**1.格式**

**jsTree需要一个具体格式JSON数据，在标准的语法没有那个字段是必须的-而是那些是你需要的。请记住你可以获取任何你请求的其他属性，jsTree将会不会碰他们，你将有可能在随后使用它们。**

**为了改变节点的图标你可以是用属性icon。具体的字符串需要包含/的一个图片的url路径，你可以使用任何其它字符串应用类样式去修饰<i>元素，它将会被用呈现这个图标。你可以使用boolean 值false来jsTree在渲染节点时没有图标。**

**你可以设置一个节点的状态使用state属性，它值可以使如下值得组合：opened, selected, disabled.**

**li\_attr和a\_attr可以直接通过jQuery属性函数获取。**

**当使用AJAX设置children为false，jsTree将会将渲染这个节点为关闭状态，如果需要打开的时候需要发送额外的请求。**

**如何内部children都应该遵循相同的格式，或者是普通字符串(这个字符串作为普通文本和任何其它自动生成的)**

**[java]** [view plain](http://blog.csdn.net/m0_37355951/article/details/75267407) [copy](http://blog.csdn.net/m0_37355951/article/details/75267407)

1. // Expected format of the node (there are no required fields)
2. {
3. id          : "string" // will be autogenerated if omitted
4. text        : "string" // node text
5. icon        : "string" // string for custom
6. state       : {
7. opened    : **boolean**  // is the node open
8. disabled  : **boolean**  // is the node disabled
9. selected  : **boolean**  // is the node selected
10. },
11. children    : []  // array of strings or objects
12. li\_attr     : {}  // attributes for the generated LI node
13. a\_attr      : {}  // attributes for the generated A node
14. }

**2.可选择JSON格式**

**如果你不想使用内部children的方式，你可以使用可选语法，每个节点需要包含两个必须字段：id和parent，没有children属性（其它都保持这个格式）**

**jsTree 将会自动构建这个层次关系，为表明一个节点应该是根节点可是设置parent属性为"#".**

**这个种方式大多数用于一次性渲染整棵树，这个数据存储在数据库之间有联结关系。**

**[java]** [view plain](http://blog.csdn.net/m0_37355951/article/details/75267407) [copy](http://blog.csdn.net/m0_37355951/article/details/75267407)

1. // Alternative format of the node (id & parent are required)
2. {
3. id          : "string" // required
4. parent      : "string" // required
5. text        : "string" // node text
6. icon        : "string" // string for custom
7. state       : {
8. opened    : **boolean**  // is the node open
9. disabled  : **boolean**  // is the node disabled
10. selected  : **boolean**  // is the node selected
11. },
12. li\_attr     : {}  // attributes for the generated LI node
13. a\_attr      : {}  // attributes for the generated A node
14. }

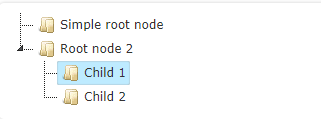
**3.使用JSON**

**为了使用JSON来渲染一棵树，你需要使用$.jstree.defaults.core.data配置选项**

**这个希望格式为一个数组节点。每个节点应该是一个如上所描述的对象或者是一个简单的字符串（这种情况字符串被用来作为一个节点的文本替换自动生成的文本），任何内部子节点格式是一样的。**

**[java]** [view plain](http://blog.csdn.net/m0_37355951/article/details/75267407) [copy](http://blog.csdn.net/m0_37355951/article/details/75267407)

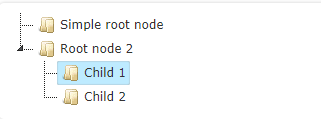
1. $('#using\_json').jstree({ 'core' : {
2. 'data' : [
3. 'Simple root node',
4. {
5. 'text' : 'Root node 2',
6. 'state' : {
7. 'opened' : **true**,
8. 'selected' : **true**
9. },
10. 'children' : [
11. { 'text' : 'Child 1' },
12. 'Child 2'
13. ]
14. }
15. ]
16. } });



**4.使用可选json格式**

**[java]** [view plain](http://blog.csdn.net/m0_37355951/article/details/75267407) [copy](http://blog.csdn.net/m0_37355951/article/details/75267407)

1. $('#using\_json\_2').jstree({ 'core' : {
2. 'data' : [
3. { "id" : "ajson1", "parent" : "#", "text" : "Simple root node" },
4. { "id" : "ajson2", "parent" : "#", "text" : "Root node 2" },
5. { "id" : "ajson3", "parent" : "ajson2", "text" : "Child 1" },
6. { "id" : "ajson4", "parent" : "ajson2", "text" : "Child 2" },
7. ]
8. } });



**5.使用AJAX**

**你可以使用AJAX向服务器请求返回一个json数据来渲染树，这个格式如上所示，这里唯一不同就是JSON是不可见，它是服务器返回的。**

**为了使用这个特性，你需要使用$.jstree.defaults.core.data 配置选项**

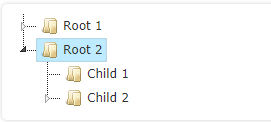
**仅仅是使用标准jQuery像AJAX配置和jstree将会自动做出一个AJAX请求而返回数据。**

**除了标准jQuery ajax选项，你可以提供data函数和url路径，这个功能将会运行当前的实例范围内，一个参数被通过表明这个节点被加载了，这个返回值将会用作各自的URL和data**

**如果你并不会返回json头部信息，至少设置数据类型 jQuery AJAX的选项为“json”**

**[java]** [view plain](http://blog.csdn.net/m0_37355951/article/details/75267407) [copy](http://blog.csdn.net/m0_37355951/article/details/75267407)

1. $('#tree').jstree({
2. 'core' : {
3. 'data' : {
4. 'url' : function (node) {
5. **return** node.id === '#' ?
6. 'ajax\_roots.json' :
7. 'ajax\_children.json';
8. },
9. 'data' : function (node) {
10. **return** { 'id' : node.id };
11. }
12. }
13. });



**6.使用函数**

**你可以提供一个函数，这个函数将会接受两个参数，节点加载和回调函数。**

**[java]** [view plain](http://blog.csdn.net/m0_37355951/article/details/75267407) [copy](http://blog.csdn.net/m0_37355951/article/details/75267407)

1. $('#tree').jstree({
2. 'core' : {
3. 'data' : function (obj, cb) {
4. cb.call(**this**,
5. ['Root 1', 'Root 2']);
6. }
7. }});

