**REST - COUNTRY WEB SERVICE**

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

**URL: /country Controller: com.cognizant.spring-learn.controller.**

**CountryController Method Annotation: @RequestMapping**

**Method Name: getCountryIndia()**

**Method Implementation: Load India bean from spring xml configuration and return**

**Sample Request:** [**http://localhost:8083/country**](http://localhost:8083/country)

**Sample Response:**

**{**

**"code": "IN",**

**"name": "India"**

**}**

**APPROACH**

**· Create POJO Country with code and name fields and getter/setter methods.**

**· Define XML Bean in in src/main/resources/country.xml with values "IN" and "India".**

**· Create REST Controller CountryController with @RestController and @RequestMapping("/country").**

**· Load XML Bean using ClassPathXmlApplicationContext and return the Country object in getCountryIndia().**

**· Run App and Test at http://localhost:8083/country to receive JSON response via browser or Postman.**

**CODE**

**Country.java**

**package com.cognizant.spring\_learn;**

**public class Country {**

**private String code;**

**private String name;**

**public Country() {}**

**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;**

**}**

**}**

**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);**

**}**

**}**

**CountryController.java**

**package com.cognizant.spring\_learn.controller;**

**import com.cognizant.spring\_learn.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 = (Country) context.getBean("in");**

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

**return country;**

**}**

**}**

**HelloController.java**

**package com.cognizant.spring\_learn.controller;**

**import org.slf4j.Logger;**

**import org.slf4j.LoggerFactory;**

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

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

**@RestController**

**public class HelloController {**

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

**@GetMapping("/hello")**

**public String sayHello() {**

***LOGGER*.info("START sayHello()");**

***LOGGER*.info("END sayHello()");**

**return "Hello World!!";**

**}**

**}**

**application.properties**

**logging.level.com.cognizant.spring\_learn=DEBUG**

**server.port=8083**

**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**

**https://www.springframework.org/schema/beans/spring-beans.xsd">**

**<bean id="in" class="com.cognizant.spring\_learn.Country">**

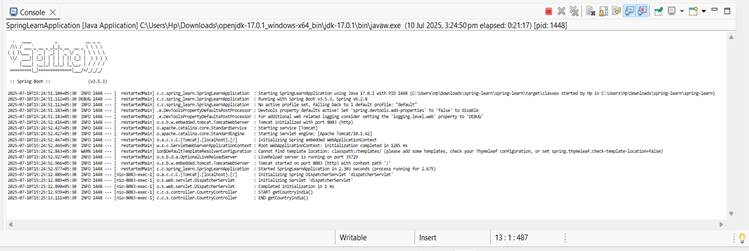
**<property name="code" value="IN"/>**

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

**</bean>**

**</beans>**

**OUTPUT**

****

****