**Spring Data JPA with Spring Boot, Hibernate**

**Spring Data JPA - Quick Example**

**Code:**

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class DemoApplication {

    public static void main(String[] args) {

        SpringApplication.run(DemoApplication.class, args);

    }

}

package com.example.demo.controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class HelloController {

    @GetMapping("/")

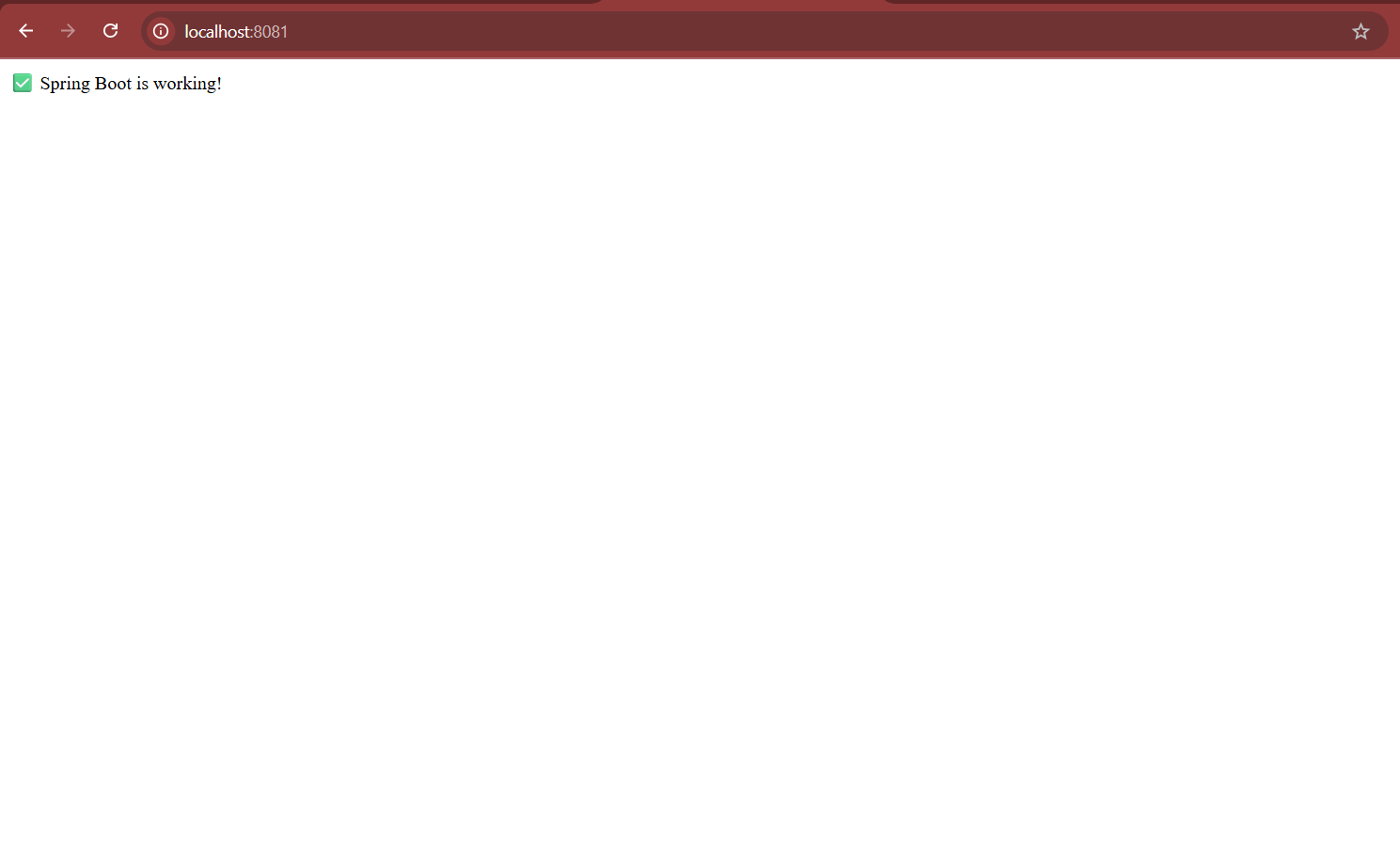
    public String home() {

        return "✅ Spring Boot is working!";

    }

}

**Output:**

****

**Difference between JPA, Hibernate and Spring Data JPA**

**JPA (Java Persistence API) – The Specification**

**What it is:**

**A standard/specification for Java ORM (Object-Relational Mapping).**

**Provides:**

**Interfaces like EntityManager, @Entity, @Table, @Id, etc.**

**Doesn't do:**

**It doesn't provide implementation — just defines what needs to be done.**

**Needs:**

**A provider (like Hibernate) to actually work.**

**Example:**

**@Entity**

**public class Country {**

**@Id**

**private Long id;**

**private String name;**

**}**

**2. Hibernate – The Implementation**

**What it is:**

**A JPA provider, meaning it's the library that actually implements JPA and provides extra features.**

**Extends JPA:**

**Has additional features like lazy/eager loading strategies, caching, custom types, etc.**

**Can be used standalone or as the engine behind JPA.**

**Example:**

**java**

**Session session = sessionFactory.openSession(); // Hibernate specific**

**3. Spring Data JPA – The Abstraction/Helper Layer**

**What it is:**

**A Spring module that builds on top of JPA and Hibernate.**

**Purpose:**

**Makes it easy to use JPA by eliminating boilerplate code.**

**Provides:**

**\* Auto-implementation of repository interfaces**

**\* Powerful query methods like findByName(), findAll()**

**\* Pagination, sorting, query derivation, etc.**

**Example:**

**java**

**public interface CountryRepository extends JpaRepository<Country, Long> {**

**List<Country> findByName(String name);**

**}**

**Implement services for managing Country**

**Code:**

package com.example.demo.entity;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

@Entity

public class Country {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Long id;

    private String name;

    private String code;

    public Country() {}

    public Country(String name, String code) {

        this.name = name;

        this.code = 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;

    }

}

package com.example.demo;

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.Bean;

import com.example.demo.entity.Country;

import com.example.demo.repository.CountryRepository;

@SpringBootApplication

public class DemoApplication {

    public static void main(String[] args) {

        SpringApplication.run(DemoApplication.class, args);

    }

    @Bean

    CommandLineRunner run(CountryRepository countryRepository) {

        return args -> {

            countryRepository.save(new Country("India", "IN"));

            countryRepository.save(new Country("Japan", "JP"));

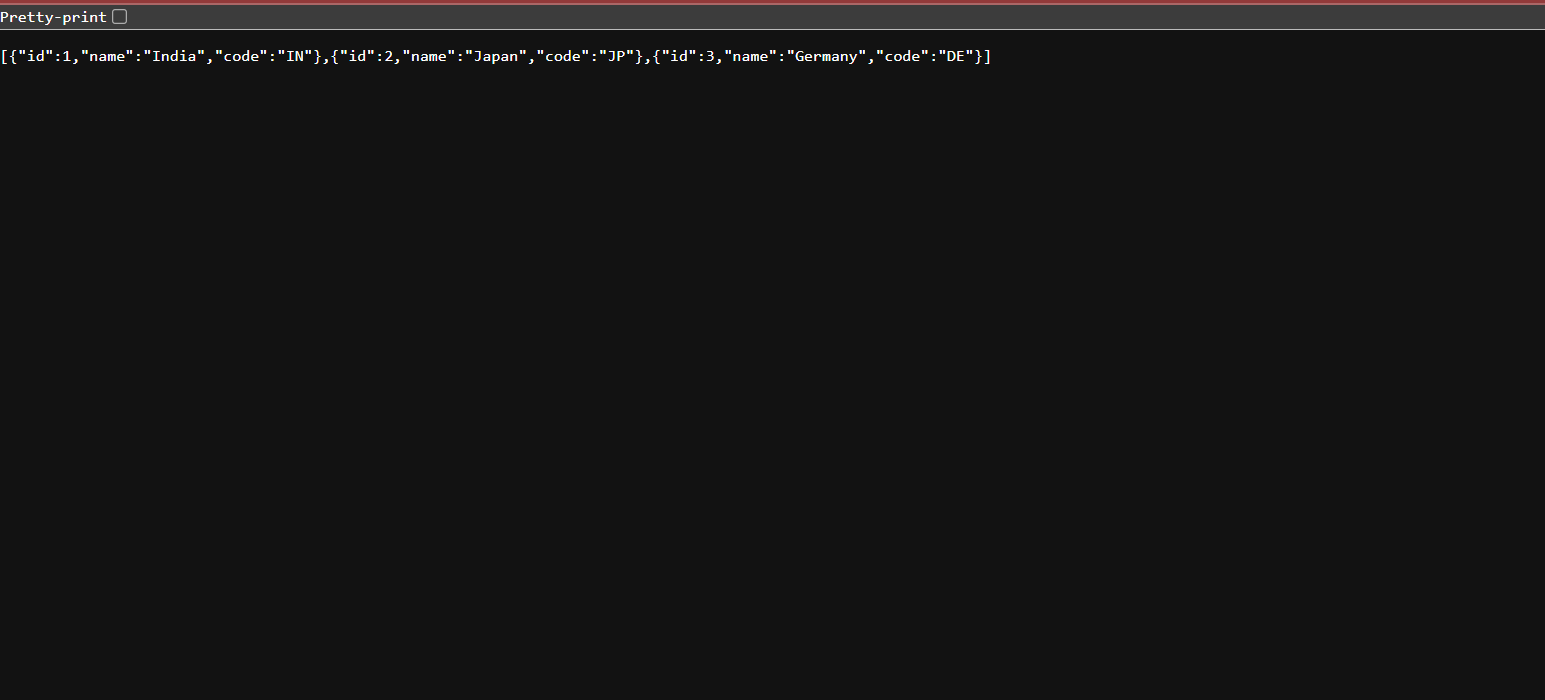
            countryRepository.save(new Country("Germany", "DE"));

        };

    }

}

**Output:**

****

**Find a country based on country code**

**Code:**

package com.example.demo.controller;

import com.example.demo.model.Country;

import com.example.demo.service.CountryService;

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

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

@RestController

@RequestMapping("/countries")

public class CountryController {

    @Autowired

    private CountryService countryService;

    @GetMapping("/{code}")

    public Country getCountry(@PathVariable String code) {

        return countryService.getCountryByCode(code)

                .orElseThrow(() -> new RuntimeException("Country not found"));

    }

}

package com.example.demo.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

@Entity

public class Country {

    @Id

    private String code; // like "IN", "US"

    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;

    }

}

package com.example.demo.repository;

import com.example.demo.model.Country;

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

import java.util.Optional;

public interface CountryRepository extends JpaRepository<Country, String> {

    Optional<Country> findByCode(String code);

}

package com.example.demo;

import com.example.demo.model.Country;

import com.example.demo.repository.CountryRepository;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class DemoApplication implements CommandLineRunner {

    @Autowired

    private CountryRepository countryRepository;

    public static void main(String[] args) {

        SpringApplication.run(DemoApplication.class, args);

    }

    @Override

    public void run(String... args) {

        Country india = new Country();

        india.setCode("IN");

        india.setName("India");

        Country usa = new Country();

        usa.setCode("US");

        usa.setName("United States");

        countryRepository.save(india);

        countryRepository.save(usa);

    }

}

package com.example.demo.service;

import com.example.demo.model.Country;

import com.example.demo.repository.CountryRepository;

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

import org.springframework.stereotype.Service;

import java.util.Optional;

@Service

public class CountryService {

    @Autowired

    private CountryRepository countryRepository;

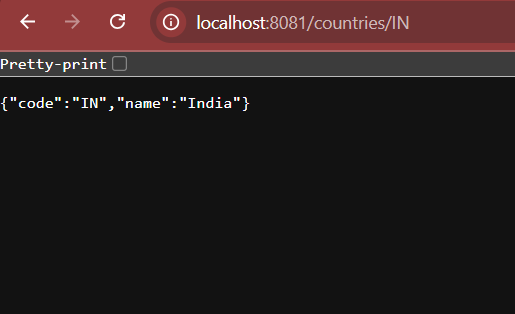
    public Optional<Country> getCountryByCode(String code) {

        return countryRepository.findByCode(code);

    }

}

**Output:**

****