**Week-4 Spring REST using Spring Boot 3**

**Create a Spring Web Project using Maven**

**SpringLearnApplication.java:**

package com.cognizant.springlearn;

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) {

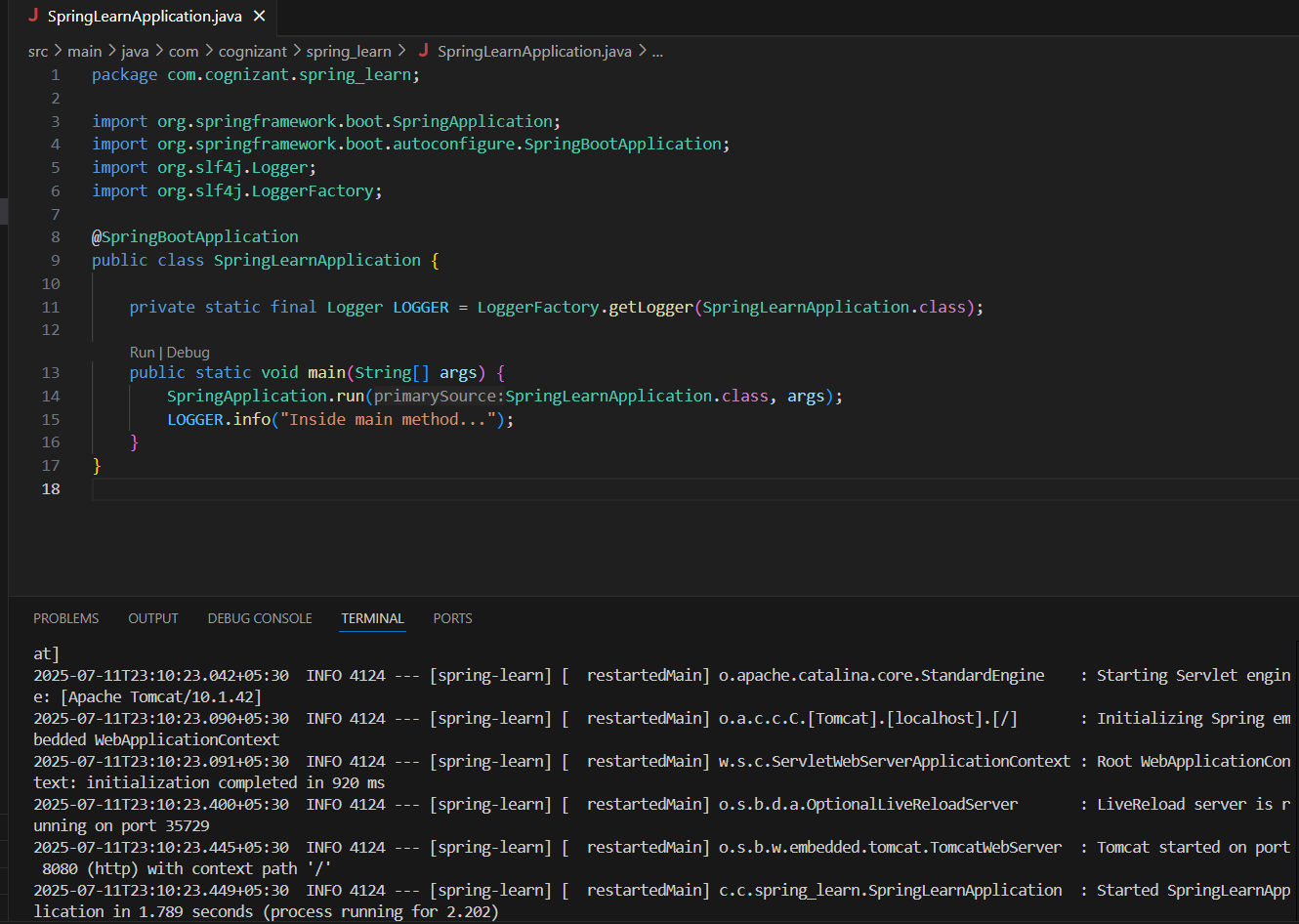
SpringApplication.run(SpringLearnApplication.class, args);

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

}

}

**OUTPUT:**



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

**Country.java:**

package com.cognizant.springlearn;

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 + "]";

}

}

**SpringLearnApplication.java** (Main class):

package com.cognizant.spring\_learn;

import java.util.\*;

public class SpringLearnApplication {

public static void main(String[] args) {

System.out.println("START");

List<Country> countryList = new ArrayList<>();

countryList.add(new Country("IN", "India"));

countryList.add(new Country("US", "United States"));

countryList.add(new Country("CN", "China"));

countryList.add(new Country("JP", "Japan"));

countryList.add(new Country("DE", "Germany"));

countryList.add(new Country("FR", "France"));

countryList.add(new Country("BR", "Brazil"));

countryList.add(new Country("ZA", "South Africa"));

countryList.add(new Country("AU", "Australia"));

countryList.add(new Country("RU", "Russia"));

for (Country country : countryList) {

System.out.println(country);

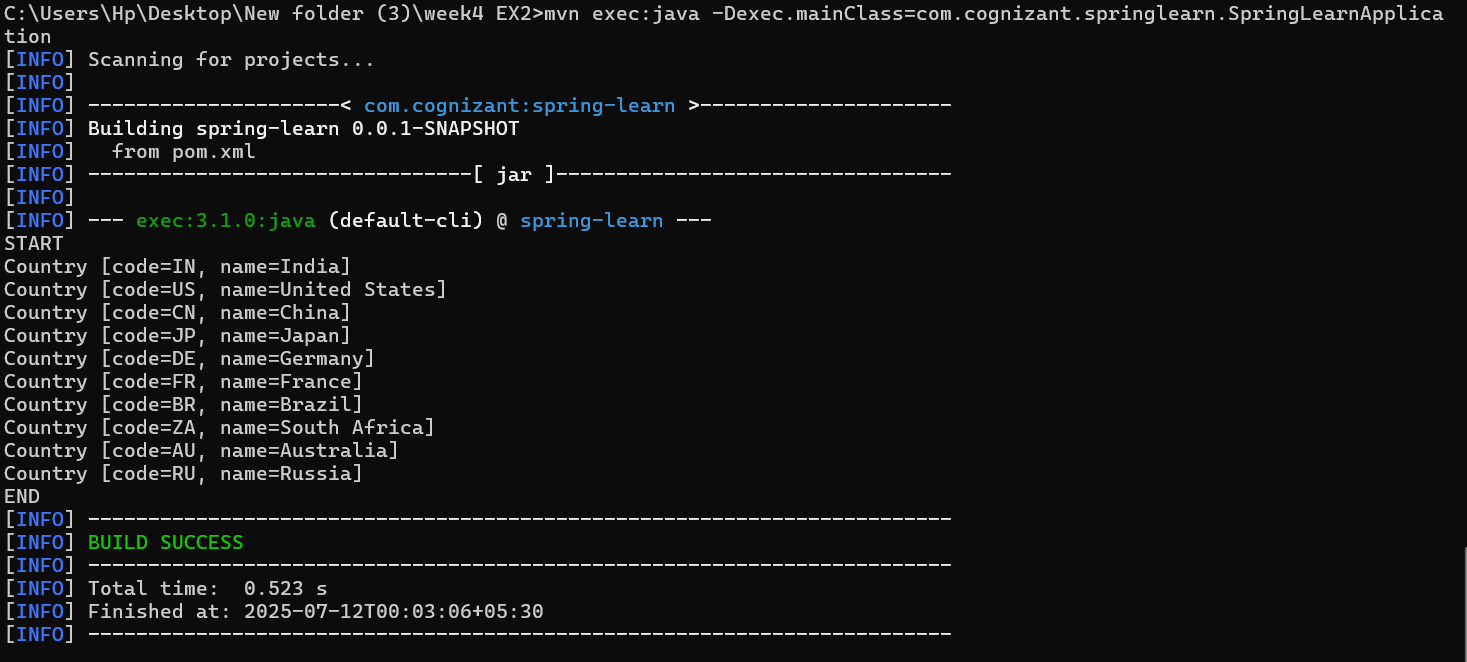
}

System.out.println("END");

}

}

**OUTPUT:**



================================================================

**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.\*;

@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) {

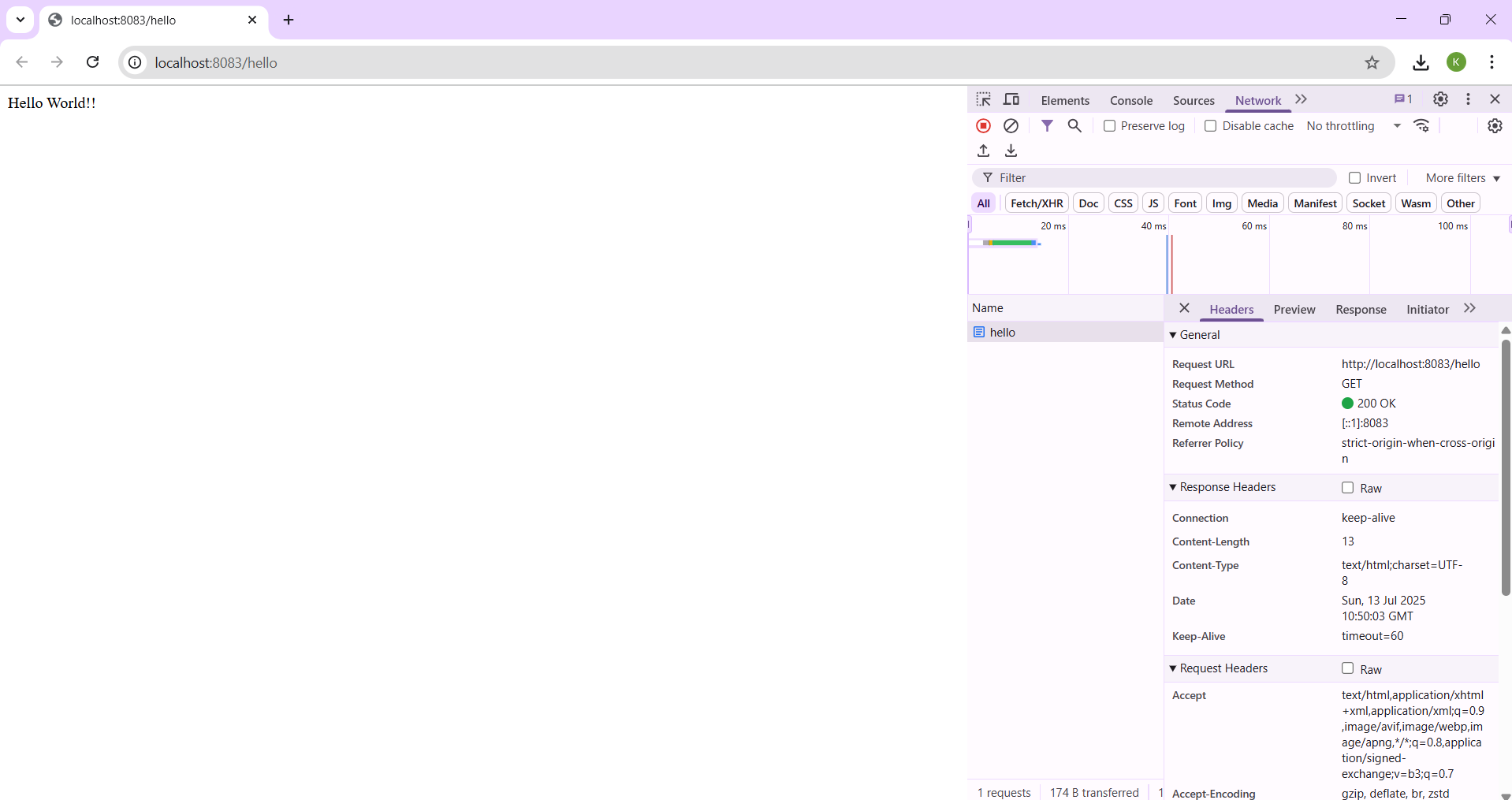
SpringApplication.run(SpringLearnApplication.class, args);

}

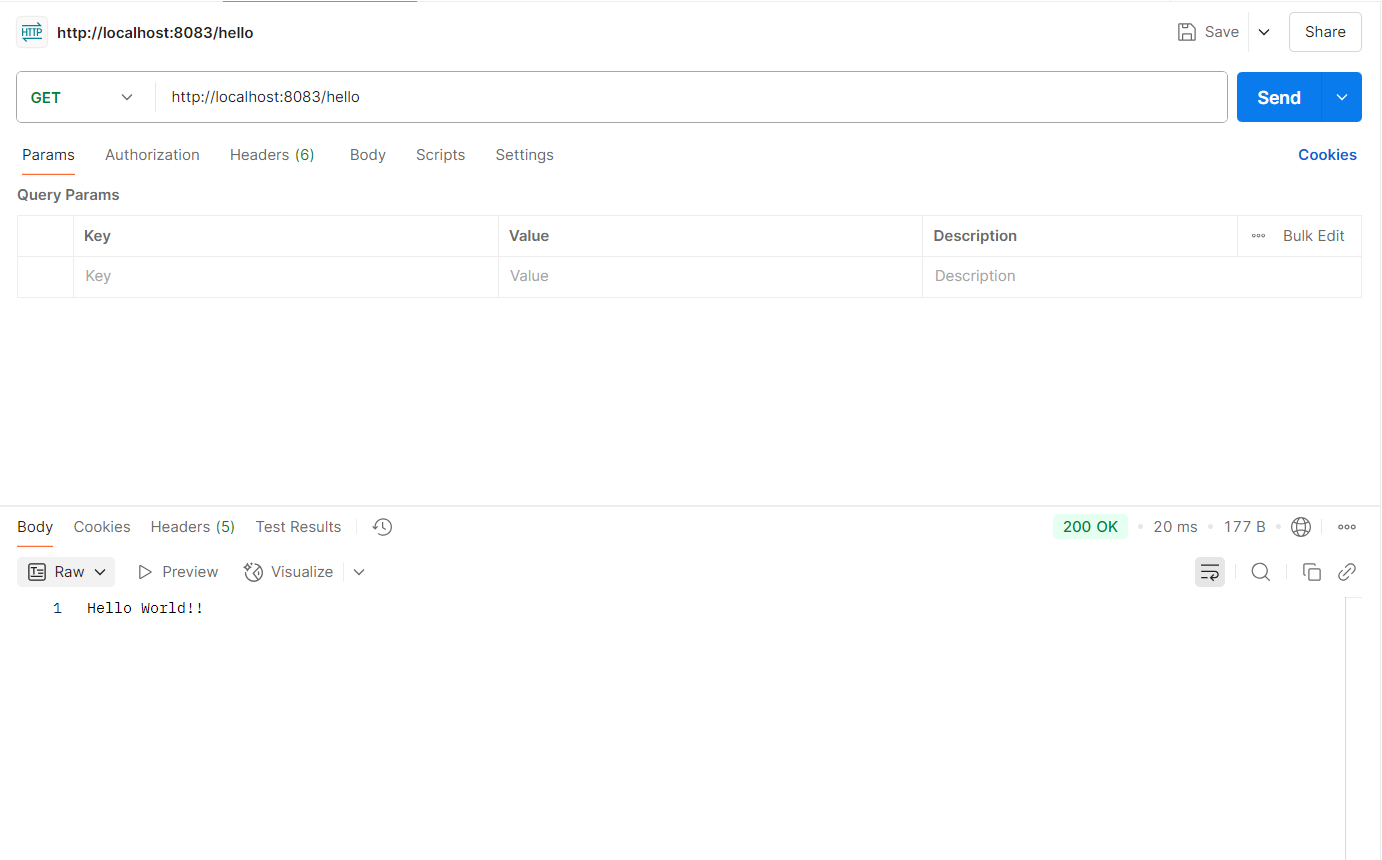
}

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

**Browser:**



**Postman:**



================================================================

**REST - Country Web Service** 

**CountryController.java:**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.model.Country;

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 {

@RequestMapping("/country")

public Country getCountryIndia() {

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

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

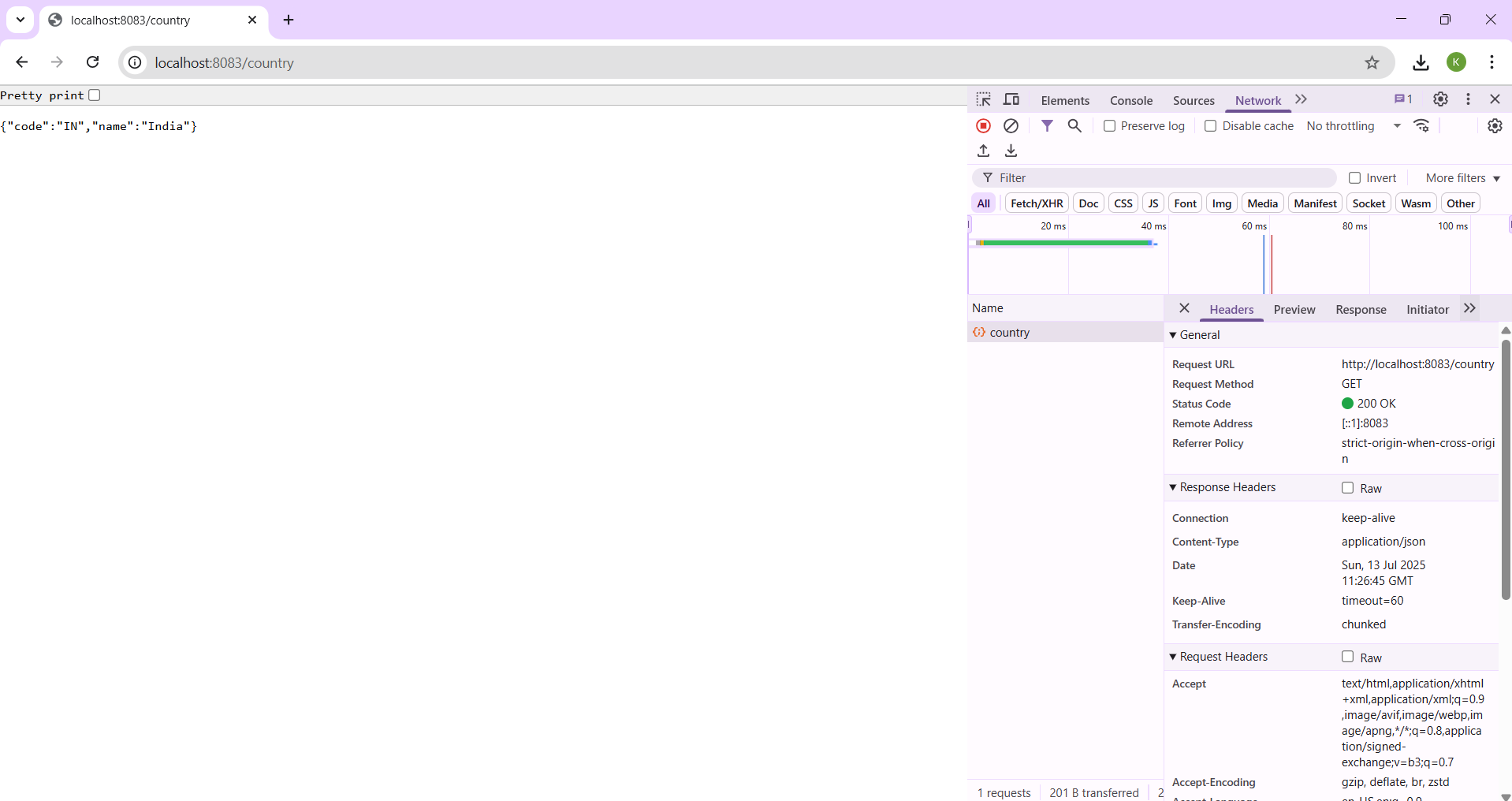
return country;

}

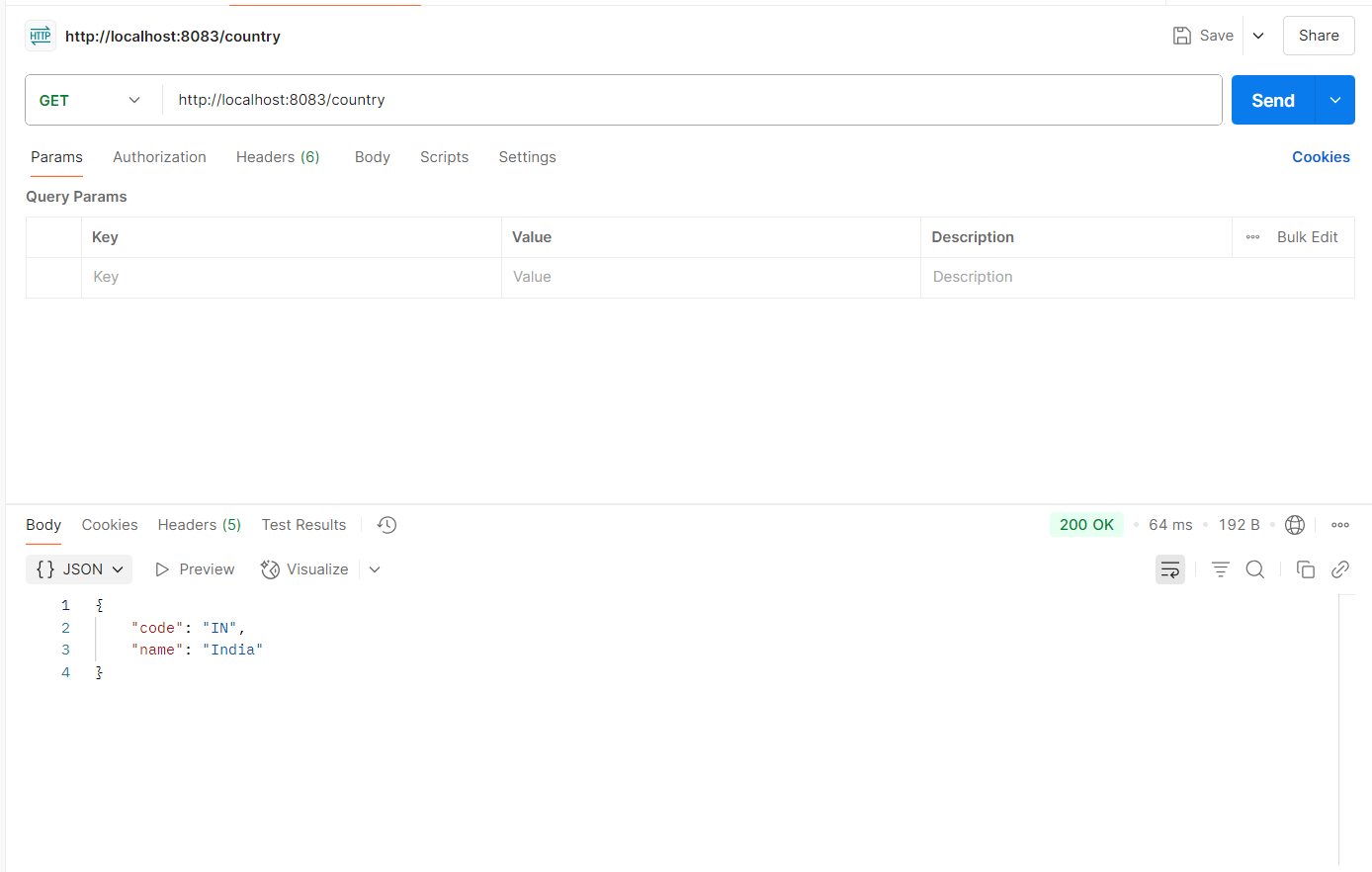
}

**OUTPUT:** http://localhost:8083/countries

**Browser:**



**Postman:**



================================================================

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

**CountryController.java:**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.model.Country;

import com.cognizant.springlearn.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) throws Exception {

return countryService.getCountry(code);

}

}

**CountryService.java:**

package com.cognizant.springlearn.service;

import com.cognizant.springlearn.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) throws Exception {

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

return countryList.stream()

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

.findFirst()

.orElseThrow(() -> new Exception("Country not found for code: " + code));

}

}

**Country.java:**

package com.cognizant.springlearn.model;

public class Country {

private String code;

private String name;

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getName() {

return name;

}

public void setName(String name) {

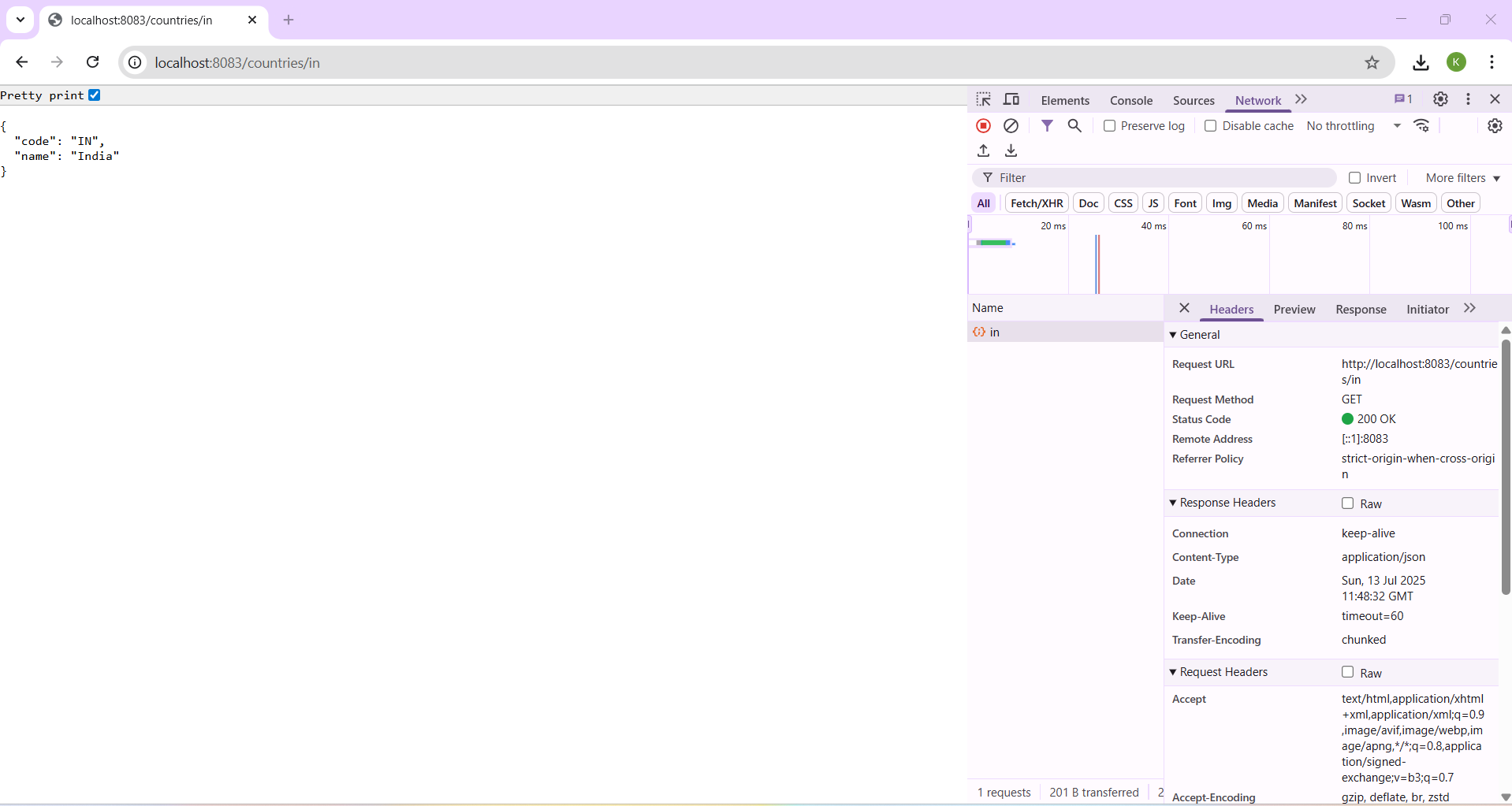
this.name = name;

}

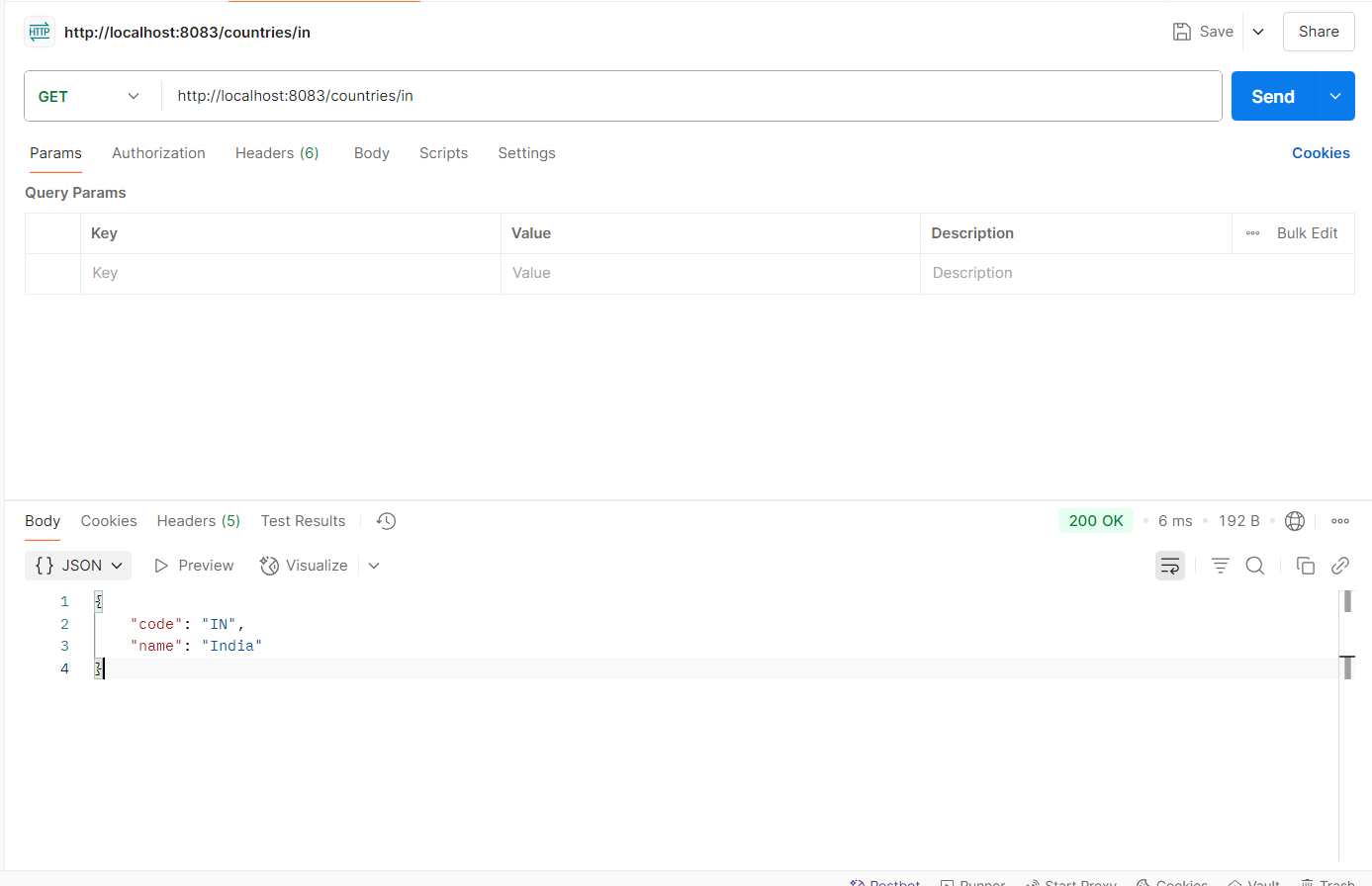
}

**OUTPUT:** http://localhost:8083/countries/in

**Browser:**



**Postman:**



================================================================

**Create authentication service that returns JWT** 

**AuthenticationController.java:**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.util.JwtUtil;

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

import org.springframework.http.ResponseEntity;

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

import java.util.Base64;

@RestController

public class AuthenticationController {

@Autowired

private JwtUtil jwtUtil;

@GetMapping("/authenticate")

public ResponseEntity<?> authenticate(@RequestHeader(value = "Authorization", required = false) String authHeader) {

System.out.println("Incoming request to /authenticate");

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

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

}

try {

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

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

String credentials = new String(decodedBytes);

String[] parts = credentials.split(":", 2);

if (parts.length != 2) {

return ResponseEntity.status(401).body("Invalid Authorization format");

}

String username = parts[0];

String password = parts[1];

if (!"user".equals(username) || !"pwd".equals(password)) {

return ResponseEntity.status(401).body("Invalid username or password");

}

String token = jwtUtil.generateToken(username);

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

} catch (Exception e) {

return ResponseEntity.status(500).body("Internal server error: " + e.getMessage());

}

}

}

**JwtUtil.java:**

package com.cognizant.springlearn.util;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import org.springframework.stereotype.Component;

import java.util.Date;

@Component

public class JwtUtil {

private final String SECRET\_KEY = "secret";

public String generateToken(String username) {

return Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date())

.setExpiration(new Date(System.currentTimeMillis() + 3600000))

.signWith(SignatureAlgorithm.HS256, SECRET\_KEY)

.compact();

}

}

**SecurityConfig.java:**

package com.cognizant.springlearn.security;

import org.springframework.context.annotation.Configuration;

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

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

@Configuration

public class SecurityConfig extends WebSecurityConfigurerAdapter {

@Override

protected void configure(HttpSecurity http) throws Exception {

http

.csrf().disable()

.authorizeRequests()

.antMatchers("/authenticate").permitAll()

.anyRequest().authenticated()

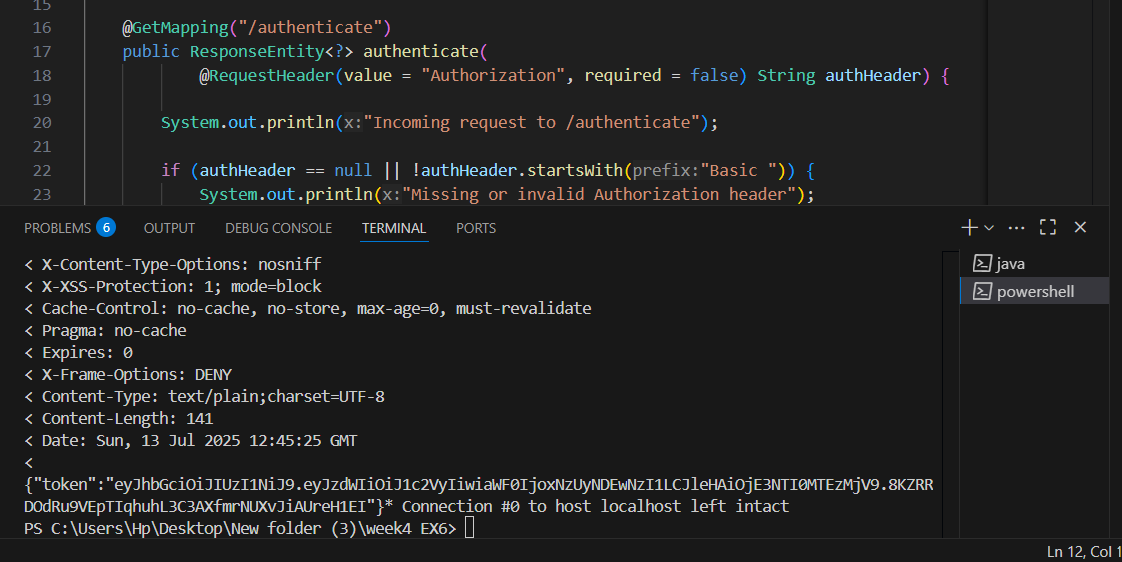
.and()

.httpBasic();

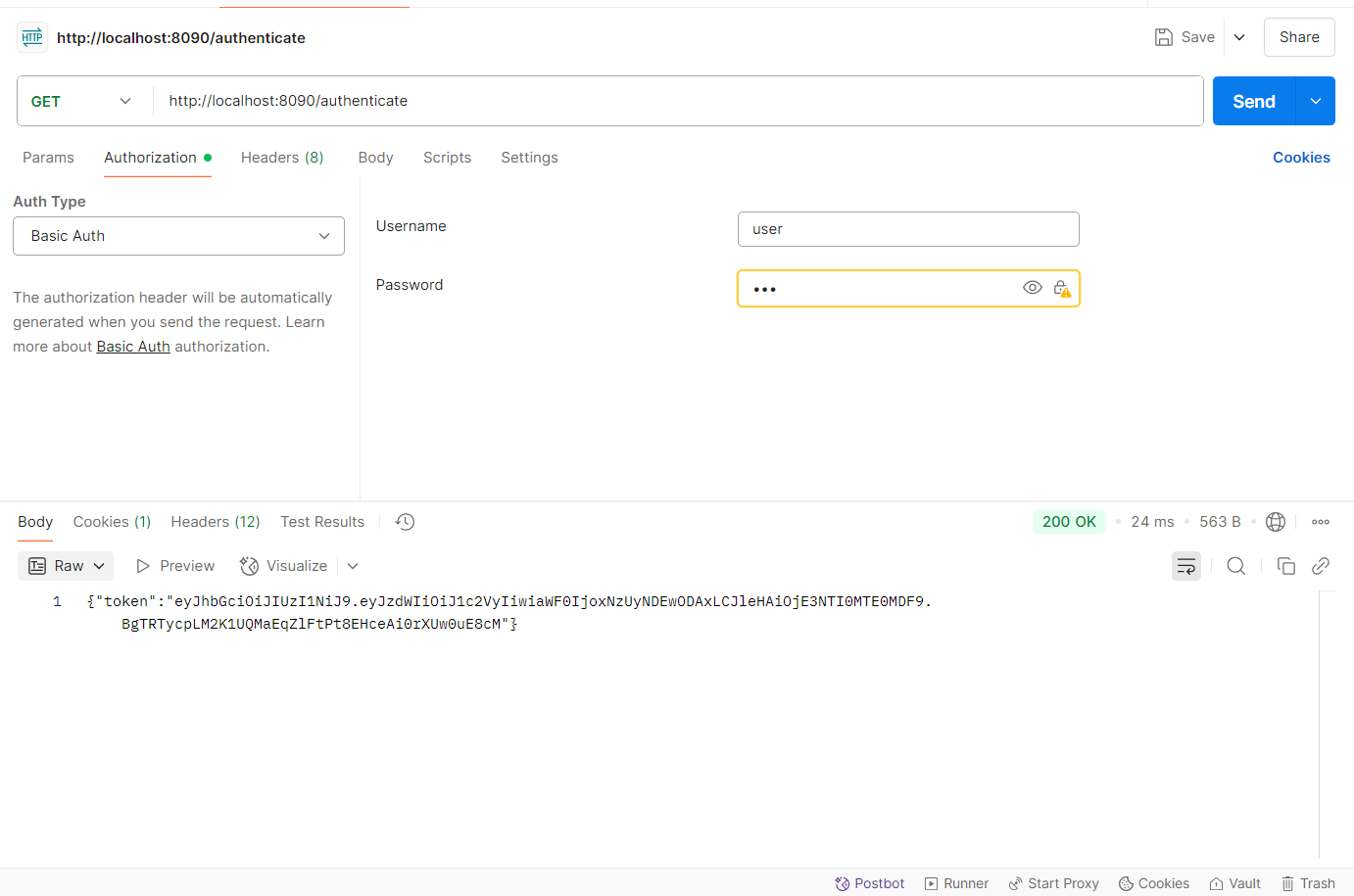
}

}

**OUTPUT:** http://localhost:8090/authenticate



**Postman:**



================================================================