EXERCISE 5: REST-Get Country based on Country Code

**Solution:**

**SpringLearnApplication.java:**

package com.cognizant.spring\_country;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(SpringLearnApplication.class, args);  
 }  
}

**CountryController.java:**

package com.cognizant.spring\_country.controller;  
  
import com.cognizant.spring\_country.model.Country;  
import com.cognizant.spring\_country.service.CountryService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
  
@RestController  
@RequestMapping("/country")  
public class CountryController {  
  
 @Autowired  
 private CountryService countryService;  
  
 @GetMapping("/countries/{code}")  
 public ResponseEntity<Country> getCountry(@PathVariable String code) {  
 Country country = countryService.getCountry(code);  
  
 if (country != null) {  
 return new ResponseEntity<>(country, HttpStatus.*OK*);  
 } else {  
 return new ResponseEntity<>(HttpStatus.*NOT\_FOUND*);  
 }  
 }  
}

**CountryService.java:**

package com.cognizant.spring\_country.service;  
  
import com.cognizant.spring\_country.model.Country;  
import com.cognizant.spring\_country.model.CountryList;  
import org.springframework.core.io.ClassPathResource;  
import org.springframework.stereotype.Service;  
  
import javax.xml.bind.JAXBContext;  
import javax.xml.bind.JAXBException;  
import javax.xml.bind.Unmarshaller;  
import java.io.IOException;  
import java.util.List;  
import java.util.Optional;  
  
@Service  
public class CountryService {  
  
 public Country getCountry(String code) {  
 try {  
 List<Country> countries = getCountriesFromXml();  
  
 // Using lambda expression for case insensitive matching  
 Optional<Country> country = countries.stream()  
 .filter(c -> c.getCode().equalsIgnoreCase(code))  
 .findFirst();  
  
 return country.orElse(null);  
  
 } catch (Exception e) {  
 throw new RuntimeException("Error retrieving country: " + e.getMessage(), e);  
 }  
 }  
  
 private List<Country> getCountriesFromXml() throws JAXBException, IOException {  
 ClassPathResource resource = new ClassPathResource("country.xml");  
 JAXBContext jaxbContext = JAXBContext.*newInstance*(CountryList.class);  
 Unmarshaller unmarshaller = jaxbContext.createUnmarshaller();  
 CountryList countryList = (CountryList) unmarshaller.unmarshal(resource.getInputStream());  
 return countryList.getCountries();  
 }  
}

**Country.java:**

package com.cognizant.spring\_country.model;  
  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlRootElement;  
  
@XmlRootElement(name = "country")  
public class Country {  
 private String code;  
 private String name;  
  
 public Country() {}  
  
 public Country(String code, String name) {  
 this.code = code;  
 this.name = name;  
 }  
  
 @XmlElement  
 public String getCode() {  
 return code;  
 }  
  
 public void setCode(String code) {  
 this.code = code;  
 }  
  
 @XmlElement  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 @Override  
 public String toString() {  
 return "Country{code='" + code + "', name='" + name + "'}";  
 }  
}

**CountryList.java:**

package com.cognizant.spring\_country.model;  
  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlRootElement;  
import java.util.List;  
  
@XmlRootElement(name = "countries")  
public class CountryList {  
 private List<Country> countries;  
  
 @XmlElement(name = "country")  
 public List<Country> getCountries() {  
 return countries;  
 }  
  
 public void setCountries(List<Country> countries) {  
 this.countries = countries;  
 }  
}

**pom.xml:**

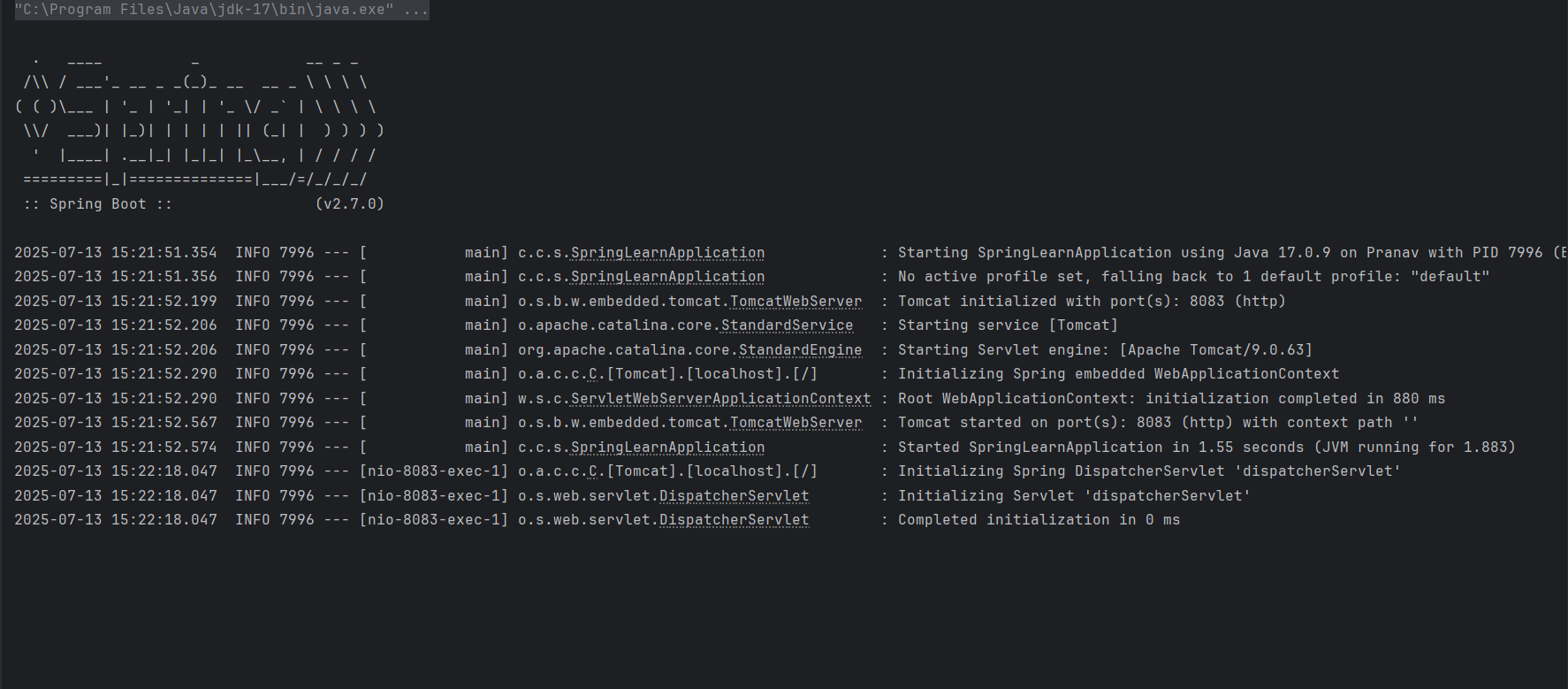
<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.7.0</version>  
 <relativePath/>  
 </parent>  
  
 <groupId>com.cognizant</groupId>  
 <artifactId>spring-learn</artifactId>  
 <version>1.0.0</version>  
 <packaging>jar</packaging>  
  
 <name>spring-learn</name>  
 <description>Spring Boot Country REST Service</description>  
  
 <properties>  
 <java.version>8</java.version>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>javax.xml.bind</groupId>  
 <artifactId>jaxb-api</artifactId>  
 <version>2.3.1</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.glassfish.jaxb</groupId>  
 <artifactId>jaxb-runtime</artifactId>  
 <version>2.3.1</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**Application.properties:**

server.port=8083  
spring.application.name=country-service

**country.xml:**

<?xml version="1.0" encoding="UTF-8"?>  
<countries>  
 <country>  
 <code>IN</code>  
 <name>India</name>  
 </country>  
 <country>  
 <code>US</code>  
 <name>United States</name>  
 </country>  
 <country>  
 <code>UK</code>  
 <name>United Kingdom</name>  
 </country>  
 <country>  
 <code>CA</code>  
 <name>Canada</name>  
 </country>  
 <country>  
 <code>AU</code>  
 <name>Australia</name>  
 </country>  
 <country>  
 <code>DE</code>  
 <name>Germany</name>  
 </country>  
 <country>  
 <code>FR</code>  
 <name>France</name>  
 </country>  
 <country>  
 <code>JP</code>  
 <name>Japan</name>  
 </country>  
 <country>  
 <code>CN</code>  
 <name>China</name>  
 </country>  
 <country>  
 <code>BR</code>  
 <name>Brazil</name>  
 </country>  
</countries>

**Output**:

A screenshot of a computer

AI-generated content may be incorrect.