

Torus Solana Adapter

[@web3auth/torus-solana-adapter](#)

Torus solana adapter allows to connect with torus wallet for solana blockchain. You can read more about torus wallet [here](#).

Basic Details

Adapter Name:torus-solana

[^](#)

Package Name:[@web3auth/torus-solana-adapter](#)

[^](#)

authMode:DAPP

[^](#)

chainNamespace:SOLANA

[^](#)

Default:YES

[^](#)

Installation

- npm
- Yarn
- pnpm

npm install --save @web3auth/torus-solana-adapter yarn add @web3auth/torus-solana-adapter pnpm add @web3auth/torus-solana-adapter

Arguments

- Table
- Interface

Parameter type chainConfig CustomChainConfig adapterSettings [TorusCtorArgs](#) loginSettings [LoginParams](#) initParams [Omit](#) clientId? string sessionTime? number web3AuthNetwork? OPENLOGIN_NETWORK_TYPE // refer to <https://docs.tor.us/solana-wallet/api-reference/class> to know more about interface field types interface

SolanaWalletOptions

{ adapterSettings ? :

TorusCtorArgs ; loginSettings ? :

LoginParams ; initParams ? :

Omit < TorusParams ,

"network"

; chainConfig ? :

CustomChainConfig ; clientId ? :

string ; sessionTime ? :

number ; web3AuthNetwork ? :

```
OPENLOGIN_NETWORK_TYPE ; }
```

Custom Chain Config

chainConfig

warning While you can pass yourchainConfig , it is not required since you can directly pass it over to theWeb3Auth /Web3AuthNoModal configuration while instantiating it.

Read more about it in their respective sections:

- [web3auth/modal](#)
- [web3auth/no-modal](#) warning If you do passchainConfig in an Adapter, it overwrites thechainConfig passed over to theWeb3Auth /Web3AuthNoModal constructor.

Change Adapter Settings

You can change the adapter settings by calling thesetAdapterSettings() function on the adapter instance.

Arguments

- Table
- Interface

Parameter type clientId? string sessionTime? number chainConfig? CustomChainConfig web3AuthNetwork? OPENLOGIN_NETWORK_TYPE interface

BaseAdapterSettings

{ clientId ? :

string ; sessionTime ? :

number ; chainConfig ? :

CustomChainConfig ; web3AuthNetwork ? :

OPENLOGIN_NETWORK_TYPE ; }

Example

```
import
```

```
{
```

```
SolanaWalletAdapter
```

```
}
```

```
from
```

```
"@web3auth/torus-solana-adapter" ; const solanaWalletAdapter =
```

```
new
```

```
SolanaWalletAdapter ( { adapterSettings :
```

```
{ modalZIndex :
```

```
99999 , } , loginSettings :
```

```
{ loginProvider :
```

```
"google" , } , initParams :
```

```
{ buildEnv :
```

```
"testing" , } , chainConfig :
```

```

{ chainNamespace :
CHAIN_NAMESPACES . SOLANA , rpcTarget :
"https://api.mainnet-beta.solana.com" , blockExplorer :
"https://explorer.solana.com" , chainId :
"0x1" ,
// Please use 0x1 for Mainnet, 0x2 for Testnet, 0x3 for Devnet displayName :
"Solana Mainnet Beta" , ticker :
"sol" , tickerName :
"solana" , } , clientId :
"BPi5PB_UiIZ-cPz1GtV5i1I2iOSOHuimiXBI0e-Oe_u6X3oVAbCiAZOTEBtTXw4tsluTITPqA8zMsfxIKMjiqNQ" , sessionTime :
3600 ,
// 1 hour in seconds web3AuthNetwork :
"sapphire_mainnet" , } ) ;
// it will add/update the torus-solana adapter in to web3auth class web3auth . configureAdapter ( solanaWalletAdapter ) ;
// you can also change the adapter settings by calling the setAdapterSettings() function on the adapter instance
solanaWalletAdapter . setAdapterSettings ( { sessionTime :
86400 ,
// 1 day in seconds chainConfig :
{ chainNamespace :
CHAIN_NAMESPACES . SOLANA , chainId :
"0x1" , rpcTarget :
"https://rpc.ankr.com/solana" ,
// This is the public RPC we have added, please pass on your own endpoint while creating an app } , web3AuthNetwork :
"sapphire_mainnet" , } ) ;Edit this page Previous Torus EVM Wallet Next Metamask Wallet

```