**Hands on: Demonstrate writing Hibernate Query Language and Native Query**

* + HQL stands for Hibernate Query Language, JPQL stands for Java Persistence Query Language, Compare HQL and JPQL, @Query annotation, HQL fetch keyword, aggregate functions in HQL, Native Query, nativeQuery attribute
    - Reference - https://docs.jboss.org/hibernate/orm/4.3/devguide/en-US/html/ch11.html
    - Features of JPA Query - https://www.baeldung.com/spring-data-jpa-query

**Solution code:**

**NationLearnApplication.java**

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

**Country.java**

package com.example.nationlearn.entity;  
  
import jakarta.persistence.Entity;  
import jakarta.persistence.Id;  
import jakarta.persistence.Table;  
  
@Entity  
@Table(name = "country")  
public class Country {  
  
 @Id  
 private int id;  
  
 private String name;  
 private String continent;  
  
 public Country() {}  
  
 public Country(int id, String name, String continent) {  
 this.id = id;  
 this.name = name;  
 this.continent = continent;  
 }  
  
 public int getId() { return id; }  
 public void setId(int id) { this.id = id; }  
  
 public String getName() { return name; }  
 public void setName(String name) { this.name = name; }  
  
 public String getContinent() { return continent; }  
 public void setContinent(String continent) { this.continent = continent; }  
}

**CountryRepository.java**

package com.example.nationlearn.repository;  
  
import com.example.nationlearn.entity.Country;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.data.jpa.repository.Query;  
import org.springframework.data.repository.query.Param;  
  
import java.util.List;  
  
public interface CountryRepository extends JpaRepository<Country, Integer> {  
 @Query("SELECT c FROM Country c WHERE c.continent = :continent")  
 List<Country> findByContinentHQL(@Param("continent") String continent);  
}

**CountryService.java**

package com.example.nationlearn.service;  
  
import com.example.nationlearn.entity.Country;  
import com.example.nationlearn.repository.CountryRepository;  
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 List<Country> getCountriesByContinent(String continent) {  
 return countryRepository.findByContinentHQL(continent);  
 }  
}

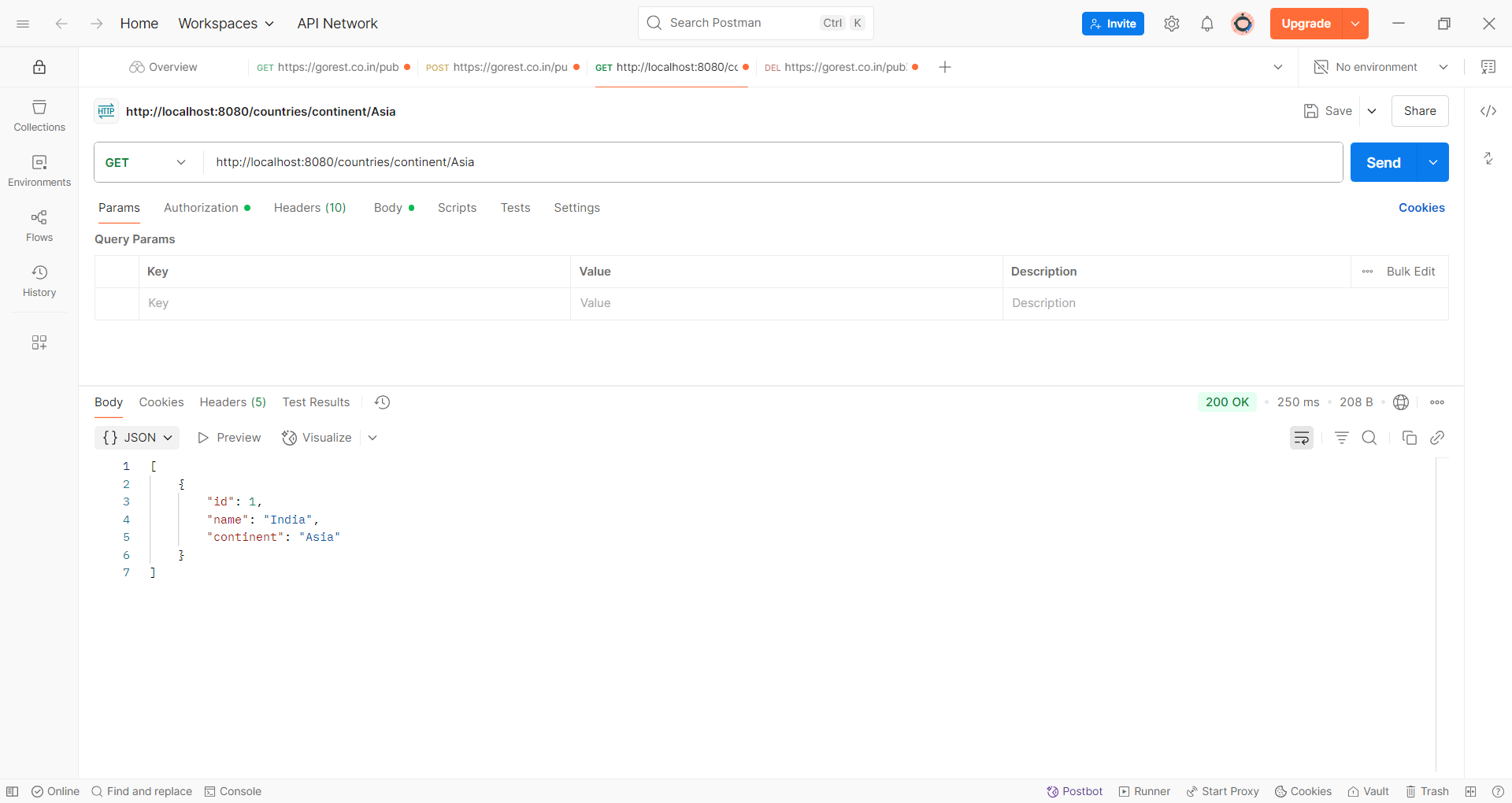
**CountryController.java**

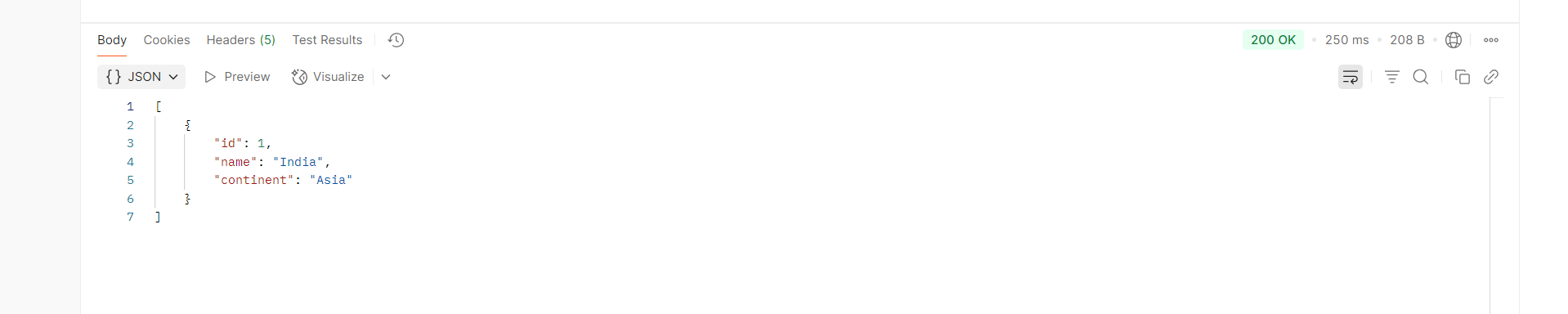
package com.example.nationlearn.controller;  
  
import com.example.nationlearn.entity.Country;  
import com.example.nationlearn.service.CountryService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
@RequestMapping("/countries")  
public class CountryController {  
  
 @Autowired  
 private CountryService countryService;  
  
 @GetMapping("/continent/{continent}")  
 public List<Country> getCountriesByContinent(@PathVariable String continent) {  
 return countryService.getCountriesByContinent(continent);  
 }  
}

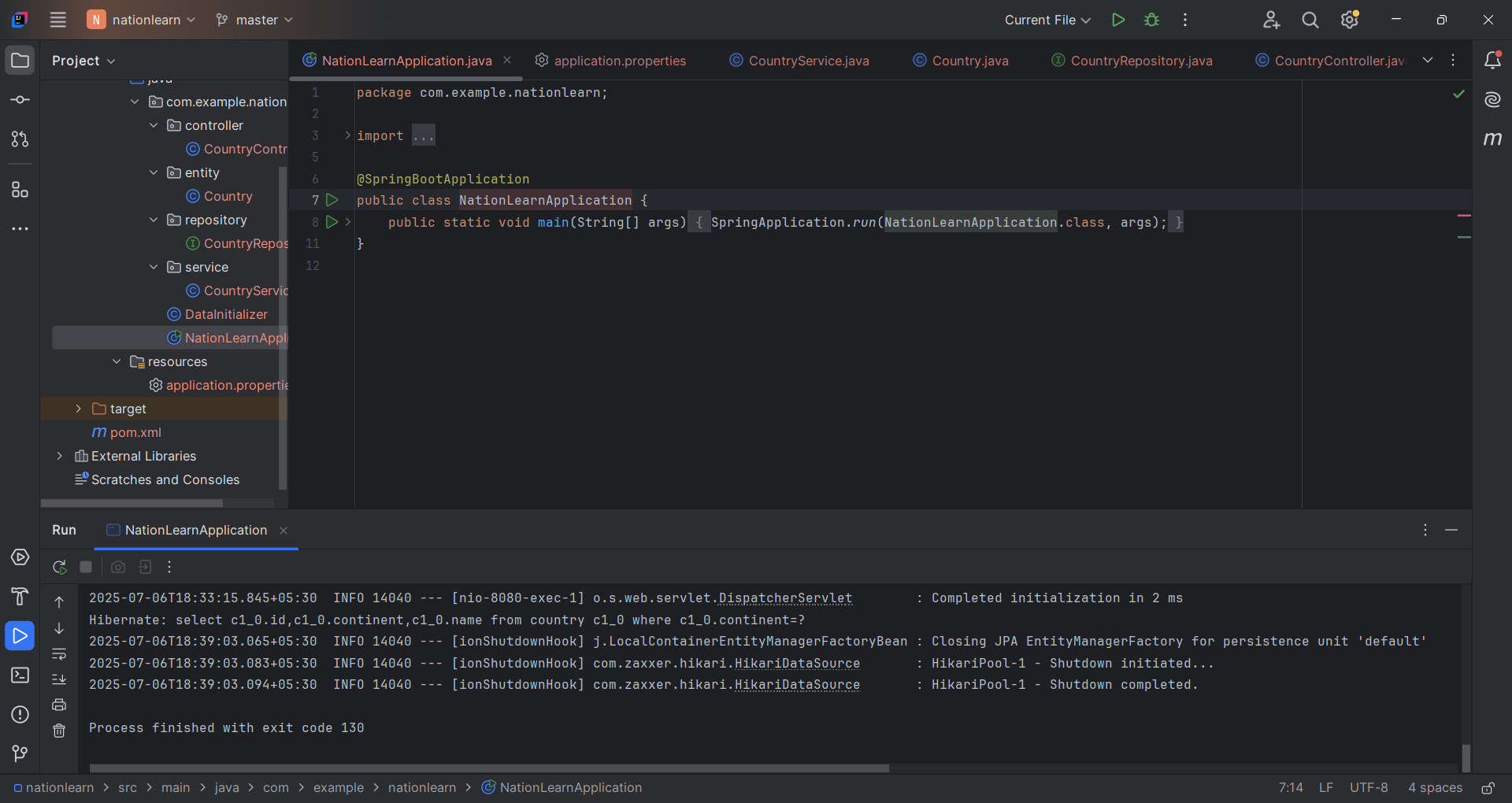
**data.sql**

INSERT INTO country (id, name, continent) VALUES (1, 'India', 'Asia');  
INSERT INTO country (id, name, continent) VALUES (2, 'Germany', 'Europe');  
INSERT INTO country (id, name, continent) VALUES (3, 'Brazil', 'South America');

**Output:**

****

****

****