# Spring Boot – RESTful Country Web Service

Objective:

To create a RESTful web service using Spring Boot that provides country data via an HTTP GET request.

## 1. Prerequisites

- JDK 8 or above  
- Maven or Gradle  
- Spring Boot with Spring Web dependency  
- IDE (e.g., IntelliJ IDEA, Eclipse, or VS Code)

## 2. Folder Structure

CountryRestService/  
├── src/  
│ ├── main/  
│ │ ├── java/  
│ │ │ └── com/example/country/  
│ │ │ ├── Country.java  
│ │ │ ├── CountryController.java  
│ │ │ └── CountryRestApplication.java  
│ │ └── resources/  
│ │ └── application.properties  
├── pom.xml (for Maven) or build.gradle (for Gradle)

## 3. Step-by-Step Implementation

### Step 1: Create the Country Model

File: Country.java

package com.example.country;  
  
public class Country {  
 private String name;  
 private String capital;  
  
 public Country() {}  
  
 public Country(String name, String capital) {  
 this.name = name;  
 this.capital = capital;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getCapital() {  
 return capital;  
 }  
  
 public void setCapital(String capital) {  
 this.capital = capital;  
 }  
}

### Step 2: Create REST Controller

File: CountryController.java

package com.example.country;  
  
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 getCountry() {  
 return new Country("India", "New Delhi");  
 }  
}

### Step 3: Create Spring Boot Main Class

File: CountryRestApplication.java

package com.example.country;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class CountryRestApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(CountryRestApplication.class, args);  
 }  
}

### Step 4: Add Application Properties

File: application.properties

# Optional server port configuration  
# server.port=8081

### Step 5: Build and Run the Application

For Maven:

mvn spring-boot:run

For Gradle:

./gradlew bootRun

## 4. Testing the Endpoint

Open a browser or REST client and access:

http://localhost:8080/country

Expected JSON Output:

{  
 "name": "India",  
 "capital": "New Delhi"  
}