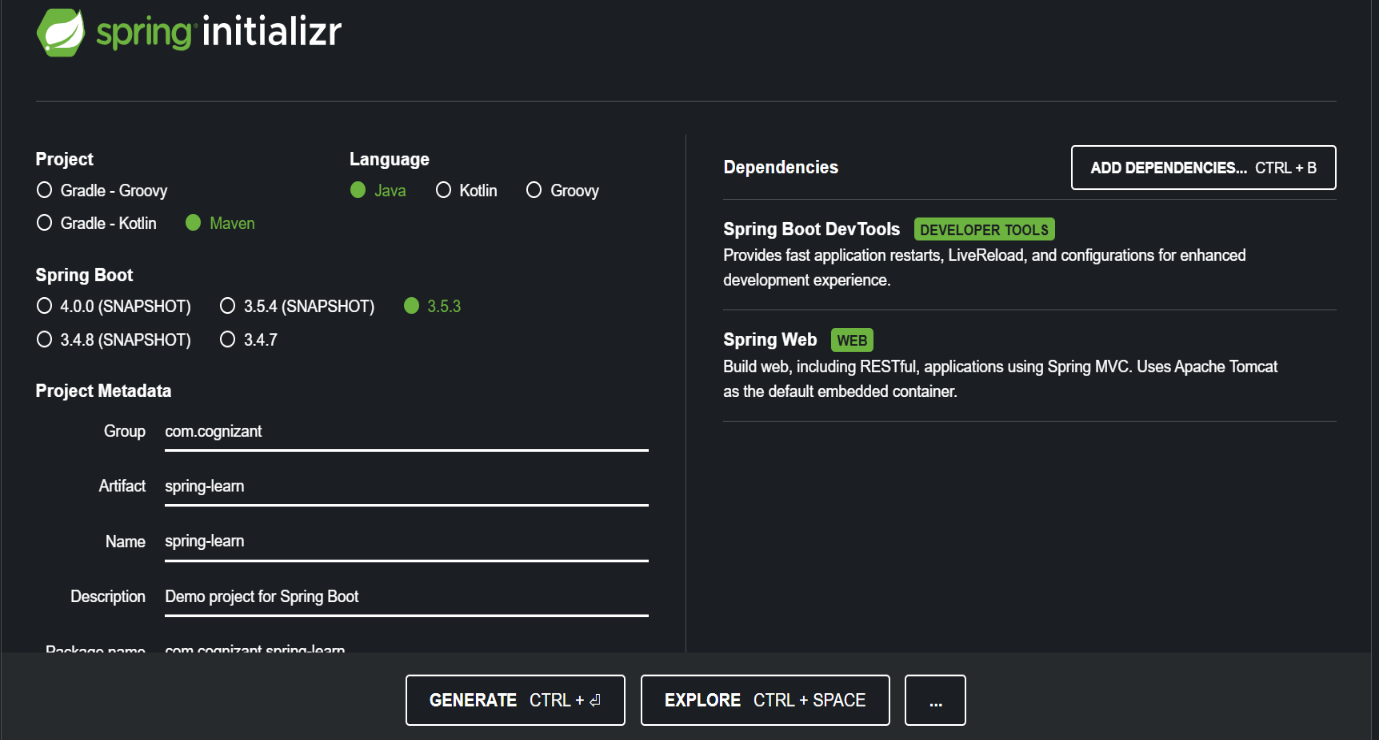
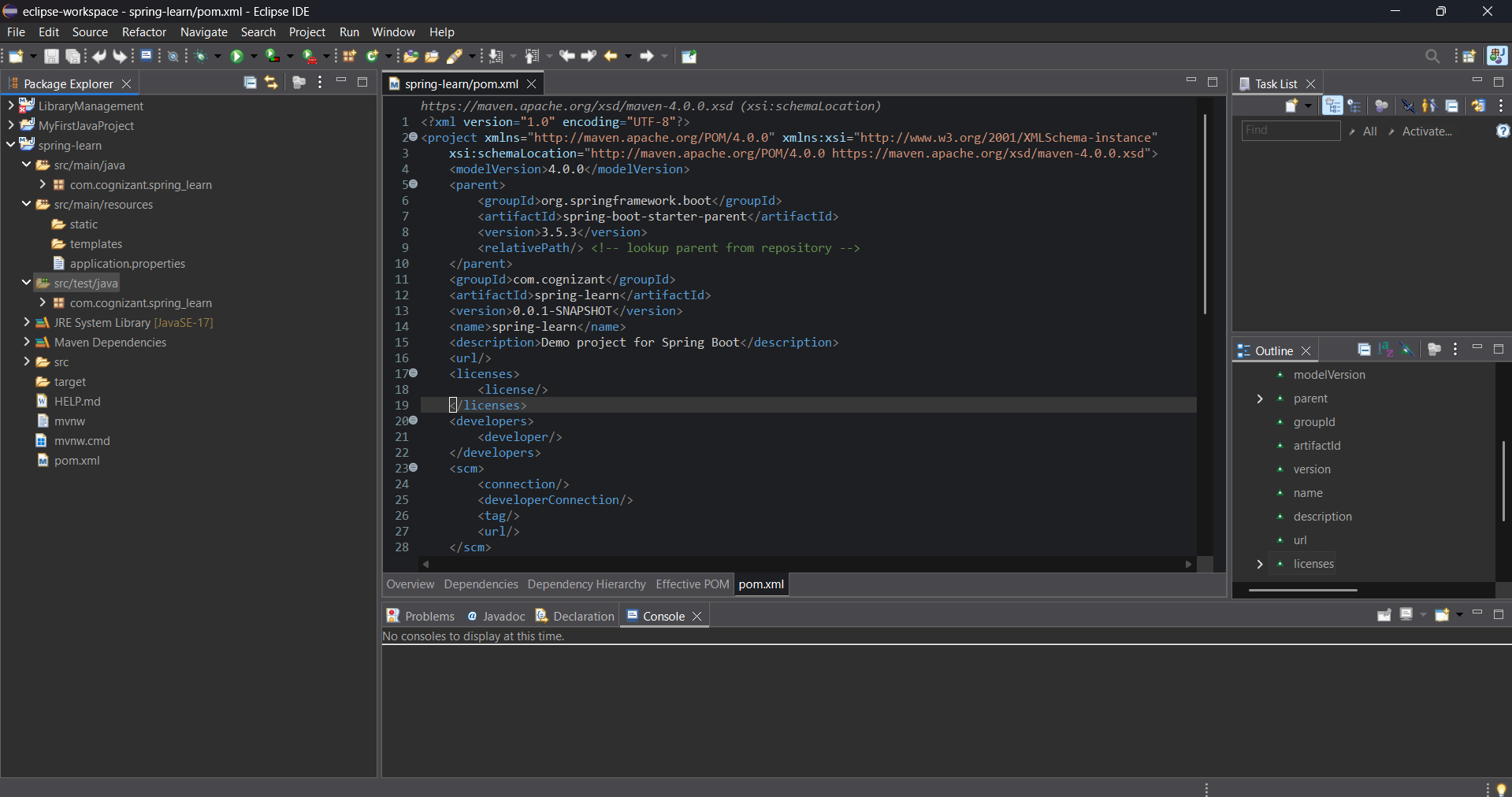
**Week – 4 Spring REST using Spring Boot**

**Hands on 1**

**Create a Spring Web Project using Maven**





**Hands on 2**

**Spring Core – Load SimpleDateFormat from Spring Configuration XML**

date-format.xml

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

https://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="dateFormat" class="java.text.SimpleDateFormat">

<constructor-arg value="dd/MM/yyyy" />

</bean>

</beans>

SpringLearnApplication.java

package com.example.springlearn;

import java.text.SimpleDateFormat;

import java.util.Date;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpringLearnApplication {

public static void main(String[] args) {

displayDate(); // Call the method

}

public static void displayDate() {

ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml")

SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);

try {

Date date = format.parse("31/12/2018");

System.out.println("Parsed Date: " + date);

} catch (Exception e) {

System.out.println("Error parsing date: " + e.getMessage());

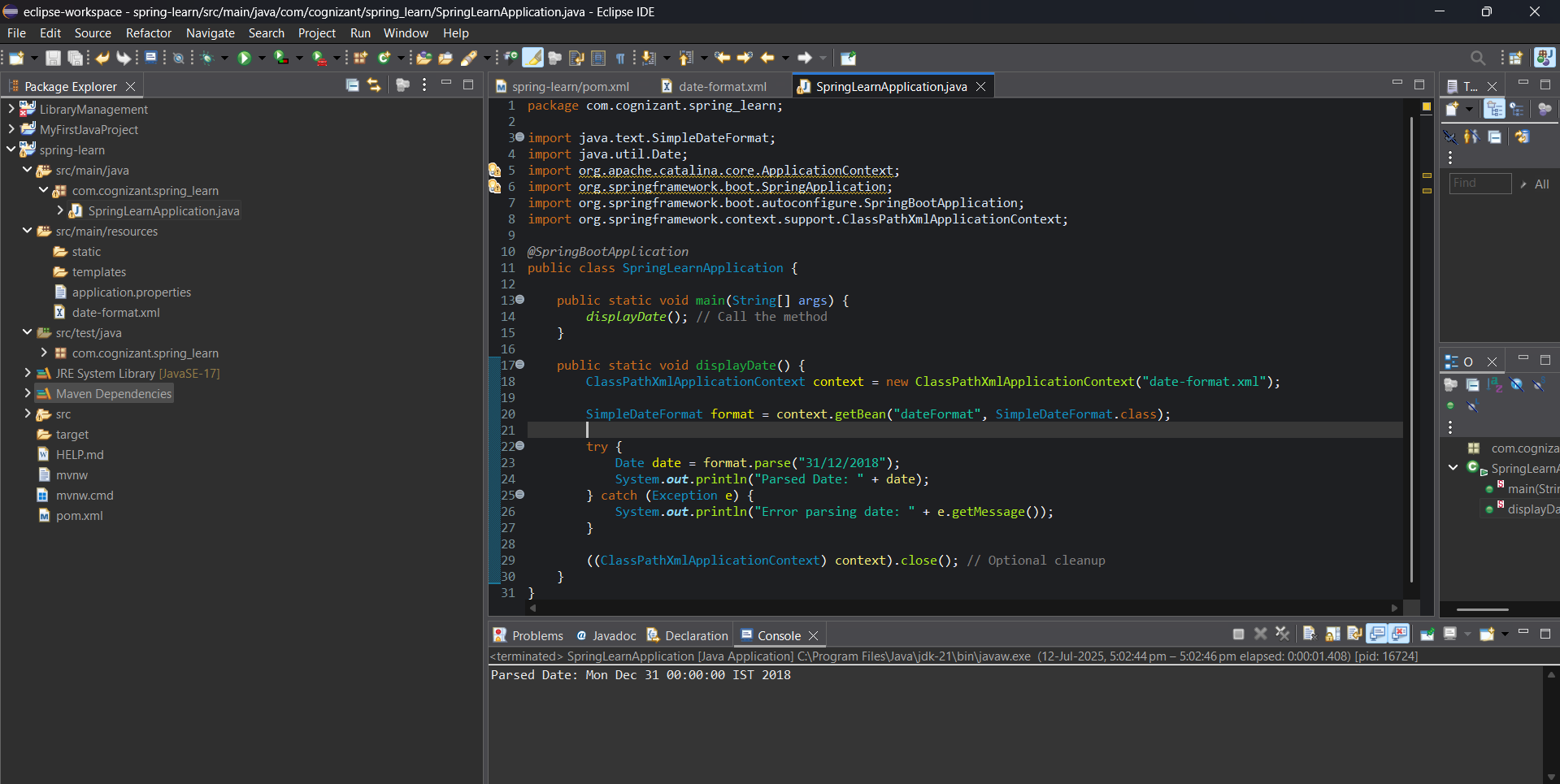
}

((ClassPathXmlApplicationContext) context).close(); // Optional cleanup

}

}

Output



**Hands on 3**

**Spring Core - Incorporate Logging**

SpringLearnApplication.java

package com.cognizant.spring\_learn;

import java.text.SimpleDateFormat;

import java.util.Date;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.support.ClassPathXmlApplicationContext;

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args); // Proper Spring Boot start

displayDate();

}

public static void displayDate() {

LOGGER.info("START");

ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");

SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);

try {

Date date = format.parse("31/12/2018");

LOGGER.debug("Parsed Date: {}", date);

} catch (Exception e) {

LOGGER.error("Error parsing date", e);

}

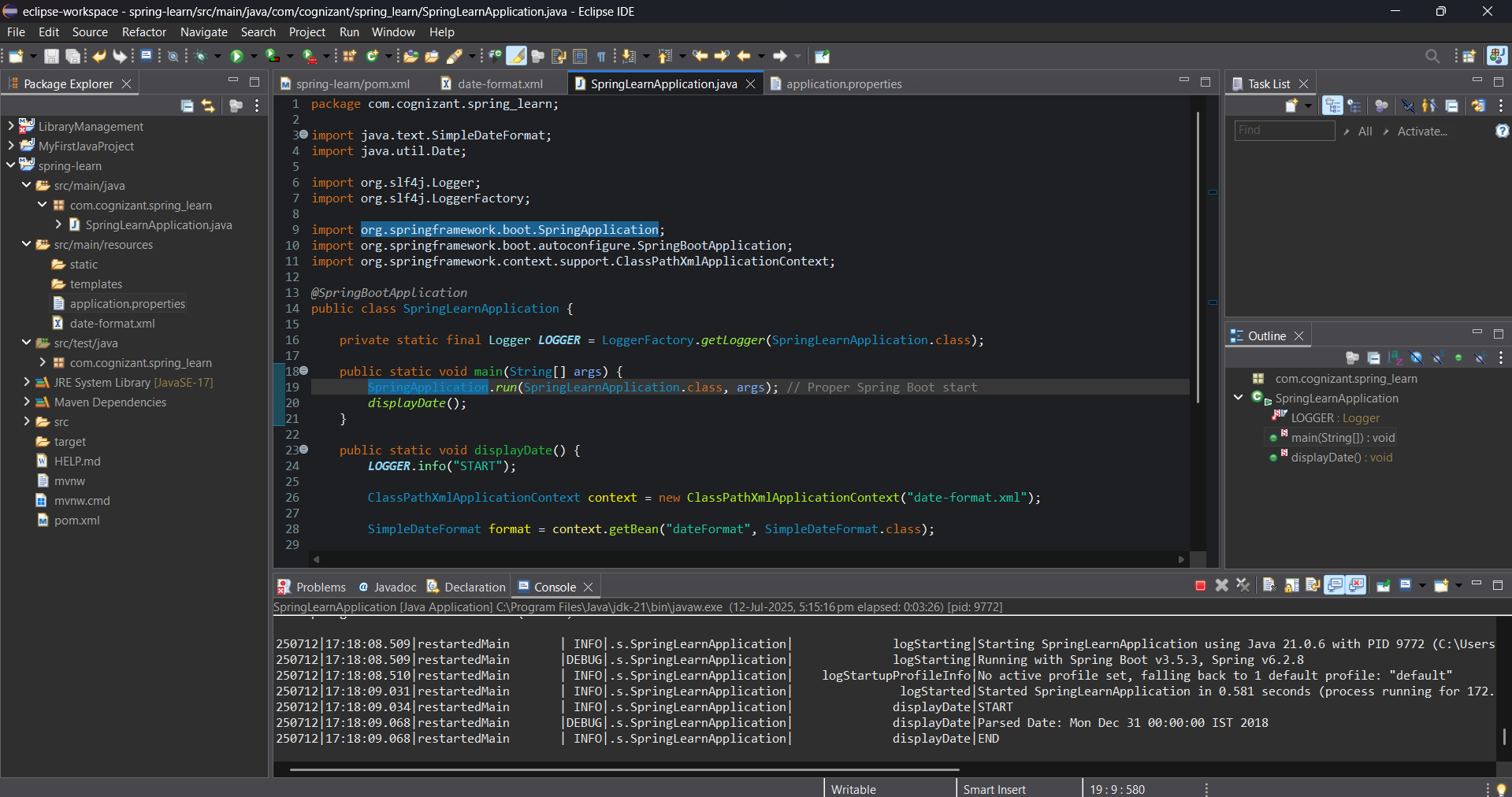
context.close();

LOGGER.info("END");

}

}

Output



**Hands on 4**

**Spring Core – Load Country from Spring Configuration XML**   
  
country.xml

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="country" class="com.cognizant.spring\_learn.Country">

<property name="code" value="IN" />

<property name="name" value="India" />

</bean>

</beans>

Country.java

package com.cognizant.spring\_learn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class Country {

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

private String code;

private String name;

public Country() {

***LOGGER***.debug("Inside Country Constructor.");

}

public String getCode() {

***LOGGER***.debug("Inside getCode()");

return code;

}

public void setCode(String code) {

***LOGGER***.debug("Inside setCode()");

this.code = code;

}

public String getName() {

***LOGGER***.debug("Inside getName()");

return name;

}

public void setName(String name) {

***LOGGER***.debug("Inside setName()");

this.name = name;

}

*@Override*

public String toString() {

return "Country [code=" + code + ", name=" + name + "]";

}

}

SpringLearnApplication.java

package com.cognizant.spring\_learn;

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 org.springframework.context.support.ClassPathXmlApplicationContext;

*@SpringBootApplication*

public class SpringLearnApplication {

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

public static void main(String[] args) {

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

*displayCountry*();

}

public static void displayCountry() {

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

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

Country country = context.getBean("country", Country.class);

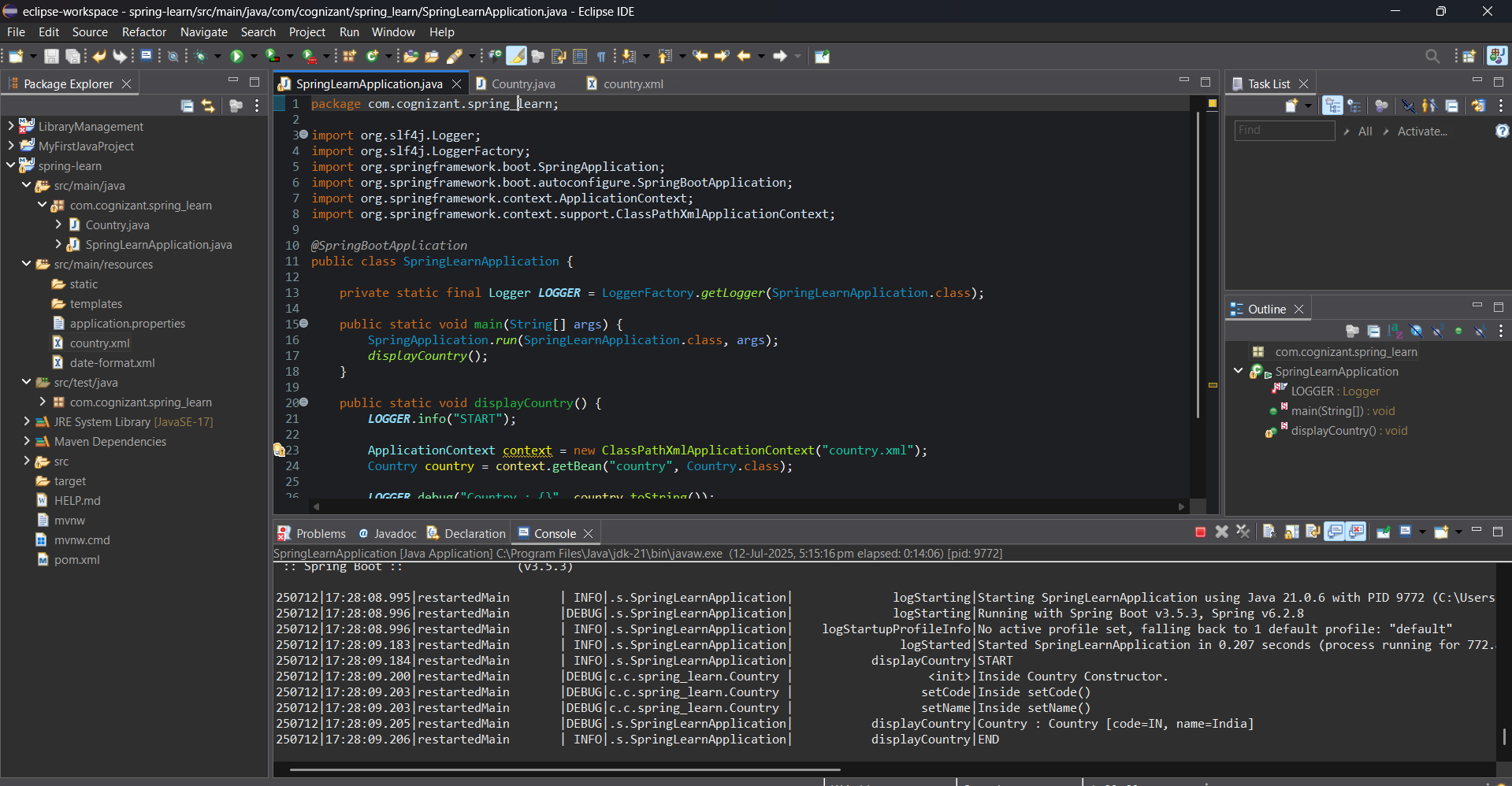
***LOGGER***.debug("Country : {}", country.toString());

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

}

}

Output



**Hands on 5**

**Spring Core – Demonstration of Singleton Scope and Prototype Scope**

SpringLearnApplication.java

package com.cognizant.spring\_learn;

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 org.springframework.context.support.ClassPathXmlApplicationContext;

*@SpringBootApplication*

public class SpringLearnApplication {

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

public static void main(String[] args) {

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

*displayCountry*();

}

public static void displayCountry() {

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

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

Country country = context.getBean("country", Country.class);

Country anotherCountry = context.getBean("country", Country.class); // key line to test scope

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

***LOGGER***.debug("Another Country : {}", anotherCountry.toString());

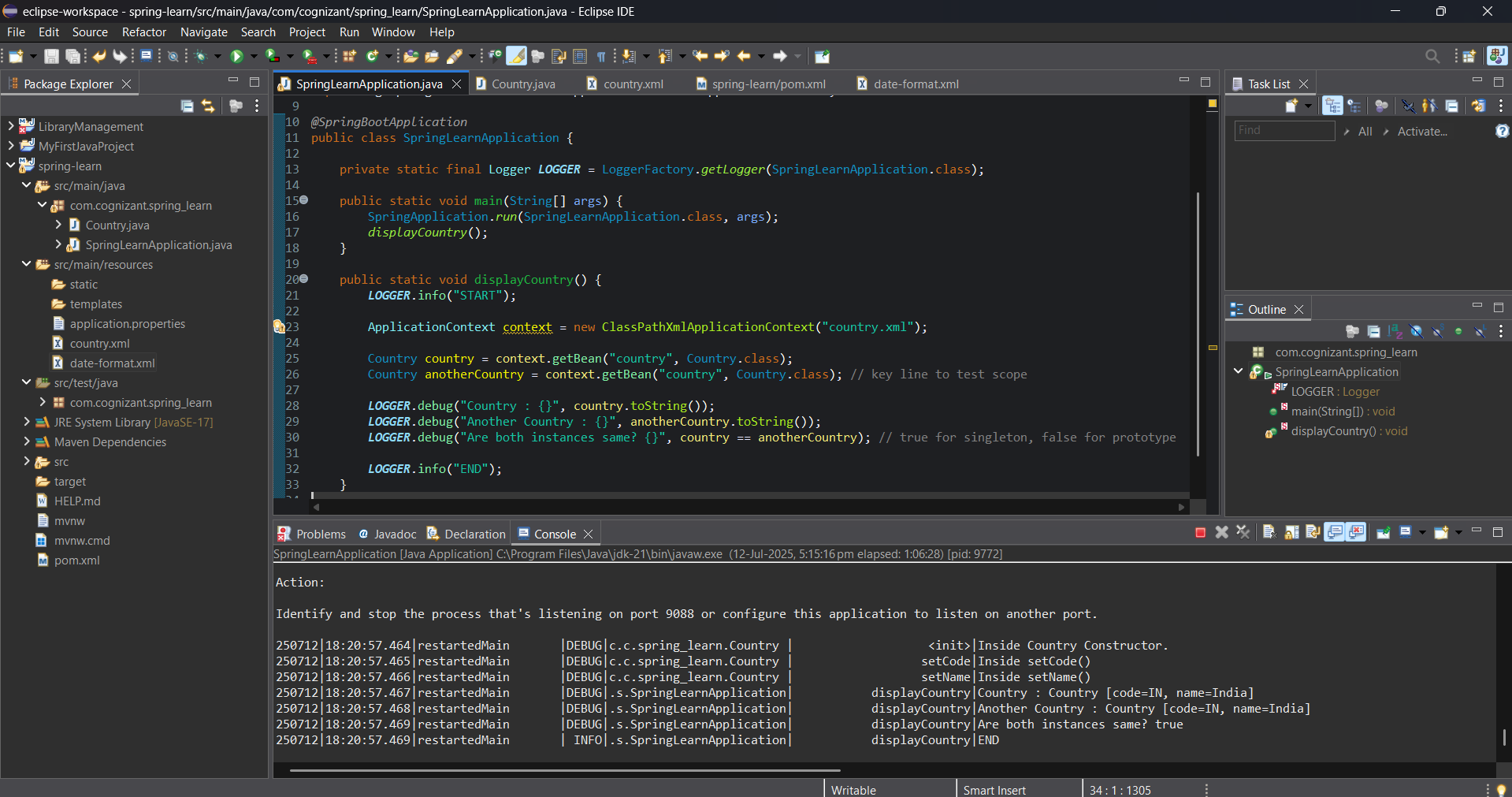
***LOGGER***.debug("Are both instances same? {}", country == anotherCountry); // true for singleton, false for prototype

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

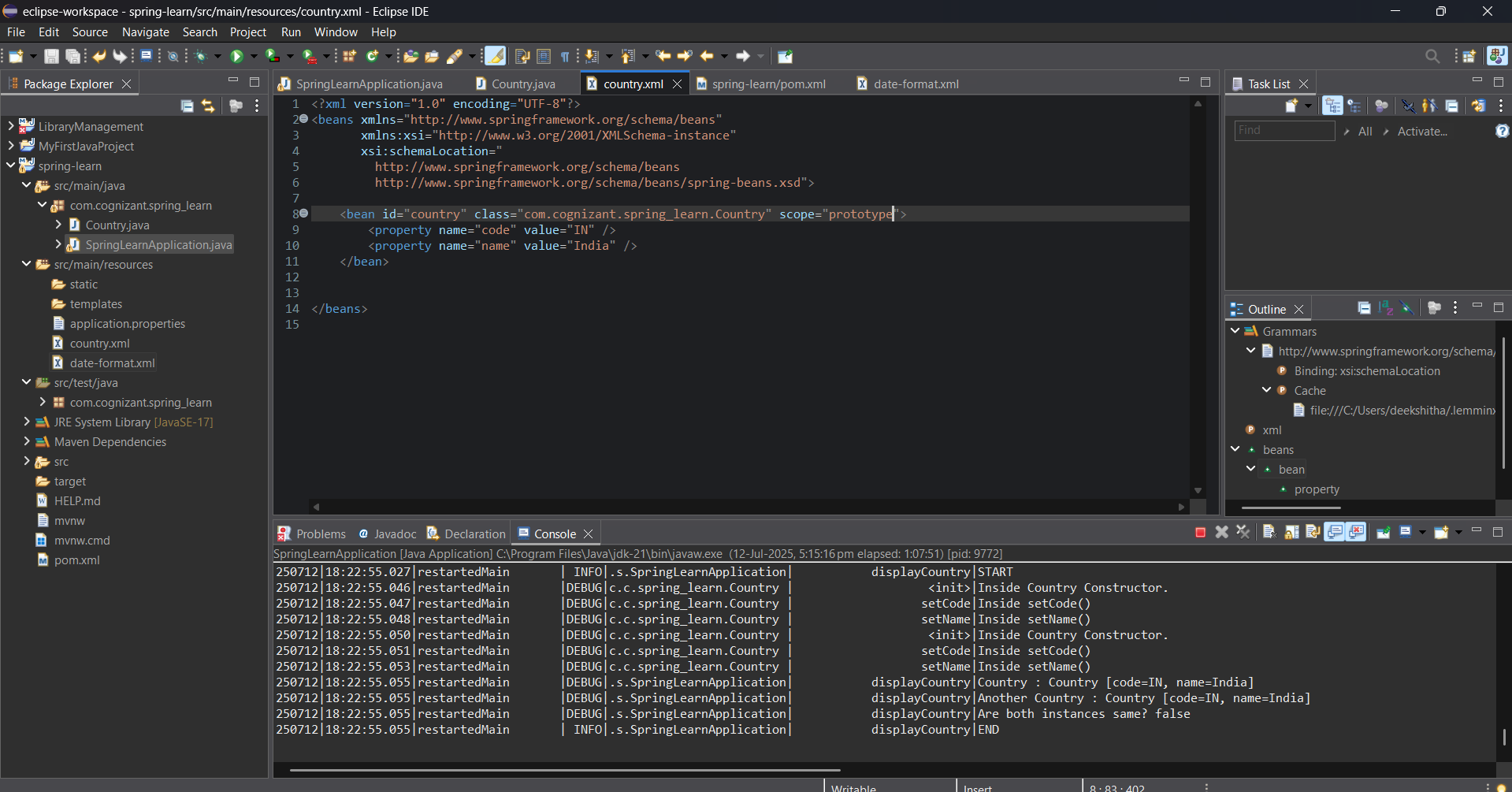
}

}

Output for singleton Scope



Output for Prototype Scope



**Hands on 6**

**Spring Core – Load list of countries from Spring Configuration XML**

country.xml

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<!-- Individual Country Beans -->

<bean id="in" class="com.cognizant.spring\_learn.Country">

<property name="code" value="IN" />

<property name="name" value="India" />

</bean>

<bean id="us" class="com.cognizant.spring\_learn.Country">

<property name="code" value="US" />

<property name="name" value="United States" />

</bean>

<bean id="de" class="com.cognizant.spring\_learn.Country">

<property name="code" value="DE" />

<property name="name" value="Germany" />

</bean>

<bean id="jp" class="com.cognizant.spring\_learn.Country">

<property name="code" value="JP" />

<property name="name" value="Japan" />

</bean>

<!-- List of Countries -->

<bean id="countryList" class="java.util.ArrayList">

<constructor-arg>

<list>

<ref bean="in"/>

<ref bean="us"/>

<ref bean="de"/>

<ref bean="jp"/>

</list>

</constructor-arg>

</bean>

</beans>

SpringLearnApplication.java

package com.cognizant.spring\_learn;

import java.util.ArrayList;

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;

import org.springframework.context.support.ClassPathXmlApplicationContext;

*@SpringBootApplication*

public class SpringLearnApplication {

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

public static void main(String[] args) {

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

*displayCountries*();

}

public static void displayCountries() {

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

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

List<Country> countryList = context.getBean("countryList", ArrayList.class);

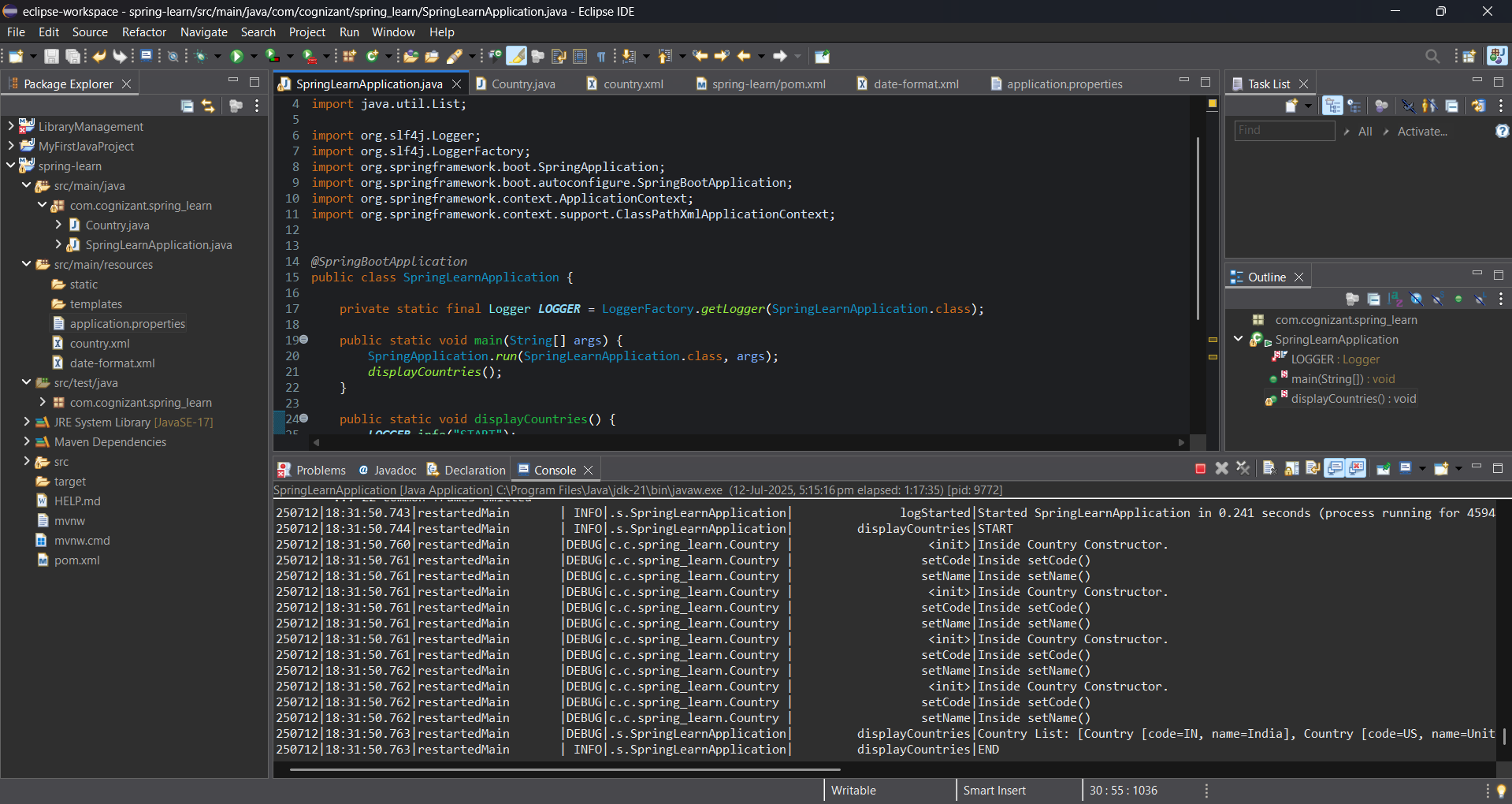
***LOGGER***.debug("Country List: {}", countryList);

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

}

}

Output



**Hello World RESTful Web Service**

HelloController.java

package com.cognizant.spring\_learn.controller;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class HelloController {

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

@GetMapping("/hello")

public String sayHello() {

LOGGER.info("START");

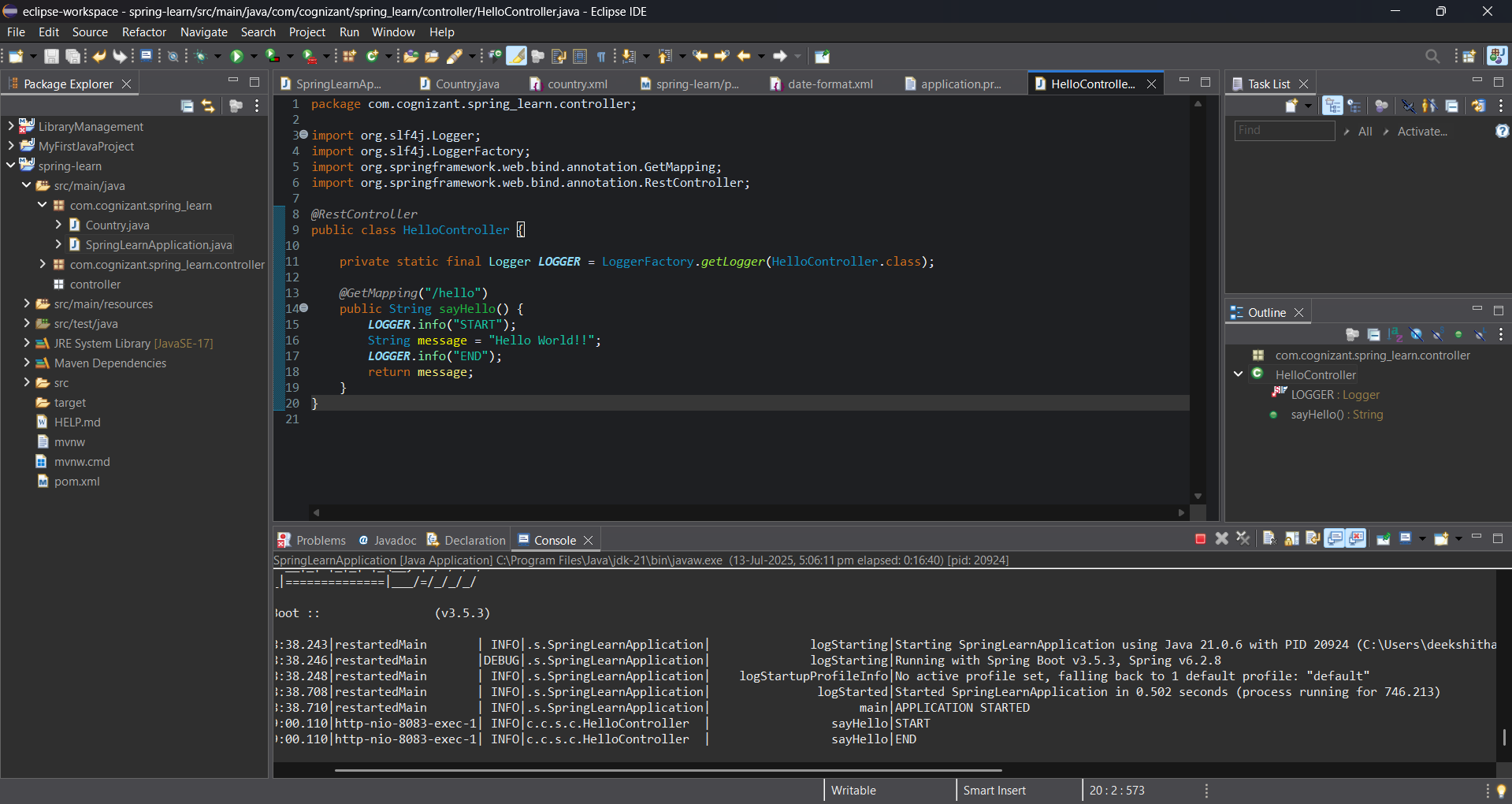
String message = "Hello World!!";

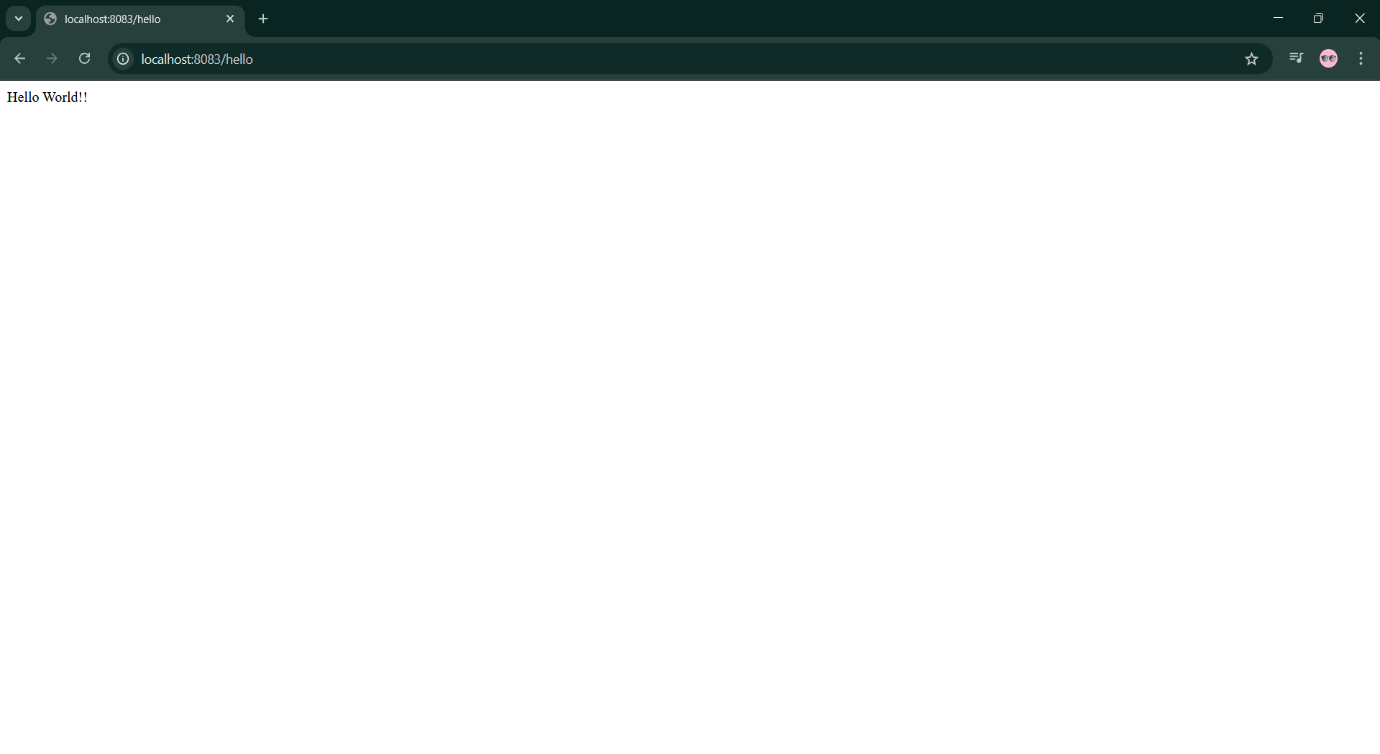
LOGGER.info("END");

return message;

}

}





**REST - Country Web Service**

country.xml

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="in" class="com.cognizant.spring\_learn.Country">

<property name="code" value="IN"/>

<property name="name" value="India"/>

</bean>

</beans>

Country.java

package com.cognizant.spring\_learn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class Country {

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

private String code;

private String name;

public Country() {

LOGGER.debug("Inside Country Constructor.");

}

public String getCode() {

LOGGER.debug("Inside getCode()");

return code;

}

public void setCode(String code) {

LOGGER.debug("Inside setCode()");

this.code = code;

}

public String getName() {

LOGGER.debug("Inside getName()");

return name;

}

public void setName(String name) {

LOGGER.debug("Inside setName()");

this.name = name;

}

@Override

public String toString() {

return "Country [code=" + code + ", name=" + name + "]";

}

}

CountryController.java

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.Country;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class CountryController {

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

@RequestMapping("/country")

public Country getCountryIndia() {

LOGGER.info("START");

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

Country country = context.getBean("in", Country.class);

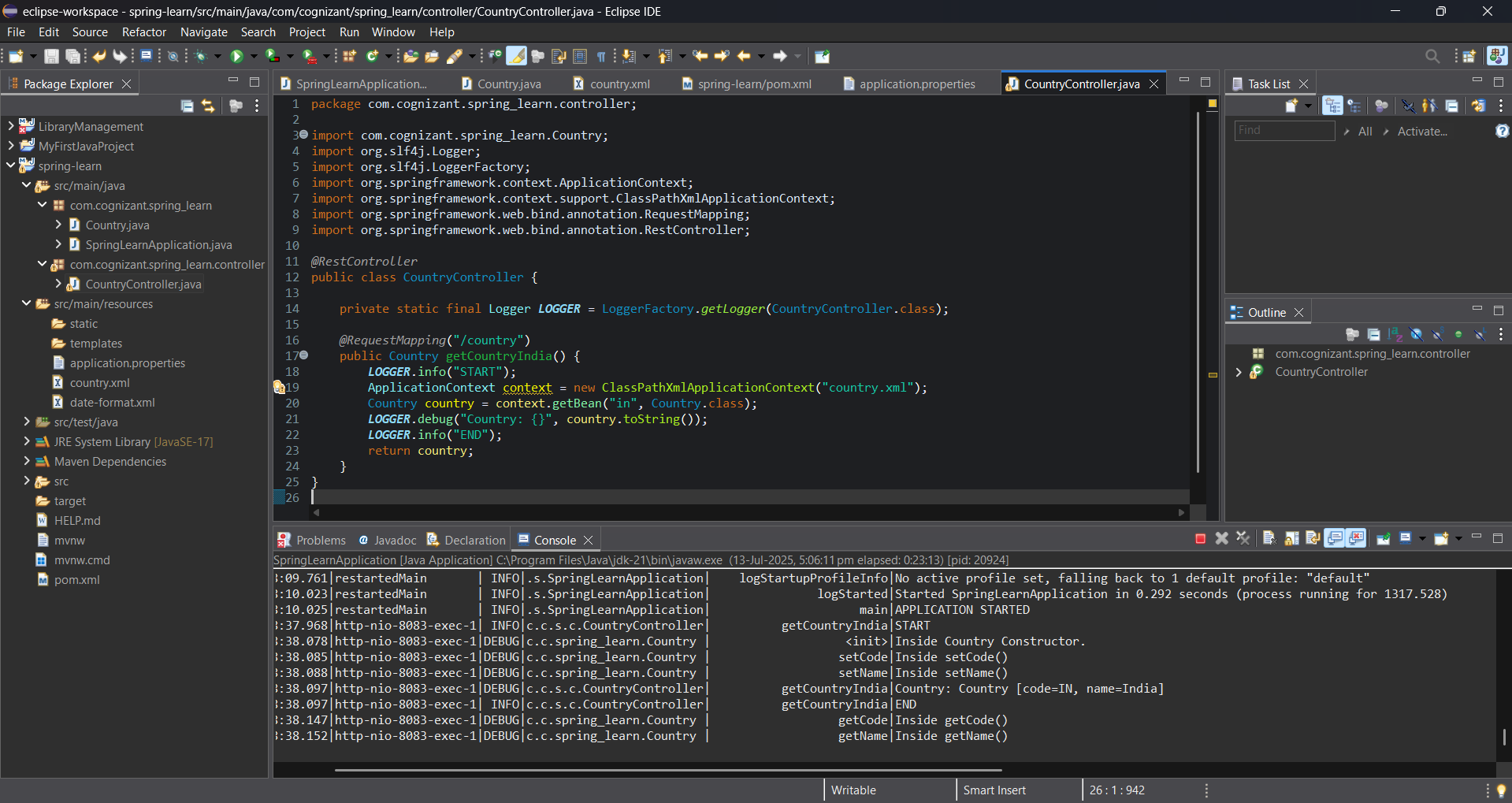
LOGGER.debug("Country: {}", country.toString());

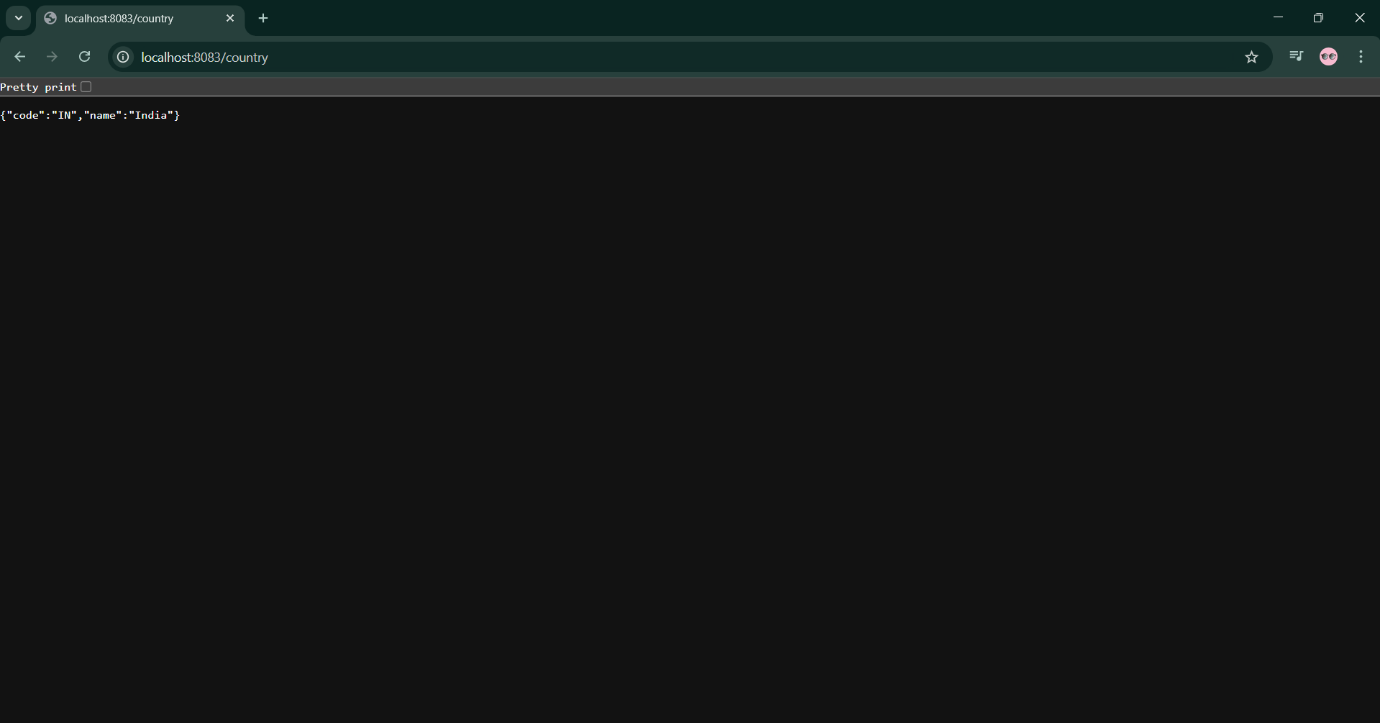
LOGGER.info("END");

return country;

}

}





**REST - Get all countries**

country.xml

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd"

<bean id="in" class="com.cognizant.spring\_learn.Country">

<property name="code" value="IN"/>

<property name="name" value="India"/>

</bean>

<bean id="us" class="com.cognizant.spring\_learn.Country">

<property name="code" value="US"/>

<property name="name" value="United States"/>

</bean>

<bean id="jp" class="com.cognizant.spring\_learn.Country">

<property name="code" value="JP"/>

<property name="name" value="Japan"/>

</bean>

<bean id="de" class="com.cognizant.spring\_learn.Country">

<property name="code" value="DE"/>

<property name="name" value="Germany"/>

</bean>

<!-- List of countries -->

<bean id="countryList" class="java.util.ArrayList">

<constructor-arg>

<list>

<ref bean="in"/>

<ref bean="us"/>

<ref bean="jp"/>

<ref bean="de"/>

</list>

</constructor-arg>

</bean>

</beans>

CountryController.java

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.Country;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

import java.util.List;

@RestController

public class CountryController {

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

@GetMapping("/country")

public Country getCountryIndia() {

LOGGER.info("START - /country");

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

Country country = context.getBean("in", Country.class);

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

LOGGER.info("END - /country");

return country;

}

@GetMapping("/countries")

public List<Country> getAllCountries() {

LOGGER.info("START - /countries");

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

List<Country> countries = context.getBean("countryList", List.class);

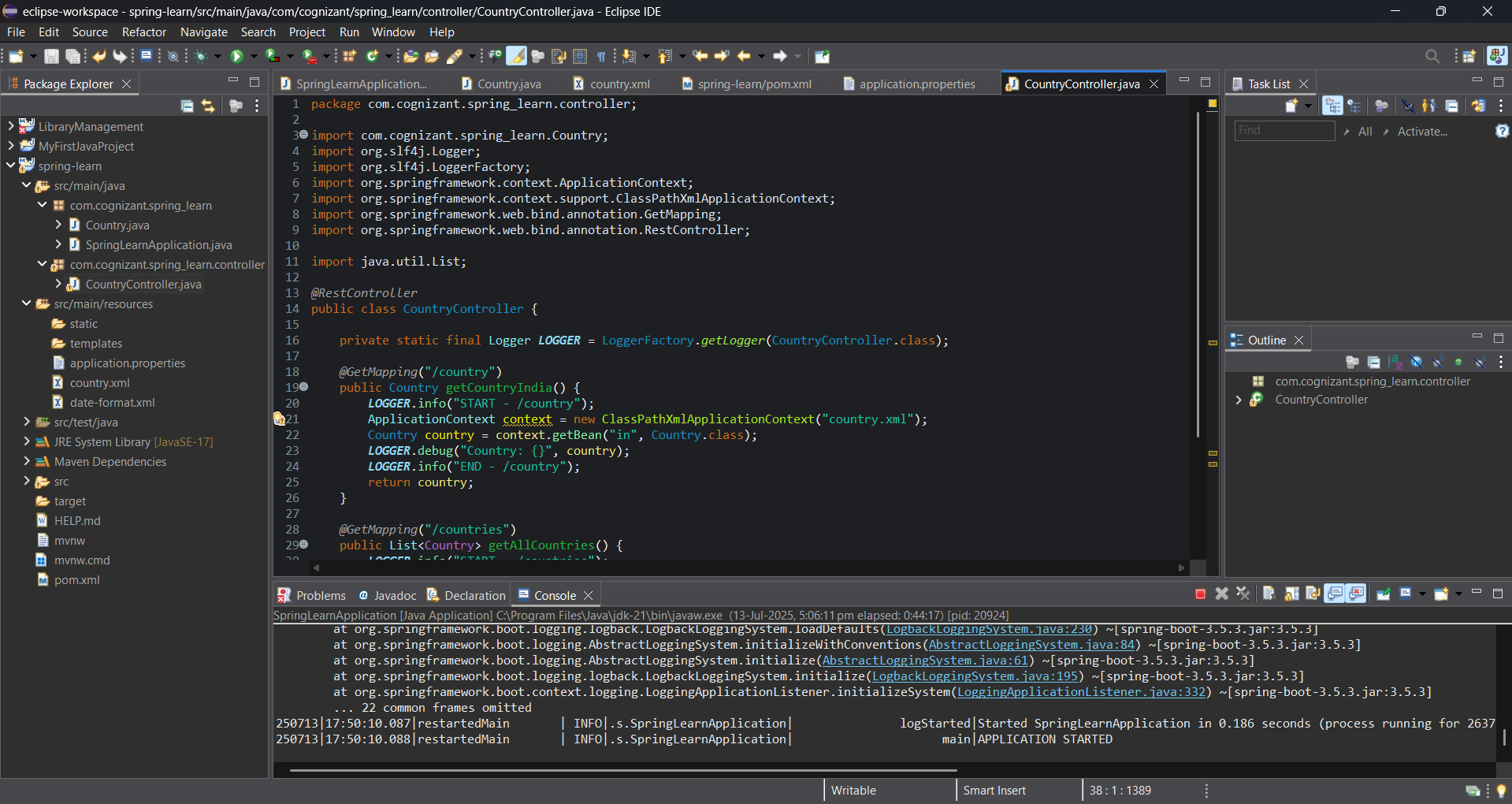
LOGGER.debug("Countries: {}", countries);

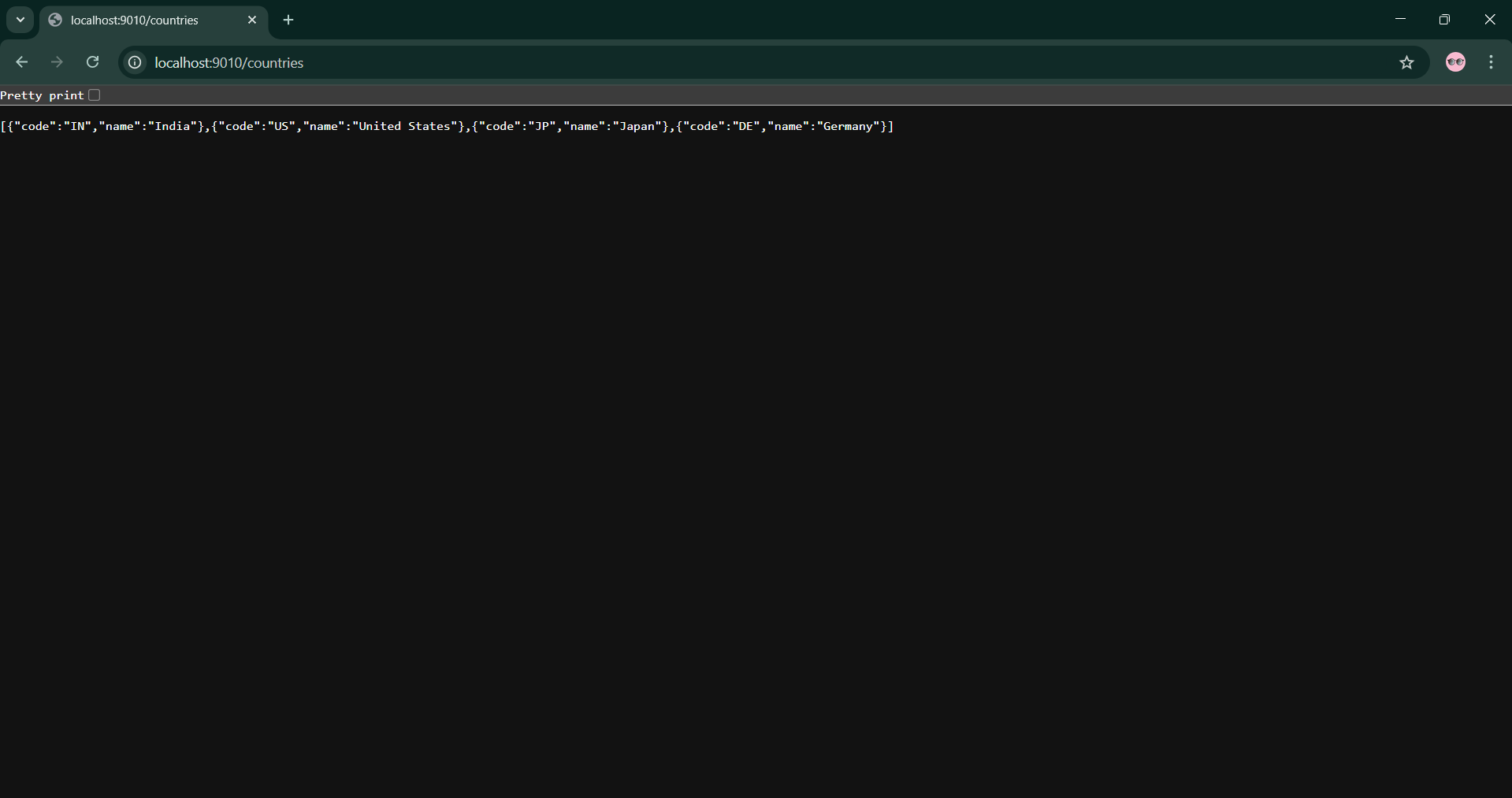
LOGGER.info("END - /countries");

return countries;

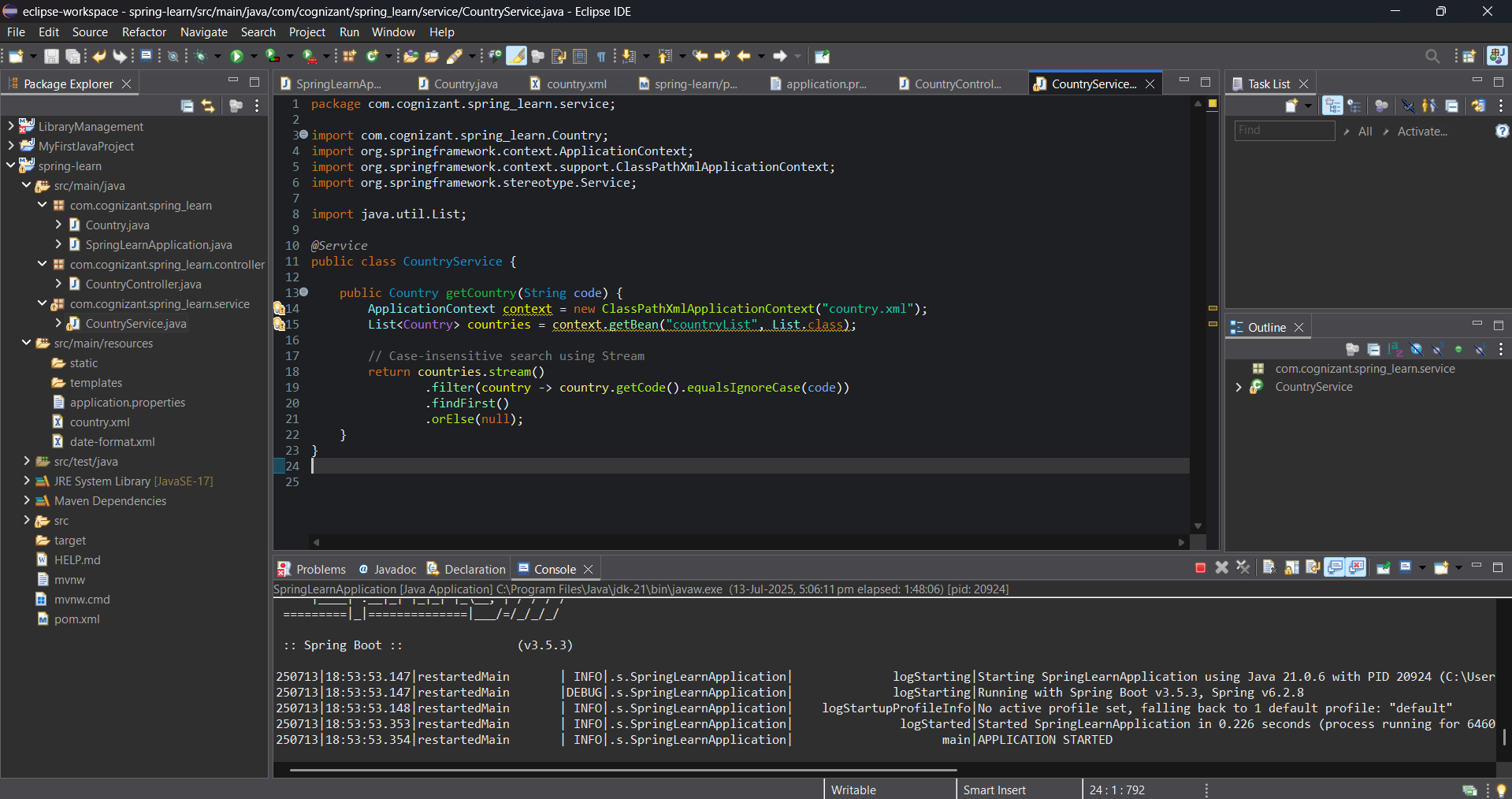
}

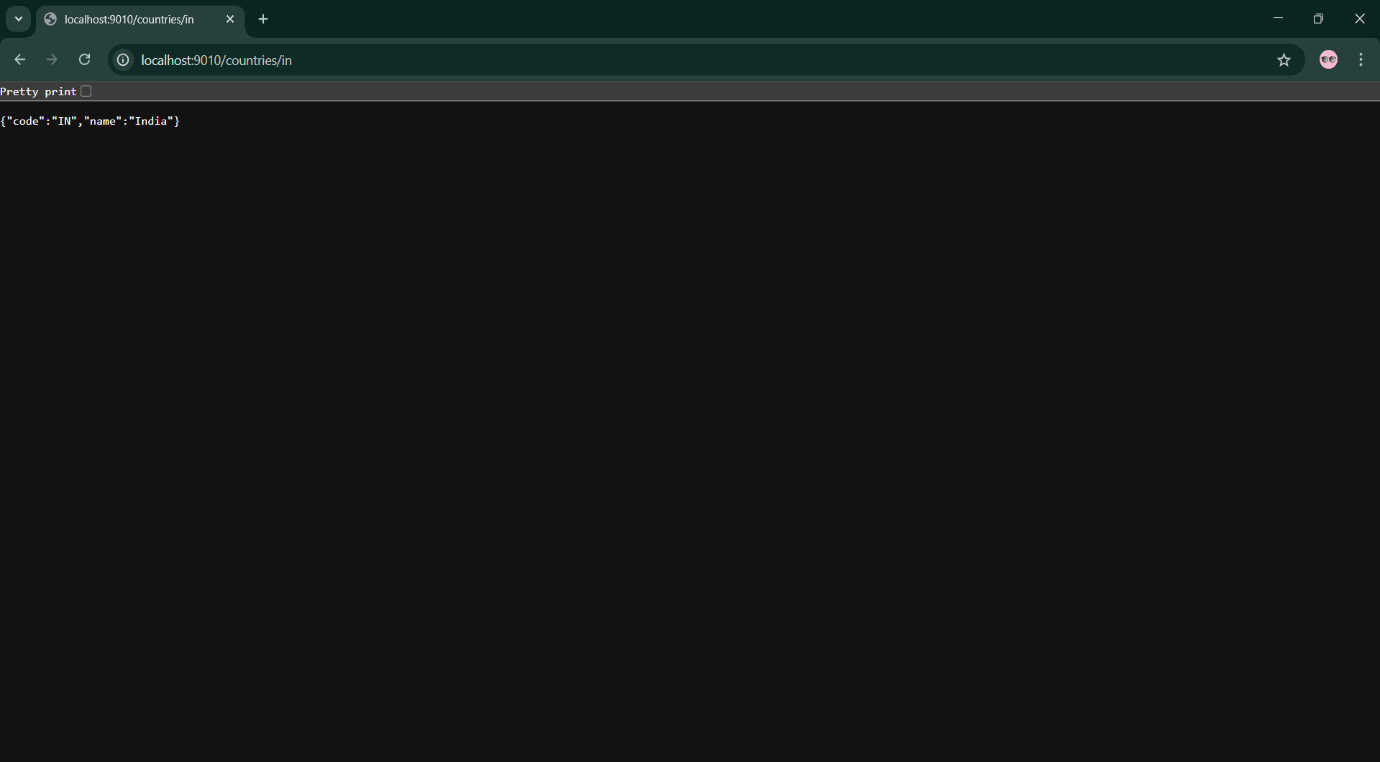
}



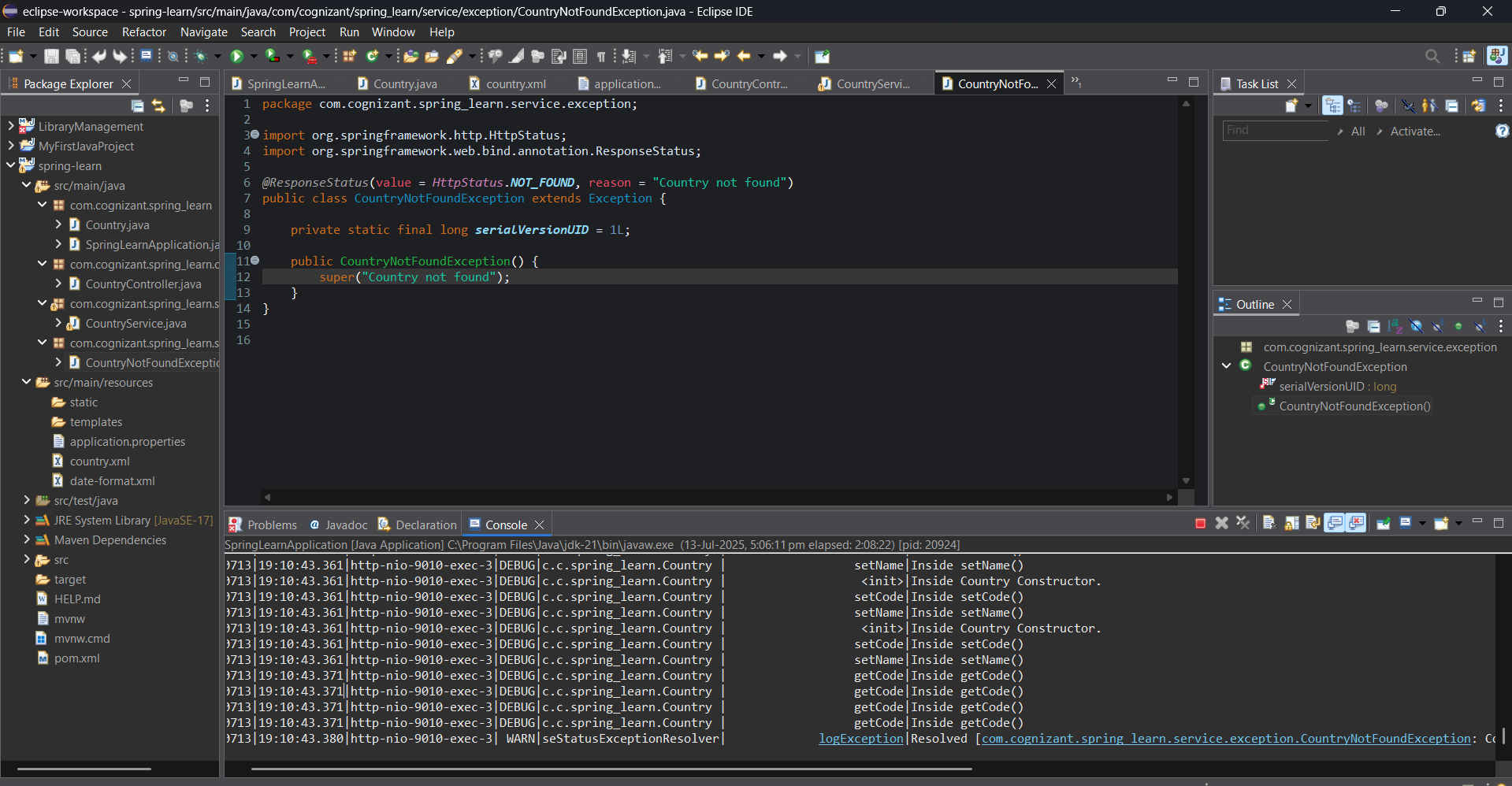


**REST - Get country based on country code**

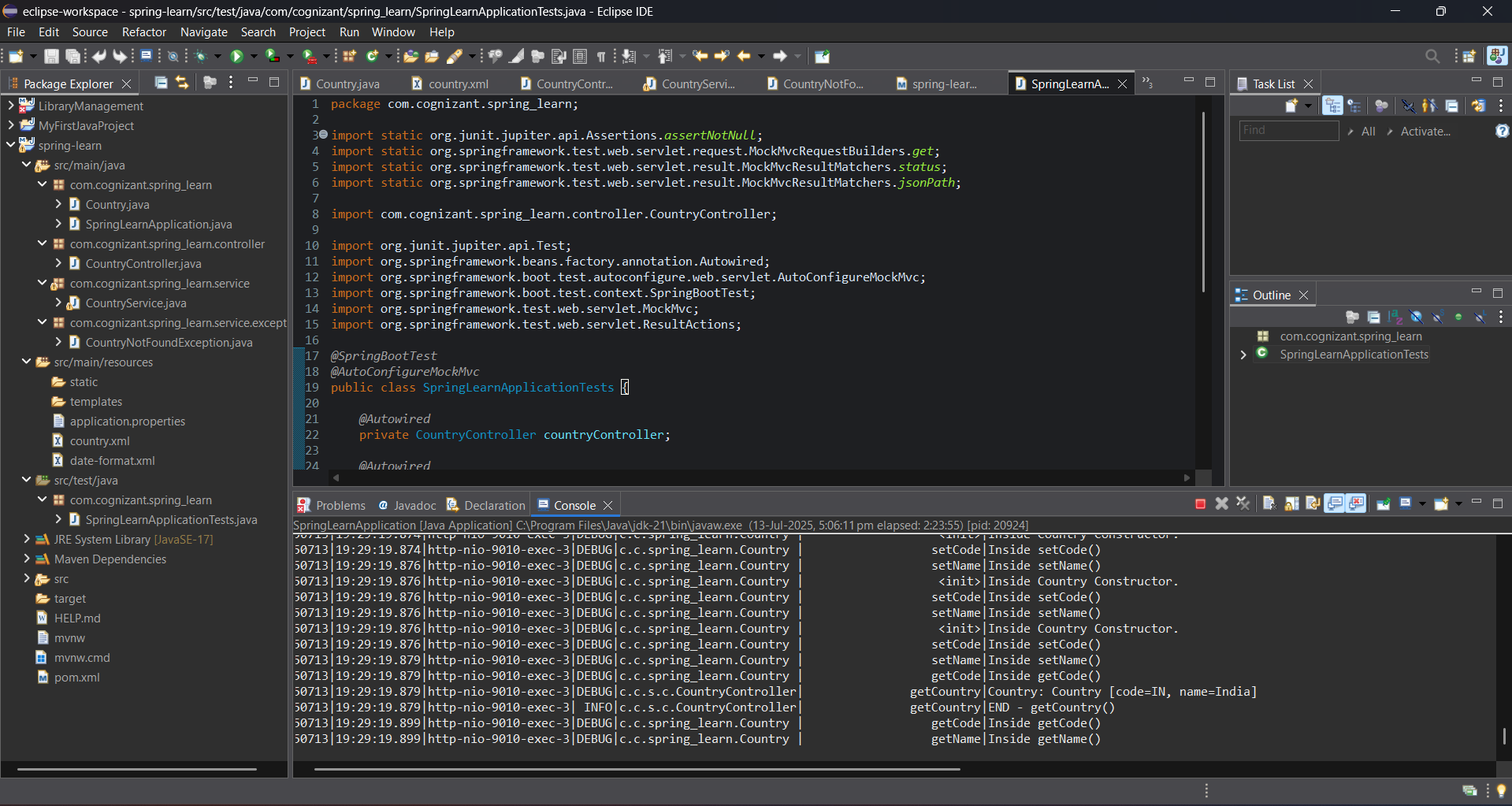




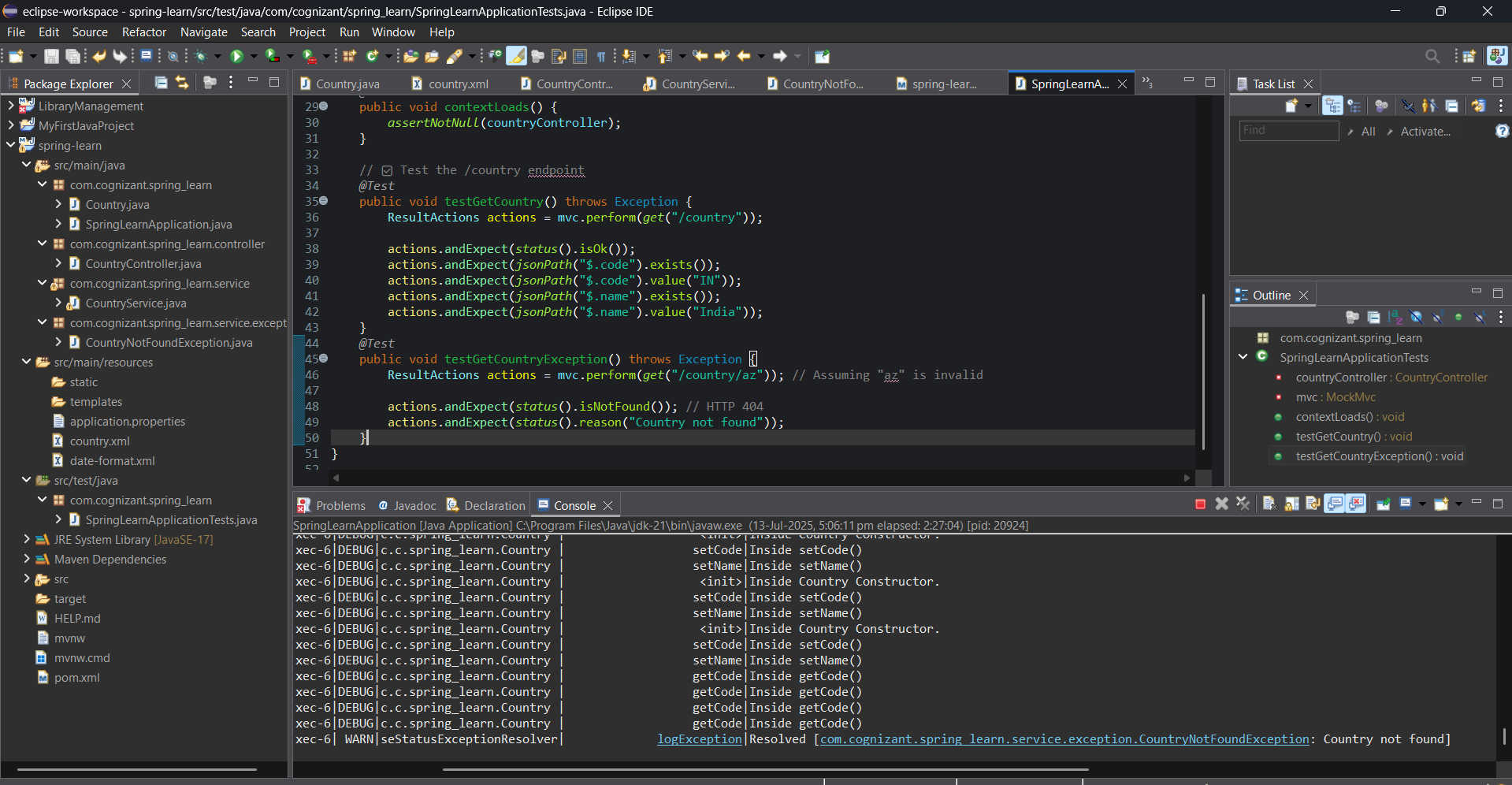
**REST - Get country exceptional scenario** 



**MockMVC - Test get country service** 



**MockMVC - Test get country service for exceptional scenario**



**Problem Statement - Display Employee List and Edit Employee form using RESTful Web Service** 

File: EmployeeProjectApplication.java

package com.cognizant.employee\_project;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class EmployeeProjectApplication {

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

public static void main(String[] args) {

LOGGER.info("START");

SpringApplication.run(EmployeeProjectApplication.class, args);

LOGGER.info("END");

}

}

File: controller/EmployeeController.java

package com.cognizant.employee\_project.controller;

import com.cognizant.employee\_project.model.Employee;

import com.cognizant.employee\_project.service.EmployeeService;

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

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

import java.util.List;

@RestController

public class EmployeeController {

@Autowired

private EmployeeService employeeService;

@GetMapping("/employees")

public List<Employee> getAllEmployees() {

return employeeService.getAllEmployees();

}

}

File: controller/DepartmentController.java

package com.cognizant.employee\_project.controller;

import com.cognizant.employee\_project.model.Department;

import com.cognizant.employee\_project.service.DepartmentService;

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

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

import java.util.List;

@RestController

public class DepartmentController {

@Autowired

private DepartmentService departmentService;

@GetMapping("/departments")

public List<Department> getAllDepartments() {

return departmentService.getAllDepartments();

}

}

File: dao/EmployeeDao.java

package com.cognizant.employee\_project.dao;

import com.cognizant.employee\_project.model.Employee;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Repository;

import java.util.List;

@Repository

public class EmployeeDao {

private static List<Employee> EMPLOYEE\_LIST;

static {

ApplicationContext context = new ClassPathXmlApplicationContext("employee.xml");

EMPLOYEE\_LIST = context.getBean("employeeList", List.class);

}

public List<Employee> getAllEmployees() {

return EMPLOYEE\_LIST;

}

}

// File: dao/DepartmentDao.java

package com.cognizant.employee\_project.dao;

import com.cognizant.employee\_project.model.Department;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Repository;

import java.util.List;

@Repository

public class DepartmentDao {

private static List<Department> DEPARTMENT\_LIST;

static {

ApplicationContext context = new ClassPathXmlApplicationContext("department.xml");

DEPARTMENT\_LIST = context.getBean("departmentList", List.class);

}

public List<Department> getAllDepartments() {

return DEPARTMENT\_LIST;

}

}

// File: service/EmployeeService.java

package com.cognizant.employee\_project.service;

import com.cognizant.employee\_project.dao.EmployeeDao;

import com.cognizant.employee\_project.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

public class EmployeeService {

@Autowired

private EmployeeDao employeeDao;

@Transactional

public List<Employee> getAllEmployees() {

return employeeDao.getAllEmployees();

}

}

File: service/DepartmentService.java

package com.cognizant.employee\_project.service;

import com.cognizant.employee\_project.dao.DepartmentDao;

import com.cognizant.employee\_project.model.Department;

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

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class DepartmentService {

@Autowired

private DepartmentDao departmentDao;

public List<Department> getAllDepartments() {

return departmentDao.getAllDepartments();

}

}

File: model/Employee.java

package com.cognizant.employee\_project.model;

import java.util.List

public class Employee {

private int id;

private String name;

private double salary;

private boolean permanent;

private Department department;

private List<Skill> skillList;

public Employee() {

}

public Employee(int id, String name, double salary, boolean permanent,

Department department, List<Skill> skillList) {

this.id = id;

this.name = name;

this.salary = salary;

this.permanent = permanent;

this.department = department;

this.skillList = 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 Department getDepartment() {

return department;

}

public void setDepartment(Department department) {

this.department = department;

}

public List<Skill> getSkillList() {

return skillList;

}

public void setSkillList(List<Skill> skillList) {

this.skillList = skillList;

}

@Override

public String toString() {

return "Employee{" +

"id=" + id +

", name='" + name + '\'' +

", salary=" + salary +

", permanent=" + permanent +

", department=" + department +

", skills=" + skillList +

'}';

}

}

File: model/Department.java

package com.cognizant.employee\_project.model;

public class Department {

private int id;

private String name;

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;

}

@Override

public String toString() {

return "Department{" + "id=" + id + ", name='" + name + '\'' + '}';

}

}

File: model/Skill.java

package com.cognizant.employee\_project.model;

public class Skill {

private int id;

private String name;

public Skill() {

}

public Skill(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;

}

@Override

public String toString() {

return "Skill{" +

"id=" + id +

", name='" + name + '\'' +

'}';

}

}

employee.xml

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="dept1" class="com.cognizant.employee\_project.model.Department">

<property name="id" value="1"/>

<property name="name" value="HR"/>

</bean>

<bean id="dept2" class="com.cognizant.employee\_project.model.Department">

<property name="id" value="2"/>

<property name="name" value="IT"/>

</bean>

<bean id="skill1" class="com.cognizant.employee\_project.model.Skill">

<property name="id" value="1"/>

<property name="name" value="Java"/>

</bean>

<bean id="skill2" class="com.cognizant.employee\_project.model.Skill">

<property name="id" value="2"/>

<property name="name" value="Angular"/>

</bean>

<bean id="emp1" class="com.cognizant.employee\_project.model.Employee">

<property name="id" value="101"/>

<property name="name" value="Deekshitha"/>

<property name="salary" value="50000"/>

<property name="permanent" value="true"/>

<property name="department" ref="dept1"/>

<property name="skillList">

<list>

<ref bean="skill1"/>

<ref bean="skill2"/>

</list>

</property>

</bean>

<bean id="employeeList" class="java.util.ArrayList">

<constructor-arg>

<list>

<ref bean="emp1"/>

</list>

</constructor-arg>

</bean>

</beans>

Department.xml

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="department1" class="com.cognizant.employee\_project.model.Department">

<property name="id" value="1"/>

<property name="name" value="HR"/>

</bean>

<bean id="department2" class="com.cognizant.employee\_project.model.Department">

<property name="id" value="2"/>

<property name="name" value="IT"/>

</bean>

<bean id="departmentList" class="java.util.ArrayList">

<constructor-arg>

<list>

<ref bean="department1"/>

<ref bean="department2"/>

</list>

</constructor-arg>

</bean>

</beans>

EmployeeProjectApplicationTests.java

package com.cognizant.employee;

import com.cognizant.employee.controller.EmployeeController;

import org.junit.jupiter.api.Test;

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

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

import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;

import org.springframework.test.web.servlet.MockMvc;

import org.springframework.test.web.servlet.ResultActions;

import static org.junit.jupiter.api.Assertions.assertNotNull;

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.get;

import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.\*;

@SpringBootTest

@AutoConfigureMockMvc

public class EmployeeProjectApplicationTests {

@Autowired

private EmployeeController employeeController;

@Autowired

private MockMvc mockMvc;

@Test

public void contextLoads() {

assertNotNull(employeeController);

}

@Test

public void testGetAllEmployees() throws Exception {

ResultActions actions = mockMvc.perform(get("/employees"));

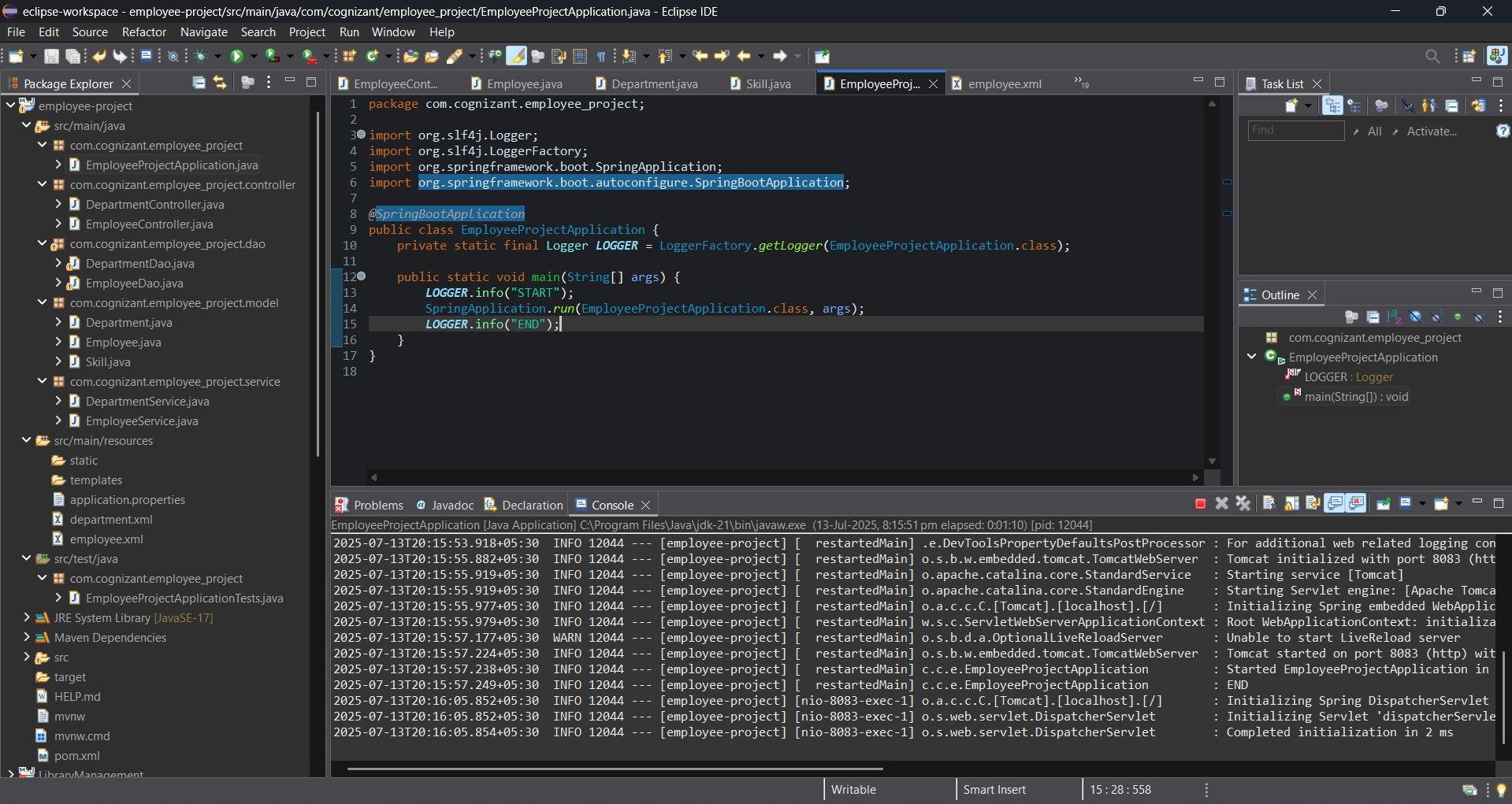
actions.andExpect(status().isOk());

actions.andExpect(jsonPath("$[0].id").exists());

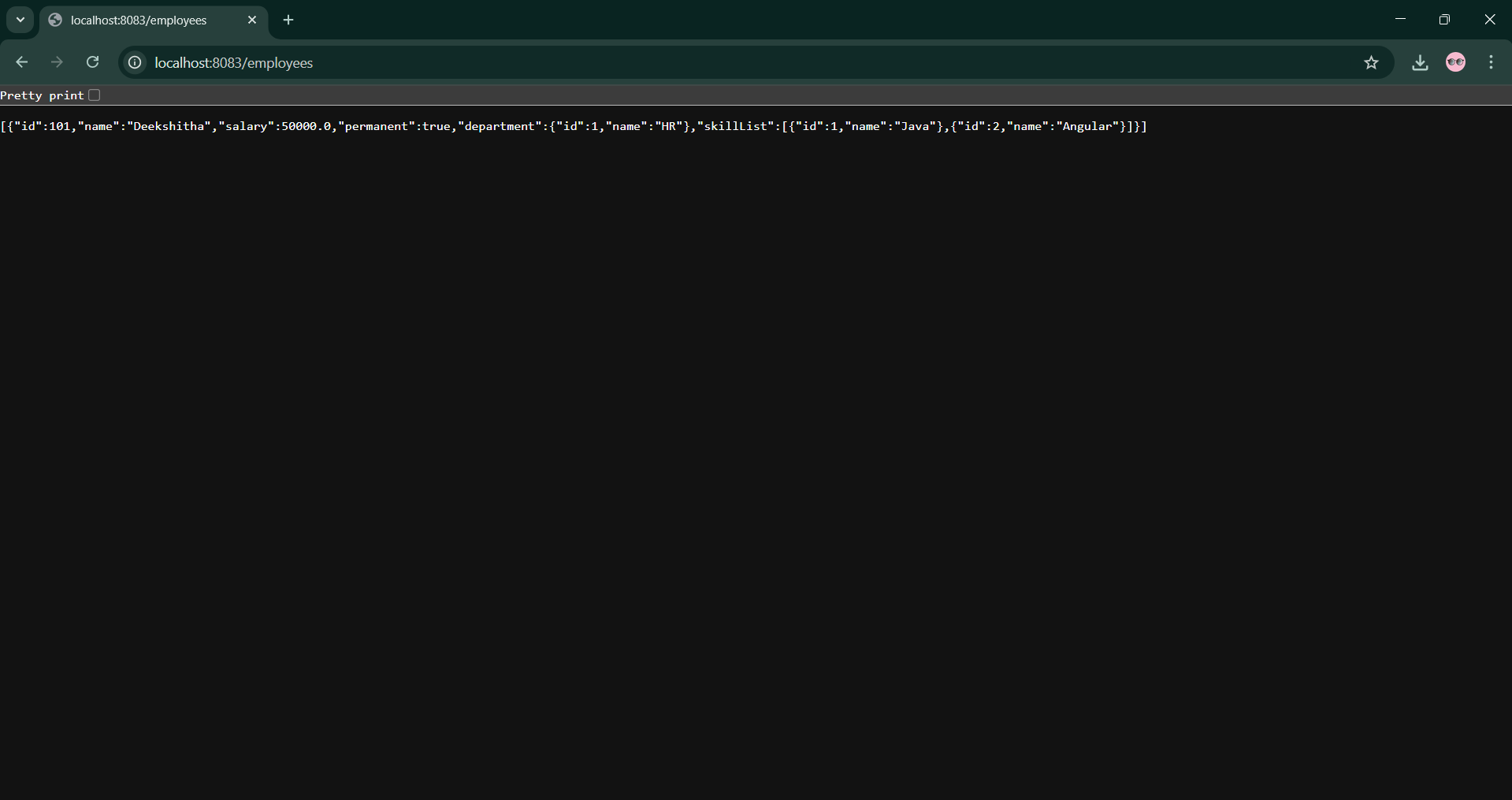
actions.andExpect(jsonPath("$[0].name").value("Deekshitha"));

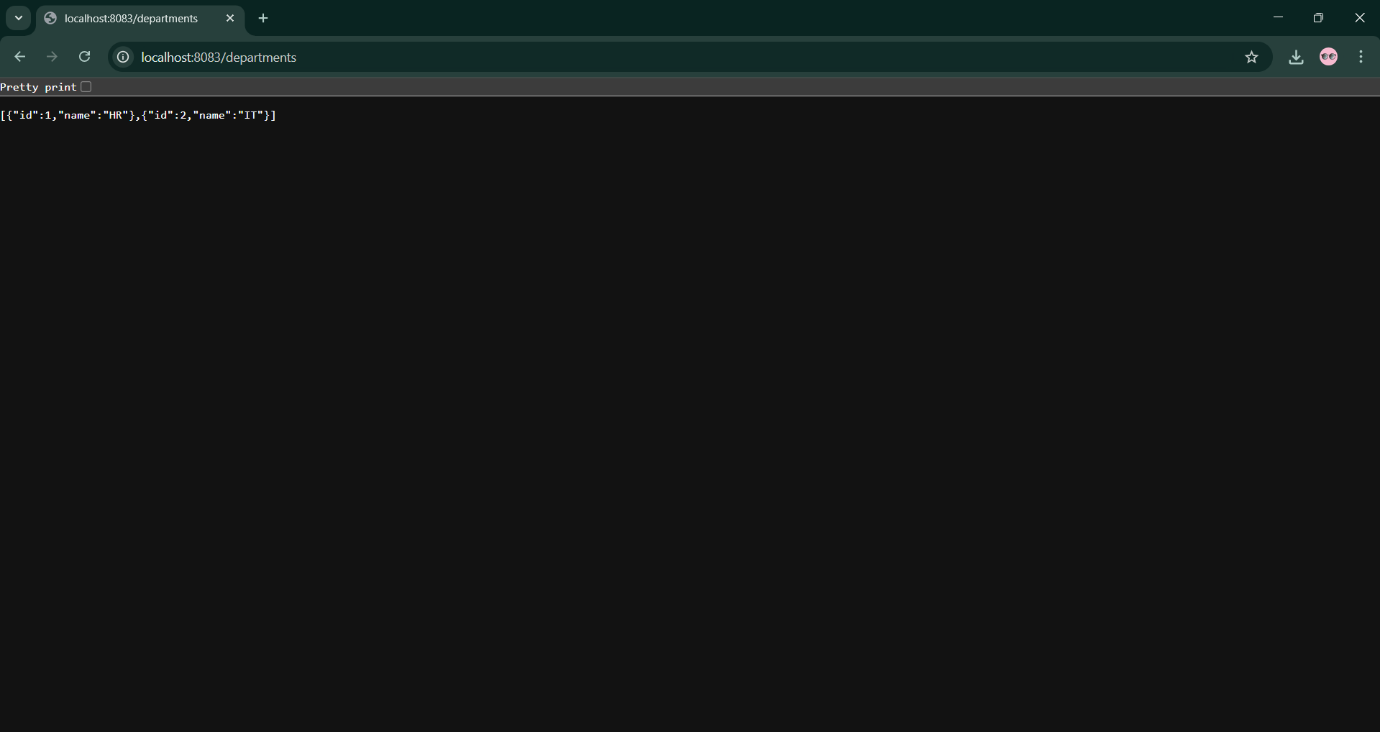
}

}



**Create REST service to gets all employees**

  
  
**Create REST service for department**



**Create authentication service that returns JWT**

**JwtUtil.java (Token Generator)**

package com.cognizant.spring\_learn.util;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import io.jsonwebtoken.security.Keys;

import org.springframework.stereotype.Component;

import java.security.Key;

import java.util.Date;

@Component

public class JwtUtil {

private final Key key = Keys.secretKeyFor(SignatureAlgorithm.HS256);

private final long expiry = 1000 \* 60 \* 60 \* 10; // 10 hours

public String generateToken(String username) {

return Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date(System.currentTimeMillis()))

.setExpiration(new Date(System.currentTimeMillis() + expiry))

.signWith(key)

.compact();

}

}

**JwtResponse.java**

package com.cognizant.spring\_learn.model;

public class JwtResponse {

private String token;

public JwtResponse(String token) {

this.token = token;

}

public String getToken() {

return token

public void setToken(String token) {

this.token = token;

}

}

**AuthenticationController.java**

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.model.JwtResponse;

import com.cognizant.spring\_learn.util.JwtUtil;

import org.springframework.security.core.Authentication;

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

@RestController

public class AuthenticationController {

private final JwtUtil jwtUtil;

public AuthenticationController(JwtUtil jwtUtil) {

this.jwtUtil = jwtUtil;

}

@GetMapping("/authenticate")

public JwtResponse generateToken(Authentication authentication) {

String token = jwtUtil.generateToken(authentication.getName());

return new JwtResponse(token);

}

}

**SecurityConfig.java**

package com.cognizant.spring\_learn.config;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.web.SecurityFilterChain;

import org.springframework.security.crypto.password.NoOpPasswordEncoder;

import org.springframework.security.crypto.password.PasswordEncoder;

import org.springframework.security.core.userdetails.User;

import org.springframework.security.provisioning.InMemoryUserDetailsManager;

import org.springframework.security.core.userdetails.UserDetails;

import org.springframework.security.core.userdetails.UserDetailsService;

@Configuration

public class SecurityConfig {

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http.csrf(csrf -> csrf.disable())

.authorizeHttpRequests(auth -> auth

.requestMatchers("/authenticate").authenticated()

.anyRequest().permitAll()

)

.httpBasic(); // Basic authentication

return http.build();

}

@Bean

public UserDetailsService userDetailsService() {

UserDetails user = User.withUsername("user")

.password("pwd")

.roles("USER")

.build();

return new InMemoryUserDetailsManager(user);

}

@Bean

public PasswordEncoder passwordEncoder() {

return NoOpPasswordEncoder.getInstance();

}

}

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

}

}

**Output:**

C:\Users\deekshitha>cd eclipse-workspace

C:\Users\deekshitha\eclipse-workspace>cd spring-learn

C:\Users\deekshitha\eclipse-workspace\spring-learn>curl -X POST -u user:pwd <http://localhost:9031/authenticate>

{

"token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyIiwiaWF0IjoxNzI4NzE2ODAwLCJleHAiOjE3Mjg3NTI4MDB9.DC3W1jZ...7HfPQ"

}