRepository.save(country);

}

}

*@Transactional*

public Country findCountryByCode(String code) { Optional<Country> result = countryRepository.findById(code); return result.orElse(null);

}

*@Transactional*

public void deleteCountry(String code) { countryRepository.deleteById(code);

}

}

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

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

*@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);

***LOGGER***.info("Inside main");

// 1. Fetch all countries

*testGetAllCountries*();

// 2. Add a new country

*testAddCountry*();

// 3. Find a country by code

*testFindCountryByCode*();

// 4. Delete a country

*testDeleteCountry*();

// 5. Final country list

*testGetAllCountries*();

}

private static void testGetAllCountries() {

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

List<Country> countries = *countryService*.getAllCountries(); ***LOGGER***.debug("countries={}", countries); ***LOGGER***.info("End");

}

private static void testAddCountry() { ***LOGGER***.info("Start - Add Country"); Country country = new Country(); country.setCode("JP"); country.setName("Japan");

*countryService*.addCountry(country); ***LOGGER***.info("Added country: {}", country); ***LOGGER***.info("End");

}

private static void testFindCountryByCode() {

***LOGGER***.info("Start - Find Country");

Country country = *countryService*.findCountryByCode("JP"); ***LOGGER***.info("Country found: {}", country); ***LOGGER***.info("End");

}

private static void testDeleteCountry() {

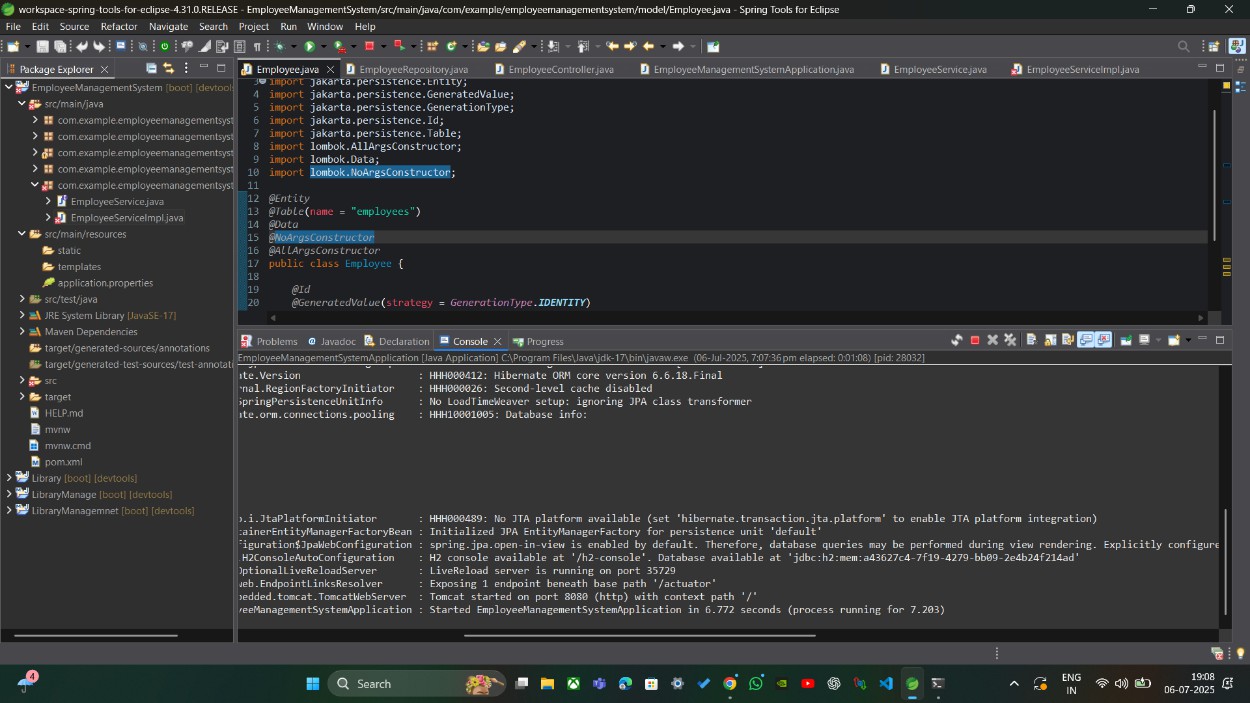
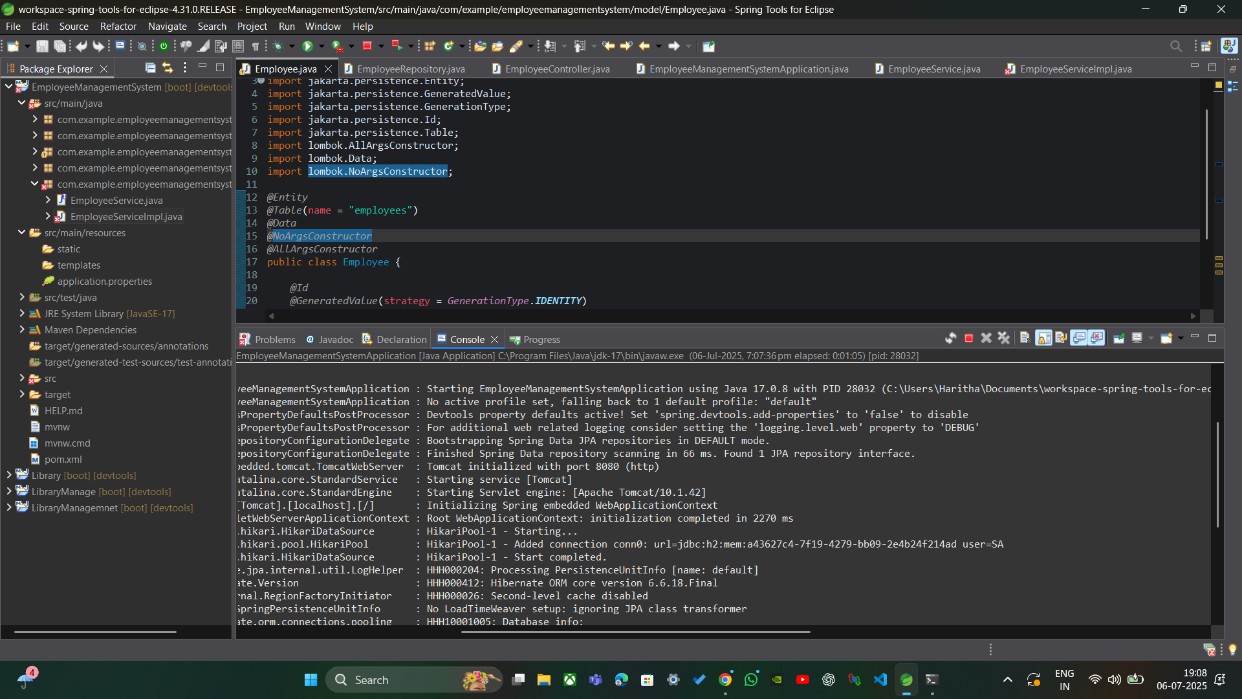
***LOGGER***.info("Start - Delete Country");

*countryService*.deleteCountry("JP"); ***LOGGER***.info("Deleted country with code JP"); ***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

# Solution:

**Employee.java**

package com.cognizant.orm\_learn.model; import jakarta.persistence.\*;

import java.util.Date; import java.util.Set; *@Entity*

*@Table*(name = "employee") public class Employee {

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.***IDENTITY***) *@Column*(name = "em\_id")

private int id;

*@Column*(name = "em\_name") private String name;

*@Column*(name = "em\_salary") private double salary;

*@Column*(name = "em\_permanent") private boolean permanent;

*@Column*(name = "em\_date\_of\_birth") private Date dateOfBirth;

*@ManyToOne*

*@JoinColumn*(name = "em\_dp\_id") private Department department;

*@ManyToMany*(fetch = *FetchType*.***EAGER***) *@JoinTable*(

name = "employee\_skill",

joinColumns = *@JoinColumn*(name = "es\_em\_id"), inverseJoinColumns = *@JoinColumn*(name = "es\_sk\_id")

)

private Set<Skill> skillList; 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 double getSalary() { return salary; }

public void setSalary(double salary) { this.salary = salary; } public boolean isPermanent() { return permanent; }

public void setPermanent(boolean permanent) { this.permanent = permanent; }

public Date getDateOfBirth() { return dateOfBirth; }

public void setDateOfBirth(Date dateOfBirth) { this.dateOfBirth = dateOfBirth; } public Department getDepartment() { return department; }

public void setDepartment(Department department) { this.department = department; } public Set<Skill> getSkillList() { return skillList; }

public void setSkillList(Set<Skill> skillList) { this.skillList = skillList; }

*@Override*

public String toString() {

return "Employee [id=" + id + ", name=" + name + ", salary=" + salary + ", permanent=" + permanent + ", dateOfBirth=" + dateOfBirth + "]";

}

}

# Department.java

package com.cognizant.orm\_learn.model; import jakarta.persistence.\*;

import java.util.Set;

*@Entity*

*@Table*(name = "department") public class Department {

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.*IDENTITY*) *@Column*(name = "dp\_id")

private int id;

*@Column*(name = "dp\_name")

private String name;

*@OneToMany*(mappedBy = "department", fetch = *FetchType*.*EAGER*) private Set<Employee> employeeList;

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 + "]";

}

}

# Skill.java

package com.cognizant.orm\_learn.model; import jakarta.persistence.\*;

import java.util.Set;

*@Entity*

*@Table*(name = "skill") public class Skill {

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.***IDENTITY***) *@Column*(name = "sk\_id")

private int id;

*@Column*(name = "sk\_name") private String name;

*@ManyToMany*(mappedBy = "skillList") private Set<Employee> employeeList; 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 "Skill [id=" + id + ", name=" + name + "]";

}

}

# EmployeeRepository.java

package com.cognizant.orm\_learn.repository;

import org.springframework.data.jpa.repository.JpaRepository; import com.cognizant.orm\_learn.model.Employee;

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {}

# DepartmentRepository.java

package com.cognizant.orm\_learn.repository;

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

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

public interface DepartmentRepository extends JpaRepository<Department, Integer> {}

# SkillRepository.java

package com.cognizant.orm\_learn.repository;

import org.springframework.data.jpa.repository.JpaRepository; import com.cognizant.orm\_learn.model.Skill;

public interface SkillRepository extends JpaRepository<Skill, Integer> {}

# EmployeeService.java

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 jakarta.transaction.Transactional; import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

*@Service*

public class EmployeeService {

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

*@Autowired*

private EmployeeRepository employeeRepository;

*@Transactional*

public Employee get(int id) {

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

return employeeRepository.findById(id).get();

}

*@Transactional*

public void save(Employee employee) { ***LOGGER***.info("Start"); employeeRepository.save(employee); ***LOGGER***.info("End");

}

}

# 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;

import jakarta.transaction.Transactional;

*@Service*

public class DepartmentService {

*@Autowired*

private DepartmentRepository departmentRepository;

*@Transactional*

public Department get(int id) {

return departmentRepository.findById(id).get();

}

*@Transactional*

public void save(Department department) { departmentRepository.save(department);

}

}

# SkillService.java

package com.cognizant.orm\_learn.service; import com.cognizant.orm\_learn.model.Skill;

import com.cognizant.orm\_learn.repository.SkillRepository; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.stereotype.Service;

import jakarta.transaction.Transactional;

*@Service*

public class SkillService {

*@Autowired*

private SkillRepository skillRepository;

*@Transactional*

public Skill get(int id) {

return skillRepository.findById(id).get();

}

*@Transactional*

public void save(Skill skill) { skillRepository.save(skill);

}

}

# 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.model.Skill;

import com.cognizant.orm\_learn.service.DepartmentService; import com.cognizant.orm\_learn.service.EmployeeService; import com.cognizant.orm\_learn.service.SkillService; 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 java.util.Date; import java.util.HashSet; import java.util.Set;

*@SpringBootApplication*

public class OrmLearnApplication { private static final Logger ***LOGGER*** =

LoggerFactory.*getLogger*(OrmLearnApplication.class);

private static EmployeeService *employeeService*;

private static DepartmentService *departmentService*; private static SkillService *skillService*;

public static void main(String[] args) {

ApplicationContext context = SpringApplication.*run*(OrmLearnApplication.class, args);

*employeeService* = context.getBean(EmployeeService.class);

*departmentService* = context.getBean(DepartmentService.class); *skillService* = context.getBean(SkillService.class); ***LOGGER***.info("Inside main");

// Uncomment the methods below one by one to test

*testAddEmployee*();

//testGetEmployee();

*testUpdateEmployee*(); *testGetDepartment*();

*testAddSkillToEmployee*();

}

private static void testAddEmployee() {

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

Employee employee = new Employee(); employee.setName("Jane Doe"); employee.setSalary(60000.0); employee.setPermanent(true); employee.setDateOfBirth(new Date());

// Set Department

Department department = *departmentService*.get(1); // ensure this exists in DB employee.setDepartment(department);

// Set Skills

Set<Skill> skillList = new HashSet<>(); skillList.add(*skillService*.get(1)); // Java skillList.add(*skillService*.get(2)); // Spring Boot employee.setSkillList(skillList);

// Save Employee

*employeeService*.save(employee);

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

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

}

private static void testGetEmployee() {

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

Employee employee = *employeeService*.get(1); // use valid employee id ***LOGGER***.debug("Employee: {}", employee); ***LOGGER***.debug("Department: {}", employee.getDepartment()); ***LOGGER***.debug("Skills: {}", employee.getSkillList()); ***LOGGER***.info("End");

}

private static void testUpdateEmployee() {

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

Employee employee = *employeeService*.get(1); // use valid employee id Department newDept = *departmentService*.get(2); // different department ID employee.setDepartment(newDept);

*employeeService*.save(employee); ***LOGGER***.debug("Updated Employee: {}", employee); ***LOGGER***.info("End");

}

private static void testGetDepartment() {

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

Department department = *departmentService*.get(1); // department with multiple employees

***LOGGER***.debug("Department: {}", department); ***LOGGER***.debug("Employees: {}", department.getEmployeeList()); ***LOGGER***.info("End");

}

private static void testAddSkillToEmployee() {

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

Employee employee = *employeeService*.get(1); // valid employee id

Skill skill = *skillService*.get(2); // valid skill id not already assigned Set<Skill> skillList = employee.getSkillList();

if (skillList == null) {

skillList = new HashSet<>();

}

skillList.add(skill); employee.setSkillList(skillList);

*employeeService*.save(employee);

***LOGGER***.debug("Updated Employee with new skill: {}", employee);

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

}

}

**Output:**

A screen shot of a computer

AI-generated content may be incorrect.