

Getting Started with React Native

The following instructions will guide you through setting up your computer for React Native development and starting a new Brightlayer UI project. Alternatively, you can start your project by downloading one of our code samples from [GitHub](#) (the Login example is a good project to start from) once you configure your environment.

Setting Up Your Environment

You'll need to start by following the general instructions for setting up your development environment for mobile found in our [Environment Guide](#).

Creating Your Application

Using the Brightlayer UI CLI

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

```
```sh npx react-native init MyApp ```
```

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

```
```sh npx @react-native-community/cli@latest init MyApp --template @brightlayer-  
ui/react-native-template-blank-typescript npx @react-native-community/cli@latest init  
MyApp --template @brightlayer-ui/react-native-template-routing-typescript npx @react-  
native-community/cli@latest init MyApp --template @brightlayer-ui/react-native-template-  
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 configurations files.

The Brightlayer UI React Native Templates will automatically include:

- React Native Paper Component Library ([react-native-paper](#))
- Brightlayer UI Component Library ([@brightlayer-ui/react-native-components](#))
- Brightlayer UI Themes ([@brightlayer-ui/react-native-themes](#))
- Material Icons ([react-native-vector-icons](#))
- Brightlayer UI Icons ([@brightlayer-ui/icons-svg](#))
- Brightlayer UI Colors ([@brightlayer-ui/colors](#))
- Open Sans font
- Brightlayer UI ESLint and Prettier configurations, if selected ([@brightlayer-ui/eslint-config](#) / [@brightlayer-ui/prettier-config](#))
- Testing libraries ([jest](#) / [react-test-renderer](#))

For detailed usage instructions, refer to the [Brightlayer UI React Native 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 CLI.

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 CLI to start your project, these steps have already been done for you automatically.

Including the Component Libraries

Material Components

React Native projects should make use of the [React-Native-Paper](#) component library. Our Brightlayer UI themes are designed to work with these components. You'll also want to install [react-native-vector-icons](#) to access the Material icons.

```
``sh npm install --save react-native-paper@4.0.0 react-native-vector-icons // or yarn add react-native-paper@4.0.0 react-native-vector-icons ``
```

NOTE: *react-native-paper@5.0.0 is currently available, however, we only support React Native Paper v4 at this time. Please be sure to keep peer-dependencies aligned as defined above to avoid potential conflicts.*

Brightlayer UI Components

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-native-components ``
```

Adding Brightlayer UI Themes

If you are using the recommended component libraries, your application will be themed with the Google Material theme by default. To use the Brightlayer UI theme, first install it:

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

*Using the Brightlayer UI React Native theme **requires** that you add the Open Sans font to your application. You can learn how to do this by reading the instructions for [Vanilla React Native](#). You can find the fonts [here](#).*

Applying the Theme

Import the theme files and React Native Paper theme provider into your application, e.g. in App.js:

```
`` `tsx
```

```
...
```

[ThemeProvider Component]

[App Component - Interactive React component]

```
`` `
```

For detailed information about theming, refer to the react-native-paper [theming guide](#).

NOTE: This theme only applies to the components in the react-native-paper and @brightlayer-ui/react-native-components package. It will not theme the components from other third party libraries. If you wish to use our theme to style your own components, refer to the react-native-paper [theming guide](#).

Avoiding Repeated Styles

To avoid adding repeatedly adding inline styles to standard components, you can define your own component that applies the styles and use that throughout your application, e.g.:

```
```tsx // MyText.js export default MyText = () => (
```

*[Text Component]*

```
);
```

```
// OtherFile.js
```

```
...
```

*[Text Component]*

*Themed Text Object*

```
```
```

Writing a React Native Application

If you are familiar with React development, React Native will be relatively simple to understand. The application logic remains the same, while the UI requires a little adjustment. Rather than typical HTML elements, React Native UIs are built with a special library of [Components](#), including `<Text>`, `<Image>`, `<View>`, etc. React Native layouts are built exclusively using FlexBox.

Running your application

After you've created your application with a CLI, it can be run from the command line:

```
```sh cd YourAppName
```

```
npm run // or yarn ```
```

When you make changes to the source code, the application will automatically rebuild and update on the device.

*To run on an Android emulator, the emulator must already be running or you will see an error message, "No connected devices found."*

*On a Mac, you may be prompted to install watchman (``brew install watchman``) or update a few settings to run the application.*

## Final Build

Now, you can open the project folders in xCode (`/ios/projectName.xcodeproj`) or Android Studio (`/android`) and perform builds from the IDE. Depending on the version of React Native you are using, you may have to [manually bundle your JS code](#) for Android.

---

## 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.*

## Component recommendations

There are a lot of resources and components available on the market to help speed up your development and improve reusability. React Native also comes with a built-in component library. We recommend the following:

- Navigation - [React Navigation](#) (JS-only implementation) - [React Native Navigation](#) (native implementation)
- Component Libraries - [React Native Built-in Components](#) - [React Native Paper](#) - [React Native Elements](#)

## License Information

[React Native](#) is available under the MIT License.