**WEEK 4  
 Spring Rest Using Spring Boot 3**

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

**CODE:**

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

    public static void main(String[] args) {

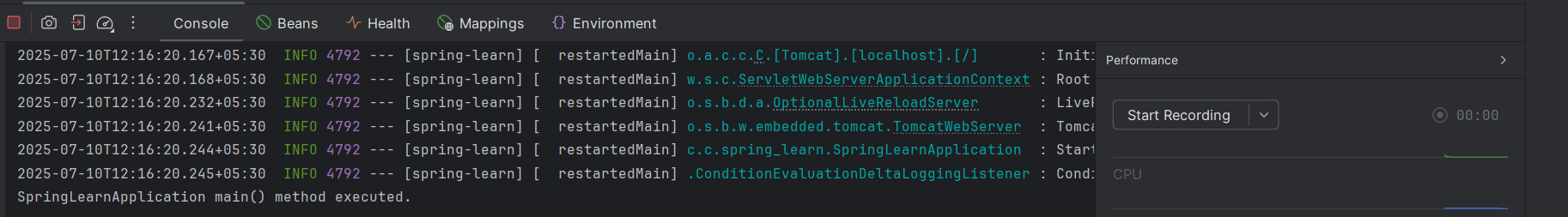
        SpringApplication.run(SpringLearnApplication.class, args);

        System.out.println("SpringLearnApplication main() method executed.");

    }

}

**O/P**



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

**Code :**

package com.cognizant.spring;

public class Country {

    private String code;

    private String name;

    public Country() {

    }

    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;

    }

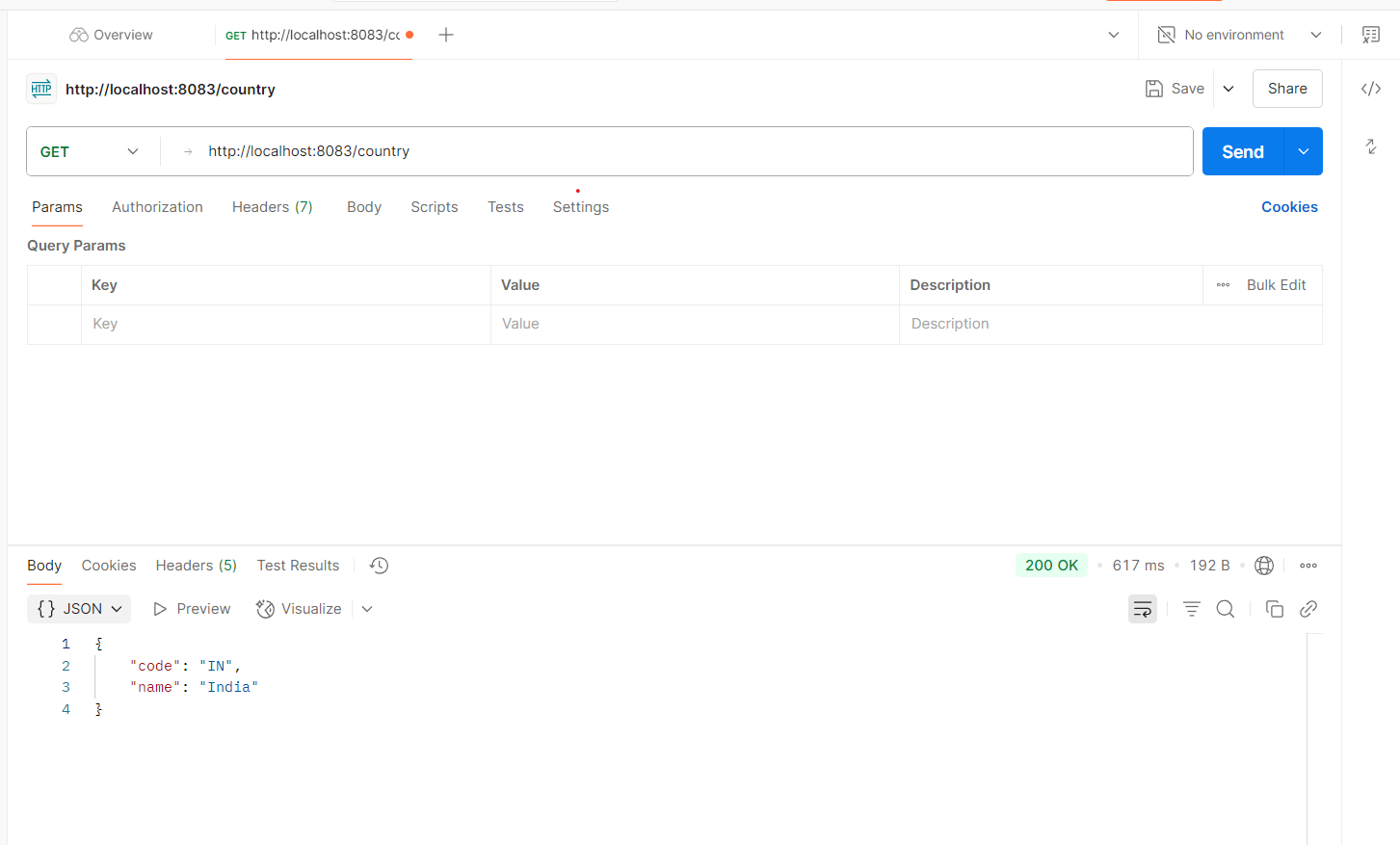
    @Override

    public String toString() {

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

    }

}

**OUTPUT:**  


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

package com.cognizant.spring.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.debug("START sayHello()");

String message = "Hello World!!";

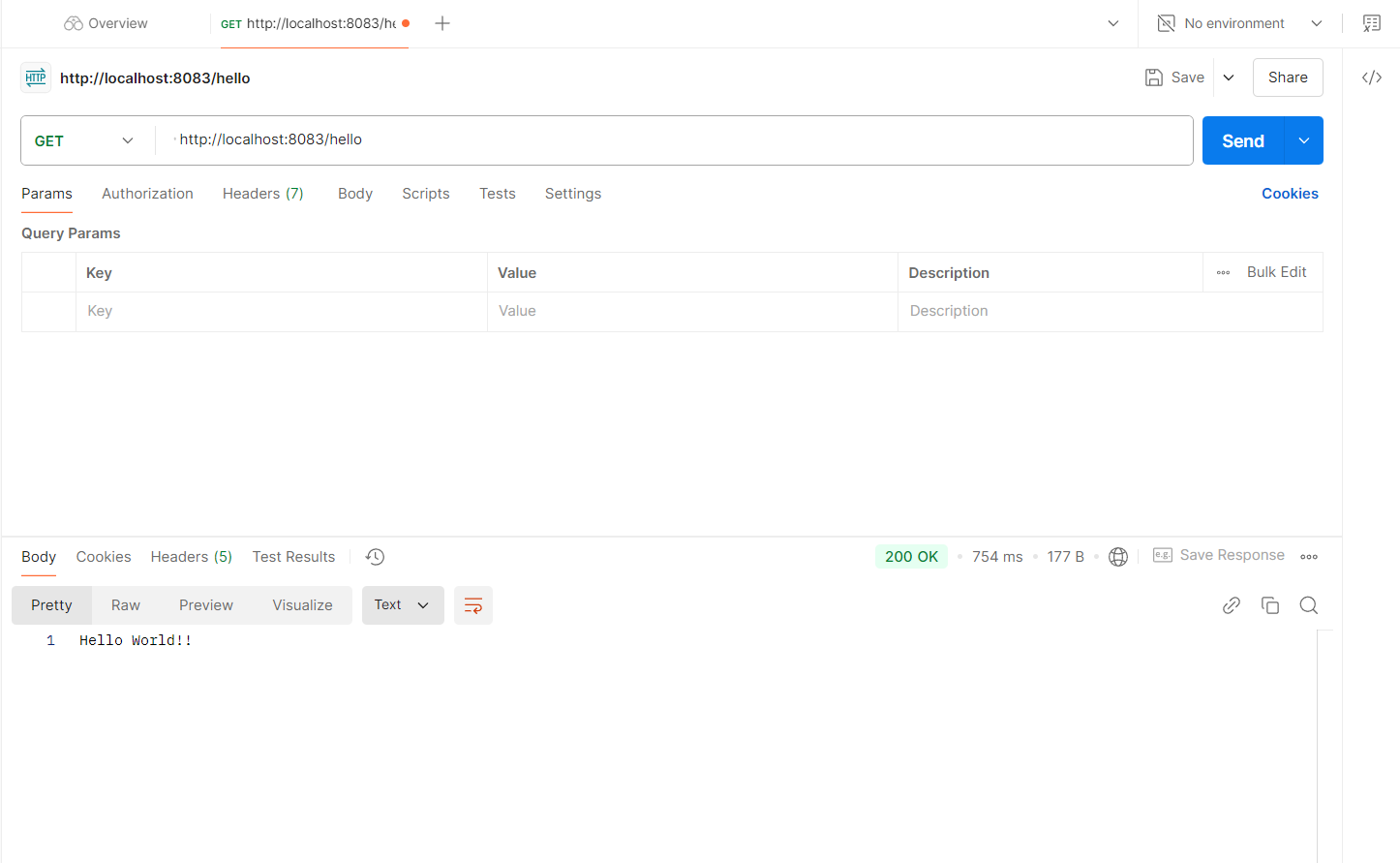
LOGGER.debug("END sayHello()");

return message;

}

}

**OUTPUT:**



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

package com.cognizant.spring;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpringApplication {

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

    public static void displayCountry() {

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

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

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

    }

    public static void main(String[] args) {

        LOGGER.debug("START of main()");

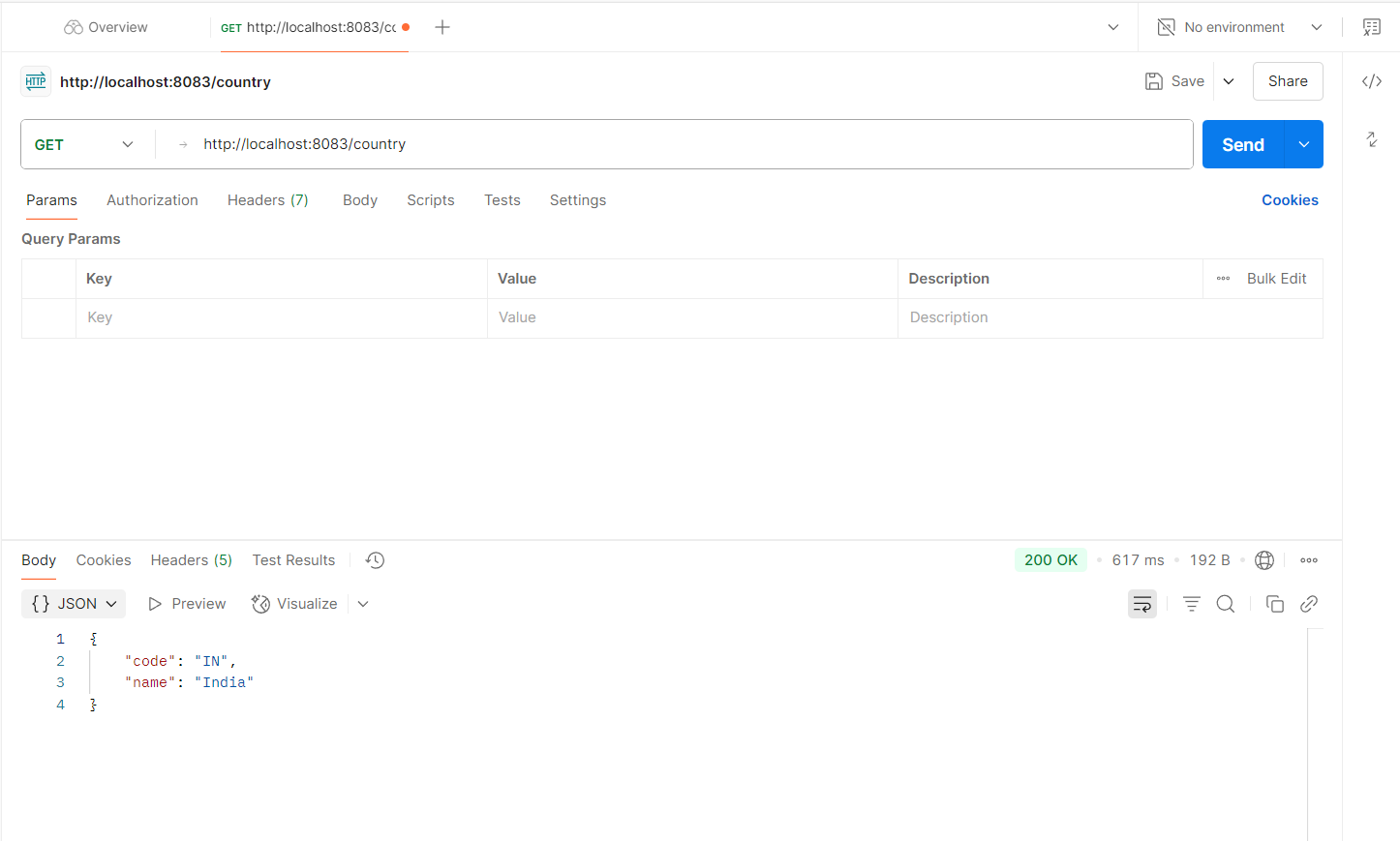
        displayCountry();

        LOGGER.debug("END of main()");

    }

}

**OUTPUT:**



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

**Code**:

package com.cognizant.spring.controller;

import com.cognizant.spring.Country;

import com.cognizant.spring.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("/countries/{code}")

    public Country getCountry(@PathVariable String code) {

        LOGGER.debug("START getCountry() with code: {}", code);

        Country country = countryService.getCountry(code);

        LOGGER.debug("END getCountry() : {}", country);

        return country;

    }

}

//CountryService.java

package com.cognizant.spring.service;

import com.cognizant.spring.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))

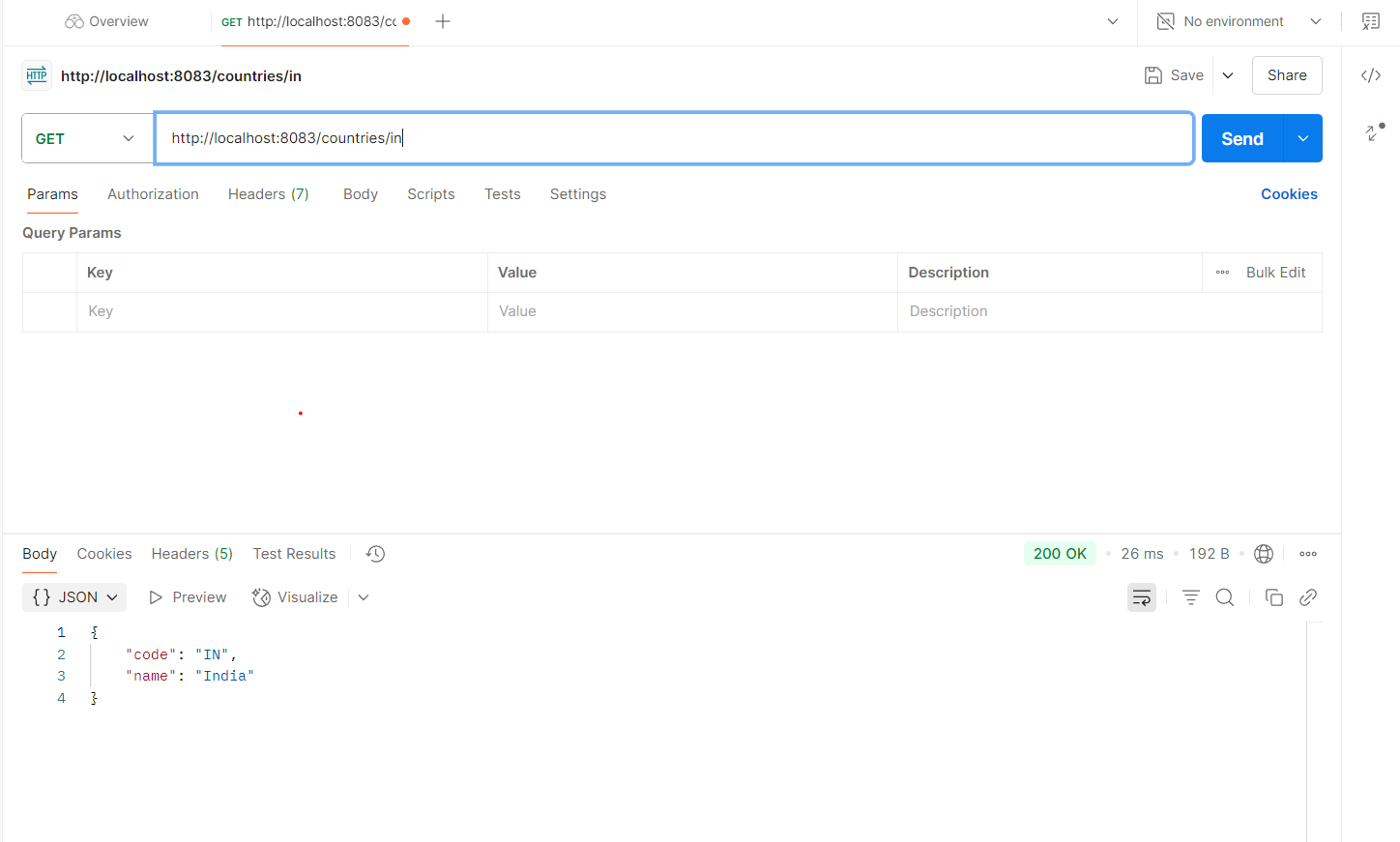
                .findFirst()

                .orElse(null);

    }

}

**OUTPUT:**



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

**Code**:

package com.cognizant.springlearn.controller;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import io.jsonwebtoken.security.Keys;

import org.springframework.http.ResponseEntity;

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

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

import jakarta.servlet.http.HttpServletRequest;

import java.security.Key;

import java.util.Base64;

import java.util.Date;

@RestController

public class AuthController {

private static final Key SECRET\_KEY = Keys.secretKeyFor(SignatureAlgorithm.HS256);

@GetMapping("/authenticate")

public ResponseEntity<?> authenticate(HttpServletRequest request) {

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

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

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

}

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

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

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

String username = values[0];

String password = values[1];

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

return ResponseEntity.status(401).body("Invalid credentials.");

}

String jwt = Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date())

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

.signWith(SECRET\_KEY)

.compact();

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

}

}

**OUTPUT**:

