**Implement services for managing Country**

**Country.java**

package com.example.demo;

import jakarta.persistence.\*;

@Entity

public class Country {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String code;

// Getters and Setters

public Long getId() { return id; }

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

public String getName() { return name; }

public void setName(String name) { this.name = name; }

public String getCode() { return code; }

public void setCode(String code) { this.code = code; }

}

**CountryController.java**

package com.example.demo;

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 addCountry(@RequestBody Country country) {

return countryService.addCountry(country);

}

@GetMapping

public List<Country> getAllCountries() {

return countryService.getAllCountries();

}

@GetMapping("/{code}")

public Country getByCode(@PathVariable String code) {

return countryService.getCountryByCode(code);

}

// Find by exact name

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

public Country getByName(@PathVariable String name) {

return countryService.getByName(name);

}

// Find all countries that contain a keyword (case-insensitive)

@GetMapping("/search")

public List<Country> search(@RequestParam String keyword) {

return countryService.searchCountriesByName(keyword);

}

// Find all countries starting with a prefix

@GetMapping("/starts")

public List<Country> startsWith(@RequestParam String prefix) {

return countryService.countriesStartingWith(prefix);

}

}

**CountryRepository.java**

package com.example.demo;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

public interface CountryRepository extends JpaRepository<Country, Long> {

Country findByName(String name);

// Find by code

Country findByCode(String code);

// Find all countries where name contains something (case-insensitive)

List<Country> findByNameContainingIgnoreCase(String keyword);

// Find all countries starting with a letter

List<Country> findByNameStartingWith(String prefix);

}

**CountryService.java**

package com.example.demo;

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

import org.springframework.stereotype.Service;

import java.util.List;

@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 Country getCountryByCode(String code) {

return countryRepository.findByCode(code);

}

public List<Country> searchCountriesByName(String keyword) {

return countryRepository.findByNameContainingIgnoreCase(keyword);

}

public List<Country> countriesStartingWith(String prefix) {

return countryRepository.findByNameStartingWith(prefix);

}

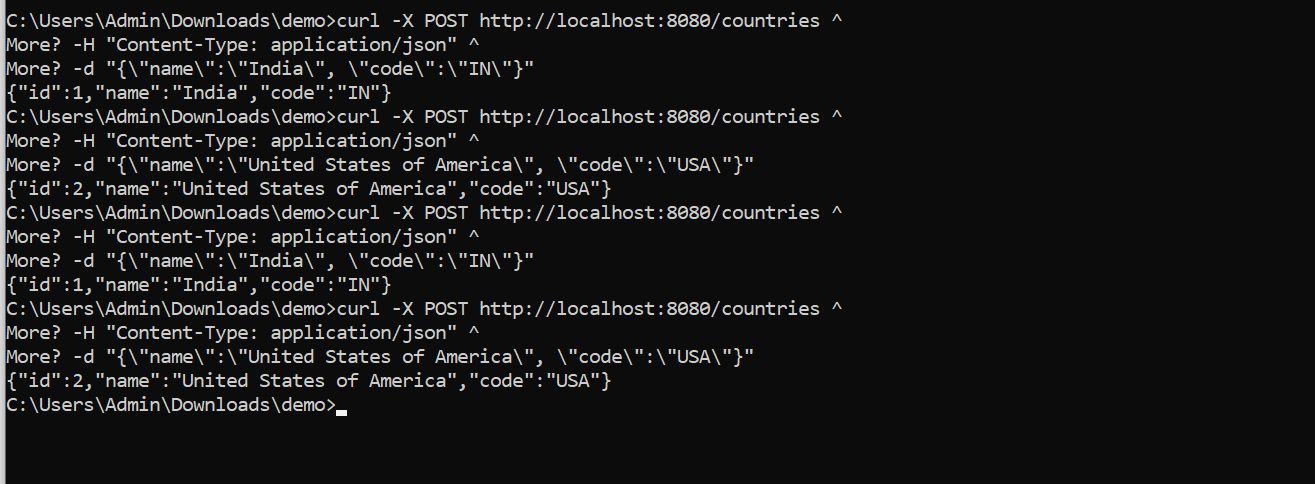
public Country getByName(String name) {

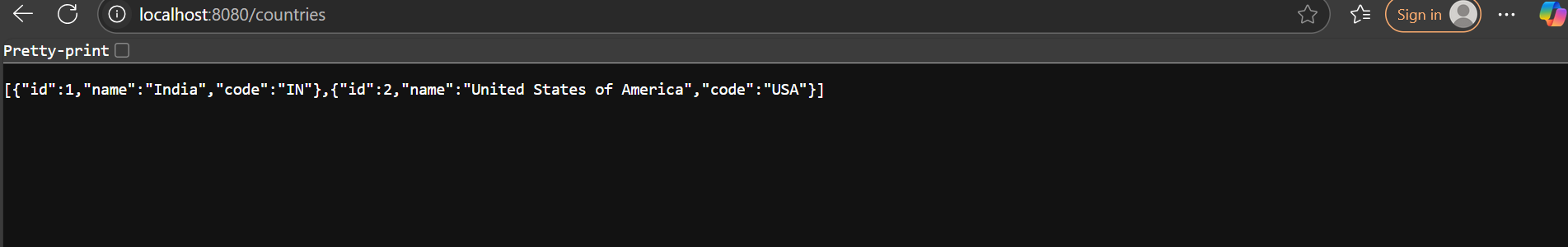
return countryRepository.findByName(name);

}

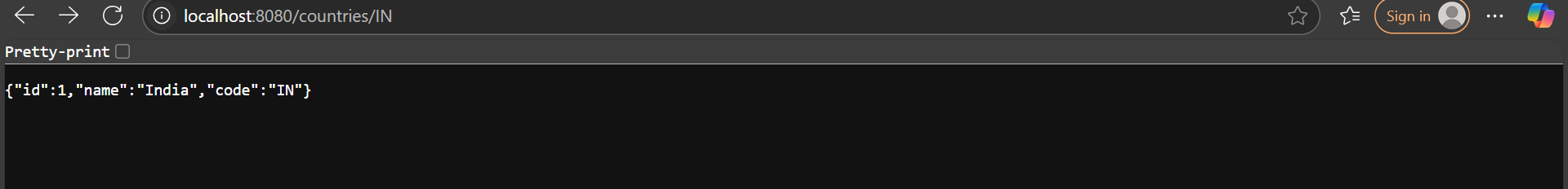
}

**Add a new country**

****

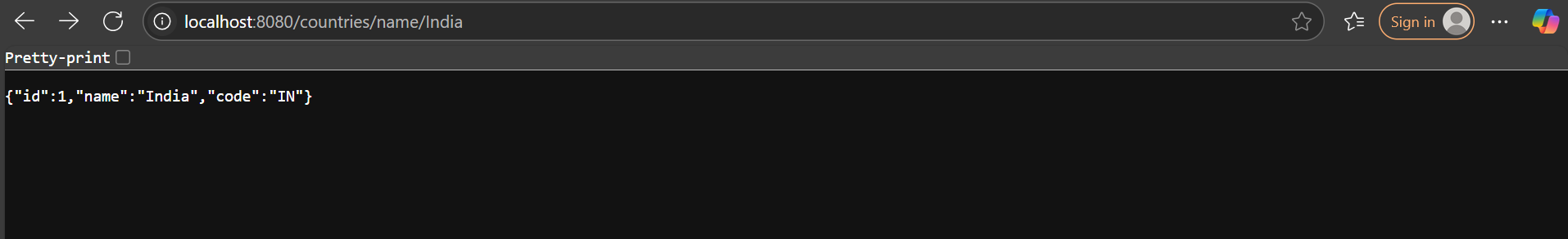
****

**Find a country based on country code**

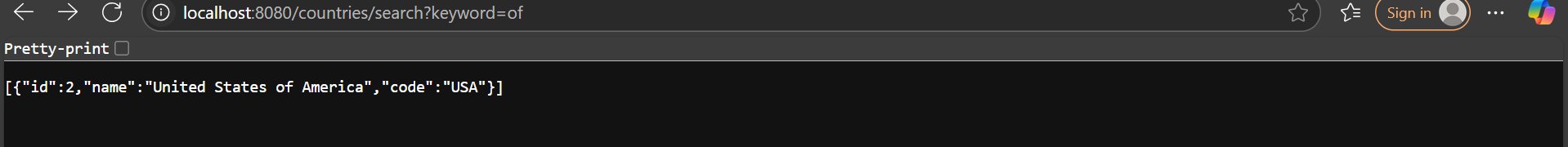
****

**Demonstrate implementation of Query Methods feature of Spring Data JPA**

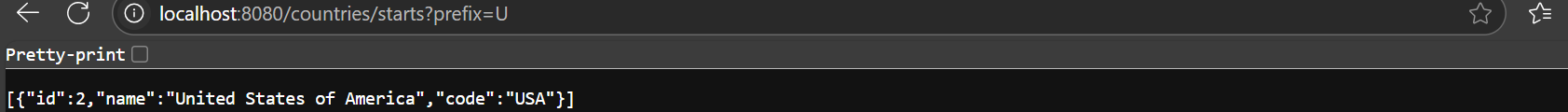
**Byname**

****

By keyword



By prefix



**Demonstrate implementation of O/R Mapping**

package com.example.demo;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

@Entity // Maps this class to a table

public class Country {

@Id // Marks this field as primary key

@GeneratedValue(strategy = GenerationType.IDENTITY) // Auto-increment

private Long id;

private String name; // Will map to column 'name'

private String code; // Will map to column 'code'

// Constructors, Getters, Setters

public Country() {

}

public Country(String name, String code) {

this.name = name;

this.code = code;

}

public Long getId() {

return id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

}