spring data 接口之 PagingAndSortingRepository

PagingAndSortingRepository 接口继承于 CrudRepository 接口，拥有CrudRepository 接口的所有方法， 并新增两个方法：分页和排序。 但是这两个方法不能包含筛选条件。

**[java]** [view plain](http://blog.csdn.net/zgf19930504/article/details/50537225) [copy](http://blog.csdn.net/zgf19930504/article/details/50537225)

1. Iterable<T> findAll(Sort sort); //查询所有,排序，不进行分页
2. ;T> findAll(Pageable pageable);

**一 声明继承PagingAndSortingRepository接口的接口：IStudentdPageSortRepository**

**[java]** [view plain](http://blog.csdn.net/zgf19930504/article/details/50537225) [copy](http://blog.csdn.net/zgf19930504/article/details/50537225)

1. **package** org.zgf.spring.data.dao;
3. **import** org.springframework.data.repository.PagingAndSortingRepository;
4. **import** org.zgf.spring.data.entity.StudentPO;
5. /\*\*
6. \* 继承PagingAndSortingRepository 接口
7. \* @author: zonggf
8. \* @date: 2016年1月15日-下午12:58:20
9. \*/
10. **public** **interface** IStudentdPageSortRepository **extends** PagingAndSortingRepository<StudentPO, Integer>{
12. /\*\*  PagingAndSortingRepository 接口继承于 CrudRepository 接口，拥有CrudRepository 接口的所有方法
13. \*   特有方法有两个 分页，和 排序， 这两个方法的缺陷是不能有筛选条件
14. \*/
15. //Iterable<T> findAll(Sort sort); //查询所有,排序，不进行分页
16. //Page<T> findAll(Pageable pageable);
17. }

**二 测试类**

**[java]** [view plain](http://blog.csdn.net/zgf19930504/article/details/50537225) [copy](http://blog.csdn.net/zgf19930504/article/details/50537225)

1. **package** org.zgf.spring.data.dao;
3. **import** java.util.ArrayList;
4. **import** java.util.List;
6. **import** javax.naming.directory.DirContext;
8. **import** org.junit.Test;
9. **import** org.springframework.beans.factory.annotation.Autowired;
10. **import** org.springframework.data.domain.Page;
11. **import** org.springframework.data.domain.PageRequest;
12. **import** org.springframework.data.domain.Sort;
13. **import** org.springframework.data.domain.Sort.Direction;
14. **import** org.springframework.data.domain.Sort.Order;
15. **import** org.zgf.spring.data.base.BaseTest;
16. **import** org.zgf.spring.data.dao.IStudentdPageSortRepository;
17. **import** org.zgf.spring.data.entity.StudentPO;
19. /\*\*
20. \* @ClassName: Test\_IStudentdPageSortRepository
21. \* @Description: 测试分页排序
22. \* @author: zonggf
23. \* @date: 2016年1月15日-下午1:01:46
24. \*/
25. **public** **class** Test\_IStudentdPageSortRepository **extends** BaseTest{
27. @Autowired
28. **private** IStudentdPageSortRepository studentdPageSortRepository;

31. /\*\*
32. \* 查询所有记录，指定排序方式,不能自定义查询条件
33. \* Hibernate: select studentpo0\_.id as id1\_0\_, studentpo0\_.age as age2\_0\_, studentpo0\_.birth as birth3\_0\_, studentpo0\_.create\_time as create\_t4\_0\_, studentpo0\_.name as name5\_0\_, studentpo0\_.sex as sex6\_0\_ from spj\_student\_2 studentpo0\_ order by studentpo0\_.id desc, studentpo0\_.name desc
34. \*/
35. @Test
36. **public** **void** test\_findAll\_Sort(){
37. Order idOrder = **new** Order(Direction.DESC, "id");
38. Order nameOrder = **new** Order(Direction.DESC,"name");
40. Sort sort = **new** Sort(idOrder, nameOrder);
41. List<StudentPO> students = (List<StudentPO>) **this**.studentdPageSortRepository.findAll(sort);
42. **for** (StudentPO studentPO : students) {
43. System.out.println(studentPO);
44. }
45. }

48. /\*\*
49. \* 此种排序方式不能指定筛选条件
50. \* 排序会执行两条sql 语句
51. \* Hibernate: select count(\*) as col\_0\_0\_ from spj\_student\_2 studentpo0\_
52. \* Hibernate: select studentpo0\_.id as id1\_0\_, studentpo0\_.age as age2\_0\_, studentpo0\_.birth as birth3\_0\_, studentpo0\_.create\_time as create\_t4\_0\_, studentpo0\_.name as name5\_0\_, studentpo0\_.sex as sex6\_0\_ from spj\_student\_2 studentpo0\_ order by studentpo0\_.id desc, studentpo0\_.name asc limit ?
53. \*/
54. @Test
55. **public** **void** test\_findAll\_page(){
56. **int** currentPage =0; //当前页从0 开始
57. **int** pageSize = 5;
59. //排序
60. Order idOrder = **new** Order(Direction.DESC, "id");
61. Order nameOrder = **new** Order(Direction.ASC,"name");
62. Sort sort = **new** Sort(idOrder,nameOrder);
63. PageRequest pageRequest  = **new** PageRequest(currentPage, pageSize, sort);
65. Page<StudentPO> page = **this**.studentdPageSortRepository.findAll(pageRequest);
66. System.out.println("总记录数:" + page.getTotalElements());
67. System.out.println("总页数:" + page.getTotalPages());
68. System.out.println("当前页（request):" + page.getNumber());
69. System.out.println("当前页总记录数（request):" + page.getSize());
70. System.out.println("当前页记录总数：" + page.getNumberOfElements());
71. List<StudentPO> students = page.getContent();
72. **for** (StudentPO studentPO : students) {
73. System.out.println(studentPO);
74. }
75. }
77. }