

# Hello World REST API:

## Steps:

### 1. CountryController.java:

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

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RestController;

import com.cognizant.spring_learn.model.Country;
import com.cognizant.spring_learn.service.CountryService;

@RestController

public class CountryController {

    @Autowired

    private CountryService countryService;

    @GetMapping("/countries/{code}")

    public Country getCountry(@PathVariable String code) {
```

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

## 2. Country.java:

```
package com.cognizant.spring_learn.model;  
  
public class Country {  
  
    private String code;  
  
    private String name;  
  
    // Default constructor  
    public Country() {}  
  
    // Parameterized constructor  
    public Country(String code, String name) {  
  
        this.code = code;  
  
        this.name = name;  
  
    }  
  
    // Getters and Setters  
    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;

}

@Override

public String toString() {

    return "Country{" +

        "code='" + code + '\'' +

        ", name='" + name + '\'' +

        '}';

}

}
```

### 3. CountryService.java:

```
//
src/main/java/com/cognizant/springlearn/service/CountryService.java

package com.cognizant.spring_learn.service;

import java.util.ArrayList;

import java.util.List;

import org.springframework.stereotype.Service;

import com.cognizant.spring_learn.model.Country;

@Service

public class CountryService {

    private List<Country> countries = new ArrayList<>();

    public CountryService() {

        // Initialize with some countries

        countries.add(new Country("IN", "India"));

        countries.add(new Country("US", "United States"));

        countries.add(new Country("DE", "Germany"));

        countries.add(new Country("JP", "Japan"));
    }
}
```

```
}

    public Country getCountry(String code) {

        return countries.stream()

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

            .findFirst()

            .orElse(null);

    }

}
```

Output:

Pretty-print ☐

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

X	Headers	Preview	Response	Initiator	Timing	Cookies
<b>▼ General</b>						
Request URL		http://localhost:8080/countries/IN				
Request Method		GET				
Status Code		<span style="color: green;">●</span> 200 OK				
Remote Address		[::1]:8080				
Referrer Policy		no-referrer				
<b>▼ Response Headers</b> <input type="checkbox"/> Raw						
Connection		keep-alive				
Content-Type		application/json				
Date		Fri, 11 Jul 2025 08:13:02 GMT				
Keep-Alive		timeout=60				
Transfer-Encoding		chunked				
<b>▼ Request Headers</b> <input type="checkbox"/> Raw						
Accept		text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7				
Accept-Encoding		gzip, deflate, br, zstd				
Accept-Language		en-US,en;q=0.9,fr;q=0.8				
Cache-Control		max-age=0				
Connection		keep-alive				
Cookie		ajs_anonymous_id=ade18666-7c89-4325-948f-8ba5c6478286; username=localhost-8888="2 1:0 10:1750691772 23:username=localhost-8888 200:eyJ1c2VybmFTZSI6ICh4OGlSMjQxZGE4MTIONzZjYTRQ3OWE2YT8kNmQzMmVjNSIsICJ1YW11ljogIkFub255bW91cyBIZWdlbW9uZSIsICIkaXNwbGF5X25hbWUOIAlQW5vbmltb3ZlElhIZ2Yt25liiwglmluaXRpYWxzljogIkFlIiwglmNvbG9yIjogbnVsbH0=Ij5035ed8b58f83436a8319d1e791bf3af95188528d62cb690cd163325edd6";				

The screenshot shows the REST Client application interface. At the top, there's a URL bar with the URL `http://localhost:8080/countries/IN`. Below the URL bar, there's a dropdown menu showing the HTTP method `GET`. To the right of the URL bar, there are buttons for `Save` and `Share`. Below the URL bar, there's a `Send` button. Below the `Send` button, there's a table with columns `Key`, `Value`, and `Description`. The table has two rows, both with empty cells. Below the table, there's a `Params` section with a `Key` and a `Value` column. The `Params` section is currently empty. Below the `Params` section, there's a `Body` section with a `JSON` tab selected. The `JSON` tab shows the response body as a JSON object: `{"code": "IN", "name": "India"}`. The response status is `200 OK` with a response time of `17 ms` and a response size of `192 B`. The response is also shown in a `Preview` tab.

GET

http://localhost:8080/countries/IN

Send

Params

Authorization

Headers (6)

Body

Scripts

Settings

Cookies

Key	Value	Description
Key	Value	Description

Body

Cookies

Headers (5)

Test Results

200 OK • 17 ms • 192 B • •

Key	Value
Content-Type	application/json
Transfer-Encoding	chunked
Date	Fri, 11 Jul 2025 08:13:54 GMT
Keep-Alive	timeout=60
Connection	keep-alive

http://localhost:8080/hello 

Save

Share

GET

http://localhost:8080/hello

Send

Params

Authorization

Headers (6)

Body

Scripts

Settings

Cookies

Body

Cookies

Headers (5)

Test Results

200 OK • 306 ms • 177 B • •

Key	Value
-----	-------