

Package ‘muiTreeView’

October 12, 2026

Type Package

Title 'MUI X Tree View' for 'shiny' Apps and 'Quarto'

Version 0.2.0

Description

Wraps the 'MUI X Tree View' React components <<https://mui.com/x/react-tree-view/>> for use in 'shiny' applications and 'quarto' documents, letting users navigate hierarchical data with nested levels that can be expanded and collapsed. Bundles the MIT-licensed community edition of the '@mui/x-tree-view' JavaScript library (the commercial 'Pro' tier is not included).

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Depends R (>= 4.1)

Imports muiMaterial (>= 0.2.1), htmltools, shiny.react (>= 0.4.0),
utils

Suggests shiny, testthat (>= 3.0.0)

RoxygenNote 7.3.3

URL <https://felixluginbuhl.com/muiTreeView/>,
<https://github.com/lgnbhl/muiTreeView>

BugReports <https://github.com/lgnbhl/muiTreeView/issues>

Config/testthat/edition 3

NeedsCompilation no

Author Felix Luginbuhl [aut, cre, cph] (ORCID:
<<https://orcid.org/0009-0008-6625-2899>>),
MUI [cph] (Copyright holder of the bundled '@mui/x-tree-view',
'@mui/x-internals', '@base-ui/utils' and '@mui/system' (subpath
imports) JavaScript libraries; see LICENSE.note),
Meta Platforms, Inc. and affiliates [cph] (Copyright holder of the
bundled 'react-is', 'prop-types' and 'use-sync-external-store'
JavaScript libraries; 'react' and 'react-dom' are declared as peer

dependencies and provided at runtime by 'shiny.react'),
React Community [cph] (Copyright holder of the bundled
'react-transition-group' JavaScript library; see LICENSE.note),
Luke Edwards [cph] (Copyright holder of the bundled 'clsx' JavaScript
library; see LICENSE.note),
Reselect contributors [cph] (Copyright holder of the bundled 'reselect'
JavaScript library; see LICENSE.note)

Maintainer Felix Luginbuhl <felix.luginbuhl@protonmail.ch>

Repository CRAN

Date/Publication 2026-10-12 16:40:02 UTC

Contents

asMuiTree	2
muiTreeViewDependency	3
print.muiTreeView	4
RichTreeView	4
RichTreeViewRoot	7
SimpleTreeView	8
SimpleTreeViewRoot	10
TreeItem	10
TreeItemCheckbox	12
TreeItemContent	12
TreeItemGroupTransition	13
TreeItemIcon	13
TreeItemIconContainer	14
TreeItemLabel	14
TreeItemLabelInput	15
TreeItemProvider	15
TreeItemRoot	16
TreeViewCollapseIcon	17
TreeViewExpandIcon	17

Index	18
--------------	-----------

asMuiTree	Create choice structure for [RichTreeView()]
-----------	--

Description

This function is adapted from ‘shinyWidgets::create_tree()’. A big thank you to the authors of the shinyWidgets for creating the ‘create_tree()’ function.

Usage

```
asMuiTree(data, levels = names(data), levels_id = NULL, ...)
```

Arguments

data	A 'data.frame'.
levels	Variables identifying hierarchical levels, values of those variables will be used as text displayed.
levels_id	Variable to use as ID for nodes. Careful! Spaces are not allowed in IDs.
...	Additional arguments.

Value

a 'list' that can be used in [RichTreeView()].

See Also

'shinyWidgets::create_tree()'

`muiTreeViewDependency` *Mui X Tree View JS dependency*

Description

Mui X Tree View JS dependency

Usage

```
muiTreeViewDependency()
```

Value

A list of HTML dependency objects (`htmltools::htmlDependency`) that load the bundled MUI X Tree View JavaScript and its shared MUI Material runtime. The `muiMaterial` dependency is listed first so its bundle (which provides `react`, `react-dom`, `@mui/material`, `@mui/system` and the emotion runtime that the tree view consumes as externals) loads before the tree view script. Attach it to custom HTML when building a tree view by hand.

Examples

```
# Inspect the dependencies attached to every muiTreeView element.  
muiTreeViewDependency()
```

```
print.muiTreeView
```

Print muiTreeView components

Description

When called interactively, renders the component in the IDE viewer panel. Otherwise, falls back to standard shiny.tag printing (raw HTML text).

Usage

```
## S3 method for class 'muiTreeView'
print(x, browse = interactive(), ...)
```

Arguments

x	A muiTreeView object (also inherits shiny.tag).
browse	Whether to render in viewer. Defaults to TRUE in interactive sessions.
...	Additional arguments passed to print.

Value

Invisibly returns x.

Examples

```
tree <- RichTreeView(
  items = list(list(id = "1", label = "Item"))
)
# browse = FALSE prints the underlying HTML instead of opening the viewer.
print(tree, browse = FALSE)
```

```
RichTreeView
```

RichTreeView

Description

<https://mui.com/x/api/tree-view/rich-tree-view/>

Usage

```
RichTreeView(...)
```

Arguments

...	Props to pass to the component.
-----	---------------------------------

Details

- `apiRef` { `current`?: { `focusItem`?: func, `getItem`?: func, `getItemDOMElement`?: func, `getItemOrderedChildren`?: func }
Default is - The ref object that allows Tree View manipulation. Can be instantiated with `useTreeViewApiRef()`.
- `checkboxSelection` bool
Default is FALSE If true, the Tree View renders a checkbox at the left of its label that allows selecting it.
- `classes` object
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `defaultExpandedItems` Array string
Default is [] Expanded item ids. Used when the item's expansion is not controlled.
- `defaultSelectedItems` any
Default is [] Selected item ids. (Uncontrolled) When `multiSelect` is true this takes an array of strings; when false (default) a string.
- `disabledItemsFocusable` bool
Default is FALSE If true, will allow focus on disabled items.
- `disableSelection` bool
Default is FALSE If true selection is disabled.
- `expandedItems` Array string
Default is - Expanded item ids. Used when the item's expansion is controlled.
- `expansionTrigger` 'content' | 'iconContainer'
Default is 'content' The slot that triggers the item's expansion when clicked.
- `getItemChildren` func
Default is (item) = item.children Used to determine the children of a given item. Signature: function(item: R) = Arrayitem The item to check.
- `getItemId` func
Default is (item) = item.id Used to determine the id of a given item. Signature: function(item: R) = stringitem The item to check.
- `getItemLabel` func
Default is (item) = item.label Used to determine the string label for a given item. Signature: function(item: R) = stringitem The item to check.
- `id` string
Default is - This prop is used to help implement the accessibility logic. If you don't provide this prop. It falls back to a randomly generated id.
- `isItemDisabled` func
Default is - Used to determine if a given item should be disabled. Signature: function(item: R) = booleanitem The item to check.
- `isItemEditable` func | bool
Default is () = FALSE Determine if a given item can be edited.
- `itemChildrenIndentation` number | string
Default is 12px Horizontal indentation between an item and its children. Examples: 24, "24px", "2rem", "2em".

- **multiSelect** bool
Default is FALSE If true, ctrl and shift will trigger multiselect.
- **onExpandedItemsChange** func
Default is - Callback fired when Tree Items are expanded/collapsed. Signature: `function(event: React.SyntheticEvent, itemIds: array) = void` event The DOM event that triggered the change. Can be null when the change is caused by the `publicAPI.setItemExpansion()` method. `itemIds` The ids of the expanded items.
- **onItemClick** func
Default is - Callback fired when the content slot of a given Tree Item is clicked. Signature: `function(event: React.MouseEvent, itemId: string) = void` event The DOM event that triggered the change. `itemId` The id of the focused item.
- **onItemExpansionToggle** func
Default is - Callback fired when a Tree Item is expanded or collapsed. Signature: `function(event: React.SyntheticEvent | null, itemId: array, isExpanded: boolean) = void` event The DOM event that triggered the change. Can be null when the change is caused by the `publicAPI.setItemExpansion()` method. `itemId` The `itemId` of the modified item. `isExpanded` true if the item has just been expanded, false if it has just been collapsed.
- **onItemFocus** func
Default is - Callback fired when a given Tree Item is focused. Signature: `function(event: React.SyntheticEvent | null, itemId: string) = void` event The DOM event that triggered the change. Warning: This is a generic event not a focus event. `itemId` The id of the focused item.
- **onItemLabelChange** func
Default is - Callback fired when the label of an item changes. Signature: `function(itemId: TreeViewItemId, newLabel: string) = void` `itemId` The id of the item that was edited. `newLabel` The new label of the items.
- **onItemSelectionToggle** func
Default is - Callback fired when a Tree Item is selected or deselected. Signature: `function(event: React.SyntheticEvent, itemId: array, isSelected: boolean) = void` event The DOM event that triggered the change. Can be null when the change is caused by the `publicAPI.setItemSelection()` method. `itemId` The `itemId` of the modified item. `isSelected` true if the item has just been selected, false if it has just been deselected.
- **onSelectedItemsChange** func
Default is - Callback fired when Tree Items are selected/deselected. Signature: `function(event: React.SyntheticEvent, itemIds: Array | string) = void` event The DOM event that triggered the change. Can be null when the change is caused by the `publicAPI.setItemSelection()` method. `itemIds` The ids of the selected items. When `multiSelect` is true, this is an array of strings; when false (default) a string.
- **selectedItems** any
Default is - Selected item ids. (Controlled) When `multiSelect` is true this takes an array of strings; when false (default) a string.
- **selectionPropagation** { `descendants?: bool, parents?: bool` }
Default is `parents: FALSE, descendants: FALSE` When `selectionPropagation.descendants` is set to true.- Selecting a parent selects all its descendants automatically. - Deselecting a parent deselects all its descendants automatically. When `selectionPropagation.parents` is set to true.-

Selecting all the descendants of a parent selects the parent automatically. - Deselecting a descendant of a selected parent deselects the parent automatically. Only works when multiSelect is true. On the SimpleTreeView / , only the expanded items are considered (since the collapsed item are not passed to the Tree View component at all)

- slotProps object
Default is The props used for each component slot.
- slots object
Default is Overridable component slots. See Slots API below for more details.
- sx Array func | object | bool | func | object
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

RichTreeViewRoot	<i>RichTreeViewRoot</i>
------------------	-------------------------

Description

Styled root element of RichTreeView(). See <https://mui.com/x/react-tree-view/rich-tree-view/customization/>.

Usage

```
RichTreeViewRoot(...)
```

Arguments

... Props to pass to the component.

Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

SimpleTreeView

*SimpleTreeView***Description**

<https://mui.com/x/api/tree-view/simple-tree-view/>

Usage

```
SimpleTreeView(...)
```

Arguments

... Props to pass to the component.

Details

- `apiRef` { `current`?: { `focusItem`?: func, `getItem`?: func, `getItemDOMElement`?: func, `getItemOrderedChild`?: func } }
Default is - The ref object that allows Tree View manipulation. Can be instantiated with `useTreeViewApiRef()`.
- `checkboxSelection` bool
Default is FALSE If true, the Tree View renders a checkbox at the left of its label that allows selecting it.
- `children` node
Default is - The content of the component.
- `classes` object
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `defaultExpandedItems` Array string
Default is [] Expanded item ids. Used when the item's expansion is not controlled.
- `defaultSelectedItems` any
Default is [] Selected item ids. (Uncontrolled) When `multiSelect` is true this takes an array of strings; when false (default) a string.
- `disabledItemsFocusable` bool
Default is FALSE If true, will allow focus on disabled items.
- `disableSelection` bool
Default is FALSE If true selection is disabled.
- `expandedItems` Array string
Default is - Expanded item ids. Used when the item's expansion is controlled.
- `expansionTrigger` 'content' | 'iconContainer'
Default is 'content' The slot that triggers the item's expansion when clicked.
- `id` string
Default is - This prop is used to help implement the accessibility logic. If you don't provide this prop. It falls back to a randomly generated id.

- `itemChildrenIndentation` `number | string`
Default is 12px Horizontal indentation between an item and its children. Examples: 24, "24px", "2rem", "2em".
- `multiSelect` `bool`
Default is FALSE If true, ctrl and shift will trigger multiselect.
- `onExpandedItemsChange` `func`
Default is - Callback fired when Tree Items are expanded/collapsed. Signature: `function(event: React.SyntheticEvent, itemIds: array) = void` event The DOM event that triggered the change. Can be null when the change is caused by the `publicAPI.setItemExpansion()` method. `itemIds` The ids of the expanded items.
- `onItemClick` `func`
Default is - Callback fired when the content slot of a given Tree Item is clicked. Signature: `function(event: React.MouseEvent, itemId: string) = void` event The DOM event that triggered the change. `itemId` The id of the focused item.
- `onItemExpansionToggle` `func`
Default is - Callback fired when a Tree Item is expanded or collapsed. Signature: `function(event: React.SyntheticEvent | null, itemId: array, isExpanded: boolean) = void` event The DOM event that triggered the change. Can be null when the change is caused by the `publicAPI.setItemExpansion()` method. `itemId` The `itemId` of the modified item. `isExpanded` true if the item has just been expanded, false if it has just been collapsed.
- `onItemFocus` `func`
Default is - Callback fired when a given Tree Item is focused. Signature: `function(event: React.SyntheticEvent | null, itemId: string) = void` event The DOM event that triggered the change. Warning: This is a generic event not a focus event. `itemId` The id of the focused item.
- `onItemSelectionToggle` `func`
Default is - Callback fired when a Tree Item is selected or deselected. Signature: `function(event: React.SyntheticEvent, itemId: array, isSelected: boolean) = void` event The DOM event that triggered the change. Can be null when the change is caused by the `publicAPI.setItemSelection()` method. `itemId` The `itemId` of the modified item. `isSelected` true if the item has just been selected, false if it has just been deselected.
- `onSelectedItemsChange` `func`
Default is - Callback fired when Tree Items are selected/deselected. Signature: `function(event: React.SyntheticEvent, itemIds: Array | string) = void` event The DOM event that triggered the change. Can be null when the change is caused by the `publicAPI.setItemSelection()` method. `itemIds` The ids of the selected items. When `multiSelect` is true, this is an array of strings; when false (default) a string.
- `selectedItems` `any`
Default is - Selected item ids. (Controlled) When `multiSelect` is true this takes an array of strings; when false (default) a string.
- `selectionPropagation` `{ descendants?: bool, parents?: bool }`
Default is `parents: FALSE, descendants: FALSE` When `selectionPropagation.descendants` is set to true.- Selecting a parent selects all its descendants automatically. - Deselecting a parent deselects all its descendants automatically. When `selectionPropagation.parents` is set to true.- Selecting all the descendants of a parent selects the parent automatically. - Deselecting a descendant of a selected parent deselects the parent automatically. Only works when `multiSelect`

is true. On the SimpleTreeView / , only the expanded items are considered (since the collapsed item are not passed to the Tree View component at all)

- slotProps object
Default is - The props used for each component slot.
- slots object
Default is - Overridable component slots. See Slots API below for more details.
- sx Array func | object | bool | func | object
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the sx page for more details.

Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

SimpleTreeViewRoot	<i>SimpleTreeViewRoot</i>
--------------------	---------------------------

Description

Styled root element of SimpleTreeView(). See <https://mui.com/x/react-tree-view/simple-tree-view/customization/>.

Usage

```
SimpleTreeViewRoot(...)
```

Arguments

... Props to pass to the component.

Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

TreeItem	<i>TreeItem</i>
----------	-----------------

Description

<https://mui.com/x/api/tree-view/tree-item/>

Usage

```
TreeItem(...)
```

Arguments

... Props to pass to the component.

Details

- `itemId` string
Default is - The id of the item. Must be unique.
- `children` any
Default is - The content of the component.
- `classes` object
Default is - Override or extend the styles applied to the component. See CSS classes API below for more details.
- `disabled` bool
Default is FALSE. If true, the item is disabled.
- `id` string
Default is - The id attribute of the item. If not provided, it will be generated.
- `label` node
Default is - The label of the item.
- `onBlur` func
Default is - Callback fired when the item root is blurred.
- `onFocus` unsupportedProp
Default is - This prop isn't supported. Use the `onItemFocus` callback on the tree if you need to monitor an item's focus.
- `onKeyDown` func
Default is - Callback fired when a key is pressed on the keyboard and the tree is in focus.
- `slotProps` object
Default is The props used for each component slot.
- `slots` object
Default is Overridable component slots. See Slots API below for more details.
- `sx` Array | func | object | bool | func | object
Default is - The system prop that allows defining system overrides as well as additional CSS styles. See the `sx` page for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

TreeItemCheckbox	<i>TreeItemCheckbox</i>
------------------	-------------------------

Description

Checkbox slot of a tree item, rendered when `checkboxSelection = TRUE`. Building block for customizing `TreeItem()`. See <https://mui.com/x/react-tree-view/tree-item-customization/>.

Usage

```
TreeItemCheckbox(...)
```

Arguments

... Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

TreeItemContent	<i>TreeItemContent</i>
-----------------	------------------------

Description

Content slot of a tree item (the clickable row). Building block for customizing `TreeItem()`. See <https://mui.com/x/react-tree-view/tree-item-customization/>.

Usage

```
TreeItemContent(...)
```

Arguments

... Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

TreeItemGroupTransition

TreeItemGroupTransition

Description

Transition slot wrapping the children of a tree item. Building block for customizing TreeItem(). See <https://mui.com/x/react-tree-view/tree-item-customization/>.

Usage

```
TreeItemGroupTransition(...)
```

Arguments

... Props to pass to the component.

Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

TreeItemIcon

TreeItemIcon

Description

<https://mui.com/x/api/tree-view/tree-item-icon/>

Usage

```
TreeItemIcon(...)
```

Arguments

... Props to pass to the component.

Details

- slotProps object
Default is The props used for each component slot.
- slots object
Default is Overridable component slots. See Slots API below for more details.

Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

TreeItemIconContainer	<i>TreeItemIconContainer</i>
-----------------------	------------------------------

Description

Icon container slot of a tree item (expand/collapse icon). Building block for customizing TreeItem(). See <https://mui.com/x/react-tree-view/tree-item-customization/>.

Usage

```
TreeItemIconContainer(...)
```

Arguments

... Props to pass to the component.

Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

TreeItemLabel	<i>TreeItemLabel</i>
---------------	----------------------

Description

Label slot of a tree item. Building block for customizing TreeItem(). See <https://mui.com/x/react-tree-view/tree-item-customization/>.

Usage

```
TreeItemLabel(...)
```

Arguments

... Props to pass to the component.

Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

TreeItemLabelInput	<i>TreeItemLabelInput</i>
--------------------	---------------------------

Description

Input rendered in place of the label when label editing is enabled with `isItemEditable`. See <https://mui.com/x/react-tree-view/rich-tree-view/editing/>.

Usage

```
TreeItemLabelInput(...)
```

Arguments

... Props to pass to the component.

Details

- `value` string
Default is - The value of the input.
- `onChange` func
Default is - Callback fired when the value changes.
- `onKeyDown` func
Default is - Callback fired when a key is pressed.
- `onBlur` func
Default is - Callback fired when the input is blurred.
- `autoFocus` bool
Default is - If TRUE, the input is focused on mount.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

TreeItemProvider	<i>TreeItemProvider</i>
------------------	-------------------------

Description

<https://mui.com/x/api/tree-view/tree-item-provider/>
Provides the context of a tree item when building a custom tree item.

Usage

```
TreeItemProvider(...)
```

Arguments

... Props to pass to the component.

Details

- children node
Default is - The content of the component.
- itemId string
Default is - The id of the item.
- id string
Default is - The id attribute of the item.

Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

TreeItemRoot	<i>TreeItemRoot</i>
--------------	---------------------

Description

Root slot of a tree item. Building block for customizing TreeItem(). See <https://mui.com/x/react-tree-view/tree-item-customization/>.

Usage

TreeItemRoot(...)

Arguments

... Props to pass to the component.

Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

TreeViewCollapseIcon	<i>TreeViewCollapseIcon</i>
----------------------	-----------------------------

Description

Default icon shown on expanded items (click to collapse). See <https://mui.com/x/react-tree-view/rich-tree-view/customization/>.

Usage

```
TreeViewCollapseIcon(...)
```

Arguments

...	Props to pass to the component.
-----	---------------------------------

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

TreeViewExpandIcon	<i>TreeViewExpandIcon</i>
--------------------	---------------------------

Description

Default icon shown on collapsed items (click to expand). See <https://mui.com/x/react-tree-view/rich-tree-view/customization/>.

Usage

```
TreeViewExpandIcon(...)
```

Arguments

...	Props to pass to the component.
-----	---------------------------------

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

Index

`asMuiTree`, [2](#)

`muiTreeViewDependency`, [3](#)

`print.muiTreeView`, [4](#)

`RichTreeView`, [4](#)

`RichTreeViewRoot`, [7](#)

`SimpleTreeView`, [8](#)

`SimpleTreeViewRoot`, [10](#)

`TreeItem`, [10](#)

`TreeItemCheckbox`, [12](#)

`TreeItemContent`, [12](#)

`TreeItemGroupTransition`, [13](#)

`TreeItemIcon`, [13](#)

`TreeItemIconContainer`, [14](#)

`TreeItemLabel`, [14](#)

`TreeItemLabelInput`, [15](#)

`TreeItemProvider`, [15](#)

`TreeItemRoot`, [16](#)

`TreeViewCollapseIcon`, [17](#)

`TreeViewExpandIcon`, [17](#)