Generate Your NFT Metadata

NFT Metadata, a crucial element in an NFT smart contract

Edward Jones Nov 10, 2021 · 3 min read ★

Part two of the four-part NFT Creator series

This is the second part of the four-part NFT Creator series:

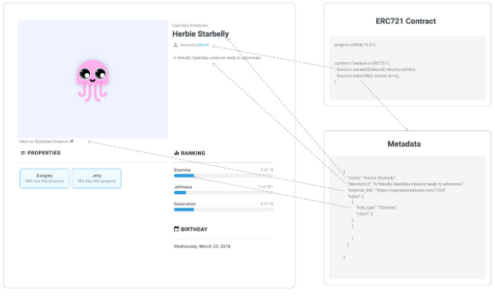
1. *Generate NFT images*
2. **Generate NFT metadata**
3. *Deploy NFT smart contracts*
4. *NFT minting*

What is NFT Metadata?

NFT metadata is the core of an NFT. It is a JSON document that often contains the following:

- NFT's name
- Description of the NFT
- Link to the hosted image
- Traits
- …

This NFT metadata will be the input of your NFT smart contract which you will deploy on the Ethereum network in the third part.



Example of NFT Metadata (source: https://docs.opensea.io/docs/metadata-standards)

Upload your NFT images in the cloud

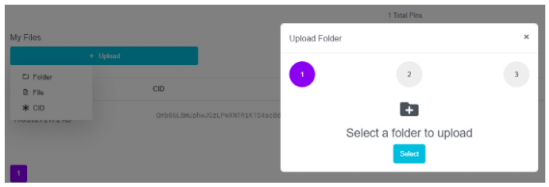
Uploading images to the blockchain is very expensive since they are large in size.

The best practice is to only upload the link of your image to the blockchain and store your image on an Interplanetary File System (more on that later).

Pinata allows you to upload NFT images for free using IPFS (InterPlanetary File System). This is a distributed file-sharing system.

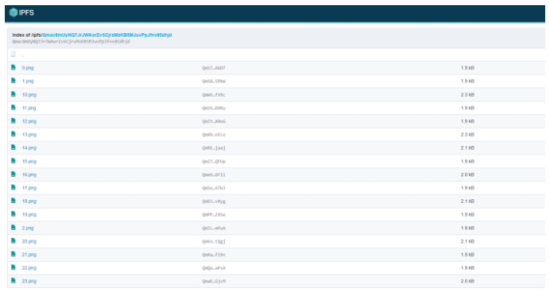
Sign up for a free account and **upload your NFT images folder** to the pinata cloud.

I called my project "NFT Creator" but feel free to choose a different name.



Pinata allows you to upload your entire folder at once

If the upload is successful, you should be able to see your uploaded files as shown below:



Click on your project and copy the link.

This is your "BASE URL", which you will need later on.

The BASE URL of my project is https://gateway.pinata.cloud/ipfs/Qmb86L8mUphwJGzLPwXNTRiK1S4scBdj9cc2Sev3s8uLiB

Generate NFT metadata

all-traits.json

In the first part of the NFT creator series, you have created a list called "all_images" specifying the traits for each image.

Simply, dump this list into a .json file using the json.dump() function.

```
            "Nose :  n1 ,
            "tokenId": 1
        },
```

all-traits.json file

[token_id].json

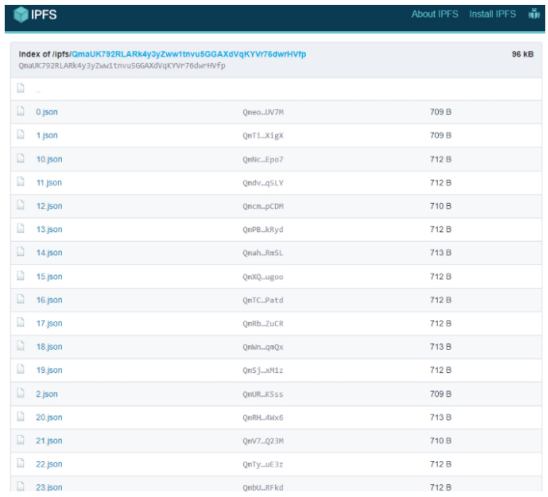
Next, you want to create a specific .json file for each image:

- load in the all_traits.json
- Specify your images "BASE URL" which you copied earlier on the Pinata website. Make sure you add an additional "/" on the end!
- Specify your project name
- Loop over the all_traits .json dictionary using the keys accessor and output an individual .json file for each unique NFT image.

For this image, you will for example receive the following .json file:

Upload the metadata to pinata

Upload your generated metadata to Pinata in the same way as for the images.



Example NFT Metadata

Below you can find an example of the NFT_metadata file for tokenID 0 & the corresponding picture.

In the next part of the series, you will learn how to deploy an NFT smart contract.

Happy programming!


```
        "trait_type": "Eyes",
        "value": "regular"
    },
    {
        "trait_type": "Ears",
        "value": "ears1"
    },
    {
        "trait_type": "Hair",
        "value": "hair7"
    },
    {
        "trait_type": "Nose",
        "value": "n2"
    },
    {
        "trait_type": "Mouth",
        "value": "m4"
    }
  ]
}
```


NFT Metadata of tokenID 0

. . .

Roadmap

1. *Create Your Own NFT Collection With Python*
2. **Create NFT Metadata**
3. *Deploy NFT smart contracts*
4. *NFT minting*

Resources

1. Ethereum Developer Resources | ethereum.org
2. GitHub — UniqueNetwork/substrapunks: Substrate based remake of CryptoPunks game
3. GitHub — benyaminahmed/nft-image-generator

Thanks to Anupam Chugh.

Programming Nft Ethereum Python Blockchain

Ismail GÖK · Nov 10, 2021

Retain Cycles and Memory Management in Swift

Build performant iOS applications