**HandsOn : REST - Get country based on country code**

Here, the objective is to create a Spring Boot REST service that returns country details based on a case-insensitive country code passed as a path variable.

**Steps :**

1. **Updating ‘*country.xml’* to look like this :**

* <?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">
* <!-- Single country bean (optional, used in previous hands-on) -->
* <bean id="country" class="com.cognizant.spring\_learn.Country">
* <property name="code" value="IN" />
* <property name="name" value="India" />
* </bean>
* <!-- List of countries -->
* <bean id="countryList" class="java.util.ArrayList">
* <constructor-arg>
* <list>
* <bean class="com.cognizant.spring\_learn.Country">
* <property name="code" value="IN" />
* <property name="name" value="India" />
* </bean>
* <bean class="com.cognizant.spring\_learn.Country">
* <property name="code" value="US" />
* <property name="name" value="United States" />
* </bean>
* <bean class="com.cognizant.spring\_learn.Country">
* <property name="code" value="JP" />
* <property name="name" value="Japan" />
* </bean>
* <bean class="com.cognizant.spring\_learn.Country">
* <property name="code" value="DE" />
* <property name="name" value="Germany" />
* </bean>
* </list>
* </constructor-arg>
* </bean>
* </beans>

1. **A class *‘CountryService’*  is created.**

package com.cognizant.spring\_learn.service;

import com.cognizant.spring\_learn.Country;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Service;

import java.util.List;

*@Service*

public class CountryService {

private static final Logger *LOGGER* = LoggerFactory.*getLogger*(CountryService.class);

public Country getCountry(String code) {

*LOGGER*.info("START");

try (ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("country.xml")) {

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

return countryList.stream()

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

.findFirst()

.orElse(null); // or throw an exception if preferred

} finally {

*LOGGER*.info("END");

}

}

}

1. **A class *‘Country*** ***Controller’*  is created.**

* package com.cognizant.spring\_learn.controller;
* import com.cognizant.spring\_learn.Country;
* import com.cognizant.spring\_learn.service.CountryService;
* import org.slf4j.Logger;
* import org.slf4j.LoggerFactory;
* import org.springframework.beans.factory.annotation.Autowired;
* import org.springframework.web.bind.annotation.\*;
* *@RestController*
* public class CountryController {
* private static final Logger ***LOGGER*** = LoggerFactory.*getLogger*(CountryController.class);
* *@Autowired*
* private CountryService countryService;
* *@GetMapping*("/countries/{code}")
* public Country getCountry(*@PathVariable* String code) {
* ***LOGGER***.info("START");
* Country country = countryService.getCountry(code);
* ***LOGGER***.debug("Country Found: {}", country);
* ***LOGGER***.info("END");
* return country;
* }
* }

1. **Ran the Application –**

After running SpringLearnApplication as java application, we get console output as –



1. **Testing the Endpoint –**

We got following JSON responses by sending requests in Postman, like-

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.