

NFT Smart Contract

The point of this tutorial is not to serve as another basic solidity tutorial but let's quickly review our basic NFT smart contract. Feel free to deploy your own version of this contract while completing this tutorial or simply use our implementation of it available [here](#) . This contract has been deployed on the Base Goerli test network. If you need Base Goerli Eth check out the [official documentation](#) on how to get test funds.

Let's take a look at NFT contract:

```
// SPDX-License-Identifier: MIT pragma solidity ^ 0.8 .9 ;

import

"@openzeppelin/contracts/token/ERC721/ERC721.sol" ; import

"@openzeppelin/contracts/utils/Counters.sol" ;

contract BasedBicoNFT is ERC721

{ using Counters

for

Counters . Counter ;

Counters . Counter

private _tokenIdCounter ;

constructor ( )

ERC721 ( "Based Biconomy SDK NFT" ,

"BBNFT" )

{ }

function

safeMint ( address to )

public

{ uint256 tokenId = _tokenIdCounter . current ( ) ; _tokenIdCounter . increment ( ) ; _safeMint ( to , tokenId ) ; } } This is a
basic smart contract which inherits from ERC721 and simply allows the end user to mint an NFT using the defined safeMint
function. It additionally assigns a token id to each minted NFT.
```

Now with the initial smart contract ready, lets get our Paymaster and Bundler ready by using the Biconomy Dashboard.
[Previous Introduction](#) [Next Paymaster, Bundler, And The Biconomy Dashboard](#)