

# Getting Started with React

---

The following instructions will guide you through starting a new Brightlayer UI project using React. Alternatively, you can also start your project by downloading one of our code samples from [Github](#) (the Login or Navigation examples are good projects to start from).

## Using the Brightlayer UI Templates

The fastest way to start a new Brightlayer UI project is to use the [Brightlayer UI Templates](#). The Brightlayer UI templates will automatically create a starter project for you using [Vite](#) to scaffold the project with all of the necessary Brightlayer UI dependencies pre-configured. To start a React project:

```
```sh npx create-blui-react-app ```
```

You can also select the template you want to use when starting a new project:

```
```sh npx create-react-app app-name --template @brightlayer-ui/blank-typescript npx  
create-react-app app-name --template @brightlayer-ui/routing-typescript npx create-react-  
app app-name --template @brightlayer-ui/authentication-typescript ```
```

You will be asked to provide a name for your project and whether you want to use the Brightlayer UI eslint and prettier configuration files.

The Brightlayer UI React Templates will automatically include:

- MUI Component Library and Icons ([@mui/material](#) / [@mui/icons-material](#))
- Brightlayer UI Component Library ([@brightlayer-ui/react-components](#))
- Brightlayer UI Themes ([@brightlayer-ui/react-themes](#))
- Brightlayer UI Icons ([@brightlayer-ui/icons-mui](#))
- Brightlayer UI Colors ([@brightlayer-ui/colors](#))
- Open Sans web font ([@fontsource/open-sans](#))
- Brightlayer UI ESLint and Prettier configurations, if selected ([@brightlayer-ui/eslint-config](#) / [@brightlayer-ui/prettier-config](#))
- Enzyme testing library ([enzyme](#) / [enzyme-adapter-react-16](#))

For detailed usage instructions, refer to the [Brightlayer UI React Templates documentation](#).

## Using an Existing Project

If you have an existing project and would like to integrate Brightlayer UI, you can follow the manual integration instructions below. If you have not already started your project, we recommend that you scaffold your project with the Brightlayer UI Templates.

---

# Manual Brightlayer UI Integration

---

Before taking the following steps, you'll want to make sure that you have an existing application. If you used the Brightlayer UI Templates to start your project, these steps have already been done for you automatically.

# Including the Component Libraries

## MUI

React projects should make use of the [MUI](#) component library, formerly known as Material UI. Our Brightlayer UI themes are designed to work with these components. You'll also want to install the icons.

```
```sh npm install --save @mui/material @mui/icons-material @emotion/react @emotion/styled // or yarn add @mui/material @mui/icons-material @emotion/react @emotion/styled ```
```

For detailed documentation about the various components available in MUI, as well as sample code and API reference, refer to the [MUI Documentation](#).

## Brightlayer UI Components

Additionally, we have a [supplemental library](#) of components designed specifically for Brightlayer UI applications. If you wish to use these components, you can install them by running:

```
```sh npm install --save @brightlayer-ui/react-components // or yarn add @brightlayer-ui/react-components ```
```

## Adding Brightlayer UI Themes

By default, your application will be themed with the Google Material theme. To use the [Brightlayer UI theme](#), first install it:

```
```sh npm install --save @brightlayer-ui/react-themes // or yarn add @brightlayer-ui/react-themes ```
```

Installing the theme package will automatically include the Open Sans web font package as well as the Brightlayer UI [colors](#) package.

## Applying the Theme

Import the theme files, MUI theme provider, and **Open Sans** font into your application, e.g. in App.js:

```
````tsx
...

```

To add the theme to your whole application, you would simply add a `

*[ThemeProvider Component]*

*` around your top-level component and pass it the theme you want to use, e.g.:*

```
````tsx

```

*[CssBaseline Component - Interactive React component]*

*[MyApp Component - Interactive React component]*

```
...

```

You can read about [React Theming](#) to learn more.

---

## Additional Information

---

### Code Formatting / Style

Brightlayer UI provides packages for code standards and style using ESLint and Prettier. These packages are used internally by the Brightlayer UI team, but they can also be used

in your projects. You can find instructions for adding these packages to your project on [GitHub](#).

*NOTE: the ESLint package is only available for TypeScript projects.*

## Browser Support

Brightlayer UI supports the same browsers as React and MUI. For specifics, please check out their support pages:

- [React](#)
- [MUI](#)

*With the end-of-life of Internet Explorer 11, Brightlayer UI will no longer be supporting IE11 as of 2021.*

## License Information

[React](#) and [MUI](#) are both available under the MIT License.