

# Tree View

A tree view widget presents a hierarchical list.

Tree views can be used to represent a file system navigator displaying folders and files, an item representing a folder can be expanded to reveal the contents of the folder, which may be files, folders, or both.

```
{{"component": "modules/components/ComponentLinkHeader.js"}}
```

## Basic tree view

```
{{"demo": "FileSystemNavigator.js"}}
```

## Multi-selection

Tree views also support multi-selection.

```
{{"demo": "MultiSelectTreeView.js"}}
```

## Controlled tree view

The tree view also offers a controlled API.

```
{{"demo": "ControlledTreeView.js"}}
```

## Rich object

While the `TreeView` / `TreeItem` component API maximizes flexibility, an extra step is needed to handle a rich object.

Let's consider a data variable with the following shape, recursion can be used to handle it.

```
const data = {
  id: 'root',
  name: 'Parent',
  children: [
    {
      id: '1',
      name: 'Child - 1',
    },
    // ...
  ],
};
```

```
{{"demo": "RichObjectTreeView.js", "defaultCodeOpen": false}}
```

## ContentComponent prop

You can use the `ContentComponent` prop and the `useTreeItem` hook to further customize the behavior of the `Treeltem`.

Such as limiting expansion to clicking the icon:

```
{{"demo": "IconExpansionTreeView.js", "defaultCodeOpen": false}}
```

Or increasing the width of the state indicator:

```
{{"demo": "BarTreeView.js", "defaultCodeOpen": false}}
```

## Customization

### Custom icons, border and animation

```
{{"demo": "CustomizedTreeView.js"}}
```

### Gmail clone

```
{{"demo": "GmailTreeView.js"}}
```

## Disabled tree items

```
{{"demo": "DisabledTreeItems.js"}}
```

The behavior of disabled tree items depends on the `disabledItemsFocusable` prop.

If it is false:

- Arrow keys will not focus disabled items and, the next non-disabled item will be focused.
- Typing the first character of a disabled item's label will not focus the item.
- Mouse or keyboard interaction will not expand/collapse disabled items.
- Mouse or keyboard interaction will not select disabled items.
- Shift + arrow keys will skip disabled items and, the next non-disabled item will be selected.
- Programmatic focus will not focus disabled items.

If it is true:

- Arrow keys will focus disabled items.
- Typing the first character of a disabled item's label will focus the item.
- Mouse or keyboard interaction will not expand/collapse disabled items.
- Mouse or keyboard interaction will not select disabled items.
- Shift + arrow keys will not skip disabled items but, the disabled item will not be selected.
- Programmatic focus will focus disabled items.

## Accessibility

(WAI-ARIA: <https://www.w3.org/TR/wai-aria-practices/#TreeView>)

The component follows the WAI-ARIA authoring practices.

To have an accessible tree view you must use `aria-labelledby` or `aria-label` to reference or provide a label on the `TreeView`, otherwise screen readers will announce it as "tree", making it hard to understand the context of a specific tree item.