<http://blog.csdn.net/daniel7443/article/details/51159865>

 分类：

SpringJPA（3）

目录[(?)](http://blog.csdn.net/daniel7443/article/details/51159865" \o "系统根据文章中H1到H6标签自动生成文章目录)[[+]](http://blog.csdn.net/daniel7443/article/details/51159865)

在[使用JPA访问关系型数据库](http://tianmaying.com/tutorial/spring-boot-jpa" \t "_blank)和[Spring Data Jpa: 分页和排序](http://tianmaying.com/tutorial/spring-jpa-page-sort)两篇文章中我们学会了如何使用[**spring**](http://lib.csdn.net/base/javaee) Data Jpa 进行简单的查询以及分页等功能，Spring Data Jpa本身所支持的功能已经非常强大了，也能够支持大部分的场景。但是，现实场景永远比想象的要复杂，有时候我们确实需要像Sql语句这样更加强大以及灵活的方式来进行查询。Spring Data Jpa 当然给我们提供了这样的方式，我们可以使用@Query标注轻松的使用类似sql查询的功能。今天我们来尝试一下如何使用@Query标注来自定义查询吧。

## 基础对象

Spring Data Jpa所提供的通过方法名进行查询的功能已经能够覆盖绝大多数单表查询了。但是，我们的查询绝不仅限于单表查询，很多时候我们还是需要进行多表查询的，因此我们今天设计两个表，blog以及user，通过这两个表的联合查询来试验@Query标注。

@Entity

public class User {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Integer id;

private String username;

private String role;

......

}

@Entity

public class Blog {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Integer id;

private String title;

private String content;

@ManyToOne

private User creator;

......

}

## 关联查询

blog表和user表通过creator相关联。在user表中，我们设计了一个属性角色--role。如果我们想找到某种角色的用户所创建的blog列表应该怎么办？让我们通过@Query标注去实现它吧：

public interface BlogRepository extends PagingAndSortingRepository<Blog, Integer> {

@Query("select blog from Blog blog join blog.creator creator where creator.role = ?1")

Page<Blog> findByRole(String role, Pageable pageable);

}

在BlogRepository中，我们通过@Query标注使用了HQL进行查询，通过它，我们可以更加灵活的进行各种查询（当然，Spring Data JPA同样支持多表联查），如果不喜欢使用方法名来定义查询条件或者查询过于复杂的话，@Query标注是一个很不错的选择。

## 查询结果

我们来写一个Controller展示我们的查询结果：

@RestController

public class TestController {

@Autowired BlogRepository blogRepository;

@RequestMapping(value = "", method=RequestMethod.GET)

public Page<Blog> getEntryByPageable(@PageableDefault(value = 15, sort = { "id" }, direction = Sort.Direction.DESC)

Pageable pageable, @RequestParam(value = "role", defaultValue = "") String role) {

if("".equals(role)){

return blogRepository.findAll(pageable);

}

return blogRepository.findByRole(role, pageable);

}

}

然后，我们进入根目录，运行mvn spring-boot:run将应用run起来。所有数据会在应用启动时添加进**[数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "_blank)**当中。然后我们访问<http://localhost:8080/>，我们可以得到所有blog的分页结果：

{

"content":[

{

"id":246,

"title":"blog122",

"content":"this is teacher blog content",

"creator":{"id":1,"username":"teacher","role":"teacher"}

},

{

"id":245,

"title":"blog121",

"content":"this is teacher blog content",

"creator":{"id":1,"username":"teacher","role":"teacher"}

},

{

"id":244,

"title":"blog120",

"content":"this is teacher blog content",

"creator":{"id":1,"username":"teacher","role":"teacher"}

},

{

"id":243,

"title":"blog119",

"content":"this is teacher blog content",

"creator":{"id":1,"username":"teacher","role":"teacher"}

},

{

"id":242,

"title":"blog118",

"content":"this is teacher blog content",

"creator":{"id":1,"username":"teacher","role":"teacher"}

}

],

"last":false,

"totalElements":246,

"totalPages":50,"size":5,

"number":0,

"first":true,

"sort":[{"direction":"DESC","property":"id","ignoreCase":false,"nullHandling":"NATIVE","ascending":false}],

"numberOfElements":5

}

接着，再访问<http://localhost:8080/?role=student>，我们可以得到所有角色为student的用户所创建的blog的分页结果：

{

"content":[

{"id":123,"title":"blog122","content":"this is student blog content","creator":{"id":2,"username":"student","role":"student"}},

{"id":122,"title":"blog121","content":"this is student blog content","creator":{"id":2,"username":"student","role":"student"}},

{"id":121,"title":"blog120","content":"this is student blog content","creator":{"id":2,"username":"student","role":"student"}},

{"id":120,"title":"blog119","content":"this is student blog content","creator":{"id":2,"username":"student","role":"student"}},

{"id":119,"title":"blog118","content":"this is student blog content","creator":{"id":2,"username":"student","role":"student"}}

],

"last":false,

"totalElements":123,

"totalPages":25,

"size":5,

"number":0,

"first":true,

"sort":[{"direction":"DESC","property":"id","ignoreCase":false,"nullHandling":"NATIVE","ascending":false}],

"numberOfElements":5}

版权声明本文由[Cliff](http://www.tianmaying.com/user/xierui)创作，转载需署名作者且注明文章出处