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

**Pom.xml:**

<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

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.cognizant</groupId>

<artifactId>spring-learn</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>spring-learn</name>

<description>Spring Boot REST project</description>

<parent>

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

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

<version>3.2.4</version>

<relativePath/>

</parent>

<properties>

<java.version>21</java.version>

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

<groupId>jakarta.xml.bind</groupId>

<artifactId>jakarta.xml.bind-api</artifactId>

<version>4.0.0</version>

</dependency>

<dependency>

<groupId>org.glassfish.jaxb</groupId>

<artifactId>jaxb-runtime</artifactId>

<version>4.0.2</version>

</dependency>

<!-- Optional: dev tools -->

<dependency>

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

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

<scope>runtime</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.11.0</version>

<configuration>

<source>21</source>

<target>21</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

**Country.java:**

package com.cognizant.spring\_learn;

import jakarta.xml.bind.annotation.XmlElement;

import jakarta.xml.bind.annotation.XmlRootElement;

@XmlRootElement(name = "country")

public class Country {

private String code;

private String name;

@XmlElement

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

@XmlElement

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

**CountryList.java:**

package com.cognizant.spring\_learn;

import jakarta.xml.bind.annotation.XmlElement;

import jakarta.xml.bind.annotation.XmlRootElement;

import java.util.List;

@XmlRootElement(name = "countries")

public class CountryList {

private List<Country> countryList;

@XmlElement(name = "country")

public List<Country> getCountryList() {

return countryList;

}

public void setCountryList(List<Country> countryList) {

this.countryList = countryList;

}

}

**CountryController.java:**

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.Country;

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

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

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

@RestController

public class CountryController {

@Autowired

private CountryService countryService;

@GetMapping("/countries/{code}")

public Country getCountry(@PathVariable String code) throws Exception {

return countryService.getCountry(code);

}

}

**CountryService.java:**

package com.cognizant.spring\_learn.service;

import com.cognizant.spring\_learn.Country;

import com.cognizant.spring\_learn.CountryList;

import jakarta.annotation.PostConstruct;

import jakarta.xml.bind.JAXBContext;

import jakarta.xml.bind.Unmarshaller;

import org.springframework.stereotype.Service;

import java.io.InputStream;

import java.util.List;

import java.util.Optional;

@Service

public class CountryService {

private List<Country> countryList;

@PostConstruct

public void init() {

try {

InputStream is = getClass().getResourceAsStream("/country.xml");

JAXBContext context = JAXBContext.newInstance(CountryList.class);

Unmarshaller unmarshaller = context.createUnmarshaller();

CountryList countries = (CountryList) unmarshaller.unmarshal(is);

countryList = countries.getCountryList();

} catch (Exception e) {

e.printStackTrace();

}

}

public Country getCountry(String code) throws Exception {

Optional<Country> country = countryList.stream()

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

.findFirst();

if (country.isPresent()) {

return country.get();

} else {

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

}

}

}

**SpringLearnApplication.java:**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication(scanBasePackages = "com.cognizant.spring\_learn")

public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

}

}

**application.properties:**

server.port=8083

**country.xml:**

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

<countries>

<country>

<code>IN</code>

<name>India</name>

</country>

<country>

<code>US</code>

<name>United States</name>

</country>

<country>

<code>CN</code>

<name>China</name>

</country>

</countries>

**Output:**





