

# Particle Network

One way to utilize Social Logins is via Particle Network. This section will give you code snippets for creating Biconomy Smart Accounts with Particle Network. Particle Network allows you to introduce familiar Web2 experiences, with the following code snippets you can unlock: authentication with email to create a smart account as well as authentication with different social providers to create a Smart Account.

tip Check out an end-to-end integration of Particle with Biconomy on this [example app](#) and [repo](#) !

## Dependencies

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

```
yarn add @biconomy/account @biconomy/particle-auth ethers@5.7.2
```

## Imports

```
import
{ createSmartAccountClient , LightSigner }
from
"@biconomy/account" ; import
{ Wallet , providers , ethers }
from
"ethers" ; import
{ ParticleAuthModule , ParticleProvider }
from
"@biconomy/particle-auth" ;
```

## Particle Auth Configuration

Particle auth will require api keys which you can get from the [Particle Dashboard](#) .

```
const particle =
new
ParticleAuthModule . ParticleNetwork ( { projectId :
"", clientKey :
"", appld :
"", wallet :
{ displayWalletEntry :
true , defaultWalletEntryPosition : ParticleAuthModule . WalletEntryPosition . BR , } , } ) ;
```

## Create the Biconomy Smart Account

```
const
connect
=
async
( )
```

```

=>
{ try
{ const userInfo =
await particle . auth . login ( ) ; console . log ( "Logged in user:" , userInfo ) ; const particleProvider =
new
ParticleProvider ( particle . auth ) ; const web3Provider =
new
ethers . providers . Web3Provider ( particleProvider , "any" ) ;
const smartAccount =
await
createSmartAccountClient ( { signer : web3Provider . getSigner ( )
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 Dynamic Next Privy

```