

React Hooks for Wallet Services Plugin for PnP Web SDK

Web3Auth provides essential React hooks for the Wallet Services Plugin SDK for managing wallet services such as WalletConnect, TopUp, and Wallet UI. These hooks can be directly imported from the [@web3auth/wallet-services-plugin-react-hooks](#) package.

Installation

`npm install --save @web3auth/wallet-services-plugin-react-hooks` For more information on React hooks, refer to the [official React documentation](#).

Initialization

To initialize the Wallet Services plugin, pass the `WalletServicesPlugin` instance to the list of plugins in `Web3AuthContextConfig`. For more details, refer to the [Wallet Services Plugin initialization documentation](#).

```
import
{ WalletServicesPlugin }
from
"@web3auth/wallet-services-plugin" ; import
{ Web3AuthContextConfig }
from
"@web3auth/modal-react-hooks" ;
const walletServicesPlugin =
new
WalletServicesPlugin ( ) ;
export
const web3AuthContextConfig : Web3AuthContextConfig =
{ web3AuthOptions , plugins :
[ walletServicesPlugin ] , } ;
```

Once `Web3AuthContextConfig` is created you can use `Web3AuthProvider` component to wrap the main component and inject the Web3Auth-related context. For more details, please refer to the [Web3AuthProvider documentation](#).

Available Hooks

To access the Wallet Services Plugin context, which is initialized via the `WalletServicesProvider` component, use the `useWalletServicesPlugin` function.

Hook Context Interface

- Table
- Interface

Parameter Description

| | |
|---------------------------------------|---|
| <code>isPluginConnected</code> | Indicates whether the plugin is connected and ready to be used. |
| <code>showWalletConnectScanner</code> | 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. |
| <code>showCheckout</code> | Shows the TopUp modal to select local currency and amount to top up the wallet. |
| <code>showWalletUI</code> | Shows the Wallet Services modal UI to be used as a wallet UI. |

```
export
interface
IBaseWalletServicesHookContext
{ isPluginConnected :
boolean ; showWalletConnectScanner ( showWalletConnectParams ? :
```

```
BaseEmbedControllerState [ "showWalletConnect" ] , ) :
```

```
Promise < void
```

```
    ; showCheckout ( showCheckoutParams ? :
```

```
BaseEmbedControllerState [ "showCheckout" ] ) :
```

```
Promise < void
```

```
    ; showWalletUI ( showWalletUiParams ? :
```

```
BaseEmbedControllerState [ "showWalletUi" ] ) :
```

```
Promise < void
```

```
    ; }
```

isPluginConnected[^](#)

Indicates whether the plugin is connected and ready to be used

Usage[^](#)

```
import
```

```
{ useWalletServicesPlugin }
```

```
from
```

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

```
const
```

```
{ isPluginConnected }
```

```
=
```

```
useWalletServicesPlugin ( ) ;
```

showWalletConnectScanner[^](#)

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[^](#)

```
import
```

```
{ useWalletServicesPlugin }
```

```
from
```

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

```
const
```

```
{ showWalletConnectScanner }
```

```
=
```

```
useWalletServicesPlugin ( ) ; await
```

```
showWalletConnectScanner ( ) ;
```

showCheckout

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

```
{ useWalletServicesPlugin }
```

from

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

const

```
{ showCheckout }
```

=

```
useWalletServicesPlugin ( ) ; await
```

```
showCheckout ( ) ;
```

showSwap

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
```

showWalletUI

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
{ useWalletServicesPlugin }
from
"@web3auth/wallet-services-plugin-react-hooks" ;
const
{ showWalletUI }
=
useWalletServicesPlugin ( ) ; await
showWalletUI ( ) ; Edit this page Previous Usage Next Overview
```