**Spring REST using Spring Boot 3**

**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>24</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.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

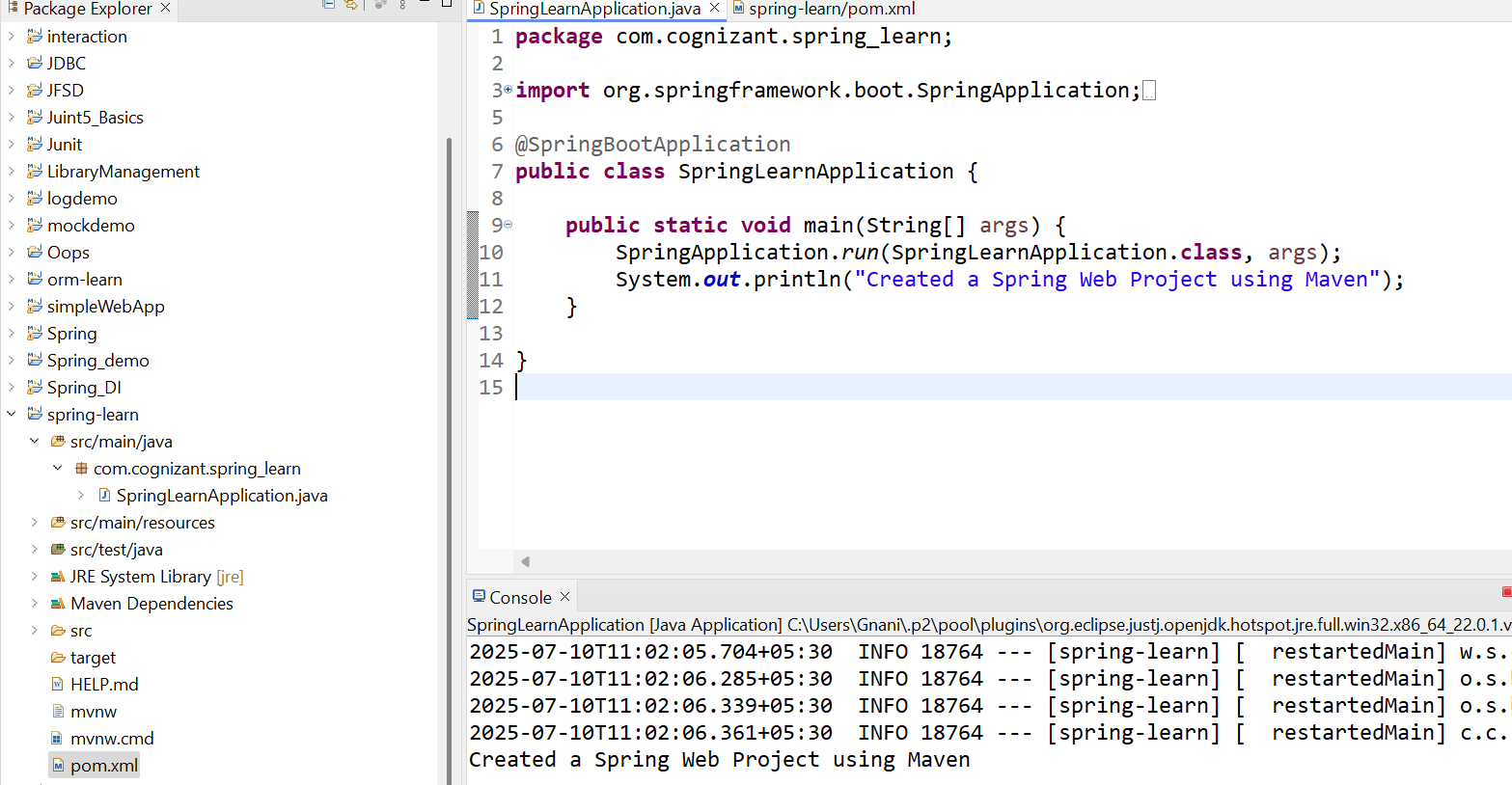
SpringApplication.run(SpringLearnApplication.class, args);

System.out.println("Created a Spring Web Project using Maven");

}

}

**Output:**



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

**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() {

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 + "]";

}

}

**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=*"country"* class=*"com.cognizant.spring\_learn.Country"*>

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

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

</bean>

</beans>

**application.properties:**

spring.application.name=spring-learn

logging.level.org.springframework=info

logging.level.com.cognizant.spring\_learn=debug

logging.pattern.console=%d{yyMMdd}|%d{HH:mm:ss.SSS}|%-20.20thread|%5p|%-25.25logger**{25}**|%25M|%m%n

**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);

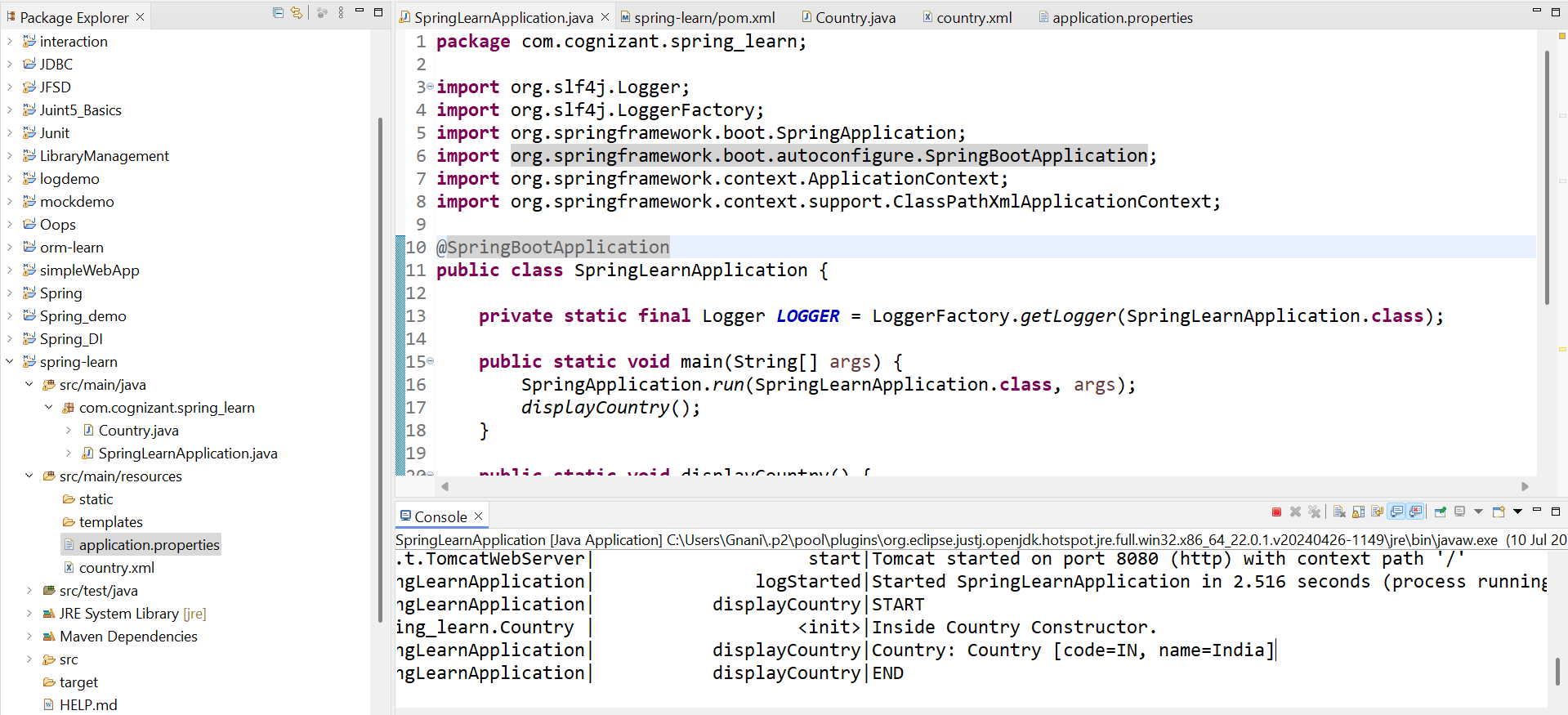
*LOGGER*.debug("Country: {}", country.toString());

*LOGGER*.info("END");

}

}

**Output:**

****

**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*.debug("Returning: {}", message);

*LOGGER*.info("END");

return message;

}

}

**SpringLearnApplication.java:**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) {

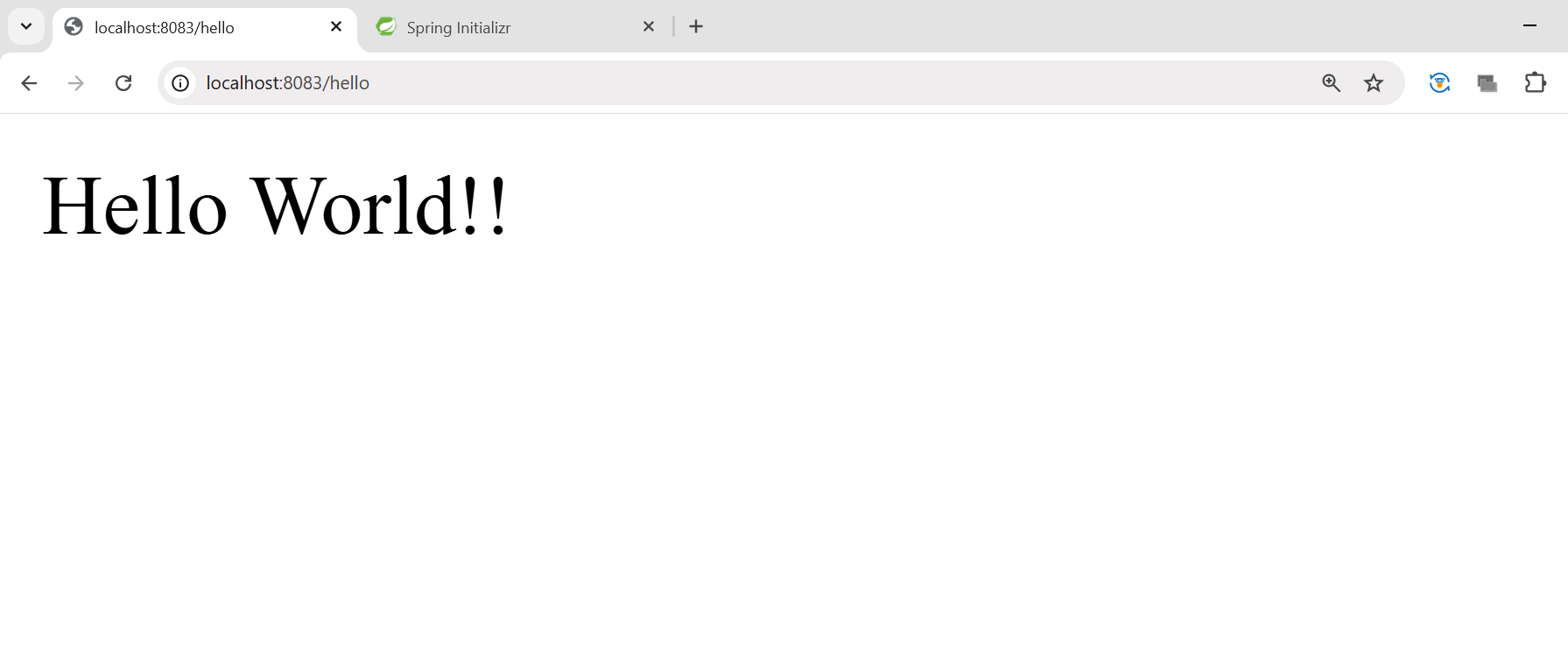
*LOGGER*.info("START");

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

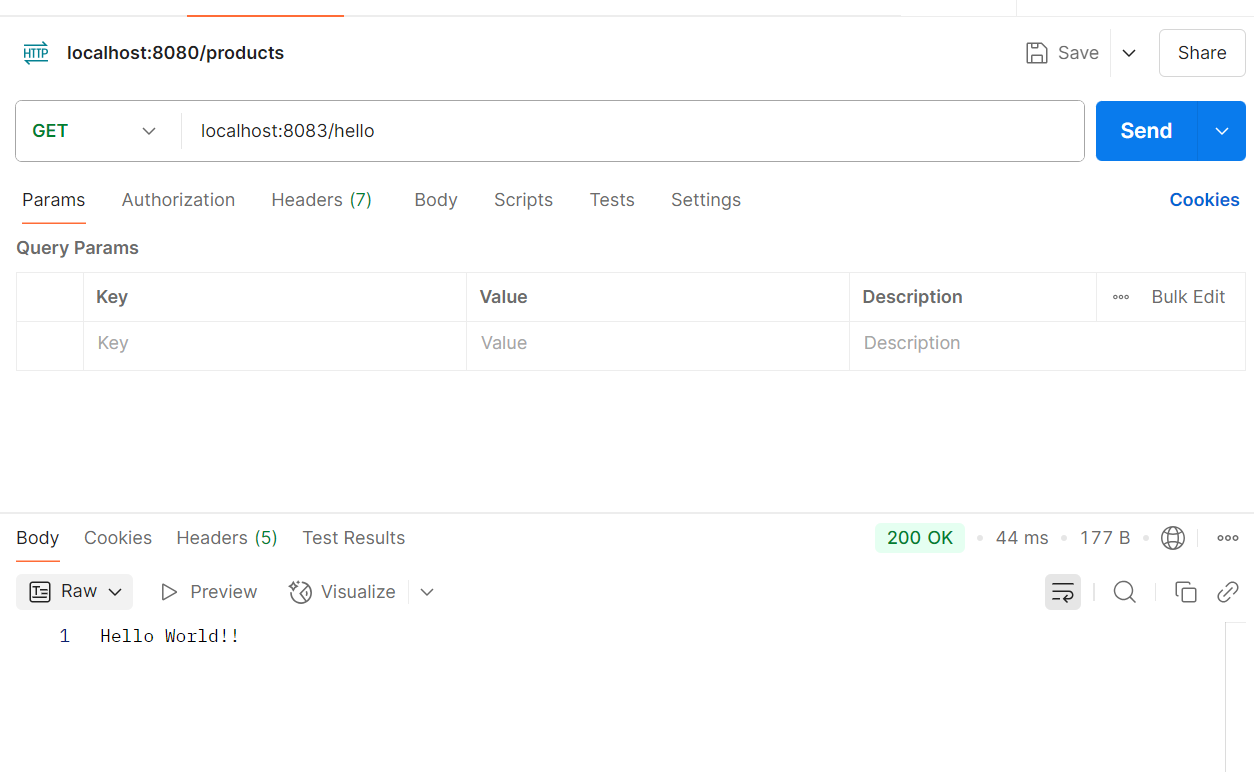
*LOGGER*.info("END");

} }

**Output in Browser:**

****

**Output in Postman:**

****

**REST - Country Web Service**

**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("country", Country.class);

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

*LOGGER*.info("END");

return country;

}

}

**SpringLearnApplication.java:**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) {

*LOGGER*.info("START");

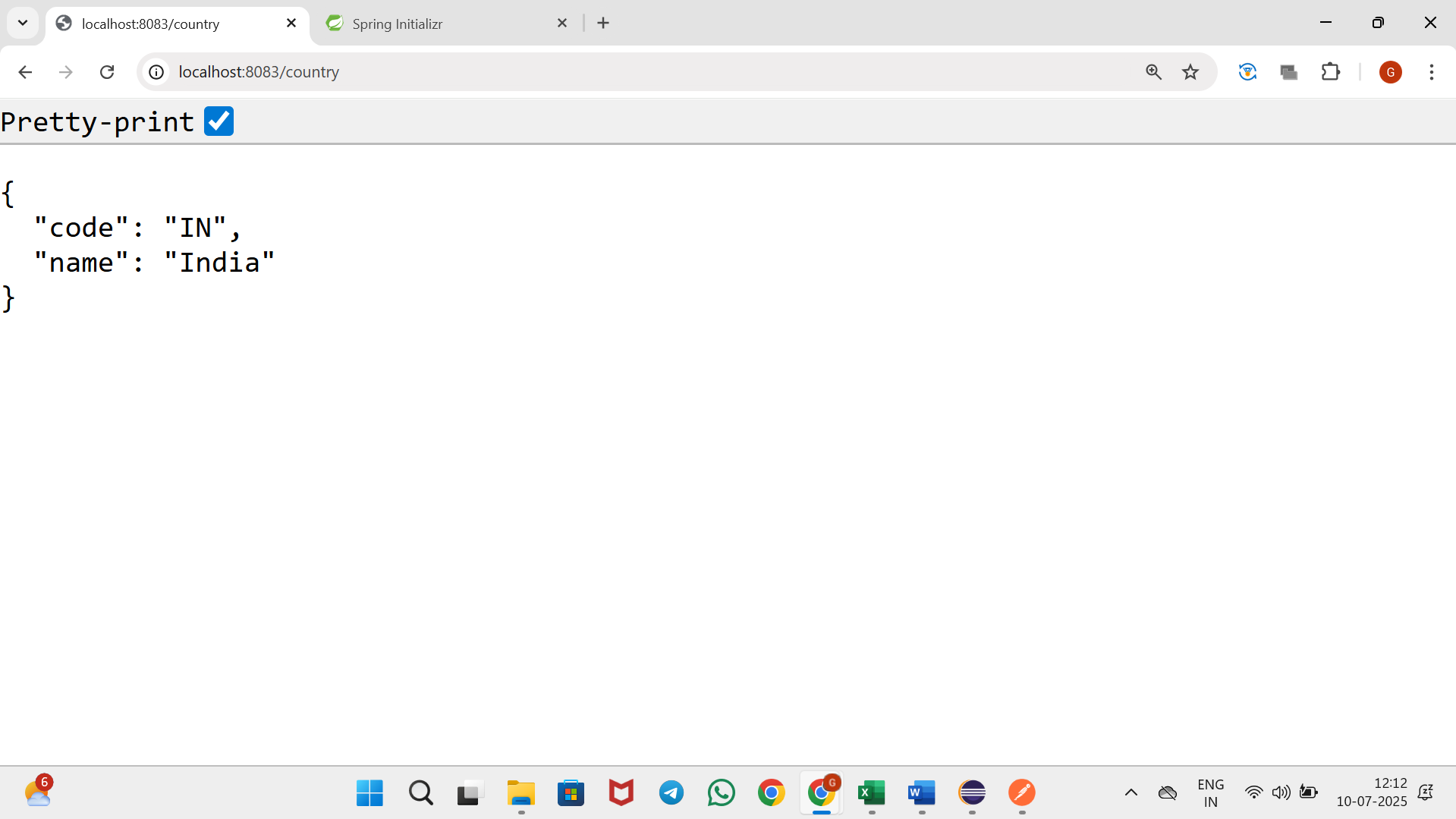
SpringApplication.*run*(SpringLearnApplication.class, args);

*LOGGER*.info("END");

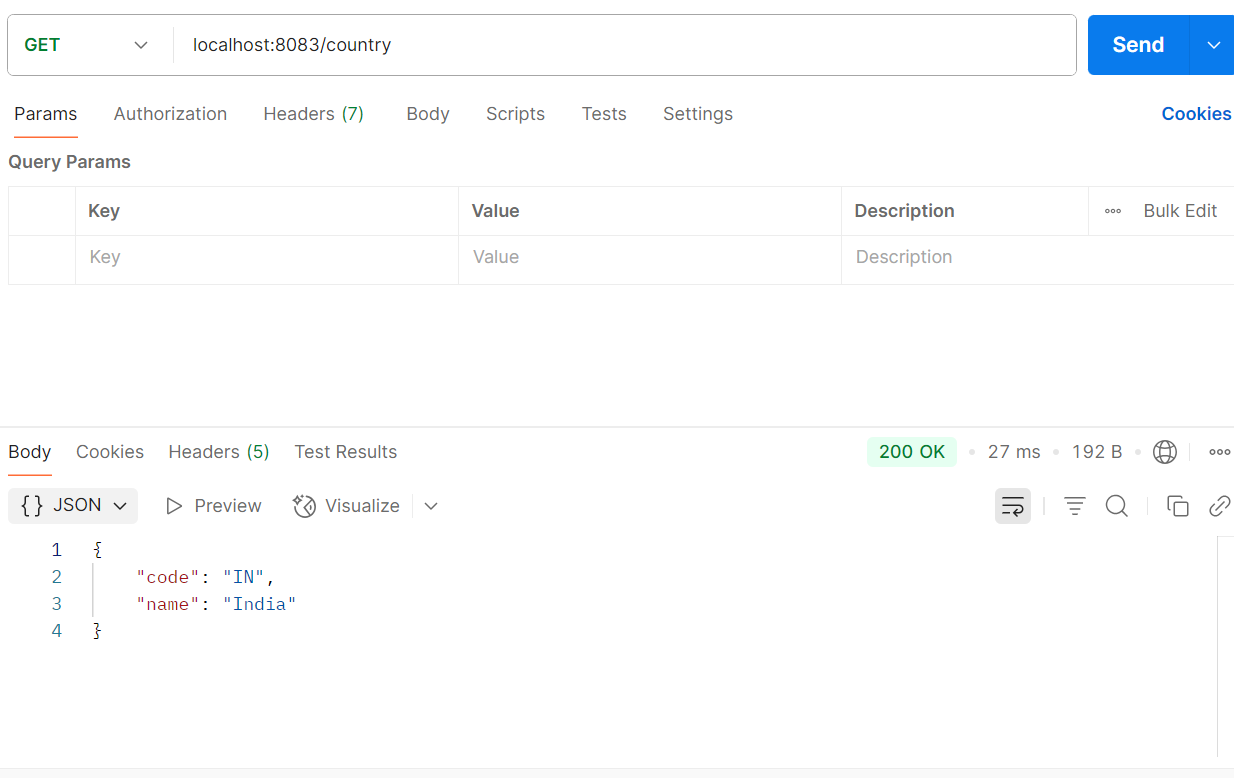
}

}

**Output in Browser:**

****

**Output in Postman:**

****

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

**CountryController.java:**

package com.cognizant.spring\_learn.Controller;

import com.cognizant.spring\_learn.Country;

import com.cognizant.spring\_learn.Service.CountryService;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

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

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

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

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

@RestController

public class CountryController {

@Autowired

private CountryService countryService;

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("country", Country.class);

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

*LOGGER*.info("END");

return country;

}

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

public Country getCountry(@PathVariable String code) {

*LOGGER*.info("START");

Country country = countryService.getCountry(code);

*LOGGER*.info("END");

return country;

}

}

**CountryService.java:**

package com.cognizant.spring\_learn.Service;

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.stereotype.Service;

import java.util.List;

@Service

public class CountryService {

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

public Country getCountry(String code) {

*LOGGER*.info("START");

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

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

Country result = countryList.stream()

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

.findFirst()

.orElse(null);

*LOGGER*.debug("Returning country: {}", result);

*LOGGER*.info("END");

return result;

}

}

<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.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>

<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>

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

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

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

</bean>

</beans>

**SpringLearnApplication.java:**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) {

*LOGGER*.info("START");

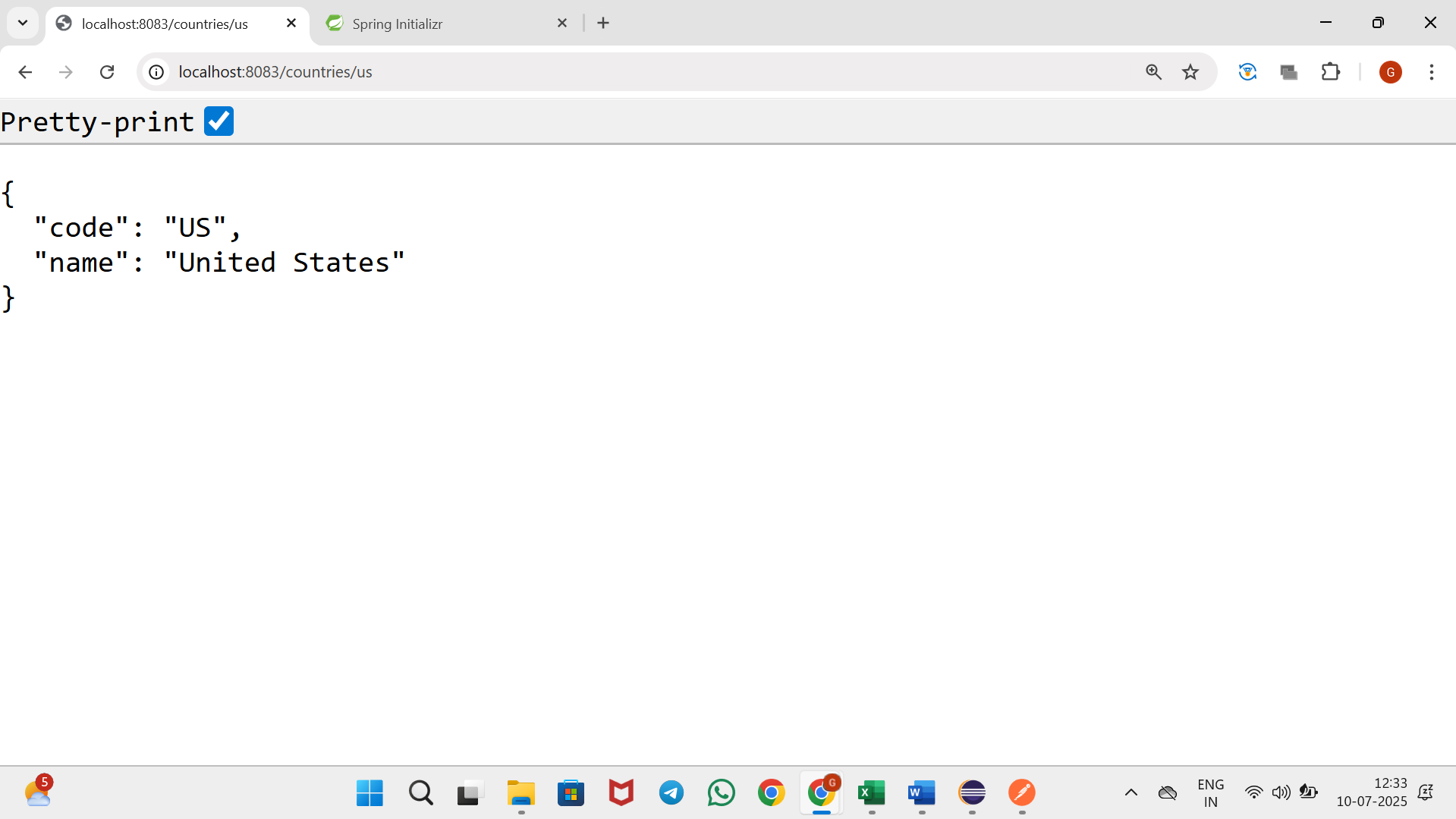
SpringApplication.*run*(SpringLearnApplication.class, args);

*LOGGER*.info("END");

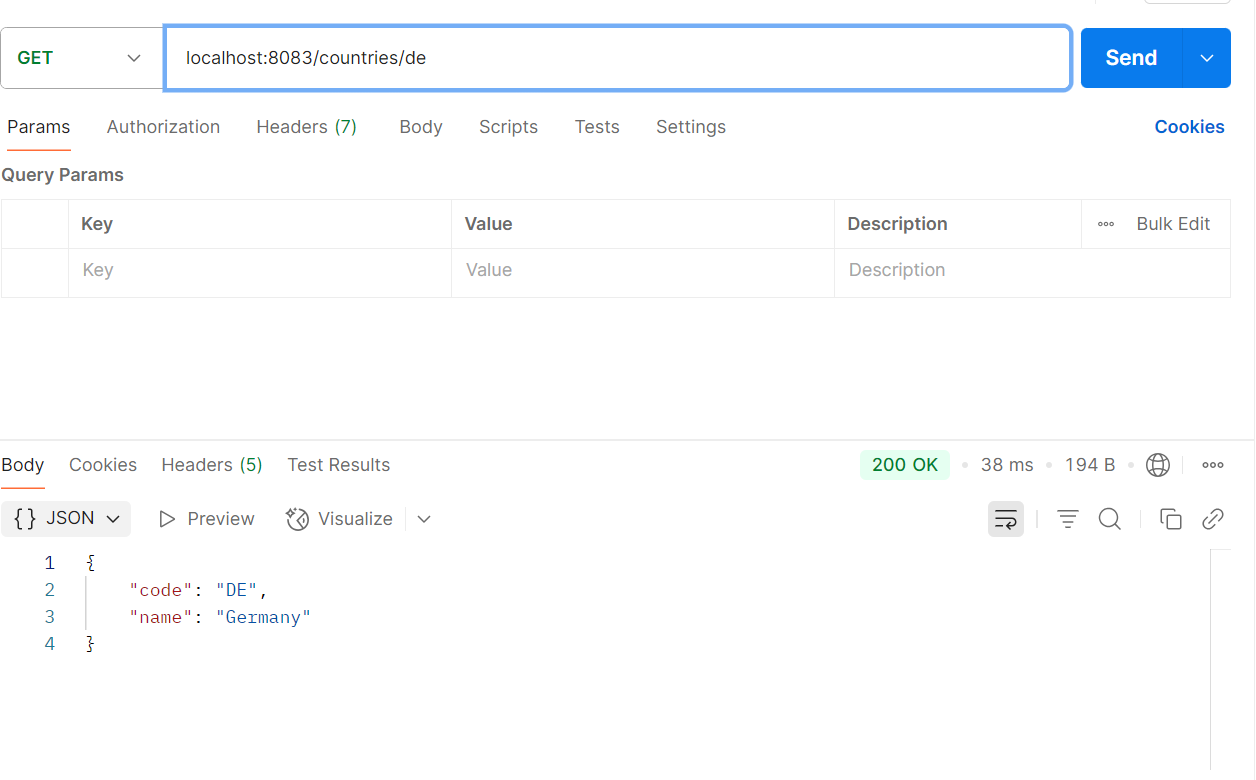
}

}

**Output in Browser:**



**Output in Postman:**

****

**Create authentication service that returns JWT**

**JwtFilter.java:**

package com.example.jwtauth.filter;

import com.example.jwtauth.util.JwtUtil;

import jakarta.servlet.\*;

import jakarta.servlet.http.HttpServletRequest;

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

import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;

import org.springframework.security.core.context.SecurityContextHolder;

import org.springframework.stereotype.Component;

import org.springframework.web.filter.GenericFilterBean;

import java.io.IOException;

import java.util.Collections;

@Component

public class JwtFilter extends GenericFilterBean {

@Autowired

private JwtUtil jwtUtil;

@Override

public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)

throws IOException, ServletException {

HttpServletRequest httpRequest = (HttpServletRequest) request;

System.*out*.println("Incoming URI: " + httpRequest.getRequestURI());

if (httpRequest.getRequestURI().startsWith("/authenticate")) {

chain.doFilter(request, response);

return; }

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

if (header != null && header.startsWith("Bearer ")) {

String token = header.substring(7);

if (jwtUtil.validateToken(token)) {

String username = jwtUtil.extractUsername(token);

UsernamePasswordAuthenticationToken auth =

new UsernamePasswordAuthenticationToken(username, null, Collections.*emptyList*());

SecurityContextHolder.*getContext*().setAuthentication(auth);

}

}

chain.doFilter(request, response);

}

}

**JwtUtil.java:**

package com.example.jwtauth.util;

import io.jsonwebtoken.\*;

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 SECRET\_KEY = Keys.*hmacShaKeyFor*("this\_is\_a\_very\_secure\_secret\_key\_123456".getBytes());

public String generateToken(String username) {

return Jwts.*builder*()

.setSubject(username)

.setIssuedAt(new Date())

.setExpiration(new Date(System.*currentTimeMillis*() + 1000 \* 60 \* 20))

.signWith(SECRET\_KEY, SignatureAlgorithm.*HS256*)

.compact();

}

public String extractUsername(String token) {

return Jwts.*parserBuilder*()

.setSigningKey(SECRET\_KEY)

.build()

.parseClaimsJws(token)

.getBody()

.getSubject();

}

public boolean validateToken(String token) {

try { Jwts.*parserBuilder*().setSigningKey(SECRET\_KEY).build().parseClaimsJws(token);

return true;

} catch (JwtException e) {

return false;

}

} }

**AuthenticationController.java:**

package com.example.jwtauth.controller;

import com.example.jwtauth.util.JwtUtil;

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

import org.springframework.security.authentication.\*;

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

import java.util.\*;

@RestController

public class AuthenticationController {

@Autowired

private AuthenticationManager authenticationManager;

@Autowired

private JwtUtil jwtUtil;

@GetMapping("/authenticate")

public Map<String, String> generateToken(@RequestHeader("Authorization") String authHeader) {

String base64Credentials = authHeader.substring("Basic ".length());

String decoded = new String(Base64.getDecoder().decode(base64Credentials));

String[] parts = decoded.split(":");

authenticationManager.authenticate(

new UsernamePasswordAuthenticationToken(parts[0], parts[1])

);

String token = jwtUtil.generateToken(parts[0]);

Map<String, String> map = new HashMap<>();

map.put("token", token);

return map;

}

@GetMapping("/hello")

public String hello() {

return "Hello! You are authenticated!";

}

}

**SecurityConfig.java:**

package com.example.jwtauth.config;

import com.example.jwtauth.filter.JwtFilter;

import jakarta.servlet.Filter;

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.configuration.AuthenticationConfiguration;

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

import org.springframework.security.config.http.SessionCreationPolicy;

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

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

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

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

import org.springframework.security.provisioning.InMemoryUserDetailsManager;

import org.springframework.security.web.SecurityFilterChain;

import org.springframework.security.web.authentication.UsernamePasswordAuthenticationFilter;

@Configuration

public class SecurityConfig {

@Bean

public PasswordEncoder passwordEncoder() {

return new BCryptPasswordEncoder();

}

@Bean

public AuthenticationManager authenticationManager(AuthenticationConfiguration config) throws Exception {

return config.getAuthenticationManager();

}

@Bean

public InMemoryUserDetailsManager userDetailsService(PasswordEncoder encoder) {

UserDetails user = User.builder()

.username("user")

.password(encoder.encode("pwd"))

.roles("USER")

.build();

return new InMemoryUserDetailsManager(user);

}

@Bean

public SecurityFilterChain filterChain(HttpSecurity http, JwtFilter jwtFilter) throws Exception {

return http

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

.authorizeHttpRequests(auth -> auth

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

.anyRequest().authenticated()

)

.sessionManagement(sess -> sess.sessionCreationPolicy(SessionCreationPolicy.STATELESS))

.addFilterBefore(jwtFilter, UsernamePasswordAuthenticationFilter.class)

.build();

}

}

**JwtAuthApplication.java:**

package com.example.jwtauth;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class JwtAuthApplication {

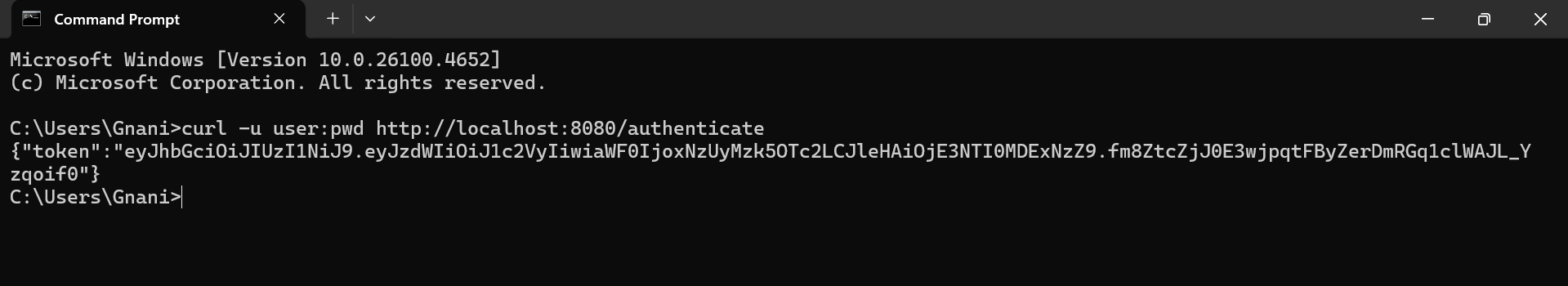
public static void main(String[] args) {

SpringApplication.run(JwtAuthApplication.class, args);

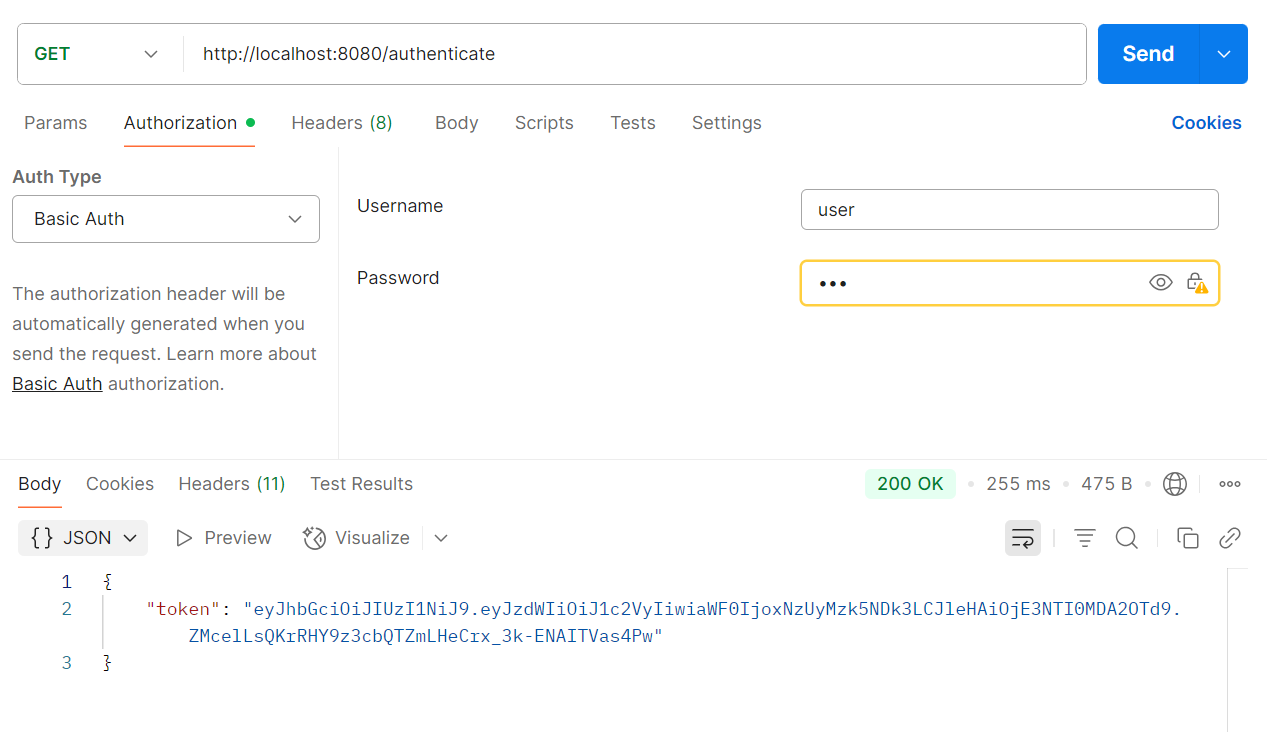
}

}

**Output:**



**Output in Postman:**

****