next.js)

- Widget Skeleton
- Remix, Nuxt, Gatsby etc.

Was this helpful? Export as PDF

Compatibility with Next.js, Remix, Nuxt, etc.

Integrate the widget with your Next.js, Remix or Nuxt application

Next.js

LI.FI Widget is fully compatible with Next.js applications and requires minimal configuration for seamless integration.

Due to the nature of server-side rendering (SSR) in Next.js and how different wallet libraries manage their connections to wallet extensions, the LI.FI Widget needs to be specifically rendered on the client side. To achieve this, make use of the'use client' directive, which ensures that the widget component is only rendered in the client-side environment.

Please take a look at our complete example in the widget repositoryhere . Let's take a look at the example of how you can add the widget to a Next.js page:

Widget.tsx ClientOnly.tsx page.tsx ```

Copy 'use client';

importtype{ WidgetConfig }from'@lifi/widget'; import{ LiFiWidget,WidgetSkeleton }from'@lifi/widget'; import{ ClientOnly }from'./ClientOnly';

exportfunctionWidget() { constconfig={ appearance:'light', theme:{ container:{ boxShadow:'0px 8px 32px rgba(0, 0, 0, 0.08)', borderRadius:'16px', }, }, }asPartial;

return(

}>

); }

Copy import{ useSyncExternalStore,typePropsWithChildren }from'react';

functionsubscribe() { return()=>{}; }

/* * Return a boolean indicating if the JS has been hydrated already. * When doing Server-Side Rendering, the result will always be false. * When doing Client-Side Rendering, the result will always be false on the * first render and true from then on. Even if a new component renders it will * always start with true. / exportfunctionuseHydrated() { returnuseSyncExternalStore(subscribe, ()=>true, ()=>false,); }

interfaceClientOnlyPropsextendsPropsWithChildren{ fallback?:React.ReactNode; }

/* * Render the children only after the JS has loaded client-side. Use an optional * fallback component if the JS is not yet loaded. / exportfunctionClientOnly({ children,fallback=null}:ClientOnlyProps) { consthydrated=useHydrated(); returnhydrated?children:fallback; }

Copy import{ Widget }from'./components/Widget';

exportdefaultfunctionHome() { return(

); }

```

Widget Skeleton

As you can see from the Next.js example, for users convenience we provide theWidgetSkeleton component that can be used as a fallback while the main widget component is being loaded.

TheWidgetSkeleton component is currently only tested within the Next.js environment and might not work with other SSR
```

frameworks. Widget Skeleton

Remix, Nuxt, Gatsby etc.

Please check out our complete examples for Remix, Nuxt and Gatsby.

- Remix (Example
-)
- Nuxt.js (Example
-)
- Gatsby (Example
-)

We are continuously working on improving our support for more frameworks.

React Router Compatibility Last updated5 months ago