**MockMVC - Test get country service for exceptional scenario   
  
Include MockMVC test that checks if correct response is received when there is an error.**

**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.model.Country;

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

import com.cognizant.spring\_learn.service.exception.CountryNotFoundException;

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 CountryNotFoundException {

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

    }

}

**CountryService.java**

package com.cognizant.spring\_learn.service;

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

import com.cognizant.spring\_learn.service.exception.CountryNotFoundException;

import org.springframework.stereotype.Service;

import java.util.ArrayList;

import java.util.List;

@Service

public class CountryService {

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

    static {

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

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

        countryList.add(new Country("CN", "China"));

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

    }

    public Country getCountry(String code) throws CountryNotFoundException {

        for (Country country : countryList) {

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

                return country;

            }

        }

        throw new CountryNotFoundException();

    }

}

**CountryNotFoundException.java**

package com.cognizant.spring\_learn.service.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() {

        super("Country not found");

    }

}

**SpringLearnApplicationTests.java**

package com.cognizant.spring\_learn;

import com.cognizant.spring\_learn.controller.CountryController;

import org.junit.jupiter.api.Test;

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

import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.test.web.servlet.MockMvc;

import org.springframework.test.web.servlet.ResultActions;

import static org.junit.jupiter.api.Assertions.assertNotNull;

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.get;

import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.\*;

@SpringBootTest

@AutoConfigureMockMvc

public class SpringLearnApplicationTests {

    @Autowired

    private CountryController countryController;

    @Autowired

    private MockMvc mvc;

    public void contextLoads() {

        assertNotNull(countryController);

    }

    @Test

    public void testGetCountry() throws Exception {

        ResultActions actions = mvc.perform(get("/country"));

        actions.andExpect(status().isOk());

        actions.andExpect(jsonPath("$.code").exists());

        actions.andExpect(jsonPath("$.code").value("IN"));

        actions.andExpect(jsonPath("$.name").exists());

        actions.andExpect(jsonPath("$.name").value("India"));

    }

    @Test

    public void testGetCountryException() throws Exception {

        ResultActions actions = mvc.perform(get("/countries/az"));

        actions.andExpect(status().isNotFound()); // 404

        actions.andExpect(status().reason("Country not found"));

    }

}

**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-learn</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <name>spring-learn</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-web</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>

    </dependencies>

    <build>

        <plugins>

            <plugin>

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

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

            </plugin>

        </plugins>

    </build>

</project>

**Output:**

