Add inputs Checkbox Radio Switch

Input Select Textarea Options tree Events

Methods

የስ

Getting started Components

Search..

About Population Aiax Animation Visibility Effects

Subtree functions Attributes Html, text and value Css and classes

Position and size Manipulation DataSet Events Utils

Containers Grid system Typography Tables Buttons Images

Checkbox

Lists

File input Input Input materia Keypad Rating

Radio Select Slider Double Slider

Spinner Tag input Textarea

App bar Bottom navigation Bottom sheet Menu Ribbon Menu Side bar Side navigation

Accordion Badge Carousel Cards Cube Counter

Charms Chat Image compare

Image magnifier Gravatar List ListView Master

NavView

Panels Progress & Activity

Stepper Splitter Tabs Tabs material Table

Tiles TreeView

Wizard

Dialogs Info box Hints Notify system Popovers

Windows Calendar

Calendar picker Date picker Time picker Countdown

Media

TreeView

A tree view that presents a hierarchical view of information. Create treeview simple and easy with Metro 4.

About

To creating treeview we use element with role data-role="treeview" . Nodes are defined with element. To create subtree just add an <11> inside the <1i>.

```
data-role="treeview">
                                                      đ
 :...
   <1i>...</1i>
```

Add node

Node is a li> with special data-* attributes. The attributes are define: icon and caption of node.

```
→ Favorites (3)

  m Projects
  Downloads
  Desktop
OneDrive (2)
  Documents
 > Projects (4)
data-role="treeview">
                                                                ረት
  </span>" data-caption="Favorites">
    </span>" data-caption="Projects">
      data-icon="<span class='mif-download'></span>" data-caption="Downloads">
      " data-caption="Desktop">
    </span>" data-caption="OneDrive">
    data-caption="Projects" data-collapsed="true">
        data-caption="Web">
          data-caption="iOS">
        </111>
```

Add inputs

You can add input controls into treeview nodes. A **prerequisite** is the need to put each <code>input</code> in a separate <code>.</code>.

Checkbox

```
<input type="checkbox" data-role="checkbox">
                                                                                                    đ
```

Radio

```
<input type="radio" data-role="radio">
                                                                                                 4
```

Switch

```
<input type="checkbox" data-role="switch">
                                                                                                   đ
```

Input

```
<input type="text" data-role="input">
                                                                                                đ
```

Select

```
<select data-role="select">...</select>
                                                                                                  đ
```

Video player
Audio player
Tools
Collapse
Color module
Draggable
Dropdown
Form validator
Hotkeys
Micro templates
Ripple
Storage
Session storage
Sorter
Touch and swipe

Utilities Animations Additional

```
<textarea data-role="textarea"></textarea>
                                                                                                                   đ

    Checkboxes (4)

       Play animation
       ✓ Play sound

    Radios (4)

    Play animation

       Play sound

    Switches (4)

             Play animation
             Play sound
             Disabled

    Disabled che

   Inputs (3)
        Value 1
 ch
     <input type="checkbox" data-role="checkbox" data-caption="Play animation">
             cli><input type="checkbox" data-role="checkbox" data-caption="Play sound" checked><input type="checkbox" data-role="checkbox" data-caption="Play sound" checked><input type="checkbox" data-role="checkbox" data-caption="Disabled" disabled><input type="checkbox" data-role="checkbox" data-caption="Disabled checked" checked disabled>
     </1i>
     <u1>
             <ii>>input type="radio" name="r1" data-role="radio" data-caption="Play animation">
             <input type="radio" name="r1" data-role="radio" data-caption="Play sound" checked>
             <ii><input type="radio" data-role="radio" data-caption="Disabled" disabled>
             <ii><iiput type="radio" data-role="radio" data-caption="Disabled checked" checked disabled>
         <input type="checkbox" data-role="switch" data-caption="Play animation">
             <input type="checkbox" data-role="switch" data-caption="Play sound" checked>
             <input type="checkbox" data-role="switch" data-caption="Disabled" disabled>
             <ii><ii>input type="checkbox" data-role="switch" data-caption="Disabled checked" checked disabled>
         <u1>
             style="width: 220px"><input type="text" data-role="input">
             <option value="1">Value 1</option>
<option value="2">Value 2</option>
                    <option value="3">Value 3</option>
             </1i>
             style="width: 220px"><textarea data-role="textarea"></textarea>
         </1i>
 Options tree
```

If you need to create options tree, you can deal with this easy with Metro 4 treeview component. You do not need any special actions, just add the checkboxes to nodes. Indeterminate state for checkboxes will be enabled automatically.

```
✓ □ Play video
      AVI
       MPEG
     ✓ VIDX
       MP4
     ✓ XVID
    Play audio
   Subtitles
       English
     ✓ Ukrainian
     Dutch
data-role="treeview">
                                                                                                                          4
    <1i>>
       <input type="checkbox" data-role="checkbox" data-caption="Play video" title="">
           <input type="checkbox" data-role="checkbox" data-caption="AVI" title="">
            <input type="checkbox" data-role="checkbox" data-caption="MPEG" title="">
            <ii>checked type="checkbox" data-role="checkbox" data-caption="VIDX" title="">
           <in>your type="checkbox" data-role="checkbox" data-caption="MP4" title="">
<in>your type="checkbox" data-role="checkbox" data-caption="MP4" title="">
```

<input type="checkbox" data-role="checkbox" data-caption="Play audio" title="">

</1i>

Events

When treeview works, it generated the numbers of events. You can use callback for this events to behavior with treeview.

Event	Data-*	Desc
onNodeClick(node, tree)	data-on-node-click	Fired when user click on node caption
onNodeDblClick(node, tree)	data-on-node-dblclick	Fired when user dblclick on node caption
onNodeInsert(node, tree)	data-on-node-insert	Fired when node was inserted
onNodeDelete(node, tree)	data-on-node-delete	Fired when node was deleted
onNodeClean(node, tree)	data-on-node-clean	Fired when node was cleaned
onCheckClick(state, check, node, tree)	data-on-check-click	Fired when user click on checkbox
onRadioClick(state, check, node, tree)	data-on-radio-click	Fired when user click on radio input
onExpandNode(node, tree)	data-on-expand-node	Fired when node was expanded
onCollapseNode(node, tree)	data-on-collapse-node	Fired when node was collapsed
onTreeviewCreate(node, tree)	data-on-treeview-create	Fired when tree was created

Methods

You can use treeview methods to interact with the component.

- toggleNode(node) toggle node state
- addTo(node, data) add child node
- insertBefore(node, data) add new node before specified node
- insertAfter(node, data) add new node after specified node
- del(node) delete node from tree
- clean(node) clean node

```
Add node Add subnode Add checkbox Insert before Insert after Delete Clear
  <button class="button" onclick="</pre>
                                                                                                                                                                                                                                                                                                                                                                4
           $('#tv_1').data('treeview').addTo(null, {
                      caption: 'New node',
                      icon: '<span class=\'mif-air\'></span>'
">Add node</button>
<button class="button" onclick="
    $('\data('treeview').addTo(\(\frac{\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\text{"\data('\)\capex('\text{"\data('\)\)}\)}\)}\).") \dispir\dispir\dispir\dispir\)} \dispir\dispir\dispir\)}\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dispir\dir\dispir\dispir\dispir\dispir\dir)}} \dispir\dispir\disp
">Add subnode</button>
<button class="button" onclick="</pre>
            var tv = $('#tv 1');
            tv.data('treeview').addTo(tv.find('.current'), {
                     html: '<input data-role=checkbox data-caption=Checkbox>'
">Add checkbox</button>
<button class="button" onclick="</pre>
           ('\#tv_1').data('treeview').insertBefore(('\#tv_1').find('.current'), {
                   caption: 'Before node'
">Insert before</button>
<button class="button" onclick="</pre>
           \\$('$tv_1').data('treeview').insertAfter($('$tv_1').find('.current'), \\
                   caption: 'After node'
">Insert after</button>
<button class="button" onclick="</pre>
           $('#tv_1').data('treeview').del($('#tv_1').find('.current'))
">Delete</button>
<button class="button" onclick="
$('\div_1').data('\treeview').clean(\div(\div_1').find('.current'))</pre>
">Clear</button>
data-role="treeview" id="tv_1">
```