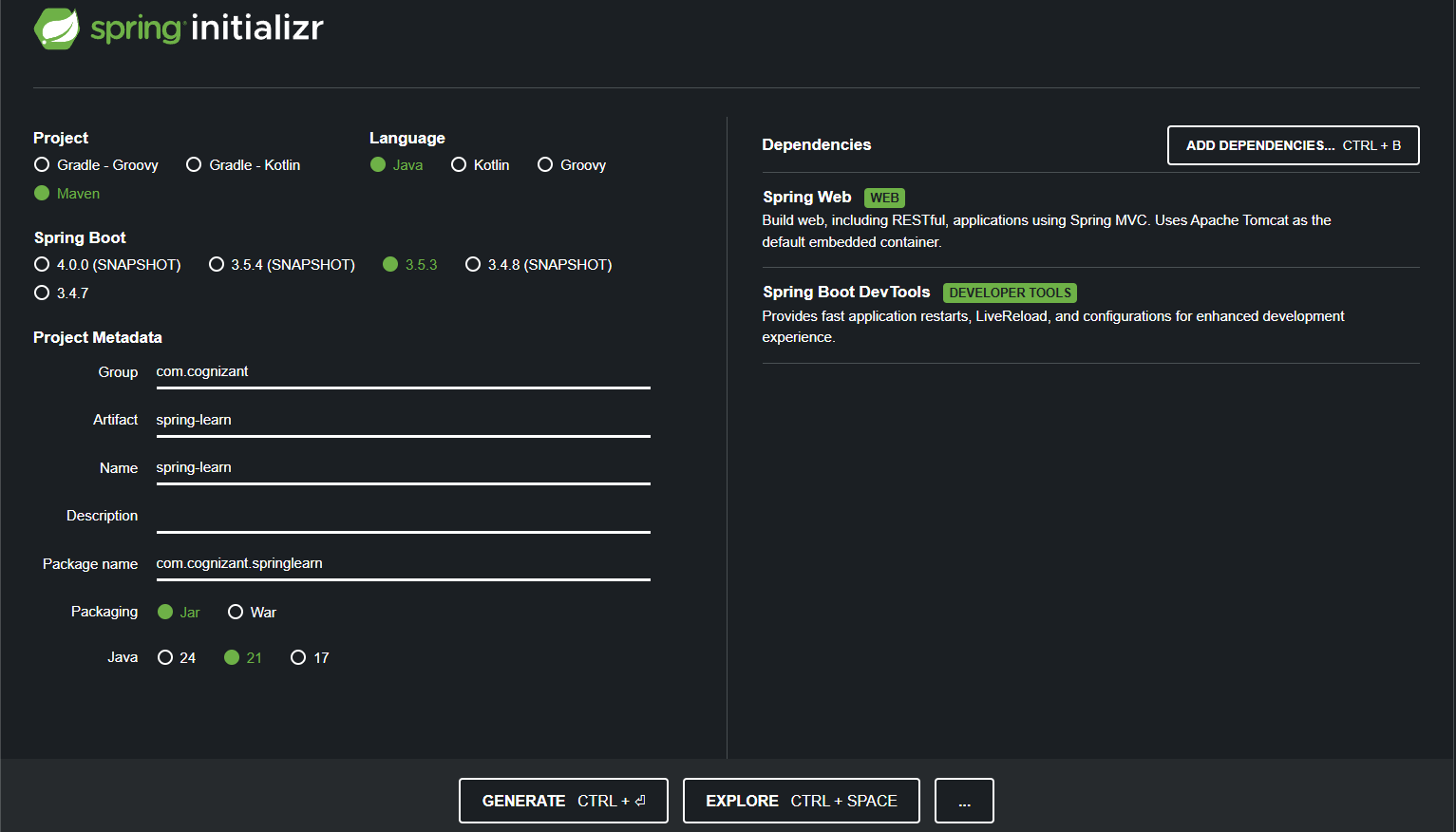
**Week 4 – Hands-on : Spring Rest hands-on**

Hands-on 1 : Create a Spring Web Project using maven

We follow the steps as listed in the document.

1. Use the Spring Initializr to add required dependencies, then generate and download the Spring boot starter project.



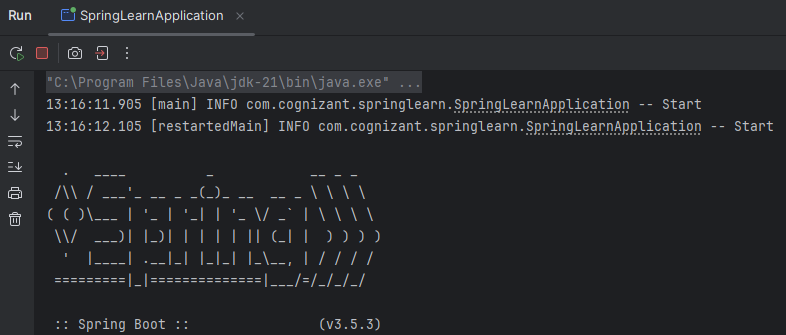
2. We extract and open the extracted folder in Intellij

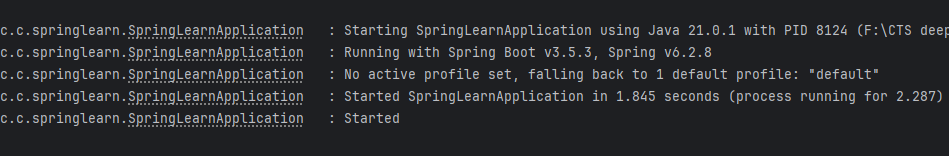
3. We skip the Proxy Setup step as it is only necessary when using Cognizant’s internal network or VPN which is not the case now.

4. Then we add the log info in the main class that is “SpringLearnApplication.java”

package com.cognizant.springlearn;  
  
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) {  
 *LOGGER*.info("Start");  
 SpringApplication.*run*(SpringLearnApplication.class, args);  
 *LOGGER*.info("Started");  
 }  
}

OUTPUT





Hands-on 2 : Spring Core – Load Country from Spring Configuration XML

1. First we create a country class in model package as “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;  
 }  
  
 @Override  
 public String toString() {  
 return "Country [code=" + code + ", name=" + name + "]";  
 }  
}

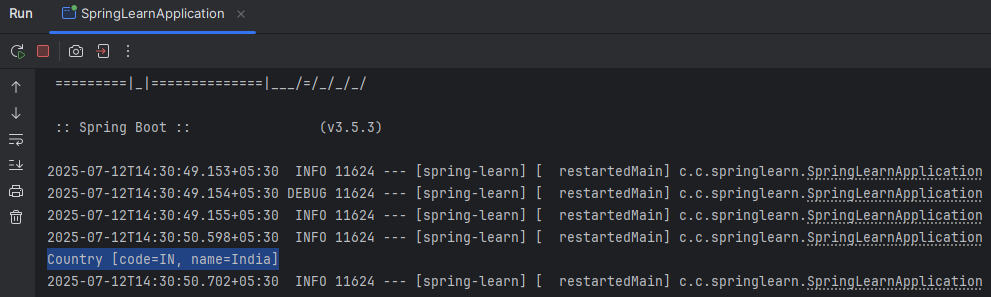
2. Under resources folder we create a new Spring XML config file for as “country.xml”

<?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  
 http://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <bean id="country" class="com.cognizant.springlearn.model.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
</beans>

3. Then we update our main method in “SpringLearnApplication.java” to load and display the country bean

package com.cognizant.springlearn;  
  
import com.cognizant.springlearn.model.Country;  
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) {  
 *LOGGER*.info("Start");  
 SpringApplication.*run*(SpringLearnApplication.class, args);  
 *displayCountry*();  
 *LOGGER*.info("Started");  
 }  
 private static void displayCountry() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = (Country) context.getBean("country", Country.class);  
 System.*out*.println(country.toString());  
 }  
}

OUTPUT



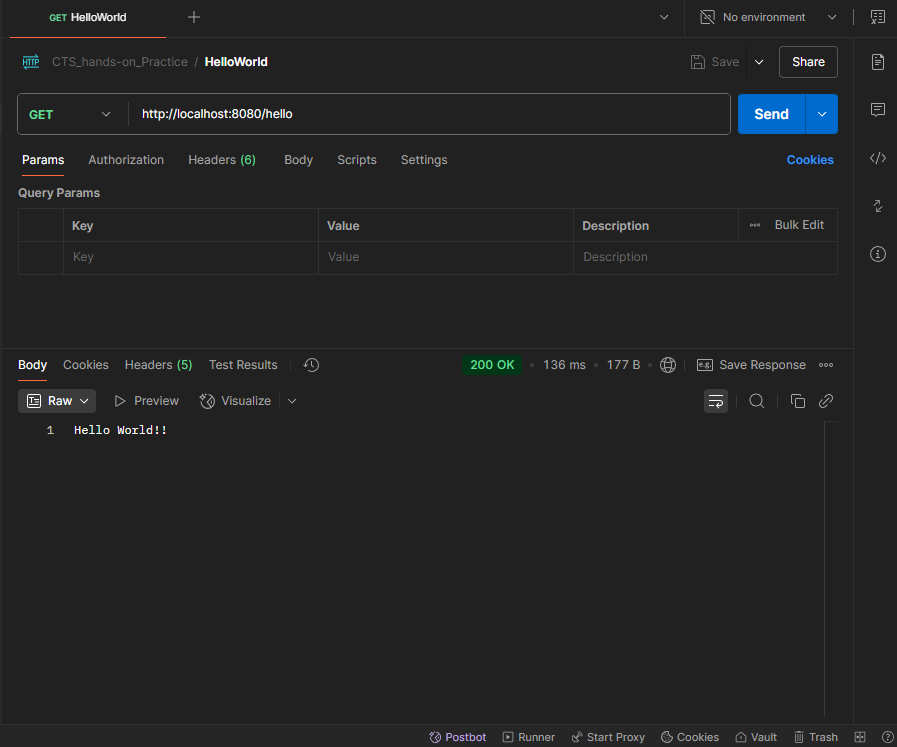
Hands-on 3 : Hello World RESTful Web Service

1. We first create a class “HelloController.java” inside controller package

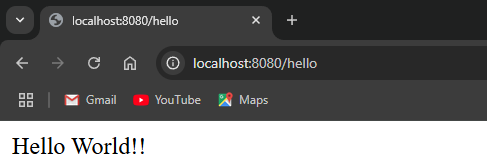
package com.cognizant.springlearn.controller;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class HelloController {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(HelloController.class);  
  
 @GetMapping("/hello")  
 public String sayHello() {  
 *LOGGER*.info("START");  
 *LOGGER*.info("END");  
 return "Hello World!!";  
 }  
}

After creating the method as given we run the Spring application and test the output in **Postman** and **Browser** window.

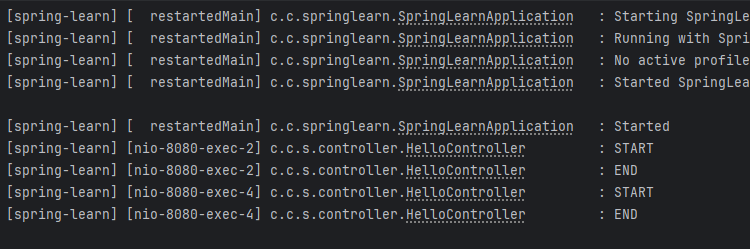
OUTPUT (Postman)



OUTPUT (Browser)



OUTPUT (IDE Console)



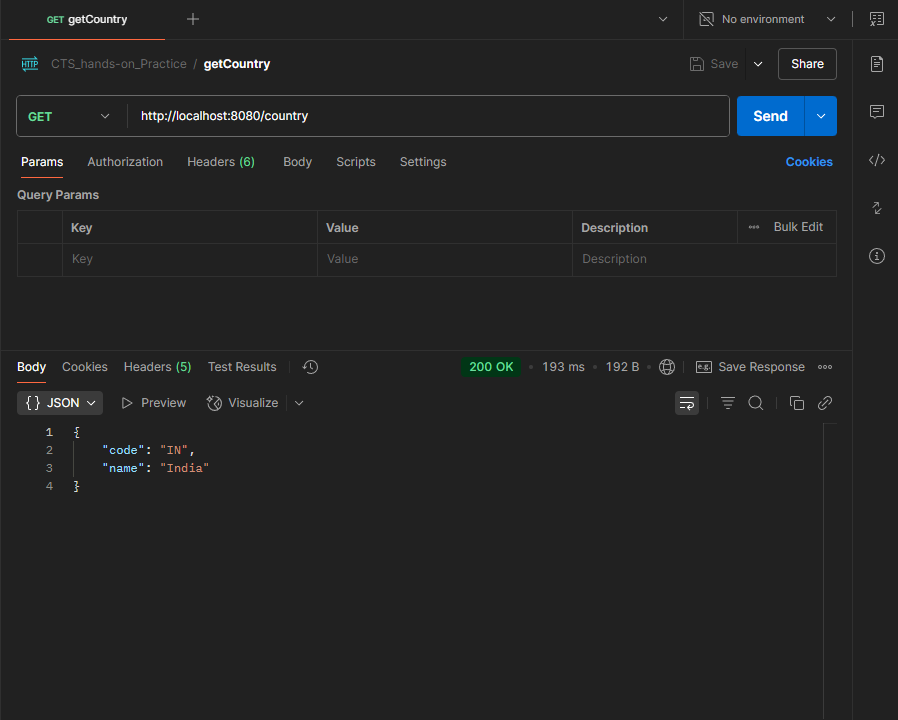
Hands-on 4 : REST – Country Web Service

1. We have already created “Country.java” class and “country.xml” file which we will reuse.

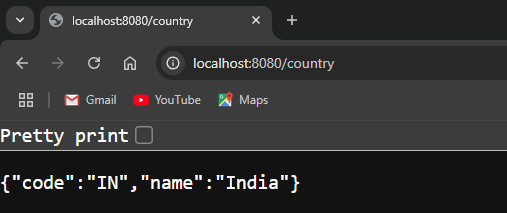
2. In the same controller package we create “CountryController.java” class file

package com.cognizant.springlearn.controller;  
  
import com.cognizant.springlearn.model.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;  
  
@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 = (Country) context.getBean("country");  
  
 *LOGGER*.info("END");  
 return country;  
 }  
}

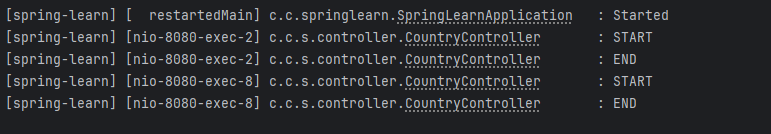
OUTPUT (Postman)



OUTPUT (Browser)

****

OUTPUT (IDE Console)

****

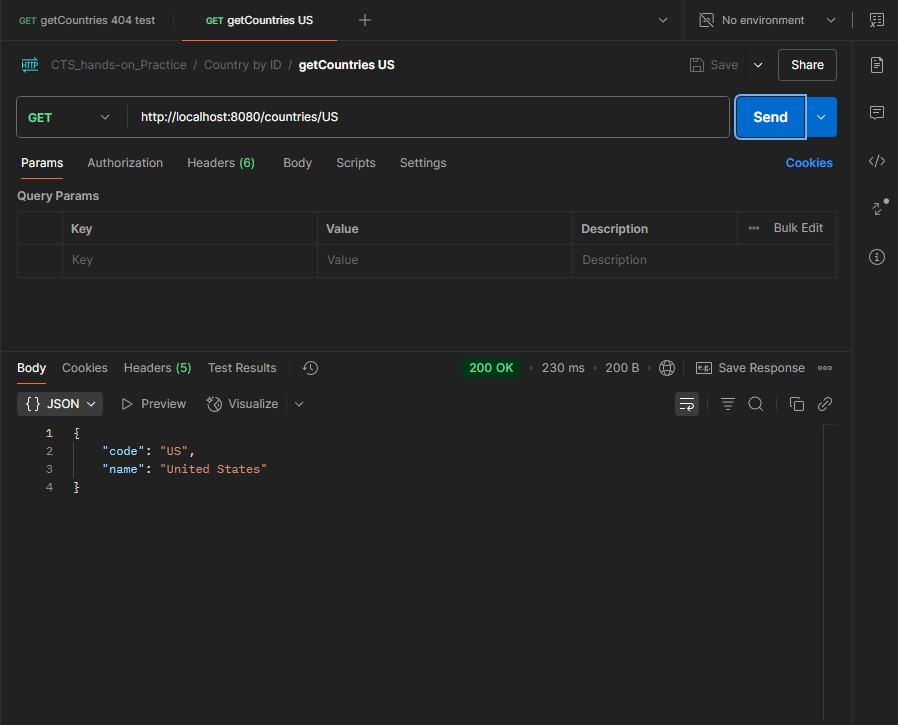
Hands-on 5 : REST – Get Country based on Country code

1. In the resources folder we create a new XML file “countryList.xml”

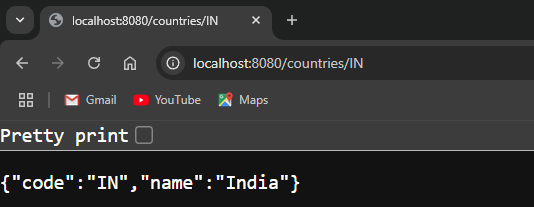
2. Update the existing CountryController class by adding new method “getCountryByCode”

@GetMapping("/countries/{code}")  
public ResponseEntity<Country> getCountryByCode(@PathVariable String code) {  
 *LOGGER*.info("START");  
  
 ApplicationContext context = new ClassPathXmlApplicationContext("countryList.xml");  
 List<Country> countryList = context.getBean("countryList", List.class);  
  
 for (Country country : countryList) {  
 if (country.getCode().equalsIgnoreCase(code)) {  
 *LOGGER*.info("END");  
 return new ResponseEntity<>(country, HttpStatus.*OK*);  
 }  
 }  
  
 *LOGGER*.warn("Country not found for code: {}", code);  
 return new ResponseEntity<>(HttpStatus.*NOT\_FOUND*);  
}

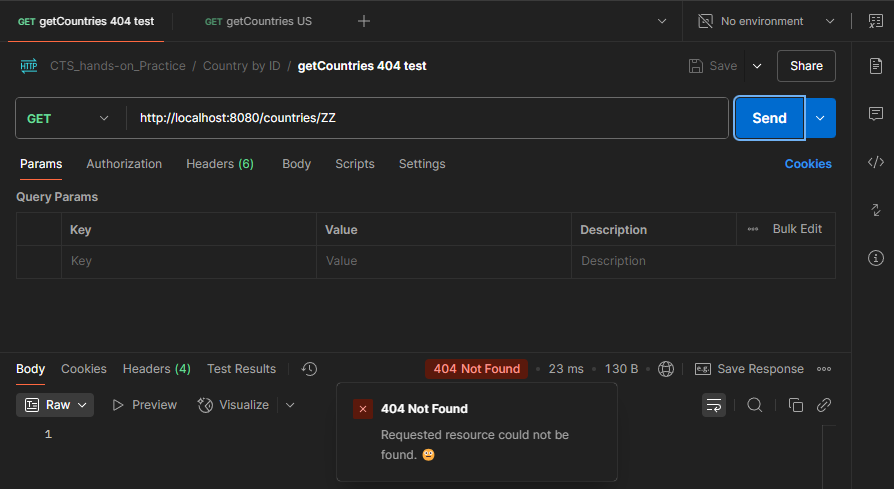
OUTPUT (Postman)



OUTPUT (Browser)



OUTPUT (Postman – Invalid country code)

****

OUTPUT (IDE console)

