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

* **Step 1:**

Open spring initiallizr

A screenshot of a computer

AI-generated content may be incorrect.

* Do the following things as shown in the picture and then generate it.
* And then extract the generated folder and open it in IntelliJ.

OR

Open intelliJ and in the existing project SPRING\_LEARN add the following :

* **src/main/java/com/cognizant/spring\_learn/service/CountryService.java**

package com.cognizant.spring\_learn.service;

import com.cognizant.spring\_learn.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> countries = (List<Country>) context.getBean("countryList");

// Find country by code (case insensitive)

return countries.stream()

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

.findFirst()

.orElse(null); // or throw a custom exception

}

}

* **Update the CountryController:**

package com.cognizant.spring\_learn.controller;  
  
import com.cognizant.spring\_learn.model.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.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
public class CountryController {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(CountryController.class);  
  
 @Autowired  
 private CountryService countryService;  
  
 // Get India using country bean  
 @RequestMapping("/country")  
 public Country getCountryIndia() {  
 *LOGGER*.info("START getCountryIndia()");  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = (Country) context.getBean("in");  
 *LOGGER*.info("END getCountryIndia()");  
 return country;  
 }  
  
 // Get all countries  
 @GetMapping("/countries")  
 public List<Country> getAllCountries() {  
 *LOGGER*.info("START getAllCountries()");  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 List<Country> countries = (List<Country>) context.getBean("countryList");  
 *LOGGER*.info("END getAllCountries()");  
 return countries;  
 }  
  
 // Get country by code  
 @GetMapping("/countries/{code}")  
 public Country getCountry(@PathVariable String code) {  
 *LOGGER*.info("START getCountry() with code: {}", code);  
 Country country = countryService.getCountry(code);  
 *LOGGER*.info("END getCountry()");  
 return country;  
 }  
}

* **Update the 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="in" class="com.cognizant.spring\_learn.model.Country">

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

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

</bean>

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

<constructor-arg>

<list>

<bean class="com.cognizant.spring\_learn.model.Country">

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

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

</bean>

<bean class="com.cognizant.spring\_learn.model.Country">

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

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

</bean>

<bean class="com.cognizant.spring\_learn.model.Country">

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

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

</bean>

<bean class="com.cognizant.spring\_learn.model.Country">

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

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

</bean>

</list>

</constructor-arg>

</bean>

</beans>

**Screenshot of output:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**OUTPUT OF BROWSER:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**OUTPUT OF POSTMAN :**

**A screenshot of a computer

AI-generated content may be incorrect.**