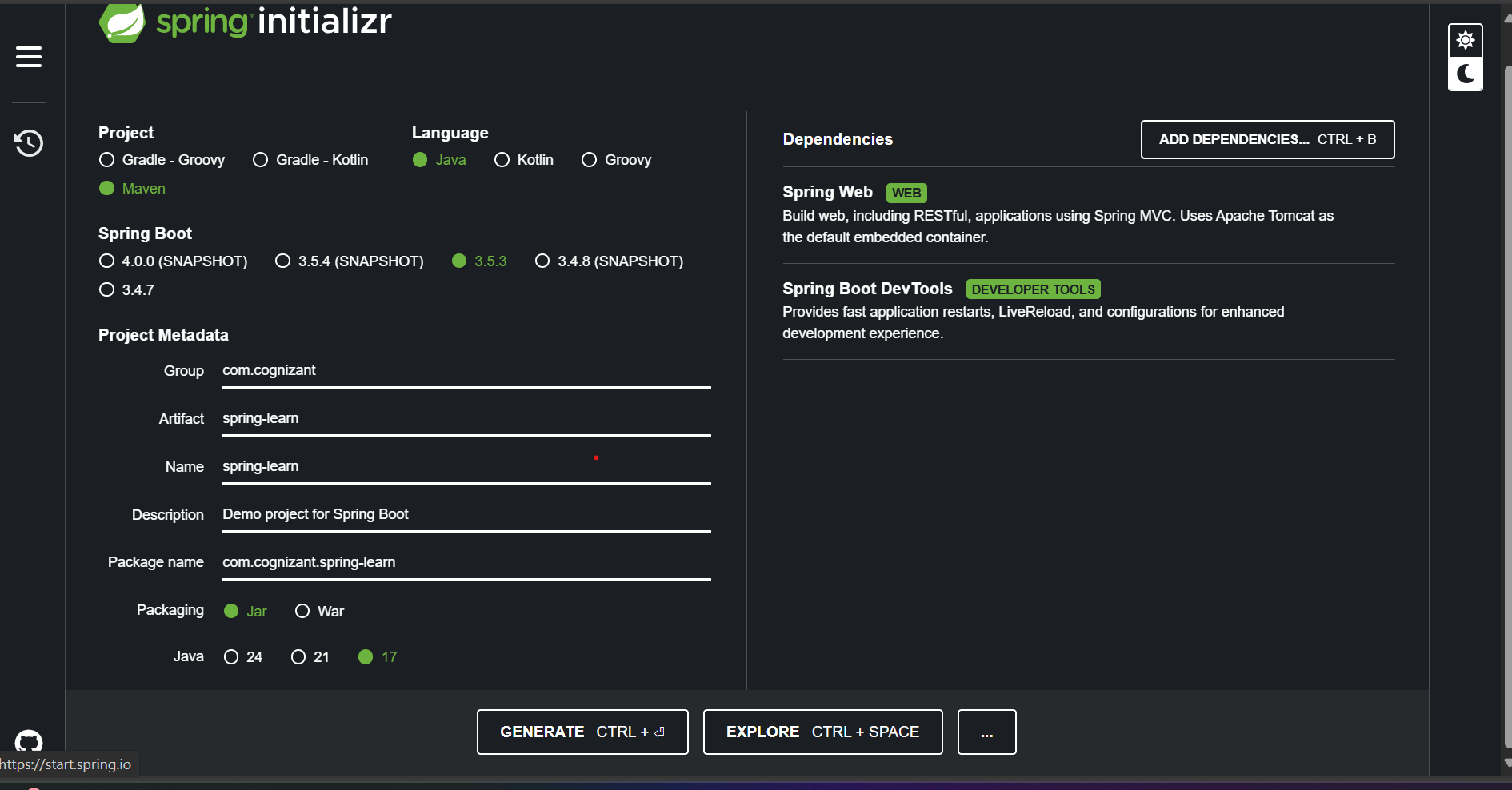
**Spring REST & JWT Hands-On**

1. **Create a Spring Web Project using Maven**



**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>spring-learn</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>spring-learn</name>

<description>Demo project for Spring Boot</description>

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

**SpringLearnApplication.java :**

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) {

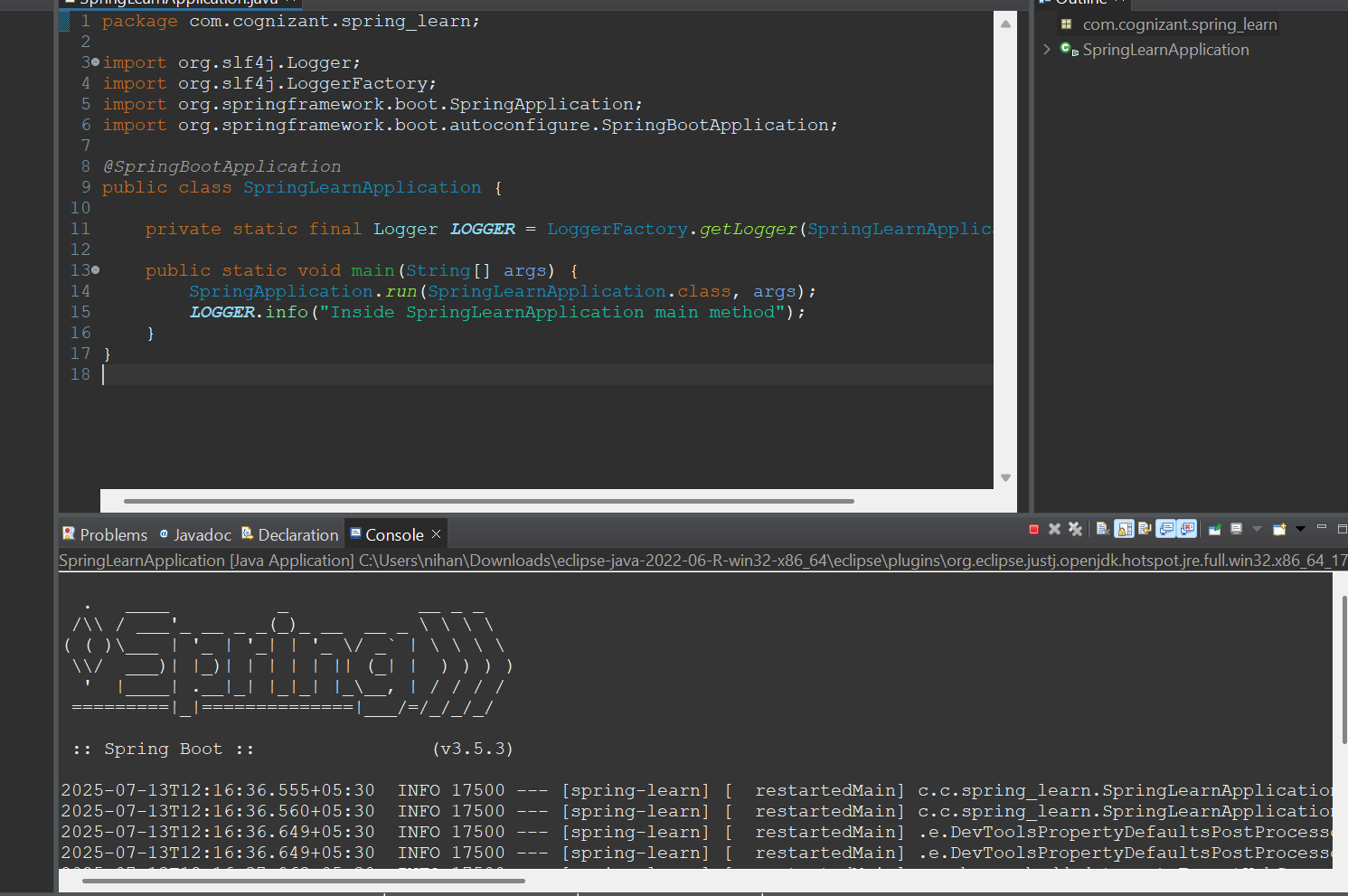
SpringApplication.run(SpringLearnApplication.class, args);

LOGGER.info("Inside SpringLearnApplication main method");

}

}

**Output :**



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

### **1. Create XML File**

**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>

**2. Add displayDate() Method in SpringLearnApplication.java**

package com.cognizant.springlearn;

import java.text.ParseException;

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.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) throws ParseException {

SpringApplication.run(SpringLearnApplication.class, args);

LOGGER.info("Inside SpringLearnApplication main method");

displayDate();

}

public static void displayDate() throws ParseException {

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

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

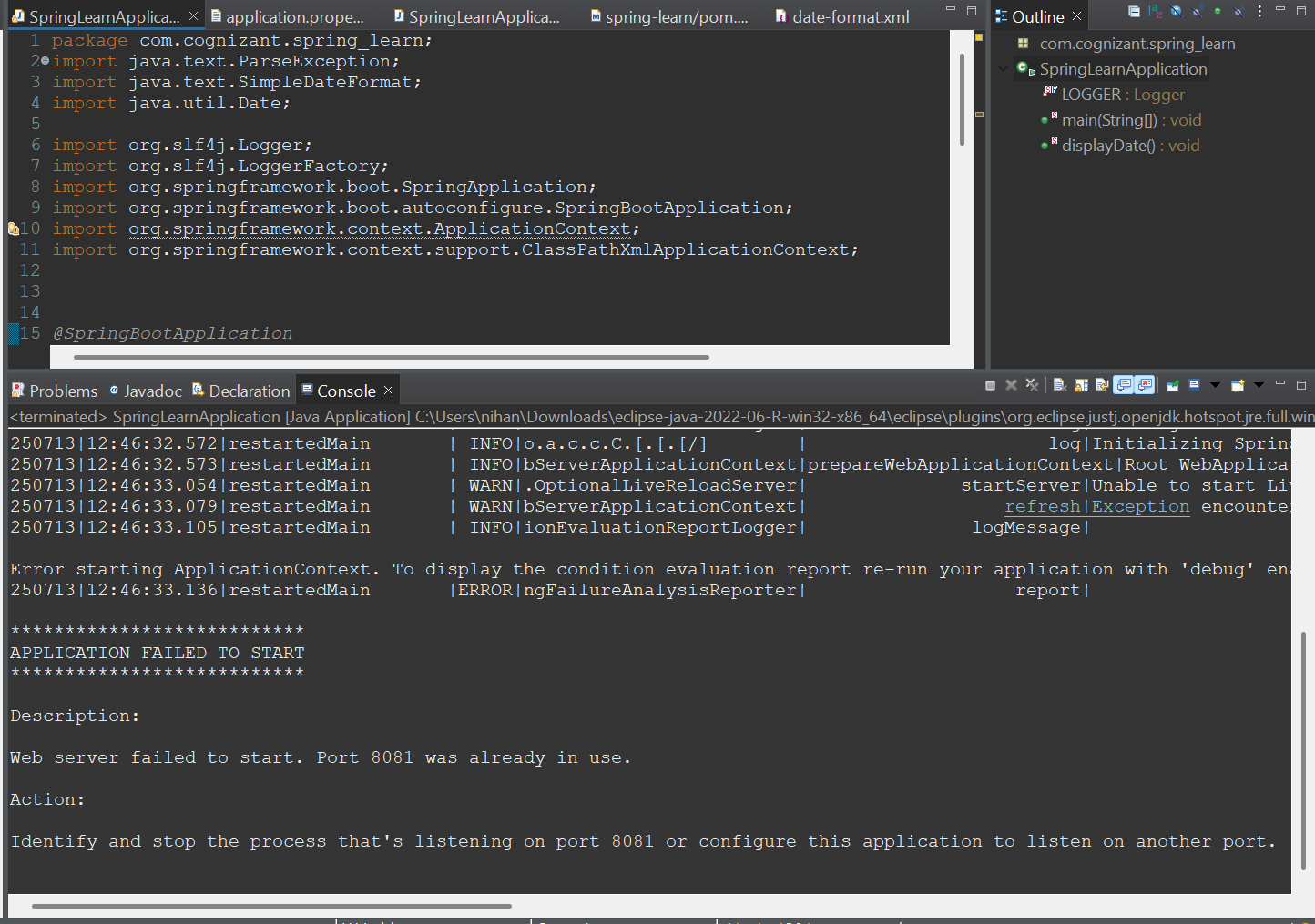
Date date = format.parse("31/12/2018");

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

}

}

**OUTPUT :**



**3.Hello World RESTful Web Service**

Write a REST service in the spring learn application created earlier, that returns the text "Hello World!!" using Spring Web Framework.

**Create the HelloController Class**

**HelloController.java :**

package com.cognizant.springlearn.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.debug("Returning message: {}", message);

LOGGER.info("END");

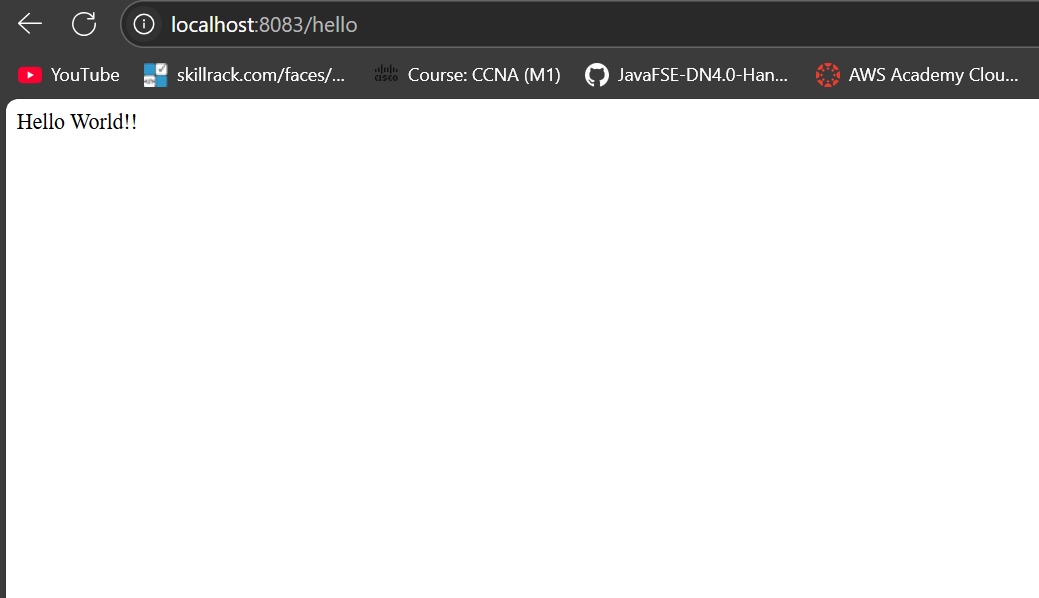
return message;

}

}

**OUTPUT :**

**http://localhost:8083/hello**



**4.REST - Country Web Service**

Write a REST service that returns India country details in the earlier created spring learn application.

**Create XML Bean (spring.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="in" class="com.cognizant.springlearn.model.Country">

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

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

</bean>

</beans>

**Create Country Model Class :**

package com.cognizant.springlearn.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;

}

@Override

public String toString() {

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

}

}

**Create CountryController Class :**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.model.Country;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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");

Country country;

try (ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("spring.xml")) {

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

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

}

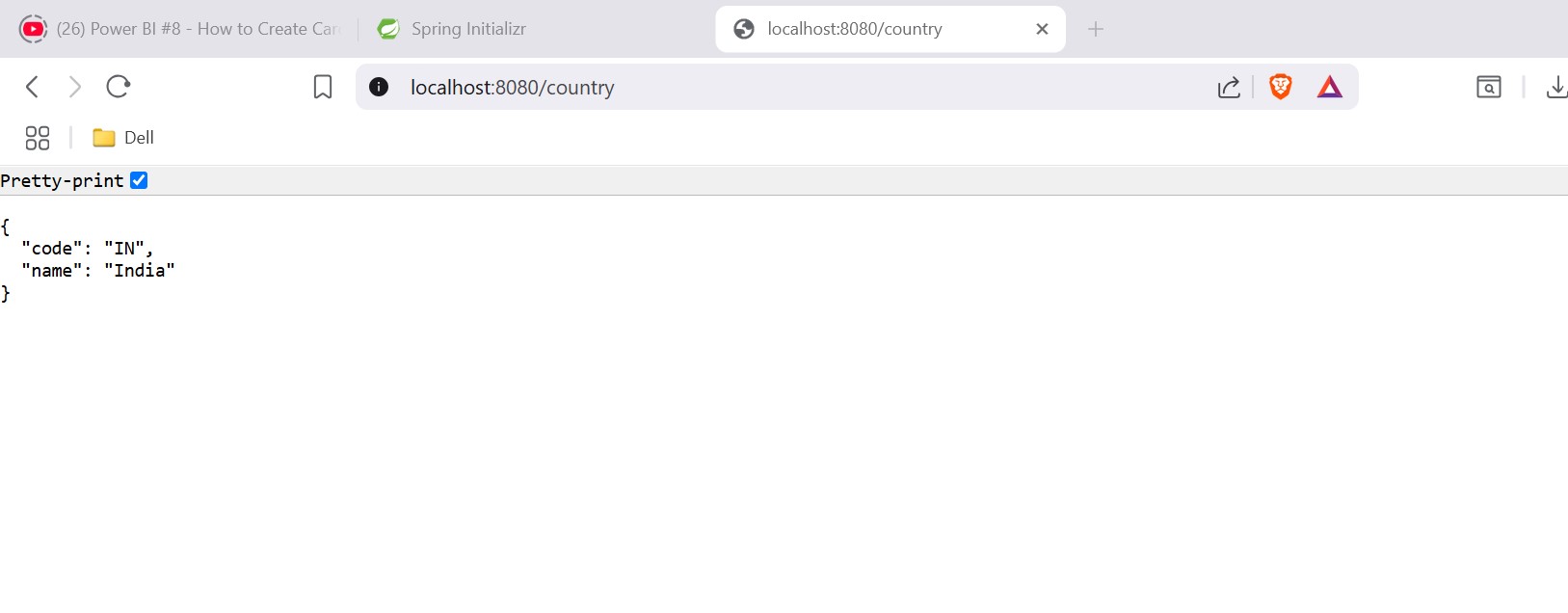
LOGGER.info("END");

return country;

}

}

**OUTPUT :**

****

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

Write a REST service that returns a specific country based on country code. The country code should be case insensitive.

**SpringCountryApiApplication.java :**

package com.cognizant.spring\_country\_api;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

*@SpringBootApplication*

public class SpringCountryApiApplication {

public static void main(String[] args) {

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

}

}

**Country.java:**

package com.cognizant.springcountryapi.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;

}

*@Override*

public String toString() {

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

}

}

**CountryApplication.java:** package com.cognizant.spring\_country\_api.controller;

import com.cognizant.springcountryapi.model.Country;

import com.cognizant.springcountryapi.service.CountryService;

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

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

*@RestController*

public class CountryController {

*@Autowired*

private CountryService countryService;

*@GetMapping*("/countries/{code}")

public Country getCountry(*@PathVariable* String code) {

return countryService.getCountry(code);

}

}

**Countryservice.java:** package com.cognizant.spring country api.service;

import com.cognizant.springcountryapi.model.Country;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Service;

import java.util.List;

*@Service*

public class CountryService {

public Country getCountry(String code) {

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

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

return countryList.stream()

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

.findFirst()

.orElse(null);

}

}

**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.springcountryapi.model.Country"*>

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

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

</**bean**>

<**bean** class=*"com.cognizant.springcountryapi.model.Country"*>

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

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

</**bean**>

<**bean** class=*"com.cognizant.springcountryapi.model.Country"*>

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

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

</**bean**>

<**bean** class=*"com.cognizant.springcountryapi.model.Country"*>

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

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

</**bean**>

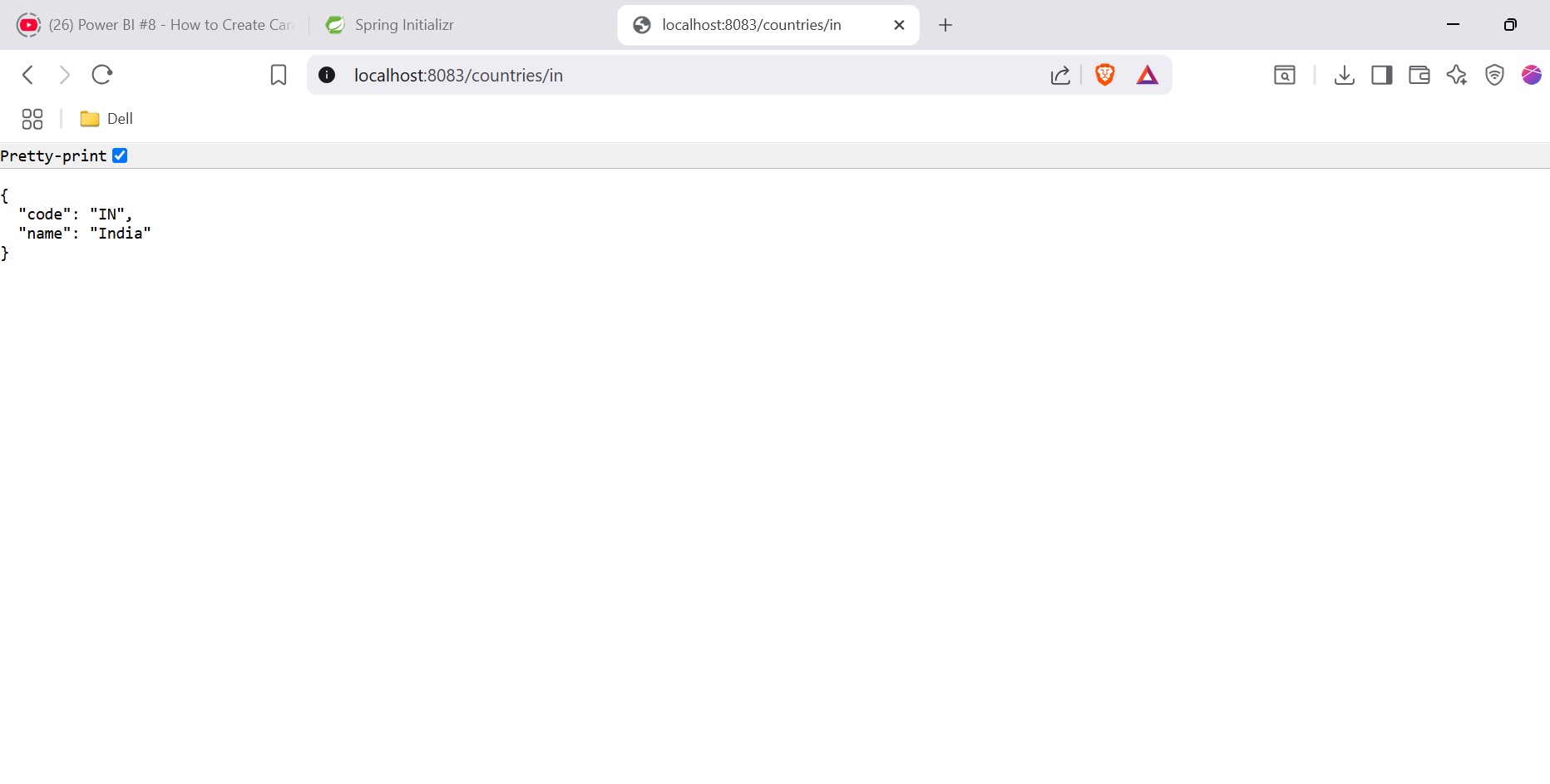
</**list**>

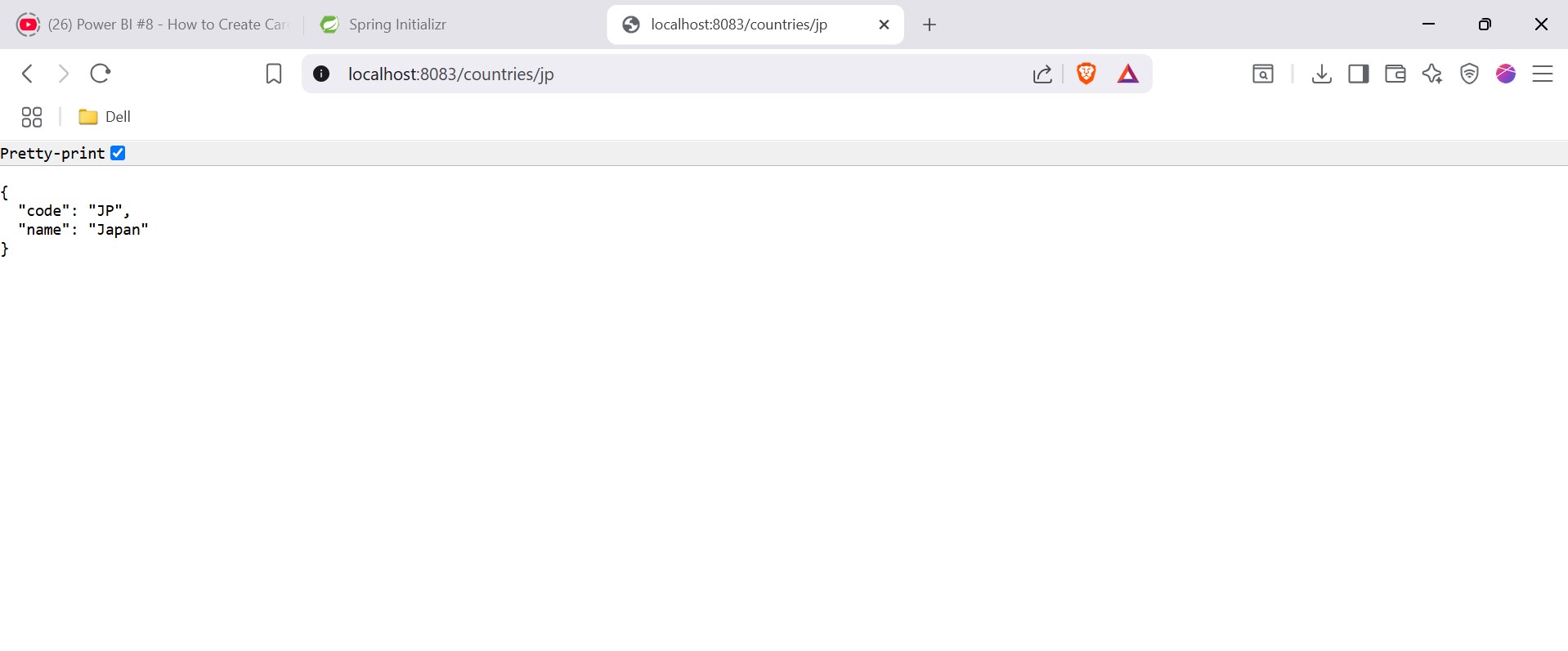
</**constructor-arg**>

</**bean**>

</**beans**>

**OUTPUT :**

****

****

**6.Create authentication service that returns JWT**

As part of first step of JWT process, the user credentials needs to be sent to authentication service request that generates and returns the JWT.

**pom.xml :**

<dependencies>

<!-- Spring Web -->

<dependency>

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

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

</dependency>

<!-- Spring Security -->

<dependency>

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

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

</dependency>

<!-- JWT Library -->

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt</artifactId>

<version>0.9.1</version>

</dependency>

</dependencies>

**JwtUtil.java :**

package com.cognizant.springlearn.util;

import java.util.Date;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

public class JwtUtil {

private static final String SECRET\_KEY = "your-secret-key";

public static String generateToken(String username) {

return Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date())

.setExpiration(new Date(System.currentTimeMillis() + 10 \* 60 \* 1000)) // 10 minutes

.signWith(SignatureAlgorithm.HS256, SECRET\_KEY)

.compact();

}

}

**AuthenticationController.java :**

package com.cognizant.springlearn.controller;

import java.util.Base64;

import javax.servlet.http.HttpServletRequest;

import com.cognizant.springlearn.util.JwtUtil;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.http.ResponseEntity;

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

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

@RestController

public class AuthenticationController {

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

@GetMapping("/authenticate")

public ResponseEntity<?> authenticate(HttpServletRequest request) {

LOGGER.info("START /authenticate");

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

if (authHeader == null || !authHeader.startsWith("Basic ")) {

return ResponseEntity.status(401).body("Missing or invalid Authorization header");

}

// Decode Base64

String[] credentials = new String(Base64.getDecoder().decode(authHeader.substring(6))).split(":");

String username = credentials[0];

String password = credentials[1];

LOGGER.debug("Received credentials: {} / {}", username, password);

// Simple validation (normally check against DB or service)

if (username.equals("user") && password.equals("pwd")) {

String token = JwtUtil.generateToken(username);

LOGGER.info("END /authenticate");

return ResponseEntity.ok().body("{\"token\":\"" + token + "\"}");

} else {

return ResponseEntity.status(403).body("Invalid credentials");

}

}

}

**SecurityConfig.java :**

package com.cognizant.springlearn.config;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.Customizer;

import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import org.springframework.security.web.SecurityFilterChain;

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

@Configuration

@EnableWebSecurity

public class SecurityConfig {

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http

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

.authorizeHttpRequests(authz -> authz

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

.anyRequest().authenticated()

)

.httpBasic(Customizer.withDefaults());

return http.build();

}

}