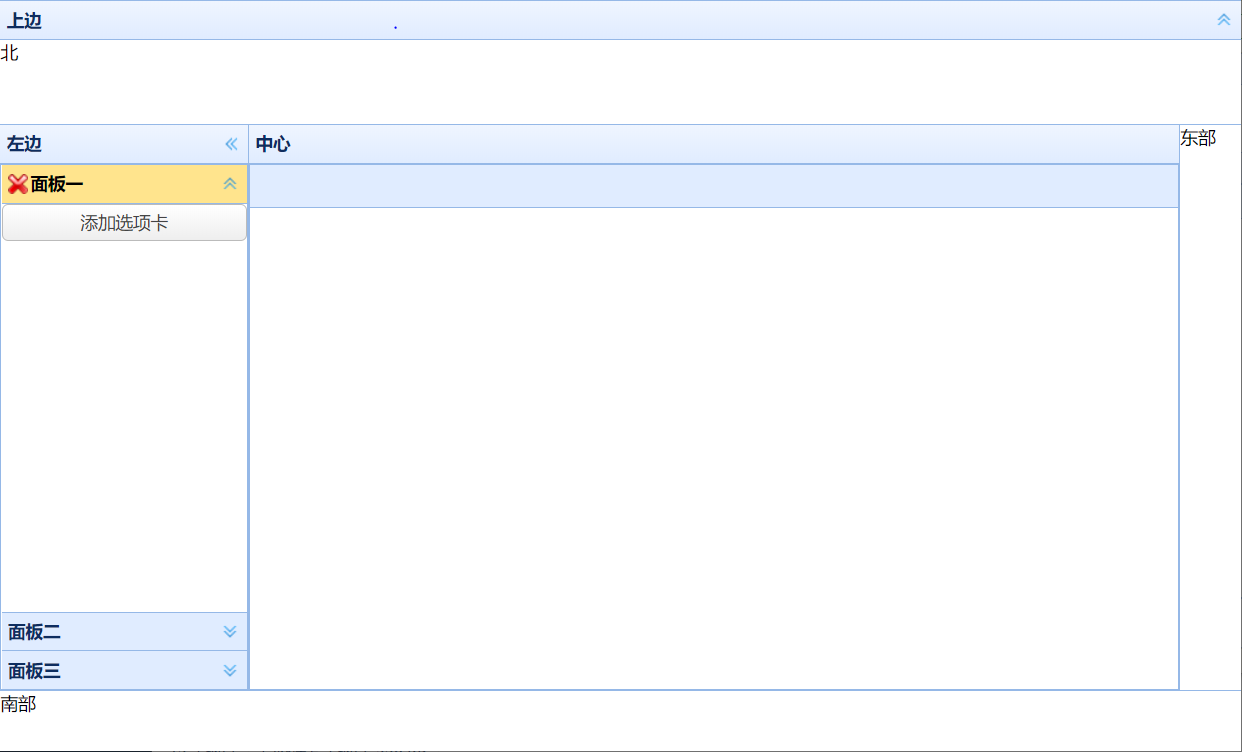
# EasyUI

## 最简单布局步骤

### 效果图



### 第一步 导包

注意：要先引入jquery的js,再引入easyui的js和css。有先后顺序的。

<!-- 引入JQuery -->

<script type="text/javascript" src="${pageContext.request.contextPath} /jquery-easyui/jquery.min.js"></script>

<!-- 引入EasyUI -->

<script type="text/javascript" src=" /jquery-easyui/jquery.easyui.min.js"></script>

<!-- 引入EasyUI的中文国际化js，让EasyUI支持中文 -->

<script type="text/javascript" src=" /jquery-easyui/locale/easyui-lang-zh\_CN.js"></script>

<!-- 引入EasyUI的样式文件-->

<link rel="stylesheet" href="/jquery-easyui/themes/default/easyui.css" type="text/css"/>

<!-- 引入EasyUI的图标样式文件-->

<link rel="stylesheet" href="/jquery-easyui/themes/icon.css" type="text/css"/>

### 第二步 布局

layout布局：north上、west左、east右、south下、center中心

在body标签 定义属性：class="easyui-layout"

在body标签体内加入div标签中有定义data-options属性

<!--使用div元素描述每个区域 region:定义区域的关键字，-->

<body class="easyui-layout">

<div title="上边" data-options="region:'north'" style="height:100px">北</div>

<div title="左边 " data-options="region:'west'" style="width:200px">西</div>

<div title="中心" data-options="region:'center'">中间</div>

<div data-options="region:'east'" style="width:200px">东部</div>

<div data-options="region:'south'" style="height:50px">南部</div>

注意：需要定义样式style进行分离：高度上下和宽度左右，中心不用管，加了title就会有标题样式

### 第三步 定义折叠面板

定义折叠面板：在zuo标签div定义：accordion，标签内定义属性class="easyui-accordion"

可以添加属性fit:true 填充整个父容器 也可以添加：title、添加图标等

<!-- 在左面布局 折叠面板操作 -->

<div class="easyui-accordion" data-options="fit:true">

<div data-options="iconCls:'icon-cancel'" title="面板一">

<a id="show" class="easyui-linkbutton">查看 </a>

</div>

<div title="面板二">222</div>

<div title="面板三">333</div>

</div>

### 第四步 定义选项卡面板

选项卡面板 在center(中间)的div中放入如下内容

<div title="显示内容" data-options="region:'center'">

<div id="mytabs" class="easyui-tabs" data-options="fit:true">

<div data-options="iconCls:'icon-cancel',closable:true" title="选项一">111</div>

<div data-options="iconCls:'icon-cancel',closable:true" title="选项二">222</div>

<div data-options="iconCls:'icon-cancel',closable:true" title="选项三">333</div>

</div>

</div>

动态点击跳出选项卡操作

使用到 tabs的两个方法exists：判断是否已经打开选项卡，select：显示指定的选项卡

<script type="text/javascript">

$("#show").click(function(){

//第一步：通过按钮触发事件:如果已经打开此标题选项卡返回true，否则false

var e=$("#mytabs").tabs("exists","选项卡标题"); //此标签必须是属性带：class="easyui-tabs"

if(e){

//第二步 如果是true直接使用”select”方法打开此标题的选项卡窗口

$("#mytabs").tabs("select","选项卡标题");

}else{

第三步 ：没有此标题的方法，创建一个此标题名称的选项卡

$("#mytabs").tabs("add",{

title:" 选项卡标题",

closable:true, //标题中添加关闭按钮

content:”111”

})

}

})

</script>

## 选项卡

### HTMl静态调出

在layout的中间进行显示

1、创建一个div；

2、使用属性class="easyui-tabs"；

3、填充整个父窗口(即中间显示父窗口)：fit:true

<div data-options="region:'center'">

<div id="tt" class="easyui-tabs" data-options="fit:true">

<!-- 使用子div表示每个面板 -->

<div data-options="iconCls:'icon-cut'" title="面板一">1111</div>

<div data-options="closable:true" title="面板二">2222</div>

<div title="面板三">3333</div>

</div>

</div>

### JS动态调出

使用到方法有：

tabs("exists", "标题名");//有此标题名：返回true

tabs("select", "选项卡标题"); //指定标题名进行选中

tabs("add",{这里是json格式}; //添加选项卡，大括号有很多属性

使用iframe标签来显示选项卡中的内容

<iframe frameborder="0" width="100%" height="100%" src="index.html"></iframe>

iframe：老师说页面上挖出那么一个坑：可以单独显示独立页面

参数：frameborder="0"：边框为0，src：uri地址(1可以百度url，2相对资源文件)

第一步 html配置

1、在需要显示的位置创建一个div：如在布局的中间显示

<div data-options="region:'center'" style="padding:5px;background:#eee;">

<!—属性不能少：class="easyui-tabs"；fit:true是填充整个父容体-->

<div id="tt" class="easyui-tabs" data-options="fit:true"></div>

</div>

第二步 js代码

<script>

//第一步： 通过按钮触发事件

$("# tt ").click(function() {

//第二步 使用exists方法判断是否已经打开：标题为a的选项卡

var e = $("# tt ").tabs("exists", "a");

if (e) {

//第三步 如果有则使用：使用select方法选中a选项卡

$("# tt ").tabs("select", "a");

} else {

$("# tt ").tabs("add", {

//第四步：没有标题为a的 使用add增加此标题a的选项卡

title: "a",

//标题中添加关闭按钮

closable: true,

//选项卡内容:这里引用iframe

content: '<iframe frameborder="0" width="100%" height="100%" src="index.html"></iframe>'

})

}

})

</script>

## 菜单

效果图



使用步骤

1、使用a标签(菜单div标签也可以的)作为菜单栏，div为下拉菜单选项

2、菜单a标签：有属性class="easyui-menubutton"、menu的值用于关联另一个div的，

3、菜单选项div：嵌套div每一个子div代表一个选项

<!-- 制作菜单 -->

<a data-options="iconCls:'icon-help',menu:'#mm'" class="easyui-menubutton">控制面板</a>

<!-- 使用div元素制作下拉菜单 -->

<div id="mm"> 事件就用普通的jquery或原生

<div onclick="alert(1111)" data-options="iconCls:'icon-edit'">修改密码</div>

<div>联系管理员</div>

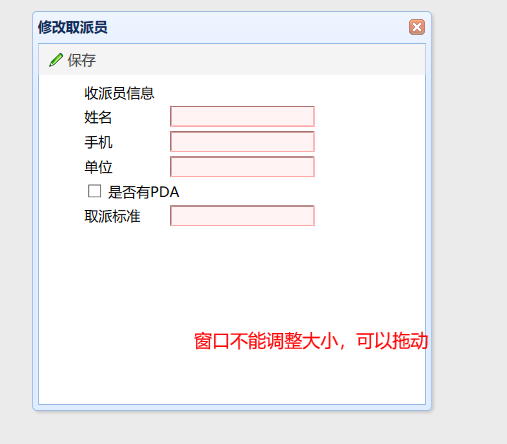
<div class="menu-sep"></div> <!-- 分割线效果-->

<div>退出系统</div>

</div>

## Windows(窗口)

效果图



第一步 HTML内容

<!-- 修改取派员窗口 -->

<div class="easyui-window" title="对收派员进行添加或者修改" id="editStaffWindow" collapsible="false"

minimizable="false" maximizable="false" style="top:20px;left:200px">

<div region="north" style="height:31px;overflow:hidden;" split="false" border="false" >

<div class="datagrid-toolbar">

<a id="edit" icon="icon-edit" href="#" class="easyui-linkbutton" plain="true" >保存</a>

</div>

</div>

<div region="center" style="overflow:auto;padding:5px;" border="false">

<form id="editStaffForm" action="staffAction\_edit.action" method="post">

<input type="hidden" name="id">

<table class="table-edit" width="80%" align="center">

<tr class="title">

<td colspan="2">收派员信息</td>

</tr>

<tr>

<td>姓名</td>

<td><input type="text" name="name" class="easyui-validatebox" required="true"/></td>

</tr>

<tr>

<td>手机</td>

<td>

<script type="text/javascript">

$(function(){

//为保存按钮绑定事件

$("#edit").click(function(){

//表单校验，如果通过，提交表单

var v = $("#editStaffForm").form("validate");

if(v){

//$("#addStaffForm").form("submit");

$("#editStaffForm").submit();

}

});

var reg = /^1[3|4|5|7|8][0-9]{9}$/;

//扩展手机号校验规则

$.extend($.fn.validatebox.defaults.rules, {

telephone: {

validator: function(value,param){

return reg.test(value);

},

message: '手机号输入有误！'

}

});

});

</script>

<input type="text" data-options="validType:'telephone'"

name="telephone" class="easyui-validatebox" required="true"/></td>

</tr>

<tr>

<td>单位</td>

<td><input type="text" name="station" class="easyui-validatebox" required="true"/></td>

</tr>

<tr>

<td colspan="2">

<input type="checkbox" name="haspda" value="1" />

是否有PDA</td>

</tr>

<tr>

<td>取派标准</td>

<td>

<input type="text" name="standard" class="easyui-validatebox" required="true"/>

</td>

</tr>

</table>

</form>

</div>

</div>

第二步 JS内容

$(function(){

// 先将body隐藏，再显示，不会出现页面刷新效果

$("body").css({visibility:"visible"});

// 修改取派员窗口

$('#editStaffWindow').window({

title: '修改取派员',

width: 400,

modal: true,//遮罩效果

shadow: true,//阴影效果

closed: true,//关闭

height: 400,

resizable:false //不可以更改窗口大小

});

});

//数据表格绑定的双击行事件对应的函数

function doDblClickRow(rowIndex, rowData){

//打开修改取派员窗口

$('#editStaffWindow').window("open");

//使用form表单对象的load方法回显数据

$("#editStaffForm").form("load",rowData);

}

## 数据表格

### 方式一 纯HTML

在html发送ajax请求获取json数据创建datagrid

1、制定一个table标签：在标签中使用data-options属性：url：请求地址；class="easyui-datagrid"

<table data-options="url:',method:'get'" class="easyui-datagrid">

<thead>

<tr>

<!—注意：field中的值必须和json对象key一一对应 -->

<th data-options="field:'id'">编号</th>

<th data-options="field:'name'">姓名</th>

<th data-options="field:'age'">年龄</th>

</tr>

</thead>

</table>

2、json数据：它是一个json对象数组

[

{"id":"1","name":"温文亮","address":"每周"}, #每个key和field中一一对应才会显示值

{"id":"2","name":"地方","address":"每周"},

{"id":"3","name":"温文亮","address":"每周"},

]

效果



### 方式二 js方式

#### 最简单方式

最简单方式：$("#mytable").datagrid(json对象)

第一步 HTML中的定义：只要一个table标签即可

<table id="mytable"></table>

第二步 js方式操作：最主要的两个属性：url和columns的二维数组格式

<script>

$(function(){

$("#mytable").datagrid({

columns:[[

{field:'id',title:"编码"},

{field:'name',title:"姓名"},

{field:'address',title:"地址"}

]],

method:'get',

url:'json/table.json'

})

})

</script>

第三步：json数据格式：json对象的数组:[{}]

[

{"id":"1","name":"温文亮","address":"每周"},

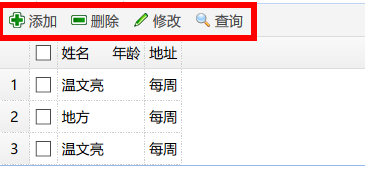
{"id":"2","name":"地方","address":"每周"},

{"id":"3","name":"温文亮","address":"每周"},

]

#### 带工具栏

带工具栏的数据表格



第一步 一样，要json数据不变 第二步如下：

<script>

$(function(){

//页面加载完成后，创建数据表格datagrid

$("#mytable").datagrid({

//定义标题行所有的列

columns:[[

{title:'编号',field:'id',checkbox:true},

{title:'姓名',field:'name'},

{title:'年龄',field:'age'},

{title:'地址',field:'address'}

]],

//指定数据表格发送ajax请求的地址

url:'json/table.json',

rownumbers:true, //每行显示顺序编号

singleSelect:true,//若为true，只允许选中一行

method:'get',

//定义工具栏

toolbar:[

{text:'添加',iconCls:'icon-add',

//为按钮绑定单击事件

handler:function(){

alert('add...');

}

},

{text:'删除',iconCls:'icon-remove'},

{text:'修改',iconCls:'icon-edit'},

{text:'查询',iconCls:'icon-search'}

]

});

});

</script>

小知识：注意 工具栏内容较多是可以抽取出来的

toolbar:aaa,

注意：函数不要放在加载完网页后再执行，即：$(function(){不要函数放这里})

var aaa=[

{text:'添加',iconCls:'icon-add',

//为按钮绑定单击事件

handler:function(){

alert('add...');

}

},

{text:'删除',iconCls:'icon-remove'},

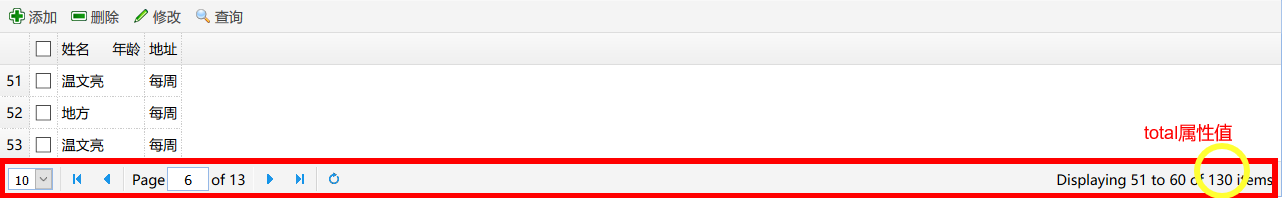
{text:'修改',iconCls:'icon-edit'},

{text:'查询',iconCls:'icon-search'}

];

#### 带分页

带分页的数据表格



Json数据格式有变化：一个json对象两个key：total值为总数、rows值为json对象数组，如下：

{

"total":"130",

"rows":[{"id":"1","name":"温文亮","address":"每周"},

{"id":"2","name":"地方","address":"每周"},

{"id":"3","name":"温文亮","address":"每周"}]

}

Js的操作

<script type="text/javascript">

$(function(){

//页面加载完成后，创建数据表格datagrid

$("#mytable").datagrid({

//定义标题行所有的列

columns:[[

//一般id都不显示可以设置为复选框，有办法可以获取值

{title:'编号',field:'id',checkbox:true},

{title:'姓名',field:'name'},

{title:'年龄',field:'age'},

{title:'地址',field:'address'}

]],

//指定数据表格发送ajax请求的地址

url:'${pageContext.request.contextPath }/json/datagrid\_data.json',

rownumbers:true,

singleSelect:true,

//显示分页条

pagination:true,

pageList:[3,5,7,10]

});

});

</script>

### 方法的使用

使用方式：$("#grid").datagrid("这里传入定义的方法名");（方法名看说明文档）

//获取数据表格中所有选中的行，返回数组对象

var rows = $("#grid").datagrid("getSelections");//定义的方法名,获得数组

if(rows.length == 0){

//没有选中记录，弹出提示

$.messager.alert("提示信息","请选择需要删除的取派员！","warning");

}else{}

# Ztree的使用

## 第一步 导包

Js引入要注意先后顺序，jquery必须第一位

<!-- 引入JQuery -->

<script type="text/javascript" src="js/jquery.min.js"></script>

<link rel="stylesheet" href="css/zTreeStyle/zTreeStyle.css" type="text/css">

<script type="text/javascript" src="js/jquery.ztree.all.js"></script>

## 第二步 使用

例子

1. 简单的使用
2. 使用ajax请求形成树

1、简单的使用

第一步 定义页面那个位置使用Ztree

//两个必须：1、要使用ul标签，2、属性必须有class="ztree"

<ul id="ztree1" class="ztree"></ul>

第二步 js部分

构造tree(树)初始化的方法：$.fn.zTree.init($("#ztree1"),setting,zNodes);

参数一：ul的id；

参数二：核心配置：setting有很多可以设置的：默认简单的是关闭的需要开启

参数三：数据的内容它是一个数组开始以[]包裹：每个json对象（也就是大括号包裹）都有3个属性：id、pId(注意i是大写，它对于父节点的id，没有就是根)、name(标题名称)

$(function(){

//第一步核心配置：var setting={}(直接空就是默认的ztree属性)，需要开启简单json数据

var setting={

data:{

simpleData:{

enable:true //使用简单的json数据构造ztree节点

}

}

};

//第二步 设置参数的数组：构造节点数据,这里设置三个json对象

var zNodes=[

//每个json对象表示一个节点：id代表自己的id，pId：代表父类的节点(没匹配节点自己就是根)

{"id":"1","pId":"0","name":"节点一"},

{"id":"2","pId":"1","name":"节点二"},

{"id":"3","pId":"2","name":"节点三"}

];

//第三步 初始化tree：第一个参数：要使用的对象，setting设置的属性，节点数据

$.fn.zTree.init($("#ztree1"),setting,zNodes);

})

1. 使用ajax请求形成树

<!-- 使用ajax获取数据 形成tree -->

<ul id="ztree2" class="ztree">

</ul>

<script>

$(function(){

var setting2={data:{simpleData:{enable:true}}};

<!-- 使用ajax获取json对象数组 -->

var url="json/dfg.json"

$.get(url,{},function(data){

$.fn.zTree.init($("#ztree2"), setting2, data);

},"json");

})

</script>

Json格式：一个json对象(大括号包裹的就是一个对象)的数组

#数组和Java格式一样：中括号[]

[

{"id":"1","pId":"0","name":"根标题名称","user":"这是多余的"},

{"id":"2","pId":"1","name":"我是根下的节点","user":"这是多余的"},

{"id":"3","pId":"0","name":"我是根2","user":"这是多余的"},

{"id":"4","pId":"3","name":"我是根2的子节点","user":"这是多余的"}

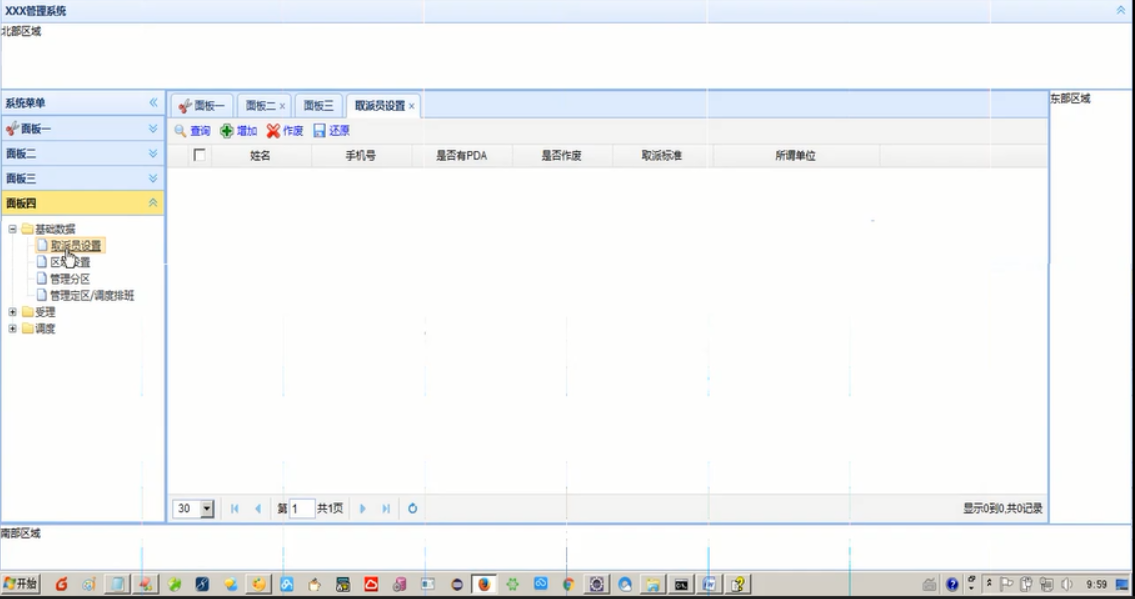
]

# easyUI与ztree配合使用

动态在tree中打开选项卡并显示内容

效果

点击叶子节点显示erayUI的选项卡并且弹出相关的操作页面



第一步 请求的 json数据格式

1、ztree数据都是json对象数组：json对象都有3个属性：id(自己id)、pId(父类id)、name(节点标题)

2、叶子节点：在必须有的3个属性再添加一个page：用于跳转的页面url

3、配合html<iframe>标签来为easyUI显示页面内容

[

{"id":"1","pId":"0","name":"根标题名称"},

{"id":"2","pId":"1","name":"我是根下的叶子节点","page":"这是url请求地址"},

{"id":"3","pId":"0","name":"我是根2"},

{"id":"4","pId":"3","name":"我是根2的叶子节点","page":"这是url请求地址"}

]

第二步 使用步骤

$(function(){

var setting={

data:{

simpleData:{

enable:true //使用简单的json数据构造ztree节点

}

},

//第一步 ztree核心配置中使用回调事件

callback:{

//为ztree节点绑定单击事件

//参数treeId：标签<ul>的id名称

//参数treeNode：点击的每个节点数据(即每个json对象的数据)

onClick:function(event,treeId,treeNode){

//判断选项卡是否已经存在

var e=$("#mytabs").tabs("exists",treeNode.name);

if(e){

$("#mytabs").tabs("select",treeNode.name);

}else{

//判断treeNode中有没有page：因为设计的json数据只有叶子节点才有此数据

if(treeNode.page != undefined){

$("#mytabs").tabs("add",{

title:treeNode.name,

closable:true,

//使用iframe来显示选项卡中的页面内容

content:'<iframe frameborder="0" height="100%" width="100%"

src="+ treeNode.page+ "></iframe>”' //跳转页面

});

}

}

}

}

};

//第二步 构造节点数据 动态使用json数据：ajax调用操作

$.get("${pageContext.request.contextPath}/js/menu.json",{},function(data){

//调用ztree的初始化方法

$.fn.zTree.init($("#ztree1"),setting,data);

},"json")

})