1）定义3个DataType：CONDITION、PUB\_CODECLASS、PUB\_CODENAME

DataType：数据类型。数据类型是对所有系统中可能使用到的数据类型的抽象。

此参数具有多态性，当我们传入一个String类型的参数时，该String值表示数据

类型的name。当我们传入的参数时一个JSON对象时，系统会自动将该JSON对象中的属性复制到数据类型中。

2）定义3个DataSet：dsCondition、dsCodeClass、dsCodeName

dsCondition数据集中的数据类型为[CONDITION]。绑定onReady事件，组件被激活时（真正可用时），设置dsCondition封装的数据为空。self.setData({});

dsCodeClass数据集中的数据类型是[PUB\_CODECLASS]，DataProvider为pubBaseManage#getCodeClass。loadMode属性（数据的自动装载方式）为manual：当数据集的flush()或flushAsync()方法被调用时才开始装载数据。

dsCodeName数据集中的数据类型是[PUB\_CODENAME]，loadMode为manual

DataSet：数据集，用于封装一组数据（指实体对象或实体集合）的控件。

3）定义一个AutoForm：frmCondition，其绑定的数据集为dsCondition。

frmCondition绑定的数据集为dsCondition。cols属性设置表单布局的分栏方式。（\*,\*,\*,\*）4栏自适应。

AutoForm：自动表单。AutoForm的get方法除了获取指定属性值外。还提供了getElement方法，当传入一个以#开头的字符串时，#后面的内容江北识别成

AutoFormElement的名称，表示根据名称获取AutoFormElement。

4）定义1个ToolBar，里面包含1个DataPilot和3个按钮（查询按钮、新增按钮、编辑按钮）。

DataPilot绑定的数据集为dsCodeClass。itemCodes属性用于定义数据导航条中要显示哪些子对象的表达式。设置为pages（用于简单翻页按钮定义的特殊代码，表示|<，<，goto，>，>|这5个子对象的组合）。

查询按钮设置action属性来绑定一个actLoad动作。

新增按钮绑定一个onClick事件，view.get(“dsCodeClass”).insert;view.get(“dlCodeClass”).show();数据集的insert方法向数据集中添加一条顶层记录。使对话框dlCodeClass显示出来。

编辑按钮绑定一个onClick事件，view.get(“dlCodeClass”).show();弹出编辑对话框。

ToolBar：工具栏。

DataPilot：数据导航条。建议将数据导航条放置在ToolBar中。

5）定义1个Dialog：dlCodeClass。

对话框里有2个Button（btnSaveClass、btnCancelClass）和1个AutoForm。

AutoForm绑定dsCodeClass数据集。

保存按钮绑定onClick事件，先获取数据集dsCodeClass的当前实体数据，得到其属性cc\_nama、cc\_code、cc\_id。然后判断几个属性是否为空，空则抛出异常。cc\_id为空时（利用AjaxAction来设置参数传递到后台，执行后回调函数中的参数result是后台返回的结果。判断编码分类是否已经维护过，没有则执行UpdateAction来更新操作并隐藏对话框。），否则执行UpdateAction保存分类并隐藏对话框。

Dialog：对话框。提供了最大化、最小化、标题拖动、拖拽大小的功能，以及有

自己的皮肤。如不需要这些功能可以考虑使用FloatPanel。

6）定义1个Container：其中包括1个SplitPanel，其MainControl下有一个DataGrid（dgCodeClass），SideControl下有一个SplitPanel

SplitPanel：分隔面板。用来布局的一个组件，使用该组件需要指定一个边组件，边组件可以被收缩起来，并且可以指定边组件的宽度或者高度的最大值和最小值。

该组件的边组件也有预览功能。

direction属性：边组件的显示位置。（left、right、top、bottom，默认值left）。

position属性：边组建的当前位置。如果direction是left、right，则尺寸指的是宽，如果direction是top、bottom，则尺寸指的是高。

7）定义1个UpdateAction：actSaveAll。其下包含2个UpdateItem。

UpdateAction：提交动作控件。其updateItems属性为更新项的数组。更新项即要提交的DataSet的信息。数组中的每一个子对象包含下列属性：

dataSet

alias：用于描述这些被提交数据的别名

dataPath

validate

refreshMode

firstResultOnly

submitDeleteEntity

submitOldData

options

autoResetEntityState

## 增

1）点击新增按钮，绑定点击事件：

view.id(“dsCodeClass”).insert(); //向数据集中添加一条顶层记录

view.id(“dlCodeClass”).show(); //显示浮动控件——对话框

2）为对话框dlCodeClass绑定onHide()事件（在隐藏之后触发）：

view.id(“dsCodeClass”).getData().cancel();

//getData()返回数据集封装的数据 //cancel()

对话框里有保存按钮和取消按钮：

为保存按钮绑定点击事件：

//获取dsCodeClass当前的Entity对象

var entity = view.id(“dsCodeClass”).getData(“#”);

var cc\_name = entity.get(“CC\_NAME”);

var cc\_code = entity.get(“CC\_CODE”);

var cc\_id = entity.get(“CC\_ID”);

//判断cc\_name和cc\_code是否为空

if(cc\_name == null || cc\_name ==””) {

throw …

}

if(cc\_code == null || cc\_code ==””) {

throw …

}

//判断cc\_id是否为空

if(cc\_id == null || cc\_id ==””) {

view.id(“cmdCheckCodeClass”).set(“parameter”,{

CC\_CODE:cc\_code}).execute(function(result) {

if(result == “N”) {

throw …

}else{

view.id(“actSaveCodeClass”).execute(function() {

view.id(“dlCodeClass”).hide();

})

}

});

}else{

view.id(“actSaveCodeClass”).execute(function() {

view.id(“dlCodeClass”).hide();

});

}

执行远程过程的动作控件cmdCheckCodeClass，设置service为pubBaseCodeManage#checkCodeClass。

@Expose

public String checkCodeClass(Map<String, Object> parameter) throws Exception {

String cc\_code=(String) parameter.get("CC\_CODE");

String result="Y";

String checksql=" SELECT COUNT(\*) FROM PUB\_CODECLASS WHERE CC\_CODE='"+cc\_code+"'";

int count=jdbcUtils.getJdbcTemplate().queryForInt(checksql);

if(count>0){

result="N";

}

return result;

}

前台传入parameter值，通过后台校验CC\_CODE是否已存在，返回一个String字符串。execute()方法中传入回调对象，传入回调对象的参数即为命令执行后的返回结果。

所有判断都没问题后执行actSaveCodeClass，actSaveCodeClass设置了dataResolver为pubBaseCode#saveAll，并附带parameter——Entity，执行此方法对传递到后台的数据进行持久化。actSaveCodeClass下需要绑定一个UpdateItem（更新项的数组）。

dataPath用于描述希望提交DataSet中哪些数据的数据路径。（[#dirty]判断数据实体自身的状态）

@DataResolver

public void saveAll(List<Map<String, Object>> datas,Map<String, Object> parameter) throws Exception{

pubBaseCodeManage.saveAll(datas,parameter);

}

//pubBaseCodeManage继承了AbstractCurdManage类，其定义了对数据库进行CURD处理的方法

## 删

## 改

## 查