Spring Native Query

Github

Running native queries to relational database using Java often leaves the source code confusing and extensive, when one has too many filter conditions and also changes in table bindings.

Because of this I decided to create the "Spring Native Query" library to facilitate the execution of native queries, with a focus on simplifying the source code, making it more readable and clean, creating files that contain the native queries and dynamically injecting assets to execute those queries.

The library's idea is to run convention queries, similar to Spring Data, and was built to work only with Spring Boot and Spring Data Jpa.

When creating a new interface that extends the NativeQuery interface, we create fake objects from these interfaces, where we use proxy to intercept method calls and execute queries, in the end we register the beans of those interfaces dynamically, so we can inject the interfaces into all the components of the Spring.

The convention works as follows, the method name is the name of the file that contains the sql query, the parameters of the methods will be passed as parameters to the entity manager, the method return is the object that will be transformed with the result returned from the query.

The file that contains the SQL query is a Jtwig template, where we can apply validations modifying the whole query, adding filters, changing links between tables, finally any changes in sql.

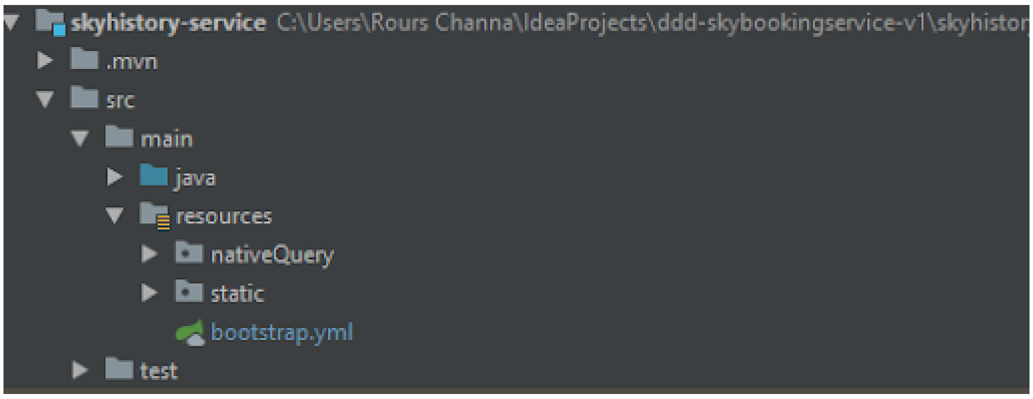
By default native query files must be added to a folder named "nativeQuery" inside the resource folder. Remember, the file name must be the same as the method name.

repository: https://github.com/gasparbarancelli/spring-native-query

Maven

```xml
<dependency>
    <groupId>io.github.gasparbarancelli</groupId>
    <artifactId>spring-native-query</artifactId>
    <version>1.0.1</version>
</dependency>
```

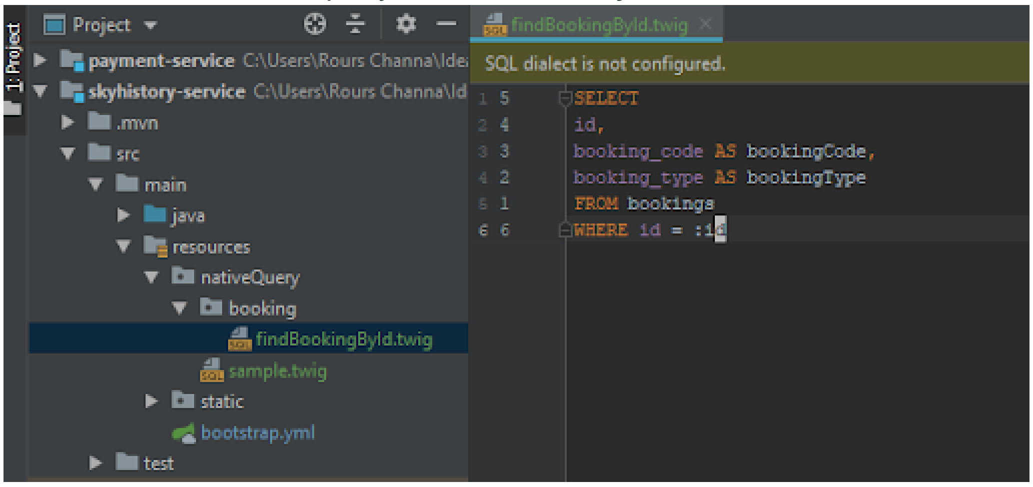
Configuration


skyhistory-service C:\Users\Rours Channa\IdeaProjects\ddd-skybookingservice-v1\skyhistor
.mvn
src
main
java
resources
nativeQuery
static
bootstrap.yml
test

1. Add configuration string to **bootstrap** File: src -> main -> resources -> bootstrap.yml

```
native-query:
  package-scan: com.skybooking.skyhistoryservice
```

2. Create folder to store query file **nativeQuery** in folder resources



SQL dialect is not configured.

```
SELECT
id,
booking_code AS bookingCode,
booking_type AS bookingType
FROM bookings
WHERE id = :id
```

File: src -> main -> resources -> nativeQuery -> booking -> findBookingByid.twig

```
SELECT
id,
booking_code AS bookingCode,
booking_type AS bookingType
FROM bookings
WHERE id = :id
```

3. Implementation code

   - Create Transfer object class to get data from query result: *BookingTO.java*

```
package com.skybooking.skyhistoryservice.v1_0_0.io.nativeQuery.booking;

public class BookingTO {
    private Integer id;
    private String bookingCode;
    private String bookingType;

    public Integer getId() {
```

```java
         return id;
        }

        public void setId(Integer id) {
         this.id = id;
        }

        public String getBookingCode() {
         return bookingCode;
        }

        public void setBookingCode(String bookingCode) {
         this.bookingCode = bookingCode;
        }

        public String getBookingType() {
         return bookingType;
        }

        public void setBookingType(String bookingType) {
         this.bookingType = bookingType;
        }
    }
```

- Create Interface by extends from **NativeQuery** interface

```java
package com.skybooking.skyhistoryservice.v1_0_0.io.nativeQuery.booking;

import io.github.gasparbarancelli.NativeQuery;
import io.github.gasparbarancelli.NativeQueryFolder;
import io.github.gasparbarancelli.NativeQueryParam;
import org.springframework.stereotype.Component;

import java.util.List;

@NativeQueryFolder("booking")
@Component
public interface BookingNQ extends NativeQuery {
    BookingTO findBookingById(@NativeQueryParam(value = "id") long id);
}
```

Note**: The **method name** must be the same the **file name**

4. Getting result from query

```java
package com.skybooking.skyhistoryservice;
import com.skybooking.skyhistoryservice.v1_0_0.io.nativeQuery.booking.BookingNQ;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.junit4.SpringRunner;
import org.springframework.transaction.annotation.Transactional;

@SpringBootTest
@RunWith(SpringRunner.class)
public class NativeQueryTest {

    @Autowired
    private BookingNQ bookingNQ;

    @Test
    @Transactional  public void getBookingById(){
        var booking = bookingNQ.findBookingById(1);
        System.out.println(booking);
    }
}
```

> Result: BookingTO{id=1, bookingCode='SBFT01000119', bookingType='flight'}