**RESTful Web Service resource naming guidelines** 

**SpringLearn2Application.java**

package com.cognizant.spring\_learn2;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearn2Application {

    public static void main(String[] args) {

        SpringApplication.run(SpringLearn2Application.class, args);

    }

}

**Country.java**

package com.cognizant.spring\_learn2.model;

public class Country {

    private String code;

    private String name;

    public Country() {}

    public Country(String code, String name) {

        this.code = code;

        this.name = name;

    }

    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;

    }

}

**CountryController.java**

package com.cognizant.spring\_learn2.controller;

import com.cognizant.spring\_learn2.model.Country;

import com.cognizant.spring\_learn2.service.CountryService;

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

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

import java.util.List;

@RestController

@RequestMapping("/countries")

public class CountryController {

    @Autowired

    private CountryService countryService;

    @GetMapping

    public List<Country> getAllCountries() {

        return countryService.getAllCountries();

    }

    @GetMapping("/{code}")

    public Country getCountryByCode(@PathVariable String code) {

        return countryService.getCountry(code);

    }

}

**CountryService.java**

package com.cognizant.spring\_learn2.service;

import com.cognizant.spring\_learn2.model.Country;

import org.springframework.stereotype.Service;

import java.util.\*;

@Service

public class CountryService {

    private static final List<Country> countries = new ArrayList<>();

    static {

        countries.add(new Country("IN", "India"));

        countries.add(new Country("US", "United States"));

        countries.add(new Country("JP", "Japan"));

    }

    public Country getCountry(String code) {

        for (Country country : countries) {

            if (country.getCode().equalsIgnoreCase(code)) {

                return country;

            }

        }

        throw new NoSuchElementException("Country not found for code: " + code);

    }

    public List<Country> getAllCountries() {

        return countries;

    }

}

**pom.xml**

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <parent>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-parent</artifactId>

        <version>3.5.4</version>

        <relativePath/> <!-- lookup parent from repository -->

    </parent>

    <groupId>com.cognizant</groupId>

    <artifactId>spring-learn2</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <name>spring-learn2</name>

    <description>Demo project for Spring Boot</description>

    <url/>

    <licenses>

        <license/>

    </licenses>

    <developers>

        <developer/>

    </developers>

    <scm>

        <connection/>

        <developerConnection/>

        <tag/>

        <url/>

    </scm>

    <properties>

        <java.version>17</java.version>

    </properties>

    <dependencies>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-webflux</artifactId>

        </dependency>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-devtools</artifactId>

            <scope>runtime</scope>

            <optional>true</optional>

        </dependency>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-test</artifactId>

            <scope>test</scope>

        </dependency>

        <dependency>

            <groupId>io.projectreactor</groupId>

            <artifactId>reactor-test</artifactId>

            <scope>test</scope>

        </dependency>

    </dependencies>

    <build>

        <plugins>

            <plugin>

                <groupId>org.springframework.boot</groupId>

                <artifactId>spring-boot-maven-plugin</artifactId>

            </plugin>

        </plugins>

    </build>

</project>

**Output:**







