

Find a country based on country code  
Add a new country

```
package com.example.countryservice.entity;
```

```
import jakarta.persistence.*;
```

```
@Entity
```

```
public class Country {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private int id;
```

```
    @Column(unique = true)
```

```
    private String code;
```

```
    private String name;
```

```
    private String capital;
```

```
    public int getId() { return id; }
```

```
    public void setId(int id) { this.id = id; }
```

```
    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; }
```

```
    public String getCapital() { return capital; }  
    public void setCapital(String capital) { this.capital = capital; }  
}
```

```
package com.example.countryservice.repository;
```

```
import java.util.Optional;  
import com.example.countryservice.entity.Country;  
import org.springframework.data.jpa.repository.JpaRepository;
```

```
public interface CountryRepository extends JpaRepository<Country, Integer> {  
    Optional<Country> findByCode(String code);  
}
```

```
package com.example.countryservice.service;
```

```
import com.example.countryservice.entity.Country;  
import com.example.countryservice.repository.CountryRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;
```

```
import java.util.List;  
import java.util.Optional;
```

```
@Service
```

```
public class CountryService {
```

@Autowired

private CountryRepository countryRepository;

```
public Country addCountry(Country country) {  
    return countryRepository.save(country);  
}
```

```
public List<Country> getAllCountries() {  
    return countryRepository.findAll();  
}
```

```
public Optional<Country> getCountryById(int id) {  
    return countryRepository.findById(id);  
}
```

```
public Country updateCountry(Country country) {  
    return countryRepository.save(country);  
}
```

```
public void deleteCountry(int id) {  
    countryRepository.deleteById(id);  
}
```

```
public Optional<Country> getCountryByCode(String code) {  
    return countryRepository.findByCode(code);  
}
```

```
}
```

```
package com.example.countryservice.controller;
```

```
import com.example.countryservice.entity.Country;
```

```
import com.example.countryservice.service.CountryService;
```

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

```
import org.springframework.web.bind.annotation.*;
```

```
import java.util.List;
```

```
@RestController
```

```
@RequestMapping("/countries")
```

```
public class CountryController {
```

```
    @Autowired
```

```
    private CountryService countryService;
```

```
    @PostMapping
```

```
    public Country add(@RequestBody Country country) {
```

```
        return countryService.addCountry(country);
```

```
    }
```

```
    @GetMapping
```

```
    public List<Country> getAll() {
```

```
        return countryService.getAllCountries();
```

```
    }
```

```
    @GetMapping("/{id}")
```

```
    public Country getById(@PathVariable int id) {
```

```
        return countryService.getCountryById(id).orElse(null);
    }
}
```

@PutMapping

```
public Country update(@RequestBody Country country) {
    return countryService.updateCountry(country);
}
```

@DeleteMapping("/{id}")

```
public String delete(@PathVariable int id) {
    countryService.deleteCountry(id);
    return "Country with ID " + id + " deleted";
}
```

@GetMapping("/code/{code}")

```
public Country getByCode(@PathVariable String code) {
    return countryService.getCountryByCode(code).orElse(null);
}
```

```
}
```

```
package com.example.countryservice;
```

```
import org.springframework.boot.SpringApplication;
```

```
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

@SpringBootApplication

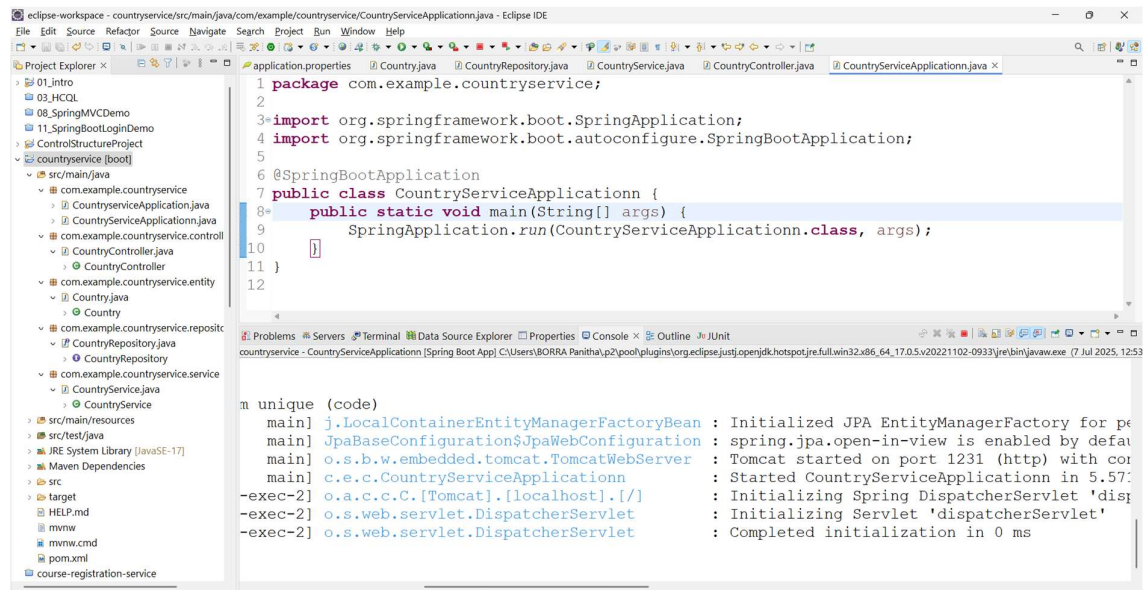
```
public class CountryServiceApplicationn {
```

```
    public static void main(String[] args) {
```

```
        SpringApplication.run(CountryServiceApplicationn.class, args);
```

```
}  
  
}
```

OUTPUT:

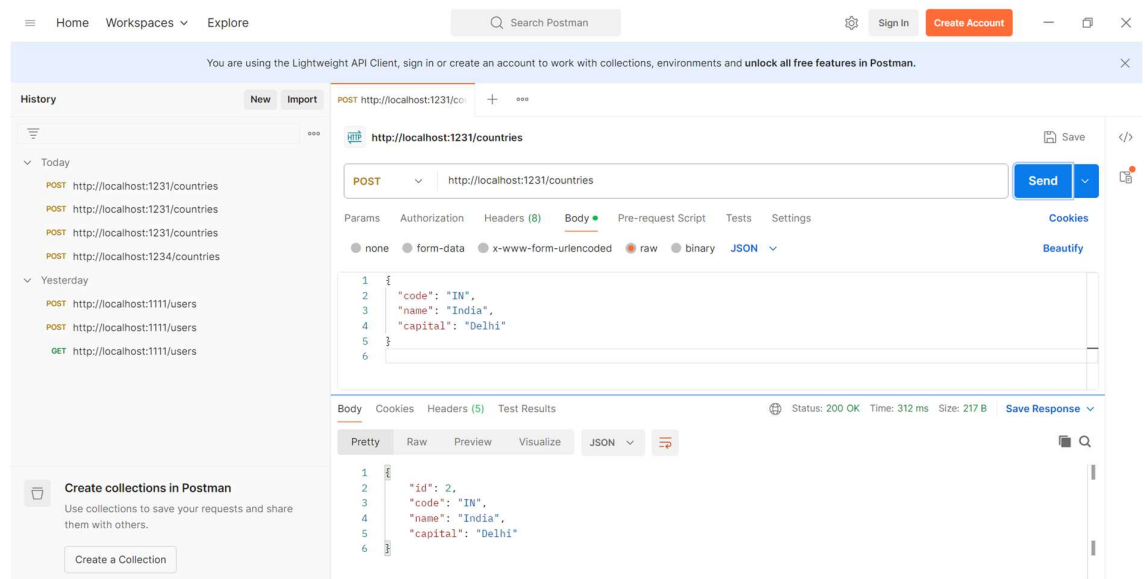


The screenshot shows the Eclipse IDE with the following components:

- Project Explorer:** Displays the project structure, including the `CountryServiceApplicationn.java` file.
- Code Editor:** Shows the source code of `CountryServiceApplicationn.java`. The code is as follows:

```
1 package com.example.countryservice;  
2  
3 import org.springframework.boot.SpringApplication;  
4 import org.springframework.boot.autoconfigure.SpringBootApplication;  
5  
6 @SpringBootApplication  
7 public class CountryServiceApplicationn {  
8     public static void main(String[] args) {  
9         SpringApplication.run(CountryServiceApplicationn.class, args);  
10    }  
11 }  
12
```
- Console:** Displays the output of the application. The output is as follows:

```
m unique (code)  
main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for po  
main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by defau  
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 1231 (http) with cor  
main] c.e.c.CountryServiceApplicationn : Started CountryServiceApplicationn in 5.57:  
-exec-2] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dis  
-exec-2] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'  
-exec-2] o.s.web.servlet.DispatcherServlet : Completed initialization in 0 ms
```



HomeWorkspacesExplore

Search Postman

Sign InCreate Account

You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman.

History

NewImport

GET http://localhost:1231/cou...+...

http://localhost:1231/countries/code/IN

Save</>

GET http://localhost:1231/countries/code/IN

POST http://localhost:1231/countries

GET http://localhost:1231/countries/code/IN

POST http://localhost:1231/countries

POST http://localhost:1231/countries

POST http://localhost:1231/countries

POST http://localhost:1234/countries

Yesterday

POST http://localhost:1111/users

POST http://localhost:1111/users

GET http://localhost:1111/users

Create collections in Postman

Use collections to save your requests and share them with others.

Create a Collection

GET http://localhost:1231/cou...

GET

http://localhost:1231/countries/code/IN

Send

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettings

Query Params

Key	Value	Bulk Edit
Key	Value	

BodyCookiesHeaders (5)Test Results

Status: 200 OKTime: 13 msSize: 217 BSave Response

PrettyRawPreviewVisualizeJSON

```
1  {
2    "id": 2,
3    "code": "IN",
4    "name": "India",
5    "capital": "Delhi"
6  }
```