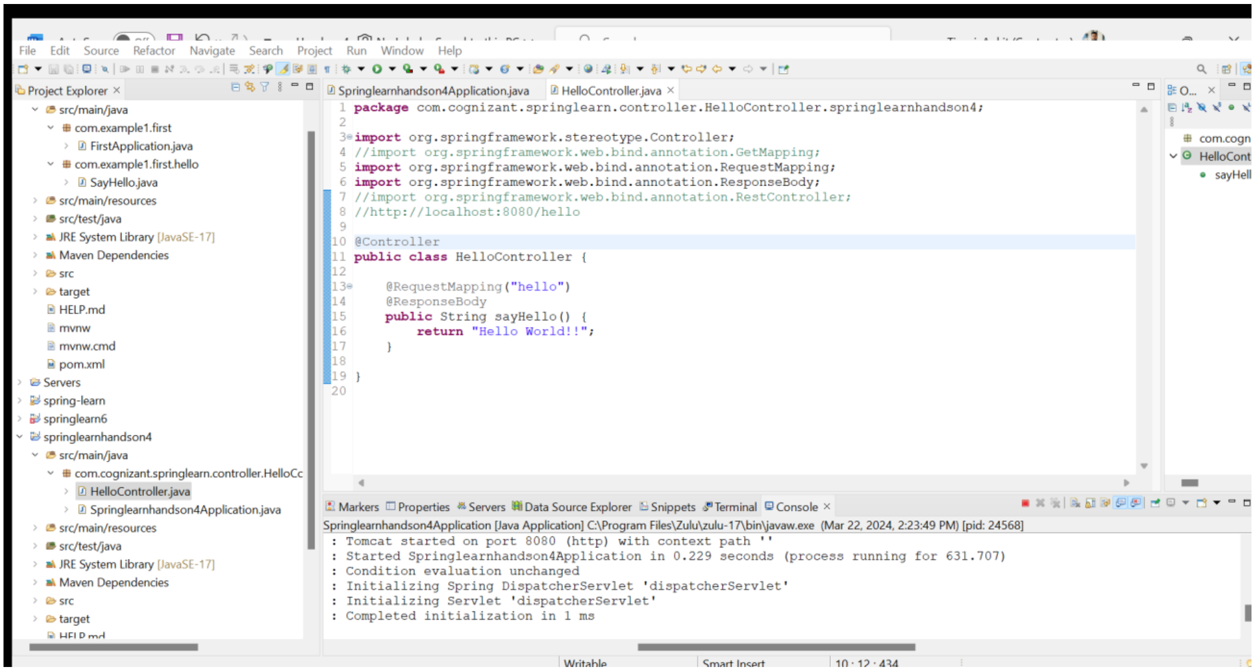
Hello World RESTful Web Service

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.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!!

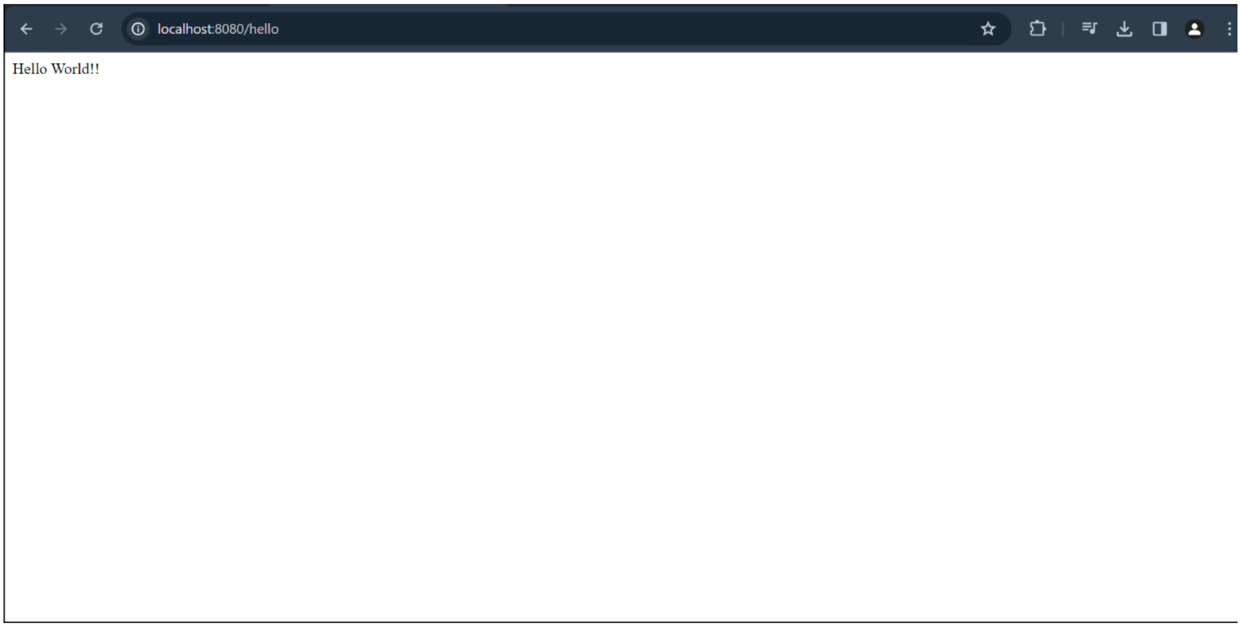


```
package com.cognizant.springlearn.controller.HelloController.springlearnhandson4;

import org.springframework.stereotype.Controller;
//import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;
//import org.springframework.web.bind.annotation.RestController;
//http://localhost:8080/hello

@Controller
public class HelloController {

    @RequestMapping("hello")
    @ResponseBody
    public String sayHello() {
        return "Hello World!!";
    }

}
```

Springlearnhandson4Application [Java Application] C:\Program Files\Zulu\zulu-17\bin\javaw.exe (Mar 22, 2024, 2:23:49 PM) [pid: 24568]

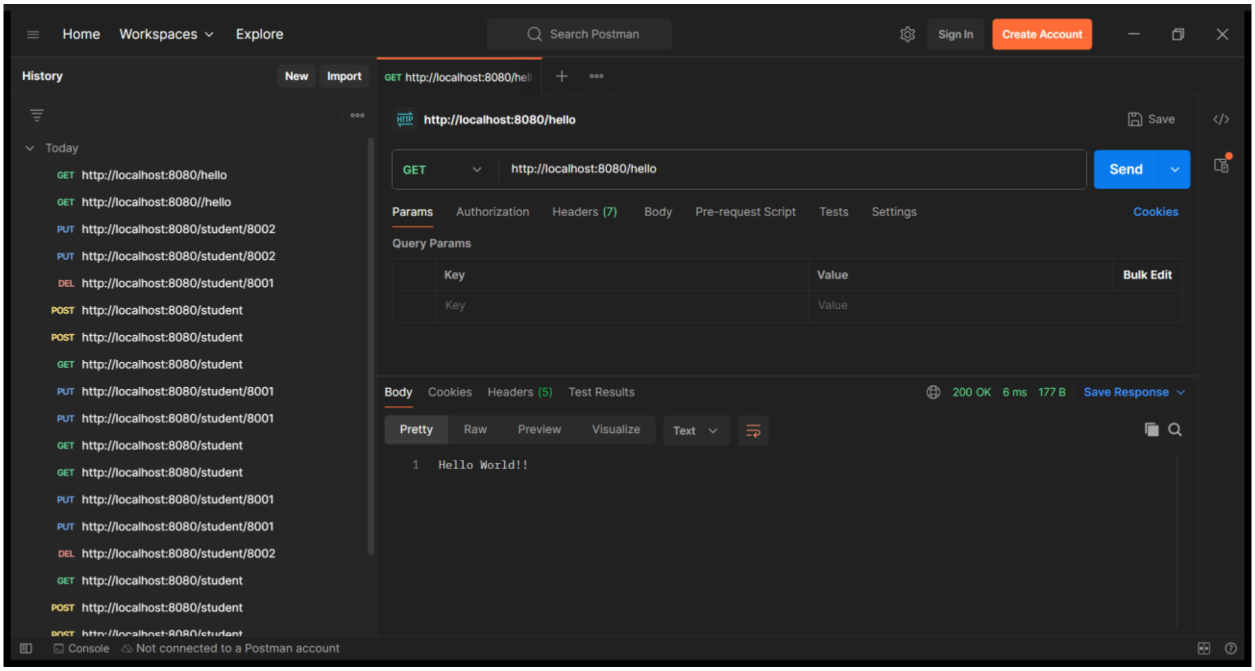
```
: Tomcat started on port 8080 (http) with context path ''
: Started Springlearnhandson4Application in 0.229 seconds (process running for 631.707)
: Condition evaluation unchanged
: Initializing Spring DispatcherServlet 'dispatcherServlet'
: Initializing Servlet 'dispatcherServlet'
: Completed initialization in 1 ms
```

OUTPUT:



```
Hello World!!
```

GET:OUTPUT


Home
Workspaces
Explore
Search Postman
Sign In
Create Account
History
New
Import
Today
GET http://localhost:8080/hello
GET http://localhost:8080//hello
PUT http://localhost:8080/student/8002
PUT http://localhost:8080/student/8002
DEL http://localhost:8080/student/8001
POST http://localhost:8080/student
POST http://localhost:8080/student
GET http://localhost:8080/student
PUT http://localhost:8080/student/8001
PUT http://localhost:8080/student/8001
GET http://localhost:8080/student
GET http://localhost:8080/student
PUT http://localhost:8080/student/8001
PUT http://localhost:8080/student/8001
DEL http://localhost:8080/student/8002
GET http://localhost:8080/student
POST http://localhost:8080/student
POST http://localhost:8080/student
GET http://localhost:8080/hel
Save
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
Query Params
Key
Value
Bulk Edit
Body
Cookies
Headers (5)
Test Results
Save Response
Pretty
Raw
Preview
Visualize
Text
Console
Not connected to a Postman account

GET http://localhost:8080/hello

200 OK 6 ms 177 B

```
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

Country.java:

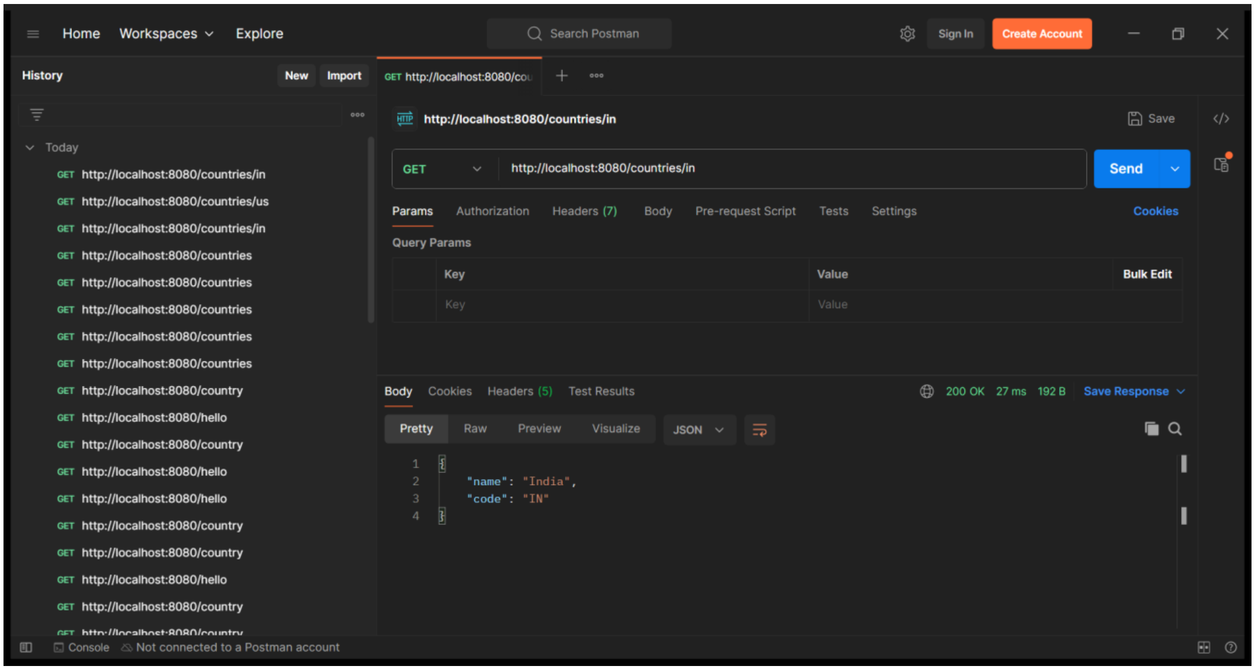
```
package com.cognizant.springlearn;

public class Country {
	private String name;
	private String code;
	public Country() {
		System.out.println("Inside Country Constructor");
	}
	public String getName() {
		System.out.println("Getting name: " + name);
		return name;
	}
	public void setName(String name) {
		System.out.println("Setting name: " + name);
		this.name = name;
	}
	public String getCode() {
		System.out.println("Getting country code: " + code);
		return code;
	}
	public void setCode(String code) {
		System.out.println("Setting country code: " + code);
		this.code = code;
	}
	@Override
	public String toString() {
		return "Country [name=" + name + ", code=" + code + "]";
	}

}
```

SpringlearnApplication.java:

```
package com.cognizant.springlearn;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
```

```
@SpringBootApplication
public class SpringLearnApplication {
	private static final Logger LOGGER =
LoggerFactory.getLogger(SpringLearnApplication.class);
	public static void main(String[] args) {
		SpringApplication.run(SpringLearnApplication.class, args);
	}
}
```

OUTPUT:



```
GET http://localhost:8080/countries/in
GET http://localhost:8080/countries/us
GET http://localhost:8080/countries/in
GET http://localhost:8080/countries
GET http://localhost:8080/countries
GET http://localhost:8080/countries
GET http://localhost:8080/countries
GET http://localhost:8080/countries
GET http://localhost:8080/country
GET http://localhost:8080/hello
GET http://localhost:8080/country
GET http://localhost:8080/hello
GET http://localhost:8080/hello
GET http://localhost:8080/country
GET http://localhost:8080/country
GET http://localhost:8080/hello
GET http://localhost:8080/country
GET http://localhost:8080/country
```

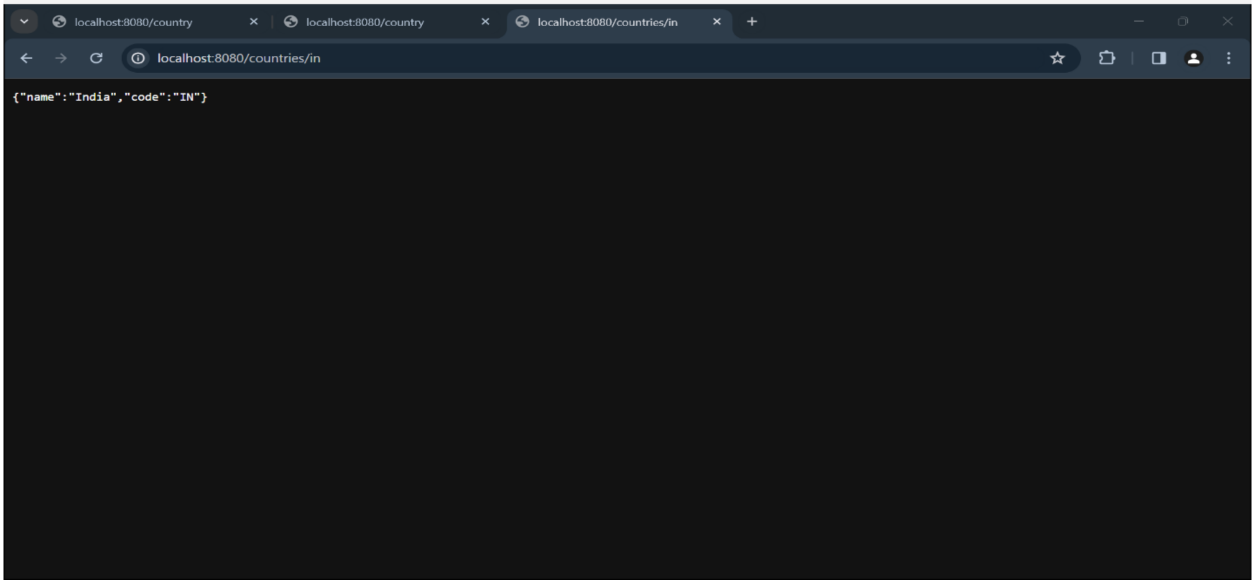
http://localhost:8080/countries/in

GET http://localhost:8080/countries/in

200 OK 27 ms 192 B

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

BROWSER OUTPUT:



localhost:8080/countries/in

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

REST - Get all countries

Write a REST service that returns all the countries.

Controller: com.cognizant.spring-learn.controller.CountryController
Method Annotation: @GetMapping("/countries")
Method Name: getAllCountries()
Method Implementation: Load country list from country.xml and return
Sample Request: http://localhost:8083/countries
Sample Response:

CountryController.java:

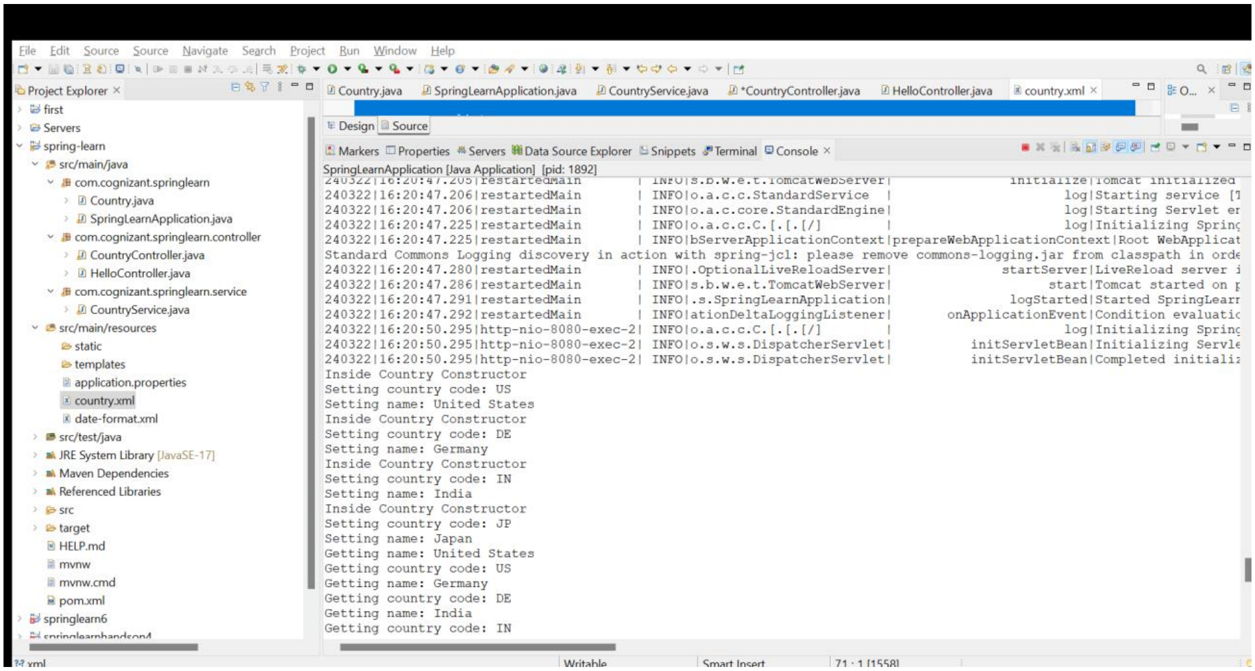
```
package com.cognizant.springlearn.controller;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import com.cognizant.springlearn.Country;
import com.cognizant.springlearn.service.*;
@RestController
public class CountryController {
    @RequestMapping(value="/country")
      public Country getCountryIndia(){
            ApplicationContext context = new
ClassPathXmlApplicationContext("country.xml");
        Country country = context.getBean("country", Country.class);
        return country;
      }
      private List<Country> countryList;
      @GetMapping("/countries")
      public List<Country> getAllCountries(){
            ApplicationContext context = new
ClassPathXmlApplicationContext("country.xml");
        ArrayList<Country> countryList = context.getBean("countryList",
ArrayList.class);

        return countryList;
      }
      @GetMapping("/countries/{code}")
    public Country getCountry(@PathVariable String code) {
            CountryService countryService = new CountryService();
         return countryService.getCountry(code);
    }

}
```

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

       <bean id="country" class="com.cognizant.springlearn.Country"
scope="prototype">

        <property name="code" value="IN" />

        <property name="name" value="India" />

    </bean>

    <bean id="us" class="com.cognizant.springlearn.Country">

        <property name="code" value="US" />

        <property name="name" value="United States" />

    </bean>

    <bean id="de" class="com.cognizant.springlearn.Country">

        <property name="code" value="DE" />

        <property name="name" value="Germany" />

    </bean>

    <bean id="in" class="com.cognizant.springlearn.Country">

        <property name="code" value="IN" />

        <property name="name" value="India" />

    </bean>

    <bean id="jp" class="com.cognizant.springlearn.Country">

        <property name="code" value="JP" />

        <property name="name" value="Japan" />

    </bean>

    <bean id="countryList" class="java.util.ArrayList">

        <constructor-arg>
            <list>
```

```
                <ref bean="us" />
                <ref bean="de" />
                <ref bean="in" />
                <ref bean="jp" />
            </list>
        </constructor-arg>
    </bean>
</beans>
```

OUTPUT:



SpringLearnApplication [Java Application] [pid: 1892]

```
240322|16:20:47.205|restartedMain        | INFO|s.b.w.e.t.TomcatWebServer|         initialize|Tomcat initialized
240322|16:20:47.206|restartedMain        | INFO|o.a.c.c.StandardService  |                log|Starting service [T
240322|16:20:47.206|restartedMain        | INFO|o.a.c.core.StandardEngine|                log|Starting Servlet er
240322|16:20:47.225|restartedMain        | INFO|o.a.c.c.C.[.[.[/]        |                log|Initializing Spring
240322|16:20:47.225|restartedMain        | INFO|bServerApplicationContext|prepareWebApplicationContext|Root WebApplicat
Standard Commons Logging discovery in action with spring-jcl: please remove commons-logging.jar from classpath in orde
240322|16:20:47.280|restartedMain        | INFO|.OptionalLiveReloadServer|        startServer|LiveReload server i
240322|16:20:47.286|restartedMain        | INFO|s.b.w.e.t.TomcatWebServer|              start|Tomcat started on p
240322|16:20:47.291|restartedMain        | INFO|.s.SpringLearnApplication|         logStarted|Started SpringLearr
240322|16:20:47.292|restartedMain        | INFO|ationDeltaLoggingListener| onApplicationEvent|Condition evaluatic
240322|16:20:50.295|http-nio-8080-exec-2| INFO|o.a.c.c.C.[.[.[/]        |                log|Initializing Spring
240322|16:20:50.295|http-nio-8080-exec-2| INFO|o.s.w.s.DispatcherServlet|     initServletBean|Initializing Servle
240322|16:20:50.295|http-nio-8080-exec-2| INFO|o.s.w.s.DispatcherServlet|     initServletBean|Completed initializ
Inside Country Constructor
Setting country code: US
Setting name: United States
Inside Country Constructor
Setting country code: DE
Setting name: Germany
Inside Country Constructor
Setting country code: IN
Setting name: India
Inside Country Constructor
Setting country code: JP
Setting name: Japan
Getting name: United States
Getting country code: US
Getting name: Germany
Getting country code: DE
Getting name: India
Getting country code: IN
```

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:

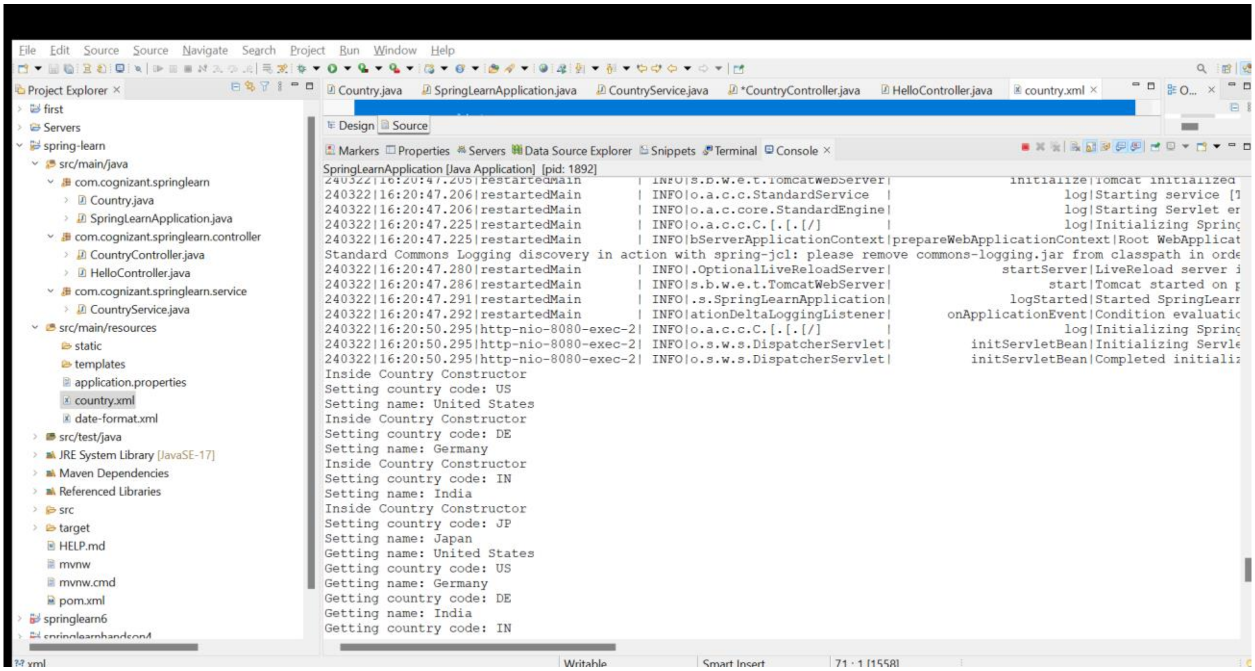
- Get the country code using @PathVariable
- Get country list from country.xml
- Iterate through the country list
- Make a case insensitive matching of country code and return the country.
- Lambda expression can also be used instead of iterating the country list

Sample Request: http://localhost:8083/country/in

CountryService.java:

```
package com.cognizant.springlearn.service;

import java.util.*;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.stereotype.Service;
import com.cognizant.springlearn.Country;
@Service
public class CountryService {
      public List<Country> getAllCountries(){
            ApplicationContext context = new
ClassPathXmlApplicationContext("country.xml");
        ArrayList<Country> countryList = context.getBean("countryList",
ArrayList.class);
        return countryList;
      }
      public Country getCountry(String code) {
        List<Country> countries = getAllCountries();
        return countries.stream().filter(c ->
c.getCode().equalsIgnoreCase(code))
                .findFirst()
                .orElse(null);
   }
}
```

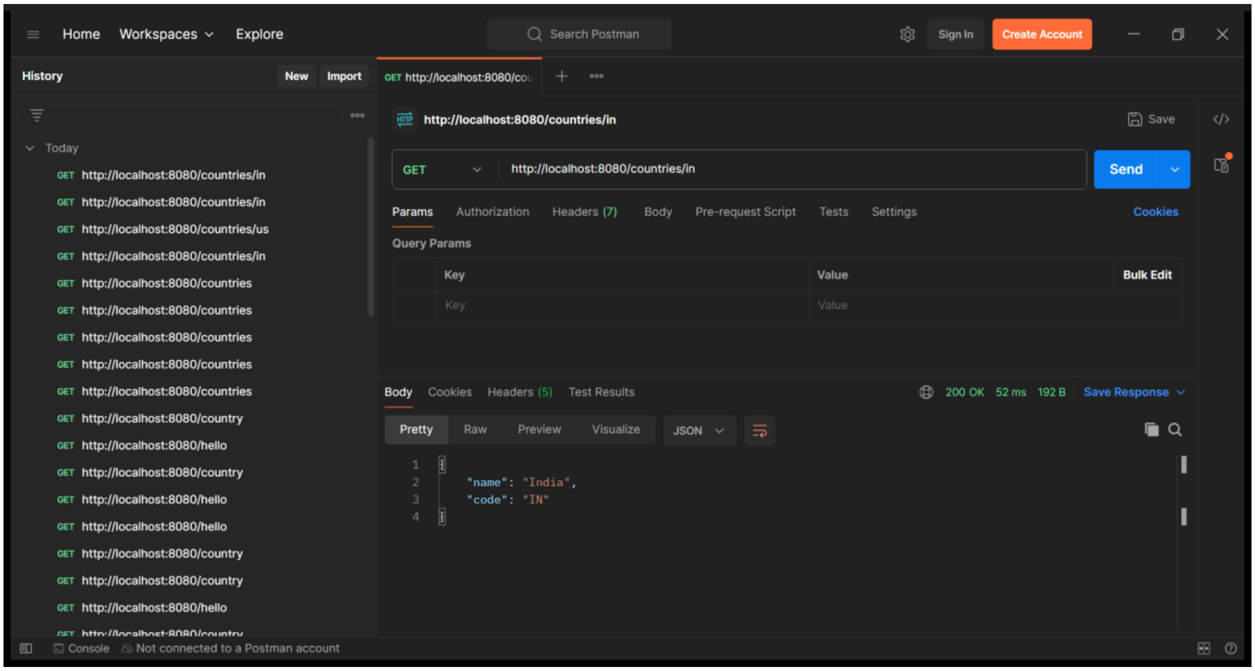
OUTPUT: ECLIPSE



SpringLearnApplication [Java Application] [pid: 1892]

```
240322|16:20:47.205|restartedMain        | INFO|s.b.w.e.t.TomcatWebServer|         initialize|Tomcat initialized
240322|16:20:47.206|restartedMain        | INFO|o.a.c.c.StandardService  |                log|Starting service [T
240322|16:20:47.206|restartedMain        | INFO|o.a.c.core.StandardEngine|                log|Starting Servlet er
240322|16:20:47.225|restartedMain        | INFO|o.a.c.c.C.[.[.[/]        |                log|Initializing Spring
240322|16:20:47.225|restartedMain        | INFO|bServerApplicationContext|prepareWebApplicationContext|Root WebApplicat
Standard Commons Logging discovery in action with spring-jcl: please remove commons-logging.jar from classpath in orde
240322|16:20:47.280|restartedMain        | INFO|.OptionalLiveReloadServer|        startServer|LiveReload server i
240322|16:20:47.286|restartedMain        | INFO|s.b.w.e.t.TomcatWebServer|              start|Tomcat started on p
240322|16:20:47.291|restartedMain        | INFO|.s.SpringLearnApplication|         logStarted|Started SpringLearr
240322|16:20:47.292|restartedMain        | INFO|ationDeltaLoggingListener|   onApplicationEvent|Condition evaluatic
240322|16:20:50.295|http-nio-8080-exec-2| INFO|o.a.c.c.C.[.[.[/]        |                log|Initializing Spring
240322|16:20:50.295|http-nio-8080-exec-2| INFO|o.s.w.s.DispatcherServlet|     initServletBean|Initializing Servle
240322|16:20:50.295|http-nio-8080-exec-2| INFO|o.s.w.s.DispatcherServlet|     initServletBean|Completed initializ
Inside Country Constructor
Setting country code: US
Setting name: United States
Inside Country Constructor
Setting country code: DE
Setting name: Germany
Inside Country Constructor
Setting country code: IN
Setting name: India
Inside Country Constructor
Setting country code: JP
Setting name: Japan
Getting name: United States
Getting country code: US
Getting name: Germany
Getting country code: DE
Getting name: India
Getting country code: IN
```

GET:OUTPUT



GET http://localhost:8080/countries/in
GET http://localhost:8080/countries/in
GET http://localhost:8080/countries/us
GET http://localhost:8080/countries/in
GET http://localhost:8080/countries
GET http://localhost:8080/countries
GET http://localhost:8080/countries
GET http://localhost:8080/countries
GET http://localhost:8080/countries
GET http://localhost:8080/country
GET http://localhost:8080/hello
GET http://localhost:8080/country
GET http://localhost:8080/hello
GET http://localhost:8080/hello
GET http://localhost:8080/country
GET http://localhost:8080/country
GET http://localhost:8080/hello
GET http://localhost:8080/country

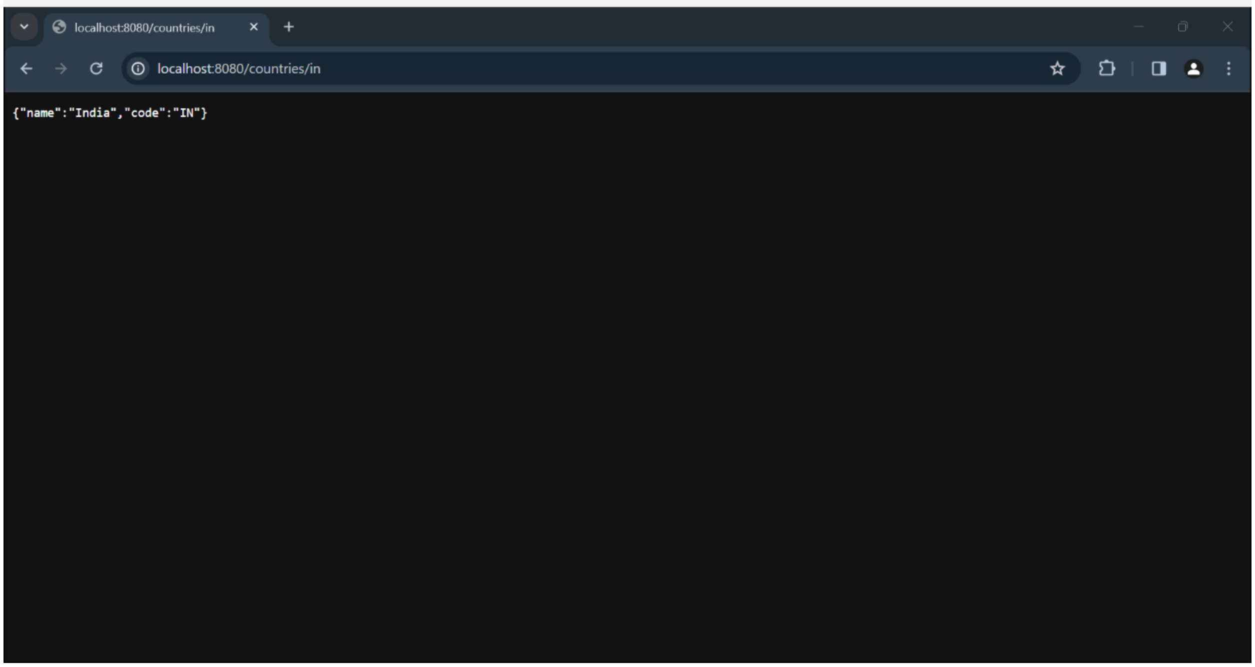
http://localhost:8080/countries/in

GET http://localhost:8080/countries/in

200 OK 52 ms 192 B

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

BROWSER OUTPUT:



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