DRAFT – WORK IN PROGRESS

NFT Making on the IC

MY NOTES AS A BEGINNER

@tedreinhardt

Ver 0.7
11 Feb 2022

Aim

- Share what I have been learning about creating an NFTs so that others may be able to advance their own projects.

- I have very little experience. I minted my first 360 degree NFTs in August/2021 using DepartureLabs Minter. Here is the NFT content.
  - https://jb6ng-naaaa-aaaaf-qacvq-cai.raw.ic0.app/mmiwg.html

Disclaimer

- These are my notes and may contain inaccuracies.
- Use the information at your own risk.

A word of advice: Never share your wallet recovery seed with anyone.

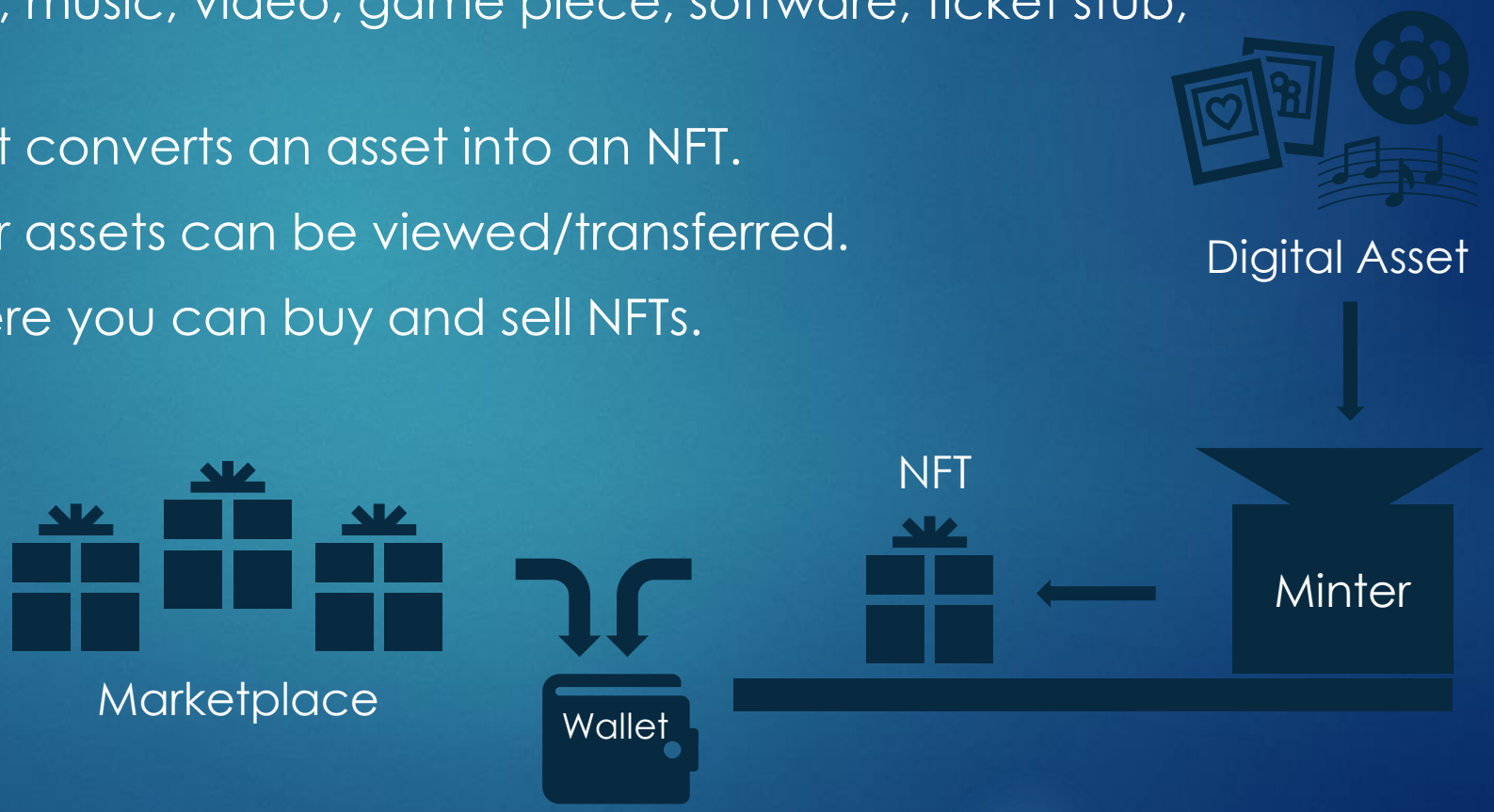
Non-Fungible Token (NFT)

- A non-fungible token (NFT) is a unique and non-interchangeable unit of data stored on a digital ledger (blockchain).
- NFTs can be associated with reproducible digital files such as photos, videos, and audio.
- NFTs use a digital ledger to provide a public certificate of authenticity or proof of ownership, but it does not restrict the sharing or copying of the underlying digital file.
- The lack of interchangeability (fungibility) distinguishes NFTs from blockchain cryptocurrencies, such as Bitcoin.

Source: Wikipedia contributors. (2021, December 15). Non-fungible token.
In Wikipedia, The Free Encyclopedia. Retrieved 19:53, December 15, 2021, from https://en.wikipedia.org/w/index.php?title=Non-fungible_token&oldid=1060457068

The NFT terminology

- Digital asset: photo, music, video, game piece, software, ticket stub, etc.
- Minter: process that converts an asset into an NFT.
- Wallet – where your assets can be viewed/transferred.
- Marketplace – where you can buy and sell NFTs.



Difference between a digital asset and a NFT?

Digital Asset + Certificate of ownership + Built-in transfer, registration and access services.

Optionally a program to transform or display the asset or deliver a service.

An NFT

The IC enables you to do all this on chain in a single canister (smart contract).
It also economical on the IC.

Costs of minting an NFT on the IC

- NFTs are stored in canisters.
- Minting a canister costs about $3 USD on the IC:
  - Canister creation costs 2 trillion Cycles.
  - 1 trillion cycles costs 1 Standard Define Rate (SDR), a term defined by the International Monetary Fund or approximately $1.42 USD as at 15 Dec 2021.
- In addition, there is an ongoing fee for storage, transmittal and processing.
- Overtime, the cost of storage and execution will have to be topped up unless it is loaded with more cycles to maintain it. If not, it will be removed from the blockchain when it is depleted.

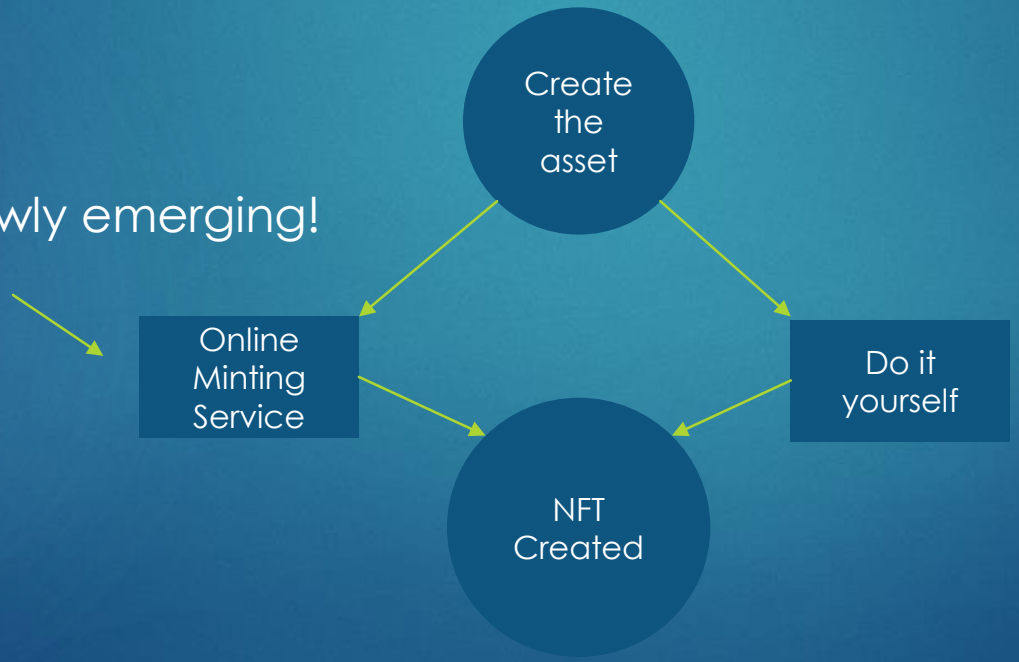
Multiple NFTs can also be stored in a single canister reducing cost.

- Using a multi-NFT canister, minting 100 000 tokens would costs around $3 vs $300 000.

How do you make an NFT?

- Create an image and save as a .png file. (Any digital asset supported by the minter can be used.)
- Use an online minter to create the NFT or do it yourself.

They are slowly emerging!



Online minting of your content

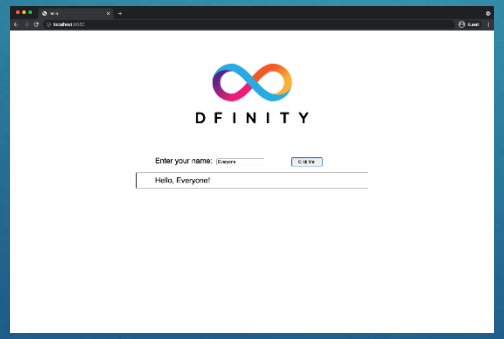
- DepartureLabs had an experimental minter – offline now.
- NFT anvil is in development.

  https://5rttq-yqaaa-aaaai-qa2ea-cai.raw.ic0.app/mint
- Toniqlabs will mint collections as part of the collect intake process if deemed of interest into Entrepot.
- BobNFTs.com is a new minting service with potential.
- NFTStudio.biz is a new minting service with potential.
- Blocks Editor has a DIP721 which can be built to mint.

Let me know if you find any others.

Create it yourself

- You will need to have the DFINITY Canister SDK and dependencies installed and running on a computer using :
  - Windows with WSL2 and linux subsystem installed (e.g. Ubuntu2004);
  - Linux; or
  - Mac
- Follow the quick start tutorial and deploy the sdk Hello example.
- Test to see if you can see the front end that looks like this:



So what did that achieve?

- Following the quick start tutorial mean you created a smart contract owned by you that displays an image (the dfinity symbol) stored on the local replica. It has an index.html, index.js and the dfinity logo png.
- Turning into an NFT means adding the data elements for the certificate of ownership data and the functions to enable transfer and management of the NFT.
- The data elements follow a format called ERC-721.

ERC-721

- A standard called ERC-721 (ERC is Ethereum Request for Comment) defines data elements and methods that must be present when constructing an NFT for trading within the Ethereum eco-system to aid interoperability with wallets and ecosystems.
- The Internet Computer community has not defined a standard however a variety of ERC-721 inspired implementations have been implemented:
  - Toniq-Labs/extendable-token
  - DepartureLabsIC/non-fungible-token
  - C3-Protocol/NFT-standards
  - rocklabs-io/ic-nft
  - Psychedelic DIP721 a rust implementation.
- These implementations are all available on github.

ERC-1155 – Multi-NFT

- ERC1155 uses a single smart contract to represent multiple tokens at once.
- Storing multiple NFTs into one smart contract reduces the cost of creating a canister for each and every NFT.
- Toniq-Labs/extendable-token examples has an advanced token that enables multiple NFTs to be stored in a single canister available on GITHUB.

For the purpose of this document, building is shown using an ERC-721-like token not the multi-NFT canister.

A word about wallets

- Wallets are a place to load, view and transfer NFTs. They will accept NFTs in formats they support.
- Popular wallets are:
  - Stoic Wallet by Toniq Labs and
  - Plug Wallet
  - EarthWallet by EarthDAO

Wallets only work with NFTs in certain formats. If it is in the wrong format, it may be possible to use a wrapper to transform one NFT format into another NFT format. For example, the metadata may need certain encoding.

Creating a NFT and moving it to a Wallet

- Here is a process overview.

1. Create the asset
2. Build your minter
3. Mint your NFT
4. Load it in your wallet

Steps to build (after you installed the SDK)

- Create a new project called nft1

```
$ dfx new nft1
```

- It will install and when successful displays the following.

Clone the NFT code repository from GITHUB

- Change directory into ~nft1.
- Install github (if not already done google for your particular platform).
- Clone the repository (repo) locally.

```
$ git clone git@github.com:Toniq-Labs/extendable-token.git
```

- Copy the erc721.mo file in the examples subdirectory into the ~nft1/src/nft1 subdirectory.
- Rename the existing main.mo to main.old or delete it.
- Rename the erc721.mo to main.mo

This replaces the Motoko code with the ERC721 code.

Move the dependencies

- Copy the extendable-token motoko sub-directory and its contents into the ~nft1/src directory.
- Check make ext and util directories are their and the files are there:

```
$ ls -lR motoko
motoko:
total 8
drwxr-xr-x 2 jhm jhm 4096 Dec 16 10:24 ext
drwxr-xr-x 2 jhm jhm 4096 Dec 16 10:24 util

motoko/ext:
total 48
-rw-r--r-- 1 jhm jhm  621 Dec 16 10:24 Allowance.mo
-rw-r--r-- 1 jhm jhm  831 Dec 16 10:24 Archive.mo
-rw-r--r-- 1 jhm jhm  445 Dec 16 10:24 Batch.mo
-rw-r--r-- 1 jhm jhm  550 Dec 16 10:24 Common.mo
-rw-r--r-- 1 jhm jhm 6672 Dec 16 10:24 Core.mo
-rw-r--r-- 1 jhm jhm  400 Dec 16 10:24 Fee.mo
-rw-r--r-- 1 jhm jhm  701 Dec 16 10:24 Ledger.mo
-rw-r--r-- 1 jhm jhm  394 Dec 16 10:24 NonFungible.mo
-rw-r--r-- 1 jhm jhm  999 Dec 16 10:24 Operator.mo
-rw-r--r-- 1 jhm jhm  546 Dec 16 10:24 Secure.mo
-rw-r--r-- 1 jhm jhm  202 Dec 16 10:24 Subscribe.mo

motoko/util:
total 20
-rw-r--r-- 1 jhm jhm 1564 Dec 16 10:24 AccountIdentifier.mo
-rw-r--r-- 1 jhm jhm 3991 Dec 16 10:24 CRC32.mo
-rw-r--r-- 1 jhm jhm 1866 Dec 16 10:24 Hex.mo
-rw-r--r-- 1 jhm jhm 5392 Dec 16 10:24 SHA224.mo
```

Start the local replica

In a separate window/terminal session.

- dfx start --clean warning: --clean wipes out the old replica state

OR

- dfx start

```
$ Dfx start --clean
Starting webserver for /_/
binding to: 127.0.0.1:45621
Dec 18 11:30:16.222 INFO ic-starter. Configuration: ValidatedConfig { replica_path: Some("/home/ted/.cache/dfinity/versions/0.8.4/replica"),
replica_version: "0.8.0", log_level: Warning, cargo_bin: "cargo", cargo_opts: "", state_dir: "/home/ted/nft1/.dfx/state/replicated_state",
http_listen_addr: 127.0.0.1:0, http_port_file: Some("/home/ted/nft1/.dfx/replica-configuration/replica-1.port"), metrics_addr: None,
provisional_whitelist: Some(All), artifact_pool_dir: "/home/ted/nft1/.dfx/state/replicated_state/node-100/ic_consensus_pool", crypto_root:
"/home/ted/nft1/.dfx/state/replicated_state/node-100/crypto", state_manager_root: "/home/ted/nft1/.dfx/state/replicated_state/node-
100/state", registry_local_store_path: "/home/ted/nft1/.dfx/state/replicated_state/ic_registry_local_store", unit_delay: None, initial_notary_delay:
Some(600ms), detect_consensus_starvation: None, consensus_pool_backend: Some("rocksdb"), state_dir_holder: None }, Application: starter
Dec 18 11:30:16.223 INFO Initialize replica configuration "/home/ted/nft1/.dfx/state/replicated_state/ic.json5", Application: starter
Dec 18 11:30:17.154 INFO Executing "/home/ted/.cache/dfinity/versions/0.8.4/replica" "--replica-version" "0.8.0" "--config-file"
"/home/ted/nft1/.dfx/state/replicated_state/ic.json5", Application: starter
Dec 18 11:30:23.398 WARN s:knvjx-zgkm4-hkvvw-3ueag-xlh7y-zje4m-54tyb-vp2bf-xigrc-mrffa-xae/n:6ufqp-7ok73-oxrj6-zgchl-xllg5-dukck-4zeln-
dlz3g-megws-lov4f-qae/ic_p2p/download_management PeerManagerImpl::new(): relay_config = None
version: 0.7.0
 Dec 18 06:30:23.433 INFO Log Level: INFO
 Dec 18 06:30:23.433 INFO Starting server. Listening on http://127.0.0.1:8000/
```

Create an empty Canister


```
$ dfx canister create nft1
Creating a wallet canister on the local network.
The wallet canister on the "local" network for user "default" is "rwlgt-iiaaa-aaaaa-aaaaa-cai"
Creating canister "nft1"...
"nft1" canister created with canister id: "rrkah-fqaaa-aaaaa-aaaaq-cai"
```

Build the wasm code

- dfx build nft1

These warnings are not an issue

```
$ dfx build nft1
Building canisters...
/home/ted/nft1/src/motoko/util/Hex.mo:48.17-48.30: warning [M0154], field unwrap is deprecated:
Option.unwrap is unsafe and fails if the argument is null; it will be removed soon; use a `switch` or `do?` expression instead
/home/ted/nft1/src/motoko/util/Hex.mo:49.17-49.30: warning [M0154], field unwrap is deprecated:
Option.unwrap is unsafe and fails if the argument is null; it will be removed soon; use a `switch` or `do?` expression instead
/home/ted/nft1/src/motoko/util/Hex.mo:48.17-48.30: warning [M0154], field unwrap is deprecated:
Option.unwrap is unsafe and fails if the argument is null; it will be removed soon; use a `switch` or `do?` expression instead
/home/ted/nft1/src/motoko/util/Hex.mo:49.17-49.30: warning [M0154], field unwrap is deprecated:
Option.unwrap is unsafe and fails if the argument is null; it will be removed soon; use a `switch` or `do?` expression instead
/home/ted/nft1/src/motoko/util/AccountIdentifier.mo:38.14-38.27: warning [M0154], field unwrap is deprecated:
Option.unwrap is unsafe and fails if the argument is null; it will be removed soon; use a `switch` or `do?` expression instead
```

Get the principal

- dfx identity get-principal will show you the identity you are using on the local replica. Your principal response will be different to this.

```
$ dfx identity get-principal
zht7g-jivec-azc2g-f5bkj-oxsfc-nvyo6-ch4jb-el2tb-ebyuf-zeo4d-gae
```

Copy your result to the clipboard

Install the code into the canister

- The dfx canister install is used to move the code into the canister and to set the owner of the registry to your principal. Your principle will be different (the output from the dfx identity get-principal)

```
dfx canister install nft1 --argument="(principal \"zht7g-jivec-azc2g-f5bkj-oxsfc-nvyo6-ch4jb-el2tb-ebyuf-zeo4d-gae\")"
```


Change to match your principal from the previous step

Status check

- You now have a canister with the dfinity logo in it and the NFT logic to manage the token. It also has an HTML page to render it and a no longer needed index.js which can be removed.
- Check to see which identity will be the minter.

```
$ dfx canister call nft1 getMinter
(principal "zht7g-jivec-azc2g-f5bkj-oxsfc-nvyo6-ch4jb-el2tb-ebyuf-zeo4d-gae")
```

Should match your principal

Mint your first NFT

- Call the mintNFT method passing the principal (vs a stoic address) with the appropriate principal

One line

```
$dfx canister call nft1 mintNFT "(record { to = (variant { \"principal\" = principal \"zht7g-jivec-azc2g-f5bkj-oxsfc-nvyo6-ch4jb-el2tb-ebyuf-zeo4d-gae\" })} )“

(0 : nat32)
```

Result

Minted a NFT in position 0 of the registry.

Run it again and the next one will be in position 1

Sets the NFT token owner and may be different for each token.

You can use dfx to call public methods of the canister

- getMinter() e.g. dfx canister call nft1 getMinter
- setMinter(minter : Principal)
- mintNFT(request : MintRequest)
- transfer(request: TransferRequest)
- approve(request: ApproveRequest)
- extensions()
- balance(request : BalanceRequest)
- allowance(request : AllowanceRequest)

…continued

- bearer(token : TokenIdentifier)
- supply(token : TokenIdentifier)
- getRegistry()
- getAllowances()
- getTokens()
- metadata(token
- public func acceptCycles()
- availableCycles()

This is what was achieved

- You minted an NFT with ownership information stored within the canister on the local replica.
- There is no access control implemented – anyone can see the content of the canister or call the methods – not just the NFT holder.
- Upgrading the contract may clobber persistent data (I have not checked this) so be aware.
- You can mint additional tokens by running the mint command again and it will store it in the same canister with a index increment.

Deploying an NFT costs cycles

- Creating a canister costs 2 Trillion cycles.
- Sign up to GITHUB to get free cycles from the Dfinity cycles faucet:

  https://faucet.dfinity.org/auth

- With a GITHUB account created more than 90 days ago, you can get free cycles to test deployment on the IC.
- The Faucet will give about $20 worth of cycles once.

Now moving to the real world

- Doing it on the IC involves:
  - creating a canister (incurs a cost of 2 x SDR) using the NNS or using dfx
  - Noting the canister ID.
  - Replacing the Dfinity logo image file with your digital asset image.
  - Clean up the code (index.js and index.html).
  - **Add access controls in the Motoko code.**
  - Installing the code into the canister on the IC with the correct principle.
  - Then run the dfx minter command pointing to the canister on the IC.

```
dfx deploy --network ic --argument="(principal \"zht7g-jivec-azc2g-f5bkj-oxsfc-nvyo6-ch4jb-el2tb-ebyuf-zeo4d-gae\")"
dfx canister --network ic call nft1 mintNFT "(record { to = (variant { \"principal\" = principal \"zht7g-jivec-azc2g-f5bkj-oxsfc-nvyo6-ch4jb-el2tb-ebyuf-zeo4d-gae\" }); metadata=opt vec
{92;120;48;65;116;105;100;34;114;121;106;108;51;45;116;121;97;97;97;45;97;97;97;97;97;45;97;97;97;98;97;45;99;97;105;48;48;48;49}  } )"
dfx canister --network ic call nft1 getTokens
```

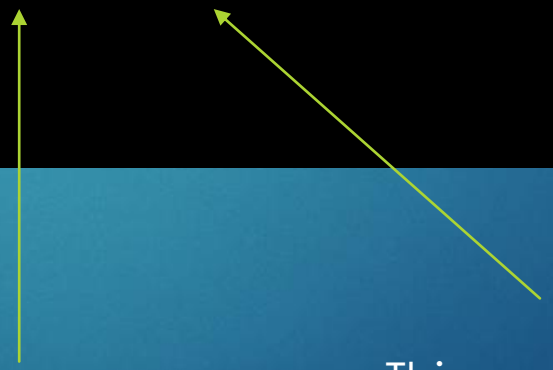
Now moving to the real world

- Doing it on the IC involves:

```
dfx deploy --network ic --argument="(principal \"zht7g-jivec-azc2g-f5bkj-oxsfc-nvyo6-ch4jb-el2tb-ebyuf-zeo4d-gae\")"

dfx canister --network ic call nft1 mintNFT "(record { to = (variant { \"principal\" = principal \"zht7g-jivec-azc2g-f5bkj-oxsfc-nvyo6-ch4jb-el2tb-ebyuf-zeo4d-gae\" }); metadata=opt vec {92;120;48;65;116;105;100;34;114;121;106;108;51;45;116;121;97;97;97;45;97;97;97;97;97;45;97;97;97;98;97;45;99;97;105;48;48;48;49}  } )"

dfx canister --network ic call nft1 getTokens
```



This is one line (dfx… })"

This metadata I encoded for it to be accepted by a particular wallet - to be tested.

Moving into a wallet (theory):

In order for this to work, the NFT must be minted with the specific meta data and format supported by the wallet.

- Install a wallet that supports NFTs (e.g. Stoic wallet)
- Go to the NFT option
- Add NFTs
- Enter the canister ID when you created it on the IC
- Boom! Done! (if the canister is in a compatible format).

To be verified and not working yet.

Also, need to create an Endpoint to display it.