

7 Navigation

7.1 Menu

The Menu UI component is a panel with clickable items. A click on an item opens a drop-down menu, which can contain several submenus.

We can display Menu *items* from the items array or a *dataSource*.

Configure the Menu

We can use the *orientation* property to specify whether the Menu has horizontal or vertical orientation. Also we can use the *animation* property to specify the *type*, *delay*, *duration*, and other options of *show* and *hide* menu actions.

Configure the Submenus

Clicking or a hovering a Menu item opens a drop-down menu that can contain several *submenus*. To specify drop-down menu mode (*'onClick' | 'onHover'*), use the *showSubMenuMode* property. If we need only to specify the first level of drop-down menus, use the *showFirstSubMenuMode* property.

If we want to hide the submenu when the mouse pointer leaves it, set the *hideSubMenuOnMouseLeave* property to **true**.

Customize Item Appearance

We can define specific fields in the item data objects to change the appearance of an item. For example, use the *icon* property to supply items with icons. Define an *itemTemplate* to customize item appearance.

7.2 TreeView

The TreeView component displays a tree of text nodes from a local or remote source.

Bind the TreeView to Data

We use *dataSource* or *items* properties to bind the data.

The TreeView supports *plain* and *hierarchical* data structures. To use plain data, set the *dataStructure* property to *"plain"*. In this case, each data object should contain the *id*, *text*, and *parentId* fields. If objects use custom field names, use the *keyExpr*, *displayExpr*, and *parentIdExpr* properties to specify them.

Each object in the TreeView's *hierarchical* data structure should include the following fields: *id*(Unique item identifier), *text*(Text displayed by the item), *items*(Nested objects (optional))

We can respectively use the *keyExpr*, *displayExpr*, and *itemsExpr* properties to specify custom names for the above-mentioned fields.

Customize Node Appearance

We can use the *itemTemplate* property to specify a template for all. If we want to add a template for a specific node, we can use the template node's property. This setting overrides the global template.

We can also use additional fields to customize node appearance. For example, we can enable the *expanded* field for those nodes that should be expanded on startup.

Search Data

The `TreeView` can display a search bar that allows users to search nodes. Enable the *searchEnabled* property to add the search bar. We can use the *searchMode* property to specify whether nodes should *contain (default)*, *start with*, or *match the search string*.

Select Nodes

To configure node selection, we can use the following properties:

selectByClick : Enables selection by a click.

showCheckBoxesMode : Adds checkboxes if we set this property to "*none*",

"*normal*" or "*selectAll*". The latter mode additionally displays the "*Select All*" checkbox at the top of the `TreeView`.

selectionMode : Sets the selection mode to "*single*" or "*none*" (default).

onItemSelectionChanged : Specifies a function that is executed when a single node is selected or deselected.