YUI Library: TreeView

2009-9-8 **v2.8**

Simple Use Case

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

Places a TreeView 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,
   oConfig);
```

Arguments:

(1) **Element id or reference:** HTML ID or element reference for the element into which the Tree's DOM structure will be inserted. If the given element contains a series of nested ordered or unordered lists, they will be used to build the tree.

(2) **Object literal:** an object containing the full tree definition

Nodes: Text, Menu, HTML, Date

Node (abstract base class for all others)

```
YAHOO.widget.TextNode(obj | str oData, Node obj
   oParent);
```

Arguments:

(1) **Associated data:** A string containing the node label or an object containing values for any public properties of the node

(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.

TextNode (for simple labeled nodes):

If `oData` is a string it will be used as the label. If an object, it should contain a `label` property. If no `oData.href` is provided, clicking on the TextNode's will invoke the node's `expand` method.

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):

A string containing markup for the node's label or an object containing at least an `html` property

DateNode (for nodes containing dates):

Same as TextNode, will use Calendar widget for cell editing

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*
clickEvent	...when node is clicked	Node clicked and event
dblClickEvent	... when node is double clicked	Node clicked and event
enterKeyPressed	... when Enter key is pressed when a node has the focus	Node obj with the focus

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

TreeView object definition

```
var tree = new YAHOO.widget.TreeView("treeDiv1", [
 "label0",
 {type:"text", label: "label1", … , children: [… ]}, …
]);
```

Tree definition is an array containing node definitions. If node definition is a string, a TextNode is build. If an object, it should have a `type` property of `"text"`, `"menu"` or `"html"` or the full name of a node type (i.e.: `"HTMLNode"`) plus any other properties as would be provided to a Node constructor. Each node can have an optional `children` property with further node definitions.

TreeView from existing markup

```
<ul><li>List 0
 <ul><li>List 0-0</li>
       …
 </ul></li>
 <li><a href="www.elsewhere.com">elsewhere</a></li>
</ul>
```

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);
var node2 = new YAHOO.widget.TextNode("2nd", root);
node2.isLeaf = true; // leaf node, not dynamic
tree.render();
```

Dependencies

TreeView requires Yahoo, Dom and Event. Animation is optional; the the Calendar Control may be used for date editing.

YAHOO.widget.TreeView: Properties

id (str)

YAHOO.widget.TreeView: Methods

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

YAHOO.widget.Node: Properties

Inherited by Text, Menu, & HTML nodes

data (obj)
expanded (b)
hasIcon (b)
href (str)
isLeaf (b)
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)
editable (b)

YAHOO.widget.Node: Methods

Inherited by Text, Menu, & HTML nodes

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