

查询过滤器用法

目录索引:

- 功能描述
- 查询规则
 - 1. 全匹配查询
 - 2. 模糊查询
 - 3. 范围查询
 - 4. 包含查询
 - 5. 等等。。。
- 查询过滤器如何集成
- 更多查询规则参考

查询过滤器

一、功能描述

查询过滤器可以帮助快速生成查询条件，不需要编码通过配置实现，支持模糊查询、匹配查询、范围查询、不匹配查询等规则。

二、查询规则

****说明:** **页面查询字段，需跟后台Controller中Page的字段对应一致，后台不需写代码自动生成查询条件SQL;

默认查询条件是全匹配，想实现模糊查询需求在查询值的前后加: *;

查询匹配方式规则:

[1].全匹配查询: 查询数据没有特殊格式，默认为全匹配查询

[2].模糊查询: 查询数据格式需加星号: { * }

例如:

格式一: 张*

格式二: *张

格式三: *张*

格式四: *张*三*

(后模糊匹配)

(前模糊匹配)

(全模糊匹配)

(更高级匹配)

[3].包含查询: 查询数据格式采用逗号分隔: { , }

例如:

格式: 张三,李四

(含义: In('张三','李四'))

[4].不匹配查询: 查询数据格式需要加叹号前缀: { ! }

例如:

格式: !张三

(含义: 不等于'张三')

特殊说明: 查询不为Null的语法: !null(大小写没关系);

查询不为空字符串的方法: !(只有一个叹号);

[5].范围查询, 支持数字, 时间的范围查询, 针对范围查询页面会生成两个查询控件

1. 如果是单一匹配方式, 则页面查询控件的name, 跟实体字段命名一样

2. 如果是范围匹配方式, 则页面查询控件需要变成两个分别名 {*_}_begin, {*_}_end

{*_}_begin: 表示查询范围开始值

{*_}_end: 表示查询范围结束值

举例：
字段名称 orderDate
查询开始时间 ： orderDate_begin
查询结束时间 ： orderDate_end

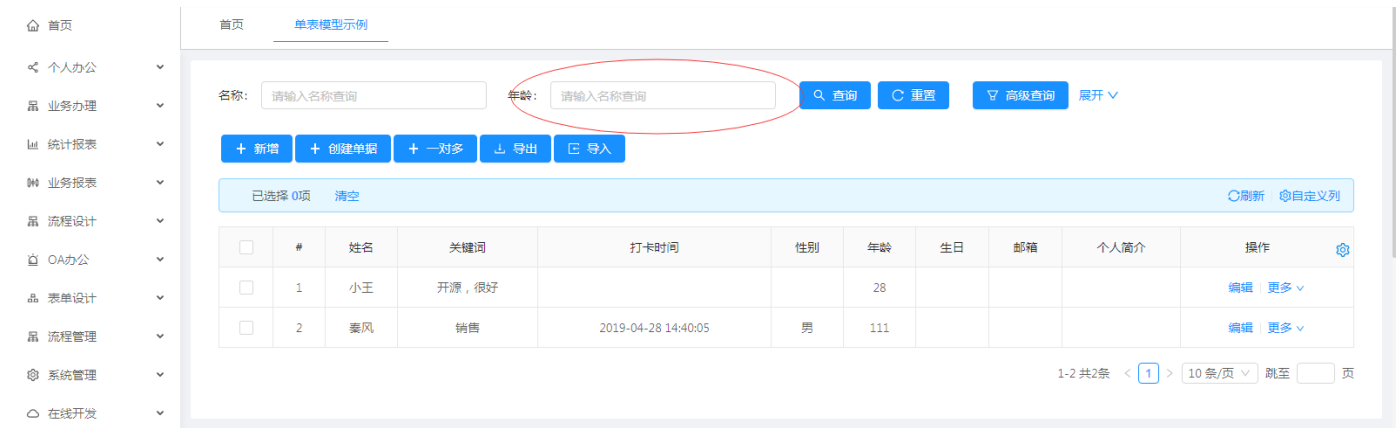
三、查询过滤器如何集成

第一步：页面实现查询条件

在线列表的查询区域，增加需要的查询字段，如下图所示。

```
JeecgDemoList.vue
1 <template>
2 <a-card :bordered="false">
3
4 <!-- 查询区域 -->
5 <div class="table-page-search-wrapper">
6   <a-form layout="inline">
7     <a-row :gutter="24">
8
9       <a-col :md="6" :sm="8">
10        <a-form-item label="名称">
11          <a-input placeholder="请输入名称查询" v-model="queryParam.name"></a-input>
12        </a-form-item>
13      </a-col>
14      <a-col :md="6" :sm="8">
15        <a-form-item label="年龄">
16          <a-input placeholder="请输入名称查询" v-model="queryParam.age"></a-input>
17        </a-form-item>
18      </a-col>
19    </a-row>
20    <template v-if="toggleSearchStatus">
21      <a-col :md="6" :sm="8">
22        <a-form-item label="字典下拉">
23          <j-dict-select-tag v-model="queryParam.sex" placeholder="请选择用户名称" dictCode="sex"/>
24        </a-form-item>
25      </a-col>
26      <a-col :md="6" :sm="8">
27        <a-form-item label="字典表下拉">
28          <j-dict-select-tag v-model="queryParam.realname" placeholder="请选择用户" dictCode="sys_user,realname,id"/>
29        </a-form-item>
30      </a-col>
31    </template>
32    <span style="..." class="table-page-search-submitButtons">
33      <a-col :md="6" :sm="24">
```

效果:



第二步：controller层处理

Controller中对应的处理逻辑中追加如下代码：

```
QueryWrapper<?> queryWrapper = QueryGenerator.initQueryWrapper(?, req.getParameterMap());
```

代码示例：

```
@GetMapping(value = "/list")
public Result<IPage<JeecgDemo>> list(JeecgDemo jeecgDemo, @RequestParam(name = "pageNo", defaultValue = "1") Integer pageNo, @RequestParam(name = "pageSize", defaultValue = "10") Integer pageSize, @RequestParam(name = "keyword", defaultValue = "") String keyword, @RequestParam(name = "sex", defaultValue = "") String sex, @RequestParam(name = "realname", defaultValue = "") String realname, @RequestParam(name = "age", defaultValue = "") String age, @RequestParam(name = "start", defaultValue = "") String start, @RequestParam(name = "end", defaultValue = "") String end, HttpServletRequest req) {
```

```
Result<IPage<JeecgDemo>> result = new Result<IPage<JeecgDemo>>();

//调用QueryGenerator的initQueryWrapper
QueryWrapper<JeecgDemo> queryWrapper = QueryGenerator.initQueryWrapper(jeecgDemo, req.getParam

Page<JeecgDemo> page = new Page<JeecgDemo>(pageNo, pageSize);
IPage<JeecgDemo> pageList = jeecgDemoService.page(page, queryWrapper);
result.setSuccess(true);
result.setResult(pageList);
return result;
}
```

四、更多查询规则参考

查询模式	用法	说明
模糊查询	支持左右模糊和全模糊 需要在查询输入框内前或后带*或是前后全部带*	
取非查询	在查询输入框前面输入! 则查询该字段不等于输入值的数据	(数值类型不支持此种查询,可以将数值字段定义为字符串类型的)
in查询	若传入的数据带,(逗号) 则表示该查询为in查询	
多选字段模糊查询	例如 现在name传入值 ,a,b,c, 那么结果sql就是 name like '%a%' or name like '%b%' or name like '%c%'	上述4 有一个特例, 若某一查询字段前后都带逗号 则会将其视为走这种查询方式 ,该查询方式是将查询条件以逗号分割再遍历数组将每个元素作like查询 用or拼接,

- 高级值规则用法 (查询内容, 带有查询规则符号)

查询模式	用法	举例
<	小于查询。 查询内容值规则: "lt+ 空格 + 内容"	输入值: "lt 100"
<=	小于等于查询。 查询内容值规则: "le+ 空格+ 内容"	输入值: "le 100"
>	大于查询。 查询内容值规则: "gt+ 空格+ 内容"	输入值: "gt 100"
>=	大于等于查询。 查询内容值规则: "ge+ 空格+ 内容"	输入值: "ge 100"