


zTree v2.6 --
v2.6

Hunter.z

 v2.6 ; *



zTree
bug



zTree

zTree(setting,
[zTreeNode])

skin

zTreeStyle.css

zTreeIcons.css



setting

addDiyDom

addHoverDom

async

* asyncDataFilter

asyncParam

asyncParamOther

asyncUrl

callback

checkable

checkedCol

checkStyle

checkRadioType

checkType

* dragCopy

* dragMove

edit_removeBtn

edit_renameBtn

editable

expandSpeed

fontCss

isSimpleData

keepLeaf

keepParent

nameCol

nodesCol

removeHoverDom

root

rootPID

showIcon

showLine

treeNodeKey

treeNodeParentKey

[checkRadioChecke

[curEditTreeNode]

[curTreeNode]

[dragNodeShowBefr



zTreeNodees

checked

click

icon

* iconClose

* iconOpen

iconSkin

isParent

name

* nocheck

nodes

open

target

url

**

[check_False_Full]

[check_True_Full]

[checkboxFocus]

[checkedOld]

[editNameStatus]

[isAjaxing]

[isFirstNode]


[isHover]

[isLastNode]

[level]

[parentNode]

[tId]



```
getChangeCheckedNodes()
getCheckedNodes(checked)
getNodeByParam(key, value)
getNodeByTId(tID)
getNodeIndex(treeNode)
getNodes()
getNodesByParam(key, value,
parentNode)
getNodesByParamFuzzy(key,
value, parentNode)
getSelectedNode()
getSetting()
transformToArray(treeNodes)
transformTozTreeNodes(simple
```

```
addNodes(parentNode,
newNodes, isSilent)
* cancelInput(newName)
cancelSelectedNode()
checkAllNodes(checked)
* copyNode(targetNode,
treeNode, moveType)
expandAll(expandSign)
# expandNode(treeNode,
expandSign, sonSign, focus)
* inputNodeName(treeNode)
moveNode(targetNode,
treeNode, moveType)
reAsyncChildNodes(parentNode,
reloadType)
refresh()
removeNode(treeNode)
selectNode(treeNode)
setEditable(editable)
updateNode(treeNode,
```

checkTypeFlag)



callback

beforeAsync

beforeChange

beforeClick

beforeCollapse

* beforeDbclick

beforeDrag

beforeDrop

beforeExpand

beforeMouseDown

beforeMouseUp

beforeRemove

beforeRename

beforeRightClick

* confirmDragOpen

* confirmRename

asyncError

asyncSuccess

change

click

collapse

* dblclick

drag

drop

expand

mouseDown

mouseUp

nodeCreated

remove

rename


rightClick



ZTREE_ASYNC_EF
ZTREE_ASYNC_SL
ZTREE_CHANGE
ZTREE_CLICK
ZTREE_DRAG
ZTREE_DROP
ZTREE_NODECRE
ZTREE_REMOVE
ZTREE_RENAME

ID

IDMark_A
IDMark_Check
IDMark_Edit
IDMark_Icon
IDMark_Input
IDMark_Remove
IDMark_Span
IDMark_Switch
IDMark_UI



LineMark_Bottom
LineMark_Center
LineMark_Line
LineMark_NoLine
LineMark_Root
LineMark_Roots

FolderMark_Close
FolderMark_Docu
FolderMark_Open

className

Class_CurSelectedM
Class_CurSelectedM
Class_TmpTargetNo
Class_TmpTargetTr

CheckBox & Radio

Check_Style_Box
Check_Style_Radio
CheckBox_Default
CheckBox_False
CheckBox_Focus
CheckBox_Full
CheckBox_Part
CheckBox_True
Radio_Type_All
Radio_Type_Level

MinMoveSize
MoveType_After
MoveType_Before
MoveType_Inner

zTree

JQuery Tree

zTree

zTree (Demo)

1 IEFireFoxChrome

2 Tree

3 JSON

4 Ajax

5

6 Tree

7 checkbox & radio

8 / css

9

[Hunter.z](#) ,,

[zTree](#) .

zTree v2.6

- 01
- 02checkbox
- 03jQuery 1.3.2 / jQuery 1.6
- 04zTree jsjQueryjquery.ztree-2.6.js
- 05 [expandNode](#) focus (boolean) focus false
- 06 constructor refresh zTreeld = 0; Treetreeld
id Bug
- 07contextmenu/mousedown/mouseup unbind Bug
refresh
- 08 Bug
- 09IE , Text Bug
- 10 [checkStyle: {"Y": "", "N": ""}](#) [checkAllNodes\(checked\)](#)
Bug
- 11setting. [asyncDataFilter](#) zTree
- 12setting. [dragCopy/dragMove](#) copy movefalseTree
trueCtrlcopymove
- 13treeNode. [iconOpen/iconClose](#)
- 14 [beforeDbclickdblclick](#) rightClick
- 15 [confirmRename](#)
- 16 [confirmDragOpen](#) , target
- 17 [inputNodeName\(treeNode\)](#)
- 18 [copyNode\(targetNode, treeNode, moveType\)](#) Move

- 19 v2.6 beta [addHoverDom/removeHoverDom](#) Bug
- 20 v2.6 beta zTree IE

21 v2.6 beta zTreeNodeCache [refresh\(\)](#)

22 v2.6 beta confirmRename false updateNode

Bug

23 v2.6 beta [nocheck](#) checkbox/radio

24 v2.6 beta [cancelInput\(newName\)](#) js

bug

Bug:

<http://code.google.com/p/jquerytree/issues/list> <http://hi.baidu.com/ztree>

[Email](#)

:

<http://code.google.com/p/jquerytree/downloads/list>

zTree(setting, [zTreeNodes])

: zTreePlugin

: [zTree]

JSON setting JSON zTreeNodes Tree

Web Tree zTreePlugin

Tree Web jquery-1.4.2.js / jquery.ztree-2.6.js / zTreeStyle.css

Tree class="tree" tree css

zTreeIcons.css

W3C <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

setting Json

zTree -> [setting](#)

zTreeNodes (
) JSON

zTree -> zTreeNodes
setting.root.nodes

zTree

setting :

```
var setting = {  
    showLine: true,  
    checkable: true  
};
```

zTreeNodes :


```
var zTreeNodes = [
  { "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":"_bla
];
```

zTreeNodes :

```
var zTreeNodes = [
  {"id":1, "name":"test1", "nodes":[
    {"id":11, "name":"test11", "nodes":[
      {"id":111, "name":"test111"}
    ]},
    {"id":12, "name":"test12"}
  ]},
  .....
];
```

Array zTreeNodes Array [isSimpleData :](#)

```
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"},
  .....
];
```

Html :

```
<ul id="tree" class="tree" style="width:300px; overflow:auto;"></ul
```

Tree:

```
var zTree = $("#tree").zTree(setting, zTreeNodes);
```

zTreeStyle.css

: [] --> [skin]

zTreecss

Tree zTree demo.css class="tree" tree
CSS

zTree

```
.tree{...}
```

zTreeul liTree

```
.tree li{...}
```

zTreeul

```
.tree li ul{...}
```

zTree

```
.tree li ul.line{...}
```

zTree

```
.tree li a{...}  
.tree li a:hover{...}
```

zTree

```
.tree li a.curSelectedNode{...}
```

zTree

```
.tree li a.curSelectedNode_Edit{...}
```

zTree

```
.tree li a.tmpTargetNode{...}
```

zTreename

```
.tree li a input.rename{...}
```

zTree

```
.tree li button{...}
```

zTree

```
.tree li button.ico{...}
```

zTree

```
.tree li button.edit{...}
```

zTree

```
.tree li button.del{...}
```

zTree checkbox & radio

```
.tree li button.chk{...}
.tree li button.chk.checkbox_false_full{...}
.tree li button.chk.checkbox_false_full_focus{...}
.tree li button.chk.checkbox_false_part{...}
.tree li button.chk.checkbox_false_part_focus{...}
.tree li button.chk.checkbox_true_full{...}
.tree li button.chk.checkbox_true_full_focus{...}
.tree li button.chk.checkbox_true_part{...}
.tree li button.chk.checkbox_true_part_focus{...}

.tree li button.chk.radio_false_full{...}
.tree li button.chk.radio_false_full_focus{...}
.tree li button.chk.radio_false_part{...}
.tree li button.chk.radio_false_part_focus{...}
.tree li button.chk.radio_true_full{...}
```

```
.tree li button.chk.radio_true_full_focus{...}
.tree li button.chk.radio_true_part{...}
.tree li button.chk.radio_true_part_focus{...}
```

zTree

```
.tree li button.switch_root_open{...}
.tree li button.switch_root_close{...}
.tree li button.switch_roots_open{...}
.tree li button.switch_roots_close{...}
.tree li button.switch_center_open{...}
.tree li button.switch_center_close{...}
.tree li button.switch_bottom_open{...}
.tree li button.switch_bottom_close{...}
.tree li button.switch_noLine_open{...}
.tree li button.switch_noLine_close{...}
```

zTree

```
.tree li button.switch_root_docu{...}
.tree li button.switch_roots_docu{...}
.tree li button.switch_center_docu{...}
.tree li button.switch_bottom_docu{...}
.tree li button.switch_noLine_docu{...}
```

zTree

```
.tree li button.ico_loading{...} //loading v2.2
.tree li button.ico_open{...}
.tree li button.ico_close{...}
.tree li button.ico_docu{...}
```

zTree CheckBox

```
.tree INPUT.checkbox{...}
```

zTree

```
.tmpTargetTree{...}
```

zTree

```
button.tmpzTreeMove_arrow{...} //v2.2
```

zTree

```
.zTreeDragUL{...}
```

zTreeiframeiFrameiframe

```
.zTreeMask{...}
```

zTreeIcons.css

: [] --> [skin]

zTree

zTree **v1.02**

```
.tree li button.sim1.ico_open{...}  
.tree li button.sim1.ico_close{...}  
.tree li button.sim2.ico_docu{...}  
.tree li button.sim3.ico_docu{...}  
.....
```

setting

JSON

: []

setting zTree JSON

v2.6 setting v3.x

addDiyDom

Function(treeId, treeNode

: [] --> [setting]

zTree

Function

treeId String

zTree [treeObjId](#)

treeNode JSON

JSON

setting & function :

```
var setting = {
  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.i"
    + "' 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.n
});
  .....
```


[addHoverDom](#) | [removeHoverDom](#)

addHoverDom

Function(treeId, treeNode

: [] --> [setting]

zTree

[removeHoverDom](#) zTree

Function

treeId String

zTree [treeObjId](#)

treeNode JSON

JSON

setting & function :

```
var setting = {
  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.n
});
function removeHoverDom(treeId, treeNode) {
  $("#diyBtn_" + treeNode.id).unbind().remove();
```

```
    $("#diyBtn_space_" + treeNode.id).unbind().remove();  
};  
.....
```

[removeHoverDom](#) | [addDiyDom](#)

async

Boolean

: [] --> [setting]

zTree isParent = true

false

setting :

```
var setting = {  
    async : true,  
    .....  
};
```

[asyncUrl](#) | [asyncParam](#) | [asyncParamOther](#) | [asyncDataFilter](#)

[asyncSuccess\(event, treeld, treeNode, msg\)](#) [asyncError\(event, treeld, treeNode, XMLHttpRequest, textStatus, errorThrown\)](#)

[reAsyncChildNodes\(parentNode, reloadType\)](#)

asyncDataFilter

Function

: [] --> [setting]

async = true zTree Function

null

```
function function ajaxDataFilter(treeId, parentNode, childNodes) {...}  
rootparentNode = null
```

Function

treeId String

zTree zTree callback zTree [treeObjId](#)

parentNode JSON

JSON

childNodes Array(JSON)

Array(JSON)

Function

treeNodes Array(JSON)

Function zTree

name
setting :

```
function ajaxDataFilter(treeId, parentNode, childNodes) {  
  if (childNodes) {  
    for(var i =0; i < childNodes.length; i++) {  
      childNodes[i].name += "_filter";  
    }  
  }  
  return childNodes;  
}
```

```
};  
var setting = {  
  asyncDataFilter : ajaxDataFilter,  
  .....  
};
```

[async](#) | [asyncUrl](#) | [asyncParam](#) | [asyncParamOther](#)

[reAsyncChildNodes\(parentNode, reloadType\)](#)

asyncParam

Array(String)

: [] --> [setting]

```
async = true asyncUrl name
```

```
[]
```

```
"name"
```

setting :

```
var setting = {  
    asyncParam : ["name"],  
    .....  
};
```

[async](#) | [asyncUrl](#) | [asyncParamOther](#) | [asyncDataFilter](#)

asyncParamOther

Array(String) / JSON

: [] --> [setting]

async = true asyncUrl

[]

[key value [key, value]

v2.5 JSON {key1:value1, key2:value2}

Array key, value

setting :

```
var setting = {  
    asyncParamOther : ["key", "value"],  
    .....  
};
```

JSON key, value

setting :

```
var setting = {  
    asyncParamOther : {"key":"value"},  
    .....  
};
```

[async](#) | [asyncUrl](#) | [asyncParam](#) | [asyncDataFilter](#)

asyncUrl

String / Function

: [] --> [setting]

async = true URL

""

v2.2 function url

function function getAsyncUrl(treeNode) {...} **rootTreeNode =**
null

URL nodes.php

setting :

```
var setting = {  
    asyncUrl : "nodes.php",  
    .....  
};
```

URL function

setting :

```
function getAsyncUrl(treeNode) {  
    var url = "";  
    .....  
    return url;  
};  
var setting = {  
    asyncUrl : getAsyncUrl,  
    .....  
};
```

Server

zTreeNode :

```
[
  { "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": "_bla
]
```

zTreeNode :

```
[
  { "id": 1, "name": "test1", "nodes": [
    { "id": 11, "name": "test11", "nodes": [
      { "id": 111, "name": "test111" }
    ] },
    { "id": 12, "name": "test12" }
  ] },
  .....
]
```

Array zTreeNode Array

[isSimpleData](#) :

```
[
  { "id": 1, "pId": 0, "name": "test1" },
  { "id": 11, "pId": 1, "name": "test11" },
  { "id": 12, "pId": 1, "name": "test12" },
  { "id": 111, "pId": 11, "name": "test111" },
  .....
]
```

[async](#) | [asyncParam](#) | [asyncParamOther](#) | [asyncDataFilter](#)

[reAsyncChildNodes\(parentNode, reloadType\)](#)

callback

```
var setting = {
  callback : {
    beforeAsync:null, //: beforeAsync
    beforeChange:null, //: beforeChange
    beforeClick:null, //: beforeClick
    beforeCollapse:null, //: beforeCollapse
    beforeDbclick:null, //: beforeDbclick
    beforeDrag:null, //: beforeDrag
    beforeDrop:null, //: beforeDrop
    beforeExpand:null, //: beforeExpand
    beforeMouseDown:null, //: beforeMouseDown
    beforeMouseUp:null, //: beforeMouseUp
    beforeRemove:null, //: beforeRemove
    beforeRename:null, //: beforeRename
    beforeRightClick:null, //: beforeRightClick

    confirmDragOpen:null, //: confirmDragOpen
    confirmRename:null, //: confirmRename

    asyncError:null //: asyncError
    asyncSuccess:null, //: asyncSuccess
    change:null, //: change
    click:null, //: click
    collapse:null, //: collapse
    dbclick:null, //: dbclick
    drag:null, //: drag
    drop:null, //: drop
    expand:null, //: expand
    mouseDown:null, //: mouseDown
    mouseUp:null, //: mouseUp
    nodeCreated:null, //: nodeCreated
    rename:null, //: rename
    remove:null, //: remove
    rightClick:null, //: rightClick
  },
  .....
};
```

checkable

Boolean

: [] --> [setting]

zTree CheckBox

checkbox/radio nodeCreated checkbox

false

CheckBox

setting :

```
var setting = {  
    checkable : true,  
    .....  
};
```

[checkType](#)

[change](#)

checkedCol

String

: [] --> [setting]

zTree check

"checked"

zTreeNodes

checkedOld

zTree treeNode child

setting :

```
var setting = {  
    checkedCol : "checked",  
    .....  
};
```

checkRadioType

String

: [] --> [setting]

checkable = true checkStyle = "radio" checkRadioType radio

checkRadioType = "level"
checkRadioType = "all"

"level"

setting :

```
var setting = {  
    checkable : true,  
    checkStyle : "radio",  
    checkRadioType : "all",  
    .....  
};
```

[checkable](#) | [checkStyle](#)

checkStyle

String

: [] --> [setting]

checkable = true checkStyle checkbox radio

2.0radio radio

checkStyle = "checkbox" checkbox [checkType](#)

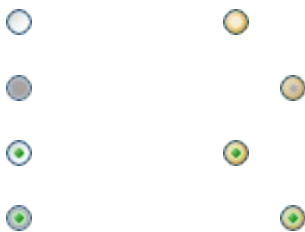
checkStyle = "radio" radio [checkRadioType](#)

checkbox

checkbox



radio



radio

setting :

```
var setting = {  
    checkable : true,
```

```
    checkStyle : "radio",  
    .....  
};
```

[checkable](#) | [checkType](#) | [checkRadioType](#)

checkType

JSON

: [] --> [setting]

checkable = true checkStyle = "checkbox" checkType
checkbox

Y CheckBox

N CheckBox

"p"

"s"

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

CheckBox

setting :

```
var setting = {  
    checkable : true,  
    checkType : { "Y": "p", "N": "s" },  
    .....  
};
```

checkable

dragCopy

Boolean

: [] --> [setting]

editable = true

1 dragCopy = true; dragMove = true Ctrl copy move

2 dragCopy = true; dragMove = false copy

3 dragCopy = false; dragMove = true move

4 dragCopy = false; dragMove = false

false ()

setting :

```
var setting = {  
  dragCopy : true,  
  dragMove : true,  
  .....  
};
```

[edit_renameBtn](#) | [edit_removeBtn](#) | [dragMove](#)

[beforeDrag\(treeId, treeNode\)](#) | [beforeDrop\(treeId, treeNode, targetNode, moveType\)](#)

[drag\(event, treeId, treeNode\)](#) | [drop\(event, treeId, treeNode, targetNode, moveType\)](#)

dragMove

Boolean

: [] --> [setting]

editable = true

1 dragCopy = true; dragMove = true Ctrl copy move

2 dragCopy = true; dragMove = false copy

3 dragCopy = false; dragMove = true move

4 dragCopy = false; dragMove = false

true ()

setting :

```
var setting = {  
  dragCopy : true,  
  dragMove : true,  
  .....  
};
```

[edit_renameBtn](#) | [edit_removeBtn](#) | [dragCopy](#)

[beforeDrag\(treeId, treeNode\)](#) | [beforeDrop\(treeId, treeNode, targetNode, moveType\)](#)

[drag\(event, treeId, treeNode\)](#) | [drop\(event, treeId, treeNode, targetNode, moveType\)](#)

edit_removeBtn

Boolean

: [] --> [setting]

zTree

1 [beforeRemove](#)

true

setting :

```
var setting = {  
  editable : true,  
  edit_removeBtn : false,  
  .....  
};
```

[editable](#) | [edit_renameBtn](#)

edit_renameBtn

Boolean

: [] --> [setting]

zTree

1

2

true

setting :

```
var setting = {  
  editable : true,  
  edit_renameBtn : false,  
  .....  
};
```

[editable](#) | [edit_removeBtn](#)

editable

Boolean

: [] --> [setting]

zTree

editable = true

1 [url](#)

2 v2.5

3 v2.3

4 **v2.6 / :** [dragCopy](#) / [dragMove](#)

5 name

6

false

setting :

```
var setting = {  
  editable : true,  
  .....  
};
```

[edit_renameBtn](#) | [edit_removeBtn](#) | [dragCopy](#) | [dragMove](#)

[beforeDrag\(treeId, treeNode\)](#) | [beforeDrop\(treeId, treeNode, targetNode, moveType\)](#)
[drag\(event, treeId, treeNode\)](#) | [drop\(event, treeId, treeNode, targetNode, moveType\)](#)

expandSpeed

String,Number

: [] --> [setting]

zTree JQuery speed

"" ("slow", "normal", or "fast") (1000)

"fast"

setting :

```
var setting = {  
    expandSpeed : "slow",  
    .....  
};
```

fontCss

JSON / Function

: [] --> [setting]

```
zTree <A>
```

```
{
```

```
  JSON JQuery css JSON      {color:"#ff0011",  
background:"blue"}
```

```
  function function setzTreeFont(treeId, treeNode) {...}
```

CSS name

setting :

```
var setting = {  
  fontCss : {color:"red"},  
  .....  
};
```

level=0 name

setting :

```
function setFontCss(treeId, treeNode) {  
  if (treeNode.level == 0) {  
    return {color:"red"};  
  }  
  return {};  
};  
var setting = {  
  fontCss : setFontCss,  
  .....  
};
```


isSimpleData

Boolean

: [] --> [setting]

zTree [addNodes\(parentNode, newNodees, isSilent\)](#)
newNodes Array

List JSON

true [treeNodeKey](#) [treeNodeParentKey](#)

false

Array

setting :

```
var setting = {  
  isSimpleData : true,  
  treeNodeKey : "id",  
  treeNodeParentKey : "pId",  
  .....  
};
```

Array :

```
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"},  
  .....  
];
```

[treeNodeKey](#) | [treeNodeParentKey](#)

[transformTozTreeNodes\(simpleTreeNodes\)](#)

keepLeaf

Boolean

: [] --> [setting]

zTree isParent=false

zTree isParent=false

false

setting :

```
var setting = {  
    keepLeaf : true,  
    .....  
};
```

[keepParent](#)

keepParent

Boolean

: [] --> [setting]

zTree isParent=true

zTree isParent=true

false

setting :

```
var setting = {  
    keepParent : true,  
    .....  
};
```

[keepLeaf](#)

nameCol

String

: [] --> [setting]

zTree

"name"

zTree treeNode ename

setting :

```
var setting = {  
    nameCol : "ename",  
    .....  
};
```

nodesCol

String

: [] --> [setting]

zTree

"nodes"

root nodes zTreeNodes

nodes

zTree treeNode child

setting :

```
var setting = {  
  nodesCol : "child",  
  root:{ child:[....] },  
  .....  
};
```

removeHoverDom

Function(treeId, treeNode

: [] --> [setting]

zTree

[addHoverDom](#) zTree

Function

treeId String

zTree [treeObjId](#)

treeNode JSON

JSON

setting & function :

```
var setting = {
  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.i"
    + "' 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.n
});
function removeHoverDom(treeId, treeNode) {
  $("#diyBtn_"+treeNode.id).unbind().remove();
```

```
    $("#diyBtn_space_" + treeNode.id).unbind().remove();  
};  
.....
```

[addHoverDom](#) | [addDiyDom](#)

root

JSON

: [] --> [setting]

zTree root.nodes

zTree setting.root.nodes zTreeNode

{ nodes:[] }

[nodesCol](#) root nodes zTreeNode

[node](#)

setting

setting :

```
var setting = {
  root : {
    nodes: [
      { "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":"_b
    ]
  },
  .....
};
```

[zTreeNode](#)

rootPID

: [] --> [setting]

[isSimpleData](#) truezTree [treeNodeParentKey](#)

rootPID zTree [treeNodeParentKey](#)

rootPID [treeNodeParentKey](#)

null

Array

setting :

```
var setting = {
  isSimpleData : true,
  rootPID : -1,
  treeNodeKey : "id",
  treeNodeParentKey : "pId",
  .....
};
```

Array :

```
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"},
  .....
];
```

[isSimpleData](#) | [treeNodeKey](#) | [treeNodeParentKey](#)

[transformTozTreeNodes\(simpleTreeNodes\)](#)

showIcon

Boolean / Function

: [] --> [setting]

zTree

true

function

```
function function showIconForTree(treeId, treeNode) {...}
```

zTree

setting :

```
var setting = {  
    showIcon : false,  
    .....  
};
```

zTree level=2

setting :

```
function showIconForTree(treeId, treeNode) {  
    return treeNode.level != 2;  
};  
var setting = {  
    showIcon : showIconForTree,  
    .....  
};
```

showLine

Boolean

: [] --> [setting]

zTree

true

zTree

setting :

```
var setting = {  
    showLine : false,  
    .....  
};
```

treeNodeKey

String

: [] --> [setting]

[isSimpleData](#) true

[transformTozTreeNodes\(simpleTreeNodes\)](#)

""

Array

setting :

```
var setting = {
    isSimpleData : true,
    rootPID : -1,
    treeNodeKey : "id",
    treeNodeParentKey : "pId",
    .....
};
```

Array :

```
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"},
    .....
];
```

[isSimpleData](#) | [rootPID](#) | [treeNodeParentKey](#)

[transformTozTreeNodes\(simpleTreeNodes\)](#)

treeNodeParentKey

String

: [] --> [setting]

[isSimpleData](#) true

[transformTozTreeNodes\(simpleTreeNodes\)](#)

""

Array

setting :

```
var setting = {
    isSimpleData : true,
    rootPID : -1,
    treeNodeKey : "id",
    treeNodeParentKey : "pId",
    .....
};
```

Array :

```
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"},
    .....
];
```

[isSimpleData](#) | [rootPID](#) | [treeNodeKey](#)

[transformTozTreeNodes\(simpleTreeNodes\)](#)

checkRadioCheckedList

Array(Object)

: [] --> [setting]

checkable = true && checkStyle = "radio" && checkRadioType = "all"
checkRadioCheckedList

curEditTreeNode

JSON

: [] --> [setting]

zTree JSON

curTreeNode

JSON

: [] --> [setting]

zTree JSON

[getSelectedNode\(\)](#)

[selectNode\(treeNode\)](#)

dragNodeShowBefore

Boolean

: [] --> [setting]

zTree

true false

dragStatus

Boolean

: [] --> [setting]

zTree

false

true

expandTriggerFlag

Boolean

: [] --> [setting]

zTree expand collapse

treeObjId

Boolean

: [] --> [setting]

zTree zTree id

zTreeNodes

Array(JSON)

: []

zTreeNode zTree JSON

zTreeNode = setting.root.nodes

[getNodes\(\)](#) zTree

[root](#)

checked

Boolean

: [] --> [zTreeNode]

setting. [checkable](#) = true CheckBox

checked [checkedCol](#)

false

CheckBox

zTreeNode :

```
var zTreeNode = [{  
  checked : true,  
  .....  
}];
```

[checkedOld](#) | [checkedCol](#)

click

String

: [] --> [zTreeNode]

onclick="...."

[click](#)

zTreeNode :

```
var zTreeNode = [{  
    click : "alert('myName')",  
    .....  
}];
```

[click](#)

icon

String

: [] --> [zTreeNode]

CSS

[zTreeStyle.css](#)

zTreeNode :

```
var zTreeNode = [{  
    icon : "folder.gif",  
    .....  
}];
```

[iconOpen](#) | [iconClose](#)

iconClose

String

: [] --> [zTreeNode]

[icon](#) iconOpen iconClose css

iconOpen iconClose

zTreeNode :

```
var zTreeNode = [{  
    iconOpen : "folderOpen.gif",  
    iconClose : "folderClose.gif",  
    .....  
}];
```

[icon](#) | [iconOpen](#)

iconOpen

String

: [] --> [zTreeNode]

[icon](#) iconOpen iconClose css

iconOpen iconClose

zTreeNode :

```
var zTreeNode = [{  
    iconOpen : "folderOpen.gif",  
    iconClose : "folderClose.gif",  
    .....  
}];
```

[icon](#) | [iconClose](#)

iconSkin

String

: [] --> [zTreeNode]

CSS ClassName **v1.02**

CSS [zTreeStyle.css](#)

CSS IE6 IE6

**CSS CSS “.tree li button.ico_open”“.tree li
button.ico_close”“.tree li button.ico_docu”**

sim

zTreeNode :

```
var zTreeNode = [{  
    iconSkin : "sim1",  
    .....  
}];
```

isParent

Boolean

: [] --> [zTreeNode]

setting. [async](#) = true isParent = true [nodes](#) length = 0

zTreeNode :

```
var zTreeNode = [{  
  isParent : true,  
  .....  
}];
```

[nodes](#) | [parentNode](#)

name

String

: [] --> [zTreeNode]

name

nameCol

Test

zTreeNode :

```
var zTreeNode = [{  
    name : "Test",  
    .....  
}];
```

nameCol

nocheck

Boolean

: [] --> [zTreeNode]

[checkable](#) true checkbox radio

1 checkbox radio zTree checked

2 [getChangeCheckedNodes\(\)](#) [getCheckedNodes\(checked\)](#)
nocheck = true

checkbox

zTreeNode :

```
var zTreeNode = [{  
  nocheck : true,  
  .....  
}];
```


nodes

Array(JSON)

: [] --> [zTreeNode]

setting. [async](#) = true [isParent](#) = true nodes length = 0

nodes [nodesCol](#)

zTreeNode :

```
var zTreeNode = [{  
  nodes : [ {...}, {...}, ...],  
  .....  
}];
```

[isParent](#) | [parentNode](#)

open

Boolean

: [] --> [zTreeNode]

true

zTreeNode :

```
var zTreeNode = [{  
  open : true,  
  .....  
}];
```

target

String

: [] --> [zTreeNode]

[url](#) target "_blank", "_self"

url

zTreeNode :

```
var zTreeNode = [{  
    target : "_blank",  
    .....  
}];
```

[url](#)

url

String

: [] --> [zTreeNode]

URL

[click](#)

g.cn

zTreeNode :

```
var zTreeNode = [{  
    url : "http://g.cn",  
    .....  
}];
```

[target](#)

String

: [] --> [zTreeNodes]

zTree

zTreeNodes :

```
var zTreeNodes = [{  
    id : "001",  
    .....  
}];
```

check_False_Full

Boolean

: [] --> [zTreeNode]

checkBox radio

(checkType = "checkbox")

1check_True_Full = true false

2check_False_Full = true false

(checkType = "radio")

1check_True_Full = true false

2check_True_Full = true false

check_True_Full

Boolean

: [] --> [zTreeNode]

checkBox radio

(checkType = "checkbox")

1check_True_Full = true false

2check_False_Full = true false

(checkType = "radio")

1check_True_Full = true false

2check_True_Full = true false

checkboxFocus

Boolean

: [] --> [zTreeNode]

checkboxBox radio mouseover

checkedOld

Boolean

: [] --> [zTreeNode]

checkBox radio

[checked](#)

v2.2 checkedNew

[checked](#)

editNameStatus

Boolean

: [] --> [zTreeNode]

editable true

editable

isAjaxing

Boolean

: [] --> [zTreeNode]

zTree

isFirstNode

Boolean

: [] --> [zTreeNode]

isLastNode

isLastNode

Boolean

: [] --> [zTreeNode]

isFirstNode

level

Number

: [] --> [zTreeNode]

level = 0

parentNode

node

: [] --> [zTreeNode]

node parentNode = null

[nodes](#) | [isParent](#)

tId

String

: [] --> [zTreeNode]

zTreeID [treeObjId](#) + "_" + zTree ID

[treeObjId](#)

getChangeCheckedNodes() : Array(JSON)

: [] --> []

zTree checkBox / radio [checked](#) != [checkedOld](#)

For Array

zTree [zTree\(setting, \[zTreeNode\]\)](#) zTreePlugin

zTree zTree [change](#) [checked](#)
[checkedOld](#)

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
var nodes = zTreeObj.getChangeCheckedNodes();  
.....
```

[getCheckedNodes\(checked\)](#) | [checkAllNodes\(checked\)](#)

getCheckedNodes(*checked*) : Array(JSON)

: [] --> []

zTree checkBox / radio

For Array

zTree [zTree\(setting, \[zTreeNode\]\)](#) zTreePlugin

1.x [getSelectedNodes\(selected\)](#)

checked Boolean

--- true: ();

false:

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
var nodes = zTreeObj.getCheckedNodes(); // zTreeObj.getCheckedNode  
.....
```

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
var nodes = zTreeObj.getCheckedNodes(false);  
.....
```

[getChangeCheckedNodes\(\)](#) | [checkAllNodes\(checked\)](#)

getNodeByParam(key, value)

: JSON Object

: [] --> []

JSON

[getNodeByParam\(key, value, parentNode\)](#)

zTree [zTree\(setting, \[zTreeNode\]\)](#) zTreePlugin

null

key String

value

id = 10

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
var node = zTreeObj.getNodeByParam("id", 10);  
.....
```

[getNodeByParam\(key, value, parentNode\)](#) | [getNodeByParamFuzzy\(value, parentNode\)](#)
[getNodeById\(tid\)](#)

getNodeByTid(tID)

: JSON Object

: [] --> []

[tid](#) JSON

zTree [zTree\(setting, \[zTreeNode\]\)](#) zTreePlugin

tID null

tID String

zTree [tid](#)

tID

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
var tID = "abc_1"  
var node = zTreeObj.getNodeByTid(tID);  
.....
```

[getNodeByParam\(key, value\)](#) | [getNodesByParam\(key, value, parentNo](#)

getNodeIndex(treeNode)

: Number

: [] --> []

0

zTree [zTree\(setting, \[zTreeNode\]\)](#) zTreePlugin

-1

treeNode JSON

JSON

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
var zIndex = zTreeObj.getNodeIndex(zTreeObj.getSelectedNode());  
.....
```

getNodes()

: Array(JSON)

: [] --> []

zTree **zTree**

zTree [zTree\(setting, \[zTreeNode\]\)](#) zTreePlugin

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
var nodes = zTreeObj.getNodes();  
.....
```

isSimpleData = true JSON :

```
[  
  {"id":1, "name":"test1", "nodes":[  
    {"id":11, "name":"test11", "nodes":[  
      {"id":111, "name":"test111"}  
    ]},  
    {"id":12, "name":"test12"}  
  ]},  
  .....  
];
```

[addNodes\(parentNode, newNodes, isSilent\)](#) | [updateNode\(treeNode, checkTypeFlag\)](#)
[moveNode\(targetNode, treeNode, moveType\)](#) | [copyNode\(targetNode, treeNode, moveType\)](#)
[removeNode\(treeNode\)](#) | [inputNodeName\(treeNode\)](#)

[root](#)

getNodesByParam(key, value, parentNode) : Array(JSON)

: [] --> []

parentNode JSON

[getNodeByParam\(key, value\)](#)

zTree [zTree\(setting, \[zTreeNode\]\)](#) zTreePlugin

Array 0 Array

key String

value

parentNode JSON Object

parentNode null

level = 1

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
var nodes = zTreeObj.getNodesByParam("level", 1);  
.....
```

"test"

js :

.....


```
var zTreeObj = zTree(setting, zTreeNodes);  
var pNode = zTreeObj.getSelectedNode();  
var nodes = zTreeObj.getNodesByParam("name", "test", pNode);  
.....
```

[getNodesByParamFuzzy\(key, value, parentNode\)](#) | [getNodeByParam\(key, value\)](#) | [getNodeById\(tid\)](#)

getNodeByParamFuzzy(key, value, parentNode) : Array(JSON)

: [] --> []

parentNode JSON

string

zTree [zTree\(setting, \[zTreeNode\]\)](#) zTreePlugin

Array 0 Array

key String

value

parentNode JSON Object

parentNode null

“”

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
var nodes = zTreeObj.getNodeByParamFuzzy("name", "");  
.....
```

“”

js :

.....

```
var zTreeObj = zTree(setting, zTreeNodes);
var pNode = zTreeObj.getSelectedNode();
var nodes = zTreeObj.getNodesByParamFuzzy("name", "", pNode);
.....
```

[getNodesByParam\(key, value, parentNode\)](#) | [getNodeByParam\(key, value, parentNode\)](#)
| [getNodeById\(tid\)](#)

getSelectedNode()

: JSON Object

: [] --> []

zTree JSON

zTree [zTree\(setting, \[zTreeNode\]\)](#) zTreePlugin

null

1.x [getCurNode\(\)](#)

zTree

js:

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
var selectedNode = zTreeObj.getSelectedNode();  
.....
```

[selectNode\(treeNode\)](#) | [cancelSelectedNode\(\)](#)

[curTreeNode](#)

getSetting()

: JSON Object

: [] --> []

zTree JSON

zTree [zTree\(setting, \[zTreeNode\]\)](#) zTreePlugin

1 JSON zTree setting

2 [root](#) , [getNodes\(\)](#)

3 setting [updateSetting\(setting\)](#)

zTree

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
var xSetting = zTreeObj.getSetting();  
.....
```

[updateSetting\(setting\)](#)

transformToArray(treeNodes) : Array(JSON)

: [] --> []

zTree Array **Array**

zTree [zTree\(setting, \[zTreeNodes\]\)](#) zTreePlugin

treeNodes Array(JSON) / JSON

zTree

zTree Array

js :

```
.....  
var zTreeObj = zTree(setting, treeNodes);  
var treeNodes = zTreeObj.getNodes();  
var simpleTreeNodes = zTreeObj.transformToArray(treeNodes);  
.....
```

[transformTozTreeNodes\(simpleTreeNodes\)](#)

`transformTozTreeNodes(simpleTreeNodes)` : Array(JSON)

: [] --> []

Array zTree **zTree**
 treeNodeKey treeNodeParentKey
zTree zTree(setting, [zTreeNodes]) zTreePlugin

simpleTreeNodes Array(JSON) / JSON

Array JSON 1 Array

Array zTreezTree

js :

```
.....  
var setting = {  
    isSimpleData : true,  
    treeNodeKey : "id",  
    treeNodeParentKey : "pId",  
    .....  
};  
var zTreeObj = zTree(setting, zTreeNodes);  
var simpleTreeNodes = [  
    {"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 treeNodes = zTreeObj.transformTozTreeNodes(simpleTreeNodes);  
zTreeObj.addNodes(null, treeNodes);  
.....
```

[transformToArray\(treeNodes\)](#)

[treeNodeKey](#) | [treeNodeParentKey](#)


```
addNodes(parentNode, newNodes, :  
isSilent)
```

```
: [] --> []
```

```
zTree zTree\(setting, \[zTreeNode\]\) zTreePlugin
```

```
parentNode    JSON Object
```

```
parentNode null
```

```
newNodes     Array(JSON)
```

```
JSON nzTree    zTreeNode
```

```
isSilent     Boolean
```

```
isSilent = true
```

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
var newNodes = [ {...}, {...}, ...];  
var nodes = zTreeObj.addNodes(null, newNodes);  
.....
```

```
getNodes\(\) | updateNode\(treeNode, checkTypeFlag\)  
moveNode\(targetNode, treeNode, moveType\) | copyNode\(targetNode,  
treeNode, moveType\)  
removeNode\(treeNode\) | inputNodeName\(treeNode\)
```

[nodes](#)

cancelInput(newName)

:

: [] --> []

zTree [zTree\(setting, \[zTreeNode\]\)](#) zTreePlugin

newName String

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
zTreeObj.cancelInput();  
.....
```

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
zTreeObj.cancelInput("newName test");  
.....
```

[inputNodeName\(treeNode\)](#)

cancelSelectedNode()

:

: [] --> []

zTree [zTree\(setting, \[zTreeNode\]\)](#) zTreePlugin

v2.2 [cancelSelectedNode](#) [cancelSelectedNode](#)
[cancelSelectedNode](#)

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
.....  
zTreeObj.cancelSelectedNode();  
.....
```

[getSelectedNode\(\)](#) | [selectNode\(treeNode\)](#)

checkAllNodes(checked) :

: [] --> []

checkable true

zTree zTree(setting, [zTreeNode]) zTreePlugin

checked Boolean

truefalse

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
zTreeObj.checkAllNodes(true);  
.....
```

getCheckedNodes(checked) | getChangeCheckedNodes()

copyNode(targetNode, treeNode, moveType) : JSON

: [] --> []

[moveNode\(targetNode, treeNode, moveType\)](#)

zTree [zTree\(setting, \[zTreeNode\]\)](#) zTreePlugin

targetNode JSON

JSON targetNode null

treeNode JSON

JSON

moveType String

"inner""before""after"

1treeNode12treeNode2

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
zTreeObj.copyNode(treeNode2, treeNode1);  
.....
```

1treeNode12treeNode2

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);
```

```
zTreeObj.copyNode(treeNode2, treeNode1, "before");  
.....
```

[getNodes\(\)](#) | [addNodes\(parentNode, newNodes, isSilent\)](#) |
[updateNode\(treeNode, checkTypeFlag\)](#)
[moveNode\(targetNode, treeNode, moveType\)](#) | [removeNode\(treeNode\)](#)
[inputNodeName\(treeNode\)](#)

expandAll(expandSign)

:

: [] --> []

zTree

zTree [zTree\(setting, \[zTreeNode\]\)](#) zTreePlugin

expandSign Boolean

(true) (false)

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
zTreeObj.expandAll(true);  
.....
```

[expandNode\(treeNode, expandSign, sonSign, focus\)](#)

expandNode(treeNode, expandSign, sonSign, focus)

: [] --> []

zTree

2.0

zTree [zTree\(setting, \[zTreeNodes\]\)](#) zTreePlugin

treeNode JSON

JSON

expandSign Boolean

(true) (false)

sonSign Boolean

focus Boolean

true

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNodes);  
zTreeObj.expandNode(treeNode, true, false);  
.....
```

[expandAll\(expandSign\)](#)

inputnodeName(treeNode)

:

: [] --> []

zTree [zTree\(setting, \[zTreeNodes\]\)](#) zTreePlugin

js [cancelInput\(newName\)](#)

treeNode JSON

JSON

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNodes);  
zTreeObj.inputnodeName(zTreeObj.getSelectedNode());  
.....
```

[getNodes\(\)](#) | [addNodes\(parentNode, newNodes, isSilent\)](#) |
[updateNode\(treeNode, checkTypeFlag\)](#)
[moveNode\(targetNode, treeNode, moveType\)](#) | [copyNode\(targetNode,
treeNode, moveType\)](#)
[removeNode\(treeNode\)](#) | [cancelInput\(newName\)](#)

**moveNode(targetNode, treeNode, :
moveType)**

: [] --> []

2.2 moveType

zTree [zTree\(setting, \[zTreeNodes\]\)](#) zTreePlugin

targetNode JSON

JSON targetNode null

treeNode JSON

JSON

moveType String

"inner""before""after"

1treeNode12treeNode2

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNodes);  
zTreeObj.moveNode(treeNode2, treeNode1);  
.....
```

1treeNode12treeNode2

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNodes);
```

```
zTreeObj.moveNode(treeNode2, treeNode1, "before");  
.....
```

[getNodes\(\)](#) | [addNodes\(parentNode, newNodes, isSilent\)](#) |
[updateNode\(treeNode, checkTypeFlag\)](#)
[copyNode\(targetNode, treeNode, moveType\)](#) | [removeNode\(treeNode\)](#)
[inputNodeName\(treeNode\)](#)

```
reAsyncChildNodes(parentNode, :  
reloadType)
```

```
: [] --> []
```

```
zTree zTree(setting, [zTreeNode]) zTreePlugin
```

```
parentNode JSON
```

```
JSON
```

```
parentNode = null Root
```

```
parentNode.isParent = false
```

```
reloadType String
```

```
reloadType = "refresh"
```

```
treeNode1
```

```
js :
```

```
.....
```

```
var zTreeObj = zTree(setting, zTreeNode);
```

```
zTreeObj.reAsyncChildNodes(treeNode1, "refresh");
```

```
.....
```

```
asyncUrl
```

refresh()

:

: [] --> []

zTree

[updateNode\(treeNode, checkTypeFlag\)](#)

[reAsyncChildNodes\(parentNode, reloadType\)](#)

zTree [zTree\(setting, \[zTreeNodes\]\)](#) zTreePlugin

zTree

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNodes);  
zTreeObj.refresh();  
.....
```

removeNode(treeNode)

:

: [] --> []

zTree [zTree\(setting, \[zTreeNodes\]\)](#) zTreePlugin

treeNode JSON

JSON

1treeNode1

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNodes);  
zTreeObj.removeNode(treeNode1);  
.....
```

[getNodes\(\)](#) | [addNodes\(parentNode, newNodes, isSilent\)](#) |
[updateNode\(treeNode, checkTypeFlag\)](#)
[moveNode\(targetNode, treeNode, moveType\)](#) | [copyNode\(targetNode,
treeNode, moveType\)](#)
| [inputNodeName\(treeNode\)](#)

selectNode(treeNode)

:

: [] --> []

2.0

zTree [zTree\(setting, \[zTreeNodes\]\)](#) zTreePlugin

treeNode JSON

JSON

1treeNode1

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNodes);  
zTreeObj.selectNode(treeNode1);  
.....
```

[getSelectedNode\(\)](#) | [cancelSelectedNode\(\)](#)

setEditable(editable)

:

: [] --> []

zTree

zTree [zTree\(setting, \[zTreeNode\]\)](#) zTreePlugin

editable Boolean

(true) (false)

zTree

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
zTreeObj.setEditable(true);  
.....
```

[editable](#)

**updateNode(treeNode, :
checkTypeFlag)**

: [] --> []

zTree [zTree\(setting, \[zTreeNodes\]\)](#) zTreePlugin
[nameCol](#)

v2.2updateNodenametarget urliconiconSkinzTreeNodes
expandNodeopen

treeNode JSON

JSON

checkTypeFlag Boolean

setting.checkType
false

(treeNode1) zTree

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNodes);  
.....  
treeNode1.name = "test Name";  
zTreeObj.updateNode(treeNode1, true);  
.....
```

[getNodes\(\)](#) | [addNodes\(parentNode, newNodes, isSilent\)](#)
[moveNode\(targetNode, treeNode, moveType\)](#) | [copyNode\(targetNode,](#)

[treeNode, moveType\)](#)
[removeNode\(treeNode\) | inputNodeName\(treeNode\)](#)

[checkType | nameCol](#)

updateSetting(setting)

:

: [] --> []

zTree

zTree [zTree\(setting, \[zTreeNode\]\)](#) zTreePlugin

zTree

1 [showLineEditableCheckable...](#) **zTree** [refresh\(\)](#)

2

[asyncUrl](#)[asyncParam](#)[asyncParamOther](#)[checkType](#)[expandSpeed](#)[callb](#)

3 [treeObjId](#) [root](#) , **Node**

4 [getSetting\(\)](#)

zTree URL (Demo“CheckBox ”)

js :

```
.....  
var zTreeObj = zTree(setting, zTreeNode);  
var xSetting = zTreeObj.getSetting();  
xSetting.asyncUrl = "node.jsp";  
zTreeObj.updateSetting(xSetting);  
.....
```

[getSetting\(\)](#)

beforeAsync(treeId, treeNode) : Boolean

: [] --> [callback]

zTree beforeAsync(treeId, treeNode) setting
[callback](#) zTree

false [asyncSuccess](#) [asyncError](#)

treeId String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON

zTree

js:

```
.....  
var setting = {  
  callback : {  
    beforeAsync: zTreeBeforeAsync,  
    .....  
  },  
  .....  
};  
.....  
function zTreeBeforeAsync(treeId, treeNode) {  
  if (treeNode.id == 1) return false;  
  return true;  
}  
.....
```

[asyncSuccess\(event, treeId, treeNode, msg\)](#) [asyncError\(event, treeId,](#)

treeNode, XMLHttpRequest, textStatus, errorThrown)

beforeChange(treeId, treeNode) : Boolean

: [] --> [callback]

zTree beforeChange(treeId, treeNode) setting
[callback](#) zTree change checkbox radio

checkbox radio false change [change](#)

treeId String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

checkbox radio JSON

zTree change

js:

```
.....  
var setting = {  
  callback : {  
    beforeChange: zTreeBeforeChange,  
    .....  
  },  
  .....  
};  
.....  
function zTreeBeforeChange(treeId, treeNode) {  
  return false;  
}  
.....
```

[change\(event, treeId, treeNode\)](#)

[checkable](#)

beforeClick(treeId, treeNode) : Boolean

: [] --> [callback]

zTree beforeClick(treeId, treeNode) setting
[callback](#) zTree click

false click [click](#)

treeId String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON

zTree click

js:

```
.....  
var setting = {  
    callback : {  
        beforeClick: zTreeBeforeClick,  
        .....  
    },  
    .....  
};  
.....  
function zTreeBeforeClick(treeId, treeNode) {  
    return false;  
}  
.....
```

[click\(event, treeId, treeNode\)](#)

beforeCollapse(treeId, treeNode) : Boolean

: [] --> [callback]

zTree beforeCollapse(treeId, treeNode) setting
[callback](#) zTree collapse

(+) **false collapse** [collapse](#)

treeId String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON

zTree collapse

js:

```
.....  
var setting = {  
  callback : {  
    beforeCollapse: zTreeBeforeCollapse,  
    .....  
  },  
  .....  
};  
.....  
function zTreeBeforeCollapse(treeId, treeNode) {  
  return false;  
}  
.....
```

[collapse\(event, treeId, treeNode\)](#) [beforeExpand\(treeId, treeNode\)](#) | [expand\(event, treeId, treeNode\)](#)

beforeDbclick(treeId, treeNode) : Boolean

: [] --> [callback]

zTree beforeDbclick(treeId, treeNode) setting
[callback](#) zTree dbclick

zTree false dbclick

[dbclick](#)

treeId String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

A JSON null

zTree dbclick

js:

```
.....  
var setting = {  
    callback : {  
        beforeDbclick: zTreeBeforeDbclick,  
        .....  
    },  
    .....  
};  
.....  
function zTreeBeforeDbclick(treeId, treeNode) {  
    return false;  
}  
.....
```

[dbclick\(event, treeId, treeNode\)](#)

beforeDrag(treeId, treeNode) : Boolean

: [] --> [callback]

zTree beforeDrag(treeId, treeNode) setting
[callback](#) zTree drag

false drag [drag](#)

treeId String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON

zTree drag

js:

```
.....  
var setting = {  
  callback : {  
    beforeDrag: zTreeBeforeDrag,  
    .....  
  },  
  .....  
};  
.....  
function zTreeBeforeDrag(treeId, treeNode) {  
  return false;  
}  
.....
```

[drag\(event, treeId, treeNode\)](#) | [beforeDrop\(treeId, treeNode, targetNode, moveType\)](#)

[drop\(event, treeId, treeNode, targetNode, moveType\)](#)

[editable](#)

beforeDrop(treeld, treeNode, targetNode, moveType) : Boolean

: [] --> [callback]

zTree beforeDrop(treeld, treeNode, targetNode, moveType)
setting callback zTree drop

false drop drop

treeld String

zTree zTree callback zTree treeObjId
zTree treeObjId

treeNode JSON

JSON
null

targetNode JSON

treeNode JSON
null

moveType String

"inner""before""after"

zTree drop

js :

```
.....  
var setting = {  
  callback : {  
    beforeDrop: zTreeBeforeDrop,  
    .....
```

```
    },
    .....
};
.....
function zTreeBeforeDrop(treeId, treeNode, targetNode, moveType) {
    return false;
}
.....
```

[drop\(event, treeId, treeNode, targetNode, moveType\) beforeDrag\(treeId, treeNode\) | drag\(event, treeId, treeNode\)](#)

[editable](#)

beforeExpand(treeId, treeNode) : Boolean

: [] --> [callback]

zTree beforeExpand(treeId, treeNode) setting
[callback](#) zTree expand

(+) false expand [expand](#)

treeId String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON

zTree expand

js:

```
.....  
var setting = {  
  callback : {  
    beforeExpand: zTreeBeforeExpand,  
    .....  
  },  
  .....  
};  
.....  
function zTreeBeforeExpand(treeId, treeNode) {  
  return false;  
}  
.....
```

[expand\(event, treeId, treeNode\)](#) [beforeCollapse\(treeId, treeNode\)](#) |
[collapse\(event, treeId, treeNode\)](#)

beforeMouseDown(treeId, treeNode) : Boolean

: [] --> [callback]

zTree beforeMouseDown(treeId, treeNode) setting
[callback](#) zTree mouseDown

false [mouseDown](#)

treeId String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

mouseDown A JSON null

zTree mouseDown

js:

```
.....  
var setting = {  
  callback : {  
    beforeMouseDown: zTreeBeforeMouseDown,  
    .....  
  },  
  .....  
};  
.....  
function zTreeBeforeMouseDown(treeId, treeNode) {  
  return false;  
}  
.....
```

[mouseDown\(event, treeId, treeNode\)](#)

beforeMouseUp(treeId, treeNode) : Boolean

: [] --> [callback]

zTree beforeMouseUp(treeId, treeNode) setting
[callback](#) zTree mouseUp

false [mouseUp](#)

treeId String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

mouseUp A JSON null

zTree mouseUp

js:

```
.....  
var setting = {  
  callback : {  
    beforeMouseUp: zTreeBeforeMouseUp,  
    .....  
  },  
  .....  
};  
.....  
function zTreeBeforeMouseUp(treeId, treeNode) {  
  return false;  
}  
.....
```

[mouseUp\(event, treeId, treeNode\)](#)

beforeRemove(treeId, treeNode)

: [] --> [callback]

zTree beforeRemove(treeId, treeNode) setting
[callback](#)

false remove

[remove](#)

treeId String

zTree zTree callback zTree

[treeObjId](#)

treeNode JSON

JSON

js :

```
.....  
var setting = {  
  callback : {  
    beforeRemove: zTreeBeforeDel,  
    .....  
  },  
  .....  
};  
.....  
function zTreeBeforeDel(treeId, treeNode) {  
  return false;  
}  
.....
```

[remove\(event, treeId, treeNode\)](#)

[editable](#) | [edit_removeBtn](#)

beforeRename(treeId, treeNode) : Boolean

: [] --> [callback]

zTree beforeRename(treeId, treeNode) setting
[callback](#)

false [rename](#)

v2.6 ESC

treeId String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON

js:

```
.....  
var setting = {  
  callback : {  
    beforeRename: zTreeBeforeRename,  
    .....  
  },  
  .....  
};  
.....  
function zTreeBeforeRename(treeId, treeNode) {  
  return false;  
}  
.....
```

[rename\(event, treeld, treeNode\)](#)

[editable](#) | [edit_renameBtn](#)

beforeRightClick(treeId, treeNode) : Boolean

: [] --> [callback]

zTree beforeRightClick(treeId, treeNode) setting
[callback](#) zTree rightClick

false rightClick

[rightClick](#)

treeId String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

A JSON null

zTree rightClick

js:

```
.....  
var setting = {  
    callback : {  
        beforeRightClick: zTreeBeforeRightClick,  
        .....  
    },  
    .....  
};  
.....  
function zTreeBeforeRightClick(treeId, treeNode) {  
    return false;  
}  
.....
```

[rightClick\(event, treeId, treeNode\)](#)

confirmDragOpen(treeId, treeNode) : Boolean

: [] --> [callback]

zTree confirmDragOpen(treeId, treeNode) setting
[callback](#)

zTree false

treeId String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON

zTree

js:

```
.....  
var setting = {  
    callback : {  
        confirmDragOpen: zTreeConfirmDragOpen,  
        .....  
    },  
    .....  
};  
.....  
function zTreeConfirmDragOpen(treeId, treeNode) {  
    return false;  
}  
.....
```


confirmRename(treeId, treeNode, newName) : Boolean

: [] --> [callback]

zTree confirmRename(treeId, treeNode, newName)
setting [callback](#) [rename](#)
false [rename](#)

v2.6 ESC

confirmRename false

treeId String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON

newName String

5

js :

```
.....  
var setting = {  
  callback : {  
    confirmRename: zTreeConfirmRename,  
    .....  
  },  
  .....  
};  
.....  
function zTreeConfirmRename(treeId, treeNode, newName) {
```

```
        return newName.length > 5;
    }
    .....
```

asyncError(event, treeld, treeNode, XMLHttpRequest, textStatus, errorThrown)

: [] --> [callback]

zTree asyncError(event, treeld, treeNode, XMLHttpRequest, textStatus, errorThrown) zTree

zTree v1.x

event js event

js event

treeld String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON

XMLHttpRequest String

XMLHttpRequest JQuery API

textStatus String

successerror JQuery API

errorThrown String

errorThrown JQuery API

js :

.....

```
function zTreeOnAsyncError(event, treeId, treeNode, XMLHttpRequest,
    alert(XMLHttpRequest));
}
.....
```

[beforeAsync\(treeId, treeNode\) | asyncSuccess\(event, treeId, treeNode, msg\)](#)

[async](#)

asyncSuccess(event, treeld, treeNode, msg)

: [] --> [callback]

zTree asyncSuccess(event, treeld, treeNode, msg)
zTree

zTree v1.x

event js event

js event

treeld String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON

msg String

js :

```
.....  
function zTreeOnAsyncSuccess(event, treeId, treeNode, msg) {  
    alert(msg);  
}  
.....
```

[beforeAsync\(treeId, treeNode\)](#) | [asyncError\(event, treeId, treeNode,](#)

`XMLHttpRequest, textStatus, errorThrown)`

`async`

change(event, treeld, treeNode)

: [] --> [callback]

zTree onChange(event, treeld, treeNode) setting
[callback](#) zTree change

CheckBox

[beforeChange](#) false change

zTree v1.x

event js event

onChange event

treeld String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

checkbox radio JSON

zTree checkbox radio tId name

js:

```
.....
var setting = {
  callback : {
    change: zTreeOnChange,
    .....
  },
  .....
};
.....
function zTreeOnChange(event, treeId, treeNode) {
  alert(treeNode.tId + ", " + treeNode.name);
}
```

.....

[beforeChange\(treeId, treeNode\)](#)

[checkable](#)

click(event, treeld, treeNode)

: [] --> [callback]

zTree onClick(event, treeld, treeNode) setting
[callback](#) zTree click

[beforeClick](#) false click

zTree v1.x

event js event

onClick event

treeld String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON

zTree tId name

js:

```
.....  
var setting = {  
    callback : {  
        click: zTreeOnClick,  
        .....  
    },  
    .....  
};  
.....  
function zTreeOnClick(event, treeId, treeNode) {  
    alert(treeNode.tId + ", " + treeNode.name);  
}
```

.....

[beforeClick\(treeId, treeNode\)](#)

collapse(event, treeld, treeNode)

: [] --> [callback]

zTree onCollapse(event, treeld, treeNode) setting
[callback](#) zTree collapse

[beforeCollapse](#) false collapse

js zTree...

event js event

event

treeld String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON

zTree tId name

js:

```
.....  
var setting = {  
    callback : {  
        collapse: zTreeOnCollapse,  
        .....  
    },  
    .....  
};  
.....  
function zTreeOnCollapse(event, treeId, treeNode) {  
    alert(treeNode.tId + ", " + treeNode.name);  
}
```

.....

[beforeCollapse\(treeId, treeNode\)](#) [beforeExpand\(treeId, treeNode\)](#) | [expand\(event, treeId, treeNode\)](#)

dblclick(event, treeld, treeNode)

: [] --> [callback]

zTree onDbclick(event, treeld, treeNode) setting
[callback](#) zTree dblclick

[beforeDbclick](#) false **dblclick**

zTree v1.x

event js event

onClick event

treeld String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

A JSON null

zTree tld name

js:

```
.....
var setting = {
  callback : {
    dblclick: zTreeOnDbclick,
    .....
  },
  .....
};
.....
function zTreeOnDbclick(event, treeId, treeNode) {
  if (treeNode == null) {
    alert(' -- zTree -- ');
  }
}
```

```
    } else {  
      alert(treeNode.tId + ", " + treeNode.name);  
    }  
  }  
  .....  
}
```

[beforeClick\(treeId, treeNode\)](#)

drag(event, treeld, treeNode)

: [] --> [callback]

zTree onDrag(event, treeld, treeNode) setting
[callback](#) zTree drag

[beforeDrag](#) false drag

zTree v1.x

event js event

js event

treeld String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON

tId name

js :

```
.....
var setting = {
  callback : {
    drag: zTreeOnDrag,
    .....
  },
  .....
};
.....
function zTreeOnDrag(event, treeId, treeNode) {
  alert(treeNode.tId + ", " + treeNode.name);
}
```

.....

[beforeDrag\(treeId, treeNode\) | beforeDrop\(treeId, treeNode, targetNode, moveType\)](#)
[drop\(event, treeId, treeNode, targetNode, moveType\)](#)

[editable](#)

drop(event, treeld, treeNode, targetNode, moveType)

: [] --> [callback]

zTree onDrop(event, treeld, treeNode, targetNode, moveType)
setting [callback](#) zTree drop

[beforeDrop](#) false drop

zTree v1.x

event js event

js event

treeld String

zTree zTree callback zTree
zTree [treeObjId](#)

treeNode JSON

JSON
null

targetNode JSON

treeNode JSON
null

moveType String

"inner""before""after"

tld name

js :

```
.....
var setting = {
  callback : {
    drop: zTreeOnDrop,
    .....
  },
  .....
};
.....
function zTreeOnDrop(event, treeId, treeNode, targetNode, moveType)
  if (treeNode) alert("treeNode = " + treeNode.tId + ", " + treeNode.tText);
  if (targetNode) alert("targetNode = " + targetNode.tId + ", " + targetNode.tText);
}
.....
```

[beforeDrop\(treeId, treeNode, targetNode, moveType\) beforeDrag\(treeId, treeNode\) | drag\(event, treeId, treeNode\)](#)

[editable](#)

expand(event, treeld, treeNode)

: [] --> [callback]

zTree onExpand(event, treeld, treeNode) setting
[callback](#) zTree expand

[beforeExpand](#) false expand

js zTree...

event js event

event

treeld String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON

zTree tId name

js:

```
.....  
var setting = {  
    callback : {  
        expand: zTreeOnExpand,  
        .....  
    },  
    .....  
};  
.....  
function zTreeOnExpand(event, treeId, treeNode) {  
    alert(treeNode.tId + ", " + treeNode.name);  
}
```

.....

beforeExpand(treeId, treeNode) beforeCollapse(treeId, treeNode) |
collapse(event, treeId, treeNode)

mouseDown(event, treeld, treeNode)

: [] --> [callback]

zTree onMouseDown(event, treeld, treeNode)
setting [callback](#) zTree mouseDown

[beforeMouseDown](#) **false mouseDown**

zTree

event js event

onMouseDown event

treeld String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

mouseDown A JSON null

tId name

js:

```
.....  
var setting = {  
  callback : {  
    mouseDown: zTreeOnMouseDown,  
    .....  
  },  
  .....  
};  
.....  
function zTreeOnMouseDown(event, treeId, treeNode) {  
  if (treeNode)  
    alert(treeNode.tId + ", " + treeNode.name);  
}
```

```
    else
      alert("is root");
}
.....
```

[beforeMouseDown\(treeId, treeNode\)](#)

mouseUp(event, treeld, treeNode)

: [] --> [callback]

zTree onMouseUp(event, treeld, treeNode) setting
[callback](#) zTree mouseUp

[beforeMouseUp](#) false mouseUp

zTree

event js event

onMouseUp event

treeld String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

mouseUp A JSON null

tId name

js:

```
.....
var setting = {
  callback : {
    mouseUp: zTreeOnMouseUp,
    .....
  },
  .....
};
.....
function zTreeOnMouseUp(event, treeId, treeNode) {
  if (treeNode)
    alert(treeNode.tId + ", " + treeNode.name);
}
```

```
    else
      alert("is root");
}
.....
```

[beforeMouseUp\(treeId, treeNode\)](#)

nodeCreated(event, treeld, treeNode)

: [] --> [callback]

zTree onNodeCreated(event, treeld, treeNode)
setting [callback](#) zTree nodeCreated

zTree

event js event

js event

treeld String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON

zTree

js:

```
.....  
var setting = {  
  callback : {  
    nodeCreated: zTreeOnNodeCreated,  
    .....  
  },  
  .....  
};  
.....  
var lastRenderingNode = null;  
function zTreeOnNodeCreated(event, treeId, treeNode) {  
  lastRenderingNode = treeNode;  
}  
.....
```

remove(event, treeld, treeNode)

: [] --> [callback]

zTree onRemove(event, treeld, treeNode) setting
[callback](#) zTree remove

[beforeRemove](#) **false** **remove**

event js event

js event

treeld String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON

tId name

js:

```
.....  
var setting = {  
    callback : {  
        remove: zTreeOnRemove,  
        .....  
    },  
    .....  
};  
.....  
function zTreeOnRemove(event, treeId, treeNode) {  
    alert(treeNode.tId + ", " + treeNode.name);  
}  
.....
```

[beforeRemove\(treeId, treeNode\)](#)

[editable](#) | [edit_removeBtn](#)

rename(event, treeld, treeNode)

: [] --> [callback]

zTree onRename(event, treeld, treeNode) setting
[callback](#) zTree rename

onblur

[beforeRename](#) **false**

event js event

js event

treeld String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON

tId name

js:

```
.....  
var setting = {  
    callback : {  
        rename: zTreeOnRename,  
        .....  
    },  
    .....  
};  
.....  
function zTreeOnRename(event, treeId, treeNode) {  
    alert(treeNode.tId + ", " + treeNode.name);  
}  
.....
```

[beforeRename\(treeId, treeNode\)](#)

[editable](#) | [edit_renameBtn](#)

rightClick(event, treeld, treeNode)

: [] --> [callback]

zTree onRightClick(event, treeld, treeNode) setting
[callback](#) zTree rightClick

[beforeRightClick](#) false rightClick

function rightClick zTree

event js event

onClick event

treeld String

zTree zTree callback zTree [treeObjId](#)

treeNode JSON

JSON **treeNode = null, zTree Root**
zTree treeNode

zTree tId name

js:

```
.....
var setting = {
  callback : {
    rightClick: zTreeOnRightClick,
    .....
  },
  .....
};
.....
function zTreeOnRightClick(event, treeId, treeNode) {
```

```
    if (treeNode)
        alert(treeNode.tId + ", " + treeNode.name);
    else
        alert("is root");
}
.....
```

[beforeRightClick\(treeId, treeNode\)](#)

ZTREE_ASYNC_ERROR

: [] --> []

zTree [asyncError](#)

ZTREE_ASYNC_SUCCESS

: [] --> []

zTree [asyncSuccess](#)

ZTREE_CHANGE

: [] --> []

zTree [change](#)

1.x ZTREE_CHECK

ZTREE_CLICK

: [] --> []

zTree [click](#)

ZTREE_DRAG

: [] --> []

zTree [drag](#)

ZTREE_DROP

: [] --> []

zTree [drop](#)

ZTREE_NODECREATED

: [] --> []

zTree [nodeCreated](#)

ZTREE_REMOVE

: [] --> []

zTree [remove](#)

ZTREE_RENAME

: [] --> []

zTree [rename](#)

IDMark_A

: [] --> [ID]

zTree id

IDMark_Check

: [] --> [ID]

zTree checkBox radio id

IDMark_Edit

: [] --> [ID]

zTree

IDMark_Icon

: [] --> [ID]

zTree id

IDMark_Input

: [] --> [ID]

zTree name

IDMark_Remove

: [] --> [ID]

zTree

IDMark_Span

: [] --> [ID]

zTree name

IDMark_Switch

: [] --> [ID]

zTree id

IDMark_UI

: [] --> [ID]

zTree UL id

LineMark_Bottom

: [] --> []

zTree id

LineMark_Center

: [] --> []

zTree id

LineMark_Line

: [] --> []

zTree id

LineMark_NoLine

: [] --> []

zTree id

LineMark_Root

: [] --> []

zTree id

LineMark_Roots

: [] --> []

zTree id

FolderMark_Close

: [] --> []

zTree id

FolderMark_Docu

: [] --> []

zTree id

FolderMark_Open

: [] --> []

zTree id

Class_CurSelectedNode

: [] --> [className]

zTree class

Class_CurSelectedNode_Edit

: [] --> [className]

zTree class

Class_TmpTargetNode

: [] --> [className]

zTree class

Class_TmpTargetTree

: [] --> [className]

zTree class

Check_Style_Box

: [] --> [CheckBox & Radio

checkStyle CheckBox

Check_Style_Radio

: [] --> [CheckBox & Radio

checkStyle Radio

CheckBox_Default

: [] --> [CheckBox & Radio

CheckBox Radio class

CheckBox_False

: [] --> [CheckBox & Radio

CheckBox Radio class

CheckBox_Focus

: [] --> [CheckBox & Radio

CheckBox Radio class

CheckBox_Full

: [] --> [CheckBox & Radio

CheckBox Radio class

CheckBox_Part

: [] --> [CheckBox & Radio

CheckBox Radio class

CheckBox_True

: [] --> [CheckBox & Radio

CheckBox Radio class

Radio_Type_All

: [] --> [CheckBox & Radio

Radio

Radio_Type_Level

: [] --> [CheckBox & Radio

Radio

MinMoveSize

: [] --> []

zTree

MoveType_After

: [] --> []

zTree

MoveType_Before

: [] --> []

zTree

MoveType_Inner

: [] --> []

zTree