控件组操作简介

控件组的使用需要在高级－》配置表达式下面进行配置。
打开表达式配置页，输入一个基本框架

```
{
  on : {
    init : function(event) {
    }
  }
}
```

大括号中的所有属性on代表此组建的事件。
init代表事件明，init为系统默认的一个事件当组建初始化的时候触发此事件。

紧接着我们建立一个控件组。
空间组的格式为：

```
[
        //第一行组件的内容
        [{
                //第一行第一列的组件的内容
                …
        },{
                //第一行第二列的组件的内容
                …
        } …],
        //第二行组件的内容
        [{
                //第二行第一列的组件的内容
                …
        },{
                //第二行第二列的组件的内容
                …
        } …]
        //第n行组件的内容
        [{
                //第n行第一列的组件的内容
                …
        },{
                //第n行第二列的组件的内容
                …
        }…]
]
```

下面举例创建一个三行的控件组，第一行是文字加一个选择输入框，第二列一个文字加一个树状下拉列表，第三列只有一个按钮。如下图：

```
var ctrlGroups = [
        [{
          type : "label",
          value : "项目名称"
        },{
          type : "autoComplete",
          value : "search",
          options : projects,
          on : {
            change : function(elem){
              var projectName = elem.value;
              target.setValue("projectName", projectName);
            }
          },
          format : {
            "id" : "id",
            "label" : "label",
            "searchkey" : "label"
          }
        }],[{
          type : "label",
          value : "管理域"
        },{
          type : "dropdowntree",
          value : "search",
          options : {
            "domainInfos" : domaintree
          },
          on : {
            change : function(elem){
              var domain = elem.value;
              target.setValue("domainPath", domain.domainPath);
            }
          },
          format : {
            "children" : "domainInfos"
          }
        }],[{
          type : "label",
          value : ""
        },{
          type : "button",
          value : "搜索",
          icon : "glyphicon glyphicon-search",
          class : "btn-primary",
          on : {
            click : function(elem){
              global.fire("refresh")
            }
          }
        }]
      ];
```

当创建完 ctrlGroups数组（对象）后需要应用它才能，我们需要执 event.target.render(ctrlGroups);

配置完成的代码应如下代码。

```
var a = {
  on : {
    init : function(event) {
      var ctrlGroups = [
        [{
          type : "label",
          value : "项目名称"
        },{
          type : "autoComplete",
          value : "search",
          options : projects,
          on : {
            change : function(elem){
              var projectName = elem.value;
              target.setValue("projectName", projectName);
            }
          },
          format : {
            "id" : "id",
            "label" : "label",
            "searchkey" : "label"
          }
        }],[{
          type : "label",
          value : "管理域"
        },{
          type : "dropdowntree",
          value : "search",
          options : {
            "domainInfos" : domaintree
          },
          on : {
            change : function(elem){
              var domain = elem.value;
              target.setValue("domainPath", domain.domainPath);
            }
          },
          format : {
            "children" : "domainInfos"
          }
        }],[{
          type : "label",
          value : ""
        },{
          type : "button",
          value : "搜索",
          icon : "glyphicon glyphicon-search",
          class : "btn-primary",
          on : {
            click : function(elem){
              global.fire("refresh")
            }
          }
        }]
      ];
```

```
    event.target.render(ctrlGroups)
  }
 }
}
```

执行的结果应该如下图。

前面创建的每一个单元都含有一个type属性，如“label”， “autoComplete”， “domaintree”， “button”，下面我们来分别介绍每一种组件的作用。

1.label是文字

属性名	作用	例子
value	设置标签显示的文字	“sample text”

2.input是输入框

属性名	作用	例子
value	设置输入框墨人显示的文字	“sample text”

on	事件集合	on : {
		change : function(elem){
		console.log(elem);
		}
		}

事件名	**触发条件**	**参数**
change	当输入框内容发生变化时触发	“sample text”

3.select是普通下拉框

属性名	作用	例子
value	设置输入框墨人显示的文字	设置为option中id的值，如1

on	事件集合	on : {
		change : function(elem){
		console.log(elem);
		}
		}

options	下拉列表的内容	[{
		id : 0,
		label : "项目1"
		},{
		id : 1,
		label : "项目名称2"
		},{
		id : 2,
		label : "名称3"
		}]

format	下拉列表解析格式	{
		"id" : "id",
		"label" : "label"
		}

事件名	触发条件	参数
change	当输入框内容发生变化时触发	“sample text”

4.datePicker是时间选择器.

事件名	触发条件	参数
change	当输入框内容发生变化时触发	“sample text”

5.autoComplete是选择输入框.

属性名	作用	例子

on	事件集合	on : {
		change : function(elem){
		console.log(elem);
		}
		}

options	下拉列表的内容	[{
		id : 0,
		label : "项目1"
		},{
		id : 1,
		label : "项目名称2"
		},{
		id : 2,
		label : "名称3"
		}]

format	下拉列表解析格式	{
		"id" : "id",
		"label" : "label"
		}

事件名	触发条件	参数
change	当输入框内容发生变化时触发	“sample text”

6.dropdowntree是树状结构下拉框。

属性名	作用	例子
value	设置输入框墨人显示的文字	设置为option中id的值，如1

on	事件集合	on : {
		change : function(elem){
		console.log(elem);
		}
		}

options	下拉列表的内容	[{
		id : 0,
		label : "项目1"
		},{
		id : 1,
		label : "项目名称2"
		},{
		id : 2,
		label : "名称3"
		}]

format	下拉列表解析格式	{
		"id" : "id",
		"label" : "label"
		}