

Crypto Collectable Wallet

Introduction

There are roughly two types of virtual assets on blockchain:

1. Fungible Token (currency, shares):
For this kind of token, all that matters is the amount. 100 units of mine is exactly the same as 100 units of yours.
2. Non-Fungible Token (collectable, art, music, player cards):
Every single unit of this kind of token is unique.

ERC-721/ERC-1155 is the most common standard for non-fungible tokens on blockchain. In this interview project, you have to implement a simple display page for ERC-721/ERC-1155 collectables, to showcase all the collectables in an account.

An ERC-721 token looks like this:

<https://opensea.io/assets/0x4164692f986daea213ba582cfed9ec1155462107/0>

Basic Requirements

1. There are two pages: list page and detail page. Use client side navigation to switch between different pages.
2. You can use any frontend framework you like (react, vue, angular, ...).
3. Use any library when you see fit. There's no point reinventing the wheel.
4. Do not spend too much time on this project. Spend 2-4 hours.

List page:

1. Use OpenSea API to get the collectables in this Ethereum account `0x960DE9907A2e2f5363646d48D7FB675Cd2892e91`. You can hard-code this address.
2. Implement pagination. Get 20 items each time and fetch the next 20 when user reaches the end of the page (infinite scrolling)

API Spec

EndPoint: <https://api.opensea.io/api/v1/assets> GET Method

Parameters:

- format=json
- owner={address} // 0x960DE9907A2e2f5363646d48D7FB675Cd2892e91
- offset=0...N
- limit=20

Response:

- image_url
- name
- collection.name
- description

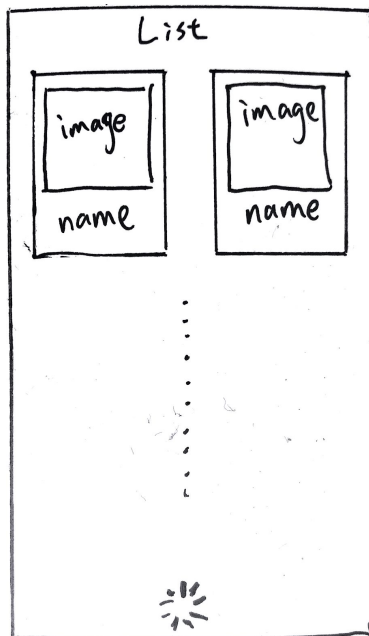
- permalink

More Documentation: <https://docs.opensea.io/reference#getting-assets>

UI Spec

Show two columns of items. Each item shows the name and the image.

Implement basic RWD that works on both desktop and mobile. But don't spend too much time on the design. The design will not be scored.



When a user clicks on one of the items, go to the detail page.

Detail Page:

API Spec

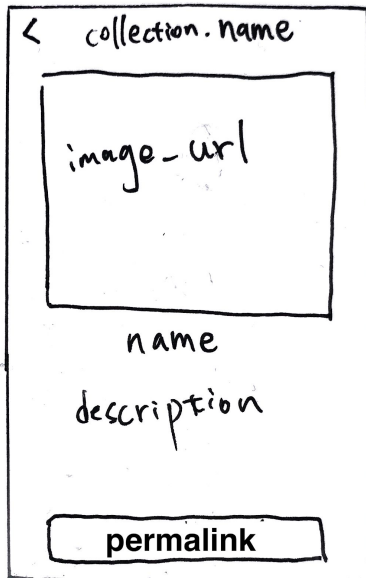
EndPoint: [https://api.opensea.io/api/v1/asset/\[contract_address\]/\[token_id\]](https://api.opensea.io/api/v1/asset/[contract_address]/[token_id]) GET Method

More Documentation: <https://docs.opensea.io/reference#retrieving-a-single-asset>

UI Spec

- Show collection name on the header
- Fit the image on the screen. Image should take up the entire width.
- Display the name of the collectable
- Display the description of the collectable, note that some collectibles have a very long description.
- A fixed button on the bottom that goes to the permalink of the collectable

Again. Implement basic RWD that works on both desktop and mobile. But don't spend too much time on the design. The design will not be scored.



Bonus

- Use RxJS for data retrieval
- Use web3.js and use the address from the Metamask Chrome/Firefox extension instead of a hard-coded address

Reference: <https://web3js.readthedocs.io/en/v1.2.6/getting-started.html>

Scoring

- Readability of the documents/comments
- Readability of the code
- How modularized is the code: high cohesion, low coupling
- Performance (rendering and scripting)
- (optional) the bonus items

Submission

Push the code to Github or Gitlab and send the link to careers@portto.io