**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  
  
**Sample Response**:

{

  "code": "IN",

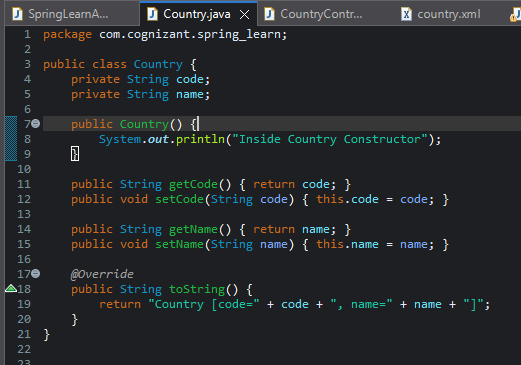
  "name": "India"

}

**Solution:**

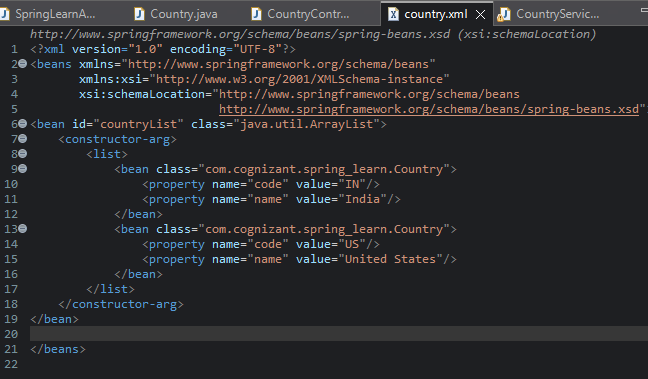
**Country Model Class**

This class defines the country object returned as a JSON response.

****

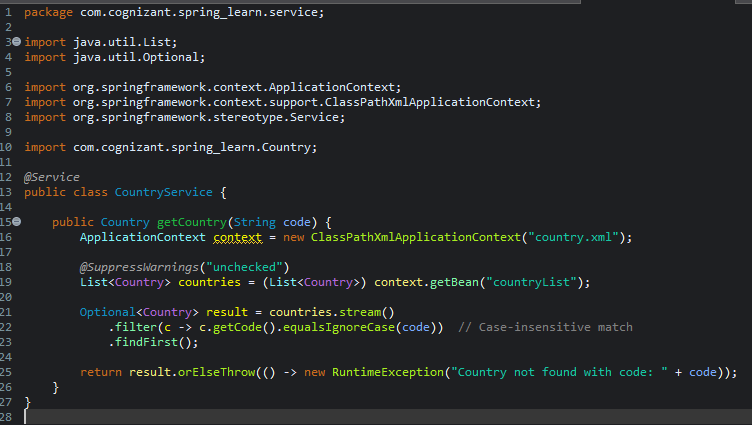
**country.xml – XML Bean Configuration**

Stores a **list of countries** that the REST service will search.



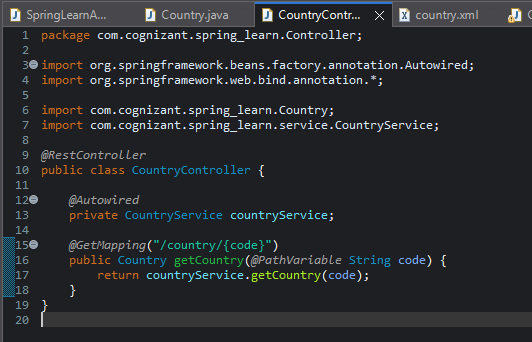
**CountryService.java – Business Logic**

Finds a matching country by code using a **lambda expression and stream filter**.



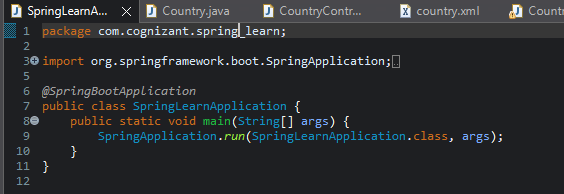
**CountryController.java – REST Endpoint**

Handles incoming HTTP GET requests and delegates to the service.

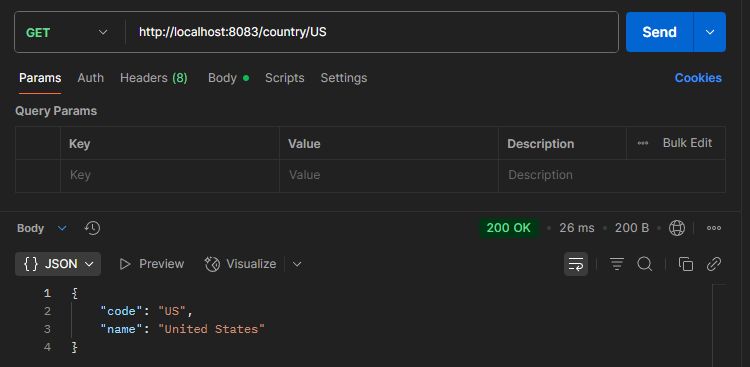


**Main Class: SpringLearnApplication.java**

Starts the Spring Boot application.



**OUTPUT:**



**Get Country by Code (Case-Insensitive) API**

* 🔹 **REST Endpoint**: GET /country/{code}  
  Returns country details based on country code.
* 🔹 **Controller**: CountryController
  + Uses @GetMapping("/country/{code}")
  + Accepts path variable code (e.g., "in", "US")
* 🔹 **Service**: CountryService
  + Loads country list from country.xml via ApplicationContext
  + Uses Java Stream API with equalsIgnoreCase() to match code
  + Returns Country object if found
  + Throws RuntimeException if no match is found
* 🔹 **Bean Configuration**: country.xml
  + Stores a list of <bean>s with different country codes and names

**Model Class**: Country.java

* Contains code and name properties
* Includes getters, setters, and toString()

🔹 **Automatic JSON Conversion**:

* Spring Boot uses **Jackson** to convert Country object to JSON