REST - Country Web Service Implementation

1. Spring XML Configuration (applicationContext.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.springlearn.model.Country">

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

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

</bean>

</beans>

2. Country Model Class

package com.cognizant.springlearn.model;

public class Country {

private String code;

private String name;

// Getters and Setters

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;

}

@Override

public String toString() {

return "Country [code=" + code + ", name=" + name + "]";

}

}

3. Country Controller Implementation

package com.cognizant.springlearn.controller;

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

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.cognizant.springlearn.model.Country;

@RestController

@RequestMapping

public class CountryController {

@Autowired

private ApplicationContext context;

@GetMapping("/country")

public Country getCountryIndia() {

// Load the India bean from XML configuration

Country india = (Country) context.getBean("in", Country.class);

return india;

}

}

4. Main Application Class (to load XML config)

package com.cognizant.springlearn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.ImportResource;

@SpringBootApplication

@ImportResource("classpath:applicationContext.xml")

public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

}

}

5. What Happens in the Controller Method?

Request Handling:

When a GET request comes to /country, Spring routes it to getCountryIndia()

The method retrieves the "in" bean from Spring's application context

The bean was previously configured in applicationContext.xml

JSON Conversion:

Spring automatically converts the Country object to JSON using Jackson

This happens because:

The method returns a Country object

@RestController includes @ResponseBody semantics

Spring Boot auto-configures Jackson for JSON processing

Response Generation:

The returned Country object is serialized to JSON format

Appropriate HTTP headers are set (Content-Type: application/json)

6. Expected Response

Request: GET http://localhost:8083/country

Response Body:

{

"code": "IN",

"name": "India"

}

7. HTTP Headers Analysis

In Chrome Developer Tools (Network Tab):

HTTP/1.1 200 OK

Content-Type: application/json

Transfer-Encoding: chunked

Date: [current date and time]

In Postman (Headers Tab):

text

Content-Type → application/json

Date → [current date and time]

Transfer-Encoding → chunked

Key Headers Explanation:

Content-Type: application/json - Indicates the response is in JSON format

Status Code: 200 OK - The request was successful

Date - Timestamp of the response

8. Testing Verification

Verify XML Loading:

Check that the bean is properly loaded from XML

Verify no exceptions occur during context loading

JSON Structure:

Confirm the response matches exactly:

{

"code": "IN",

"name": "India"

}

Content-Type:

Ensure the response header shows application/json

Logging (optional enhancement):

You could add logging to track bean retrieval

9. Troubleshooting

If you encounter issues:

404 Error: Check that:

The controller is in the right package (com.cognizant.spring-learn.controller)

The application is running on port 8083

500 Error: Check that:

The bean name "in" matches in both XML and controller

The Country class has proper getter methods

No JSON Response: Ensure:

You have Jackson in your dependencies

The method returns the Country object directly