

Migration Guide from v7 to v8 for Web3Auth No Modal SDK

Overview[^](#)

This migration guide provides steps for upgrading from version 7 (v7) to version 8 (v8) of the Web3Auth No Modal SDK. The guide outlines significant changes and enhancements, including simplification of the whitelabeling process, introduction of `getDefaultExternalAdapters` for easy adapter management, and updates to the `CustomChainConfig` interface.

Changes in Detail[^](#)

Whitelabeling Configuration Update[^](#)

Before (v7):[^](#)

Whitelabel configuration was passed directly to the `OpenLoginAdapter`. This often required importing the `OpenLoginAdapter` just for this purpose.

```
// v7 code
const openloginAdapter =
  new
  OpenloginAdapter ( { adapterSettings :
    { whiteLabel :
      { appName :
        "W3A Heroes" , appUrl :
        "https://web3auth.io" , // ... other settings } , } , // ... other settings } ) ;
```

After (v8):[^](#)

The whitelabel configuration can now be passed to the `uiConfig` parameter, streamlining the process and reducing unnecessary imports.

```
// v8 code
const web3AuthOptions :
  Web3AuthNoModalOptions
  =
  { clientId , chainConfig , web3AuthNetwork :
    "sapphire_mainnet" , uiConfig :
      { appName :
        "W3A Heroes" , appUrl :
        "https://web3auth.io" , theme :
          { primary :
            "#768729" , } , logoLight :
            "https://web3auth.io/images/web3auth-logo.svg" , logoDark :
            "https://web3auth.io/images/web3auth-logo---Dark.svg" , defaultLanguage :
            "en" , mode :
            "auto" , useLogoLoader :
            true , } , privateKeyProvider , } ;
const web3auth =
  new
```

Web3AuthNoModal (web3AuthOptions) ;

Adapter Management with `getDefaultExternalAdapters`

[^](#)

Before (v7): [^](#)

Adapters were manually configured and instantiated.

After (v8): [^](#)

Adapters can be automatically fetched and configured using the `getDefaultExternalAdapters` method, streamlining the setup process.

```
// v8 code import
{ getDefaultExternalAdapters }
from
"@web3auth/default-evm-adapter" ;

const adapters =
await
getDefaultExternalAdapters ( { options : web3AuthOptions } ) ;
adapters . forEach ( ( adapter )
=>
{ if
( adapter . name
===
"walletconnectv2" )
{ web3auth . configureAdapter ( adapter ) ; } } ) ;
```

ChainConfig

[^](#)

Before (v7): [^](#)

ChainConfig did not require `alogo` parameter, and `blockExplorer` was used for the block explorer URL.

```
const chainConfig :
CustomChainConfig
=
{ chainNamespace :
CHAIN_NAMESPACES . OTHER , chainId :
"0x1" , rpcTarget :
"https://mainnet-algorand.api.purestake.io/ps2" , displayName :
"Algorand Mainnet" , blockExplorer :
"" , ticker :
"ALGO" , tickerName :
"Algorand" , } ;
```

After (v8):[â](#)

ChainConfig now requires a logo parameter for the chain's logo and the parameter formerly called blockExplorer has been renamed to blockExplorerUrl. Additionally, isTestnet has been introduced which can be used to define whether the network is testnet or not.

note From v8, parameters apart from chainId, rpcTarget, and chainNamespace are now optional. Please note, while using [Wallet Services](#) you have to also pass the optional parameters. // v8 code export

type

CustomChainConfig

=

{ chainNamespace :

ChainNamespaceType ; chainId :

string ; rpcTarget :

string ; wsTarget ? :

string ; displayName ? :

string ; blockExplorerUrl ? :

string ; ticker ? :

string ; tickerName ? :

string ; decimals ? :

number ; logo ? :

string ; isTestnet ? :

boolean ; } ; const chainConfig =

{ chainId :

"0x3" , displayName :

"Solana Testnet" , chainNamespace :

CHAIN_NAMESPACES . SOLANA , tickerName :

"SOLANA" , ticker :

"SOL" , decimals :

18 , rpcTarget :

"https://api.testnet.solana.com" , blockExplorerUrl :

"https://explorer.solana.com/?cluster=testnet" , logo :

"https://images.toruswallet.io/sol.svg" , isTestnet :

true , } ;

web3AuthNoModalOptions

[â](#)

Before (v7):[â](#)

Earlier, the privateKeyProvider provider was not mandatory, and chainConfig was mandatory.

After (v8):[â](#)

In v8, the `privateKeyProvider` is introduced to `Web3AuthNoModalOptions` , and `chainConfig` is now optional.

```
const chainConfig =
{ chainId :
"0x1" ,
// Please use 0x1 for Mainnet rpcTarget :
"https://rpc.ankr.com/eth" , displayName :
"Ethereum Mainnet" , blockExplorerUrl :
"https://etherscan.io/" , ticker :
"ETH" , tickerName :
"Ethereum" , logo :
"https://images.toruswallet.io/eth.svg" , } ;
const ethereumPrivateKeyProvider =
EthereumPrivateKeyProvider ( { config :
{ chainConfig } } ) ;
const web3auth =
new
Web3Auth ( { clientId :
"" ,
// Get your Client ID from the Web3Auth Dashboard web3AuthNetwork :
WEB3AUTH_NETWORK . SAPPHIRE_MAINNET , uiConfig :
{ // Access the LANGUAGES from @toruslabs/openlogin-utils defaultLanguage :
LANGUAGES . tr ,
// Supported languages en, de, ja, ko, zh, es, fr, pt, nl } privateKeyProvider : ethereumPrivateKeyProvider , } ) ;
```

@web3auth/default-solana-adapter

[â](#)

Default Solana adapter allows you to create & configure external adapters for SOLANA namespace, like `@web3auth/torus-solana` , and `@web3auth/phantom` easily.

Installation[â](#)

- npm
- Yarn
- pnpm

```
npm install --save @web3auth/default-solana-adapter yarn add @web3auth/default-solana-adapter pnpm add
@web3auth/default-solana-adapter
```

Usage[â](#)

```
const chainConfig =
{ chainId :
"0x3" , displayName :
"Solana Testnet" , chainNamespace :
```

```

CHAIN_NAMESPACES . SOLANA , tickerName :
"SOLANA" , ticker :
"SOL" , decimals :
18 , rpcTarget :
"https://api.testnet.solana.com" , blockExplorerUrl :
"https://explorer.solana.com/?cluster=testnet" , logo :
"https://images.toruswallet.io/sol.svg" , isTestnet :
true , } ;

const privateKeyProvider =
new
SolanaPrivateKeyProvider ( { config :
{ chainConfig }
} ) ;

const web3AuthNoModalOptions :
Web3AuthNoModalOptions
=
{ clientId , web3AuthNetwork :
WEB3AUTH_NETWORK . SAPPHIRE_MAINNET , privateKeyProvider : privateKeyProvider , } ;

const adapters =
await
getDefaultExternalAdapters ( { options : web3AuthNoModalOptions } ) ;
adapters . forEach ( ( adapter )
=>
{ web3auth . configureAdapter ( adapter ) ; } ) ;

```

Summary^â

Migrating from v7 to v8 of the Web3Auth No Modal SDK involves:

- Moving whitelabel configuration to theuiConfig parameter.
- UtilizinggetDefaultExternalAdapters for streamlined adapter configuration.
- UpdatingChainConfig to include thelogo parameter and renamingblockExplorer toblockExplorerUrl
- . [Edit this page](#) [Previous v8 to v9](#) [Next v6 to v7](#)