Hands-on 4: REST – Country Web Service

## 1. Objective

Create a RESTful GET service that returns Country details (India) in JSON format.

## 2. Create Country Bean

File: src/main/java/com/cognizant/springlearn/model/Country.java  
  
Code:  
  
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; }  
}

## 3. Define Bean in XML

File: src/main/resources/country.xml  
  
Content:  
  
<?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="in" class="com.cognizant.springlearn.model.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
</beans>

## 4. Create Controller

File: src/main/java/com/cognizant/springlearn/controller/CountryController.java  
  
Code:  
  
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.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class CountryController {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);  
  
 @RequestMapping("/country")  
 public Country getCountryIndia() {  
 LOGGER.info("START: getCountryIndia()");  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = context.getBean("in", Country.class);  
 LOGGER.info("END: getCountryIndia()");  
 return country;  
 }  
}

## 5. Run and Test

• Start SpringLearnApplication  
• Open browser or Postman → http://localhost:8083/country  
  
Expected JSON Response:  
{  
 "code": "IN",  
 "name": "India"  
}

## 6. SME Explanation Points

• Controller loads 'in' bean using Spring XML and returns it.  
• Spring Boot uses Jackson to convert Country object to JSON.  
• View HTTP Headers:  
 - In Chrome: DevTools → Network → Headers tab  
 - In Postman: Headers tab after request

Output:

