Servoy Treeview plugin

The Treeview plugin is a tree-table control,
that uses dataset as data model.

In order to install it, you need to have at least Servoy 5,

and copy the treeview.jar to the installation's beans directory.

In order to use a dataset as a treeview model,
it has to satisfy the following requirements:

– the column names are internal identifiers used by the treeview control to read the dataset entries; all the data for a column are used by the identifier specified in the column name; here are all the possible identifiers

- **id** (object) [*required*] : row id
- **pid** (object) [*required*] : parent row id
- **treeColumn** (string) [*required*] : tree nodes label
- **column**1, ... n (string) [*optional*] : row columns label
- **icon** (string) [*optional*] : servoy media file name
- **editable** (string) [*optional*] : 'false' | null row columns editable flag, null means editable
- **type** (string) [*optional*] : 'folder'| null tree node type, null means it depends on the number of children

– the first row in the dataset contains the treeview header labels;

Short description of treeview functions :

- setDataSet(JSDataSet jsDataSet);

Set the treeview data model, see upper for detailed description of the dataset.

- refresh(restoreExpandedNodes);

Refresh the control and optionally restore expanded nodes, call this if you changed the data model.

- **expandNode(Object nodeId)**;

Expand the tree node with nodeId.

- isNodeExpanded(Object nodeId);

Returns true if the node is expanded.

- **collapseNode(Object nodeId);**

Collapse the tree node with nodeId.

- **setMultipleSelect(boolean bMultipleSelect);**

Set whatever multiple rows can be selected at the same time.

- **setSelectedNodes(Object[] nodesId);**

Mark the nodesId as selected in the treeview.

- **getSeletedNodes();**

Return the selected nodesId.

- **setHeaderVisible(boolean visible);**

Set whatever the treeview header is visible.

- setColumnWidth(String columnProperty, int width);

Set column width in pixels. The columnProperty is the treeview internal

identifier from the dataset column names

- setColumnHeaderTextFormat(String columnProperty, String textFormat);

Set column header text format. The columnProperty is the treeview internal

identifier from the dataset column names. The textFormat should be a

html block, where the model data is shown using $text.

Ex.: <html><font color='red'>$text</font></html>

- setColumnTextFormat(String columnProperty, String textFormat);

Set column text format. The columnProperty is the treeview internal

identifier from the dataset column names. The textFormat should be a

html block, where the model data is shown using $text.

Ex.: <html><font color='red'>$text</font></html>

-setCellTextFormatScript(Function cellTextFormatScript);

Set the cell text format to what the 'cellTextFormatScript' returns;
the script is called with the following arguments: 'columnid', 'rowid', 'celltext'

- setColumnCount(int columnCount);

Set number of visible columns, additional for the tree nodes column.

- setRowHeight(int height);

Set the treeview rows height in pixels.

- showTreeLines(boolean bShowLines);

Set whatever lines are drawn between the tree nodes.

- setStyleSheet(String styleSheet);

Set the treeview css style sheet, the element name should be 'treeview'.

Ex.: treeview { background-color: transparent; color: #0000FF; border-style: solid; }

- **getChildNodes(Object nodeId);**

Returns the child nodes id for nodeId.

- **getParentNode(Object nodeId);**

Returns the parent node id for nodeId.

- **getNodeLevel(Object nodeId);**

Returns the level of nodeID

- **getRootNodes();**

Returns the root node ids.

- **getElementType();**

Returns "TREEVIEW".

- **onNodeExpanded(Function f);**

Set node expand callback. arguments[0] contains the node id.

- **onNodeCollapsed(Function f);**

Set node collapse callback. arguments[0] contains the node id.

- **onNodeClicked(Function f);**

Set node clicked callback. arguments[0] contains the node id,
arguments[1] contains the column id.

- **onNodeSelected(Function f);**

Set node selected callback. arguments[0] contains the node id

- **onNodeRightClicked(Function f);**

Set node right click callback. arguments[0] contains the node id,
argument[1] , argument[2] the x, y coordinate of the right click

- onNodeIconClicked(Function f);

Set node icon clicked callback. arguments[0] contains the node id.

- onEditStarting(Function f);

Set row edit starting callback. arguments[0] contains the node id.

- onEditFinished(Function f);

Set row edit finished callback. arguments[0] contains the node id,

argument[1] the column identifier and argument[2] the entered value.

- onRightClick(Function f);

Set treeview right click callback. arguments[0] is null,
argument[1] , argument[2] the x, y coordinate of the right click