**Find a country based on country code**

**SpringlearnApplication.java**

package com.cognizant.springlearn;  
  
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);  
 }  
}

**Country.java**

package com.cognizant.springlearn.model;  
  
import jakarta.persistence.Entity;  
import jakarta.persistence.Id;  
import jakarta.persistence.Table;  
  
@Entity  
@Table(name = "country")  
public class Country {  
  
 @Id  
 private String code;  
 private String name;  
  
 public String getCode() {  
 return code;  
 }  
  
 public void setCode(String code) {  
 this.code = code;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 @Override  
 public String toString() {  
 return "Country [code=" + code + ", name=" + name + "]";  
 }  
}

**CountryRepository.java**

package com.cognizant.springlearn.repository;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import com.cognizant.springlearn.model.Country;  
  
public interface CountryRepository extends JpaRepository<Country, String> {  
}

**CountryNotFoundException.java**

package com.cognizant.springlearn.service.exception;  
  
public class CountryNotFoundException extends Exception {  
  
 public CountryNotFoundException(String message) {  
 super(message);  
 }  
}

**CountryService.java**

package com.cognizant.springlearn.service;  
  
import java.util.List;  
import com.cognizant.springlearn.model.Country;  
import com.cognizant.springlearn.service.exception.CountryNotFoundException;  
  
public interface CountryService {  
  
 List<Country> getAll();  
  
 Country findCountryByCode(String code) throws CountryNotFoundException;  
  
 Country save(Country country);  
  
 void delete(String code) throws CountryNotFoundException;  
}

**CountryServiceImpl.java**

package com.cognizant.springlearn.service;  
  
import java.util.List;  
import java.util.Optional;  
  
import org.springframework.stereotype.Service;  
import org.springframework.transaction.annotation.Transactional;  
  
import com.cognizant.springlearn.model.Country;  
import com.cognizant.springlearn.repository.CountryRepository;  
import com.cognizant.springlearn.service.exception.CountryNotFoundException;  
  
@Service  
public class CountryServiceImpl implements CountryService {  
  
 private final CountryRepository countryRepository;  
  
 public CountryServiceImpl(CountryRepository countryRepository) {  
 this.countryRepository = countryRepository;  
 }  
  
 @Override  
 public List<Country> getAll() {  
 return countryRepository.findAll();  
 }  
  
 @Override  
 @Transactional  
 public Country findCountryByCode(String code) throws CountryNotFoundException {  
 Optional<Country> opt = countryRepository.findById(code);  
 if (opt.isEmpty()) {  
 throw new CountryNotFoundException("Country not found: " + code);  
 }  
 return opt.get();  
 }  
  
 @Override  
 public Country save(Country country) {  
 return countryRepository.save(country);  
 }  
  
 @Override  
 public void delete(String code) throws CountryNotFoundException {  
 if (!countryRepository.existsById(code)) {  
 throw new CountryNotFoundException("Country not found: " + code);  
 }  
 countryRepository.deleteById(code);  
 }  
}

**CountryController.java**

package com.cognizant.springlearn.controller;  
  
import java.util.List;  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
  
import com.cognizant.springlearn.model.Country;  
import com.cognizant.springlearn.service.CountryService;  
import com.cognizant.springlearn.service.exception.CountryNotFoundException;  
  
@RestController  
@RequestMapping("/countries")  
public class CountryController {  
  
 private final CountryService countryService;  
  
 public CountryController(CountryService countryService) {  
 this.countryService = countryService;  
 }  
  
 @GetMapping  
 public List<Country> getAll() {  
 return countryService.getAll();  
 }  
  
 @GetMapping("/{code}")  
 public ResponseEntity<Country> getByCode(@PathVariable String code) throws CountryNotFoundException {  
 Country country = countryService.findCountryByCode(code);  
 return ResponseEntity.ok(country);  
 }  
  
 @PostMapping  
 public ResponseEntity<Country> add(@RequestBody Country country) {  
 Country saved = countryService.save(country);  
 return new ResponseEntity<>(saved, HttpStatus.CREATED);  
 }  
  
 @DeleteMapping("/{code}")  
 public ResponseEntity<Void> delete(@PathVariable String code) throws CountryNotFoundException {  
 countryService.delete(code);  
 return ResponseEntity.noContent().build();  
 }  
}

**Configuration & SQL Files**

**application.properties**

server.port=8081  
  
spring.datasource.url=jdbc:mysql://localhost:3306/springlearn  
spring.datasource.username=root  
spring.datasource.password=root  
  
spring.jpa.hibernate.ddl-auto=none  
spring.jpa.show-sql=true  
spring.jpa.open-in-view=false

**schema.sql**

CREATE TABLE IF NOT EXISTS country (  
 code CHAR(2) PRIMARY KEY,  
 name VARCHAR(50) NOT NULL  
);

**data.sql**

INSERT INTO country (code, name) VALUES ('IN', 'India');  
INSERT INTO country (code, name) VALUES ('US', 'United States');  
INSERT INTO country (code, name) VALUES ('CN', 'China');

**pom.xml**

<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  
 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>com.cognizant</groupId>  
 <artifactId>springlearn</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>Spring Learn</name>  
 <description>Spring Boot JPA MySQL Application</description>  
 <packaging>jar</packaging>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.2.5</version>  
 <relativePath/>  
 </parent>  
  
 <properties>  
 <java.version>17</java.version>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-jpa</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>com.mysql</groupId>  
 <artifactId>mysql-connector-j</artifactId>  
 <version>8.4.0</version>  
 <scope>runtime</scope>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 </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>

**Output:**







