**HANDSON EXERCISES - WEEK 4**

**Skill : Spring REST using Spring Boot 3**

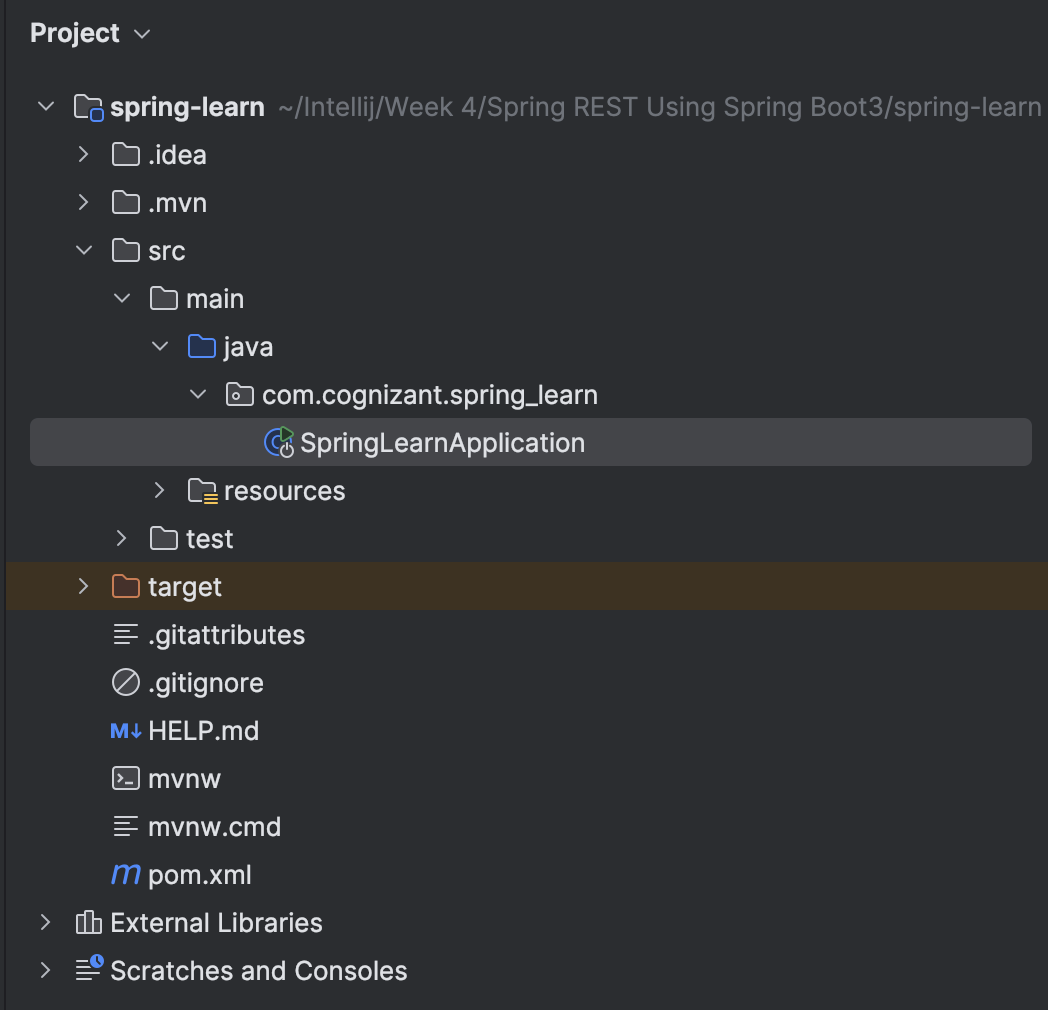
**File : 1. spring-rest-handson.docx**

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

**Spring Initializr Configuration for spring-learn Project**



**Project Structure :**

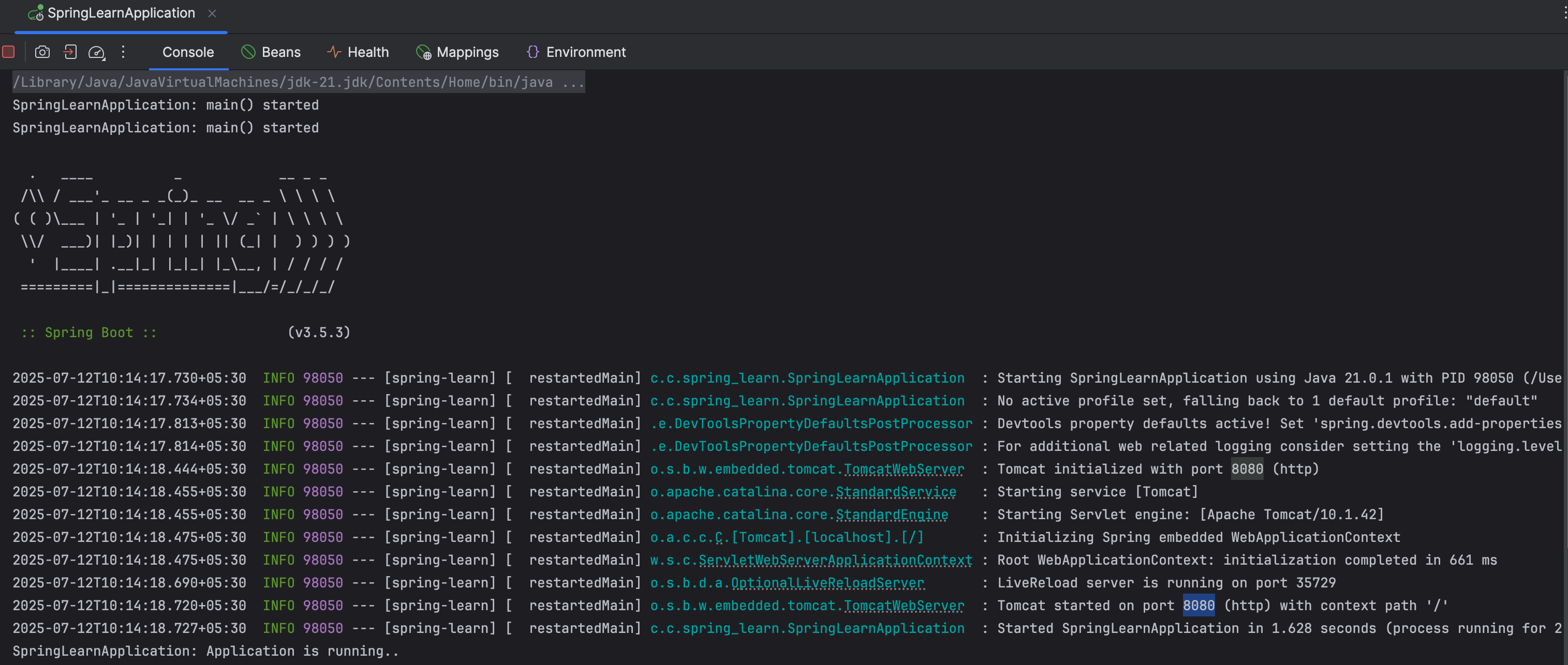


**CODE :**

**SpringLearnApplication.java :**

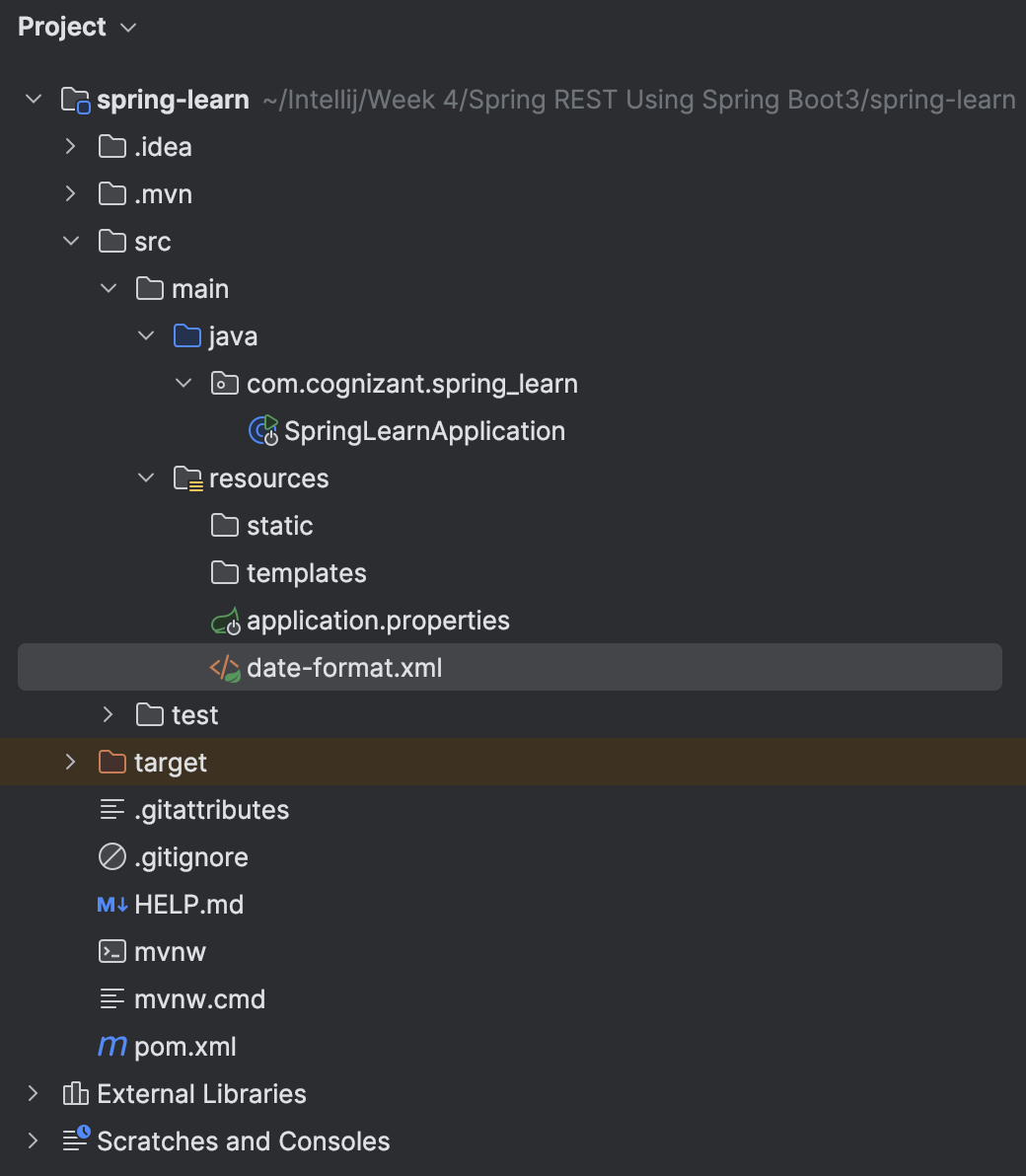
package com.cognizant.spring\_learn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
  
 public static void main(String[] args) {  
 System.out.println("SpringLearnApplication: main() started");  
 SpringApplication.run(SpringLearnApplication.class, args);  
 System.out.println("SpringLearnApplication: Application is running..");  
 }  
}

**OUTPUT :**



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

**Project Structure :**



**CODE :**

**src/resources/date-fromat.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>

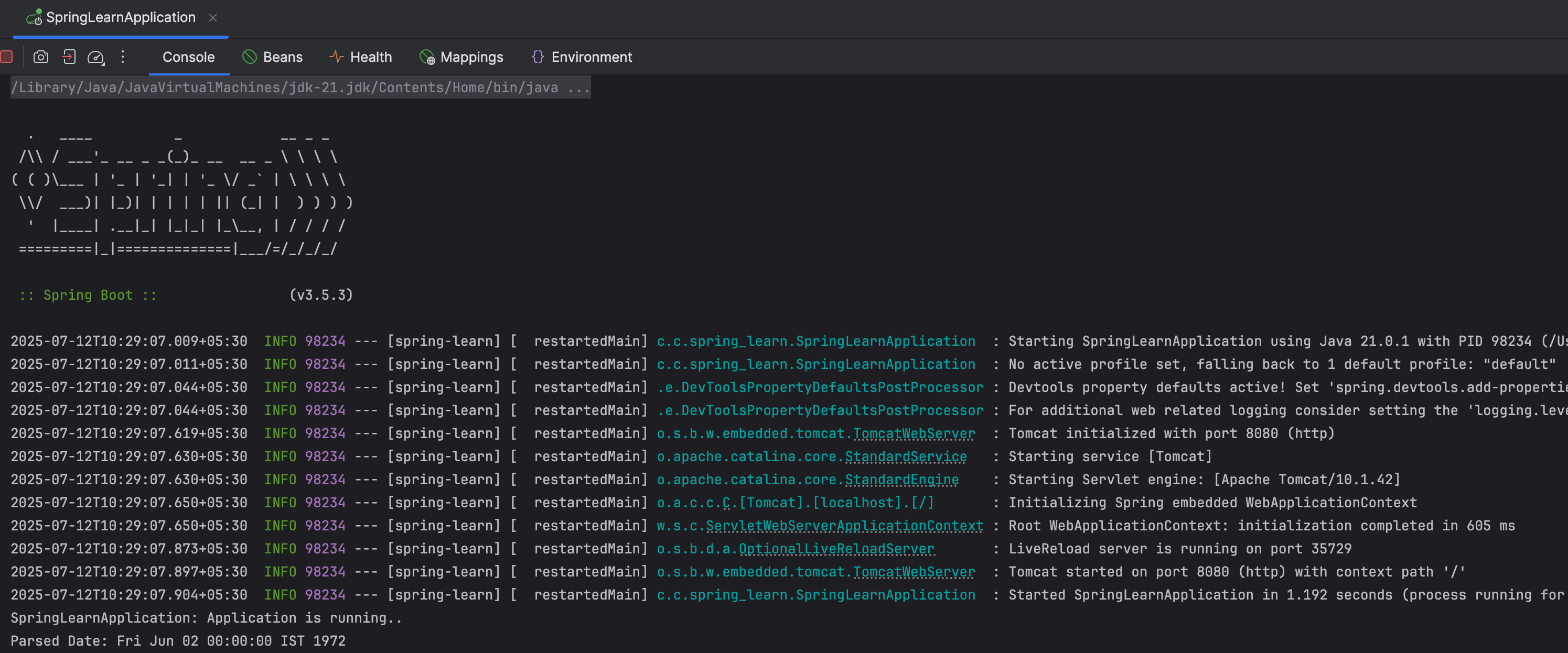
**SpringLearnApplication.java :**

package com.cognizant.spring\_learn;  
  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
import java.text.SimpleDateFormat;  
import java.util.Date;

@SpringBootApplication  
public class SpringLearnApplication {  
  
 public static void main(String[] args) throws Exception {  
 SpringApplication.*run*(SpringLearnApplication.class, args);  
 System.*out*.println("SpringLearnApplication: Application is running..");  
  
 *displayDate*();

}  
  
 public static void displayDate() throws Exception {  
 ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");  
  
 SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);  
  
 Date parsedDate = format.parse("31/12/2018");  
  
 System.*out*.println("Parsed Date: " + parsedDate);  
 }  
}

**OUTPUT :**



**Exercise 3 : Spring Core - Incorporate Logging**

**CODE :**

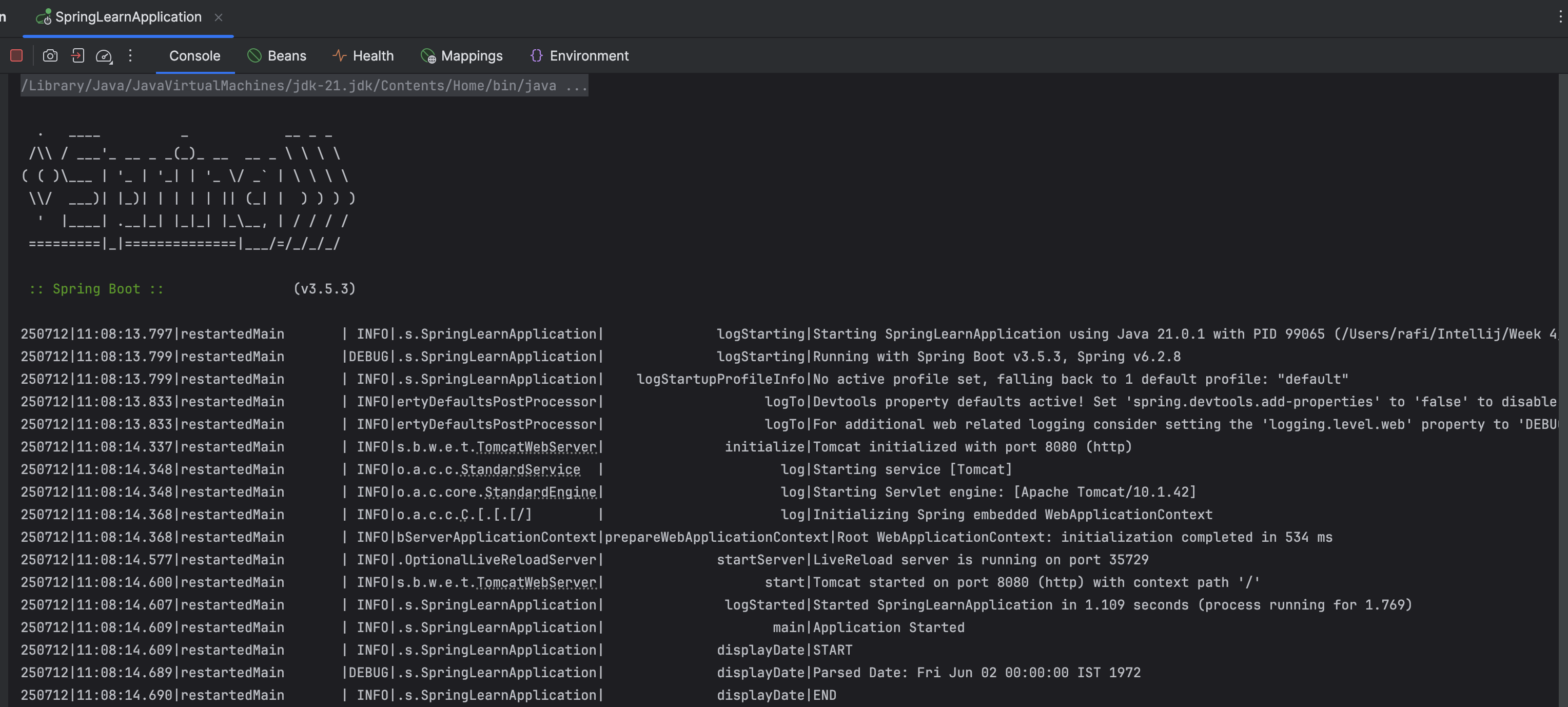
**resources/application.properties :**

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.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
import java.text.SimpleDateFormat;  
import java.util.Date;  
  
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) throws Exception {  
 SpringApplication.run(SpringLearnApplication.class, args);  
 LOGGER.info("Application Started");  
  
 displayDate();   
 }  
  
 public static void displayDate() throws Exception {  
 LOGGER.info("START");  
  
 ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");  
 SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);  
  
 Date parsedDate = format.parse("02/06/1972");  
  
 LOGGER.debug("Parsed Date: {}", parsedDate);  
  
 LOGGER.info("END");  
 }  
}

**OUTPUT :**



**Exercise 4 : Spring Core – Load Country from Spring Configuration XML**

**resources/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>

**SpringLearnApplication.java :**

package com.cognizant.spring\_learn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;

import java.text.SimpleDateFormat;  
import java.util.Date;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);  
  
 public static void main(String[] args) throws Exception {  
 SpringApplication.run(SpringLearnApplication.class, args);

LOGGER.info("Application Started");  
  
 displayDate();

displayCountry();

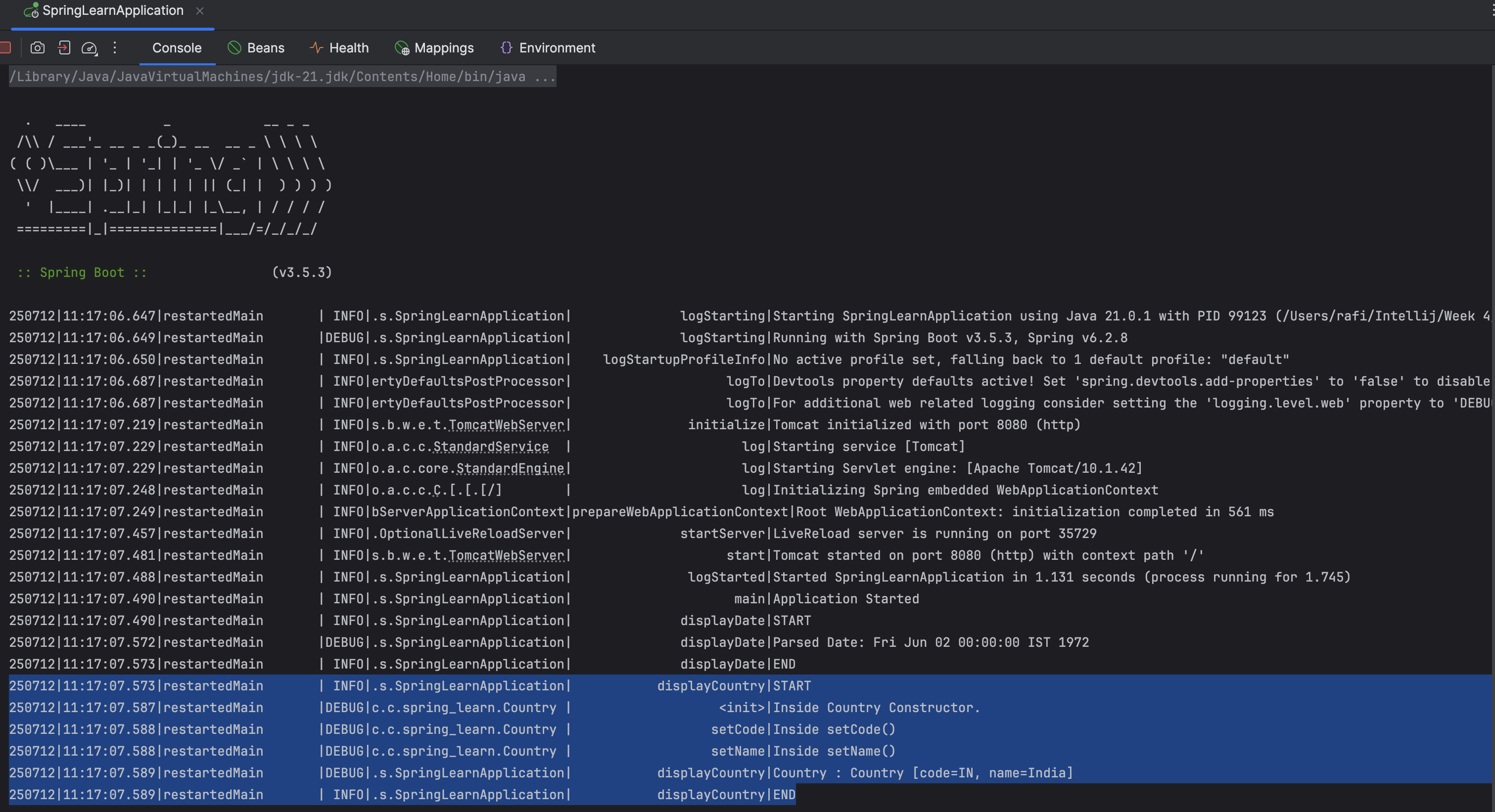
}  
  
 public static void displayDate() throws Exception {  
 LOGGER.info("START");  
  
 ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");  
 SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);  
  
 Date parsedDate = format.parse("02/06/1972");  
 LOGGER.debug("Parsed Date: {}", parsedDate);  
  
 LOGGER.info("END");

}  
  
 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 :**



**Exercise 5 : Spring Core – Demonstration of Singleton Scope and Prototype Scope**

**Singleton Scope :**

**resources/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>

**SpringLearnApplication.java :**

package com.cognizant.spring\_learn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
import java.text.SimpleDateFormat;  
import java.util.Date;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);  
  
 public static void main(String[] args) throws Exception {  
 SpringApplication.run(SpringLearnApplication.class, args);  
 LOGGER.info("Application Started");  
  
 displayDate();  
 displayCountry();  
 }  
  
 public static void displayDate() throws Exception {  
 LOGGER.info("START");  
  
 ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");  
 SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);  
  
 Date parsedDate = format.parse("02/06/1972");  
 LOGGER.debug("Parsed Date: {}", parsedDate);  
  
 LOGGER.info("END");  
 }  
  
 public static void displayCountry() {  
 LOGGER.info("START");  
  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
  
 Country country1 = context.getBean("country", Country.class);  
 Country country2 = context.getBean("country", Country.class);  
  
 LOGGER.debug("Country 1 : {}", country1);  
 LOGGER.debug("Country 2 : {}", country2);  
  
 LOGGER.info("END");  
 }  
}

**Prototype Scope -**

**country.xml :**

<bean id="country" class="com.cognizant.spring\_learn.Country" scope="prototype">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
</bean>

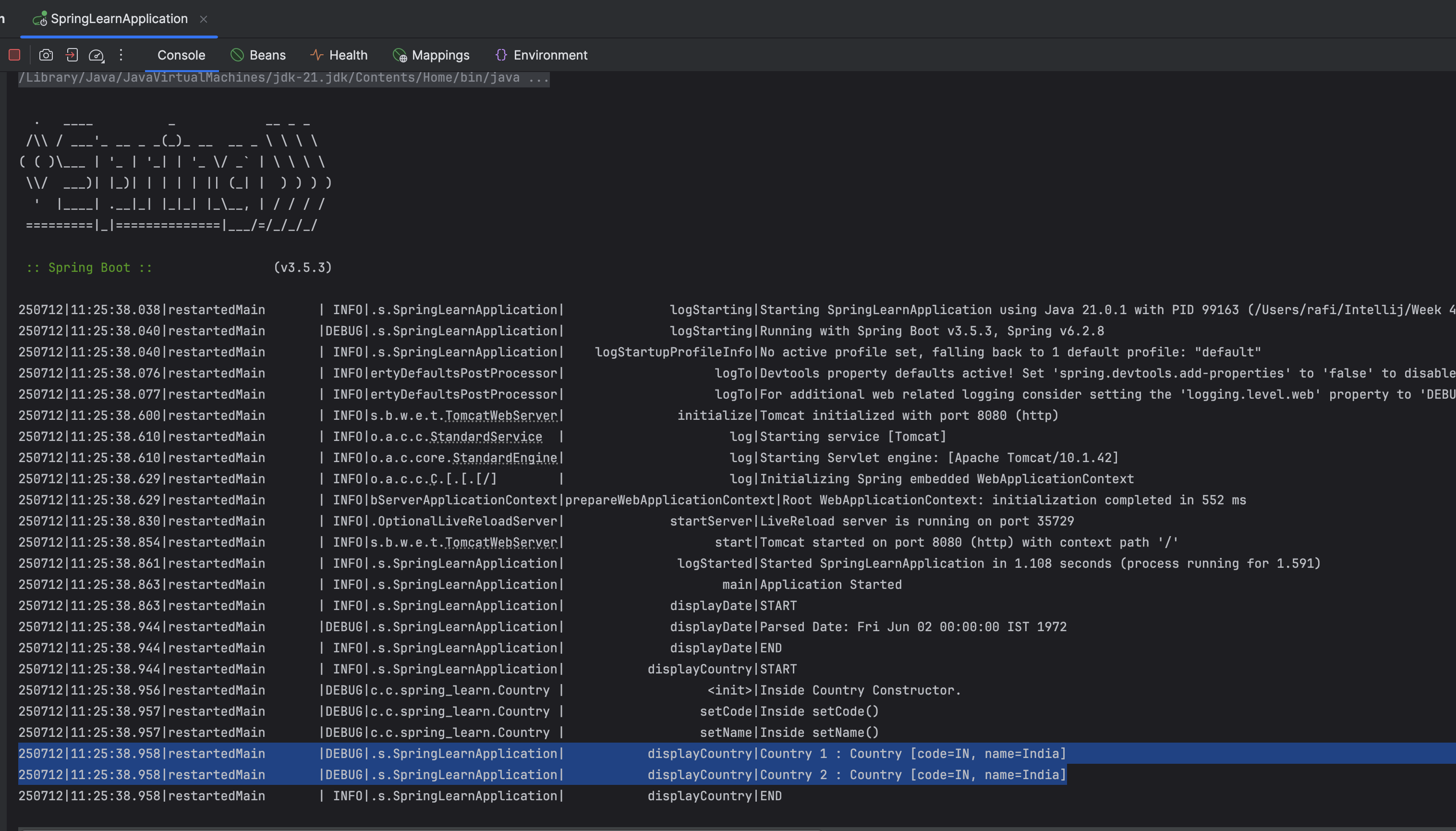
**SpringLearnApplication.java :**

**The same code , here we’ll be checking whether the objects are same objects or not.**

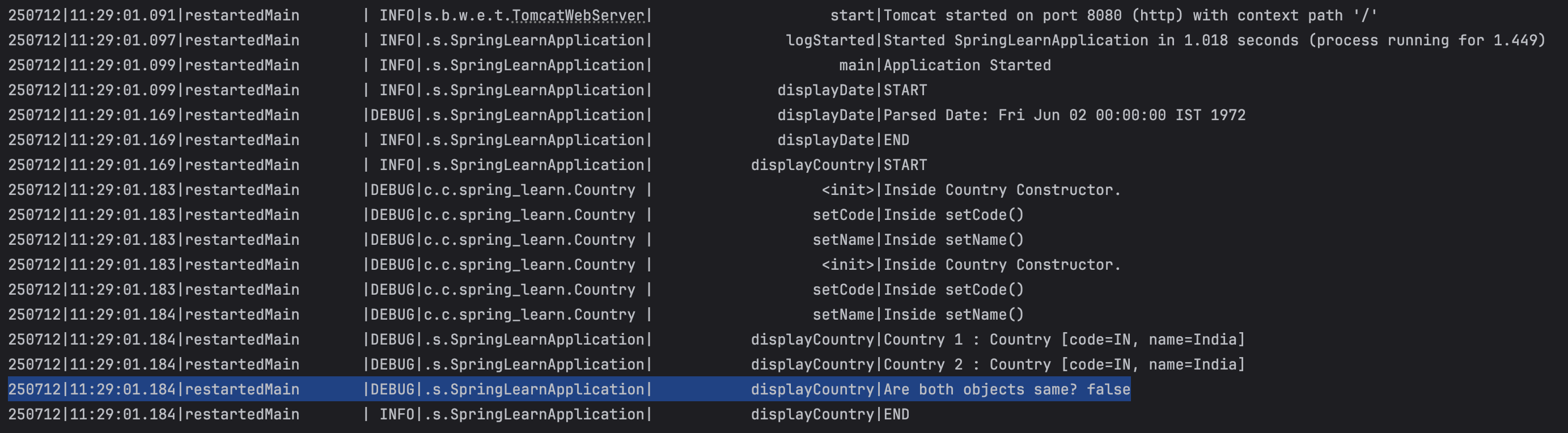
public static void displayCountry() {  
 LOGGER.info("START");  
  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
  
 Country country1 = context.getBean("country", Country.class);  
 Country country2 = context.getBean("country", Country.class);  
  
 LOGGER.debug("Country 1 : {}", country1);  
 LOGGER.debug("Country 2 : {}", country2);  
 LOGGER.debug("Are both objects same? {}", country1 == country2);  
  
 LOGGER.info("END");  
}

**OUTPUT :**

**Singleton Scope -**



**Prototype Scope -**



**Exercise 6 : Spring Core – Load list of countries from Spring Configuration XML**

**CODE:**

**resources/country.xml :**

<?xml version="1.0" encoding="UTF-8"?>

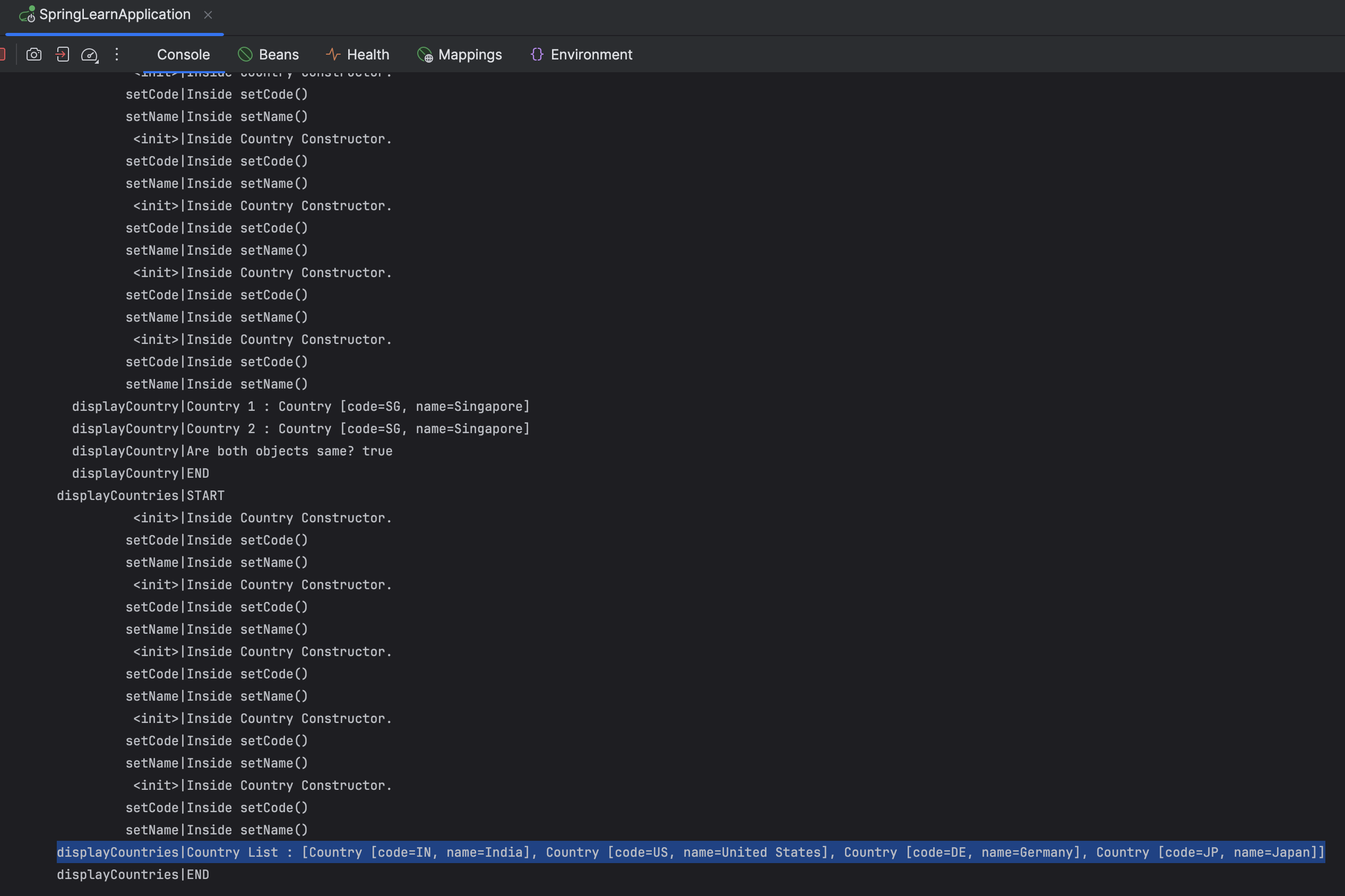
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans  
 https://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <bean id="country" class="com.cognizant.spring\_learn.Country" scope="singleton">  
 <property name="code" value="SG"/>  
 <property name="name" value="Singapore"/>  
 </bean>  
  
 <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="de" class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="DE"/>  
 <property name="name" value="Germany"/>  
 </bean>  
  
 <bean id="jp" class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="JP"/>  
 <property name="name" value="Japan"/>  
 </bean>  
  
 <bean id="countryList" class="java.util.ArrayList">  
 <constructor-arg>  
 <list>  
 <ref bean="in"/>  
 <ref bean="us"/>  
 <ref bean="de"/>  
 <ref bean="jp"/>  
 </list>  
 </constructor-arg>  
 </bean>

</beans>

**SpringLearnApplication.java :**

package com.cognizant.spring\_learn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.List;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);  
  
 public static void main(String[] args) throws Exception {  
 SpringApplication.run(SpringLearnApplication.class, args);  
 LOGGER.info("Application Started");  
  
 displayDate();  
 displayCountry();  
 displayCountries();  
 }  
  
 public static void displayDate() throws Exception {  
 LOGGER.info("START");  
  
 ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");  
 SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);  
  
 Date parsedDate = format.parse("02/06/1972");  
 LOGGER.debug("Parsed Date: {}", parsedDate);  
  
 LOGGER.info("END");  
 }  
  
 public static void displayCountry() {  
 LOGGER.info("START");  
  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
  
 Country country1 = context.getBean("country", Country.class);  
 Country country2 = context.getBean("country", Country.class);  
  
 LOGGER.debug("Country 1 : {}", country1);  
 LOGGER.debug("Country 2 : {}", country2);  
 LOGGER.debug("Are both objects same? {}", country1 == country2);  
  
 LOGGER.info("END");  
 }  
  
 public static void displayCountries() {  
 LOGGER.info("START");  
  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
  
 List<Country> countryList = context.getBean("countryList", List.class);  
 LOGGER.debug("Country List : {}", countryList);  
  
 LOGGER.info("END");  
 }  
}

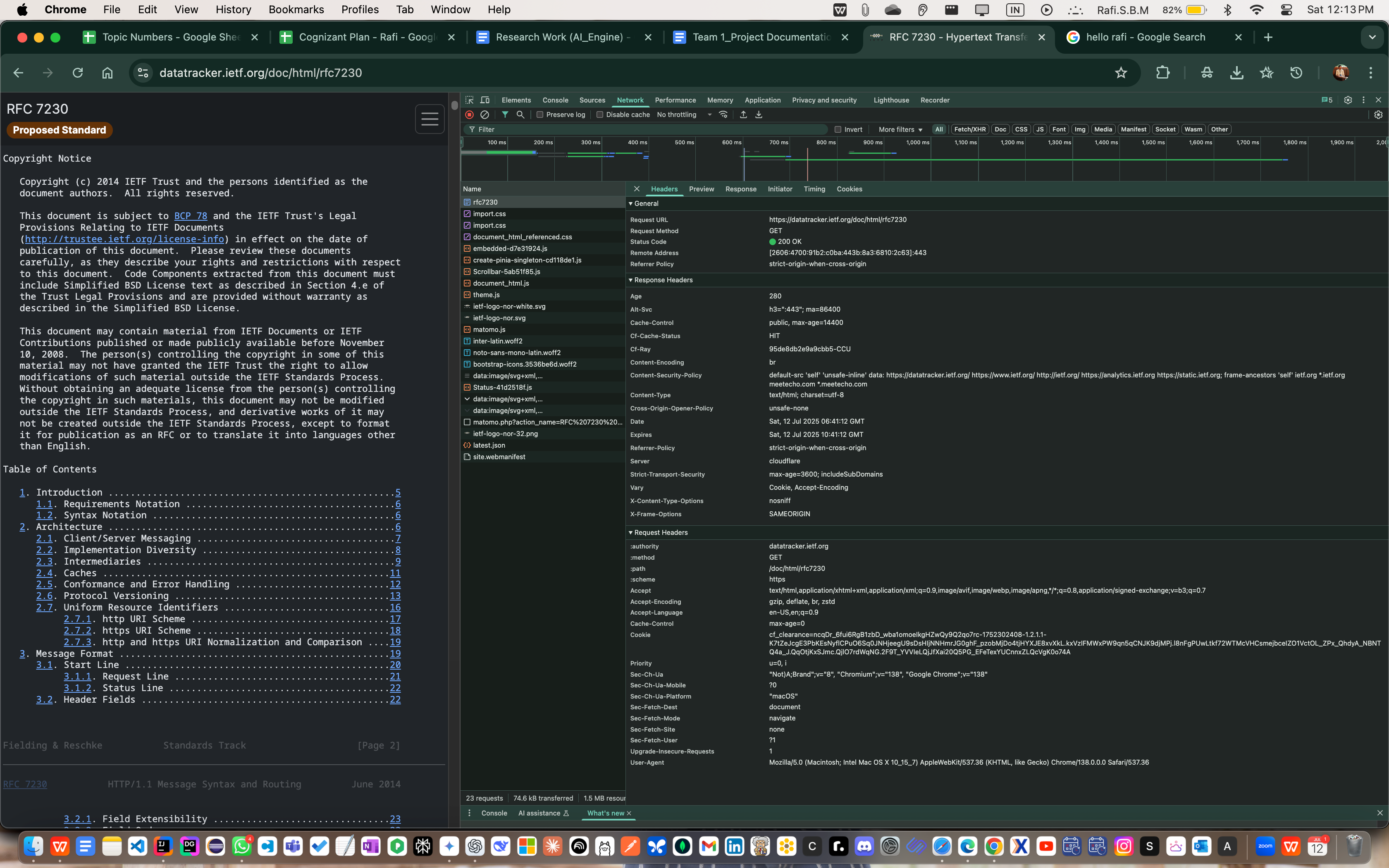
**OUTPUT:**



**File : 2. spring-rest-handson.docx**

**Exercise 1 : HTTP Request Response**

**OUTPUT :** General , Response & Request Headers.



**Exercise 2 : Hello World RESTful Web Service**

**CODE:**

**application.properties :**

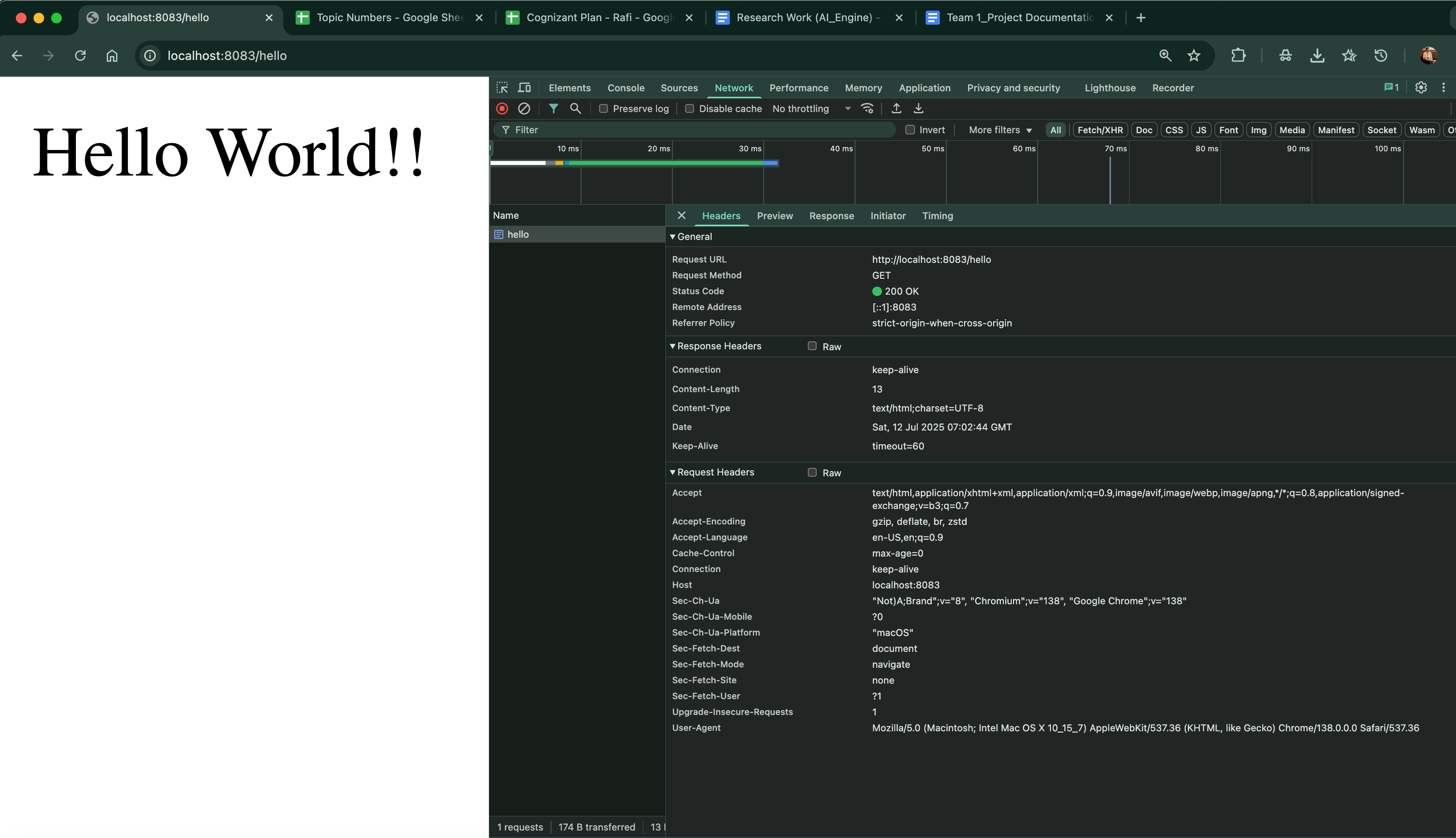
spring.application.name=spring-learn  
  
logging.level.org.springframework=info  
logging.level.com.cognizant.spring\_learn=debug  
logging.pattern.console=%d{yyMMdd}|%d{HH:mm:ss.SSS}|%-20.20thread|%5p|%-25.25logger{25}|%25M|%m%n  
  
server.port=8083

**com.cognizant.spring\_learn.controller/HelloController.java :**

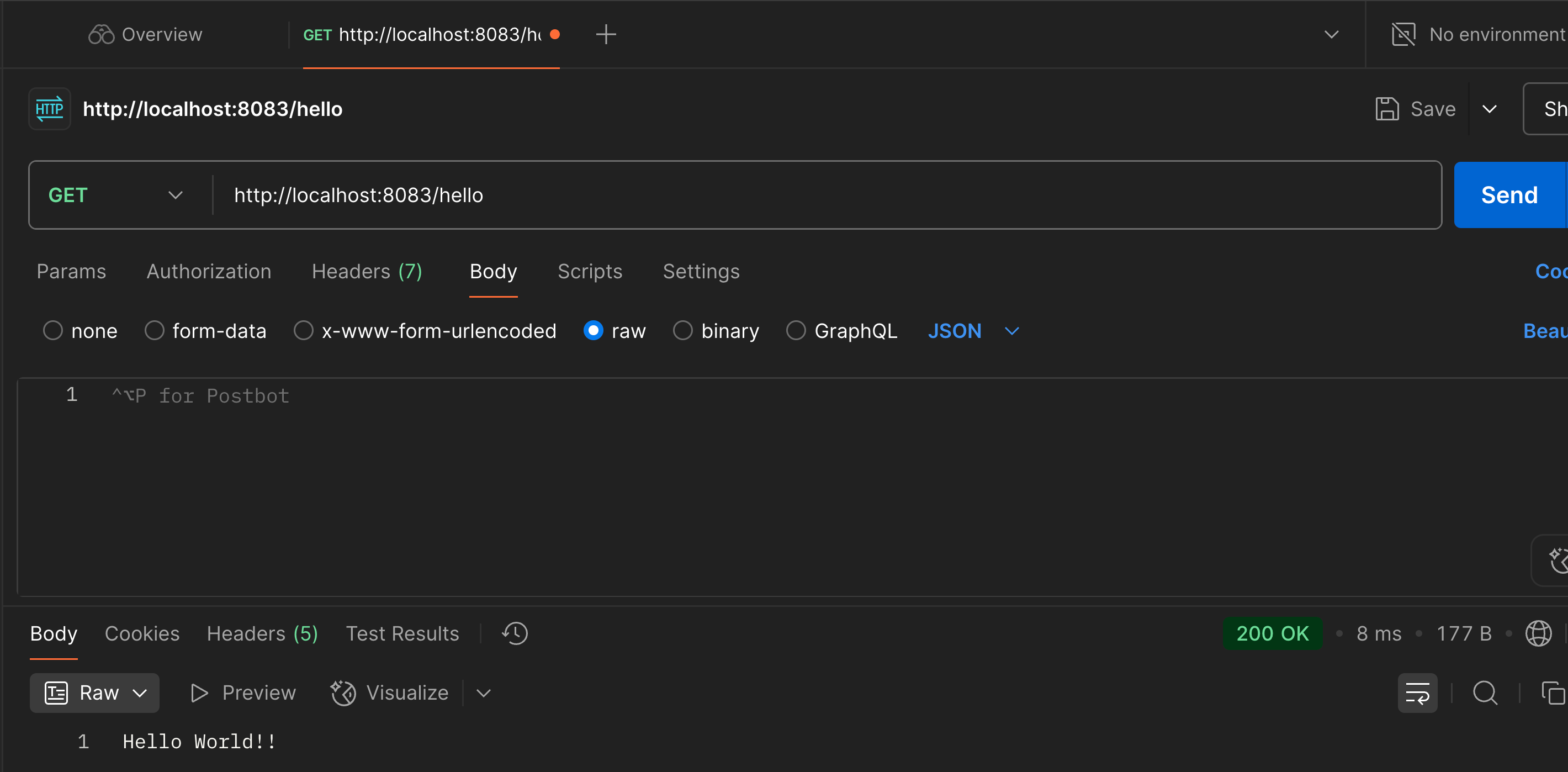
package com.cognizant.spring\_learn.controller;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class HelloController {

private static final Logger LOGGER = LoggerFactory.getLogger(HelloController.class);  
  
 @GetMapping("/hello")  
 public String sayHello() {  
 LOGGER.info("START");  
 String message = "Hello World!!";  
 LOGGER.info("END");  
 return message;  
 }  
}

**OUTPUT : DevTools**



**Postman -**



**Exercise 3 : REST - Country Web Service**

**com.cognizant.spring\_learn.controller/CountryController.java :**

package com.cognizant.spring\_learn.controller;  
  
import com.cognizant.spring\_learn.Country;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
@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 india = context.getBean("in", Country.class);  
  
 LOGGER.debug("Country : {}", india);  
  
 LOGGER.info("END");  
 return india;  
 }  
}

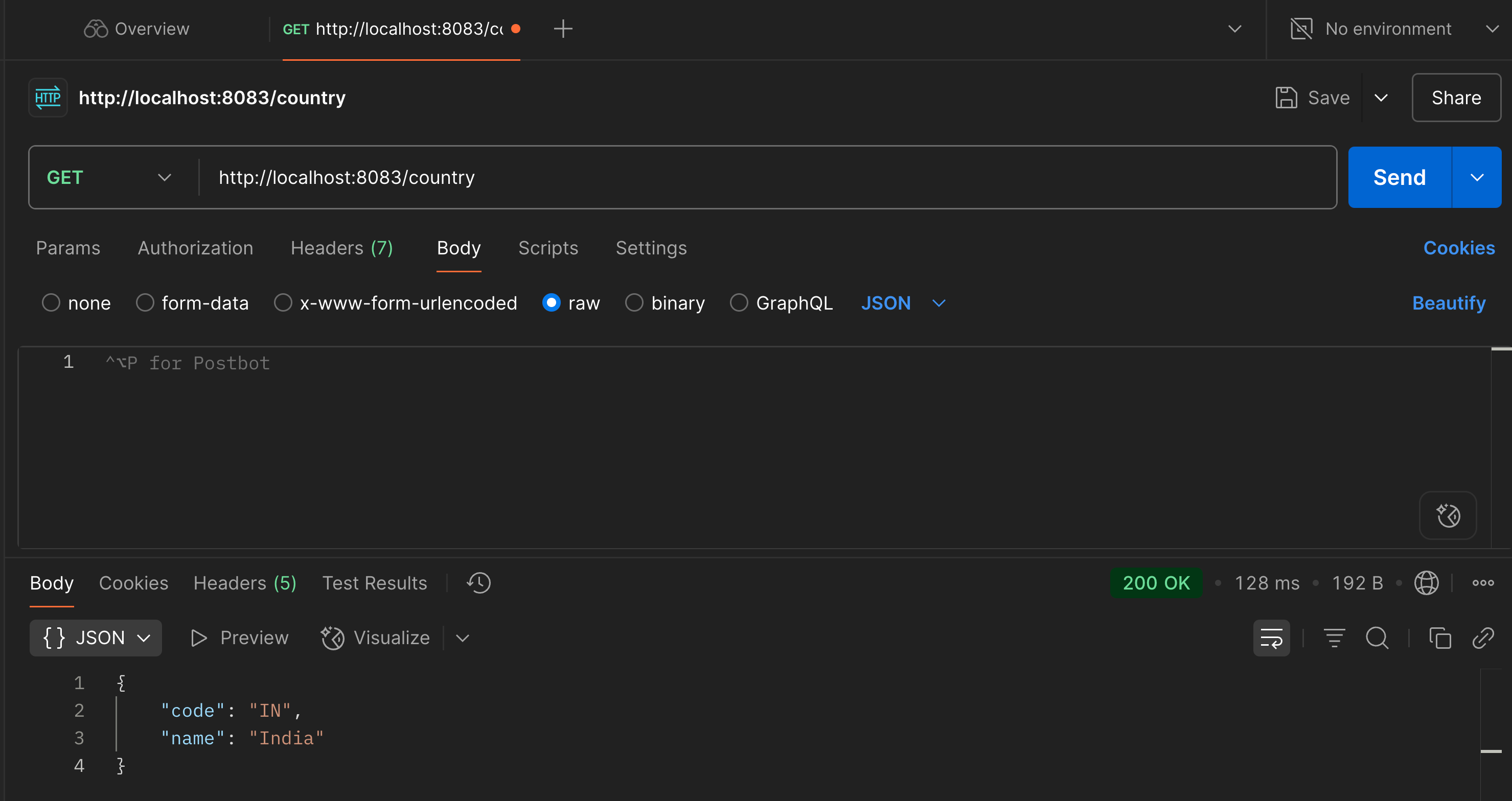
**resources/country.xml :**

<?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

**OUTPUT:**



**Exercise 4 : REST - Get all countries**

**CODE:**

**resources/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>  
  
 <bean id="us" class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="US"/>  
 <property name="name" value="United States"/>  
 </bean>  
  
 <bean id="de" class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="DE"/>  
 <property name="name" value="Germany"/>  
 </bean>  
  
 <bean id="jp" class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="JP"/>  
 <property name="name" value="Japan"/>  
 </bean>

<bean id="countryList" class="java.util.ArrayList">

<constructor-arg>  
 <list>  
 <ref bean="country"/>  
 <ref bean="us"/>  
 <ref bean="de"/>  
 <ref bean="jp"/>  
 </list>  
 </constructor-arg>

</bean>  
  
</beans>

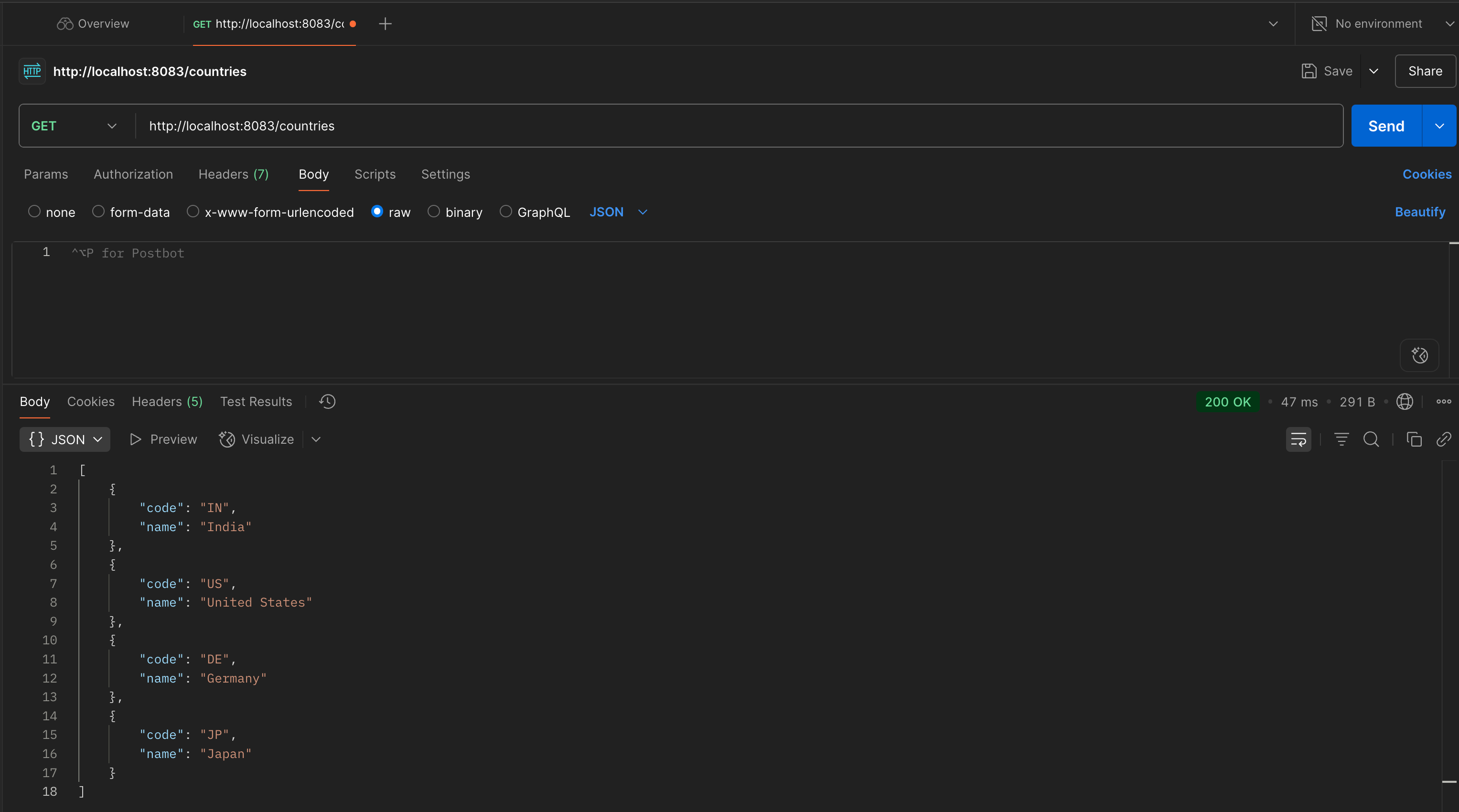
**CountryController.java :**

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.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
import java.util.List;  
  
@RestController  
public class CountryController {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);  
  
 @GetMapping("/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")  
 public List<Country> getAllCountries() {  
 LOGGER.info("START");

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 List<Country> countryList = context.getBean("countryList", List.class);  
  
 LOGGER.debug("Country List: {}", countryList);  
 LOGGER.info("END");  
  
 return countryList;  
 }  
}

**OUTPUT:**



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

**CODE:**

**CountryController.java :**

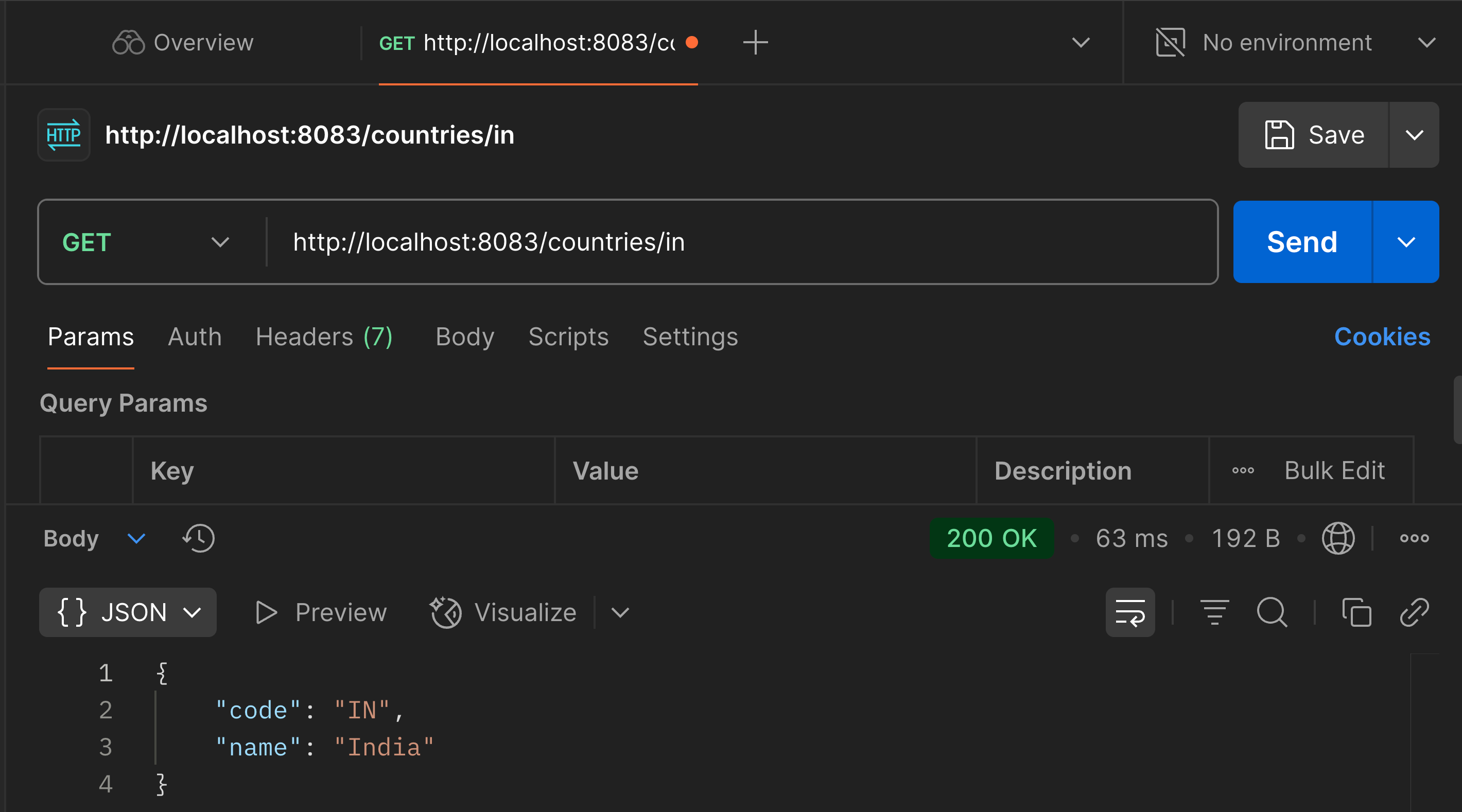
**Added a method -**

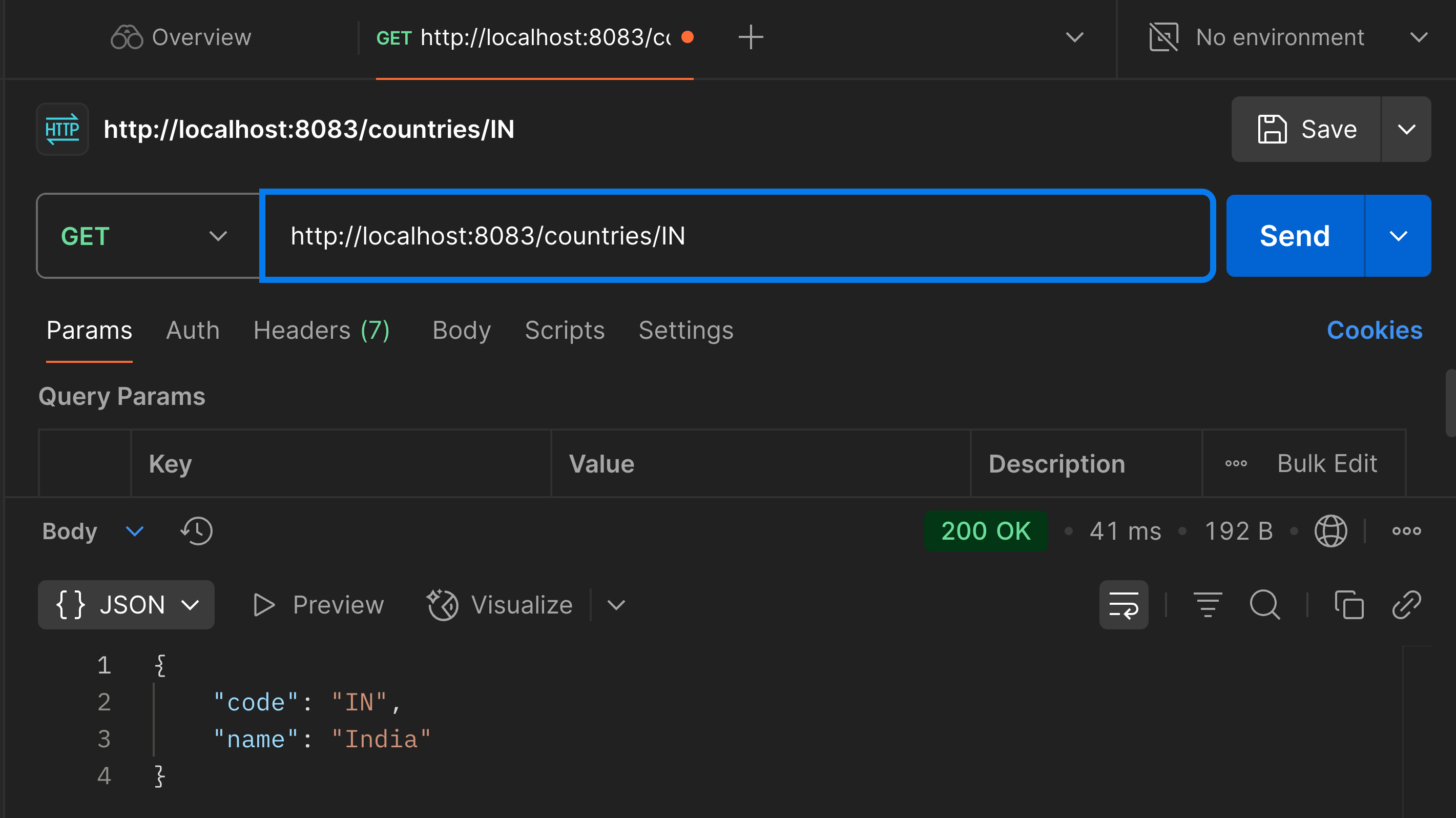
@GetMapping("/countries/{code}")  
 public Country getCountry(@PathVariable String code) {  
 LOGGER.info("START");  
 Country country = countryService.getCountry(code);  
 LOGGER.debug("Fetched Country for code {}: {}", code, country);  
 LOGGER.info("END");  
 return country;  
 }

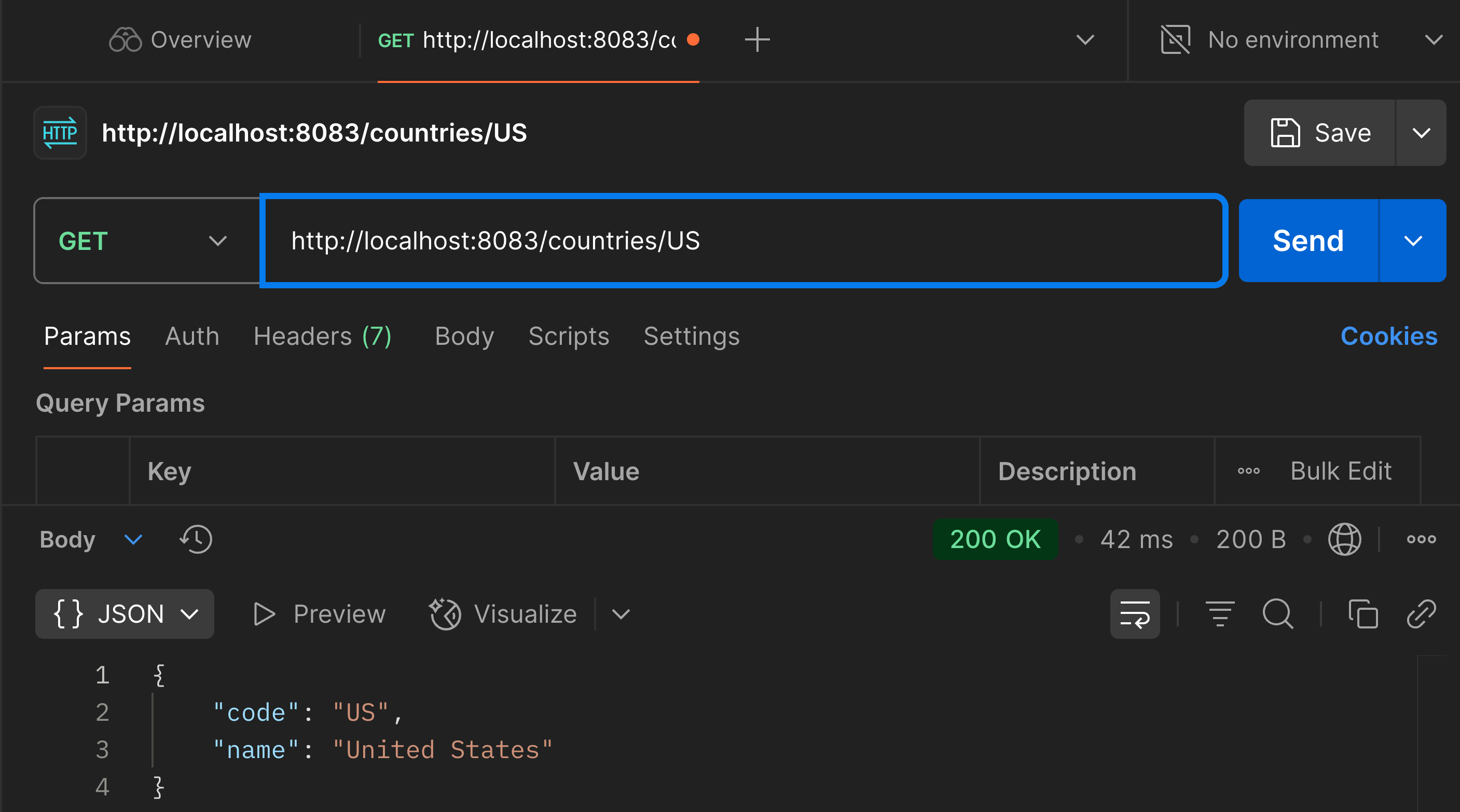
**service/CountryService.java :**

package com.cognizant.spring\_learn.service;  
  
import com.cognizant.spring\_learn.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> countries = context.getBean("countryList", List.class);  
  
 return countries.stream()  
 .filter(c -> c.getCode().equalsIgnoreCase(code))  
 .findFirst()  
 .orElse(null);  
 }  
}

**OUTPUT :**







**File : 5. JWT-handson**

**com.cognizant.spring\_learn.controller/AuthController.java :**

package com.cognizant.spring\_learn.controller;  
  
import com.cognizant.spring\_learn.security.JwtUtil;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.ResponseEntity;  
import org.springframework.security.core.Authentication;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.\*;  
  
@RestController  
public class AuthController {  
  
 @Autowired  
 private JwtUtil jwtUtil;  
  
 @GetMapping("/authenticate")  
 public ResponseEntity<Map<String, String>> authenticate(Authentication authentication) {  
 String username = authentication.getName();  
 String token = jwtUtil.generateToken(username);  
  
 Map<String, String> result = new HashMap<>();  
 result.put("token", token);  
  
 return ResponseEntity.ok(result);  
 }  
}

**JwtUtil.java :**

package com.cognizant.spring\_learn.security;  
  
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 String SECRET\_KEY = "ThisIsASecretKeyForJwtGeneration1234567890!";  
 private final long EXPIRATION\_TIME = 1000 \* 60 \* 60;  
  
 private Key getSigningKey() {  
 return Keys.hmacShaKeyFor(SECRET\_KEY.getBytes());  
 }  
  
 public String generateToken(String username) {  
 return Jwts.builder()  
 .setSubject(username)  
 .setIssuedAt(new Date(System.currentTimeMillis()))  
 .setExpiration(new Date(System.currentTimeMillis() + EXPIRATION\_TIME))  
 .signWith(getSigningKey(), SignatureAlgorithm.HS256)  
 .compact();  
 }  
  
 public String extractUsername(String token) {  
 return Jwts.parserBuilder()  
 .setSigningKey(getSigningKey())  
 .build()  
 .parseClaimsJws(token)  
 .getBody()  
 .getSubject();  
 }  
  
 public boolean validateToken(String token, String username) {  
 String extractedUsername = extractUsername(token);  
 return extractedUsername.equals(username) && !isTokenExpired(token);  
 }  
  
 private boolean isTokenExpired(String token) {  
 Date expiration = Jwts.parserBuilder()  
 .setSigningKey(getSigningKey())  
 .build()  
 .parseClaimsJws(token)  
 .getBody()  
 .getExpiration();  
 return expiration.before(new Date());  
 }  
}

**SecurityConfig.java :**

package com.cognizant.spring\_learn.security;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.security.authentication.AuthenticationManager;  
import org.springframework.security.authentication.AuthenticationManagerResolver;  
import org.springframework.security.config.annotation.authentication.configuration.AuthenticationConfiguration;  
import org.springframework.security.config.annotation.method.configuration.EnableMethodSecurity;  
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;  
import org.springframework.security.core.userdetails.User;  
import org.springframework.security.core.userdetails.UserDetailsService;  
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.config.annotation.web.builders.HttpSecurity;  
  
@Configuration  
@EnableWebSecurity  
@EnableMethodSecurity  
public class SecurityConfig {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(SecurityConfig.class);  
  
 @Bean  
 public PasswordEncoder passwordEncoder() {  
 LOGGER.info("Creating BCryptPasswordEncoder");  
 return new BCryptPasswordEncoder();  
 }  
  
 @Bean  
 public UserDetailsService userDetailsService(PasswordEncoder passwordEncoder) {  
 LOGGER.info("Creating in-memory users");  
 InMemoryUserDetailsManager manager = new InMemoryUserDetailsManager();  
 manager.createUser(User.withUsername("user").password(passwordEncoder.encode("pwd")).roles("USER").build());  
 manager.createUser(User.withUsername("admin").password(passwordEncoder.encode("pwd")).roles("ADMIN").build());  
 return manager;  
 }  
  
 @Bean  
 public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {  
 http.csrf(csrf -> csrf.disable())  
 .authorizeHttpRequests(auth -> auth  
 .requestMatchers("/authenticate").permitAll()  
 .requestMatchers("/countries/\*\*").hasRole("USER")  
 .anyRequest().authenticated()  
 )  
 .httpBasic();  
 return http.build();  
 }  
  
 @Bean  
 public AuthenticationManager authenticationManager(AuthenticationConfiguration config) throws Exception {  
 return config.getAuthenticationManager();  
 }  
}

**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/>  
</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>  
  
<properties>  
 <java.version>21</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-starter-security</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
  
 <dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt-api</artifactId>  
 <version>0.11.5</version>  
 </dependency>

<dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt-impl</artifactId>  
 <version>0.11.5</version>  
 <scope>runtime</scope>  
 </dependency>

<dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt-jackson</artifactId>  
 <version>0.11.5</version>  
 <scope>runtime</scope>  
 </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>

**OUTPUT :**

