**Spring Data JPA with Spring Boot, Hibernate**

**Spring Data JPA - Quick Example**

**Code:**

**<!--pom.xml-->**

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

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.5.3</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.cognizant</groupId>

<artifactId>orm-learn</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>orm-learn</name>

<description>Demo project for Spring Data JPA and Hibernate</description>

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

<java.version>21</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>jakarta.persistence</groupId>

<artifactId>jakarta.persistence-api</artifactId>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**#application.properties**

spring.application.name=orm-learn

# Spring Framework and application log

logging.level.org.springframework=info

logging.level.com.cognizant=debug

# Hibernate logs for displaying executed SQL, input and output

logging.level.org.hibernate.SQL=trace

logging.level.org.hibernate.type.descriptor.sql=trace

# Log pattern

logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %-20.20thread %5p %-25.25logger{25} %25M %4L %m%n

# Database 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=examly

# Hibernate configuration

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

@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 + "]";

}

}

**//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> {

}

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

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();

}

}

**//OrmLearnApplication.java**

package com.cognizant.orm\_learn;

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;

import java.util.List;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@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);

LOGGER.info("Inside main");

countryService = context.getBean(CountryService.class);

testGetAllCountries();

}

private static void testGetAllCountries() {

LOGGER.info("Start");

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

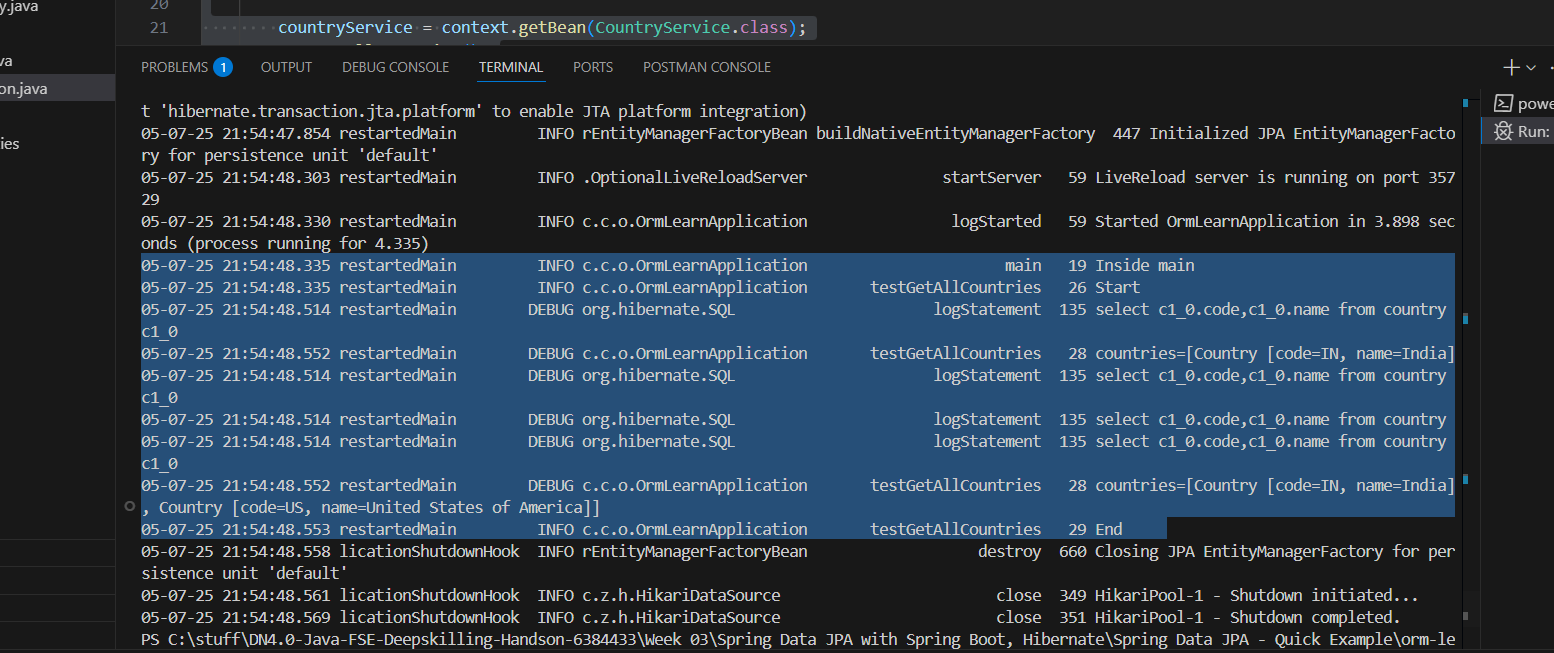
LOGGER.debug("countries={}", countries);

LOGGER.info("End");

}

}

**Output:**



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

**Code:**

**#application.properties**

spring.application.name=difference

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

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

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

spring.jpa.show-sql=false

spring.jpa.properties.hibernate.format\_sql=true

spring.h2.console.enabled=true

spring.h2.console.path=/h2-console

logging.level.org.hibernate.SQL=DEBUG

logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE

logging.level.com.cognizant.difference=DEBUG

**//Employee.java**

package com.cognizant.difference.model;

import jakarta.persistence.\*;

import java.time.LocalDate;

@Entity

@Table(name = "employees")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Integer id;

@Column(name = "first\_name", nullable = false)

private String firstName;

@Column(name = "last\_name", nullable = false)

private String lastName;

@Column(name = "email", unique = true)

private String email;

@Column(name = "salary")

private Double salary;

@Column(name = "hire\_date")

private LocalDate hireDate;

public Employee() {}

public Employee(String firstName, String lastName, String email, Double salary) {

this.firstName = firstName;

this.lastName = lastName;

this.email = email;

this.salary = salary;

this.hireDate = LocalDate.now();

}

public Integer getId() { return id; }

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

public String getFirstName() { return firstName; }

public void setFirstName(String firstName) { this.firstName = firstName; }

public String getLastName() { return lastName; }

public void setLastName(String lastName) { this.lastName = lastName; }

public String getEmail() { return email; }

public void setEmail(String email) { this.email = email; }

public Double getSalary() { return salary; }

public void setSalary(Double salary) { this.salary = salary; }

public LocalDate getHireDate() { return hireDate; }

public void setHireDate(LocalDate hireDate) { this.hireDate = hireDate; }

@Override

public String toString() {

return "Employee{" +

"id=" + id +

", firstName='" + firstName + '\'' +

", lastName='" + lastName + '\'' +

", email='" + email + '\'' +

", salary=" + salary +

", hireDate=" + hireDate +

'}';

}

}

**//SpringDataConfig.java**

package com.cognizant.difference.config;

import org.springframework.boot.autoconfigure.domain.EntityScan;

import org.springframework.context.annotation.Configuration;

import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

import org.springframework.transaction.annotation.EnableTransactionManagement;

@Configuration

@EnableJpaRepositories(basePackages = "com.cognizant.difference.springdata")

@EntityScan(basePackages = "com.cognizant.difference.model")

@EnableTransactionManagement

public class SpringDataConfig {

}

**//HibernateEmployeeDAO.java**

package com.cognizant.difference.hibernate;

import com.cognizant.difference.model.Employee;

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

import jakarta.persistence.\*;

import java.util.List;

@Repository

public class HibernateEmployeeDAO {

@PersistenceContext

private EntityManager entityManager;

@Transactional

public Integer addEmployee(Employee employee) {

try {

entityManager.persist(employee);

entityManager.flush();

System.out.println("Hibernate: Employee saved with ID: " + employee.getId());

return employee.getId();

} catch (Exception e) {

System.err.println("Hibernate Error: " + e.getMessage());

throw e;

}

}

@Transactional(readOnly = true)

public List<Employee> listEmployees() {

try {

TypedQuery<Employee> query = entityManager.createQuery("FROM Employee", Employee.class);

List<Employee> employees = query.getResultList();

System.out.println("Hibernate: Retrieved " + employees.size() + " employees");

return employees;

} catch (Exception e) {

System.err.println("Hibernate Error: " + e.getMessage());

throw e;

}

}

}

**//EmployeeRepository.java**

package com.cognizant.difference.springdata;

import com.cognizant.difference.model.Employee;

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

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

import org.springframework.data.repository.query.Param;

import org.springframework.stereotype.Repository;

import java.util.List;

@Repository

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**//EmployeeService.java**

package com.cognizant.difference.springdata;

import com.cognizant.difference.model.Employee;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import java.util.List;

@Service

@Transactional

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

public Employee addEmployee(Employee employee) {

Employee savedEmployee = employeeRepository.save(employee);

System.out.println("Spring Data JPA: Employee saved with ID: " + savedEmployee.getId());

return savedEmployee;

}

public List<Employee> getAllEmployees() {

return employeeRepository.findAll();

}

}

**//DifferenceApplication.java**

package com.cognizant.difference;

import com.cognizant.difference.model.Employee;

import com.cognizant.difference.hibernate.HibernateEmployeeDAO;

import com.cognizant.difference.springdata.EmployeeService;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import java.util.List;

@SpringBootApplication

public class DifferenceApplication implements CommandLineRunner {

@Autowired

private EmployeeService employeeService;

@Autowired

private HibernateEmployeeDAO hibernateDAO;

public static void main(String[] args) {

SpringApplication.run(DifferenceApplication.class, args);

}

@Override

public void run(String... args) throws Exception {

System.out.println("\n" + "=".repeat(80));

System.out.println("DEMONSTRATING DIFFERENCES BETWEEN JPA, HIBERNATE, AND SPRING DATA JPA");

System.out.println("=".repeat(80) + "\n");

demonstrateHibernate();

System.out.println("\n" + "=".repeat(80) + "\n");

demonstrateSpringDataJPA();

System.out.println("\n" + "=".repeat(80));

}

private void demonstrateHibernate() {

System.out.println("HIBERNATE DIRECT USAGE - Manual EntityManager Operations");

System.out.println("-".repeat(70));

try {

// Create employees using Hibernate

Employee emp1 = new Employee("John", "Hibernate", "john.hibernate@cognizant.com", 50000.0);

Employee emp2 = new Employee("Jane", "Sessions", "jane.sessions@cognizant.com", 60000.0);

Employee emp3 = new Employee("Mike", "Transaction", "mike.transaction@cognizant.com", 55000.0);

System.out.println("Creating employees using Hibernate EntityManager...");

Integer id1 = hibernateDAO.addEmployee(emp1);

Integer id2 = hibernateDAO.addEmployee(emp2);

Integer id3 = hibernateDAO.addEmployee(emp3);

// List all employees

System.out.println("\nListing all employees (Hibernate):");

List<Employee> employees = hibernateDAO.listEmployees();

employees.forEach(emp -> System.out.println(" " + emp));

} catch (Exception e) {

System.err.println("Error in Hibernate demonstration: " + e.getMessage());

e.printStackTrace();

}

}

private void demonstrateSpringDataJPA() {

System.out.println("SPRING DATA JPA USAGE - Automatic Repository Management");

System.out.println("-".repeat(75));

try {

// Create employees using Spring Data JPA

Employee emp1 = new Employee("Alice", "SpringData", "alice.spring@cognizant.com", 70000.0);

Employee emp2 = new Employee("Bob", "Repository", "bob.repo@cognizant.com", 80000.0);

Employee emp3 = new Employee("Charlie", "JPA", "charlie.jpa@cognizant.com", 90000.0);

System.out.println("Creating employees using Spring Data JPA...");

Employee savedEmp1 = employeeService.addEmployee(emp1);

Employee savedEmp2 = employeeService.addEmployee(emp2);

Employee savedEmp3 = employeeService.addEmployee(emp3);

// List all employees

System.out.println("\nListing all employees (Spring Data JPA):");

employeeService.getAllEmployees().forEach(emp -> System.out.println(" " + emp));

} catch (Exception e) {

System.err.println("Error in Spring Data JPA demonstration: " + e.getMessage());

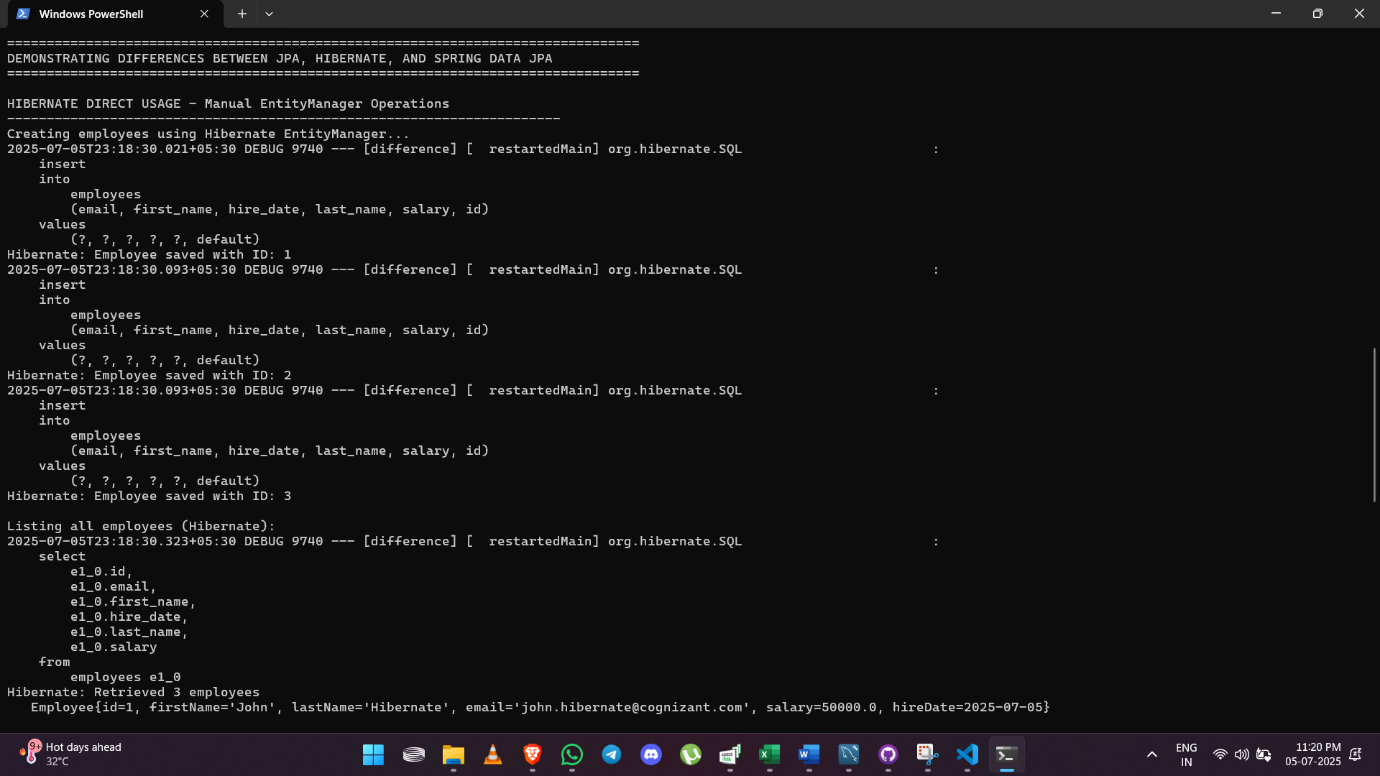
e.printStackTrace();

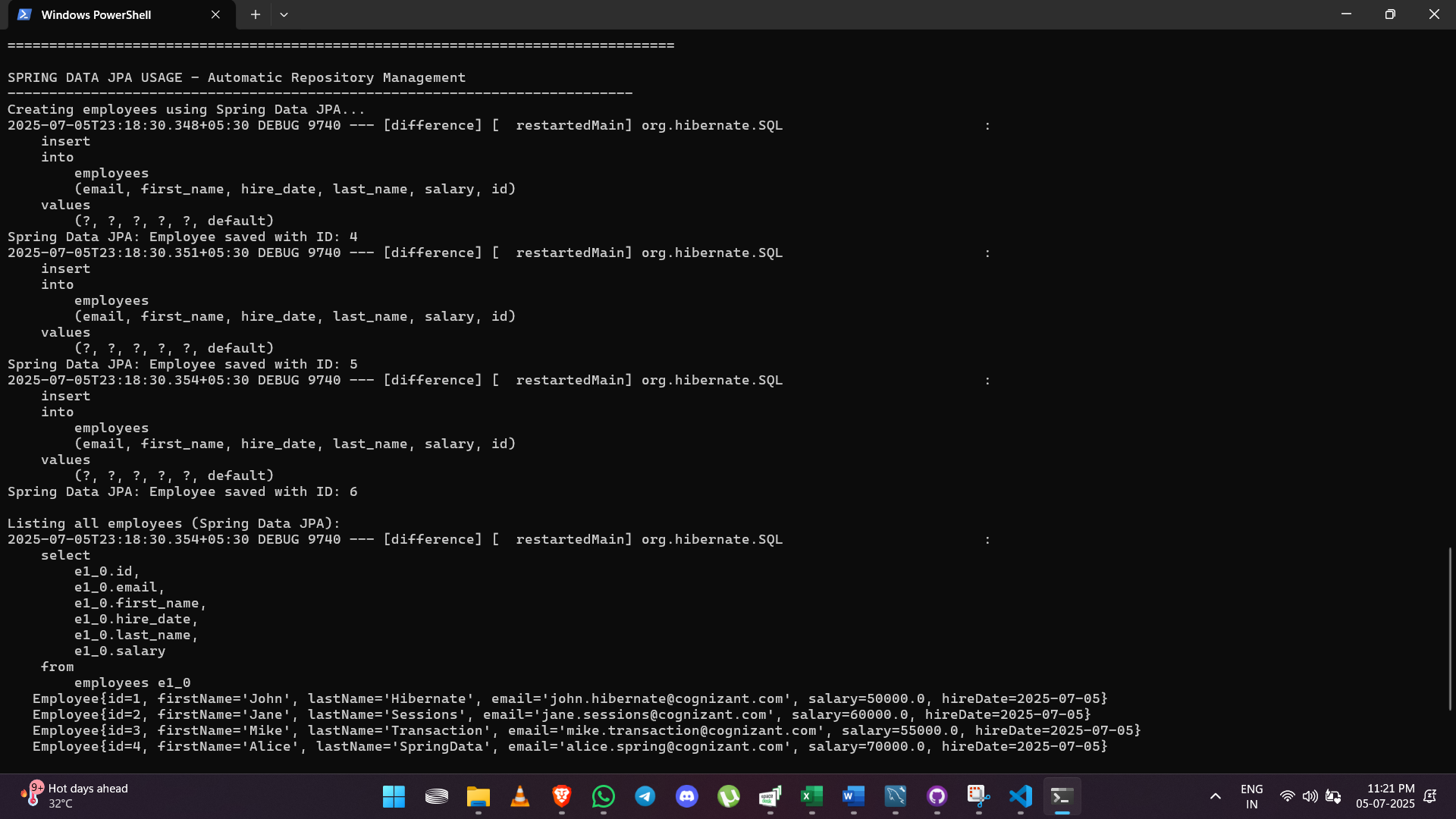
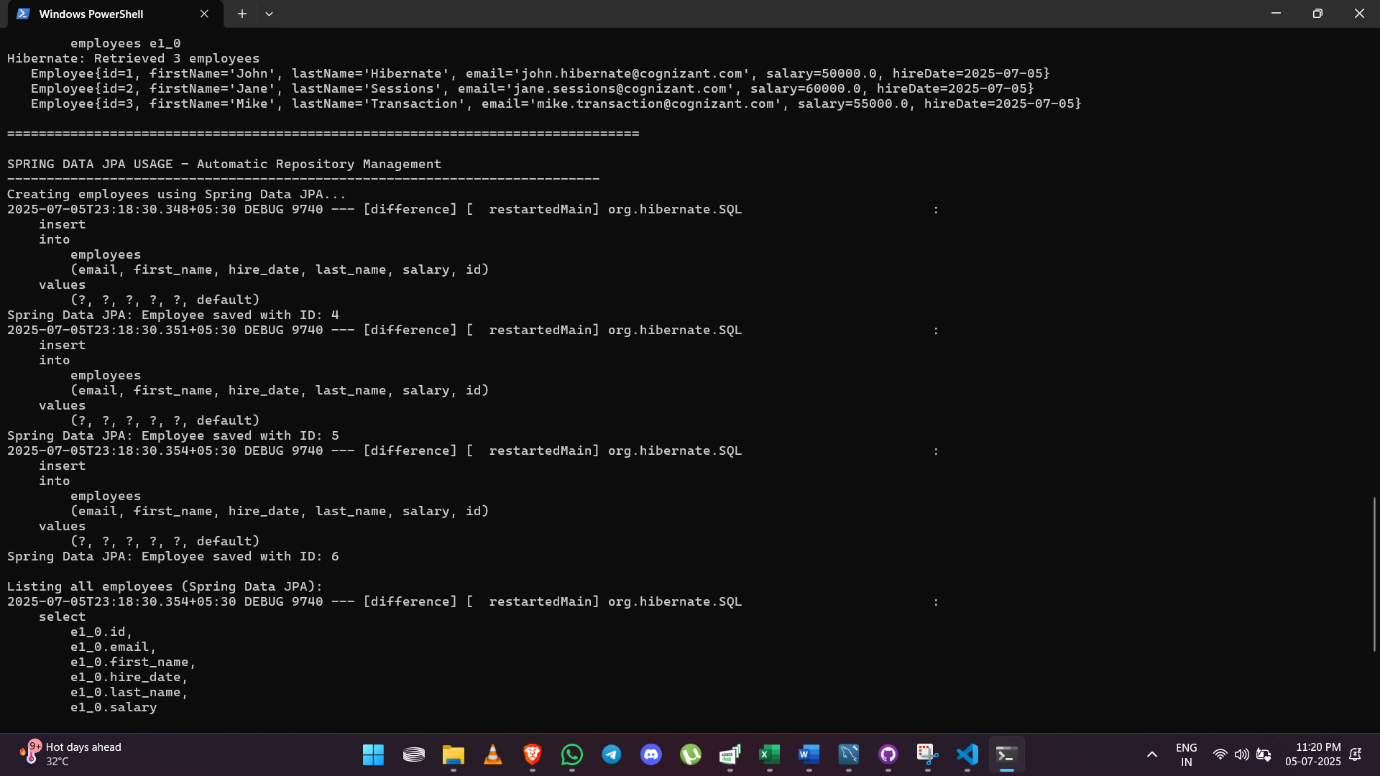
}

}

}

**Output:**





**Find a country based on country code**

**Code:**

**//CountryNotFoundException.java**

package com.cognizant.orm\_learn.service.exception;

public class CountryNotFoundException extends Exception {

public CountryNotFoundException(String message) {

super(message);

}

}

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

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

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

import com.cognizant.orm\_learn.service.exception.CountryNotFoundException;

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

@Transactional

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

@Transactional

public Country findCountryByCode(String countryCode) throws CountryNotFoundException {

Optional<Country> result = countryRepository.findById(countryCode);

if (!result.isPresent()) {

throw new CountryNotFoundException("Country not found with code: " + countryCode);

}

return result.get();

}

}

**//OrmLearnApplication.java**

package com.cognizant.orm\_learn;

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;

import com.cognizant.orm\_learn.service.exception.CountryNotFoundException;

import java.util.List;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@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);

LOGGER.info("Inside main");

countryService = context.getBean(CountryService.class);

testGetAllCountries();

testFindCountryByCode();

}

private static void testGetAllCountries() {

LOGGER.info("Start");

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

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

LOGGER.info("End");

}

private static void testFindCountryByCode() {

LOGGER.info("Start");

try {

Country country = countryService.findCountryByCode("IN");

LOGGER.debug("Country: {}", country);

} catch (CountryNotFoundException e) {

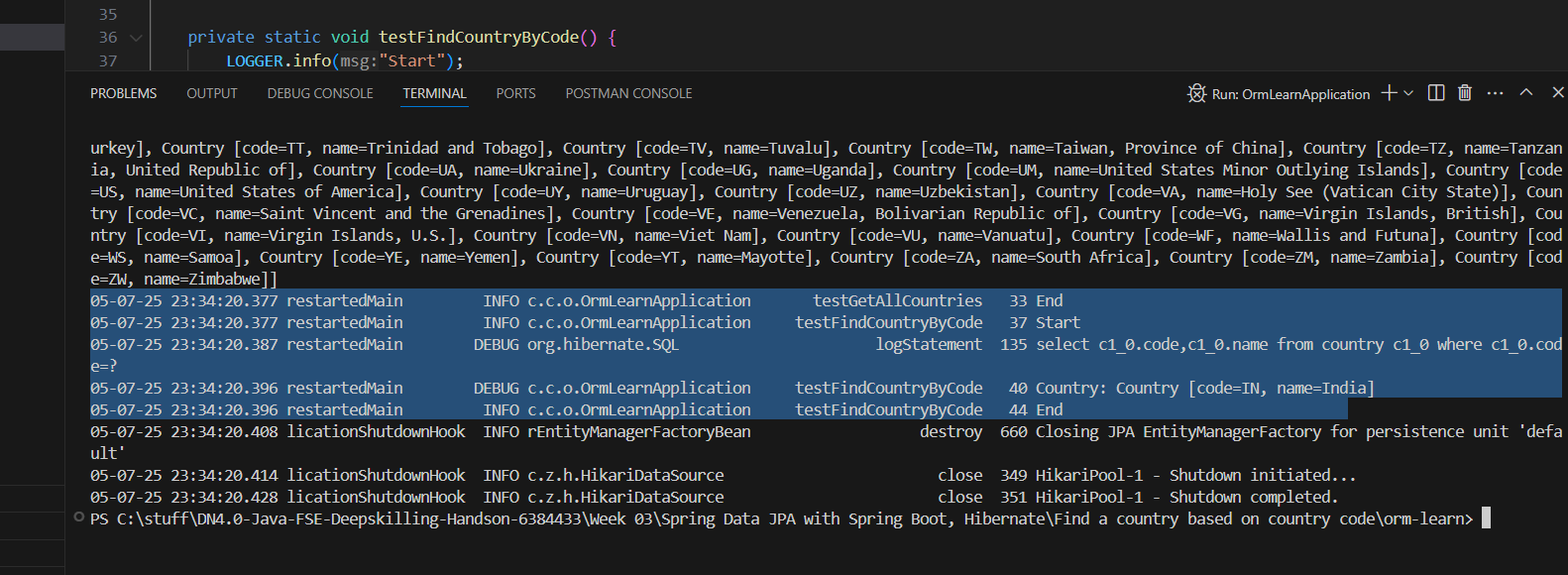
LOGGER.error("Exception: {}", e.getMessage());

}

LOGGER.info("End");

}

}

**Output:**

**Add a new country**

**Code:**

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

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

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

import com.cognizant.orm\_learn.service.exception.CountryNotFoundException;

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

@Transactional

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

@Transactional

public Country findCountryByCode(String countryCode) throws CountryNotFoundException {

Optional<Country> result = countryRepository.findById(countryCode);

if (!result.isPresent()) {

throw new CountryNotFoundException("Country not found with code: " + countryCode);

}

return result.get();

}

@Transactional

public void addCountry(Country country) {

countryRepository.save(country);

}

}

**//OrmLearnApplication.java**

package com.cognizant.orm\_learn;

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;

import com.cognizant.orm\_learn.service.exception.CountryNotFoundException;

import java.util.List;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@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);

LOGGER.info("Inside main");

countryService = context.getBean(CountryService.class);

testGetAllCountries();

testFindCountryByCode();

testAddCountry();

}

private static void testGetAllCountries() {

LOGGER.info("Start");

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

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

LOGGER.info("End");

}

private static void testFindCountryByCode() {

LOGGER.info("Start");

try {

Country country = countryService.findCountryByCode("IN");

LOGGER.debug("Country: {}", country);

} catch (CountryNotFoundException e) {

LOGGER.error("Exception: {}", e.getMessage());

}

LOGGER.info("End");

}

private static void testAddCountry() {

LOGGER.info("Start");

Country newCountry = new Country();

newCountry.setCode("JP");

newCountry.setName("Japan");

countryService.addCountry(newCountry);

try {

Country retrievedCountry = countryService.findCountryByCode("JP");

LOGGER.debug("Added Country: {}", retrievedCountry);

} catch (CountryNotFoundException e) {

LOGGER.error("Country not found after insert: {}", e.getMessage());

}

LOGGER.info("End");

}

}

**Output:**

