Name: SASHMITHA R

Superset ID: 6387562

1. **spring-data-jpa-handson**
    - **Implement services for managing Country**
    - **Find a country based on country code**
    - **Add a new country**

**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
https://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>3.5.3</version>
		<relativePath/> <!-- lookup parent from repository -->
	</parent>
	<groupId>com.example</groupId>
	<artifactId>CountryManagementSystem</artifactId>
	<version>0.0.1-SNAPSHOT</version>
	<name>CountryManagementSystem</name>
	<description>Demo project for Spring Boot</description>
	<url/>
	<licenses>
		<license/>
	</licenses>
	<developers>
		<developer/>
```

```xml
</developers>
<scm>
	<connection/>
	<developerConnection/>
	<tag/>
	<url/>
</scm>
<properties>
	<java.version>17</java.version>
</properties>
<dependencies>
	<dependency>
		<groupId>org.springframework.boot</groupId>
		<artifactId>spring-boot-starter-data-jpa</artifactId>
	</dependency>
	<dependency>
		<groupId>org.springframework.boot</groupId>
		<artifactId>spring-boot-starter-web</artifactId>
	</dependency>

	<dependency>
		<groupId>org.springframework.boot</groupId>
		<artifactId>spring-boot-devtools</artifactId>
		<scope>runtime</scope>
		<optional>true</optional>
	</dependency>
	<dependency>
		<groupId>com.mysql</groupId>
		<artifactId>mysql-connector-j</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>
```

**Country.java**

```
package com.example;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
@Entity
@Table(name = "countries")
public class Country {
   @Id
   private String code;
   private String name;
   public Country() {
   }
```

```
    public Country(String code, String name) {
        this.code = code;
        this.name = 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;
    }
}
```

**CountryController.java**

```
package com.example;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import java.util.List;
@RestController
@RequestMapping("/api/countries")
@CrossOrigin(origins = "*")
public class CountryController
{
    @Autowired
    private CountryService countryService;
    @GetMapping
    public List<Country> getAllCountries() {
        return countryService.getAllCountries();
    }
    @GetMapping("/{code}")
    public Country getCountryByCode(@PathVariable String code) {
        return countryService.getCountryByCode(code);
    }
    @PostMapping
    public Country addCountry(@RequestBody Country country) {
        return countryService.addCountry(country);
    }
    @PostMapping("/bulk")
    public List<Country> addCountries(@RequestBody List<Country> countries) {
        return countryService.addCountries(countries);
    }
    @PutMapping
    public Country updateCountry(@RequestBody Country country) {
        return countryService.updateCountry(country);
    }
    @DeleteMapping("/{code}")
    public void deleteCountry(@PathVariable String code) {
        countryService.deleteCountry(code);
    }
    @GetMapping("/search")
    public List<Country> searchByName(@RequestParam String name) {
        return countryService.searchByName(name);
    }
}
```

**CountryService.java**

```
package com.example;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.List;

import org.springframework.transaction.annotation.Transactional;

@Service

public class CountryService {

  @Autowired

  private CountryRepository countryRepository;

  public List<Country> getAllCountries() {

    return countryRepository.findAll();

  }

  public List<Country> addCountries(List<Country> countries) {

    return countryRepository.saveAll(countries);

  }

  public Country getCountryByCode(String code) {

    return countryRepository.findById(code).orElse(null);

  }

  @Transactional

  public Country addCountry(Country country) {

    return countryRepository.save(country);

  }

  @Transactional

  public Country updateCountry(Country country) {

    return countryRepository.save(country);

  }

  public void deleteCountry(String code) {

    countryRepository.deleteById(code);

  }

  public List<Country> searchByName(String partialName) {
        return countryRepository.findByNameContainingIgnoreCase(partialName);
    }
}
```

**CountryRepository.java**

```
package com.example;
import org.springframework.data.jpa.repository.JpaRepository;
import java.util.List;
public interface CountryRepository extends JpaRepository<Country, String> {
    List<Country> findByNameContainingIgnoreCase(String name);
}
```

**Application.properties**

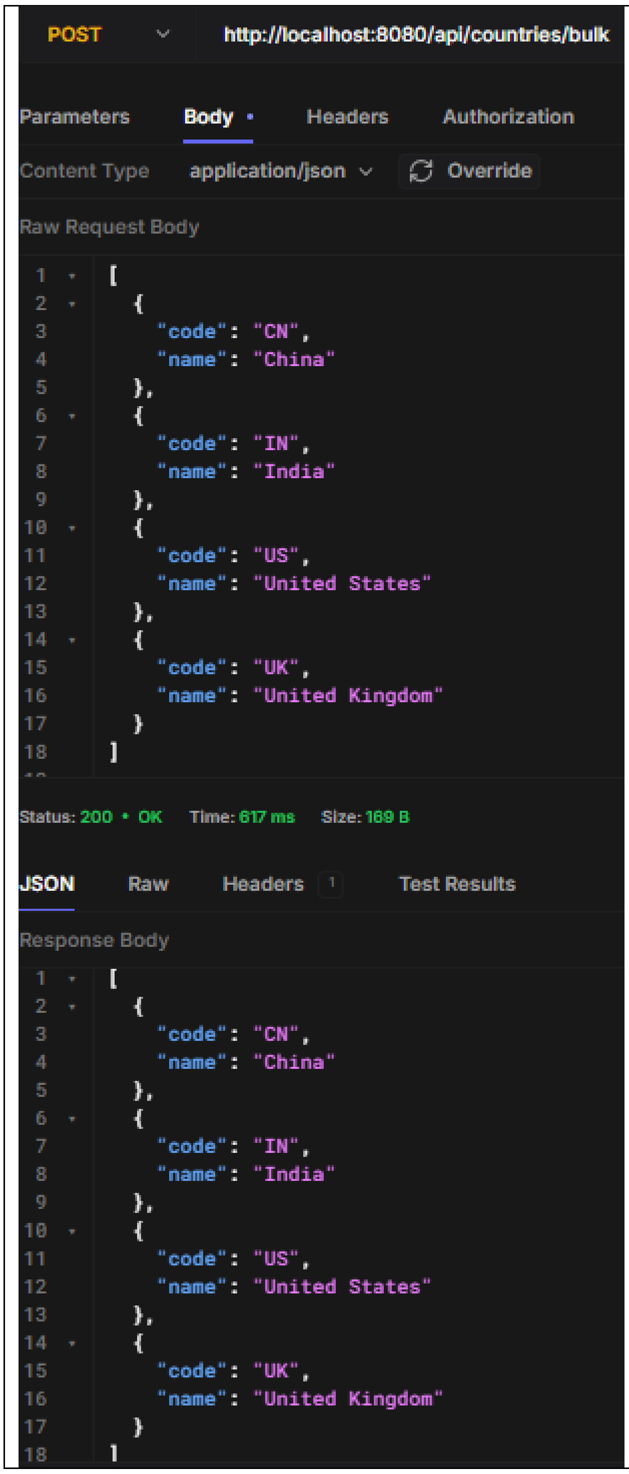
```
spring.application.name=CountryManagementSystem
spring.datasource.url=jdbc:mysql://localhost:3306/countrydb
spring.datasource.username=root
spring.datasource.password=<password>
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect

spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true
logging.level.org.hibernate.SQL=DEBUG
logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE
```

**Output :**

**Posting values**

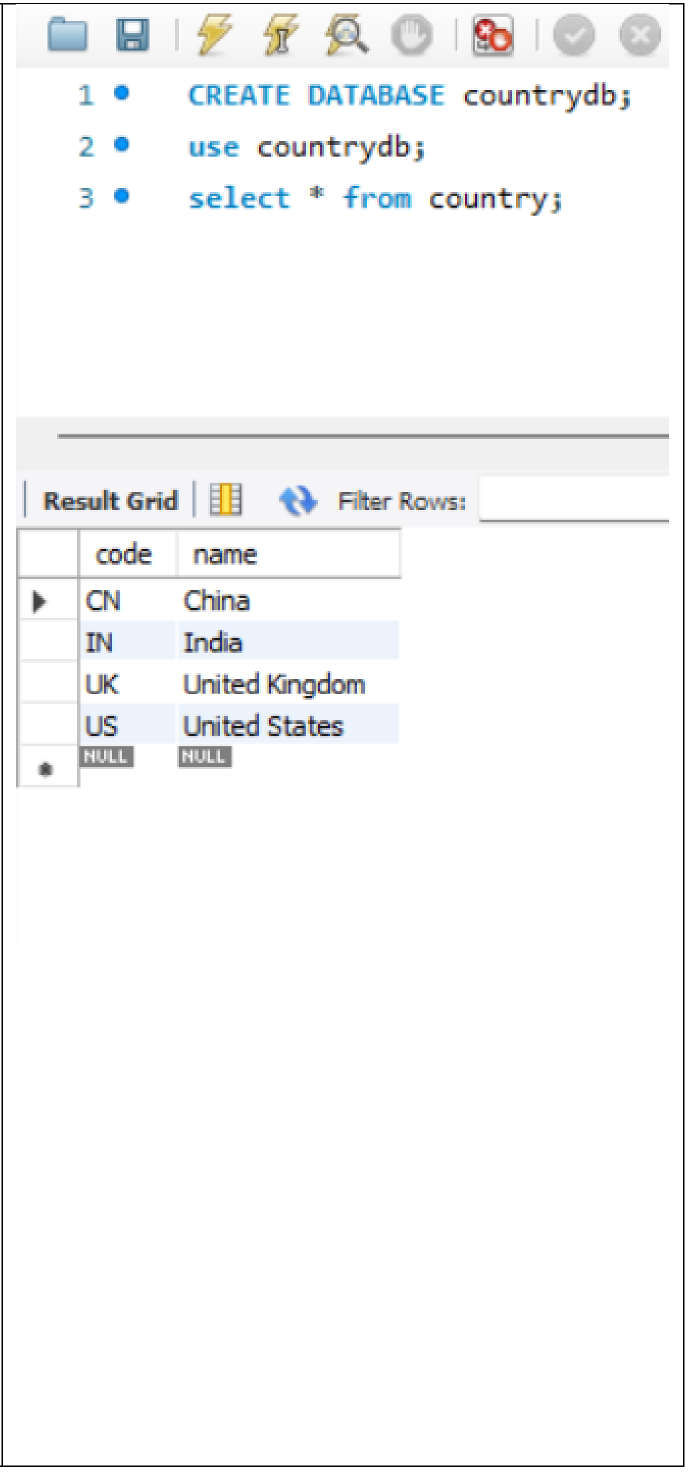


POST http://localhost:8080/api/countries/bulk

```
[
  {
    "code": "CN",
    "name": "China"
  },
  {
    "code": "IN",
    "name": "India"
  },
  {
    "code": "US",
    "name": "United States"
  },
  {
    "code": "UK",
    "name": "United Kingdom"
  }
]
```

Status: 200 • OK Time: 617 ms Size: 169 B

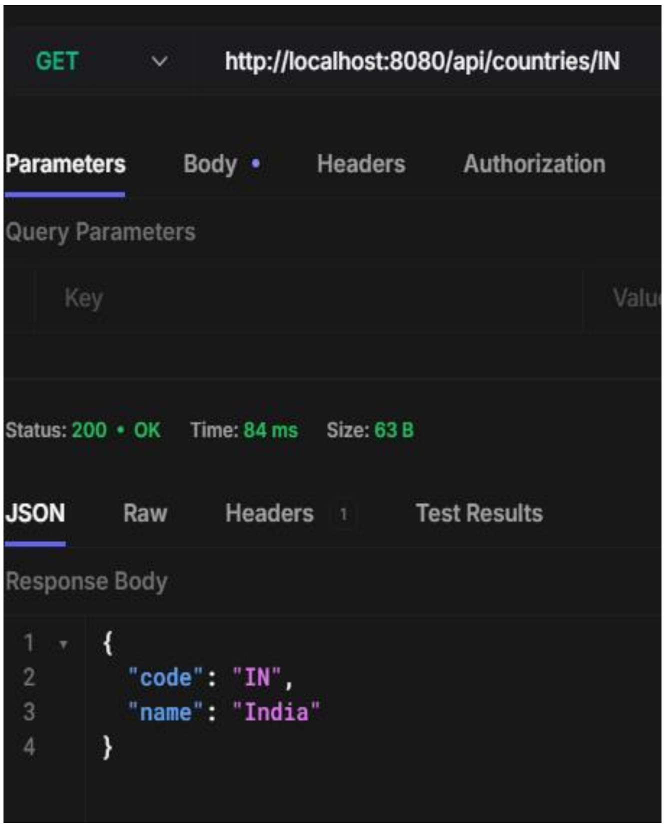
```
[
  {
    "code": "CN",
    "name": "China"
  },
  {
    "code": "IN",
    "name": "India"
  },
  {
    "code": "US",
    "name": "United States"
  },
  {
    "code": "UK",
    "name": "United Kingdom"
  }
]
```



```
CREATE DATABASE countrydb;
use countrydb;
select * from country;
```

| code | name |
|---|---|
| CN | China |
| IN | India |
| UK | United Kingdom |
| US | United States |
| NULL | NULL |

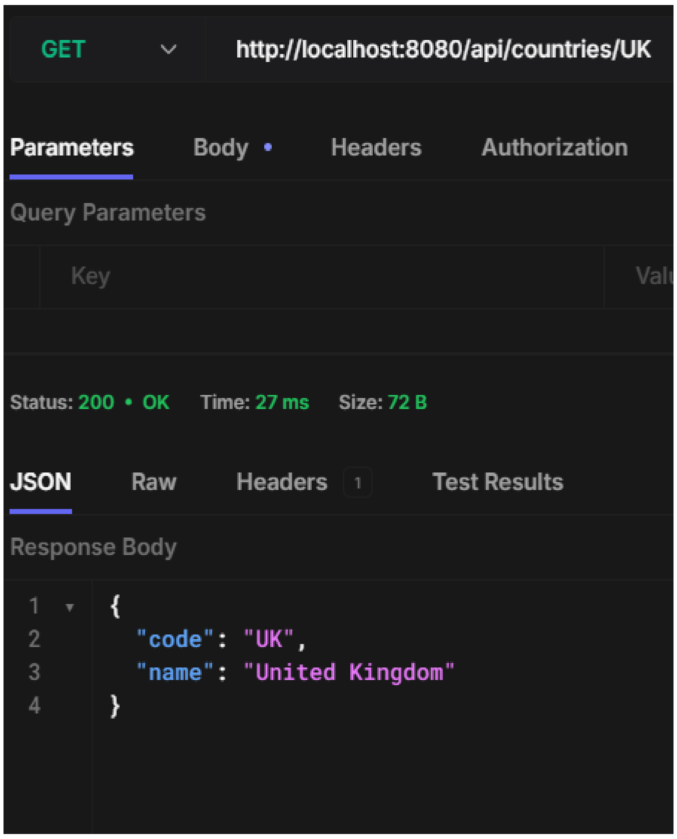
**Getting details based on country code**



GET http://localhost:8080/api/countries/IN

Status: 200 • OK Time: 84 ms Size: 63 B

```
{
  "code": "IN",
  "name": "India"
}
```

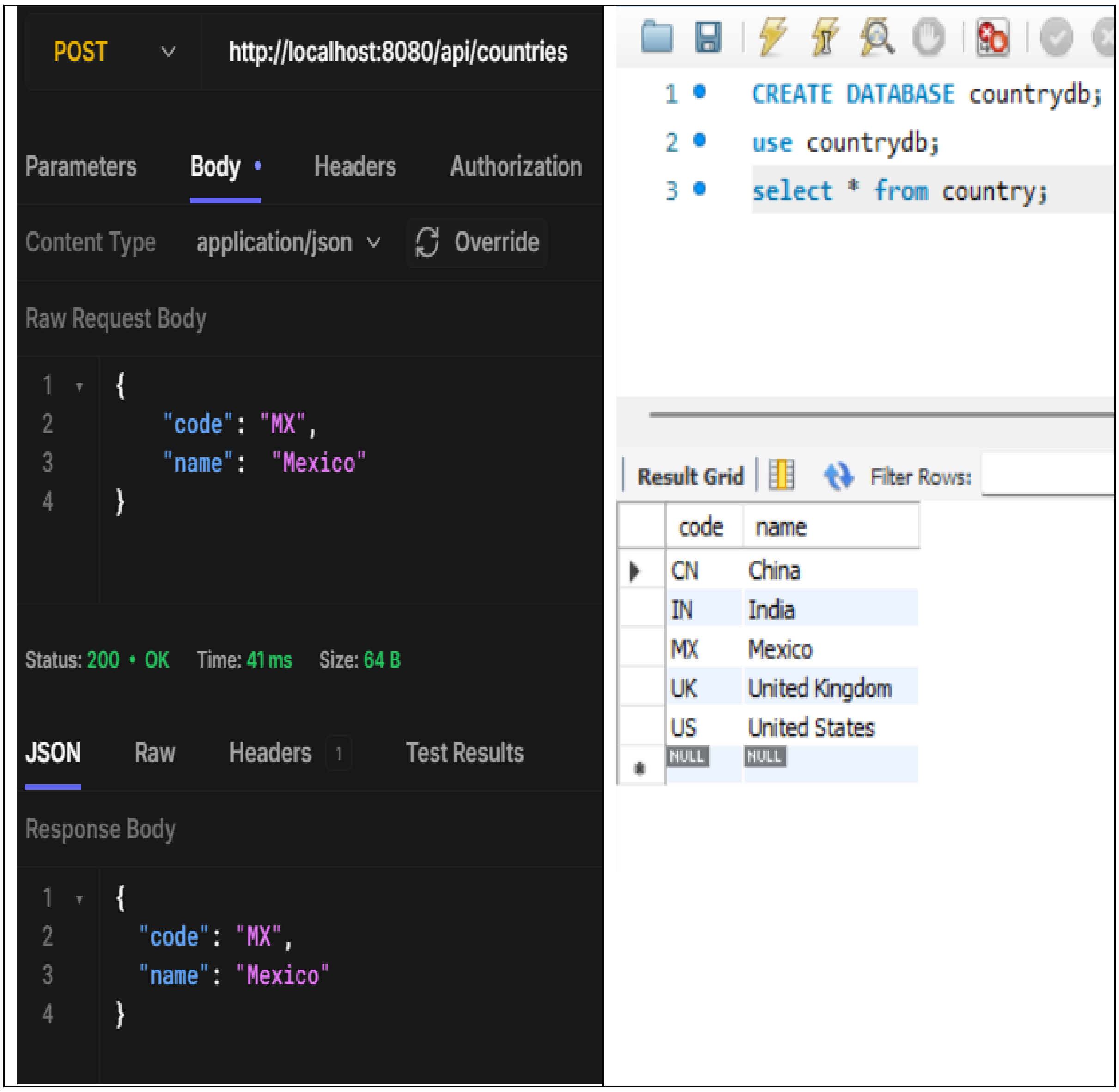


GET http://localhost:8080/api/countries/UK

Status: 200 • OK Time: 27 ms Size: 72 B

```
{
  "code": "UK",
  "name": "United Kingdom"
}
```

**Adding new values**



POST http://localhost:8080/api/countries

Parameters | Body | Headers | Authorization

Content Type: application/json | Override

Raw Request Body

```
{
    "code": "MX",
    "name":  "Mexico"
}
```

Status: 200 • OK   Time: 41 ms   Size: 64 B

JSON | Raw | Headers 1 | Test Results

Response Body

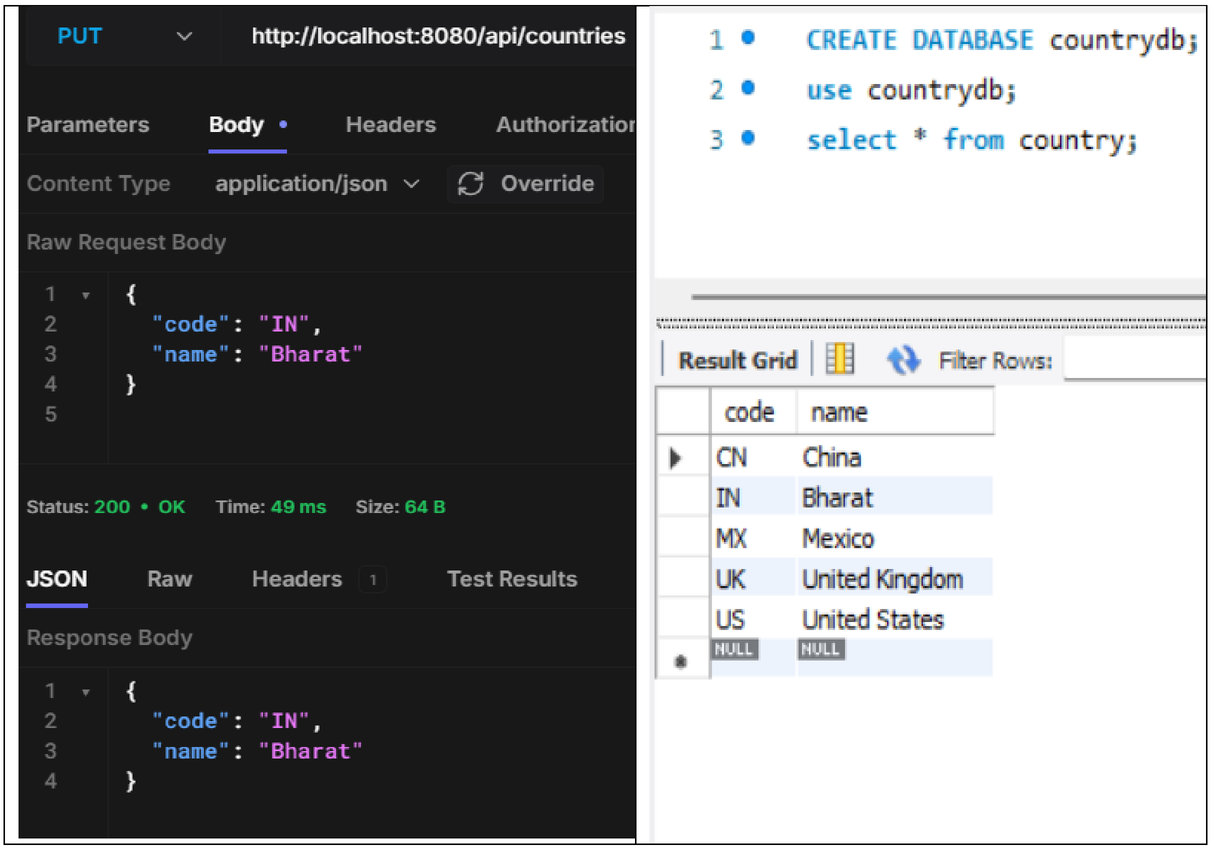
```
{
  "code": "MX",
  "name": "Mexico"
}
```

```
CREATE DATABASE countrydb;
use countrydb;
select * from country;
```

Result Grid | Filter Rows:

| code | name |
|---|---|
| CN | China |
| IN | India |
| MX | Mexico |
| UK | United Kingdom |
| US | United States |
| NULL | NULL |

**Updating the Country name**



```
http://localhost:8080/api/countries
```

Raw Request Body:

```
{
  "code": "IN",
  "name": "Bharat"
}
```

Status: 200 • OK   Time: 49 ms   Size: 64 B

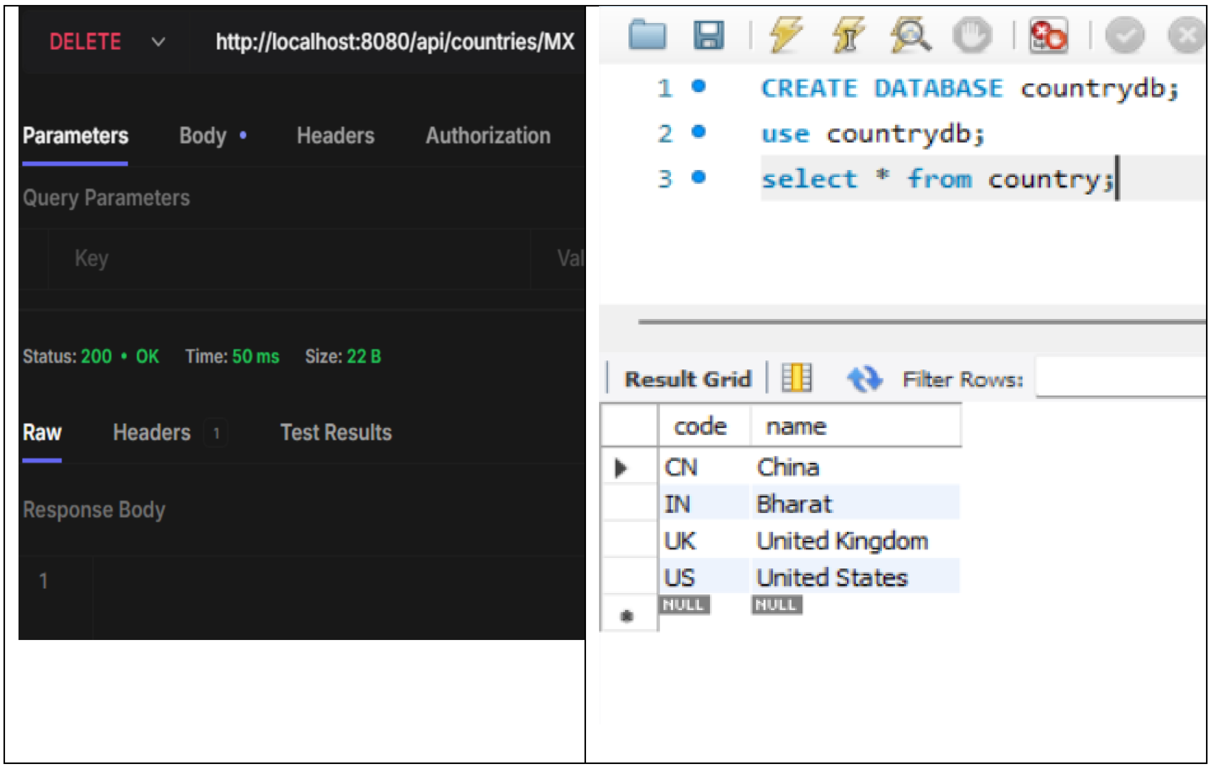
Response Body:

```
{
  "code": "IN",
  "name": "Bharat"
}
```

```
CREATE DATABASE countrydb;
use countrydb;
select * from country;
```

| code | name |
|---|---|
| CN | China |
| IN | Bharat |
| MX | Mexico |
| UK | United Kingdom |
| US | United States |
| NULL | NULL |

**Deleting Country based on code**



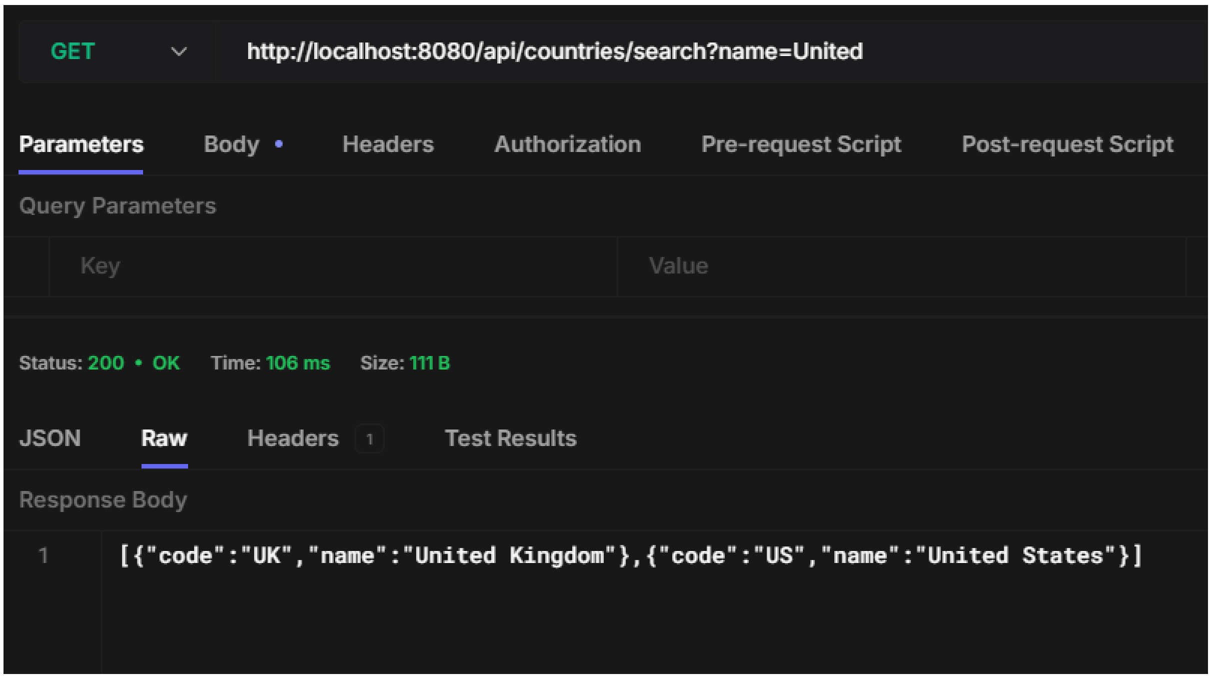
```
http://localhost:8080/api/countries/MX
```

Status: 200 • OK   Time: 50 ms   Size: 22 B

```
CREATE DATABASE countrydb;
use countrydb;
select * from country;
```

| code | name |
|---|---|
| CN | China |
| IN | Bharat |
| UK | United Kingdom |
| US | United States |
| NULL | NULL |

**Searching similar countries based on keywords**



```
GET http://localhost:8080/api/countries/search?name=United
```

Status: 200 • OK Time: 106 ms Size: 111 B

```
[{"code":"UK","name":"United Kingdom"},{"code":"US","name":"United States"}]
```