**Spring Data JPA - Quick Example**

**Country.java**

package com.cognizant.orm\_learn.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 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.repositry;

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

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

import org.springframework.stereotype.Repository;

@Repository

public interface CountryRepositry extends JpaRepository<Country,String> {

}

**CountryService.java**

package com.cognizant.orm\_learn.service;

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

import com.cognizant.orm\_learn.repositry.CountryRepositry;

import jakarta.transaction.Transactional;

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

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class CountryService {

@Autowired

private CountryRepositry countryRepository;

@Transactional

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

}

**ormlearnApplication.java**

package com.cognizant.orm\_learn;

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

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

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.List;

@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);

testGetAllCountries();

}

private static void testGetAllCountries() {

LOGGER.info("Start");

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

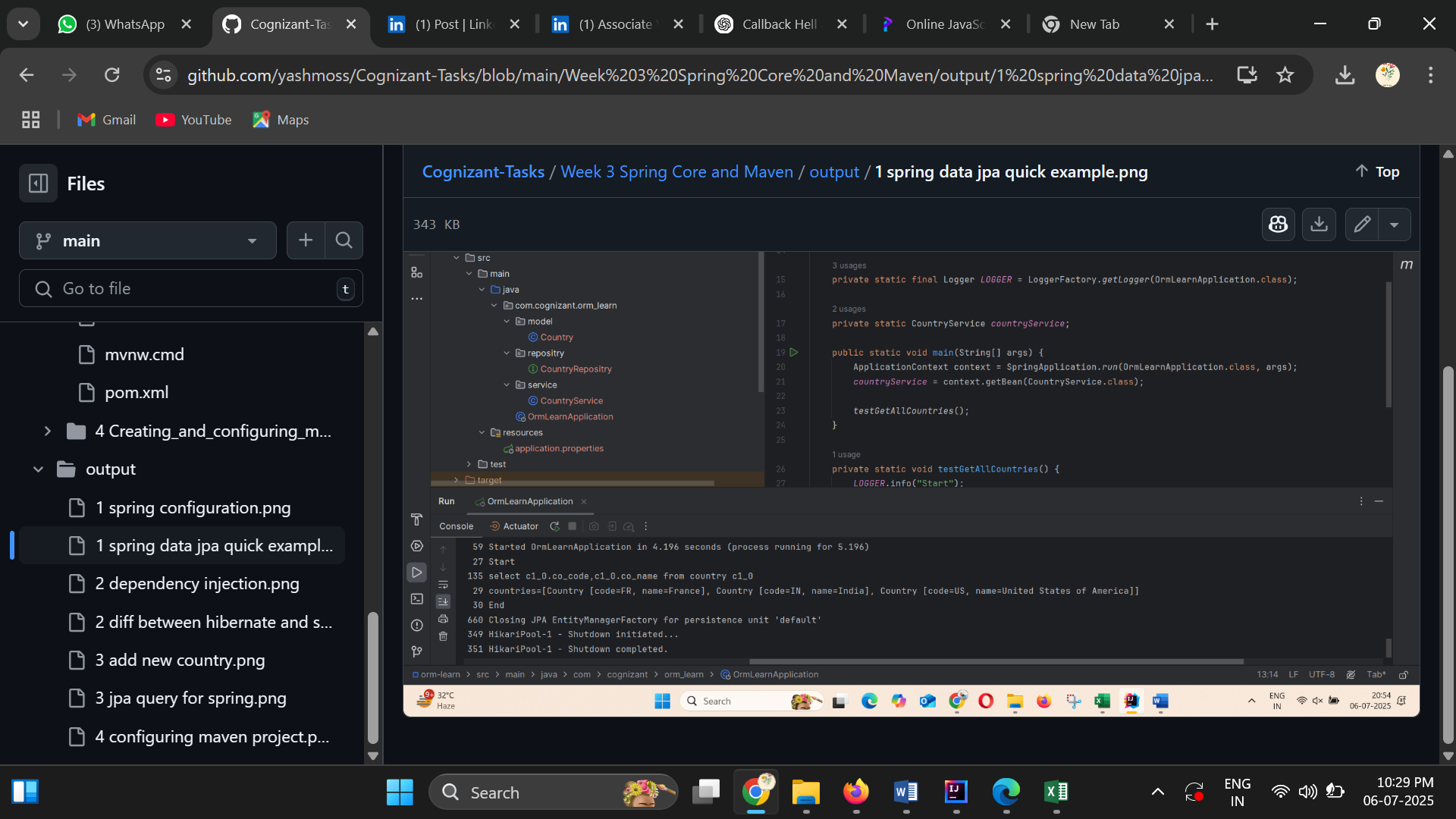
LOGGER.debug("countries={}", countries);

LOGGER.info("End");

}

}

Output:



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

**Employee.java**

package com.cognizant.hibernate\_example;

import javax.persistence.\*;

@Entity

@Table(name = "employee")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private String department;

public void setName(String name) {

this.name=name;

}

public void setDepartment(String department) {

this.department=department;

}

}  
**HibrenateJpaApplication.java**package com.cognizant.hibernate\_example;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class HibernateJpaApplication {

public static void main(String[] args) {

Employee emp = new Employee();

emp.setName("Yashwanth");

emp.setDepartment("CSE");

Session session = HibernateUtil.getSessionFactory().openSession();

Transaction tx = null;

try {

tx = session.beginTransaction();

session.save(emp);

tx.commit();

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

} catch (Exception e) {

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

e.printStackTrace();

} finally {

session.close();

}

}

} **HibernateUtil.java**package com.cognizant.hibernate\_example;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

public class HibernateUtil {

private static final SessionFactory factory;

static {

try {

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

} catch (Throwable ex) {

throw new ExceptionInInitializerError(ex);

}

}

public static SessionFactory getSessionFactory() {

return factory;

}

}

**HibernateJpaApplicationTests**package com.cognizant.hibernate\_example;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

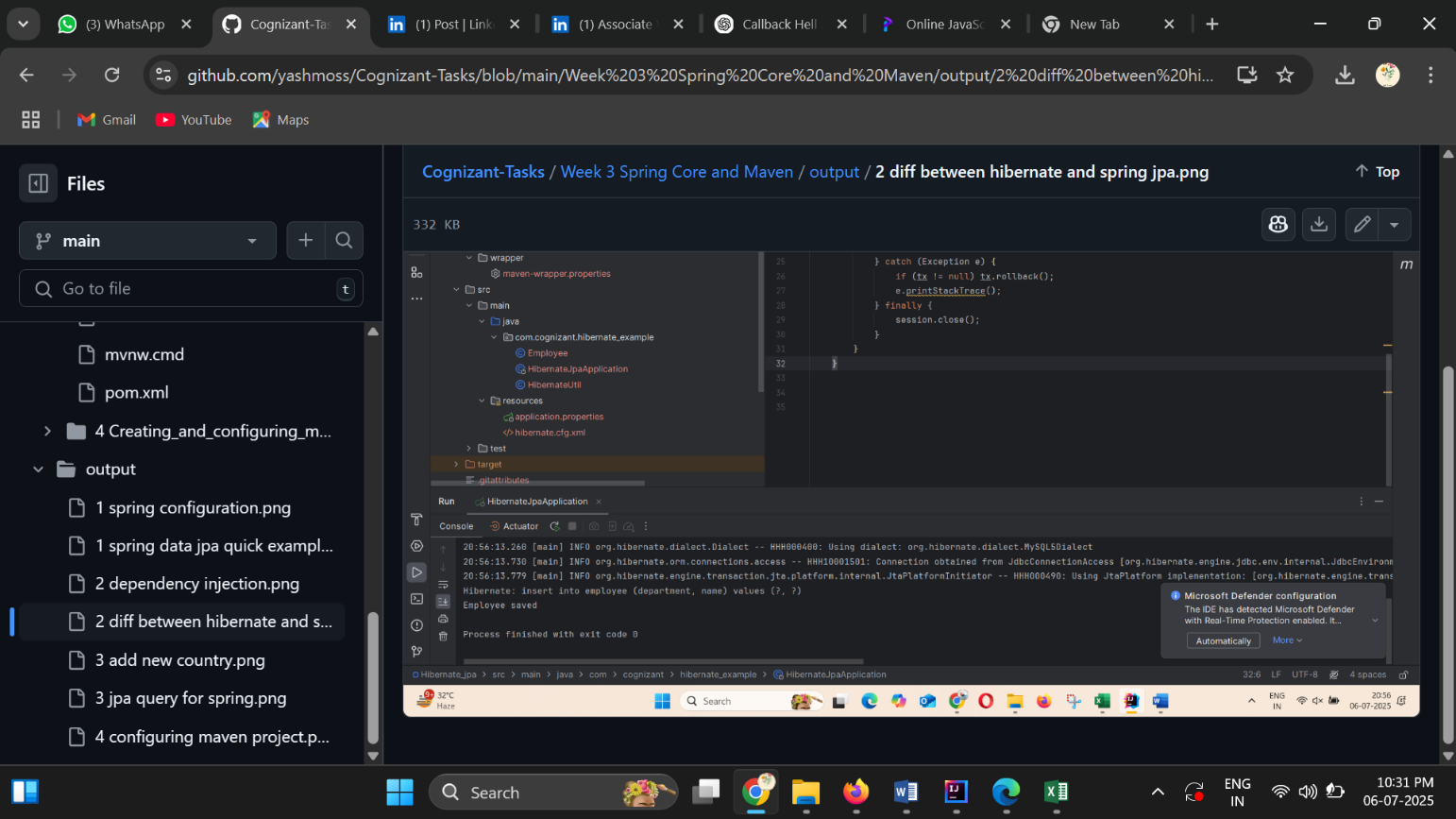
class HibernateJpaApplicationTests {

@Test

void contextLoads() {

}

Output:



**Add a new country**

**Country.java**package com.example.country.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.example.country.repository;

import com.example.country.model.Country;

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

public interface Countryrepository extends JpaRepository<Country, String> {

}

**CountryApplication.java**package com.example.country;

import com.example.country.model.Country;

import com.example.country.repository.Countryrepository;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class CountryApplication implements CommandLineRunner {

@Autowired

private Countryrepository repository;

public static void main(String[] args) {

SpringApplication.run(CountryApplication.class, args);

}

@Override

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

Country country = new Country("FR", "France");

repository.save(country);

System.out.println("New country added: " + country.getName());

}

}

**CountryApplicationTests**package com.example.country;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

class CountryApplicationTests {

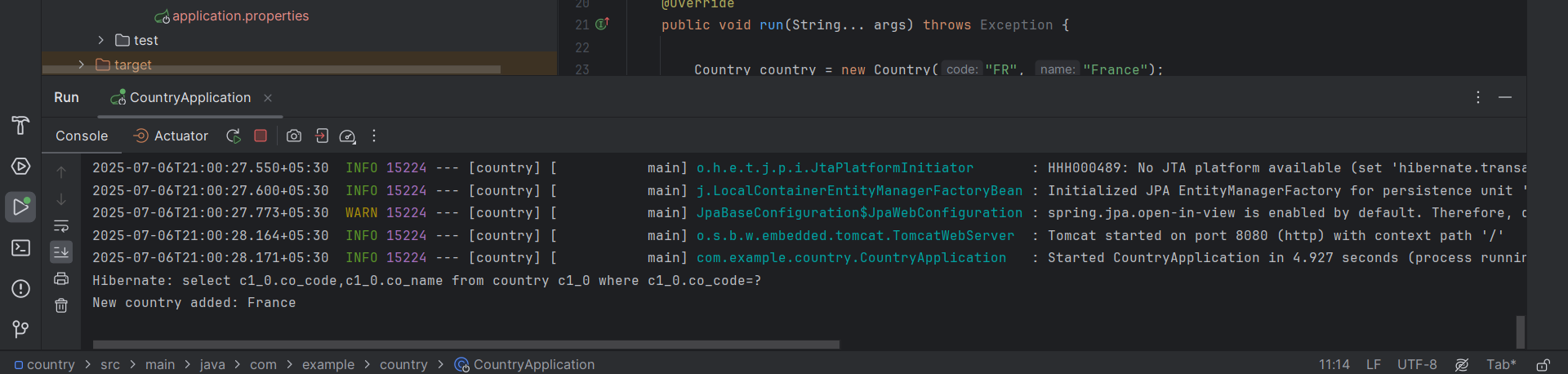
@Test

void contextLoads() {

}

}

Output:



**Demonstrate implementation of Query Methods feature of Spring Data JPA**

**Country.java**

package com.cognizant.jpa\_query\_methods.model;

import jakarta.persistence.\*;

@Entity

@Table(name="country")

public class Country {

@Id

@Column(name = "co\_code")

private String cocode;

@Column(name = "co\_name")

private String coname;

public String getCo\_code() {

return cocode;

}

public String getCo\_name() {

return coname;

}

public void setCo\_code(String co\_code) {

this.cocode = co\_code;

}

public void setCo\_name(String co\_name) {

this.coname = co\_name;

}

}

**JpaQueryMethodsApplication.java**

package com.cognizant.jpa\_query\_methods;

import com.cognizant.jpa\_query\_methods.model.Country;

import com.cognizant.jpa\_query\_methods.repository.CountryRepository;

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 JpaQueryMethodsApplication implements CommandLineRunner {

@Autowired

public CountryRepository countryRepository;

public static void main(String[] args) {

SpringApplication.run(JpaQueryMethodsApplication.class, args);

}

@Override

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

System.out.println("Countries containing N");

List<Country> result= countryRepository.findByConameContainingOrderByConameAsc("N");

result.forEach(c-> System.out.println(c.getCo\_code()+" "+c.getCo\_name()));

List<Country> result2=countryRepository.findByConameContaining("N");

result2.forEach(c-> System.out.println(c.getCo\_code()+" "+c.getCo\_name()));

}

}

Output:

