Lists

[Design System Image]

Lists are frequently used throughout Brightlayer UI applications. They are intended to represent groups of similar logical items, such as a device, an alarm, or a setting. They can display data from simple name-value pairs up to complex timelines or device lists.

[MaterialDesignDescription Component - Interactive React component]

[TOC Component - Interactive React component]

Common Use Cases

[Design System Image]

You can use lists in many ways, including for:

- Details panels
- Timelines (Alarms / Notifications)
- Inventories / List of Devices
- Contacts
- Logs
- Settings
- File trees

Lists should **NOT** be used where:

- Data represents different logical items (e.g., don't mix alarms and devices in a single list)

For information about the differences between lists and tables, check out our Tables pattern.

Supporting UI Elements

Lists typically appear inside of a panel (mobile) or a card (desktop). Occasionally, an entire screen may use a single list as its main body content.

[Design System Image]

Variations

Basic Lists

[Design System Image]

[DemoCard Component - Interactive React component]

The simplest type of list simply presents a single line (or multiple lines) of text for each element in the collection.

Lists with Status

[Design System Image]

[DemoCard Component - Interactive React component]

Commonly in Brightlayer UI applications, list items also show a visual indication of an item's current status, such as online, offline, or an error condition. You can indicate this status by using avatars and status stripes with Brightlayer UI's status color palette.

[Design System Image]

If avatars and status stripes do not work well based on the context, such as in the body of a scorecard component, you can color-code your list item icons and labels to indicate status. However, for accessibility concerns, you should never rely _solely_ on the color to convey the message.

[Design System Image]

Using tags is another way to display list item status. They typically appear at the very end of a list item.

[Design System Image]

Timestamps

Depending on the data being displayed, you can add additional UI elements to the list items to show more information. One of the most common such elements is a timestamp, which usually occurs in timelines or device logs. If you have a large number of items in a list with timestamps, you should group them by date to make them easier to scan.

[Design System Image]

Behaviors

Lists can be interactive. In general, there are two types of actions: global list actions and local list item actions.

[DemoCard Component - Interactive React component]

Global List Actions

Global list actions are the actions that manipulate the entire list. The buttons typically sit in an Appbar, a panel header, or a button panel (desktop-only).

[Design System Image]

Common global list actions include:

- Adding a new list item
- Sorting and filtering the list
- Toggling a different view, such as a grid view
- Exporting or downloading the list data

[DemoCard Component - Interactive React component]

[DemoCard Component - Interactive React component]

Drag and Drop

[Design System Image]

[DemoCard Component - Interactive React component]

Drag and drop is a special case for list sorting. It re-orders the items in a list manually. This should be accomplished by placing a drag handle as the leftmost element in the row, and pushing all the list item content to its right. In most cases, the ability to re-order the list should be locked behind an Edit Mode toggle mechanism.

Actions on Selected Items

[Design System Image]

[DemoCard Component - Interactive React component]

You may want to take action on several list items at once, but not all. The easiest way to do this is through a multi-select list. You may optionally add a highlight color for the selected items.

Local List Item Actions

[Design System Image]

[DemoCard Component - Interactive React component]

[DemoCard Component - Interactive React component]

If there are actions that you need to perform on individual list items, you may add icon buttons or an action menu. If you have three or fewer actions, they can be represented by individual icon buttons. If there are more than three, you should move them into an action menu.

> *If your entire list item is clickable, you should not include local list actions, as the two interactions can interfere with one another.*

Actions on Hover

On desktop applications, you may use hover events to display applicable list actions. Hover list actions can save space, leaving more room for displaying static information.

[Design System Image]

Navigating Down a Tree Structure

[Design System Image]

[DemoCard Component - Interactive React component]

Lists can be used to navigate a tree-like structure in an application. See the navigation pattern.

Responsive Design

[Design System Image]

When it comes to responsive design and progressive disclosure, it can be tricky to fit all the information from a list onto a smaller screen. Here are some best practices on designing for mobile:

- Remove less critical information from lists on mobile.
- Use summary cards with a "view more" option to view the full data.
- Use gestures (e.g., long-press, swipe-left) that align with the target device's native behavior.

Design Specifications

[Design System Image]

[MaterialDesignDescription Component - Interactive React component]

Developers

Use the following components to implement this pattern:

Angular:

- Angular Material - List
- @brightlayer-ui/angular-components - Info List Item

> *In Angular, the CDK (Component Developer's Kit) version 7+ is required to use the drag-and-drop feature.*

React:

- MUI - List - List Item
- @brightlayer-ui/react-components - Info List Item

React Native:

- React-Native - Flat List
- React Native Paper - List Item
- @brightlayer-ui/react-native-components - Info List Item