Wallet Management

Configure your widget for seamless wallet management To manage wallet connection, switching chains, etc., the widget uses the@lifi/wallet-management package internally and provides UI to connect to a wallet.

If you already have your wallet management UI, you can utilize a set of callbacks provided for your convenience and help the widget communicate with your wallet management solution.

The example below shows how to preconfigure a basic wallet management.

```

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( ); }

```

In the next major version, we will migrate towagmi and simplify wallet management integration.

Check out ourexample of how to integrate yourwagmi wallet management with the widget. Also, see thewagmi ethers adapters (react / core) documentation for wallet client -> signer conversion. Some callbacks are optional, and we will use our implementation if you don't provide any.

```

Copy import{ Signer }from'ethers';

exportinterfaceWidgetWalletManagement{ // will be called when the user presses Connect Wallet button connect():Promise; // will be called to disconnect the wallet by the user or internally disconnect():Promise; // will be called to switch the current wallet chain by the user or internally switchChain?(chainId:number):Promise; // will be called to add a token to a wallet if not present addToken?(token:Token,chainId:number):Promise; // will be called to add a chain to a wallet if not present addChain?(chainId:number):Promise; // we retrieve information about the user account from ethers Signer abstraction signer?:Signer; }

```

You can find an example of how we use these callbacks to communicate withJumper.exchange external wallet management solution in the repositoryhere .

Last updated5 months ago On this page Was this helpful?Export as PDF