

Adapters for Web3Auth PnP Web SDKs

Adapters are essentially connectors between Web3Auth and the underlying wallet provider. For example, an adapter for connecting with torus wallet is available under web3auth as [@web3auth/torus-evm-adapter](#). Every adapter follows a common interface which is required by Web3Auth to communicate with the wallet.

An adapter emits certain events like `CONNECTED`, `CONNECTING` and `DISCONNECTED` etc during login lifecycle of a user. Each adapter exposes provider on successful user login that can be used to invoke RPC calls on wallet and on connected blockchain.

Hence, adapters are needed to connect your Web3Auth instance to a wallet provider. Web3Auth by default contains a pre-configured list of login adapters. It also provides flexibility to add/remove adapters or to use some existing adapter with custom configurations. To configure an adapter create an instance of the adapter by using its corresponding package and pass the returned adapter instance in the `configureAdapter` function.

For example, to configure openlogin adapter,

- Import the `OpenloginAdapter`
- from the `@web3auth/openlogin-adapter` package.
- Create an instance of the adapter along with the configuration
- Pass the returned instance to `web3auth.configureAdapter`
- as shown in the example below.

```
import
{
  OpenloginAdapter
}
from
"@web3auth/openlogin-adapter" ;

const openloginAdapter =
new
OpenloginAdapter ( { adapterSettings :
{ network :
"sapphire_mainnet" , uxMode :
"popup" , } , } ) ;

web3auth . configureAdapter ( openloginAdapter ) ;
```

Currently Available Wallet Adapters

By default Web3Auth's modal UI supports a set of default adapters depending on the `authMode` being used. By default, Web3Auth requires basic configuration `client_id`, and registering a redirect if you are using CustomAuth adapter from the dashboard.

The table below summarizes all the available adapters and their properties

Adapter	AuthMode	ChainNamespace	Default	Package Name	openlogin	WALLET	DAPP	EIP155	SOLANA	YES
@web3auth/openlogin-adapter	torus-evm	DAPP	EIP155	NO	@web3auth/torus-evm-adapter	torus-solana	DAPP	SOLANA	NO	@web3auth/torus-solana-adapter
@web3auth/metamask-adapter	phantom	DAPP	SOLANA	NO	@web3auth/phantom-adapter	walletconnect-v1	DAPP	EIP155	NO	@web3auth/wallet-connect-v1-adapter
@web3auth/coinbase	slope	DAPP	SOLANA	NO	@web3auth/slope	solflare	DAPP	SOLANA	NO	@web3auth/solflare
@web3auth/default-evm-adapter	default-solana-adapter	DAPP	SOLANA	NO	@web3auth/default-solana-adapter					

Types of Wallet Adapters

Currently Web3Auth supports two categories of wallet adapters:

- IN_APP
- Application Wallets
- EXTERNAL
- External Wallets

Application Wallets

- Application wallets are those wallets which are scoped to dApp. This means that if you are logging in with the same application wallet on different applications, you will get different wallets (private keys).
- openlogin
- ,torus-evm
- andtorus-solana
- are Application Wallets.

External Wallets

- External wallet are those which are global for all applications. No matter where you are logging in, you will get same keys if same credentials are being provided.
- metamask
- ,phantom
- andwalletconnect-v1
- ,coinbase
- ,slope
- ,solflare
- are External Wallets in Web3Auth.

Common Configuration Interfaces

type

CustomChainConfig

=

{ chainNamespace :

ChainNamespaceType ; /* * The chain id of the chain/ chainId :

string ; /* * RPC target Url for the chain/ rpcTarget :

string ; /* * Display Name for the chain/ displayName :

string ; /* * Url of the block explorer/ blockExplorerUrl :

string ; /* * Default currency ticker of the network (e.g: ETH) ticker :

string ; /* * Name for currency ticker (e.g:Ethereum) / tickerName :

string ;

/* * Logo of the chain/ logo : string ; } ; warning If you do passchainConfig in an Adapter, it overwrites thechainConfig passed over to theWeb3Auth /Web3AuthNoModal constructor. [Edit this page](#) [Previous](#) [Show Wallet Connect Scanner](#) [Next](#) [Openlogin \(Social Logins\)](#)