Spring REST using Spring Boot 3(2. spring-rest-handson)

REST - Country Web Service

Write a REST service in the spring learn application created earlier, that returns the text "Hello World!!" using Spring Web Framework. Refer details below:

Method: GET
URL: /hello
Controller: com.cognizant.spring-learn.controller.HelloController
Method Signature: public String sayHello()
Method Implementation: return hard coded string "Hello World!!"
Sample Request: http://localhost:8083/hello
Sample Response: Hello World!!

IMPORTANT NOTE: Don't forget to include start and end log in the sayHello() method.

Try the URL http://localhost:8083/hello in both chrome browser and postman.

SME to explain the following aspects:

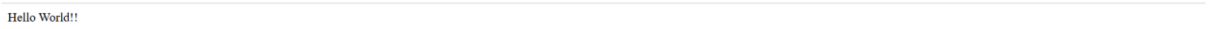
- In network tab of developer tools show the HTTP header details received
- In postman click on "Headers" tab to view the HTTP header details received

Code:

HelloController.java

```java
package com.cognizant.spring_learn.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()");
        String message = "Hello World!!";
        LOGGER.info("END sayHello()");
        return message;
    }
}
```

Output:



Hello World!!

REST - Country Web Service

Write a REST service that returns India country details in the earlier created spring learn application.

URL: /country
Controller: com.cognizant.spring-learn.controller.CountryController
Method Annotation: @RequestMapping
Method Name: getCountryIndia()
Method Implementation: Load India bean from spring xml configuration and return
Sample Request: http://localhost:8083/country
Sample Response:

```
{

  "code": "IN",

  "name": "India"

}
```

SME to explain the following aspects:

- What happens in the controller method?
- How the bean is converted into JSON reponse?
- In network tab of developer tools show the HTTP header details received
- In postman click on "Headers" tab to view the HTTP header details received

Code:

CountryController.java

```java
package com.cognizant.spring_learn.controller;

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 com.cognizant.spring_learn.Country;

@RestController
public class CountryController {
    private static final Logger LOGGER =
LoggerFactory.getLogger(CountryController.class);

    @RequestMapping("/country")
    public Country getCountryIndia() {
        LOGGER.info("START");
        ApplicationContext context = new
ClassPathXmlApplicationContext("country.xml");
        Country country = (Country) context.getBean("in");
        LOGGER.info("END");
        return country;
    }
}
```

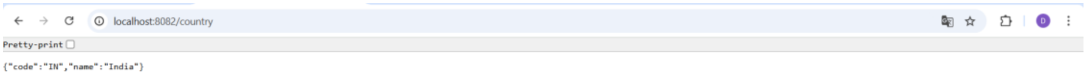
Country.xml

```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
        http://www.springframework.org/schema/beans/spring-beans.xsd">

    <!-- This is the bean named 'country' -->
    <bean id="country" class="com.cognizant.spring_learn.Country">
        <property name="code" value="IN"/>
        <property name="name" value="India"/>
    </bean>

    <!-- Optional: If your controller is expecting a bean named 'in', include this
too -->
    <bean id="in" class="com.cognizant.spring_learn.Country">
        <property name="code" value="IN"/>
        <property name="name" value="India"/>
    </bean>
</beans>
```

Country.java

```java
package com.cognizant.spring_learn;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class Country {
	private static final Logger LOGGER =
LoggerFactory.getLogger(Country.class);
          private String code;
          private String name;
          public Country() {
              LOGGER.debug("Inside Country Constructor.");
          }
          public String getCode() {
              LOGGER.debug("Getter for code called");
              return code;
          }
          public void setCode(String code) {
              LOGGER.debug("Setter for code called");
              this.code = code;
          }
          public String getName() {
              LOGGER.debug("Getter for name called");
              return name;
          }
          public void setName(String name) {
              LOGGER.debug("Setter for name called");
              this.name = name;
          }

          @Override
          public String toString() {
              return "Country [code=" + code + ", name=" + name + "]";
          }
}
```

Output:



```
{"code":"IN","name":"India"}
```

REST - Get country based on country code

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"
}
```

Code:

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;
    }
```

```java
    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;
    }
}
```

Country.xml

```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
       http://www.springframework.org/schema/beans/spring-beans.xsd">

    <bean id="country" class="com.cognizant.spring_learn.Country">
        <property name="code" value="IN"/>
        <property name="name" value="India"/>
    </bean>

    <bean id="countryList" class="java.util.ArrayList">
        <constructor-arg>
            <list>
                <bean class="com.cognizant.spring_learn.Country">
                    <property name="code" value="IN"/>
                    <property name="name" value="India"/>
                </bean>
                <bean class="com.cognizant.spring_learn.Country">
                    <property name="code" value="US"/>
                    <property name="name" value="United States"/>
                </bean>
                <bean class="com.cognizant.spring_learn.Country">
                    <property name="code" value="DE"/>
                    <property name="name" value="Germany"/>
                </bean>
            </list>
        </constructor-arg>
    </bean>
</beans>
```

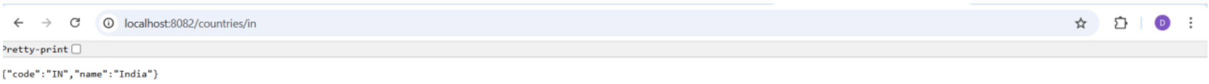
CountryService.java

```java
package com.cognizant.spring_learn.service;
import com.cognizant.spring_learn.Country;
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> countryList = (List<Country>)
context.getBean("countryList");
        return countryList.stream()
                .filter(c -> c.getCode().equalsIgnoreCase(code))
                .findFirst()
                .orElse(null); // or throw new CountryNotFoundException();
    }
}
```

CountryController.java

```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) {
        return countryService.getCountry(code);
    }
}
```

Output:



Pretty-print □

```
{"code":"IN","name":"India"}
```