**SPRING REST USING SPRING BOOT 3**

**Create a Spring Web Project using Maven**

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/> <!-- lookup parent from repository -->

</parent>

<groupId>com.cognizant</groupId>

<artifactId>spring-learn</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>spring-learn</name>

<description> Spring Web Project using Maven</description>

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

<java.version>17</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-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</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>

SpringLearnApplication.java

package com.example.springwebdemo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringWebDemoApplication {

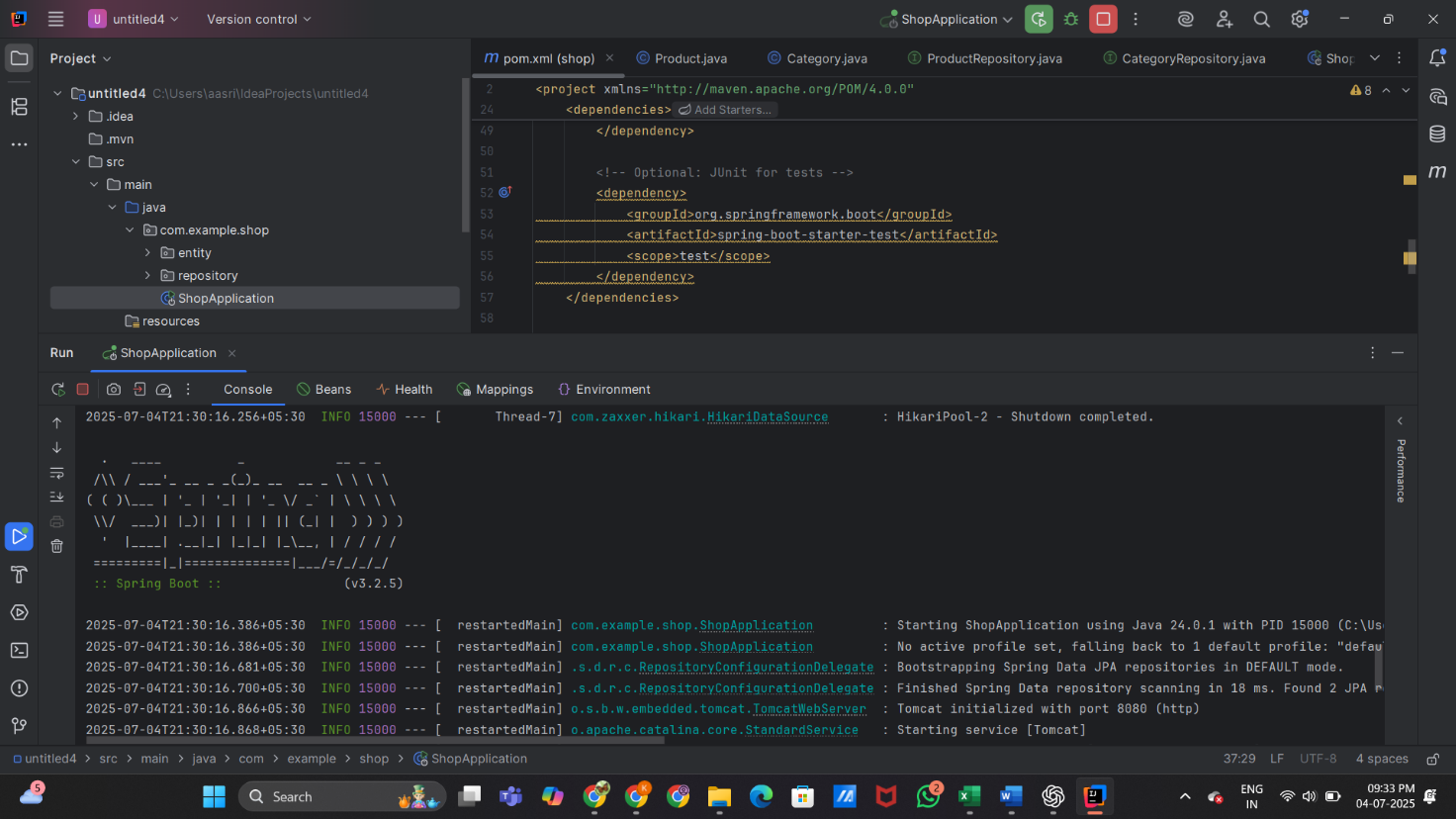
public static void main(String[] args) {

SpringApplication.run(SpringWebDemoApplication.class, args);

}

}

**Output:**

****

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

Country.java

package com.cognizant.spring\_learn;

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

http://www.springframework.org/schema/beans ">

<!--suppress SpringXmlModelInspection -->

<bean id="country" class="com.cognizant.spring\_learn.Country">

<!--suppress SpringXmlModelInspection -->

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

<!--suppress SpringXmlModelInspection -->

<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;

*@*SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) throws Exception {

LOGGER.info("START");

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

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

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

LOGGER.info("END");

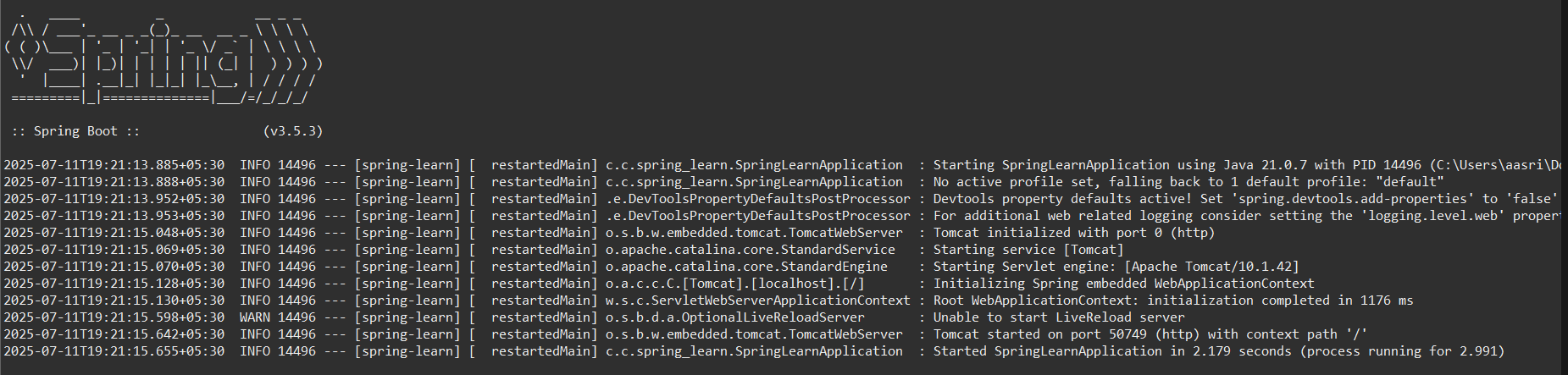
SpringApplication.run(SpringLearnApplication.class, args);

}

}

**Output:**

**Hello World RESTful Web Service**

HelloController.java

package com.cognizant.spring\_learn.controller;

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

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

@RestController

public class HelloController {

@GetMapping("/hello")

public String sayHello() {

return "Hello World!!";

}

}

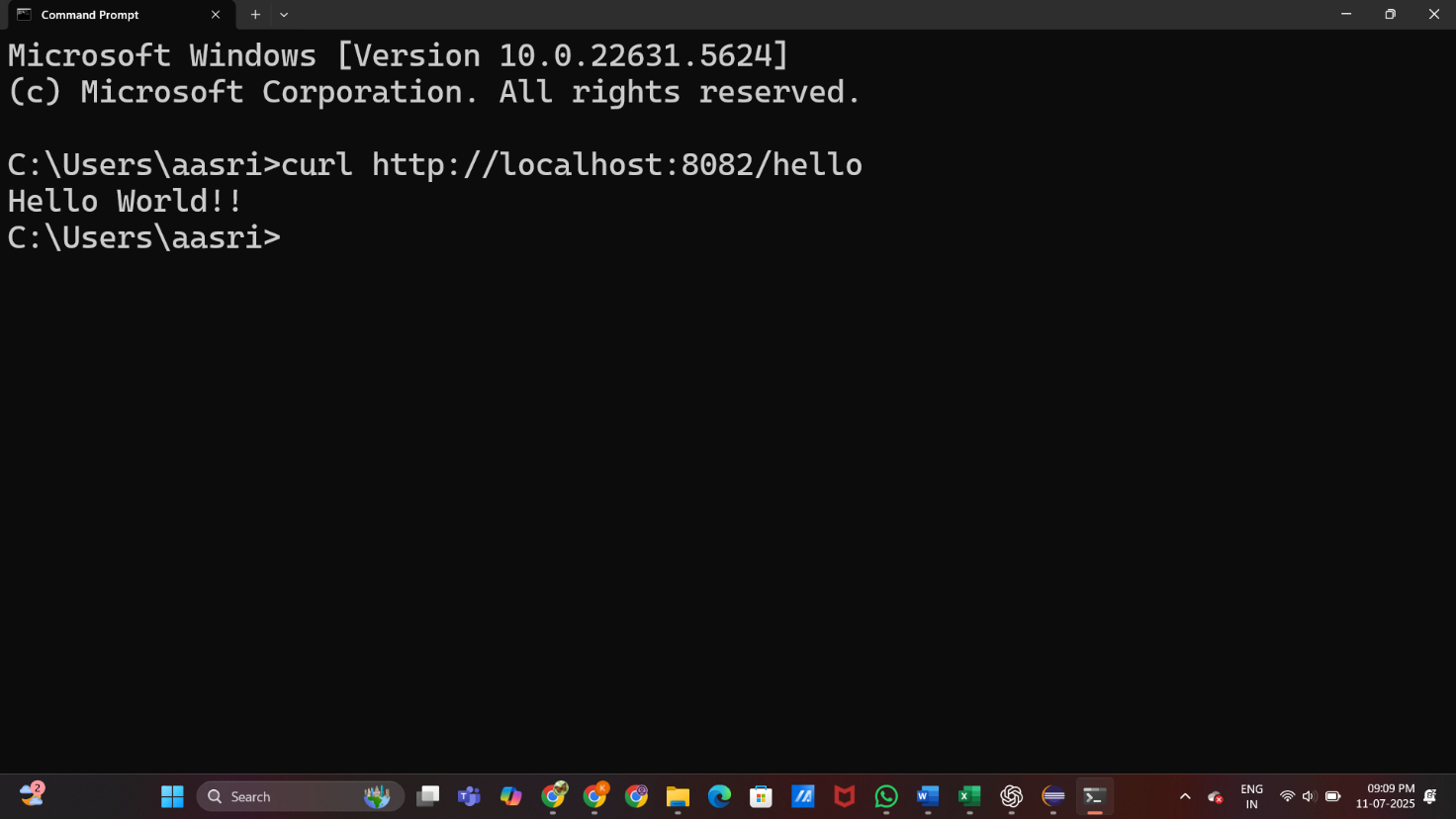
application.properties:

spring.application.name=spring-learn

server.port=8082

**Output:**

In Command Prompt:



**REST - Country Web Service**

Country.java

package com.cognizant.spring\_learn;

public class Country {

private String code;

private String name;

@Override

public String toString() {

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

}

}

CountryController.java

package com.cognizant.spring\_learn.controller;

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

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

import com.cognizant.spring\_learn.Country;

@RestController

public class CountryController {

@GetMapping("/country")

public Country getCountry() {

Country country = new Country();

country.setCode("IN");

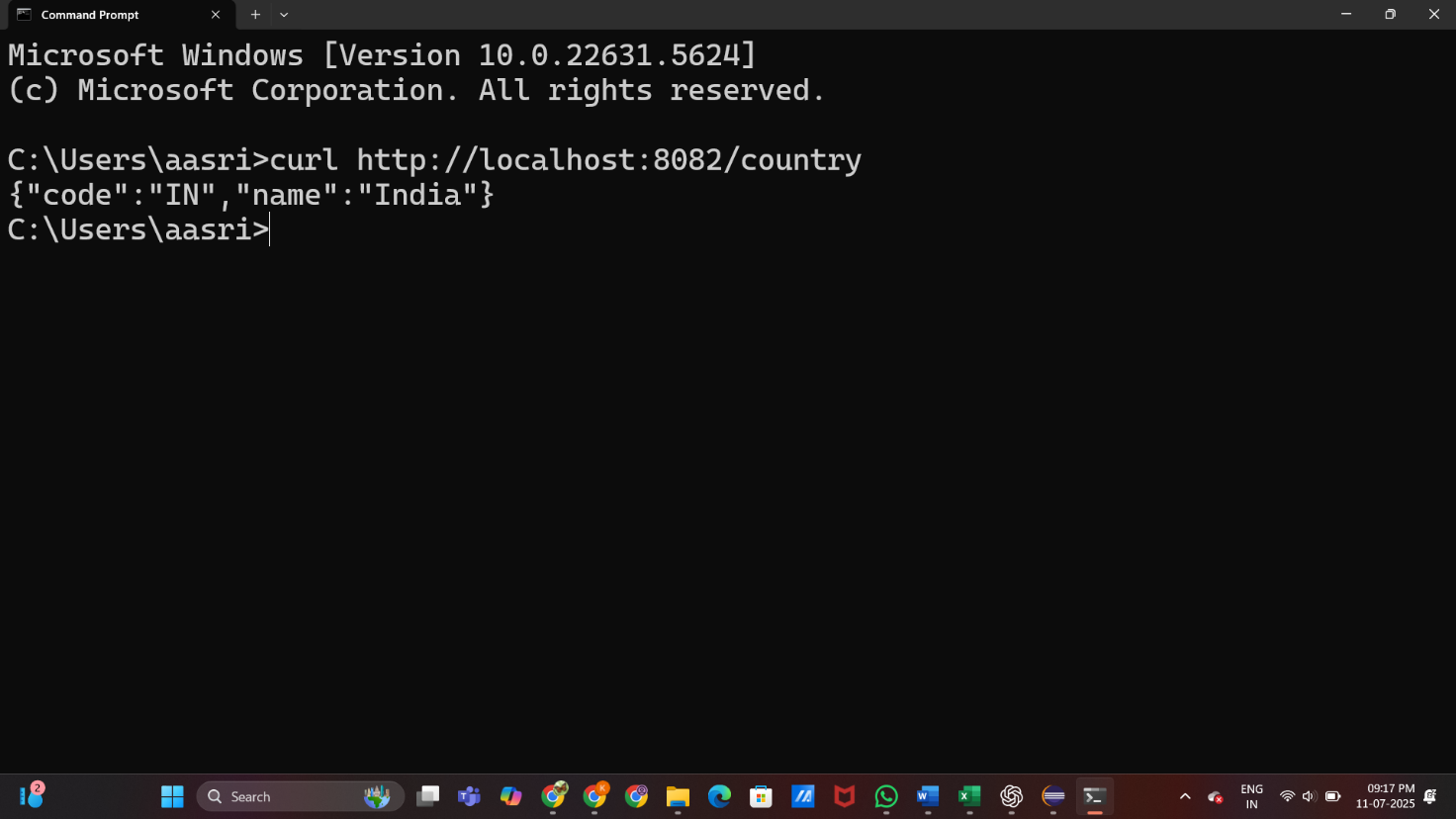
country.setName("India");

return country;

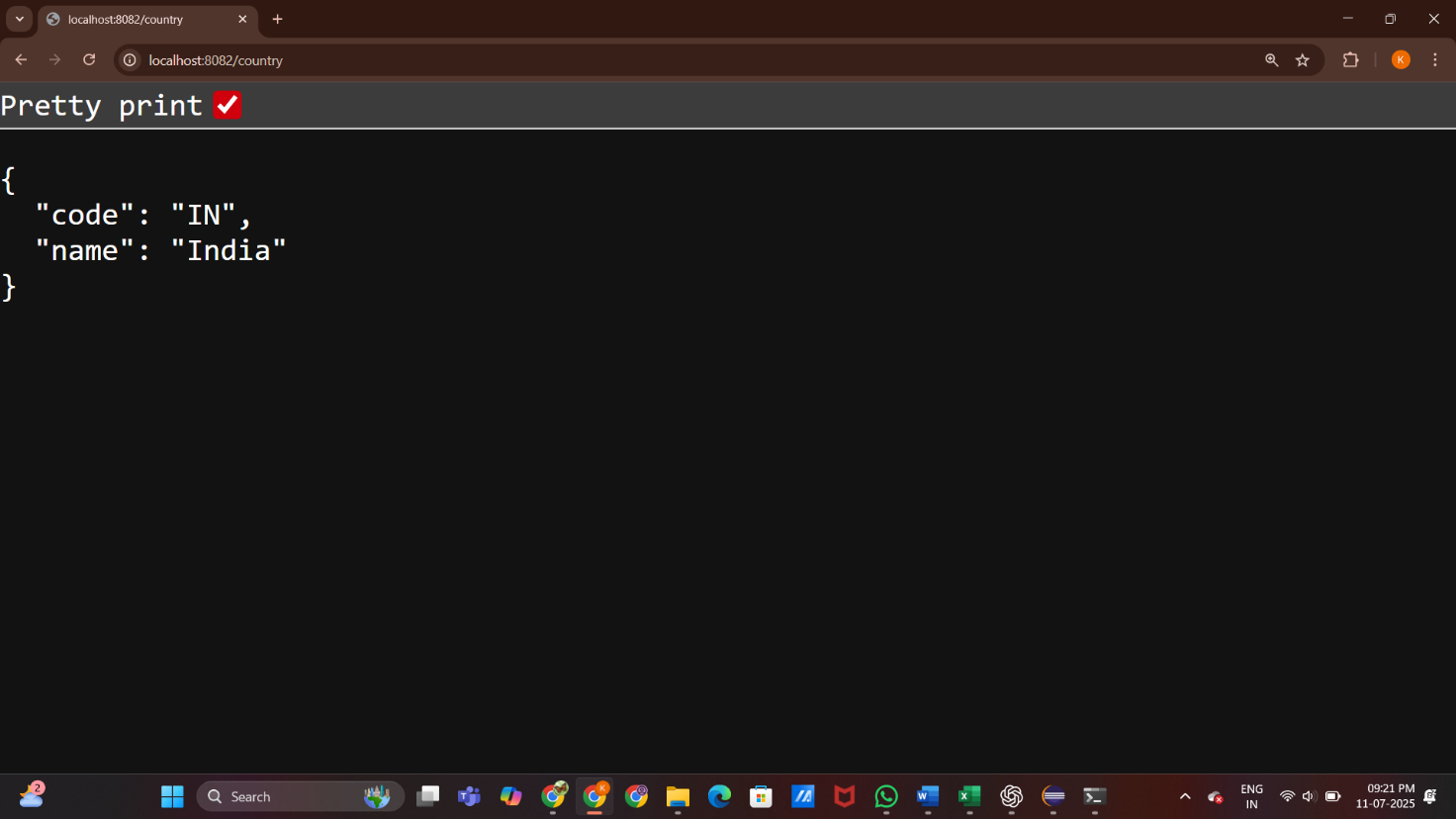
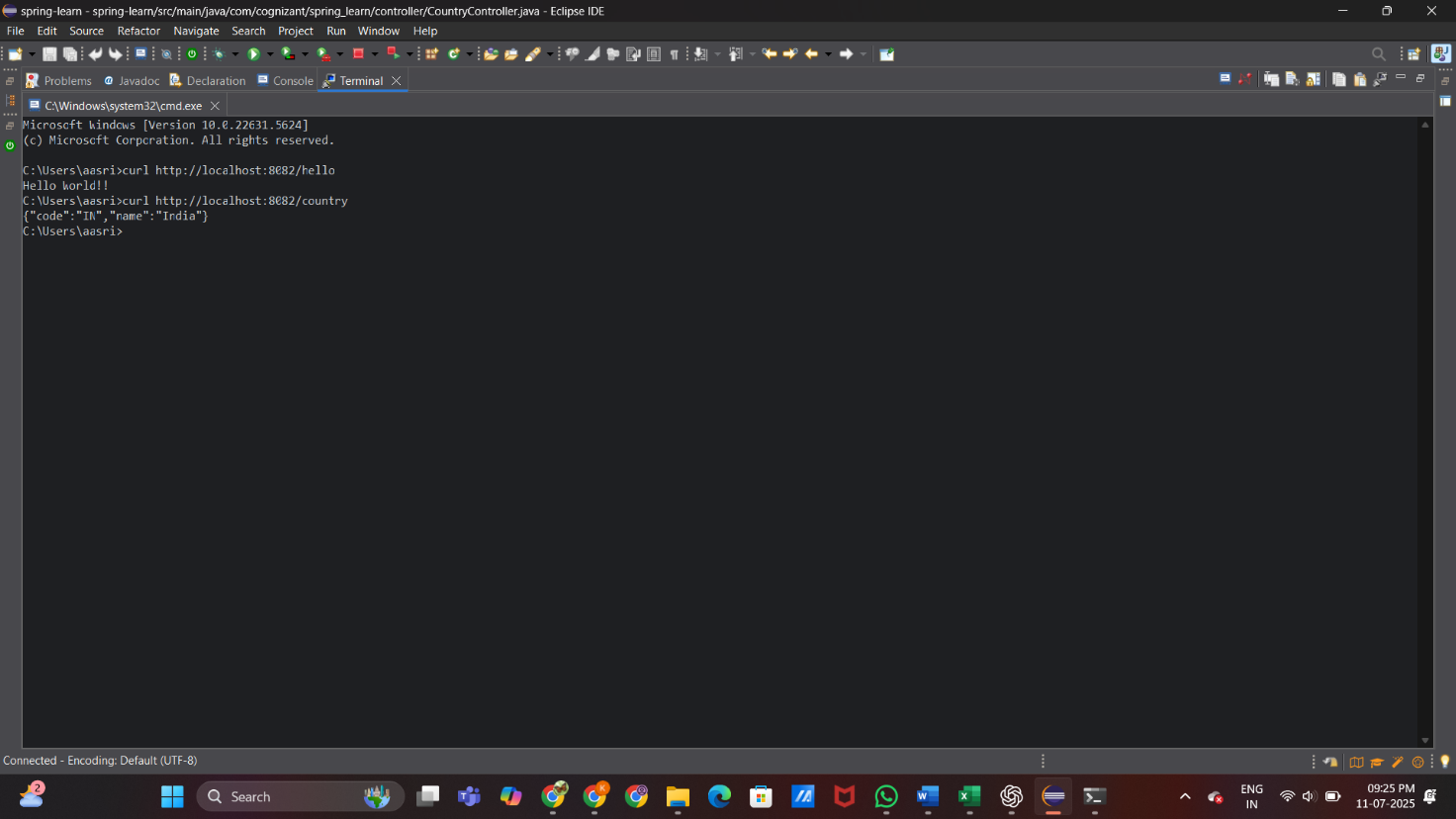
}

}

**Output:**



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

****CountryController.java

package com.cognizant.spring\_learn.controller;

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

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

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

import com.cognizant.spring\_learn.Country;

@RestController

public class CountryController {

@GetMapping("/country")

public Country getCountry() {

Country country = new Country();

country.setCode("IN");

country.setName("India");

return country;

}

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

public Country getCountryByCode(@PathVariable String code) {

Country country = new Country();

switch (code.toUpperCase()) {

case "IN":

country.setCode("IN");

country.setName("India");

break;

case "US":

country.setCode("US");

country.setName("United States");

break;

case "JP":

country.setCode("JP");

country.setName("Japan");

break;

case "FR":

country.setCode("FR");

country.setName("France");

break;

case "DE":

country.setCode("DE");

country.setName("Germany");

break;

case "IT":

country.setCode("IT");

country.setName("Italy");

break;

case "CN":

country.setCode("CN");

country.setName("China");

break;

case "BR":

country.setCode("BR");

country.setName("Brazil");

break;

default:

country.setCode("NA");

country.setName("Not Available");

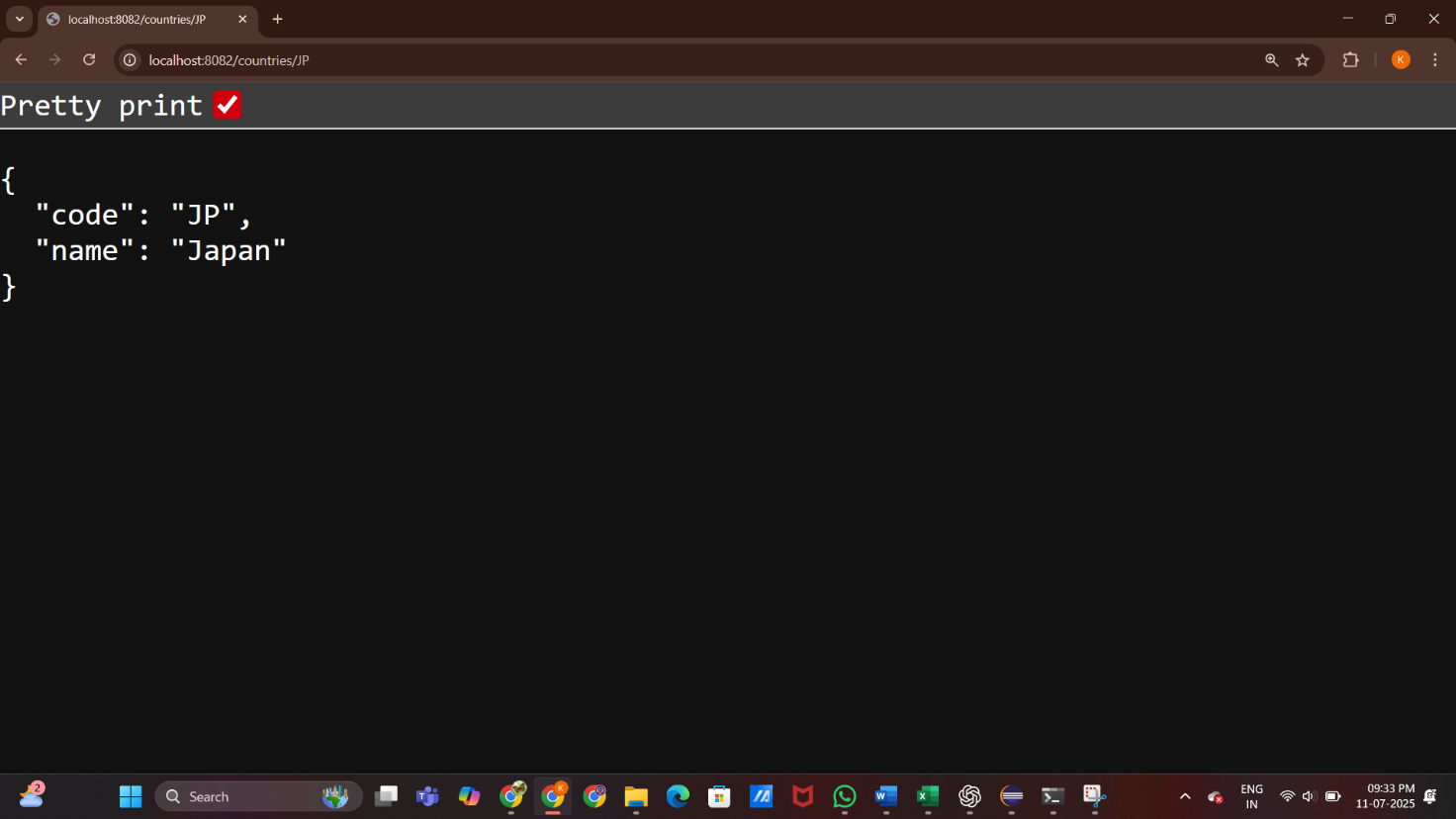
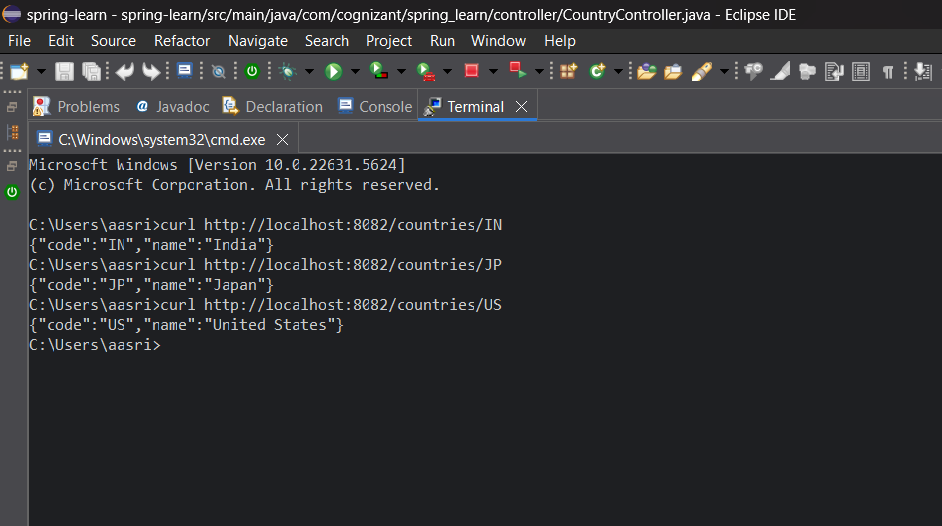
}

return country;

}

}

**Output:**

****