

Themes

One of the key benefits of Brightlayer UI is the consistent look & feel between different applications. This is achieved primarily through the use of Brightlayer UI themes. Brightlayer UI offers two themes: a light theme (blue) and a dark theme (blueDark).

[MaterialDesignDescription Component - Interactive React component]

[MaterialDesignDescription Component - Interactive React component]

Both themes inherit from Material Design's baseline theme and expand upon it with Brightlayer UI specific styles. The major differences between Brightlayer UI themes and the baseline Material theme are the color palettes and [typographies](#).

[TOC Component - Interactive React component]

Explore Our Themes

Have a look at the component theme explorer below and learn how Brightlayer UI themes work at a high level. Here we showcase some of the most representative components from Material Design and Brightlayer UI, and how our themes affect them.

[Theme Explorer - Interactive theme customization tool]

You are also encouraged to explore our more detailed theme examples where you can modify the code and see the effects in real-time.

[DemoCard Component - Interactive React component]

Blue Theme / Light Theme

[Design System Image]

The blue theme is the default theme for all Brightlayer UI applications. With a clean color vibrance and high text legibility, it is well-suited for everyday use on a computer monitor, on-site facility inspection on a tablet, or on a mobile phone screen.

The light theme uses our iconic [Blue 500](#) color (`#007bc1`) throughout the application, notably in toolbars, buttons, and icons. In some places, Light Blue 500 is applied as a secondary color on interactive UI elements such as form controls. The light theme's base background color is White 200, although the surface elements are often shaded in White 50 (pure white).

[Design System Image]

Dark Theme / Blue Dark Theme

[Design System Image]

The dark theme is intended primarily for applications used in low-light conditions. As the name indicates, it utilizes mostly dark colors (grays and blacks). Other colors are typically reserved for interactive components and status displays.

[Design System Image]

To accommodate for [accessibility requirements](#), Brightlayer UI uses the 200 shade of colors when text is used on a dark background, such as in an outlined button and a text field label. For "contained" elements, such as in an elevated / contained button, the same style is used as in the light theme (500 shades).

[Design System Image]

[Design System Image]

Developers

Angular

Angular themes are available from our `@brightlayer-ui/angular-themes` package, which is a collection of SCSS stylesheets that are imported into your application and applied via classes (``blui-blue`` or ``blui-dark``) to both Brightlayer UI components and Angular Material components.

- `@brightlayer-ui/angular-themes`: [NPM](#), [GitHub](#)
- [Theming Guide](#) on Angular Material

React

React themes are available from our `@brightlayer-ui/react-themes` package, which contains JSON objects that are consumed by the theme mechanism for MUI components (```

[ThemeProvider Component - Interactive React component]

```), including some specific styling rules defined in [JSS](#).

- `@brightlayer-ui/react-themes`: [NPM](#), [GitHub](#)
- [Theming Guide](#) on MUI

# React Native

React Native themes are available from our [@brightlayer-ui/react-themes](#) package, which utilize React Native Paper's theming mechanism. This package contains two separate dark themes which must be applied to different components to achieve the correct appearance (see the package readme for details).

- [@brightlayer-ui/react-native-themes](#): [NPM](#), [GitHub](#)
- [Theming Guide](#) on React Native Paper

# Color Palette

All of the [Brightlayer UI colors](#) are available individually ([@brightlayer-ui/colors](#)) for use in your application. However, in most cases, you should use values from the themes to alter styles for better future-proofing.