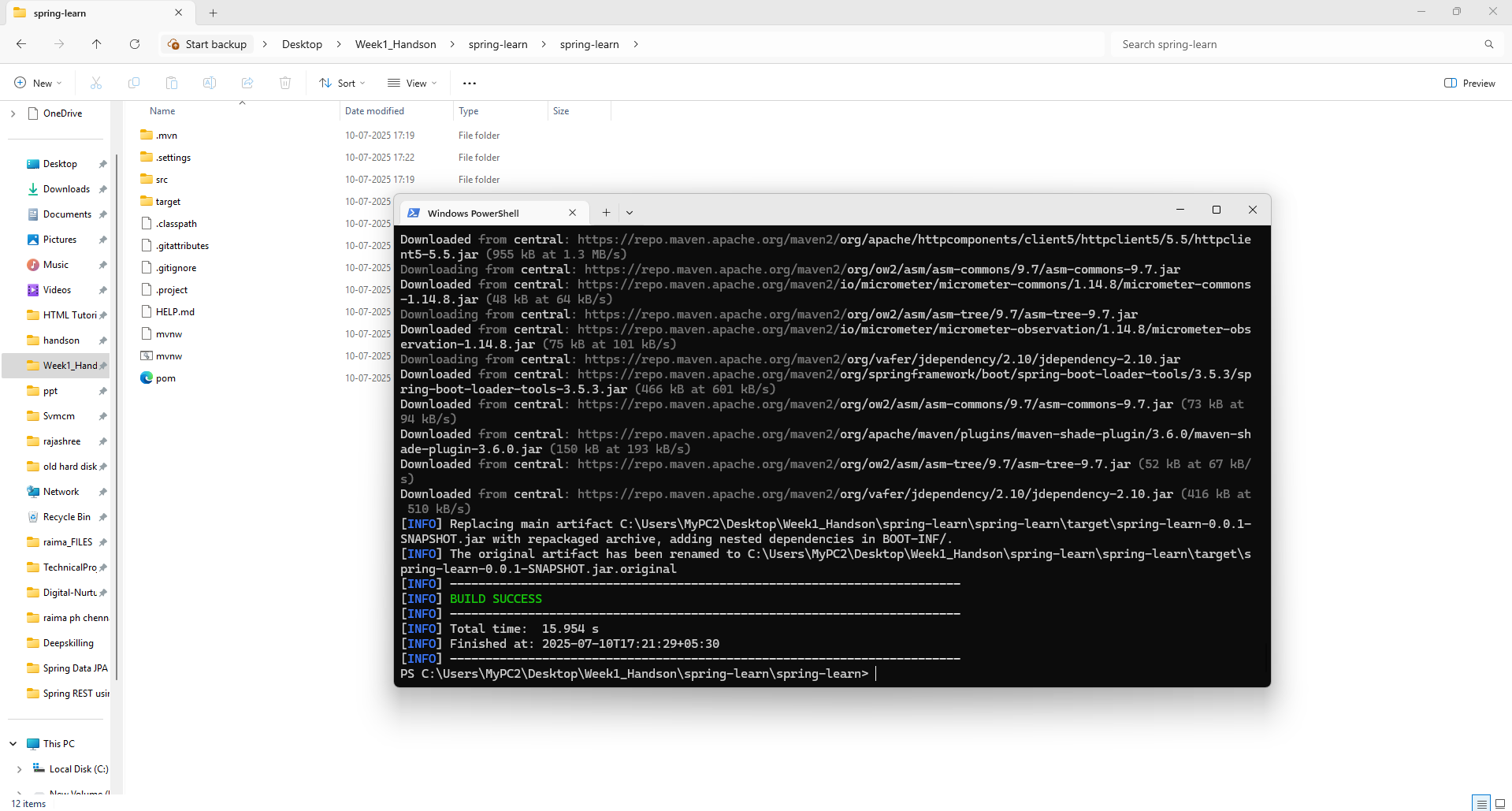
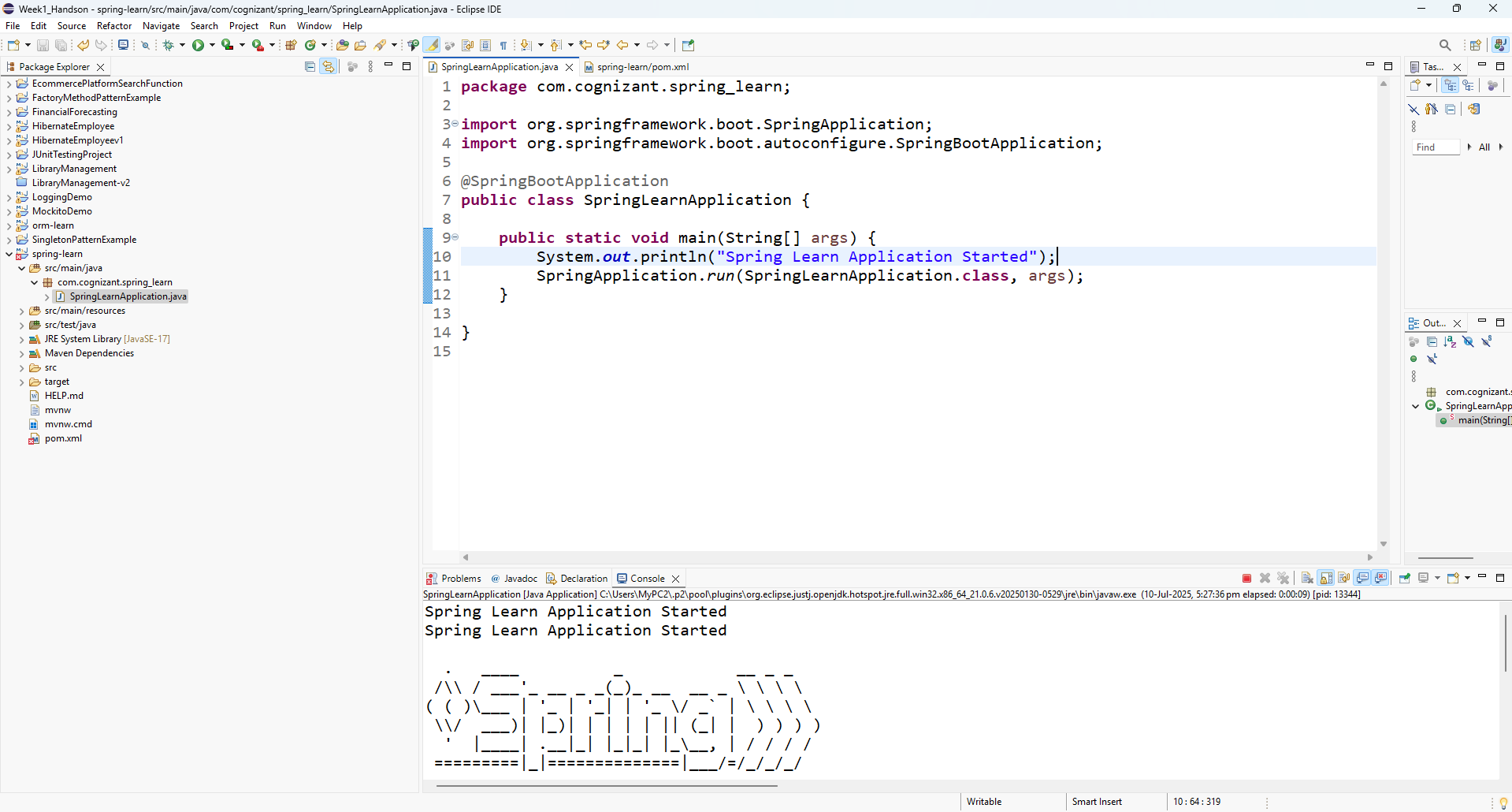
**File 1**

**Hands on 1**

**Create a Spring Web Project using Maven**





src/main/java

com.cognizant.spring\_learn

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) {

System.out.println("Spring Learn Application Started");

SpringApplication.run(SpringLearnApplication.class, args);

}

}

src/main/resources

application.properties

spring.application.name=spring-learn

src/test/java

com.cognizant.spring\_learn

SpringLearnApplicationTests.java

package com.cognizant.spring\_learn;

import org.junit.jupiter.api.Test;

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

@SpringBootTest

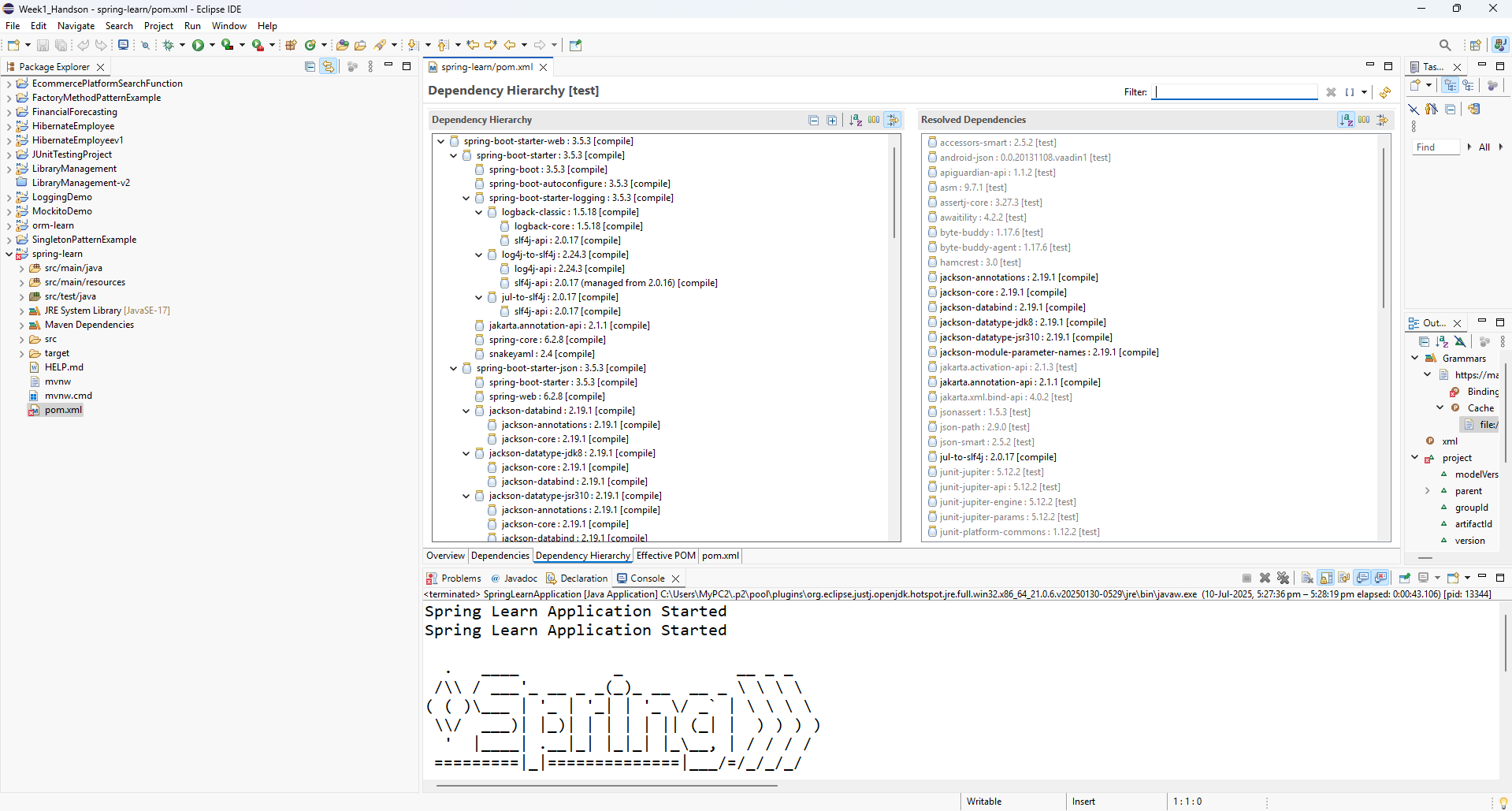
class SpringLearnApplicationTests {

@Test

void contextLoads() {

}

}



**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.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpringLearnApplication {

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

public static void main(String[] args) {

System.out.println("App Started");

LOGGER.debug("START");

displayCountry();

LOGGER.debug("END");

}

public static void displayCountry() {

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

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

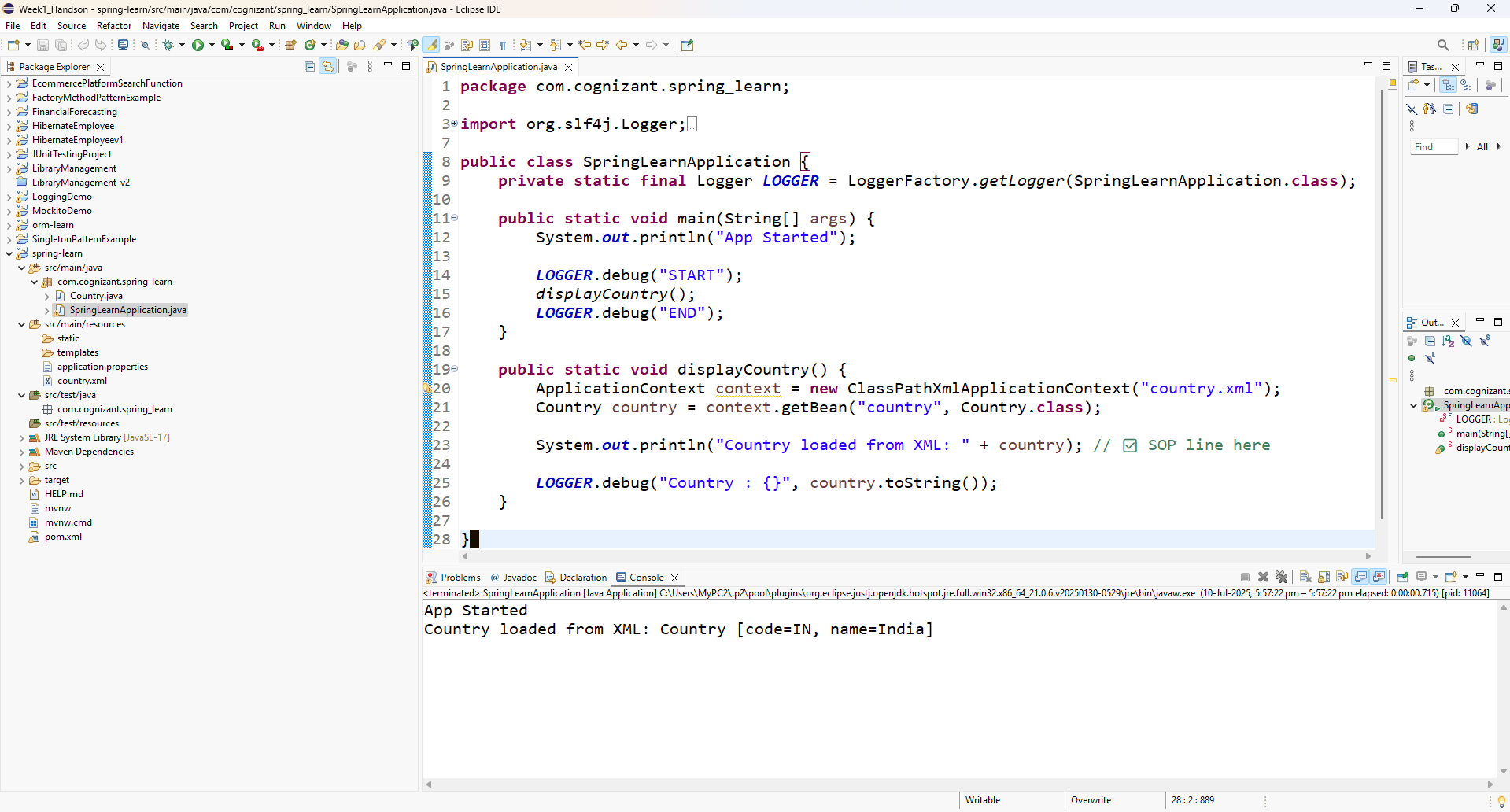
System.out.println("Country loaded from XML: " + country); // ✅ SOP line here

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

}

}

**Output:**



**File 2**

**Hello World RESTful Web Service**

**Project name** - spring-hello-world

src/main/java

com.cognizant.helloworld.controller

**HelloController.java**

package com.cognizant.helloworld.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 - sayHello()");

String message = "Hello World!!";

LOGGER.info("END - sayHello()");

return message;

}

}

**SpringHelloWorldApplication.java**

package com.cognizant.helloworld;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

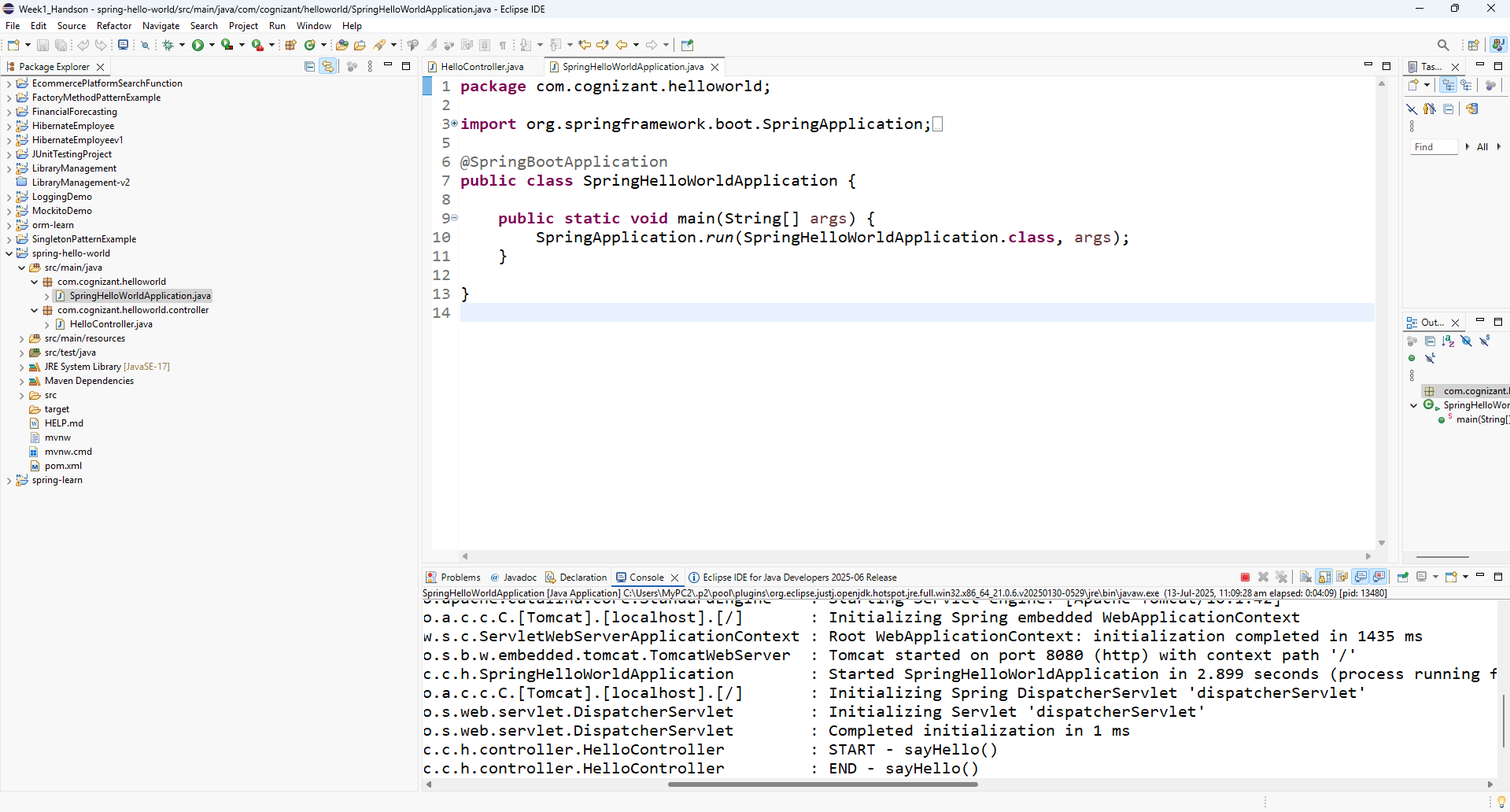
public class SpringHelloWorldApplication {

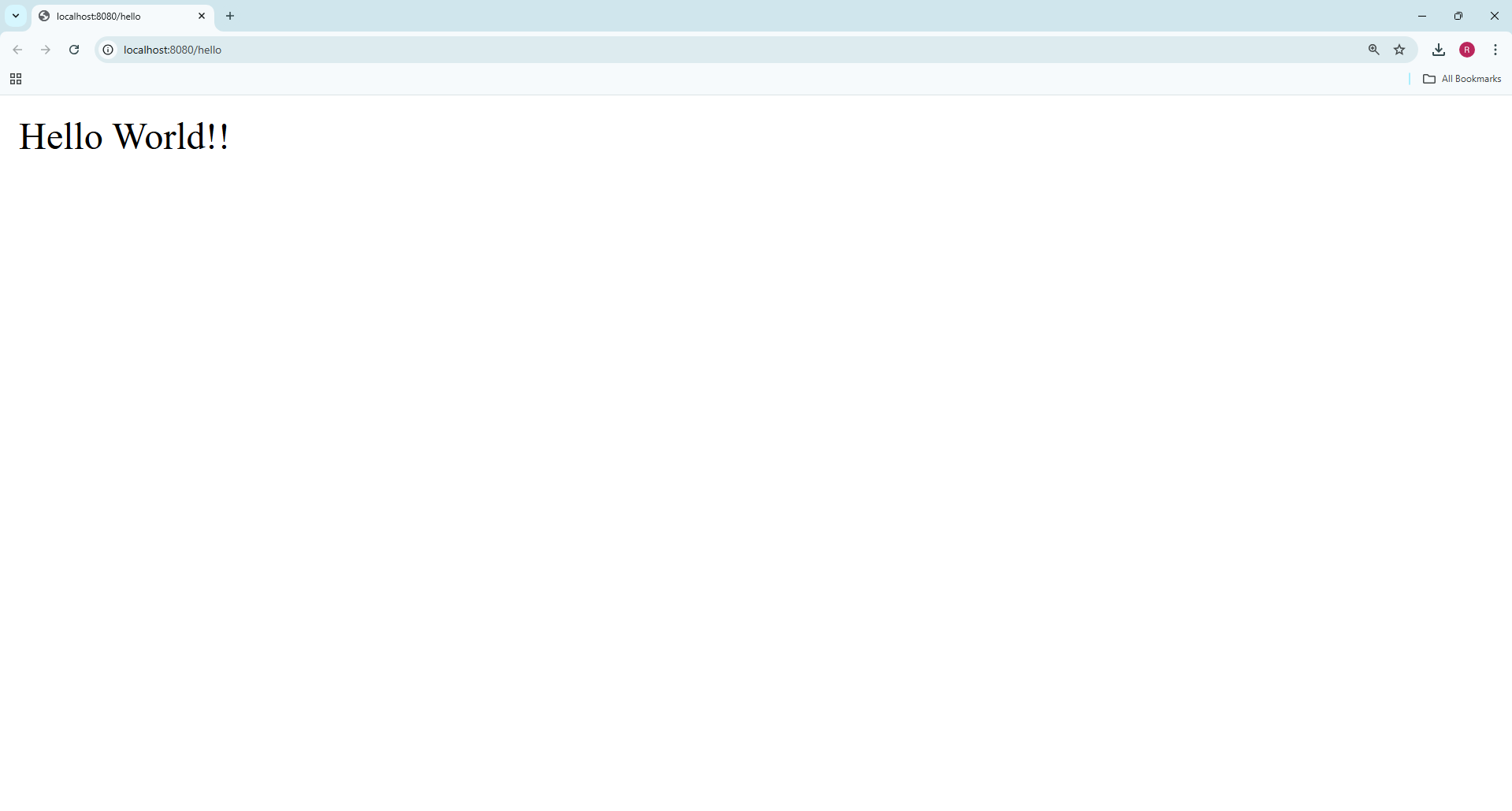
public static void main(String[] args) {

SpringApplication.run(SpringHelloWorldApplication.class, args);

}

}

**Output:**  




**REST - Country Web Service**

**Project name** - spring-country-service

**src/main/java**

com.cognizant.countryservice.model

**Country.java**

package com.cognizant.countryservice.model;

public class Country {

private String code;

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;

}

}

com.cognizant.countryservice.controller

**CountryController.java**

package com.cognizant.countryservice.controller;

import com.cognizant.countryservice.model.Country;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.context.ApplicationContext;

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);

@Autowired

private ApplicationContext context;

@RequestMapping("/country")

public Country getCountryIndia() {

LOGGER.info("START - getCountryIndia()");

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

LOGGER.info("END - getCountryIndia()");

return country;

}

}

**src/main/resources/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.countryservice.model.Country">

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

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

</bean>

</beans>

**SpringCountryServiceApplication.java**

package com.cognizant.countryservice;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.ImportResource;

@SpringBootApplication

@ImportResource("classpath:country.xml") // ✅ Add this line

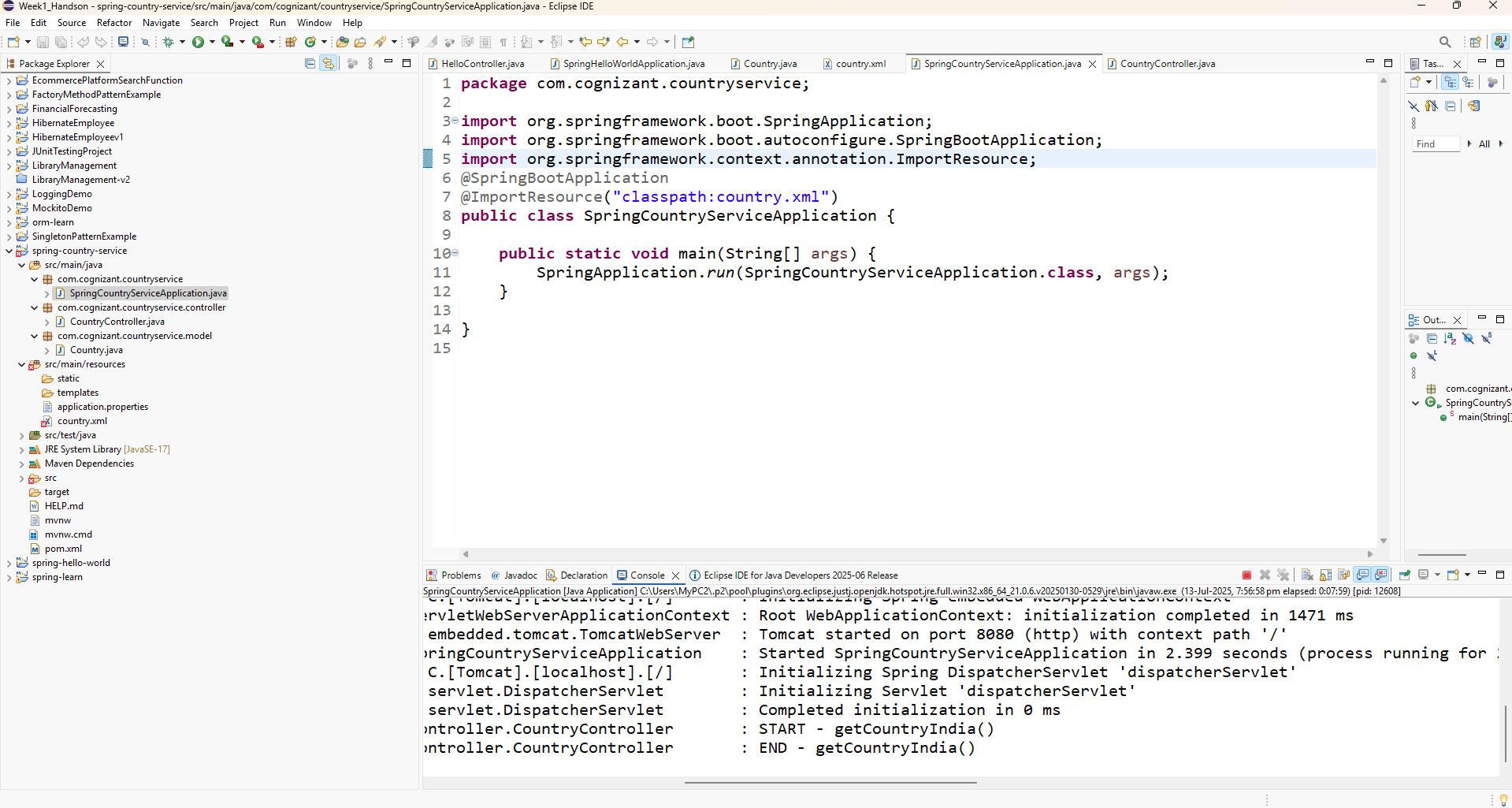
public class SpringCountryServiceApplication {

public static void main(String[] args) {

SpringApplication.run(SpringCountryServiceApplication.class, args);

}

}

Output:  




**REST - Get country based on country code**   
Implementing in the same project

**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

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

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

<constructor-arg>

<list>

<bean class="com.cognizant.countryservice.model.Country">

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

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

</bean>

<bean class="com.cognizant.countryservice.model.Country">

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

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

</bean>

<bean class="com.cognizant.countryservice.model.Country">

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

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

</bean>

</list>

</constructor-arg>

</bean>

</beans>

com.cognizant.countryservice.model

**Country.java**

package com.cognizant.countryservice.model;

public class Country {

private String code;

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;

}

}

com.cognizant.countryservice.service

**CountryService.java**

package com.cognizant.countryservice.service;

import com.cognizant.countryservice.model.Country;

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

import org.springframework.context.ApplicationContext;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class CountryService {

@Autowired

private ApplicationContext context;

public Country getCountry(String code) {

@SuppressWarnings("unchecked")

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

// Case-insensitive match using stream

return countries.stream()

.filter(c -> c.getCode().equalsIgnoreCase(code))

.findFirst()

.orElse(null); // (Optional) Replace with exception

}

}

com.cognizant.countryservice.controller

**CountryController.java**

package com.cognizant.countryservice.controller;

import com.cognizant.countryservice.model.Country;

import com.cognizant.countryservice.service.CountryService;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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

@RestController

public class CountryController {

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

@Autowired

private CountryService countryService;

@GetMapping("/country/{code}")

public Country getCountry(@PathVariable String code) {

LOGGER.info("START - getCountry({})", code);

Country country = countryService.getCountry(code);

LOGGER.info("END - getCountry({})", code);

return country;

}

}

**SpringCountryServiceApplication.java**

package com.cognizant.countryservice;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.ImportResource;

@SpringBootApplication

@ImportResource("classpath:country.xml")

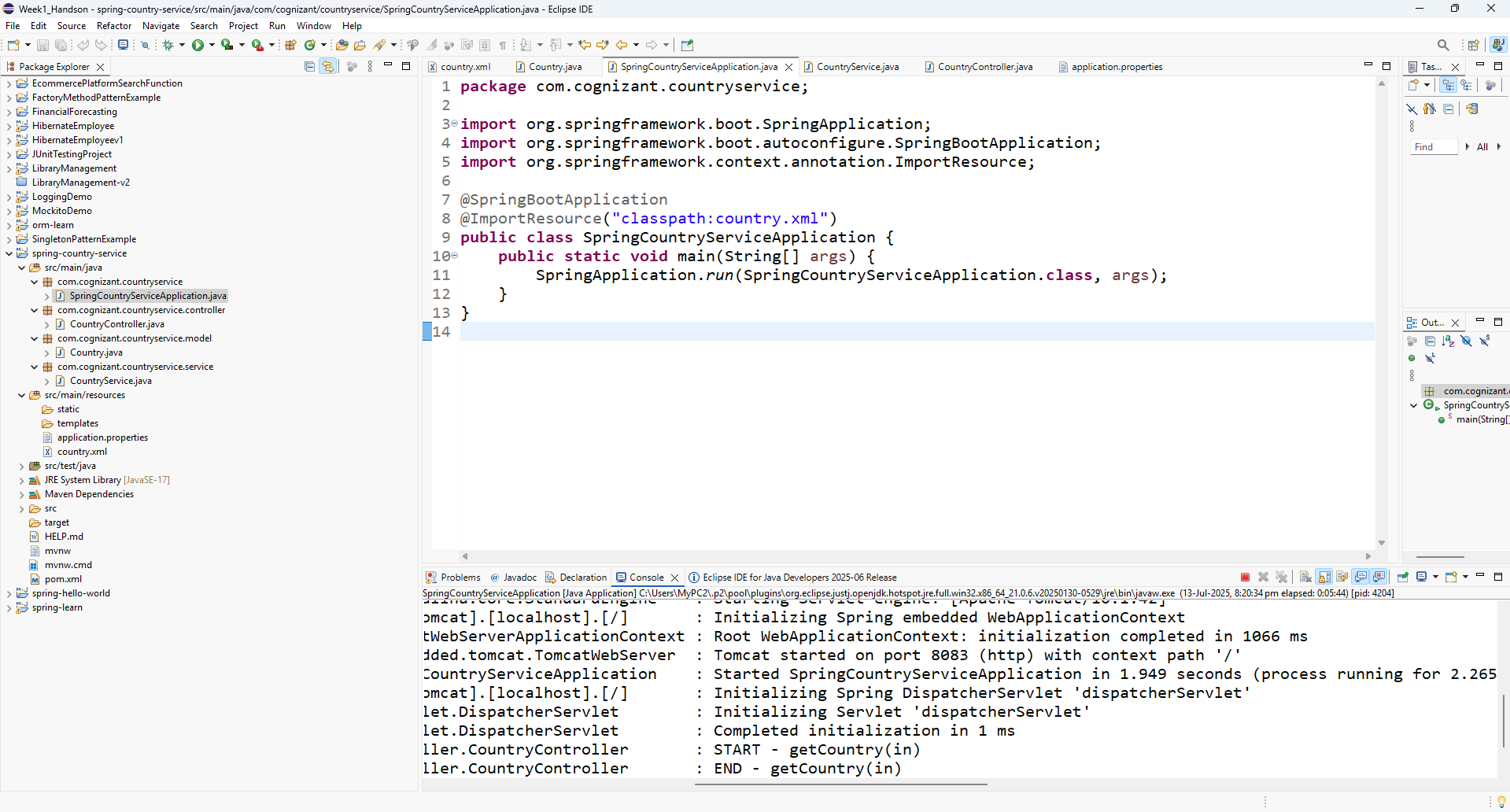
public class SpringCountryServiceApplication {

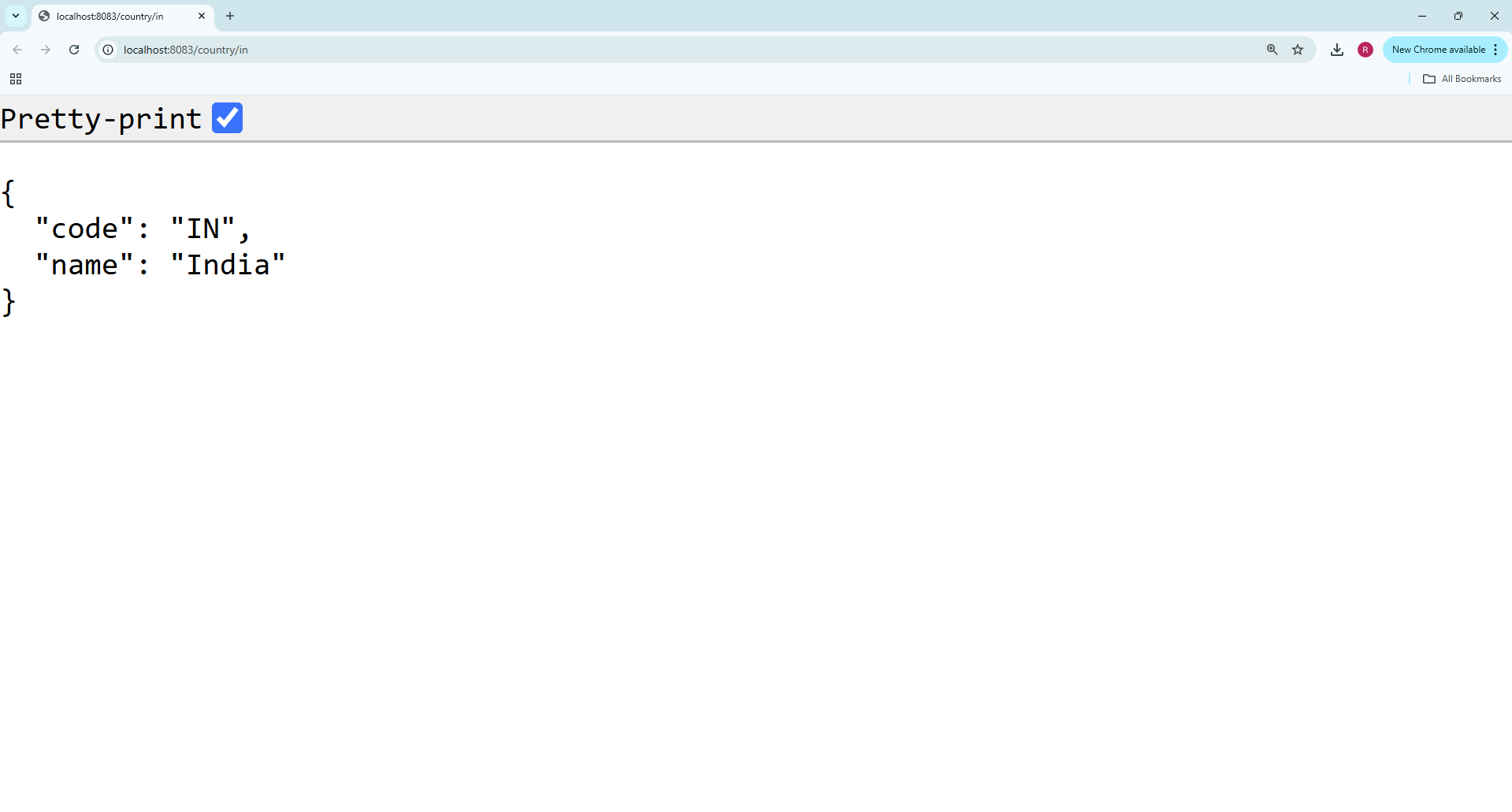
public static void main(String[] args) {

SpringApplication.run(SpringCountryServiceApplication.class, args);

}

}

**Output:** 



**Create authentication service that returns JWT**

Project name - jwt-auth-service

src/main/java

**JwtAuthServiceApplication.java**

package com.cognizant.countryservice;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class JwtAuthServiceApplication {

public static void main(String[] args) {

SpringApplication.run(JwtAuthServiceApplication.class, args);

}

}com.cognizant.countryservice.auth

**AuthenticationController.java**

package com.cognizant.countryservice.auth;

import org.springframework.http.ResponseEntity;

import com.cognizant.countryservice.auth.TokenService;

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

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

import jakarta.servlet.http.HttpServletRequest;

@RestController

public class AuthenticationController {

private final TokenService tokenService;

public AuthenticationController(TokenService tokenService) {

this.tokenService = tokenService;

}

@GetMapping("/authenticate")

public ResponseEntity<TokenResponse> authenticate(HttpServletRequest request) {

var creds = BasicAuthUtils.*extract*(request); // username & password

tokenService.verify(creds); // raises on bad creds

String jwt = tokenService.generate(creds.username());

return ResponseEntity.*ok*(new TokenResponse(jwt));

}

/\*\* Small record = JSON body {"token":"..."} \*/

record TokenResponse(String token) {}

}

**BasicAuthUtils.java**

package com.cognizant.countryservice.auth;

import java.nio.charset.StandardCharsets;

import java.util.Base64;

public class BasicAuthUtils {

/\*\* Simple value object for the pair \*/

public record Creds(String username, String password) {}

public static Creds extract(jakarta.servlet.http.HttpServletRequest request) {

String header = request.getHeader("Authorization");

if (header == null || !header.startsWith("Basic "))

throw new IllegalArgumentException("Missing or invalid Basic header");

String base64 = header.substring(6);

String decoded = new String(

Base64.getDecoder().decode(base64),

StandardCharsets.UTF\_8);

int colon = decoded.indexOf(':');

if (colon < 0) throw new IllegalArgumentException("Malformed credentials");

return new Creds(decoded.substring(0, colon), decoded.substring(colon + 1));

}

}

**TokenService.java**

package com.cognizant.countryservice.auth;

import java.time.Instant;

import java.util.Date;

import javax.crypto.SecretKey;

import org.springframework.security.authentication.AuthenticationManager;

import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;

import org.springframework.stereotype.Service;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.security.Keys;

@Service

public class TokenService {

private final AuthenticationManager authManager;

private final SecretKey key = Keys.hmacShaKeyFor(

// 32‑byte (256‑bit) key minimum – move to config in production

"change‑this‑to‑a‑real‑256‑bit‑secret‑key‑please".getBytes());

public TokenService(AuthenticationManager authManager) {

this.authManager = authManager;

}

/\*\* Delegates credential‑check to Spring Security \*/

public void verify(BasicAuthUtils.Creds creds) {

var auth = new UsernamePasswordAuthenticationToken(

creds.username(), creds.password());

authManager.authenticate(auth); // throws BadCredentialsException if wrong

}

public String generate(String username) {

Instant now = Instant.now();

return Jwts.builder()

.setSubject(username)

.setIssuedAt(Date.from(now))

.setExpiration(Date.from(now.plusSeconds(600))) // 10 min

.signWith(key)

.compact();

}

}

com.cognizant.countryservice.config

**SecurityConfig.java**

package com.cognizant.countryservice.config;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.authentication.AuthenticationManager;

import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;

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

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

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

import org.springframework.security.provisioning.InMemoryUserDetailsManager;

import org.springframework.security.web.SecurityFilterChain;

@Configuration

public class SecurityConfig {

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

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

.authorizeHttpRequests(auth -> auth

.requestMatchers("/authenticate").permitAll()

.anyRequest().authenticated())

.httpBasic(basic -> {}); // Enables basic auth

return http.build();

}

@Bean

public UserDetailsService users() {

var user = User.withUsername("user")

.password("{noop}pwd") // {noop} for plaintext (for demo)

.roles("USER")

.build();

return new InMemoryUserDetailsManager(user);

}

@Bean

public AuthenticationManager authenticationManager(HttpSecurity http) throws Exception {

return http.getSharedObject(AuthenticationManagerBuilder.class).build();

}

}

Output:

