

ArtZero API and BackEnd

Git URL:

<https://github.com/ArtZero-io/az-backend>

IP:

18.138.185.86

Location: AWS Singapore

APIs	3
1. Ask Backend to update Collection Data	3
2. Ask Backend to update NFT Data	3
3. Get Collection Contract Address using by the Backend	3
4. Get total Collections in Database	3
5. Get Collections	3
6. Get Collection Information by ID	4
7. Get Collection Information by Collection Address	4
8. Get list of NFTs	4
9. Get NFT Information by token ID	4
10. Get Collections by Volume	5
11. Get Floor Price	5
12. Ask Backend to cache an Image	5
13. Get Cached Image	5
14. Get Collections by Collection Owner	6
15. Get NFTs by NFT Owner	6
16. Get NFTs by NFT Owner and Collection	6
17. Ask Backend to cache JSON	6
18. Get Cached JSON	6
19. Get NFTs by Collection Address	7
20. Ask Backend to update Bid Data	7
21. Get Bids from DB by Bidder Address	7
22. Count Number of Collections by Collection Owner	7
23. Get Featured Collections	8
24. Get listed NFTs	8
25. Get unlisted NFTs	8
26. Get new Listed NFT Events	8
27. Get new UnListed NFT Events	8
27. Get Purchase Events	9
28. Get Accepted Bid Events	9

29. Search Collections	9
30. Get Add Reward History	9
31. Get Reward Claim History	10

APIs

URL: <https://api.artzero.io/>

Code Location: **az_api.js**

Run as pm2 service in backend server

1. Ask Backend to update Collection Data

/updateCollection

Method: **POST**

Inputs:

collection_address: address of NFT Contract Address (Collection Address)

2. Ask Backend to update NFT Data

/updateNFT

Method: **POST**

Inputs:

collection_address: address of NFT Contract Address (Collection Address)

token_id: Token ID

3. Get Collection Contract Address using by the Backend

/getCollectionContract

Method: **GET**

Inputs:

N/A

4. Get total Collections in Database

/getCollectionCount

Method: **GET**

Inputs:

N/A

5. Get Collections

/getCollections

Method: **POST**

Inputs:

limit: how many collections to get

offset: where to start

sort: 1 = ASC and -1 = DESC

isActive: true/false

Example:

```
{limit: 6, offset:0,sort:-1,isActive:true}
```

trang 1: Lấy 6 collections, bắt đầu từ 0, sắp xếp theo thứ tự mới nhất

```
{limit: 6, offset:6,sort:-1,isActive:true}
```

trang 2: Lấy 6 collections, bắt đầu từ 6, sắp xếp theo thứ tự mới nhất

6. Get Collection Information by ID

/getCollectionByID

Method: **POST**

Inputs:

id: index của collection trong DB trùng với index trên contract

7. Get Collection Information by Collection Address

/getCollectionByAddress

Method: **POST**

Inputs:

collection_address: address of NFT Contract Address (Collection Address)

8. Get list of NFTs

/getNFTs

Method: **POST**

Inputs:

limit: how many collections to get

offset: where to start

sort: 1 = ASC and -1 = DESC

Example:

```
{limit: 6, offset:0,sort:-1}
```

trang 1: Lấy 6 collections, bắt đầu từ 0, sắp xếp theo thứ tự mới nhất

```
{limit: 6, offset:6,sort:-1}
```

trang 2: Lấy 6 collections, bắt đầu từ 6, sắp xếp theo thứ tự mới nhất

9. Get NFT Information by token ID

/getNFTByID

Method: **POST**

Inputs:

collection_address: address of NFT Contract Address (Collection Address)

token_id: Token ID

10. Get Collections by Volume

/getCollectionsByVolume

Method: **POST**

Inputs:

limit: how many collections to get

offset: where to start

sort: 1 = ASC and -1 = DESC

isActive: true/false

Example:

{limit: 6, offset:0,sort:-1}

trang 1: Lấy 6 collections, bắt đầu từ 0, sắp xếp theo volume nhỏ nhất

{limit: 6, offset:6,sort:-1}

trang 2: Lấy 6 collections, bắt đầu từ 6, sắp xếp theo volume lớn nhất

11. Get Floor Price

/getFloorPrice

Method: **POST**

Inputs:

collection_address: address of NFT Contract Address (Collection Address)

12. Ask Backend to cache an Image

/cacheImage

Method: **POST**

Inputs:

input: IPFS hash hoặc link hoặc URL tới image

is1024: có cache cỡ 1024 ko (true or false) default: false

13. Get Cached Image

/getImage

Method: **GET**

Inputs:

input: IPFS hash hoặc link hoặc URL tới image

size: image size 100, 500 or 1024

url: trả về url này khi ko lấy được image từ cached

14. Get Collections by Collection Owner

/getCollectionsByOwner

Method: **POST**

Inputs:

owner: address of Collection Owner

limit: how many collections to get

offset: where to start

sort: 1 = ASC and -1 = DESC

15. Get NFTs by NFT Owner

/getNFTsByOwner

Method: **POST**

Inputs:

owner: address of NFT Owner

limit: how many NFT to get

offset: where to start

sort: 1 = ASC and -1 = DESC

16. Get NFTs by NFT Owner and Collection

/getNFTsByOwnerAndCollection

Method: **POST**

Inputs:

owner: address of NFT Owner

collection_address: Collection Address

limit: how many NFT to get

offset: where to start

sort: 1 = ASC and -1 = DESC

17. Ask Backend to cache JSON

/cacheJSON

Method: **POST**

Inputs:

input: IPFS hash hoặc link hoặc URL tới JSON

18. Get Cached JSON

/getJSON

Method: **GET**

Inputs:

input: IPFS hash hoặc link hoặc URL tới image

19. Get NFTs by Collection Address

/getNFTsByCollectionAddress

Method: **POST**

Inputs:

collection_address: Collection Address

limit: how many NFT to get

offset: where to start

sort: 1 = ASC and -1 = DESC

20. Ask Backend to update Bid Data

/updateBids

Method: **POST**

Inputs:

collection_address: address of NFT Contract (Collection Address)

seller: address of Seller

token_id: Token ID

21. Get Bids from DB by Bidder Address

/getBidsByBidderAddress

Method: **POST**

Inputs:

bidder: Bidder Address

limit: how many bids to get

offset: where to start

sort: 1 = ASC and -1 = DESC

22. Count Number of Collections by Collection Owner

/countCollectionsByOwner

Method: **POST**

Inputs:

owner: address of Collection Owner

noNFT: include collection with No NFT. Default is false

23. Get Featured Collections

/getFeaturedCollections

Method: **GET**

Inputs:

N/A

24. Get listed NFTs

/getListedNFTs

Method: **POST**

Inputs:

limit: how many record to get

offset: where to start

sort: 1 = ASC and -1 = DESC

25. Get unlisted NFTs

/getUnlistedNFTs

Method: **POST**

Inputs:

limit: how many record to get

offset: where to start

sort: 1 = ASC and -1 = DESC

26. Get new Listed NFT Events

/getNewListEvents

Method: **POST**

Inputs:

collection_address: address of NFT Contract (Collection Address) blank for all

limit: how many record to get

offset: where to start

sort: 1 = ASC and -1 = DESC

27. Get new UnListed NFT Events

/getUnlistEvents

Method: **POST**

Inputs:

collection_address: address of NFT Contract (Collection Address) blank for all

limit: how many record to get

offset: where to start
sort: 1 = ASC and -1 = DESC

27. Get Purchase Events

/getPurchaseEvents

Method: **POST**

Inputs:

collection_address: address of NFT Contract (Collection Address) blank for all

limit: how many record to get

offset: where to start

sort: 1 = ASC and -1 = DESC

28. Get Accepted Bid Events

/getBidWinEvents

Method: **POST**

Inputs:

collection_address: address of NFT Contract (Collection Address) blank for all

limit: how many record to get

offset: where to start

sort: 1 = ASC and -1 = DESC

29. Search Collections

/searchCollections

Method: **POST**

Inputs:

keywords: what to search for

limit: how many record to get (default 15)

isActive: is Collection must be active? Default true

ignoreNoNFT: false if nft count must > 0, true include 0 nft count

30. Get Add Reward History

/getAddRewardHistory

Method: **POST**

Inputs:

limit: how many record to get

offset: where to start

sort: 1 = ASC and -1 = DESC

31. Get Reward Claim History

/getClaimRewardHistory

Method: **POST**

Inputs:

staker_address: staker address if not provided, show all

limit: how many record to get

offset: where to start

sort: 1 = ASC and -1 = DESC

32. Get Ownership History for an NFT

/getOwnershipHistory

Method: **POST**

Inputs:

owner: address of the owner

collection_address: address of NFT Contract

token_id: NFT Token ID