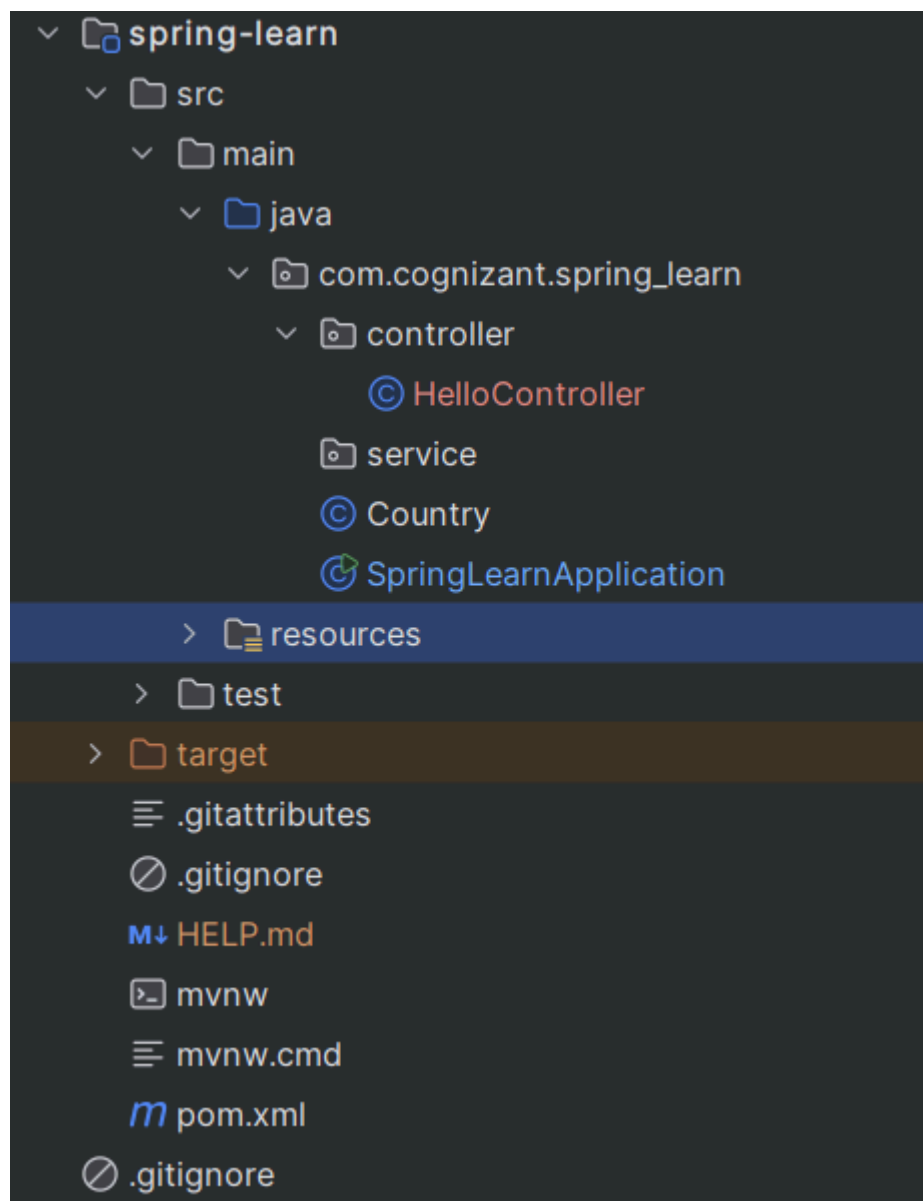


Spring REST Hands on

Source Code: [Here](#)

Hello World RESTful Web Service

Source Structure



Code

```
package com.cognizant.spring_learn.controller;  
  
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/hello")
public class HelloController {
    @GetMapping
    public String sayHello(){
        return "Hello World!!";
    }
}
```

Output

Console logging messages using Logger

```
250710|12:36:47.505|http-nio-8083-exec-2|DEBUG|a.j.AuthConfigFactoryImpl| log>Loading persistent provider registrations from [C:\Users\hp\AppData\Local\Te
250710|12:36:47.505|http-nio-8083-exec-2| INFO|o.a.c.c.C.[.[/] | log|Initializing Spring DispatcherServlet 'dispatcherServlet'
250710|12:36:47.505|http-nio-8083-exec-2| INFO|o.s.w.s.DispatcherServlet| initServletBean|Initializing Servlet 'dispatcherServlet'
250710|12:36:47.505|http-nio-8083-exec-2| INFO|o.s.w.s.DispatcherServlet| initServletBean|Completed initialization in 0 ms
250710|12:36:47.536|http-nio-8083-exec-2| INFO|c.c.s.c.HelloController | sayHello|Start sayHello()
250710|12:36:47.536|http-nio-8083-exec-2| INFO|c.c.s.c.HelloController | sayHello|End sayHello()
```

Calling /hello route from Postman

GET Say Hello

Week-4 / Spring Rest / Say Hello

GET http://localhost:8083/hello

Headers (8)

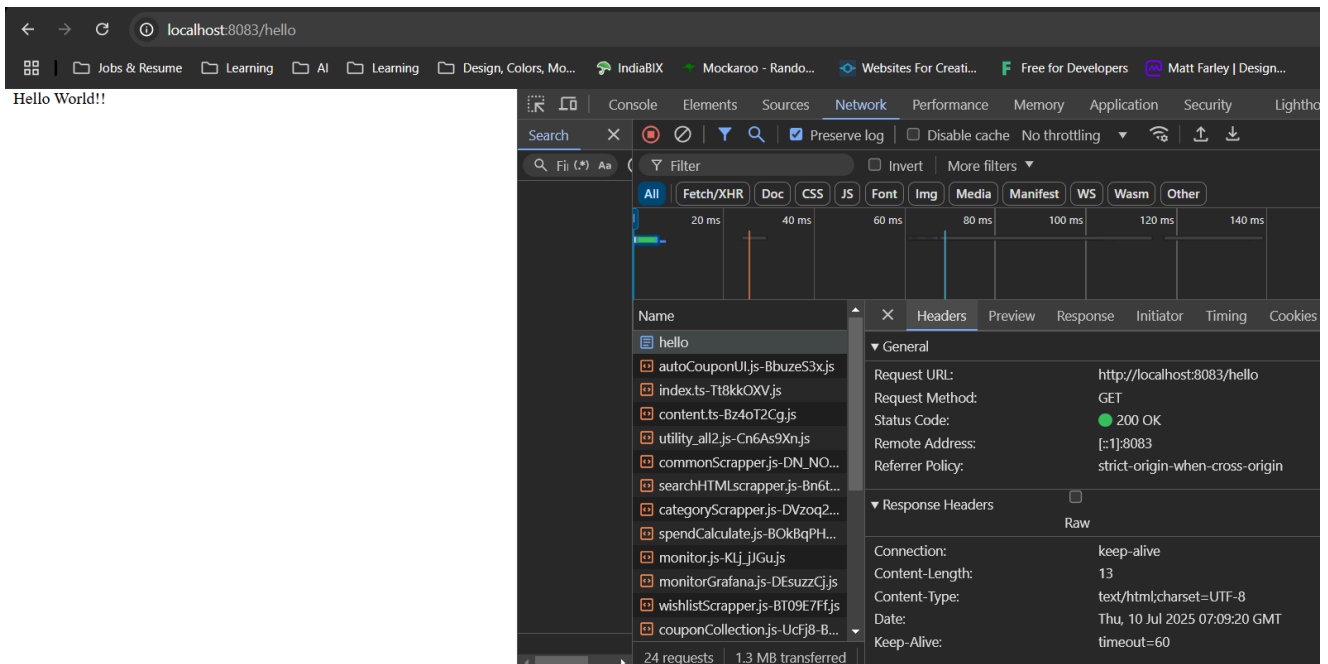
Key	Value	Description
Cookie	refresh=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlb...	
Cache-Control	no-cache	
Postman-Token	<calculated when request is sent>	
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.43.2	
Accept	/*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	

Body

200 OK · 119 ms · 177 B

1 Hello World!!

Calling /hello route from Chrome



REST - Country Web Service

Code

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

import com.cognizant.spring_learn.Country;
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.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/country")
public class CountryController {

    @GetMapping
    public Country getCountryIndia(){
        ApplicationContext context = new
ClassPathXmlApplicationContext("country.xml");
        return context.getBean("country", Country.class);
    }
}
```

Output

What happens in the controller method?

Here we have created an endpoint `/country` that returns information about a country (in this case, India). But instead of using the modern annotation-based approach, I have used **XML configuration file** (`country.xml`) to define the `Country` bean (As we have done in previous hands on). This controller reads this bean and returns it as a JSON response (By default).

Here we have used `@RestController` which is combination of `@Controller` and `@ResponseBody`.

`@ResponseBody` tells Spring **not to render HTML** but to **write the return value into the HTTP response body** (i.e., as JSON)

How the bean is converted into JSON response?

Spring Boot uses a library called **Jackson** to automatically convert Java objects (beans) into **JSON format** when returning them from a `@RestController`.

Jackson only recognizes **Non-static** fields, that have public **getters** or are accessible and which are not marked with `@JsonIgnore`.

Calling `/hello` route from Postman

The screenshot shows the Postman interface for a GET request to `http://localhost:8083/country`. The **Headers** tab is active, displaying a list of headers:

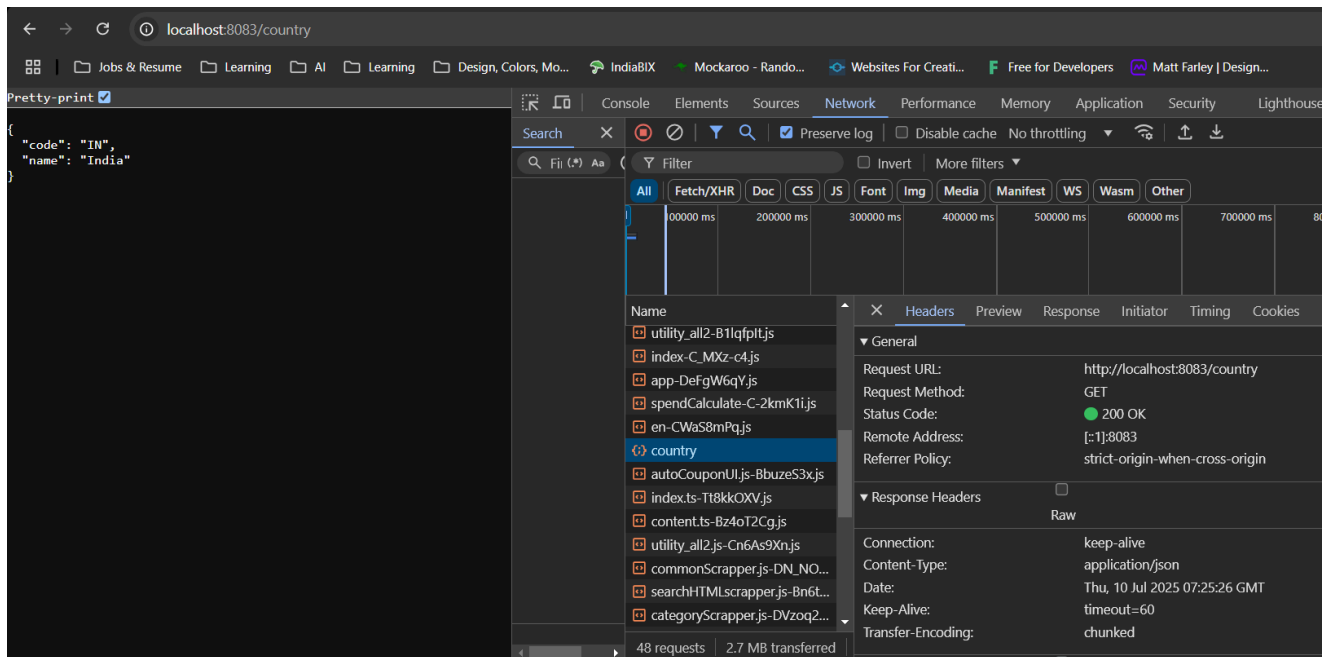
Key	Value	Description
Cookie	refresh=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlb...	
Cache-Control	no-cache	
Postman-Token	<calculated when request is sent>	
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.43.2	
Accept	/*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	

The **Body** tab is also visible, showing the JSON response:

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

The status bar at the bottom indicates a **200 OK** response with a time of 309 ms and a size of 192 B.

Calling `/hello` route from Chrome



REST - Get country based on country code

Code

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

// Import all the required packages

@RestController
@RequestMapping("/country")
public class CountryController {
    @Autowired
    private CountryService countryService;
    private static final Logger logger =
        LoggerFactory.getLogger(CountryController.class);

    @GetMapping("/{code}")
    public Country getCountry(@PathVariable String code){
        try{
            return countryService.getCountry(code);
        }
        catch (Exception e){
            logger.error(e.toString());
            return null;
        }
    }
}
```

```

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> countries = context.getBean("countryList",
List.class);

        return countries.stream()
            .filter(c -> c.getCode().equalsIgnoreCase(code))
            .findFirst()
            .orElseThrow(() -> new RuntimeException("Country not found
with code: " + code));
    }
}

```

Output

Week-4 / Spring Rest / Get Country Details with Code

GET http://localhost:8083/country/us

Overview Params Authorization Headers (8) Body Scripts Settings Cookies

Hide auto-generated headers

Key	Value	Description
Cookie	refresh=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbW...	
Cache-Control	no-cache	
Postman-Token	<calculated when request is sent>	
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.43.2	
Accept	*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	

Body Cookies (1) Headers (5) Test Results

200 OK • 32 ms • 200 B • Save Response

JSON Preview Visualize

```

1 {
2   "code": "US",
3   "name": "United States"
4 }

```

GET http://localhost:8083/country/in Send

Overview Params Authorization **Headers (8)** Body Scripts Settings

Cookies

Headers Hide auto-generated headers

	Key	Value	Description		Bulk Edit	Presets
<input checked="" type="checkbox"/>	Cookie	refresh=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbW...				
<input checked="" type="checkbox"/>	Cache-Control	no-cache				
<input checked="" type="checkbox"/>	Postman-Token	<calculated when request is sent>				
<input checked="" type="checkbox"/>	Host	<calculated when request is sent>				
<input checked="" type="checkbox"/>	User-Agent	PostmanRuntime/7.43.2				
<input checked="" type="checkbox"/>	Accept	*/*				
<input checked="" type="checkbox"/>	Accept-Encoding	gzip, deflate, br				
<input checked="" type="checkbox"/>	Connection	keep-alive				
	Key	Value	Description			

Body Cookies (1) Headers (5) Test Results

200 OK • 47 ms • 192 B • Save Response

{ } JSON Preview Visualize

Copy Paste Search

```

1  {}
2  |
3  |   "code": "IN",
4  |   "name": "India"
5  |
6  }
```