Introduction

The NFT-playbook is a command line tool written in TypeScript. The goal of the tool is to make the process of minting new NFTs as easy as possible. Currently, Ethereum and Solana are supported. There are multiple independent steps required in order to mint a NFT. First of all, you need a smart contract on which to mint the NFT. Additionally, you need a file that the NFT should point to. The NFT-playbook combines all these functionalities into one simple tool. So you can easily: 1. Upload a file to IPFS via Pinata 1. Deploy a new smart contract 1. Mint new NFTs on a smartcontract

Every function can be used independently. So if you just want to upload a file to IPFS, you can do that. But if you want to upload and afterwards use it to mint a NFT, the NFT-playbook makes this as easy as possible for you.

For the usage of the tool we recommend to create an Pinata account: https://www.pinata.cloud/ through this service you can upload your NFTs to IPFS automatically through our tool.

Minting a single NFT

1. Select IPFS/Pinata from the main menu (Optional, but recommended)

1. Provide your Pinata API key & API Secret manually or alternatively over a .env file (see section ".env" below)
2. Provide the path to the file you want to upload
3. Confirm the upload

2. Select Blockchain Settings

1. Select all the blockchains on which you want to mint the NFT at the same time and with the same values
2. Enter your Private Key to authenticate (For privacy reasons, the input is not displayed)
3. For Ethereum: Provide the address of an existing smart contract on which to mint or choose to deploy a new contract

3. Select NFT Minting

1. Provide a name for the NFT
2. Provide a link that the NFT should point to (If you uploaded your file with Pinata, this is already pre-filled)
3. Double-check the summary and confirm to mint

Mint multiple NFTs (Bulk Minting)

For Bulk Minting a Pinata Account is mandatory!

1. Create a JSON- file with all information about the NFTs you want to mint in the root folder. The file must have the following structure:

```
{
   "name": "Desired Name for your NFT",
       "path": "local Path to Object",
       "blockchains": [
         {
           "name": "Ethereum",
           "receiver": "Public key of the receiver of the NFT"
         },
         {
           "name": "Solana",
           "receiver": "Public key of the receiver of the NFT"
        }
       ]
    }
```

2. Open the CLI and enter your Pinata credentials (See Step 1 of "Minting a single NFT")
3. Provide all your Blockchain credentials in the Blockchain menu (see Step 2 of "Minting a single NFT")
4. Go to Bulk Minting menu and provide the path to your created JSON file
5. Check the summary of your minting process and press enter

settings.json

This file holds some additional settings that are relevant for every blockchain you want to mint an NFT on. Currently there are only 2: Ethereum and Solana. Structure: Structure: 1. `config` is an array that holds information for every blockchain. 1. `blockchain` blockchain name. 2. `settings` 1. `GAS_LIMIT` a gas fee limit that you can set 2. `endPoint` the URI or chainID to the Ethereum network. Can be set for testing purposes e.g. to the address of some test network or localhost 2. `log_file` path to the file that shall be used for logging.

.env

In this file you can save your Pinata credentials so that you don't have to enter them manually after every program start. This is an optional step. If you do not provide this file, the CLI will ask for your Pinata Key & Secret at runtime. The file has to look like this:

`PINATA_API_KEY="YOUR KEY HERE" PINATA_API_SEC="YOUR SEC HERE"`

logfile

The NFT-playbook writes a log to the file specified in settings.json. Events that are logged include: 1. IPFS/Pinata: file uploads; IPFS Hash 2. NFT Minting: successful mint; failed mint