

Web3Auth PnP Web SDKs

In the Plug and Play product, we offer a modular package of Web3Auth which helps you connect the Social Logins (using Openlogin Adapter) alongside external wallet via a modular adapter interface. This helps you make Web3Auth, a one stop solution for all your authentication needs and get started quickly with your user onboarding journey. An adapter is just an extension/ connector of the underlying wallet to Web3Auth. This modular approach helps the package remain lightweight and easy to use, while providing you the flexibility of using multiple underlying packages in an efficient way.

For using Web3Auth in the web, you have two choices of SDKs to get started with.

- Web3Auth Plug and Play Modal SDK [@web3auth/modal](#)
- :
- A simple and easy to use SDK that will give you a simple modular way
- of implementing Web3Auth directly within your application. You can use the pre-configured Web3Auth Modal UI and whitelabel it according to your
- needs.
- Web3Auth Plug and Play No Modal SDK [@web3auth/no-modal](#)
- :
- The core module implementing all the Web3Auth features you
- need and giving you the flexibility of using your own UI with the Web3Auth SDK working in the backend.

Web3Auth Plug and Play Modal SDK [@web3auth/modal](#)

[@web3auth/modal](#)

[@web3auth/modal](#)

This package provides main class for using default Web3Auth Modal. The package includes all of our packages and gives you a simple way of implementing Web3Auth within your interface. Additionally, it is a child class of [@web3auth/no-modal](#) package. Hence, you can still call all the functions available in the [@web3auth/no-modal](#) package.

tip Head on to the [@web3auth/modal SDK Reference](#) section to get started.

Web3Auth Plug and Play No Modal SDK [@web3auth/no-modal](#)

[@web3auth/no-modal](#)

[@web3auth/no-modal](#)

This package provides the core logic for handling adapters within Web3Auth. This package acts as a manager for all the wallet adapters we offer. You should use this package to build your custom login UI on top of Web3Auth. The SDK size for this is comparatively smaller than the Plug and Play Modal SDK, since it doesn't have the Web3Auth Modal UI within it.

tip Head on to the [@web3auth/no-modal SDK Reference](#) to get started.

Additional SDKs [@web3auth/](#)

Adapter packages [@web3auth/](#)

Adapter acts as a connector between the Web3Auth and an underlying wallet provider. Web3Auth uses the default [@web3auth/openlogin-adapter](#) for social logins. For the rest of the external wallets, every adapter follows a common interface which is required by Web3Auth to communicate with the wallet.

tip Checkout the [Adapters SDK Reference](#) to learn more.

Currently we have the following adapters available for utilisation:

- [@web3auth/openlogin-adapter](#)
- - default
- (Enables social logins)
- [@web3auth/torus-evm-adapter](#)
- [@web3auth/torus-solana-adapter](#)
- [@web3auth/metamask-adapter](#)
- [@web3auth/phantom-adapter](#)
- [@web3auth/wallet-connect-v1-adapter](#)
- [@web3auth/coinbase-adapter](#)

- [@web3auth/slope-adapter](#)
- [@web3auth/solflare-adapter](#)

wagmi Connector[^](#)

wagmi is a collection of React Hooks containing everything you need to start working with Ethereum. Web3Auth can be used with wagmi library to ease the integration process for a dApp using wagmi hooks.

Simply use [@web3auth/web3auth-wagmi-connector](#) which connects with both Modal and No Modal SDKs in your dApp to manage the interactions seamlessly.

tip Head on to the [@web3auth/web3auth-wagmi-connector](#) to get started.

Helper SDKs[^](#)

Wallet Services[^](#)

Web3Auth Wallet Services offer modular, pluggable enhancements for your Web3Auth-integrated wallet, designed to elevate both functionality and user experience within your application. These services facilitate seamless integration, providing a comprehensive set of features without necessitating additional UI development. From sophisticated wallet functionality to user engagement tools and efficient transaction management, Web3Auth Wallet Services are tailored to meet the needs of modern day dApp experiences.

tip Head on to the [@web3auth/wallet-services-plugin](#) to get started.

Provider packages[^](#)

For making RPC calls within your dApp, Web3Auth exposes respective providers for different chains. This provider can be used to interact with the connected chain using exposed functions within the provider. Currently Web3Auth supports providers for both EVM and Solana chains. For other chains, one can easily get the private key from the Web3Auth SDK.

tip Checkout the [Providers SDK Reference](#) to learn more.

- [For EVM based Chains@web3auth/ethereum-provider](#)
- [For Solana Blockchain@web3auth/solana-provider](#)
- [Common Key Provider for other Blockchains@web3auth/base-provider](#)

Common Types and Interfaces[^](#)

[@web3auth/base](#)

[^](#)

This package gives access to common types and interfaces for Web3Auth. This comes in handy by providing you a standard way of importing the values you need to work with the SDKs. We highly recommend using it while working with Typescript. [Edit this page](#) [Previous v1 to v2](#) [Next Overview](#)