EIP1193 (EVM) Private Key Provider for PnP Web SDKs

@web3auth/ethereum-provider

â

TheEIP1193 Provider can be used to interact with any EVM compatible blockchain. This is a wrapper around theEthereum JavaScript Provider API with some additional functionalities around Web3Auth Private Key handling.

In this section we'll explore more about how you can use this provider with our SDKs.

Installationâ

@web3auth/ethereum-provider

â

npm install --save @web3auth/ethereum-provider

Initialisationâ

ImportEthereumPrivateKeyProvider from@web3auth/ethereum-provider .

import

{

EthereumPrivateKeyProvider

}

from

"@web3auth/ethereum-provider" ;

const privateKeyProvider =

new

EthereumPrivateKeyProvider ({ config :

EthereumPrivKeyProviderConfig , }) ; This constructor takes an object with a config ofEthereumPrivKeyProviderConfig as input.

Argumentsâ

EthereumPrivKeyProviderConfig

export

interface

EthereumPrivKeyProviderConfig

extends

BaseProviderConfig

{ chainConfig :

CustomChainConfig ; }

export

type

CustomChainConfig

=

{ chainNamespace :

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

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

string ; /* * *web socket target Url for the chain*/ wsTarget ? :

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 ; /* * *Number of decimals for the currency ticker (e.g: 18)*/ decimals ? :

number ; /* * *Logo for the token*/ logo ? :

string ; /* * *Whether the network is testnet or not*/ isTestnet ? :

boolean ; } ; export

interface

BaseProviderConfig

extends

BaseConfig

{ chainConfig :

Partial < CustomChainConfig

; networks ? :

Record < string ,

CustomChainConfig

; skipLookupNetwork ? :

boolean ; } export

interface

BaseConfig

{ /* * *Determines if this controller is enabled* / disabled ? :

boolean ; }

Chain Configâ

While connecting your preferred chain, you need to pass the chainConfig as a parameter. The Chain IDs for the supported chains can be found onChainList . Please note that you need to pass over the hex value of the chain id in the provider config.

Some of the commonly used L2s and the Ethereum chain ids are listed below.

Hex Decimal Network 0x1 1 Ethereum Mainnet 0xAA36A7 11155111 Sepolia Testnet 0x38 56 Binance Smart Chain Mainnet 0x89 137 Polygon Mainnet 0xA86A 43114 Avalanche C-Chain 0xA 10 Optimism 0xE 14 Flare 0x13 19 Songbird const chainConfig =

{ chainNamespace :

CHAIN_NAMESPACES . EIP155 , chainId :

"0x1" , rpcTarget :

"https://rpc.ankr.com/eth" , // Avoid using public rpcTarget in production. // Use services like Infura, Quicknode etc displayName :

"Ethereum Mainnet" , blockExplorerUrl :

"https://etherscan.io" , ticker :

"ETH" , tickerName :

"Ethereum" , logo :

"https://cryptologos.cc/logos/ethereum-eth-logo.png" , } ;

Setting up the providerâ

- PnP Modal SDK
- PnP NoModal SDK
- CoreKit SFA Web SDK
- MPC CoreKit JS SDK

import

{

Web3Auth

}

from

"@web3auth/modal" ; import

{

EthereumPrivateKeyProvider

}

from

"@web3auth/ethereum-provider" ; import

{

WEB3AUTH_NETWORK

}

from

"@web3auth/base" ;

const privateKeyProvider =

new

EthereumPrivateKeyProvider ({ config :

{ chainConfig : chainConfig } , }) ;

const web3auth =

new

Web3Auth ({ // Get it from Web3Auth Dashboard clientId , web3AuthNetwork :

WEB3AUTH_NETWORK . SAPPHIRE_MAINNET , privateKeyProvider , }) ; import

{

Web3AuthNoModal

}

```
from
"@web3auth/no-modal" ; import
{
AuthAdapter
}
from
"@web3auth/auth-adapter" ; import
{
EthereumPrivateKeyProvider
}
from
"@web3auth/ethereum-provider" ; import
{
WEB3AUTH_NETWORK
}
from
"@web3auth/base" ;
const privateKeyProvider =
new
EthereumPrivateKeyProvider ( { config :
{ chainConfig } , } ) ;
const web3auth =
new
Web3AuthNoModal ( { clientId ,
// Get it from Web3Auth Dashboard web3AuthNetwork :
WEB3AUTH_NETWORK . SAPPHIRE_MAINNET , privateKeyProvider , } ) ;
const authAdapter =
new
AuthAdapter ( { privateKeyProvider } ) ; web3auth . configureAdapter ( authAdapter ) ; import
{
Web3Auth
}
from
"@web3auth/single-factor-auth" ; import
{
EthereumPrivateKeyProvider
}
```

```
from
"@web3auth/ethereum-provider" ;
const privateKeyProvider =
new
EthereumPrivateKeyProvider ( { config :
{ chainConfig } , } ) ;
const web3auth =
new
Web3Auth ( { clientId ,
// Get your Client ID from Web3Auth Dashboard web3AuthNetwork :
"sapphire_mainnet" , privateKeyProvider , } ) ; import
{
Web3AuthMPCCoreKit ,
WEB3AUTH_NETWORK , makeEthereumSigner }
from
"@web3auth/mpc-core-kit" ; import
{
EthereumSigningProvider
}
from
"@web3auth/ethereum-mpc-provider" ; import
{
CHAIN_NAMESPACES
}
from
"@web3auth/base" ; import
{ tssLib }
from
"@toruslabs/tss-dkls-lib" ;
const coreKitInstance =
new
Web3AuthMPCCoreKit ( { web3AuthClientId , web3AuthNetwork :
WEB3AUTH_NETWORK . MAINNET , storage :
window . localStorage , manualSync :
true ,
// This is the recommended approach tssLib : tssLib , } ) ;
// Setup provider for EVM Chain const evmProvider =
```

new

EthereumSigningProvider ({ config :

{ chainConfig }

}) ; evmProvider . setupProvider (makeEthereumSigner (coreKitInstance)) ;

Using the providerâ

On connection, you can use thisprovider as anEIP1193 provider withweb3.js ,ethers orviem library.

- web3.js
- ethers.js
- viem

import

Web3

from

"web3" ;

const web3 =

new

Web3 (provider) ; import

{ ethers }

from

"ethers" ;

const provider =

new

ethers . providers . Web3Provider (provider) ; import

{ createWalletClient }

from

"viem" ;

const walletClient =

createWalletClient ({ transport :

custom (provider) , }) ; Once you have set up the provider, you can use the standard functions in the web3 library to get user's account, perform transactions, sign a message etc. Here we have listed a few examples to help you get started.

info ** Please refer to all the updated JSON RPC Methods with the Provider on theOfficial Ethereum Documentation ** tip Please refer to ourEVM Connect Blockchain Reference for more information.

Examplesâ

Integrate PnP Modal SDK with EVM based chains SAMPLE APP Use any EVM Blockchain like Ethereum, Polygon, Arbitrum, Base etc. with Plug and Play Modal SDK Source Code Guide pnp web @web3auth/modal javascript evm ### Integrate PnP No Modal SDK with EVM based chains SAMPLE APP Use any EVM Blockchain like Ethereum, Polygon, Arbitrum, Base etc. with Plug and Play No Modal SDK Source Code Guide pnp web @web3auth/no-modal javascript evm
Edit this page Previous Overview Next Account Abstraction Provider