Hands-on 5: REST – Get Country Based on Country Code

## 1. Objective

Create a RESTful GET service that returns a specific Country object based on country code (case-insensitive).

## 2. Update country.xml with Multiple Countries

File: src/main/resources/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">  
  
 <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="CN"/>  
 <property name="name" value="China"/>  
 </bean>  
 </list>  
 </constructor-arg>  
 </bean>  
</beans>

## 3. Create CountryService

File: src/main/java/com/cognizant/springlearn/service/CountryService.java  
  
package com.cognizant.springlearn.service;  
  
import com.cognizant.springlearn.model.Country;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.stereotype.Service;  
  
import java.util.List;  
  
@Service  
public class CountryService {  
  
 public Country getCountry(String code) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 List<Country> countryList = context.getBean("countryList", List.class);  
 return countryList.stream()  
 .filter(country -> country.getCode().equalsIgnoreCase(code))  
 .findFirst()  
 .orElse(null);  
 }  
}

## 4. Update CountryController

File: src/main/java/com/cognizant/springlearn/controller/CountryController.java  
  
// Add inside the class:  
@Autowired  
private CountryService countryService;  
  
@GetMapping("/country/{code}")  
public Country getCountry(@PathVariable String code) {  
 LOGGER.info("START: getCountry()");  
 Country country = countryService.getCountry(code);  
 LOGGER.info("END: getCountry()");  
 return country;  
}

## 5. Run and Test

Run SpringLearnApplication  
Test in browser or Postman:  
GET http://localhost:8083/country/in  
  
Expected Output:  
{  
 "code": "IN",  
 "name": "India"  
}

## 6. SME Explanation Points

• Controller accepts country code from URL as @PathVariable  
• Service loads country list from XML config  
• Matching is done case-insensitively using Java streams  
• Spring Boot (Jackson) converts the Country object to JSON  
• View HTTP headers using browser DevTools or Postman

Output:

