**Hands-on: REST - Get Country Based on Country Code**

**1. Objective**

The objective of this hands-on is to build a RESTful service using Spring Boot that returns a specific country's details based on the country code passed in the URL. The country code must be case-insensitive. Country details are stored in a Spring XML configuration file and accessed using a service class.

**2. Implementation Steps**

* Update country.xml to include a list of countries using <list> and <bean>.
* Create a CountryService class to load the country list and fetch country by code.
* Create a CountryController class with endpoint /countries/{code} using @GetMapping.
* Use Stream API with case-insensitive filtering to return the matching country object.

**3. XML Configuration (country.xml)**

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

<constructor-arg>

<list>

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

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

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

</bean>

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

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

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

</bean>

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

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

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

</bean>

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

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

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

</bean>

</list>

</constructor-arg>

</bean>

**4. CountryService.java**

package com.cognizant.springlearn.service;

import com.cognizant.springlearn.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 = (List<Country>) context.getBean("countryList");

return countryList.stream()

.filter(c -> c.getCode().equalsIgnoreCase(code))

.findFirst()

.orElse(null);

}

}

**5. CountryController.java**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.Country;

import com.cognizant.springlearn.service.CountryService;

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

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

@RestController

public class CountryController {

@Autowired

private CountryService countryService;

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

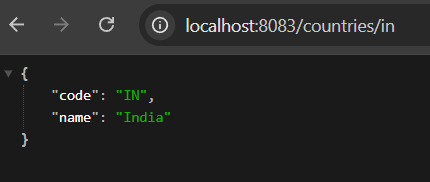
public Country getCountry(@PathVariable String code) {

return countryService.getCountry(code);

}

}

**6. Output**

****

**7. Conclusion**

This hands-on demonstrates how to create a REST endpoint that dynamically serves country data based on a path parameter. It uses Spring XML configuration for bean definition, a service layer for business logic, and automatic JSON conversion through Spring Boot.