

pciMVC功能说明

目录

- [设计思路](#)
- [数据描述规则](#)
- [API参考](#)

修改记录

修改日期	版本号	修改内容	修改人
2014/3/6	0.0.1a	发布pciMVC	古晶
2014/3/10	0.0.2a	修改widget与item关联关系以适配一个组件对应多个数据项的场景。新增Util模块提供工具类方法，调整widget，data部分属性	古晶
2014/3/17	0.0.3a	修复IE9以下版本不支持数组forEach方法的问题。调整container所具有的属性,增加groupType属性。 widget增加id属性，可以通过id获取widget对象	古晶
2014/3/17	0.0.3a	form新增executeByNameList方法。item增加对初始值的存储	古晶
2014/4/8	0.0.4a	增加buttons属性用于添加辅助按钮，并提供外设组件的封装	古晶

设计思路

通过对数据项的描述代替之前创建组件所需的html代码和JS代码。建立组件与数据项的关联关系，并提供对于数据通用性操作的统一处理。

数据描述规则

数据描述分为四类 form,container,widget,data。示例：

```
{category: 'form', el: 'form_trans', children: [
  {
    category: 'container', type: 'ul', el: 'ul_part1', children: [
      {category: 'widget', attr: {desc: '文本框1', type: 'textfield', pjfAttr:{name: 'test'}}}
    ]
  },
  {
    category: 'container', type: 'ul', el: 'ul_part1', groupName: 'grp1', children: [
      {category: 'widget', attr: {desc: '文本框2', type: 'dateinput', pjfAttr:{name: 'group1'}}},
      {
        category: 'container', type: 'ul', el: 'ul_part1', groupName: 'grp3', children: [
          {category: 'widget', attr: {desc: '下拉框', type: 'selector', pjfAttr:{name: 'group1', categoryId: '116426'}}},
          {category: 'widget', attr: {desc: 'auto', type: 'auto', pjfAttr:{name: 'group2', categoryId: '116341'}}}
        ]
      }
    ]
  }
]
}]}
```

form

form用于描述提交到后台的数据集合

属性名	说明
category	类别定义，此处为form
el	对应html中form的id属性
children	数组格式，描述form中所具有的数据项，只能包含container

container

container用于描述组件在界面上放置的位置和数据项所属的层级

属性名	说明
category	类别定义，此处为container
el	对应html中包含组件dom的容器的id属性(如列式布局则为ul的id)
groupName	组名（可选）,表示数据项所属的层级,不影响界面展示
groupType	类型，表示组的存储方式为对象或数组，值为 <code>Group</code> 表示对象，值为 <code>Array</code> 表示数组。当groupName属性不为空时必填
children	数组格式，描述container中所具有的数据项，可以包含widget,data以及container

widget

widget用于描述需要在界面上展示的组件

属性名	子属性	说明
category	-	类别定义，此处为widget
attr	-	初始化所需属性
-	id	(可选)定义widget对象id，通过 <code>pciMVC.Util.getWidgetById</code> 方法获取widget对象，此id不同于pjf组件的id属性
-	type	实例化组件类型，现支持 <code>'textfield': PJF.ui.textfield, 'dateinput': PJF.ui.dateinput, 'selector': PJF.ui.select, 'auto': PJF.ui.autoComplete, 'radio': PJF.ui.radio, 'dateSpan': PJF.ui.dateSpan</code>
-	desc	组件描述，代替之前代码中label组件
-	pjfAttr	用于实例化pjf组件所需传入参数（dom属性无需传入）
-	initialize	(可选)pjf组件实例化完成调用完成额外初始化工作
-	getValuePost	(可选)取值后置操作
-	setValuePre	(可选)设值前置操作
-	buttons	(可选)定义组件的辅助按钮，格式为数组

button

button为widget辅助按钮的属性

属性名	说明
type	按钮类型，自定义为 <code>defined</code> ，其他为封装的外设包括 <code>card</code> 划卡， <code>IcCard</code> 划IC卡, <code>IdCard</code> 读身份证
id	(可选)用于通过widget组件的getAssistButtonById方法获取辅助按钮
desc	按钮显示的文字描述，类型为 <code>defined</code> 时有效
onclick	按钮点击的事件，具有参数 <code>field</code> 指向当前的widget，类型为 <code>defined</code> 时有效
successful	外设调用成功的回调函数，参数 <code>field</code> 指向当前的widget，参数 <code>res</code> 为外设调用返回的结果
failure	(可选)自定义失败处理逻辑

data

data用于描述不在界面上展示的数据项

属性名	子属性	说明
category	-	类别定义，此处为data
attr	-	初始化所需属性
-	name	数据项对应key值
-	defaultValue	(可选)默认值
-	getValuePost	(可选)取值后置操作
-	setValuePre	(可选)设值前置操作

API参考

pciMVC.Model.FormInstantiator

用于初始化form创建

方法名	参数	说明
getForm	-	获取创建的form
add	data	动态增加数据项，data为需要动态增加的数据项，格式与初始化时相同

pciMVC.Model.Form

数据集封装对象

方法名	参数	说明
getItems	-	获取所有Item对象
getFormData	-	根据初始化结构返回所有数据项的值
setFormData	data	根据与初始化结构相符的JSON对象对存储的Item对象进行赋值
getItemByName	key	根据name属性查找对应的Item并返回，对于深层对象，各个层级用 . 分隔
executeInGroup	groupName, funcname	对一个group内的数据项进行全量操作。groupName 为需要操作的组名，funcname 为要执行操作的方法名，方法所需的参数按次序加入实参中。
executeByNameList	names, funcname	根据name对数据项进行操作。names 为需要操作的数据项的name数组，funcname 为要执行操作的方法名，方法所需的参数按次序加入实参中。

pciMVC.Model.Item

数据项封装，对widget和data提供统一的取值设值接口

方法名	参数	说明
setValue	data	设值
getValue	-	取值
getType	-	返回封装的数据项类型 widget 或 data
execute	funcname	widget类型特有方法，用于执行组件方法，funcname 为要执行操作的方法名，方法所需的参数按次序加入实参中。

pciMVC.Util

工具方法模块，提供通用方法

方法名	参数	说明	示例
createNamespace	context,namespace	context 为需要创建命名空间的根对象， namespace 为命名空间层级描述。方法返回创建的命名空间	<pre>var root = pciMVC.Util.createNamespace(window, 'pci.test.MVCTest');</pre>
getWidgetById	id	id 为widget定义时id属性的值	<pre>var widget = pciMVC.Util.getWidgetById('cstNameInput');</pre>
each	obj,callback,args	对数组或对象中的属性进行遍历操作	<pre>pciMVC.Util.each(container.getDatas(), function (data) {safeInsertData(itemsInGroup, data.name.split('.'), pciMVC.Model.Item(data));});</pre>