#### CryptoX wallet – NFT

#### Functional requirements

* We need to provide ability for user to view his tokens
* User need to have special tab or screen with list of tokens
* User need to have option to import/refresh/sync his list of tokens
* User need to have ability to manually import tokens (specifying base API url), that returns list of tokens
* User need to have ability to search NFT by name or description

CryptoX general behavior:

1. User press button for sync Token list with Spaceseven.

Or press button “Import” and enter base url of NFT marketplace

2. After some waiting for response, NFT list, that user owns is displayed as gallery

3. User have ability to make “refresh” tokens

4. When user press NFT card, it prompt to open url in browser, that leads to marketplace NFT page with all details

5. When user press search button/icon, only relevant tokens is shown

CryptoX need to save NFT data:

* marketplace base url
* blockchain (Concordium)
* Smartcontract address (index, subindex)
* NFT id
* NFT page url at marketplace
* NFT icon image
* NFT name
* NFT metadata structure (owner, etc.)

#### Data Model for NFT

Token unique identifier:

* Blockchain (always Concordium)
* SmartContract address (Index int, Subindex int)
* NFT id (bigint ?)

Token metadata

* marketplace base url (NFT page at marketplace)
* NFT metadata url (saved in blockchain)
* NFT name
* owner address (=user address)
* author address
* author royalty
* link to small preview image
* link to image
* sale status (fixed, auction, not for sale)

#### Backend API

json-rpc call to API (POST) "get\_tokens\_by\_address"

**Marketplace endpoint must be mapped to following full path:**

**<$marketplaceDomain>/api/v2/nft/get-by-address**

**When user add additional marketplace for NFT sync, he enter base url, without path to api, for example:**

**https://spaceseven.com**

**https://dev.spaceseven.com**

Marketplace API must accept following input (: blockchain, address

{

"blockchain\_id":"<blockchain\_id",

"address":"<user\_address>",

"page\_start": 1, // pagination start item

"page\_limit": 6, //items count to load

"search\_string": "" //optional, will limit tokens with pattern match NFT name or NFT description

}

where: blockchain\_id could be 1 or 2 as value. Optional attribute

we use 2 for Concordium and its default, if not specified.

address - user wallet address. In case several address is exists, CryptoX need to make several requests for each address.

page\_start - starting item, default 0

page\_lilmit - items count limit, default 6

Request example (curl)

curl --location --request POST 'https://dev.nftmarketplace.com/api/v2/nft/get-by-address' \

--header 'Content-Type: application/json' \

--data-raw '{

"blockchain\_id":"2",

"address":"4NUoN8NeMXXXXXXXXXXXXXXX9Mn2jnFk8xBPavUonzN",

"page\_start": 0,

"page\_limit": 6,

"search\_string": ""

}'

Backend should get user tokens by address and with pattern match (search). Sorting should be new->old Search keyword will be applied to NFT name or description (e.g.

nft\_name like 'search\_string%' or description like 'search\_string%'

)

output:

Backend must handle request with following rules:

* handle and validate request structure
* handle address (check if valid)
* make query to database and retrieve all minted tokens, where address is stated as owner - with any hidden status and published=true, copies>0; filtered by blockchain and search\_string pattern
* multiple NFT (1155) must be returned as 1 record, and quantity reflects how many tokens is owned
* for multiple NFT (1155), in case several different lots is active (auction and fixed), status should be Fixed\_Sale

* JSON array with tokens, if found:

{"response\_meta":{"status":"OK"},

"tokens":

[ {

"blockchain\_id": "ID",

"smartcontract\_address": [

"INDEX",

"SUBINDEX"

],

"nft\_id": "ID",//nft id in blockchain smartcontract

"marketplace": "base\_url",//e.g. https://spaceseven.com/api/get\_tokens\_by\_address

"marketplace\_name":"<e.g. Spaceseven>",

"nft\_page":"nft\_marketplace\_url",//e.g. https://spaceseven.com/nft/164941167388000272

"nft\_metadata\_url": "ipfs\_metadata\_link",//e.g. https://gateway.pinata.cloud/ipfs/bafkreia2tyrybdwr7j46reqmqn6fkwohid5cuvydtug4tf3zyavncpftiu

"nft\_name": "name",

"owner\_address": "user\_address",

"author\_address": "author\_address",

"author\_royalty": "royalty\_pct", //formated to NN.NN

"icon\_preview\_url": "", //e.g. https://storage.googleapis.com/spaceseven-public/nft/preview-opt/06/20085185-ce1f-4db4-8f93-43d30f3f1dbd\_small.webp

"image\_url": "", //e.g. https://storage.googleapis.com/spaceseven-public-dev/nft/preview-opt/06/20085185-ce1f-4db4-8f93-43d30f3f1dbd\_large.webp

"quantity":"", // always 1 for single, X for multpiple

"total\_minted":"", // always 1 for single, Y for multiple

"sale\_status": "" //not for sale, fixed price sale, auction

},

{}

],

"page\_start":"1",

"page\_limit":"6",

"total\_items":"54"

}

To display each token, CryptoX need to extract and save icon\_preview image from external resources by link.

page\_start - item, which must be 1st in output, starting from 0

page\_limit - items count limit in response

total\_items - total count of items found in backend database, according to input (address, search\_string )

Other information should be displayed as is (depends, how UI designed).

* error message with description, to show for user, in case no token found or some validation/system error

{

"response\_meta": {

"status": "NotFound",

"error": {

"messages": [

"no tokens found"

],

"code": "308320188405645580"

}

}

}

Error messages:  
"no tokens found"  
"invalid request" (validation or request handling fails)

Pagination behavior:

If items count is less than requested limit, this mean, no more items left

If request start item greater than items count, empty should be returned

If start item less then 0, empty results should be returned

Empty result message:

{

"response\_meta":{"status":"OK"},

"tokens": [],

"page\_start":"1",

"page\_limit":"6",

}