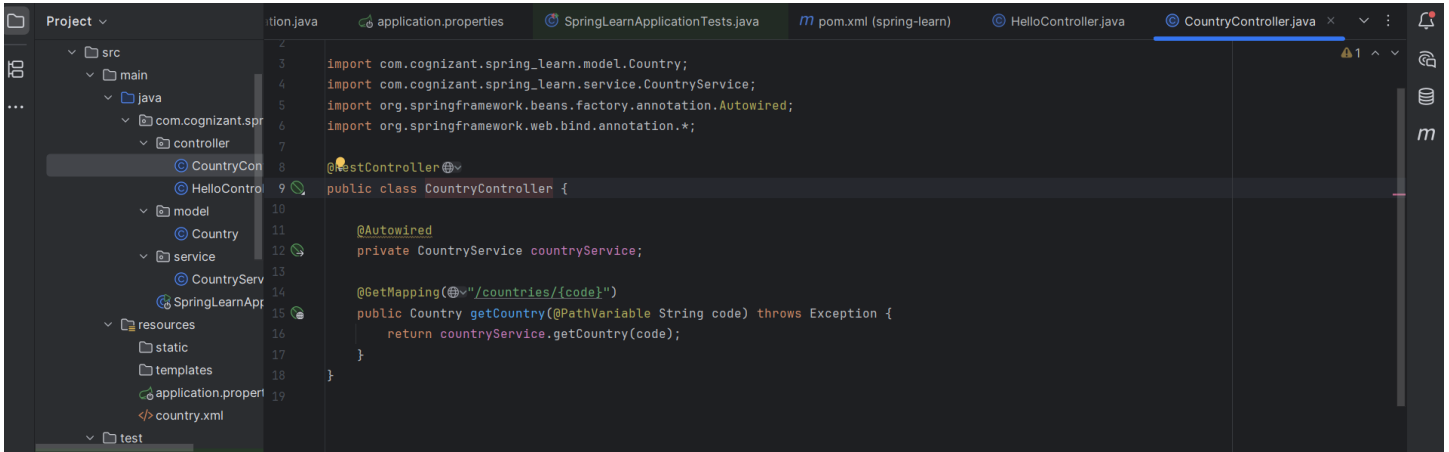


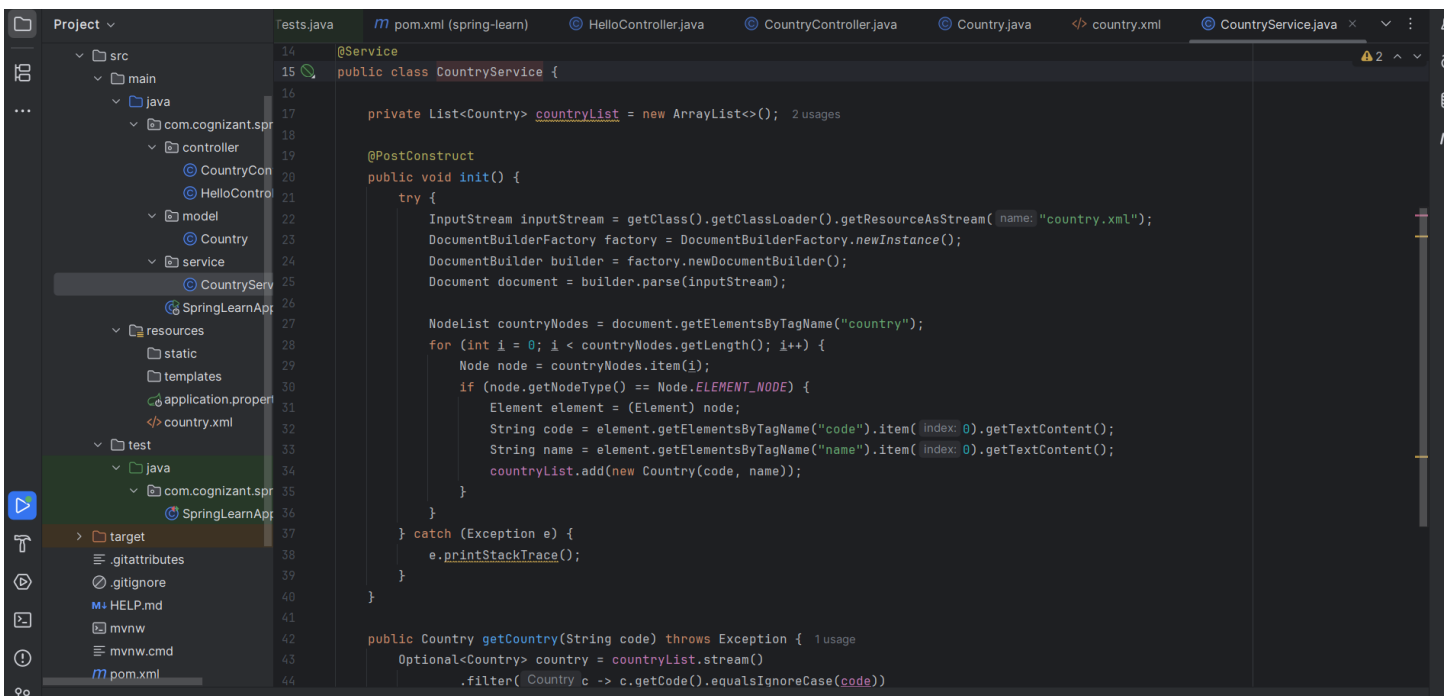
# REST - Get country based on country code

## CountryController.java



```
1  import com.cognizant.spring_learn.model.Country;
2
3  import com.cognizant.spring_learn.service.CountryService;
4
5  import org.springframework.beans.factory.annotation.Autowired;
6  import org.springframework.web.bind.annotation.*;
7
8  @RestController
9  public class CountryController {
10
11     @Autowired
12     private CountryService countryService;
13
14     @GetMapping("/countries/{code}")
15     public Country getCountry(@PathVariable String code) throws Exception {
16         return countryService.getCountry(code);
17     }
18 }
19
```

## CountryService.java



```
14 @Service
15 public class CountryService {
16
17     private List<Country> countryList = new ArrayList<>();
18
19     @PostConstruct
20     public void init() {
21         try {
22             InputStream inputStream = getClass().getClassLoader().getResourceAsStream("country.xml");
23             DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
24             DocumentBuilder builder = factory.newDocumentBuilder();
25             Document document = builder.parse(inputStream);
26
27             NodeList countryNodes = document.getElementsByTagName("country");
28             for (int i = 0; i < countryNodes.getLength(); i++) {
29                 Node node = countryNodes.item(i);
30                 if (node.getNodeType() == Node.ELEMENT_NODE) {
31                     Element element = (Element) node;
32                     String code = element.getElementsByTagName("code").item(0).getTextContent();
33                     String name = element.getElementsByTagName("name").item(0).getTextContent();
34                     countryList.add(new Country(code, name));
35                 }
36             }
37         } catch (Exception e) {
38             e.printStackTrace();
39         }
40     }
41
42     public Country getCountry(String code) throws Exception {
43         Optional<Country> country = countryList.stream()
44             .filter(c -> c.getCode().equalsIgnoreCase(code))
45
```

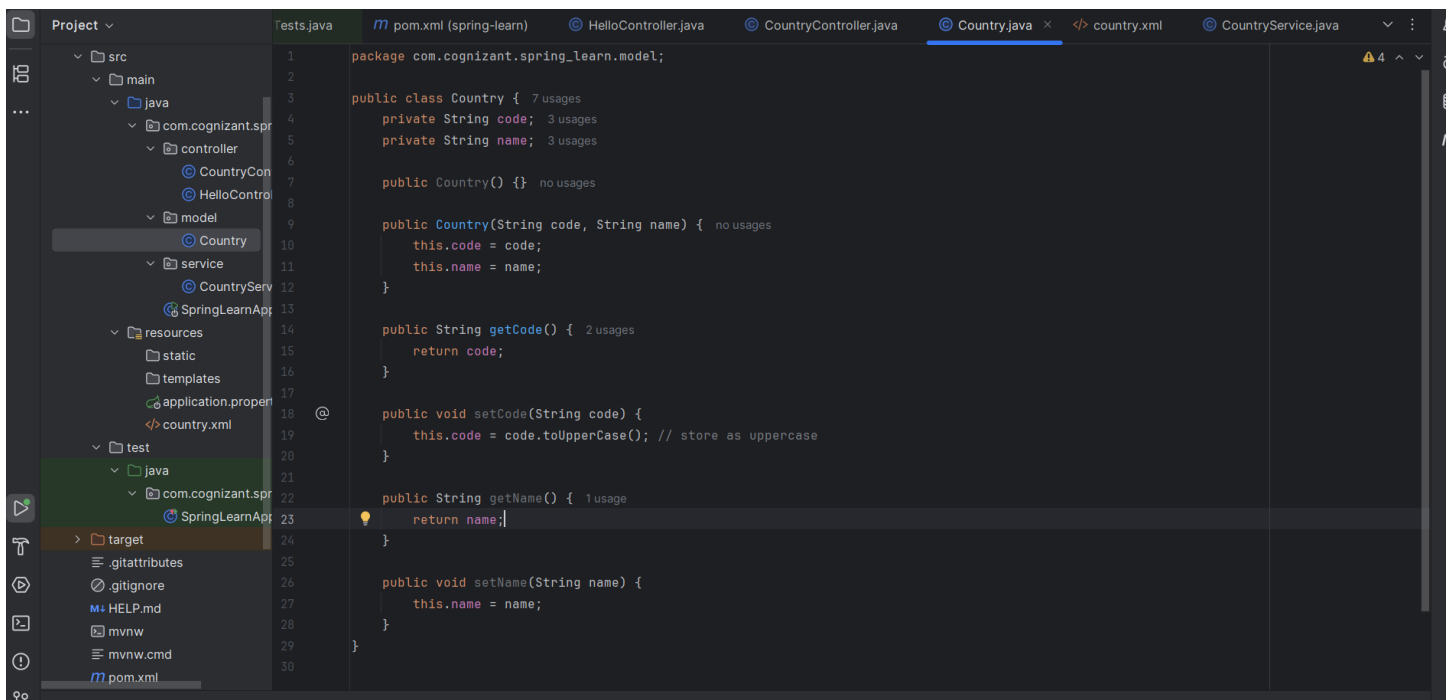
```

    public Country getCountry(String code) throws Exception { 1 usage
        Optional<Country> country = countryList.stream()
            .filter( Country c -> c.getCode().equalsIgnoreCase(code))
            .findFirst();

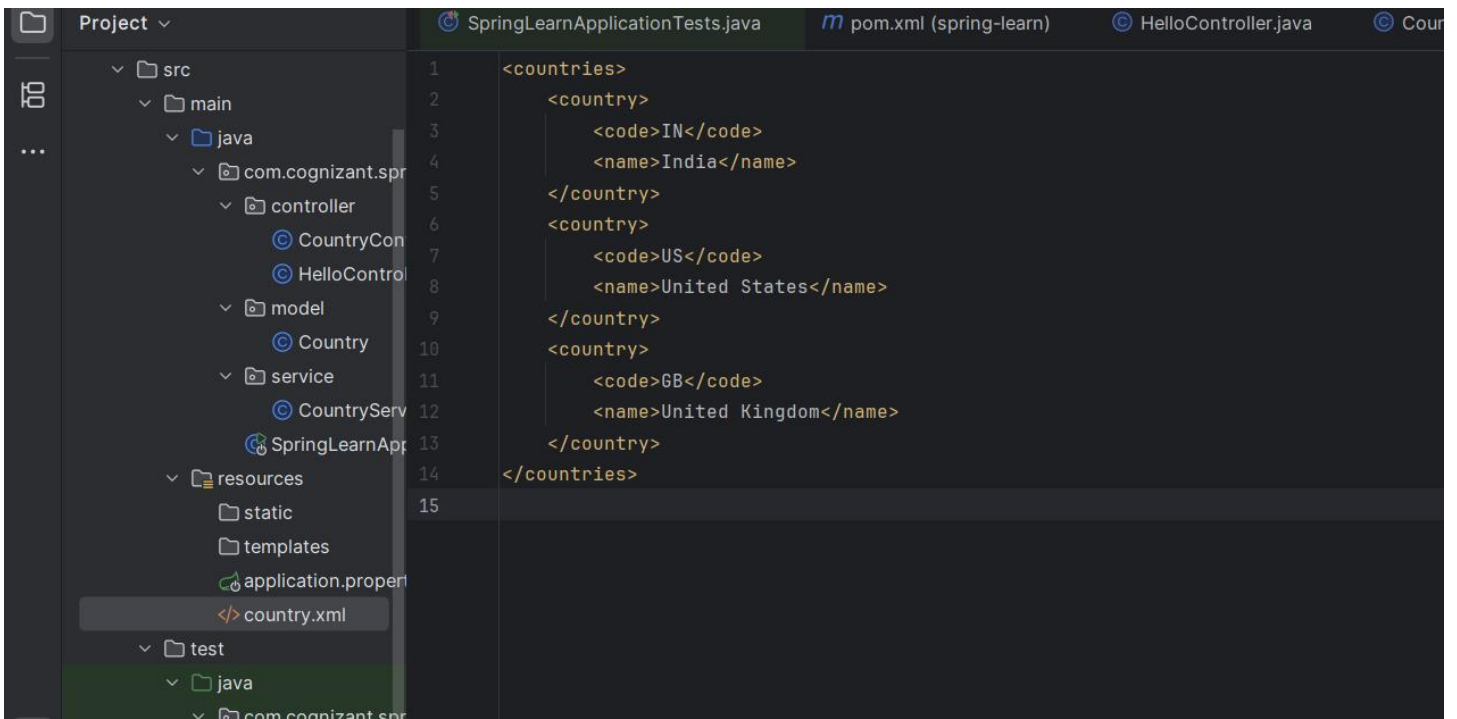
        if (country.isPresent()) {
            return country.get();
        } else {
            throw new Exception("Country with code " + code + " not found");
        }
    }
}

```

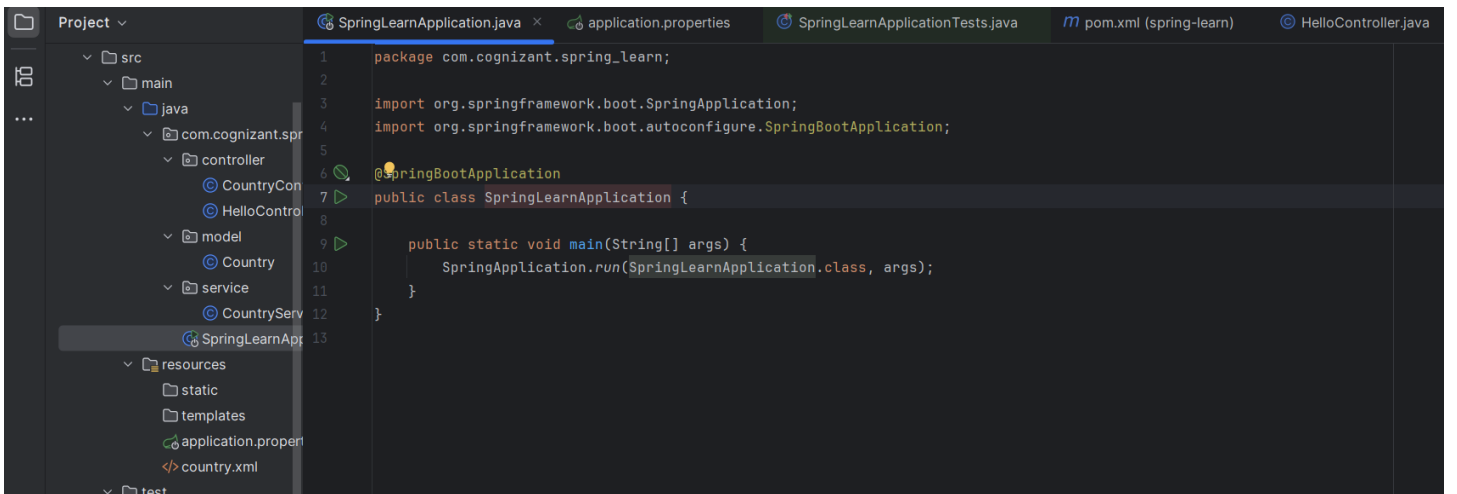
## Country.java



## Country.xml



## SpringLearnApplication.java



## POSTMAN

training / New Request

GET

http://localhost:8083/countries/in

Send

Params

Authorization

Headers (6)

Body

Scripts

Tests

Settings

Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Output:

Body

Cookies

Headers (5)

Test Results

200 OK • 88 ms • 192 B • Save Response

{ } JSON

Preview

Visualize

1 {

2     "code": "IN",

3     "name": "India"

4 }

If I want the output for United States,

GET

http://localhost:8083/countries/us

Send

Params

Authorization

Headers (6)

Body

Scripts

Tests

Settings

Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (5)

Test Results

200 OK

4 ms

200 B

e.g. Save Response

...

{ } JSON

Preview

Visualize

1 {

2 | "code": "US",

3 | "name": "United States"

4 }