overview)

Migrate from v2 to v3

Migration guide for upgrading LI.FI Widget v2 to v3

Overview

LI.FI Widget v3 introduces a comprehensive overhaul of its core, enhancing compatibility with popular account management libraries likeWagmi andRainbowKit and incorporating many new features, including multi-ecosystem support starting withSolana . Therefore, there are some breaking changes and deprecations to be aware of, as outlined in this guide. Additionally, we encourage you to read the updated documentation, which includes new features that are not mentioned here.

To get started, install the latest version of Widget.

yarn pnpm bun npm ```

Copy yarnadd@lifi/widget

Copy pnpmadd@lifi/widget

Copy bunadd@lifi/widget

Copy npminstall@lifi/widget

```

Configuration

Renamed variants

The following widget variants were renamed to better reflect their functionality:

default is nowcompact

expandable is nowwide

SeeSelect Widget Variants for more details.

Updated toAddress type

In Widget v3 we updated destination wallet functionality to have wallet bookmarks, recently used addresses and more. Previously, thetoAddress option had a typestring . To extend the feature in v3 we added a specific type for this field.

```

Copy // Widget v2 interfaceWidgetConfig{ // ... toAddress?:string; // ... }

// Widget v3 interfaceToAddress{ name?:string; address:string; chainType:ChainType; logoURI?:string; }

interfaceWidgetConfig{ // ... toAddress?:ToAddress; // ... }

```

Updated subvariantOptions type

In Widget v3 we extended the functionality of subvariants andsubvariantOptions required type update. Previously,subvariantOptions supported onlysplit subvariant and hadSplitSubvariantOptions type and now we added support forcustom subvariant with possibly more subvariant options coming in the future.

```

Copy // Widget v2 typeSplitSubvariantOptions='bridge'|'swap'; interfaceWidgetConfig{ // ... subvariantOptions?:SplitSubvariantOptions; // ... }

// Widget v3 typeSplitSubvariant='bridge'|'swap'; typeCustomSubvariant='checkout'|'deposit'; interfaceSubvariantOptions{ split?:SplitSubvariant; custom?:CustomSubvariant; }

interfaceWidgetConfig{ // ... subvariantOptions?:SubvariantOptions; // ... }

```

SDK configuration

Following the release of LI.FI SDK v3 there are some changes to SDK configuration in the widget.

SDKConfig type is nowWidgetSDKConfig .

defaultRouteOptions renamed torouteOptions .

Removed pre-built drawer button

Previously in Widget v2 there were two ways of controlling the drawer. The first one was pre-built button which came with drawer variant to open and close it. The second one was hiding the button viahiddenUI option and controlling the drawer by attaches the ref. Since the title and positioning often required some adjustments we decided to remove pre-built drawer button in favor of controlling the drawer with external button.

Widget v2 drawer variant pre-built button Here is the example of controlling the drawer with ref and you can customize the button as you like.

```

Copy exportconstWidgetPage=()=>{ constdrawerRef=useRef(null);

consttoggleWidget=()=>{ drawerRef.current?.toggleDrawer(); };

return(

Open LI.FI Widget

); }

```

We also removedHiddenUI.DrawerButton option since there is no default button anymore. However, we added a new optionHiddenUI.DrawerCloseButton to hide the close button inside the drawer.

Theming

In Widget v3, we revamped customization and theming possibilities and did a small cleanup.

The top-levelcontainerStyle option is moved under the umbrella of thetheme option and was renamed tocontainer .

```

Copy // Widget v2 exportconstWidgetPage=()=>{ return(1px solid rgb(234, 234, 234), borderRadius:'16px', }, }} />); };

// Widget v3 exportconstWidgetPage=()=>{ return(1px solid rgb(234, 234, 234), borderRadius:'16px', }, }, }} />); };

```

Also,theme option type was renamed fromThemeConfig toWidgetTheme .
```

Please check out more inCustomize Widget section.

Miscellaneous

disableLanguageDetector configuration option was removed. Language detection should now be inherited from the integration dApp.

Wallet Management

Moved from Ethers.js to Wagmi

We dropped support for Ethers.js. Instead, Widget now has a first-class Wagmi and all Wagmi-based libraries support like RainbowKit. You can still use Ethers.js in your project and convert Signer/Provider object to use Wagmi'sinjected connector before wrapping the Widget withWagmiProvider . However, we suggest moving your dApp's wallet management to use Wagmi as a more future proof solution.

Widget v2 setup with Ethers.js

```

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

exportconstWidgetPage=()=>{ const{account,connect,disconnect,switchChain}=useWallet();

constwidgetConfig=useMemo(():WidgetConfig=>{ return{ walletManagement:{ signer:account.signer, connect:async()=>{ constsigner=awaitconnect(); returnsigner; }, disconnect:async()=>{ awaitdisconnect(); }, switchChain:async(chainId:number)=>{ awaitswitchChain(chainId); if(account.signer) { returnaccount.signer; }else{ throwError('No signer object is found after the chain switch.'); } }, }, }; },[account.signer,connect,disconnect,switchChain]);

return( ); }

```

Widget v3 setup with Wagmi

No more complicated callbacks, just wrap the Widget inWagmiProvider and you are set. If you already have one in your dApp, just make sure to keep the Wagmi chains configuration in sync with the Widget chain list, so all functionality like switching chains can keep working.

SeeWallet Management for more details.

```

Copy import{ LiFiWidget }from'@lifi/widget'; import{ createClient }from'viem'; import{ WagmiProvider,createConfig,http }from'wagmi'; import{ mainnet }from'wagmi/chains'; import{ injected }from'wagmi/connectors';

constwagmiConfig=createConfig({ // Make sure to provide the full list of chains // you would like to support in the Widget // and keep them in sync, so all functionality // like switching chains can work correctly. chains:[mainnet], connectors: [injected()], client({ chain }) { returncreateClient({ chain,transport:http() }); }, });

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

```

Updated wallet management configuration

Since we don't need to provide callbacks and signers to widget configuration,walletManagement option was renamed towalletConfig and now has the following interface:

```

Copy interfaceWidgetWalletConfig{ // Can be used to open the external wallet menu, // if the user is not connected and dApp uses external wallet management onConnect():void; // Provide your projectId and other WalletConnect properties // when using Widget's internal wallet management walletConnect?:WalletConnectParameters; // Provide your app name and other Coinbase properties // when using Widget's internal wallet management coinbase?:CoinbaseWalletParameters; }

```

Dependencies

Wagmi is a new peer dependency

Since we moved fromethers.js toWagmi we added it to our peer dependencies.

Dropped CommonJS support

LI.FI Widget v3 no longer publishes a separate CommonJS build since most of the modern front-end tooling supports ESM and ESM is the future. SeeSindre Sorhus' guide for more info about switching to ESM.

Install Widget Last updated4 months ago