

## TELUS Component Library

## ExpandCollapse

Multi-Platform Component | [Figma UI KIT](#)

✓ First panel

---

✓ Second panel

---

Non-interactive item

---

✓ Last panel

---

```
<ExpandCollapse>
  {(expandProps) => (
    <StackView divider={{ variant: { decorative: true } }}>
      <ExpandCollapse.Panel
        {...expandProps}
        panelId="first"
        control={<Typography variant={{ size: 'h4' }}>First
panel</Typography>}
      >
        <Typography>Some content within the first
panel</Typography>
      </ExpandCollapse.Panel>
      <ExpandCollapse.Panel
        {...expandProps}
        panelId="second"
        control={<Typography variant={{ size: 'h4' }}>Second
panel</Typography>}
      >
        <Typography>Some content within the second
panel</Typography>
      </ExpandCollapse.Panel>
      <ExpandCollapse.Panel panelId="third" content>
        <Typography>Non-interactive item</Typography>
      </ExpandCollapse.Panel>
      <ExpandCollapse.Panel
```

```
    {...expandProps}
    panelId="fourth"
    control={<Typography variant={{ size: 'h4' }}>Last
panel</Typography>}
  >
    <Typography>Some content within the last
panel</Typography>
  </ExpandCollapse.Panel>
  <Spacer space={0} />
</StackView>
)}
</ExpandCollapse>
```

## Introduction

Flexible base component for lists where some or all items are collapsible headers.

[Follow the appropriate instructions](#) to add this component in to your app.

## Guidance

ExpandCollapse is designed for flexibility - components for a particular interaction pattern can be built on top of it. [SideNav](#) is an example of a component built on top of ExpandCollapse that facilitates a particular interaction pattern.

### ExpandCollapse

The outer `ExpandCollapse` represents the container for one or more collapsible `ExpandCollapse.Panel`s and manages state on which panels are open and closed.

### Children

`ExpandCollapse` takes a render function as its child and passes it an object, `expandProps`, which can be spread onto any `ExpandCollapse.Panel`s. These don't need to be direct children and can be anywhere inside the `ExpandCollapse` container.

Any number or arrangement of non-expandable items may be included alongside expandable panels.

✓ First panel

✓ Second panel

Some non-expandable content

```
<ExpandCollapse>
  {(expandProps) => (
    <StackView space={3}>
      <ExpandCollapse.Panel
        {...expandProps}
        panelId="first"
        control={<Typography>First panel</Typography>}
      >
        <Typography>Some content within the first
panel</Typography>
      </ExpandCollapse.Panel>
      <ExpandCollapse.Panel
        {...expandProps}
        panelId="second"
        control={<Typography>Second panel</Typography>}
      >
        <Typography>Some content within the second
panel</Typography>
      </ExpandCollapse.Panel>
      <ExpandCollapse.Panel panelId="third" content>
        <Typography>Some non-expandable content</Typography>
      </ExpandCollapse.Panel>
    </StackView>
  )}
</ExpandCollapse>
```

maxOpen prop

By default, any number of items may be opened at once. If desired, a number may be passed to ExpandCollapse's `maxItems` props to pin it only allow that

many items to be open at a time.

The `Accordion` component is a shorthand for `ExpandCollapse` with `maxOpen` set as 1.

`initialOpen` prop

An array of panel ids may be passed to the `initialOpen` prop to set those panels as open by default.

^ First panel

Some content within the first panel

^ Second panel

Some content within the second panel

✓ Third panel

```
<ExpandCollapse maxOpen={2} initialOpen={['first', 'second']}>
  {(expandProps) => (
    <StackView>
      <ExpandCollapse.Panel
        {...expandProps}
        panelId="first"
        control={<Typography>First panel</Typography>}
      >
        <Typography>Some content within the first
panel</Typography>
      </ExpandCollapse.Panel>
      <ExpandCollapse.Panel
        {...expandProps}
        panelId="second"
        control={<Typography>Second panel</Typography>}
      >
        <Typography>Some content within the second
panel</Typography>
      </ExpandCollapse.Panel>
      <ExpandCollapse.Panel
        {...expandProps}
```

```
        panelId="third"
        control={<Typography>Third panel</Typography>}
      >
        <Typography>Some content within the third
panel</Typography>
      </ExpandCollapse.Panel>
    </StackView>
  )}
</ExpandCollapse>
```

### Controlled behaviour with `open` prop

An ExpandCollapse may be controlled by a parent by passing an array of panel IDs to its `open` prop and a function that updates state to `onChange`. When delegating responsibility to the parent like this, `initialOpen` prop should not be used and `open` should always be defined as an array (pass an empty array when nothing is open).

No panels open.

✓ First panel

---

✓ Second panel

---

✓ Third panel

Expand editable code

`ExpandCollapse.Panel`

Each section that may be opened and closed should be wrapped in an `ExpandCollapse.Panel` and the `expandProps` from the `ExpandCollapse` parent should be spread to each of these.

### `panelId` prop (required)

Every `ExpandCollapse.Panel` must also take a `panelId` prop. This string will be used by the `ExpandCollapse` parent to manage whether this panel is open or closed.

Note that unlike HTML element `id` attributes, this does not need to be unique on the page. It is only used in the scope of the one `ExpandCollapse` parent. It should normally be unique within this one `ExpandCollapse` parent. If a panel id was to appear more than once, opening and closing each panel with the shared ID would open all others with the same ID.

### `control` prop (required)

The `control` prop contains the rendered content in the pressable section that opens and closes the panel.

The icon and spacing around this content are controlled by the theme, which may be overridden by passing the panel a `controlTokens` object, for example, `controlTokens={{ icon: null }}` removes the panel's control's icon.

## Accessibility

An `ExpandCollapse` panel's control is announced to assistive technology as a `button` with platform-appropriate notification of its expanded state automatically applied (`aria-expanded` on web and React Native's `accessibilityState.expanded` applied in native apps).

## Platform considerations

`ExpandCollapse` is available on both web and native apps.

## Props

### ExpandCollapse

Name	Type	Platform	Default	Description
variant	variant	standard		System variant prop, see <a href="#">variants</a> for more details
tokens	tokens	standard		System tokens prop, see <a href="#">tokens</a> for more details
children*	union	standard		ExpandCollapse usually takes a render function to which it passes an object with properties { openIds, onToggle, resetValues, setValues, unsetValues }. openIds (array of ids) and onToggle (function) will usually be passed to <code>&lt;ExpandCollapse.Panel&gt;</code>
maxOpen	number	standard		If set, enforces a specific maximum number of open items, closing already-open items when new items are opened. By default, any number of items may be open at once.
open	arrayOf	standard		If passed, the item or items that are expanded will be controlled by a parent component
onChange	func	standard		Function called when which panels are currently open changes. Necessary if `open` is passed.

Name	Type	Platform	Default	Description
initialOpen	arrayOf	standard		Allows items with specified ids to be open by default on first load. Should not be used with `open`.

## ExpandCollapse.Panel

Name	Type	Platform	Default	Description
variant	variant	standard		System variant prop, see <a href="#">variants</a> for more details
tokens	tokens	standard		System tokens prop, see <a href="#">tokens</a> for more details
openIds	arrayOf	standard		Array of the ids of currently open groups
panelId*	string	standard		Unique name by which the ExpandCollapse controls this element.
onToggle	func	standard		Function to call on pressing the panel's control, which should open or close the panel. If Panel is a direct child of `ExpandCollapse`, this prop will be provided



Name	Type	Platform	Default	Description
				by the ExpandCollapse parent.
onPress	func	standard		Optional function to call on pressing the panel's control, in addition to opening or closing the panel.
children	union	standard		ExpandCollapsePanel's children are the content that is hidden or revealed on open or close, and may be any single React element or a render function which takes this group's isExpanded state, onToggle function and panelId
control	custom	standard		The content inside the control element that opens and closes the ExpandCollapse when pressed.
controlTokens	custom	standard		Optional theme token overrides that may be passed to the ExpandCollapseControl element.
controlRef	custom	standard		An optional ref to be attached to the control

Name	Type	Platform	Default	Description
content	bool	standard		A boolean prop to determine if the panel is a content panel or not. If true, the panel will not have a control

## Tokens

In exceptional circumstances, the following tokens can be passed to this component to override its default styles. **Do not do this unless absolutely necessary.** [Read more about overriding styles.](#)

► View Tokens

## Variants

### Compact

^ First panel

Some content within the first panel

^ Second panel

Some content within the second panel

✓ Third panel

```
<ExpandCollapse maxOpen={2} initialOpen={['first', 'second']}>
  {(expandProps) => (
    <StackView>
      <ExpandCollapse.Panel
        {...expandProps}
        panelId="first"
      />
    )
  )
</ExpandCollapse>
```

```
    variant={{ compact: true }}
    control={<Typography>First panel</Typography>}
  >
    <Typography>Some content within the first
panel</Typography>
  </ExpandCollapse.Panel>
  <ExpandCollapse.Panel
    {...expandProps}
    panelId="second"
    variant={{ compact: true }}
    control={<Typography>Second panel</Typography>}
  >
    <Typography>Some content within the second
panel</Typography>
  </ExpandCollapse.Panel>
  <ExpandCollapse.Panel
    {...expandProps}
    panelId="third"
    variant={{ compact: true }}
    control={<Typography>Third panel</Typography>}
  >
    <Typography>Some content within the third
panel</Typography>
  </ExpandCollapse.Panel>
</StackView>
)}
</ExpandCollapse>
```

## Feedback

- Spotted a problem with this component? Raise an [issue on GitHub](#)
- See any [existing issues](#) for this component
- Contact the team on slack in [#ds-support](#)