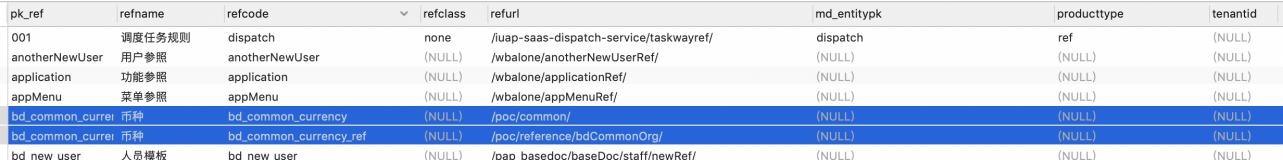
## baseservice3.5.3 下数据权限控制

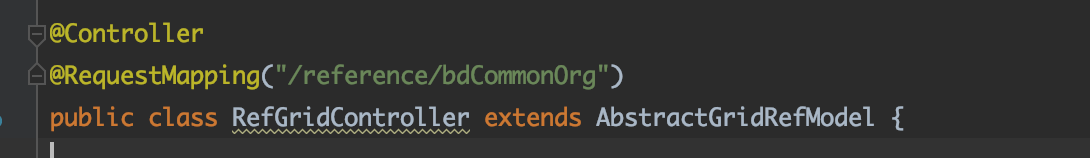
## 1.自定义参照

1.自定义参照需要根据参照的不同类型选择继承不同的类,如单表参照继承AbstractGridRefModel,树形参照继承AbstractTreeRefModel,多选参照继承AbstractCommonRefModel,树形参照继承AbstractTreeRefModel,树表参照继承AbstractTreeGridRefModel.

2.实现抽象类中的方法,除了AbstractTreeGridRefModel外,其他均实现各自类中的getCommonRefData方法即可,AbstractTreeGridRefModel需要实现blobRefClassSearch和blobRefSearch,这块案例中均有提供demo案例,按照案例实现即可.

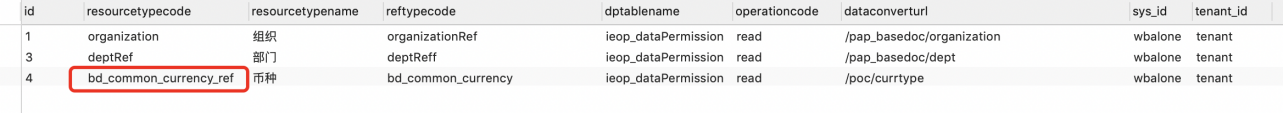
3.因为参照和数据权限是两套机制,所以他们之间在ref\_refinfo表中无法共用一套,所以还需要在ref\_refinfo表中增加一条数据.这条记录的refurl中的内容为 第一步中你自定义controller类中定义的requestMapping.如下图所示:





注意:  
正常的参照还是按照ref.xml中配置然后在ref\_refinfo表中进行配置 /项目名/common/ 的方法配置即可.与数据权限配置不产生冲突.

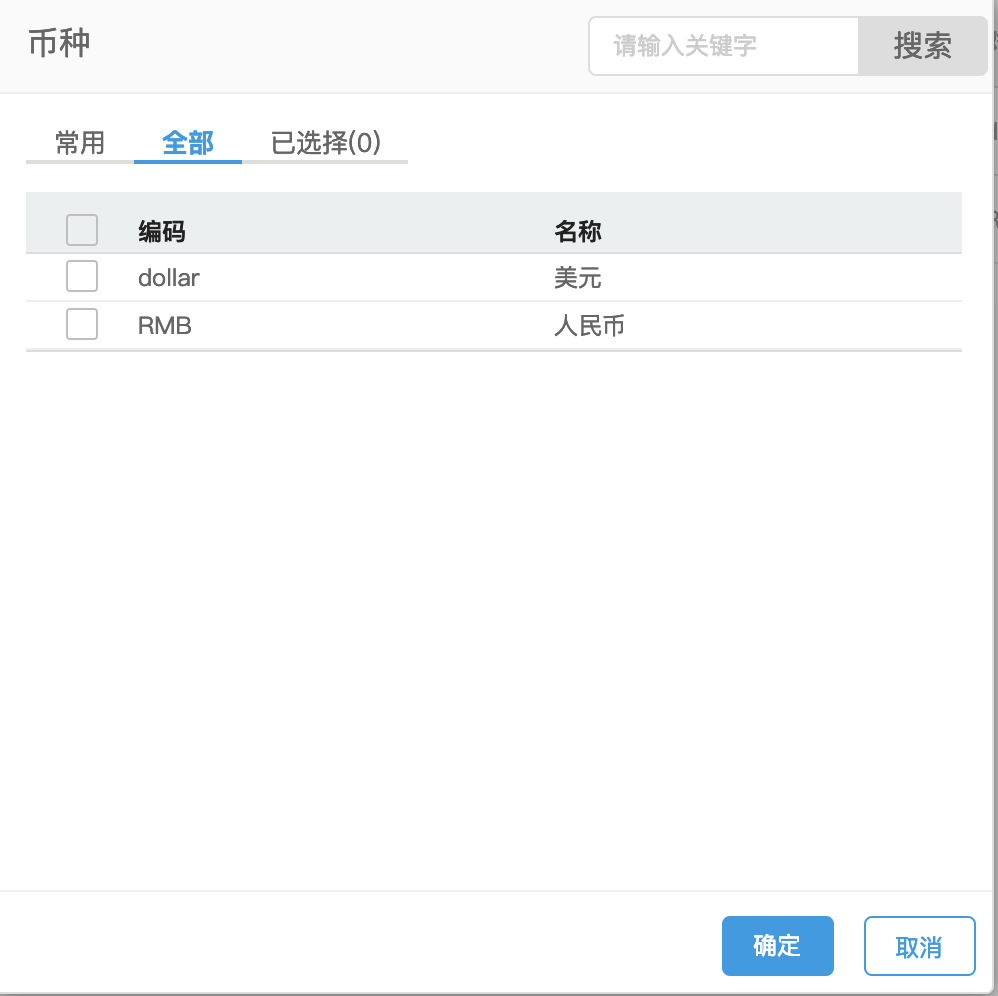
4. 在ieop\_dpprofile\_reg表中注册一条记录其中resourcetypecode为ref\_refinfo表中的ref\_refcode的值,dataconverturl的内容为该表格的controller对应的requestMapping,其中必须要指定一个requestMapping为getByIds的方法.具体可以参考demo案例.



5. 配置完这些内容以后,就可以使用管理员账号登录,进入管理中心-数据权限模块,点击分配对象,会出现如下内容:



选择币种,然后点击右侧的分配 按钮,会出现如下界面:



选择该角色能看到的内容,然后点击确定即可.

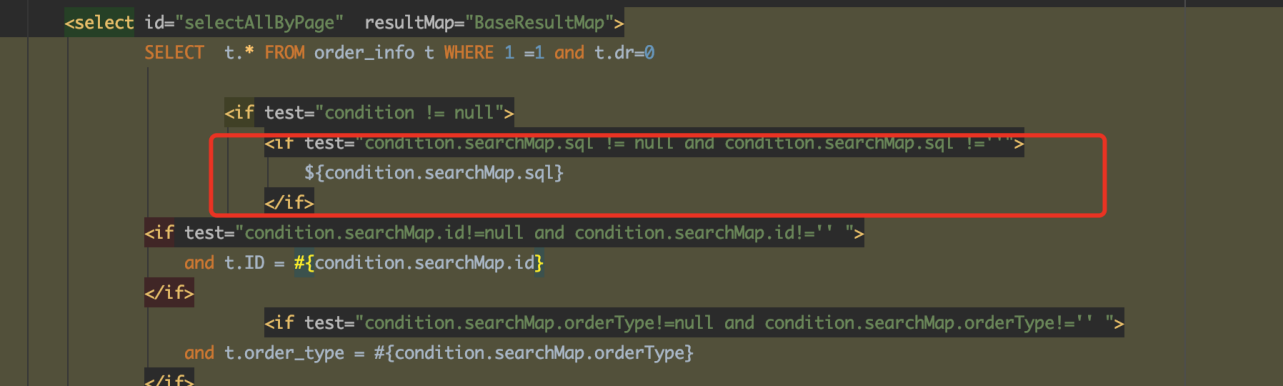
6. 此时在前台react代码中对参照进行配置就可以完成 该用户打开参照看到部分数据,在参照组件中增加如下代码clientParam: '{"isUseDataPower":"true"}',此时打开表单用户看到的参照就已经是通过过滤的了.

* 注意:  
  如果加了数据权限过滤后,如果该用户所属的角色均为进行数据权限分配的话,该用户是无法看到该参照下的数据的,打开参照时数据为空.
* 

7. 接下来就要进行列表的数据权限控制,在该列表所在的service中重写selectAllByPage方法,增加searchParams.addCondition("sql",buildPermSql());,详情可以参考demo案例,注意案例中的下面这段代码中的map中指定的内容,map的key为需要进行数据权限控制的数据库字段,不是属性值,value为ieop\_dpprofile\_reg中的resoucetypecode字段.

* /\*\*  
   \* 需要支持数据权限的资源  
   \* @return  
   \*/  
   private HashMap<String, String> processFieldDataPermResTypeMap(){  
   HashMap<String, String> fieldDataPermResTypeMap = new HashMap<String, String>();  
   fieldDataPermResTypeMap.put("apply\_no","bd\_common\_currency\_ref"); //数据库字段与权限资源名称的对应关系  
   return fieldDataPermResTypeMap;  
   }

8. 然后在mapper文件中增加一个查询条件

* <if test="condition.searchMap.sql != null and condition.searchMap.sql !=''">  
   ${condition.searchMap.sql}  
   </if>
* 

通过以上步骤就完成了一个自定义参照的数据权限过滤.

## 2. 系统参照

如果系统预置的数据权限参照比如组织和部门的时候,只需要进行自定义参照配置的第五步以后的内容即可.