

PnP No Modal SDK - v5 to v6

General[a](#)

web3auth.connected

is introduced[a](#)

Manage session using web3auth.connected

instead of web3auth.provider [a](#)

With V6, users can manage their session using web3auth.connected instead of web3auth.provider .

// With V6 if

```
( web3auth . connected )
```

```
{ setLoggedIn ( true ) ; }
```

provider

is now always available[a](#)

In V5, we used to add a check for setting the provider only if the web3auth.provider was present. But now with V6 we always have a provider available even if the user is not logged in.

// With V5 if

```
( web3auth . provider )
```

```
{ setProvider ( web3auth . provider ) ; } // With V6 setProvider ( web3auth . provider ) ;
```

// before the connect() or connectTo(), provider is available.

rpcTarget

and chainId is now a mandatory parameter[a](#)

Previously, the Web Modal SDK required chainConfig as a parameter which had rpcTarget & chainId as the optional parameter. But with V6, it's mandatory to add rpcTarget & chainId in the chainConfig object.

```
const web3auth =
```

```
new
```

```
Web3Auth ( { clientId , chainConfig :
```

```
{ chainNamespace :
```

```
CHAIN_NAMESPACES . EIP155 , chainId :
```

```
"0x1" , rpcTarget :
```

```
"https://rpc.ankr.com/eth" ,
```

```
// This is the public RPC we have added, please pass on your own custom endpoint while creating an app } ,  
web3AuthNetwork :
```

```
"sapphire_mainnet" , } ) ;
```

privateKeyProvider

is introduced[a](#)

privateKeyProvider

is now a mandatory parameter to be passed into the OpenLoginAdapter[a](#)

With V6, we have added a new parameter `privateKeyProvider` which is mandatory to be passed into the `OpenLoginAdapter` .

For EVM chains

For EVM chains use the `EthereumPrivateKeyProvider` from the `@web3auth/ethereum-provider` package. Please note that `EthereumPrivateKeyProvider` requires `chainConfig` as the `config` parameter at the time of initialization.

```
import
{
  EthereumPrivateKeyProvider
}
from
"@web3auth/ethereum-provider" ;
const chainConfig =
{ chainNamespace :
  CHAIN_NAMESPACES . EIP155 , chainId :
  "0x1" , rpcTarget :
  "https://rpc.ankr.com/eth" , displayName :
  "Ethereum Mainnet" , blockExplorer :
  "https://etherscan.io" , ticker :
  "ETH" , tickerName :
  "Ethereum" , } ; const privateKeyProvider =
new
EthereumPrivateKeyProvider ( { config :
{ chainConfig }
} ) ; const openloginAdapter =
new
OpenloginAdapter ( { privateKeyProvider , } ) ; web3authInstance . configureAdapter ( openloginAdapter ) ;
```

For Solana

For Solana use the `SolanaPrivateKeyProvider` from the `@web3auth/solana-provider` package. Please note that `SolanaPrivateKeyProvider` requires `chainConfig` as the `config` parameter at the time of initialization.

```
import
{
  SolanaPrivateKeyProvider
}
from
"@web3auth/solana-provider" ;
const chainConfig =
{ chainNamespace :
  CHAIN_NAMESPACES . SOLANA , chainId :
  "0x1" ,
```

```
// Please use 0x1 for Mainnet, 0x2 for Testnet, 0x3 for Devnet rpcTarget :
"https://rpc.ankr.com/solana" , displayName :
"Solana Mainnet" , blockExplorer :
"https://explorer.solana.com" , ticker :
"SOL" , tickerName :
"Solana" , } ; const privateKeyProvider =
new
SolanaPrivateKeyProvider ( { config :
{ chainConfig }
} ) ; const openloginAdapter =
new
OpenloginAdapter ( { privateKeyProvider , } ) ; web3auth . configureAdapter ( openloginAdapter ) ;
```

For all other chains

For all Non-EVM & Non-Solana chains use theCommonPrivateKeyProvider from the@web3auth/base-provider package. Please note thatCommonPrivateKeyProvider requireschainConfig as theconfig parameter at the time of initialization.

```
import
{
CommonPrivateKeyProvider
}
from
"@web3auth/base-provider" ;
const chainConfig =
{ chainNamespace :
CHAIN_NAMESPACES . SOLANA , chainId :
"0x1" ,
// Please use 0x1 for Mainnet, 0x2 for Testnet, 0x3 for Devnet rpcTarget :
"https://rpc.ankr.com/solana" , displayName :
"Solana Mainnet" , blockExplorer :
"https://explorer.solana.com" , ticker :
"SOL" , tickerName :
"Solana" , } ; const privateKeyProvider =
new
CommonPrivateKeyProvider ( { config :
{ chainConfig }
} ) ;
const openloginAdapter =
new
OpenloginAdapter ( { privateKeyProvider , } ) ; web3authInstance . configureAdapter ( openloginAdapter ) Edit this page
```

[Previous v6 to v7](#)