**jQuery EasyUI中文帮助手册**

### API版本：1.2.4

目录

**[jQuery EasyUI中文帮助手册](#_Toc354081610)** [1](#_Toc354081610)

[API版本：1.2.4 1](#_Toc354081611)

[1. 基本 2](#_Toc354081612)

[1.1 语法解析 2](#_Toc354081613)

[语法解析 2](#_Toc354081614)

[1.2 简单载入器 2](#_Toc354081615)

**[简单载入器](#_Toc354081616)** [2](#_Toc354081616)

[1.3 一般拖动 4](#_Toc354081617)

**[可拖动](#_Toc354081618)** [4](#_Toc354081618)

[1.4 拖动至容器 5](#_Toc354081619)

**[拖动至容器](#_Toc354081620)** [5](#_Toc354081620)

[1.5 缩放 6](#_Toc354081621)

**[缩放](#_Toc354081622)** [6](#_Toc354081622)

[1.6 分页 7](#_Toc354081623)

**[分页](#_Toc354081624)** [7](#_Toc354081624)

[1.7 搜索框 9](#_Toc354081625)

**[搜索框](#_Toc354081626)** [9](#_Toc354081626)

[1.8 进度条 11](#_Toc354081627)

**[进度条](#_Toc354081628)** [11](#_Toc354081628)

[2. 布局管理器 12](#_Toc354081629)

[2.1 控制面板 12](#_Toc354081630)

[2.2 选项卡 18](#_Toc354081631)

**[选项卡切换](#_Toc354081632)** [18](#_Toc354081632)

[2.3 可伸缩面板 22](#_Toc354081633)

**[可伸缩面板](#_Toc354081634)** [22](#_Toc354081634)

[2.4 布局面板 24](#_Toc354081635)

**[布局面板](#_Toc354081636)** [24](#_Toc354081636)

[3. 菜单和按钮 27](#_Toc354081637)

[3.1 菜单 27](#_Toc354081638)

**[菜单](#_Toc354081639)** [27](#_Toc354081639)

[3.2 链接按钮 30](#_Toc354081640)

**[链接按钮](#_Toc354081641)** [30](#_Toc354081641)

[3.3 菜单按钮 31](#_Toc354081642)

**[菜单按钮](#_Toc354081643)** [31](#_Toc354081643)

[3.4 分隔按钮 32](#_Toc354081644)

**[分隔按钮](#_Toc354081645)** [32](#_Toc354081645)

[4. 表单 33](#_Toc354081646)

[4.1 表单 33](#_Toc354081647)

**[表单](#_Toc354081648)** [33](#_Toc354081648)

[4.2 表单验证 35](#_Toc354081649)

**[表单验证](#_Toc354081650)** [35](#_Toc354081650)

[4.3 自定义组合框 37](#_Toc354081651)

**[自定义组合框](#_Toc354081652)** [37](#_Toc354081652)

[4.4 可装载组合框 39](#_Toc354081653)

**[可装载组合框](#_Toc354081654)** [39](#_Toc354081654)

[4.5 组合树 42](#_Toc354081655)

**[组合树](#_Toc354081656)** [42](#_Toc354081656)

[4.6 组合表格 43](#_Toc354081657)

**[组合表格](#_Toc354081658)** [43](#_Toc354081658)

[4.7 数字验证框 45](#_Toc354081659)

**[数字验证框](#_Toc354081660)** [45](#_Toc354081660)

[4.8 日期组合框 46](#_Toc354081661)

**[日期组合框](#_Toc354081662)** [46](#_Toc354081662)

[4.9 日期时间组合框 48](#_Toc354081663)

**[日期时间组合框](#_Toc354081664)** [48](#_Toc354081664)

[4.10 日历 49](#_Toc354081665)

**[日历](#_Toc354081666)** [49](#_Toc354081666)

[4.11 调节器 51](#_Toc354081667)

**[调节器](#_Toc354081668)** [51](#_Toc354081668)

[4.12 数字调节器 52](#_Toc354081669)

**[数字调节器](#_Toc354081670)** [52](#_Toc354081670)

[4.13 时间调节器 53](#_Toc354081671)

**[时间调节器](#_Toc354081672)** [53](#_Toc354081672)

[5. 窗口 54](#_Toc354081673)

[5.1 窗口 54](#_Toc354081674)

**[窗口](#_Toc354081675)** [54](#_Toc354081675)

[5.2 对话窗口 56](#_Toc354081676)

**[对话窗口](#_Toc354081677)** [56](#_Toc354081677)

[5.3 消息窗口 58](#_Toc354081678)

**[消息窗口](#_Toc354081679)** [58](#_Toc354081679)

[6. 数据表格和树形菜单 60](#_Toc354081680)

[6.1 数据表格 60](#_Toc354081681)

**[数据表格](#_Toc354081682)** [60](#_Toc354081682)

[6.2 属性表格 69](#_Toc354081683)

**[属性表格](#_Toc354081684)** [69](#_Toc354081684)

[6.3 树形菜单 69](#_Toc354081685)

**[树形菜单](#_Toc354081686)** [69](#_Toc354081686)

[6.4 树形表格 75](#_Toc354081687)

**[树形表格](#_Toc354081688)** [75](#_Toc354081688)

# 基本

## 语法解析

### 语法解析

#### 使用方法

1. $.parser.parse();       // 解析所有的页面
2. $.parser.parse('#cc');  // 解析特定的代码

#### 属性

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| $.parser.auto | boolean（布尔型） | 定义是否自动解析easyui组件。 | true |

#### 事件

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| $.parser.onComplete | context | 当语法解析结束时触发事件。 |

#### 方法

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| $.parser.parse | context | 解析easyui组件。 |

## 简单载入器

**简单载入器**

**使用方法**

1. easyloader.base = '../';    // 设置easyui根目录
2. easyloader.load('messager', **function**(){        // 载入特定模块
3. $.messager.alert('Title', 'load ok');
4. });

**属性**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| modules（模块） | object（对象） | 预定义模块。 |  |
| locales（多语言） | object（预定义对象） | 预定义多语言。 |  |
| base（很目录） | string（字符串） | easyui根目录,必须以'/'结尾。 | easyui根目录将会被自动设置为相对于easyload.js的位置。 |
| theme（主题名称） | string（字符串） | 'themes'目录中的主题的名称。 | default |
| css（层叠样式表） | boolean（布尔型） | 定义是否在加载模块的同时加载css文件。 | true |
| locale（自定义语言） | string（字符串） | 语言名称。 | null |
| timeout（超时） | number（数字） | 超时单位为毫秒，出现超时就触发。 | 2000 |

**预定义语言**

* de（德语）
* en（英语）
* fr（法语）
* zh\_CN（简体中文）
* zh\_TW（繁体中文）

**事件**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onProgress | name | 当一个模块成功载入时触发。 |
| onLoad | name | 当一个模块及其所有依赖关系（可以理解为载入该模块所需要的其他模块、属性、方法等）载入时触发。 |

**方法**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| load | module, callback | 载入特定的模块。当载入成功时将调用回调函数。 有效的模块参数类型如下: 单一的模块名称 一个存储有模块名称的数组 一个以'.css'结尾的样式文件 一个以'.js'结尾的js文件 |

## 一般拖动

**可拖动**

使用$.fn.draggable.defaults重载默认值。

**使用方法**

1. **<div** id="dd" style="width:100px;height:100px;"**>**
2. **<div** id="title" style="background:#ccc;"**>**title**</div>**
3. **</div>**
4. $('#dd').draggable({
5. handle:'#title'
6. });

**属性**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| proxy（替代） | string,function | 拖动时的替代元素，当被设置为'clone'时，将使用该元素的一个复制元素来作为替代元素。如果指定了一个函数, 它将返回一个jquery对象。 | null |
| revert（复原） | boolean（布尔型） | 如果设置为true, 当拖动停止时元素将返回它的初始位置。 | false |
| cursor（指针） | string（字符串） | 拖动时的CSS指针。 | move |
| deltaX（水平增量，X轴） | number（数字） | 被拖动元素对应于当前指针的水平位置。 | null |
| deltaY（垂直增量，y轴） | number（数字） | 被拖动元素对应于当前指针的垂直位置。 | null |
| handle（句柄） | selector（选择器） | 开始拖动的句柄。 | null |
| disabled（停止拖动） | boolean（布尔型） | 当设置为true时停止拖动。 | false |
| edge（边缘） | number（数字） | 可以在其中拖动的容器的宽度 | 0 |
| axis（拖动轴） | string（字符串） | 可用值为'v'或者'h',当设置为空时，元素可以同时在水平和垂直方法拖动。 | null |

**事件**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onBeforeDrag | e | 在拖动之前触发，返回false将取消拖动。 |
| onStartDrag | e | 当目标对象开始被拖动时触发。 |
| onDrag | e | 在拖动过程中触发，当不能再拖动时返回false。 |
| onStopDrag | e | 当拖动停止时触发。 |

**方法**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回属性对象。 |
| proxy | none | 如果替代元素被设置，返回将被用于拖动的替代元素。 |
| enable | none | 允许拖动。 |
| disable | none | 禁止拖动。 |

## 拖动至容器

**拖动至容器**

使用$.fn.droppable.defaults重载默认值。

**使用方法**

1. **<div** id="dd" style="width:100px;height:100px;"**></div>**
2. $('#dd').droppable({
3. accept:'#d1,#d3'
4. });

**属性**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| accept | selector | 决定哪个可拖动元素将会被接收。 | null |

**事件**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onDragEnter | e,source | 当可拖动元素被拖入目标容器的时候触发，参数source是被拖动的DOM元素。 |
| onDragOver | e,source | 当可拖动元素被拖至某个元素之上的时候触发，参数source是被拖动的DOM元素。 |
| onDragLeave | e,source | 当可拖动元素被拖离目标容器的时候触发，参数source是被拖动的DOM元素。 |
| onDrop | e,source | 当可拖动元素被拖入目标容器并放置的时候触发，参数source是被拖动的DOM元素。 |

## 缩放

**缩放**

使用$.fn.resizable.defaults重载默认值。

**使用方法**

1. **<div** id="rr" style="width:100px;height:100px;border:1px solid #ccc;"**></div>**
2. $('#rr').resizable({
3. maxWidth:800,
4. maxHeight:600
5. });

**属性**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| disabled（禁止缩放） | boolean（布尔型） | 如果设置为true将禁止缩放。 | false |
| handles（句柄） | string（字符串） | 说明缩放的方向，'n'表示向上（北）,'e'表示向右（东）等。 | n, e, s, w, ne, se, sw, nw, all |
| minWidth（最小宽度） | number（数字） | 缩放时所允许的最小宽度。 | 10 |
| minHeight（最小高度） | number（数字） | 缩放时所允许的最小高度。 | 10 |
| maxWidth（最大宽度） | number（数字） | 缩放时所允许的最大宽度。 | 10000 |
| maxHeight | number（数字） | 缩放时所允许的最大高度。 | 10000 |
| edge（边界） | number（数字） | 将被缩放的边框的边界。 | 5 |

**事件**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onStartResize | e | 当开始缩放时触发。 |
| onResize | e | 在缩放过程中触发，当返回false时, DOM元素将不再缩放。 |
| onStopResize | e | 当缩放停止是触发。 |

## 分页

**分页**

使用$.fn.pagination.defaults重载默认值。

**依赖关系**

* 链接按钮

**使用指南**

1. **<div** id="pp" style="background:#efefef;border:1px solid #ccc;"**></div>**
2. $('#pp').pagination({
3. total:2000,
4. pageSize:10
5. });

**属性**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| total（总数） | number（数字） | 总记录数，当创建分页时必须设置。 | 1 |
| pageSize（每页记录数） | number（数字） | 每页显示的记录数。 | 10 |
| pageNumber（当前页码） | number（数字） | 当分页创建完毕时显示当前页码。 | 1 |
| pageList（可选择的每页记录数） | array（数组） | 用户能够改变每页显示的记录数。 | [10,20,30,50] |
| loading（是否正在载入） | boolean（布尔型） | 定义数据是否正在载入。 | false |
| buttons（按钮） | array（数组） | 自定义按钮，每个按钮包含2个属性: iconCls： 显示背景图片的CSS类 handler： 当按钮被点击时调用的一个句柄函数 | null |
| showPageList（显示可选择的每页记录数） | boolean（布尔型） | 定义是否显示可选择的每页记录数。 | true |
| showRefresh（显示刷新） | boolean（布尔型） | 定义是否显示刷新按钮。 | true |
| beforePageText（输入框之前的文本） | string（字符串） | 在输入框之前显示的符号。 | Page |
| afterPageText（输入框之后的文本） | string（字符串） | 在输入框之后显示的符号。 | of {pages} |
| displayMsg（显示信息） | string（字符串） | 在插件右上方显示分页信息。 | Displaying {from} to {to} of {total} items |

**事件**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onSelectPage | pageNumber, pageSize | 当用户进行翻页时触发，回调函数包含2个参数: pageNumber： 下一页的页码 pageSize： 下一页的显示记录数 |
| onBeforeRefresh | pageNumber, pageSize | 刷新之前触发, 返回false将取消刷新。 |
| onRefresh | pageNumber, pageSize | 刷新之后触发。 |
| onChangePageSize | pageSize | 当用户修改每页显示记录数时触发。 |

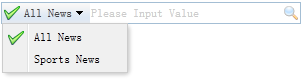
**方法**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回分页属性对象。 |
| loading | none | 提醒分页插件正在载入。 |
| loaded | none | 提醒分页插件已经载入。 |

## 搜索框

**搜索框**

使用$.fn.searchbox.defaults重载默认值。



**依赖关系**

* 菜单按钮

**使用方法**

**创建搜索框**

1. 使用标签创建。对input标签引用'easyui-searchbox'类。

1. **<script** type="text/javascript"**>**
2. function qq(value,name){
3. alert(value+":"+name)
4. }
5. **</script>**
6. **<input** id="ss" class="easyui-searchbox"
7. searcher="qq"
8. prompt="Please Input Value" menu="#mm" style="width:300px"**></input>**
9. **<div** id="mm" style="width:120px"**>**
10. **<div** name="all" iconCls="icon-ok"**>**All News**</div>**
11. **<div** name="sports"**>**Sports News**</div>**
12. **</div>**

2. 使用脚本创建。

1. <input id="ss"></input>
2. <div id="mm" style="width:120px">
3. <div name="all" iconCls="icon-ok">All News</div>
4. <div name="sports">Sports News</div>
5. </div>
6. $('#ss').searchbox({
7. width:200,
8. searcher:**function**(value,name){
9. alert(value + "," + name)
10. },
11. menu:'#mm',
12. prompt:'Please Input Value'
13. });

**属性**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| width（宽度） | number（数字） | 设置组建的宽度。 | auto |
| propmt（提醒） | string（字符串） | 显示在搜索框的提醒信息。 | '' |
| value（值） | string（字符串） | 搜索框的默认值。 | '' |
| menu（菜单） | selector（选择器） | 搜索类型下拉菜单。 | null |
| searcher（搜索器） | function(value,name) | 当用户点击搜索按钮或者按下ENTER键的时候搜索函数将被调用。 | null |

**方法**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回搜索框属性对象。 |
| menu | none | 返回搜索类型菜单对象。 |
| textbox | none | 返回文本框对象。 |
| getValue | none | 返回当前搜索关键字。 |
| setValue | value | 设置新的搜索关键字。 |
| getName | none | 返回当前搜索类型。 |
| destroy | none | 清除搜索框组件 |
| resize | width | 重置搜索框组建的宽度。 |

﻿

## 进度条

**进度条**

使用$.fn.progressbar.defaults重载默认值。



**依赖关系**

* 无

**使用方法**

**创建进度条**

进度条组件可以通过html标签或脚本两种方法创建。使用html标签创建更容易，只需要对<div>标签引用'easyui-progressbar'类。

1. **<div** id="p" class="easyui-progressbar" style="width:400px;"**></div>**

**获取/设置进度值**

为进度条获取当前进度值并且设置一个新的进度值。

1. **var** value = $('#p').progressbar('getValue');
2. **if** (value < 100){
3. value += Math.floor(Math.random() \* 10);
4. $('#p').progressbar('setValue', value);
5. }

**属性**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| width（宽度） | string（字符串） | 设置进度条的宽度。 | auto |
| value（值） | number（数字） | 当前进度，百分比数值。 | 0 |
| text（文本） | string（字符串） | 显示在组件中的文本模板。 | {value}% |

**事件**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onChange | newValue,oldValue | 当进度发生改变时触发。 |

**方法**

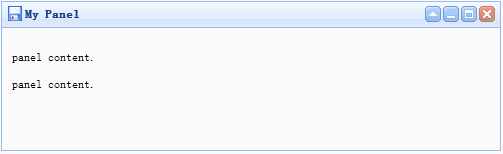
|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回进度条属性对象。 |
| resize | width | 重置进度条。 |
| getValue | none | 返回当前进度值。 |
| setValue | value | 设置一个新的进度值。 |

﻿

# 布局管理器

## 控制面板

使用$.fn.panel.defaults重载默认值。



**使用方法**

**创建控制面板**

1. 使用标签创建控制面板

使用标签创建控制面板十分简单，只须要对<div>标签引用'easyui-panel'类。

1. **<div** id="p" class="easyui-panel" title="My Panel" style="width:500px;height:150px;padding:10px;background:#fafafa;"
2. iconCls="icon-save"  closable="true"
3. collapsible="true" minimizable="true" maximizable=true**>**
4. **<p>**panel content.**</p>**
5. **<p>**panel content.**</p>**
6. **</div>**

2. 使用脚本创建控制面板

下面的代码将创建一个工具栏在右上方的控制面板。

1. <div id="p" style="padding:10px;">
2. <p>panel content.</p>
3. <p>panel content.</p>
4. </div>
5. $('#p').panel({
6. width:500,
7. height:150,
8. title: 'My Panel',
9. tools: [{
10. iconCls:'icon-add',
11. handler:**function**(){alert('new')}
12. },{
13. iconCls:'icon-save'
14. handler:**function**(){alert('save')}
15. }]
16. });

**移动控制面板**

调用'move'方法可以将控制面板移动到一个新的位置。

1. $('#p').panel('move',{
2. left:100,
3. top:100
4. });

**载入内容**

下面的代码将使用ajax载入控制面板内容并且在载入成功时显示信息。

1. $('#p').panel({
2. href:'content\_url.php',
3. onLoad:**function**(){
4. alert('loaded successfully');
5. }
6. });

**属性**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| title（标题） | string（字符串） | 显示在控制面板顶部的标题文本。 | null |
| iconCls（图标CSS类） | string（字符串） | 在控制面板标题前显示一个16x16大小图标的CSS类。 | null |
| width（宽度） | number（数字） | 设置控制面板的宽度。 | auto |
| height（高度） | number（数字） | 设置控制面板的高度。 | auto |
| left（左边距） | number（数字） | 设置控制面板的左边距。 | null |
| top（顶边距） | number（数字） | 设置控制面板的顶边距。 | null |
| cls（类） | string（字符串） | 对控制面板引用一个CSS类。 | null |
| headerCls（头部css类） | string（字符串） | 对控制面板头部引用一个CSS类。 | null |
| bodyCls（主体类） | string（字符串） | 对控制面板主体引用一个CSS类。 | null |
| style（样式） | object（对象） | 给控制面板添加一个自定义的样式。 | {} |
| fit(铺满浏览器) | boolean（布尔型） | 设置为true时，控制面板的大小将铺满它所在的容器（浏览器）。 | false |
| border（边框） | boolean（布尔型） | 定义是否显示控制面板边框。 | true |
| doSize（调整大小） | boolean（布尔型） | 如果设置为true，在控制面板被创建时将被重置大小并且自动布局。 | true |
| noheader（无头部） | boolean（布尔型） | 如果设置为true，控制面板头部将不被创建。 | false |
| content（内容） | string（字符串） | 控制面板主体的内容。 | null |
| collapsible（可折叠） | boolean（布尔型） | 定义是否显示可折叠按钮。 | false |
| minimizable（最小化） | boolean（布尔型） | 定义是否显示最小化按钮。 | false |
| maximizable（最大化） | boolean（布尔型） | 定义是否显示最大化按钮。 | false |
| closable（关闭） | boolean（布尔型） | 定义是否显示关闭按钮。 | false |
| tools（工具栏） | array（数组） | 自定义工具栏，每一个工具都可以包含2个属性：图标类 和句柄。 | [] |
| collapsed（已折叠） | boolean（布尔型） | 定义控制面板初始化时是否折叠。 | false |
| minimized（已最小化） | boolean（布尔型） | 定义控制面板初始化时是否最小化。 | false |
| maximized（已最大化） | boolean（布尔型） | 定义控制面板初始化时是否最大化。 | false |
| closed（已关闭） | boolean（布尔型） | 定义控制面板初始化时是否关闭。 | false |
| href（超链接） | string（字符串） | 载入远程数据的超链接，载入的数据将显示在控制面板中。 | null |
| cache（缓存） | boolean（布尔型） | 如果设置为true，从超链接载入的数据将被缓存。 | true |
| loadingMessage（载入时信息） | string（字符串） | 在正在载入远程数据时显示在控制面板中的信息。 | Loading… |
| extractor（提取器） | function（函数） | 定义如何从ajax返回值中提取内容,返回被提取的数据。  extractor: function(data){  var pattern = /<body[^>]\*>((.|[\n\r])\*)<\/body>/im;  var matches = pattern.exec(data);  if (matches){  return matches[1]; // only extract body content  } else {  return data;  }  } |  |

**事件**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onLoad | none | 在远程数据被载入时触发。 |
| onBeforeOpen | none | 在控制面板被打开之前触发，返回false将停止打开。 |
| onOpen | none | 在控制面板被打开之后触发。 |
| onBeforeClose | none | 在控制面板被关闭之前触发，返回false将取消关闭。 |
| onClose | none | 在控制面板被关闭后触发。 |
| onBeforeDestroy | none | 在控制面板被注销前触发，返回false将取消注销。 |
| onDestroy | none | 在控制面板被注销后触发。 |
| onBeforeCollapse | none | 在控制面板被折叠之前触发，返回false将停止折叠。 |
| onCollapse | none | 在控制面板被折叠之后触发。 |
| onBeforeExpand | none | 在控制面板被扩展之前触发，返回false将停止扩展（这里应该是指扩展区域，宽、高等）。 |
| onExpand | none | 在控制面板被扩展之后触发。 |
| onResize | width, height | 在控制面板被缩放后触发。 width： 新的控制面板宽度 height：新的控制面板高度 |
| onMove | left,top | 在控制面板被移动后触发。 left：新的控制面板左边距 top：新的控制面板顶边距 |
| onMaximize | none | 在控制面板被最大化后触发 |
| onRestore | none | 在控制面板被重置为初始大小后触发。 |
| onMinimize | none | 在控制面板被最小化后触发。 |

**方法**

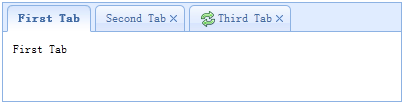
|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回选项属性。 |
| panel | none | 返回控制面板对象。 |
| header | none | 返回控制面板头对象。 |
| body | none | 返回控制面板主体对象。 |
| setTitle | title | 设置控制面板头部的标题文本。 |
| open | forceOpen | 当参数forceOpen设置为true时，控制面板将被打开，不受onBeforeOpen回调函数的约束。 |
| close | forceClose | 当参数forceClose设置为true时, 控制面板将被打开，不受onBeforeClose回调函数的约束。 |
| destroy | forceDestroy | 当参数forceDestroy设置为true时,控制面板将被打开，不受onBeforeDestroy回调函数的约束。 |
| refresh | href | 当href属性被设置时，刷新控制面板以载入远程数据。 |
| resize | options | 这是控制面板的尺寸并且进行布局。options对象包含以下2个属性: width： 新的控制面板宽度 height： 新的控制面板高度 left： 新的控制面板左边距 top： 新的控制面板顶边距 |
| move | options | 移动控制面板到一个新的位置。options对象包含以下2个属性: left： 新的控制面板左边距 top： 新的控制面板顶边距 |
| maximize | none | 使控制面板铺满整个容器。 |
| minimize | none | 最小化控制面板。 |
| restore | none | 使最大化的控制面板重置为其初始化时的大小和位置。 |
| collapse | animate | 折叠控制面板主体。 |
| expand | animate | 扩展控制面板主体。 |

﻿

## 选项卡

**选项卡切换**

使用$.fn.tabs.defaults重载默认值。



**依赖关系**

* 控制面板
* 链接按钮

**使用方法**

**创建选项卡**

1. 使用HTML标签创建选项卡

使用HTML标签创建选项卡十分简单，不需要写任何javascript代码，只需要<div>标签引用'easyui-panel'类。每个选项卡面板都可以使用闭合的<div>标签对创建，使用方法跟创建控制面板一样。

1. **<div** id="tt" class="easyui-tabs" style="width:500px;height:250px;"**>**
2. **<div** title="Tab1" style="padding:20px;display:none;"**>**
3. tab1
4. **</div>**
5. **<div** title="Tab2" closable="true" style="overflow:auto;padding:20px;display:none;"**>**
6. tab2
7. **</div>**
8. **<div** title="Tab3" iconCls="icon-reload" closable="true" style="padding:20px;display:none;"**>**
9. tab3
10. **</div>**
11. **</div>**

2. 使用脚本创建选项卡

下面的代码演示如何使用脚本创建选项卡，当该选项卡被选择时将会触发'onSelect'事件。

1. $('#tt').tabs({
2. border:**false**,
3. onSelect:**function**(title){
4. alert(title+' is selected');
5. }
6. });

**添加一个选项卡面板**

1. //添加一个选项卡面板
2. $('#tt').tabs('add',{
3. title:'New Tab',
4. content:'Tab Body',
5. closable:**true**
6. });

**获取当前被选择的选项卡**

1. //获取当前被选择的选项卡以及相应的选项卡对象
2. **var** pp = $('#tt').tabs('getSelected');
3. **var** tab = pp.panel('options').tab;    //相应的选项卡对象

**属性**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| width（宽度） | number（数字） | 选项卡所在容器（控制面板）的宽度。 | auto |
| height（高度） | number（数字） | 选项卡所在容器（控制面板）的高度。 | auto |
| plain（简洁模式） | boolean（布尔型） | 如果设置为true，将不显示控制面板背景。 | false |
| fit（铺满浏览器） | boolean（布尔型） | 设置为true时，选项卡的大小将铺满它所在的容器（浏览器）。 | false |
| border（边框） | boolean（布尔型） | 如果设置true，将显示选项卡所在容器（控制面板）的边框。 | true |
| scrollIncrement（滚动增量） | number（数字） | 选项卡滚动条每次滚动的像素值。 | 100 |
| scrollDuration（滚动时间） | number（数字） | 每次滚动持续的时间，单位为毫秒。 | 400 |
| tools（工具栏） | array（数组） | 控制面板右侧的工具栏，每个工具选项都跟链接按钮一样。 | null |

**事件**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onLoad | panel | 当一个选项卡完成ajax远程载入对象时触发。 |
| onSelect | title | 当用户选择一个选项卡时触发。 |
| onBeforeClose | title | 在一个选项卡被关闭之前触发，返回false将取消关闭。 |
| onClose | title | 在用户关闭一个选项卡面板后触发。 |
| onAdd | title | 在一个选项卡面板被添加后触发。 |
| onUpdate | title | 在一个选项卡面板被更新后触发。 |
| onContextMenu | e, title | 在一个选项卡面板被鼠标右键单击后触发。 |

**方法**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回选项卡属性对象。 |
| tabs | none | 返回所有的选项卡面板。 |
| resize | none | 重置选项卡容器（控制面板）大小并且自动布局。 |
| add | options | 添加一个新的选项卡面板，options参数是一个可以配置的对象，查看选项卡控制面板属性获取更多信息。 |
| close | title | 关闭一个选项卡面板，title参数表示哪个选项卡将被关闭。 |
| getTab | title | 获取特定的选项卡面板名称。 |
| getSelected | none | 获取被选择的选项卡面板名称。 |
| select | title | 选择一个选项卡面板。 |
| exists | title | 验证一个特定的选项卡面板是否存在。 |
| update | param | 更新特定的选项卡面板，param参数包含2个属性: tab: 将被更新的选项卡。 options: 选项卡相关配置项。 |

**选项卡面板**

选项卡面板属性在控制面板组件中定义，以下是一些常用属性。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| title（标题） | string（字符串） | 选项卡面板的标题。 |  |
| content（内容） | string（字符串） | 选项卡面板的内容。 |  |
| href（超链接） | string（字符串） | 从超链接载入远程内容到选项卡 面板。 | null |
| cache（缓存） | boolean（布尔型） | 设置为true将缓存选项卡面板，当href（超链接）属性被设置时有效。 | true |
| iconCls（图标CSS类） | string（字符串） | 一个显示选项卡面板标题图标的CSS类。 | null |
| width（宽度） | number（数字） | 选项卡面板的宽度。 | auto |
| height（高度） | number（数字） | 选项卡面板的高度。 | auto |

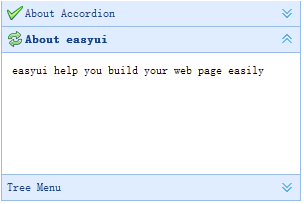
一些额外属性

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| closable（可关闭） | boolean（布尔型） | 当这是为true时，选项卡面板将会显示一个关闭按钮，点击该按钮将关闭选项卡。 | false |
| selected（默认选项卡） | boolean（布尔型） | 当设置为true时，将会默认显示该选项卡下的内容。 | false |

## 可伸缩面板

**可伸缩面板**

使用$.fn.accordion.defaults重载默认值



**依赖关系**

* 控制面板

**使用方法**

**创建可伸缩面板**

使用html标签创建可伸缩面板，只须要对<div>标签引用'easyui-accordion'类。

1. **<div** id="aa" class="easyui-accordion" style="width:300px;height:200px;"**>**
2. **<div** title="Title1" iconCls="icon-save" style="overflow:auto;padding:10px;"**>**
3. **<h3** style="color:#0099FF;"**>**Accordion for jQuery**</h3>**
4. **<p>**Accordion is a part of easyui framework for jQuery.
5. It lets you define your accordion component on web page more easily.**</p>**
6. **</div>**
7. **<div** title="Title2" iconCls="icon-reload" selected="true" style="padding:10px;"**>**
8. content2
9. **</div>**
10. **<div** title="Title3"**>**
11. content3
12. **</div>**
13. **</div>**

在后面将介绍如何改变或者重建可伸缩面板以及修改一些特性。

1. $('#aa').accordion({
2. animate:**false**
3. });

**刷新可伸缩面板的内容**

调用'getSelected'方法获取当前面板（的名称），接着可以调用控制面板的'refresh'方法来载入新的内容。

1. **var** pp = $('#aa').accordion('getSelected'); //获取当前选择的可伸缩面板
2. **if** (pp){
3. pp.panel('refresh','new\_content.php');  //调用'refresh'方法载入新内容
4. }

**容器属性（可伸缩面板所在的容器，通常称为‘盒子’）**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| width（宽度） | number（数字） | 可伸缩面板所在容器的宽度。 | auto |
| height（高度） | number（数字） | 可伸缩面板所在容器的高度。 | auto |
| fit（铺满浏览器） | boolean（布尔型） | 如果设置为true，可伸缩面板所在容器将铺满它所在的父容器（浏览器） | false |
| border（边框） | boolean（布尔型） | 定义是否显示边框。 | true |
| animate（动画效果） | boolean（布尔型） | 定义当延伸或者折叠面板时是否显示动画效果。 | true |

**可伸缩面板属性**

可伸缩面板继承控制面板的属性，以下是额外的属性:

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| selected（被选择） | boolean（布尔型） | 设置为true将默认展开面板。 | false |

**事件**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onSelect | title | 当一个可伸缩面板被选择时触发。 |
| onAdd | title | 在一个新面板被添加时触发。 |
| onBeforeRemove | title | 在可伸缩面板被移除之前触发，返回false将取消移除。 |
| onRemove | title | 在一个可伸缩面板被移除时触发。 |

**方法**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回容器属性对象。 |
| panels | none | 获取所有的面板。 |
| resize | none | 重置可伸缩面板。 |
| getSelected | none | 获取被选择的面板。 |
| getPanel | title | 获取特定的可伸缩面板。 |
| select | title | 选择特定的面板。 |
| add | options | 添加一个先的可伸缩面板。 |
| remove | title | 移除特定的面板。 |

## 布局面板

**布局面板**



**依赖关系**

* 控制面板
* 缩放

**使用方法**

**创建布局面板**

使用HTML标签创建布局面板，只须要对<div>标签引用'easyui-layout'类。

1. **<div** id="cc" class="easyui-layout" style="width:600px;height:400px;"**>**
2. **<div** region="north" title="North Title" split="true" style="height:100px;"**></div>**
3. **<div** region="south" title="South Title" split="true" style="height:100px;"**></div>**
4. **<div** region="east" iconCls="icon-reload" title="East" split="true" style="width:100px;"**></div>**
5. **<div** region="west" split="true" title="West" style="width:100px;"**></div>**
6. **<div** region="center" title="center title" style="padding:5px;background:#eee;"**></div>**
7. **</div>**

在整个页面创建布局面板。

1. **<body** class="easyui-layout"**>**
2. **<div** region="north" title="North Title" split="true" style="height:100px;"**></div>**
3. **<div** region="south" title="South Title" split="true" style="height:100px;"**></div>**
4. **<div** region="east" iconCls="icon-reload" title="East" split="true" style="width:100px;"**></div>**
5. **<div** region="west" split="true" title="West" style="width:100px;"**></div>**
6. **<div** region="center" title="center title" style="padding:5px;background:#eee;"**></div>**
7. **</body>**

**折叠布局面板**

1. $('#cc').layout();
2. //折叠左边的面板
3. $('#cc').layout('collapse','west');

**布局面板属性**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| title（标题） | string（字符串） | 布局面板的标题。 | null |
| region（区域） | string（字符串） | 定义布局面板的位置（方向），可以取下列值的其中之一：north, south, east, west, center。 |  |
| border（边框） | boolean（布尔型） | 设置为true将显示布局面板的边框。 | true |
| split（分隔条） | boolean（布尔型） | 如果设置为true将显示一个分隔条，用户可以拖动分隔条来改变布局面板的尺寸。 | false |
| iconCls（图标CSS类） | string（字符串） | 一个用来显示布局面板头部图标的css类。 | null |
| href（超链接） | string（字符串） | 一个用来从远程站点载入数据的超链接。 | null |

**方法**

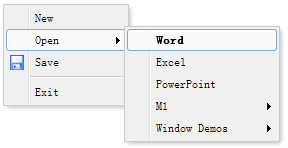
|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| resize | none | 设置布局面板的尺寸大小。 |
| panel | region | 返回特定的布局面板，'region'参数的可取值为：'north','south','east','west','center'。 |
| collapse | region | 折叠特定的布局面板，'region'参数的可取值为：'north','south','east','west'。 |
| expand | region | 延伸特定的布局面板，'region'参数的可取值为：'north','south','east','west'。 |

# 菜单和按钮

## 菜单

**菜单**

使用$.fn.menu.defaults重载默认值。



**使用方法**

**创建菜单**

使用HTML标签创建菜单必须对<div>标签引用'easyui-menu'类，每个菜单项都将使用<div>标签创建。 为菜单添加'iconCls'属性可以指定一个显示在菜单项左侧的图标。 对菜单引用'menu-sep'类将生成一条菜单分隔线。

1. **<div** id="mm" class="easyui-menu" style="width:120px;"**>**
2. **<div>**New**</div>**
3. **<div>**
4. **<span>**Open**</span>**
5. **<div** style="width:150px;"**>**
6. **<div><b>**Word**</b></div>**
7. **<div>**Excel**</div>**
8. **<div>**PowerPoint**</div>**
9. **</div>**
10. **</div>**
11. **<div** iconCls="icon-save"**>**Save**</div>**
12. **<div** class="menu-sep"**></div>**
13. **<div>**Exit**</div>**
14. **</div>**

使用脚本创建菜单并且监听'onClick'事件

1. $('#mm').menu({
2. onClick:**function**(item){
3. //...
4. }
5. });

**显示菜单**

当一个菜单创建完毕，它是隐藏而不可见的，调用'show'方法可以显示菜单。

1. $('#mm').menu('show', {
2. left: 200,
3. top: 100
4. });

**属性**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| zIndex（堆叠顺序） | number（数字） | 菜单的z-index样式,从第一个菜单开始递增。 | 110000 |
| left（左边距） | number（数字） | 菜单的左边距。 | 0 |
| top（顶边距） | number | 菜单的顶边距。 | 0 |

**事件**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onShow | none | 当一个菜单被显示后触发。 |
| onHide | none | 在一个菜单被隐藏后触发。 |
| onClick | item | 当一个菜单被点击时触发。 |

**方法**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| show | pos | 在一个特定的位置显示菜单。 pos参数有2个属性： left：新的左边距。 top：新的顶边距。 |
| hide | none | 隐藏一个菜单。 |
| getItem | itemEl | 获取菜单项数据并将其返回, 数据包含以下属性： target：DOM对象，菜单项目。 id：字符串，分配给元素的ID。 text：字符串，菜单项的文本。 href：字符串，超链接地址。 disabled：布尔型，菜单项是启用还是禁用。 onclick：函数, 当用户点击菜单时将要执行的函数。 iconCls：字符串，图标css类。 |
| setText | param | 设置特定的菜单文本，'param'参数包含2个属性： target：DOM对象，将要被设置的菜单项。 text：字符串，新的文本值。 |
| setIcon | param | 设置特定的菜单项图标，'param'参数包含2个属性： target：DOM对象，菜单项目。 iconCls: 新的图标css类。 |
| findItem | text | 查找特定的菜单项，和getItem方法返回的对象相同。 |
| appendItem | param | 添加菜单，'param'参数包含以下属性： parent: 待添加新菜单的DOM对象，如果没有被设置，新菜单项将会被作为顶级菜单添加。 text：字符串，菜单项文本。 href：字符串，超链接地址。 onclick：字符串或者函数，当用户点击菜单项时将要被执行的脚本代码或者函数。 iconCls: string, the icon class. |
| removeItem | itemEl | 移除特定的菜单项。 |
| enableItem | itemEl | 启用菜单项。 |
| disableItem | itemEl | 禁用菜单项。 |

## 链接按钮

**链接按钮**

使用$.fn.linkbutton.defaults重载默认值



**使用方法**

1. **<a** href="#" id="btn" iconCls="icon-search"**>**easyui**</a>**
2. $('#btn').linkbutton({
3. plain:**true**
4. });
5. $('#btn').linkbutton('disable');    //禁用按钮
6. $('#btn').linkbutton('enable');     //启用按钮

**属性**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| id（ID） | string（字符串） | 分配给该组件的ID。 | null |
| disabled（禁用） | boolean（布尔型） | 设置为true将禁用按钮。 | false |
| plain（简洁模式） | boolean（布尔型） | 设置为true将显示简洁效果。 | false |
| text（文本） | string（字符串） | 显示在按钮上的文本。 | '' |
| iconCls（图标CSS类） | string（字符串） | 用来在左边显示一个16x16大小图标的css类。 | null |

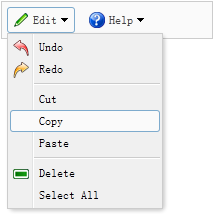
**方法**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回属性对象。 |
| disable | none | 禁用按钮。 |
| enable | none | 启用按钮。 |

## 菜单按钮

**菜单按钮**

继承$.fn.linkbutton.defaults，使用$.fn.menubutton.defaults重载默认值。



**依赖关系**

* 菜单
* 链接按钮

**使用方法**

1. **<a** href="javascript:void(0)" id="mb" iconCls="icon-edit"**>**Edit**</a>**
2. **<div** id="mm" style="width:150px;"**>**
3. **<div** iconCls="icon-undo"**>**Undo**</div>**
4. **<div** iconCls="icon-redo"**>**Redo**</div>**
5. **<div** class="menu-sep"**></div>**
6. **<div>**Cut**</div>**
7. **<div>**Copy**</div>**
8. **<div>**Paste**</div>**
9. **<div** class="menu-sep"**></div>**
10. **<div** iconCls="icon-remove"**>**Delete**</div>**
11. **<div>**Select All**</div>**
12. **</div>**
13. $('#mb').menubutton({
14. menu: '#mm'
15. });

**属性**

继承linkbutton的属性, 以下是菜单按钮独有的属性。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| plain（简洁模式） | boolean（布尔型） | 设置为true将显示简洁效果。 | true |
| menu（菜单） | string（字符串） | 用来创建一个相应菜单的选择器。 | null |
| duration（持续时间） | number（数字） | 定义鼠标划过按钮时显示菜单所持续的时间，单位为毫秒。 | 100 |

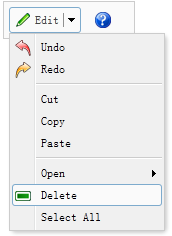
**方法**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回属性对象。 |
| disable | none | 禁用菜单按钮。 |
| enable | none | 启用菜单按钮。 |

## 分隔按钮

**分隔按钮**

继承$.fn.linkbutton.defaults，使用$.fn.splitbutton.defaults重载默认值。



**依赖关系**

* 菜单
* 链接按钮

**使用方法**

1. **<a** href="javascript:void(0)" id="sb" iconCls="icon-ok" onclick="javascript:alert('ok')"**>**Ok**</a>**
2. **<div** id="mm" style="width:100px;"**>**
3. **<div** iconCls="icon-ok"**>**Ok**</div>**
4. **<div** iconCls="icon-cancel"**>**Cancel**</div>**
5. **</div>**
6. $('#sb').splitbutton({
7. menu:'#mm'
8. });

**Properties**

属性继承链接按钮, 以下是分隔按钮的独有属性。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| plain（简洁模式） | boolean（布尔型） | 设置为true将显示简洁效果。 | true |
| menu（菜单） | string（字符串） | 用来创建一个相应菜单的选择器。 | null |
| duration（持续时间） | number（数字） | 定义鼠标划过按钮时显示菜单所持续的时间，单位为毫秒。 | 100 |

**方法**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回属性对象。 |
| disable | none | 禁用分隔按钮。 |
| enable | none | 启用分隔按钮。 |

# 表单

## 表单

**表单**

**使用方法**

1. **<form** id="ff" method="post"**>**
2. ...
3. **</form>**

使用ajax方式提交表单

1. $('#ff').form({
2. url:...,
3. onSubmit: **function**(){
4. //进行表单验证
5. //如果返回false阻止提交
6. },
7. success:**function**(data){
8. alert(data)
9. }
10. });
11. //提交表单
12. $('#ff').submit();

普通方式提交

1. //调用表单插件的'submit'方法提交
2. $('#ff').form('submit', {
3. url:...,
4. onSubmit: **function**(){
5. //进行表单验证
6. //如果返回false阻止提交
7. },
8. success:**function**(data){
9. alert(data)
10. }
11. });

**属性**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| url（目标网址） | string（字符串） | 表单提交的目标网址 | null |

**事件**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onSubmit | none | 在表单提交之前触发，返回false将阻止表单提交。 |
| success | data | 当表单成功提交时触发。 |
| onBeforeLoad | param | 在发送载入远程数据请求之前触发，返回false将取消请求。 |
| onLoadSuccess | data | 在表单数据被载入时触发。 |
| onLoadError | none | 在载入表单数据发生错误时触发。 |

**方法**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| submit | options | 执行表单提交，options参数是一个包含以下属性的对象： url：表单提交的目标网址 onSubmit：在表单提交之前执行的回调函数 success: 在表单成功提交后执行的回调函数 |
| load | data | 载入记录填充表单。 data参数可以是字符串或者对象类型，当传入字符串时，data参数必须是一个远程url地址，否则用户必须自定义要载入的数据 |
| clear | none | 清除表单数据 |
| validate | none | 执行表单域验证，如果全部表单域有效返回true，该方法与表单验证插件一起使用。 |

## 表单验证

**表单验证**

使用$.fn.validatebox.defaults重载默认值。



**使用方法**

1. **<input** id="vv" required="true" validType="email"**>**
2. $('#vv').validatebox({
3. required:**true**
4. });

**验证规则**

验证规则使用required和validType属性来定义, 以下列出的是插件内置的验证规则。

* email: 正则匹配电子邮件。
* url: 正则匹配url。
* length[0,100]: 验证长度范围。
* remote['http://.../action.do','paramName']: 发送ajax请求来验证，验证有效时返回true。

要自定义规则，重载$.fn.validatebox.defaults。你所定义的验证规则必须定义一个验证函数和验证无效时的提示信息。 例如，定义一个验证最小长度的规则：

1. $.extend($.fn.validatebox.defaults.rules, {
2. minLength: {
3. validator: **function**(value, param){
4. **return** value.length >= param[0];
5. },
6. message: 'Please enter at least {0} characters.'
7. }
8. });

现在你可以使用最小长度验证规则来定义一个至少要输入5个字符的输入框。

1. **<input** class="easyui-validatebox" validType="minLength[5]"**>**

**属性**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| required（必填） | boolean（布尔型） | 定义表单域必须填写。 | false |
| validType（验证类型） | string（字符串） | 定义表单域的验证类型，比如：email, url等。 | null |
| missingMessage（未填提示） | string（字符串） | 当表单域未填写时出现的提示信息。 | This field is required. |
| invalidMessage（无效提示） | string（字符串） | 当表单域的内容被验证为无效时出现的提示。 | null |

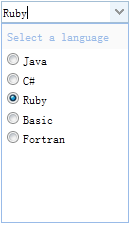
**方法**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| destroy | none | 移除并注销组件。 |
| validate | none | 验证表单域的内容是否有效。 |
| isValid | none | 调用validate方法并且返回验证结果，true或者false。 |

## 自定义组合框

**自定义组合框**

继承$.fn.validatebox.defaults，使$.fn.combo.defaults重载。



**依赖关系**

* 表单验证
* 控制面板

**使用方法**

1. **<input** id="cc" value="001"**>**
2. **<select** id="cc"**></select>**
3. $('#cc').combo({
4. required:**true**,
5. multiple:**true**
6. });

**属性**

属性继承表单验证,以下是只读组合框独有的属性。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| width（宽度） | number（数字） | 组合框的宽度。 | auto |
| panelWidth（下拉框宽度） | number（数字） | 下拉框的宽度。 | null |
| panelHeight（下拉框高度） | number（数字） | 下拉框的高度。 | 200 |
| multiple（可多选） | boolean（布尔型） | 定义是否支持多选。 | false |
| separator（分隔条） | string（字符串） | 当支持多选时，用以分隔文本的字符。 | , |
| editable（可编辑） | boolean（布尔型） | 定义用户是否可以直接向组合框输入内容。 | true |
| disabled（禁用） | boolean（布尔型） | 定义是否禁用输入框。 | false |
| hasDownArrow（下拉图标） | boolean（布尔型） | 定义是否显示下拉按钮。 | true |
| value（值） | string（字符串） | 默认值。 |  |
| delay（延迟） | number（数字） | 最后一次输入事件与搜索（自动完成）之间的延迟间隔。 | 200 |
| keyHandler（按键助手） | object（对象） | 当用户按下一个键位时调用的函数，默认的按键助手定义如下：  keyHandler: {  up: function(){},  down: function(){},  enter: function(){},  query: function(q){}  } |  |

**事件**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onShowPanel | none | 当显示下拉面板的时候触发。 |
| onHidePanel | none | 当隐藏下拉面板的时候触发。 |
| onChange | newValue, oldValue | 当组合框的值发生改变时触发。 |

**方法**

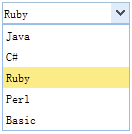
方法继承表单验证，以下是只读组合框独有的方法。

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回属性对象。 |
| panel | none | 返回下拉面板对象。 |
| textbox | none | 返回文本框对象。 |
| destroy | none | 销毁组件。 |
| resize | width | 调整组件的宽度。 |
| showPanel | none | 显示下拉面板。 |
| hidePanel | none | 隐藏下拉面板。 |
| disable | none | 禁用组合框。 |
| enable | none | 启用组合框。 |
| validate | none | 验证输入的值。 |
| isValid | none | 返回验证结果。 |
| clear | none | 清除组件的值。 |
| getText | none | 获取组合框的值。 |
| setText | text | 设置组合框的值。 |
| getValues | none | 返回一个存储下拉面板值的数组。 |
| setValues | values | 设置下拉面板的值，参数values是一个数组。 |
| getValue | none | 获取下拉面板的值。 |
| setValue | value | 设置下拉面板的值。 |

## 可装载组合框

**可装载组合框**

继承$.fn.combo.defaults，使用$.fn.combobox.defaults重载默认值。



**依赖关系**

* 自定义组合框

**使用方法**

1. **<select** id="cc" name="dept" style="width:200px;"**>**
2. **<option** value="aa"**>**aitem1**</option>**
3. **<option>**bitem2**</option>**
4. **<option>**bitem3**</option>**
5. **<option>**ditem4**</option>**
6. **<option>**eitem5**</option>**
7. **</select>**
8. **<input** id="cc" name="dept" value="aa"**>**
9. $('#cc').combobox({
10. url:'combobox\_data.json',
11. valueField:'id',
12. textField:'text'
13. });

The json data format sample:

1. [{
2. "id":1,
3. "text":"text1"
4. },{
5. "id":2,
6. "text":"text2"
7. },{
8. "id":3,
9. "text":"text3",
10. "selected":**true**
11. },{
12. "id":4,
13. "text":"text4"
14. },{
15. "id":5,
16. "text":"text5"
17. }]

**属性**

继承自定义组合框的属性, 以下是可装载组合框的独有属性。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| valueField（值域） | string（字符串） | 绑定到组合框下的数据的值域的名称。 | value |
| textField（文本域） | string（字符串） | 绑定到组合框下数据的文本域的名称。 | text |
| mode（模式） | string（字符串） | 定义当文本发生改变时，如何载入数据列表。如果组合框要从远程地址载入数据，设置为'remote'。 | local |
| url（超链接地址） | string（字符串） | 可供载入数据列表的远程地址。 | null |
| method（方法） | string（字符串） | 获取数据的HTTP方法。 | post |
| data（数据） | array（数组） | 将被载入的数据列表。 | null |
| filter（过滤器） | function（函数） | 定义当模式设置为'local'时如何过滤本地数据，function有2个参数： q: 用户输入的文本。 row: 数据列表 返回true将允许显示数据列表。 |  |
| formatter(格式) | function（函数） | 定义如何返回数据，function有一个参数：row。 |  |

**事件**

事件继承自定义组合框，以下是可装载组合框的独有事件。

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onLoadSuccess | none | 当远程数据载入成功时触发。 |
| onLoadError | none | 当远程数据载入错误时触发。 |
| onSelect | record | 当用户选择一个列表项时触发。 |
| onUnselect | record | 当用户取消选择一个列表项时触发。 |

**方法**

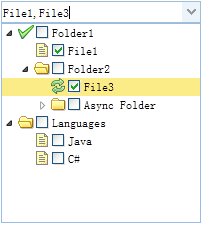
方法继承自定义组合框，以下是可装载组合框独有的或者重载自定义组合框的方法。

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回属性对象。 |
| getData | none | 返回已载入的数据。 |
| loadData | data | 载入本地数据列表。 |
| reload | url | 请求远程数据列表。 |
| setValues | values | 设置组合框的值，参数values是一个数组。 |
| setValue | value | 设置组合框的值。 |
| clear | none | 清除组合框的值。 |
| select | value | 选择特定的项目。 |
| unselect | value | 取消选择特定的项目。 |

## 组合树

**组合树**

继承$.fn.combo.defaults和$.fn.tree.defaults，使用$.fn.combotree.defaults重载默认值。



**依赖关系**

* 自定义组合框
* 树形菜单

**使用方法**

1. **<select** id="cc" style="width:200px;"**></select>**
2. **<input** id="cc" value="01"**>**
3. $('#cc').combotree({
4. url:'tree\_data.json'
5. });

**属性**

属性继承自定义组合框和树形菜单，以下是组合树重载的属性。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| editable（可编辑） | boolean（布尔型） | 定义用户是否可以直接输入。 | false |

**事件**

事件继承自定义组合框和树形菜单

**方法**

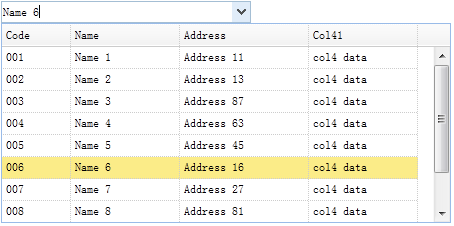
方法继承自定义组合框，以下是组合树独有的或者重载自定义组合框的方法。

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回属性对象。 |
| tree | none | 返回树形菜单对象。 |
| loadData | data | 载入本地树形菜单数据。 |
| reload | url | 再次向远程树形菜单数据发起请求。 |
| clear | none | 清除组件值。 |
| setValues | values | 设置组合框的值，参数values是一个数组。 |
| setValue | value | 设置组合框的值。 |

## 组合表格

**组合表格**

继承$.fn.combo.defaults和$.fn.datagrid.defaults，使用$.fn.combogrid.defaults重载默认值。



**依赖关系**

* 自定义组合框
* 数据表格

**使用方法**

1. **<select** id="cc" name="dept" style="width:250px;"**></select>**
2. **<input** id="cc" name="dept" value="01"**>**
3. $('#cc').combogrid({        panelWidth:450,     value:'006',            idField:'code',     textField:'name',       url:'datagrid\_data.json',       columns:[[          {field:'code',title:'Code',width:60},           {field:'name',title:'Name',width:100},          {field:'addr',title:'Address',width:120},           {field:'col4',title:'Col41',width:100}      ]]  });

**属性**

属性继承自定义组合框和数据表格，以下是组合表格独有的属性。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| loadMsg（载入时信息） | string（字符串） | 当数据表格载入远程数据时的提示信息。 | null |
| idField（id字段） | string（字符串） | 标识表格行数据唯一性的字段名。 | null |
| textField（文本域） | string（字符串） | 显示在文本框中的文本域的名称。 | null |
| mode（模式） | string（字符串） | 定义当文本发生改变时如何载入组合表格的数据。如果组合表格要从远程服务器载入数据，设置为'remote'。 | local |
| filter（过滤器） | function(q, row) | 定义当模式设置为'local'时如何选择本地数据，返回true将选择一行。 |  |

**事件**

事件继承自定义组合框和数据表格。

**方法**

方法继承自定义组合框，以下是组合表格独有的或者重载自定义组合框的方法。

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回属性对象。 |
| grid | none | 返回数据表格对象。 |
| setValues | values | 设置组合框的值，参数values是一个数组。 |
| setValue | value | 设置组合框的值 |
| clear | none | 清除组合框的值。 |

﻿

## 数字验证框

**数字验证框**

继承$.fn.validatebox.defaults，使用$.fn.numberbox.defaults重载默认值。

**依赖关系**

* 表单验证

**使用方法**

1. **<input** type="text" id="nn"**></input>**
2. $('#nn').numberbox({
3. min:0,
4. precision:2
5. });

**属性**

属性继承表单验证，以下是数字验证框的独有属性。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| disabled（禁用） | boolean（布尔型） | 定义是否禁用输入框。 | false |
| min（最小值） | number（数字） | 允许的最小值。 | null |
| max（最大值） | number（数字） | 允许的最大值。 | null |
| precision（精度） | number（数字） | 精确到小数点后的最大位数。 | 0 |

**Methods（方法）**

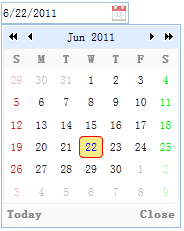
方法继承表单验证，以下是数字验证独有的或者重载表单验证的方法。

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| disable | none | 禁用输入框。 |
| enable | none | 启用输入框。 |
| fix | none | 将输入框的值修正为一个有效值。 |

## 日期组合框

**日期组合框**

继承$.fn.combo.defaults，使用$.fn.datebox.defaults重载默认值。



**依赖关系**

* 自定义组合框
* 日历

**使用方法**

1. **<input** id="dd" type="text"**></input>**
2. $('#dd').datebox({
3. required:**true**
4. });

**属性**

属性继承自定义组合框，以下是日期组合框独有的属性。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| panelWidth（面板宽度） | number（数字） | 下拉日历面板的宽度。 | 180 |
| panelHeight（面板高度） | number（数字） | 下拉日历面板的高度。 | auto |
| currentText（当前文本） | string（字符串） | 显示在日历左下角用以选择今天日期的按钮上的文本。 | Today |
| closeText（关闭） | string（字符串） | 显示在关闭按钮上的文本。 | Close |
| okText（OK按钮文本） | string（字符串） | 显示在OK按钮上的文本。 | Ok |
| disabled（禁用） | boolean（布尔型） | 当设置为true时禁用输入框。 | false |
| formatter（格式） | function（函数） | 一个格式化日期的函数，该函数的参数是一个被选择的日期并且一个字符串。 |  |
| parser（解析器） | function（函数） | 一个解析日期的字符串，该函数的参数是一个被选择的日期并且返回一个解析后的日期值。 |  |

**事件**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onSelect | date | 当用户选择日期时触发。 |

**方法**

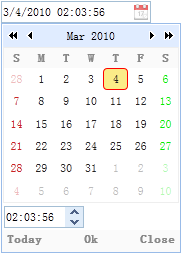
方法继承自定义组合框，以下是重载自定义组合框的方法。

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回属性对象。 |
| calendar | none | 获取日历对象。 |
| setValue | value | 设置日期组合框的值。 |

## 日期时间组合框

**日期时间组合框**

继承$.fn.datebox.defaults，使用$.fn.datetimebox.defaults重载默认值。



**依赖关系**

* 日期组合框
* 事件调节器

**使用方法**

1. **<input** id="dt" type="text"**></input>**
2. $('#dt').datetimebox({
3. showSeconds:**false**
4. });

**属性**

属性继承日期组合框，以下是日期时间组合框独有的属性。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| showSeconds（显示秒） | boolean（布尔型） | 定义是否显示秒数信息。 | true |

**方法**

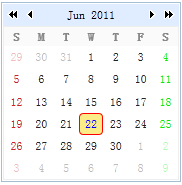
方法继承日期组合框，以下是重载日期组合框的方法。

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回属性对象。 |
| spinner | none | 返回时间调节器对象。 |
| setValue | value | 设置日期时间组合框的值。 |

## 日历

**日历**

使用$.fn.calendar.defaults重载默认值。



**使用方法**

1. **<div** id="cc" style="width:180px;height:180px;"**></div>**
2. $('#cc').calendar({
3. width:600,
4. height:600,
5. current:**new** Date()
6. });

**属性**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| width（宽度） | number（数字） | 日历组件的宽度。 | 180 |
| height（高度） | number（数字） | 日历组件的高度。 | 180 |
| fit（铺满浏览器） | boolean（布尔型） | 当设置为true时，日历组件的大小将铺满装载它的父容器（浏览器）。 | false |
| border（边框） | boolean（布尔型） | 定义是否显示边框。 | true |
| weeks（星期代码列表） | array（数组） | 显示星期几的代码列表。 | ['S','M','T','W','T','F','S'] |
| months（月份代码列表） | array（数组） | 显示月份的代码列表。 | ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'] |
| year（年份） | number（数字） | 日历组件的起始年份 | 当前年份(4位数字) |
| month（月份） | number（数字） | 日历组件默认显示的月份。 | 当前月份，从1开始。 |
| current(当前日期) | Date（日期） | 当前日期。 | 当前日期。 |

**事件**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onSelect | date | 当用户选择日期的时候触发。 |

**方法**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回属性对象。 |
| resize | none | 重置日历的大小。 |
| moveTo | date | 将日历设置到一个指定的日期。 |

## 调节器

**调节器**

继承$.fn.validatebox.defaults，使用$.fn.spinner.defaults重载默认值。

**依赖关系**

* 表单验证

**使用方法**

1. **<input** id="ss" value="2"**>**
2. $('#ss').spinner({
3. required:**true**,
4. increment:10
5. });

**属性**

属性继承表单验证，以下是调节器的独有属性。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| width（宽度） | number（数字） | 组件的宽度 | auto |
| value（值） | string（字符串） | 初始值。 |  |
| min（最小值） | string（字符串） | 允许的最小值。 | null |
| max（最大值） | string（字符串） | 允许的最大值。 | null |
| increment（增量） | number（数字） | 当点击调节器按钮时的增量值。 | 1 |
| editable（可编辑） | boolean（布尔型） | 定义用户是否可以直接输入。 | true |
| disabled（禁用） | boolean（布尔型） | 定义是否禁用输入框。 | false |
| spin（调节函数） | function(down) | 当用户点击调节按钮时调用的函数。 |  |

**事件**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onSpinUp | none | 当用户点击增加按钮时触发。 |
| onSpinDown | none | 当用户点击减少按钮时触发。 |

**方法**

方法继承表单验证，以下是调节器独有的方法。

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回属性对象。 |
| destroy | none | 销毁组件。 |
| resize | width | 调整组件宽度。 |
| enable | none | 启用组件。 |
| disable | none | 禁用组件。 |
| getValue | none | 获取组件的值。 |
| setValue | value | 设置组件的值。 |
| clear | none | 清除组件的值。 |

## 数字调节器

**数字调节器**

继承$.fn.spinner.defaults和$.fn.numberbox.defaults，使用$.fn.numberspinner.defaults重载默认值。



**依赖关系**

* 调节器
* 数字验证框

**使用方法**

1. **<input** id="ss" min="10" max="100" required="true" style="width:80px;"**>**
2. $('#ss').numberspinner({
3. editable:**false**
4. });

**属性**

属性继承调节器和数字验证框。

**事件**

事件继承调节器。

**方法**

方法继承调节器，以下是重载调节器的方法。

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回属性对象。 |
| setValue | value | 设置数字调节器的值。 |

## 时间调节器

**时间调节器**

继承$.fn.spinner.defaults，使用$.fn.timespinner.defaults重载默认值。



**依赖关系**

* 调节器

**使用方法**

1. **<input** id="ss" required="true" style="width:80px;"**>**
2. $('#ss').timespinner({
3. showSeconds:**true**
4. });

**属性**

属性继承调节器，以下是重载调节器的属性。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| separator（分隔符） | string（字符串） | 时分秒直接的分隔符号。 | : |
| showSeconds（显示秒数） | boolean（布尔型） | 定义是否显示描述信息。 | false |
| highlight（高亮） | number（数字） | 显示高亮背景的时间要素，0=时，1=分，… | 0 |

**事件**

事件继承调节器

**方法**

方法继承调节器，以下是重载调节器的方法。

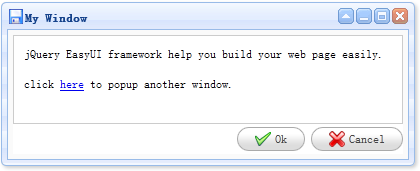
|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回属性对象。 |
| setValue | value | 设置时间调节器的值。 |
| getHours | none | 获取当前小时的值。 |
| getMinutes | none | 获取当前分钟的值。 |
| getSeconds | none | 获取当前秒钟的值。 |

# 窗口

## 窗口

**窗口**

继承$.fn.panel.defaults，使用$.fn.window.defaults重载默认值。



**依赖关系**

* 一般拖动
* 缩放
* 控制面板

**使用方法**

1. **<div** id="win" iconCls="icon-save" title="My Window"**>**
2. Window Content
3. **</div>**
4. $('#win').window({
5. width:600,
6. height:400,
7. modal:**true**
8. });
9. $('#win').window('close');  //关闭窗口

**属性**

继承控制面板，以下是窗口独有的或者重载控制面板的方法。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| title（标题） | string（字符串） | 窗口的标题。 | New Window |
| collapsible（可折叠） | boolean（布尔型） | 定义是否显示折叠按钮。 | true |
| minimizable（最小化） | boolean（布尔型） | 定义是否显示最小化按钮。 | true |
| maximizable（最大化） | boolean（布尔型） | 定义是否显示最大化按钮。 | true |
| closable（关闭） | boolean（布尔型） | 定义是否显示关闭按钮。 | true |
| closed（已关闭） | boolean（布尔型） | 定义是否在初始化组件时关闭（隐藏）窗口 | false |
| zIndex（堆叠顺序） | number（数字） | 窗口的堆叠顺序，从第一个窗口的zIndex值开始递增。 | 9000 |
| draggable（可拖放） | boolean（布尔型） | 定义窗口是否可以被拖放。 | true |
| resizable（可缩放） | boolean（布尔型） | 定义窗口是否可以被缩放。 | true |
| shadow（阴影） | boolean（布尔型） | 如果设置为true，显示窗口的时候将显示阴影。 | true |
| inline（行内显示） | boolean（布尔型） | 定义如何布局窗口，如果设置为true，窗口将显示在它的父容器中，否则将显示在所有元素的上面。 | false |
| modal（遮罩） | boolean（布尔型） | 定义窗口是否带有遮罩效果。 | true |

**事件**

事件继承控制面板。

**方法**

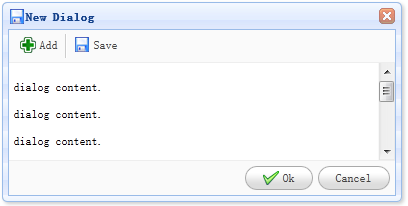
方法继承控制面板，以下是窗口独有的方法。

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| window | none | 返回窗口对象。 |

## 对话窗口

**对话窗口**

继承$.fn.window.defaults，使用$.fn.dialog.defaults重载默认值。



**依赖关系**

* 窗口
* 链接按钮

**使用方法**

1. **<div** id="dd" title="My Dialog" style="width:400px;height:200px;"**>**
2. Dialog Content.
3. **</div>**
4. $('#dd').dialog({
5. modal:**true**
6. });

**属性**

属性继承窗口，以下是重载窗口的属性。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| title（标题） | string（字符串） | 对话标题。 | New Dialog |
| collapsible（可折叠） | boolean（布尔型） | 定义是否显示折叠按钮。 | false |
| minimizable（最小化） | boolean（布尔型） | 定义是否显示最小化按钮。 | false |
| maximizable（最大化） | boolean（布尔型） | 定义是否显示最大化按钮。 | false |
| resizable（可缩放） | boolean（布尔型） | 定义对话窗口是否可以被缩放。 | false |
| toolbar（工具栏） | array（数组） | 对话窗口顶部的工具栏，每个工具的属性都跟链接按钮的属性一样。 | null |
| buttons（按钮） | array（数组） | 对话窗口底部的按钮，每个按钮的属性都跟链接按钮一样。 | null |

**事件**

事件继承窗口。

**方法**

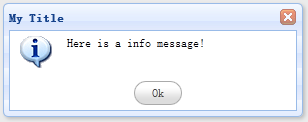
方法继承窗口，以下是对话窗口独有的方法。

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| dialog | none | 返回对话窗口对象。 |

## 消息窗口

**消息窗口**

使用$.messager.defaults重载默认值。



**依赖关系**

* 窗口
* 链接按钮
* 进度条

**使用方法**

1. $.messager.alert('Warning','The warning message');
2. $.messager.confirm('Confirm','Are you sure you want to delete record?',**function**(r){
3. **if** (r){
4. alert('ok');
5. }
6. });

**属性**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| ok（确认） | string（字符串） | 显示在“确认”按钮上的文本。 | Ok |
| cancel（取消） | string（字符串） | 显示在“取消”按钮上的文本。 | Cancel |

**方法**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| $.messager.show | options | 在屏幕的右下方显示一个消息窗口 ，options参数是一个可配置的对象: showType：定义如何显示消息窗口，可选值: null,slide,fade,show，默认为slide。 showSpeed：定义窗口显示的时间，单位毫秒，默认为 600。 width：定义消息窗口的宽度，默认为250。 height：定义消息窗口的高度，默认为100。 msg：显示在消息窗口的文本。 title：显示在窗口头部的标题。 timeout：如果定义为0，消息窗口将不会关闭直到用户点击关闭为止。如果定义为非0，消息窗口在超时后将自动关闭。 |
| $.messager.alert | title, msg, icon, fn | 显示警告窗口。参数如下: title：显示在窗口头部的标题文本。 msg：显示在窗口中的文本。 icon：显示的图片，可选值：error,question,info,warning。 fn：当窗口关闭时触发的回调函数。 |
| $.messager.confirm | title, msg, fn | 显示一个带有确认和取消按钮的确认信息窗口。参数如下： title：显示在窗口头部的文本。 msg：显示在窗口中的文本。 fn(b)：回调函数，当用户点击确认按钮时，传递一个true值给回调函数，否则传递一个false值。 |
| $.messager.prompt | title, msg, fn | 显示一个带有确认和取消的输入信息窗口。 参数如下： title：显示在窗口头部的标题文本。 msg：显示在窗口中的信息 fn(val)：接受用户输入作为参数的回调函数。 |
| $.messager.progress | options or method | 显示一个带进度条信息的窗口。 属性定义如下： title：显示在面板头部的标题文本，默认为''。 msg：显示在主窗体的文本，默认为''。 text：显示在进度条中的信息默认为undefined。 interval：每次进度增加所耗费的时间，单位为毫秒，默认为300.  方法定义如下： bar：获取进度条对象。 close：关闭进程窗口。  **示例代码**  显示进程消息窗口。  $.messager.progress();  使用如下代码关闭窗口。  $.messager.progress('close'); |

# 数据表格和树形菜单

## 数据表格

**数据表格**

继承$.fn.panel.defaults，使用$.fn.datagrid.defaults重载默认值。

**依赖关系**

* 控制面板
* 缩放
* 链接按钮
* 分页

**使用方法**

1. **<table** id="tt"**></table>**
2. $('#tt').datagrid({
3. url:'datagrid\_data.json',
4. columns:[[
5. {field:'code',title:'Code',width:100},
6. {field:'name',title:'Name',width:100},
7. {field:'price',title:'Price',width:100,align:'right'}
8. ]]
9. });

**数据表格属性**

属性继承控制面板，以下是数据表格独有的属性。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| columns（列） | array（数组） | 数据表格列配置对象，查看列属性以获取更多细节。 | null |
| frozenColumns（固定列） | array（数组） | 跟列属性一样，但是这些列固定在左边，不会滚动。 | null |
| fitColumns（自适应列宽） | boolean（布尔型） | 设置为true将自动使列适应表格宽度以防止出现水平滚动。 | false |
| striped（显示条纹） | boolean（布尔型） | 设置为true将交替显示行背景。 | false |
| method（方法） | string（字符串） | 请求远程数据的方法类型。 | post |
| nowrap（截取） | boolean（布尔型） | 设置为true，当数据长度超出列宽时将会自动截取。 | true |
| idField（id字段） | string（字符串） | 表明该列是一个唯一列。 | null |
| url（超链接） | string（字符串） | 一个用以从远程站点请求数据的超链接地址。 | null |
| loadMsg（载入时信息） | string（字符串） | 当从远程站点载入数据时，显示的一条快捷信息。 | Processing, please wait … |
| pagination（分页） | boolean（布尔型） | 设置true将在数据表格底部显示分页工具栏。 | false |
| rownumbers（行数） | boolean（布尔型） | 设置为true将显示行数。 | false |
| singleSelect（单选模式） | boolean（布尔型） | 设置为true将只允许选择一行。 | false |
| pageNumber（当前页码） | number（数字） | 当设置分页属性时，初始化分页码。 | 1 |
| pageSize（每页记录数） | number（数字） | 当设置分页属性时，初始化每页记录数。 | 10 |
| pageList（可选择的每页记录数） | array | 当设置分页属性时，初始化每页记录数列表。 | [10,20,30,40,50] |
| queryParams（查询参数） | object（对象） | 当请求远程数据时，发送的额外参数。 | {} |
| sortName（默认排序） | string（字符串） | 当数据表格初始化时以哪一列来排序。 | null |
| sortOrder（排序顺序） | string（字符串） | 定义排序顺序，可以是'asc'或者'desc'（正序或者倒序）。 | asc |
| remoteSort（远程排序） | boolean（布尔型） | 定义是否通过远程服务器对数据排序。 | true |
| showFooter（显示行底） | boolean（布尔型） | 定义是否显示行底（如果是做统计表格，这里可以显示总计等）。 | false |
| rowStyler（行样式） | function（函数） | 返回样式，如：'background:red'，function有2个参数： index：行索引，从0开始. row：对应于该行记录的对象。 |  |
| loadFilter（载入过滤器） | function（函数） | 返回用以显示的已过滤数据，function有一个参数'data'表示原始数据，你可以将原始数据改变为规范的数据格式，该函数必须返回包含 'total'和'rows'属性的标准数据对象。 |  |
| editors（编辑模式） | object（对象） | 定义当编辑一行时的编辑模式。 | predefined editors |
| view（视图） | object（对象） | 定义数据表格的视图。 | default view |

**列属性**

数据表格的列是一个对象数组，即这个对象中的元素也是一个数组（js中数组是对象）。 对象数组中的每一个元素都是可配置的对象，每个可配置对象定义一个列。

示例代码:

1. columns:[[
2. {field:'itemid',title:'Item ID',rowspan:2,width:80,sortable:**true**},
3. {field:'productid',title:'Product ID',rowspan:2,width:80,sortable:**true**},
4. {title:'Item Details',colspan:4}
5. ],[
6. {field:'listprice',title:'List Price',width:80,align:'right',sortable:**true**},
7. {field:'unitcost',title:'Unit Cost',width:80,align:'right',sortable:**true**},
8. {field:'attr1',title:'Attribute',width:100},
9. {field:'status',title:'Status',width:60}
10. ]]

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| title（标题） | string（字符串） | 列标题。 | undefined |
| field（字段） | string（字符串） | 列字段。 | undefined |
| width（宽度） | number（数字） | 列宽。 | undefined |
| rowspan（跨行数） | number（数字） | 表明一个单元格跨几行。 | undefined |
| colspan（跨列数） | number（数字） | 表明一个单元格跨几列。 | undefined |
| align（对其方式） | string（字符串） | 表明如何对其列数据，可选值：'left'，'right'，'center'。 | undefined |
| sortable（可排序） | boolean（布尔型） | 设置为true允许对该列排序。 | undefined |
| resizable（缩放） | boolean（布尔型） | 设置为true允许该列被缩放。 | undefined |
| hidden（隐藏） | boolean（布尔型） | 设置为true将隐藏列。 | undefined |
| checkbox（复选框） | boolean（布尔型） | 设置为true将显示复选框。 | undefined |
| formatter（格式化） | function（函数） | 格式化单元格函数，有3个参数： value：字段的值。 rowData：行数据。 rowIndex：行索引。 | undefined |
| styler（样式） | function（函数） | 单元格样式函数，返回样式字符串装饰表格如'background:red'，function有3个参数： value：字段值。 rowData：行数据。 rowIndex：行索引。 | undefined |
| sorter（排序器） | function（函数） | T自定义字段排序函数，有2个参数： a：该列的第一个值。 b：该列的第二个值。 | undefined |
| editor（编辑器） | string,object（字符串，对象） | 表明编辑类型。如果属性是字符串类型表示编辑类型，如果是对象则包含2个参数： type：字符串，编辑类型，可选值：text，textarea，checkbox，numberbox，validatebox，datebox，combobox，combotree。 options：对象，对象于编辑类型的编辑器属性。 | undefined |

**编辑器**

使用$.fn.datagrid.defaults.editors重载默认值。

每个编辑器都有以下方法：

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| init | container, options | 初始化编辑器并返回目标对象。 |
| destroy | target | 销毁编辑器。 |
| getValue | target | 获取编辑框的值。 |
| setValue | target , value | 设置编辑框的值。 |
| resize | target , width | 调整编辑器 |

例如，如下代码将定义一个文本编辑器：

1. $.extend($.fn.datagrid.defaults.editors, {
2. text: {
3. init: **function**(container, options){
4. **var** input = $('<input type="text" class="datagrid-editable-input">').appendTo(container);
5. **return** input;
6. },
7. getValue: **function**(target){
8. **return** $(target).val();
9. },
10. setValue: **function**(target, value){
11. $(target).val(value);
12. },
13. resize: **function**(target, width){
14. **var** input = $(target);
15. **if** ($.boxModel == **true**){
16. input.width(width - (input.outerWidth() - input.width()));
17. } **else** {
18. input.width(width);
19. }
20. }
21. }
22. });

**数据表格视图**

使用$.fn.datagrid.defaults.view重载默认值。

视图是一个告诉数据表格如何呈现行记录的对象，对象必须定义以下方法：

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| render | target, container, frozen | 当数据载入时调用。 target: DOM对象，数据网格对象。 container: 行记录容器。 frozen: 是否呈现固定容器。 |
| renderFooter | target, container, frozen | 这是一个可选函数用以展现行底。 |
| renderRow | target, fields, frozen, rowIndex, rowData | 这是一个可选函数，它可以被render函数调用。 |
| refreshRow | target, rowIndex | 定义如何刷新指定的行。 |
| onBeforeRender | target, rows | 在视图被呈现之前触发。 |
| onAfterRender | target | 在视图被程序之后触发。 |

**事件**

事件继承控制面板，以下是数据表格独有的属性。

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onLoadSuccess | data | 当数据载入成功时触发。 |
| onLoadError | none | 当载入远程数据发生错误时触发。 |
| onBeforeLoad | param | 在请求载入数据之前触发，如果返回false将取消载入。 |
| onClickRow | rowIndex, rowData | 当用户点击行时触发，参数如下： rowIndex：被点击的行索引，从0开始。 rowData：对应于被点击的行的记录。 |
| onDblClickRow | rowIndex, rowData | 当用户双击一行时触发，参数如下： rowIndex：被点击的行索引，从0开始。 rowData：对应于被点击的行的记录。 |
| onClickCell | rowIndex, field, value | 当用户点击单元格时触发。 |
| onDblClickCell | rowIndex, field, value | 当用户双击单元格时触发。 |
| onSortColumn | sort, order | 当用户对列排序时触发，参数如下： sort：排序字段名称。 order：排序顺序。 |
| onResizeColumn | field, width | 当用户调整列宽时触发。 |
| onSelect | rowIndex, rowData | 当用户选择一行是触发，参数如下： rowIndex：被选择的行索引，从0开始。 rowData：对应于被选择行的记录。 |
| onUnselect | rowIndex, rowData | 当用户取消选择一行时触发，参数如下： rowIndex：被取消选择的行索引，从0开始。 rowData：对应于被取消选择行的记录。 |
| onSelectAll | rows | 当用户选择所有行时触发。 |
| onUnselectAll | rows | 当用户取消选择所有行时触发。 |
| onBeforeEdit | rowIndex, rowData | 当用户开始编辑一行时触发，参数如下： rowIndex：正在编辑的行索引，从0开始。 rowData：对应于正在编辑的行的记录。 |
| onAfterEdit | rowIndex, rowData, changes | 当用户编辑完成时触发，参数如下： rowIndex：正在编辑的行索引，从0开始。 rowData：对应于正在编辑的行的记录。 changes：被改变的字段内容，对应方式为字段：值。 |
| onCancelEdit | rowIndex, rowData | 当用户取消编辑行时触发，参数如下： rowIndex：正在编辑的行索引，从0开始。 rowData：对应于正在编辑的行的记录。 |
| onHeaderContextMenu | e, field | 当数据表格的列标题被鼠标右键单击时触发。 |
| onRowContextMenu | e, rowIndex, rowData | 当一行被鼠标右键单击时触发。 |

**方法**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回属性对象。 |
| getPager | none | 返回页面对象。 |
| getPanel | none | 返回控制面板对象。 |
| getColumnFields | frozen | 返回列字段，如果设置了frozen属性为true，将返回固定列的字段名。 |
| getColumnOption | field | 返回特定的列属性。 |
| resize | param | 缩放和布局。 |
| load | param | 载入并显示第一页的记录，如果传递了'param'参数，它将会覆盖查询参数属性的值。 |
| reload | param | 重载记录，跟'load'方法一样但是重载的是当前页的记录而非第一页。 |
| reloadFooter | footer | 重载行底记录。 |
| loading | none | 显示载入状态。 |
| loaded | none | 隐藏载入状态。 |
| fitColumns | none | 让列宽自动适应数据表格的宽度。 |
| fixColumnSize | none | 固定列尺寸。 |
| fixRowHeight | index | 固定特定列的高度。 |
| loadData | data | 载入本地数据，旧记录将被移除。 |
| getData | none | 返回已载入数据。 |
| getRows | none | 返回当前页的记录。 |
| getFooterRows | none | 返回行底记录。 |
| getRowIndex | row | 返回指定行的索引，row参数可以是行记录或者是一个id字段的值。 |
| getSelected | none | 返回第一个被选择的行记录或null。 |
| getSelections | none | 返回所有被选择的行，当没有记录被选择时，将返回一个空数组。 |
| clearSelections | none | 取消所有的已选择项。 |
| selectAll | none | 全选。 |
| unselectAll | none | 取消全选。 |
| selectRow | index | 选择一行，行索引从0开始。 |
| selectRecord | idValue | 通过传递id参数来选择一行。 |
| unselectRow | index | 取消选择一行。 |
| beginEdit | index | 开始编辑一行。 |
| endEdit | index | 结束编辑。 |
| cancelEdit | index | 取消编辑。 |
| getEditors | index | 获取指定行的编辑器，每个编辑器有如下属性： actions：编辑器可以做的行为。 target：目标编辑器jQuery对象。 field：字段名。 type：编辑器类型。 |
| getEditor | options | 获取特定的编辑器，options参数有2个属性： index：行索引。 field:字段名。 |
| refreshRow | index | 刷新一行。 |
| validateRow | index | 校验指定的行，如果有效返回true。 |
| updateRow | param | 更新指定的行，param参数包含如下属性： index：要更新的行索引。 row：新的行数据。 |
| appendRow | row | 添加一行。 |
| insertRow | param | 插入一个新行，param参数包含如下属性： index：要插入的行索引，如果没有定义则在最后面添加一个新行。 row：行数据。 |
| deleteRow | index | 删除一行。 |
| getChanges | type | 获取从最后一次提交开始的被修改的所有行，type参数表明修改的类型，可选值：inserted，deleted，updated等 。当没有传递type参数时，返回所有被修改的行。 |
| acceptChanges | none | 提交所有修改的数据，提交后的数据将不能再修改或者回滚。 |
| rejectChanges | none | 回滚所有被删除的行。 |
| mergeCells | options | 合并单元格，options参数包含如下属性： index：行索引。 field：字段名。 rowspan：整合单元格要跨的行数。 colspan：整合单元格要跨的列数。 |
| showColumn | field | 显示特定的列。 |
| hideColumn | field | 隐藏特定的列。 |

## 属性表格

**属性表格**

继承$.fn.datagrid.defaults，使用$.fn.propertygrid.defaults重载默认值。

**依赖关系**

* 数据表格

**使用方法**

1. **<table** id="pg"**></table>**
2. $('#pg').propertygrid({
3. url:'propertygrid\_data.json',
4. showGroup:**true**
5. });

**属性**

属性继承数据表格，以下是属性表格独有的属性。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| showGroup（显示分组） | boolean（布尔型） | 定义是否显示属性分组。 | false |
| groupField（分组字段） | string（字符串） | 定义要分组的字段名称。 | group |
| groupFormatter（分组格式） | function（函数） | 定义如何格式化分组中的值。 |  |

## 树形菜单

**树形菜单**

使用$.fn.tree.defaults重载默认值。

**依赖关系**

* 一般拖动
* 拖动至容器

**使用方法**

树形菜单可以用<ul>标签定义，它可以定义叶子节点和子节点，代码示例如下：

1. **<ul** id="tt"**>**
2. **<li>**
3. **<span>**Folder**</span>**
4. **<ul>**
5. **<li>**
6. **<span>**Sub Folder 1**</span>**
7. **<ul>**
8. **<li>**
9. **<span><a** href="#"**>**File 11**</a></span>**
10. **</li>**
11. **<li>**
12. **<span>**File 12**</span>**
13. **</li>**
14. **<li>**
15. **<span>**File 13**</span>**
16. **</li>**
17. **</ul>**
18. **</li>**
19. **<li>**
20. **<span>**File 2**</span>**
21. **</li>**
22. **<li>**
23. **<span>**File 3**</span>**
24. **</li>**
25. **</ul>**
26. **</li>**
27. **<li>**
28. **<span>**File21**</span>**
29. **</li>**
30. **</ul>**

也可以用不带<li>的<ul>标签来定义：

1. **<ul** id="tt"**></ul>**
2. $('#tt').tree({
3. url:'tree\_data.json'
4. });

**树形菜单的数据格式**

每个节点都拥有以下属性：

* id：节点id，对载入远程数据很重要。
* text：显示在节点的文本。
* state：节点状态，'open' or 'closed'，默认为'open'。当设置为'closed'时，拥有子节点的节点将会从远程站点载入它们。
* checked：表明节点是否被选择。
* attributes：可以为节点添加的自定义属性。
* children：子节点，必须用数组定义。

示例代码：

1. [{
2. "id":1,
3. "text":"Folder1",
4. "iconCls":"icon-save",
5. "children":[{
6. "text":"File1",
7. "checked":**true**
8. },{
9. "text":"Books",
10. "state":"open",
11. "attributes":{
12. "url":"/demo/book/abc",
13. "price":100
14. },
15. "children":[{
16. "text":"PhotoShop",
17. "checked":**true**
18. },{
19. "id": 8,
20. "text":"Sub Bookds",
21. "state":"closed"
22. }]
23. }]
24. },{
25. "text":"Languages",
26. "state":"closed",
27. "children":[{
28. "text":"Java"
29. },{
30. "text":"C#"
31. }]
32. }]

**属性**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| url（超链接） | string（字符串） | 用以载入远程数据的超链接地址。 | null |
| method（方法） | string（字符串） | 获取数据的HTTP方法。 | post |
| animate（动画） | boolean（布尔型） | 定义当节点打开或关闭时是否显示动画效果。 | false |
| checkbox（复选框） | boolean（布尔型） | 定义是否在每个节点之前显示复选框。 | false |
| cascadeCheck（级联选择） | boolean（布尔型） | 定义是否支持级联选择。 | true |
| onlyLeafCheck（只选叶子节点） | boolean（布尔型） | 定义是否只在叶子节点之前显示复选框。 | false |
| dnd（拖放） | boolean（布尔型） | 定义是否支持拖放。 | false |
| data（数据） | array（数组） | 将被载入的节点数据。 | null |

**事件**

多数事件回调函数都有'node'参数，该参数包含如下属性：

* id：节点的唯一标识。
* text：显示在节点上的文本。
* checked：节点是否被选择。
* attributes：节点的自定义属性。
* target：目标DOM对象。

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onClick | node | 当用户点击节点时触发，node参数包含如下属性： id：节点id。 text：显示在节点上的文本。 checked：节点是否被选择。 attributes：节点的自定义属性。 target：被点击的目标DOM对象。 |
| onDblClick | node | 当用户双击节点时触发。 |
| onBeforeLoad | node, param | 在请求载入数据之前触发，返回false将取消载入。 |
| onLoadSuccess | node, data | 当数据载入成功时触发。 |
| onLoadError | arguments | 当数据载入失败时触发，arguments参数跟jQuery.ajax的'error'函数一样。 |
| onBeforeExpand | node | 在节点打开之前触发，返回false将取消打开。 |
| onExpand | node | 在节点被打开时触发。 |
| onBeforeCollapse | node | 在节点被关闭之前触发，返回false将取消关闭。 |
| onCollapse | node | 当节点被关闭时触发。 |
| onCheck | node, checked | 当用户点击复选框时触发。 |
| onBeforeSelect | node | 在节点被选择之前触发，返回false将取消选择。 |
| onSelect | node | 当节点被选择时触发。 |
| onContextMenu | e, node | 当节点被鼠标右键点击时触发。 |
| onDrop | target, source, point | 当节点位置被（拖动）更换时触发。 target：DOM对象，需要被拖动动的目标节点。 source：原始节点。 point：指明拖动方式，可选值：'append'，'top'或者'bottom'。 |
| onBeforeEdit | node | 在编辑节点之前触发。 |
| onAfterEdit | node | 在编辑节点之后触发。 |
| onCancelEdit | node | 当取消编辑时触发。 |

**方法**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回树形菜单属性对象。 |
| loadData | data | 载入树形菜单数据。 |
| getNode | target | 获取特定的节点对象。 |
| getData | target | 获取特定的节点数据，包括它的子节点。 |
| reload | target | 重新载入树形菜单数据。 |
| getRoot | none | 获取根节点，返回节点对象。 |
| getRoots | none | 获取根节点，返回节点数组。 |
| getParent | target | 获取父节点，target是一个节点DOM对象。 |
| getChildren | target | 获取子节点，target参数是一个节点DOM对象。 |
| getChecked | none | 获取所有被选择的节点。 |
| getSelected | none | 获取被选择的节点并返回，如果没有节点被选择则返回null。 |
| isLeaf | target | 判断指定的节点是否是叶子节点，target参数是一个节点DOM对象。 |
| find | id | 查找指定的节点并返回节点对象。 |
| select | target | 选择一个节点，target参数是一个节点DOM对象。 |
| check | target | 设置指定的节点为已选择状态。 |
| uncheck | target | 设置指定的节点为未选择状态。< |
| collapse | target | 关闭节点，target参数是一个节点DOM对象。 |
| expand | target | 打开节点，target参数是一个节点DOM对象。 |
| collapseAll | target | 关闭所有的节点。 |
| expandAll | target | 打开所有的节点。 |
| expandTo | target | 打开从根节点到指定节点之间的所有节点。 |
| append | param | 添加若干子节点到一个父节点，param参数有2个属性： parent：DOM对象，将要被添加子节点的父节点，如果未指定，子节点将被添加至根节点。 data：数组，节点数据。 |
| toggle | target | 打开或关闭节点的触发器，target参数是一个节点DOM对象。 |
| insert | param | 在一个指定节点之前或之后插入节点，'param'参数包含如下属性： before：DOM对象，在某个节点之前插入。 after: DOM对象，在某个节点之后插入。 data：对象，节点数据。 |
| remove | target | 移除一个节点和它的子节点，target参数是一个节点DOM对象。 |
| pop | target | 移除一个节点和它的子节点，该方法跟remove方法一样，不同的是它将返回被移除的节点数据。 |
| update | param | 更新指定的节点，param参数有如下属性： target(DOM对象，将被更新的目标节点)，id，text，iconCls，checked等。 |
| enableDnd | none | 启用拖动特性。 |
| disableDnd | none | 禁用拖动特性。 |
| beginEdit | nodeEl | 开始编辑节点。 |
| endEdit | nodeEl | 结束编辑节点。 |
| cancelEdit | nodeEl | 取消编辑节点。 |

## 树形表格

**树形表格**

继承$.fn.datagrid.defaults，使用$.fn.treegrid.defaults重载默认值。

**依赖关系**

* 数据表格

**使用方法**

1. **<table** id="tt"**></table>**
2. $('#tt').treegrid({
3. url:'treegrid\_data.json',
4. treeField:'name',
5. columns:[[
6. {title:'Task Name',field:'name',width:180},
7. {field:'persons',title:'Persons',width:60,align:'right'},
8. {field:'begin',title:'Begin Date',width:80},
9. {field:'end',title:'End Date',width:80}
10. ]]
11. });

**属性**

属性继承数据表格，以下是树形表格独有的树形。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **描述** | **默认值** |
| treeField（树形字段） | string（字符串） | 定义作为树节点的字段。 | null |
| animate（动画） | boolean（布尔型） | 定义当打开或关闭节点时是否显示动画效果。 | false |

**事件**

事件继承数据表格，以下是树形表格的独有事件。

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| onClickRow | row | 当用户点击节点时触发 。 |
| onDblClickRow | row | 当用户双击节点时触发 。 |
| onBeforeLoad | row, param | 在请求载入数据之前触发，返回false将取消载入。 |
| onLoadSuccess | row, data | 当数据载入成功时触发。 |
| onLoadError | arguments | 当数据载入失败时触发，arguments参数和jQuery.ajax的'error'函数一样。 |
| onBeforeExpand | row | 在节点打开之前触发 ，返回false将取消打开。 |
| onExpand | row | 当节点打开时触发。 |
| onBeforeCollapse | row | 在节点关闭之前触发，返回false将取消关闭。 |
| onCollapse | row | 当节点关闭时触发。 |
| onContextMenu | e, row | 当节点被鼠标右键点击时触发 。 |
| onBeforeEdit | row | 当用户开始编辑节点时触发。 |
| onAfterEdit | row,changes | 当用户结束编辑时触发。 |
| onCancelEdit | row | 当用户取消编辑节点时触发。 |

**方法**

|  |  |  |
| --- | --- | --- |
| **名称** | **参数** | **描述** |
| options | none | 返回树形表格的属性对象。 |
| resize | options | 设置树形表格的大小，options包含2个属性： width：新的树形表格宽度 height：新的树形表格高度。 |
| fixRowHeight | id | 固定特定行的高。 |
| loadData | data | 载入树形表格数据。 |
| reload | id | 重新载入树形表格数据。 |
| reloadFooter | footer | 重新载入树形表格的行底数据。 |
| getData | none | 获取已载入的数据。 |
| getFooterRows | none | 获取行底数据。 |
| getRoot | none | 获取根节点，返回节点对象。 |
| getRoots | none | 获取根节点，返回节点数组。 |
| getParent | id | 获取父节点。 |
| getChildren | id | 获取子节点。 |
| getSelected | none | 获取被选择节点并返回，如果没有节点被选择则返回null。 |
| getSelections | none | 获取所有被选择的节点。 |
| getLevel | id | 获取指定的节点的层级。 |
| find | id | 查找指定的节点并返回节点数据。 |
| select | id | 选择一个节点。 |
| unselect | id | 取消选择一个节点。 |
| selectAll | none | 选择所有的节点。 |
| unselectAll | none | 取消选择所有的节点。 |
| collapse | id | 关闭节点。 |
| expand | id | 打开节点。 |
| collapseAll | id | 关闭所有节点。 |
| expandAll | id | 打开所有节点。 |
| expandTo | id | 打开从根节点到指定节点直接的所有节点。 |
| toggle | id | 打开或关闭节点的触发器。 |
| append | param | 添加若干子节点到一个父节点，'param'参数包含如下属性： parent：DOM对象，将要被添加子节点的父节点，如果未指定，子节点将被添加至根节点。 data：数组，节点数据。 |
| remove | id | 移除一个节点和它的子节点。 |
| refresh | id | 属性指定的节点。 |
| beginEdit | id | 开始编辑节点。 |
| endEdit | id | 结束编辑节点。 |
| cancelEdit | id | 取消编辑节点。 |
| getEditors | id | 获取指定行的编辑器，每个编辑器都有如下属性： actions：编辑器可以做的行为。 target：目标编辑器jQuery对象。 field：字段名。 type：编辑类型。 |
| getEditor | options | 获取指定的编辑器，options包含2个属性： id：行节点 id。 field：字段名。 |