

Button

Buttons allow the user to perform actions like submitting a form (Send, Sign In, Sign Up, Save), initiating a task (Edit Profile, Add User Slots, Share, Run, Resolve Conflicts, etc.), or showing/hiding UI elements (Close, Advanced options for collection generation).

🔗 Find the migration guide from legacy `Button` component to aether `Button` component [Here](#).

Import

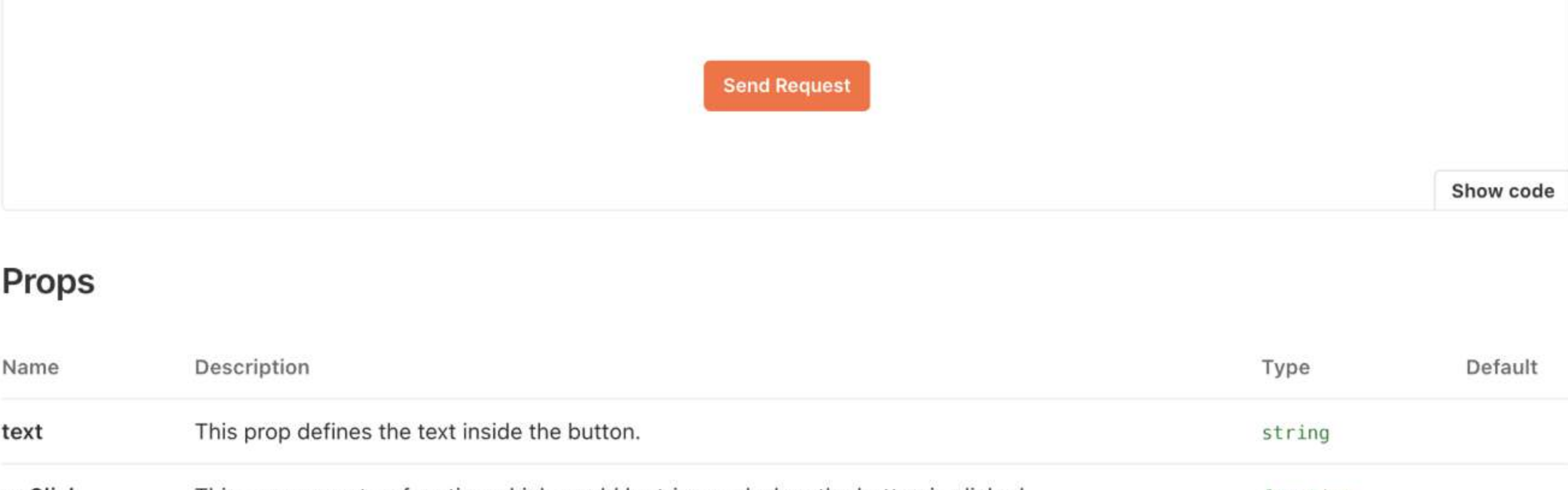
```
import { Button } from '@postman/aether';
```

Copy

Usage

```
<Button
  type='primary'
  size='medium'
  text='Send Request'
  onClick={() => {}}
/>
```

Copy



Show code

Props

Name	Description	Type	Default
text	This prop defines the text inside the button.	string	
onClick	This prop accepts a function which would be triggered when the button is clicked.	function	
onDoubleClick	This prop accepts a function which would be triggered when the button is double clicked using a pointing device.	function	
type	This prop defines the appearance of the button.	string	secondary
size	This prop defines the size of the button. Possible values are: small, medium.	string	medium
icon	This prop allows to add an icon inside the button. The value of this prop must be from the Aether Icons .	Aether Icon	
tooltip	This prop displays the tooltip over the button.	node,Tooltip component	
shortcutLabel	This prop allows you to display a shortcut label inside a tooltip, that is displayed when we hover over the button.	string	
isActive	This prop allows to keep the Button component in the active state.	boolean	false
isDisabled	This props makes the button disabled.	boolean	
isLoading	This prop allows to make the Button disabled and shows an animated loader inside button while the action is under process.	boolean	false
showIndicator	This prop shows a dot-indicator on the top-right corner of the Button, this prop works only for the tertiary variant of the Button and for the CounterButton	boolean	false
minWidth	This prop defines the min-width for the button, by default button will take as much width as required by the content inside it. It accepts any value which is valid to min-width CSS property.	string	
fullWidth	This prop expands the button to take 100% width of its immediate parent.	boolean	false
onFocus	This prop accepts a function which would be triggered when the button is focused. The button becomes focused when we interact with it using the Tab key.	function	
onBlur	This prop accepts a function which would be triggered when the focus of button is lost.	function	
className	This prop accepts classname property for this component. It is a recommended practice to use classNames only when needed.	string	
autoFocus	Use this prop to focus Button component on the first mount	boolean	false
data-testid	Recommended way to set unique identifiers for finding elements when writing test cases. Use of classname or ids to find elements in test cases is discouraged. Instead follow semantic properties or use dataTestId as an alternative to classname or ids. Read more here ↗ for better testing strategy.	string	

Examples

Button types

- prop: `type`
- Values: `secondary` (default) | `primary` | `tertiary` | `outline` | `destructive` | `plain` | `monochrome-plain` | `muted-plain`
- Value type: `String`



Show code

Button sizes

- prop: `size`
- Values: `medium` (default) | `small`
- Value type: `String`



Show code

Button label

- prop: `text`
- Value type: `String`
- Defines the text inside the button.

Button icon

- prop: `icon`
- Value type: `String` | `Node`
- Adds an icon on the left side of the button-text. It also allows to display icon-only button.

Important notes:

- `icon='icon->'` syntax is deprecated and will be removed in future.
- It is recommended to use the icons from the [Aether icons library](#), and ONLY use custom component as the value of `icon` prop when the required icon is not in the list. Also, when using Aether icons, consumer won't need to handle the icon dimensions explicitly. Means, it is not recommended to pass the `size` prop along with the icon. This is abstracted within the Button component based on the size of the Button.
- As per the design guidelines, icons can only be used with tertiary type buttons, so make sure you add `type='tertiary'` prop whenever using icons.
- Recommended way to render icon inside the button:

```
import { IconActionAddStroke } from '@postman/aether-icons'
...
<Button icon={<IconActionAddStroke />} />
```

Copy



Show code

Custom button icon

We suggest passing custom component as button icon only when required for example for some branding purposes.

Some brand logo icons might not be available in [Aether icons library](#) in this case. For such cases, consumers can pass custom component as value of `icon` prop, but in this case they own the output completely. So in this case, make sure to test the icon size and color, based on the button size and type respectively.

Please feel free to reach out to [#design-system](#) channel if you have any queries around when to use custom icon inside button.



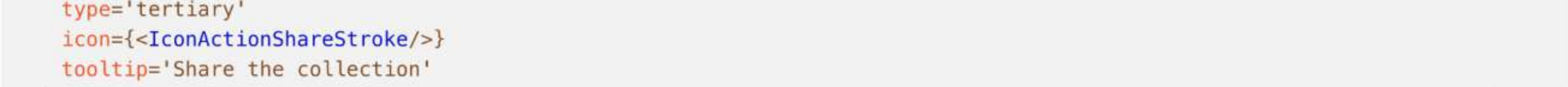
Show code

Button tooltip

- prop: `tooltip`
- Value type: `String` or `Object`
- To display a tooltip over the button

```
<Button
  onClick={() => {}}
  type='tertiary'
  icon={<IconActionShareStroke/>}
  tooltip='Share the collection'
/>
```

Copy

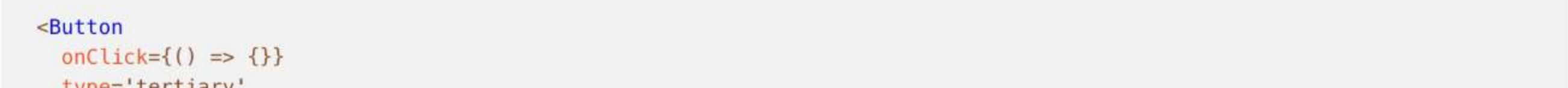


Customize Button tooltip

```
<Button
  onClick={() => {}}
  type='tertiary'
  icon={<IconActionShareStroke/>}
  tooltip={
    <Tooltip
      content='Share collection'
      placement='top'
    />
  }
/>
```

Copy

- Values: `bottom` (default) | `left` | `top` | `bottom` | `right` | `auto`



Show code

Button shortcut

To display a shortcut label inside the button tooltip.

- prop: `shortcutLabel`

```
<Button
  onClick={() => {}}
  type='tertiary'
  icon={<IconActionShareStroke/>}
  tooltip='Share collection'
  shortcutLabel='⌘ + S'
/>
```

Copy



Disabling Button

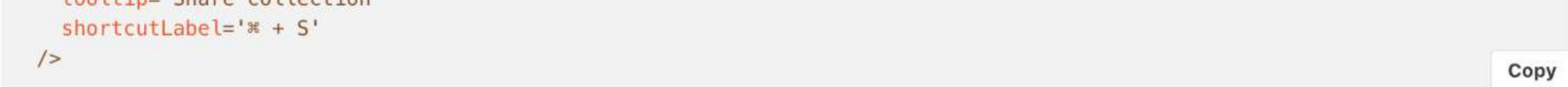
- prop: `isDisabled`
- Value type: `Boolean`
- Default value: `false`

Use the `tooltip` prop to conditionally show a message to inform users why the button is disabled.

```
const isSignedIn = false;

<Button
  onClick={() => {}}
  type='tertiary'
  text='Share'
  icon={<IconActionShareStroke/>}
  tooltip={!isSignedIn && 'Sign-in to share collection'}
  isDisabled={!isSignedIn}
/>
```

Copy



Loading inside Button

- prop: `isLoading`
- Value type: `Boolean`
- Default value: `false`
- To show a loader inside a button and makes button disabled during that time.

```
<Button
  onClick={() => {}}
  type='primary'
  text='Button'
  isLoading
/>
```

Copy



Indicator on Button

- prop: `showIndicator`
- Value type: `Boolean`
- Default value: `false`
- To show a dot-indicator on the top-right corner of the Button. This prop works only for the tertiary variant of the `Button` and for the `CounterButton`.



Show code

Full width Button

- prop: `fullWidth`

className

- Value type: `String`
- This prop allows to add a classname to the Button component.
- The majority of the appearances and visual behaviours of the Button should be handled by the other props available.

⚠️ Don't override the styles using the `className` prop.

```
<Button
  type='primary'
  text='Button'
  onClick={() => {}}
  className='custom-className'
/>
```

Copy