## WEEK 4 HANDS\_ON EXERCISES

**EXERCISE 1:** Create a Spring Web Project using Maven

**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;

@SpringBootApplication

public class SpringLearnApplication {

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

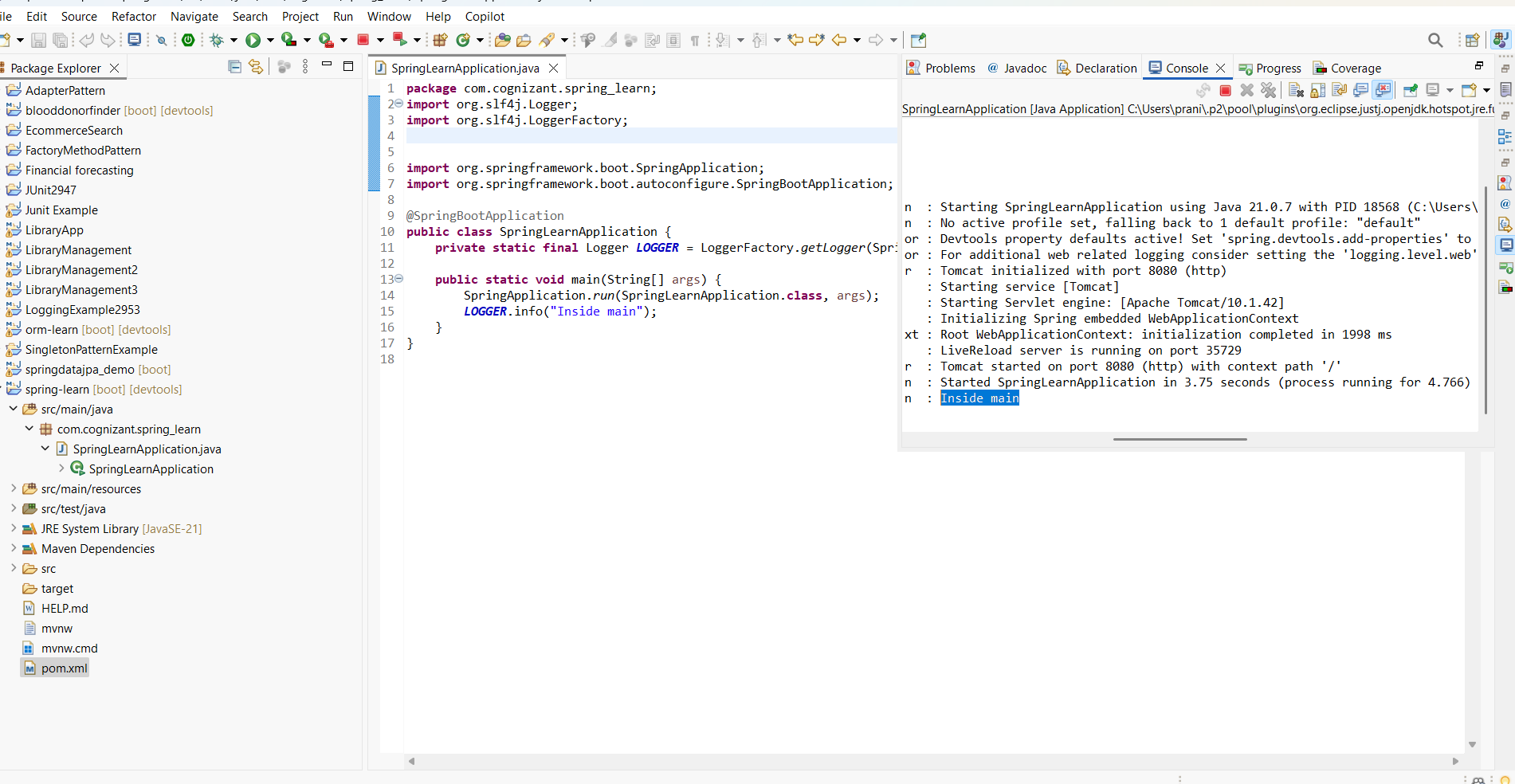
public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

LOGGER.info("Inside main");

}

}

**Output:**

**Exercise 2:** Spring Core – Load Country from Spring Configuration XML

**Country.java:**

package com.cognizant.springcore;

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 + "]";

}

}

**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.springcore.Country">

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

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

</bean>

</beans>

**SpringLearnApplication.java:**

**package** com.cognizant.springcore;

//import org.slf4j.Logger;

//import org.slf4j.LoggerFactory;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**public** **class** SpringCoreApplication {

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

**public** **static** **void** main(String[] args) {

System.***out***.println("START");

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

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

System.***out***.println("Country: " + country);

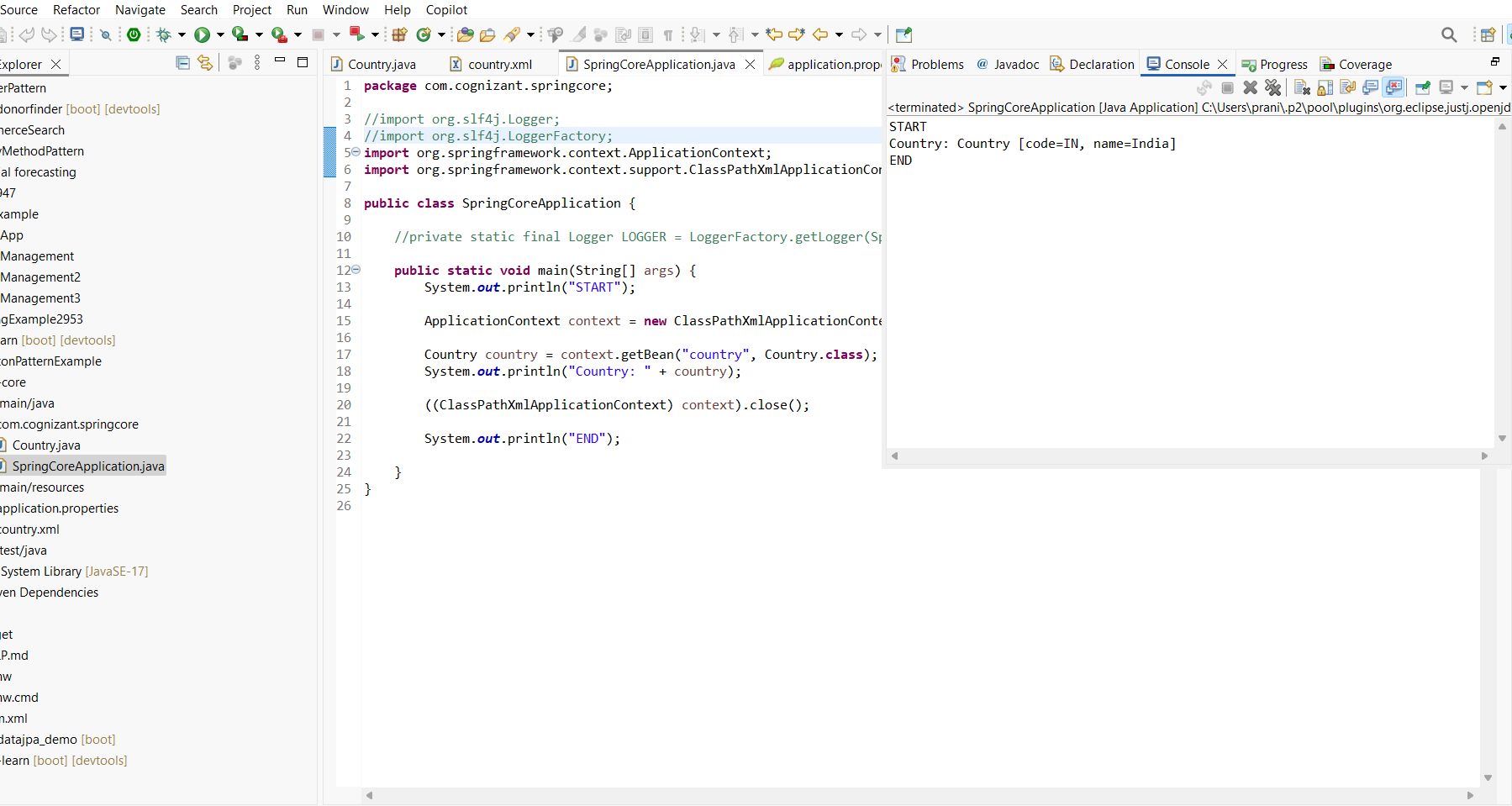
((ClassPathXmlApplicationContext) context).close();

System.***out***.println("END");

}

}

**Output:**



**Exercise 3:** Hello World RESTful Web Service

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

**return** message;

}

}

**SpringLearnApplication.java:**

**package** com.cognizant.springlearn;

**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);

}

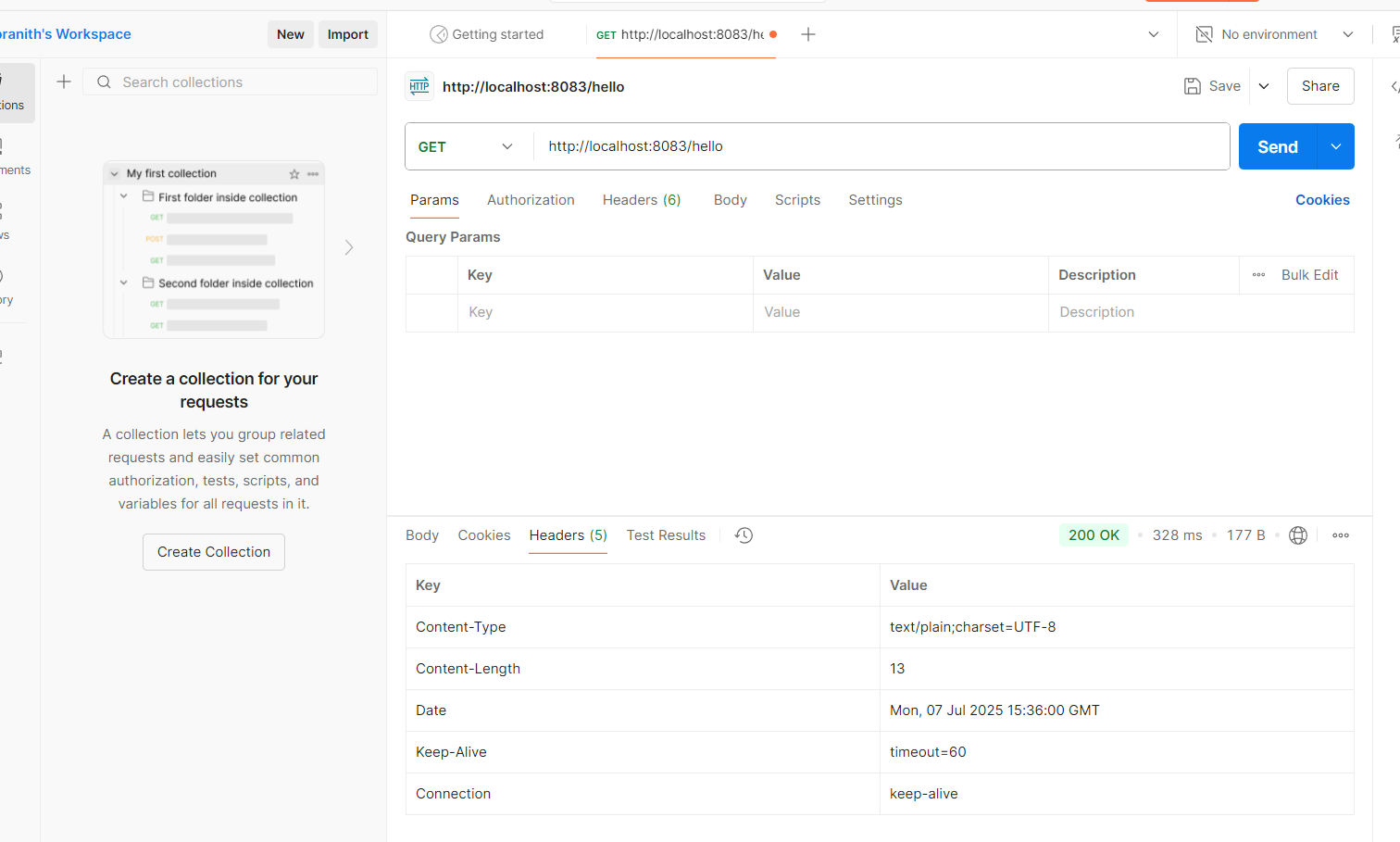
}

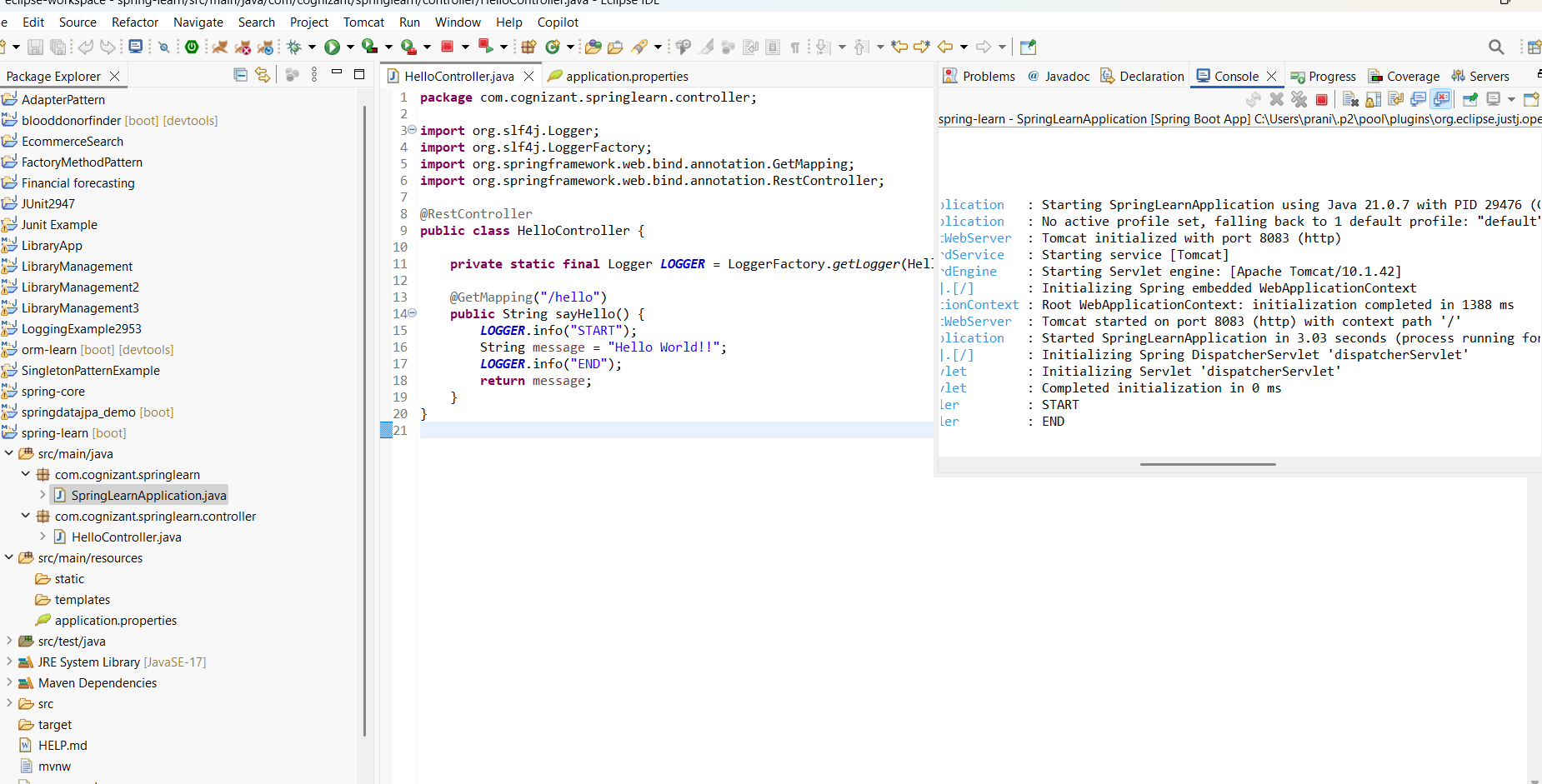
**Output:**

**Network Tab:**



**Postman:**

****

**Exercise 4:** REST - Country Web Service

**Country.java:**

**package** com.example.countryrest;

**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;

}

}

**CountryController.java:**

**package** com.example.countryrest;

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

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

@RestController

**public** **class** CountryController {

@GetMapping("/country")

**public** Country getCountryIndia() {

**return** **new** Country("IN", "India");

}

}

**CountryrestApplication.java:**

**package** com.example.countryrest;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** CountryrestApplication {

**public** **static** **void** main(String[] args) {

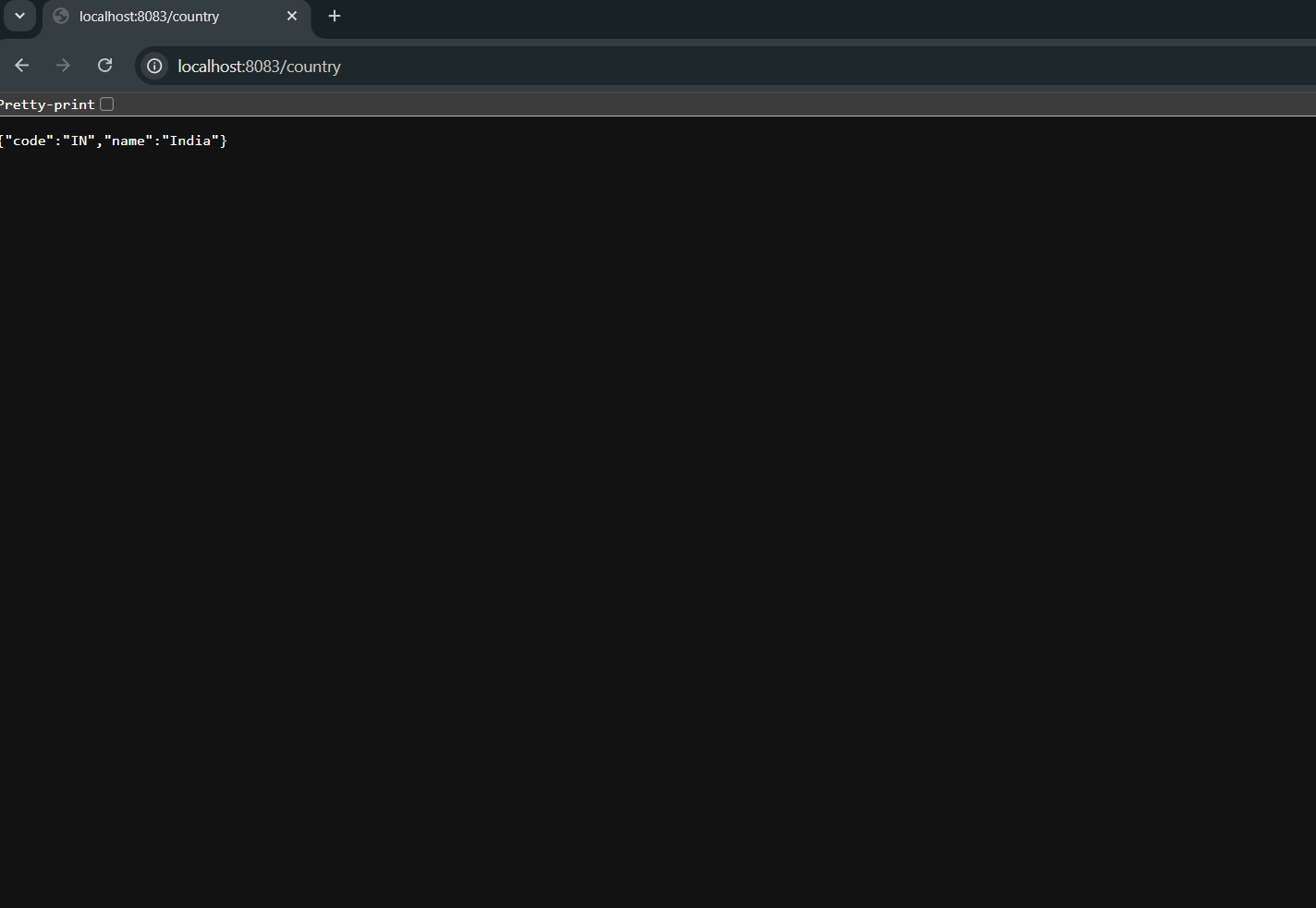
SpringApplication.*run*(CountryrestApplication.**class**, args);

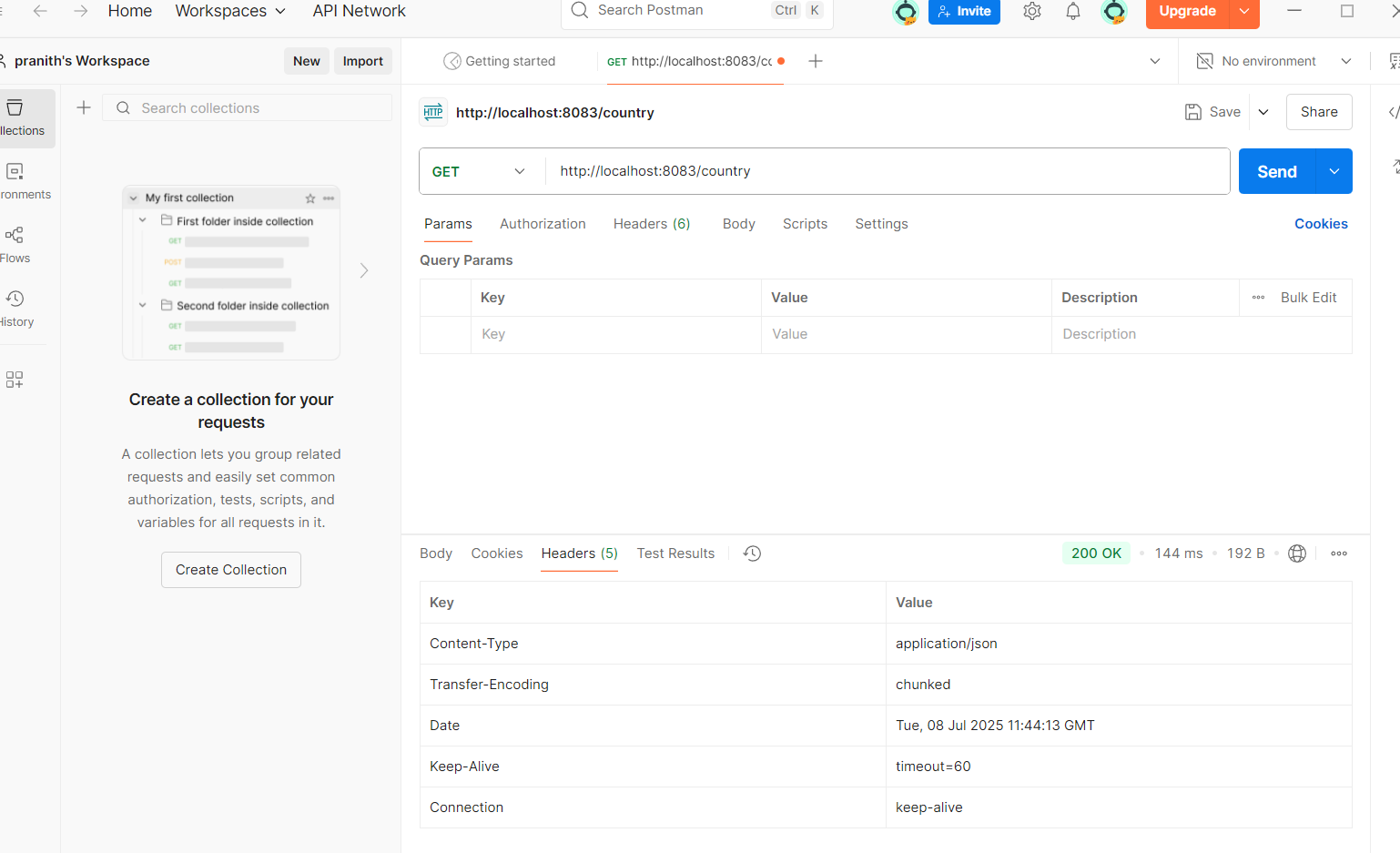
}

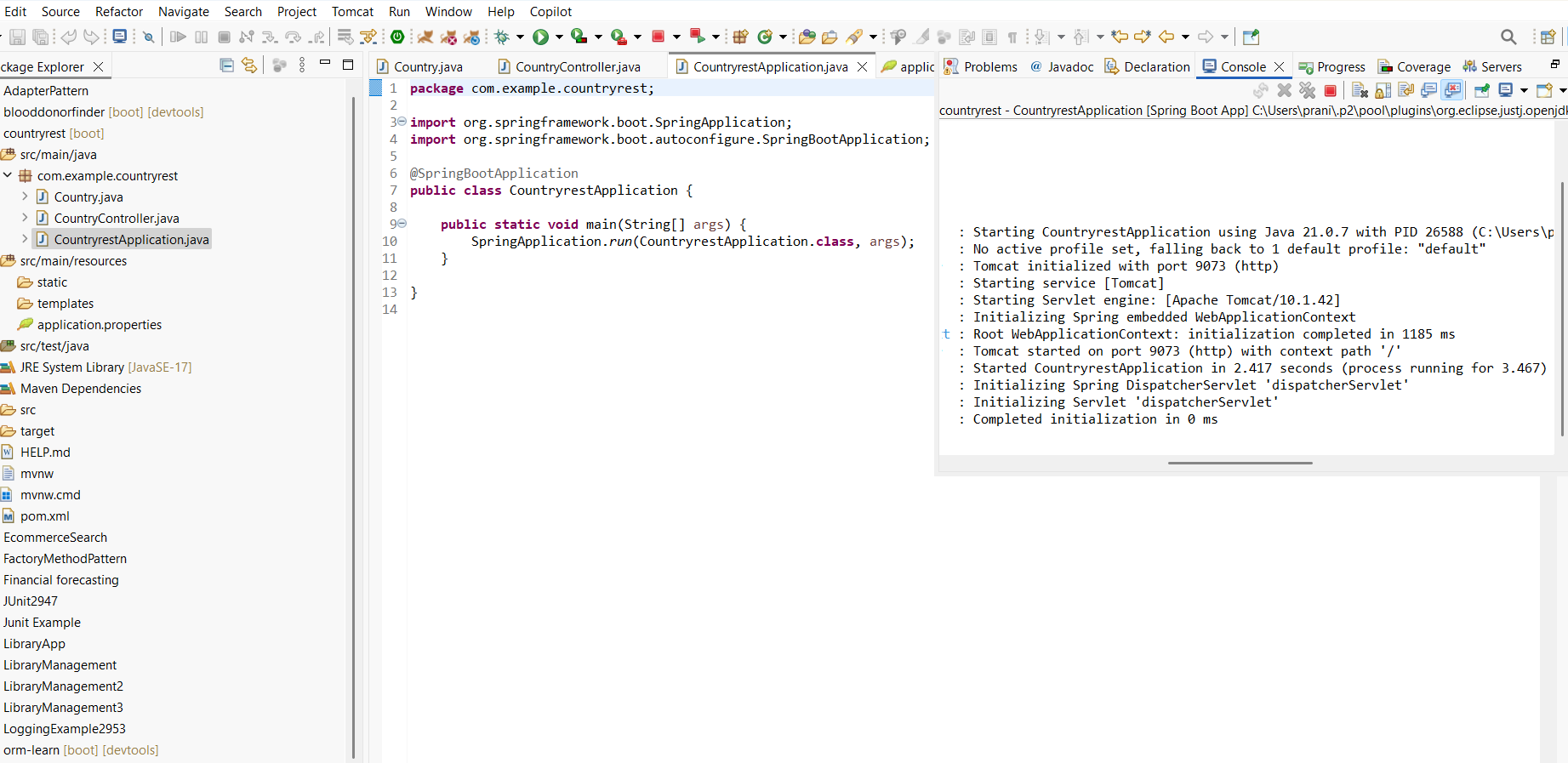
}

**Output:**

**Network Tab:**



**Postman:**



**Exercise 5: REST - Get country based on country code**

**Country.java:**

**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;

}

}

**CountryController.java:**

**package** com.cognizant.springlearn.controller;

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

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

**import** com.cognizant.springlearn.model.Country;

**import** com.cognizant.springlearn.service.CountryService;

@RestController

**public** **class** CountryController {

@Autowired

**private** CountryService countryService;

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

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

**return** countryService.getCountry(code);

}

}

**CountryService.java:**

**package** com.cognizant.springlearn.service;

**import** java.util.List;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**import** org.springframework.stereotype.Service;

**import** com.cognizant.springlearn.model.Country;

@Service

**public** **class** CountryService {

**public** Country getCountry(String code) **throws** Exception {

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

@SuppressWarnings("unchecked")

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

Country country = countryList.stream()

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

.findFirst()

.orElseThrow(() -> **new** Exception("Country Not Found"));

context.close();

**return** country;

}

}

**SpringlearnApllication.java:**

**package** com.cognizant.springlearn;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** SpringLearnApplication {

**public** **static** **void** main(String[] args) {

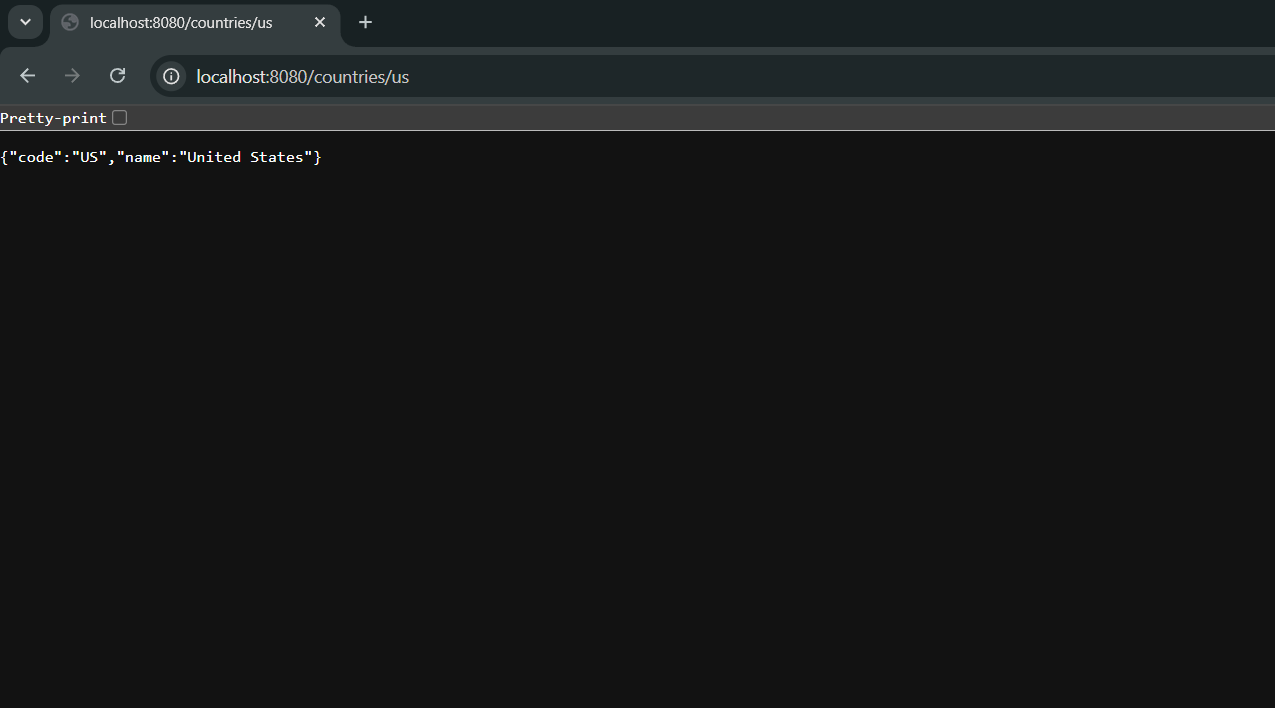
SpringApplication.*run*(SpringLearnApplication.**class**, args);

}

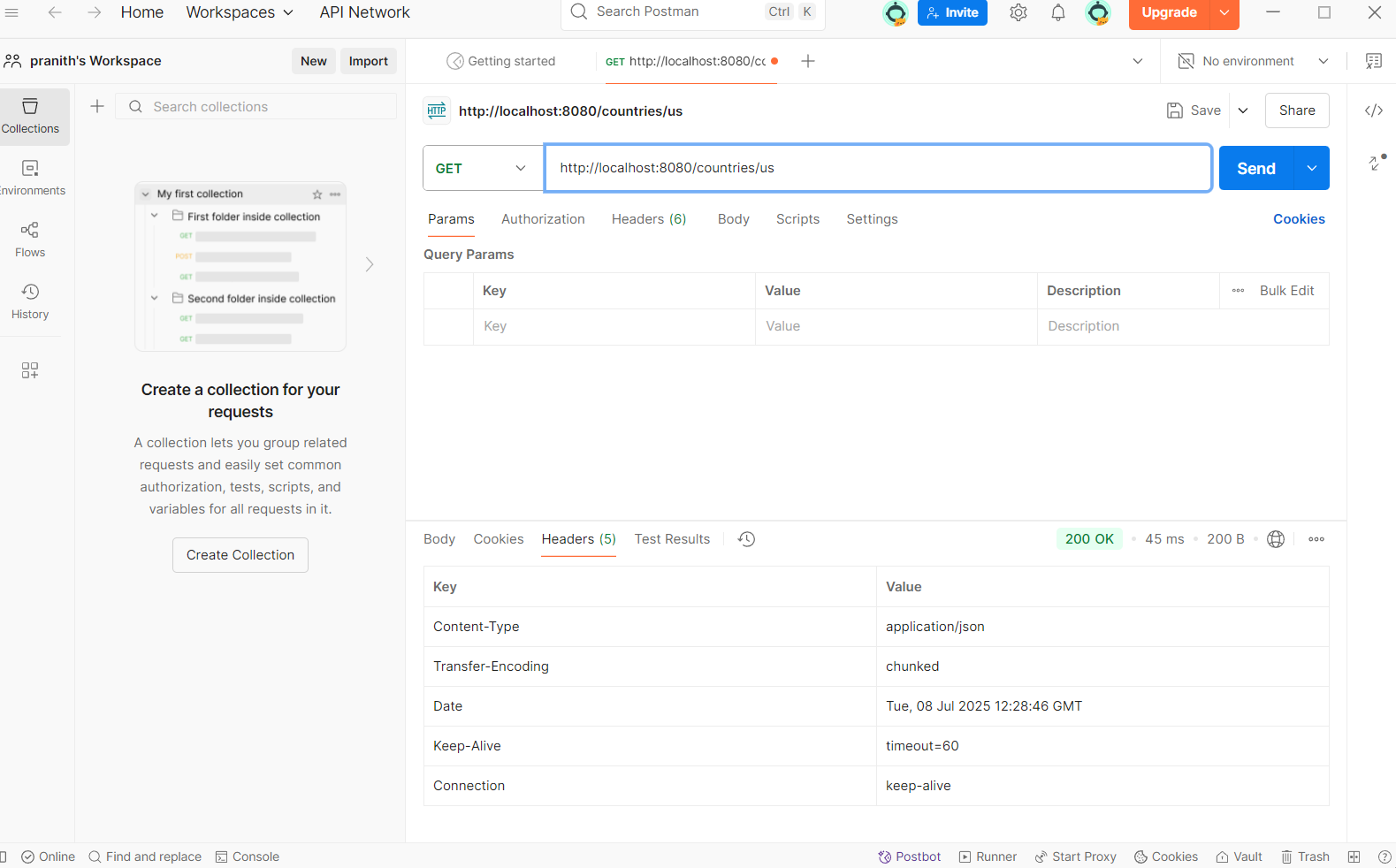
}

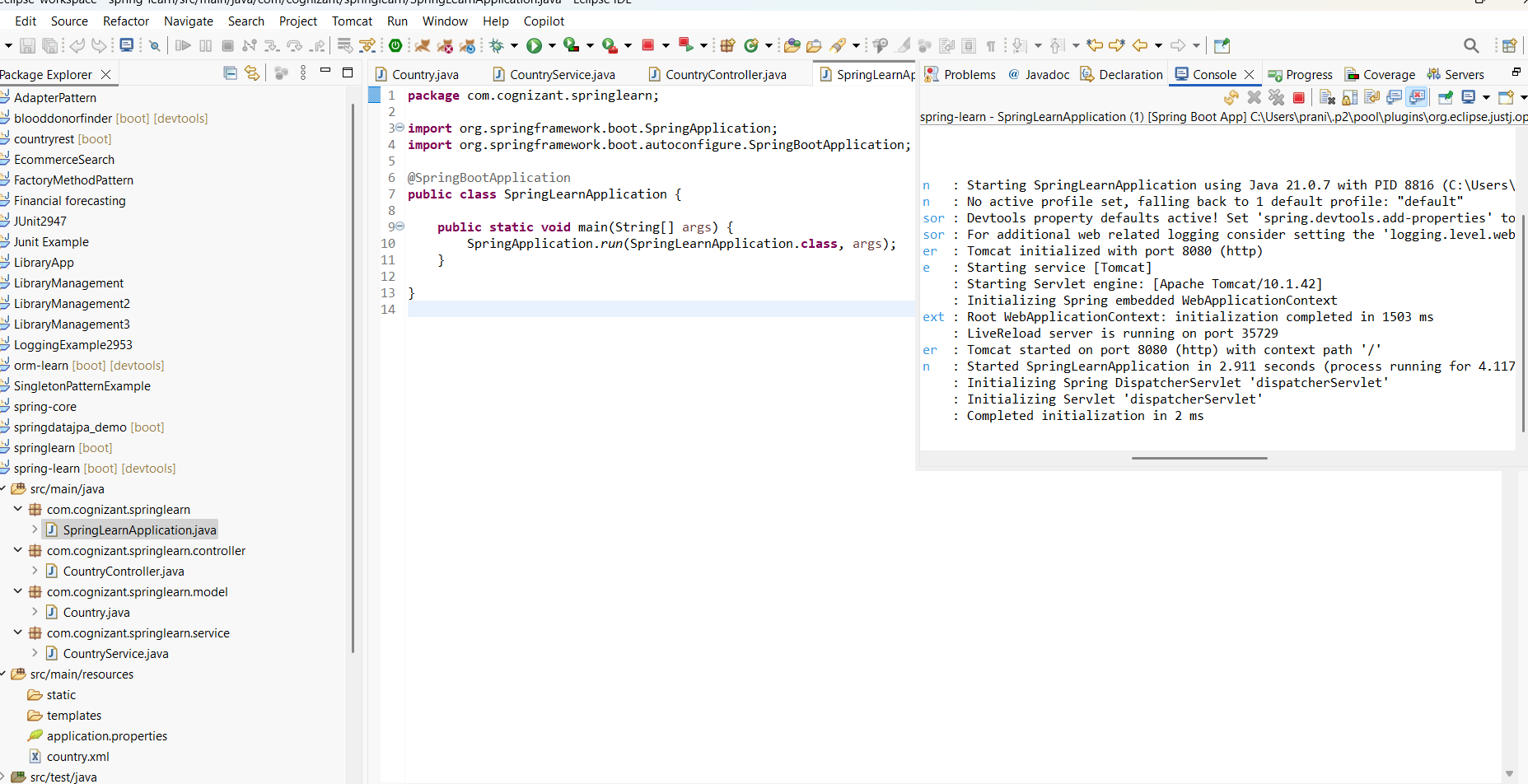
**Output:**

**Network tab**

****

**Postman**



****

**Exercise 6:** Create authentication service that returns JWT

**AuthController.java:**

**package** com.cognizant.jwtauth.controller;

**import** com.cognizant.jwtauth.util.JwtUtil;

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

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

**import** jakarta.servlet.http.HttpServletRequest;

**import** java.util.Base64;

@RestController

**public** **class** AuthController {

@GetMapping("/authenticate")

**public** String authenticate(HttpServletRequest request) {

System.***out***.println("🔐 /authenticate endpoint called");

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

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

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

**byte**[] decoded = Base64.*getDecoder*().decode(base64Credentials);

String[] credentials = **new** String(decoded).split(":", 2);

String username = credentials[0];

String password = credentials[1];

System.***out***.println("👤 Username: " + username);

System.***out***.println("🔑 Password: " + password);

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

String token = JwtUtil.*generateToken*(username);

**return** "{\"token\":\"" + token + "\"}";

}

}

**return** "{\"error\":\"Invalid credentials\"}";

}

}

**SecurityConfig.java:**

**package** com.cognizant.jwtauth.security;

**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;

@Configuration

**public** **class** SecurityConfig {

@Bean

**public** SecurityFilterChain filterChain(HttpSecurity http) **throws** Exception {

http

.csrf().disable()

.authorizeHttpRequests(auth -> auth

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

.anyRequest().authenticated()

);

**return** http.build();

}

}

**JwtUtil.java:**

**package** com.cognizant.jwtauth.util;

**import** io.jsonwebtoken.Jwts;

**import** io.jsonwebtoken.SignatureAlgorithm;

**import** java.util.Date;

**public** **class** JwtUtil {

**private** **static** **final** String ***SECRET\_KEY*** = "secret";

**public** **static** String generateToken(String username) {

**return** Jwts.*builder*()

.setSubject(username)

.setIssuedAt(**new** Date())

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

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

.compact();

}

}

**JwtauthapiApplication.java:**

**package** com.cognizant.jwtauth;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** JwtauthapiApplication {

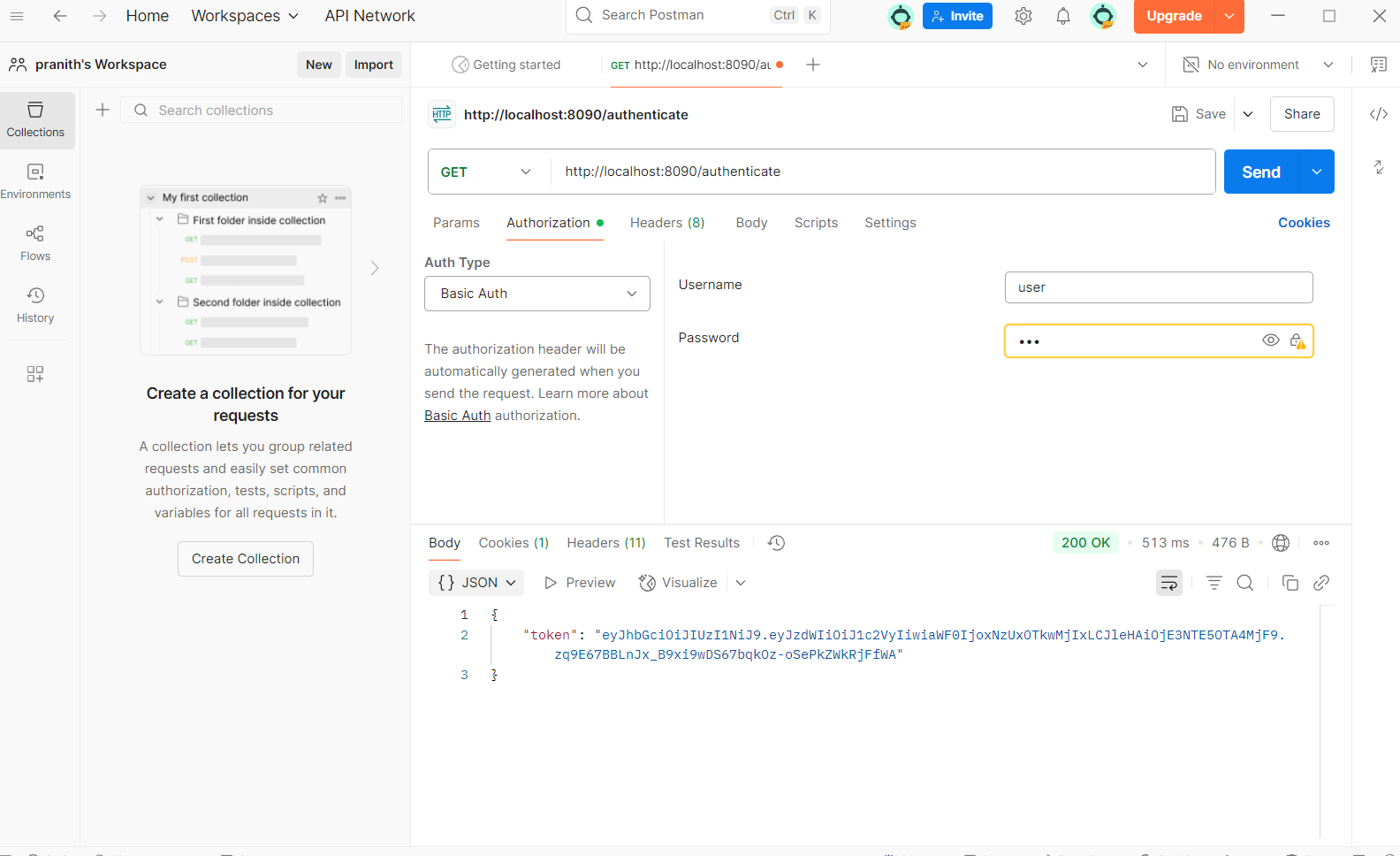
**public** **static** **void** main(String[] args) {

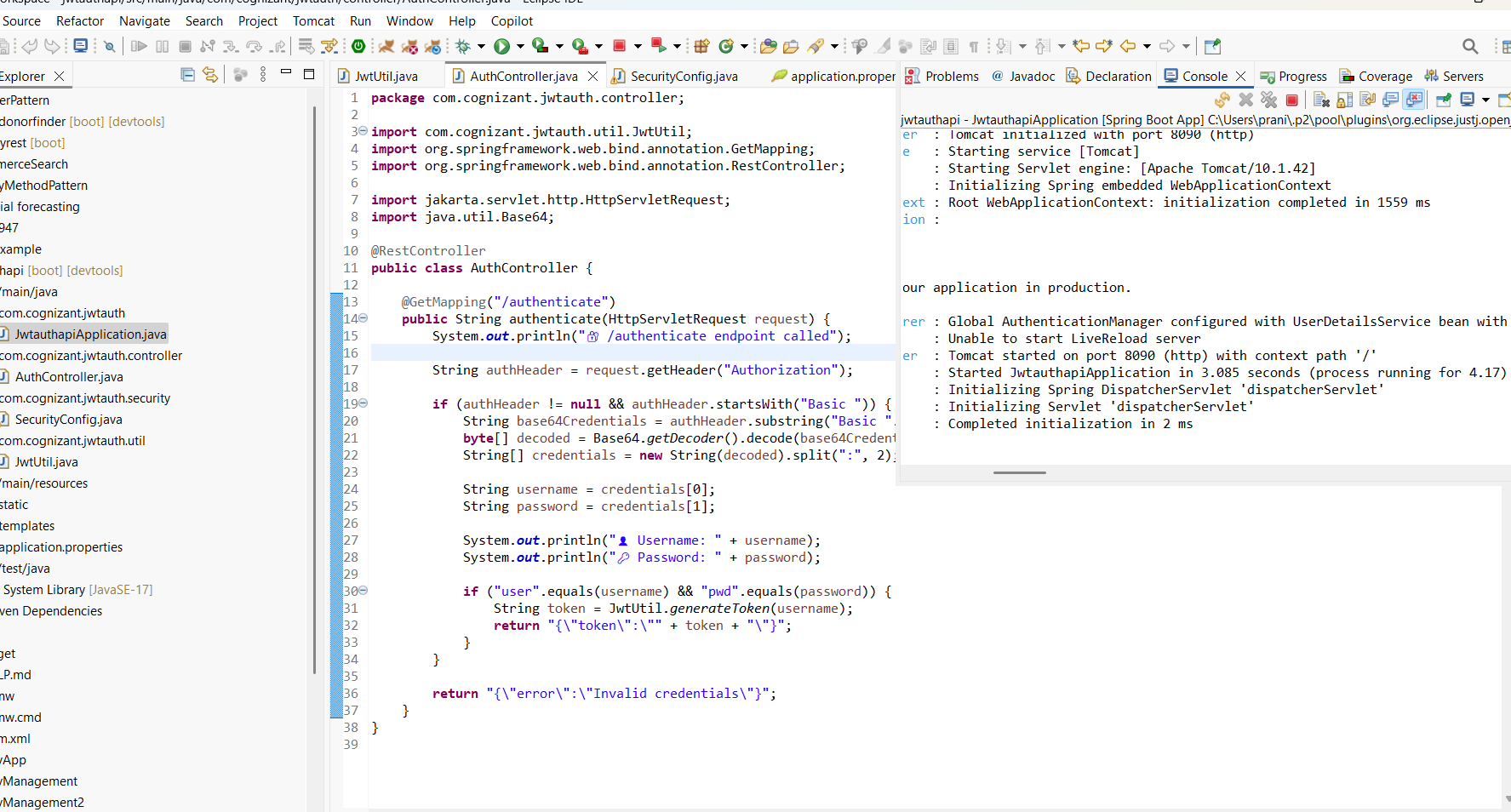
SpringApplication.*run*(JwtauthapiApplication.**class**, args);

}

}

**Output:**

**Postman**

****