Set up guide

For more information on web wallet integrations, refer to the StarknetKit documentation

Installation

To get started with integrating the web wallet SDK in your dApp, you will need to install the `starknetkit` package and its peer dependencies:

```

Copy yarnaddstarknetkit

```

or for usage with NPM:

```

Copy npminstallstarknetkit

```

Imports

After installation, we get access to different methods, such asconnect ,disconnect , etc which we should import for use in our application:

```

Copy import{ connect,disconnect }from'starknetkit'

```

Establishing a connection

To establish a wallet connection, we need to call theconnect method which was imported earlier like this:

```

Copy constconnection=awaitconnect();

```

Below is an example function that establishes a connection, then sets theconnection ,provider , andaddress states:

```

Copy constconnectWallet=async()=>{ constconnection=awaitconnect({webWalletUrl:"https://web.argent.xyz"});

if(connection&&connection.isConnected) { setConnection(connection) setProvider(connection.account) setAddress(connection.selectedAddress) } }

```

And to reconnect to a previously connected wallet on load:

```

Copy constconnection=awaitconnect({modalMode:"neverAsk",webWalletUrl:"https://web.argent.xyz"})

```

It's important to note that web wallet is only available for users on mainnet. If as a dApp for integration and testing purposes, you need access to an internal testnet environment, please contact Argent privately. For a detailed guide on how to integrate web wallet in starknet-react based applications, please refer to theStarknetKit documentation .

Signing transactions

Signing transactions in web wallet is similar to how it's done using the Argent X browser extension.

```
```

Copy consttx=awaitconnection.account.execute({ //let's assume this is an erc20 contract contractAddress:"0x...", selector:"transfer", calldata:["0x...", // ...] })

```

It will show up like this to the user:

?

If the user's wallet is already funded it will ask the user to confirm the transaction. The dapp will get feedback if the user has confirmed or rejected the transaction request. If confirmed, the dapp will get a transaction hash.

Users wallet is not funded

In a case where the user has no funds, the user is guided through the “Add funds” screens where they can go to on-ramps and more. We tried to make the funding process of new wallets as easy as possible with regards KYC. Once this process is completed, the user's wallet will be funded and now ready to be deployed.

Users wallet is not deployed

After a user has funded their wallet, they are ready for their first transaction. The wallet deployment will be done along the first transaction and is almost invisible to the user. Just note that a connected wallet may not be deployed yet.

Previous Web Wallet SDK Next Web Wallet UX

Last updated1 month ago
```