

# DFNS

Dfns provides wallet-as-a-service infrastructure that enables crypto developers to forget about private key management so they can focus on building what matters most — their applications. This guide will show you how to create a smart account using DFNS as a signer.

tip Check out an end-to-end integration of DFNS with Biconomy on this [repo](#) ! Before using the code in this guide you will need to onboard onto DFNS, check out their onboarding guide [here](#) .

Code snippets below will be taken from the repository mentioned above and include some environment variables you will be able to fill in. The example will follow a node js script but can be adopted for frontend frameworks as well.

The first thing we need to do is generate a DFNS wallet. Let's take a look at the imports and then the code for generating the wallet.

## Imports

```
import
{ createSmartAccountClient , LightSigner }
from
"@biconomy/account" ; import
{ DfnsWallet }
from
"@dfns/lib-ethersjs5" ; import
{ DfnsApiClient }
from
"@dfns/sdk" ; import
{ AsymmetricKeySigner }
from
"@dfns/sdk-keysigner" ; import
{ Interface }
from
"@ethersproject/abi" ; import
{ JsonRpcProvider }
from
"@ethersproject/providers" ; import dotenv from
"dotenv" ;
dotenv . config ( ) ;
```

## Generate DFNS Wallet

```
const provider =
new
JsonRpcProvider ( process . env . POLYGON_PROVIDER_URL ) ;
const
initDfnsWallet
```

```

=
( walletId :
string , provider : JsonRpcProvider )

=>
{ const signer =
new
AsymmetricKeySigner ( { privateKey : process . env . DFNS_PRIVATE_KEY ! , credId : process . env . DFNS_CRED_ID ! ,
appOrigin : process . env . DFNS_APP_ORIGIN ! , } ) ;

const dfnsClient =
new
DfnsApiClient ( { appld : process . env . DFNS_APP_ID ! , authToken : process . env . DFNS_AUTH_TOKEN ! , baseUrl :
process . env . DFNS_API_URL ! , signer , } ) ;

return
new
DfnsWallet ( { walletId , dfnsClient , maxRetries :
10 , } ) . connect ( provider ) ; } ;

const mumbaiWallet =
initDfnsWallet ( process . env . POLYGON_WALLET_ID ! , provider ) ; Remember to get your API keys from the DFNS
dashboard and follow theirgetting started guides for this part of the process.

```

## Create Smart Account

```

const createAccount =
async
( ) :
Promise < BiconomySmartAccountV2
=> createSmartAccountClient ( { signer : mumbaiWallet 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 } ) ; You can get all paymaster
and bundler URLs from yourBiconomy Dashboard for this step.

```

With this integration you will have everything you need to begin using the Biconomy Smart Account with DFNS. Remember to check out the[full integration here](#) to see this in action with an example that includes executing a gasless transaction as well. [Previous Magic Next Capsule](#)