Digital Nurture 4.0

Deep Skilling - Java FSE

WEEK-3 HANDS ON

By Kaviya P

**2.SPRING DATA JPA HANSON**

**Hands on 1**

**Write queries on country table using Query Methods**   
  
Following are the list of queries that is required for an application. Implement these queries using Query Methods feature of Spring Data JPA. Click [**here**](https://docs.spring.io/spring-data/jpa/docs/2.2.0.RELEASE/reference/html/#jpa.query-methods.query-creation) for reference. Include appropriate methods in OrmLearnApplication and test the same. 

* An application has a search text box for searching by country. When typing characters on the text box, a list of all the matching countries should be displayed. For example, if 'ou' is entered in the search box the following countries should be displayed. Write a Query Method to achieve this feature. Implement this method in CountryRepository.

**SpringAndMavenApplication.java**

**package** com.week3.SpringAndMaven;

**import** org.springframework.boot.CommandLineRunner;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**import** com.week3.SpringAndMaven.model.Country;

**import** com.week3.SpringAndMaven.service.CountryService;

**import** com.week3.SpringAndMaven.service.exception.CountryNotFoundException;

**import** jakarta.annotation.PostConstruct;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** java.util.List;

**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;

@SpringBootApplication

**public** **class** SpringAndMavenApplication **implements** CommandLineRunner{

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

@Autowired

**private** CountryService countryService;

**public** **static** **void** main(String[] args) {

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

}

@Override

**public** **void** run(String... args) {

testFindByNameContaining();

testFindByNameContainingOrderByNameAsc();

testFindByNameStartingWith();

}

**private** **void** testFindByNameContaining() {

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

List<Country> countries = countryService.findByNameContaining("ou");

countries.forEach(c -> ***LOGGER***.debug("Country: {} - {}", c.getCode(), c.getName()));

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

}

**private** **void** testFindByNameContainingOrderByNameAsc() {

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

List<Country> countries = countryService.findByNameContainingOrderByNameAsc("ou");

countries.forEach(c -> ***LOGGER***.debug("Country: {} - {}", c.getCode(), c.getName()));

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

}

**private** **void** testFindByNameStartingWith() {

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

List<Country> countries = countryService.findByNameStartingWith("Z");

countries.forEach(c -> ***LOGGER***.debug("Country: {} - {}", c.getCode(), c.getName()));

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

}

}

**CountryController.java**

**package** com.week3.SpringAndMaven.controller;

**import** java.util.List;

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

**import** org.springframework.http.HttpStatus;

**import** org.springframework.http.ResponseEntity;

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

**import** com.week3.SpringAndMaven.model.Country;

**import** com.week3.SpringAndMaven.service.CountryService;

**import** com.week3.SpringAndMaven.service.exception.CountryNotFoundException;

@RestController

@RequestMapping("/countries")

**public** **class** CountryController {

@Autowired

**private** CountryService countryService;

// GET /countries/search?name=ou

@GetMapping("/search")

**public** List<Country> getCountriesByNameContains(@RequestParam String name) {

**return** countryService.findByNameContaining(name);

}

// GET /countries/search-sorted?name=ou

@GetMapping("/search-sorted")

**public** List<Country> getCountriesByNameContainsSorted(@RequestParam String name) {

**return** countryService.findByNameContainingOrderByNameAsc(name);

}

// GET /countries/starts-with?prefix=Z

@GetMapping("/starts-with")

**public** List<Country> getCountriesStartingWith(@RequestParam String prefix) {

**return** countryService.findByNameStartingWith(prefix);

}

}

**COUNTRY.java**

**package** com.week3.SpringAndMaven.model;

**import** jakarta.persistence.\*;

@Entity

@Table(name = "country")

**public** **class** Country {

@Id

@Column(name = "co\_code")

**private** String code;

@Column(name = "co\_name")

**private** String name;

**public** Country() {}

**public** Country(String code, String name) {

**this**.code = code;

**this**.name = 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;

}

}

**CountryRepository java**

**package** com.week3.SpringAndMaven.repository;

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

**import** org.springframework.data.jpa.repository.Query;

**import** org.springframework.stereotype.Repository;

**import** com.week3.SpringAndMaven.model.Country;

**import** java.util.List;

@Repository

**public** **interface** CountryRepository **extends** JpaRepository<Country, String> {

List<Country> findByNameContaining(String substring);

List<Country> findByNameContainingOrderByNameAsc(String substring);

List<Country> findByNameStartingWith(String prefix);

}

**CountryService.java**

**package** com.week3.SpringAndMaven.service;

**import** java.util.\*;

**import** javax.management.loading.ClassLoaderRepository;

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

**import** org.springframework.stereotype.Service;

**import** com.week3.SpringAndMaven.model.Country;

**import** com.week3.SpringAndMaven.repository.CountryRepository;

**import** com.week3.SpringAndMaven.service.exception.CountryNotFoundException;

**import** jakarta.transaction.Transactional;

@Service

**public** **class** CountryService {

@Autowired

**private** CountryRepository countryRepository;

**public** List<Country> findByNameContaining(String substring) {

**return** countryRepository.findByNameContaining(substring);

}

**public** List<Country> findByNameContainingOrderByNameAsc(String substring) {

**return** countryRepository.findByNameContainingOrderByNameAsc(substring);

}

**public** List<Country> findByNameStartingWith(String prefix) {

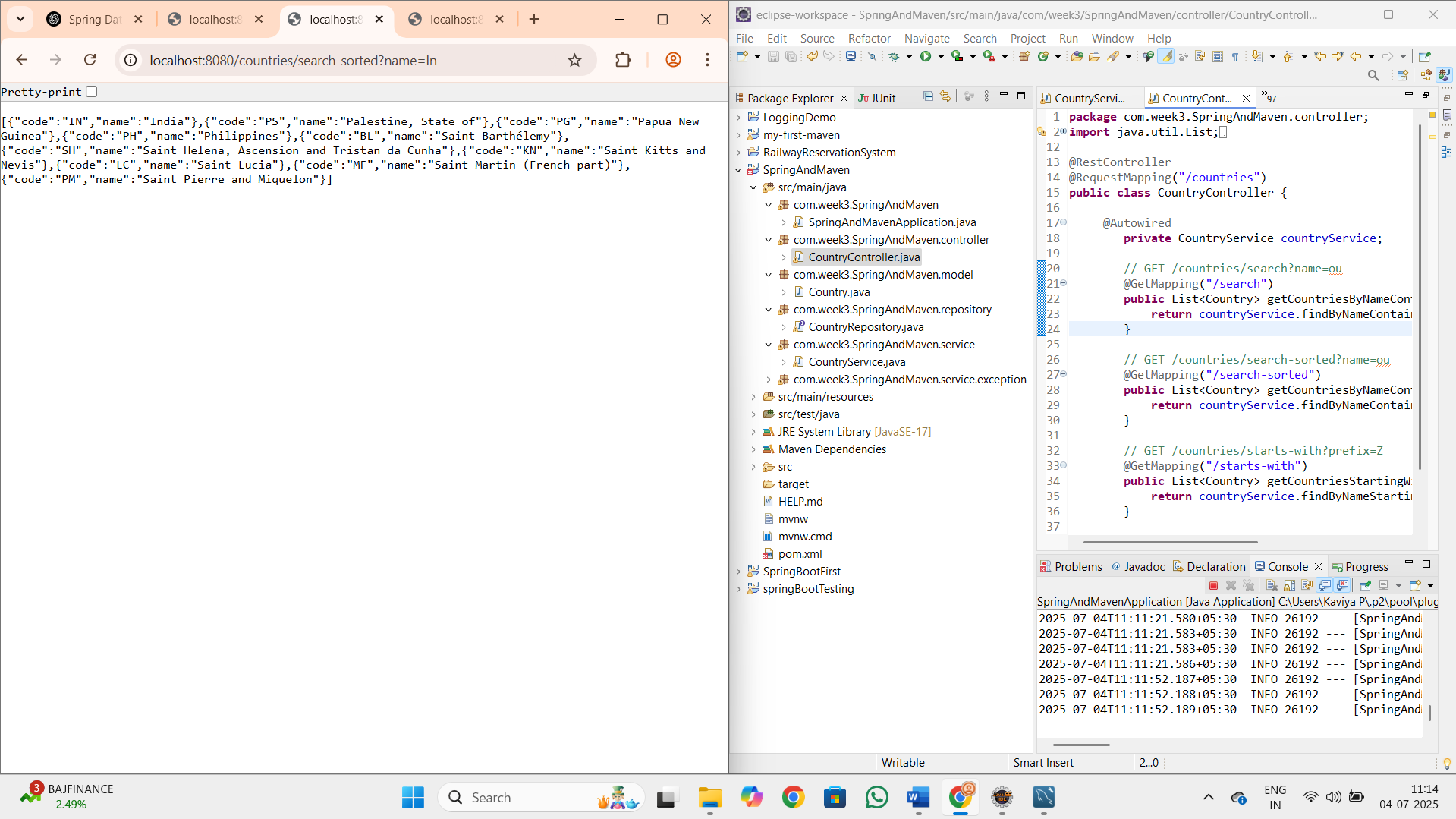
**return** countryRepository.findByNameStartingWith(prefix);

}

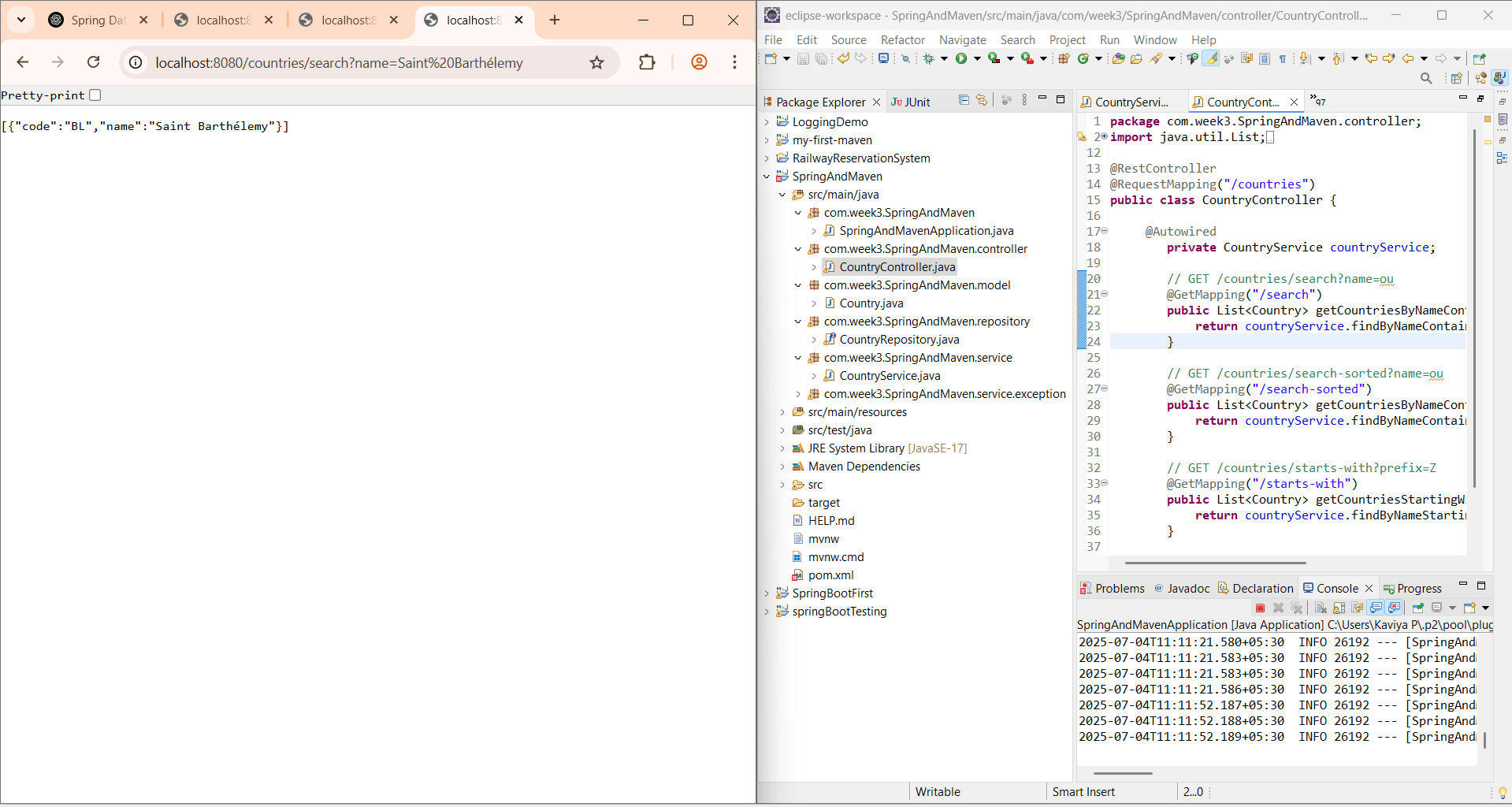
}

**OUTPUT**

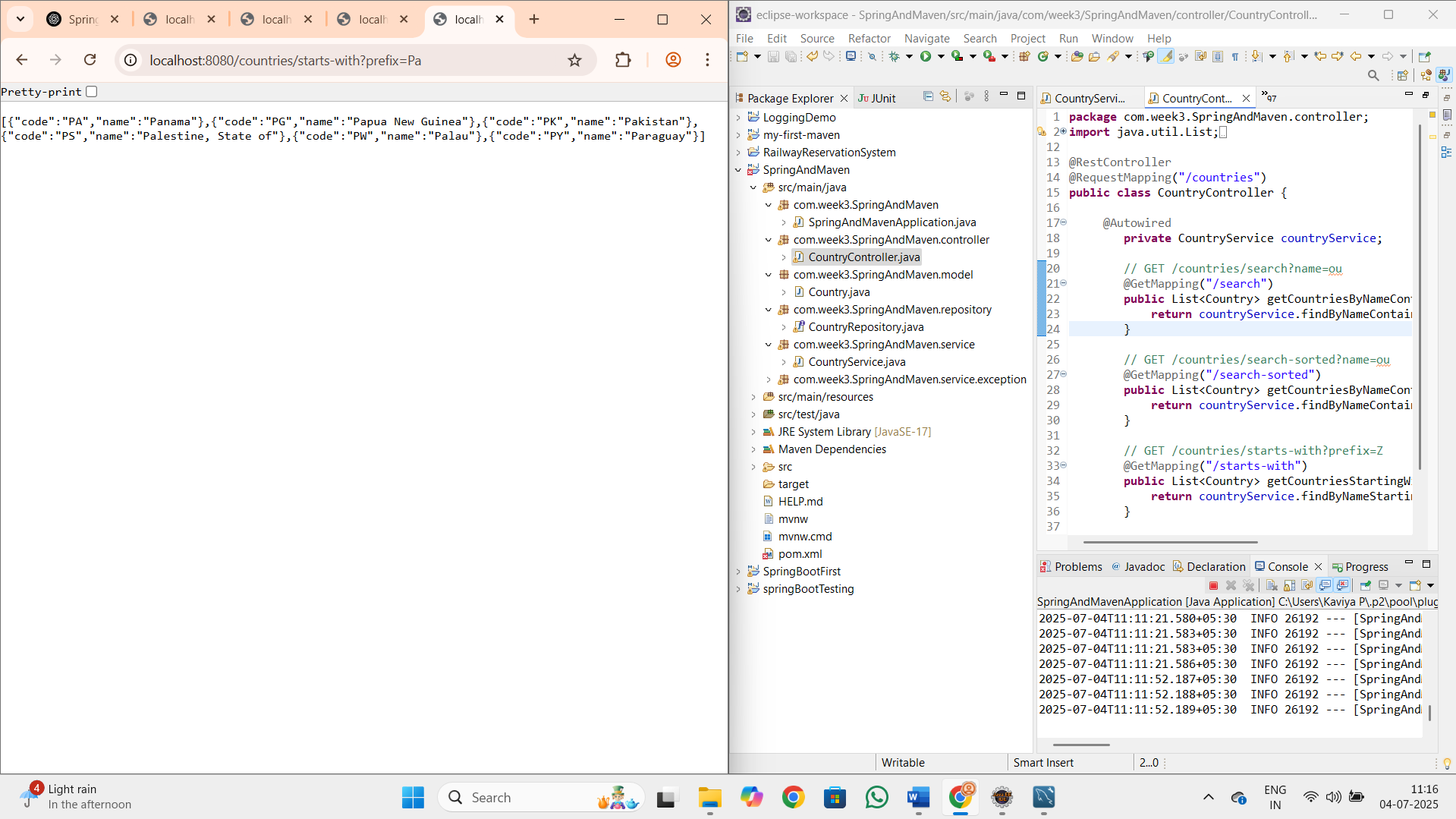
**FOR SEARCH**

****

**FOR SORT**

****

**FOR PREFIX**

****

**Hands on 2**

**Write queries on stock table using Query Methods**

**SpringAndMavenApplication.java**

**package** com.week3.SpringAndMaven;

**import** java.text.SimpleDateFormat;

**import** java.util.\*;

**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** org.springframework.boot.CommandLineRunner;

**import** org.springframework.context.annotation.Bean;

**import** com.week3.SpringAndMaven.model.Stock;

**import** com.week3.SpringAndMaven.repository.StockRepository;

@SpringBootApplication

**public** **class** SpringAndMavenApplication {

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

@Autowired

**private** StockRepository stockRepository;

**public** **static** **void** main(String[] args) {

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

}

@Bean

**public** CommandLineRunner testStockQueries() {

**return** args -> {

SimpleDateFormat sdf = **new** SimpleDateFormat("yyyy-MM-dd");

// 1. Facebook in Sep 2019

***LOGGER***.info("Facebook stocks in September 2019");

List<Stock> fbStocks = stockRepository.findByCodeAndDateBetween("FB",

sdf.parse("2019-09-01"), sdf.parse("2019-09-30"));

fbStocks.forEach(stock -> ***LOGGER***.info(stock.toString()));

// 2. Google stock > 1250

***LOGGER***.info("Google stocks with close > 1250");

List<Stock> googStocks = stockRepository.findByCodeAndCloseGreaterThan("GOOGL", **new** java.math.BigDecimal("1250"));

googStocks.forEach(stock -> ***LOGGER***.info(stock.toString()));

// 3. Top 3 volumes

***LOGGER***.info("Top 3 volumes");

List<Stock> topVolume = stockRepository.findTop3ByOrderByVolumeDesc();

topVolume.forEach(stock -> ***LOGGER***.info(stock.toString()));

// 4. Netflix lowest closing prices

***LOGGER***.info("Lowest Netflix stocks");

List<Stock> lowestNetflix = stockRepository.findTop3ByCodeOrderByCloseAsc("NFLX");

lowestNetflix.forEach(stock -> ***LOGGER***.info(stock.toString()));

};

}

}

**Stock.java**

**package** com.week3.SpringAndMaven.model;

**import** jakarta.persistence.\*;

**import** java.math.BigDecimal;

**import** java.util.Date;

@Entity

@Table(name = "stock")

**public** **class** Stock {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

@Column(name = "st\_id")

**private** **int** id;

@Column(name = "st\_code")

**private** String code;

@Temporal(TemporalType.***DATE***)

@Column(name = "st\_date")

**private** Date date;

@Column(name = "st\_open")

**private** BigDecimal open;

@Column(name = "st\_close")

**private** BigDecimal close;

@Column(name = "st\_volume")

**private** BigDecimal volume;

// Getters and Setters

**public** **int** getId() { **return** id; }

**public** **void** setId(**int** id) { **this**.id = id; }

**public** String getCode() { **return** code; }

**public** **void** setCode(String code) { **this**.code = code; }

**public** Date getDate() { **return** date; }

**public** **void** setDate(Date date) { **this**.date = date; }

**public** BigDecimal getOpen() { **return** open; }

**public** **void** setOpen(BigDecimal open) { **this**.open = open; }

**public** BigDecimal getClose() { **return** close; }

**public** **void** setClose(BigDecimal close) { **this**.close = close; }

**public** BigDecimal getVolume() { **return** volume; }

**public** **void** setVolume(BigDecimal volume) { **this**.volume = volume; }

}

**StockRepository.java**

**package** com.week3.SpringAndMaven.repository;

**import** java.math.BigDecimal;

**import** java.util.\*;

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

**import** com.week3.SpringAndMaven.model.Stock;

**public** **interface** StockRepository **extends** JpaRepository<Stock, Integer> {

// 1. Facebook stocks in Sep 2019

List<Stock> findByCodeAndDateBetween(String code, Date start, Date end);

// 2. Google stock details where close > 1250

List<Stock> findByCodeAndCloseGreaterThan(String code, BigDecimal close);

// 3. Top 3 by highest volume

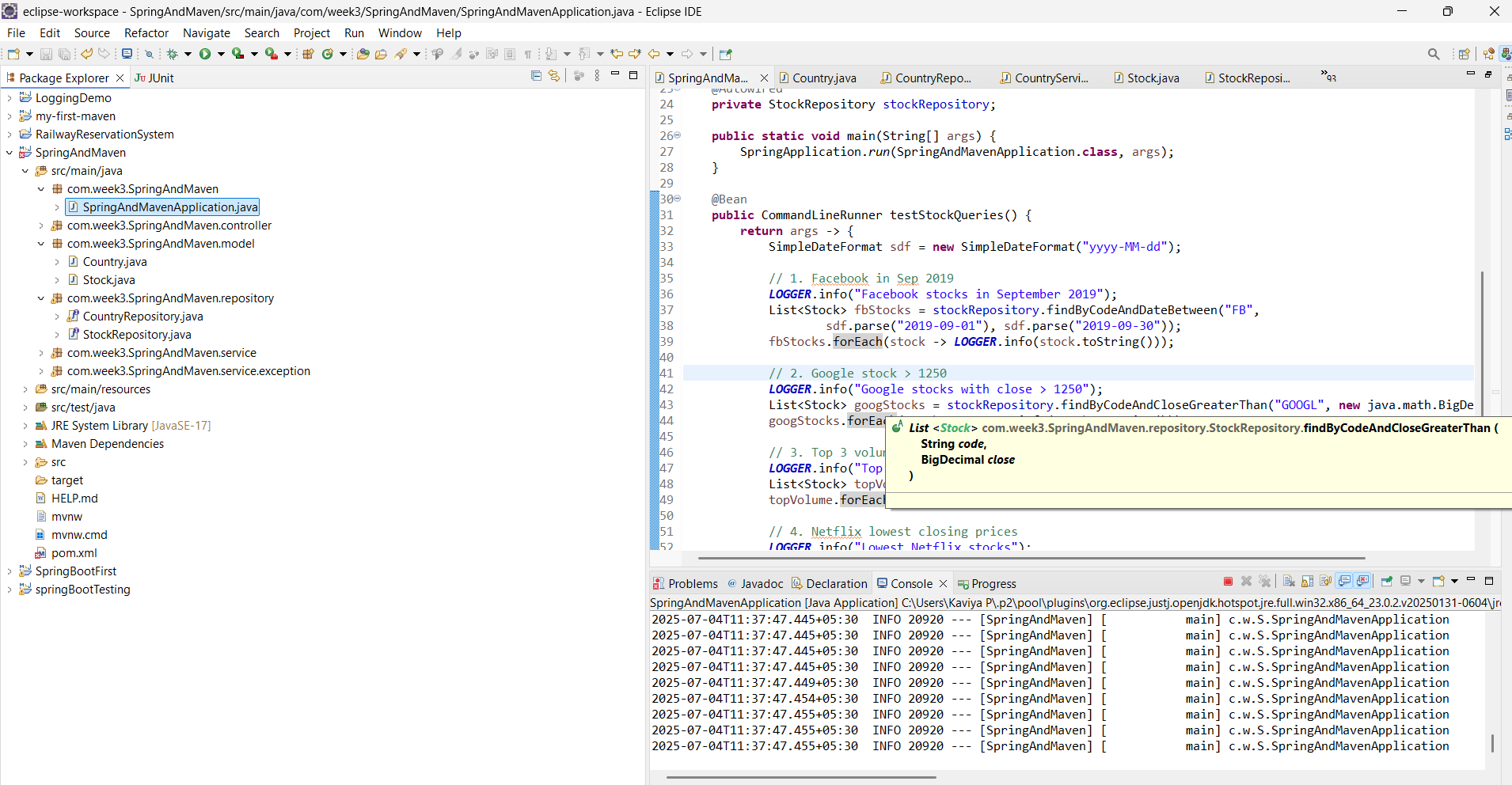
List<Stock> findTop3ByOrderByVolumeDesc();

// 4. Netflix stocks lowest - ascending sort

List<Stock> findTop3ByCodeOrderByCloseAsc(String code);

}

**OUTPUT :**

****

**Hands on 3**

**Create payroll tables and bean mapping**

**SQL**

create database Employee;

use Employee;

CREATE TABLE department (

    dp\_id INT AUTO\_INCREMENT PRIMARY KEY,

    dp\_name VARCHAR(45)

);

CREATE TABLE skill (

    sk\_id INT AUTO\_INCREMENT PRIMARY KEY,

    sk\_name VARCHAR(45)

);

CREATE TABLE employee (

    em\_id INT AUTO\_INCREMENT PRIMARY KEY,

    em\_name VARCHAR(45),

    em\_salary DECIMAL(10,2),

    em\_permanent BOOLEAN,

    em\_date\_of\_birth DATE,

    em\_dp\_id INT,

    FOREIGN KEY (em\_dp\_id) REFERENCES department(dp\_id)

);

CREATE TABLE employee\_skill (

    es\_id INT AUTO\_INCREMENT PRIMARY KEY,

    es\_em\_id INT,

    es\_sk\_id INT,

    FOREIGN KEY (es\_em\_id) REFERENCES employee(em\_id),

    FOREIGN KEY (es\_sk\_id) REFERENCES skill(sk\_id)

);

**Department.java**

**package** com.week3.SpringAndMaven.model;

**import** java.util.List;

**import** jakarta.persistence.\*;

@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")

**private** List<Employee> employeeList;

// Getters, Setters, toString()

}

**Employee.java**

**package** com.week3.SpringAndMaven.model;

**import** java.util.\*;

**import** jakarta.persistence.\*;

@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

@JoinTable(name = "employee\_skill",

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

inverseJoinColumns = @JoinColumn(name = "es\_sk\_id"))

**private** List<Skill> skillList;

// Getters, Setters, toString()

}

**Skill.java**

**package** com.week3.SpringAndMaven.model;

**import** java.util.List;

**import** jakarta.persistence.\*;

@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** List<Employee> employeeList;

// Getters, Setters, toString()

}

**DepartmentRepository.java**

**package** com.week3.SpringAndMaven.repository;

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

**import** org.springframework.stereotype.Repository;

**import** com.week3.SpringAndMaven.model.Department;

@Repository

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

}

**EmployeeRepository.java**

**package** com.week3.SpringAndMaven.repository;

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

**import** org.springframework.stereotype.Repository;

**import** com.week3.SpringAndMaven.model.Employee;

@Repository

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

}

**SkillRepository.java**

**package** com.week3.SpringAndMaven.repository;

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

**import** org.springframework.stereotype.Repository;

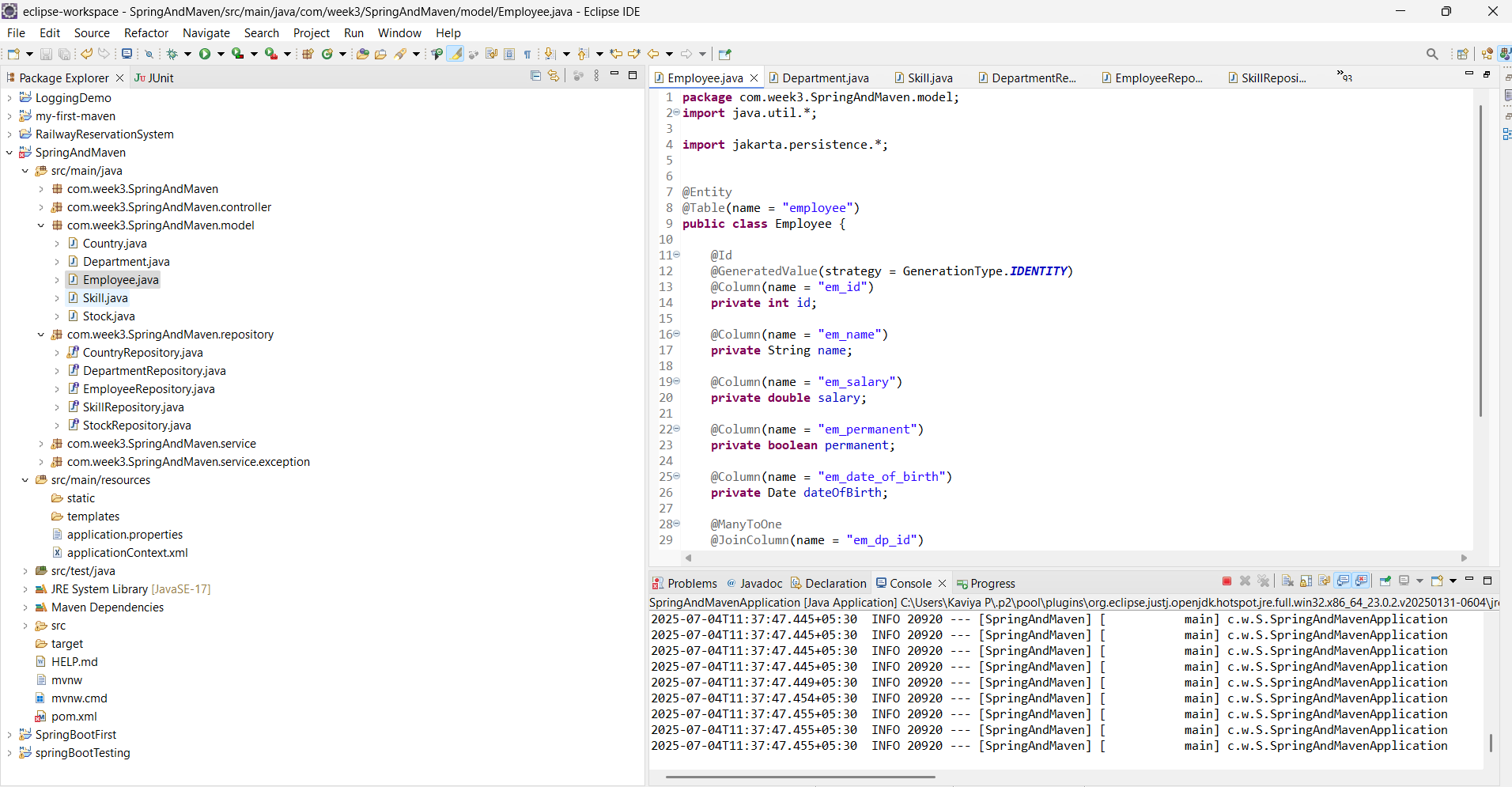
**import** com.week3.SpringAndMaven.model.Skill;

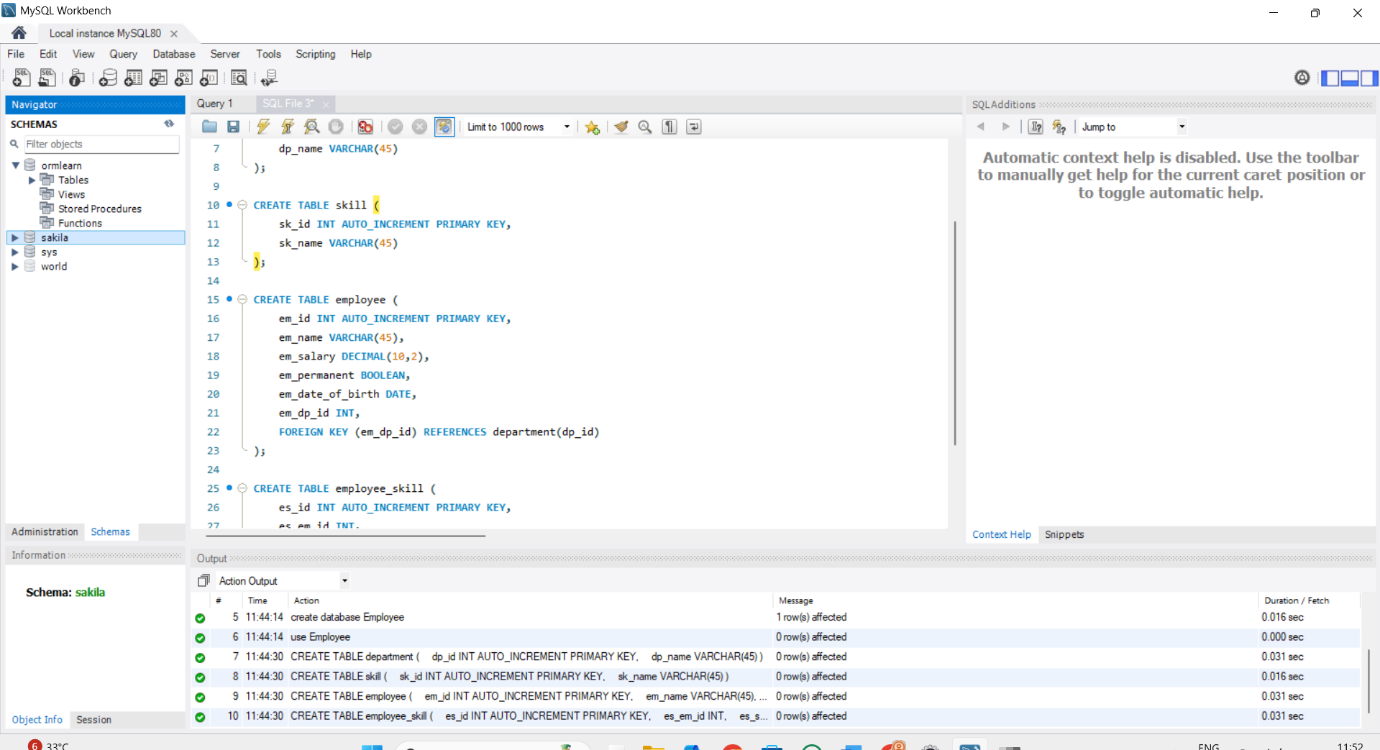
@Repository

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

}

**OUTPUT**





**Hands on 4**

**Implement many to one relationship between Employee and Department**

**DepartmentService.java**

**package** com.week3.SpringAndMaven.service;

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

**import** org.springframework.stereotype.Service;

**import** com.week3.SpringAndMaven.model.Department;

**import** com.week3.SpringAndMaven.repository.DepartmentRepository;

**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);

}

}

**EmployeeService.java**

**package** com.week3.SpringAndMaven.service;

**import** org.springframework.stereotype.Service;

**import** org.springframework.transaction.annotation.Transactional;

**import** com.week3.SpringAndMaven.model.Employee;

**import** com.week3.SpringAndMaven.repository.EmployeeRepository;

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

**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");

}

}

**Department.java**

**package** com.week3.SpringAndMaven.model;

**import** java.util.List;

**import** jakarta.persistence.\*;

@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")

**private** List<Employee> employeeList;

// Getters, Setters, toString()

}

**Employee.java**

**package** com.week3.SpringAndMaven.model;

**import** java.util.\*;

**import** jakarta.persistence.\*;

@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

@JoinTable(name = "employee\_skill",

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

inverseJoinColumns = @JoinColumn(name = "es\_sk\_id"))

**private** List<Skill> skillList;

// Getters, Setters, toString()

}

**Skill.java**

**package** com.week3.SpringAndMaven.model;

**import** java.util.List;

**import** jakarta.persistence.\*;

@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** List<Employee> employeeList;

// Getters, Setters, toString()

}

**DepartmentRepository.java**

**package** com.week3.SpringAndMaven.repository;

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

**import** org.springframework.stereotype.Repository;

**import** com.week3.SpringAndMaven.model.Department;

@Repository

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

}

**EmployeeRepository.java**

**package** com.week3.SpringAndMaven.repository;

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

**import** org.springframework.stereotype.Repository;

**import** com.week3.SpringAndMaven.model.Employee;

@Repository

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

}

**SkillRepository.java**

**package** com.week3.SpringAndMaven.repository;

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

**import** org.springframework.stereotype.Repository;

**import** com.week3.SpringAndMaven.model.Skill;

@Repository

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

}

**SpringAndMavenApplication.java**

**package** com.week3.SpringAndMaven;

**import** java.text.SimpleDateFormat;

**import** java.util.\*;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

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

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.boot.CommandLineRunner;

**import** org.springframework.context.annotation.Bean;

**import** com.week3.SpringAndMaven.model.\*;

**import** com.week3.SpringAndMaven.repository.StockRepository;

**import** com.week3.SpringAndMaven.service.\*;

@SpringBootApplication

**public** **class** SpringAndMavenApplication {

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

**private** **static** EmployeeService *employeeService*;

**private** **static** DepartmentService *departmentService*;

**public** **static** **void** main(String[] args) **throws** Exception {

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

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

*departmentService* = context.getBean(DepartmentService.**class**);

// Uncomment each to test

// testGetEmployee();

// testAddEmployee();

// testUpdateEmployee();

}

**private** **static** **void** testGetEmployee() {

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

Employee employee = *employeeService*.get(1);

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

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

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

}

**private** **static** **void** testAddEmployee() **throws** Exception {

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

Employee employee = **new** Employee();

employee.setName("John");

employee.setSalary(45000.0);

employee.setPermanent(**true**);

employee.setDateOfBirth(**new** SimpleDateFormat("yyyy-MM-dd").parse("1990-05-01"));

Department department = *departmentService*.get(1); // Assume dept id 1 exists

employee.setDepartment(department);

*employeeService*.save(employee);

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

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

}

**private** **static** **void** testUpdateEmployee() **throws** Exception {

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

Employee employee = *employeeService*.get(1); // Assume emp id 1 exists

Department department = *departmentService*.get(2); // Change to a different dept

employee.setDepartment(department);

*employeeService*.save(employee);

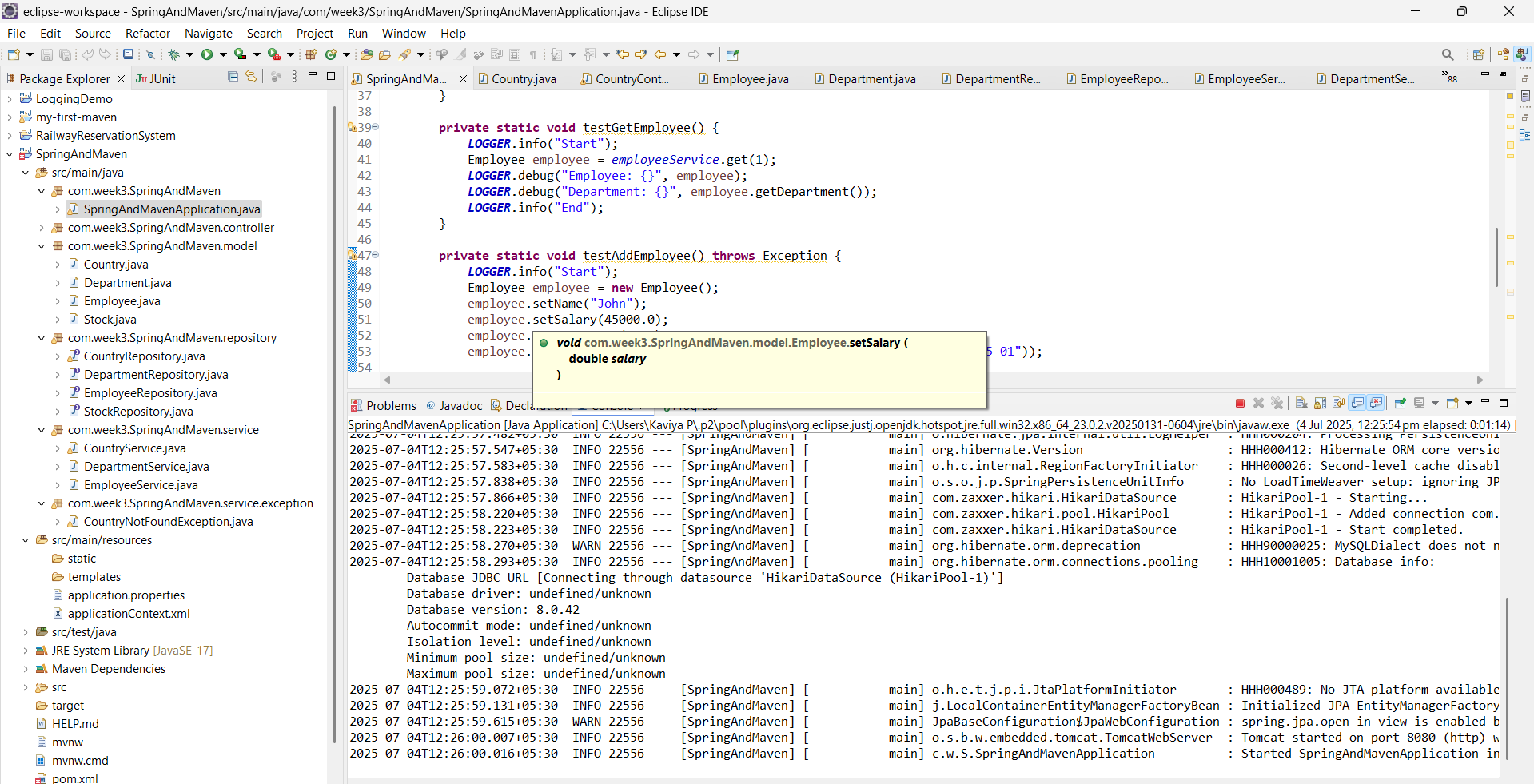
***LOGGER***.debug("Updated Employee: {}", employee);

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

}

}

**OUTPUT**

****

**Hands on 5**

**Implement one to many relationship between Employee and Department**   
  
**Employee.java**

**package** com.week3.SpringAndMaven.model;

**import** java.util.\*;

**import** jakarta.persistence.\*;

@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

@JoinTable(name = "employee\_skill",

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

inverseJoinColumns = @JoinColumn(name = "es\_sk\_id"))

**private** List<Skill> skillList;

// Getters, Setters, toString()

}

**Skill.java**

**package** com.week3.SpringAndMaven.model;

**import** java.util.List;

**import** jakarta.persistence.\*;

@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** List<Employee> employeeList;

// Getters, Setters, toString()

}

**DepartmentRepository.java**

**package** com.week3.SpringAndMaven.repository;

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

**import** org.springframework.stereotype.Repository;

**import** com.week3.SpringAndMaven.model.Department;

@Repository

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

}

**EmployeeRepository.java**

**package** com.week3.SpringAndMaven.repository;

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

**import** org.springframework.stereotype.Repository;

**import** com.week3.SpringAndMaven.model.Employee;

@Repository

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

}

**Department.java**

**package** com.week3.SpringAndMaven.model;

**import** java.util.Set;

**import** jakarta.persistence.\*;

@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***) // EAGER fetch to avoid LazyInitializationException

**private** Set<Employee> employeeList;

// Getters and Setters

**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 + "]";

}

}

**SkillRepository.java**

**package** com.week3.SpringAndMaven.repository;

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

**import** org.springframework.stereotype.Repository;

**import** com.week3.SpringAndMaven.model.Skill;

@Repository

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

}

**DepartmentService.java**

**package** com.week3.SpringAndMaven.service;

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

**import** org.springframework.stereotype.Service;

**import** com.week3.SpringAndMaven.model.Department;

**import** com.week3.SpringAndMaven.repository.DepartmentRepository;

**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);

}

}

**EmployeeService.java**

**package** com.week3.SpringAndMaven.service;

**import** org.springframework.stereotype.Service;

**import** org.springframework.transaction.annotation.Transactional;

**import** com.week3.SpringAndMaven.model.Employee;

**import** com.week3.SpringAndMaven.repository.EmployeeRepository;

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

**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");

}

}

**SpringAndMavenApplication.java**

**package** com.week3.SpringAndMaven;

**import** java.text.SimpleDateFormat;

**import** java.util.\*;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

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

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.boot.CommandLineRunner;

**import** org.springframework.context.annotation.Bean;

**import** com.week3.SpringAndMaven.model.\*;

**import** com.week3.SpringAndMaven.repository.StockRepository;

**import** com.week3.SpringAndMaven.service.\*;

@SpringBootApplication

**public** **class** SpringAndMavenApplication {

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

**private** **static** EmployeeService *employeeService*;

**private** **static** DepartmentService *departmentService*;

**public** **static** **void** main(String[] args) **throws** Exception {

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

*departmentService* = context.getBean(DepartmentService.**class**);

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

*testGetDepartment*();

// Uncomment each to test

// testGetEmployee();

// testAddEmployee();

// testUpdateEmployee();

}

**private** **static** **void** testGetDepartment() {

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

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

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

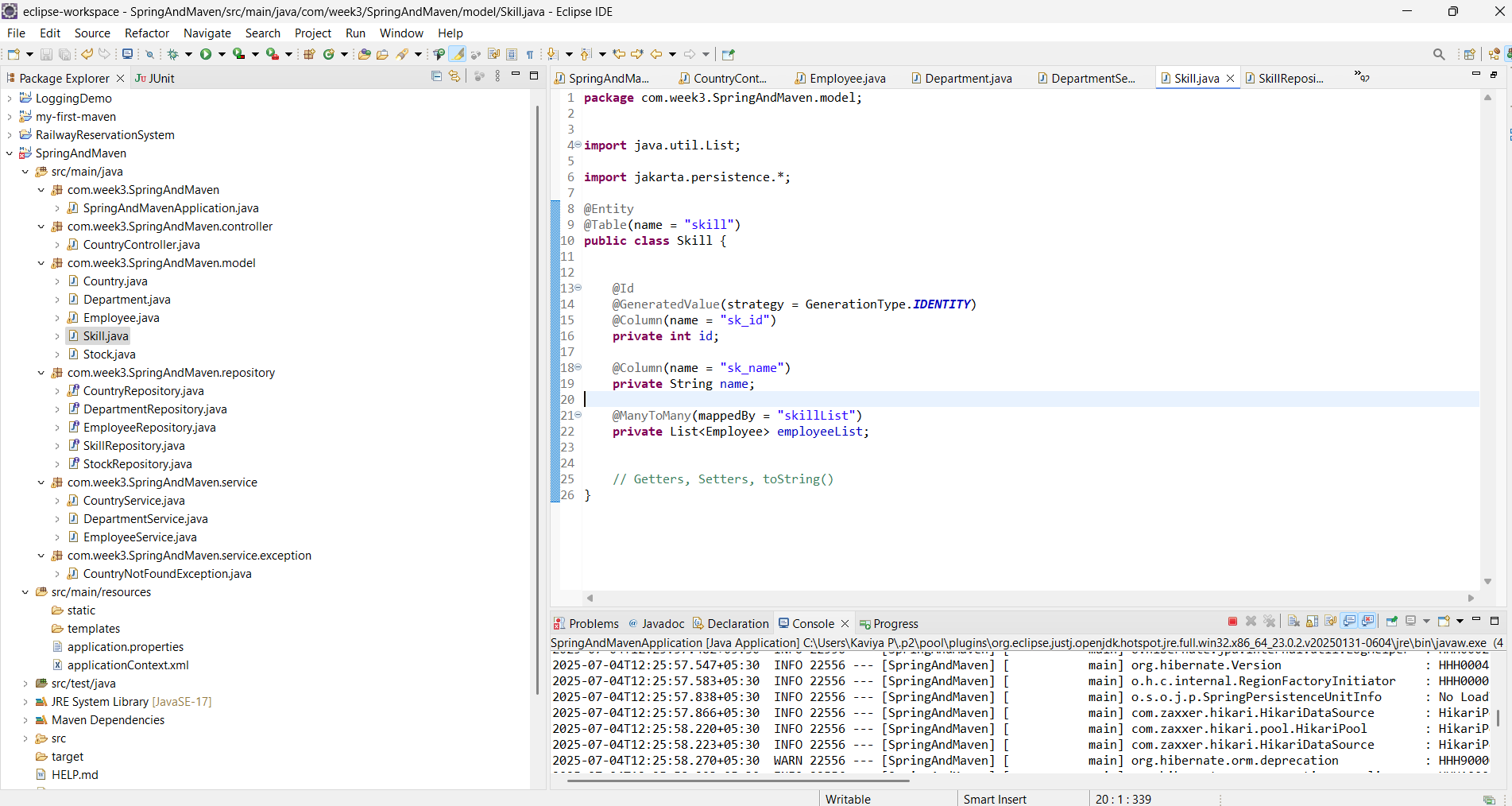
***LOGGER***.debug("Employees: {}", department.getEmployeeList());

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

}

}

**OUTPUT**

****

**Hands on 6**

**Implement many to many relationship between Employee and Skill**

**Employee.java**

**package** com.week3.SpringAndMaven.model;

**import** jakarta.persistence.\*;

**import** java.util.Date;

**import** java.util.HashSet;

**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;

@Temporal(TemporalType.***DATE***)

@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 = **new** HashSet<>();

// Getters and Setters

**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 + "]";

}

}

**Skill.java**

**package** com.week3.SpringAndMaven.model;

**import** java.util.Set;

**import** jakarta.persistence.\*;

@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;

// Getters and Setters

**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 + "]";

}

}

**DepartmentRepository.java**

**package** com.week3.SpringAndMaven.repository;

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

**import** org.springframework.stereotype.Repository;

**import** com.week3.SpringAndMaven.model.Department;

@Repository

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

}

**EmployeeRepository.java**

**package** com.week3.SpringAndMaven.repository;

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

**import** org.springframework.stereotype.Repository;

**import** com.week3.SpringAndMaven.model.Employee;

@Repository

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

}

**Department.java**

**package** com.week3.SpringAndMaven.model;

**import** java.util.Set;

**import** jakarta.persistence.\*;

@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***) // EAGER fetch to avoid LazyInitializationException

**private** Set<Employee> employeeList;

// Getters and Setters

**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 + "]";

}

}

**SkillRepository.java**

**package** com.week3.SpringAndMaven.repository;

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

**import** org.springframework.stereotype.Repository;

**import** com.week3.SpringAndMaven.model.Skill;

@Repository

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

}

**DepartmentService.java**

**package** com.week3.SpringAndMaven.service;

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

**import** org.springframework.stereotype.Service;

**import** com.week3.SpringAndMaven.model.Department;

**import** com.week3.SpringAndMaven.repository.DepartmentRepository;

**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);

}

}

**EmployeeService.java**

**package** com.week3.SpringAndMaven.service;

**import** org.springframework.stereotype.Service;

**import** org.springframework.transaction.annotation.Transactional;

**import** com.week3.SpringAndMaven.model.Employee;

**import** com.week3.SpringAndMaven.repository.EmployeeRepository;

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

**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)

.orElseThrow(() -> **new** RuntimeException("Employee not found with id: " + id));

}

@Transactional

**public** **void** save(Employee employee) {

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

employeeRepository.save(employee);

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

}

}

**SkillService.java**

**package** com.week3.SpringAndMaven.service;

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

**import** org.springframework.stereotype.Service;

**import** com.week3.SpringAndMaven.model.Skill;

**import** com.week3.SpringAndMaven.repository.SkillRepository;

**import** jakarta.transaction.Transactional;

@Service

**public** **class** SkillService {

@Autowired

**private** SkillRepository skillRepository;

@Transactional

**public** Skill get(**int** id) {

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

}

}

**SpringAndMavenApplication.java**

**package** com.week3.SpringAndMaven;

**import** java.text.SimpleDateFormat;

**import** java.util.\*;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

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

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.boot.CommandLineRunner;

**import** org.springframework.context.annotation.Bean;

**import** com.week3.SpringAndMaven.model.\*;

**import** com.week3.SpringAndMaven.repository.StockRepository;

**import** com.week3.SpringAndMaven.service.\*;

@SpringBootApplication

**public** **class** SpringAndMavenApplication {

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

**private** **static** EmployeeService *employeeService*;

**private** **static** DepartmentService *departmentService*;

**public** **static** **void** main(String[] args) **throws** Exception {

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

*departmentService* = context.getBean(DepartmentService.**class**);

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

*testGetDepartment*();

// Uncomment each to test

// testGetEmployee();

// testAddEmployee();

// testUpdateEmployee();

}

**private** **static** **void** testGetDepartment() {

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

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

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

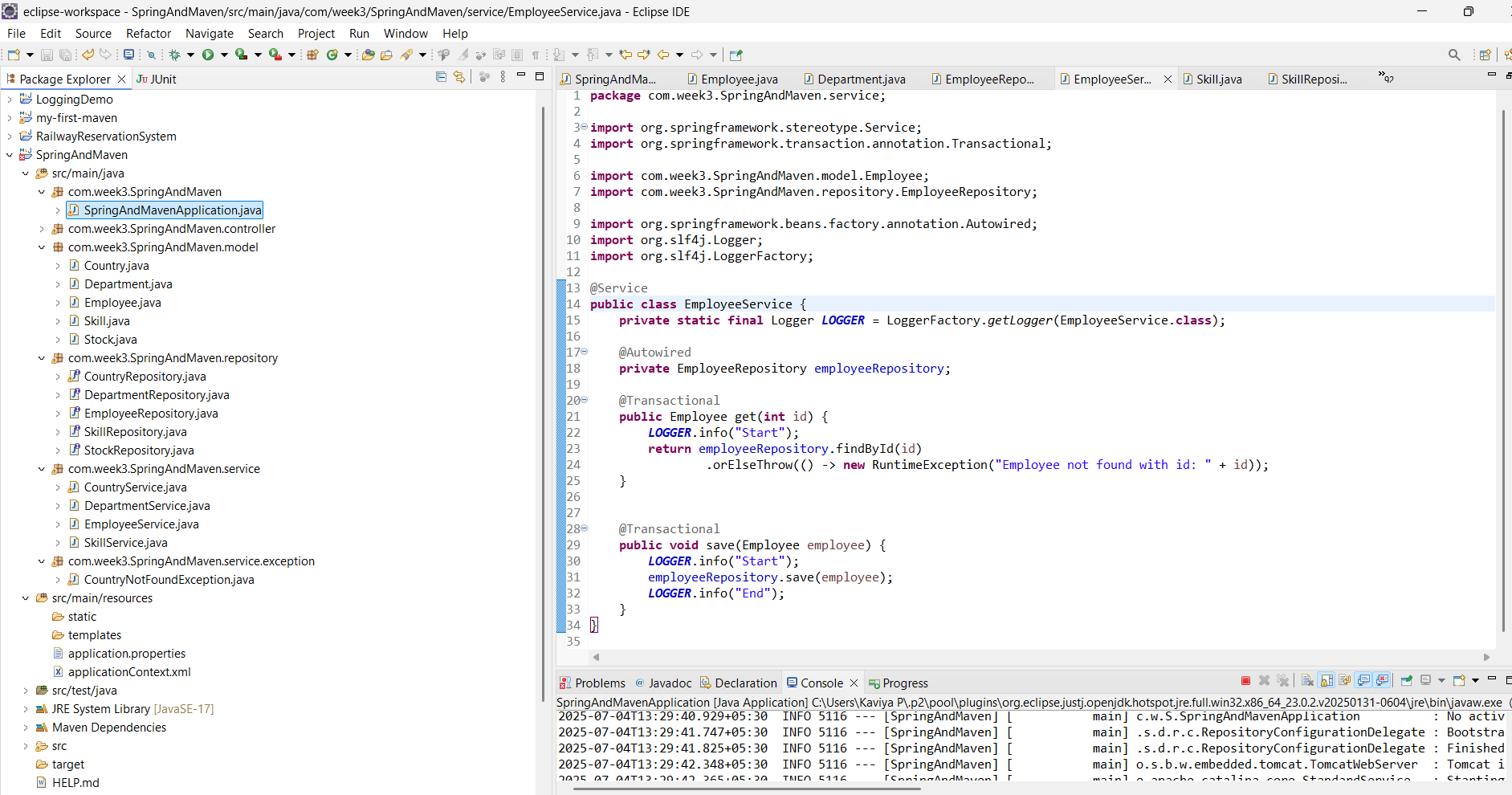
***LOGGER***.debug("Employees: {}", department.getEmployeeList());

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

}

}

**OUTPUT**

****