[基于Extjs的大型信息管理平台[商业版]](http://www.cnblogs.com/CoolHu/)

<http://www.cnblogs.com/CoolHu/archive/2012/08/03/2622186.html>

# [Extjs4-布局](http://www.cnblogs.com/CoolHu/archive/2012/08/03/2622186.html)

本篇讲解Ext另一个重要的概念：布局。一般的容器类控件都是通过配置项items添加子控件的，这些子控件相对于父控件怎么定位呢，这里就要用到布局。某些容器类控件，它本身默认就集成了一种布局方式，例如比较典型的是：Ext.container.Viewport 布局控件，它其实就是一个border布局的容器，还有Ext.form.Panel、Ext.tab.Panel等。本节我们系统的分析各种布局方式。

## 一、absolute

这种方式的布局可以对子元素相对于父级容器控件进行绝对定位，它包含了x、y两个配置项用于定位。

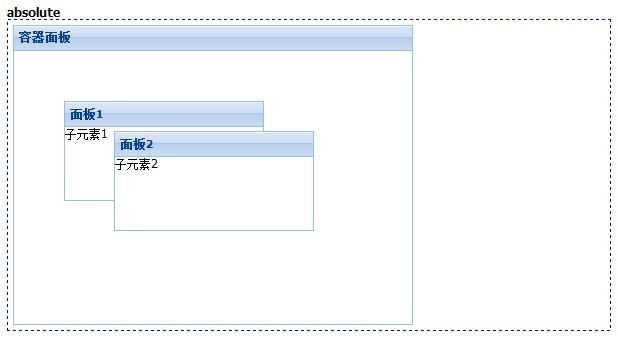
我们来看看一个例子：

[Js]

[?](http://www.cnblogs.com/lipan/archive/2011/12/27/2302719.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | //absolute  Ext.create('Ext.Panel', {      title: '容器面板',      renderTo: 'div1',      width: 400,      height: 300,      layout: 'absolute',      items: [{          title: '面板1',          xtype: "panel",          html: "子元素1",          width: 200,          height: 100,          x: 50,          y: 50        }, {          title: '面板2',          xtype: "panel",          html: "子元素2",          width: 200,          height: 100,          x: 100,          y: 80        }]  }); |

效果如下：



## 二、accordion

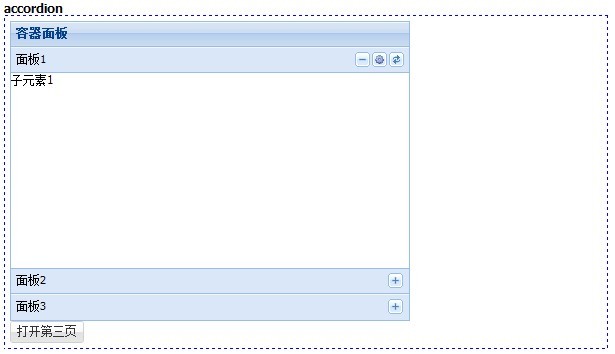
有的js插件里面accordion都是一个ui控件，但是Ext是通过布局的方式实现的，我们可以用面板控件作为它的折叠项，并且还可以用js来翻动活动项。

[Js]

[?](http://www.cnblogs.com/lipan/archive/2011/12/27/2302719.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | //accordion  Ext.create('Ext.Panel', {      title: '容器面板',      renderTo: 'div2',      width: 400,      height: 300,      layout: 'accordion',      items: [{          tools: [{ type: 'gear', handler: function () {              Ext.Msg.alert('提示', '配置按钮被点击。');          }          }, { type: 'refresh'}],          title: '面板1',          xtype: "panel",          html: "子元素1"        }, {          title: '面板2',          xtype: "panel",          html: "子元素2"      }, {          id: 'panel3',          title: '面板3',          xtype: "panel",          html: "子元素3"      }]  });  Ext.create("Ext.Button", {      renderTo: 'div2',      text: "打开第三页",      handler: function () {          Ext.getCmp('panel3').expand(true);      }  }); |

效果如下：



## 三、anchor

这个布局就是表单面板默认支持的，每一项占据一行，支持用anchor配置项分配各个子项的高度和宽度。为百分比时表示当前大小占父容器的百分比，为数字的时一般为负数，表示父容器的值减去差值，剩下的为子项的大小。

[Js]

[?](http://www.cnblogs.com/lipan/archive/2011/12/27/2302719.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | //anchor  Ext.create('Ext.Panel', {      title: '容器面板',      renderTo: 'div3',      width: 400,      height: 300,      layout: 'anchor',      items: [{          tools: [{ type: 'gear', handler: function () {              Ext.Msg.alert('提示', '配置按钮被点击。');          }          }, { type: 'refresh'}],          title: '面板1',          xtype: "panel",          html: "子元素1",          anchor: '80% 20%'        }, {          title: '面板2',          xtype: "panel",          html: "子元素2",          anchor: '-50 -200'      }, {          title: '面板3',          xtype: "panel",          html: "子元素3",          anchor: '100% 30%'      }]  }); |

效果如下：



## 四、border

这个布局可以定义东南西北四个方向的子元素，还有一个居中的子元素，一般用它来做页面整页布局，所以Ext.container.Viewport默认就支持了这个布局方式。

[Js]

[?](http://www.cnblogs.com/lipan/archive/2011/12/27/2302719.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45 | //border  Ext.create('Ext.Panel', {      title: '容器面板',      renderTo: 'div4',      width: 400,      height: 300,      layout: 'border',      defaults: {          split: true,                 //是否有分割线          collapsible: true,           //是否可以折叠          bodyStyle: 'padding:15px'      },      items: [{          region: 'north',            //子元素的方位：north、west、east、center、south          title: '北',          xtype: "panel",          html: "子元素1",          height: 70      }, {          region: 'west',          title: '西',          xtype: "panel",          html: "子元素2",          width: 100        }, {          region: 'east',          title: '东',          xtype: "panel",          html: "子元素2",          width: 100        }, {          region: 'center',          title: '主体',          xtype: "panel",          html: "子元素3"      }, {          region: 'south',          title: '南',          xtype: "panel",          html: "子元素4",          height: 70      }]  }); |

效果如下：



## 五、card

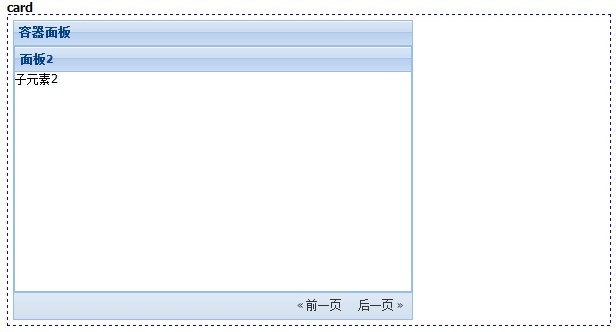
这个布局可以像卡片一样的切换每个子元素，各个子元素都会独占父元素的容器空间。我们可以定义翻页按钮来控制当前处于活动状态的子元素。

[Js]

[?](http://www.cnblogs.com/lipan/archive/2011/12/27/2302719.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45 | //card  var cardNav = function (incr) {      var l = Ext.getCmp('cardPanel').getLayout();      var i = l.activeItem.id.split('card')[1];      var next = parseInt(i, 10) + incr;      l.setActiveItem(next);      Ext.getCmp('cardPrev').setDisabled(next === 0);      Ext.getCmp('cardNext').setDisabled(next === 2);  };  Ext.create('Ext.Panel', {      title: '容器面板',      renderTo: 'div5',      width: 400,      height: 300,      layout: 'card',      activeItem: 1,                  //默认活动项      id: 'cardPanel',      items: [{          id: 'card0',          title: '面板1',          xtype: "panel",          html: "子元素1"        }, {          id: 'card1',          title: '面板2',          xtype: "panel",          html: "子元素2"      }, {          id: 'card2',          title: '面板3',          xtype: "panel",          html: "子元素3"      }],      bbar: ['->', {          id: 'cardPrev',          text: '« 前一页',          handler: Ext.Function.bind(cardNav, this, [-1])      }, {          id: 'cardNext',          text: '后一页 »',          handler: Ext.Function.bind(cardNav, this, [1])      }]    }); |

效果如下：



## 六、column

这个布局把子元素按照列进行划分。

[Js]

[?](http://www.cnblogs.com/lipan/archive/2011/12/27/2302719.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | //column  Ext.create('Ext.Panel', {      title: '容器面板',      renderTo: 'div6',      width: 400,      height: 300,      layout: 'column',      defaults: {                     //设置没一列的子元素的默认配置          layout: 'anchor',          defaults: {              anchor: '100%'          }      },      items: [{          columnWidth: 4 / 10,        //设置列的宽度          items: [{              title: '面板1',              border: false,              html: '子元素1'          }, {              title: '面板2',              border: false,              html: '子元素2'          }]      }, {          width: 120,          items: [{              title: '面板3',              border: false,              html: '子元素3'          }]      }, {          columnWidth: .40,          items: [{              title: '面板4',              border: false,              html: '子元素4'          }]      }]    }); |

效果如下：



## 七、fit

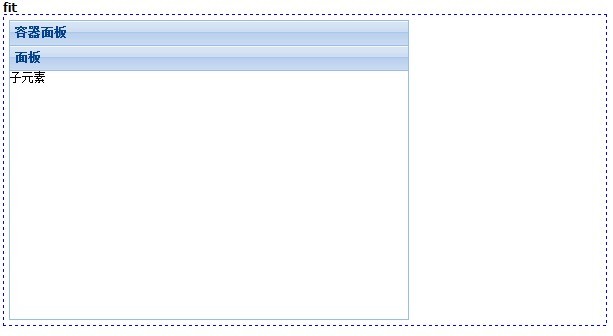
这个布局下子元素会独占全部的容器空间，一般用于只有一个子项的情况。

[Js]

[?](http://www.cnblogs.com/lipan/archive/2011/12/27/2302719.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | //fit  Ext.create('Ext.Panel', {      title: '容器面板',      renderTo: 'div7',      width: 400,      height: 300,      layout: 'fit',      items: [{          title: '面板',          html: '子元素',          border: false      }]  }); |

效果如下：



## 八、table

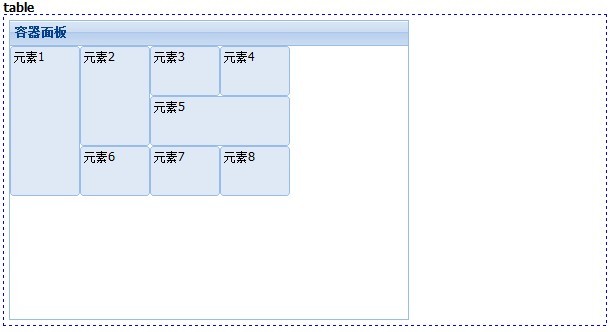
这个布局用表格定位的方式去组织子元素，我们可以像表格一样设置rowspan和colspan。

[Js]

[?](http://www.cnblogs.com/lipan/archive/2011/12/27/2302719.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | //table  Ext.create('Ext.Panel', {      title: '容器面板',      renderTo: 'div8',      width: 400,      height: 300,      layout: {          type: 'table',          columns: 4      },      defaults: { frame: true, width: 70, height: 50 },      items: [          { html: '元素1', rowspan: 3, height: 150 },          { html: '元素2', rowspan: 2, height: 100 },          { html: '元素3' },          { html: '元素4' },          { html: '元素5', colspan: 2, width: 140 },          { html: '元素6' },          { html: '元素7' },          { html: '元素8' }      ]  }); |

效果如下：



## 九、vbox

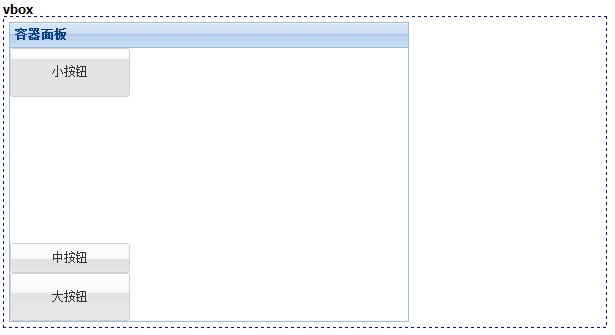
这个布局把所有的子元素按照纵向排成一列。

[Js]

[?](http://www.cnblogs.com/lipan/archive/2011/12/27/2302719.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | //vbox  Ext.create('Ext.Panel', {      title: '容器面板',      renderTo: 'div9',      width: 400,      height: 300,      layout: {          type: 'vbox',          pack: 'start',              //纵向对齐方式 start：从顶部；center：从中部；end：从底部          align: 'stretchmax'             //对齐方式 center、left、right：居中、左对齐、右对齐；stretch：延伸；stretchmax：以最大的元素为标准延伸      },      defaults: {          xtype: 'button'      },      items: [{          text: '小按钮',          flex: 1                      //表示当前子元素尺寸所占的均分的份数。      }, {          xtype: 'tbspacer',          //插入的空填充          flex: 3      },        {          text: '中按钮',          scale: 'medium'      }, {          text: '大按钮',          width: 120,          scale: 'large',          flex: 1      }]  }); |

效果如下：



## 十、hbox

跟vbox类似，只不过变成了横向的。

[Js]

[?](http://www.cnblogs.com/lipan/archive/2011/12/27/2302719.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | //hbox  Ext.create('Ext.Panel', {      title: '容器面板',      renderTo: 'div10',      width: 400,      height: 300,      layout: {          type: 'hbox',          pack: 'end',          align: 'middle'             //对齐方式 top、middle、bottom：顶对齐、居中、底对齐；stretch：延伸；stretchmax：以最大的元素为标准延伸      },      defaults: {          xtype: 'button'      },      items: [{          text: '小按钮'      },{          text: '中按钮',          scale: 'medium'      }, {          text: '大按钮',          width: 120,          scale: 'large'      }]  }); |

效果如下：



分类: [ExtJs](http://www.cnblogs.com/CoolHu/category/387085.html)

[基于Extjs的大型信息管理平台[商业版]](http://www.cnblogs.com/CoolHu/)

<http://www.cnblogs.com/CoolHu/archive/2012/08/03/2622360.html>

[**Extjs-xtype**](http://www.cnblogs.com/CoolHu/archive/2012/08/03/2622360.html)

|  |  |  |
| --- | --- | --- |
| **基本组件：** | | |
| ***xtype*** | ***Class*** | ***描述*** |
| button | Ext.Button | 按钮 |
| splitbutton | Ext.SplitButton | 带下拉菜单的按钮 |
| cycle | Ext.CycleButton | 带下拉选项菜单的按钮 |
| buttongroup | Ext.ButtonGroup | 编组按钮(Since 3.0) |
| slider | Ext.Slider | 滑动条 |
| progress | Ext.ProgressBar | 进度条 |
| statusbar | Ext.StatusBar | 状态条，2.2加进来，3.0 又去了 |
| colorpalette | Ext.ColorPalette | 调色板 |
| datepicker | Ext.DatePicker | 日期选择面板 |
|  | | |
| 容器及数据类组件 | | |
| ***xtype*** | ***Class*** | ***描述*** |
| window | Ext.Window | 窗口 |
| viewport | Ext.ViewPort | 视口，即浏览器的视口，能随之伸缩 |
| box | Ext.BoxComponent | 盒子组件，相当于一个 <div> |
| component | Ext.Component | 组件 |
| container | Ext.Container | 容器 |
| panel | Ext.Panel | 面板 |
| tabpanel | Ext.TabPanel | 选项面板 |
| treepanel | Ext.tree.TreePanel | 树型面板 |
| flash | Ext.FlashComponent | 显示 Flash 的组件(Since 3.0) |
| grid | Ext.grid.GridPanel | 表格 |
| editorgrid | Ext.grid.EditorGridPanel | 可编辑的表格 |
| propertygrid | Ext.grid.PropertyGrid | 属性表格 |
| editor | Ext.Editor | 编辑器 |
| dataview | Ext.DataView | 数据显示视图 |
| listview | Ext.ListView | 列表视图 |
|  | | |
| 工具栏组件： | | |
| ***xtype*** | ***Class*** | ***描述*** |
| paging | Ext.PagingToolbar | 分页工具条 |
| toolbar | Ext.Toolbar | 工具栏 |
| tbbutton | Ext.Toolbar.Button | 工具栏按钮 |
| tbfill | Ext.Toolbar.Fill | 工具栏填充区 |
| tbitem | Ext.Toolbar.Item | 工具条项目 |
| tbseparator | Ext.Toolbar.Separator | 工具栏分隔符 |
| tbspacer | Ext.Toolbar.Spacer | 工具栏空白 |
| tbsplit | Ext.Toolbar.SplitButton | 工具栏分隔按钮 |
| tbtext | Ext.Toolbar.TextItem | 工具栏文本项 |
|  | | |
| 菜单组件： | | |
| ***xtype*** | ***Class*** | ***描述*** |
| menu | Ext.menu.Menu | 菜单 |
| colormenu | Ext.menu.ColorMenu | 颜色选择菜单 |
| datemenu | Ext.menu.DateMenu | 日期选择菜单 |
| menubaseitem | BaseItem |  |
| menucheckitem | Ext.menu.CheckItem | 选项菜单项 |
| menuitem | Ext.menu.Item |  |
| menuseparator | Ext.menu.Separator | 菜单分隔线 |
| menutextitem | Ext.menu.TextItem | 文本菜单项 |
|  | | |
| 表单及表单域组件： | | |
| ***xtype*** | ***Class*** | ***描述*** |
| form | Ext.FormPanel/Ext.form.FormPanel | 表单面板 |
| checkbox | Ext.form.Checkbox | 多选框 |
| combo | Ext.form.ComboBox | 下拉框 |
| datefield | Ext.form.DateField | 日期选择项 |
| timefield | Ext.form.TimeField | 时间录入项 |
| field | Ext.form.Field | 表单字段 |
| fieldset | Ext.form.FieldSet | 表单字段组 |
| hidden | Ext.form.Hidden | 表单隐藏域 |
| htmleditor | Ext.form.HtmlEditor | HTML 编辑器 |
| label | Ext.form.Label | 标签 |
| numberfield | Ext.form.NumberField | 数字编辑器 |
| radio | Ext.form.Radio | 单选按钮 |
| textarea | Ext.form.TextArea | 多行文本框 |
| textfield | Ext.form.TextField | 表单文本框 |
| trigger | Ext.form.TriggerField | 触发录入项 |
| checkboxgroup | Ext.form.CheckboxGroup | 编组的多选框(Since 2.2) |
| displayfield | Ext.form.DisplayField | 仅显示，不校验/不被提交的文本框 |
| radiogroup | Ext.form.RadioGroup | 编组的单选按钮(Since 2.2) |
|  | | |
| 图表组件： | | |
| ***xtype*** | ***Class*** | ***描述*** |
| chart | Ext.chart.Chart | 图表组件 |
| barchart | Ext.chart.BarChart | 柱状图 |
| cartsianchart | Ext.chart.CartesianChart |  |
| columnchart | Ext.chart.ColumnChart |  |
| linechart | Ext.chart.LineChart | 连线图 |
| piechart | Ext.chart.PieChart | 扇形图 |
|  | | |
| 数据集 Store： | | |
| ***xtype*** | ***Class*** | ***描述*** |
| arraystore | Ext.data.ArrayStore |  |
| directstore | Ext.data.DirectStore |  |
| groupingstore | Ext.data.GroupingStore |  |
| jsonstore | Ext.data.JsonStore |  |
| simplestore | Ext.data.SimpleStore |  |
| store | Ext.data.Store |  |
| xmlstore | Ext.data.XmlStore |  |

分类: [ExtJs](http://www.cnblogs.com/CoolHu/category/387085.html)

# [ExtJs开发总结](http://www.cnblogs.com/CoolHu/archive/2012/11/23/2784056.html)

**读后很实用，感谢这位具有奉献精神的朋友！**

<http://www.cnblogs.com/CoolHu/archive/2012/11/23/2784056.html>

之前做过一阵的ExtJs开发，从最傻的符号问题，到后来的渲染问题都碰到过。下面是个人开发过程中做的一些总结，多是问题的应对：

1、引入js和css文件时注意文件的路径问题；

2、导入ext-base.js后注意设置Ext.BLANK\_IMAGE\_URL的值（透明图片s.gif位置）；

3、IE提示“缺少标识符，字符串或数字”错误，为配置时 “}”前多了逗号，且所处位置在Ext.onReady在同一个js文件

4、IE提示变量未定义，一般为有语法错误，如多余了”,”等

5、“无效字符”错误，可能是将”,”打成了”，”了

6、”缺少’}'”错误，问题是多个参数之间缺少了”,”导致

7、加载文件较多时，使用ExtJs可做一个加载提示

8、ext-base.js引入必须在ext-all.js之前

9、Ext.state.Manager.setProvider(new Ext.state.CookieProvider());  
初始化Ext状态管理器，在Cookie中记录用户的操作状态，如果不启用，象刷新时就不会保存当前的状态，而是重新加载  
如果窗口中有用可拖动面板的话，你在拖动后如果启动了Ext.state.Manager.setProvider(new Ext.state.CookieProvider()),就算刷新后面板仍然会在你拖动后的位置。如果不启用的话是不是就会按照默认的排列方式排列

10、对浏览器禁用javascript的提示  
<noscript class=”noticeDialog”><div>  
 <p>请开启浏览器的 JavaScript 支持！否则不能正常使用本系统<br>启用之后，刷新页面即可</p>  
</div></noscript>

11、TabPanel中放置复杂组件时，注意需要设置TabPanel的Width

12、extjs中文字体在firefox里显示偏小的问题，解决方法是再创建一个名为ext-patch.css的css文件，内容见<http://www.phpchina.com/html/78/78-28624.html>

13、中文化问题，在 ext-all.js 后面，挂上 ext-lang-zh\_CN.js 即可，如：  
   <script type=”text/javascript” src=”<%=contextPath%>/public/js/ext-base.js”></script>   
   <script type=”text/javascript” src=”<%=contextPath%>/public/js/ext-all.js”></script>   
   <script type=”text/javascript” src=”<%=contextPath%>/public/js/ext-lang-zh\_CN.js”></script>

14、ComboBox加载后自动选择第一项  
var pn\_zlfx\_cbb\_grade = new Ext.form.ComboBox({  
 displayField: ‘name’,  
 valueField: ‘id’,  
 triggerAction: ‘all’,  
 width: 80,  
 lazyInit: false,  
 mode: ‘local’,  
 editable: false,  
 forceSelection: true,  
 store: new Ext.data.JsonStore({  
  url: SITE\_URL+’get\_zlfx.asp?grade\_id=0′,  
  fields: ['id', 'name'],  
  root: ‘data’,  
  autoLoad: true,  
  listeners:{  
   load: function(store, records, options){  
    if (records.length != 0){  
     pn\_zlfx\_cbb\_grade.setValue(records[0].data.id);  
    }  
   }  
  }  
 })

15、JsonStore排序：sortInfo: {field: “name”, direction: “ASC|DESC”}

16、Extjs类的配置属性是不能被动态配置的,就如同上面这样的写法,当然,可能可以通过调用或设置某些公共的方法和属性来改变这些配置属性,但不能直接设置.

17、xhtml strict模式：<!DOCTYPE html  
PUBLIC “-//W3C//DTD XHTML 1.0 Strict//EN”  
“<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>“>

18、Html文档<script>标签中的js代码放在<![CDATA[和]]>之间，使xhtml验证时忽略中间的内容

19、JsonStore添加Record  
var orgaListRecord = new Ext.data.Record.create([  
  {name: 'id\_combo1'},  
  {name : 'name'}  
]);  
var orgaList = new Ext.data.JsonStore({  
  url: ‘./get\_organisme.php’,  
  root: ‘orga’,  
  fields: orgaListRecord  
});  
orgaList.add([  
 new orgaListRecord({'id\_combo1': 3,'name': 'Other'})  
]);  
orgaList.load({add: true}); // instead of orgaList.load();

20、数据加载前须判断是否有数据：  
if (typeof(pn\_zlfx\_gp\_mark\_subjects)==”undefined” || typeof(pn\_zlfx\_gp\_mark\_subjects.data)==”undefined”)  
    pn\_zlfx\_gp\_setting.store.removeAll();  
else  
    pn\_zlfx\_gp\_setting.store.loadData(pn\_zlfx\_gp\_mark\_subjects);

21、尽量将处理过程放在关键位置，减少重复代码

22、默认排序方式：sortDir

23、服务器返回的JSON数据字段用’或”括起来，避免与js关键词与保留字冲突

24、Ext.encode()出错，问题是json数据有问题

25、store增加totalProperty配置，可用store.getCount()获取

26、store中文排序异常问题，通过重载Ext.data.Store.prototype.applySort函数来解决

27、store.load()是异步操作，如果希望在数据载入后再执行操作，可以通过以下方式执行：  
store.load({  
  callback: function (records) {  
    alert(store.getTotalCount());  
  }  
});

28、 EXTJS items不能显示问题收藏  
有时经常碰到添加items时不能显示，郁闷不得行~~  
经不断测试可能有如下几个方法可以解决：  
1)试添加：Layout:’fit’  
2)若xtype为’panel’，可试添加：  
        listeners: {  
             ‘activate’: function(p) {p.doLayout();}  
             ,single:true  
         }  
3)若xtype为’tabpanel’，可试添加：  
        layoutOnTabChange:true  
总的来说是cmp.doLayout()问题….

29、Ext.ajax.request增加form和isUpload:true时，请求方式为enctype=”multipart/form-data”传值，asp中需特别解析

30、ComboBox增加mode:’local’配置来手动控制数据载入，同时可防止数据的二次加载

31、要使ComboBox手动输入的值能得到提交，需要增加ComboBox的Blur事件处理函数  
    onComboBoxBlur: function(field){  
     field.setValue(field.getRawValue());  
    }

32、Firebug显示所有Extjs组件事件  
Ext.util.Observable.prototype.fireEvent = Ext.util.Observable.prototype.fireEvent.createInterceptor(function() {  
    console.log(arguments);  
    return true;  
});

33、工具栏Toolbar内容的增删  
    var tb = this.gp.getTopToolbar();  
    // 删除工具栏内容  
    var i = tb.items.getCount();  
    while(i){  
  Ext.fly(tb.items.get(i).getEl()).remove();  
  tb.items.removeAt(i);  
 }  
 // 重新添加工具栏内容  
    tb.add(‘包括（’, this.cbgType, ‘）’);

34、Ext.ux.form.LovCombo不能设置forceSelection为true，否则当控件失去焦点时显示值会清空

35、IE下Ext.GridPanel的autoWidth或者layout:’fit’会将宽度拉的很长的解决办法：

var grid = new Ext.grid.GridPanel({

        bodyStyle:’width:100%’,

        autoWidth:true,

…….

});  
36、Ext.GridPanel加了CellSelection后，如设置viewConfig:{forceFit:true}点击单元格会导致表格跳动  
解决方案：<http://www.extjs.com/forum/showthread.php?p=282928#post282928>

37、ComboBox执行Filter第一次无效的解决是 增加lastQuery:” 配置

38、Panel内部高度自适应+自动出现滚动条，增加以下配置  
,autoScroll: true  
,style:”height:100%;”  
,bodyStyle:”height:100%;”

39、某些情况下TabPanel里加载iframe第一次不显示，可在panel显示后重新设置一下iframe的src强制加载一遍

分类: [ExtJs](http://www.cnblogs.com/CoolHu/category/387085.html)