**Country\_Queries**

**Pom.xml**

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>

  <artifactId>demo</artifactId>

  <version>1.0-SNAPSHOT</version>

  <name>demo</name>

  <!-- FIXME change it to the project's website -->

  <url>http://www.example.com</url>

  <properties>

    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

    <maven.compiler.source>1.8</maven.compiler.source>

    <maven.compiler.target>1.8</maven.compiler.target>

  </properties>

  <dependencies>

    <dependency>

      <groupId>junit</groupId>

      <artifactId>junit</artifactId>

      <version>4.11</version>

      <scope>test</scope>

    </dependency>

<dependency>

  <groupId>org.springframework.boot</groupId>

  <artifactId>spring-boot-starter</artifactId>

  <version>2.7.18</version>

</dependency>

<dependency>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-logging</artifactId>

    <version>2.7.18</version>

</dependency>

<dependency>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-data-jpa</artifactId>

    <version>2.7.18</version>

</dependency>

<dependency>

    <groupId>jakarta.persistence</groupId>

    <artifactId>jakarta.persistence-api</artifactId>

    <version>2.2.3</version>

</dependency>

  </dependencies>

  <build>

    <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->

      <plugins>

        <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean\_Lifecycle -->

        <plugin>

          <artifactId>maven-clean-plugin</artifactId>

          <version>3.1.0</version>

        </plugin>

        <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_jar\_packaging -->

        <plugin>

          <artifactId>maven-resources-plugin</artifactId>

          <version>3.0.2</version>

        </plugin>

        <plugin>

          <artifactId>maven-compiler-plugin</artifactId>

          <version>3.8.0</version>

        </plugin>

        <plugin>

          <artifactId>maven-surefire-plugin</artifactId>

          <version>2.22.1</version>

        </plugin>

        <plugin>

          <artifactId>maven-jar-plugin</artifactId>

          <version>3.0.2</version>

        </plugin>

        <plugin>

          <artifactId>maven-install-plugin</artifactId>

          <version>2.5.2</version>

        </plugin>

        <plugin>

          <artifactId>maven-deploy-plugin</artifactId>

          <version>2.8.2</version>

        </plugin>

        <!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#site\_Lifecycle -->

        <plugin>

          <artifactId>maven-site-plugin</artifactId>

          <version>3.7.1</version>

        </plugin>

        <plugin>

          <artifactId>maven-project-info-reports-plugin</artifactId>

          <version>3.0.0</version>

        </plugin>

      </plugins>

    </pluginManagement>

  </build>

</project>

**/test/java/com/example/**

**AppTest.java**

package com.example;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

class ApplicationTests {

    @Test

    void contextLoads() {

    }

}

**/main/java/com/example/**

**ApplicationTest.java**

package com.example;

import java.util.List;

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

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ConfigurableApplicationContext;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import com.cognizant.spring\_learn.model.Country;

import com.cognizant.spring\_learn.repository.CountryRepository;

@SpringBootApplication

public class Application {

    private static final Logger LOGGER = LoggerFactory.getLogger(Application.class);

    @Autowired

    private static CountryRepository countryRepository;

    public static void main(String[] args) {

        ConfigurableApplicationContext context = SpringApplication.run(Application.class, args);

        countryRepository = context.getBean(CountryRepository.class);

        testFindByPartialName();

        testFindByPartialNameSorted();

        testFindByStartingAlphabet();

         System.exit(0);

    }

    private static void testFindByPartialName() {

        LOGGER.info("Searching countries containing 'ou' (unsorted):");

        List<Country> countries = countryRepository.findByNameContainingIgnoreCase("ou");

        countries.forEach(c -> LOGGER.debug("→ {}", c));

    }

    private static void testFindByPartialNameSorted() {

        LOGGER.info("Searching countries containing 'ou' (sorted):");

        List<Country> countries = countryRepository.findByNameContainingIgnoreCaseOrderByNameAsc("ou");

        countries.forEach(c -> LOGGER.debug("→ {}", c));

    }

    private static void testFindByStartingAlphabet() {

        LOGGER.info("Searching countries starting with 'Z':");

        List<Country> countries = countryRepository.findByNameStartingWithIgnoreCase("Z");

        countries.forEach(c -> LOGGER.debug("→ {}", c));

    }

}

**/respository/CountryRepository.java**

package com.example.repository;

import java.util.List;

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

import org.springframework.stereotype.Repository;

import com.example.models.Country;

@Repository

public interface CountryRepository extends JpaRepository<Country, String> {

     // Part 1 – Matching countries containing a substring (case-insensitive)

    List<Country> findByNameContainingIgnoreCase(String keyword);

    // Part 2 – Same query, but results sorted alphabetically

    List<Country> findByNameContainingIgnoreCaseOrderByNameAsc(String keyword);

    // Part 3 – Countries starting with a specific letter (case-insensitive)

    List<Country> findByNameStartingWithIgnoreCase(String prefix);

}

**/models/Country.java**

package com.example.models;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

@Entity

@Table(name="country")

public class Country {

    @Id

    @Column(name="co\_code")

    private String code;

    @Column(name="co\_name")

    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;

    }

    // Optional toString() for logging

    @Override

    public String toString() {

        return "Country [code=" + code + ", name=" + name + "]";

    }

}