## horizontal line

Spring Data Series Part4:@Query Annotation

# Overview

We know the fact that we can delegate Query creation to The Spring data. Spring Data is enough smart to derive query based on the method name. But sometimes delegation is not fulfilled our needs. The way Spring data creates the query is not fitted our requirements so we need to customize the query. Using @Query annotation we can customize our query and instruct Spring data to take this customize query, not yours.

# When @Query annotation needed

1. Suppose you want to build a query which takes many numbers of filter criteria say I want to fetch Person based on name,country,gender, and age. We can achieve this by Spring Data Method convention like **findPersonbyNameAndCountryAndGenderAndAge** but certainly readability is pretty bad , so avoid this and use @Query annotation and create your own query with a suitable method name.

**@Query("select p from Person p where p.name like ?1 and p.country like ?2 and gender like ?3 and age=?4 order by country")**

**findPerson(String name,String country,String gender,Integer age);**

2. Suppose you want to sort your query by a specific property say name.

3. Suppose you want to override CRUDRepository **findOne** method , just redefine this method maintaining signature and put a @Query annotation with your customize query , Now Spring data takes your version

**@Query("select p from Person p where p.id = ?1 and country='America'")**

**Person findOne(Long id);**

4. You need to define a functionality which does not provide by Repository interface.

# Example

We take and Example where we create a custom method **findByCountryContains and Override** CrudRepository **findoneMethod.**

package com.example.repo;

import java.util.List;

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

import org.springframework.data.querydsl.QueryDslPredicateExecutor;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

import com.example.person.Person;

@Repository

public interface PersonRepositary extends CrudRepository<Person, Long>,QueryDslPredicateExecutor<Person> {

@Query("select p from Person p where p.country like ?1 order by country")

List<Person> findByCountryContains(String country);

List<Person> findPersonByHobbyName(String name);

@Query("select p from Person p where p.id = ?1 and country='America'")

Person findOne(Long id);

}

package com.example.person;

import java.util.ArrayList;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToMany;

@Entity

public class Person {

@Id

@GeneratedValue(strategy=GenerationType.AUTO)

private Long id;

private String name;

private String country;

private String gender;

@OneToMany(mappedBy="person",targetEntity=Hobby.class,

fetch=FetchType.EAGER,cascade=CascadeType.ALL)

List<Hobby> hobby;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCountry() {

return country;

}

public void setCountry(String country) {

this.country = country;

}

public String getGender() {

return gender;

}

public void setGender(String gender) {

this.gender = gender;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public List<Hobby> getHobby() {

return hobby;

}

public void setHobby(List<Hobby> hobby) {

this.hobby = hobby;

}

public void addHobby(Hobby ihobby)

{

if(hobby == null)

{

hobby = new ArrayList<Hobby>();

}

hobby.add(ihobby);

}

@Override

public String toString() {

return "Person [id=" + id + ", name=" + name + ", country=" + country + ", gender=" + gender + "]";

}

}

package com.example.person;

import java.util.List;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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.context.annotation.Bean;

import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

import com.example.repo.HobbyRepository;

import com.example.repo.PersonRepositary;

import com.querydsl.core.types.dsl.BooleanExpression;

@SpringBootApplication

@EnableJpaRepositories("com.example.repo")

public class PersonApplication {

@Autowired

HobbyRepository hRepo;

private static final Logger log = LoggerFactory.getLogger(PersonApplication.class);

@Bean

public CommandLineRunner demo(PersonRepositary repository) {

//insertPerson(repository);

//findPersonByHobbyname(repository,"Photography");

//findPersonByNameQueryDsl(repository,"Photography");

//findPersonByCountry(repository,"India");

Person p = repository.findOne(13L);

log.info("Person " + p);

return null;

}

private void findPersonByNameQueryDsl(PersonRepositary repository,String hobby)

{

BooleanExpression nameExpr = QPerson.person.name.contains("Shamik");

Iterable<Person> pList = repository.findAll(nameExpr);

for(Person p : pList)

log.info("Person " + p);

}

private void findPersonByHobbyname(PersonRepositary repository,String hobby)

{

List<Person> pList = repository.findPersonByHobbyName(hobby);

for(Person p : pList)

log.info("Person " + p);

}

private void findPersonByCountry(PersonRepositary repository,String country)

{

List<Person> pList = repository.findByCountryContains(country);

for(Person p : pList)

log.info("Person " + p);

}

private void insertPerson(PersonRepositary repository)

{

Person p = new Person();

p.setName("Samir mitra");

p.setCountry("America");

p.setGender("male");

Person p1 = new Person();

p1.setName("Shamik mitra");

p1.setCountry("India");

p1.setGender("male");

repository.save(p);

repository.save(p1);

Hobby hobby1 =new Hobby();

hobby1.setName("Photography");

hobby1.setPerson(p);

Hobby hobby2 =new Hobby();

hobby2.setName("Dancing");

hobby2.setPerson(p1);

Hobby hobby3 =new Hobby();

hobby3.setName("Photography");

hobby3.setPerson(p1);

hRepo.save(hobby1);

hRepo.save(hobby2);

hRepo.save(hobby3);

}

public static void main(String[] args) {

SpringApplication.run(PersonApplication.class, args);

}

}

Output:

**Hibernate:**

**select**

**person0\_.id as id1\_1\_,**

**person0\_.country as country2\_1\_,**

**person0\_.gender as gender3\_1\_,**

**person0\_.name as name4\_1\_**

**from**

**person person0\_**

**where**

**person0\_.id=?**

**and person0\_.country='America'**

**2016-10-07 12:29:39.570 INFO 5744 --- [ main] com.example.person.PersonApplication : Person Person [id=13, name=Samir mitra, country=America, gender=male]**