# React Stepper component

## Stepper

Steppers convey progress through numbered steps. It provides a wizard-like workflow.

Steppers display progress through a sequence of logical and numbered steps. They may also be used for navigation. Steppers may display a transient feedback message after a step is saved.

- **Types of Steps**: Editable, Non-editable, Mobile, Optional
- **Types of Steppers**: Horizontal, Vertical, Linear, Non-linear

{{“component”: “modules/components/ComponentLinkHeader.js”}}

> **Note:** Steppers are no longer documented in the Material Design guidelines, but MUI will continue to support them.

### Horizontal stepper

Horizontal steppers are ideal when the contents of one step depend on an earlier step.

Avoid using long step names in horizontal steppers.

#### Linear

A linear stepper allows the user to complete the steps in sequence.

The `Stepper` can be controlled by passing the current step index (zero-based) as the `activeStep` prop. `Stepper` orientation is set using the `orientation` prop.

This example also shows the use of an optional step by placing the `optional` prop on the second `Step` component. Note that it’s up to you to manage when an optional step is skipped. Once you’ve determined this for a particular step you must set `completed={false}` to signify that even though the active step index has gone beyond the optional step, it’s not actually complete.

{{“demo”: “HorizontalLinearStepper.js”}}

### Non-linear

Non-linear steppers allow the user to enter a multi-step flow at any point.

This example is similar to the regular horizontal stepper, except steps are no longer automatically set to `disabled={true}` based on the `activeStep` prop.

The use of the `StepButton` here demonstrates clickable step labels, as well as setting the `completed` flag. However because steps can be accessed in a non-linear fashion, it's up to your own implementation to determine when all steps are completed (or even if they need to be completed).

{{"demo": "HorizontalNonLinearStepper.js"}}

### Alternative label

Labels can be placed below the step icon by setting the `alternativeLabel` prop on the `Stepper` component.

{{"demo": "HorizontalLinearAlternativeLabelStepper.js"}}

### Error step

{{"demo": "HorizontalStepperWithError.js"}}

### Customized horizontal stepper

Here is an example of customizing the component. You can learn more about this in the overrides documentation page.

{{"demo": "CustomizedSteppers.js"}}

## Vertical stepper

Vertical steppers are designed for narrow screen sizes. They are ideal for mobile. All the features of the horizontal stepper can be implemented.

{{"demo": "VerticalLinearStepper.js"}}

### Performance

The content of a step is unmounted when closed. If you need to make the content available to search engines or render expensive component trees inside your modal while optimizing for interaction responsiveness it might be a good idea to keep the step mounted with:

```
<StepContent TransitionProps={{ unmountOnExit: false }} />
```

## Mobile stepper

This component implements a compact stepper suitable for a mobile device. It has more limited functionality than the vertical stepper. See mobile steps for its inspiration.

The mobile stepper supports three variants to display progress through the available steps: text, dots, and progress.

### Text

The current step and total number of steps are displayed as text.

{{“demo”: “TextMobileStepper.js”, “bg”: true}}

### Text with carousel effect

This demo uses react-swipeable-views to create a carousel.

{{“demo”: “SwipeableTextMobileStepper.js”, “bg”: true}}

### Dots

Use dots when the number of steps is small.

{{“demo”: “DotsMobileStepper.js”, “bg”: true}}

### Progress

Use a progress bar when there are many steps, or if there are steps that need to be inserted during the process (based on responses to earlier steps).

{{“demo”: “ProgressMobileStepper.js”, “bg”: true}}