**2. spring-data-jpa-handson**

**Demonstrate writing Hibernate Query Language and Native Query**

**CODE:**

controller 🡪 CountryController.java

package org.example.springdatajpahandson.controller;  
  
import org.example.springdatajpahandson.model.Country;  
import org.example.springdatajpahandson.service.CountryService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
  
@RestController  
@RequestMapping("/countries")  
public class CountryController {  
  
 @Autowired  
 private CountryService countryService;  
  
 @PostMapping  
 public ResponseEntity<Country> saveCountry(@RequestBody Country country) {  
 return ResponseEntity.*ok*(countryService.saveCountry(country));  
 }  
  
 @GetMapping("/jpql/{name}")  
 public ResponseEntity<Country> getCountryByNameJPQL(@PathVariable String name) {  
 return ResponseEntity.*ok*(countryService.getByNameJPQL(name));  
 }  
  
 @GetMapping("/native/{code}")  
 public ResponseEntity<Country> getCountryByCodeNative(@PathVariable String code) {  
 return ResponseEntity.*ok*(countryService.getByCodeNative(code));  
 }  
}

model🡪Country.java

package org.example.springdatajpahandson.model;  
  
import jakarta.persistence.\*;  
  
@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;  
 }  
  
 public Long getId() {  
 return id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public String getCode() {  
 return code;  
 }  
  
 public void setId(Long id) {  
 this.id = id;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public void setCode(String code) {  
 this.code = code;  
 }  
}

repository🡪CountryRepository.java

package org.example.springdatajpahandson.repository;  
import org.example.springdatajpahandson.model.Country;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.data.jpa.repository.Query;  
import org.springframework.data.repository.query.Param;  
public interface CountryRepository extends JpaRepository<Country, Long> {  
 @Query("SELECT c FROM Country c WHERE c.name = :name")  
 Country findByNameJPQL(@Param("name") String name);  
 @Query(value = "SELECT \* FROM country WHERE code = :code", nativeQuery = true)  
 Country findByCodeNative(@Param("code") String code);  
}

service🡪CountryService

package org.example.springdatajpahandson.service;  
  
import org.example.springdatajpahandson.model.Country;  
import org.example.springdatajpahandson.repository.CountryRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service  
public class CountryService {  
  
 @Autowired  
 private CountryRepository countryRepository;  
  
 public Country getByNameJPQL(String name) {  
 return countryRepository.findByNameJPQL(name);  
 }  
  
 public Country getByCodeNative(String code) {  
 return countryRepository.findByCodeNative(code);  
 }  
  
 public Country saveCountry(Country country) {  
 return countryRepository.save(country);  
 }  
}

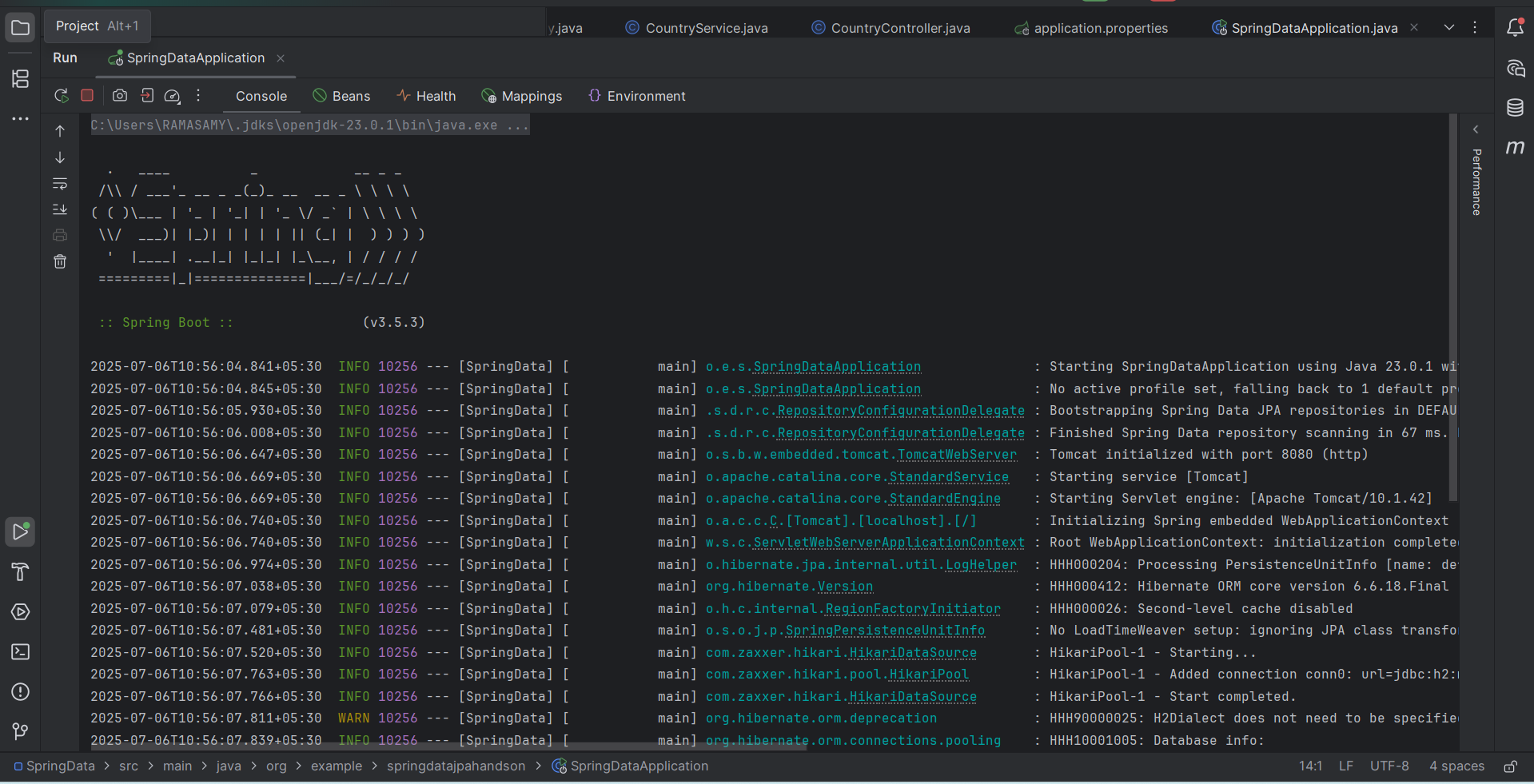
applicaton.properties

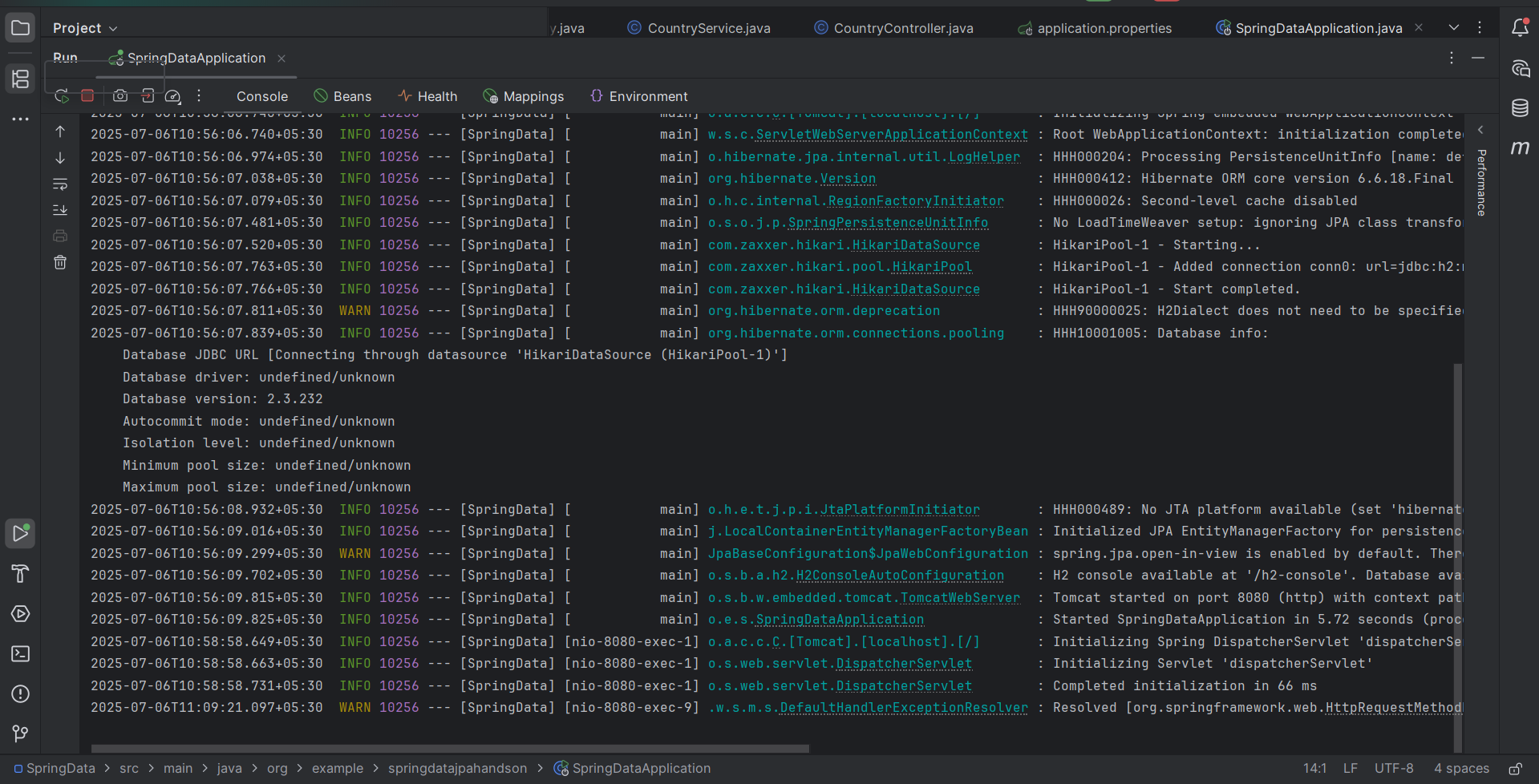
spring.application.name=SpringData  
spring.datasource.url=jdbc:h2:mem:testdb  
spring.datasource.driverClassName=org.h2.Driver  
spring.datasource.username=sa  
spring.datasource.password=  
spring.jpa.hibernate.ddl-auto=update  
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect  
spring.h2.console.enabled=true

SpringDataApplication.java

package org.example.springdatajpahandson;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringDataApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(SpringDataApplication.class, args);  
 }  
  
}

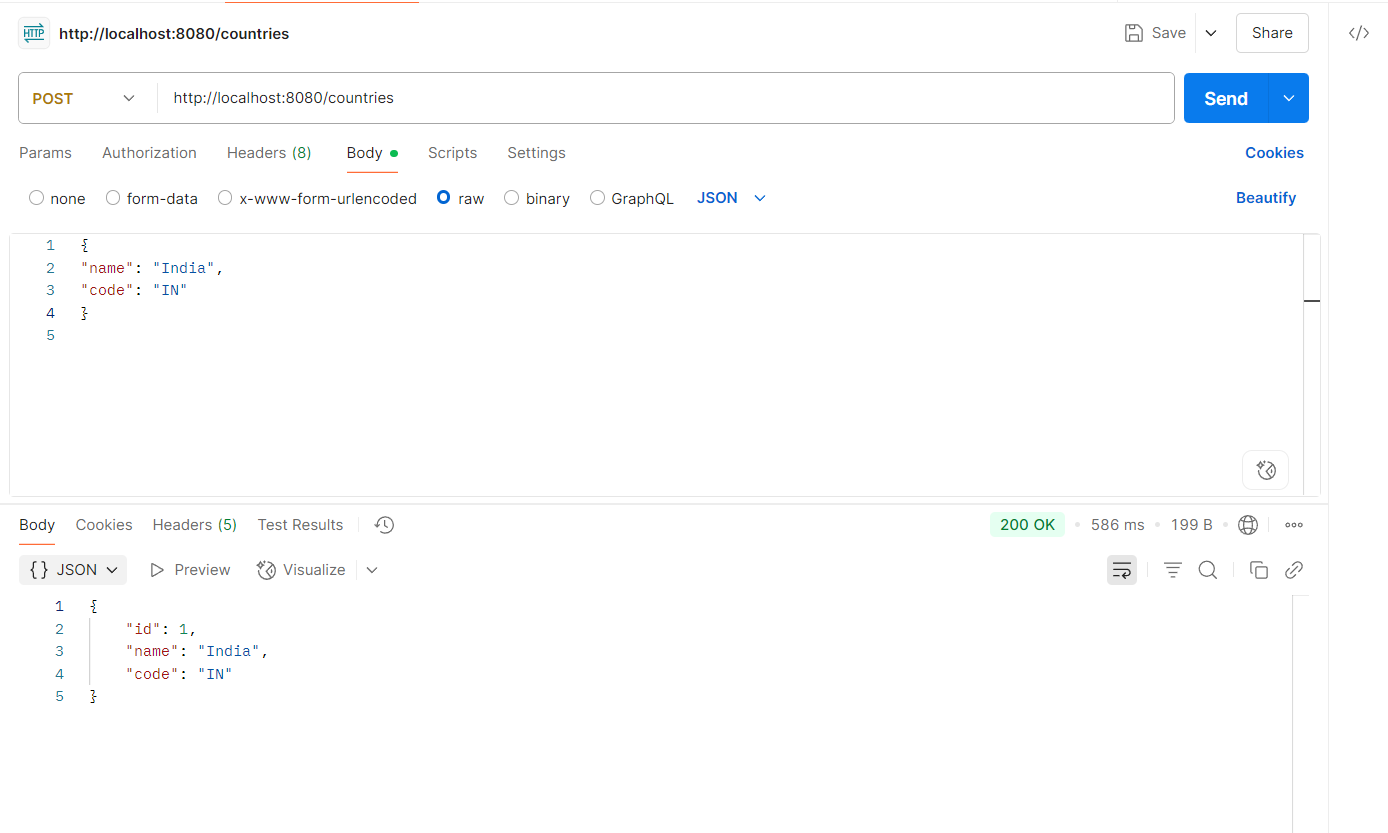
**OUTPUT**

****

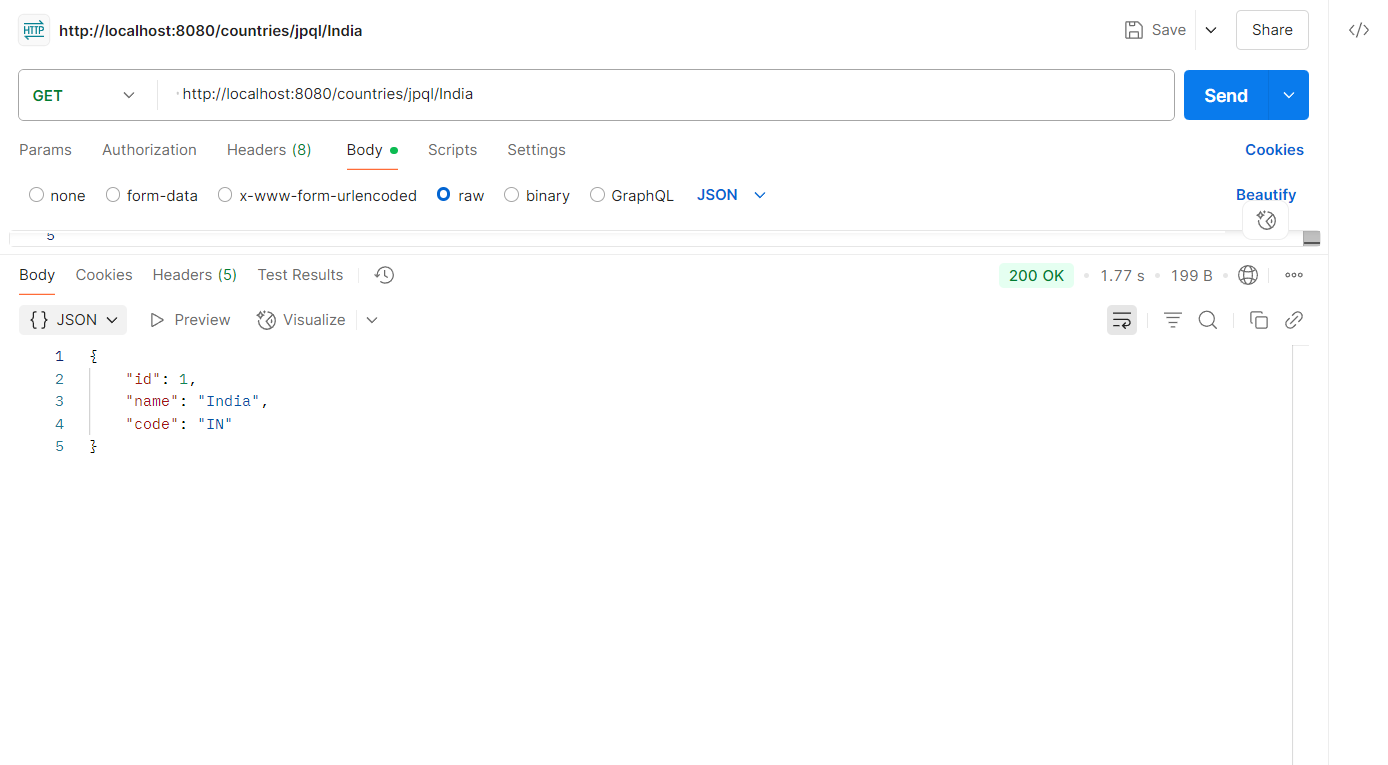
****

**POSTMAN:**

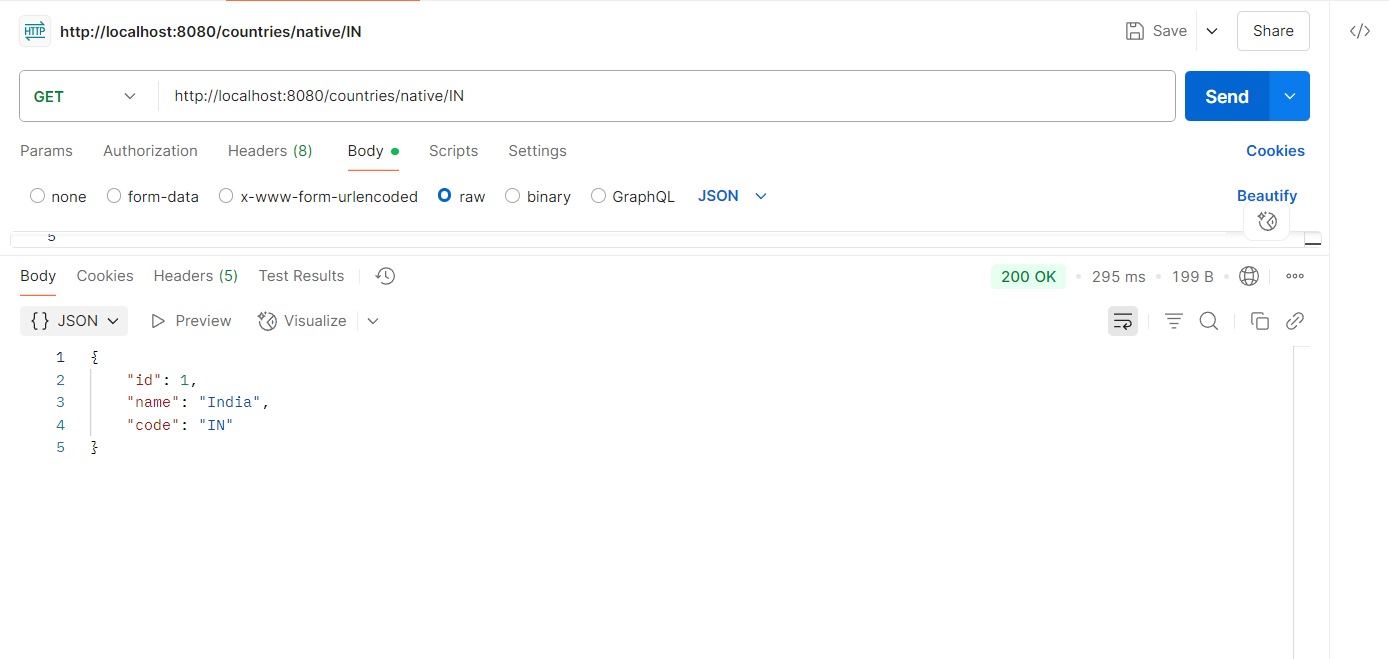
**POST**

**GET**

**Retrieve using JPQL**

****

**Retrieve using Native Query**

****