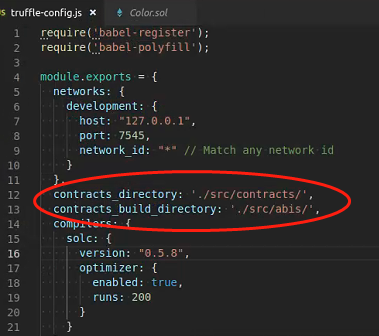
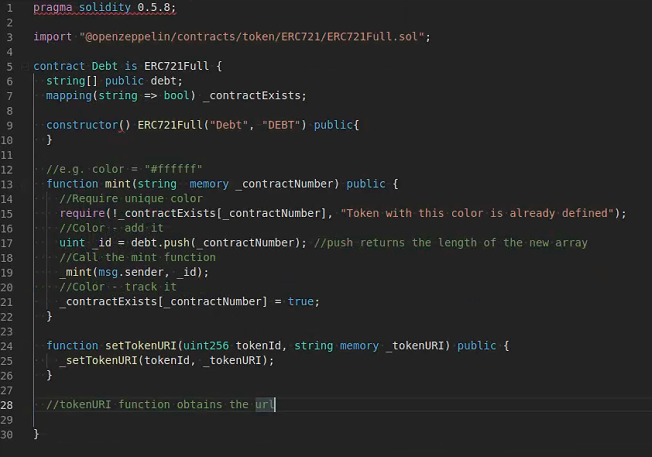
# Create an ERC721 Debt Token

<https://github.com/onebit256/erc-721>

## Explain the Color code



## Modify Color Contract to Debt Contract



Const dbt = await Debt.deployed()

Dbt.address

Dbt.mint(“123”)

Await dbt.totalSupply()

Dbt.debt(0)

Dbt.debt(1)

Dbt.totalSupply()

Dbt.setTokenURI(1,”http://127.0.0.1:30000”)

Dbt.tokenURI(1)

So Application can read TokenURI to get metadata

The Metadata could be stored on IPFS or centralized server, to display on the website

Dbt.ownerOf(1)

Dbt.transferFrom(accounts[0],accounts[1])

Dbt.ownerOf(1)

# 2 Opensea

## 2.1 Deployment

https://docs.opensea.io/docs/getting-started

git clone https://github.com/ProjectOpenSea/opensea-creatures.git

Step1: Update truffle config

export ALCHEMY\_KEY="<your\_alchemy\_project\_id>"

export MNEMONIC="<metamask>"

export NETWORK="rinkeby"

truffle deploy --network rinkeby

Step 2: deploy the smartcontract

Step 3: create the frontend

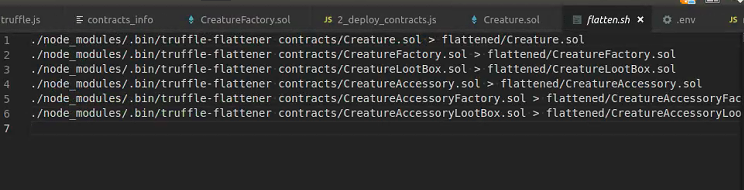
testnets.opensea.io/get-listed/step-two

## 2.2 Code Explanation

Factory Address

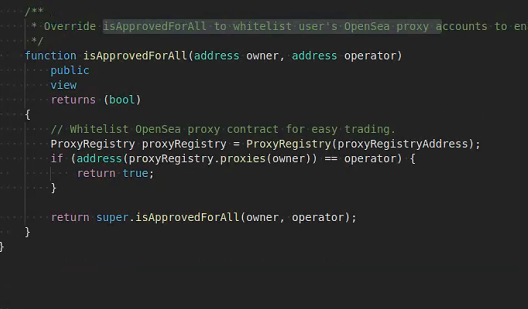
https://testnets.opensea.io/assets/<factory\_contract\_address>/<option\_id>

### 2.2.1 Flattner

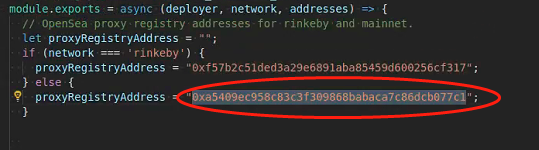


When we want to call a stable library, we use flattner to throw all dependencies into one file

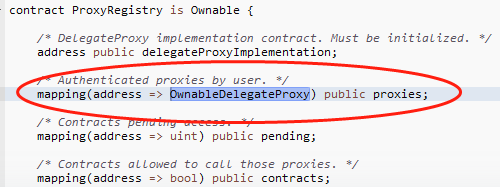
### 2.2.2 Creature, ERC721Tradeable, Proxy and logic contract



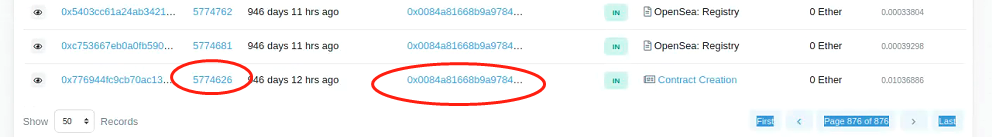
deployjs

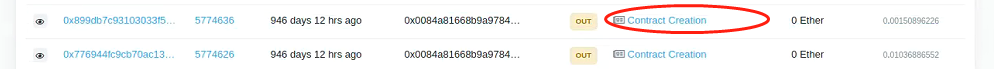


<https://etherscan.io/address/0xa5409ec958c83c3f309868babaca7c86dcb077c1#code>





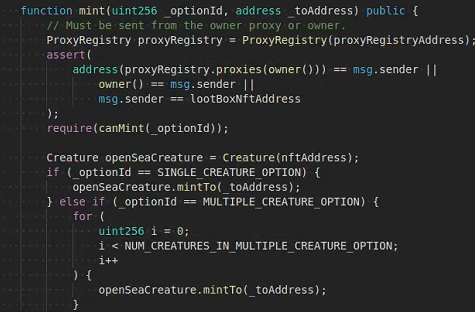




Already registered

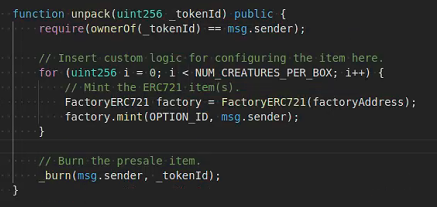
### 2.2.3 Factory Function

Auto mint



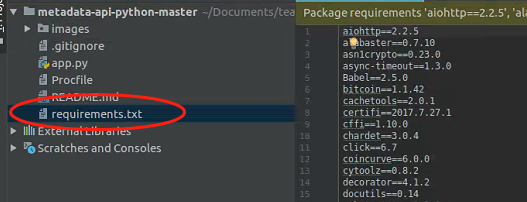
2.2.4 Lootbox

Core function unpack



# 3 Metadata and Flask

Pycharm



Pip install -r requirements.txt