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

**Question:**

Write a REST service that returns a specific country based on country code. The country code should be case insensitive.  
  
**Controller**: com.cognizant.spring-learn.controller.CountryController  
**Method Annotation:** @GetMapping("/countries/{code}")  
**Method Name**: getCountry(String code)  
**Method Implemetation**: Invoke countryService.getCountry(code)   
**Service Method:**com.cognizant.spring-learn.service.CountryService.getCountry(String code)  
  
**Service Method Implementation**:

* Get the country code using @PathVariable
* Get country list from country.xml
* Iterate through the country list
* Make a case insensitive matching of country code and return the country.
* Lambda expression can also be used instead of iterating the country list

**Sample Request**: http://localhost:8083/country/in  
  
**Sample Response**:

{

  "code": "IN",

  "name": "India"

}

**Solution:**

**CountryService.java**

**package** **com.example.service**;

**import** **java.util.List**;

**import** **org.springframework.context.ApplicationContext**;

**import** **org.springframework.context.support.ClassPathXmlApplicationContext**;

**import** **org.springframework.stereotype.Service**;

**import** **com.example.exception.CountryNotFoundException**;

**import** **com.example.model.Country**;

@**Service**

**public** **class** CountryService {  *// ✅ class wrapper added*

**public** **Country** getCountry(**String** code) **throws** **CountryNotFoundException** {

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

**List**<**Country**> countries **=** (**List<**Country**>**) context.getBean("countryList");

**return** countries.stream()

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

                .findFirst()

                .orElseThrow(() **->** **new** CountryNotFoundException("Country not found: " **+** code));

    }

**public** **List**<**Country**> getAllCountries() {

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

**return** (**List<**Country**>**) context.getBean("countryList");

    }

}

**CountryController.java**

**package** **com.example.controller**;

**import** **com.example.model.Country**;

**import** **com.example.service.CountryService**;

**import** **com.example.exception.CountryNotFoundException**;

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

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

**import** **java.util.List**;

@**RestController**

**public** **class** CountryController {

    @**Autowired**

**private** **CountryService** countryService;

    @**GetMapping**("/country")

**public** **Country** getCountryIndia() {

**return** **new** Country("IN", "India");  *// hardcoded for /country*

    }

    @**GetMapping**("/countries")

**public** **List**<**Country**> getAllCountries() {

**return** countryService.getAllCountries();  *// if you've added this*

    }

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

**public** **Country** getCountryByCode(@**PathVariable** **String** code) **throws** **CountryNotFoundException** {

**return** countryService.getCountry(code);

    }

}

**CountryNotFoundException.java**

**package** **com.example.exception**;

**import** **org.springframework.http.HttpStatus**;

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

@**ResponseStatus**(value **=** HttpStatus.NOT\_FOUND, reason **=** "Country not found")

**public** **class** CountryNotFoundException **extends** Exception {

**public** CountryNotFoundException(**String** message) {

        super(message);

    }

}

**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.example.model.Country">

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

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

    </bean>

    <bean *id*="countryList" *class*="java.util.ArrayList">

        <constructor-arg>

            <list>

                <ref *bean*="in"/>

                <bean *class*="com.example.model.Country">

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

                    <property *name*="name" *value*="United States"/>

                </bean>

                <bean *class*="com.example.model.Country">

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

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

                </bean>

                <bean *class*="com.example.model.Country">

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

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

                </bean>

            </list>

        </constructor-arg>

    </bean>

</beans>

**Country.java**

**package** **com.example.model**;

**public** **class** Country {

**private** **String** code;

**private** **String** name;

**public** Country() {}  *// default constructor*

**public** Country(**String** code, **String** name) {  *// ✅ fix*

        this.code **=** code;

        this.name **=** name;

    }

**public** **String** getCode() {

**return** code;

    }

**public** **void** setCode(**String** code) {

        this.code **=** code;

    }

**public** **String** getName() {

**return** name;

    }

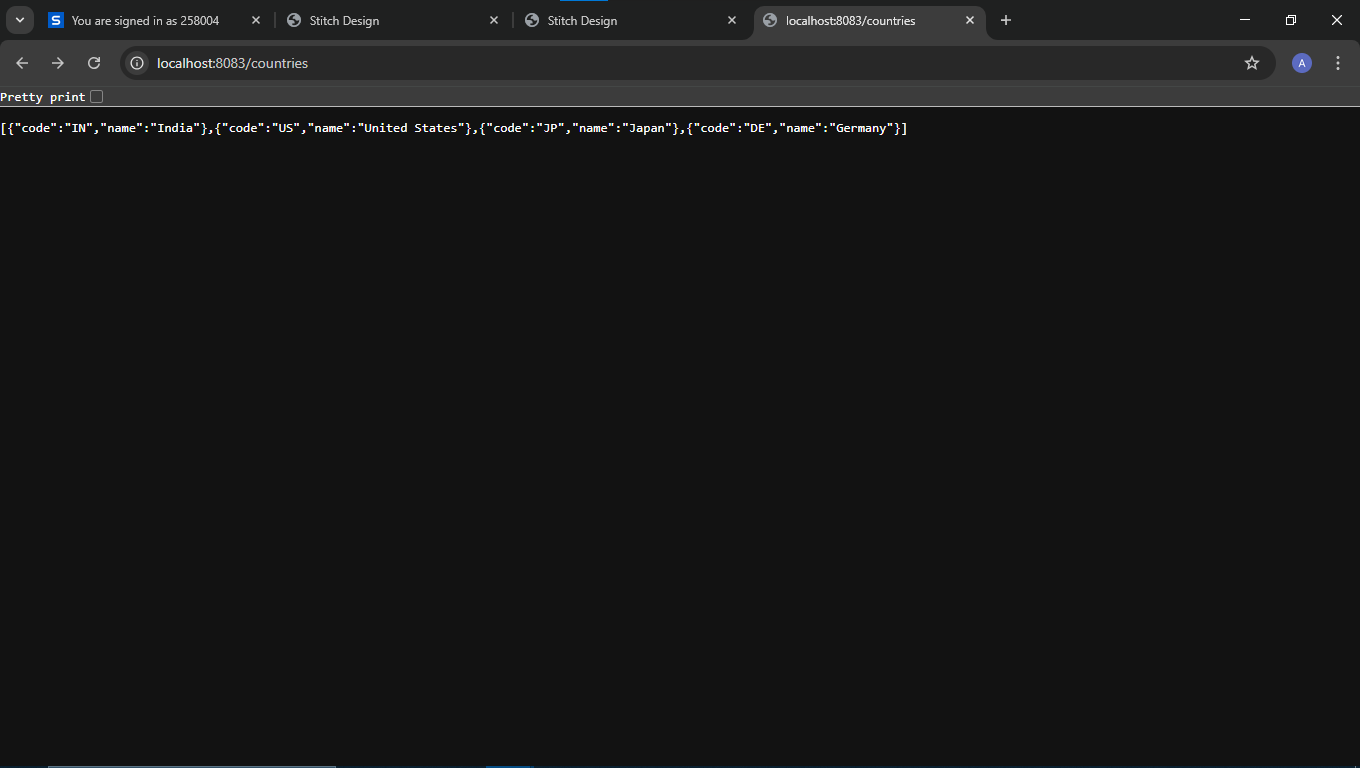
**public** **void** setName(**String** name) {

        this.name **=** name;

    }

}

**Output:**

****