

TELUS Component Library

Modal

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

```
function ModalExample() {  
  const [isOpen, setIsOpen] = useState(false)  
  
  return (  
    <>  
      <Button onPress={() => setIsOpen(true)}>Open "Modal"  
    </Button>  
      <Modal isOpen={isOpen} onClose={() => setIsOpen(false)}>  
        <Typography variant={{ size: 'h2' }}>This is a  
Modal</Typography>  
        <Box vertical={4}>  
          <Typography>  
            This is some modal content. You can pass any  
children to the Modal to achieve your  
desired look.  
          </Typography>  
        </Box>  
      </Modal>  
    </>  
  )  
}
```

Introduction

A modal window is a secondary window that opens on top of the main one.

Users have to interact with it before they can carry out their task and return to the main window.

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

Guidance

Use to reveal additional information to a user after they have performed an explicit interaction.

They are a strongly discouraged pattern; it's preferred to have all relevant information within a page, and irrelevant information either linked externally or omitted.

- Must only appear after a customer interaction, not on page load or any other circumstance
- Open a modal based on explicit customer action e.g. clicking on a button/link/form field
- Only one modal should be "current" at any time
- Read [WebAIM's documentation](#) to create accessible modals
- Don't use modals to reinforce or repeat information already available in the parent page or view
- Don't use modals consecutively
- Default modal width is 576px, except for the XS / SM viewports where the modal takes up 100% of the screen

On the web, `Modal` component has a set of web only properties that applies some specific decisions about content structure and styling to the base Modal component that is exported from `@telus-uds/components-base`.

CAUTION

ContentModal and DialogModal are web only recipes.

ContentModal

A content modal provides a way for the customer to access additional information without leaving the current page or view, and maintain context. It's also an effective technique to catch the customer's attention to something vital, a specific task, or set of information.

Open "Content Modal"

```
function ContentModalExample() {
  const [isOpen, setIsOpen] = useState(false)

  return (
    <>
      <Button onPress={() => setIsOpen(true)}>Open "Content
Modal"</Button>
      <Modal
        isOpen={isOpen}
        onClose={() => setIsOpen(false)}
        heading="Make a statement for a content modal"
        subHeading="Subheading for context"
        subHeadingSize="large"
        bodyText="Use a content modal for dismissible content. A
content modal provides a way for the user to access additional
information without leaving the current page or view, and
maintain context"
        confirmButtonText="Continue"
        onConfirm={() => setIsOpen(false)}
      />
    </>
  )
}
```

DialogModal

A dialog modal provides information to the customer and prompts them for a response.

- Use dialog modals as a means for customers to safely make actions that require their full attention, such as confirmations of destructive actions
- The copy on the modal should clearly lead to confirm or reject an impending decision and if possible the outcomes
- Avoid dialog modals:
 - if there are more than 2 options for customer consideration
 - as alerts to display errors, warnings, and success messages
 - being shown consecutively

Open "Dialog Modal"

```
function DialogModalExample() {
  const [isOpen, setIsOpen] = useState(false)

  return (
    <>
      <Button onPress={() => setIsOpen(true)}>Open "Dialog
Modal"</Button>
      <Modal
        isOpen={isOpen}
        onClose={() => setIsOpen(false)}
        onCancel={() => setIsOpen(false)}
        heading="Ask a question that confirms the customer
action which initiated the dialog modal"
        bodyText="Use a dialog modal to provide information and
prompt for a response to customer action, which may lead to a
significant impact on account or lead to unrecoverable states."
        confirmButtonText="Confirm"
        onConfirm={() => setIsOpen(false)}
        cancelButtonText="Cancel"
      />
    </>
  )
}
```

Accessibility

- Be aware that a modal's appearance will put users of assistive technology in a new focus context. This may be disorientating, so be sure to ensure a modal's content and role is easy to understand and easy to get out of.
- Long `bodyText` may scroll within the modal. Not all devices and operating systems make it visually apparent that scrollable sections of a page are scrollable, so try to avoid long text that might overflow.

Platform considerations

The `Modal` component is currently only supported on web.

The `BaseModal` component is available on both native platforms and web.

Props

Name	Type	Platform	Default	Description
onCancel	func	web only	onClose	Action to perform c pressing t cancel bu Defaults t `onClose` action if undefinec
variant	variant	standard		System variant pr see variari for more details
children	union	standard		
tokens	tokens	standard		System tokens pr see token more deta
isOpen	bool	standard		Setting th `true` wil open the `Modal`
onClose	func	standard		This func is trigger when the `Modal` i closed

Name	Type	Platform	Default	Description
maxWidth	bool	standard		`max-width` is applied to the `Modal` container.
heading	string	web only		Heading text shown prominently with heading semantic tags.
headingLevel	'h3' 'h4'	web only		Stylistic size and semantic level of the modal heading.
subHeading	string	web only		Short content rendered below the heading. If there is no body content that is scrollable, the subHeading sits above the scrollable area.
subHeadingSize	'small' 'medium' 'large'	web only		Short content rendered below the heading. If there is no

Name	Type	Platform	Default	Description
				body content that is scrollable subHeader sits above scrollable area.
bodyText	string	web only		If the modal requires longer text include it here. This will become scrollable if too long the allowed height of modal.
confirmButtonText	string	web only		Text for the button controlling primary action of the modal
confirmButtonVariant	object	web only		Button variant object to be passed through to Button to choose its visual style
onConfirm	function	web only		Action on pressing the confirm button

Name	Type	Platform	Default	Description
				confirm button.
cancelButtonText	string	web only		Text for the optional cancel button provided the user cannot want to do the confirm action.
cancelButtonType	elementType	web only		Component use for the cancel button (default: `TextButton`) Should act similar props to `TextButton`
copy	'en' 'fr'	standard		To define locale of the copy
closeButton	node	standard		Pass a react node to override the default close button or pass `null` to hide the close button.

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

Width

Use corresponding `width` variant to specify the desired width of the modal:



```
function ModalExample() {
  const [isOpen, setIsOpen] = useState(false)

  return (
    <>
      <Button onPress={() => setIsOpen(true)}>Open "Modal"
    </Button>
      <Modal isOpen={isOpen} onClose={() => setIsOpen(false)}
        variant={{ width: 's' }}>
        <Typography variant={{ size: 'h2' }}>This is a
Modal</Typography>
        <Box vertical={4}>
          <Typography>
            This is some modal content. You can pass any
children to the Modal to achieve your
desired look.
          </Typography>
        </Box>
        <Button onPress={() => setIsOpen(false)}>Close
Modal</Button>
      </Modal>
    </>
  )
}
```

```
}
```

Currently available widths are the following:

- `'s'` : 320px,
- `'m'` : 480px (also max-width on MD viewports),
- `'l'` : 630px (also max-width on LG viewports),
- `'xl'` : 768px (also max-width on XL viewports)

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](#)