**Hands on: Demonstrate implementation of Query Methods feature of Spring Data JPA**

* + Query Methods - Search by containing text, sorting, filter with starting text, fetch between dates, greater than or lesser than, top
    - Query methods - https://docs.spring.io/spring-data/jpa/docs/2.2.0.RELEASE/reference/html/#jpa.query-methods.query-creation

**Solution code:**

**Application.properties:**

# H2 DB settings  
spring.datasource.url=jdbc:h2:mem:testdb  
spring.datasource.driver-class-name=org.h2.Driver  
spring.datasource.username=sa  
spring.datasource.password=  
  
# Hibernate should create the schema from your @Entity  
spring.jpa.hibernate.ddl-auto=update  
  
# Tell Spring to always run data.sql  
spring.sql.init.mode=always  
spring.h2.console.enabled=true

**File name is Country.java:**

package com.example.world\_learn.model;  
  
import jakarta.persistence.Entity;  
import jakarta.persistence.Id;  
import jakarta.persistence.Table;  
  
@Entity  
@Table(name = "country")  
public class Country {  
  
 @Id  
 private String code;  
  
 private String name;  
  
 // Getters and setters  
 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;  
 }  
}

**file name is CountryRepository.java:**

package com.example.world\_learn.repository;  
  
import com.example.world\_learn.model.Country;  
import org.springframework.data.jpa.repository.JpaRepository;  
import java.util.List;  
  
public interface CountryRepository extends JpaRepository<Country, String> {  
 List<Country> findByNameContaining(String keyword);  
 List<Country> findByNameContainingOrderByNameAsc(String keyword);  
 List<Country> findByNameStartingWith(String prefix);  
}

**file name is CountryService.java:**

package com.example.world\_learn.service;  
  
import com.example.world\_learn.model.Country;  
import com.example.world\_learn.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> searchByNameContains(String keyword) { return countryRepository.findByNameContaining(keyword);  
 }  
  
 public List<Country> searchByNameContainsSorted(String keyword) {  
 return countryRepository.findByNameContainingOrderByNameAsc(keyword);  
 }  
  
 public List<Country> searchByStartingLetter(String letter) {  
 return countryRepository.findByNameStartingWith(letter);  
 }  
}

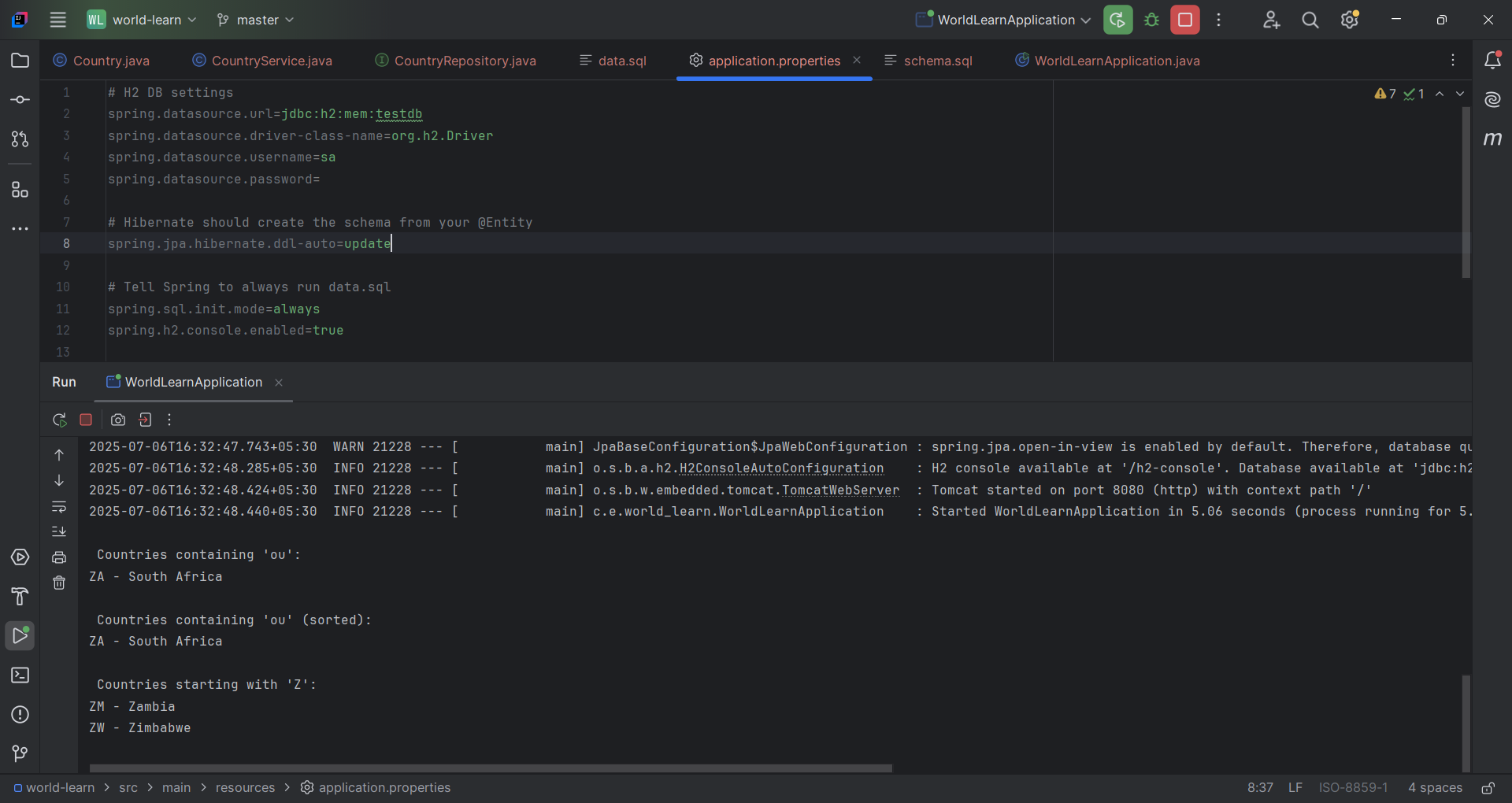
**data.sql:**

INSERT INTO country (code, name) VALUES ('IN', 'India');  
INSERT INTO country (code, name) VALUES ('US', 'United States');  
INSERT INTO country (code, name) VALUES ('ZA', 'South Africa');  
INSERT INTO country (code, name) VALUES ('FR', 'France');  
INSERT INTO country (code, name) VALUES ('CA', 'Canada');  
INSERT INTO country (code, name) VALUES ('ZM', 'Zambia');  
INSERT INTO country (code, name) VALUES ('ZW', 'Zimbabwe');

**Testing file WorldLearnApplication.java:**

package com.example.world\_learn;  
  
import com.example.world\_learn.model.Country;  
import com.example.world\_learn.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;  
import org.springframework.boot.autoconfigure.domain.EntityScan;  
import org.springframework.data.jpa.repository.config.EnableJpaRepositories;  
  
import java.util.List;  
  
  
@SpringBootApplication  
@EntityScan("com.example.world\_learn.model")  
@EnableJpaRepositories("com.example.world\_learn.repository")  
  
public class WorldLearnApplication implements CommandLineRunner {  
  
 @Autowired  
 private CountryRepository countryRepository;  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(WorldLearnApplication.class, args);  
 }  
  
 @Override  
 public void run(String... args) throws Exception {  
 testFindByNameContaining();  
 testFindByNameContainingSorted();  
 testFindByNameStartingWith();  
 }  
  
 private void testFindByNameContaining() {  
 System.*out*.println("\n Countries containing 'ou':");  
 List<Country> result = countryRepository.findByNameContaining("ou");  
 result.forEach(c -> System.*out*.println(c.getCode() + " - " + c.getName()));  
 }  
  
 private void testFindByNameContainingSorted() {  
 System.*out*.println("\n Countries containing 'ou' (sorted):");  
 List<Country> result = countryRepository.findByNameContainingOrderByNameAsc("ou");  
 result.forEach(c -> System.*out*.println(c.getCode() + " - " + c.getName()));  
 }  
  
 private void testFindByNameStartingWith() {  
 System.*out*.println("\n Countries starting with 'Z':");  
 List<Country> result = countryRepository.findByNameStartingWith("Z");  
 result.forEach(c -> System.*out*.println(c.getCode() + " - " + c.getName()));  
 }  
}

**OUTPUT:**

****