

# Wallet Services Plugin for PnP Web SDK

The [Wallet Services Plugin](#) enables your application to integrate essential features like wallet top-ups, WalletConnect, and customizable wallet and transaction UI screens. You gain full control to personalize the wallet interface to suit your brand, while also having the option to embed a seamless wallet experience directly for your users.

Note Access to Wallet Services is gated and available on the Scale plan and above. You can use this feature insapphire\_devnet for free.

Wallet Services is currently available for all EVM chains.

## Installation

```
npm install --save @web3auth/wallet-services-plugin
```

## Initialization

After creating your Web3Auth instance, you need to initialize the Wallet Services Plugin and add it to the Web3Auth instance for further usage.

### Parameters

- Table
- Interface

Parameter Description wsEmbedOpts? Configuration options for Wallet Services. It accepts Partial as a value.  
walletInitOptions? Parameters to customize your Wallet Services UI within the application. It accepts Partial as a value.  
constructor ( options ? :

```
{ wsEmbedOpts ? :
```

```
Partial < CtorArgs
```

```
    ; walletInitOptions ? :
```

```
Partial < WsEmbedParams
```

```
    ; } ) ;
```

### wsEmbedOpts

[â](#)

wsEmbedOpts is an optional parameter that allows you to pass in the configuration options for Wallet Services. It takes the CtorArgs object as input which contains the following parameters:

- Table
- Interface

CtorArgs

Parameter Description modalZIndex? Z-index of the modal and iframe. Default value is 99999 . web3AuthClientId Web3Auth Client ID instring . You can obtain this from the Web3Auth dashboard. web3AuthNetwork Web3Auth network to use. Accepts WEB3AUTH\_NETWORK\_TYPE . export

interface

CtorArgs

```
{ /* * Z-index of the modal and iframe * @defaultValue 99999 modalZIndex ? :
```

```
number ; /* * You can get your clientId/projectId by registering your * dapp on {@link "https://dashboard.web3auth.io/" developer dashboard} / web3AuthClientId :
```

```
string ; /* * network specifies the web3auth network to be used/ web3AuthNetwork :
```

```
WEB3AUTH_NETWORK_TYPE ; }
```

## walletInitOptions



The `walletInitOptions` from Wallet Services is an optional parameter that allows you to customize your Wallet Services UI within the application. It takes the object `WsEmbedParams` as input which contains multiple parameters that allow you to customize the UI of the plugin.

- Table
- Interface

### WsEmbedParams

Parameter Description `chainConfig?` Chain config of the Blockchain to connect with. It accepts `EthereumProviderConfig`. `whiteLabel?` White Label parameters to whitelisting the Wallet Services UI. It accepts `WhiteLabelParams`. export

interface

### WsEmbedParams

```
{ /* * Chain to connect with/ chainConfig ? :
```

```
EthereumProviderConfig ; /* * Build Environment of WsEmbed. * * production uses https://wallet.web3auth.io, * * staging uses https://staging-wallet.web3auth.io, * * testing uses https://develop-wallet.web3auth.io (latest internal build) * * development uses http://localhost:4050 (expects wallet-services-frontend to be run locally), * * @defaultValue production / buildEnv ? :
```

```
WS_EMBED_BUILD_ENV_TYPE ; /* * Enables or disables logging. * * Defaults to false in prod and true in other environments / enableLogging ? :
```

```
boolean ; /* * url of widget to load * * Defaults to true * @defaultValue true walletUrls ? :
```

```
Partial < Record < WS_EMBED_BUILD_ENV_TYPE ,
```

### WalletUrlConfig

```
/* * Allows you to customize the look & feel of the widget whiteLabel ? :
```

```
WhiteLabelParams ; }
```

## whiteLabel



The `whiteLabel` object provides additional options for customizing the appearance and behavior of Wallet Services UI.

- Table
- Interface

Parameter Description `showWidgetButton?` Show or hide the wallet widget button. Default is `true`. `buttonPosition?` Position of the widget button on the screen. Default is `bottom-left`. `hideNftDisplay?` Hide or show NFT display in the wallet. Default is `false`. `hideTokenDisplay?` Hide or show token display in the wallet. Default is `false`. `hideTransfers?` Hide or show the transfer option in the wallet. Default is `false`. `hideTopup?` Hide or show the top-up option in the wallet. Default is `false`. `hideReceive?` Hide or show the receive option in the wallet. Default is `false`. `defaultPortfolio?` Set the default portfolio type. Either `token` or `nft`. Default is `token`. export

interface

### WhiteLabelParams

```
{ /* * Show or hide the wallet widget button. * @defaultValue true showWidgetButton ? :
```

```
boolean ; /* * Position of the widget button. * @defaultValue bottom-left buttonPosition ? :
```

```
BUTTON_POSITION_TYPE ; /* * Hide or show NFT display in the wallet. * @defaultValue false hideNftDisplay ? :
```

```
boolean ; /* * Hide or show token display in the wallet. * @defaultValue false hideTokenDisplay ? :
```

```
boolean ; /* * Hide or show the transfer option in the wallet. * @defaultValue false hideTransfers ? :
```

```

boolean ; /* * Hide or show the top-up option in the wallet. * @defaultValue false/ hideTopup ? :
boolean ; /* * Hide or show the receive option in the wallet. * @defaultValue false/ hideReceive ? :
boolean ; /* * Set the default portfolio type. * @defaultValue token/ defaultPortfolio ? :
"token"

```

```
|
```

"nft" ; } note For more information on customizing the whitelabel options using theWhiteLabelData object, refer to the[whitelabel documentation](#) .

## Example

```

import
{ WalletServicesPlugin }
from
"@web3auth/wallet-services-plugin" ;
const walletServicesPlugin =
new
WalletServicesPlugin ( ) ; // Use the existing Web3Auth instance web3auth . addPlugin ( walletServicesPlugin ) Edit this page Previous Coinbase Wallet Next Usage

```