

Capsule

[Capsule](#) is a signing solution that you can use to create secure, embedded MPC wallets with just an email or a social login. Capsule wallets are portable across applications, recoverable, and programmable, so your users do not need to create different signers or contract accounts for every application they use.

Read below to learn how to configure your app to create smart accounts for all your users using Capsule and Biconomy. This guide assumes you are using React or a React based web framework such as Next JS. For Capsule guides in other frameworks, check out the [Capsule developer docs](#) or get in touch via hello@usecapsule.com.

Dependencies

You will need the following dependencies to create a Smart Account this way:

```
yarn add @biconomy/account @usecapsule/web-sdk ethers@5.7.2
```

Imports

```
import
{ createSmartAccountClient , LightSigner }
from
"@biconomy/account" ; import
{ Wallet , providers , ethers }
from
"ethers" ;
```

Capsule Configuration

You will need to do some initial setup and get API keys from capsule, for additional help check out the [Capsule Starter Guide](#).

```
import
{ Capsule , Environment }
from
"@usecapsule/web-sdk" ; const capsule =
new
Capsule ( Environment . BETA ,
YOUR_API_KEY ) ; After setup we now need to get a signer from capsule update the import statement to get the Capsule
ethers signer and pass the capsule instance we created above.
import Capsule ,
{ Environment , CapsuleEthersSigner }
from
"@usecapsule/web-sdk" ;
const provider =
new
ethers . JsonRpcProvider ( CHAIN_PROVIDER ,
CHAIN ) ;
const ethersSigner =
```

new

CapsuleEthersSigner (capsule , provider) ;

Create the Biconomy Smart Account

const

connect

=

async

()

=>

{ try

{ const smartAccount =

await

createSmartAccountClient ({ signer : ethersSigner as LightSigner , bundlerUrl :

"" ,

// <-- Read about this at <https://docs.biconomy.io/dashboard#bundler-url> biconomyPaymasterApiKey :

"" ,

// <-- Read about at <https://docs.biconomy.io/dashboard/paymaster> rpcUrl :

""

// <-- read about this at <https://docs.biconomy.io/Account/methods#createsmartaccountclient> }) ;

const address =

await smartAccount . getAccountAddress () ; }

catch

(error)

{ console . error (error) ; } } [Previous DFNS Next Turnkey](#)