Spring Rest Hands on 2 Solutions

**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 expressions can also be used instead of iterating the country.

1. **country.xml** (inside src/main/resources):-

<countries>

<country>

<code>IN</code>

<name>India</name>

</country>

<country>

<code>US</code>

<name>United States</name>

</country>

<country>

<code>JP</code>

<name>Japan</name>

</country>

</countries>

2. **Country.java (model):-**

package com.cognizant.springlearn.model;

public class Country {

private String code;

private String name;

// Getters and Setters

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code.toUpperCase(); // to store code in uppercase

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

**3. Countries.java (Wrapper for List<Country>):-**

package com.cognizant.springlearn.model;

import java.util.List;

import javax.xml.bind.annotation.\*;

@XmlRootElement(name = "countries")

@XmlAccessorType(XmlAccessType.FIELD)

public class Countries {

@XmlElement(name = "country")

private List<Country> countryList;

public List<Country> getCountryList() {

return countryList;

}

public void setCountryList(List<Country> countryList) {

this.countryList = countryList;

}

}

4. **CountryService.java:-**

package com.cognizant.springlearn.service;

import com.cognizant.springlearn.model.Country;

import com.cognizant.springlearn.model.Countries;

import org.springframework.core.io.ClassPathResource;

import org.springframework.stereotype.Service;

import javax.xml.bind.JAXBContext;

import javax.xml.bind.Unmarshaller;

import java.io.InputStream;

import java.util.List;

@Service

public class CountryService {

public Country getCountry(String code) throws Exception {

code = code.toUpperCase();

// Load XML and unmarshall

InputStream is = new ClassPathResource("country.xml").getInputStream();

JAXBContext context = JAXBContext.newInstance(Countries.class);

Unmarshaller unmarshaller = context.createUnmarshaller();

Countries countries = (Countries) unmarshaller.unmarshal(is);

List<Country> countryList = countries.getCountryList();

// Search using lambda

return countryList.stream()

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

.findFirst()

.orElseThrow(() -> new Exception("Country not found with code: " + code));

}

}

### **5. CountryController.java:-**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.model.Country;

import com.cognizant.springlearn.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}")

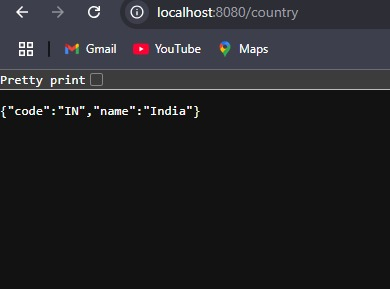
public Country getCountry(@PathVariable String code) throws Exception {

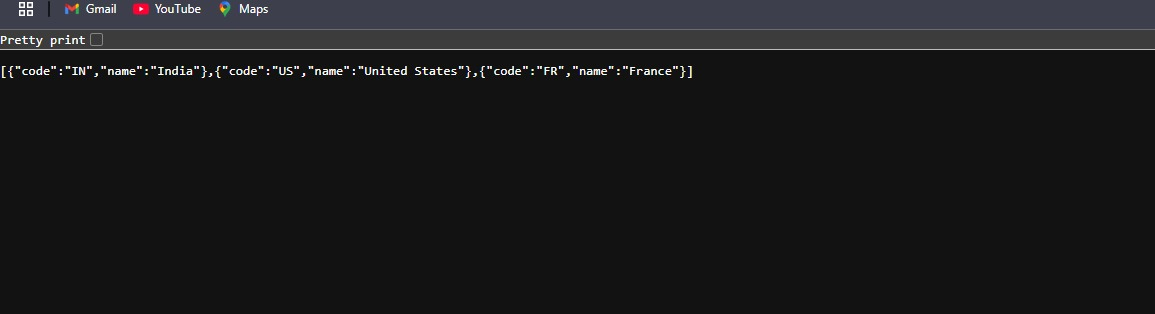
return countryService.getCountry(code);

}

}

OUTPUTS:-





**Hello World RESTful 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.springlearn.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 : HelloContoller.java file

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")

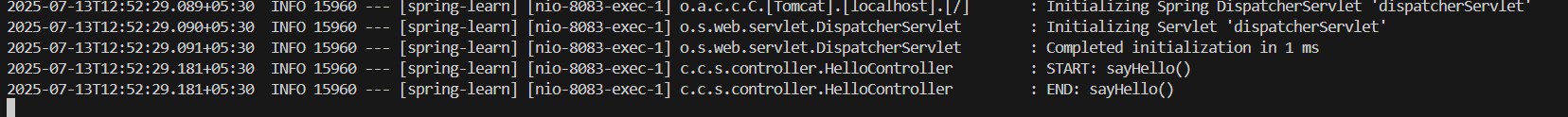
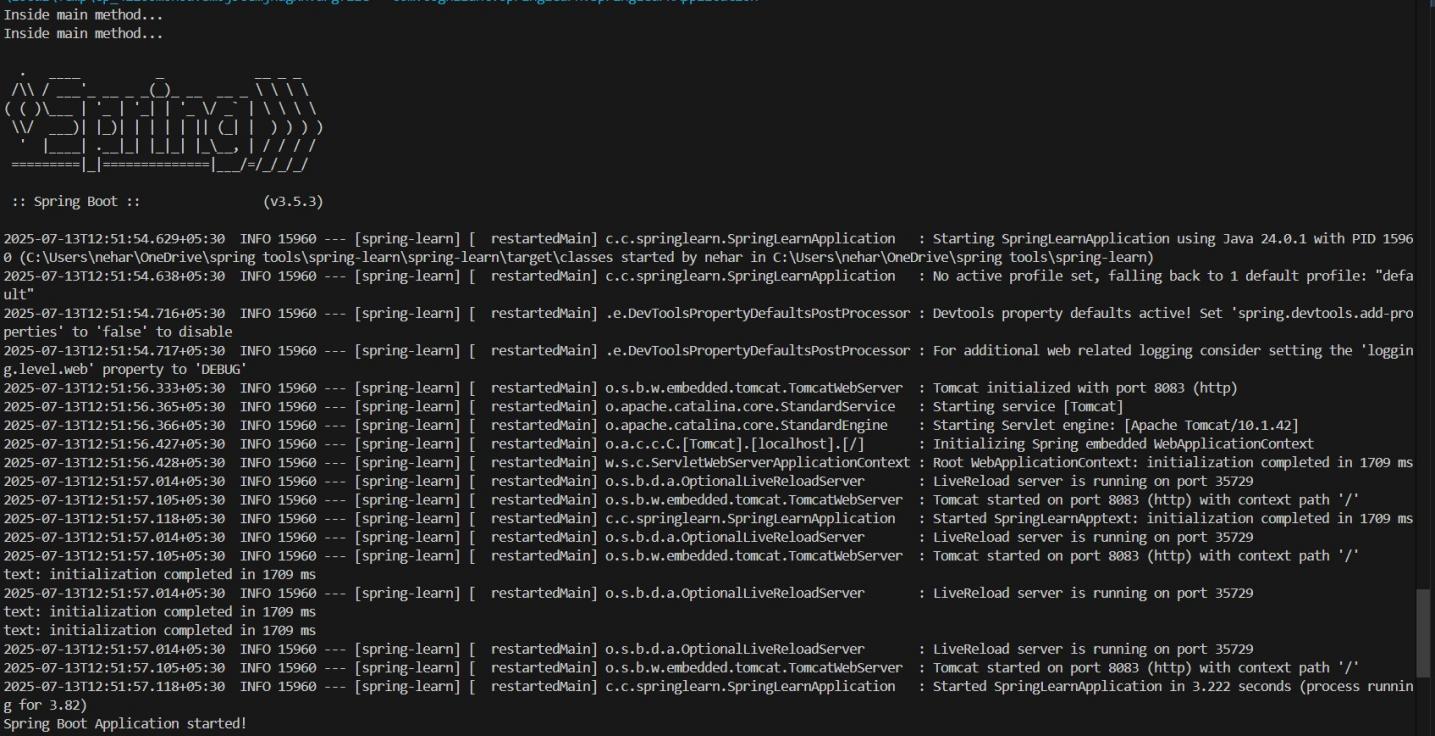
public String sayHello() { LOGGER.info("START: sayHello()");

String response = "Hello World!!"; LOGGER.info("END: sayHello()"); return response;

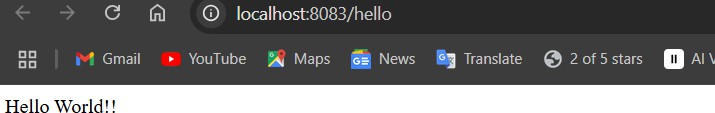
}

}

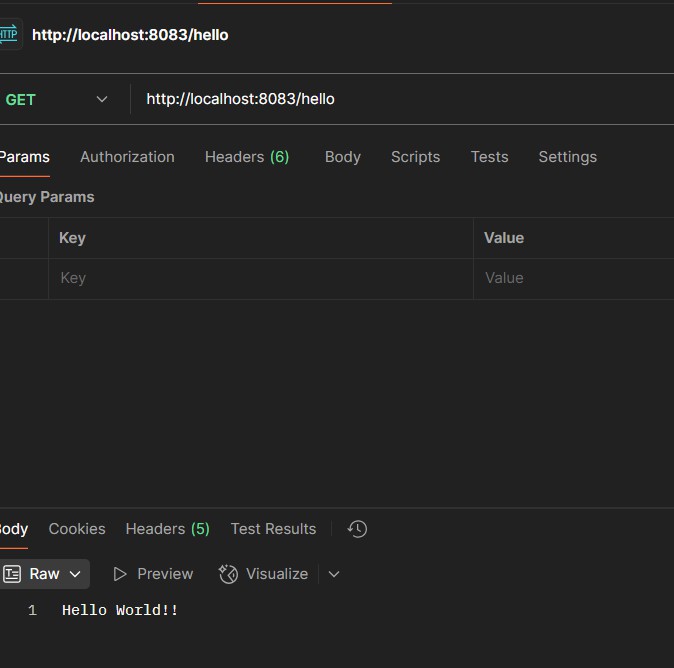
**OUTPUT:**

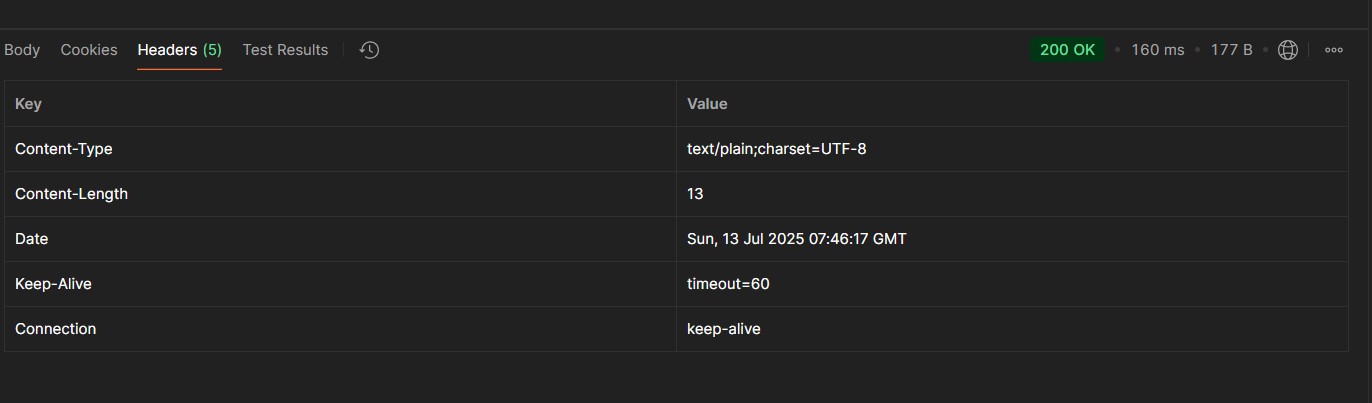
****

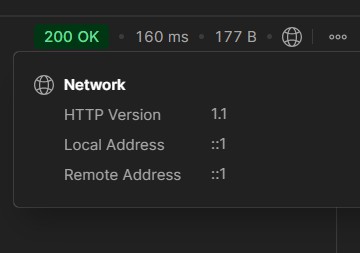
**WEB BROWSER:**

****

# Postman Console, Header, Network tabs:

****



****

**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:**

package com.cognizant.springlearn.controller; import com.cognizant.springlearn.country; import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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);

@Autowired

private ApplicationContext context;

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

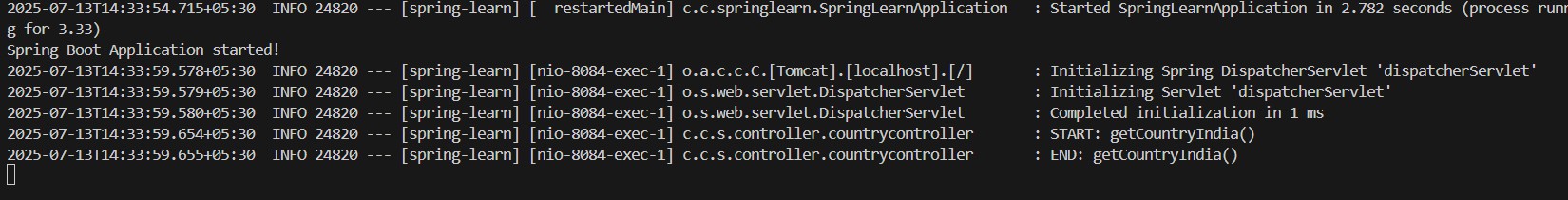
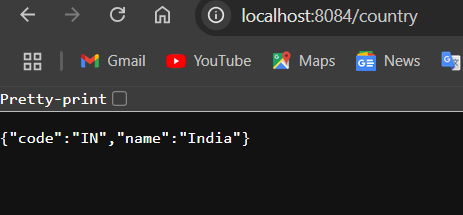
LOGGER.info("START: getCountryIndia()");

country country = (country) context.getBean("country"); LOGGER.info("END: getCountryIndia()");

return country;

}

}

****