# Week 4 – Spring REST using Spring Boot 3

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

Code:

#### ****RestServiceApplication.java****

package com.example.restservice;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class RestServiceApplication {

public static void main(String[] args) {

SpringApplication.run(RestServiceApplication.class, args);

}

}

#### ****Greeting.java****

package com.example.restservice;

public class Greeting {

private final long id;

private final String content;

public Greeting(long id, String content) {

this.id = id;

this.content = content;

}

public long getId() {

return id;

}

public String getContent() {

return content;

}

}

#### ****GreetingController.java****

package com.example.restservice;

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

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

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

import java.util.concurrent.atomic.AtomicLong;

@RestController

public class GreetingController {

private final AtomicLong counter = new AtomicLong();

@GetMapping("/greeting")

public Greeting greeting(@RequestParam(value = "name", defaultValue = "World") String name) {

return new Greeting(counter.incrementAndGet(), String.format("Hello, %s!", name));

}

}

**GET Request to:**  
http://localhost:8080/greeting

**Response:**

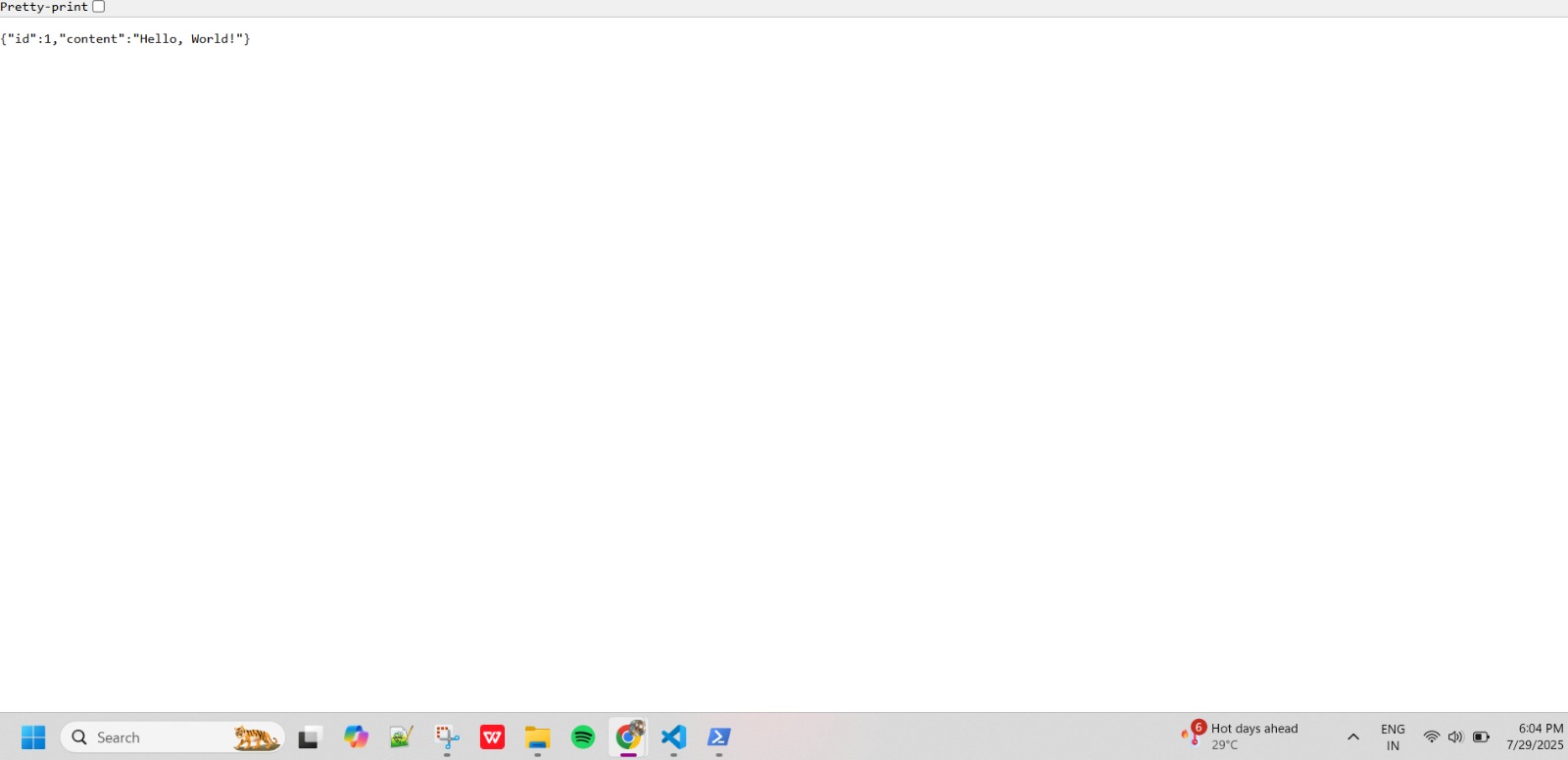
{

"id": 1,

"content": "Hello, World!"

}

Output:



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

Code:

#### Country.java

package com.example.restservice;

public class Country {

private String code;

private String name;

public Country() {

System.out.println("Inside Country Constructor.");

}

public String getCode() {

System.out.println("Getting country code: " + code);

return code;

}

public void setCode(String code) {

System.out.println("Setting country code: " + code);

this.code = code;

}

public String getName() {

System.out.println("Getting country name: " + name);

return name;

}

public void setName(String name) {

System.out.println("Setting country name: " + name);

this.name = name;

}

@Override

public String toString() {

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

}

}

#### RestServiceApplication.java

package com.example.restservice;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.Bean;

@SpringBootApplication

public class RestServiceApplication {

public static void main(String[] args) {

ApplicationContext context = SpringApplication.run(RestServiceApplication.class, args);

// Manually get the Country bean and call its methods

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

// Setting values

country.setCode("IN");

country.setName("India");

// Getting values (triggers getters with print)

String code = country.getCode();

String name = country.getName();

System.out.println("Country : " + country);

}

@Bean

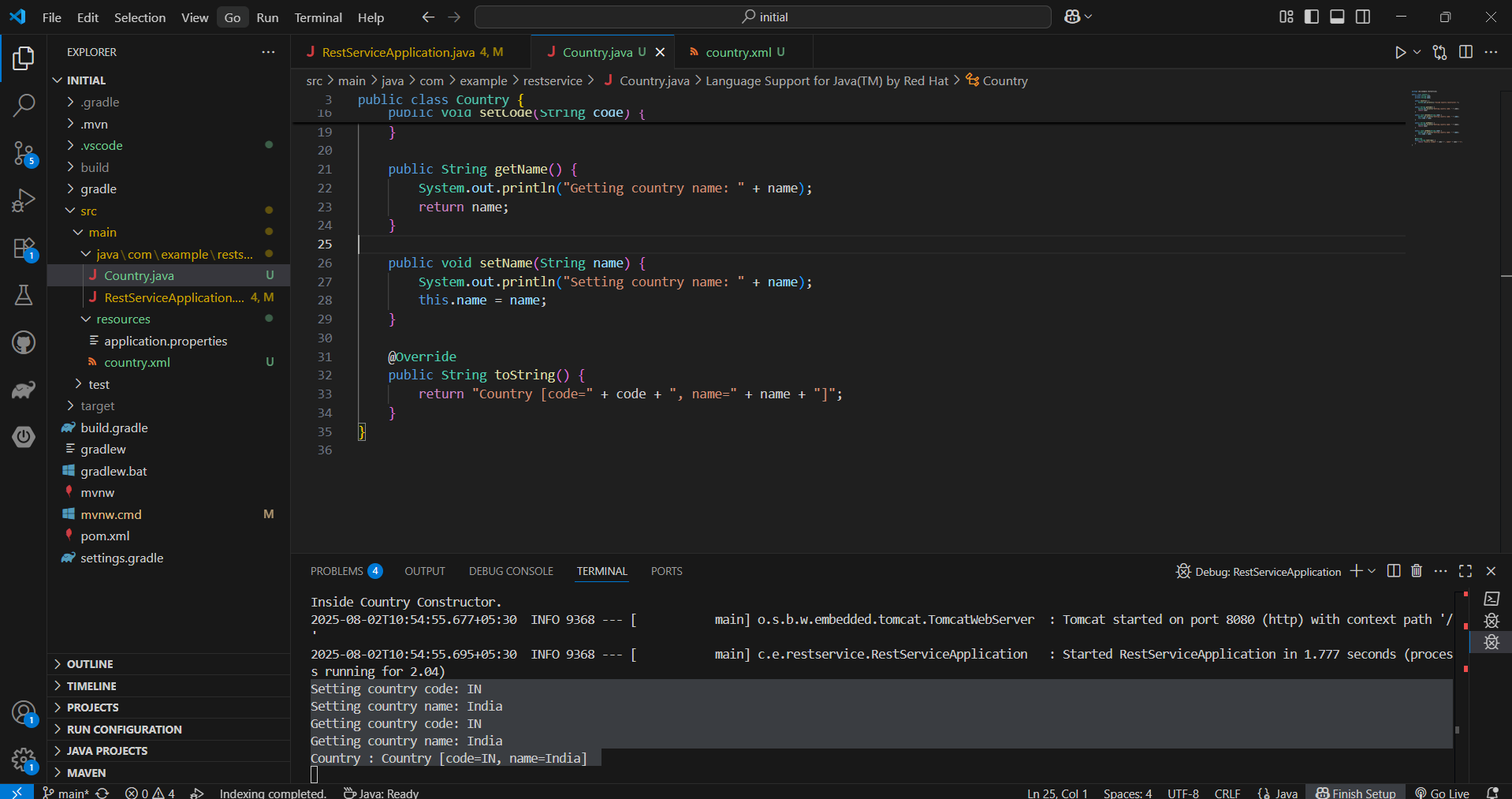
public Country country() {

return new Country(); // Initially empty

}

}

Output:



## Exercise 3: Hello World RESTful Web Service

Code:

### pom.xml

<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

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.cognizant</groupId>

<artifactId>spring-learn</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>spring-learn</name>

<description>Spring Boot Hello App</description>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.1.2</version>

</parent>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

</dependencies>

<properties>

<java.version>17</java.version>

</properties>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

### application.properties

**Path**: src/main/resources/application.properties

### SpringLearnApplication.java

**Path**: src/main/java/com/cognizant/springlearn/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);

}

}

### HelloController.java

**Path**: src/main/java/com/cognizant/springlearn/controller/HelloController.java

package com.cognizant.springlearn.controller;

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

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

@RestController

public class HelloController {

@GetMapping("/hello")

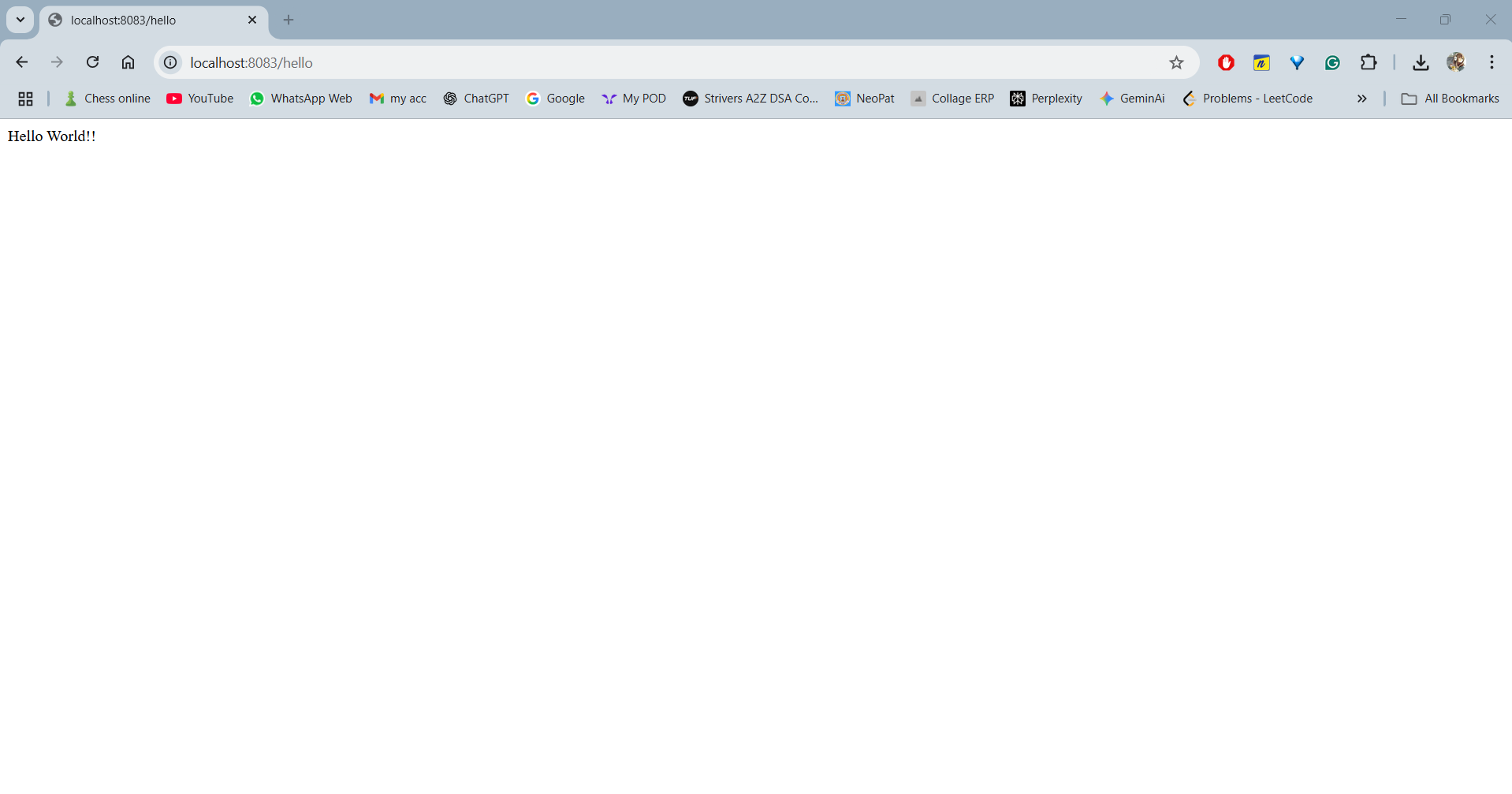
public String sayHello() {

return "Hello World!!";

}

}

Output:



## Exercise 4: REST - Country Web Service

Code:

### 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;

}

}

### spring.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>

### CountryController.java

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.model.Country;

import org.springframework.beans.factory.xml.XmlBeanFactory;

import org.springframework.core.io.ClassPathResource;

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

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

@RestController

public class CountryController {

@RequestMapping("/country")

public Country getCountryIndia() {

XmlBeanFactory factory = new XmlBeanFactory(new ClassPathResource("spring.xml"));

Country country = (Country) factory.getBean("country");

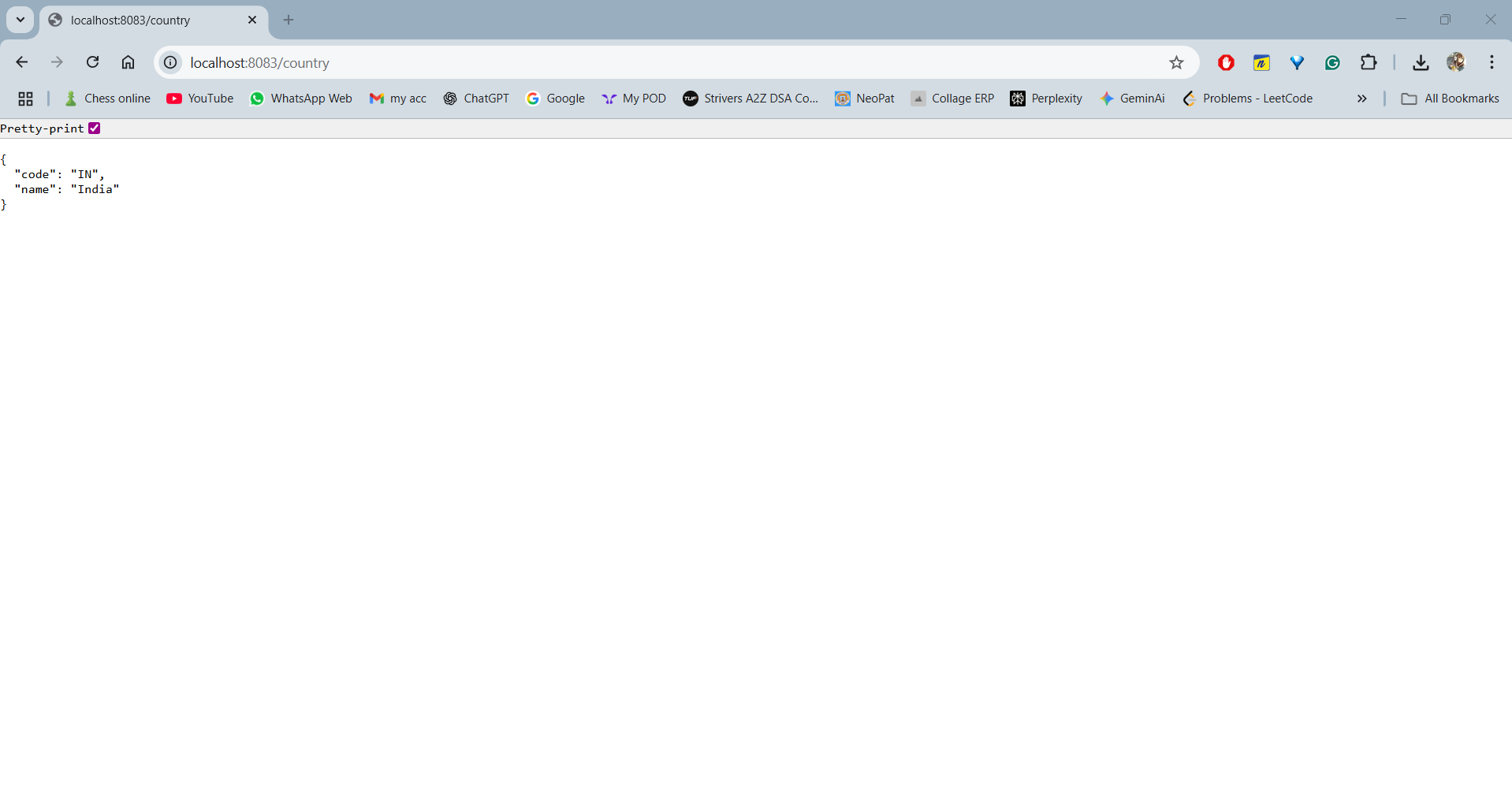
return country;

}

}

URL: http://localhost:8083/country

Output:



## Exercise 5: REST - Country Web Service

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;

}

}

### CountryController.java

package com.cognizant.springlearn.controller;

import java.util.List;

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

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

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

import com.cognizant.springlearn.model.Country;

@RestController

public class CountryController {

@GetMapping("/country")

public Country getCountryIndia() {

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

return (Country) context.getBean("country");

}

@GetMapping("/countries")

public List<Country> getAllCountries() {

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

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

}

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

public Country getCountryByCode(@PathVariable String code) {

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

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

for (Country country : countries) {

if (country.getCode().equalsIgnoreCase(code)) {

return country;

}

}

return null;

}

}

### spring.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>

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

<constructor-arg>

<list>

<bean class="com.cognizant.springlearn.model.Country">

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

<property name="name" value="India"/>

</bean>

<bean class="com.cognizant.springlearn.model.Country">

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

<property name="name" value="United States"/>

</bean>

<bean class="com.cognizant.springlearn.model.Country">

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

<property name="name" value="Germany"/>

</bean>

<bean class="com.cognizant.springlearn.model.Country">

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

<property name="name" value="Japan"/>

</bean>

</list>

</constructor-arg>

</bean>

</beans>

URL: http://localhost:8083/countries

Output:

## 