YUI Library: TreeView

2007-12-4 **v2.4**

Simple Use Case

```
var tree = new YAHOO.widget.TreeView("treeDiv1");
var root = tree.getRoot();
var tmpNode = new YAHOO.widget.TextNode("mylabel",
   root, false);
tree.draw();
```

Places a Tree control in the HTML element whose ID attribute is "treediv1"; adds one node to the top level of the Tree and renders.

Constructor: YAHOO.widget.TreeView

```
YAHOO.widget.TreeView(str | element target);
```

Arguments:
(1) **Element id or reference:** HTML ID or element reference for the element being into which the Tree's DOM structure will be inserted.

Nodes: TextNode, MenuNode, HTMLNode

TextNode (for simple labeled nodes):

```
YAHOO.widget.TextNode(obj | str oData, Node obj
   oParent[, b expanded]);
```

Arguments:
(1) **Associated data:** A string containing the node label or an object containing str `label`, str `href`, and any other custom members desired. If no `oData.href` is provided, clicking on the TextNode's intrinsic `<a>` tag will invoke the node's `expand` method.
(2) **Parent node:** The node object of which the new node will be a child; for top-level nodes, the parent is the Tree's root node.
(3) **Expanded state:** A boolean indicating whether the node is expanded when the Tree is rendered.

MenuNode (for auto-collapsing node navigation):

MenuNodes are identical to TextNodes in construction and behavior, except that only one MenuNode can be open at any time for a given level of depth.

HTMLNode (for nodes with customized HTML for labels):

```
YAHOO.widget.HTMLNode(obj | str HTML, Node obj
   oParent[, b expanded,b hasIcon]);
```

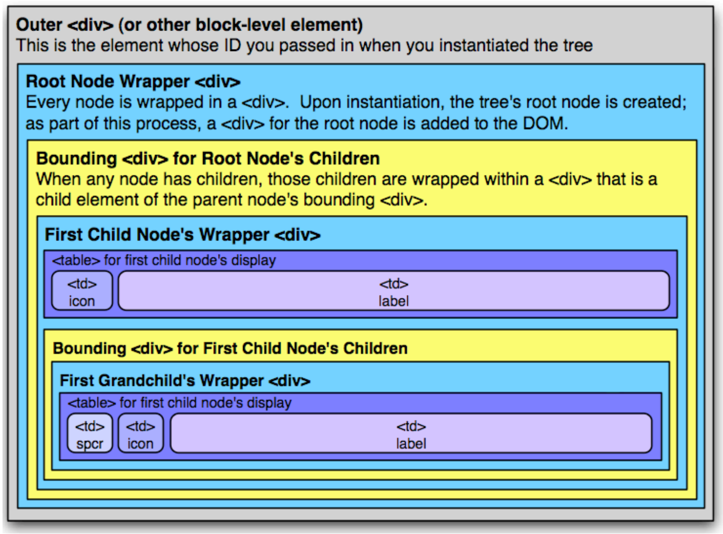
Arguments:
(1) **HTML:** A string containing markup for the node's label; no event handlers are provided by default for this markup.
(2) **Parent node:** See TextNode.
(3) **Expanded state:** See TextNode.
(4) **Has Icon:** Stipulates whether the expanded/contracted icon (and its horizontal space) should be rendered for this node.

Interesting Moments in TreeView see docs for complete list

Event	Fires...	Arguments
expand	...before a node expands; return false to cancel.	Node obj *expanding node*
collapse	...before a node collapses; return false to cancel	Node obj *collapsing node*
labelClick	...when text label clicked	Node obj *clicked nd*

TreeView events are Custom Events; subscribe to them by name using the following syntax: `tree.subscribe("expand", fn);`.

TreeView DOM Structure



Outer <div> (or other block-level element)
This is the element whose ID you passed in when you instantiated the tree

Root Node Wrapper <div>
Every node is wrapped in a <div>. Upon instantiation, the tree's root node is created; as part of this process, a <div> for the root node is added to the DOM.

Bounding <div> for Root Node's Children
When any node has children, those children are wrapped within a <div> that is a child element of the parent node's bounding <div>.

First Child Node's Wrapper <div>
<table> for first child node's display
<td> icon
<td> label
Bounding <div> for First Child Node's Children
First Grandchild's Wrapper <div>
<table> for first child node's display
<td> spcr
<td> icon
<td> label

Solutions:

Dynamically load child nodes:

```
fnLoadData = function(oNode, fnCallback) {
 //create child nodes for oNode
 var tmp = new YAHOO.widget.TextNode("lbl", oNode);
 fnCallback(); //then fire callback}
var tree = new Yahoo.widget.TreeView(targetEl);
tree.setDynamicLoad(fnLoadData);
var root = tree.getRoot();
var node1 = new YAHOO.widget.TextNode("1st", root);
tree.draw();
```

Dependencies

TreeView requires the YAHOO global object and the Event Utility.

YAHOO.widget. TreeView: Properties

id (str)
nodeCount (int)

YAHOO.widget. TreeView: Methods

collapseAll()
draw()
expandAll()
getNodesByProperty()
getRoot()
popNode(node**)** returns detached node, which can then be reinserted
removeChildren(node**)**
removeNode(node, b autorefresh**)**
setDynamicLoad(fn**)**

YAHOO.widget.Node: Properties

Inherited by Text, Menu, & HTML nodes

data (obj)
expanded (b)
hasIcon (b)
href (str)
iconMode (i)
labelStyle (s) Text/MenuNodes only. Use to style label area, e.g. for custom icons. Use `contentStyle` property for HTMLNodes
nextSibling (node obj)
parent (node obj)
previousSibling (node obj)
target (str)
tree (TreeView obj)

YAHOO.widget.Node: Methods

Inherited by Text, Menu, & HTML nodes

appendTo()
collapse()
collapseAll()
expand()
expandAll()
getEl() returns node's wrapper <div> element
getHTML() includes children
getNodeHTML() sans children
hasChildren()
insertBefore()
insertAfter()
isDynamic()
isRoot()
setDynamicLoad()
toggle()