

StarknetKit SDK

A beginner's guide to the StarknetKit SDK

Installation

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

```
...
```

```
Copy npm install starknetkit
```

```
...
```

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 the connect method which was imported earlier like this:

```
...
```

```
Copy const connection = await connect();
```

```
...
```

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

```
...
```

```
Copy const connectWallet = async () => { const connection = await connect();
```

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

```
...
```

And to reconnect to a previously connected wallet on load:

```
...
```

```
Copy const connection = await connect( { modalMode: "neverAsk" } )
```

```
...
```

Example:

```
...
```

```
Copy useEffect(() => {
```

```
  const connectToStarknet = async () => {
```

```
    const connection = await connect( { modalMode: "neverAsk" } )
```

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

```
    connectToStarknet(); }, [])
```

```
...
```

Connection Parameters

...

```
Copy { alwaysShowDiscovery=true, modalMode={"canAsk","neverAsk"}, storeVersion=getStoreVersionFromBrowser(),
modalTheme->"system" is the default | "dark" | "light" dappName->name of dapp webWalletUrl=DEFAULT_WEBWALLET_URL,
argentMobileOptions, connectors=[], }
```

...

Please refer to the [ArgentMobileConnector](#) section to view available options for `argentMobileOptions`

Disconnecting wallet

To disconnect an existing connection, simply call the `disconnect` method from our imports, then set previously defined states to `undefined` :

...

```
Copy await disconnect();
```

...

Example:

...

```
Copy const disconnectWallet = async () => {
  await disconnect();
  setConnection(undefined); setProvider(undefined); setAddress(""); }
...
```

Disconnection Params

...

```
Copy await disconnect({ clearLastWallet: true })
```

...

Available methods

1. `isConnected`
2.
 - This method is available after an attempt to establish a connection, can be used to confirm if an account was truly connected.
3. `selectedAddress`
4.
 - This method can be called to get the wallet address of a connected account.
5. `account`
6.
 - This method gives us access to the account object. It uses [starknet.js \(opens in a new tab\)](#)
7. `AccountInterface` and extends the `starknet.js Provider`.
- 8.

For more detailed info on the different connectors, how to use StarknetKit with `starknet-react` and many more guides, please check out the official [starknetkit docs](#) .

[Previous Starknet](#) [Next ArgentX](#)

Last updated 2 months ago