

Customize Widget

Customize the look and feel of the widget to match the design of your dApp. Li.FI Widget supports visual customization, allowing you to match your web app's design. The widget's layout stays consistent, but you can modify colors, fonts, border radius, container styles, and more.

Start customizing the widget by tweaking some of the following options:

...

```
Copy interfaceWidgetConfig{ // ... // modifies the style of the container element containerStyle?:CSSProperties; // provides
colors, fonts, border-radius theme?:ThemeConfig; // sets default appearance - light, dark, or auto appearance?:Appearance;
// disables parts of the UI disabledUI?:DisabledUIType[]; // hides parts of the UI hiddenUI?:HiddenUIType[]; // makes parts of
the UI required requiredUI?:RequiredUIType[]; }
```

...

Check out our [Li.FI Playground](#) to play with customization options in real time.

Container style

Container style can be changed by using the corresponding `containerStyle` option. In the example below, we adjust border and border-radius properties.

...

```
Copy import{ LiFiWidget,WidgetConfig }from'@lifi/widget'; import{ useMemo }from'react';
```

```
exportconstWidgetPage=()=>=>{ constwidgetConfig:WidgetConfig=useMemo(()=>({ containerStyle:{ boxShadow:'0px 8px 32px
rgba(0, 0, 0, 0.08)', borderRadius:'16px', }, },[]));
```

```
return( );};
```

...

Before After

Theme

Let's move further and modify the inner elements' primary and secondary colors, font family, and border radius.

...

```
Copy import{ LiFiWidget,WidgetConfig }from'@lifi/widget'; import{ useMemo }from'react';
```

```
exportconstWidgetPage=()=>=>{ constwidgetConfig:WidgetConfig=useMemo(()=>({ containerStyle:{ boxShadow:'0px 8px 32px
rgba(0, 0, 0, 0.08)', borderRadius:'16px', }, theme:{ palette:{ primary:{ main:'#7B3FE4'}, secondary:{ main:'#F5B5FF'},
background:{ paper:'#000000',// bg color for cards default:'#000000',// bg color container }, grey:{ 300:'#000000',// border
light theme 800:'#000000',// border dark theme }, }, shape:{ borderRadius:0, borderRadiusSecondary:0, }, typography:{
fontFamily:'Comic Sans MS', }, }, },[]));
```

```
return( );};
```

...

Before After

Appearance

The widget has complete Dark Mode support out of the box. By default, the `appearance` option is set to `auto`, matching the user's system settings for dark and light modes. There is also an option to disable UI for manually switching appearance, for example, if your web app only needs dark mode.

Now let's set the default appearance to dark and hide the appearance settings UI.

...

```
Copy import{ LiFiWidget,WidgetConfig }from'@lifi/widget'; import{ useMemo }from'react';
```

```
export const WidgetPage = () => {
  const widgetConfig: WidgetConfig = useMemo(() => ({
    containerStyle: {
      boxShadow: '0px 8px 32px rgba(0, 0, 0, 0.08)',
      borderRadius: '16px',
    },
    theme: {
      palette: {
        primary: { main: '#7B3FE4' },
        secondary: { main: '#F5B5FF' },
      },
      shape: {
        borderRadius: 0,
        borderRadiusSecondary: 0,
      },
      typography: {
        fontFamily: 'Comic Sans MS',
      },
      appearance: 'dark',
      hiddenUI: ['appearance'],
    },
  })), []);

  return ( );
};

...

```

Before After Before After If you are interested in additional customization options for your service, reach out via our [Discord](#) or [Partnership](#) page. As you can see, widget customization is pretty straightforward. We are eager to see what combinations you will come up with while we continue constantly adding new customization options.

Last updated 5 months ago On this page * [Container style](#) * [Theme](#) * [Appearance](#)

Was this helpful? [Export as PDF](#)