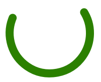
TELUS Component Library

Spinner

Web Component | Figma UI KIT



Fetching products

```
<Spinner label="Fetching products" show />
```

Introduction

Spinner can be used to show a visual loading state either as a standalone element, on a specific section, or the entire screen.

Follow the appropriate instructions to add this component in to your app.

Guidance

- A spinner is most appropriate for providing feedback for actions that last between 2 and 10 seconds
- Set `show` prop to `true` to display the spinner and unset it or set to `false` to hide it
- Label should be short and concise. Spinner label should accurately explain the state of the requested action along with relevant items being loaded, "Loading page content", "Logging in", "Processing payment"
- Both the small and large spinner have an overlay background to visually indicate that the content behind it is inaccessible
- Use the small spinner when applied on a button or other small interactive elements such as toggles or links
- Use the large spinner when affecting an entire web page or content block, whose content has not finished loading, such as paginated content

- Use the standalone spinner (`small` / `large`) as an interstitial placeholder for loading content
- If the Spinner should only cover its children, set the `inline` prop to `true` . If the Spinner should cover the full width of its parent regardless of the size of its children, `inline` should be set to `false`

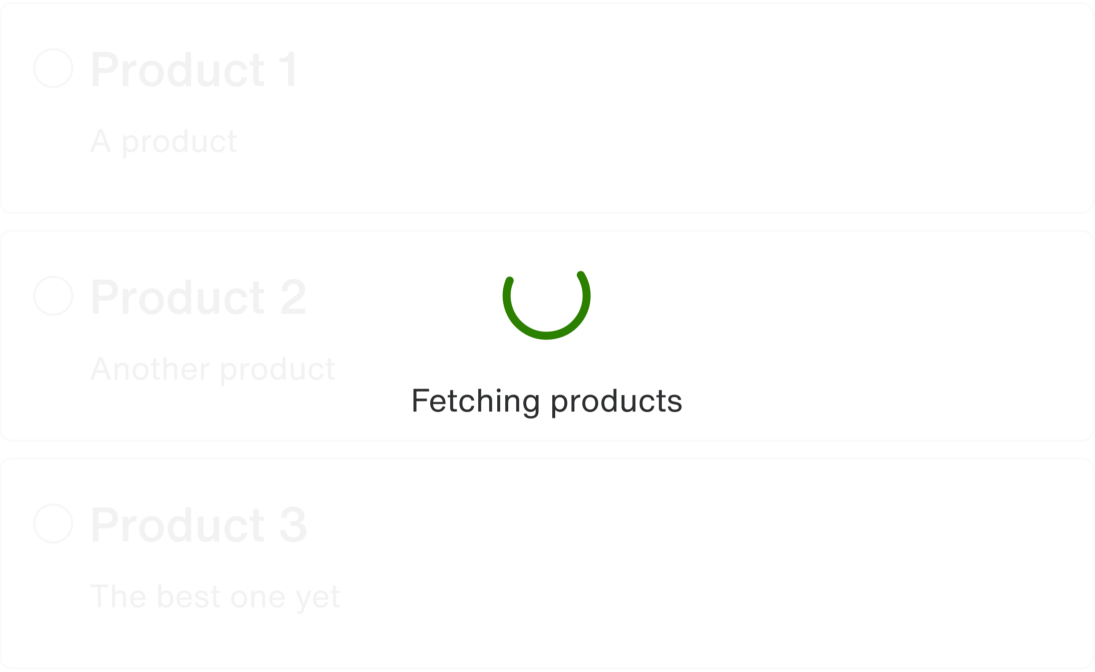
Accessibility

When using a `Spinner` , ensure that you provide a `label` prop to explain what is happening to assistive technologies. If you are using a Spinner as a placeholder until content either arrives or changes, ensure that you are using an [`aria-live` region](#) or appropriate `role` on the parent.

Platform considerations

This component is currently only supported on web.

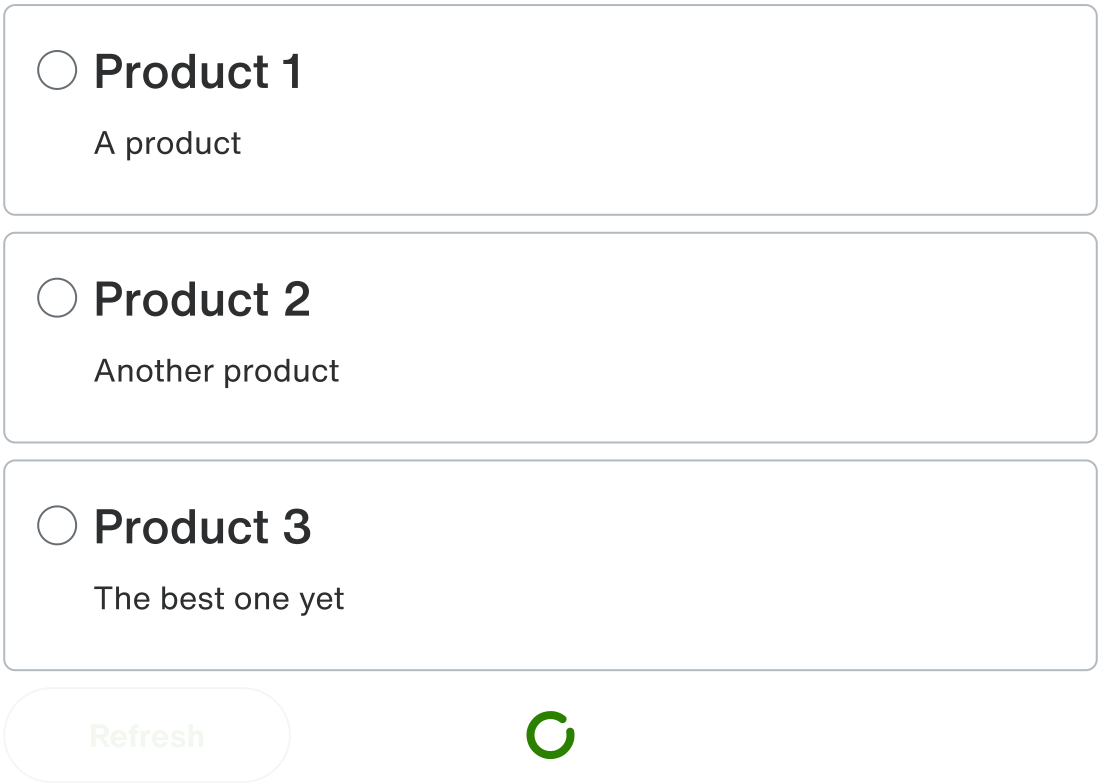
Overlaying children



```
<Spinner label="Fetching products" show>
  <StackView space={2}>
    <RadioCard title="Product 1">
      <Typography>A product</Typography>
    </RadioCard>
    <RadioCard title="Product 2">
      <Typography>Another product</Typography>
    </RadioCard>
    <RadioCard title="Product 3">
      <Typography>The best one yet</Typography>
    </RadioCard>
  </StackView>
</Spinner>
```

Overlaying buttons

Set the `size` variant to `small` and `inline` prop to `true` in order to fully cover each individual button:



○ Product 1

A product

○ Product 2

Another product

○ Product 3

The best one yet

```
<>
  <StackView space={2}>
    <RadioCard title="Product 1">
      <Typography>A product</Typography>
    </RadioCard>
    <RadioCard title="Product 2">
      <Typography>Another product</Typography>
    </RadioCard>
    <RadioCard title="Product 3">
      <Typography>The best one yet</Typography>
    </RadioCard>
  </StackView>
  <Spacer space={2} />
  <Spinner inline label="Fetching products" variant={{ size:
'small' }} show>
    <Button onPress={() => {}}>Refresh</Button>
  </Spinner>
</>
```

Full screen spinner

○ **Product 1**

A product

○ **Product 2**

Another product

○ **Product 3**

The best one yet

Refresh

```
function fullScreenSpinner() {
  const [show, setShow] = useState(false)
  const handleRefresh = () => {
    setShow(true)
    setTimeout(() => setShow(false), 2000)
  }

  return (
    <>
      <StackView space={2}>
        <RadioCard title="Product 1">
          <Typography>A product</Typography>
        </RadioCard>
        <RadioCard title="Product 2">
          <Typography>Another product</Typography>
        </RadioCard>
        <RadioCard title="Product 3">
          <Typography>The best one yet</Typography>
        </RadioCard>
        <Button onPress={handleRefresh}>Refresh</Button>
      </StackView>
      <Spinner fullScreen label="Fetching products" show={show}
/>
    </>
  )
}
```

Props

Name	Type	Platform	Default	Description
fullScreen	bool	web only	false	Enables body locking.
inline	bool	web only	false	Set the inline prop to true if the Spinner should only cover its children; if the Spinner should cover the full width of its parent regardless of the size of its children, inline should be set to false
show	bool	web only	false	Whether or not to render the spinner.
isStatic	bool	web only	false	If true, it should render a static spinner

Name	Type	Platform	Default	Description
variant	variant	web only	{}	System variant prop, see variants for more details
tokens	tokens	web only		System tokens prop, see tokens for more details
children	node	web only		Content to be overlaid while the spinner is active. Can be text, any HTML element, or any component.
label*	string	web only		Communicates a message to assistive technology while visible. This same message will appear underneath the spinner when its `size` is `large`.
labelPosition	string	web only		Determine where the label of the spinner should be placed, left, right, bottom or top.

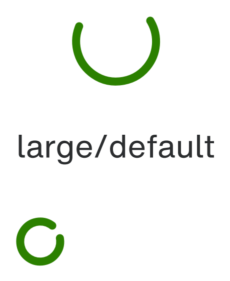
Tokens

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

Size



```
<StackView space={4}>
  <Spinner label="large/default" show={true} />
  <Spinner variant={{ size: 'small' }} label="small" show={true}
/>
</StackView>
```

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