

# Pay gas in ERC20 tokens

## Overview

This tutorial demonstrates how to use the Biconomy Smart Account to perform a transaction with ERC-20 token payment. The provided code includes creating a Biconomy Smart Account, encoding a function call, building a transaction, and sending it with ERC-20 token payment.

## Prerequisites

- Node.js installed on your machine
- A Biconomy API key and Bundler url, get from [dashboard](#)

For this tutorial we are going to mint and NFT on Mumbai, this means we can use the following:

- bundlerUrl --> <https://bundler.biconomy.io/api/v2/80001/nJPK7B3ru.dd7f7861-190d-41bd-af80-6877f74b8f44>
- preferred ERC20 token for gas payment --> 0xda5289fcaaf71d52a80a254da614a192b693e977 (USDC)
- nft address --> 0x1758f42Af7026fBbB559Dc60EcE0De3ef81f665e

## Step 1: Create Biconomy Smart Account and Encode Function Call

```
const smartWallet =
await
createSmartAccountClient ( { signer ,
// can be viem client or ethers signer bundlerUrl , biconomyPaymasterApiKey , } ) ;
const encodedCall =
encodeFunctionData ( { abi :
parseAbi ( [ "function safeMint(address _to)" ] ) , functionName :
"safeMint" , args :
[ recipient ] , } ) ; Create a Biconomy Smart Account instance and encode the function call for the transaction.
```

## Step 2: Build the Transaction

```
const nftAddress =
"0x1758f42Af7026fBbB559Dc60EcE0De3ef81f665e" ; const transaction =
{ to : nftAddress , data : encodedCall , } ; Build the transaction with the encoded function call.
```

## Step 3: Send Transaction with ERC-20 Preferred Token Payment

- Preferred token
- List of tokens

Pay the gas in your preferred token.

```
const
{ wait }
=
await smartWallet . sendTransaction ( transaction ,
{ paymasterServiceData :
{ mode : PaymasterMode . ERC20 , preferredToken :
"0xda5289fcaaf71d52a80a254da614a192b693e977" , } , } ) ;
const
```

```

{ receipt :
{ transactionHash } , userOpHash , success , }

=

await
wait ( ) ;

if ( success ==

'true' )

{ console . log ( "UserOp receipt" , receipt ) console . log ( "Transaction receipt" , receipt . receipt ) } You can choose from a
list of ERC20 tokens to pay for the gas, not giving a preferred token will allow you to select from current supported tokens.

const feeQuotesResponse =

await smartWallet . getTokenFees ( transaction ,

{ paymasterServiceData :

{ mode : PaymasterMode . ERC20

} , } ) ;

const userSelectedFeeQuote = feeQuotesResponse . feeQuotes ? . [ 0 ] ;

// Allow user to pick desired token

const

{ wait }

=

await smartWallet . sendTransaction ( transaction ,

{ paymasterServiceData :

{ mode : PaymasterMode . ERC20 , feeQuote : userSelectedFeeQuote , spender : feeQuotesResponse .
tokenPaymasterAddress , maxApproval :

true , } , } ) ;

const

{ receipt :

{ transactionHash } , userOpHash , success , }

=

await
wait ( ) ;

if ( success ==

'true' )

{ console . log ( "UserOp receipt" , receipt ) console . log ( "Transaction receipt" , receipt . receipt ) } Send the transaction
using the Biconomy Smart Account, specifying ERC-20 token payment details. Wait for the transaction to be mined and get
the transaction hash.

```

That's it! You've successfully performed a transaction with ERC-20 token payment using the Biconomy Smart Account. Feel free to customize this example based on your specific use case. [Previous Send a gasless transaction](#) [Next Session Keys Tutorial](#)