How to use a Web3Auth signer with permissionless.js

Web3Auth is a popular embedded wallet provider that supports social logins. While social logins are great, your users still need to onramp in order to pay for gas, which introduces significant friction.

By combining permissionless.js with Web3Auth, you can use Web3Auth to enable a smooth social login experience, while using permissionless.js accounts as the smart wallets to sponsor gas for users, batch transactions, and more.

Setup

To use Web3Auth with permissionless.js, first create an application that integrates with Web3Auth.

- Refer to theWeb3Auth documentation site
- for instructions on setting up an application with the Web3Auth.
- For a quick start, Web3Auth provides example starter projects, available here
- .

Integration

Integrating permissionless.js with Web3Auth is straightforward after setting up the project. Web3Auth provides an Externally Owned Account (EOA) wallet to use as a signer with permissionless.js accounts.

Create the Web3Auth object

After following the Web3Auth documentation, you will have access to aweb3auth object as shown below that you can use to create aSmartAccountSigner object:

```

import{ CHAIN_NAMESPACES, WEB3AUTH_NETWORK }from"@web3auth/base" import{ Web3Auth }from"@web3auth/modal" import{ providerToSmartAccountSigner }from"permissionless" import{ EIP1193Provider }from"viem"

// Config options here will be specific to your project. See the Web3Auth docs for more info.
constweb3auth=newWeb3Auth({ clientId, chainConfig: { chainNamespace:CHAIN_NAMESPACES.EIP155, chainId:"0xaa36a7", rpcTarget:"https://rpc.ankr.com/eth_sepolia", displayName:"Sepolia Testnet", blockExplorer:"https://sepolia.etherscan.io", ticker:"ETH", tickerName:"Ethereum", }, uiConfig: {}, web3AuthNetwork:WEB3AUTH_NETWORK.SAPPHIRE_DEVNET, })

// Get the Provider and EOA address (this will be the address of the signer) from Web3Auth
constweb3authProvider=web3auth.provider

if(!web3authProvider) { thrownewError("No provider found") }

// Create the smart account signer from the provider and signer address
constsmartAccountSigner=awaitproviderToSmartAccountSigner(web3authProviderasEIP1193Provider)

```

Use with permissionless.js

SimpleAccount Safe Account Kernel Account Biconomy Account ```

SimpleAccount import{ signerToSimpleSmartAccount }from"permissionless/accounts" import{ createPublicClient, http }from"viem" import{ generatePrivateKey, privateKeyToAccount }from"viem/accounts" import{ sepolia }from"viem/chains"

exportconstpublicClient=createPublicClient({ transport:http("https://rpc.ankr.com/eth_sepolia"), chain: sepolia, })

constsmartAccount=awaitsignerToSimpleSmartAccount(publicClient, { signer: smartAccountSigner, factoryAddress:"0x9406Cc6185a346906296840746125a0E44976454", entryPoint:ENTRYPOINT_ADDRESS_V06, })

```
```