**Find、query、count接口入参结构**

**入参结构样例,支持嵌套and、or 以及 in**

{

"params": {

"sort": "desc",

"orderBy": "name",

"filter": {

"joiner": "and",

"conditions": [

{

"conditionName": "name",

"operator": "=",

"conditionValues": [

"wyc"

]

},

{

"joiner": "or",

"conditions": [

{

"conditionName": "creator",

"operator": "=",

"conditionValues": [

"wyc"

]

},

{

"conditionName": "modifier",

"operator": "CONTAINS",

"conditionValues": [

"wyc",

"wxf"

]

}

]

}

]

}

}

}

样例中的条件最终会变成

(name=wyc) and （ (creator=wyc) or (modifier in (wyc, wxf) ) ) order by name desc

jioner控制的是相同层次的conditions之间的关系

**单个条件的写法**

{

"params": {

"sort": "desc",

"orderBy": "name",

"filter": {

"conditionName": "name",

"operator": "=",

"conditionValues": [

"wyc"

]

}

}

}

或者

{

"params": {

"sort": "desc",

"orderBy": "name",

"filter": {

"joiner": "and", *// and 或者 or都可以*

"conditions": [

{

"conditionName": "name",

"operator": "=",

"conditionValues": [

"wyc"

]

}

]

}

}

}

**ConditionName、orderBy字段说明**

只支持模型项上自己定义的属性

如果输入了不是该模型的属性则会提示错误，错误样例如下

{

"result": "FAIL",

"data": [],

"errors": [

{

"code": "rdm.coresdk.unknown.exception",

"message": "unknown exception",

"detailMessage": "org.hibernate.QueryException: could not resolve property: string of: com.huawei.it.rdm.bean.module.DM0829002 [from com.huawei.it.rdm.bean.module.DM0829002 p left join fetch p.master where p.rdmDeleteFlag = 0 and UPPER(p.string) = UPPER('string') and p.latest = '1' order by p.id desc]"

}

]

}

其中could not resolve property: string为具体错误的属性名称

**支持的operator**

* =
* like  
  -- 全模糊匹配
* startWith
* endWith
* in
* <
* >
* <=
* >=
* <>
* ISNULL
* NOTNULL

**ISNULL NOTNULL传参**

{

"params": {

"sort": "desc",

"orderBy": "name",

"filter": {

"joiner": "and", *// and 或者 or都可以*

"conditions": [

{

"conditionName": "name",

"operator": "ISNULL", *// or ISNOTNULL*

"conditionValues": [ *// 这个时候有没有conditionValues都可以*

"wyc"

]

}

]

}

}

}

**参考对象的条件写法**

SimpleEntitylong idString nameTestQuerySimpleEntity refref

那么查询TestQuery对象时，希望在过滤条件里增加对SimpleEntity的过滤（就是多表关联查询）

如果是过滤id，则如下

{

"params": {

"filter": {

"joiner": "and",

"conditions": [

{

"conditionName": "ref.id",

"operator": "=",

"conditionValues": [

"406889137980243968"

]

}

]

}

}

}

如果想过滤其他属性，则类比就行

**扩展属性过滤条件写法**

固定前缀 extAttrs  
有两类

* extAttrs.扩展属性名
* extAttrs.扩展属性名.value
  + 适用于自定义精度的扩展属性

{

"params": {

"filter": {

"joiner": "and",

"conditions": [

{

"conditionName": "extAttrs.extAttrName1",

"operator": "=",

"conditionValues": [

"406889137980243968"

]

},

{

"conditionName": "extAttrs.extAttrName2.value",

"operator": "=",

"conditionValues": [

"12.133"

]

}

]

}

}

}

**分类属性过滤条件写法**

固定前缀clsAttrs

* clsAttrs.分类名.分类内属性名
* clsAttrs.分类名.分类内属性名.value
  + 适用于自定义精度的属性

{

"params": {

"filter": {

"joiner": "and",

"conditions": [

{

"conditionName": "clsAttrs.clsAttrName.attrName1",

"operator": "=",

"conditionValues": [

"123"

]

},

{

"conditionName": "clsAttrs.clsAttrName.attrName2.value",

"operator": "=",

"conditionValues": [

"12.123"

]

}

]

}

}

}