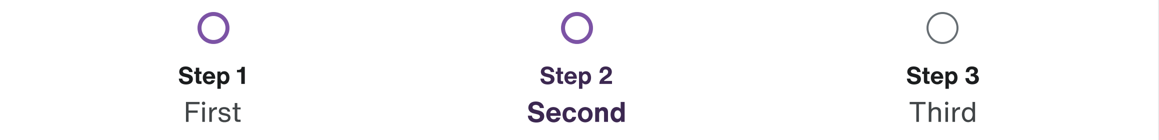
TELUS Component Library

StepTracker

Multi-Platform Component | Figma UI KIT



```
<StepTracker current={1} steps={['First', 'Second', 'Third']} />
```

Introduction

StepTracker component shows the current position in a sequence of steps. At wide viewports (`lg` and `xl`), all steps are labelled, and the current step is highlighted; at narrower viewports, progress through the steps is shown but only the current step has a visible label.

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

Accessibility

Keep in mind that in its current implementation this is not an interactive component and can't be used to navigate between steps. The application must provide its own navigation mechanism and state control. That is the main reason the component assumes the `progressbar` role in terms of accessibility. This also makes it extremely important to make sure you set the necessary accessibility attributes on controls used to modify the state of the step tracker.

That said, you can also pass standard a11y props to the `StepTracker` component, including the ones that are specific to the `progressbar` role. By default the following props are used:

- `accessibilityRole` (`role`): `progressbar` ,
- `accessibilityLabel` (`aria-label`): `Step I of N: <Current Step Label>` ,
- `accessibilityLevel` (`aria-level`): 1,

- `accessibilityValue.min` (`aria-valuemin`): `0`,
- `accessibilityValue.max` (`aria-valuemax`): `steps.length - 1`,
- `accessibilityValue.now` (`aria-valuenow`): `current`,
- `accessibilityValue.text` (`aria-valuetext`): `<Current Step Label>`,

Platform considerations

The component is available on both native platforms and web.

Props

Name	Type	Platform	Default	Description
current	number	standard	0	The current step, 0-based number
copy	union	standard	'en'	The language to use for the labels
dictionary	shape	standard	{ en: { 1: { stepLabel: 'Step %{stepNumber}', stepTrackerLabel: 'Step %{stepNumber} of %{stepCount}: %{stepLabel}' }, 2: { stepLabel: '%{stepNumber}.', stepTrackerLabel: 'Step %{stepNumber} of %	Custom dictionary containing the labels to use for the steps

Name	Type	Platform	Default	Description
			{stepCount}: %{stepLabel}' }, 3: { stepLabel: 'Step %{stepNumber}', stepTrackerLabel: 'Step %{stepNumber} of %{stepCount}: %{stepLabel}' } }, fr: { 1: { stepLabel: 'Étape %{stepNumber}', stepTrackerLabel: 'Étape %{stepNumber} sur %{stepCount}: %{stepLabel}' }, 2: { stepLabel: '%{stepNumber}.', stepTrackerLabel: 'Étape %{stepNumber} sur %{stepCount}: %{stepLabel}' }, 3: { stepLabel: 'Étape %{stepNumber}', stepTrackerLabel: 'Étape %{stepNumber} sur %{stepCount}: %{stepLabel}' } } }	
steps	arrayOf	standard	[]	An array of strings defining the step titles

Name	Type	Platform	Default	Description
tokens	tokens	standard		System tokens prop, see tokens for more details
variant	variant	standard		System variant prop, see variants for more details

Tokens

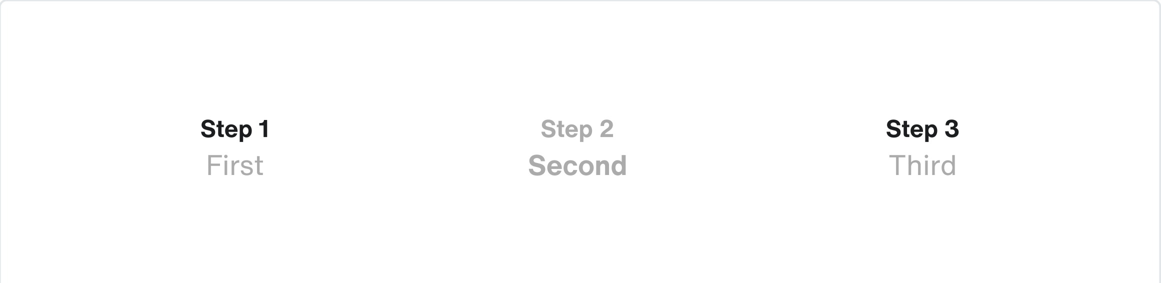
In exceptional circumstances, the following tokens can be passed to this component to override its default styles. **Do not do this unless absolutely necessary.** Read more about overriding styles.

▶ View Tokens

Variants

Inverse

Use the `inverse` variant to improve legibility and accessibility overall while displaying `StepTracker` on darker backgrounds:



```
<Box variant={{ background: 'darkest' }} space={3}>
  <StepTracker current={1} steps={['First', 'Second', 'Third']}
variant={{ inverse: true }} />
</Box>
```

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