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:

{

parentName:[{value:’asdf’},{value:’fda’}],

parentAmount:[{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）