

## SPRING REST HANDS ON 2

**Superset ID :** 6384831

**Name :** Mohana Priya N

**E-mail :** [mohanapriya.2205056@srec.ac.in](mailto:mohanapriya.2205056@srec.ac.in)

### **Mandatory Hands on:**

#### **1) Hands on 1: Hello World RESTful Web Service**

##### **Solution:**

**//HelloController.java**

```
package com.cognizant.springlearn.controller;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController

public class HelloController {

    private static final Logger LOGGER = LoggerFactory.getLogger(HelloController.class);

    @GetMapping("/hello")

    public String sayHello() {

        LOGGER.info("START sayHello");

        LOGGER.info("END sayHello");

        return "Hello World!!";

    }

}
```

#### **2) Hands on 2: REST - Country Web Service**

##### **Solution:**

**//CountryController.java**

```
package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.Country;
```

```

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class CountryController {

    private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);

    @RequestMapping("/country")
    public Country getCountryIndia() {

        LOGGER.info("START getCountryIndia");

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

        Country country = context.getBean("country", Country.class);

        LOGGER.info("END getCountryIndia");

        return country;

    }

}

```

### 3) Hands on 4: REST - Get country based on country code

#### Solution:

##### // CountryService.java

```

package com.cognizant.springlearn.service;

import com.cognizant.springlearn.Country;
import com.cognizant.springlearn.service.exception.CountryNotFoundException;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.stereotype.Service;
import java.util.List;

@Service

```

```

public class CountryService {

    public Country getCountry(String code) {

        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());

    }

}

```

#### **//CountryController.java**

```

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.Country;
import java.util.List;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.PathVariable;

@RestController

public class CountryController {

    private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);

    @RequestMapping("/country")

    public Country getCountryIndia() {

        LOGGER.info("START getCountryIndia");

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

}

```

```

        Country country = context.getBean("country", Country.class);

        LOGGER.info("END getCountryIndia");

        return country;
    }

    @Autowired
    private CountryService countryService;

    @GetMapping("/countries/{code}")
    public Country getCountry(@PathVariable String code) throws CountryNotFoundException {
        LOGGER.info("START getCountry with code: {}", code);

        Country country = countryService.getCountry(code);

        LOGGER.info("END getCountry");

        return country;
    }
}

```

### **Other Hands on:**

#### **4) Hands on 3: REST - Get all countries**

##### **Solution:**

##### **//CountryController.java**

```

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.Country;
import java.util.List;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.bind.annotation.GetMapping;

@RestController

```

```

public class CountryController {

    private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);

    @RequestMapping("/country")
    public Country getCountryIndia() {

        LOGGER.info("START getCountryIndia");

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

        Country country = context.getBean("country", Country.class);

        LOGGER.info("END getCountryIndia");

        return country;

    }

    @GetMapping("/countries")
    public List<Country> getAllCountries() {

        LOGGER.info("START getAllCountries");

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

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

        LOGGER.info("END getAllCountries");

        return countries;

    }

}

```

## 5) Hands on 5: REST - Get country exceptional scenario

### Solution:

#### //CountryNotFoundException.java

```

package com.cognizant.springlearn.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 RuntimeException {

}

```

## 6) Hands on 6: MockMVC - Test get country service

### Solution:

**//SpringLearnApplicationTests.java**

```
package com.cognizant.springlearn;

import com.cognizant.springlearn.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;

    @Test
    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").value("India"));
    }
    @Test
    public void testGetCountryException() throws Exception {
        mvc.perform(get("/country/az"))
            .andExpect(status().isNotFound())
            .andExpect(status().reason("Country not found"));
    }
}

```

## 7) Hands on 7: MockMVC - Test get country service for exceptional scenario

### Solution:

**//SpringLearnApplication.java**

```

@SpringBootTest
@AutoConfigureMockMvc
public class SpringLearnApplicationTests {
    @Autowired private CountryController countryController;
    @Autowired private MockMvc mvc;
    @Test public void contextLoads() {
        assertNotNull(countryController);
    }
    @Test public void testGetCountry() throws Exception {
        mvc.perform(get("/country"))
            .andExpect(status().isOk())
            .andExpect(jsonPath("$.code").value("IN"))
            .andExpect(jsonPath("$.name").value("India"));
    }
    @Test public void testGetCountryException() throws Exception {
        mvc.perform(get("/countries/az"))

```

```
.andExpect(status().isNotFound())  
.andExpect(status().reason("Country not found"));  
}  
}
```