

# Package Migration Instructions

---

This guide will walk you through the process of migrating your project from the old `@pxblue`-scoped packages to the new `@brightlayer-ui`-scoped packages. We have maintained the version numbers when migrating, so in most cases you will simply need to change the package name and update the import statements throughout your application. For most packages, we have migrated the latest two major versions. If you are using an older version, you will need to upgrade before changing to the packages in the new scope.

Peer dependencies have changed to **Brightlayer UI** scoped packages. Update your project's peer dependencies from `@pxblue`-scoped to new `@brightlayer-ui`-scoped packages.

## @brightlayer-ui/angular-auth-workflow

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- Change all `` component prefixes to ``.
- The below code snippets indicate the changes to be made for using Auth Module in your angular application.

```
`` ` // package.json
```

```
@pxblue/angular-auth-workflow => @brightlayer-ui/angular-auth-workflow ` ` `
```

To install the latest version of all of these peer dependencies, run the following command in your project root:

```
`` ` npm install @brightlayer-ui/colors @brightlayer-ui/angular-components @angular/cdk  
@angular/material // or yarn add @brightlayer-ui/colors @brightlayer-ui/angular-  
components @angular/cdk @angular/material ` ` `
```

Update module name from `PxbAuthModule` to `BluiAuthModule`.

```
`` ` // app.module.ts // before import from '@pxblue/angular-auth-workflow'; imports: [  
PxbAuthModule ]
```

```
// after import from '@brightlayer-ui/angular-auth-workflow'; imports: [ BluiAuthModule ]  
```
```

Update component name from `` to ``.

```
` `` // Angular component
```

```
// before
```

```
// after ` ``
```

## @brightlayer-ui/angular-components

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- Change all `` component prefixes to ``.
- The below code snippets indicate the changes to be made for using App Bar in your angular application.

```
` `` // package.json
```

```
@pxblue/angular-components => @brightlayer-ui/angular-components ` ``
```

Update package name from `@pxblue/angular-components` to `@brightlayer-ui/angular-components`.

```
` `` // app.module.ts
```

```
// before import from '@pxblue/angular-components'
```

```
// after import from '@brightlayer-ui/angular-components'
```

```
` ``
```

```
` `` // Update component name from `` to `` // before
```

```
// after ` ``
```

## @brightlayer-ui/angular-template-authentication

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- You can use these templates only with **brighlayer-ui** scoped version of the CLI:

```
``` // before npx -p @pxblue/cli pxb new angular --template=authentication
```

```
// after npx -p @brightlayer-ui/cli blui new angular --template=authentication ```
```

## @brightlayer-ui/angular-template-blank

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- You can use these templates only with **brighlayer-ui** scoped version of the CLI:

```
``` // before npx -p @pxblue/cli pxb new angular --template=blank
```

```
// after npx -p @brightlayer-ui/cli blui new angular --template=blank ```
```

## @brightlayer-ui/angular-template-routing

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- You can use these templates only with **brighlayer-ui** scoped version of the CLI:

```
``` // before npx -p @pxblue/cli pxb new angular --template=routing
```

```
// after npx -p @brightlayer-ui/cli blui new angular --template=routing ```
```

## @brightlayer-ui/angular-themes

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- Code snippet for using theme in your angular application.

```js // Angular.json file "styles" entries (there is one under "build" and one under "test")  
to include the Brightlayer UI themes and Open Sans reference:

```
// before "styles": [ "src/styles.scss", "./node_modules/@pxblue/angular-themes/theme.scss",  
"./node_modules/@pxblue/angular-themes/open-sans.scss" ],  
  
// after "styles": [ "src/styles.scss", "./node_modules/@brightlayer-ui/angular-themes/theme.scss",  
"./node_modules/@brightlayer-ui/angular-themes/open-sans.scss" ],  
```
```

Alternatively, you can also import the theme file in your top-level SCSS file.

```
```scss // in styles.scss or your top-level scss file  
  
//before import '~@pxblue/angular-themes/theme.scss' import '~@pxblue/angular-themes/open-sans.scss'  
  
//after import '~@brightlayer-ui/angular-themes/theme.scss' import '~@brightlayer-ui/angular-themes/open-sans.scss' ```
```

Change `pxb-` class name prefixes to `blui-` of your application's body element:

```
```HTML // Default Theme
```

=>

```
// Dark Theme => ```
```

## @brightlayer-ui/chartjs

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- The below code snippets indicate the changes to be made for using @brightlayer-ui/chartjs package in your angular application.

```
``` // package.json
```

```
@pxblue/chartjs => @brightlayer-ui/chartjs ```
```

To use this library, specify in your HTML where you want your graph to be rendered:

```

```

Update import statement to use `@brightlayer-ui/chartjs` package:

``` // before import from '@pxblue/chartjs';

// after import from '@brightlayer-ui/chartjs'; ```

and then use these functions to create Brightlayer UI themed chart configurations.

``` Replace `@pxblue/chartjs` to `@brightlayer-ui/chartjs` for loading default sample data to render a chart

// before import from '@pxblue/chartjs' ... const lineChartConfig = createLineChartConfig(); drawChart(lineChartConfig, 'graphId');

// after import from '@brightlayer-ui/chartjs' ... const lineChartConfig = createLineChartConfig(); drawChart(lineChartConfig, 'graphId'); ```

This will use default sample data to render a chart in your application.

## @brightlayer-ui/cli

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- The below code snippets indicate the changes to be made for using Brightlayer UI CLI command.

``` \\ before npx -p @pxblue/cli pxb new ```

``` after npx -p @brightlayer-ui/cli blui new ```

- Change all `pxb` reference to `blui` from CLI command.

``` pxb help => `blui help` pxb version => `blui version` pxb new => blui new ```

## @brightlayer-ui/colors

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- The below code snippets indicate the changes to be made for using `@brightlayer-ui/colors` package.

```
`` ` // package.json
```

```
@pxblue/colors => @brightlayer-ui/colors `` `
```

Angular

```
`` ` // In styles.scss or your top-level sass file update import statement
```

```
// before @import '~@pxblue/colors/palette.scss'
```

```
// after @import '~@brightlayer-ui/colors/palette.scss' ...
```

```
// Change all `$pxb-` color definition prefixes to `$blui-`
```

```
// before background-color: map-get($pxb-blue, 500)
```

```
// after background-color: map-get($blui-blue, 500) `` `
```

React or React Native

```
`` ` // before import * as Colors from '@pxblue/colors';
```

```
// after import * as Colors from '@brightlayer-ui/colors'; ...
```

```
`` `
```

## @brightlayer-ui/colors-branding

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- The below code snippets indicate the changes to be made for using `@brightlayer-ui/colors-branding` package.

```
`` ` // package.json
```

```
@pxblue/colors-branding => @brightlayer-ui/colors-branding `` `
```

## Angular

```
`` ` // In styles.scss or your top-level sass file update import statement

// before @import '~@pxblue/colors-branding/palette.scss'

// after @import '~@brightlayer-ui/colors-branding/palette.scss' ...

// Change all ` $pxb-` color definition prefixes to ` $blui-`

// before background-color: map-get($pxb-blue, 500)

// after background-color: map-get($blui-blue, 500) `` `
```

## React or React Native

```
`` ` // before import * as Colors from '@pxblue/colors-branding';

// after import * as Colors from '@brightlayer-ui/colors-branding'; ...

`` `
```

# @brightlayer-ui/cra-template-authentication

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- This template can be used with Create React App.

```
`` ` // before npx create-react-app myapp --template @pxblue-ui/authentication

// after npx create-react-app myapp --template @brightlayer-ui/authentication `` `
```

- Using the Brightlayer UI CLI:

```
`` ` // before npx -p @pxblue/cli pxb new react --name=myapp --language=js --
template=authentication

// after npx -p @brightlayer-ui/cli blui new react --name=myapp --language=js --
template=authentication `` `
```

## @brightlayer-ui/cra-template-authentication-typescript

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- This template can be used with Create React App:

```
``` // before npx create-react-app myapp --template @pxblue/authentication-typescript  
  
//after npx create-react-app myapp --template @brightlayer-ui/authentication-typescript  
```
```

- Using the Brightlayer UI CLI:

```
``` // before npx -p @pxblue/cli pxb new react --name=myapp --language=ts --  
template=authentication
```

```
// after npx -p @brightlayer-ui/cli blui new react --name=myapp --language=ts --  
template=authentication ```
```

## @brightlayer-ui/cra-template-blank

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- This template can be used with Create React App:

```
``` // before npx create-react-app myapp --template @pxblue/blank  
  
// after npx create-react-app myapp --template @brightlayer-ui/blank ```
```

- Using the Brightlayer UI CLI:

```
``` // before npx -p @pxblue/cli pxb new react --name=myapp --language=js --  
template=blank
```

```
// after npx -p @brightlayer-ui/cli blui new react --name=myapp --language=js --  
template=blank ```
```



## @brightlayer-ui/cra-template-blank-typescript

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- This template can be used with Create React App:

```
``` // before npx create-react-app myapp --template @pxblue/blank-typescript
```

```
// after npx create-react-app myapp --template @brightlayer-ui/blank-typescript ```
```

- Using the Brightlayer UI CLI:

```
``` // before npx -p @pxblue/cli pxb new react --name=myapp --language=ts --  
template=blank
```

```
// after npx -p @brightlayer-ui/cli blui new react --name=myapp --language=ts --  
template=blank ```
```

## @brightlayer-ui/cra-template-routing

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- This template can be used with Create React App:

```
``` // before npx create-react-app myapp --template @pxblue/routing
```

```
// after npx create-react-app myapp --template @brightlayer-ui/routing ```
```

- Using the Brightlayer UI CLI:

```
``` // before npx -p @pxblue/cli pxb new react --name=myapp --language=js --  
template=routing
```

```
// after npx -p @brightlayer-ui/cli blui new react --name=myapp --language=js --  
template=routing ```
```

## @brightlayer-ui/cra-template-routing-typescript

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- This template can be used with Create React App:

```
``` // before npx create-react-app myapp --template @pxblue/routing-typescript
```

```
// after npx create-react-app myapp --template @brightlayer-ui/routing-typescript ```
```

- Using the Brightlayer UI CLI:

```
``` // before npx -p @pxblue/cli pxb new react --name=myapp --language=ts --  
template=routing
```

```
// after npx -p @brightlayer-ui/cli blui new react --name=myapp --language=ts --  
template=routing ```
```

## @brightlayer-ui/eslint-config

- Change package namespace from `@pxblue` to `@brightlayer-ui`.

```
``` // package.json
```

```
@pxblue/eslint-config => @brightlayer-ui/eslint-config ```
```

- Update the following `.eslintrc.js` to the root of your project.

```
``` // before module.exports = ;
```

```
// after module.exports = ; ```
```

## @brightlayer-ui/icons

- Change package namespace from `@pxblue` to `@brightlayer-ui`.

```
``` // package.json
```

@pxblue/icons => @brightlayer-ui/icons ```

- Code snippet to use icon font in angular application.

```
``` // angular.json // before "styles": [ "src/styles.scss",  
"./node_modules/@pxblue/icons/iconfont/PxblueUIIcons.css" ],  
  
// after "styles": [ "src/styles.scss", "./node_modules/@brightlayer-  
ui/icons/iconfont/BrightlayerUIIcons.css" ], ```
```

- Code snippet to use icon font in React application.

```
``` // index.js  
  
// before require('@pxblue/icons/iconfont/PxblueUIIcons.css');  
  
// after require('@brightlayer-ui/icons/iconfont/BrightlayerUIIcons.css'); ```  
  
- Change all `pxb-` class name prefixes to `blui-` in component  
  
``` // before  
  
// after ```
```

## @brightlayer-ui/icons-mui

- Change package namespace from `@pxblue` to `@brightlayer-ui`.

```
``` // package.json  
  
@pxblue/icons-mui => @brightlayer-ui/icons-mui ```
```

- Replace package name from `@pxblue` to `@brightlayer-ui` to use mui icon in React application.

```
``` \\ before import myIcon from '@pxblue/icons-mui/MyIcon';  
  
\\ after import myIcon from '@brightlayer-ui/icons-mui/MyIcon';  
  
```
```

## @brightlayer-ui/icons-svg

- Change package namespace from `@pxblue` to `@brightlayer-ui`.

```
``` // package.json
```

```
@pxblue/icons-svg => @brightlayer-ui/icons-svg ```
```

- Code snippet to use svg icon in Angular application.

```
``` // app.module.ts import from '@angular/material/icon'; import from '@angular/common/http'; ... imports:[ MatIconModule, HttpClientModule ] ```
```

The simplest way to use these SVGs in Angular is to register them with the `matIconRegistry` so they can be used with the `<img>` tag. You can register icons individually, or as the entire Brightlayer UI set:

```
``` import from "@angular/core"; import from "@angular/material/icon"; import from "@angular/platform-browser";
```

```
declare var require: any; const icon = require("@brightlayer-ui/icons-svg/breaker.svg").default; // individual icon const iconSet = require("@brightlayer-ui/icons-svg/icons.svg").default; // full set ```
```

Then, in your constructor, register the icon or the icon set. It will then be available for use as a `<img>`.

#### Individual Icon

```
``` // app.component.ts export class AppComponent { ```
```

```
``` // app.component.html ```
```

#### Entire Icon Set

```
``` // app.component.ts export class AppComponent { ```
```

```
``` // app.component.ts ```
```

## @brightlayer-ui/mapbox

- Change package namespace from `@pxblue` to `@brightlayer-ui`.

```
``` // package.json
```

```
@pxblue/mapbox => @brightlayer-ui/mapbox ```
```

- Code snippet to use mapbox in Angular application. Import the theme into your target typescript file:

```
``` // app.component.ts
```

```
//      before      declare      var      require;      const      defaultTheme      =  
require('@pxblue/mapbox/default.json');          const      darkTheme      =  
require('@pxblue/mapbox/dark.json');
```

```
// after declare var require; const defaultTheme = require('@brightlayer-  
ui/mapbox/default.json'); const darkTheme = require('@brightlayer-ui/mapbox/dark.json');  
```
```

Bind the theme to the style attribute of the ``:

```
``` // app.component.html ```
```

- Code snippet to use mapbox in React application.

```
``` // App.js // before const defaultTheme = require('@pxblue/mapbox/default.json');  
const darkTheme = require('@pxblue/mapbox/dark.json');
```

```
// after const defaultTheme = require('@brightlayer-ui/mapbox/default.json'); const  
darkTheme = require('@brightlayer-ui/mapbox/dark.json'); ```
```

Apply the theme variable to the style parameter of the mapboxgl configuration:

```
``` // App.js const map = new mapboxgl.Map(); ```
```

## @brightlayer-ui/ng-progress-icons

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- Code snippet to use progress icon in angular application.

```
``` // package.json
```

```
@pxblue/ng-progress-icons => @brightlayer-ui/ng-progress-icons ```
```

Replace package name from `@pxblue` to `@brightlayer-ui` to use @brightlayer-ui/ng-progress-icons angular application.

```
``` app.module.ts // before import from '@pxblue/ng-progress-icons';
```

```
// after import from '@brightlayer-ui/ng-progress-icons'; ```
```

## @brightlayer-ui/prettier-config

- Change package namespace from `@pxblue` to `@brightlayer-ui`.

```
``` // package.json
```

```
@pxblue/prettier-config => @brightlayer-ui/prettier-config ```
```

- Update the following script to your `package.json`.

```
``` // before "prettier": "@pxblue/prettier-config", ...
```

```
// after "prettier": "@brightlayer-ui/prettier-config", ... ```
```

## @brightlayer-ui/publish

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- Change `pxb-` command prefixes to `blui-`.

```
``` // before npx -p @pxblue/publish pxb-publish -b
```

```
// after npx -p @brightlayer-ui/publish blui-publish -b ```
```

## @brightlayer-ui/react-auth-shared

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- ``` // package.json

@pxblue/react-auth-shared => @brightlayer-ui/react-auth-shared ```

## @brightlayer-ui/react-auth-workflow

- Change package namespace from `@pxblue` to `@brightlayer-ui`.

``` // package.json

@pxblue/react-auth-workflow => @brightlayer-ui/react-auth-workflow ```

- Change Local Storage key `pxb-auth-i18nextLng` to `blui-auth-i18nextLng`.

```
```tsx      //      before      localStorage.setItem('pxb-auth-i18nextLng',      'fr');  
localStorage.removeItem('pxb-auth-i18nextLng');
```

```
// after localStorage.setItem('blui-auth-i18nextLng', 'fr'); localStorage.removeItem('blui-  
auth-i18nextLng'); ```
```

- Change all `pxb` namespace to `blui` for overriding workflow strings / translations.

```
```tsx // before import i18n from 'i18next'; ... i18n.addResourceBundle('en', 'pxb', }, true,  
true);
```

```
// after import i18n from 'i18next'; ... i18n.addResourceBundle('en', 'blui', }, true, true);  
```
```

## @brightlayer-ui/react-components

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- For e.g. follow the code snippet to use App Bar in your react application.

``` // package.json

@pxblue/react-components => @brightlayer-ui/react-components ``

`` // component // before import from '@pxblue/react-components'; ...

*[AppBar Component]*

// after import from '@brightlayer-ui/react-components'; ...

*[AppBar Component]*

...

## @brightlayer-ui/react-native-progress-icons

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- For e.g follow the code snippet for using progress icons in your application.

`` // package.json

@pxblue/react-native-progress-icons => @brightlayer-ui/react-native-progress-icons ``

`` // Replace `@pxblue` to `@brightlayer-ui` in import statement of component.

// before import from '@pxblue/react-native-progress-icons'; ...

*[Battery Component - Interactive React component]*

// after import from '@brightlayer-ui/react-native-progress-icons'; ...

*[Battery Component - Interactive React component]*

...



## @brightlayer-ui/react-native-template-authentication

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- You can use these templates with the Brightlayer UI CLI:
- `` ` // before npx -p @pxblue/cli pxb new react-native --name=myapp --cli=rnc --language=js --template=authentication

// after npx -p @brightlayer-ui/cli blui new react-native --name=myapp --cli=rnc --language=js --template=authentication ``

## @brightlayer-ui/react-native-template-authentication-typescript

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- You can use these templates with the Brightlayer UI CLI:
- `` ` // before npx -p @pxblue/cli pxb new react-native --name=myapp --cli=rnc --language=ts --template=authentication

// after npx -p @brightlayer-ui/cli blui new react-native --name=myapp --cli=rnc --language=ts --template=authentication ``

## @brightlayer-ui/react-native-template-blank

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- You can use these templates with the Brightlayer UI CLI:
- `` ` // before npx -p @pxblue/cli pxb new react-native --name=myapp --cli=rnc --language=js --template=blank

```
// after npx -p @brightlayer-ui/cli blui new react-native --name=myapp --cli=rnc --  
language=js --template=blank ```
```

## @brightlayer-ui/react-native-template-blank-typescript

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- You can use these templates with the Brightlayer UI CLI:
- ``` // before npx -p @pxblue/cli pxb new react-native --name=myapp --cli=rnc --  
language=ts --template=blank

```
// after npx -p @brightlayer-ui/cli blui new react-native --name=myapp --cli=rnc --  
language=ts --template=blank ```
```

## @brightlayer-ui/react-native-template-routing

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- You can use these templates with the Brightlayer UI CLI:
- ``` // before npx -p @pxblue/cli pxb new react-native --name=myapp --cli=rnc --  
language=js --template=routing

```
// after npx -p @brightlayer-ui/cli blui new react-native --name=myapp --cli=rnc --  
language=js --template=routing ```
```

## @brightlayer-ui/react-native-template-routing-typescript

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- You can use these templates with the Brightlayer UI CLI:
- ``` // before npx -p @pxblue/cli pxb new react-native --name=myapp --cli=rnc --language=ts --template=routing

// after npx -p @brightlayer-ui/cli blui new react-native --name=myapp --cli=rnc --language=ts --template=routing ```

## @brightlayer-ui/react-native-themes

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- Follow the code snippet for using `brightlayer-ui/react-native-themes` package.

``` // package.json

@pxblue/react-native-themes => @brightlayer-ui/react-native-themes ```

- Code snippet to use @brightlayer-ui/react-native-themes package.

```tsx // before import from 'react-native-paper'; import \* as PXBThemes from '@pxblue/react-native-themes'; ... // Default Theme

*[ThemeProvider Component]*

*[App Component - Interactive React component]*

// after import from 'react-native-paper'; import \* as BLUIThemes from '@brightlayer-ui/react-native-themes'; ... // Default Theme

*[ThemeProvider Component]*

*[App Component - Interactive React component]*

...

## @brightlayer-ui/react-native-vector-icons

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- Change Icon type from `PXBlueIcon` to `BLUIIcon`.
- Follow the below instructions to use `@brightlayer-ui/react-native-vector-icons` package in iOS projects. We recommend doing this via CocoaPods:

```
``` // Podfile pod 'RNBLUIVectorIcons', :path => '../node_modules/@brightlayer-ui/react-native-vector-icons' ```
```

Edit your `Info.plist`

- Under the "Fonts Provided By Application" property (which you should have added when setting up RNVI), Rename PXBlueIcons.ttf to BrightlayerUIIcons.ttf.

After updating these files you will need to install the Pods and recompile your application:

```
``` cd ios pod install ```
```

- Follow the below instructions to use `@brightlayer-ui/react-native-vector-icons` package in Android projects. We recommend doing this via Gradle: Replace `@pxblue` reference to `@brightlayer-ui` in `android/app/build.gradle` ( NOT `android/build.gradle` ) it will look like below code.

```
```      apply      from:      "../..node_modules/@brightlayer-ui/react-native-vector-icons/fonts.gradle" ```
```

Usage

```
```tsx import BLUIIcon from '@brightlayer-ui/react-native-vector-icons'; ...
```

*[BLUIIcon Component - Interactive React component]*

```

## @brightlayer-ui/react-progress-icons

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- For e.g follow the code snippet for using progress icons in your application.

``` // package.json

@pxblue/react-progress-icons => @brightlayer-ui/react-progress-icons ```

Replace `@pxblue` to `@brightlayer-ui` in import statement.

```tsx // before import from '@pxblue/react-progress-icons'; ...

*[Battery Component - Interactive React component]*

// after import from '@brightlayer-ui/react-progress-icons'; ...

*[Battery Component - Interactive React component]*

```

## @brightlayer-ui/react-themes

- Change package namespace from `@pxblue` to `@brightlayer-ui`.
- Follow the code snippet for using `brightlayer-ui/react-themes` package.

``` // package.json

@pxblue/react-themes => @brightlayer-ui/react-themes ```

```tsx // before ... import \* as BLUIThemes from '@brightlayer-ui/react-themes'; import '@brightlayer-ui/react-themes/open-sans'; ...

*[ThemeProvider Component]*

*[App Component - Interactive React component]*

```
// after ... import * as BLUIThemes from '@brightlayer-ui/react-themes'; import  
"@brightlayer-ui/react-themes/open-sans"; ...
```

*[ThemeProvider Component]*

*[App Component - Interactive React component]*

...

## @brightlayer-ui/storybook-rtl-addon

- Change package namespace from `@pxblue` to `@brightlayer-ui`.

```
``` // package.json
```

```
@pxblue/storybook-rtl-addon => @brightlayer-ui/storybook-rtl-addon ```
```

- Replace `@pxblue` reference to `@brightlayer-ui` in import statement in your application.

```
```ts // before import from '@pxblue/storybook-rtl-addon';
```

```
// after import from '@brightlayer-ui/storybook-rtl-addon'; ```
```

## @brightlayer-ui/storybook-themes

- Change package namespace from `@pxblue` to `@brightlayer-ui`.

```
`` ` // package.json
```

```
@pxblue/storybook-themes => @brightlayer-ui/storybook-themes `` `
```

- Replace `pxblueTheme` reference with `bluiTheme` in config file, check below code for reference:

```
`` `tsx // before import from '@pxblue/storybook-themes' ...
```

```
addParameters(, {});
```

```
// after import from '@brightlayer-ui/storybook-themes' ...
```

```
addParameters(, {}); `` `
```

## @brightlayer-ui/symbols

- Change package namespace from `@pxblue` to `@brightlayer-ui`.

```
`` ` // package.json
```

```
@pxblue/symbols => @brightlayer-ui/symbols `` `
```

## @brightlayer-ui/symbols-mui

- Change package namespace from `@pxblue` to `@brightlayer-ui`.

```
`` ` // package.json
```

```
@pxblue/symbols-mui => @brightlayer-ui/symbols-mui `` `
```

Replace `@pxblue-ui` reference with `@brightlayer-ui`, check below code for reference:

```
`` `tsx // before const symbol = require('@pxblue-ui/symbols/battery.svg'); // after const  
symbol = require('@brightlayer-ui/symbols/battery.svg'); `` `
```

## @brightlayer-ui/tag

- Change package namespace from `@pxblue` to `@brightlayer-ui`.

```
``` // package.json
```

```
@pxblue/tag => @brightlayer-ui/tag ```
```

- Change `pxb-` reference to `blui-`. Follow the command change for reference.

```
``` // before npx -p @pxblue/tag pxb-tag -b -s
```

```
// after npx -p @brightlayer-ui/tag blui-tag -b -s ```
```

## @brightlayer-ui/types

- Change package namespace from `@pxblue` to `@brightlayer-ui`.

```
``` // package.json
```

```
@pxblue/types => @brightlayer-ui/types ```
```

- Change `PXBlueColor` type to `BLUIColor`. Follow the code snippet for reference.

```
``` // before import from '@pxblue/types'
```

```
// after import from '@brightlayer-ui/types'; ```
```