## **REST - Get country based on country code**

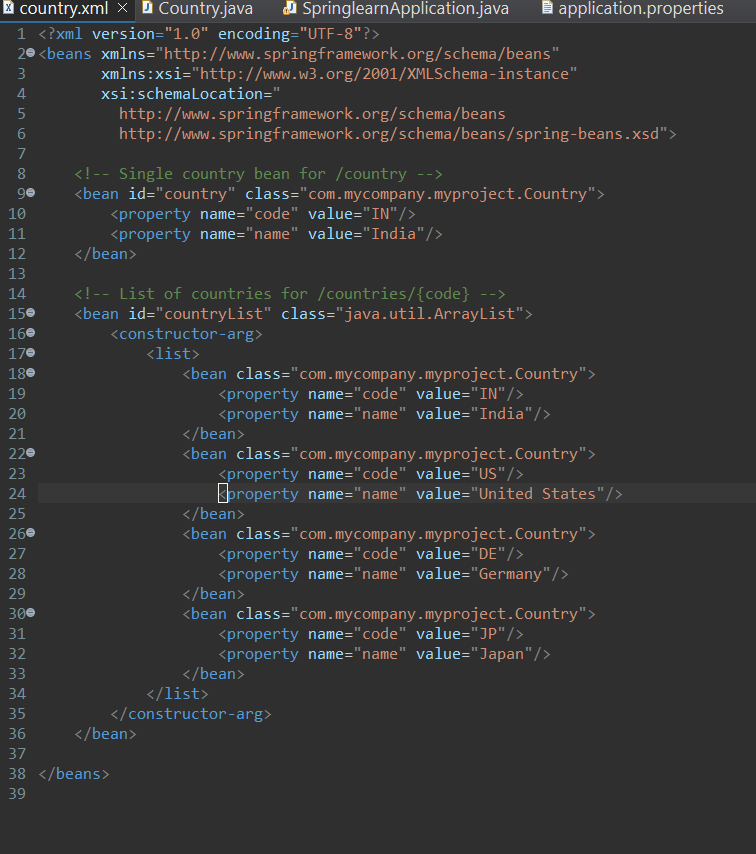
## **Objective**

* Build a REST API that returns a country based on country code.
* Code is case-insensitive.
* Country details are loaded from Spring XML configuration.

## **Updated country.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="  
 http://www.springframework.org/schema/beans  
 http://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <!-- Single country bean -->  
 <bean id="country" class="com.mycompany.myproject.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
  
 <!-- Country list bean -->  
 <bean id="countryList" class="java.util.ArrayList">  
 <constructor-arg>  
 <list>  
 <bean class="com.mycompany.myproject.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
 <bean class="com.mycompany.myproject.Country">  
 <property name="code" value="US"/>  
 <property name="name" value="United States"/>  
 </bean>  
 <bean class="com.mycompany.myproject.Country">  
 <property name="code" value="DE"/>  
 <property name="name" value="Germany"/>  
 </bean>  
 <bean class="com.mycompany.myproject.Country">  
 <property name="code" value="JP"/>  
 <property name="name" value="Japan"/>  
 </bean>  
 </list>  
 </constructor-arg>  
 </bean>  
  
</beans>

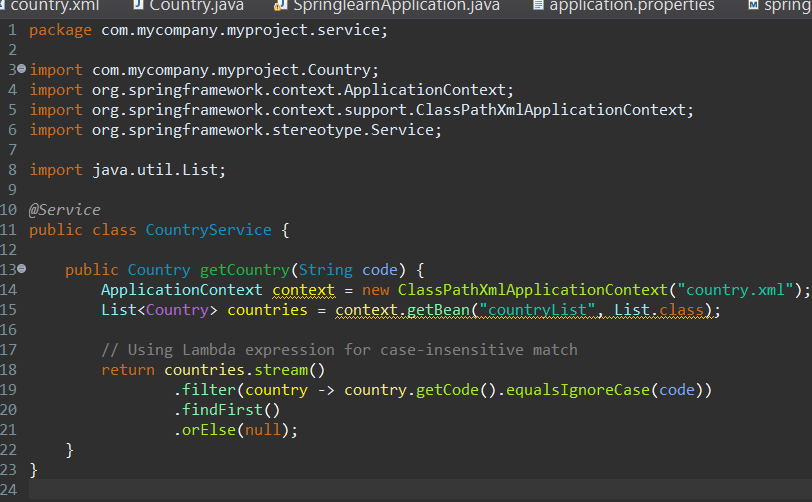
Both single country (country) and list (countryList) available.



**Create CountryService**

package com.mycompany.myproject.service;  
  
import com.mycompany.myproject.Country;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.stereotype.Service;  
  
import java.util.List;  
  
@Service  
public class CountryService {  
  
 public Country getCountry(String code) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 List<Country> countries = context.getBean("countryList", List.class);  
  
 return countries.stream()  
 .filter(c -> c.getCode().equalsIgnoreCase(code))  
 .findFirst()  
 .orElse(null);  
 }  
}

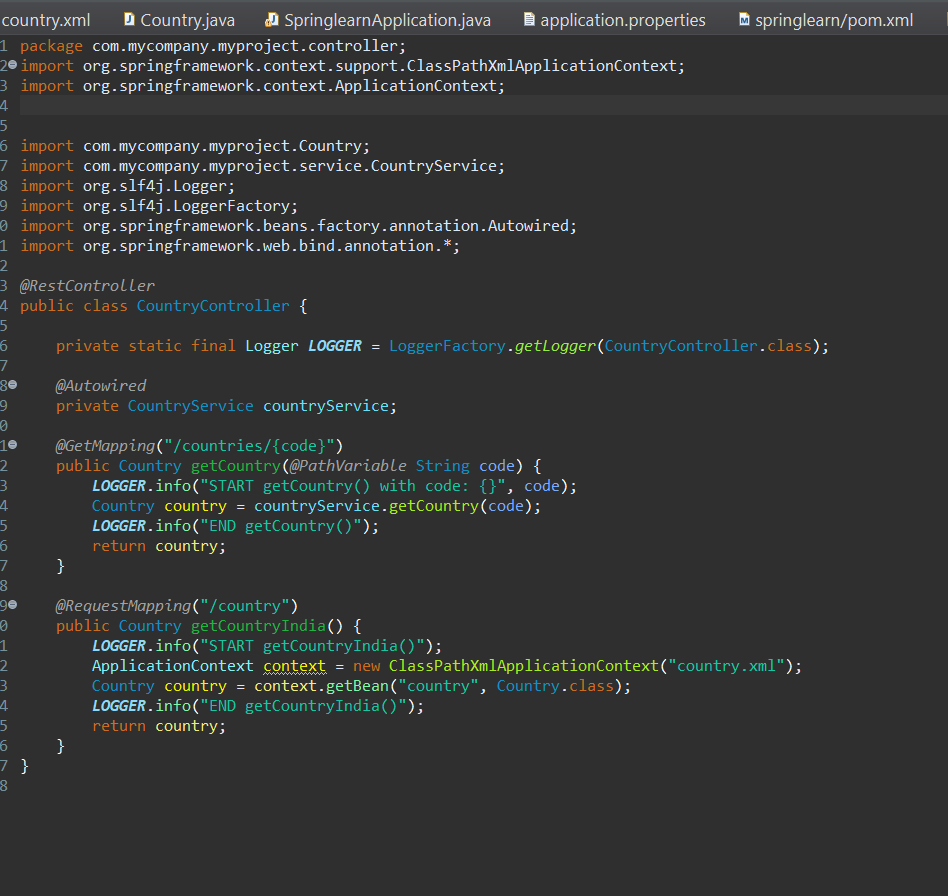
Loads country list → filters using lambda → matches case-insensitive code.



**Update CountryController**

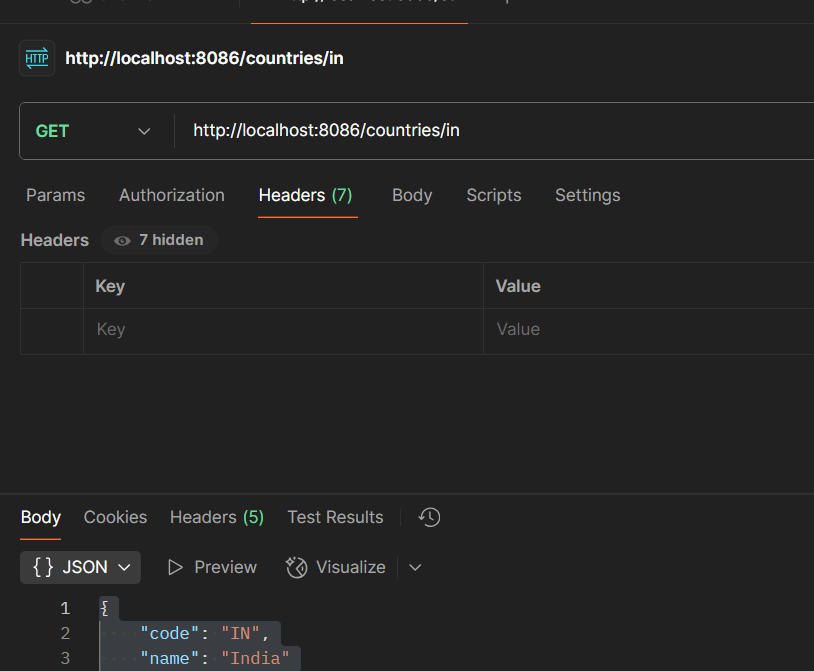
package com.mycompany.myproject.controller;  
  
import com.mycompany.myproject.Country;  
import com.mycompany.myproject.service.CountryService;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class CountryController {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);  
  
 @Autowired  
 private CountryService countryService;  
  
 @GetMapping("/countries/{code}")  
 public Country getCountry(@PathVariable String code) {  
 LOGGER.info("START getCountry() with code: {}", code);  
 Country country = countryService.getCountry(code);  
 LOGGER.info("END getCountry()");  
 return country;  
 }  
  
 @RequestMapping("/country")  
 public Country getCountryIndia() {  
 LOGGER.info("START getCountryIndia()");  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = context.getBean("country", Country.class);  
 LOGGER.info("END getCountryIndia()");  
 return country;  
 }  
}

New GET endpoint /countries/{code}.



## **Test**

* Sample Request: GET http://localhost:8086/countries/in
* Sample Response:
* {  
   "code": "IN",  
   "name": "India"  
  }
* Case-insensitive: IN, in, In, iN all work.



## SME Checkpoints

| 📌 | ✅ Explanation |
| --- | --- |
| Controller | @GetMapping extracts {code} path variable |
| Service | Loads list from XML → filters by code |
| JSON | Spring Boot auto-converts Country POJO to JSON |
| DevTools | Shows Accept: application/json + Content-Type: application/json |
| Postman | Headers tab confirms Content-Type: application/json |

Logs show START & END for method calls.