@wordpress/components

- Installation
- Setup
  - Within WordPress
  - Outside WordPress
- Usage
  - Popovers
  - TypeScript
- Docs & examples
- Contributing to this package

This package includes a library of generic WordPress components to be used for creating common UI elements shared between screens and features of the WordPress dashboard.

Installation

```
npm install @wordpress/components --save
```

*This package assumes that your code will run in an **ES2015+** environment. If you're using an environment that has limited or no support for such language features and APIs, you should include the polyfill shipped in `@wordpress/babel-preset-default` in your code.*

Setup

Many components require the package's CSS stylesheet to be loaded.

Within WordPress

To ensure proper load order, add the `wp-components` stylesheet as a dependency of your plugin's stylesheet. See wp_enqueue_style documentation for how to specify dependencies.

Outside WordPress

Load the stylesheet in your application:

```
import '@wordpress/components/build-style/style.css';
```

The RTL version of the stylesheet is available at `@wordpress/components/build-style/style-rtl.css`.

Usage

Within Gutenberg, these components can be accessed by importing from the `components` root directory:

```
import { Button } from '@wordpress/components';

export default function MyButton() {
    return <Button>Click Me!</Button>;
}
```

Popovers

By default, the `Popover` component will render within an extra element appended to the body of the document.

If you want to precisely control where the popovers render, you will need to use the `Popover.Slot` component.

The following example illustrates how you can wrap a component using a `Popover` and have those popovers render to a single location in the DOM.

```
import { Popover, SlotFillProvider } from '@wordpress/components';
import { MyComponentWithPopover } from './my-component';

const Example = () => {
    <SlotFillProvider>
```

```
        <MyComponentWithPopover />
        <Popover.Slot />
    </SlotFillProvider>;
};
```

TypeScript

This package exposes its own types for the components it exports, however it doesn't export its own types for component props. If you need to extract the props type, please use `React.ComponentProps` to get the types from the element.

```
import type { ComponentProps } from 'react';
import { Button } from '@wordpress/components';

export default function MyButton( props: ComponentProps< typeof Button > ) {
    return <Button { ...props }>Click Me!</Button>;
}
```

Docs & examples

You can browse the components docs and examples at https://wordpress.github.io/gutenberg/

Contributing to this package

This is an individual package that's part of the Gutenberg project. The project is organized as a monorepo. It's made up of multiple self-contained software packages, each with a specific purpose. The packages in this monorepo are published to npm and used by WordPress as well as other software projects.

To find out more about contributing to this package or Gutenberg as a whole, please read the project's main contributor guide.

This package also has its own contributing information where you can find additional details.

□□□ #2
□ □□□ □□□ 2026-10-31 18:17:56 UTC
□ □□□ □□□ 2026-10-31 18:17:57 UTC