**Hands On 8: MockMVC - Test get country service for exceptional scenario** 

**Country.xml**

<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.model.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
  
 <bean id="us" class="com.cognizant.spring\_learn.model.Country">  
 <property name="code" value="US"/>  
 <property name="name" value="United States"/>  
 </bean>  
  
 <bean id="de" class="com.cognizant.spring\_learn.model.Country">  
 <property name="code" value="DE"/>  
 <property name="name" value="Germany"/>  
 </bean>  
  
 <bean id="jp" class="com.cognizant.spring\_learn.model.Country">  
 <property name="code" value="JP"/>  
 <property name="name" value="Japan"/>  
 </bean>  
  
 <bean id="countryList" class="java.util.ArrayList">  
 <constructor-arg>  
 <list>  
 <ref bean="in"/>  
 <ref bean="us"/>  
 <ref bean="de"/>  
 <ref bean="jp"/>  
 </list>  
 </constructor-arg>  
 </bean>  
  
</beans>

**Country.java**

package com.cognizant.spring\_learn;  
  
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;  
 }  
  
 @Override  
 public String toString() {  
 return "Country{" +  
 "code='" + code + '\'' +  
 ", name='" + name + '\'' +  
 '}';  
 }  
}

**CountryController.java**

package com.cognizant.spring\_learn.controller;  
  
import com.cognizant.spring\_learn.Country;  
import com.cognizant.spring\_learn.service.CountryService;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
import java.util.List;  
  
@RestController  
public class CountryController {  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(CountryController.class);  
 @Autowired  
 CountryService countryService;  
  
 @GetMapping("/countries")  
 public List<Country> getAllCountries() {  
 *LOGGER*.info("START");  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 List<Country> countryList = context.getBean("countryList", List.class);  
 *LOGGER*.debug("Country List: {}", countryList);  
 *LOGGER*.info("END");  
 return countryList;  
 }  
  
 @GetMapping("/countries/{code}")  
 public Country getCountry(@PathVariable String code) throws CountryNotFoundException

{  
 *LOGGER*.info("START");  
  
 Country country = countryService.getCountry(code);  
 *LOGGER*.debug("Country Found: {}", country);  
 *LOGGER*.info("END");  
 return country;  
 }  
}

**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(String message) {  
 super(message);  
 }  
}

**CountryService.java**

package com.cognizant.spring\_learn.service;  
  
import java.util.List;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.stereotype.Service;  
import com.cognizant.spring\_learn.Country;  
  
@Service  
public class CountryService {  
  
 public Country getCountry(String code) throws CountryNotFoundException

{  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 List<Country> countryList = context.getBean("countryList", List.class);  
  
 return countryList.stream()  
 .filter(country -> country.getCode().equalsIgnoreCase(code))  
 .findFirst()  
 .orElseThrow(() -> new CountryNotFoundException("Country not found"));

}  
}

**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);  
 }  
}

**SpringLearnApplicationTests.java**

package com.cognizant.spring\_learn;  
  
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.*jsonPath*;  
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.*status*;  
  
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.context.annotation.ImportResource;  
import org.springframework.test.web.servlet.MockMvc;  
import org.springframework.test.web.servlet.ResultActions;  
  
@SpringBootTest  
@AutoConfigureMockMvc  
public class SpringLearnApplicationTests {  
  
 @Autowired  
 private CountryController countryController;  
  
 @Autowired  
 private MockMvc mockMvc;  
 @Test  
 public void testGetCountryException() throws Exception {  
 ResultActions actions = mockMvc.perform(*get*("/countries/ZZ")); actions.andExpect(*status*().isNotFound());actions.andExpect(*status*().reason("Country not found"));

}

}

**Output**

