说明：EasyUI本是前端同学所需学习的，但是身为Java WEB开发人员，也必须要了解一些前端知识。我们可利用EasyUI迅速搭建起基本的界面，这对我这种无美术细胞的人是一大福音。学习EasyUI需要掌握的前端知识仅仅是HTML、CSS、JavaScript、JSON和jQuery。

# 1. EasyUI入门

使用EasyUI能快速构建WEB页面，EasyUI适合开发后台的前端页面。EasyUI是基于JQuery的。首先在官网上下载EasyUI的社区版本，社区版本是免费使用的。解压EasyUI包，其中的目录和文件解释如下：

（1）demo和demo-mobile文件夹：EasyUI的示例文件夹。由于EasyUI也适用于移动端，因此demo-mobile包下是移动端的使用示例。

（2）locale文件夹：区域和语言文件夹，其中存放EasyUI支持的语言文件。

（3）plugins文件夹：插件文件夹。

（4）src文件夹：源码文件夹。

（5）themes文件夹：EasyUI提供的主题文件夹。

（6）目录下的jquery.min.js文件是jQuery文件，jquery.easyui.min.js是EasyUI的JS文件，而带有mobile的是支持移动端的JS文件。

（7）其余的txt文件是更新说明、自述和许可证等文件。

使用EasyUI时，我们只需要在WEB项目中引入如下目录和文件：

locale文件夹（不需使用的语言文件可删除）、themes文件夹（其中的不需使用的主题文件夹可删除）和jquery.min.js、jquery.easyui.min.js文件。

## 1.1 第一个案例

下面写第一个EasyUI案例。代码和解释如下：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="en"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  *<!-- 引入一个主题的easyui样式文件。这里选择的就是default默认主题，可按需更改 -->* <**link rel="stylesheet" type="text/css" href="easyui/themes/default/easyui.css"**>  *<!-- 引入图标样式文件 -->* <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  *<!-- 这是一个可折叠面板的示例。解释如下： -->  <!--  class：指定该div样式为easyui的面板样式。使用easyui，通常即通过class指定样式，样式的名称都是"easyui-组件名"的形式  style中指定了该面板的大小，其中padding指定了面板边框和面板中内容的间距。  title用于指定面板的标题。  iconCls即设置面板的图标。各个图标的样式就是定义在themes/icon.css文件中的。可自行添加样式名和对应的图片。  collapsible可设置为true和false。true表示可折叠，false表示不可折叠。  -->* <**div  class="easyui-panel"  style="width**: 500**px**; **height**: 200**px**; **padding**: 10**px**;**"  title="示例 - 面板"  iconCls="icon-save"  collapsible="true"**>  面板中的内容...  </**div**> </**body**> *<!-- 引入jquery文件 -->* <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> *<!-- 再引入easyui的JS文件。必须在jquery后引入，因为是基于jquery的 -->* <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> <**script**>  *// 自己的js内容* </**script**> </**html**> |

特别注意：引入CSS文件时，link中的属性：rel、type和href都要写好，否则可能引入不成功。例如，若缺少rel="stylesheet"，则CSS效果无效。

我们注意到，这些easyui的属性其实不是在HTML标准里面的。HTML提供了data-属性，EasyUI现在也支持HTML5，所以EasyUI现在支持将EasyUI的属性全部写在data-options属性中。即上述的案例可写为：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/default/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**> *<!-- 其中的minimizable是显示最小化按钮；maximizable是显示最大化按钮 -->* <**div  class="easyui-panel"  style="width**: 500**px**; **height**: 200**px**; **padding**: 10**px**;**"  title="示例 - 面板"  data-options="iconCls: 'icon-save', collapsible: true, minimizable: true, maximizable: true"** >  面板中的内容... </**div**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> </**html**> |

可以看到，data-options属性值的内容也是“键值对”的形式。那么EasyUI的一个组件有哪些属性（或者方法、事件）呢？需要自行查看EasyUI的文档（以后用的文档就是这里提供的、CSDN一个人写的中文文档，推荐使用exe格式的）。文档中说明的属性都可写在该组件的data-options中。

文档下载地址：http://download.csdn.net/album/detail/343 。

## 1.2 EasyUI的原理

EasyUI是基于JQuery实现的，现在介绍一下EasyUI实现这些效果的原理。

学习jQuery时我们学习了jQuery的核心（选择器、DOM操作、事件和AJAX），同时jQuery具有插件机制，而EasyUI就是通过插件扩展来基于jQuery实现功能的。

我们这里写一个简单的jQuery插件就知道EasyUI大致是如何工作的了。

通过$.extend()方法可扩展jQuery的全局方法，而通过$.fn.extend()方法可扩展jQuery的局部方法。什么意思呢？例如我们在my-plugin.js中分别通过$.extend()和$.fn.extend()方法扩展了jQuery的方法：

|  |
| --- |
| **$**.extend({  myFuncGlobal: **function**() {  *alert*(**"这是我扩展的全局方法"**);  } });  **$**.**fn**.extend({  myFuncLocal: **function**() {  *alert*(**"这是我扩展的局部方法"**);  } }); |

测试页面中，这样使用：

|  |
| --- |
| <!doctype **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>标题</**title**>  *<!-- 先引入jquery -->* <**script src="jquery-3.1.0.min.js"**></**script**>  *<!-- 再引入我的“插件” -->* <**script src="my-plugin.js"**></**script**> </**head**> <**body**>  <**div id="testPlugin"**>用于测试的DIV</**div**> </**body**> <**script**>  *// 直接通过“$.myFuncGlobal()”调用全局方法。* **$**.myFuncGlobal();  *// 局部方法需要通过jquery对象来调用* **$**(**"#testPlugin"**).myFuncLocal(); </**script**> </**html**> |

从上述示例看出，扩展的全局方法直接通过“$.方法”即可调用，相当于Java中的“静态方法”，而扩展的局部方法要通过jQuery对象来进行调用，相当于Java的“实例方法”。

EasyUI中，其实就是通过扩展jQuery的方法来实现效果的，使得开发者只需要稍微了解一些HTML、CSS和JS的知识就能利用EasyUI展示出较好的页面效果。EasyUI内部只要提供一些常用的API，API内部利用jQuery操作DOM、数据和样式即可。

# 2 常用组件

## 2.1 Layout布局组件

使用Layout布局组件能快速构建一个分块的页面。标准的Layout布局面板中有四个区域面板，分别是“北 南 西 东 中”这几个区域。默认这几个区域是有边框、并且四边的区域是可折叠的。Layout简单使用如下：

|  |
| --- |
| <!doctype **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**link type="text/css" rel="stylesheet" href="easyui/themes/default/easyui.css"** />  <**link type="text/css" rel="stylesheet" href="easyui/themes/icon.css"** />  <**script src="easyui/jquery.min.js"**></**script**>  <**script src="easyui/jquery.easyui.min.js"**></**script**>  <**title**>EasyUI</**title**> </**head**> <**body**>  *<!-- 用于layout布局的div。可设置fit属性，这样布局组件将自适应父容器 -->  <!-- 有时，也可直接设置body的class属性为easyui-layout，然后设置fit为true，这样就是整个页面布局 -->* <**div class="easyui-layout" style="height**: 500**px**; **width**:800**px" data-options="fit: true"**>   *<!-- 区域面板 - 北边 -->  <!--  每个面板可设置的属性有：  region：可选'north' 'south' 'west' 'east'和'center'  title: 面板的标题  split: 是否可以通过分割栏改变面板大小  border: 是否显示边界  iconCls: icon图标的class名称。  collapsible: 是否可折叠  北、南面板可设置height样式；东、西面板可设置width样式，中心面板的大小可自适应。  -->* <**div data-options="region: 'north', title: '北边', split: true, border: true, iconCls: 'icon-save', collapsible: true" style="height**: 100**px"**></**div**>  *<!-- 区域面板 - 南边 -->* <**div data-options="region: 'south', title: '南边'" style="height**: 50**px"**></**div**>  *<!-- 区域面板 - 西边 -->* <**div data-options="region: 'west', title: '西边'" style="width**: 100**px**;**"**></**div**>  *<!-- 区域面板 - 东边 -->* <**div data-options="region: 'east', title: '东边'" style="width**: 100**px**;**"**></**div**>  *<!-- 区域面板 - 中间 -->* <**div data-options="region: 'center', title: '中间'"**></**div**>  </**div**> </**body**> </**html**> |

## 2.2 EasyUI调用组件方法

在实际应用中要经常调用组件的方法。比如想在特定时候展开或者收起北边的面板等。EasyUI调用组件方法的方式是（在EasyUI中文版“文档说明”一章中提及）：

|  |
| --- |
| $('selector').plugin('method', parameter); |

即拿到组件元素，使用组件名plugin、方法名method调用。

我们实现如下效果：页面一开始就收起北边面板，3秒后再展开。则这里的plugin名为layout，查看layout的方法中，有collapse和expand方法，并且方法参数是region中可选的值。那么实现的代码如下（主要看JS）：

|  |
| --- |
| <!doctype **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**link type="text/css" rel="stylesheet" href="easyui/themes/default/easyui.css"** />  <**link type="text/css" rel="stylesheet" href="easyui/themes/icon.css"** />  <**title**>EasyUI</**title**> </**head**> <**body**>  *<!-- 面板。给一个id以便JS使用 -->* <**div id="myLayout" class="easyui-layout" data-options="fit: true"**>  *<!-- 面板区域 -->* <**div data-options="region: 'north', title: '北边', split: true, border: true, iconCls: 'icon-save', collapsible: true" style="height**: 100**px"**></**div**>  <**div data-options="region: 'south', title: '南边'" style="height**: 50**px"**></**div**>  <**div data-options="region: 'west', title: '西边'" style="width**: 100**px**;**"**></**div**>  <**div data-options="region: 'east', title: '东边'" style="width**: 100**px**;**"**></**div**>  <**div data-options="region: 'center', title: '中间'"**></**div**>  </**div**> </**body**> <**script src="easyui/jquery.min.js"**></**script**> <**script src="easyui/jquery.easyui.min.js"**></**script**> <**script**>  **$**(**function**() {  *// 需要在页面加载完成之后执行这些代码。否则EasyUI的界面还没有构建出来就执行这些代码是会报错的。  // 使用EasyUI调用方法的方式：  // 一开始先收起北边面板* **$**(**"#myLayout"**).layout(**'collapse'**, **'north'**);  *// 三秒后展开* setTimeout(**function** () {  **$**(**"#myLayout"**).layout(**'expand'**, **'north'**);  }, 3000);  }); </**script**> </**html**> |

## 2.3 Layout的嵌套布局

有时我们需要的布局不一定就是EasyUI提供的规规矩矩的布局。比如我们不需要南边和东边面板，我们还希望在中心面板中再分出一个Layout，这个里面的Layout只显示西边和中心面板。

这时只要舍弃某个面板，或者进行Layout的嵌套即可。下面是实现的案例。

|  |
| --- |
| <!doctype **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**link type="text/css" rel="stylesheet" href="easyui/themes/default/easyui.css"** />  <**link type="text/css" rel="stylesheet" href="easyui/themes/icon.css"** />  <**title**>EasyUI</**title**> </**head**> <**body**>  *<!-- 外面的面板 -->* <**div class="easyui-layout" data-options="fit: true"**>  <**div data-options="region: 'north', title: '北边', split: true, border: false, iconCls: 'icon-save', collapsible: true" style="height**: 100**px"**>  北边充当头部  </**div**>  <**div data-options="region: 'west', title: '西边'" style="width**: 200**px**;**"**>  西边充当导航栏  </**div**>  <**div data-options="region: 'center', title: '中间'"**>  *<!-- 中间的再嵌套Layout，分成西边和中间两块 -->* <**div class="easyui-layout" data-options="fit: true"**>  *<!-- 嵌套的面板区域。只有西边和中间。面板不写title属性就没有标题栏。 -->* <**div data-options="region: 'west'" style="width**: 280**px"**>  嵌套的西边面板-充当作者、文章简介  </**div**>  <**div data-options="region: 'center'"**>  嵌套的中间面板-显示文章内容  </**div**>  </**div**>  </**div**>  </**div**> </**body**> <**script src="easyui/jquery.min.js"**></**script**> <**script src="easyui/jquery.easyui.min.js"**></**script**> </**html**> |

即：不需要的面板直接不写，需要细分区域的再嵌套。

# 3. Accordion（分类器）

Accordion可分类显示条目，有伸缩功能。Accordion也由容器和其中的面板两部分组成，其各自的属性方法等参见各自的文档页面。需要注意的是，当查看面板属性时，文档说“分类面板属性继承自panel面板”，即我们也可使用panel中定义的属性，另外分类面板还有自己的两个属性。

下面是Accordion一个简单的例子：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/default/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  *<!-- 分类器容器。可使用style指定宽高 -->  <!-- 基本属性:  fit：是否适应父容器大小  border: 是否显示边框  animate：伸缩时是否显示动画效果，默认为true  multiple：是否允许展示多个分类内容。默认为false，即一次只会展示一个分类  selected：初始时被选择的（即展示的）分类。默认为0，即选择第一个展开，这里写-1就是没有分类被展开  -->* <**div class="easyui-accordion" data-options="fit:false, border: false, animate: false, multiple: false, selected: -1" style="width**: 300**px**; **height**: 200**px"**>  *<!-- 其中每个div就是一个分类 -->  <!-- 可设置图标。style中设置padding是设置分类内容和面板的距离 -->* <**div title="分类1" data-options="iconCls: 'icon-save'" style="padding**: 10**px"**>  分类1的内容  </**div**>  <**div title="分类2" data-options="iconCls: 'icon-add', collapsible:false" style="padding**: 10**px"**>  分类2的内容  </**div**>  <**div title="分类3" data-options="iconCls: 'icon-reload'" style="padding**: 10**px"**>  分类3的内容  </**div**>  </**div**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> </**html**> |

分类面板也可设置selected属性，表示是否展开。

下面我们使用JS来操作Accordion，实现动态增删和伸缩分类面板。

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/default/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  <**div id="myAccordion" class="easyui-accordion" data-options="selected: -1" style="width**: 300**px**; **height**: 200**px"**>  <**div title="分类1" data-options="iconCls: 'icon-save'" style="padding**: 10**px"**>  分类1的内容  </**div**>  <**div title="分类2" data-options="iconCls: 'icon-add'" style="padding**: 10**px"**>  分类2的内容  </**div**>  </**div**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> <**script**>  **$**(**function** () {  *// 先增加一个分类面板  // 使用add方法，文档中参数是option，就是json数据，描述分类的属性，和之前介绍的分类属性用法一致。* **$**(**"#myAccordion"**).accordion(**'add'**, {  **title**: **'分类3'**,  **content**: **'分类3内容'**,  **iconCls**: **'icon-reload'**,  **selected**: **false** *// 在默认情况下，新增的面板会变成当前面板（即被选中）。如果要添加一个非选中面板，不要忘记将'selected'属性设置为false。* });   *// 2秒后删除第一个面板* setTimeout(**function** () {  **$**(**"#myAccordion"**).accordion(**'remove'**, 0); *// 方法参数值可以使索引或者标题  // 发现删除后的第一个分类（即原来的第二个分类）就会被展开。  // 为了让他不展开，我们再收起现在的第一个分类* **$**(**"#myAccordion"**).accordion(**'unselect'**, 0);  *// 最后展开分类3面板* **$**(**"#myAccordion"**).accordion(**'select'**, **'分类3'**);  }, 3000);  }); </**script**> </**html**> |

# 4. LinkButton按钮

EasyUI中使用a标签实现按钮的效果（当然也可直接在按钮元素中使用相关的class样式达到相同的效果）。使用例子如下：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/default/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  <**a id="myBtn" class="easyui-linkbutton" data-options="iconCls: 'icon-search'"**>按钮</**a**>  <**a class="easyui-linkbutton" data-options="iconCls: 'icon-remove', disabled: true"**>禁用的按钮</**a**>  *<!-- toggle可单独使用，相当于实现checkbox效果。但是group需要和toggle一起使用，实现radio效果 -->* <**a class="easyui-linkbutton" data-options="iconCls: 'icon-add', group: 'sex', toggle: true"**>男</**a**>  <**a class="easyui-linkbutton" data-options="iconCls: 'icon-add', group: 'sex', toggle: true"**>女</**a**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> <**script**>  *// 由于linkbutton比较简单，我们可自行通过JS处理点击事件等。  // 但是这里还是用EasyUI提供的功能。EasyUI为linkbutton提供了一个onClick事件。  // EasyUI使用事件的方法：$('selector').组件名( { '事件名1': 事件处理函数1, '事件名2': 事件处理函数2... } )。即参数就是一个JSON。  // 例子：(弹出按钮文本)* **$**(**"#myBtn"**).linkbutton({  **'onClick'**: **function**() {  **var** text = **$**(**this**).text();  alert(text);  }  }); </**script**> </**html**> |

# 5. Tabs选项卡

tabs中可分为tabs容器和选项卡两部分。简单案例如下：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/default/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  *<!-- tabs容器 -->  <!-- selected为-1还是表示一开始不选择任何选项卡 -->* <**div class="easyui-tabs"  style="width**: 500**px**; **height**: 300**px**;**"  data-options="fit: false, selected: -1"**>  *<!-- 选项卡面板 -->* <**div title="选项卡1" style="padding**: 10**px**;**"**>  选项卡1内容  </**div**>  *<!-- 设置显示窗口关闭按钮、设置图标、设置选项卡显示的网页内容。  注意，不能显示别的网站的的内容，因为EasyUI是通过AJAX访问的，有跨域问题。  -->* <**div title="选项卡2" data-options="closable: true, iconCls: 'icon-save', href: 'demo.html'" style="padding**: 10**px"**></**div**>  *<!-- 也可直接设置content内容 -->* <**div title="选项卡3" data-options="content: '选项卡3的内容'"**></**div**>  </**div**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> </**html**> |

下面动态地创建选项卡。案例：通过点击按钮来创建相应的选项卡（选项卡中显示内容或者本网站其他页面，这在后台开发中很常见，把功能基本都放在选项卡所关联的页面中）。

注意的是，如果这个选项卡已经创建了，那就不再创建了，且Tabs中应该展示对应的选项卡。

代码如下：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/default/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  *<!-- 按钮 -->* <**a id="addDeptBtn" class="easyui-linkbutton" data-options="iconCls: 'icon-add'"**>添加部门</**a**>  <**a id="editDeptBtn" class="easyui-linkbutton" data-options="iconCls: 'icon-edit'"**>修改部门</**a**> <**br**> <**br**>  *<!-- Tabs -->* <**div id="myTabs" class="easyui-tabs"  style="width**: 500**px**; **height**: 300**px**;**"  data-options="fit: false, selected: 0"**>  *<!-- 选项卡面板 -->* <**div title="首页" style="padding**: 10**px**;**"**>  首页内容  </**div**>  </**div**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> <**script**>  **$**(**function**() {  *// 处理添加部门按钮的点击事件。* **$**(**"#addDeptBtn"**).linkbutton({  **'onClick'**: **function**() {  *// 在该事件处理函数中添加选项卡。在添加之前，要根据选项卡title检查该选项卡是否存在* **if** ( !**$**(**"#myTabs"**).tabs(**'exists'**, **'添加部门'**) ) {  *// 不存在则添加* **$**(**"#myTabs"**).tabs(**'add'**, {  **title**: **'添加部门'**,  **content**: **'添加部门...'**,  **closable**: **true**,  **iconCls**: **'icon-add'** });  } **else** {  *// 存在的话，就选中按钮对应的选项卡。* **$**(**"#myTabs"**).tabs(**'select'**, **'添加部门'**);  }   }  });  *// 处理修改部门按钮点击事件。代码和上述类似。* **$**(**"#editDeptBtn"**).linkbutton({  onClick: **function**() {  **if** ( !**$**(**"#myTabs"**).tabs(**'exists'**, **'修改部门'**) ) {  **$**(**"#myTabs"**).tabs(**'add'**, {  **title**: **'修改部门'**,  **href**: **'demo.html'**, *// 这里不写文本内容，而直接显示一个页面的内容。常用。* **closable**: **true**,  **iconCls**: **'icon-edit'** });  } **else** {  **$**(**"#myTabs"**).tabs(**'select'**, **'修改部门'**);  }  }  });  }); </**script**> </**html**> |

# 6. Pagination分页

EasyUI中分页是一个独立的组件。

例子一：以静态方式创建Pagination：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/bootstrap/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  *<!-- 以静态方式创建pagination -->* <**div class="easyui-pagination" data-options="total: 1000, pageSize: 10"**></**div**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> *<!-- 在最后引入中文包。否则显示的效果是英文的 -->* <**script type="text/javascript" src="easyui/locale/easyui-lang-zh\_CN.js"**></**script**> </**html**> |

下面是动态方式创建Pagination，并在改变分页数量和选择不同页面时触发事件，以便拿到pagesize等数据传给后台处理。

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/bootstrap/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  *<!-- 以静态方式创建pagination -->* <**div id="myPagination" style="width**: 600**px"**></**div**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> *<!-- 在最后引入中文包。否则显示的效果是英文的 -->* <**script type="text/javascript" src="easyui/locale/easyui-lang-zh\_CN.js"**></**script**> <**script**>  **$**(**function**() {  *// 动态构建pagination* **$**(**"#myPagination"**).pagination({  **total**: 100,  **pageSize**: 10,  **pageNumber**: 1,  **pageList**: [10, 20, 40],  **layout**: [**'list'**, **'sep'**, **'first'**, **'prev'**, **'manual'**, **'next'**, **'last'**, **'links'** ],  **beforePageText**: **'这是第'** *// 这些属性都可查看文档进行设置。很灵活* });  *// 当分页页面改变时触发的事件。拿到pageNumber和pageSize* **$**(**"#myPagination"**).pagination({  *// 事件中可接受的参数就是pageNumber和pageSize* **'onSelectPage'**: **function** (pageNumber, pageSize) {  **console**.log(**"第"** + pageNumber + **"页"**);  **console**.log(**"页面大小："** + pageSize);  }  });  }); </**script**> </**html**> |

Pagination常和下面的DataGrid使用。

# 7. DataGrid

DataGrid可以使表格数据看起来美观。表格的class属性需要为“easyui-datagrid”，并且表格需要有thead、tbody元素。在thead元素中，每个表头“单元格”（th）还要设置data-options，data-options中也是设置键值对，这里需要设置field的值，field值是自己为每列定义的，相当于是“列名”。示例代码如下：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" href="easyui/themes/default/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  *<!--  首先表格要满足thead、tbody等格式，并且要设置class样式为easyui-datagrid。  表格的data-options中可设置：  1. fitColumns：表示各个列是否会等分表格宽度。该值要配合<th>中data-options设置的宽度使用，否则设置无效。  当设置为true时，会等分宽度（表格右侧还是会留出一小段空间预存放滑动条），当设置为false时，列的宽度由th的data-options设置宽度决定。  2. singleSelect表示表格是否为单选。设置为true时，表格只能选中一行数据。  th中data-options的field属性只是设置该列的“名字”，width用来设置列宽度（还要看看上述设置的fitColumn值是否为true）。  -->* <**table class="easyui-datagrid" data-options="fitColumns: true, singleSelect: true" style="width**: 350**px"**>  <**thead**>  <**tr**>  <**th data-options="field: 'id', width: 50"**>编号</**th**>  <**th data-options="field: 'name', width: 50"**>姓名</**th**>  <**th data-options="field: 'age', width: 50"**>年龄</**th**>  <**th data-options="field: 'gender', width: 50"**>性别</**th**>  </**tr**>  </**thead**>  <**tbody**>  <**tr**>  <**td**>1</**td**>  <**td**>张三</**td**>  <**td**>21</**td**>  <**td**>男</**td**>  </**tr**>  <**tr**>  <**td**>2</**td**>  <**td**>李四</**td**>  <**td**>22</**td**>  <**td**>女</**td**>  </**tr**>  </**tbody**>  </**table**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> </**html**> |

DataGrid也可在表格中加载json数据，只要设置data-options中的url属性值为json文件即可。另外，还可利用JS自己构建DataGrid，示例如下：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" href="easyui/themes/default/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  *<!-- 直接写一个table -->* <**table id="myGrid" style="width**: 600**px"**>  </**table**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> *<!-- 要在最后引入中文语言文件，否则显示的分页信息是英文的 -->* <**script type="text/javascript" src="easyui/locale/easyui-lang-zh\_CN.js"**></**script**> <**script**>  *// 获得元素，调用datagrid方法。* **$**(**"#myGrid"**).datagrid({  **url**: **'data.json'**, *// 获得表格数据的地址* **columns**: [[  {**field**: **'id'**, **title**: **'编号'**, **width**: 100},  {**field**: **'name'**, **title**: **'姓名'**, **width**: 100},  {**field**: **'age'**, **title**: **'年龄'**, **width**: 100},  {**field**: **'gender'**, **title**: **'性别'**, **width**: 100}  ]], *// 列数据，即表头。是“二维数组”，每个一维数组就是一行表头，但使用时只用一行表头* **fitColumns**: **true**, *// 是否等分列宽* **singleSelect**: **true**, *// 是否单选* **pagination**: **true**, *// 显示分页选项* **pageSize**: 2, *// 每页分页大小* **pageList**: [2, 4, 8, 10] *// 每页显示多少条的集合* }); </**script**> </**html**> |

构造的data.json数据如下：

|  |
| --- |
| [  {**"id"**:1, **"name"**:**"张三"**, **"age"**:22, **"gender"**:**"男"**},  {**"id"**:2, **"name"**:**"李四"**, **"age"**:21, **"gender"**:**"女"**},  {**"id"**:3, **"name"**:**"王五"**, **"age"**:23, **"gender"**:**"男"**},  {**"id"**:4, **"name"**:**"赵六"**, **"age"**:22, **"gender"**:**"女"**} ] |

此时能显示数据，但是每个分页都是显示上面的数据，要使分页有效，数据格式还要进行更改。

平常，根据业务需要，会利用JS对组件进行各种各样的操作，这时我们就要重点关注组件的属性、方法和事件了，使用API详细了解这三个方面非常重要，基本上就能掌握此组件提供的功能了。

# 8. 表单Form

## 8.1 ValidateBox验证框

进行验证的例子：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/bootstrap/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  *<!-- 静态创建的验证框  属性可有：required: 是否必须，validType：验证的类型。可参见文档。这里定义长度为6-18.  -->* <**input type="text" class="easyui-validatebox" data-options="required: true, validType: 'length[6, 18]'"**>  *<!-- 也可动态创建-->* 验证中文：<**input type="text" id="myInput"**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> <**script type="text/javascript" src="easyui/locale/easyui-lang-zh\_CN.js"**></**script**> <**script**>  **$**(**function**() {  *// 动态构建validatebox* **$**(**"#myInput"**).validatebox({  *// 这里没有写必须的属性。但是只要填写了表单，就会进行验证。  // 这里validType给了一个数组，即要验证数组中定义的所有规则。  // 第一个规则是长度2-5，这个验证规则EasyUI已经提供了。但是EasyUI并没有提供验证中文validChinese的方法。  // 因此我们下面进行扩展。* **validType**: [**'length[2, 5]'**, **'validChinese'**]  });  });   *// 扩展中文验证（扩展验证的方法在文档中有说明）。这个不需要写在加载完毕的函数中，扩展可单独写在一个文件中。* **$**.extend(**$**.fn.**validatebox**.defaults.**rules**, {  *// 写验证的规则名* **validChinese** : {  *// validator中写验证的方法，方法返回布尔值表示验证成功或失败* validator: **function** (value) {  *// 参数value就是文本框中输入的值  // 这里使用正则表达式验证。基本的汉字在Unicode中范围就是4E00-9FA5。* **if** ( /^[\u4E00-\u9FA5]\*$/.test(value) ) {  **return true**;  } **else** {  **return false**;  }  },  *// 验证失败的提示信息。* **message**: **"请输入中文"** }  }); </**script**> </**html**> |

此外，文档中还演示了如何验证两个文本框数据是否一致（密码和确认密码一致），可以看一看，是根据ID号关联的。

在实际应用中，通常是在表单中进行验证，并且在表单提交前进行验证，如果验证失败，是不允许提交的。那么validatebox也提供了isValid方法用于判断是否验证成功了。案例如下：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/bootstrap/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**> <**form id="myForm" action="https://www.google.com"**>  <**p**>验证邮箱：</**p**>  <**input id="vbEmail" type="text" name="email" class="easyui-validatebox" data-options="required: true, validType: 'email', validateOnBlur: true"**>  <**p**>密码长度为6-18位：</**p**>  <**input id="vbPwd" type="password" class="easyui-validatebox" data-options="validType: 'length[6, 18]'"**>  <**p**>  <**input type="submit" value="提交"**>  </**p**> </**form**>  </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> <**script type="text/javascript" src="easyui/locale/easyui-lang-zh\_CN.js"**></**script**> <**script**>  **$**(**function** () {  *// jQuery处理表单的提交事件。若处理函数返回true，则允许提交，返回false则禁止提交。* **$**(**"#myForm"**).**submit**(**function** () {  *// 验证邮箱和密码* **var** emailFlag = **$**(**"#vbEmail"**).validatebox(**'isValid'**);  **var** pwdFlag = **$**(**"#vbPwd"**).validatebox(**'isValid'**);  *// 均通过验证才可以提交* **if** (emailFlag && pwdFlag) {  **return true**;  } **else** {  **return false**;  }  });  }); </**script**> </**html**> |

## 8.2 ComboBox下拉列表框

ComboBox其实是文本框和下拉列表框的组合。EasyUI的Combobox可展示下拉列表，也允许用户手动输入，也会根据用户输入展示匹配的选项。例子：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/bootstrap/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  *<!-- 静态下拉列表 -->* <**select class="easyui-combobox" style="width**: 200**px"**>  <**option value="张三"**>张三</**option**>  <**option value="小明"**>小明</**option**>  <**option value="小红"**>小红</**option**>  <**option value="李四"**>李四</**option**>  </**select**>   *<!-- 动态创建 -->* <**input id="myCity" name="city"**>  </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> <**script type="text/javascript" src="easyui/locale/easyui-lang-zh\_CN.js"**></**script**> <**script**>  **$**(**function** () {  **$**(**"#myCity"**).combobox({  **url**: **'city.json'**, *// 获得数据的地址。* **valueField**: **'id'**, *// 数据中要绑定到下拉列表中值value的字段* **textField**: **'name'** *// 数据中要绑定到下拉列表中显示的字段* });  }); </**script**> </**html**> |

对应的data.json数据如下：

|  |
| --- |
| [  {**"id"**:1, **"name"**: **"A城市"**},  {**"id"**:2, **"name"**: **"B城市"**},  {**"id"**:3, **"name"**: **"C城市"**, **"selected"**: **true**},  {**"id"**:4, **"name"**: **"D城市"**} ] |

## 8.3 datebox日期输入框

datebox用于输入日期。例子如下：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/bootstrap/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  *<!-- datebox -->* <**input id="myDatebox" type="text" class="easyui-datebox" data-options="required: true"**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> <**script type="text/javascript" src="easyui/locale/easyui-lang-zh\_CN.js"**></**script**> <**script**>  **$**(**function** () {  *// 处理选择日期事件，显示选择的年月日。* **$**(**"#myDatebox"**).datebox({  **'onSelect'**: **function** (date) {  **console**.log(date.getFullYear());  **console**.log(date.getMonth() + 1);  **console**.log(date.getDate());  }  });  }); </**script**> </**html**> |

## 8.4 numberspinner数字微调

可对数字进行微调。比如用于调整商品数量等。例子：

|  |
| --- |
| 选择数量：<**input type="text" class="easyui-numberspinner" data-options="min: 0, max: 20, value: 1"**> |

## 8.5 slider滑动条

滑动条可显示数值、刻度等。例如：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/bootstrap/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  <**div style="padding**: 50**px**;**"**>  <**input id="mySlider" type="text"**>  </**div**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> <**script type="text/javascript" src="easyui/locale/easyui-lang-zh\_CN.js"**></**script**> <**script**>  **$**(**function** () {  **$**(**"#mySlider"**).slider({  **height**: 300,  **mode**: **'v'**, *// 'v'表示是垂直的* **min**: 150,  **max**: 190,  **showTip**: **true**,  **value**: 160, *// 默认值* **rule**: [150, **'|'**, 160, **'|'**, 170, **'|'**, 180, **'|'**, 190] *// 表示刻度* });  }); </**script**> </**html**> |

# 9 Window 窗口

## 9.1 window窗口

可在页面上创建窗口。自行测试静态窗口的创建。下面通过按钮来显示和关闭窗口。

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/bootstrap/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  <**a id="openWindowBtn" class="easyui-linkbutton"**>打开窗口</**a**>  <**a id="closeWindowBtn" class="easyui-linkbutton"**>关闭窗口</**a**>  <**div id="myWindow"**></**div**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> <**script type="text/javascript" src="easyui/locale/easyui-lang-zh\_CN.js"**></**script**> <**script**>  **$**(**function** () {  **$**(**"#openWindowBtn"**).linkbutton({  **'onClick'**: **function** () {  *// 创建窗口* **$**(**"#myWindow"**).window({  **title**: **'新窗口'**,  **width**: 600,  **height**: 350,  **left**: 100,  **top**: 100, *// 上述四个属性定位窗口大小和位置* **minimizable**: **false**,  **content**: **'窗口内容'**, *// 同样，也可用href指定窗口显示的页面  //modal: true // 是否为模态框。若是模态框，则页面其他内容暂时不可操作。* });  }  });  **$**(**"#closeWindowBtn"**).linkbutton({  **'onClick'**: **function** () {  *// 关闭窗口* **$**(**"#myWindow"**).window(**'close'**);  }  });  }); </**script**> </**html**> |

## 9.2 dialog对话框

见代码：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/bootstrap/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  <**a id="openDialogBtn" class="easyui-linkbutton"**>打开对话框</**a**>  <**div id="myDialog"**></**div**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> <**script type="text/javascript" src="easyui/locale/easyui-lang-zh\_CN.js"**></**script**> <**script**>  **$**(**function** () {  **$**(**"#openDialogBtn"**).linkbutton({  **'onClick'**: **function** () {  *// 创建对话框* **$**(**"#myDialog"**).dialog({  **title**: **'对话框'**,  **width**: 300,  **height**: 400,  **left**: 200,  **top**: 100,  **toolbar**: [  {  **text**: **'编辑'**,  **iconCls**: **'icon-edit'**,  handler: **function**() {alert(**'a'**);}  },  {  **text**: **'帮助'**,  handler: **function**() {alert(**'b'**);}  }  ],  **buttons**: [  {  **text**: **'提交'**,  handler: **function**() {alert(**"提交"**);}  }  ],  **content**: **'对话框内容也可用href代替。'** });  }  });  }); </**script**> </**html**> |

## 9.3 messager消息窗口

EasyUI一共提供了五种消息窗口，分别是：警告框alert、确认框confirm、显示框show、输入框prompt和进度框progress。分别调用上述方法来显示这些不同的框。

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/bootstrap/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  <**a id="alertBtn" class="easyui-linkbutton"**>警告框</**a**>  <**a id="confirmBtn" class="easyui-linkbutton"**>确认框</**a**>  <**a id="showBtn" class="easyui-linkbutton"**>显示框</**a**>  <**a id="promptBtn" class="easyui-linkbutton"**>输入框</**a**>  <**a id="progressBtn" class="easyui-linkbutton"**>进度框</**a**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> <**script type="text/javascript" src="easyui/locale/easyui-lang-zh\_CN.js"**></**script**> <**script**>  **$**(**function** () {  *// 1.警告框按钮* **$**(**"#alertBtn"**).linkbutton({  **'onClick'**: **function**() {  *// 弹出警告框* **$**.**messager**.alert(**'警告标题'**, **'警告内容'**, **'warning'**, **function** () {  alert(**"点击警告确定按钮的回调函数"**);  })  }  })  *// 2. 确认框按钮* **$**(**"#confirmBtn"**).linkbutton({  **'onClick'**: **function** () {  *// 弹出确认框* **$**.**messager**.confirm(**'确认标题'**, **'确认的内容'**, **function** (value) {  *// value是布尔值，表示选择的是“确认”还是“取消”* value ? alert(**"你点击了确认"**) : alert(**"你点击了取消"**);  })  }  });  *// 3. 显示框按钮* **$**(**"#showBtn"**).linkbutton({  **'onClick'**: **function** () {  *// 弹出显示框。显示框基本就用于消息提示。* **$**.**messager**.show({  **showType**: **'slide'**,  **showSpeed**: 500,  **width**: 300,  **height**: 200,  **title**: **'消息提示'**,  **msg**: **'请阅读消息'**,  **timeout**: 3000 *// 3秒后关闭* });  }  });  *// 4. 输入框按钮* **$**(**"#promptBtn"**).linkbutton({  **'onClick'**: **function** () {  *// 弹出输入框* **$**.**messager**.**prompt**(**'输入框标题'**, **'请输入内容'**, **function** (value) {  alert(**"你输入的内容是："** + value);  });  }  });  *// 5. 进度框按钮* **$**(**"#progressBtn"**).linkbutton({  **'onClick'**: **function** () {  *// 弹出进度框* **$**.**messager**.progress({  **title**: **'进度标题'**,  **msg**: **'进度内容'**,  **text**: **'文本'**,  **interval**: 300  });  }  });  }); </**script**> </**html**> |

# 10.tree树

EasyUI提供了树结构。例如静态展示：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/bootstrap/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  <**ul id="myTree" class="easyui-tree"**>  <**li**>  <**span**>第一章</**span**>  <**ul**>  <**li**>  <**span**>第一节</**span**>  </**li**>  <**li**>  <**span**>第二节</**span**>  </**li**>  </**ul**>  </**li**>  <**li**>  <**span**>第二章</**span**>  </**li**>  </**ul**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> <**script type="text/javascript" src="easyui/locale/easyui-lang-zh\_CN.js"**></**script**> <**script**>  **$**(**function** () {  *// 收起树所有的节点* **$**(**"#myTree"**).tree(**'collapseAll'**);  }); </**script**> </**html**> |

动态加载树时，EasyUI支持以children数据结构方式构建树。

动态加载树案例：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/bootstrap/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  <**ul id="myTree"**></**ul**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> <**script type="text/javascript" src="easyui/locale/easyui-lang-zh\_CN.js"**></**script**> <**script**>  **$**(**function** () {  *// 指定数据构建树* **$**(**"#myTree"**).tree({  **url**: **'tree.json'** });  }); </**script**> </**html**> |

使用children的数据例子是：

|  |
| --- |
| [  {  **"id"**: 1,  **"text"**: **"第一章"**,  **"state"**: **"closed"**,  **"iconCls"**: **"icon-save"**,  **"children"**: [  {  **"id"**: 11,  **"text"**: **"第一节"** },  {  **"id"**: 12,  **"text"**: **"第二节"** }  ]  },  {  **"id"**: 2,  **"text"**: **"第二章"** } ] |

上述注意，如果一个节点没有子节点，那么不能执行该节点的state属性值为closed，否则树结构会循环。

但是有时我们需要使用parentId的数据结构方式构建树。此时就要自定义了。代码如下，采用的是loadFilter方法来过滤显示，其中convert方法是我们自己写的。

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-CN"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>EasyUI</**title**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/bootstrap/easyui.css"**>  <**link rel="stylesheet" type="text/css" href="easyui/themes/icon.css"**> </**head**> <**body**>  <**ul id="myTree"**></**ul**> </**body**> <**script type="text/javascript" src="easyui/jquery.min.js"**></**script**> <**script type="text/javascript" src="easyui/jquery.easyui.min.js"**></**script**> <**script type="text/javascript" src="easyui/locale/easyui-lang-zh\_CN.js"**></**script**> <**script**>  **$**(**function** () {  *// 指定数据构建树* **$**(**"#myTree"**).tree({  **url**: **'tree.json'**,  **lines**: **true**, *// 显示树结构的线条* loadFilter: **function**(rows){  **console**.log(rows);  **return** *convert*(rows);  }  });  });   *// 转换的方法* **function** *convert*(rows){  **function** *exists*(rows, parentId){  **for**(**var** i=0; i<rows.length; i++){  **if** (rows[i].**id** == parentId) **return true**;  }  **return false**;  }   **var** nodes = [];  *// get the top level nodes* **for**(**var** i=0; i<rows.length; i++){  **var** row = rows[i];  **if** (!*exists*(rows, row.parentId)){  nodes.push(row);  }  }   **var** toDo = [];  **for**(**var** i=0; i<nodes.**length**; i++){  toDo.push(nodes[i]);  }  **while**(toDo.**length**){  **var** node = toDo.shift(); *// the parent node  // get the children nodes* **for**(**var** i=0; i<rows.length; i++){  **var** row = rows[i];  **if** (row.parentId == node.**id**){  **var** child = row;  **if** (node.**children**){  node.**children**.push(child);  } **else** {  node.**children** = [child];  }  toDo.push(child);  }  }  }  **return** nodes;  } </**script**> </**html**> |

对于JSON数据，只是形式发生了改变，其他的属性照常使用没问题！这是我自己根据别人代码修改的^\_^。例如：

|  |
| --- |
| [  {  **"id"**: 1,  **"parendId"**: 0,  **"text"**: **"第一章"**,  **"iconCls"**: **"icon-add"**,  **"state"**: **"closed"** },  {  **"id"**: 2,  **"parentId"**: 0,  **"text"**: **"第二章"** },  {  **"id"**: 11,  **"parentId"**: 1,  **"text"**: **"第一节"** },  {  **"id"**: 12,  **"parentId"**: 1,  **"text"**: **"第二节"** } ] |

相关的事件和方法用到的时候自己看文档。