1. **搜索参数**(用户从客户端输入)

搜索参数分成2种，普通的字段和外键。

外键一般是objectId，无法直接显示在页面供用户来选择，所以需要外键对应记录的其他字段来进行选择。

*外键字段的名称不变，以便code处理。*

例如：coll定义如下

**billType**:{  
 **name**:{**type**:String,**unique**:true},  
 **parentBillType**:{**type**:**mongoose**.**Schema**.**Types**.**ObjectId**,**ref**:**"billTypes"**},  
 *//而外信息，简化查询处理。无validator（无inputRule），因为由程序自动处理，保证了正确性  
 //* **parentBillTypeFields**:{  
 **name**:{**type**:String}  
 },  
 **cDate**:{**type**:Date,**default**:Date.now},  
 **uDate**:{**type**:Date,**default**:Date.now},  
 **dDate**:{**type**:Date},  
},

实际存储冗余字段的名称是parentBillTypeFields（定义在fkAdditionalFieldsConfig

），而客户端传入的参数，外键的名称还是parentBillType。这样的好处是，可以直接判断字段是普通字段还是外键字段（parentBillType in fkAdditionalFieldsConfig，如果客户端传入的是parentBillTypeFields，不直观）。

parentBillType转换成parentBillTypeFields对应的搜索参数，可以通过fkAdditionalFieldsConfig很容易的完成

**格式：**

**普通字段：**

{

name:[{value:‘name1’},{value:’name2’}]

age:[{value:18,op:’gt’},{value:20,op:”eq”}]

}

外键字段：

{

~~parentBillType:~~

~~[~~

~~{name:{value:’fkname1’},title:{value:’overtime’},age:{value:18,op:gt}},~~

~~{name:{value:’fkname2’}},~~

~~]~~

parentBillType:

{

name:[{value:’asdf’},{value:’fda’}],

age:[{value:12, compOp:’gt’}, {value:24, compOp:’lt’}]

}

}

外键字段的格式在client中做最小的修改

转换成搜索参数为

{ '$or':

[

{ name: { '$in': [ 'name1', 'name2' ] } },

{ age: { '$gt': 18, '$lt': 20 } },

{ 'parentBillTypeFields.name': { '$in': [ 'asdf', 'fda' ] } },

{ 'parentBillTypeFields.age': { '$gt': 12, '$lt': 24 } }

]

}

1. **Create/update参数**

{

name:{value:’name1’}, //普通字段，

parentBillType:{value:’objectid1’} //外键字段，同时返回用户可读字段和objectid，用户在页面选择可读字段，js返回objectId

}

1. **delete参数**

因为angular的delete无法像put或者post一样传递参数，所以只能将objectId加在URL中（类似GET）

1. **Pagination参数**

4.1 server端的pagination

**{start**:start,**end**:end,**currentPage**:currentPage,**showPrevious**:showPrevious,**showNext**:showNext,**totalPage**:totalPage,**pageSize**:pageSize,**pageLength**:pageLength}

**start:** 页码起始数

**end:：**页码结束数

**currentPage：**当前页码数，**必定位于start/end之间**

**showPrevious：**boolean，**前一页**按钮是否显示

**showNext：**boolean，**后一页**按钮是否显示

**totalPage：**总共多少页

**pageSize：**每页显示多少记录

**pageLength：**每次显示几页

4.2 angular

**{'paginationInfo'**:{}, 存储从server返回的pagination信息  
**'pageRange'**:null, 根据pagination中的start和end，产生的数组（包括页码和active）  
**'goToPageNo'**:null, 存储用户输入的去第几页  
**'goToButtonEnable'**:false,} go按钮是否有效

**5. 添加/删除一个记录**

为了正确的处理提价/删除记录后的表格数据（例如，添加一个记录后，页数是否需要增加）。

**UnifiedRouterController：通过计算操作后的分页信息，与上传的currentPage比较，得出如何操作**

****

**5.1 对于create**：POST body={recorderInfo:{}, currentPage:}

recorderInfo: 添加的记录，currentPage：当前所在的页数。

考虑到如果当前有过滤条件，可能会把新加的记率过滤掉。所以添加完成后，重新读取记录的时候要：a **搜索条件重置为空**；b. **跳转到第一页**以便显示新添加的记录

**5.2 对于create**：POST body={searchParams:, currentPage:}

searchParams为当前搜索条件，以便删除后，补充上的记录还是过滤的记录

currentPage：判断删除后，页数的变化，以及记录的变化

**6. inputAttr**

**inputType**:在页面上，用什么html元素来处理用户的输入。

**Normal**：普通input

**Date**：普通input，通过datetimepicker初始化

**Select**：枚举值。例如性别，支取等。

**autoComplete**：普通input，但是通过autoComplete初始化，提过ac功能。