zTree 是利用 JQuery 的核心代码，实现一套能完成大部分常用功能的 Tree 插件

* 兼容 IE、FireFox、Chrome 等浏览器
* 在一个页面内可同时生成多个 Tree 实例
* 支持 JSON 数据
* 支持一次性静态生成 和 Ajax 异步加载 两种方式
* 支持多种事件响应及反馈
* 支持 Tree 的节点移动、编辑、删除
* 支持任意更换皮肤 / 个性化图标（依靠css）
* 支持极其灵活的 checkbox 或 radio 选择功能
* 简单的参数配置实现 灵活多变的功能

# zTree 方法详解

$.fn.zTree : {

init (obj, zSetting, zNodes)

getZTreeObj (treeId)

destroy (treeId)

\_z : {tools, view, event, data}

}

zTreeObj : {

setting

addNodes (parentNode, newNodes, isSilent)

cancelEditName (newName)

cancelSelectedNode (node)

checkAllNodes (checked)

checkNode (node, checked, checkTypeFlag, callbackFlag)

copyNode (targetNode, node, moveType, isSilent)

destroy ()

editName (node)

expandAll (expandFlag)

expandNode (node, expandFlag, sonSign, focus, callbackFlag)

getChangeCheckedNodes ()

getCheckedNodes (checked)

getNodeByParam (key, value, parentNode)

getNodeByTId (tId)

getNodeIndex (node)

getNodes ()

getNodesByFilter (filter, isSingle, parentNode, invokeParam)

getNodesByParam (key, value, parentNode)

getNodesByParamFuzzy (key, value, parentNode)

getSelectedNodes ()

hideNode (node)

hideNodes (nodes)

moveNode (targetNode, node, moveType, isSilent)

reAsyncChildNodes (parentNode, reloadType, isSilent)

refresh ()

removeChildNodes (parentNode)

removeNode (node, callbackFlag)

selectNode (node, addFlag)

setChkDisabled (node, disabled)

setEditable (editable)

showNode (node)

showNodes (nodes)

transformToArray (nodes)

transformTozTreeNodes (simpleNodes)

updateNode (node, checkTypeFlag)

}

## zTree

### $.fn.zTree.init

概述[ 依赖 jquery.ztree.core 核心 js ]

zTree 初始化方法，创建 zTree 必须使用此方法

1、页面需要进行 W3C 申明，例如：<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">。

2、需要首先加载 jquery-1.4.2.js 或其他更高版本的 jQuery 。

3、需要加载 jquery-ztree.core-3.0.js，如果需要用到 编辑功能 或 checkbox / radio 还需要分别加载 jquery-ztree.exedit-3.0.js 和 jquery-ztree.excheck-3.0.js 。

4、需要加载 zTreeStyle.css 以及 zTreeStyle 目录下的 img 文件。

5、如果需要使用自定义图标请参考相应的Demo。

6、请注意设置 zTree 的容器样式 class="ztree"，其中 "ztree" 这个 className，可以根据需要随意修改，别忘了修改 css 中对应名字就是了，对于容器如果需要增加其他特殊样式，可根据自己的需要进行修改。

Function 参数说明

**obj**jQuery Object

用于展现 zTree 的 DOM 容器

**zSetting**JSON

zTree 的配置数据，具体请参考 “setting 配置详解”中的各个属性详细说明

**zNodes**Array(JSON) / JSON

zTree 的节点数据，具体请参考 “treeNode 节点数据详解”中的各个属性详细说明

1、v3.x 支持单独添加一个节点，即如果只新增一个节点，不用必须包在数组中

2、如果需要异步加载根节点，可以设置为 null 或 [ ]

3、使用简单数据模式，请参考 setting.data.simpleData 内的属性说明

**返回值**JSON

zTree 对象，提供操作 zTree 的各种方法，对于通过 js 操作 zTree 来说必须通过此对象

如果不需要自行设定全局变量保存，可以利用 $.fn.zTree.getZTreeObj 方法随时获取

setting & function 举例

1. 简单创建 zTree 演示

<!DOCTYPE html>

<HTML>

<HEAD>

<TITLE> ZTREE DEMO </TITLE>

<meta http-equiv="content-type" content="text/html; charset=UTF-8">

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

<script type="text/javascript" src="jquery-1.4.2.js"></script>

<script type="text/javascript" src="jquery.ztree.core-3.4.js"></script>

<!--

<script type="text/javascript" src="jquery.ztree.excheck-3.4.js"></script>

<script type="text/javascript" src="jquery.ztree.exedit-3.4.js"></script>

-->

<SCRIPT type="text/javascript" >

var zTreeObj,

setting = {

view: {

selectedMulti: false

}

},

zTreeNodes = [

{"name":"网站导航", open:true, children: [

{ "name":"google", "url":"http://g.cn", "target":"\_blank"},

{ "name":"baidu", "url":"http://baidu.com", "target":"\_blank"},

{ "name":"sina", "url":"http://www.sina.com.cn", "target":"\_blank"}

]

}

];

$(document).ready(function(){

zTreeObj = $.fn.zTree.init($("#tree"), setting, zTreeNodes);

});

</SCRIPT>

</HEAD>

<BODY>

<ul id="tree" class="ztree" style="width:230px; overflow:auto;"></ul>

</BODY>

</HTML>

### $.fn.zTree.getZTreeObj

概述[ 依赖 jquery.ztree.core 核心 js ]

zTree v3.x 专门提供的根据 treeId 获取 zTree 对象的方法。

必须在初始化 zTree 以后才可以使用此方法。

有了这个方法，用户不再需要自己设定全局变量来保存 zTree 初始化后得到的对象了，而且在所有回调函数中全都会返回 treeId 属性，用户可以随时使用此方法获取需要进行操作的 zTree 对象

Function 参数说明

**treeId**String

zTree 的 DOM 容器的 id

**返回值**JSON

zTree 对象，提供操作 zTree 的各种方法，对于通过 js 操作 zTree 来说必须通过此对象

function 举例

1. 获取 id 为 tree 的 zTree 对象

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

### $.fn.zTree.destroy

概述[ 依赖 jquery.ztree.core 核心 js ]

从 zTree v3.4 开始提供销毁 zTree 的方法。

1、用此方法可以销毁指定 treeId 的 zTree，也可以销毁当前页面全部的 zTree。

2、销毁指定 treeId 的 zTree 也可以使用 zTreeObj.destroy() 方法。

3、重新使用已经被销毁的树，必须要使用 init 方法进行初始化。

Function 参数说明

**treeId**String

zTree 的 DOM 容器的 id

省略 treeId，表示销毁当前页面全部的 zTree

**返回值**无

目前无任何返回值

$.fn.zTree.destroy("treeDemo");

2. 销毁全部 的 zTree

$.fn.zTree.destroy();

### $.fn.zTree.\_z

概述[ 依赖 jquery.ztree.core 核心 js ]

zTree v3.x 内部的全部方法都可以通过 $.fn.zTree.\_z 进行调用，开放出来是为了更便于大家开发制作自己的 zTree 插件。

如无特殊需求请勿使用此对象，以及修改此对象内部的各个函数。

## zTreeObj

### zTreeObj.setting

概述[ 依赖 jquery.ztree.core 核心 js ]

zTree 对象使用的 setting 配置数据，详细请参考 “setting 配置详解”中的各个属性详细说明

v3.x 取消了原先操作 setting 的方法，让用户可以较自由的修改参数，但请注意，对于 zTree 初始化有影响的参数后期修改是不会起作用的，请对各个属性有较深入的了解以后再考虑进行修改。

### zTreeObj.addNodes

概述[ 依赖 jquery.ztree.core 扩展 js ]

添加节点。

v3.x 为了避免原先反复初始化造成的数据重复问题，在 初始化 和 添加节点 时内部进行 clone 操作。如果需要获取数据在 zTree 内的对象，请获取此方法的返回值。

请通过 zTree 对象执行此方法。

Function 参数说明

**parentNode**JSON

指定的父节点，如果增加根节点，请设置 parentNode 为 null 即可。

请务必保证此节点数据对象 是 zTree 内部的数据对象

**newNodes**JSON / Array(JSON)

需要增加的节点数据 JSON 对象集合，数据只需要满足 zTree 的节点数据必需的属性即可，详细请参考“treeNode 节点数据详解”

1、v3.x 支持单独添加一个节点，即如果只新增一个节点，不用必须包在数组中

2、使用简单数据模式，请参考 setting.data.simpleData 内的属性说明

**isSilent**Boolean

设定增加节点后是否自动展开父节点。

isSilent = true 时，不展开父节点，其他值或缺省状态都自动展开。

**返回值**Array(JSON)

返回值是 zTree 最终添加的节点数据集合

如果 newNodes 是单个节点数据 JSON，返回值也是将其包在 Array 内

请务必记住：返回值中的数据对象 是 newNodes 被 clone 后的，所以绝对不相等！

function 举例

1. 对于 id = "tree" 的 zTree 增加 1 个根节点

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

var newNode = {name:"newNode1"};

newNode = treeObj.addNodes(null, newNode);

2. 对于 id = "tree" 的 zTree 增加 3 个根节点

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

var newNodes = [{name:"newNode1"}, {name:"newNode2"}, {name:"newNode3"}];

newNodes = treeObj.addNodes(null, newNodes);

### zTreeObj.cancelEditName

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

取消节点的编辑名称状态，可以恢复原名称，也可以强行赋给新的名称。

请通过 zTree 对象执行此方法。

Function 参数说明

**newName**String

重新给定的新名称。

如果省略此参数，则恢复原名称。

**返回值**无

目前无任何返回值

function 举例

1. 取消 zTree 的编辑名称状态，恢复该节点原有名称

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

treeObj.cancelEditName();

2. 取消 zTree 的编辑名称状态，并且重新设定该节点名称

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

treeObj.cancelEditName("test\_new\_name");

### zTreeObj.cancelSelectedNode

概述[ 依赖 jquery.ztree.core 核心 js ]

取消节点的选中状态。

v3.x 支持多点同时选中，因此取消选中状态可以全部取消，也可以单独取消某个节点的选中状态。

请通过 zTree 对象执行此方法。

Function 参数说明

**treeNode**JSON

需要取消选中状态的节点。

请务必保证此节点数据对象 是 zTree 内部的数据对象

如果省略此参数，则将取消全部被选中节点的选中状态。

**返回值**无

目前无任何返回值

function 举例

1. 取消当前所有被选中节点的选中状态

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

treeObj.cancelSelectedNode();

2. 取消当前第一个被选中节点的选中状态

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

var nodes = treeObj.getSelectedNode();

if (nodes.length>0) {

treeObj.cancelSelectedNode(nodes[0]);

}

### zTreeObj.checkAllNodes

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

勾选 或 取消勾选 全部节点。[setting.check.enable = true 且 setting.check.chkStyle = "checkbox" 时有效]

此方法不会触发 beforeCheck / onCheck 事件回调函数。

请通过 zTree 对象执行此方法。

Function 参数说明

**checked**Boolean

checked = true 表示勾选全部节点

checked = false 表示全部节点取消勾选

不会影响 treeNode.nochecked = true 的节点。

不会影响未加载的节点。

**返回值**无

目前无任何返回值

function 举例

1. 勾选全部节点

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

treeObj.checkAllNodes(true);

### zTreeObj.copyNode

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

复制节点。

v3.x 复制节点时进行 clone 操作。如果需要获取数据在 zTree 内的对象，请获取此方法的返回值。

请通过 zTree 对象执行此方法。

Function 参数说明

**targetNode**JSON

要复制到的目标节点 JSON 数据

如果复制成为根节点，请设置 targetNode 为 null 即可

请务必保证此节点数据对象 是 zTree 内部的数据对象

**treeNode**JSON

需要被复制的节点数据

请务必保证此节点数据对象 是 zTree 内部的数据对象

**moveType**String

复制到目标节点的相对位置

"inner"：成为子节点，"prev"：成为同级前一个节点，"next"：成为同级后一个节点

**isSilent**Boolean

设定复制节点后是否自动展开父节点。

isSilent = true 时，不展开父节点，其他值或缺省状态都自动展开。

**返回值**JSON

返回值是最终加入到 zTree 内的节点数据

请务必记住：返回值中的数据对象 是 treeNode 被 clone 后的，所以绝对不相等！

function 举例

1. 将根节点中第二个节点 复制成为 第一个节点的子节点

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

var nodes = treeObj.getNodes();

treeObj.copyNode(nodes[0], nodes[1], "inner");

2. 将根节点中第二个节点 复制成为 第一个节点的前一个节点

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

var nodes = treeObj.getNodes();

treeObj.copyNode(nodes[0], nodes[1], "before");

### zTreeObj.destroy

概述[ 依赖 jquery.ztree.core 核心 js ]

从 zTree v3.4 开始提供销毁 zTree 的方法。

1、用此方法可以销毁 zTreeObj 代表的 zTree。

2、销毁当前页面全部的 zTree，也可以使用 $.fn.zTree.destroy() 方法。

3、重新使用已经被销毁的树，必须要使用 init 方法进行初始化。

Function Parameter Descriptions

**返回值**无

目前无任何返回值

function 举例

1. 销毁 id 为 "treeDemo" 的 zTree

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

zTreeObj.destroy();

### zTreeObj.editName

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

设置某节点进入编辑名称状态。

1、如果需要用 js 取消编辑名称状态，请使用 cancelEditName(newName) 方法。

2、可利用此方法让当前正编辑的节点 input 输入框获取焦点。

3、请通过 zTree 对象执行此方法。

Function 参数说明

**treeNode**JSON

指定进入编辑名称状态的节点 JSON 数据

请务必保证此节点数据对象 是 zTree 内部的数据对象

**返回值**无

目前无任何返回值

function 举例

1. 设置根节点第一个节点进入编辑名称状态

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

var nodes = treeObj.getNodes();

treeObj.editName(nodes[0]);

### zTreeObj.expandAll

概述[ 依赖 jquery.ztree.core 核心 js ]

展开 / 折叠 全部节点

此方法不会触发 beforeExpand / onExpand 和 beforeCollapse / onCollapse 事件回调函数。

请通过 zTree 对象执行此方法。

Function 参数说明

**expandFlag**Boolean

expandFlag = true 表示 展开 全部节点

expandFlag = false 表示 折叠 全部节点

**返回值**Boolean

返回值表示最终实际操作情况

true 表示 展开 全部节点

false 表示 折叠 全部节点

null 表示 不存在任何父节点

function 举例

1. 展开全部节点

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

treeObj.expandAll(true);

### zTreeObj.expandNode

概述[ 依赖 jquery.ztree.core 核心 js ]

展开 / 折叠 指定的节点

v3.x 中执行此方法可以触发 beforeExpand / onExpand 或 beforeCollapse / onCollapse 事件回调函数。便于减少冗余代码

请通过 zTree 对象执行此方法。

Function 参数说明

**treeNode**JSON

需要 展开 / 折叠 的节点数据

请务必保证此节点数据对象 是 zTree 内部的数据对象

**expandFlag**Boolean

expandFlag = true 表示 展开 节点

expandFlag = false 表示 折叠 节点

省略此参数，则根据对此节点的展开状态进行 toggle 切换

**sonSign**Boolean

sonSign = true 表示 全部子孙节点 进行与 expandFlag 相同的操作

sonSign = false 表示 只影响此节点，对于其 子孙节点无任何影响

sonSign = false 且 treeNode.open = expandFlag 时，不会触发回调函数，直接返回

省略此参数，等同于 false

**focus**Boolean

focus = true 表示 展开 / 折叠 操作后，通过设置焦点保证此焦点进入可视区域内

focus = false 表示 展开 / 折叠 操作后，不设置任何焦点

省略此参数，等同于 true

**callbackFlag**Boolean

callbackFlag = true 表示执行此方法时触发 beforeExpand / onExpand 或 beforeCollapse / onCollapse 事件回调函数

callbackFlag = false 表示执行此方法时不触发事件回调函数

省略此参数，等同于 false

**返回值**Boolean

返回值表示最终实际操作情况

true 表示 展开 节点

false 表示 折叠 节点

null 表示 不是父节点

### zTreeObj.getChangeCheckedNodes

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

获取输入框勾选状态被改变的节点集合（与原始数据 checkedOld 对比）。[setting.check.enable = true 时有效]

请通过 zTree 对象执行此方法。

Function 参数说明

**返回值**Array(JSON)

返回全部勾选状态被改变的节点集合 Array

如果需要获取每次操作后全部被改变勾选状态的节点数据，请在每次勾选操作后，遍历所有被改变勾选状态的节点数据，让其 checkedOld = checked 就可以了。

function 举例

1. 获取当前勾选状态被改变的节点集合

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

var nodes = treeObj.getChangeCheckedNodes();

### zTreeObj.getCheckedNodes

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

获取输入框被勾选 或 未勾选的节点集合。[setting.check.enable = true 时有效]

请通过 zTree 对象执行此方法。

Function 参数说明

**checked**Boolean

checked = true 表示获取 被勾选 的节点集合

checked = false 表示获取 未勾选 的节点集合

省略此参数，等同于 true。

对于 treeNode.nochecked = true 的节点不进行获取。

**返回值**Array(JSON)

返回全部符合要求的节点集合 Array

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

var nodes = treeObj.getCheckedNodes(true);

### zTreeObj.getNodeByParam

概述[ 依赖 jquery.ztree.core 核心 js ]

根据节点数据的属性搜索，获取条件完全匹配的节点数据 JSON 对象

请通过 zTree 对象执行此方法。

Function 参数说明

**key**String

需要精确匹配的属性名称

**value**?

需要精确匹配的属性值，可以是任何类型，只要保证与 key 指定的属性值保持一致即可

**parentNode**JSON

搜索范围，指定在某个父节点下的子节点中进行搜索

忽略此参数，表示在全部节点中搜索

**返回值**JSON

匹配精确搜索的节点数据

1、如无结果，返回 null

2、如有多个节点满足查询条件，只返回第一个匹配到的节点

function 举例

1. 查找 id = 1 的节点数据

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

var node = treeObj.getNodeByParam("id", 1, null);

### zTreeObj.getNodeByTId

概述[ 依赖 jquery.ztree.core 核心 js ]

根据 zTree 的唯一标识 tId 快速获取节点 JSON 数据对象

通过内部的 cache 获取，不需要遍历节点。

请通过 zTree 对象执行此方法。

Function 参数说明

**tId**String

节点在 zTree 内的唯一标识 tId

**返回值**JSON

tId 对应的节点 JSON 数据对象

如无结果，返回 null

function 举例

1. 获取 tId = "tree\_10" 的节点数据

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

var node = treeObj.getNodeByTId("tree\_10");

### zTreeObj.getNodeIndex

概述[ 依赖 jquery.ztree.core 核心 js ]

获取某节点在同级节点中的序号（从0开始）

请通过 zTree 对象执行此方法。

Function 参数说明

**treeNode**JSON

需要查询顺序的节点 JSON 数据对象

请务必保证此节点数据对象 是 zTree 内部的数据对象

**返回值**Number

返回值从 0 开始计数

如果不存在该节点数据，返回 -1

function 举例

1. 获取当前选中的第一个节点在同级节点中的序号

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

var nodes = treeObj.getSelectedNodes();

if (nodes.length>0) {

var index = treeObj.getNodeIndex(nodes[0]);

}

### zTreeObj.getNodes

概述[ 依赖 jquery.ztree.core 核心 js ]

获取 zTree 的全部节点数据

请通过 zTree 对象执行此方法。

Function 参数说明

**返回值**Array(JSON)

全部节点数据

1、Array 仅仅是根节点的集合（默认情况子节点都处于 children 属性下）；

2、如需遍历全部节点需要利用递归，或利用 transformToArray 方法 将数据变成简单的 Array 集合

3、对于异步加载模式下，尚未加载的子节点是无法通过此方法获取的。

function 举例

1. 获取全部节点数据

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

var nodes = treeObj.getNodes();

### zTreeObj.getNodesByFilter

概述[ 依赖 jquery.ztree.core 核心 js ]

根据自定义规则搜索节点数据 JSON 对象集合 或 单个节点数据

可自定义复杂的搜索规则

请通过 zTree 对象执行此方法。

Function 参数说明

**filter**Function

自定义过滤器函数 function filter(node) {...}

filter 参数：node (节点数据 JSON)

filter 返回值：boolean (true 表示符合搜索条件；false 表示不符合搜索条件)

**isSingle**Boolean

isSingle = true 表示只查找单个节点

isSingle = false 表示查找节点集合

忽略此参数，表示查找节点集合

**parentNode**JSON

可以指定在某个父节点下的子节点中搜索

忽略此参数，表示在全部节点中搜索

**invokeParam**任意类型

用户自定义的数据对象，用于 filter 中进行计算

**返回值**Array(JSON) / JSON

isSingle = true 返回 第一个找到的节点数据 JSON，无结果时返回 null

isSingle = false 返回 节点数据集合 Array(JSON)，无结果时返回 [ ]

function 举例

1. 查找 level = 2 & name 中包含 "test" 的节点数据

function filter(node) {

return (node.level == 2 && node.name.indexOf("test")>-1);

}

......

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

var node = treeObj.getNodesByFilter(filter, true); // 仅查找一个节点

var nodes = treeObj.getNodesByFilter(filter); // 查找节点集合

### zTreeObj.getNodesByParam

概述[ 依赖 jquery.ztree.core 核心 js ]

根据节点数据的属性搜索，获取条件完全匹配的节点数据 JSON 对象集合

请通过 zTree 对象执行此方法。

Function 参数说明

**key**String

需要精确匹配的属性名称

**value**?

需要精确匹配的属性值，可以是任何类型，只要保证与 key 指定的属性值保持一致即可

**parentNode**JSON

可以指定在某个父节点下的子节点中搜索

忽略此参数，表示在全部节点中搜索

**返回值**Array(JSON)

匹配精确搜索的节点数据集合

如无结果，返回 [ ]

function 举例

1. 查找 name = "test" 的节点数据

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

var nodes = treeObj.getNodesByParam("name", "test", null);

### zTreeObj.getNodesByParamFuzzy

概述[ 依赖 jquery.ztree.core 核心 js ]

根据节点数据的属性搜索，获取条件模糊匹配的节点数据 JSON 对象集合

请通过 zTree 对象执行此方法。

Function 参数说明

**key**String

需要模糊匹配的属性名称

**value**String

需要模糊匹配的属性值

模糊匹配只能针对 String 类型的数据

**parentNode**JSON

可以指定在某个父节点下的子节点中搜索

忽略此参数，表示在全部节点中搜索

**返回值**Array(JSON)

匹配模糊搜索的节点数据集合

如无结果，返回 [ ]

function 举例

1. 查找 name 包含 "test" 的节点数据

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

var nodes = treeObj.getNodesByParamFuzzy("name", "test", null);

### zTreeObj.getSelectedNodes

概述[ 依赖 jquery.ztree.core 核心 js ]

获取 zTree 当前被选中的节点数据集合

请通过 zTree 对象执行此方法。

Function 参数说明

**返回值**Array(JSON)

当前被选中的节点数据集合

function 举例

1. 获取当前被选中的节点数据集合

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

var nodes = treeObj.getSelectedNodes();

### zTreeObj.hideNodes

概述[ 依赖 jquery.ztree.exhide 扩展 js ]

隐藏一批节点。

1、此功能不支持 exedit 扩展，因此不要在编辑状态时使用隐藏节点的方法。

2、隐藏/显示节点，会影响节点的 isFirstNode 和 isLastNode 属性。

3、请通过 zTree 对象执行此方法。

Function 参数说明

**treeNodes**Array(JSON)

指定被隐藏的节点 JSON 数据集合

请务必保证这些节点数据对象 是 zTree 内部的数据对象

**返回值**无

目前无任何返回值

function 举例

1. 隐藏根节点第一个节点的子节点

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

var nodes = treeObj.getNodes();

treeObj.hideNodes(nodes[0].children);

### zTreeObj.hideNode

概述[ 依赖 jquery.ztree.exhide 扩展 js ]

隐藏某个节点。

1、此功能不支持 exedit 扩展，因此不要在编辑状态时使用隐藏节点的方法。

2、隐藏/显示节点，会影响节点的 isFirstNode 和 isLastNode 属性。

3、请通过 zTree 对象执行此方法。

Function 参数说明

**treeNode**JSON

指定被隐藏的节点 JSON 数据

请务必保证此节点数据对象 是 zTree 内部的数据对象

**返回值**无

目前无任何返回值

function 举例

1. 隐藏根节点第一个节点

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

var nodes = treeObj.getNodes();

treeObj.hideNode(nodes[0]);

### zTreeObj.moveNode

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

移动节点。

请通过 zTree 对象执行此方法。

Function 参数说明

**targetNode**JSON

要移动到的目标节点 JSON 数据

如果移动成为根节点，请设置 targetNode 为 null 即可

请务必保证此节点数据对象 是 zTree 内部的数据对象

**treeNode**JSON

需要被移动的节点数据

请务必保证此节点数据对象 是 zTree 内部的数据对象

**moveType**String

指定移动到目标节点的相对位置

"inner"：成为子节点，"prev"：成为同级前一个节点，"next"：成为同级后一个节点

**isSilent**Boolean

设定移动节点后是否自动展开父节点。

isSilent = true 时，不展开父节点，其他值或缺省状态都自动展开。

**返回值**JSON

返回值是最终被移动的节点数据，正常情况下与 treeNode 参数完全相同

如果 返回值 为 null，说明 移动失败，主要原因有：  
1、targetNode 是 treeNode 父节点，且 moveType = "inner"  
2、targetNode 是 treeNode 子孙节点

function 举例

1. 将根节点中第二个节点 移动成为 第一个节点的子节点

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

var nodes = treeObj.getNodes();

treeObj.moveNode(nodes[0], nodes[1], "inner");

2. 将根节点中第二个节点 移动成为 第一个节点的前一个节点

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

var nodes = treeObj.getNodes();

treeObj.moveNode(nodes[0], nodes[1], "before");

### zTreeObj.reAsyncChildNodes

概述[ 依赖 jquery.ztree.core 核心 js ]

强行异步加载父节点的子节点。[setting.async.enable = true 时有效]

已经加载过的父节点可反复使用此方法重新加载。

请通过 zTree 对象执行此方法。

Function 参数说明

**parentNode**JSON

指定需要异步加载的父节点 JSON 数据

1、parentNode = null 时，相当于从根节点 Root 进行异步加载

2、parentNode.isParent = false 时，不进行异步加载

3、请务必保证此节点数据对象 是 zTree 内部的数据对象

**reloadType**String

reloadType = "refresh" 表示清空后重新加载。

reloadType != "refresh" 时，表示追加子节点处理。

**isSilent**Boolean

设定异步加载后是否自动展开父节点。

isSilent = true 时，不展开父节点，其他值或缺省状态都自动展开。

**返回值**无

目前无任何返回值

function 举例

1. 重新异步加载 zTree

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

treeObj.reAsyncChildNodes(null, "refresh");

2. 重新异步加载当前选中的第一个节点

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

var nodes = treeObj.getSelectedNodes();

if (nodes.length>0) {

treeObj.reAsyncChildNodes(nodes[0], "refresh");

}

### zTreeObj.refresh

概述[ 依赖 jquery.ztree.core 核心 js ]

刷新 zTree 。

没有特殊必要，尽量不要使用此方法。单个节点更新请使用 updateNode 方法，异步加载模式下请使用 reAsyncChildNodes 方法。

请通过 zTree 对象执行此方法。

Function 参数说明

**返回值**无

目前无任何返回值

function 举例

1. 刷新 zTree

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

treeObj.refresh();

### zTreeObj.removeChildNodes

概述[ 依赖 jquery.ztree.core 扩展 js ]

清空某父节点的子节点。

1、清空子节点后，父节点会自动变为叶子节点，如需要父节点保持父节点状态，请设置 setting.data.keep.parent 属性。

2、请勿用此方法清空根节点，如果需要清空根节点，直接初始化 zTree，并且设置初始节点为 null 即可。

3、此方法不会触发任何事件回调函数。

请通过 zTree 对象执行此方法。

Function 参数说明

**parentNode**JSON

需要清空子节点的父节点数据

请务必保证此节点数据对象 是 zTree 内部的数据对象

**返回值**Array(JSON)

将该父节点的子节点数据返回，如果不存在则返回 null

function 举例

1. 清空选中的第一个节点的子节点

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

var nodes = treeObj.getSelectedNodes();

if (nodes && nodes.length>0) {

treeObj.removeChildNodes(nodes[0]);

}

### zTreeObj.removeNode

概述[ 依赖 jquery.ztree.core 扩展 js ]

删除节点。

v3.x 中删除节点可以触发 beforeRemove / onRemove 事件回调函数。便于减少冗余代码

请通过 zTree 对象执行此方法。

Function 参数说明

**treeNode**JSON

需要被删除的节点数据

请务必保证此节点数据对象 是 zTree 内部的数据对象

**callbackFlag**Boolean

callbackFlag = true 表示执行此方法时触发 beforeCheck & onCheck 事件回调函数

callbackFlag = false 表示执行此方法时不触发事件回调函数

省略此参数，等同于 false

**返回值**无

目前无任何返回值

function 举例

1. 删除所有选中的节点

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

var nodes = treeObj.getSelectedNodes();

for (var i=0, l=nodes.length; i < l; i++) {

treeObj.removeNode(nodes[i]);

}

### zTreeObj.selectNode

概述[ 依赖 jquery.ztree.core 核心 js ]

选中指定节点

v3.x 支持同时选中多个节点。

请通过 zTree 对象执行此方法。

Function 参数说明

**treeNode**JSON

需要被选中的节点数据

请务必保证此节点数据对象 是 zTree 内部的数据对象

**addFlag**Boolean

addFlag = true 表示追加选中，会出现多点同时被选中的情况

addFlag = false 表示单独选中，原先被选中的节点会被取消选中状态

setting.view.selectedMulti = false 时，此参数无效，始终进行单独选中

**返回值**无

目前无任何返回值

function 举例

1. 单独选中根节点中第一个节点

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

var nodes = treeObj.getNodes();

if (nodes.length>0) {

treeObj.selectNode(nodes[0]);

}

### zTreeObj.setChkDisabled

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

禁用 或 解禁 某个节点的 checkbox / radio [setting.check.enable = true 时有效]

1、节点的 checkbox / radio 被禁用后，无法勾选或取消勾选，但能够影响父节点的半选状态

2、请不要直接修改已加载节点的 treeNode.chkDisabled 属性。

3、请通过 zTree 对象执行此方法。

Function 参数说明

**treeNode**JSON

需要禁用 或 解禁 checkbox / radio 的节点数据

请务必保证此节点数据对象 是 zTree 内部的数据对象

**disabled**Boolean

disabled = true 表示禁用 checkbox / radio

disabled = false 表示解禁 checkbox / radio

省略此参数，等同于 disabled = false

不影响 treeNode.nochecked = true 的节点。

**返回值**无

目前无任何返回值

function 举例

1. 禁用当前选中的节点的 checkbox / radio

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

var nodes = treeObj.getSelectedNodes();

for (var i=0, l=nodes.length; i < l; i++) {

treeObj.setChkDisabled(nodes[i], true);

}

### zTreeObj.setEditable

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

设置 zTree 进入 / 取消 编辑状态。

对于编辑状态的各种功能需要提前设置对应 setting 中的不同属性

请通过 zTree 对象执行此方法。

Function 参数说明

**editable**Boolean

true 表示进入 编辑状态

false 表示取消 编辑状态

**返回值**无

目前无任何返回值

function 举例

1. 设置 zTree 进入编辑状态

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

treeObj.setEditable(true);

### zTreeObj.showNode

概述[ 依赖 jquery.ztree.exhide 扩展 js ]

显示某个被隐藏的节点。

1、此功能不支持 exedit 扩展，因此不要在编辑状态时使用隐藏节点的方法。

2、隐藏/显示节点，会影响节点的 isFirstNode 和 isLastNode 属性。

3、请通过 zTree 对象执行此方法。

Function 参数说明

**treeNode**JSON

指定被显示的节点 JSON 数据

请务必保证此节点数据对象 是 zTree 内部的数据对象

**返回值**无

目前无任何返回值

function 举例

1. 显示某个隐藏的节点

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

var node = treeObj.getNodeByParam("isHidden", true);

if (node) {

treeObj.showNode(node);

}

### zTreeObj.showNodes

概述[ 依赖 jquery.ztree.exhide 扩展 js ]

显示一批已经被隐藏的节点。

1、此功能不支持 exedit 扩展，因此不要在编辑状态时使用隐藏节点的方法。

2、隐藏/显示节点，会影响节点的 isFirstNode 和 isLastNode 属性。

3、请通过 zTree 对象执行此方法。

Function 参数说明

**treeNodes**Array(JSON)

指定被显示的节点 JSON 数据集合

请务必保证这些节点数据对象 是 zTree 内部的数据对象

**返回值**无

目前无任何返回值

function 举例

1. 显示全部隐藏的节点

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

var nodes = treeObj.getNodesByParam("isHidden", true);

treeObj.showNodes(nodes);

### zTreeObj.transformToArray

概述[ 依赖 jquery.ztree.core 核心 js ]

将 zTree 使用的标准 JSON 嵌套格式的数据转换为简单 Array 格式。(免去用户自行编写递归遍历全部节点的麻烦)

请通过 zTree 对象执行此方法。

Function 参数说明

**treeNodes**Array(JSON) / JSON

需要被转换的 zTree 节点数据对象集合 或 某个单独节点的数据对象

**返回值**Array(JSON)

转换后的简单 Array 数据格式

function 举例

1. 将 zTree 节点数据转换为简单 Array 格式

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

var nodes = treeObj.transformToArray(treeObj.getNodes());

### zTreeObj.transformTozTreeNodes

概述[ 依赖 jquery.ztree.core 核心 js ]

将简单 Array 格式数据转换为 zTree 使用的标准 JSON 嵌套数据格式。

使用此方法，请务必设置节点唯一标识属性名称 setting.data.simpleData.idKey 和 父节点唯一标识属性名称 setting.data.simpleData.pIdKey，并且让数据满足父子关系。

请通过 zTree 对象执行此方法。

Function 参数说明

**simpleNodes**Array(JSON) / JSON

需要被转换的简单 Array 格式数据 或 某个单独的数据对象

**返回值**Array(JSON)

zTree 使用的标准数据，子节点都存在于父节点数据的 children 属性中

如果 simpleNodes 是一个 JSON 对象，则被简单封装成长度为 1 的数组。

function 举例

1. 将简单 Array 格式转换为zTree使用的标准格式

var setting = {

data: {

simpleData: {

enable: true,

idKey: "id",

pIdKey: "pId",

rootPId: 0,

}

}

};

var simpleNodes = [

{"id":1, "pId":0, "name":"test1"},

{"id":11, "pId":1, "name":"test11"},

{"id":12, "pId":1, "name":"test12"},

{"id":111, "pId":11, "name":"test111"}

];

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

var nodes = treeObj.transformTozTreeNodes(simpleNodes);

### zTreeObj.updateNode

概述[ 依赖 jquery.ztree.core 核心 js ]

更新某节点数据，主要用于该节点显示属性的更新。

1、可针对 name、target、 url、icon、 iconSkin、checked、nocheck 等这几个用于显示效果的参数进行更新，其他用于 zTreeNodes 的参数请不要随意更新，对于展开节点，还请调用 expandNode方法，因此请勿随意修改 open 属性。

2、用此方法修改 checked 勾选状态不会触发 beforeCheck / onCheck 事件回调函数。

请通过 zTree 对象执行此方法。

Function 参数说明

**treeNode**JSON

指定需要更新的节点 JSON 数据

请务必保证此节点数据对象 是 zTree 内部的数据对象

**checkTypeFlag**Boolean

checkTypeFlag = true 表示按照 setting.check.chkboxType 属性进行父子节点的勾选联动操作

checkTypeFlag = false 表示只修改此节点勾选状态，无任何勾选联动操作

当 setting.check.enable = true 且 setting.check.chkStyle = "checkbox" 时才有效

不影响父子节点中 treeNode.nochecked = true 的节点。

**返回值**无

目前无任何返回值

function 举例

1. 更新根节点中第一个节点的名称

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

var nodes = treeObj.getNodes();

if (nodes.length>0) {

nodes[0].name = "test";

treeObj.updateNode(nodes[0]);

}

# treeNode 节点数据详解

treeNode : {

checked

children

chkDisabled

click

getCheckStatus ()

getNextNode ()

getParentNode ()

getPreNode ()

halfCheck

icon

iconClose

iconOpen

iconSkin

isHidden

isParent

name

nocheck

open

target

url

\*DIY\*

[check\_Child\_State]

[check\_Focus]

[checkedOld]

[editNameFlag]

[isAjaxing]

[isFirstNode]

[isHover]

[isLastNode]

[level]

[parentTId]

[tId]

[zAsync]

}

## treeNode

### treeNode.checked

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

节点的 checkBox / radio 的 勾选状态。[setting.check.enable = true & treeNode.nocheck = false 时有效]

1、如果不使用 checked 属性设置勾选状态，请修改 setting.data.key.checked

2、建立 treeNode 数据时设置 treeNode.checked = true 可以让节点的输入框默认为勾选状态

3、修改节点勾选状态，可以使用 treeObj.checkNode / checkAllNodes / updateNode 方法，具体使用哪种请根据自己的需求而定

4、为了解决部分朋友生成 json 数据出现的兼容问题, 支持 "false","true" 字符串格式的数据

默认值：false

Boolean 格式说明

true 表示节点的输入框被勾选

false 表示节点的输入框未勾选

treeNode 举例

1. 初始化的数据设置 默认为勾选状态

var nodes = [

{ "id":1, "name":"test1", checked:true },

{ "id":2, "name":"test2", checked:true }

]

2. 获取第一个根节点的勾选状态

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

var checked = treeObj.getNodes()[0].checked;

### treeNode.children

概述[ 依赖 jquery.ztree.core 核心 js ]

节点的子节点数据集合。

1、如果不使用 children 属性保存子节点数据，请修改 setting.data.key.children

2、异步加载时，对于设置了 isParent = true 的节点，在展开时将进行异步加载

默认值：无

Array(JSON) 格式说明

标准的 JSON 数据对象

treeNode 举例

1. 初始化的标准嵌套格式的 JSON 数据对象

var nodes = [

{ "id":1, "name":"test1",

children: [

{ "id":3, "name":"test3"},

{ "id":4, "name":"test4"},

{ "id":5, "name":"test5"}

]

},

{ "id":2, "name":"test2" }

]

2. 获取第一个根节点的子节点

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

var nodes = treeObj.getNodes()[0].children;

### treeNode.chkDisabled

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

1、设置节点的 checkbox / radio 是否禁用 [setting.check.enable = true 时有效]

2、为了解决部分朋友生成 json 数据出现的兼容问题, 支持 "false","true" 字符串格式的数据

3、请勿对已加载的节点修改此属性，禁止 或 取消禁止 请使用 setChkDisabled() 方法

默认值：false

Boolean 格式说明

true 表示此节点的 checkbox / radio 被禁用。

false 表示此节点的 checkbox / radio 可以使用。

treeNode 举例

1. 禁用节点 checkbox / radio

var nodes = [

{ "id":1, "name":"test1", "checked":true, "chkDisabled":true},

{ "id":2, "name":"test2", "chkDisabled":true},

{ "id":3, "name":"test3"}

]

### treeNode.click

概述[ 依赖 jquery.ztree.core 核心 js ]

最简单的 click 事件操作。相当于 onclick="..." 的内容。 如果操作较复杂，请使用 onClick 事件回调函数。

由于 IE 对于 onclick 和 click事件共存时的处理与其他浏览器不同，所以请不要利用此参数控制是否允许跳转的操作（例如：treeNode.click = "return false;"）。如有类似需求，请不要使用 url 属性设置网址，同时利用 onClick 回调函数控制跳转。

默认值：无

String 格式说明

标准 javascript 语法， 例如：alert("test"); 等

treeNode 举例

1. 设置某节点点击时，弹出信息框

var nodes = [

{ "id":1, "name":"Google CN", "url":"http://g.cn", "click":"alert('test');"},

......

]

### treeNode.getCheckStatus

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

获取节点 checkbox / radio 半勾选状态。[setting.check.enable = true 时有效]

初始化节点数据时，由 zTree 增加此属性，请勿提前赋值

Function 参数说明

**返回值**JSON

{

checked: true, //等同于 treeNode.checked

half: true //规则见下表

}

| **setting.check.checkType = "checkbox"** | | | |
| --- | --- | --- | --- |
| **treeNode.checked** | **treeNode.check\_Child\_State** | **treeNode.halfCheck** | **half** |
| - | - | true | true |
|  | | | |
| true | -1 | false | false |
| true | 0 | false | true |
| true | 1 | false | true |
| true | 2 | false | false |
|  | | | |
| false | -1 | false | false |
| false | 0 | false | false |
| false | 1 | false | true |
| false | 2 | false | true |

| **setting.check.checkType = "radio"** | | | |
| --- | --- | --- | --- |
| **treeNode.checked** | **treeNode.check\_Child\_State** | **treeNode.halfCheck** | **half** |
| - | - | true | true |
|  | | | |
| true | -1 | false | false |
| true | 0 | false | false |
| true | 2 | false | true |
|  | | | |
| false | -1 | false | false |
| false | 0 | false | false |
| false | 2 | false | true |

treeNode 举例

1. 获取第一个根节点的半选状态

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

var halfCheck = treeObj.getNodes()[0].getCheckStatus();

### treeNode.getNextNode

概述[ 依赖 jquery.ztree.core 核心 js ]

获取与 treeNode 节点相邻的后一个节点。

初始化节点数据时，由 zTree 增加此属性，请勿提前赋值

Function 参数说明

**返回值**JSON

与 treeNode 节点相邻的后一个节点。

如果 treeNode 是最后一个节点，返回 null 。

treeNode 举例

1. 获取当前被选中的节点的下一个节点

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

var sNodes = treeObj.getSelectedNodes();

if (sNodes.length > 0) {

var node = sNodes[0].getNextNode();

}

### treeNode.getParentNode

概述[ 依赖 jquery.ztree.core 核心 js ]

获取 treeNode 节点的父节点。

初始化节点数据时，由 zTree 增加此属性，请勿提前赋值

Function 参数说明

**返回值**JSON

treeNode 节点的父节点 JSON 数据对象。

如果 treeNode 是根节点，返回 null 。

treeNode 举例

1. 获取当前被选中的节点的父节点

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

var sNodes = treeObj.getSelectedNodes();

if (sNodes.length > 0) {

var node = sNodes[0].getParentNode();

}

### treeNode.getPreNode

概述[ 依赖 jquery.ztree.core 核心 js ]

获取与 treeNode 节点相邻的前一个节点。

初始化节点数据时，由 zTree 增加此属性，请勿提前赋值

Function 参数说明

**返回值**JSON

与 treeNode 节点相邻的前一个节点。

如果 treeNode 是第一个节点，返回 null 。

treeNode 举例

1. 获取当前被选中的节点的前一个节点

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

var sNodes = treeObj.getSelectedNodes();

if (sNodes.length > 0) {

var node = sNodes[0].getPreNode();

}

### treeNode.halfCheck

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

强制节点的 checkBox / radio 的 半勾选状态。[setting.check.enable = true & treeNode.nocheck = false 时有效]

1、强制为半勾选状态后，不再进行自动计算半勾选状态

2、设置 treeNode.halfCheck = false 或 null 才能恢复自动计算半勾选状态

3、为了解决部分朋友生成 json 数据出现的兼容问题, 支持 "false","true" 字符串格式的数据

默认值：false

Boolean 格式说明

true 表示节点的输入框 强行设置为半勾选

false 表示节点的输入框 根据 zTree 的规则自动计算半勾选状态

treeNode 举例

1. 初始化的数据设置 默认为半勾选状态

var nodes = [

{ "id":1, "name":"test1", isParent:true, checked:true, halfCheck:true },

{ "id":2, "name":"test2", isParent:true, checked:false, halfCheck:true },

{ "id":3, "name":"test3", isParent:true, checked:true },

{ "id":4, "name":"test4", isParent:true, checked:false }

]

### treeNode.icon

概述[ 依赖 jquery.ztree.core 核心 js ]

节点自定义图标的 URL 路径。

1、父节点如果只设置 icon ，会导致展开、折叠时都使用同一个图标

2、父节点展开、折叠使用不同的个性化图标需要同时设置 treeNode.iconOpen / treeNode.iconClose 两个属性

3、如果想利用 className 设置个性化图标，需要设置 treeNode.iconSkin 属性

默认值：无

String 格式说明

图标图片的 url 可以是相对路径也可以是绝对路径

设置相对路径请注意页面与图片之间的关系，确保图片能够正常加载

treeNode 举例

1. 设置节点的个性化图标

var nodes = [

//父节点展开 折叠时使用相同的图标

{ name:"父节点1", icon:"/img/parent.gif"},

//父节点展开 折叠时分别使用不同的图标

{ name:"父节点2", iconOpen:"/img/open.gif", iconClose:"/img/close.gif"},

//叶子节点个性化图标

{ name:"叶子节点", icon:"/img/leaf.gif"}

]

### treeNode.iconClose

概述[ 依赖 jquery.ztree.core 核心 js ]

父节点自定义折叠时图标的 URL 路径。

1、此属性只针对父节点有效

2、此属性必须与 iconOpen 同时使用

3、如果想利用 className 设置个性化图标，需要设置 treeNode.iconSkin 属性

默认值：无

String 格式说明

图标图片的 url 可以是相对路径也可以是绝对路径

设置相对路径请注意页面与图片之间的关系，确保图片能够正常加载

treeNode 举例

1. 设置节点的个性化图标

var nodes = [

//父节点展开 折叠时使用相同的图标

{ name:"父节点1", icon:"/img/parent.gif"},

//父节点展开 折叠时分别使用不同的图标

{ name:"父节点2", iconOpen:"/img/open.gif", iconClose:"/img/close.gif"},

//叶子节点个性化图标

{ name:"叶子节点", icon:"/img/leaf.gif"}

]

### treeNode.iconOpen

概述[ 依赖 jquery.ztree.core 核心 js ]

父节点自定义展开时图标的 URL 路径。

1、此属性只针对父节点有效

2、此属性必须与 iconClose 同时使用

3、如果想利用 className 设置个性化图标，需要设置 treeNode.iconSkin 属性

默认值：无

String 格式说明

图标图片的 url 可以是相对路径也可以是绝对路径

设置相对路径请注意页面与图片之间的关系，确保图片能够正常加载

treeNode 举例

1. 设置节点的个性化图标

var nodes = [

//父节点展开 折叠时使用相同的图标

{ name:"父节点1", icon:"/img/parent.gif"},

//父节点展开 折叠时分别使用不同的图标

{ name:"父节点2", iconOpen:"/img/open.gif", iconClose:"/img/close.gif"},

//叶子节点个性化图标

{ name:"叶子节点", icon:"/img/leaf.gif"}

]

### treeNode.iconSkin

概述[ 依赖 jquery.ztree.core 核心 js ]

节点自定义图标的 className

1、需要修改 css，增加相应 className 的设置

2、css 方式简单、方便，并且同时支持父节点展开、折叠状态切换图片

3、css 建议采用图片分割渲染的方式以减少反复加载图片，并且避免图片闪动

4、zTree v3.x 的 iconSkin 同样支持 IE6

5、如果想直接使用 图片的Url路径 设置节点的个性化图标，需要设置 treeNode.icon / treeNode.iconOpen / treeNode.iconClose 属性

默认值：无

String 格式说明

设置个性图标的 className

css & treeNode 举例

1. 设置节点的个性化图标

css 内容：

.ztree li span.button.diy01\_ico\_open, .ztree li span.button.diy01\_ico\_close{...}

.ztree li span.button.diy02\_ico\_open{...}

.ztree li span.button.diy02\_ico\_close{...}

.ztree li span.button.diy03\_ico\_docu{...}

js中节点数据：

var nodes = [

//父节点展开 折叠时使用相同的图标

{ name:"父节点1", iconSkin:"diy01"},

//父节点展开 折叠时分别使用不同的图标

{ name:"父节点2", iconSkin:"diy02"},

//叶子节点个性化图标

{ name:"叶子节点", iconSkin:"diy03"}

]

### treeNode.isHidden

概述[ 依赖 jquery.ztree.exhide 扩展 js ]

判断 treeNode 节点是否被隐藏。

1、初始化 zTree 时，如果节点设置 isHidden = true，会被自动隐藏

2、请勿对已加载的节点修改此属性，隐藏 / 显示 请使用 hideNode() / hideNodes() / showNode() / showNodes() 方法

Boolean 格式说明

true 表示被隐藏

false 表示被显示

treeNode 举例

1. 查看第一个根节点是否被隐藏

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

var sNodes = treeObj.getNodes();

if (sNodes.length > 0) {

var isHidden = sNodes[0].isHidden;

}

### treeNode.isParent

概述[ 依赖 jquery.ztree.core 核心 js ]

记录 treeNode 节点是否为父节点。

1、初始化节点数据时，根据 treeNode.children 属性判断，有子节点则设置为 true，否则为 false

2、初始化节点数据时，如果设定 treeNode.isParent = true，即使无子节点数据，也会设置为父节点

3、为了解决部分朋友生成 json 数据出现的兼容问题, 支持 "false","true" 字符串格式的数据

Boolean 格式说明

true 表示是父节点

false 表示不是父节点

treeNode 举例

1. 查看当前被选中的节点是否是父节点

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

var sNodes = treeObj.getSelectedNodes();

if (sNodes.length > 0) {

var isParent = sNodes[0].isParent;

}

### treeNode.name

概述[ 依赖 jquery.ztree.core 核心 js ]

节点名称。

1、如果不使用 name 属性保存节点名称，请修改 setting.data.key.name

默认值：无

String 格式说明

节点显示的名称字符串，标准 String 即可，所有特殊字符都会被自动转义

treeNode 举例

1. 设置节点的名称为 test1、test2、test3

var nodes = [

{ "id":1, "name":"test1"},

{ "id":2, "name":"test2"},

{ "id":3, "name":"test3"}

]

### treeNode.nocheck

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

1、设置节点是否隐藏 checkbox / radio [setting.check.enable = true 时有效]

2、为了解决部分朋友生成 json 数据出现的兼容问题, 支持 "false","true" 字符串格式的数据

默认值：false

Boolean 格式说明

true 表示此节点不显示 checkbox / radio，不影响勾选的关联关系，不影响父节点的半选状态。

false 表示节点具有正常的勾选功能

treeNode 举例

1. 不显示某个节点的 checkbox / radio

var nodes = [

{ "id":1, "name":"test1", "nocheck":true},

{ "id":2, "name":"test2"},

{ "id":3, "name":"test3"}

]

### treeNode.open

概述[ 依赖 jquery.ztree.core 核心 js ]

记录 treeNode 节点的 展开 / 折叠 状态。

1、初始化节点数据时，如果设定 treeNode.open = true，则会直接展开此节点

2、叶子节点 treeNode.open = false

3、为了解决部分朋友生成 json 数据出现的兼容问题, 支持 "false","true" 字符串格式的数据

默认值：false

Boolean 格式说明

true 表示节点为 展开 状态

false 表示节点为 折叠 状态

treeNode 举例

1. 查看当前被选中的节点的 展开 / 折叠 状态

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

var sNodes = treeObj.getSelectedNodes();

if (sNodes.length > 0) {

var isOpen = sNodes[0].open;

}

### treeNode.target

概述[ 依赖 jquery.ztree.core 核心 js ]

设置点击节点后在何处打开 url。[treeNode.url 存在时有效]

默认值：无

String 格式说明

同超链接 target 属性: "\_blank", "\_self" 或 其他指定窗口名称

省略此属性，则默认为 "\_blank"

treeNode 举例

1. 设置点击某节点时，弹出新页面

var nodes = [

{ "id":1, "name":"test1", "url":"http://myTest.com", "target":"\_blank"},

......

]

### treeNode.url

概述[ 依赖 jquery.ztree.core 核心 js ]

节点链接的目标 URL

1、编辑模式 (setting.edit.enable = true) 下此属性功能失效，如果必须使用类似功能，请利用 onClick 事件回调函数自行控制。

2、如果需要在 onClick 事件回调函数中进行跳转控制，那么请将 URL 地址保存在其他自定义的属性内，请勿使用 url

默认值：无

String 格式说明

同超链接 href 属性

treeNode 举例

1. 设置某节点点击时，跳转到 g.cn

var nodes = [

{ "id":1, "name":"Google CN", "url":"http://g.cn"},

......

]

### treeNode.\* DIY \*

概述[ 依赖 jquery.ztree.core 核心 js ]

用于保存节点的其他自定义数据信息，不要与 zTree 使用的属性相同即可，用户可随意设定。

treeNode 举例

1. 设置节点的备用英文名称

var node = { "id":1, "name":"test1", "ename":"test eName"};

### treeNode.check\_Child\_State

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

用于设置节点的子节点的 checkBox / radio 的半选状态。[setting.check.enable = true 时有效]

v3.x 针对节点数据对象提供 treeNode.getCheckStatus() 方法获取标准的半选状态

zTree 内部使用，请勿进行初始化 或 随意修改

默认值：true

Number 格式说明

规则如下：

| **setting.check.checkType = "checkbox"** | | | |
| --- | --- | --- | --- |
| **treeNode.check\_Child\_State** | **勾选状态说明** |  |  |
| -1 | 不存在子节点 或 子节点全部设置为 nocheck = true |  |  |
| 0 | 无 子节点被勾选 |  |  |
| 1 | 部分 子节点被勾选 |  |  |
| 2 | 全部 子节点被勾选 |  |  |

| **setting.check.checkType = "radio"** | | | |
| --- | --- | --- | --- |
| **treeNode.check\_Child\_State** | **勾选状态说明** |  |  |
| -1 | 不存在子节点 或 子节点全部设置为 nocheck = true |  |  |
| 0 | 无 子节点被勾选 |  |  |
| 2 | 有 子节点被勾选 |  |  |

### treeNode.check\_Focus

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

用于设置节点的 checkBox / radio 的 focus 状态。[setting.check.enable = true 时有效]

zTree 内部使用，请勿进行初始化 或 随意修改

默认值：false

Boolean 格式说明

true 表示当前鼠标移动到输入框内

false 表示当前鼠标移动到输入框外

### treeNode.checkedOld

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

节点的 checkBox / radio 在初始化时的 勾选状态。[setting.check.enable = true & treeNode.nocheck = false 时有效]

1、zTree 初始化节点数据时会对此属性进行赋值，因此请勿对此属性初始化

2、如需配合 zTreeObj.getChangeCheckedNodes 方法实现特殊功能，可以根据需求在使用中自行修改 checkedOld 数据

默认值：checked的初始化值

Boolean 格式说明

true 表示节点初始化时 输入框被勾选

false 表示节点初始化时 输入框未勾选

treeNode 举例

1. 获取第一个根节点的初始勾选状态

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

var checkedOld = treeObj.getNodes()[0].checkedOld;

### treeNode.editNameFlag

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

用于记录节点是否处于编辑名称状态。[setting.edit.enable = true 时有效]

zTree 内部使用，请勿进行初始化 或 随意修改

默认值：false

Boolean 格式说明

true 表示节点处于编辑名称状态

false 表示节点未处于编辑名称状态

### treeNode.isAjaxing

概述[ 依赖 jquery.ztree.core 核心 js ]

记录 treeNode 节点是否正在进行异步加载。

初始化节点数据时，由 zTree 增加此属性，请勿提前赋值

Boolean 格式说明

true 表示节点正在进行异步加载

false 表示节点没有进行异步加载

treeNode 举例

1. 查看当前被选中的节点是否节点正在进行异步加载

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

var sNodes = treeObj.getSelectedNodes();

if (sNodes.length > 0) {

var isAjaxing = sNodes[0].isAjaxing;

}

### treeNode.isFirstNode

概述[ 依赖 jquery.ztree.core 核心 js ]

记录 treeNode 节点是否为同级节点中的第一个节点。

使用 exhide 扩展后，只针对显示的节点设置此属性

初始化节点数据时，由 zTree 增加此属性，请勿提前赋值

Boolean 格式说明

true 表示是同级节点中的第一个节点

false 表示不是同级节点中的第一个节点

节点被隐藏后，isFirstNode = false

treeNode 举例

1. 查看当前被选中的节点是否是同级节点中的第一个节点

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

var sNodes = treeObj.getSelectedNodes();

if (sNodes.length > 0) {

var isFirstNode = sNodes[0].isFirstNode;

}

### treeNode.isHover

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

记录节点 的 hover 状态，主要用于 setting.view.addHoverDom / removeHoverDom 。

zTree 内部使用，请勿进行初始化 或 随意修改

默认值：false

Boolean 格式说明

true 表示节点处于 hover 状态

false 表示节点未处于 hover 状态

### treeNode.isLastNode

概述[ 依赖 jquery.ztree.core 核心 js ]

记录 treeNode 节点是否为同级节点中的最后一个节点。

使用 exhide 扩展后，只针对显示的节点设置此属性

初始化节点数据时，由 zTree 增加此属性，请勿提前赋值

Boolean 格式说明

true 表示是同级节点中的最后一个节点

false 表示不是同级节点中的最后一个节点

节点被隐藏后，isLastNode = false

treeNode 举例

1. 查看当前被选中的节点是否是同级节点中的最后一个节点

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

var sNodes = treeObj.getSelectedNodes();

if (sNodes.length > 0) {

var isLastNode = sNodes[0].isLastNode;

}

### treeNode.level

概述[ 依赖 jquery.ztree.core 核心 js ]

记录节点的层级

初始化节点数据时，由 zTree 增加此属性，请勿提前赋值

Number 格式说明

根节点 level = 0，依次递增

treeNode 举例

1. 查看当前被选中的节点的级数

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

var sNodes = treeObj.getSelectedNodes();

if (sNodes.length > 0) {

var level = sNodes[0].level;

}

### treeNode.parentTId

概述[ 依赖 jquery.ztree.core 核心 js ]

treeNode 节点的父节点唯一标识 tId。

1、v3.x 用 parentTId 替换了原先的 parentNode 属性，同时增加了 getParentNode 方法，以避免原先 parentNode 造成的 clone 死循环

2、初始化节点数据时，由 zTree 增加此属性，请勿提前赋值

String 格式说明

zTree 内部生成的节点唯一标识，请参考 treeNode.tId 的说明

如果 treeNode 是根节点，则 parentTId = null

treeNode 举例

1. 查看当前被选中的节点的父节点 tId

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

var sNodes = treeObj.getSelectedNodes();

if (sNodes.length > 0) {

var parentTId = sNodes[0].parentTId;

}

### treeNode.tId

概述[ 依赖 jquery.ztree.core 核心 js ]

treeNode 节点的唯一标识 tId。

初始化节点数据时，由 zTree 增加此属性，请勿提前赋值

String 格式说明

生成规则：setting.treeId + "\_" + 内部计数

treeNode 举例

1. 查看当前被选中的节点的 tId

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

var sNodes = treeObj.getSelectedNodes();

if (sNodes.length > 0) {

var tId = sNodes[0].tId;

}

### treeNode.zAsync

概述[ 依赖 jquery.ztree.core 核心 js ]

记录 treeNode 节点是否已经进行过异步加载，避免父节点反复异步加载数据。

初始化节点数据时，由 zTree 增加此属性，请勿提前赋值

默认值：false （无子节点的父节点）； true （有子节点的父节点 & 叶子节点）

Boolean 格式说明

true 表示父节点展开时不需要自动异步加载

false 表示父节点展开时需要自动异步加载

此参数不会对 reAsyncChildNodes 方法造成任何影响

treeNode 举例

1. 查看当前被选中的节点是否节点是否需要自动异步加载

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

var sNodes = treeObj.getSelectedNodes();

if (sNodes.length > 0) {

var zAsync = sNodes[0].zAsync;

}

# setting 配置详解

var setting = {

treeId : "",

treeObj : null,

async : {

autoParam : [],

contentType : "application...",

dataFilter : null,

dataType : "text",

enable : false,

otherParam : [],

type : "post",

url : ""

},

callback : {

beforeAsync : null,

beforeCheck : null,

beforeClick : null,

beforeCollapse : null,

beforeDblClick : null,

beforeDrag : null,

beforeDragOpen : null,

beforeDrop : null,

beforeEditName : null,

beforeExpand : null,

beforeMouseDown : null,

beforeMouseUp : null,

beforeRemove : null,

beforeRename : null,

beforeRightClick : null,

onAsyncError : null,

onAsyncSuccess : null,

onCheck : null,

onClick : null,

onCollapse : null,

onDblClick : null,

onDrag : null,

onDrop : null,

onExpand : null,

onMouseDown : null,

onMouseUp : null,

onNodeCreated : null,

onRemove : null,

onRename : null,

onRightClick : null

},

check : {

autoCheckTrigger : false,

chkboxType : {"Y": "ps", "N": "ps"},

chkStyle : "checkbox",

enable : false,

nocheckInherit : false

radioType : "level"

},

data : {

keep : {

leaf : false,

parent : false

},

key : {

checked : "checked",

children : "children",

name : "name",

title : ""

url : "url"

},

simpleData : {

enable : false,

idKey : "id",

pIdKey : "pId",

rootPId : null

}

},

edit : {

drag : {

autoExpandTrigger : true,

isCopy : true,

isMove : true,

prev : true,

next : true,

inner : true,

borderMax : 10,

borderMin : -5,

minMoveSize : 5,

maxShowNodeNum : 5,

autoOpenTime : 500

},

editNameSelectAll : false,

enable : false,

removeTitle : "remove",

renameTitle : "rename",

showRemoveBtn : true,

showRenameBtn : true

},

view : {

addDiyDom : null,

addHoverDom : null,

autoCancelSelected : true,

dblClickExpand : true,

expandSpeed : "fast",

fontCss : {},

nameIsHTML : false,

removeHoverDom : null,

selectedMulti : true,

showIcon : true,

showLine : true,

showTitle : true

}

}

## setting

### setting.treeId

概述[ 依赖 jquery.ztree.core 核心 js ]

zTree 的唯一标识，初始化后，等于 用户定义的 zTree 容器的 id 属性值。

请勿进行初始化 或 修改，属于内部参数。

### setting.treeObj

概述[ 依赖 jquery.ztree.core 核心 js ]

zTree 容器的 jQuery 对象，主要功能：便于操作。

请勿进行初始化 或 修改，属于内部参数。

## setting.async

### setting.async.autoParam

概述[ 依赖 jquery.ztree.core 核心 js ]

异步加载时需要自动提交父节点属性的参数。[setting.async.enable = true 时生效]

默认值：[ ]

Array(String) 格式说明

1、将需要作为参数提交的属性名称，制作成 Array 即可，例如：["id", "name"]

2、可以设置提交时的参数名称，例如 server 只接受 zId : ["id=zId"]

setting 举例

1. 设置 id 属性为自动提交的参数

var setting = {

async: {

enable: true,

url: "http://host/getNode.php",

autoParam: ["id"]

}

};

假设 异步加载 父节点(node = {id:1, name:"test"}) 的子节点时，将提交参数 id=1

......

2. 设置 id 属性作为 zId 成为自动提交的参数

var setting = {

async: {

enable: true,

url: "http://host/getNode.php",

autoParam: ["id=zId"]

}

};

假设 对父节点 node = {id:1, name:"test"}，进行异步加载时，将提交参数 zId=1

......

### setting.async.contentType

概述[ 依赖 jquery.ztree.core 核心 js ]

Ajax 提交参数的数据类型。[setting.async.enable = true 时生效]

默认值："application/x-www-form-urlencoded"

String 格式说明

contentType = "application/x-www-form-urlencoded" 可以满足绝大部分请求，按照标准的 Form 格式提交参数

contentType = "application/json" 可以满足 .Net 的编程需要，按照 JSON 格式提交参数

setting 举例

1. 设置 Ajax 提交参数的数据类型为 JSON 格式

var setting = {

async: {

enable: true,

contentType: "application/json",

url: "http://host/getNode.php",

autoParam: ["id", "name"]

}

};

......

### setting.async.dataFilter

概述[ 依赖 jquery.ztree.core 核心 js ]

用于对 Ajax 返回数据进行预处理的函数。[setting.async.enable = true 时生效]

默认值：null

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**parentNode**JSON

进行异步加载的父节点 JSON 数据对象

对根进行异步加载时，parentNode = null

**childNodes**Array(JSON) / JSON / String

异步加载获取到的数据转换后的 Array(JSON) / JSON / String 数据对象

v3.4开始 支持 XML 数据格式的 String

**返回值**Array(JSON) / JSON

返回值是 zTree 支持的JSON 数据结构即可。

v3.x 支持单个 JSON 节点数据进行加载

setting & function 举例

1. 修改异步获取到的节点name属性

function ajaxDataFilter(treeId, parentNode, childNodes) {

if (childNodes) {

for(var i =0; i < childNodes.length; i++) {

childNodes[i].name += "\_filter";

}

}

return childNodes;

};

var setting = {

async: {

enable: true,

url: "http://host/getNode.php",

dataFilter: ajaxDataFilter

}

};

......

### setting.async.dataType

概述[ 依赖 jquery.ztree.core 核心 js ]

Ajax 获取的数据类型。[setting.async.enable = true 时生效]

默认值："text"

String 格式说明

dataType = "text" 可以满足绝大部分请求

其余 dataType 类型请参考 jQuery ajax 中的 dataType 参数

setting 举例

1. 设置 Ajax 获取的数据类型为 纯文本

var setting = {

async: {

enable: true,

dataType: "text",

url: "http://host/getNode.php",

autoParam: ["id", "name"]

}

};

......

### setting.async.enable

概述[ 依赖 jquery.ztree.core 核心 js ]

设置 zTree 是否开启异步加载模式

默认值：false

Boolean 格式说明

true 表示 开启 异步加载模式

false 表示 关闭 异步加载模式

如果设置为 true，请务必设置 setting.async 内的其它参数。

如果需要根节点也异步加载，初始化时 treeNodes 参数设置为 null 即可。

setting 举例

1. 需要开启异步加载模式

var setting = {

async: {

enable: true,

url: "http://host/getNode.php",

autoParam: ["id", "name"]

}

};

......

### setting.async.otherParam

概述[ 依赖 jquery.ztree.core 核心 js ]

Ajax 请求提交的静态参数键值对。[setting.async.enable = true 时生效]

默认值：[ ]

Array(String) 格式说明

可以为空[ ]，如果有 key，则必须存在 value。 例如：[key, value]

JSON 格式说明

直接用 JSON 格式制作键值对，例如：{ key1:value1, key2:value2 }

setting 举例

1. 设置 Array(String) 格式的参数

var setting = {

async: {

enable: true,

url: "http://host/getNode.php",

otherParam: ["id", "1", "name", "test"]

}

};

进行异步加载时，将提交参数 id=1&name=test

2. 设置 JSON 格式的参数

var setting = {

async: {

enable: true,

url: "http://host/getNode.php",

otherParam: { "id":"1", "name":"test"}

}

};

进行异步加载时，将提交参数 id=1&name=test

### setting.async.type

概述[ 依赖 jquery.ztree.core 核心 js ]

Ajax 的 http 请求模式。[setting.async.enable = true 时生效]

默认值："post"

String 格式说明

type = "post" 表示异步加载采用 post 方法请求

type = "get" 表示异步加载采用 get 方法请求

对应于 jQuery ajax 中的 type 参数

setting 举例

1. 设置使用 get 方式请求数据

var setting = {

async: {

enable: true,

type: "get",

url: "http://host/getNode.php",

autoParam: ["id", "name"]

}

};

......

### setting.async.url

概述[ 依赖 jquery.ztree.core 核心 js ]

Ajax 获取数据的 URL 地址。[setting.async.enable = true 时生效]

默认值：""

String 格式说明

设置固定的异步加载 url 字符串，请注意地址的路径，确保页面能正常加载

url 内也可以带参数，这些参数就只能是通过 get 方式提交了，并且请注意进行转码

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

需要异步加载子节点的的父节点 JSON 数据对象

针对根进行异步加载时，treeNode = null

**返回值**String

返回值同 String 格式的数据

setting & function 举例

1. 设置异步获取节点的 URL 为 nodes.php

var setting = {

async: {

enable: true,

url: "nodes.php",

autoParam: ["id", "name"]

}

};

......

2. 设置异步获取节点的 URL 为 function 动态获取

function getAsyncUrl(treeId, treeNode) {

return treeNode.isParent ? "nodes1.php" : "nodes2.php";

};

var setting = {

async: {

enable: true,

url: getAsyncUrl,

autoParam: ["id", "name"]

}

};

......

## setting.callback

### setting.callback.beforeAsync

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获异步加载之前的事件回调函数，zTree 根据返回值确定是否允许进行异步加载

默认值：null

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

进行异步加载的父节点 JSON 数据对象

针对根进行异步加载时，treeNode = null

**返回值**Boolean

返回值是 true / false

如果返回 false，zTree 将不进行异步加载，也无法触发 onAsyncSuccess / onAsyncError 事件回调函数

setting & function 举例

1. 禁止 id 为 1 的父节点进行异步加载操作

function zTreeBeforeAsync(treeId, treeNode) {

return (treeNode.id !== 1);

};

var setting = {

callback: {

beforeAsync: zTreeBeforeAsync

}

};

......

### setting.callback.beforeCheck

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

用于捕获 勾选 或 取消勾选 之前的事件回调函数，并且根据返回值确定是否允许 勾选 或 取消勾选

默认值：null

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

进行 勾选 或 取消勾选 的节点 JSON 数据对象

**返回值**Boolean

返回值是 true / false

如果返回 false，将不会改变勾选状态，并且无法触发 onCheck 事件回调函数

setting & function 举例

1. 禁止所有勾选操作，保持初始化的勾选状态

function zTreeBeforeCheck(treeId, treeNode) {

return false;

};

var setting = {

callback: {

beforeCheck: zTreeBeforeCheck

}

};

......

### setting.callback.beforeClick

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获单击节点之前的事件回调函数，并且根据返回值确定是否允许单击操作

默认值：null

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

被单击的节点 JSON 数据对象

**clickFlag**Number

节点被点击后的选中操作类型，详细看下表

| **clickFlag** | **selectedMulti** | **autoCancelSelected && event.ctrlKey** | **isSelected** | **选中操作** |
| --- | --- | --- | --- | --- |
| 1 | true | false | false | 普通选中 |
| 1 | true | false | true | 普通选中 |
| 2 | true | true | false | 追加选中 |
| 0 | true | true | true | 取消选中 |
| 1 | false | false | false | 普通选中 |
| 1 | false | false | true | 普通选中 |
| 1 | false | true | false | 普通选中 |
| 0 | false | true | true | 取消选中 |

**返回值**Boolean

返回值是 true / false

如果返回 false，zTree 将不会选中节点，也无法触发 onClick 事件回调函数

setting & function 举例

1. 禁止节点被选中

function zTreeBeforeClick(treeId, treeNode, clickFlag) {

return (treeNode.id !== 1);

};

var setting = {

callback: {

beforeClick: zTreeBeforeClick

}

};

......

### setting.callback.beforeCollapse

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获父节点折叠之前的事件回调函数，并且根据返回值确定是否允许折叠操作

默认值：null

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

要折叠的父节点 JSON 数据对象

**返回值**Boolean

返回值是 true / false

如果返回 false，zTree 将不会折叠节点，也无法触发 onCollapse 事件回调函数

setting & function 举例

1. 禁止所有已展开的父节点折叠

function zTreeBeforeCollapse(treeId, treeNode) {

return false;

};

var setting = {

callback: {

beforeCollapse: zTreeBeforeCollapse

}

};

......

### setting.callback.beforeDblClick

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获 zTree 上鼠标双击之前的事件回调函数，并且根据返回值确定触发 onDblClick 事件回调函数

默认值：null

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

鼠标双击时所在节点的 JSON 数据对象

如果不在节点上，则返回 null

**返回值**Boolean

返回值是 true / false

如果返回 false，将仅仅无法触发 onDblClick 事件回调函数，对其他操作无任何影响

此事件回调函数对双击节点展开功能无任何影响，如果需要设置请参考 setting.view.dblClickExpand 属性

setting & function 举例

1. 禁止 onDblClick 事件

function zTreeBeforeDblClick(treeId, treeNode) {

return false;

};

var setting = {

callback: {

beforeDblClick: zTreeBeforeDblClick

}

};

......

### setting.callback.beforeDrag

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

用于捕获节点被拖拽之前的事件回调函数，并且根据返回值确定是否允许开启拖拽操作

默认值：null

Function 参数说明

**treeId**String

被拖拽的节点 treeNodes 所在 zTree 的 **treeId**，便于用户操控

**treeNodes**Array(JSON)

要被拖拽的节点 JSON 数据集合

v3.x 允许多个同级节点同时被拖拽，因此将此参数修改为 Array(JSON)

如果拖拽时多个被选择的节点不是同级关系，则只能拖拽鼠标当前所在位置的节点

**返回值**Boolean

返回值是 true / false

如果返回 false，zTree 将终止拖拽，也无法触发 onDrag / beforeDrop / onDrop 事件回调函数

setting & function 举例

1. 禁止全部拖拽操作

function zTreeBeforeDrag(treeId, treeNodes) {

return false;

};

var setting = {

edit: {

enable: true

},

callback: {

beforeDrag: zTreeBeforeDrag

}

};

......

### setting.callback.beforeDragOpen

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

用于捕获拖拽节点移动到折叠状态的父节点后，即将自动展开该父节点之前的事件回调函数，并且根据返回值确定是否允许自动展开操作

默认值：null

Function 参数说明

**treeId**String

需要被展开的父节点 treeNode 所在 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

要被自动展开的父节点 JSON 数据对象

**返回值**Boolean

返回值是 true / false

如果返回 false，zTree 将无法进行自动展开操作

setting & function 举例

1. 禁止全部拖拽时的自动展开操作

function zTreeBeforeDragOpen(treeId, treeNode) {

return false;

};

var setting = {

edit: {

enable: true

},

callback: {

beforeDragOpen: zTreeBeforeDragOpen

}

};

......

### setting.callback.beforeDrop

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

用于捕获节点拖拽操作结束之前的事件回调函数，并且根据返回值确定是否允许此拖拽操作

默认值：null

如未拖拽到有效位置，则不触发此回调函数，直接将节点恢复原位置

Function 参数说明

**treeId**String

目标节点 targetNode 所在 zTree 的 **treeId**，便于用户操控

**treeNodes**Array(JSON)

被拖拽的节点 JSON 数据集合

无论拖拽操作为 复制 还是 移动，treeNodes 都是当前被拖拽节点的数据集合。

**targetNode**JSON

treeNodes 被拖拽放开的目标节点 JSON 数据对象。

如果拖拽成为根节点，则 targetNode = null

**moveType**String

指定移动到目标节点的相对位置

"inner"：成为子节点，"prev"：成为同级前一个节点，"next"：成为同级后一个节点

**isCopy**Boolean

拖拽节点操作是 复制 或 移动

true：复制；false：移动

**返回值**Boolean

返回值是 true / false

如果返回 false，zTree 将恢复被拖拽的节点，也无法触发 onDrop 事件回调函数

setting & function 举例

1. 禁止将节点拖拽成为根节点

function zTreeBeforeDrop(treeId, treeNodes, targetNode, moveType) {

return !(targetNode == null || (moveType != "inner" && !targetNode.parentTId));

};

var setting = {

edit: {

enable: true

},

callback: {

beforeDrop: zTreeBeforeDrop

}

};

......

### setting.callback.beforeEditName

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

用于捕获节点编辑按钮的 click 事件，并且根据返回值确定是否允许进入名称编辑状态

此事件回调函数最主要是用于捕获编辑按钮的点击事件，然后触发自定义的编辑界面操作。

默认值：null

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

将要进入编辑名称状态的节点 JSON 数据对象

**返回值**Boolean

返回值是 true / false

如果返回 false，节点将无法进入 zTree 默认的编辑名称状态

setting & function 举例

1. 禁止修改父节点的名称

function zTreeBeforeEditName(treeId, treeNode) {

return !treeNode.isParent;

}

var setting = {

edit: {

enable: true

},

callback: {

beforeEditName: zTreeBeforeEditName

}

};

......

### setting.callback.beforeExpand

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获父节点展开之前的事件回调函数，并且根据返回值确定是否允许展开操作

默认值：null

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

要展开的父节点 JSON 数据对象

**返回值**Boolean

返回值是 true / false

如果返回 false，zTree 将不会展开节点，也无法触发 onExpand 事件回调函数

setting & function 举例

1. 禁止所有已折叠的父节点展开

function zTreeBeforeExpand(treeId, treeNode) {

return false;

};

var setting = {

callback: {

beforeExpand: zTreeBeforeExpand

}

};

......

function zTreeBeforeMouseDown(treeId, treeNode) {

return false;

};

var setting = {

callback: {

beforeMouseDown: zTreeBeforeMouseDown

}

};

......

### setting.callback.beforeMouseDown

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获 zTree 上鼠标按键按下之前的事件回调函数，并且根据返回值确定触发 onMouseDown 事件回调函数

默认值：null

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

鼠标按键按下时所在节点的 JSON 数据对象

如果不在节点上，则返回 null

**返回值**Boolean

返回值是 true / false

如果返回 false，将仅仅无法触发 onMouseDown 事件回调函数，对其他操作无任何影响

setting & function 举例

1. 禁止 onMouseDown 事件

function zTreeBeforeMouseDown(treeId, treeNode) {

return false;

};

var setting = {

callback: {

beforeMouseDown: zTreeBeforeMouseDown

}

};

......

### setting.callback.beforeMouseUp

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获 zTree 上鼠标按键松开之前的事件回调函数，并且根据返回值确定触发 onMouseUp 事件回调函数

默认值：null

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

鼠标按键松开时所在节点的 JSON 数据对象

如果不在节点上，则返回 null

**返回值**Boolean

返回值是 true / false

如果返回 false，将仅仅无法触发 onMouseUp 事件回调函数，对其他操作无任何影响

setting & function 举例

1. 禁止 onMouseUp 事件

function zTreeBeforeMouseUp(treeId, treeNode) {

return false;

};

var setting = {

callback: {

beforeMouseUp: zTreeBeforeMouseUp

}

};

......

### setting.callback.beforeRemove

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

用于捕获节点被删除之前的事件回调函数，并且根据返回值确定是否允许删除操作

默认值：null

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

将要删除的节点 JSON 数据对象

**返回值**Boolean

返回值是 true / false

如果返回 false，zTree 将不删除节点，也无法触发 onRemove 事件回调函数

setting & function 举例

1. 禁止全部删除操作

function zTreeBeforeRemove(treeId, treeNode) {

return false;

}

var setting = {

edit: {

enable: true

},

callback: {

beforeRemove: zTreeBeforeRemove

}

};

......

### setting.callback.beforeRename

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

用于捕获节点编辑名称结束（Input 失去焦点 或 按下 Enter 键）之后，更新节点名称数据之前的事件回调函数，并且根据返回值确定是否允许更改名称的操作

节点进入编辑名称状态后，按 ESC 键可以放弃当前修改，恢复原名称，取消编辑名称状态

默认值：null

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

将要更改名称的节点 JSON 数据对象

**newName**String

修改后的新名称

**返回值**Boolean

返回值是 true / false

如果返回 false，zTree 将保持名称编辑状态，无法触发 onRename 事件回调函数，并且会导致屏蔽其它事件，直到修改名称使得 beforeRename 返回 true

如果返回 false，不会让 input 输入框获取焦点，避免由于警告信息而导致反复触发 beforeRename。 请在关闭提示警告信息后，利用 editName 方法让 input 重新获取焦点。

setting & function 举例

1. 禁止修改的名称的长度小于 5

function zTreeBeforeRename(treeId, treeNode, newName) {

return newName.length > 5;

}

var setting = {

edit: {

enable: true

},

callback: {

beforeRename: zTreeBeforeRename

}

};

......

### setting.callback.beforeRightClick

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获 zTree 上鼠标右键点击之前的事件回调函数，并且根据返回值确定触发 onRightClick 事件回调函数

默认值：null

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

鼠标右键点击时所在节点的 JSON 数据对象

如果不在节点上，则返回 null

**返回值**Boolean

返回值是 true / false

如果返回 false，将仅仅无法触发 onRightClick 事件回调函数，对其他操作无任何影响

setting & function 举例

1. 禁止 onRightClick 事件

function zTreeBeforeRightClick(treeId, treeNode) {

return false;

};

var setting = {

callback: {

beforeRightClick: zTreeBeforeRightClick

}

};

......

### setting.callback.onAsyncError

Function(event, treeId, treeNode, XMLHttpRequest, textStatus, errorThrown)

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获异步加载出现异常错误的事件回调函数

如果设置了 setting.callback.beforeAsync 方法，且返回 false，将无法触发 onAsyncSuccess / onAsyncError 事件回调函数。

默认值：null

Function 参数说明

**event**js event 对象

标准的 js event 对象

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

进行异步加载的父节点 JSON 数据对象

针对根进行异步加载时，treeNode = null

**XMLHttpRequest**String

标准 XMLHttpRequest 对象，请参考 JQuery API 文档。

**textStatus**String

请求状态：success，error，请参考 JQuery API 文档。

**errorThrown**String

errorThrown 只有当异常发生时才会被传递，请参考 JQuery API 文档。

setting & function 举例

1. 异步加载出现异常后，弹出错误信息

function zTreeOnAsyncError(event, treeId, treeNode, XMLHttpRequest, textStatus, errorThrown) {

alert(XMLHttpRequest);

};

var setting = {

callback: {

onAsyncError: zTreeOnAsyncError

}

};

......

### setting.callback.onAsyncSuccess

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获异步加载正常结束的事件回调函数

如果设置了 setting.callback.beforeAsync 方法，且返回 false，将无法触发 onAsyncSuccess / onAsyncError 事件回调函数。

默认值：null

Function 参数说明

**event**js event 对象

标准的 js event 对象

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

进行异步加载的父节点 JSON 数据对象

针对根进行异步加载时，treeNode = null

**msg**String / Object

异步获取的节点数据字符串，主要便于用户调试使用。

实际数据类型会受 setting.async.dataType 的设置影响，请参考 JQuery API 文档。

setting & function 举例

1. 异步加载成功后，弹出提示信息

function zTreeOnAsyncSuccess(event, treeId, treeNode, msg) {

alert(msg);

};

var setting = {

callback: {

onAsyncSuccess: zTreeOnAsyncSuccess

}

};

......

### setting.callback.onCheck

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

用于捕获 checkbox / radio 被勾选 或 取消勾选的事件回调函数

如果设置了 setting.callback.beforeCheck 方法，且返回 false，将无法触发 onCheck 事件回调函数。

默认值：null

Function 参数说明

**event**js event 对象

标准的 js event 对象

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

被勾选 或 取消勾选的节点 JSON 数据对象

setting & function 举例

1. 每次点击 checkbox 或 radio 后， 弹出该节点的 tId、name 以及当前勾选状态的信息

function zTreeOnCheck(event, treeId, treeNode) {

alert(treeNode.tId + ", " + treeNode.name + "," + treeNode.checked);

};

var setting = {

callback: {

onCheck: zTreeOnCheck

}

};

......

### setting.callback.onClick

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获节点被点击的事件回调函数

如果设置了 setting.callback.beforeClick 方法，且返回 false，将无法触发 onClick 事件回调函数。

默认值：null

Function 参数说明

**event**js event 对象

标准的 js event 对象

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

被点击的节点 JSON 数据对象

**clickFlag**Number

节点被点击后的选中操作类型，详细看下表

| **clickFlag** | **selectedMulti** | **autoCancelSelected && event.ctrlKey** | **isSelected** | **选中操作** |
| --- | --- | --- | --- | --- |
| 1 | true | false | false | 普通选中 |
| 1 | true | false | true | 普通选中 |
| 2 | true | true | false | 追加选中 |
| 0 | true | true | true | 取消选中 |
| 1 | false | false | false | 普通选中 |
| 1 | false | false | true | 普通选中 |
| 1 | false | true | false | 普通选中 |
| 0 | false | true | true | 取消选中 |

setting & function 举例

1. 每次点击节点后， 弹出该节点的 tId、name 的信息

function zTreeOnClick(event, treeId, treeNode) {

alert(treeNode.tId + ", " + treeNode.name);

};

var setting = {

callback: {

onClick: zTreeOnClick

}

};

......

### setting.callback.onCollapse

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获节点被折叠的事件回调函数

如果设置了 setting.callback.beforeCollapse 方法，且返回 false，将无法触发 onCollapse 事件回调函数。

默认值：null

Function 参数说明

**event**js event 对象

标准的 js event 对象

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

被折叠的节点 JSON 数据对象

setting & function 举例

1. 每次折叠节点后， 弹出该节点的 tId、name 的信息

function zTreeOnCollapse(event, treeId, treeNode) {

alert(treeNode.tId + ", " + treeNode.name);

};

var setting = {

callback: {

onCollapse: zTreeOnCollapse

}

};

......

### setting.callback.onDblClick

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获 zTree 上鼠标双击之后的事件回调函数

如果设置了 setting.callback.beforeDblClick 方法，且返回 false，将无法触发 onDblClick 事件回调函数。

默认值：null

Function 参数说明

**event**js event 对象

标准的 js event 对象

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

鼠标双击时所在节点的 JSON 数据对象

如果不在节点上，则返回 null

setting & function 举例

1. 每次鼠标双击后， 弹出鼠标所在节点的 tId、name 的信息

function zTreeOnDblClick(event, treeId, treeNode) {

alert(treeNode ? treeNode.tId + ", " + treeNode.name : "isRoot");

};

var setting = {

callback: {

onDblClick: zTreeOnDblClick

}

};

......

### setting.callback.onDrag

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

用于捕获节点被拖拽的事件回调函数

如果设置了 setting.callback.beforeDrag 方法，且返回 false，将无法触发 onDrag 事件回调函数。

默认值：null

Function 参数说明

**event**js event 对象

标准的 js event 对象

**treeId**String

被拖拽的节点 treeNodes 所在 zTree 的 **treeId**，便于用户操控

**treeNodes**Array(JSON)

要被拖拽的节点 JSON 数据集合

setting & function 举例

1. 每次开始进行拖拽节点后， 弹出被拖拽节点的个数信息

function zTreeOnDrag(event, treeId, treeNodes) {

alert(treeNodes.length);

};

var setting = {

callback: {

onDrag: zTreeOnDrag

}

};

......

### setting.callback.onDrop

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

用于捕获节点拖拽操作结束的事件回调函数

如果设置了 setting.callback.beforeDrop 方法，且返回 false，将无法触发 onDrop 事件回调函数。

默认值：null

Function 参数说明

**event**js event 对象

标准的 js event 对象

**treeId**String

目标节点 targetNode 所在 zTree 的 **treeId**，便于用户操控

**treeNodes**Array(JSON)

被拖拽的节点 JSON 数据集合

如果拖拽操作为 移动，treeNodes 是当前被拖拽节点的数据集合。

如果拖拽操作为 复制，treeNodes 是复制后 clone 得到的新节点数据。

**targetNode**JSON

成为 treeNodes 拖拽结束的目标节点 JSON 数据对象。

如果拖拽成为根节点，则 targetNode = null

**moveType**String

指定移动到目标节点的相对位置

"inner"：成为子节点，"prev"：成为同级前一个节点，"next"：成为同级后一个节点

如果 moveType = null，表明拖拽无效

**isCopy**Boolean

拖拽节点操作是 复制 或 移动

true：复制；false：移动

setting & function 举例

1. 每次拖拽操作结束后， 弹出该被拖拽节点的个数以及目标节点的 tId、name 的信息

function zTreeOnDrop(event, treeId, treeNodes, targetNode, moveType) {

alert(treeNodes.length + "," + (targetNode ? (targetNode.tId + ", " + targetNode.name) : "isRoot" ));

};

var setting = {

callback: {

onDrop: zTreeOnDrop

}

};

......

### setting.callback.onExpand

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获节点被展开的事件回调函数

如果设置了 setting.callback.beforeExpand 方法，且返回 false，将无法触发 onExpand 事件回调函数。

默认值：null

Function 参数说明

**event**js event 对象

标准的 js event 对象

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

被展开的节点 JSON 数据对象

setting & function 举例

1. 每次展开节点后， 弹出该节点的 tId、name 的信息

function zTreeOnExpand(event, treeId, treeNode) {

alert(treeNode.tId + ", " + treeNode.name);

};

var setting = {

callback: {

onExpand: zTreeOnExpand

}

};

......

### setting.callback.onMouseDown

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获 zTree 上鼠标按键按下后的事件回调函数

如果设置了 setting.callback.beforeMouseDown 方法，且返回 false，将无法触发 onMouseDown 事件回调函数。

默认值：null

Function 参数说明

**event**js event 对象

标准的 js event 对象

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

鼠标按键按下时所在节点的 JSON 数据对象

如果不在节点上，则返回 null

setting & function 举例

1. 每次鼠标按键按下后， 弹出鼠标所在节点的 tId、name 的信息

function zTreeOnMouseDown(event, treeId, treeNode) {

alert(treeNode ? treeNode.tId + ", " + treeNode.name : "isRoot");

};

var setting = {

callback: {

onMouseDown: zTreeOnMouseDown

}

};

......

### setting.callback.onMouseUp

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获 zTree 上鼠标按键松开后的事件回调函数

如果设置了 setting.callback.beforeMouseUp 方法，且返回 false，将无法触发 onMouseUp 事件回调函数。

默认值：null

Function 参数说明

**event**js event 对象

标准的 js event 对象

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

鼠标按键松开时所在节点的 JSON 数据对象

如果不在节点上，则返回 null

setting & function 举例

1. 每次鼠标按键松开后， 弹出鼠标所在节点的 tId、name 的信息

function zTreeOnMouseUp(event, treeId, treeNode) {

alert(treeNode ? treeNode.tId + ", " + treeNode.name : "isRoot");

};

var setting = {

callback: {

onMouseUp: zTreeOnMouseUp

}

};

......

### setting.callback.onNodeCreated

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获节点生成 DOM 后的事件回调函数

v3.x 采用了延迟加载技术，因此对于父节点未展开的子节点来说，初始化后是不会触发此回调函数，直到其父节点被展开

大数据量的节点加载请注意：不设置 onNodeCreated，可以提升一部分初始化性能

默认值：null

Function 参数说明

**event**js event 对象

标准的 js event 对象

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

生成 DOM 完毕的节点的 JSON 数据对象

setting & function 举例

1. 创建节点 DOM 后， 弹出该节点的 tId、name 的信息

function zTreeOnNodeCreated(event, treeId, treeNode) {

alert(treeNode.tId + ", " + treeNode.name);

};

var setting = {

callback: {

onNodeCreated: zTreeOnNodeCreated

}

};

......

### setting.callback.onRemove

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

用于捕获删除节点之后的事件回调函数。

如果用户设置了 beforeRemove 回调函数，并返回 false，将无法触发 onRemove 事件回调函数。

默认值：null

Function 参数说明

**event**js event 对象

标准的 js event 对象

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

将要删除的节点 JSON 数据对象

setting & function 举例

1. 删除节点时，弹出被删除的节点的 tId 以及 name 信息

function zTreeOnRemove(event, treeId, treeNode) {

alert(treeNode.tId + ", " + treeNode.name);

}

var setting = {

callback: {

onRemove: zTreeOnRemove

}

};

......

### setting.callback.onRename

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

用于捕获节点编辑名称结束之后的事件回调函数。

1、节点进入编辑名称状态，并且修改节点名称后触发此回调函数。如果用户设置了 beforeRename 回调函数，并返回 false，将无法触发 onRename 事件回调函数。

2、如果通过直接修改 treeNode 的数据，并且利用 updateNode 方法更新，是不会触发此回调函数的。

默认值：null

Function 参数说明

**event**js event 对象

标准的 js event 对象

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

被修改名称的节点 JSON 数据对象

setting & function 举例

1. 修改名称后，弹出被修改名称的节点的 tId 以及 name 信息

function zTreeOnRename(event, treeId, treeNode) {

alert(treeNode.tId + ", " + treeNode.name);

}

var setting = {

callback: {

onRename: zTreeOnRename

}

};

......

### setting.callback.onRightClick

概述[ 依赖 jquery.ztree.core 核心 js ]

用于捕获 zTree 上鼠标右键点击之后的事件回调函数

1、如果设置了 setting.callback.beforeRightClick 方法，且返回 false，将无法触发 onRightClick 事件回调函数。

2、只要将 function 的引用赋给 onRightClick 属性，则右键点击 zTree 时，将屏蔽浏览器的右键菜单。

默认值：null

Function 参数说明

**event**js event 对象

标准的 js event 对象

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

鼠标右键点击时所在节点的 JSON 数据对象

如果不在节点上，则返回 null

setting & function 举例

1. 每次鼠标右键点击后， 弹出鼠标所在节点的 tId、name 的信息

function zTreeOnRightClick(event, treeId, treeNode) {

alert(treeNode ? treeNode.tId + ", " + treeNode.name : "isRoot");

};

var setting = {

callback: {

onRightClick: zTreeOnRightClick

}

};

......

## setting.check

### setting.check.autoCheckTrigger

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

设置自动关联勾选时是否触发 beforeCheck / onCheck 事件回调函数。[setting.check.enable = true 且 setting.check.chkStyle = "checkbox" 时生效]

1、如果设置 setting.check.chkboxType = { "Y": "", "N": "" }，将不会有任何自动关联勾选的操作。

2、如果开启触发，对于节点较多的树将会影响性能，因为所有被联动勾选的操作都会触发事件回调函数，请根据需要决定是否使用此功能。

默认值: false

Boolean 格式说明

true / false 分别表示 触发 / 不触发 事件回调函数

setting 举例

1. 需要触发自动关联勾选操作

var setting = {

check: {

enable: true,

autoCheckTrigger: true

}

};

......

### setting.check.chkboxType

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

勾选 checkbox 对于父子节点的关联关系。[setting.check.enable = true 且 setting.check.chkStyle = "checkbox" 时生效]

默认值：{ "Y": "ps", "N": "ps" }

JSON 格式说明

Y 属性定义 checkbox 被勾选后的情况；  
N 属性定义 checkbox 取消勾选后的情况；  
"p" 表示操作会影响父级节点；  
"s" 表示操作会影响子级节点。

请注意大小写，不要改变

setting 举例

1. checkbox 勾选操作，只影响父级节点；取消勾选操作，只影响子级节点

var setting = {

check: {

enable: true,

chkStyle: "checkbox",

chkboxType: { "Y": "p", "N": "s" }

}

};

......

### setting.check.chkStyle

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

勾选框类型(checkbox 或 radio）[setting.check.enable = true 时生效]

默认值："checkbox"

String 格式说明

chkStyle = "checkbox" 时，显示 checkbox 选择框，setting.check.chkboxType 属性有效。  
chkStyle = "radio" 时，显示 radio 选择框， setting.check.radioType 属性有效。

请注意大小写，不要改变

checkbox 状态说明

未勾选；如果是父节点，则无子节点被勾选。鼠标移到该节点上显示为：

未勾选；（只有父节点存在此状态）存在被勾选的子节点。鼠标移到该节点上显示为：

被勾选；如果是父节点，则全部子节点都被勾选。鼠标移到该节点上显示为：

被勾选；（只有父节点存在此状态）且部分或无子节点被勾选。鼠标移到该节点上显示为：

radio 状态说明

未勾选；如果是父节点，则没有子节点被勾选。鼠标移到该节点上显示为：

未勾选；（只有父节点存在此状态）且存在被勾选的子节点。鼠标移到该节点上显示为：

被勾选；如果是父节点，则没有子节点被勾选。鼠标移到该节点上显示为：

被勾选；（只有父节点存在此状态）且存在被勾选的子节点。鼠标移到该节点上显示为：

setting 举例

1. 设置选择框为 radio

var setting = {

check: {

enable: true,

chkStyle: "radio"

}

};

......

### setting.check.enable

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

设置 zTree 的节点上是否显示 checkbox / radio

默认值: false

Boolean 格式说明

true / false 分别表示 显示 / 不显示 复选框或单选框

setting 举例

1. 需要显示 checkbox

var setting = {

check: {

enable: true

}

};

......

### setting.check.nocheckInherit

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

当父节点设置 nocheck = true 时，设置子节点是否自动继承 nocheck = true 。[setting.check.enable = true 时生效]

1、只使用于初始化节点时，便于批量操作。 对于已存在的节点请利用 updateNode 方法单个节点设置。

默认值: false

Boolean 格式说明

true 表示 新加入子节点时，自动继承父节点 nocheck = true 的属性。

false 表示 新加入子节点时，不继承父节点 nocheck 的属性。

setting 举例

1. 需要子节点自动继承 nocheck = true

var setting = {

check: {

enable: true,

nocheckInherit: true

}

};

......

### setting.check.radioType

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

radio 的分组范围。[setting.check.enable = true 且 setting.check.chkStyle = "radio" 时生效]

默认值："level"

String 格式说明

radioType = "level" 时，在每一级节点范围内当做一个分组。  
radioType = "all" 时，在整棵树范围内当做一个分组。

请注意大小写，不要改变

setting 举例

1. 设置 radio 的判别规则为整棵树内

var setting = {

check: {

enable: true,

chkStyle: "radio",

radioType: "all"

}

};

......

## setting.data

### setting.data.keep.leaf

概述[ 依赖 jquery.ztree.core 核心 js ]

zTree 的节点叶子节点属性锁，是否始终保持 isParent = false

默认值：false

Boolean 格式说明

true / false 分别表示 锁定 / 不锁定 叶子节点属性

如果设置为 true，则所有 isParent = false 的节点，都无法添加子节点。

setting 举例

1. 需要锁定叶子节点状态

var setting = {

data: {

keep: {

leaf: true

}

}

};

......

### setting.data.keep.parent

概述[ 依赖 jquery.ztree.core 核心 js ]

zTree 的节点父节点属性锁，是否始终保持 isParent = true

默认值：false

Boolean 格式说明

true / false 分别表示 锁定 / 不锁定 父节点属性

如果设置为 true，则所有 isParent = true 的节点，即使该节点的子节点被全部删除或移走，依旧保持父节点状态。

setting 举例

1. 需要锁定父节点状态

var setting = {

data: {

keep: {

parent: true

}

}

};

......

### setting.data.key.checked

概述[ 依赖 jquery.ztree.excheck 扩展 js ]

zTree 节点数据中保存 check 状态的属性名称。

默认值："checked"

请勿与 zTree 节点数据的其他参数冲突，例如：checkedOld

setting 举例

1. 设置 zTree 显示节点时，将 treeNode 的 isChecked 属性当做节点名称

var setting = {

data: {

key: {

checked: "isChecked"

}

}

};

......

### setting.data.key.children

概述[ 依赖 jquery.ztree.core 核心 js ]

zTree 节点数据中保存子节点数据的属性名称。

默认值："children"

setting 举例

1. 设置 zTree 显示节点时，将 treeNode 的 nodes 属性当做节点名称

var setting = {

data: {

key: {

children: "nodes"

}

}

};

......

### setting.data.key.name

概述[ 依赖 jquery.ztree.core 核心 js ]

zTree 节点数据保存节点名称的属性名称。

默认值："name"

setting 举例

1. 设置 zTree 显示节点时，将 treeNode 的 ename 属性当做节点名称

var setting = {

data: {

key: {

name: "ename"

}

}

};

......

### setting.data.key.title

概述[ 依赖 jquery.ztree.core 核心 js ]

zTree 节点数据保存节点提示信息的属性名称。[setting.view.showTitle = true 时生效]

如果设置为 "" ，则自动与 setting.data.key.name 保持一致，避免用户反复设置

默认值：""

setting 举例

1. 设置 zTree 显示节点时，将 treeNode 的 fullName 属性当做节点名称

var setting = {

data: {

key: {

title: "fullName"

}

}

};

......

### setting.data.key.url

概述[ 依赖 jquery.ztree.core 核心 js ]

zTree 节点数据保存节点链接的目标 URL 的属性名称。

特殊用途：当后台数据只能生成 url 属性，又不想实现点击节点跳转的功能时，可以直接修改此属性为其他不存在的属性名称

默认值："url"

setting 举例

1. 设置 zTree 显示节点时，将 treeNode 的 xUrl 属性当做节点链接的目标 URL

var setting = {

data: {

key: {

url: "xUrl"

}

}

};

......

### setting.data.simpleData.enable

概述[ 依赖 jquery.ztree.core 核心 js ]

确定 zTree 初始化时的节点数据、异步加载时的节点数据、或 addNodes 方法中输入的 newNodes 数据是否采用简单数据模式 (Array)

不需要用户再把数据库中取出的 List 强行转换为复杂的 JSON 嵌套格式

默认值：false

Boolean 格式说明

true / false 分别表示 使用 / 不使用 简单数据模式

如果设置为 true，请务必设置 setting.data.simpleData 内的其他参数: idKey / pIdKey / rootPId，并且让数据满足父子关系。

setting 举例

1. 使用简单 Array 格式的数据

var setting = {

data: {

simpleData: {

enable: true,

idKey: "id",

pIdKey: "pId",

rootPId: 0,

}

}

};

var treeNodes = [

{"id":1, "pId":0, "name":"test1"},

{"id":11, "pId":1, "name":"test11"},

{"id":12, "pId":1, "name":"test12"},

{"id":111, "pId":11, "name":"test111"}

];

......

### setting.data.simpleData.idKey

概述[ 依赖 jquery.ztree.core 核心 js ]

节点数据中保存唯一标识的属性名称。[setting.data.simpleData.enable = true 时生效]

默认值："id"

setting 举例

1. 使用简单 Array 格式的数据

var setting = {

data: {

simpleData: {

enable: true,

idKey: "id",

pIdKey: "pId",

rootPId: 0,

}

}

};

var treeNodes = [

{"id":1, "pId":0, "name":"test1"},

{"id":11, "pId":1, "name":"test11"},

{"id":12, "pId":1, "name":"test12"},

{"id":111, "pId":11, "name":"test111"}

];

......

### setting.data.simpleData.pIdKey

概述[ 依赖 jquery.ztree.core 核心 js ]

节点数据中保存其父节点唯一标识的属性名称。[setting.data.simpleData.enable = true 时生效]

默认值："pId"

setting 举例

1. 使用简单 Array 格式的数据

var setting = {

data: {

simpleData: {

enable: true,

idKey: "id",

pIdKey: "pId",

rootPId: 0

}

}

};

var treeNodes = [

{"id":1, "pId":0, "name":"test1"},

{"id":11, "pId":1, "name":"test11"},

{"id":12, "pId":1, "name":"test12"},

{"id":111, "pId":11, "name":"test111"}

];

......

### setting.data.simpleData.rootPId

概述[ 依赖 jquery.ztree.core 核心 js ]

用于修正根节点父节点数据，即 pIdKey 指定的属性值。[setting.data.simpleData.enable = true 时生效]

默认值：null

setting 举例

1. 使用简单 Array 格式的数据

var setting = {

data: {

simpleData: {

enable: true,

idKey: "id",

pIdKey: "pId",

rootPId: 0

}

}

};

var treeNodes = [

{"id":1, "pId":0, "name":"test1"},

{"id":11, "pId":1, "name":"test11"},

{"id":12, "pId":1, "name":"test12"},

{"id":111, "pId":11, "name":"test111"}

];

......

## setting.edit

### setting.edit.drag.autoExpandTrigger

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

拖拽时父节点自动展开是否触发 onExpand 事件回调函数。[setting.edit.enable = true 时生效]

默认值：false

Boolean 格式说明

true / false 分别表示 触发 / 不触发 onExpand 事件回调函数。

setting 举例

1. 设置拖拽时父节点自动展开触发 onExpand 事件回调函数

var setting = {

edit: {

enable: true,

drag: {

autoExpandTrigger: true

}

}

};

......

### setting.edit.drag.isCopy

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

拖拽时, 设置是否允许复制节点。[setting.edit.enable = true 时生效]

默认值：true

规则说明

1、isCopy = true; isMove = true 时，拖拽节点按下 Ctrl 键表示 copy; 否则为 move

2、isCopy = true; isMove = false 时，所有拖拽操作都是 copy

3、isCopy = false; isMove = true 时，所有拖拽操作都是 move

4、isCopy = false; isMove = false 时，禁止拖拽操作

setting 举例

1. 设置所有拖拽操作都是 copy

var setting = {

edit: {

enable: true,

drag: {

isCopy: true,

isMove: false

}

}

};

......

### setting.edit.drag.isMove

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

拖拽时, 设置是否允许移动节点。[setting.edit.enable = true 时生效]

默认值：true

规则说明

1、isCopy = true; isMove = true 时，拖拽节点按下 Ctrl 键表示 copy; 否则为 move

2、isCopy = true; isMove = false 时，所有拖拽操作都是 copy

3、isCopy = false; isMove = true 时，所有拖拽操作都是 move

4、isCopy = false; isMove = false 时，禁止拖拽操作

setting 举例

1. 设置所有拖拽操作都是 move

var setting = {

edit: {

enable: true,

drag: {

isCopy: false,

isMove: true

}

}

};

......

### setting.edit.drag.prev

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

拖拽到目标节点时，设置是否允许移动到目标节点前面的操作。[setting.edit.enable = true 时生效]

拖拽目标是 根 的时候，不触发 prev 和 next，只会触发 inner

此功能主要作用是对拖拽进行适当限制（辅助箭头），需要结合 next、inner 一起使用，才能实现完整功能。

默认值：true

Boolean 格式说明

true / false 分别表示 允许 / 不允许 移动到目标节点前面

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控（多棵树拖拽时，是目标节点所在树的 treeId）

**treeNodes**Array(JSON)

被拖拽的节点 JSON 数据集合

**targetNode**JSON

拖拽时的目标节点 JSON 数据对象

**返回值**Boolean

返回值同 Boolean 格式的数据

setting & function 举例

1. 禁止拖拽到节点前面的操作

var setting = {

edit: {

enable: true,

drag: {

prev: false,

next: true,

inner: true

}

}

};

......

### setting.edit.drag.next

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

拖拽到目标节点时，设置是否允许移动到目标节点后面的操作。[setting.edit.enable = true 时生效]

拖拽目标是 根 的时候，不触发 prev 和 next，只会触发 inner

此功能主要作用是对拖拽进行适当限制（辅助箭头），需要结合 prev、inner 一起使用，才能实现完整功能。

默认值：true

Boolean 格式说明

true / false 分别表示 允许 / 不允许 移动到目标节点后面

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控（多棵树拖拽时，是目标节点所在树的 treeId）

**treeNodes**Array(JSON)

被拖拽的节点 JSON 数据集合

**targetNode**JSON

拖拽时的目标节点 JSON 数据对象

**返回值**Boolean

返回值同 Boolean 格式的数据

setting & function 举例

1. 禁止拖拽到节点后面的操作

var setting = {

edit: {

enable: true,

drag: {

prev: true,

next: false,

inner: true

}

}

};

......

### setting.edit.drag.inner

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

拖拽到目标节点时，设置是否允许成为目标节点的子节点。[setting.edit.enable = true 时生效]

拖拽目标是 根 的时候，不触发 prev 和 next，只会触发 inner

此功能主要作用是对拖拽进行适当限制（辅助箭头），需要结合 prev、next 一起使用，才能实现完整功能。

默认值：true

Boolean 格式说明

true / false 分别表示 允许 / 不允许 成为目标节点的子节点

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控（多棵树拖拽时，是目标节点所在树的 treeId）

**treeNodes**Array(JSON)

被拖拽的节点 JSON 数据集合

**targetNode**JSON

拖拽时的目标节点 JSON 数据对象

如果拖拽的节点要成为根节点，则 targetNode = null

**返回值**Boolean

返回值同 Boolean 格式的数据

setting & function 举例

1. 禁止拖拽成为目标节点的子节点

var setting = {

edit: {

enable: true,

drag: {

prev: true,

next: true,

inner: false

}

}

};

......

### setting.edit.drag.borderMax

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

拖拽节点成为根节点时的 Tree 内边界范围 (单位：px)。[setting.edit.enable = true 时生效]

默认值：10

请根据自己的需求适当调整此值

setting 举例

1. 更改拖拽操作节点成为根节点时的 Tree 内边界范围为20px

var setting = {

edit: {

enable: true,

drag: {

borderMax: 20

}

}

};

......

### setting.edit.drag.borderMin

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

拖拽节点成为根节点时的 Tree 外边界范围 (单位：px)。[setting.edit.enable = true 时生效]

默认值：-5

请根据自己的需求适当调整此值

setting 举例

1. 更改拖拽操作节点成为根节点时的 Tree 外边界范围为10px

var setting = {

edit: {

enable: true,

drag: {

borderMin: -10

}

}

};

......

### setting.edit.drag.minMoveSize

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

判定是否拖拽操作的最小位移值 (单位：px)。[setting.edit.enable = true 时生效]

根据自己的需求可适当调整此值，如果太小容易导致点击鼠标时误操作进行拖拽

默认值：5

setting 举例

1. 更改拖拽操作启动位移值为10px

var setting = {

edit: {

enable: true,

drag: {

minMoveSize: 10

}

}

};

......

### setting.edit.drag.maxShowNodeNum

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

拖拽多个兄弟节点时，浮动图层中显示的最大节点数。 多余的节点用...代替。[setting.edit.enable = true 时生效]

默认值：5

请根据自己的需求适当调整此值

setting 举例

1. 设置拖拽时最多可显示10个节点

var setting = {

edit: {

enable: true,

drag: {

maxShowNodeNum: 10

}

}

};

......

### setting.edit.drag.autoOpenTime

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

拖拽时父节点自动展开的延时间隔。 (单位：ms)[setting.edit.enable = true 时生效]

默认值：500

请根据自己的需求适当调整此值

setting 举例

1. 设置拖拽到父节点上立刻自动展开

var setting = {

edit: {

enable: true,

drag: {

autoOpenTime: 0

}

}

};

......

### setting.edit.editNameSelectAll

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

节点编辑名称 input 初次显示时,设置 txt 内容是否为全选状态。 [setting.edit.enable = true 时生效]

默认值: false

Boolean 格式说明

true 表示 全选状态

false 表示 不是全选状态，光标默认在最后

setting 举例

1. 设置节点编辑名称 input 初次显示时，txt内容为全选状态

var setting = {

edit: {

enable: true,

editNameSelectAll: true

}

};

......

### setting.edit.enable

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

设置 zTree 是否处于编辑状态

请在初始化之前设置，初始化后需要改变编辑状态请使用 zTreeObj.setEditable() 方法

默认值: false

Boolean 格式说明

true / false 分别表示 可以 / 不可以 编辑

编辑状态规则说明

1、点击节点时，不会打开 node.url 指定的 URL。  
2、全面支持 编辑 与 异步加载 状态共存。  
3、可以对节点进行拖拽，且支持多棵树之间进行拖拽。  
4、支持拖拽时 复制/移动 节点。（参考: setting.edit.drag.isCopy / setting.edit.drag.isMove）  
5、可以通过编辑按钮修改 name 属性。  
6、可以通过删除按钮删除节点。

请注意大小写，不要改变

setting 举例

1. 设置 zTree 进入编辑状态

var setting = {

edit: {

enable: true

}

};

......

### setting.edit.removeTitle

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

删除按钮的 Title 辅助信息。[setting.edit.enable = true & setting.edit.showRemoveBtn = true 时生效]

默认值："remove"

String 格式说明

鼠标移动到 删除按钮 上时，浏览器自动弹出的辅助信息内容，可根据用户需要自行修改

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

需要设置删除按钮 Title 信息的节点 JSON 数据对象

**返回值**String

返回值同 String 格式的数据

setting & function 举例

1. 设置删除按钮的 Title 辅助信息为: "删除节点"

var setting = {

edit: {

enable: true,

showRemoveBtn: true,

removeTitle: "删除节点"

}

};

......

2. 设置父节点删除按钮的 Title 辅助信息为: "删除父节点"

function setRemoveTitle(treeId, treeNode) {

return treeNode.isParent ? "删除父节点":"删除叶子节点";

}

var setting = {

edit: {

enable: true,

showRemoveBtn: true,

removeTitle: setRemoveTitle

}

};

......

### setting.edit.renameTitle

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

编辑名称按钮的 Title 辅助信息。[setting.edit.enable = true & setting.edit.showRenameBtn = true 时生效]

默认值："rename"

String 格式说明

设置鼠标移动到 编辑名称按钮 上时，浏览器自动弹出的辅助信息内容，可根据用户需要自行修改

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

需要设置编辑名称按钮 Title 信息的节点 JSON 数据对象

**返回值**String

返回值同 String 格式的数据

setting & function 举例

1. 设置编辑名称按钮的 Title 辅助信息为: "编辑节点名称"

var setting = {

edit: {

enable: true,

showRenameBtn: true,

renameTitle: "编辑节点名称"

}

};

......

2. 设置父节点编辑名称按钮的 Title 辅助信息为: "编辑父节点名称"

function setRenameTitle(treeId, treeNode) {

return treeNode.isParent ? "编辑父节点名称":"编辑叶子节点名称";

}

var setting = {

edit: {

enable: true,

showRenameBtn: true,

renameTitle: setRenameTitle

}

};

......

### setting.edit.showRemoveBtn

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

设置是否显示删除按钮。[setting.edit.enable = true 时生效]

当点击某节点的删除按钮时：

1、首先触发 setting.callback.beforeRemove 回调函数，用户可判定是否进行删除操作。

2、如果未设置 beforeRemove 或 beforeRemove 返回 true，则删除节点并触发 setting.callback.onRemove 回调函数。

默认值：true

Boolean 格式说明

true / false 分别表示 显示 / 隐藏 删除按钮

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

需要设置是否显示删除按钮的节点 JSON 数据对象

**返回值**Boolean

返回值同 Boolean 格式的数据

setting & function 举例

1. 不显示删除按钮

var setting = {

edit: {

enable: true,

showRemoveBtn: false

}

};

......

2. 设置所有的父节点不显示删除按钮

function setRemoveBtn(treeId, treeNode) {

return !treeNode.isParent;

}

var setting = {

edit: {

enable: true,

showRemoveBtn: setRemoveBtn

}

};

......

### setting.edit.showRenameBtn

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

设置是否显示编辑名称按钮。[setting.edit.enable = true 时生效]

当点击某节点的编辑名称按钮时：

1、进入节点编辑名称状态。

2、编辑名称完毕（Input 失去焦点 或 按下 Enter 键），会触发 setting.callback.beforeRename 回调函数，用户可根据自己的规则判定是否允许修改名称。

3、如果 beforeRename 返回 false，则继续保持编辑名称状态，直到名称符合规则位置 （按下 ESC 键可取消编辑名称状态，恢复原名称）。

4、如果未设置 beforeRename 或 beforeRename 返回 true，则结束节点编辑名称状态，更新节点名称，并触发 setting.callback.onRename 回调函数。

默认值：true

Boolean 格式说明

true / false 分别表示 显示 / 隐藏 编辑名称按钮

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

需要设置是否显示编辑名称按钮的节点 JSON 数据对象

**返回值**Boolean

返回值同 Boolean 格式的数据

setting & function 举例

1. 不显示编辑名称按钮

var setting = {

edit: {

enable: true,

showRenameBtn: false

}

};

......

2. 设置所有的父节点不显示编辑名称按钮

function setRenameBtn(treeId, treeNode) {

return !treeNode.isParent;

}

var setting = {

edit: {

enable: true,

showRenameBtn: setRenameBtn

}

};

......

## setting.view

### setting.view.addDiyDom

概述[ 依赖 jquery.ztree.core 核心 js ]

用于在节点上固定显示用户自定义控件

1. 大数据量的节点加载请注意：在 addDiyDom 中针对每个节点 查找 DOM 对象并且添加新 DOM 控件，肯定会影响初始化性能；如果不是必须使用，建议不使用此功能

2. 属于高级应用，使用时请确保对 zTree 比较了解。

默认值：null

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

需要显示自定义控件的节点 JSON 数据对象

setting & function 举例

1. 设置节点后面显示一个按钮

var setting = {

view: {

addDiyDom: addDiyDom

}

};

function addDiyDom(treeId, treeNode) {

var aObj = $("#" + treeNode.tId + "\_a");

if ($("#diyBtn\_"+treeNode.id).length>0) return;

var editStr = "<span id='diyBtn\_space\_" +treeNode.id+ "' > </span>"

+ "<button type='button' class='diyBtn1' id='diyBtn\_" + treeNode.id

+ "' title='"+treeNode.name+"' onfocus='this.blur();'></button>";

aObj.append(editStr);

var btn = $("#diyBtn\_"+treeNode.id);

if (btn) btn.bind("click", function(){alert("diy Button for " + treeNode.name);});

};

......

### setting.view.addHoverDom

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

用于当鼠标移动到节点上时，显示用户自定义控件，显示隐藏状态同 zTree 内部的编辑、删除按钮

请务必与 setting.view.removeHoverDom 同时使用；属于高级应用，使用时请确保对 zTree 比较了解。

默认值：null

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

需要显示自定义控件的节点 JSON 数据对象

setting & function 举例

1. 设置鼠标移到节点上，在后面显示一个按钮

var setting = {

view: {

addHoverDom: addHoverDom,

removeHoverDom: removeHoverDom,

......

}

};

function addHoverDom(treeId, treeNode) {

var aObj = $("#" + treeNode.tId + "\_a");

if ($("#diyBtn\_"+treeNode.id).length>0) return;

var editStr = "<span id='diyBtn\_space\_" +treeNode.id+ "' > </span>"

+ "<button type='button' class='diyBtn1' id='diyBtn\_" + treeNode.id

+ "' title='"+treeNode.name+"' onfocus='this.blur();'></button>";

aObj.append(editStr);

var btn = $("#diyBtn\_"+treeNode.id);

if (btn) btn.bind("click", function(){alert("diy Button for " + treeNode.name);});

};

function removeHoverDom(treeId, treeNode) {

$("#diyBtn\_"+treeNode.id).unbind().remove();

$("#diyBtn\_space\_" +treeNode.id).unbind().remove();

};

......

### setting.view.autoCancelSelected

概述[ 依赖 jquery.ztree.core 核心 js ]

点击节点时，按下 Ctrl 键是否允许取消选择操作。

如果不需要此功能，请设置为 false。

默认值: true

Boolean 格式说明

true / false 分别表示 支持 / 不支持 配合 Ctrl 键进行取消节点选择的操作

setting 举例

1. 禁止配合 Ctrl 键进行取消节点选择的操作

var setting = {

view: {

autoCancelSelected: false

}

};

......

### setting.view.dblClickExpand

概述[ 依赖 jquery.ztree.core 核心 js ]

双击节点时，是否自动展开父节点的标识

默认值: true

Boolean 格式说明

true / false 分别表示双击节点 切换 / 不切换 展开状态

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

需要设置是否双击切换展开状态的节点 JSON 数据对象

**返回值**Boolean

返回值同 Boolean 格式的数据

setting 举例

1. 取消默认双击展开父节点的功能

var setting = {

view: {

dblClickExpand: false

}

};

......

2. 设置 zTree 仅仅 level=0 的父节点取消双击展开的功能

function dblClickExpand(treeId, treeNode) {

return treeNode.level > 0;

};

var setting = {

view: {

dblClickExpand: dblClickExpand

}

};

......

### setting.view.expandSpeed

概述[ 依赖 jquery.ztree.core 核心 js ]

zTree 节点展开、折叠时的动画速度，设置方法同 JQuery 动画效果中 speed 参数。

IE6 下会自动关闭动画效果，以保证 zTree 的操作速度

默认值："fast"

String 格式说明

三种预定速度之一的字符串("slow", "normal", or "fast")

设置为 "" 时，不显示动画效果

Number 格式说明

表示动画时长的毫秒数值 (如：1000)

setting 举例

1. 设置为慢速显示动画效果

var setting = {

view: {

expandSpeed: "slow"

}

};

......

### setting.view.fontCss

概述[ 依赖 jquery.ztree.core 核心 js ]

个性化文字样式，只针对 zTree 在节点上显示的<A>对象。

默认值：{}

JSON 格式说明

JSON 格式为 JQuery css方法中的 JSON 对象格式，例如：{color:"#ff0011", background:"blue"}

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

需要设置自定义样式的节点 JSON 数据对象

**返回值**JSON

返回值同 JSON 格式的数据，例如：{color:"#ff0011", background:"blue"}

setting & function 举例

1. 不修改CSS，设置全部节点 name 显示为红色

var setting = {

view: {

fontCss : {color:"red"}

}

};

2. 设置 level=0 的节点 name 显示为红色

function setFontCss(treeId, treeNode) {

return treeNode.level == 0 ? {color:"red"} : {};

};

var setting = {

view: {

fontCss: setFontCss

}

};

### setting.view.nameIsHTML

概述[ 依赖 jquery.ztree.core 核心 js ]

设置 name 属性是否支持 HTML 脚本

如果允许 HTML 脚本，请根据自己的需求做校验，避免出现 js 注入等安全问题。

默认值: false

Boolean 格式说明

true / false 分别表示 支持 / 不支持 HTML 脚本

setting 举例

1. 设置 name 属性支持 HTML 脚本

var setting = {

view: {

nameIsHTML: true

}

};

var node = {"name":"<font color='red'>test</font>"};

......

### setting.view.removeHoverDom

概述[ 依赖 jquery.ztree.exedit 扩展 js ]

用于当鼠标移出节点时，隐藏用户自定义控件，显示隐藏状态同 zTree 内部的编辑、删除按钮

请务必与 addHoverDom 同时使用；属于高级应用，使用时请确保对 zTree 比较了解。

默认值：null

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

需要隐藏自定义控件的节点 JSON 数据对象

setting & function 举例

1. 设置鼠标移到节点上，在后面显示一个按钮

var setting = {

view: {

addHoverDom: addHoverDom,

removeHoverDom: removeHoverDom,

......

}

};

function addHoverDom(treeId, treeNode) {

var aObj = $("#" + treeNode.tId + "\_a");

if ($("#diyBtn\_"+treeNode.id).length>0) return;

var editStr = "<span id='diyBtn\_space\_" +treeNode.id+ "' > </span>"

+ "<button type='button' class='diyBtn1' id='diyBtn\_" + treeNode.id

+ "' title='"+treeNode.name+"' onfocus='this.blur();'></button>";

aObj.append(editStr);

var btn = $("#diyBtn\_"+treeNode.id);

if (btn) btn.bind("click", function(){alert("diy Button for " + treeNode.name);});

};

function removeHoverDom(treeId, treeNode) {

$("#diyBtn\_"+treeNode.id).unbind().remove();

$("#diyBtn\_space\_" +treeNode.id).unbind().remove();

};

......

### setting.view.selectedMulti

概述[ 依赖 jquery.ztree.core 核心 js ]

设置是否允许同时选中多个节点。

默认值: true

Boolean 格式说明

true / false 分别表示 支持 / 不支持 同时选中多个节点

1、设置为 true时，按下 Ctrl 键可以选中多个节点

2、设置为 true / false 都不影响按下 Ctrl 键可以让已选中的节点取消选中状态（ 取消选中状态可以参考 setting.view.autoCancelSelected ）

setting 举例

1. 禁止多点同时选中的功能

var setting = {

view: {

selectedMulti: false

}

};

......

### setting.view.showIcon

概述[ 依赖 jquery.ztree.core 核心 js ]

设置 zTree 是否显示节点的图标。

默认值：true

Boolean 格式说明

true / false 分别表示 显示 / 隐藏 图标

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

需要设置是否显示图标的节点 JSON 数据对象

**返回值**Boolean

返回值同 Boolean 格式的数据

setting & function 举例

1. 设置 zTree 不显示图标

var setting = {

view: {

showIcon: false

}

};

......

2. 设置 zTree 仅仅 level=2 的节点不显示图标

function showIconForTree(treeId, treeNode) {

return treeNode.level != 2;

};

var setting = {

view: {

showIcon: showIconForTree

}

};

......

### setting.view.showLine

概述[ 依赖 jquery.ztree.core 核心 js ]

设置 zTree 是否显示节点之间的连线。

默认值：true

Boolean 格式说明

true / false 分别表示 显示 / 不显示 连线

setting 举例

1. 设置 zTree 不显示节点之间的连线

var setting = {

view: {

showLine: false

}

};

......

### setting.view.showTitle

概述[ 依赖 jquery.ztree.core 核心 js ]

设置 zTree 是否显示节点的 title 提示信息(即节点 DOM 的 title 属性)。

请务必与 setting.data.key.title 同时使用。

默认值：true

Boolean 格式说明

true / false 分别表示 显示 / 隐藏 提示信息

如果 setting.view.showTitle = true & setting.data.key.title = ''，zTree 会自动使用 setting.data.key.name 指定的节点名称当做 title

Function 参数说明

**treeId**String

对应 zTree 的 **treeId**，便于用户操控

**treeNode**JSON

需要设置是否显示提示信息的节点 JSON 数据对象

**返回值**Boolean

返回值同 Boolean 格式的数据

setting & function 举例

1. 设置 zTree 不显示提示信息

var setting = {

view: {

showTitle: false

}

};

......

2. 设置 zTree 仅仅 level=2 的节点不显示提示信息

function showTitleForTree(treeId, treeNode) {

return treeNode.level != 2;

};

var setting = {

view: {

showTitle: showTitleForTree

}

};

......

# zTree v3.x 常见问题

## zTree 对 jQuery 的版本有要求吗?

* 基本来说没有要求，zTree 在开发中使用 jQuery 1.4.4；目前简单测试过 jQuery 1.3 ~ 1.6.1 应该都能正常使用，如果发现和 jQuery 的兼容问题，还请及时 Email: [hunter.z@263.net](mailto:hunter.z@263.net) 通知我。
* 如果希望有更好的动画效果，更高的运行效率，可以选择 jQuery 1.6.x；如果希望能有减少 js 代码，使用基本功能，那么可以选择 mini 版较低版本的jQuery，大家就各取所需吧。
* zTree 中提供的是未经压缩的 jQuery 1.4.4，主要是开发时便于调试。  
  \* 如需自行下载 jQuery，请访问：<http://docs.jquery.com/Downloading_jQuery>

## zTree v3.x 入门指南

考虑到还是会有第一次使用 zTree 的朋友，因此入门指南不能少，至少让大家快速明白如何开始制作一棵属于自己的树。

步骤 1、文件准备

将需要使用的 **zTree v3.x** 相关的 js、css、img 文件分别放置到相应目录，并且保证相对路径正确

步骤 2、编写 html 页面

按照以下代码，制作 html 页面，访问试试看吧，注意：  
1) "<!DOCTYPE html>" 是必需的！  
2) zTree 的容器 className 别忘了设置为 "ztree"  
3) 入门成功后，就可以按照顺序去看 Demo 了，直接看看源码，应该能看懂的

## 关于异步加载

1、什么是异步加载？

这个问题与 zTree 无关，但必须要解释一下，因为很多初学者都在这个问题上犯糊涂。

准确的说应该是：“什么是 AJAX ?”-- AJAX 即“Asynchronous JavaScript and XML”（异步JavaScript和XML)；AJAX 是一种技术，可以在不影响当前页面的情况下与后台服务端进行通讯，获取相应的数据。更多的解释大家可以去网上搜索一下。

2、异步加载的特性

请注意最前面的两个字 “异步”，与他类似还有一个词 “同步”，顾名思义“同步”就是前面的语句不执行完就不会执行后面的语句，“异步”就不一样了，发起异步请求后，不管是否得到结果，都不影响当前代码的后续执行。

经常有朋友问：“异步加载模式下，初始化 zTree 以后为什么无法得到全部节点数据？”、“异步加载模式下，初始化 zTree 以后为什么无法使用 expandAll 方法将全部节点展开 ？”等等类似的问题，看了前面的解释，应该明白了吧？当你发起异步请求后，代码会立刻去执行 getNodes 方法 或 expandAll 方法，这时候异步加载的数据还未收到，当然操作就失效了。所以你会发现当你先 alert 一下，会发现又一切正常了-- 就是这个原因。

所以，对于异步加载模式下，需要在获取节点后处理的事情，尽量利用 onAsyncSuccess / onAsyncError 事件回调函数去处理，这样才能保证异步加载正常完毕后执行你需要的操作。

利用 setTimeout 举个异步的例子，对照着执行看看吧：

//No.1

var a = 1;

a++;

var b = a;

alert(b);

//No.2

var a = 1;

setTimeout(function(){a++;}, 500);

var b = a;

alert(b);

## css 样式异常怎么办?

产生样式冲突的可能性很多，在这次 v3.0 的制作过程中，又更加深入了这方面的了解，大概总结如下：

1、异常原因：

页面上自定义的 css 与 zTree 的 css 产生冲突

页面上使用其他插件的 css 与 zTree 的 css 产生冲突

修改 zTree 的 css 错误影响了 zTree

2、解决方案：  
由于 css 冲突的可能性太多，与其说是解决方案，不如说是处理建议。

A、避免针对 id 进行样式设定，尽量采用 class 设置。 因为针对 id 设定的样式等级太高，zTree 的样式肯定会被覆盖。

B、zTree 默认的样式不可能将全部 css 定义都重新设置一遍，所以难免有遗漏，如果被遗漏的样式影响了效果，那么请在 zTreeStyle 内增加上对应的设置内容，而且一旦出现这种情况，也希望你能通知我，谢谢。

C、如果以上都不能解决，那么再看看是否有命名规则冲突，导致了样式冲突

D、最后再看看是否自己修改的 zTree 样式出现了错误

E、以上答案均无效，那么就请与我联系吧。

## 如何将 节点数据提交给 Server 端

你可以利用 Form 或 Ajax 提交数据。

把 JSON 数据转化为参数是比较麻烦的。在今后，我会逐渐提供一些工具包便于大家生成各种格式的数据。

制作提交数据的基本方法：拼字符串

拼接成什么格式的字符串？ 是 url? xml? 还是 json？ 这就要根据后台解析数据的方法来决定了。

## 请区分 JSON 字符串 / 对象

* 有不少朋友由于没有分清楚这两者的关系，导致使用 zTree 出现了错误，这里专门讲解一下：
* （1）JSON 对象是满足 JSON 数据格式的 JS 对象  
  例如： {name:"abc"}  
  最正规的 JSON 数据是连属性名称都要被包括起来的  
  例如： {"name":"abc"}
* （2）JSON 格式的字符串的关键问题-- 它仅仅是一个字符串，只不过满足了 JSON 的数据格式   
  例如：'{"name":"abc"}'
* 看到两个例子的对比了吗？显而易见，关键问题就出在这个双引号上。
* 所以在生成 zTree 的数据时，请不要在数据外面再增加那两个双引号了！

## zTree v3.x 使用注意事项

目前的常见问题是通过原先 v2.6 的常见问题整理、提取过来的，还不够全面，在今后会逐步完善常见问题里面的内容。

### js 核心文件

**zTree v3.x** 有 3 个 js 文件，一个 核心包 ( jquery.ztree.core-3.x.js )，两个扩展包 -- 复选框功能包 ( jquery.ztree.excheck-3.x.js ) & 编辑功能包 ( jquery.ztree.exedit-3.x.js )

使用 zTree v3.x 时，核心包必须加载，扩展包根据需要进行加载；这种结构有利于今后不断开发 zTree 的各种辅助功能或工具，而不影响基本功能。

在实际使用中，建议将使用的扩展包 与 核心包 代码合并为一个 js 文件，以节省网络资源，合并时一定要将核心包的代码放在最前面.

### css 文件

**zTree v3.x** 只使用一个 zTreeStyle.css 文件，为了尽量避免样式冲突，将 zTree 容器的 className 设置为 'ztree'

v3.x 中对于主要 DOM ( li、ul、+/-图标、a ) 的 className 增加了 level 输出，可以针对不同等级的节点设定独立的样式

### img 图片

**zTree v3.x** 将主要图标都合并为一个图片，利用图片切割方式显示图标

为了避免原先节点连线之间 1 像素间隔的问题，在 v3.x 中 +/-图标 继续保留 18\*18 的尺寸，其余图标全部修改为 16\*16，请升级的朋友务必注意修正原先使用的图标大小

在 v3.x Demo 中 自定义图标都是独立的图片文件，实际开发中建议也都采用图片切割的方式

### zTree 数据

zTree 使用 JSON 数据，关于数据部分请认真查看 API 文档 "treeNode 节点数据详解" 中的内容，并且在 Demo 中也都可以看到 treeNode 的定义

### 调试代码

进行程序开发时，建议首先使用 chrome 或 firefox 进行调试工作 -- 普通 js 调试，ajax 获取数据时网络情况的跟踪都很容易。 个人推荐：chrome 的开发人员工具或 firefox 的 firebug 插件。