**Module 2 : Spring-rest-handson**

**Hands on: Hello World RESTful Web Service**

**Objective:** Create a simple Spring Boot REST controller that returns a "Hello World!!" message when accessed via a browser or API tool like Postman.

### **Tools & Technologies Required**

* Java 8+
* Spring Boot
* Maven
* IDE (Eclipse/IntelliJ)
* Postman or Browser

**Solution:**

**Step 1: Project Setup**

#### **Using Spring Initializr (**[**https://start.spring.io/**](https://start.spring.io/)**)**

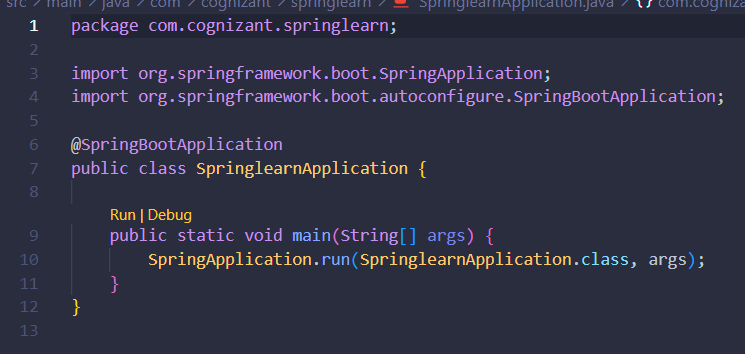
* **Project:** Maven
* **Language:** Java
* **Spring Boot Version:** 2.5+ or 3.x
* **Dependencies:**
  + Spring Web
  + Spring Boot DevTools (optional)
  + Spring Boot Test (for testing)

**Step 2: Write the Controller**

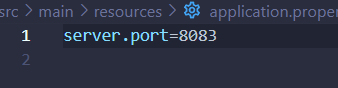
****

**Step 3: Main Application Class**

* **File: SpringLearnApplication.java**

****

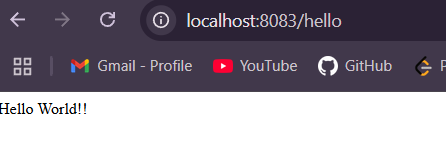
**Step 4: Configuration (Optional)**

****

**Step 5: Testing**

**Using Browser:**

* **Navigate to: http://localhost:8083/hello**
* **Expected Response: Hello World!!**

****

**Step 6: Explanation**

* **@RestController: Marks this class as a controller where every method returns a domain object instead of a view.**
* **@GetMapping("/hello"): Maps GET requests to /hello to this method.**
* **Logger: Logs method entry to track flow.**
* **Return: Plain string — "Hello World!!".**

**Hands on: REST - Country Web Service**

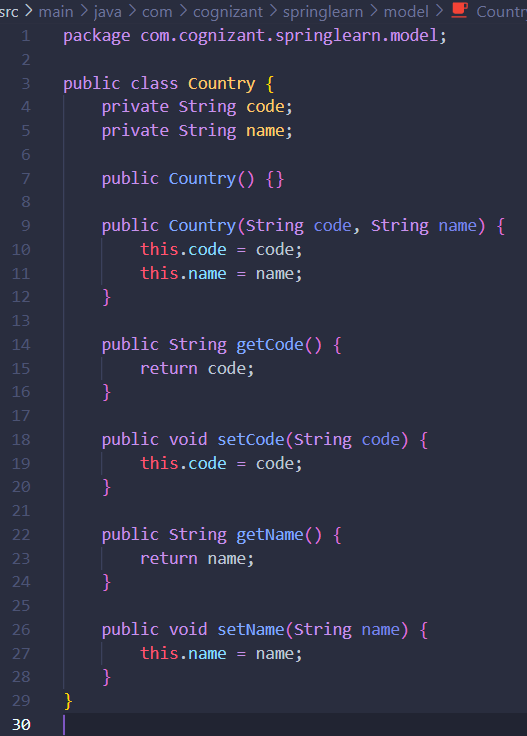
**Objective:** Build a RESTful Spring Boot web service that returns a country object (India) in JSON format. It should:

* Load the country details from a Spring XML configuration
* Return the country object via HTTP GET
* Be accessible from browser/Postman.

**Solution:**

**Step 1: Create Country Model**

**File: Country.java**

****

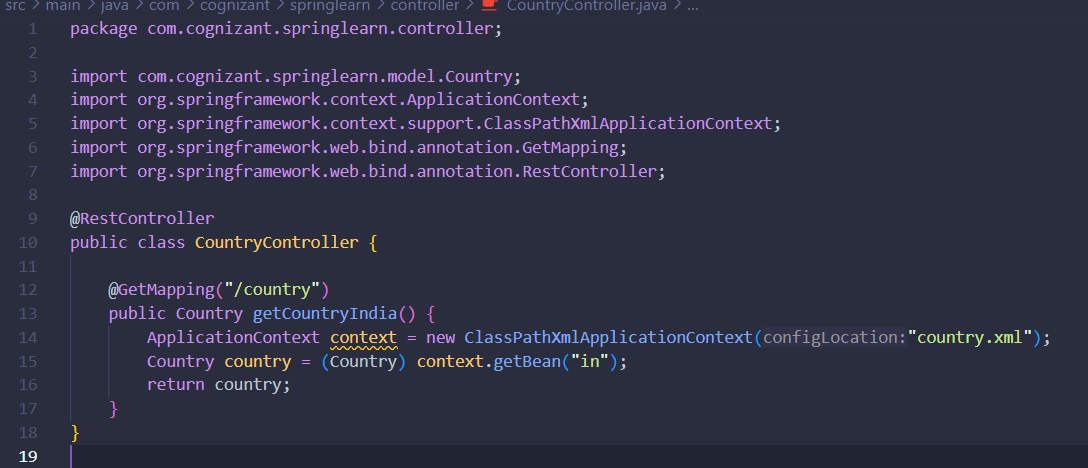
**Step 2: Define Bean in XML**

**File: country.xml (place in src/main/resources)**

****

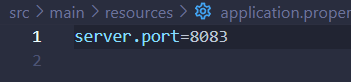
**Step 3: Create CountryController**

**File: CountryController.java**

****

**Step 4: Configure Port (Optional)**

**File: application.properties**

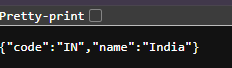
****

**Now run and test and start the Spring Boot App**

* **mvn spring-boot:run**

**Step 5: Access the EndPoint**

* **URL: http://localhost:8083/country**
* **Method: GET**

****

**Hands on: REST – Get Country Based on Country Code**

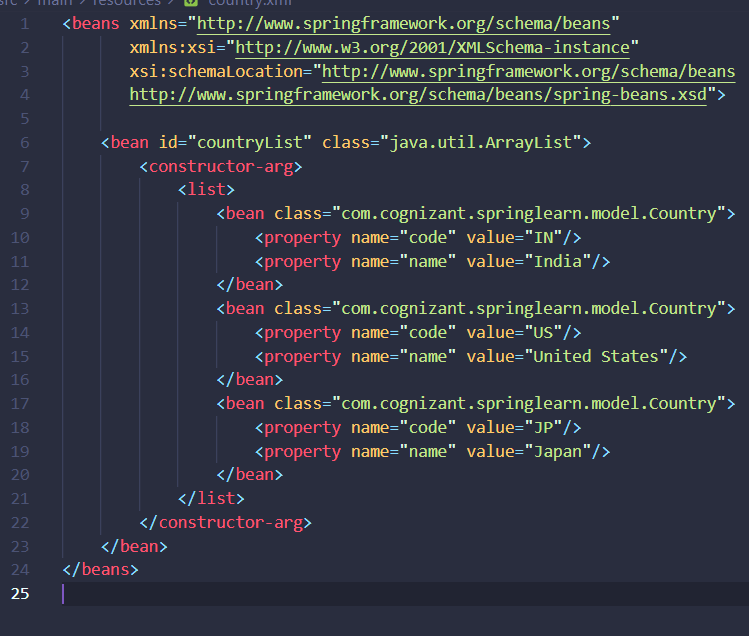
**Objective:** Enhance the Country REST web service to:

* Accept a **country code** in the URL
* Return the matching country in JSON format
* Match codes in a **case-insensitive** manner
* Return appropriate error if the code is not found

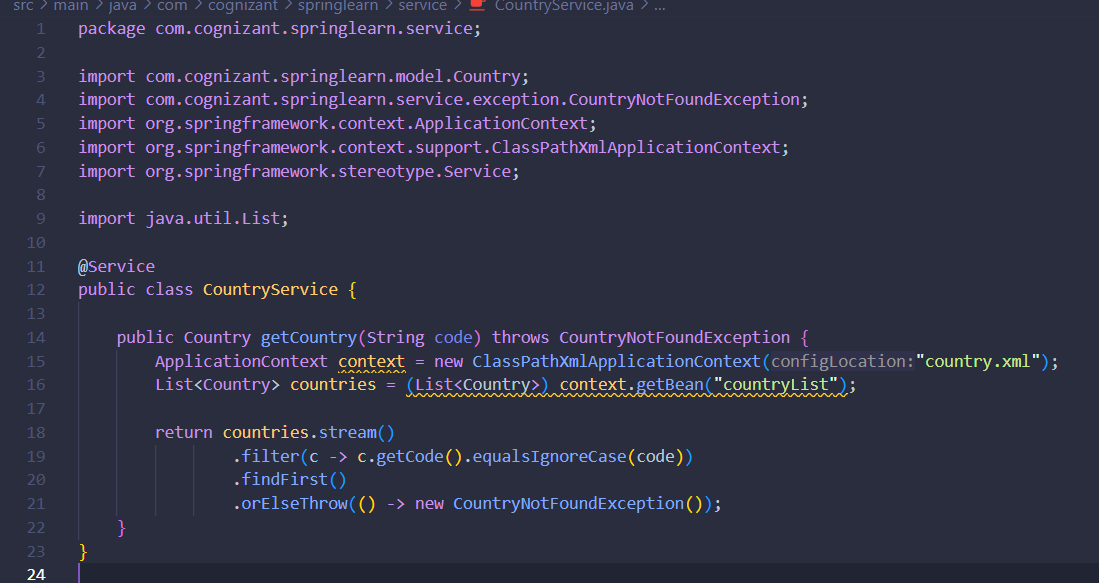
**Solution:**

**Step 1: Extend country.xml to Include Multiple Countries**

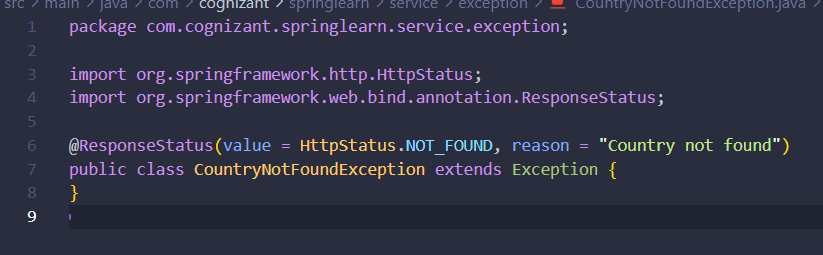
**File: country.xml**

****

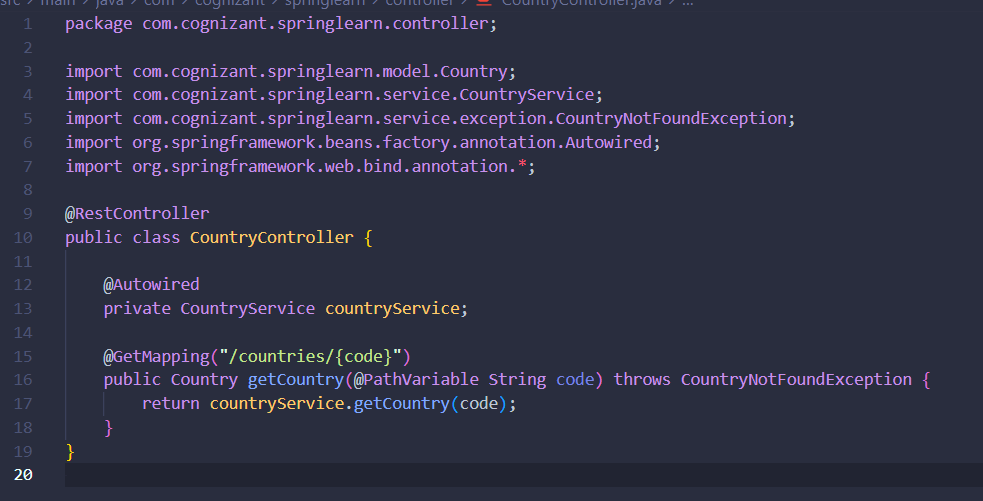
**Step 2: Create CountryService.java**

****

**Step 3: Create Exception Class CountryNotFoundException.java**

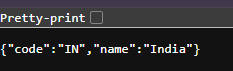
****

**Step 4: Update CountryController.java**

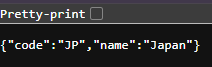
****

**Step 5: Test the API by running : mvn spring-boot:run**

[**http://localhost:8083/countries/**](http://localhost:8083/countries/in)**in**

****

[**http://localhost:8083/countries/**](http://localhost:8083/countries/in)**jp**

****