**REST – Country Web Service**

Write a REST service that returns India country details in the earlier created spring learn application.

**CountryController.java**

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.model.Country;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

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

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

@RestController

public class CountryController {

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

@RequestMapping("/country")

public Country getCountryIndia() {

LOGGER.info("START - getCountryIndia()");

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

Country country = context.getBean("country", Country.class);

LOGGER.info("END - getCountryIndia()");

return country;

}

}

**Country.java**

package com.cognizant.spring\_learn.model;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class Country {

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

private String code;

private String name;

public Country() {

LOGGER.debug("Inside Country Constructor.");

}

public String getCode() {

LOGGER.debug("Getting Code: {}", code);

return code;

}

public void setCode(String code) {

LOGGER.debug("Setting Code: {}", code);

this.code = code;

}

public String getName() {

LOGGER.debug("Getting Name: {}", name);

return name;

}

public void setName(String name) {

LOGGER.debug("Setting Name: {}", name);

this.name = name;

}

@Override

public String toString() {

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

}

}

**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="country" class="com.cognizant.spring\_learn.model.Country">

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

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

</bean>

</beans>

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

}

}

**SME:**

**Q. What happens in the controller method ?**

- The “/country” endpoint maps to the “getCountryIndia()” method.

- It loads Spring’s ApplicationContext using “country.xml”.

- It fetches the bean with id “country”, which is configured with “IN” and “India”.

- The method returns the bean object.

**Q. How the bean is converted into JSON response?**

- Spring Boot includes Jackson, which auto-converts Java POJOs into JSON.

- The returned “Country” object has “getCode()” and “getName()” -> used to generate:

{

“code” : “IN”,

“name” : “India”

}

**Q. In network tab of developer tools show the HTTP header details received?**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Q. In postman click on “Headers” tab to view the HTTP header details received?**

**A screenshot of a computer

AI-generated content may be incorrect.**