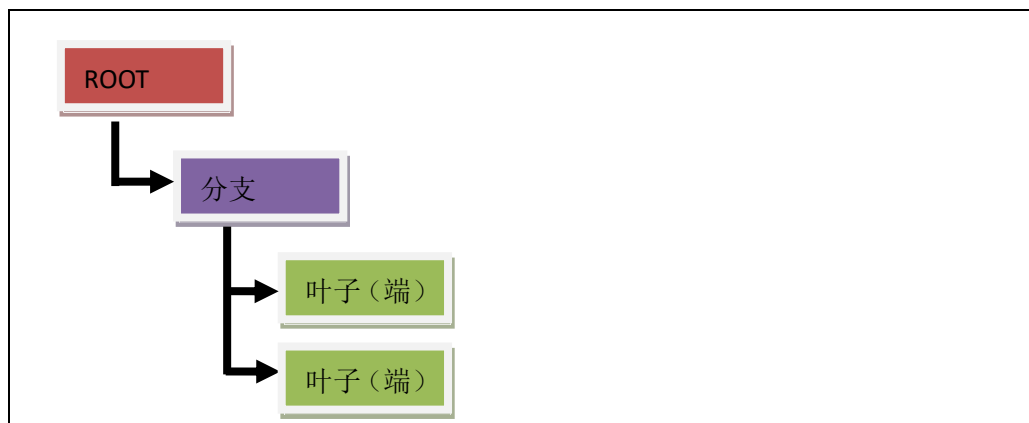


1. 菜单树

如果菜单的层次比较深，那么就会形成一种特殊的结构：树形结构（菜单树），树形结构存在不同的节点，根（顶），叉（分支），端（叶子），树形结构数据可以采用**递归方式**读取



2. 树形组件 – ztree

1) 基本操作

a) 环境

```
css: zTreeStyle.css  
js : jquery.js; jquery.ztree.all.min.js  
html : <ul id=" treeDemo" class=" ztree" >
```

b) 操作

```
var setting = {};  
var nodes = [{name:' xxx' }, {name:' yyyy' }];  
$.fn.zTree.init($( "#treeDemo" ), setting, nodes);
```

2) ztree 节点属性含义

```

var node = {
    name :    'xxx' ,// 节点名称
    open: true, // 父节点是否展开
    icon :    'xxx' ,// 图标图片的 url 可以是相对路径也可以是绝对路径
    checked:true,// 节点的 checkBox / radio 的 勾选状态
    children:[{} , {}] //节点的子节点数据集合
    level:1//记录节点的层级(不能由后台生成 , 是 ztree 组件自动生成的)
}

```

treeNode 对象的 ztree 的树形节点对象 , 包含了树形节点数据的所有属性字段和 ztree 自己生成的属性

3) ztree 树形组件配置

同步读取数据

```

var setting = {
    view: {
        selectedMulti: false,
    },
    callback: {
        onClick : function(event, treeld, json) {

        }
    }
}
Var d = [{}, {}...];
$.fn.zTree.init($("#treeDemo"), setting, d);

```

异步读取数据

```

var setting = {
    check: {
        enable: true // 复选框
    },
    view: {
        selectedMulti: false,
    },
    async: {
        enable: true,
        url:"xxxxx",
        autoParam:["id", "name=n", "level=lv"]
    },
    callback: {
        onClick : function(event, treeld, json) {

```

```

    }
  }
}
$.fn.zTree.init($("#treeDemo"), setting);
...
// 异步刷新数据
treeObj.reAsyncChildNodes(null, "refresh");

```

4) ztree 树形数据操作

读取当前树对象

```
var treeObj = $.fn.zTree.getZTreeObj("treeDemo");
```

刷新当前树对象的数据

```
treeObj.reAsyncChildNodes(null, "refresh");
```

查询节点

```
var nodes = treeObj.getSelectedNodes(); // 获取选择的节点
var node = treeObj.getNodeByParam("id", 1, null); // 根据参数获取节点
var nodes = treeObj.getCheckedNodes(true); // 获取被选中的节点

```

增加新节点

```
var newNodes = [{name:"newNode1"}, {name:"newNode2"}, {name:"newNode3"}];
newNodes = treeObj.addNodes(node, newNodes);

```

修改节点

```
var treeObj = $.fn.zTree.getZTreeObj("tree");
var nodes = treeObj.getNodes();
if (nodes.length>0) {
    nodes[0].name = "test";
    treeObj.updateNode(nodes[0]);
}

```

删除节点

```
treeObj.removeChildNodes(parentNode);
treeObj.removeNode(currentNode);

```