

Usage - Wallet Services Plugin

Fiat on Ramp

Displays the TopUp modal, allowing users to select their local currency and specify the token to top up their wallet

Parameters

Name Description
receiveWalletAddress? Specifies the recipient's address. By default, it is set to the currently connected address.
tokenList? Specifies the tokens to display in the list. By default, all tokens are shown. Use the ticker name to specify which tokens to display, such as[USDC, USDT, ETH] . Please checkout the coverage details for[full list of supported networks and tokens](#) .
fiatList? Specifies the available fiat currencies enabled for purchase. Use the currency acronym to define which fiat currencies to display, such as[USD, SGD, INR, JPY] . Please checkout the coverage details for[full list of supported currencies](#) .
show Determines whether the checkout UI is displayed. This can be used to programmatically control its visibility.

Usage

```
import
{
  WalletServicesPlugin
}
from
"@web3auth/wallet-services-plugin" ;

const walletServicesPlugin =
new
WalletServicesPlugin ( ) ; web3auth . addPlugin ( walletServicesPlugin ) ;

// Add the plugin to web3auth

await walletServicesPlugin . showCheckout ( ) ;

// Opens the TopUp modal
```

Swap

Displays the Swap modal, allowing users to swap between tokens.

Parameters

Name Description
show Controls whether the Swap UI is visible. Set to true to display the Swap interface, or false to hide it.
fromToken? Specifies the token being swapped from. Optional field.
toToken? Specifies the token being swapped to. Optional field.
fromValue? The amount of the fromToken to be swapped. Optional field.
toAddress? The recipient's address for the toToken . Optional field.

Usage

```
import
{
  WalletServicesPlugin
}
from
"@web3auth/wallet-services-plugin" ;

const walletServicesPlugin =
```

new

```
WalletServicesPlugin ( ) ; web3auth . addPlugin ( walletServicesPlugin ) ;
```

```
// Add the plugin to web3auth
```

```
await walletServicesPlugin . showSwap ( ) ;
```

```
// Opens the Swap Modal
```

Connect dApps with Wallet Connect

Shows the Wallet Connect Scanner to connect with dApps having Wallet Connect login option. This is useful for interoperability with dApps having Wallet Connect login option.

Parameters

Name Description show Determines whether the Wallet Connect UI is displayed. This can be used to programmatically control its visibility.

Usage

Shows the Wallet Connect Scanner to connect with dApps having Wallet Connect login option. This is useful for interoperability with dApps having Wallet Connect login option.

```
import
```

```
{
```

```
WalletServicesPlugin
```

```
}
```

```
from
```

```
"@web3auth/wallet-services-plugin" ;
```

```
const walletServicesPlugin =
```

```
new
```

```
WalletServicesPlugin ( ) ; web3auth . addPlugin ( walletServicesPlugin ) ;
```

```
// Add the plugin to web3auth
```

```
await walletServicesPlugin . showWalletConnectScanner ( ) ;
```

Show Wallet Embedded UI

Shows the modal, and allows you to use the templated wallet UI services.

Parameters

Name Description show Determines whether the wallet UI is displayed. This can be used to programmatically control its visibility. path? Specifies the path for the wallet services.

Usage

```
import
```

```
{
```

```
WalletServicesPlugin
```

```
}
```

```
from
```

```
"@web3auth/wallet-services-plugin" ;
```

```
const walletServicesPlugin =
new
WalletServicesPlugin ( ) ; web3auth . addPlugin ( walletServicesPlugin ) ;
// Add the plugin to web3auth
await walletServicesPlugin . showWalletUi ( ) ;
// Opens the TopUp modal
```

Private Key export

Go to Privacy & Security under Settings in the Wallet UI to export your private key. Available from version 8.7.0 .

Toggle testnet visibility

You can toggle the testnet visibility in the Wallet Services UI by going to Settings , then General . Available from version 8.7.0 .

info If you don't pass the name and chainId for a specific chain as listed in the dropdown below, the chain will appear twice in the wallet services UI dropdown: once by default and once with the name you provided.

Transaction Confirmation Screens

You can use the wallet services proxy provider to integrate prebuilt transaction confirmation screens into your application. Upon successful completion, you can retrieve the signature for the request.

Please check the list of [JSON RPC methods](#) , noting that the request method currently supports only the signing methods.

Usage

```
import
{ WalletServicesPlugin }
from
"@web3auth/wallet-services-plugin" ;
const walletServicesPlugin =
new
WalletServicesPlugin ( ) ; // Use the existing Web3Auth instance web3auth . addPlugin ( walletServicesPlugin ) ;
const message =
"Example personal_sign message" ; const accounts =
await web3auth . provider . request < never ,
string [ ]
( { method :
"eth_accounts"
} ) ; const from = accounts [ 0 ] ; const signedMessage =
await walletServicesPlugin ? . proxyProvider ? . request < [ string ,
string ] ,
string
( { method :
"personal_sign" , params :
[ message , from ] , } )
```

[Edit this page](#) [Previous Overview](#) [Next React hooks](#)

