**Spring REST using Spring Boot 3**

**2. spring-rest-handson**

**REST - Country Web Service**

**REST service that returns India country details in the earlier created spring learn application.**

* **CountryController.java**

package com.cognizant.spring\_learn.controller;

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

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

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.cognizant.spring\_learn.Country;

@RestController

public class CountryController {

@RequestMapping("/country")

public Country getCountryIndia() {

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

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

context.close(); // Close the context to avoid resource leak

return country;

}

}

* **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.spring\_learn.Country">

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

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

</bean>

</beans>

* **Country.java**

package com.cognizant.spring\_learn;

public class Country {

private String code;

private String name;

// Getters and setters

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 + "]";

}

}

* **SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication // Enables component scanning and auto-configuration

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); // Launch Spring Boot app

*LOGGER*.info("END");

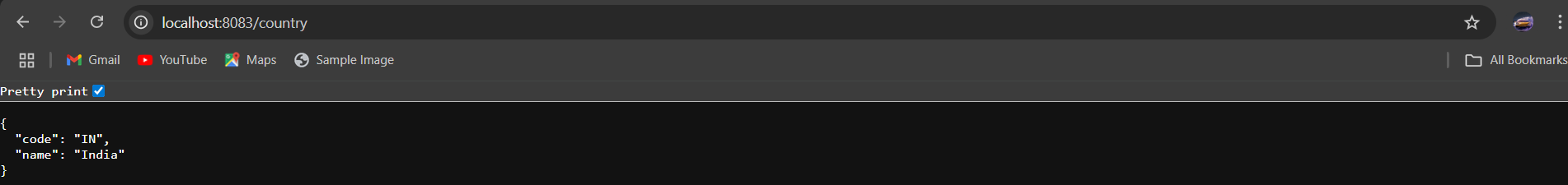
}

}

Then go to this URL:

* <http://localhost:8083/country>

**OUTPUT:**



**What happens in the controller method?**

* When /country is hit, Spring calls getCountryIndia().
* The method loads the Spring context from country.xml.
* Fetches the bean country of type Country.
* Returns it as a response.

**How is the bean converted into JSON?**

* The return type is Country, a POJO (JavaBean).
* Spring Boot uses **Jackson** (a JSON library) under the hood.
* It auto-converts the Java object into a JSON response because:
  + You're using @RestController, which combines @Controller + @ResponseBody.

**View HTTP Header in Developer Tools (Browser):**

1. Open browser → F12 → Go to **Network** tab.
2. Refresh or hit http://localhost:8083/country.
3. Click on the /country request.
4. In the **Headers** tab, you’ll see:
   * Content-Type: application/json
   * Status: 200 OK
   * Date, Server, etc.