# 最小引入集合

|  |
| --- |
| <link rel=*"stylesheet"* type=*"text/css"* href=*"../resources/css/ext-all.css"*>  <script type=*"text/javascript"* src=*"../ext-all.js"*></script>  <!--<script type="text/javascript" src="../bootstrap.js"></script>-->  <script type=*"text/javascript"* src=*"../locale/ext-lang-zh\_CN.js"*></script> |

* bootstrap.js可以根据环境判断是使用ext-all-debug.js还是使用ext-all.js

|  |
| --- |
| <html>  <head>  <link rel=*"stylesheet"* type=*"text/css"* href=*"../resources/css/ext-all.css"*>  <script type=*"text/javascript"* src=*"../ext-all.js"*></script>  <script type=*"text/javascript"*>  Ext.onReady(**function**() {  Ext.create('Ext.Viewport', {  layout: 'border',  items: [{  region: 'north',  html: '<h1 class="x-panel-header">Page Title工</h1>',  border: **false**,  margins: '0 0 5 0'  }, {  region: 'west',  collapsible: **true**,  title: 'Navigation',  width: 150  }, {  region: 'south',  title: 'South Panel',  collapsible: **true**,  html: 'Information goes here',  split: **true**,  height: 100,  minHeight: 100  }, {  region: 'east',  title: 'East Panel',  collapsible: **true**,  split: **true**,  width: 150  }, {  region: 'center',  xtype: 'tabpanel', // TabPanel itself has no title  activeTab: 0, // First tab active by default  items: {  title: 'Default Tab',  html: 'The first tab\'s content. Others may be added dynamically刘祖刚'  }  }]  //,renderTo: Ext.getBody()  });  });  </script>  </head>  <body>  </body>  </html> |

# xtype

|  |  |  |
| --- | --- | --- |
| **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** |  |  |
| 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 |  |
| arraystore | Ext.data.ArrayStore |  |
| directstore | Ext.data.DirectStore |  |
|  |  |  |
|  |  |  |

# Ext.data

## Model

### 简单例子

|  |
| --- |
| Ext.define('User', {  extend: 'Ext.data.Model',  //字段  fields: [  {name: 'name', type: 'string'},  {name: 'age', type: 'int', convert: **null**},  {name: 'phone', type: 'string'},  {name: 'alive', type: 'boolean', defaultValue: **true**, convert: **null**}  ],    //自定义方法  changeName: **function**() {  **var** oldName = **this**.get('name'),  newName = oldName + " The Barbarian";  **this**.set('name', newName);  }  });  **var** user = Ext.create('User', {  id : 'ABCD12345',  name : 'Conan',  age : 24,  phone: '555-555-5555'  });  //调用自定义方法  user.changeName();  //获取字段  alert(user.get('name')); |

### 验证

|  |
| --- |
| Ext.define('User', {  extend: 'Ext.data.Model',  //定义模型中的字段  fields: [  {name: 'name', type: 'string'},  {name: 'age', type: 'int'},  {name: 'phone', type: 'string'},  {name: 'gender', type: 'string'},  {name: 'username', type: 'string'},  {name: 'alive', type: 'boolean', defaultValue: **true**}  ],  //定义字段的验证规划  validations: [  {type: 'presence', field: 'age'},  {type: 'length', field: 'name', min: 2},  {type: 'inclusion', field: 'gender', list: ['Male', 'Female']},  {type: 'exclusion', field: 'username', list: ['Admin', 'Operator']},  {type: 'format', field: 'username', matcher: /([a-z]+)[0-9]{2,3}/}  ]  });  //生成一个实例  **var** instance = Ext.create('User', {  name: 'E',  gender: 'Male',  username: 'edspencer'  });  //验证  **var** errors = instance.validate();  //验证是否通过  alert(errors.isValid());  //错误结果条数  alert(errors.length);  //打印验证的错误结果  errors.each(**function**(item,index,len){  alert(item.field + ' : ' + item.message);  }); |

## Store

|  |
| --- |
| Ext.onReady(**function**(){  **var** myStore = Ext.create('Ext.data.Store', {  storeId:'sampleStore',  fields:[//定义model中的字段  {name: 'firstName', type: 'string'},  {name: 'lastName', type: 'string'},  {name: 'age', type: 'int'},  {name: 'eyeColor', type: 'string'},  {name: 'rocks', type: 'boolean'}  ],  proxy: {  type: 'ajax',  url: 'demo4Data.json',  reader: {  type: 'json',  root: 'users'  }  },  autoLoad: **true**  });  Ext.create('Ext.grid.Panel', {  title: 'Boolean Column Demo',  store: myStore,//也可以用'sampleStore'  columns: [//设置显示列  { text: 'firstName', dataIndex: 'firstName', flex: 1 },  { text: 'lastName', dataIndex: 'lastName', flex: 1 },  { text: 'age', dataIndex: 'age', flex: 1 },  { text: 'eyeColor', dataIndex: 'eyeColor', flex: 1 },  {  xtype: 'booleancolumn',  text: 'Rocks',  trueText: '是',  falseText: '否',  dataIndex: 'rocks'  }  ],  height: 200,  width: 400,  renderTo: Ext.getBody()  });  }); |

* **demo4Data.json**

|  |
| --- |
| {  users: [  {  firstName: 'lzg',  lastName: 'zg',  age: 11,  eyeColor: 'read',  rocks : **true**  },{  firstName: 'bba',  lastName: 'cc',  age: 21,  eyeColor: 'yellow',  rocks : **false**  },{  firstName: '111',  lastName: 'as',  age: 11,  eyeColor: 'yellow',  rocks : **false**  },{  firstName: '222',  lastName: 'fsdf',  age: 31,  eyeColor: 'yellow',  rocks : **false**  }  ]  } |

### 加载方法

方式一

|  |
| --- |
| **var** myStore = Ext.create('Ext.data.Store', {  ------  autoLoad: **true**  ------  }); |

方式二

|  |
| --- |
| **var** myStore = Ext.create('Ext.data.Store', {  ------  autoLoad: {gender: 0, hello: 25},//自加入参数  ------  }); |

方式三

|  |
| --- |
| **var** myStore = Ext.create('Ext.data.Store', {  ------  });  //请求参数  myStore.load({  params:{  hello: 1234  }  }); |

### 排序

* 在文件中store中配置

|  |
| --- |
| var store = Ext.create('Ext.data.Store', {  model: 'User',  sorters: [  {  property : 'age',  direction: 'DESC'  },  {  property : 'firstName',  direction: 'ASC'  }  ]  }); |

* 在程序中调用

|  |
| --- |
| //简单排序  myStore.sort('age', 'desc');  //多个排序  myStore.sort([  {  property : 'age',  direction: 'ASC'  },{  property : 'firstName',  direction: 'ASC'  }  ]); |

### 过滤

* 在文件中store中配置

|  |
| --- |
| var store = Ext.create('Ext.data.Store', {  model: 'User',  filters: [  {  property: 'firstName',  value : /Ed/  }  ]  }); |

* 在程序中调用

|  |
| --- |
| //简单过滤  myStore.filter('eyeColor', 'yellow');//过滤，显示字段eyeColor值为yellow的记录  //多个过滤条件  store.filter([  {property: "email", value: /\.com$/},  {filterFn: function(item) { return item.get("age") > 10; }}  ]);  //每个过滤器配置不同的root  store.filter([  Ext.create('Ext.util.Filter', {property: "email", value: /\.com$/, root: 'data'}),  Ext.create('Ext.util.Filter', {filterFn: function(item) { return item.get("age") > 10; }, root: 'data'})  ]); |

### 常用方法

|  |  |  |
| --- | --- | --- |
|  |  |  |
| add | 新加一条记录 | store.add({ name: '12', email: '33', phone: 'phone' }); |
| remove | 删除一条记录 | //获取选中行  var selection = grid.getSelectionModel().getSelection()[0];  if (selection) {  //删除  store.remove(selection);  } |
| Sync | 向后台同步数据 | store.sync(); |
| autoSync | 数据变化后是否自动同步 | store.autoSync |
| batchActions | Grid中的修改是否作一次ajax请求  默认为true | store.getProxy().batchActions |
| writeAllFields | 是否提交所有的字段，还是只提交变更字段  默认为true | store.getProxy().getWriter().writeAllFields |
| api | Proxy中的属性 | api: {//增、删、改、查的路径  read: 'users.jsp?method=read',  create: 'users.jsp?method=create',  update: 'users.jsp?method=update',  destroy: 'users.jsp?method=destroy'  } |
| reader | Proxy中的属性 | reader: {  type: 'json',//数据格式  successProperty: 'success',//是否成功字段  root: 'data',//数据的根属性  messageProperty: 'message'//消息属性  } |
| writer | Proxy中的属性 | writer: {  type: 'json',//数据格式  writeAllFields: **false**,//是否同步所有的字段  root: 'data'//数据的根属性  } |
| Listeners | 监听 | listeners: {  exception: **function**(proxy, response, operation){  Ext.MessageBox.show({  title: 'REMOTE EXCEPTION',  msg: operation.getError(),  icon: Ext.MessageBox.ERROR,  buttons: Ext.Msg.OK  });  }  } |
| convert | 转换某列的显示 | **var** ds = Ext.create('Ext.data.ArrayStore', {  fields: [  {name: 'company'},  {name: 'price', type: 'float'},  {name: 'change', type: 'float'},  {name: 'pctChange', type: 'float'},  {name: 'lastChange', type: 'date', dateFormat: 'n/j h:ia'},  // Rating dependent upon performance 0 = best, 2 = worst  {name: 'rating', type: 'int', convert: **function**(value, record) {  **var** pct = record.get('pctChange');  **if** (pct < 0) **return** 2;  **if** (pct < 1) **return** 1;  **return** 0;  }}  ],  data: myData  }); |

# 容器（container）

## Container

## Panel

Ext.panel.Panel

xtype: panel

* 常用属性

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 用途 | 值 | 备注 |
| collapsible | 面板是否可收缩 | True/false |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* 子类：

Ext.container.ButtonGroup

Ext.form.Panel

Ext.menu.Menu

Ext.panel.Table

Ext.grid.Panel

Ext.tree.Panel

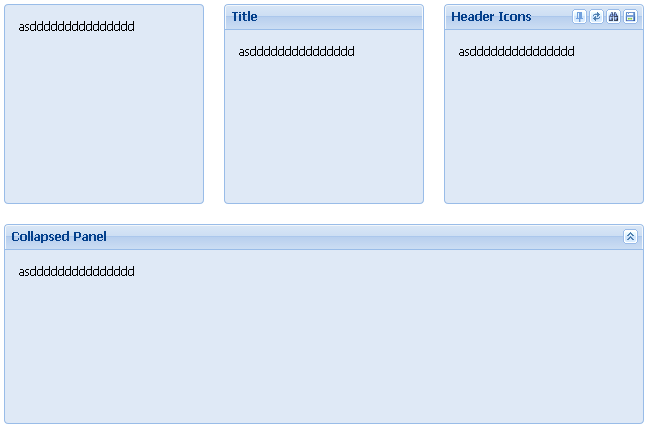
Ext.tab.Panel

Ext.tip.Tip

Ext.ux.GMapPanel

Ext.ux.event.RecorderManager

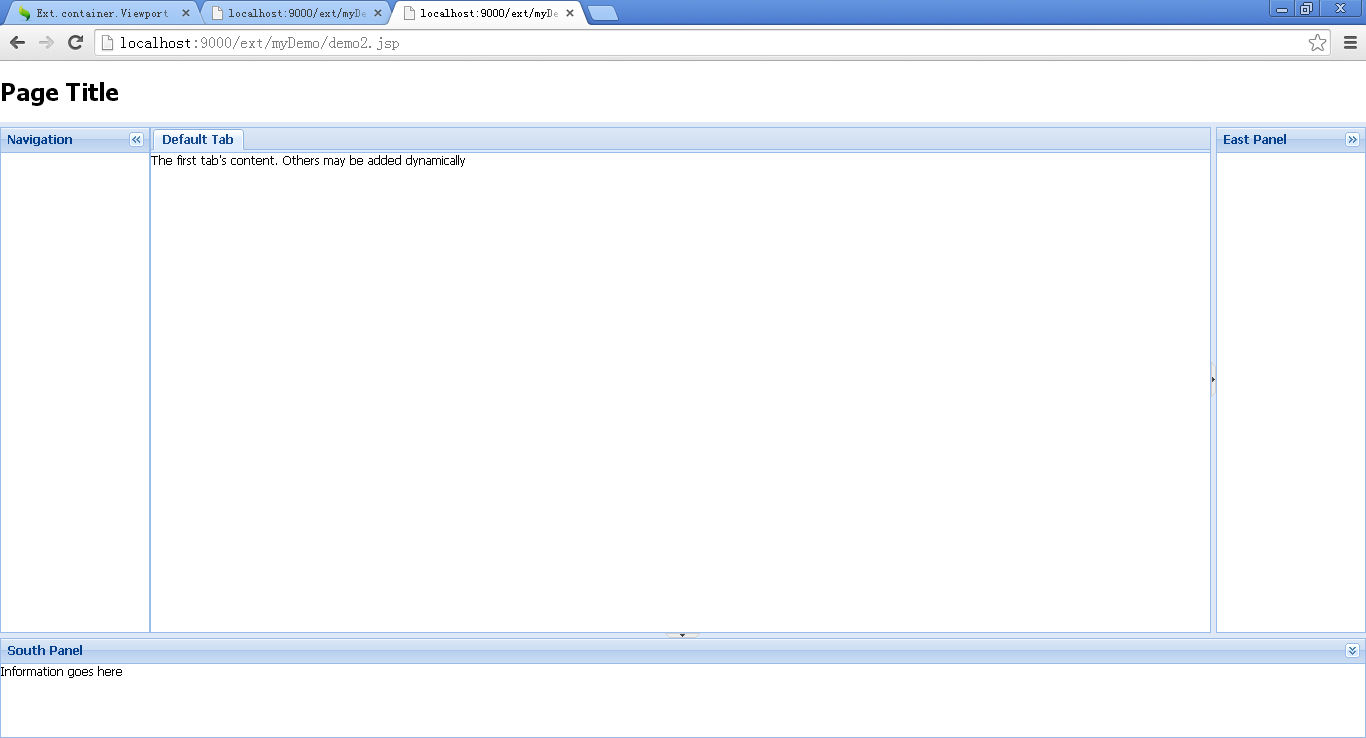
Ext.window.Window



|  |
| --- |
| Ext.define('KitchenSink.view.panel.BasicPanels', {  extend: 'Ext.Container',  //xtype: 'basic-panels',  width: 660,  requires: [  'Ext.layout.container.Table'  ],  layout: {  type: 'table',  columns: 3,  tdAttrs: { style: 'padding: 10px;' }  },  defaults: {  xtype: 'panel',  width: 200,  height: 200,  bodyPadding: 10,  frame: **true**//framed panels  },  initComponent: **function** () {  **this**.items = [  {  //无标题栏  html: 'asddddddddddddddd'  },  {  title: 'Title',//有标题栏  html: 'asddddddddddddddd'  },  {  title: 'Header Icons',  tools: [//工具栏  { type:'pin' },  { type:'refresh' },  { type:'search' },  { type:'save' }  ],  html: 'asddddddddddddddd'  },  {  title: 'Collapsed Panel',  collapsed: **true**,//初始时收缩态  collapsible: **true**,//是否可收缩  width: 640,  html: 'asddddddddddddddd',  colspan: 3  }  ];  **this**.callParent();  },  renderTo: Ext.getBody()  });  **var** v = **new** KitchenSink.view.panel.BasicPanels(); |

## Viewport

直接显示于 document body中，自动适应浏览器大小。



|  |
| --- |
| Ext.create('Ext.container.Viewport', {  layout: 'border',  items: [{  region: 'north',  html: '<h1 class="x-panel-header">Page Title</h1>',  border: **false**,  margins: '0 0 5 0'  }, {  region: 'west',  collapsible: **true**,  title: 'Navigation',  width: 150  // could use a TreePanel or AccordionLayout for navigational items  }, {  region: 'south',  title: 'South Panel',  collapsible: **true**,  html: 'Information goes here',  split: **true**,  height: 100,  minHeight: 100  }, {  region: 'east',  title: 'East Panel',  collapsible: **true**,  split: **true**,  width: 150  }, {  region: 'center',  xtype: 'tabpanel', // TabPanel itself has no title  activeTab: 0, // First tab active by default  items: {  title: 'Default Tab',  html: 'The first tab\'s content. Others may be added dynamically'  }  }]  }); |

# 布局

## Auto

Ext.layout.container.Auto

layout: auto, autocontainer

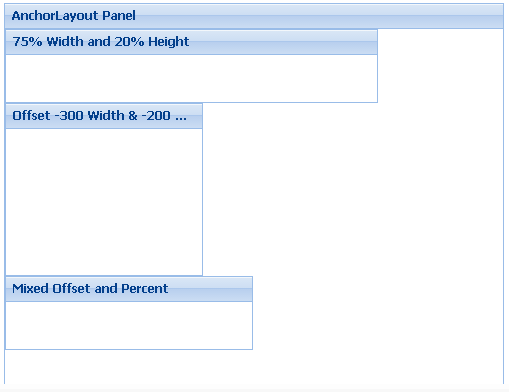


|  |
| --- |
| Ext.create('Ext.Panel', {  width: 500,  height: 280,  title: "AutoLayout Panel",  layout: 'auto',  renderTo: document.body,  items: [{  xtype: 'panel',  title: 'Top Inner Panel',  width: '75%',  height: 90  },  {  xtype: 'panel',  title: 'Bottom Inner Panel',  width: '75%',  height: 90  }]  }); |

## Anchor

Ext.layout.container.Anchor

layout: anchor



|  |
| --- |
| Ext.create('Ext.Panel', {  width: 500,  height: 400,  title: "AnchorLayout Panel",  layout: 'anchor',  renderTo: Ext.getBody(),  items: [  {  xtype: 'panel',  title: '75% Width and 20% Height',  anchor: '75% 20%'//父容器长、宽的百分比  },  {  xtype: 'panel',  title: 'Offset -300 Width & -200 Height',  anchor: '-20 -200'//父容器长、宽减去像素  },  {  xtype: 'panel',  title: 'Mixed Offset and Percent',  anchor: '-250 20%'//像素、百分比混用  }  ]  }); |

## Border

Ext.layout.container.Border

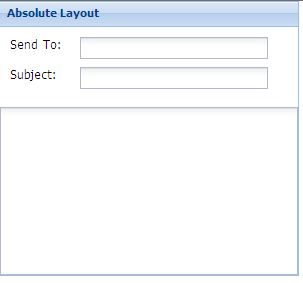
layout: border

|  |
| --- |
| Ext.create('Ext.Viewport', {  layout: 'border',  items: [{  region: 'north',  html: '<h1 class="x-panel-header">Page Title工</h1>',  border: **false**,  margins: '0 0 5 0'  }, {  region: 'west',  collapsible: **true**,  title: 'Navigation',  width: 150  }, {  region: 'south',  title: 'South Panel',  collapsible: **true**,  html: 'Information goes here',  split: **true**,  height: 100,  minHeight: 100  }, {  region: 'east',  title: 'East Panel',  collapsible: **true**,  split: **true**,  width: 150  }, {  region: 'center',  xtype: 'tabpanel', // TabPanel itself has no title  activeTab: 0, // First tab active by default  items: {  title: 'Default Tab',  html: 'The first tab\'s content. Others may be added dynamically刘祖刚'  }  }]  //,renderTo: Ext.getBody()  }); |

## Absolute（绝对）

Ext.layout.container.Absolute

layout: absolute

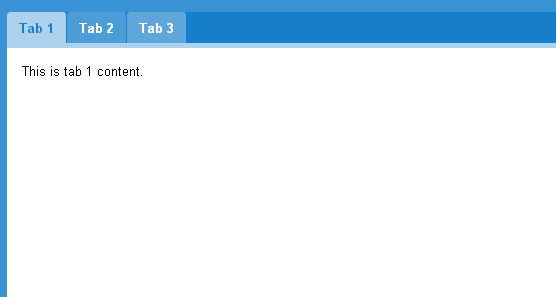


|  |
| --- |
| Ext.create('Ext.form.Panel', {  title: 'Absolute Layout',  width: 300,  height: 275,  layout: 'absolute',  url:'save-form.php',  defaultType: 'textfield',  items: [{  x: 10,  y: 10,  xtype:'label',  text: 'Send To:'  },{  x: 80,  y: 10,  name: 'to',  anchor:'90%' // anchor width by percentage  },{  x: 10,  y: 40,  xtype:'label',  text: 'Subject:'  },{  x: 80,  y: 40,  name: 'subject',  anchor: '90%' // anchor width by percentage  },{  x:0,  y: 80,  xtype: 'textareafield',  name: 'msg',  anchor: '100% 100%' // anchor width and height  }],  renderTo: Ext.getBody()  }); |

## cart(tabpanel)

Ext.tab.Panel

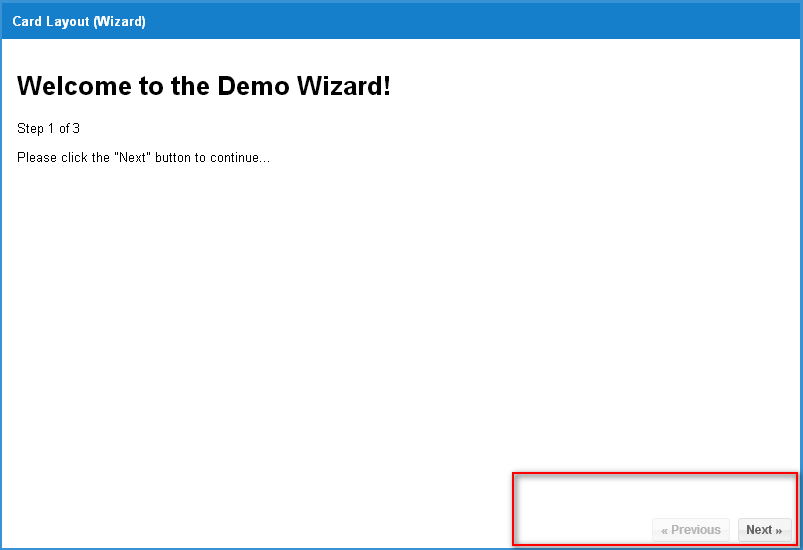
xtype: tabpanel



|  |
| --- |
| Ext.onReady(**function**() {  **var** tabs = Ext.create('Ext.tab.Panel', {  width: 400,  height: 400,  tabPosition: 'bottom', //tabBar的位置，默认时top  plain: **true**, //是否去掉tabBar的背景颜色  renderTo: document.body,//设置部署位置  activeTab: 1,//设置激活的页  listeners: {//增加tab页切换时的监听器  //id=disactive1的tab页，不能激活  beforetabchange: **function**(tabs, newTab, oldTab) {  **return** newTab.id != 'disactive1';  }  },  items: [{  title: 'Foo',  html: 'hello'  }, {  title: 'Bar',  tabConfig: {  title: 'Custom Title',  tooltip: 'A button tooltip'//提示  },  html: 'world'  }, {  id: 'disactive1',  title: '不能激活'  }, {  id: 'hiddenTab1',  title: '隐藏',  hidden: **true**, //隐藏tab页  html: '隐藏'  }, {  id: 'showTab1',  title: '显示',  html: '显示'  }, {  id: 'remove-this-tab',  title: 'remove-this-tab',  html: 'remove-this-tab'  }]  });  //延时执行函数  setTimeout(**function**(){  //获取id=showTab1的tab页，并隐藏  tabs.child('#showTab1').tab.hide();  //获取id=hiddenTab1的tab页，并显示  **var** hiddenTab1 = tabs.child('#hiddenTab1');  hiddenTab1.tab.show();  //激活某tab页  tabs.setActiveTab(hiddenTab1);  //tabs.setActiveTab(1);//也可以根据索引位置激活  }, 1000);  Ext.create('Ext.button.Button', {  text : '获取当前激活的tab',  scope : **this**,  handler : **function**() {  **var** tab = tabs.getActiveTab();//获取当前激活tab  alert('Current tab: ' + tab.title);  },  renderTo : Ext.getBody()  });  Ext.create('Ext.button.Button', {  text : '增加tab',  scope : **this**,  handler : **function**() {  //新加tab页  **var** tab = tabs.add({  //tabs.items.length获取当前共有几个tab页  title: 'Tab ' + (tabs.items.length + 1),  html : 'Another one'  });  tabs.setActiveTab(tab);  },  renderTo : Ext.getBody()  });  Ext.create('Ext.button.Button', {  text : '删除tab',  scope : **this**,  handler : **function**() {  //删除id=remove-this-tab的tab页  **var** tab = Ext.getCmp('remove-this-tab');  tabs.remove(tab);  },  renderTo : Ext.getBody()  });  }); |

## cart(wizard)（向导）

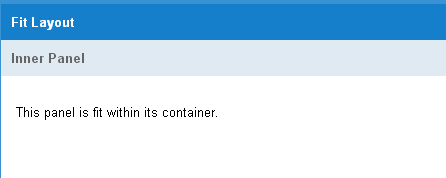
Layout managers > border brower > basic ext layouts > cart(wizard)



## fit（自动充满父容器）

Ext.layout.container.Fit

layout: fit

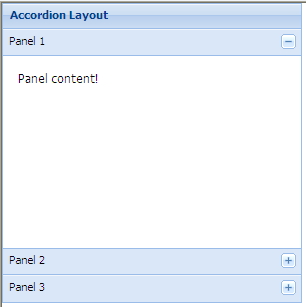


|  |
| --- |
| Ext.onReady(**function**() {  Ext.create('Ext.Viewport',{  id: 'fit-panel',  title: 'Fit Layout',  layout: 'fit',  items: {  title: 'Inner Panel',  html: '<p>This panel is fit within its container.</p>',  bodyStyle: 'padding:15px',  ui: Ext.themeName == 'neptune' ? 'light' : 'default',  border: **false**  }  });  }); |

## accordion（手风琴）

Ext.layout.container.Accordion

layout: accordion

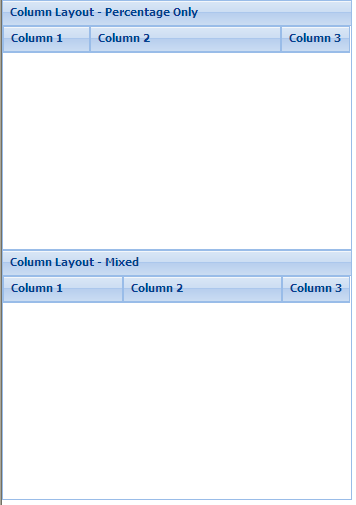


|  |
| --- |
| Ext.create('Ext.panel.Panel', {  title: 'Accordion Layout',  width: 300,  height: 300,  defaults: {  // applied to each contained panel  bodyStyle: 'padding:15px'  },  layout: {  // layout-specific configs go here  type: 'accordion',  titleCollapse: **false**, //是否点击标题栏时就可以触发  animate: **false**, //是否动画  activeOnTop: **false** //激活后是否置顶  },  items: [{  title: 'Panel 1',  html: 'Panel content!'  },{  title: 'Panel 2',  html: 'Panel content!'  },{  title: 'Panel 3',  html: 'Panel content!'  }],  renderTo: Ext.getBody()  }); |

## Column

Ext.layout.container.Column

series: column

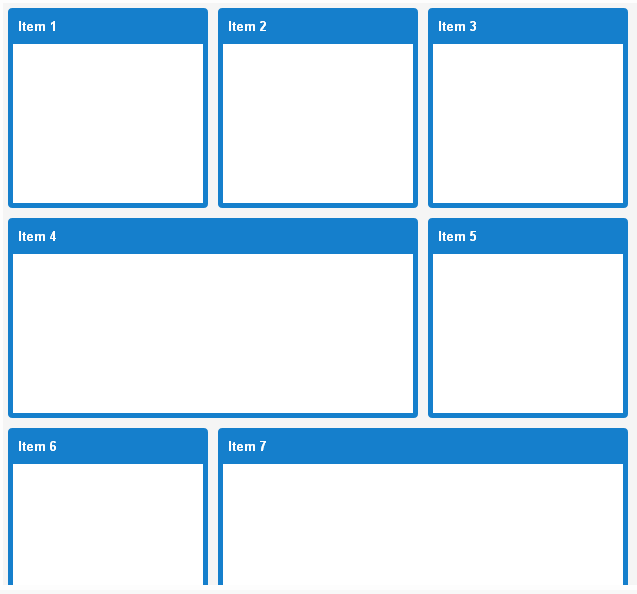


|  |
| --- |
| //所有元素的columnWidth的合要为1  Ext.create('Ext.panel.Panel', {  title: 'Column Layout - Percentage Only',  width: 350,  height: 250,  layout:'column',  items: [{  title: 'Column 1',  columnWidth: 0.25  },{  title: 'Column 2',  columnWidth: 0.55  },{  title: 'Column 3',  columnWidth: 0.20  }],  renderTo: Ext.getBody()  });  //混合使用width 和 columnWidth时  //columnWidth属性的所有值为1  //columnWidth的绝对宽是width的余值  Ext.create('Ext.Panel', {  title: 'Column Layout - Mixed',  width: 350,  height: 250,  layout:'column',  items: [{  title: 'Column 1',  width: 120  },{  title: 'Column 2',  columnWidth: 0.7  },{  title: 'Column 3',  columnWidth: 0.3  }],  renderTo: Ext.getBody()  }); |

## Table

Ext.layout.container.Table

layout: table

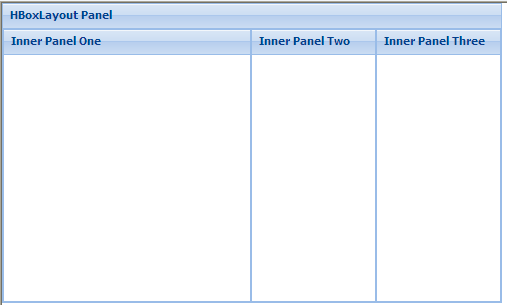


|  |
| --- |
| Ext.onReady(**function**() {  **var** panel = Ext.create('Ext.Panel', {  id:'main-panel',  baseCls:'x-plain',  renderTo: Ext.getBody(),  layout: {  type: 'table',  columns: 3  },  // applied to child components  defaults: {frame:**true**, width:200, height: 200},  items:[{  title:'Item 1'  },{  title:'Item 2'  },{  title:'Item 3'  },{  title:'Item 4',  width:410,  colspan:2  },{  title:'Item 5'  },{  title:'Item 6'  },{  title:'Item 7',  width:410,  colspan:2  },{  title:'Item 8',  rowspan: 2,  height: 410  },{  title:'Item 9'  },{  title:'Item 10'  },{  title:'Item 11',  rowspan: 2,  height: 410  },{  title:'Item 12'  },{  title:'Item 13'  },{  title:'Item 14',  rowspan: 2,  height: 410  },{  title:'Item 15'  },{  title:'Item 16'  },{  title:'Item 17',  rowspan: 2,  height: 410  },{  title:'Item 18',  width:410,  colspan:2  },{  title:'Item 19'  },{  title:'Item 20'  }]  });  }); |

## Hbox

Ext.layout.container.HBox

layout: hbox

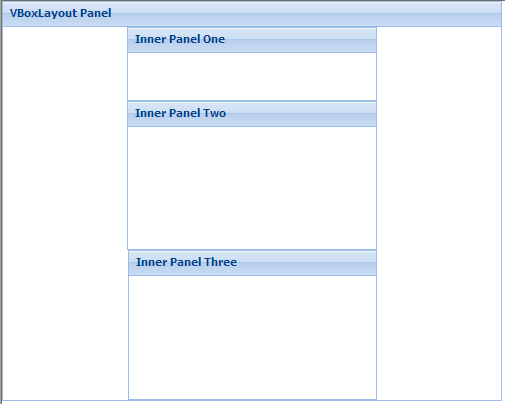


|  |
| --- |
| Ext.create('Ext.Panel', {  width: 500,  height: 300,  title: "HBoxLayout Panel",  layout: {  type: 'hbox',  align: 'stretch'  },  renderTo: document.body,  items: [{  xtype: 'panel',  title: 'Inner Panel One',  flex: 3 //显示宽度所占比重 本例中其值3,总值为5,其比重为60%  },{  xtype: 'panel',  title: 'Inner Panel Two',  flex: 1 //显示宽度所占比重 本例中其值1,总值为5,其比重为20%  },{  xtype: 'panel',  title: 'Inner Panel Three',  flex: 1 //显示宽度所占比重 本例中其值1,总值为5,其比重为20%  }]  }); |

## vBox

Ext.layout.container.VBox

layout: vbox

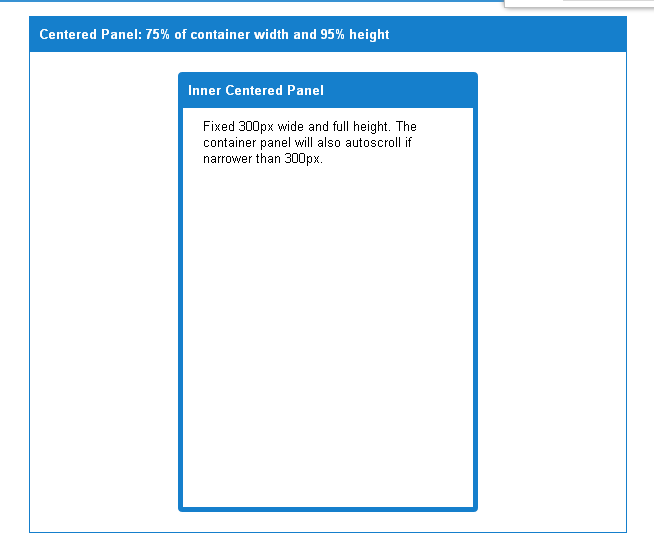


|  |
| --- |
| Ext.create('Ext.Panel', {  width: 500,  height: 400,  title: "VBoxLayout Panel",  layout: {  type: 'vbox',  align: 'center'  },  renderTo: document.body,  items: [{  xtype: 'panel',  title: 'Inner Panel One',  width: 250,  flex: 2 //显示高度所占比重 本例中其值2,总值为10,其比重为20%  },  {  xtype: 'panel',  title: 'Inner Panel Two',  width: 250,  flex: 4 //显示高度所占比重 本例中其值4,总值为10,其比重为40%  },  {  xtype: 'panel',  title: 'Inner Panel Three',  width: '50%',  flex: 4 //显示高度所占比重 本例中其值4,总值为10,其比重为40%  }]  }); |

## Center居中

Ext.ux.layout.Center

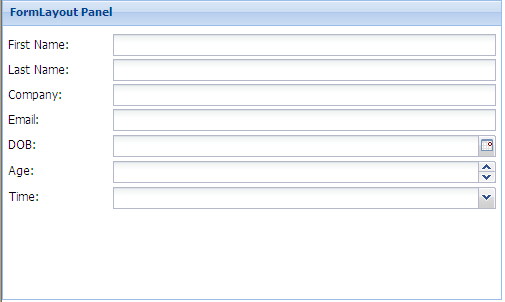
layout: ux.center



## Form

Ext.layout.container.Form

layout: form



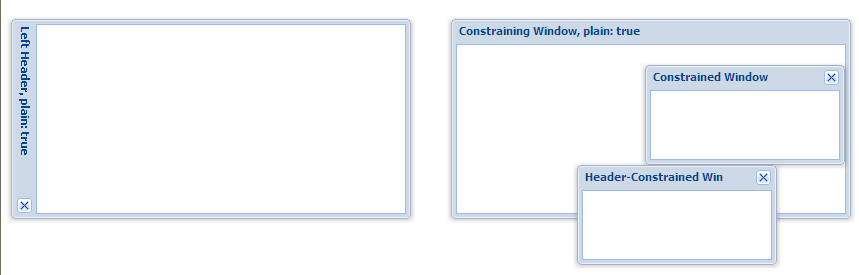
|  |
| --- |
| Ext.create('Ext.Panel', {  width: 500,  height: 300,  title: "FormLayout Panel",  layout: 'form',  renderTo: Ext.getBody(),  bodyPadding: 5,  defaultType: 'textfield',  items: [{  fieldLabel: 'First Name',  name: 'first',  allowBlank:**false**  },{  fieldLabel: 'Last Name',  name: 'last'  },{  fieldLabel: 'Company',  name: 'company'  }, {  fieldLabel: 'Email',  name: 'email',  vtype:'email'  }, {  fieldLabel: 'DOB',  name: 'dob',  xtype: 'datefield'  }, {  fieldLabel: 'Age',  name: 'age',  xtype: 'numberfield',  minValue: 0,  maxValue: 100  }, {  xtype: 'timefield',  fieldLabel: 'Time',  name: 'time',  minValue: '8:00am',  maxValue: '6:00pm'  }]  }); |

# Window

## window

Ext.window.Window

Xtype:window



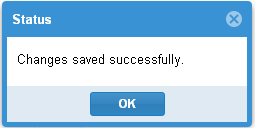
|  |
| --- |
| Ext.create('Ext.Window', {  title: 'Left Header, plain: true',  width: 400,  height: 200,  x: 10,  y: 20,  plain: **true**,  headerPosition: 'left',//标题的位置，如right,top,bottom  layout: 'fit',  items: {  border: **false**  }  }).show();  **var** constrainedWin, constrainedWin2;  **var** win2 = Ext.create('widget.window', {  height: 200,  width: 400,  x: 450,  y: 20,  title: 'Constraining Window, plain: true',  closable: **false**,  plain: **true**,  layout: 'fit',  items: [  constrainedWin = Ext.create('Ext.Window', {  title: 'Constrained Window',  width: 200,  height: 100,  x: 1000,  y: 20,  constrain: **true**,  layout: 'fit',  items: {  border: **false**  }  }),  constrainedWin2 = Ext.create('Ext.Window', {  title: 'Header-Constrained Win',  width: 200,  height: 100,  x: 120,  y: 120,  constrainHeader: **true**,  layout: 'fit',  items: {  border: **false**  }  }),  {  border: **false**  }  ]  });  win2.show();  constrainedWin.show();  constrainedWin2.show(); |

## MessageBox

Ext.window.MessageBox

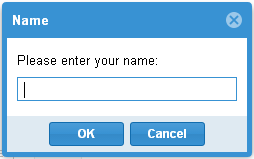
xtype: messagebox

### alert（提示框）



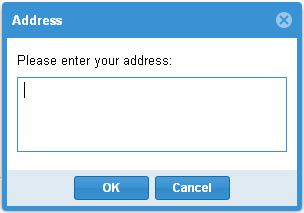
|  |
| --- |
| Ext.Msg.alert('Status', 'Changes saved successfully.'); |

### Prompt（单行文字）



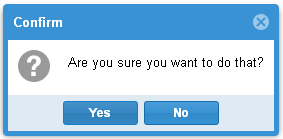
|  |
| --- |
| Ext.Msg.prompt('Name', 'Please enter your name:', **function**(btn, text){  **if** (btn == 'ok'){  alert(btn);  alert(text);  }  }); |

### Prompt Multi-line（多行文字）



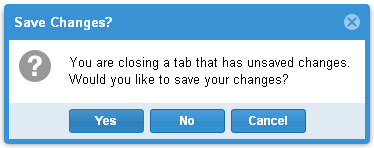
|  |
| --- |
| **function** showResultText(btn, text){  alert(btn);  alert(text);  }  Ext.MessageBox.show({  title: 'Address',  msg: 'Please enter your address:',  width:300,  buttons: Ext.MessageBox.OKCANCEL,  multiline: **true**,  fn: showResultText  }); |

### Confirm（是、否）



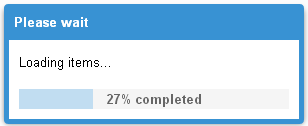
|  |
| --- |
| **function** showResult(btn){  alert(btn);  }  Ext.MessageBox.confirm('Confirm', 'Are you sure you want to do that?', showResult); |

### Yes/No/Cancel（是、否、取消）



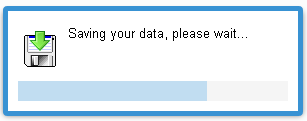
|  |
| --- |
| **function** showResult(btn){  alert(btn);  }  Ext.MessageBox.show({  title:'Save Changes?',  msg: 'You are closing a tab that has unsaved changes. <br />Would you like to save your changes?',  buttons: Ext.MessageBox.YESNOCANCEL,  fn: showResult,  icon: Ext.MessageBox.QUESTION  }); |

### Progress Dialog（进度条）



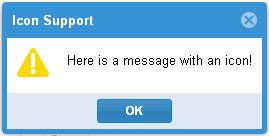
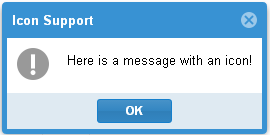
|  |
| --- |
| Ext.MessageBox.show({  title: 'Please wait',  msg: 'Loading items...',  progressText: 'Initializing...',  width:300,  progress:**true**,//是否显示进度条  closable:**true**,  });  **var** f = **function**(v){  **return** **function**(){  **if**(v == 12){  Ext.MessageBox.hide();  }**else**{  **var** i = v/11;  Ext.MessageBox.updateProgress(i, Math.round(100\*i)+'% completed');  }  };  };  **for**(**var** i = 1; i < 13; i++){  setTimeout(f(i), i\*500);  } |

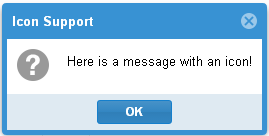
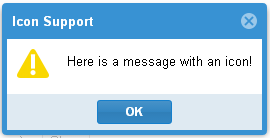
### Wait Dialog（等待对话框）



|  |
| --- |
| Ext.MessageBox.show({  msg: 'Saving your data, please wait...',  progressText: 'Saving...',  width:300,  wait:**true**,  waitConfig: {interval:200},  icon:'ext-mb-download', //custom class in msg-box.html  iconHeight: 50  });  setTimeout(**function**(){  Ext.MessageBox.hide();  }, 8000); |

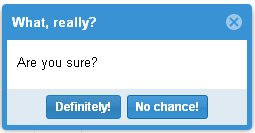
### Icons（图标）

|  |
| --- |
| **function** showResult(btn){  alert(btn);  }  Ext.MessageBox.show({  title: 'Icon Support',  msg: 'Here is a message with an icon!',  buttons: Ext.MessageBox.OK,  fn: showResult,  icon: Ext.MessageBox.INFO  //icon: Ext.MessageBox.QUESTION  //icon: Ext.MessageBox.WARNING  //icon: Ext.MessageBox.ERROR  }); |

### Custom button text（自定义按钮文字）

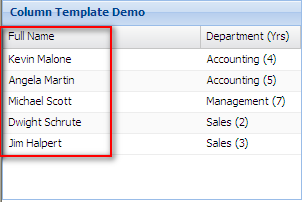


|  |
| --- |
| **function** showResultText(btn, text){  alert(btn);  alert(text);  }  Ext.MessageBox.show({  title: 'What, really?',  msg: 'Are you sure?',  buttons: Ext.MessageBox.YESNO,  buttonText:{  yes: "Definitely!",  no: "No chance!"  },  fn: showResult  }); |

# Grid表单

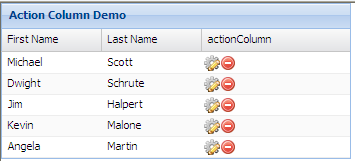
## Column（列）

### Template（合并显示多个数据列）



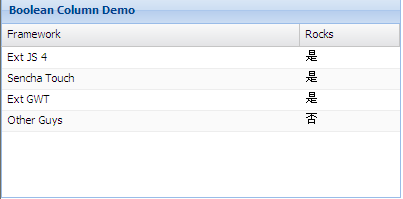
|  |
| --- |
| Ext.create('Ext.data.Store', {  storeId:'employeeStore',  fields:['firstname', 'lastname', 'seniority', 'department'],  groupField: 'department',  data:[  { firstname: "Michael", lastname: "Scott", seniority: 7, department: "Management" },  { firstname: "Dwight", lastname: "Schrute", seniority: 2, department: "Sales" },  { firstname: "Jim", lastname: "Halpert", seniority: 3, department: "Sales" },  { firstname: "Kevin", lastname: "Malone", seniority: 4, department: "Accounting" },  { firstname: "Angela", lastname: "Martin", seniority: 5, department: "Accounting" }  ]  });  Ext.create('Ext.grid.Panel', {  title: 'Column Template Demo',  store: Ext.data.StoreManager.lookup('employeeStore'),  columns: [  //firstname,lastname两列合并在Full Name列显示  { text: 'Full Name', xtype: 'templatecolumn', tpl: '{firstname} {lastname}', flex:1 },  { text: 'Department (Yrs)', xtype: 'templatecolumn', tpl: '{department} ({seniority})' }  ],  height: 200,  width: 300,  renderTo: Ext.getBody()  }); |

### actionColumn（列中画图标，并可加事件）



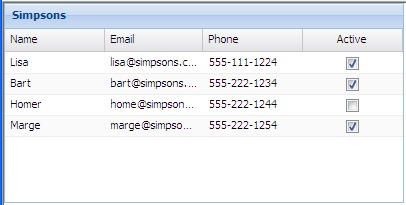
|  |
| --- |
| Ext.create('Ext.data.Store', {  storeId:'employeeStore',  fields:['firstname', 'lastname', 'seniority', 'dep', 'hired'],  data:[  {firstname:"Michael", lastname:"Scott"},  {firstname:"Dwight", lastname:"Schrute"},  {firstname:"Jim", lastname:"Halpert"},  {firstname:"Kevin", lastname:"Malone"},  {firstname:"Angela", lastname:"Martin"}  ]  });  Ext.create('Ext.grid.Panel', {  title: 'Action Column Demo',  store: Ext.data.StoreManager.lookup('employeeStore'),  columns: [  {text: 'First Name', dataIndex:'firstname'},  {text: 'Last Name', dataIndex:'lastname'},  {  text: 'actionColumn',  xtype:'actioncolumn',  width:150,  items: [{  icon: '../examples/shared/icons/fam/cog\_edit.png', // Use a URL in the icon config  tooltip: 'Edit',  handler: **function**(grid, rowIndex, colIndex) {  **var** rec = grid.getStore().getAt(rowIndex);  alert("Edit " + rec.get('firstname'));  }  },{  icon: '../examples/restful/images/delete.png',  tooltip: 'Delete',  handler: **function**(grid, rowIndex, colIndex) {  **var** rec = grid.getStore().getAt(rowIndex);  alert("Terminate " + rec.get('firstname'));  }  }]  }  ],  width: 350,  renderTo: Ext.getBody()  }); |

### booleanColumn（把true,false改变显示值）



|  |
| --- |
| Ext.create('Ext.data.Store', {  storeId:'sampleStore',  fields:[  {name: 'framework', type: 'string'},  {name: 'rocks', type: 'boolean'}  ],  data:{'items':[  { 'framework': "Ext JS 4", 'rocks': **true** },  { 'framework': "Sencha Touch", 'rocks': **true** },  { 'framework': "Ext GWT", 'rocks': **true** },  { 'framework': "Other Guys", 'rocks': **false** }  ]},  proxy: {  type: 'memory',  reader: {  type: 'json',  root: 'items'  }  }  });  Ext.create('Ext.grid.Panel', {  title: 'Boolean Column Demo',  store: Ext.data.StoreManager.lookup('sampleStore'),  columns: [  { text: 'Framework', dataIndex: 'framework', flex: 1 },  {  xtype: 'booleancolumn',  text: 'Rocks',  trueText: '是',  falseText: '否',  dataIndex: 'rocks'  }  ],  height: 200,  width: 400,  renderTo: Ext.getBody()  }); |

### CheckColumn



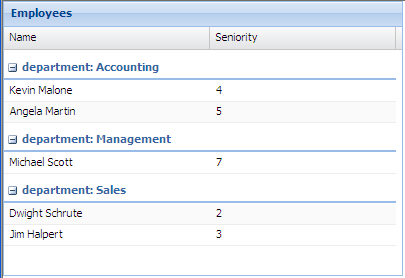
|  |
| --- |
| **var** store = Ext.create('Ext.data.Store', {  fields : ['name', 'email', 'phone', 'active'],  data : {  items : [  { name : 'Lisa', email : 'lisa@simpsons.com', phone : '555-111-1224', active : **true** },  { name : 'Bart', email : 'bart@simpsons.com', phone : '555-222-1234', active : **true** },  { name : 'Homer', email : 'home@simpsons.com', phone : '555-222-1244', active : **false** },  { name : 'Marge', email : 'marge@simpsons.com', phone : '555-222-1254', active : **true** }  ]  },  proxy : {  type : 'memory',  reader : {  type : 'json',  root : 'items'  }  }  });  Ext.create('Ext.grid.Panel', {  title : 'Simpsons',  height : 200,  width : 400,  renderTo : Ext.getBody(),  store : store,  columns : [  { text : 'Name', dataIndex : 'name' },  { text : 'Email', dataIndex : 'email', flex : 1 },  { text : 'Phone', dataIndex : 'phone' },  { xtype : 'checkcolumn', text : 'Active', dataIndex : 'active' }  ]  }); |

### 日期、数字、行号列

|  |
| --- |
| Ext.create('Ext.data.Store', {  storeId:'sampleStore',  fields:[  { name: 'symbol', type: 'string' },  { name: 'date', type: 'date' },  { name: 'change', type: 'number' },  { name: 'volume', type: 'number' },  { name: 'topday', type: 'date' }  ],  data:[  { symbol: "msft", date: '2011/04/22', change: 2.43, volume: 61606325, topday: '04/01/2010' },  { symbol: "goog", date: '2011/04/22', change: 0.81, volume: 3053782, topday: '04/11/2010' },  { symbol: "apple", date: '2011/04/22', change: 1.35, volume: 24484858, topday: '04/28/2010' },  { symbol: "sencha", date: '2011/04/22', change: 8.85, volume: 5556351, topday: '04/22/2010' }  ]  });  Ext.create('Ext.grid.Panel', {  title: 'Date Column Demo',  store: Ext.data.StoreManager.lookup('sampleStore'),  columns: [  {xtype: 'rownumberer'},//行号  { text: 'Symbol', dataIndex: 'symbol', flex: 1 },  { text: 'Date', dataIndex: 'date', xtype: 'datecolumn', format:'Y-m-d' },//日期格式  { text: 'Change', dataIndex: 'change', xtype: 'numbercolumn', format:'0.00' },//数字格式  { text: 'Volume', dataIndex: 'volume', xtype: 'numbercolumn', format:'0,000' },  { text: 'Top Day', dataIndex: 'topday', xtype: 'datecolumn', format:'l' } //星期格式  ],  height: 200,  width: 450,  renderTo: Ext.getBody()  }); |

## Feature（特色）

### Grouping（分组）

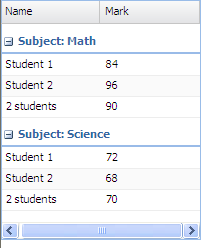


|  |
| --- |
| **var** store = Ext.create('Ext.data.Store', {  storeId:'employeeStore',  fields:['name', 'seniority', 'department'],  groupField: 'department',  data: {'employees':[  { "name": "Michael Scott", "seniority": 7, "department": "Management" },  { "name": "Dwight Schrute", "seniority": 2, "department": "Sales" },  { "name": "Jim Halpert", "seniority": 3, "department": "Sales" },  { "name": "Kevin Malone", "seniority": 4, "department": "Accounting" },  { "name": "Angela Martin", "seniority": 5, "department": "Accounting" }  ]},  proxy: {  type: 'memory',  reader: {  type: 'json',  root: 'employees'  }  }  });  Ext.create('Ext.grid.Panel', {  title: 'Employees',  store: Ext.data.StoreManager.lookup('employeeStore'),  columns: [  { text: 'Name', dataIndex: 'name' },  { text: 'Seniority', dataIndex: 'seniority' }  ],  features: [{ftype:'grouping'}],  width: 400,  height: 275,  renderTo: Ext.getBody()  }); |

### Summary（统计）

|  |
| --- |
| Ext.define('TestResult', {  extend: 'Ext.data.Model',  fields: ['student', {  name: 'mark',  type: 'int'  }]  });  Ext.create('Ext.grid.Panel', {  width: 400,  height: 200,  title: 'Summary Test',  style: 'padding: 20px',  renderTo: document.body,  features: [{  ftype: 'summary'  }],  store: {  model: 'TestResult',  data: [{  student: 'Student 1',  mark: 84  },{  student: 'Student 2',  mark: 72  },{  student: 'Student 3',  mark: 96  },{  student: 'Student 4',  mark: 68  }]  },  columns: [{  dataIndex: 'student',  text: 'Name',  summaryType: 'count',//统计Name列的个数    //统计行，对应学生列的显示值  //value统计的个数  summaryRenderer: **function**(value, summaryData, dataIndex) {  **return** Ext.String.format('{0} student{1}', value, value !== 1 ? 's' : '');  }  }, {  dataIndex: 'mark',  text: 'Mark',  summaryType: 'average'//统计Mark列的平均值  }]  }); |

### groupingSummary（分组统计）



|  |
| --- |
| Ext.define('TestResult', {  extend: 'Ext.data.Model',  fields: ['student', 'subject', {  name: 'mark',  type: 'int'  }]  });  Ext.create('Ext.grid.Panel', {  width: 200,  height: 240,  renderTo: document.body,  features: [{  //分组列表头的表达式  groupHeaderTpl: 'Subject: {name}',  //分组统计  ftype: 'groupingsummary'  }],  store: {  model: 'TestResult',  groupField: 'subject',//分组字段  data: [{  student: 'Student 1',  subject: 'Math',  mark: 84  },{  student: 'Student 1',  subject: 'Science',  mark: 72  },{  student: 'Student 2',  subject: 'Math',  mark: 96  },{  student: 'Student 2',  subject: 'Science',  mark: 68  }]  },  columns: [{  dataIndex: 'student',  text: 'Name',  //统计学生数量  summaryType: 'count',  //统计行，对应学生列的显示值  //value统计的个数  summaryRenderer: **function**(value){  **return** Ext.String.format('{0} student{1}', value, value !== 1 ? 's' : '');  }  }, {  dataIndex: 'mark',  text: 'Mark',  summaryType: 'average'//统计平均值  }]  }); |

## Plugin（插件）

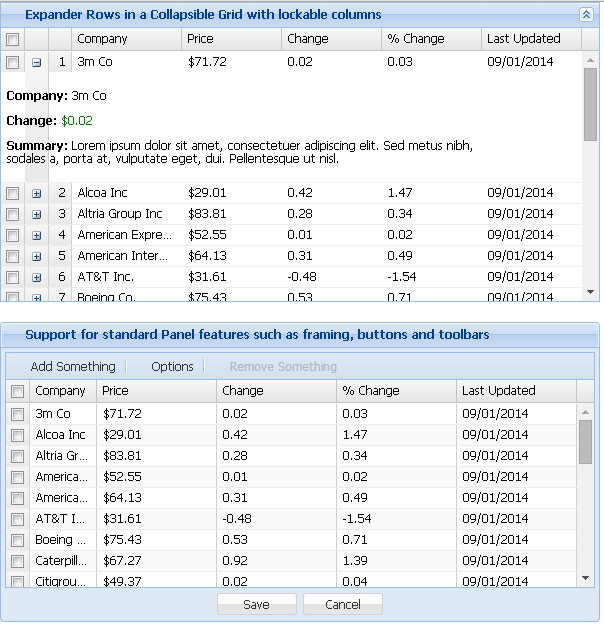
### CellEditing（可编辑单元格）

|  |
| --- |
| Ext.create('Ext.data.Store', {  storeId:'simpsonsStore',  fields:['name', 'email', 'phone','male','age','availDate','isChinese'],  data:{'items':[  {"name":"Lisa", "email":"lisa@simpsons.com", "phone":"555-111-1224", male:0,age:18, availDate:**new** Date(), isChinese:**true**},  {"name":"Bart", "email":"bart@simpsons.com", "phone":"555-222-1234", male:1 ,age:15, availDate:**new** Date(), isChinese:**true**},  {"name":"Homer", "email":"home@simpsons.com", "phone":"555-222-1244", male:0 ,age:14, availDate:**new** Date(), isChinese:**false**},  {"name":"Marge", "email":"marge@simpsons.com", "phone":"555-222-1254", male:0 ,age:12, availDate:**new** Date(), isChinese:**true**}  ]},  proxy: {  type: 'memory',  reader: {  type: 'json',  root: 'items'  }  }  });  Ext.create('Ext.grid.Panel', {  title: 'Simpsons',  store: Ext.data.StoreManager.lookup('simpsonsStore'),  columns: [  //文本  {header: 'Name', dataIndex: 'name', editor: 'textfield'},  //文本、非空验证  {header: 'Email', dataIndex: 'email', flex:1,  editor: {  xtype: 'textfield',  allowBlank: **false**  }  },  {header: 'Phone', dataIndex: 'phone'} ,  //下拉框  {  header: '性别',  dataIndex: 'male',  width: 40,  editor: **new** Ext.form.field.ComboBox({  typeAhead: **true**,  triggerAction: 'all',  store: [  [0,'女'],  [1,'男']  ]  }),  renderer: **function** (val){  **if**(val==0){  **return** "女";  }**else**{  **return** "男";  }  }  },  //数值  {  header: '年龄',  dataIndex: 'age',  width: 70,  align: 'right',  editor: {  xtype: 'numberfield',  allowBlank: **false**,  minValue: 0,  maxValue: 200  }  },  //日期  {  header: '有效期',  dataIndex: 'availDate',  width: 95,  renderer: Ext.util.Format.dateRenderer('M d, Y'),  editor: {  xtype: 'datefield',  format: 'm/d/y',  minValue: '01/01/06',  disabledDays: [0, 6],  disabledDaysText: 'Plants are not available on the weekends'  }  },  //checkbox  {  xtype: 'checkcolumn',  header: '中国人？',  dataIndex: 'isChinese',  width: 90,  stopSelection: **false**  },  //action伪列  {  xtype: 'actioncolumn',  width: 30,  sortable: **false**,  menuDisabled: **true**,  items: [{  icon: '../examples/shared/icons/fam/delete.gif',  tooltip: 'Delete',  scope: **this**,  handler: **function**(grid, rowIndex){  grid.getStore().removeAt(rowIndex);  }  }]  }  ],    selType: 'cellmodel',  //工具栏  tbar: [{  text: 'Add Plant',  scope: **this**,  handler: **function**(){  alert('add');  }  }],  //编辑触发次数  plugins: [  Ext.create('Ext.grid.plugin.CellEditing', {  clicksToEdit: 1  })  ],    height: 200,  width: 800,  renderTo: Ext.getBody()  }); |

### RowEditing（可编辑行）

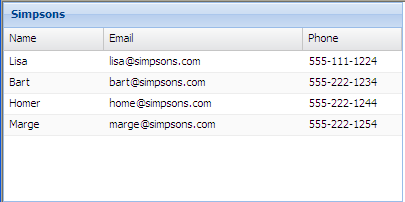
|  |
| --- |
| Ext.create('Ext.data.Store', {  storeId:'simpsonsStore',  fields:['name', 'email', 'phone'],  data: [  {"name":"Lisa", "email":"lisa@simpsons.com", "phone":"555-111-1224"},  {"name":"Bart", "email":"bart@simpsons.com", "phone":"555-222-1234"},  {"name":"Homer", "email":"home@simpsons.com", "phone":"555-222-1244"},  {"name":"Marge", "email":"marge@simpsons.com", "phone":"555-222-1254"}  ]  });  Ext.create('Ext.grid.Panel', {  title: 'Simpsons',  store: Ext.data.StoreManager.lookup('simpsonsStore'),  columns: [  {header: 'Name', dataIndex: 'name', editor: 'textfield'},  {header: 'Email', dataIndex: 'email', flex:1,  editor: {  xtype: 'textfield',  allowBlank: false  }  },  {header: 'Phone', dataIndex: 'phone'}  ],  selType: 'rowmodel',  plugins: [  Ext.create('Ext.grid.plugin.RowEditing', {  clicksToEdit: 1  })  ],  height: 200,  width: 400,  renderTo: Ext.getBody()  }); |

### Checkbox选中事件、扩展显示列、增加按钮



|  |
| --- |
| Ext.define('Company', {  extend: 'Ext.data.Model',  fields: [  {name: 'company'},  {name: 'price', type: 'float'},  {name: 'change', type: 'float'},  {name: 'pctChange', type: 'float'},  {name: 'lastChange', type: 'date', dateFormat: 'n/j h:ia'},  {name: 'industry'},  {name: 'desc'}  ]  });  // Array data for the grids  Ext.grid.dummyData = [  ['3m Co',71.72,0.02,0.03,'9/1 12:00am', 'Manufacturing'],  ['Alcoa Inc',29.01,0.42,1.47,'9/1 12:00am', 'Manufacturing'],  ['Altria Group Inc',83.81,0.28,0.34,'9/1 12:00am', 'Manufacturing'],  ['American Express Company',52.55,0.01,0.02,'9/1 12:00am', 'Finance'],  ['American International Group, Inc.',64.13,0.31,0.49,'9/1 12:00am', 'Services'],  ['AT&T Inc.',31.61,-0.48,-1.54,'9/1 12:00am', 'Services'],  ['Boeing Co.',75.43,0.53,0.71,'9/1 12:00am', 'Manufacturing'],  ['Caterpillar Inc.',67.27,0.92,1.39,'9/1 12:00am', 'Services'],  ['Citigroup, Inc.',49.37,0.02,0.04,'9/1 12:00am', 'Finance'],  ['E.I. du Pont de Nemours and Company',40.48,0.51,1.28,'9/1 12:00am', 'Manufacturing'],  ['Exxon Mobil Corp',68.1,-0.43,-0.64,'9/1 12:00am', 'Manufacturing'],  ['General Electric Company',34.14,-0.08,-0.23,'9/1 12:00am', 'Manufacturing'],  ['General Motors Corporation',30.27,1.09,3.74,'9/1 12:00am', 'Automotive'],  ['Hewlett-Packard Co.',36.53,-0.03,-0.08,'9/1 12:00am', 'Computer'],  ['Honeywell Intl Inc',38.77,0.05,0.13,'9/1 12:00am', 'Manufacturing'],  ['Intel Corporation',19.88,0.31,1.58,'9/1 12:00am', 'Computer'],  ['International Business Machines',81.41,0.44,0.54,'9/1 12:00am', 'Computer'],  ['Johnson & Johnson',64.72,0.06,0.09,'9/1 12:00am', 'Medical'],  ['JP Morgan & Chase & Co',45.73,0.07,0.15,'9/1 12:00am', 'Finance'],  ['McDonald\'s Corporation',36.76,0.86,2.40,'9/1 12:00am', 'Food'],  ['Merck & Co., Inc.',40.96,0.41,1.01,'9/1 12:00am', 'Medical'],  ['Microsoft Corporation',25.84,0.14,0.54,'9/1 12:00am', 'Computer'],  ['Pfizer Inc',27.96,0.4,1.45,'9/1 12:00am', 'Medical'],  ['The Coca-Cola Company',45.07,0.26,0.58,'9/1 12:00am', 'Food'],  ['The Home Depot, Inc.',34.64,0.35,1.02,'9/1 12:00am', 'Retail'],  ['The Procter & Gamble Company',61.91,0.01,0.02,'9/1 12:00am', 'Manufacturing'],  ['United Technologies Corporation',63.26,0.55,0.88,'9/1 12:00am', 'Computer'],  ['Verizon Communications',35.57,0.39,1.11,'9/1 12:00am', 'Services'],  ['Wal-Mart Stores, Inc.',45.45,0.73,1.63,'9/1 12:00am', 'Retail'],  ['Walt Disney Company (The) (Holding Company)',29.89,0.24,0.81,'9/1 12:00am', 'Services']  ];  // add in some dummy descriptions  **for**(**var** i = 0; i < Ext.grid.dummyData.length; i++){  Ext.grid.dummyData[i].push('Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed metus nibh, sodales a, porta at, vulputate eget, dui. Pellentesque ut nisl. ');  }  Ext.QuickTips.init();  **var** getLocalStore = **function**() {  **return** Ext.create('Ext.data.ArrayStore', {  model: 'Company',  data: Ext.grid.dummyData  });  };  **var** grid1 = Ext.create('Ext.grid.Panel', {  store: getLocalStore(),  columns: [  Ext.create('Ext.grid.RowNumberer'),//行号  {text: "Company", flex: 1, dataIndex: 'company'},  {text: "Price", renderer: Ext.util.Format.usMoney, dataIndex: 'price'},  {text: "Change", dataIndex: 'change'},  {text: "% Change", dataIndex: 'pctChange'},  {text: "Last Updated", renderer: Ext.util.Format.dateRenderer('m/d/Y'), dataIndex: 'lastChange'}  ],  selType: 'checkboxmodel',//首列增加checkbox  columnLines: **false**,//列与列之间的分隔线  enableLocking: **true**,  width: 600,  height: 300,  plugins: [{//扩展显示记录  ptype: 'rowexpander',  rowBodyTpl : **new** Ext.XTemplate(  '<p><b>Company:</b> {company}</p>',  '<p><b>Change:</b> {change:this.formatChange}</p>',  '<p><b>Summary:</b> {desc}</p>',  {  formatChange: **function**(v){  **var** color = v >= 0 ? 'green' : 'red';  **return** '<span style="color: ' + color + ';">' + Ext.util.Format.usMoney(v) + '</span>';  }  })  }],  collapsible: **true**,  animCollapse: **true**,  title: 'Expander Rows in a Collapsible Grid with lockable columns',  iconCls: 'icon-grid',  margin: '0 0 20 0',  renderTo: Ext.getBody()  });  **var** selModel = Ext.create('Ext.selection.CheckboxModel', {  listeners: {  selectionchange: **function**(sm, selections) {  grid4.down('#removeButton').setDisabled(selections.length === 0);  }  }  });    **var** grid4 = Ext.create('Ext.grid.Panel', {  id:'button-grid',  store: getLocalStore(),  columns: [  {text: "Company", flex: 1, sortable: **true**, dataIndex: 'company'},  {text: "Price", width: 120, sortable: **true**, renderer: Ext.util.Format.usMoney, dataIndex: 'price'},  {text: "Change", width: 120, sortable: **true**, dataIndex: 'change'},  {text: "% Change", width: 120, sortable: **true**, dataIndex: 'pctChange'},  {text: "Last Updated", width: 120, sortable: **true**, renderer: Ext.util.Format.dateRenderer('m/d/Y'), dataIndex: 'lastChange'}  ],  columnLines: **true**,  selModel: selModel,//checkbox选中事件  // inline buttons  dockedItems: [{//底部增加按钮  xtype: 'toolbar',  dock: 'bottom',  ui: 'footer',  layout: {  pack: 'center'  },  items: [{  minWidth: 80,  text: 'Save'  },{  minWidth: 80,  text: 'Cancel'  }]  }, {//头部增加按钮  xtype: 'toolbar',  items: [{  text:'Add Something',  tooltip:'Add a new row',  iconCls:'add'  }, '-', {  text:'Options',  tooltip:'Set options',  iconCls:'option'  },'-',{  itemId: 'removeButton',  text:'Remove Something',  tooltip:'Remove the selected item',  iconCls:'remove',  disabled: **true**  }]  }],  width: 600,  height: 300,  frame: **true**,  title: 'Support for standard Panel features such as framing, buttons and toolbars',  iconCls: 'icon-grid',  renderTo: Ext.getBody()  }); |

## Ext.grid.Panel（简单表单）



|  |
| --- |
| **var** store1 = Ext.create('Ext.data.Store', {  storeId:'simpsonsStore',//Store在Ext.data.StoreManager中的ID  fields:['name', 'email', 'phone'],//列  data:{'items':[  { 'name': 'Lisa', "email":"lisa@simpsons.com", "phone":"555-111-1224" },  { 'name': 'Bart', "email":"bart@simpsons.com", "phone":"555-222-1234" },  { 'name': 'Homer', "email":"home@simpsons.com", "phone":"555-222-1244" },  { 'name': 'Marge', "email":"marge@simpsons.com", "phone":"555-222-1254" }  ]},  proxy: {  type: 'memory', //存储于内存中  reader: {  type: 'json',  root: 'items'  }  }  });  Ext.create('Ext.grid.Panel', {  title: 'Simpsons',  //从StoreManager取出Ext.data.Store  store: Ext.data.StoreManager.lookup('simpsonsStore'),  //通过变量  //store: store1,  columns: [  //text是列名  //dataIndex是与Ext.data.Store相对应  { text: '姓名', dataIndex: 'name', flex: 1 },  { text: '邮箱', dataIndex: 'email', flex: 2 },  { text: '电话', dataIndex: 'phone', flex: 1 }  ],  height: 200,  width: 400,  renderTo: Ext.getBody()  }); |

# Form（表单）

## 简单例子

|  |
| --- |
| **var** simple = Ext.create('Ext.form.Panel', {  url:'save-form.php',  frame:**true**,  title: 'Simple Form',  bodyStyle:'padding:5px 5px 0',  width: 350,  fieldDefaults: {  msgTarget: 'side',  labelWidth: 75  },  defaultType: 'textfield',  defaults: {  anchor: '100%'  },  items: [{  fieldLabel: 'First Name',  name: 'first',  allowBlank:**false**  },{  fieldLabel: 'Last Name',  name: 'last'  },{  fieldLabel: 'Company',  name: 'company'  }, {  fieldLabel: 'Email',  name: 'email',  vtype:'email'  }, {  xtype: 'timefield',  fieldLabel: 'Time',  name: 'time',  minValue: '8:00am',  maxValue: '6:00pm'  }],  buttons: [{  text: 'Save'  },{  text: 'Cancel'  }]  });  simple.render(document.body); |

## 字段集（fieldsSet）

|  |
| --- |
| **var** fsf = Ext.create('Ext.form.Panel', {  url:'save-form.php',  frame:**true**,  title: 'Simple Form with FieldSets',  bodyStyle:'padding:5px 5px 0',  width: 350,  fieldDefaults: {  msgTarget: 'side',  labelWidth: 75  },  defaults: {  anchor: '100%'  },  items: [{  xtype:'fieldset',  checkboxToggle:**true**,//能否通过复选框控制是否显示set  title: 'User Information',  defaultType: 'textfield',  collapsed: **true**,  layout: 'anchor',  defaults: {  anchor: '100%'  },  items :[{  fieldLabel: 'First Name',  name: 'first',  allowBlank:**false**  },{  fieldLabel: 'Last Name',  name: 'last'  },{  fieldLabel: 'Company',  name: 'company'  }, {  fieldLabel: 'Email',  name: 'email',  vtype:'email'  }]  },{  xtype:'fieldset',  title: 'Phone Number',  collapsible: **true**,  defaultType: 'textfield',  layout: 'anchor',  defaults: {  anchor: '100%'  },  items :[{  fieldLabel: 'Home',  name: 'home',  value: '(888) 555-1212'  },{  fieldLabel: 'Business',  name: 'business'  },{  fieldLabel: 'Mobile',  name: 'mobile'  },{  fieldLabel: 'Fax',  name: 'fax'  }]  }],  buttons: [{  text: 'Save'  },{  text: 'Cancel'  }]  });  fsf.render(document.body); |

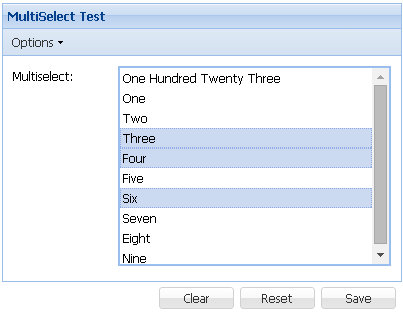
## Tabpanel布局

|  |
| --- |
| **var** tabs = Ext.create('Ext.form.Panel', {  width: 350,  border: **false**,  bodyBorder: **false**,  fieldDefaults: {  labelWidth: 75,  msgTarget: 'side'  },  defaults: {  anchor: '100%'  },  items: {  xtype:'tabpanel',  activeTab: 0,  defaults:{  bodyStyle:'padding:10px'  },  items:[{  title:'Personal Details',  defaultType: 'textfield',  items: [{  fieldLabel: 'First Name',  name: 'first',  allowBlank:**false**,  value: 'Ed'  },{  fieldLabel: 'Last Name',  name: 'last',  value: 'Spencer'  },{  fieldLabel: 'Company',  name: 'company',  value: 'Ext JS'  }, {  fieldLabel: 'Email',  name: 'email',  vtype:'email'  }]  },{  title:'Phone Numbers',  defaultType: 'textfield',  items: [{  fieldLabel: 'Home',  name: 'home',  value: '(888) 555-1212'  },{  fieldLabel: 'Business',  name: 'business'  },{  fieldLabel: 'Mobile',  name: 'mobile'  },{  fieldLabel: 'Fax',  name: 'fax'  }]  }]  },  buttons: [{  text: 'Save'  },{  text: 'Cancel'  }]  });  tabs.render(document.body); |

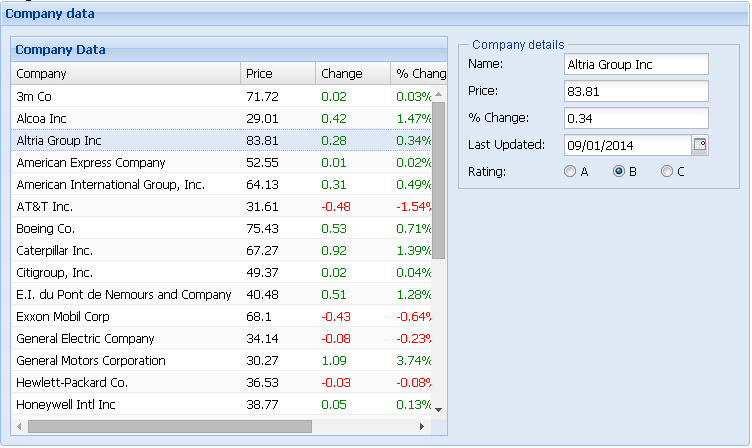
## 复杂布局

|  |
| --- |
| **var** tab2 = Ext.create('Ext.form.Panel', {  title: 'Inner Tabs',  bodyStyle:'padding:5px',  width: 600,  fieldDefaults: {  labelAlign: 'top',  msgTarget: 'side'  },  defaults: {  anchor: '100%'  },  items: [{  layout:'column',  border:**false**,  items:[{  columnWidth:.5,  border:**false**,  layout: 'anchor',  defaultType: 'textfield',  items: [{  fieldLabel: 'First Name',  name: 'first',  anchor:'95%'  }, {  fieldLabel: 'Company',  name: 'company',  anchor:'95%'  }]  },{  columnWidth:.5,  border:**false**,  layout: 'anchor',  defaultType: 'textfield',  items: [{  fieldLabel: 'Last Name',  name: 'last',  anchor:'95%'  },{  fieldLabel: 'Email',  name: 'email',  vtype:'email',  anchor:'95%'  }]  }]  },{  xtype:'tabpanel',  plain:**true**,  activeTab: 0,  height:235,  defaults:{bodyStyle:'padding:10px'},  items:[{  title:'Personal Details',  defaults: {width: 230},  defaultType: 'textfield',  items: [{  fieldLabel: 'First Name',  name: 'first',  allowBlank:**false**,  value: 'Jamie'  },{  fieldLabel: 'Last Name',  name: 'last',  value: 'Avins'  },{  fieldLabel: 'Company',  name: 'company',  value: 'Ext JS'  }, {  fieldLabel: 'Email',  name: 'email',  vtype:'email'  }]  },{  title:'Phone Numbers',  defaults: {width: 230},  defaultType: 'textfield',  items: [{  fieldLabel: 'Home',  name: 'home',  value: '(888) 555-1212'  },{  fieldLabel: 'Business',  name: 'business'  },{  fieldLabel: 'Mobile',  name: 'mobile'  },{  fieldLabel: 'Fax',  name: 'fax'  }]  },{  cls: 'x-plain',  title: 'Biography',  layout: 'fit',  items: {  xtype: 'htmleditor',  name: 'bio2',  fieldLabel: 'Biography'  }  }]  }],  buttons: [{  text: 'Save'  },{  text: 'Cancel'  }]  });  tab2.render(document.body); |

## 多选（MultiSelect）



## 与grid联合展示

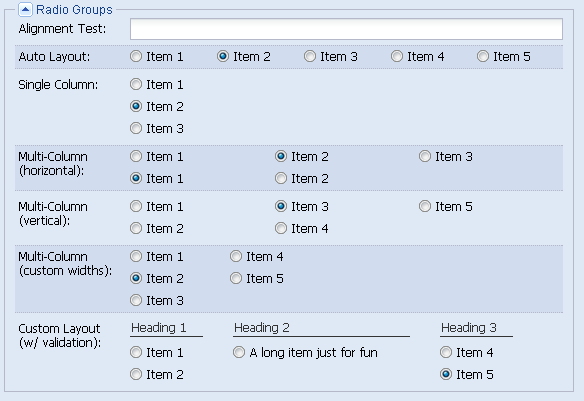


|  |
| --- |
| /\*  This file is part of Ext JS 4  Copyright (c) 2011 Sencha Inc  Contact: http://www.sencha.com/contact  Commercial Usage  Licensees holding valid commercial licenses may use this file in accordance with the Commercial Software License Agreement provided with the Software or, alternatively, in accordance with the terms contained in a written agreement between you and Sencha.  If you are unsure which license is appropriate for your use, please contact the sales department at http://www.sencha.com/contact.  \*/  Ext.require([  'Ext.form.\*',  'Ext.data.\*',  'Ext.grid.Panel',  'Ext.layout.container.Column'  ]);  Ext.onReady(**function**(){  Ext.QuickTips.init();  **var** bd = Ext.getBody();  // sample static data for the store  **var** myData = [  ['3m Co', 71.72, 0.02, 0.03, '9/1 12:00am'],  ['Alcoa Inc', 29.01, 0.42, 1.47, '9/1 12:00am'],  ['Altria Group Inc', 83.81, 0.28, 0.34, '9/1 12:00am'],  ['American Express Company', 52.55, 0.01, 0.02, '9/1 12:00am'],  ['American International Group, Inc.', 64.13, 0.31, 0.49, '9/1 12:00am'],  ['AT&T Inc.', 31.61, -0.48, -1.54, '9/1 12:00am'],  ['Boeing Co.', 75.43, 0.53, 0.71, '9/1 12:00am'],  ['Caterpillar Inc.', 67.27, 0.92, 1.39, '9/1 12:00am'],  ['Citigroup, Inc.', 49.37, 0.02, 0.04, '9/1 12:00am'],  ['E.I. du Pont de Nemours and Company', 40.48, 0.51, 1.28, '9/1 12:00am'],  ['Exxon Mobil Corp', 68.1, -0.43, -0.64, '9/1 12:00am'],  ['General Electric Company', 34.14, -0.08, -0.23, '9/1 12:00am'],  ['General Motors Corporation', 30.27, 1.09, 3.74, '9/1 12:00am'],  ['Hewlett-Packard Co.', 36.53, -0.03, -0.08, '9/1 12:00am'],  ['Honeywell Intl Inc', 38.77, 0.05, 0.13, '9/1 12:00am'],  ['Intel Corporation', 19.88, 0.31, 1.58, '9/1 12:00am'],  ['International Business Machines', 81.41, 0.44, 0.54, '9/1 12:00am'],  ['Johnson & Johnson', 64.72, 0.06, 0.09, '9/1 12:00am'],  ['JP Morgan & Chase & Co', 45.73, 0.07, 0.15, '9/1 12:00am'],  ['McDonald\'s Corporation', 36.76, 0.86, 2.40, '9/1 12:00am'],  ['Merck & Co., Inc.', 40.96, 0.41, 1.01, '9/1 12:00am'],  ['Microsoft Corporation', 25.84, 0.14, 0.54, '9/1 12:00am'],  ['Pfizer Inc', 27.96, 0.4, 1.45, '9/1 12:00am'],  ['The Coca-Cola Company', 45.07, 0.26, 0.58, '9/1 12:00am'],  ['The Home Depot, Inc.', 34.64, 0.35, 1.02, '9/1 12:00am'],  ['The Procter & Gamble Company', 61.91, 0.01, 0.02, '9/1 12:00am'],  ['United Technologies Corporation', 63.26, 0.55, 0.88, '9/1 12:00am'],  ['Verizon Communications', 35.57, 0.39, 1.11, '9/1 12:00am'],  ['Wal-Mart Stores, Inc.', 45.45, 0.73, 1.63, '9/1 12:00am']  ];  **var** ds = Ext.create('Ext.data.ArrayStore', {  fields: [  {name: 'company'},  {name: 'price', type: 'float'},  {name: 'change', type: 'float'},  {name: 'pctChange', type: 'float'},  {name: 'lastChange', type: 'date', dateFormat: 'n/j h:ia'},  // Rating dependent upon performance 0 = best, 2 = worst  {name: 'rating', type: 'int', convert: **function**(value, record) {  **var** pct = record.get('pctChange');  **if** (pct < 0) **return** 2;  **if** (pct < 1) **return** 1;  **return** 0;  }}  ],  data: myData  });  // example of custom renderer function  **function** change(val){  **if**(val > 0){  **return** '<span style="color:green;">' + val + '</span>';  }**else** **if**(val < 0){  **return** '<span style="color:red;">' + val + '</span>';  }  **return** val;  }  // example of custom renderer **function**  **function** pctChange(val){  **if**(val > 0){  **return** '<span style="color:green;">' + val + '%</span>';  }**else** **if**(val < 0){  **return** '<span style="color:red;">' + val + '%</span>';  }  **return** val;  }    // render rating as "A", "B" or "C" depending upon numeric value.  **function** rating(v) {  **if** (v == 0) **return** "A";  **if** (v == 1) **return** "B";  **if** (v == 2) **return** "C";  }  bd.createChild({tag: 'h2', html: 'Using a Grid with a Form'});  /\*  \* Here is where we create the Form  \*/  **var** gridForm = Ext.create('Ext.form.Panel', {  id: 'company-form',  frame: **true**,  title: 'Company data',  bodyPadding: 5,  width: 750,  layout: 'column', // Specifies that the items will now be arranged **in** columns  fieldDefaults: {  labelAlign: 'left',  msgTarget: 'side'  },  items: [{  columnWidth: 0.60,  xtype: 'gridpanel',  store: ds,  height: 400,  title:'Company Data',  columns: [  {  id :'company',  text : 'Company',  flex: 1,  sortable : **true**,  dataIndex: 'company'  },  {  text : 'Price',  width : 75,  sortable : **true**,  dataIndex: 'price'  },  {  text : 'Change',  width : 75,  sortable : **true**,  renderer : change,  dataIndex: 'change'  },  {  text : '% Change',  width : 75,  sortable : **true**,  renderer : pctChange,  dataIndex: 'pctChange'  },  {  text : 'Last Updated',  width : 85,  sortable : **true**,  renderer : Ext.util.Format.dateRenderer('m/d/Y'),  dataIndex: 'lastChange'  },  {  text: 'Rating',  width: 30,  sortable: **true**,  renderer: rating,  dataIndex: 'rating'  }  ],  listeners: {  selectionchange: **function**(model, records) {  **if** (records[0]) {  **this**.up('form').getForm().loadRecord(records[0]);  }  }  }  }, {  columnWidth: 0.4,  margin: '0 0 0 10',  xtype: 'fieldset',  title:'Company details',  defaults: {  width: 240,  labelWidth: 90  },  defaultType: 'textfield',  items: [{  fieldLabel: 'Name',  name: 'company'  },{  fieldLabel: 'Price',  name: 'price'  },{  fieldLabel: '% Change',  name: 'pctChange'  },{  xtype: 'datefield',  fieldLabel: 'Last Updated',  name: 'lastChange'  }, {  xtype: 'radiogroup',  fieldLabel: 'Rating',  columns: 3,  defaults: {  name: 'rating' //Each radio has the same name so the browser will make sure only one is checked at once  },  items: [{  inputValue: '0',  boxLabel: 'A'  }, {  inputValue: '1',  boxLabel: 'B'  }, {  inputValue: '2',  boxLabel: 'C'  }]  }]  }],  renderTo: bd  });  gridForm.child('gridpanel').getSelectionModel().select(0);  }); |

## 常用控件

|  |  |  |
| --- | --- | --- |
|  |  |  |
| xtype:’htmleditor’ | 简单的html编辑器 |  |
| xtype:'textfield' | 文本 |  |
| xtype:'fieldset' | 字段集 | Details中就是一个fieldset  xtype: 'fieldset',  title: 'Details',  collapsible: **true**,  defaults: {  labelWidth: 89,  anchor: '100%',  layout: {  type: 'hbox',  defaultMargins: {top: 0, right: 5, bottom: 0, left: 0}  }  },  items: [……] |
| xtype: 'timefield'  minValue: '1:30 AM',  maxValue: '9:15 PM' | 时间 |  |
| vtype:'email' | 邮箱 |  |
| fieldLabel | 字段标签 |  |
| xtype: 'textareafield' | 文本区 |  |
| xtype: 'displayfield' | 显示字段 | 如：  Display field <span style="color:green;">value</span> |
| xtype: 'textfield'  inputType: 'password' | 口令输入框 |  |
| xtype: 'numberfield'  value: 5,  minValue: 0,  maxValue: 50  allowDecimals: true,//是否允许有小数  decimalPrecision: 1,//保留几位小数  step: 0.4//每次增加减少的值  hideTrigger: **true**,//是否显示触发按钮 | 数值 | value: 默认,  minValue: 最小,  maxValue: 最大 |
| fieldDefaults: {  labelAlign: 'left',  labelWidth: 105,  anchor: '100%'  }, | 字段默认设置 |  |
| xtype: 'radiofield',  name: 'radio1',  boxLabel: 'radio 2' | 单选框 | {  xtype: 'radiofield',  name: 'radio1',  value: 'radiovalue1',  fieldLabel: 'Radio buttons',  boxLabel: 'radio 1'  }, {  xtype: 'radiofield',  name: 'radio1',  value: 'radiovalue2',  fieldLabel: '',  labelSeparator: '',  hideEmptyLabel: **false**,  boxLabel: 'radio 2'  } |
| xtype: 'checkboxfield'  boxLabel: 'box label2' | 复选框 | {  xtype: 'checkboxfield',  name: 'radio1',  value: 'radiovalue1',  fieldLabel: 'Radio buttons',  boxLabel: 'radio 1'  }, {  xtype: 'checkboxfield',  name: 'radio1',  value: 'radiovalue2',  fieldLabel: '',  labelSeparator: '',  hideEmptyLabel: **false**,  boxLabel: 'radio 2'  } |
| xtype: 'datefield' | 日期 |  |
| xtype: 'sliderfield',  fieldLabel: 'Sound Effects',  value: 50,  name: 'fx'  width: 200,  increment: 10,  minValue: 0,  maxValue: 100, | 滑块字段 |  |
| xtype: 'fieldcontainer',  //容器的标签  fieldLabel: 'Phone',  //容器的错误信息是否做为一个整体显示  combineErrors: **true**,  //错误信息显示位置  msgTarget: 'under',  defaults: {  hideLabel: **true**  },  items: [//容器中的元素] | 字段容器 | xtype: 'fieldcontainer',  fieldLabel: 'Phone',//容量的标签  combineErrors: **true**,//容器的错误信息是否做为一个整体显示  msgTarget: 'under',//错误信息显示位置  defaults: {  hideLabel: **true**  },  items: [//容器中的元素  {xtype: 'displayfield', value: '('},  {xtype: 'textfield', fieldLabel: 'Phone 1', name: 'phone-1', width: 29, allowBlank: **false**},  {xtype: 'displayfield', value: ')'},  {xtype: 'textfield', fieldLabel: 'Phone 2', name: 'phone-2', width: 29, allowBlank: **false**, margins: '0 5 0 0'},  {xtype: 'displayfield', value: '-'},  {xtype: 'textfield', fieldLabel: 'Phone 3', name: 'phone-3', width: 48, allowBlank: **false**}  ] |
| xtype: 'checkboxgroup' | 复选框组 |  |
| xtype: 'radiogroup' | 单选框组 |  |

### 单、复选框组布局

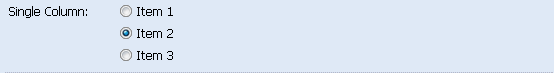


#### Auto布局



|  |
| --- |
| {  xtype: 'radiogroup',  fieldLabel: 'Auto Layout',  cls: 'x-check-group-alt',  items: [  {boxLabel: 'Item 1', name: 'rb-auto', inputValue: 1},  {boxLabel: 'Item 2', name: 'rb-auto', inputValue: 2, checked: **true**},  {boxLabel: 'Item 3', name: 'rb-auto', inputValue: 3},  {boxLabel: 'Item 4', name: 'rb-auto', inputValue: 4},  {boxLabel: 'Item 5', name: 'rb-auto', inputValue: 5}  ]  } |

#### 简单列布局



|  |
| --- |
| {  xtype: 'radiogroup',  fieldLabel: 'Single Column',  columns: 1,  items: [  {boxLabel: 'Item 1', name: 'rb-col', inputValue: 1},  {boxLabel: 'Item 2', name: 'rb-col', inputValue: 2, checked: **true**},  {boxLabel: 'Item 3', name: 'rb-col', inputValue: 3}  ]  } |

#### 横向多列布局



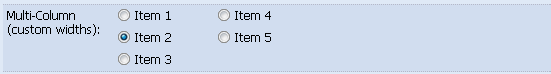
|  |
| --- |
| {  xtype: 'radiogroup',  fieldLabel: 'Multi-Column (horizontal)',  cls: 'x-check-group-alt',  columns: 3,  items: [  {boxLabel: 'Item 1', name: 'rb-horiz-1', inputValue: 1},  {boxLabel: 'Item 2', name: 'rb-horiz-1', inputValue: 2, checked: **true**},  {boxLabel: 'Item 3', name: 'rb-horiz-1', inputValue: 3},  {boxLabel: 'Item 1', name: 'rb-horiz-2', inputValue: 1, checked: **true**},  {boxLabel: 'Item 2', name: 'rb-horiz-2', inputValue: 2}  ]  } |

#### 纵向多列布局



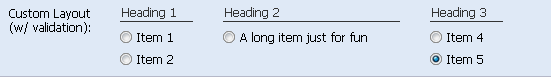
|  |
| --- |
| {  xtype: 'radiogroup',  fieldLabel: 'Multi-Column (vertical)',  columns: 3,  vertical: **true**,  items: [  {boxLabel: 'Item 1', name: 'rb-vert', inputValue: 1},  {boxLabel: 'Item 2', name: 'rb-vert', inputValue: 2, checked: **true**},  {boxLabel: 'Item 3', name: 'rb-vert', inputValue: 3},  {boxLabel: 'Item 4', name: 'rb-vert', inputValue: 4},  {boxLabel: 'Item 5', name: 'rb-vert', inputValue: 5}  ]  } |

#### 多列自定义宽度布局



|  |
| --- |
| {  xtype: 'radiogroup',  fieldLabel: 'Multi-Column<br />(custom widths)',  cls: 'x-check-group-alt',  columns: [100, 100],  vertical: **true**,  items: [  {boxLabel: 'Item 1', name: 'rb-custwidth', inputValue: 1},  {boxLabel: 'Item 2', name: 'rb-custwidth', inputValue: 2, checked: **true**},  {boxLabel: 'Item 3', name: 'rb-custwidth', inputValue: 3},  {boxLabel: 'Item 4', name: 'rb-custwidth', inputValue: 4},  {boxLabel: 'Item 5', name: 'rb-custwidth', inputValue: 5}  ]  } |

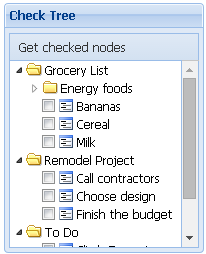
#### 自定义布局（分组中不选时不报错）



|  |
| --- |
| {  xtype: 'radiogroup',  fieldLabel: 'Custom Layout<br />(w/ validation)',  allowBlank: **false**,  msgTarget: 'side',  autoFitErrors: **false**,  anchor: '-18',  layout: 'column',  defaultType: 'container',  items: [{  columnWidth: .25,  items: [  {xtype: 'component', html: 'Heading 1', cls:'x-form-check-group-label'},  {xtype: 'radiofield', boxLabel: 'Item 1', name: 'rb-cust', inputValue: 1},  {xtype: 'radiofield', boxLabel: 'Item 2', name: 'rb-cust', inputValue: 2}  ]  },{  columnWidth: .5,  items: [  {xtype: 'component', html: 'Heading 2', cls:'x-form-check-group-label'},  {xtype: 'radiofield', boxLabel: 'A long item just for fun', name: 'rb-cust', inputValue: 3}  ]  },{  columnWidth: .25,  items: [  {xtype: 'component', html: 'Heading 3', cls:'x-form-check-group-label'},  {xtype: 'radiofield', boxLabel: 'Item 4', name: 'rb-cust', inputValue: 4},  {xtype: 'radiofield', boxLabel: 'Item 5', name: 'rb-cust', inputValue: 5}  ]  }]  } |

# 树

## 复选框树

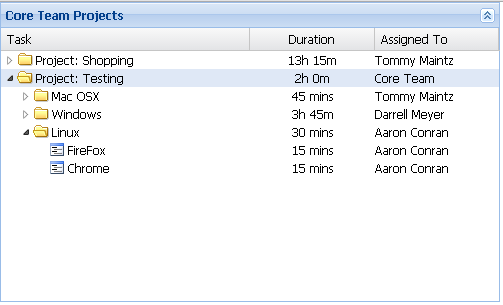


|  |
| --- |
| **var** store = Ext.create('Ext.data.TreeStore', {  proxy: {  type: 'ajax',  url: 'check-nodes.json'  },  sorters: [{  property: 'leaf',  direction: 'ASC'  }, {  property: 'text',  direction: 'ASC'  }]  });  **var** tree = Ext.create('Ext.tree.Panel', {  store: store,  rootVisible: **false**,  useArrows: **true**,  frame: **true**,  title: 'Check Tree',  renderTo: 'tree-div',  width: 200,  height: 250,  dockedItems: [{  xtype: 'toolbar',  items: {  text: 'Get checked nodes',  handler: **function**(){  **var** records = tree.getView().getChecked();  **var** names = [];    Ext.Array.each(records, **function**(rec){  names.push(rec.get('text'));  });    Ext.MessageBox.show({  title: 'Selected Nodes',  msg: names.join('<br />'),  icon: Ext.MessageBox.INFO  });  }  }  }]  }); |

check-nodes.json

|  |
| --- |
| [{  "text": "To Do",  "cls": "folder",  "expanded": **true**,  "children": [{  "text": "Go jogging",  "leaf": **true**,  "checked": **true**  },{  "text": "Take a nap",  "leaf": **true**,  "checked": **false**  },{  "text": "Climb Everest",  "leaf": **true**,  "checked": **false**  }]  },{  "text": "Grocery List",  "cls": "folder",  "children": [{  "text": "Bananas",  "leaf": **true**,  "checked": **false**  },{  "text": "Milk",  "leaf": **true**,  "checked": **false**  },{  "text": "Cereal",  "leaf": **true**,  "checked": **false**  },{  "text": "Energy foods",  "cls": "folder",  "children": [{  "text": "Coffee",  "leaf": **true**,  "checked": **false**  },{  "text": "Red Bull",  "leaf": **true**,  "checked": **false**  }]  }]  },{  "text": "Remodel Project",  "cls": "folder",  "children": [{  "text": "Finish the budget",  "leaf": **true**,  "checked": **false**  },{  "text": "Call contractors",  "leaf": **true**,  "checked": **false**  },{  "text": "Choose design",  "leaf": **true**,  "checked": **false**  }]  }] |

## Grid树



|  |
| --- |
| Ext.define('Task', {  extend: 'Ext.data.Model',  fields: [  {name: 'task', type: 'string'},  {name: 'user', type: 'string'},  {name: 'duration', type: 'string'}  ]  });  **var** store = Ext.create('Ext.data.TreeStore', {  model: 'Task',  proxy: {  type: 'ajax',  //the store will get the content from the .json file  url: 'treegrid.json'  },  folderSort: **true**  });  //Ext.ux.tree.TreeGrid is no longer a Ux. You can simply use a tree.TreePanel  **var** tree = Ext.create('Ext.tree.Panel', {  title: 'Core Team Projects',  width: 500,  height: 300,  renderTo: Ext.getBody(),  collapsible: **true**,  useArrows: **true**,  rootVisible: **false**,  store: store,  multiSelect: **true**,  singleExpand: **true**,  //the 'columns' property is now 'headers'  columns: [{  xtype: 'treecolumn', //**this** is so we know which column will show the tree  text: 'Task',  flex: 2,  sortable: **true**,  dataIndex: 'task'  },{  //we must use the templateheader component so we can use a custom tpl  xtype: 'templatecolumn',  text: 'Duration',  flex: 1,  sortable: **true**,  dataIndex: 'duration',  align: 'center',  //add **in** the custom tpl **for** the rows  tpl: Ext.create('Ext.XTemplate', '{duration:this.formatHours}', {  formatHours: **function**(v) {  **if** (v < 1) {  **return** Math.round(v \* 60) + ' mins';  } **else** **if** (Math.floor(v) !== v) {  **var** min = v - Math.floor(v);  **return** Math.floor(v) + 'h ' + Math.round(min \* 60) + 'm';  } **else** {  **return** v + ' hour' + (v === 1 ? '' : 's');  }  }  })  },{  text: 'Assigned To',  flex: 1,  dataIndex: 'user',  sortable: **true**  }]  }); |

Treejson.js

|  |
| --- |
| {"text":".","children": [  {  task:'Project: Shopping',  duration:13.25,  user:'Tommy Maintz',  iconCls:'task-folder',  expanded: **true**,  children:[{  task:'Housewares',  duration:1.25,  user:'Tommy Maintz',  iconCls:'task-folder',  children:[{  task:'Kitchen supplies',  duration:0.25,  user:'Tommy Maintz',  leaf:**true**,  iconCls:'task'  },{  task:'Groceries',  duration:.4,  user:'Tommy Maintz',  leaf:**true**,  iconCls:'task'  },{  task:'Cleaning supplies',  duration:.4,  user:'Tommy Maintz',  leaf:**true**,  iconCls:'task'  },{  task: 'Office supplies',  duration: .2,  user: 'Tommy Maintz',  leaf: **true**,  iconCls: 'task'  }]  }, {  task:'Remodeling',  duration:12,  user:'Tommy Maintz',  iconCls:'task-folder',  expanded: **true**,  children:[{  task:'Retile kitchen',  duration:6.5,  user:'Tommy Maintz',  leaf:**true**,  iconCls:'task'  },{  task:'Paint bedroom',  duration: 2.75,  user:'Tommy Maintz',  iconCls:'task-folder',  children: [{  task: 'Ceiling',  duration: 1.25,  user: 'Tommy Maintz',  iconCls: 'task',  leaf: **true**  }, {  task: 'Walls',  duration: 1.5,  user: 'Tommy Maintz',  iconCls: 'task',  leaf: **true**  }]  },{  task:'Decorate living room',  duration:2.75,  user:'Tommy Maintz',  leaf:**true**,  iconCls:'task'  },{  task: 'Fix lights',  duration: .75,  user: 'Tommy Maintz',  leaf: **true**,  iconCls: 'task'  }, {  task: 'Reattach screen door',  duration: 2,  user: 'Tommy Maintz',  leaf: **true**,  iconCls: 'task'  }]  }]  },{  task:'Project: Testing',  duration:2,  user:'Core Team',  iconCls:'task-folder',  children:[{  task: 'Mac OSX',  duration: 0.75,  user: 'Tommy Maintz',  iconCls: 'task-folder',  children: [{  task: 'FireFox',  duration: 0.25,  user: 'Tommy Maintz',  iconCls: 'task',  leaf: **true**  }, {  task: 'Safari',  duration: 0.25,  user: 'Tommy Maintz',  iconCls: 'task',  leaf: **true**  }, {  task: 'Chrome',  duration: 0.25,  user: 'Tommy Maintz',  iconCls: 'task',  leaf: **true**  }]  },{  task: 'Windows',  duration: 3.75,  user: 'Darrell Meyer',  iconCls: 'task-folder',  children: [{  task: 'FireFox',  duration: 0.25,  user: 'Darrell Meyer',  iconCls: 'task',  leaf: **true**  }, {  task: 'Safari',  duration: 0.25,  user: 'Darrell Meyer',  iconCls: 'task',  leaf: **true**  }, {  task: 'Chrome',  duration: 0.25,  user: 'Darrell Meyer',  iconCls: 'task',  leaf: **true**  },{  task: 'Internet Exploder',  duration: 3,  user: 'Darrell Meyer',  iconCls: 'task',  leaf: **true**  }]  },{  task: 'Linux',  duration: 0.5,  user: 'Aaron Conran',  iconCls: 'task-folder',  children: [{  task: 'FireFox',  duration: 0.25,  user: 'Aaron Conran',  iconCls: 'task',  leaf: **true**  }, {  task: 'Chrome',  duration: 0.25,  user: 'Aaron Conran',  iconCls: 'task',  leaf: **true**  }]  }]  }  ]} |

# Toolbar（工具栏）

## 简单例子

|  |
| --- |
| Ext.create('Ext.toolbar.Toolbar', {  renderTo: document.body,  width : 400,  items: [  {  text: 'Button'  },  {  xtype: 'splitbutton',  text : 'Split Button'  },  {  xtype : 'textfield',  name : 'field1',  emptyText: 'enter search term'  }  ]  }); |

## 菜单

|  |
| --- |
| Ext.onReady(**function**(){    **var** onItemClick = **function**(){  alert(111);  };    **var** myMenu = Ext.create('Ext.menu.Menu', {  //text : '菜单',  items:[  {  text: 'item 1',  handler: onItemClick  },{  text: 'item 2',  menu: {  items:[  {  text:'item 2-1',  handler: onItemClick  },{  text:'item 2-2',  handler: onItemClick  }  ]  }  },{  text: 'item 3',  handler: onItemClick  }  ]  });    Ext.create('Ext.toolbar.Toolbar', {  renderTo: document.body,  width : 400,  items: [  {  text: 'Button'  },  {  xtype: 'splitbutton',  text : 'Split Button'  },  {  xtype : 'textfield',  name : 'field1',  emptyText: 'enter search term'  },{  text : '菜单1',  menu : myMenu  },{  text : '菜单２',  menu : {  items:[  {  text: 'item5',  handler: onItemClick  },{  text: 'item4',  menu: {  items:[  {  text:'item 4-1',  handler: onItemClick  },{  text:'item 4-2',  handler: onItemClick  }  ]  }  }  ]  }  }  ]  });  }); |

## 填充方式和按钮

|  |
| --- |
| Ext.create('Ext.toolbar.Toolbar', {  renderTo: document.body,  width : 500,  items: [  {  //xtype: 'button', // default for Toolbars  text: 'Button',  handler: **function**() {  alert(111);  }  },  {  //Ext.button.Split  xtype: 'splitbutton',  text : 'Split Button'  },  // 元素之间填充方式，相当于{ xtype: 'tbfill' } Ext.toolbar.Fill  '->',  {  xtype : 'textfield',  name : 'field1',  emptyText: 'enter search term'  },  // 元素之间填充方式，相当于 {xtype: 'tbseparator'} Ext.toolbar.Separator  '-',  // 相当于{xtype: 'tbtext', text: 'text1'}  'text 1',  // 元素之间填充方式，相当于 ' ' Ext.toolbar.Spacer可控制宽度  { xtype: 'tbspacer' },  'text 2',  { xtype: 'tbspacer', width: 50 },  'text 3'  ]  }); |

## Ext.toolbar.Paging

Xtype: pagingtoolbar

|  |
| --- |
| **var** itemsPerPage = 200;  **var** store = Ext.create('Ext.data.Store', {  id:'simpsonsStore',  autoLoad: **false**,  fields:['name', 'email', 'phone'],  //每页显示的数据  pageSize: itemsPerPage,  proxy: {  type: 'ajax',  url: 'demo6Data.json',  reader: {  type: 'json',  root: 'items',  //共有多少条记录  totalProperty: 'total'  }  }  });  //请求参数  store.load({  params:{  hello: 1234  }  });  Ext.create('Ext.grid.Panel', {  title: 'Simpsons',  store: store,  columns: [  { header: 'Name', dataIndex: 'name' },  { header: 'Email', dataIndex: 'email', flex: 1 },  { header: 'Phone', dataIndex: 'phone' }  ],  width: 400,  height: 300,    //增加分页工具栏  dockedItems: [{  //页面工具条 Ext.toolbar.Paging  xtype: 'pagingtoolbar',  store: store,  //工具栏位置  dock: 'bottom',  //是否显示displayMsg的内容  displayInfo: **false**,  beforePageText: '第',  afterPageText: '页 共{0}页'  }],  renderTo: Ext.getBody()  }); |

# 常用属性、方法

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 用途 | 值 | 备注 |
| dataIndex | Store中的索引名 | String | [Ext.grid.column.Column](http://localhost:9000/ext/docs/index.html) |
| sortable | 此列是否可排序 | True/false | Ext.grid.column.Column |
| Locked | 是否锁定此列 | True/false |  |
| width | 列宽 | 数字 |  |
| collapsible | 面板是否可收缩 | True/false | Ext.panel.Panel |
| animCollapse | 面板收缩时，是否有动画 | True/false | Ext.panel.Table |
| columnLines | 列与列之间的分隔线 | True/false | Ext.panel.Table |
|  |  |  |  |
| renderer | 显示用途 |  | Ext.grid.column.Column |
| renderer : function(val) {  if (val > 0) {  return '<span style="color:green;">' + val + '</span>';  } else if (val < 0) {  return '<span style="color:red;">' + val + '</span>';  }  return val;  } | | |
| 多级表头语法 | Columns:[{  Text: ‘columnA’,  dataIndex: 'company'  },{  Text: ‘columnB’,  Columns:[{  Text:’columnB1’,  dataIndex: 'b1'  },{  Text:’columnB2’,  dataIndex: 'b2'  }]  }] | | |
| modal | Ext.window.Window  为true时，后阻塞主方法，直到window完成 | | |
| msgTarget | Form中的字段错误信息提示位置；  如：under, side 无等 | | |
| xtype: component',  width: 10 | 留白 | | |
| autoFitErrors | 错误提示信息是否自适应。True时，自动选择Under,side | | |
| Ext.AbstractComponent.up | var owningTabPanel = grid.up('tabpanel');  在父窗口中查找tabpanle | | |
| Ext.container.AbstractContainer.child | 容器中的直接子节点 | | |
| Ext.container.AbstractContainer.down | 容器中的下级节点 | | |
| dockedItems | dockedItems: [{  xtype: 'toolbar',  items: {  text: 'Get checked nodes',  handler: **function**(){  **var** records = tree.getView().getChecked(),  names = [];    Ext.Array.each(records, **function**(rec){  names.push(rec.get('text'));  });    Ext.MessageBox.show({  title: 'Selected Nodes',  msg: names.join('<br />'),  icon: Ext.MessageBox.INFO  });  }  }  }] | | |
| closable | 是否可关闭 | | |
| closeAction: 'hide' | 把window或其它控件的关闭事件设置为hide | | |

# 常用控件及代码

## Tooltip

Ext.tip.ToolTip

xtype: tooltip

|  |
| --- |
| var tip = [Ext.create](http://localhost:9000/ext/docs/index.html#!/api/Ext-method-create)('[Ext.tip.ToolTip](http://localhost:9000/ext/docs/index.html#!/api/Ext.tip.ToolTip)', {  target: 'clearButton',  html: 'Press this button to clear the form'  }); |

## Ext.define（定义类）

### 定义普通类

|  |
| --- |
| Ext.define('lzg.Test', {  someProperty: 'something',  someMethod: **function**(s) {  alert(s + **this**.someProperty);  }  });  **var** obj = **new** lzg.Test();  obj.someMethod('Say '); // alerts 'Say something' |

### 定义private变量

|  |
| --- |
| Ext.define('lzg.Test', **function** () {  **var** id = 0;//外部无法访问此变量，相当于private  **return** {  nextId: **function** () {  **return** ++id;  }  };  });  **var** test = **new** lzg.Test();  alert(test.nextId()); |

### 静态方法

|  |
| --- |
| Ext.define('MyApp.foo.Bar', **function** (aa) {  **return** {  statics: {  staticMethod: **function** () {  alert(1);  }  },  method: **function** () {  **return** aa.staticMethod();  }  };  });  MyApp.foo.Bar.staticMethod(); |

## Ext.Decode（把json字符串解释成Object对象）

|  |
| --- |
| **var** str = "{ id:'111', name:'lzg'}";  **var** obj = Ext.decode(str,**true**);  alert(obj.id); |

## Ext.String.format

var cls = 'my-class',

var text = 'Some text';

var s = Ext.String.format('<div class="{0}">{1}</div>', cls, text);

// s now contains the string: '<div class="my-class">Some text</div>'

## Ext.util.Format

## Ext.util.MixedCollection

|  |
| --- |
| var coll = new Ext.util.MixedCollection();  coll.add('key1', 'val1');  coll.add('key2', 'val2');  coll.add('key3', 'val3');  console.log(coll.get('key1')); // prints 'val1'  console.log(coll.indexOfKey('key3')); // prints 2 |

|  |
| --- |
| var coll = new Ext.util.MixedCollection();  coll.add('key1', 100);  coll.add('key2', -100);  coll.add('key3', 17);  coll.add('key4', 0);  var biggerThanZero = coll.filterBy(function(value){  return value > 0;  });  console.log(biggerThanZero.getCount()); // prints 2 |

## 代码片断

### 延时执行函数

|  |
| --- |
| setTimeout(function(){}, 1000); |

### 根据id获取某控件

|  |
| --- |
| Ext.ComponentManager.get('mainViewPanel'); |

### 根据id获取store

Ext.data.StoreManager.lookup('simpsonsStore')

### Ajax请求

|  |
| --- |
| Ext.Ajax.request({  url: 'page.php',  params: {  id: 1  },  success: function(response){  var text = response.responseText;  // process server response here  }  }) |

## Form

### Form重置

|  |
| --- |
| this.up('form').getForm().reset(); |

### 获取form

|  |
| --- |
| this.up('form').getForm(); |

### 循环form中的元素

|  |
| --- |
| Ext.iterate(form.getValues(), **function**(key, value) {  s += Ext.util.Format.format("{0} = {1}<br />", key, value);  }, **this**); |