1.最简单的树形结构的实现

①.首先，需要在页面中添加ul元素

<**ul id="permissionTree" class="ztree"**></**ul**>

②然后，在文档加载完毕之后，这样，写

$.*fn*.**zTree**.init($(**"#permissionTree"**), setting, zNodes);

其中，init方法中有三个参数，第一个参数就是选中那个ul即可

第二个参数时一个json数据，此处实现最简单的数所以直接写**var** setting={};

第三个参数是一个数组，格式大概如下

**var** zNodes =[  
 { **name**:**"父节点1 - 展开"**, **open**:**true**,  
 **children**: [  
 { **name**:**"父节点11 - 折叠"**,  
 **children**: [  
 { **name**:**"叶子节点111"**},  
 { **name**:**"叶子节点112"**},  
 { **name**:**"叶子节点113"**},  
 { **name**:**"叶子节点114"**}  
 ]},  
 { **name**:**"父节点12 - 折叠"**,  
 **children**: [  
 { **name**:**"叶子节点121"**},  
 { **name**:**"叶子节点122"**},  
 { **name**:**"叶子节点123"**},  
 { **name**:**"叶子节点124"**}  
 ]},  
 { **name**:**"父节点13 - 没有子节点"**, **isParent**:**true**}  
 ]},  
 { **name**:**"父节点2 - 折叠"**,  
 **children**: [  
 { **name**:**"父节点21 - 展开"**, **open**:**true**,  
 **children**: [  
 { **name**:**"叶子节点211"**},  
 { **name**:**"叶子节点212"**},  
 { **name**:**"叶子节点213"**},  
 { **name**:**"叶子节点214"**}  
 ]},  
 { **name**:**"父节点22 - 折叠"**,  
 **children**: [  
 { **name**:**"叶子节点221"**},  
 { **name**:**"叶子节点222"**},  
 { **name**:**"叶子节点223"**},  
 { **name**:**"叶子节点224"**}  
 ]},  
 { **name**:**"父节点23 - 折叠"**,  
 **children**: [  
 { **name**:**"叶子节点231"**},  
 { **name**:**"叶子节点232"**},  
 { **name**:**"叶子节点233"**},  
 { **name**:**"叶子节点234"**}  
 ]}  
 ]},  
 { **name**:**"父节点3 - 没有子节点"**, **isParent**:**true**}  
];

可以看出这个zNodes吧总体上是一个集合，每一个都写在{}里面,用逗号分隔

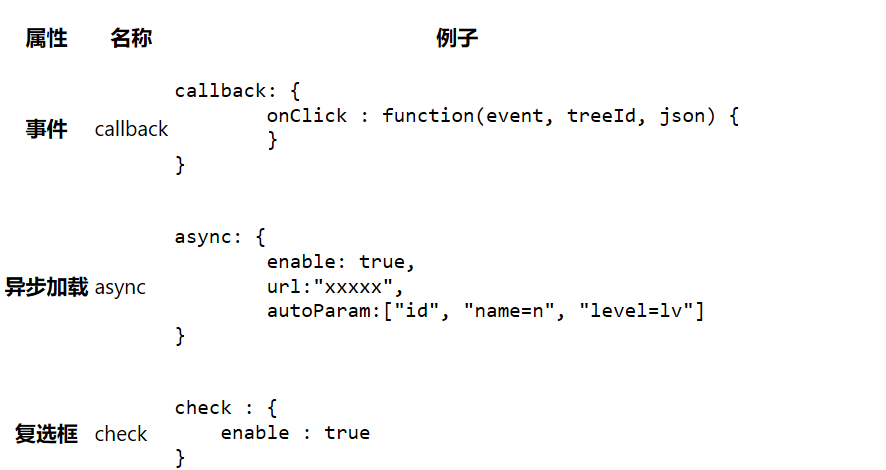
其中的name属性：是指这个节点的名称，即在页面中显示出来的那个名

open属性：是指它的直接点默认是打开的

children:里面还是一个集合，还是写在{}里面，里面还有

2.动态的从数据库中加载并形成的树形结构

上边那颗树是静态的，下面就是动态的了



①在setting里面放置一个一步加载的属性async

3.如果想在这个树后面每个节点上添加上对应的添加，删除，修改样式，即当鼠标移动过去的时候就可以看到对应的样式

**var** setting = {  
 **view**: {  
 *//是否开启多选* **selectedMulti**: **false**,  
 *//这是增加diy的，，，什么玩意，就是加图标的* addDiyDom: **function**(treeId, treeNode){  
 *//这是获取当前子节点* **var** icoObj = $(**"#"** + treeNode.**tId** + **"\_ico"**); *// tId = permissionTree\_1, $("#permissionTree\_1\_ico")  
 //这是判断如果图标存在，那么久移除默认样式，加上对应的图标* **if** ( treeNode.**icon** ) {  
 icoObj.removeClass(**"button ico\_docu ico\_open"**).addClass(treeNode.**icon**).css(**"background"**,**""**);  
 }  
  
 },  
 addHoverDom: **function**(treeId, treeNode){  
 *// <a><span></span></a>* **var** aObj = $(**"#"** + treeNode.**tId** + **"\_a"**); *// tId = permissionTree\_1, ==> $("#permissionTree\_1\_a")* aObj.attr(**"href"**, **"javascript:;"**);  
 **if** (treeNode.**editNameFlag** || $(**"#btnGroup"**+treeNode.**tId**).**length**>0) **return**;  
 **var** s = **'<span id="btnGroup'**+treeNode.**tId**+**'">'**;  
 **if** ( treeNode.**level** == 0 ) {  
 s += **'<a class="btn btn-info dropdown-toggle btn-xs" onclick="***addNode***('**+treeNode.**id**+**')" style="margin-left:10px;padding-top:0px;" href="#" >&nbsp;&nbsp;<i class="fa fa-fw fa-plus rbg "></i></a>'**;  
 } **else if** ( treeNode.**level** == 1 ) {  
 s += **'<a class="btn btn-info dropdown-toggle btn-xs" onclick="***updateNode***('**+treeNode.**id**+**')" style="margin-left:10px;padding-top:0px;" href="#" title="修改权限信息">&nbsp;&nbsp;<i class="fa fa-fw fa-edit rbg "></i></a>'**;  
 **if** (treeNode.children.length == 0) {  
 s += **'<a class="btn btn-info dropdown-toggle btn-xs " onclick="***deleteNode***('**+treeNode.**id**+**')" style="margin-left:10px;padding-top:0px;" href="#" >&nbsp;&nbsp;<i class="fa fa-fw fa-times rbg "></i></a>'**;  
 }  
 s += **'<a class="btn btn-info dropdown-toggle btn-xs" onclick="***addNode***('**+treeNode.**id**+**')" style="margin-left:10px;padding-top:0px;" href="#" >&nbsp;&nbsp;<i class="fa fa-fw fa-plus rbg "></i></a>'**;  
 } **else if** ( treeNode.**level** == 2 ) {  
 s += **'<a class="btn btn-info dropdown-toggle btn-xs" onclick="***updateNode***('**+treeNode.**id**+**')" style="margin-left:10px;padding-top:0px;" href="#" title="修改权限信息">&nbsp;&nbsp;<i class="fa fa-fw fa-edit rbg "></i></a>'**;  
 s += **'<a class="btn btn-info dropdown-toggle btn-xs" onclick="***deleteNode***('**+treeNode.**id**+**')" style="margin-left:10px;padding-top:0px;" href="#">&nbsp;&nbsp;<i class="fa fa-fw fa-times rbg "></i></a>'**;  
 }  
  
 s += **'</span>'**;  
 aObj.after(s);  
 },  
 removeHoverDom: **function**(treeId, treeNode){  
 $(**"#btnGroup"**+treeNode.**tId**).remove();  
 }  
 },  
 **async**: {  
 **enable**: **true**,  
 **url**:**"${**APP\_PATH**}/permission/loadData"**,  
 **autoParam**:[**"id"**, **"name=n"**, **"level=lv"**]  
 }  
};

注意：上边那个treeNode里面封装了所有json数据，都可以获取使用

4.执行完添加删除修改逻辑之后，重新加载整个树

**var** treeObj=$.*fn*.**zTree**.getZTreeObj(**"permissionTree"**);  
treeObj.reAsyncChildNodes(**null**,**"refresh"**);

5.如果想要加上复选框

**var** setting = {  
 **check**:{  
 **enable**:**true** },  
 **view**: {  
 *//是否开启多选* **selectedMulti**: **false**,  
 *//这是增加diy的，，，什么玩意，就是加图标的* addDiyDom: **function**(treeId, treeNode){  
 *//这是获取当前子节点* **var** icoObj = $(**"#"** + treeNode.**tId** + **"\_ico"**); *// tId = permissionTree\_1, $("#permissionTree\_1\_ico")  
 //这是判断如果图标存在，那么久移除默认样式，加上对应的图标* **if** ( treeNode.**icon** ) {  
 icoObj.removeClass(**"button ico\_docu ico\_open"**).addClass(treeNode.**icon**).css(**"background"**,**""**);  
 }  
 }  
 },  
 **async**: {  
 **enable**: **true**,  
 **url**:**"${**APP\_PATH**}/permission/loadData"**,  
 **autoParam**:[**"id"**, **"name=n"**, **"level=lv"**]  
 }  
};

5.获取选中的复选框的值

**var** treeObj=$.*fn*.**zTree**.getZTreeObj(**"permissionTree"**);  
**var** nodes=treeObj.getCheckedNodes(**true**);