

---

# **OpenSea API Python wrapper documentation**

***Release 0.1.7***

**Attila Toth**

**Jan 04, 2023**



# CONTENTS

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>OpenSea NFT API Python 3 wrapper</b>          | <b>1</b>  |
| 1.1      | Supported endpoints . . . . .                    | 1         |
| 1.2      | Prerequisite . . . . .                           | 1         |
| 1.3      | Installation . . . . .                           | 2         |
| 1.4      | Usage examples . . . . .                         | 2         |
| 1.5      | Documentation . . . . .                          | 3         |
| <b>2</b> | <b>Usage examples</b>                            | <b>5</b>  |
| 2.1      | Get data about a single asset . . . . .          | 5         |
| 2.2      | Get data about a single asset contract . . . . . | 5         |
| 2.3      | Get data about a single collection . . . . .     | 5         |
| 2.4      | Get collection stats . . . . .                   | 6         |
| 2.5      | Get data about multiple assets . . . . .         | 6         |
| 2.6      | Get data about multiple collections . . . . .    | 6         |
| 2.7      | Get data about multiple events . . . . .         | 6         |
| 2.8      | Get data about multiple bundles . . . . .        | 7         |
| <b>3</b> | <b>Contributing</b>                              | <b>9</b>  |
| <b>4</b> | <b>History</b>                                   | <b>11</b> |
| 4.1      | 0.1.8 (2022-xx-xx) . . . . .                     | 11        |
| 4.2      | 0.1.7 (2022-03-26) . . . . .                     | 11        |
| 4.3      | 0.1.6 (2022-02-25) . . . . .                     | 11        |
| 4.4      | 0.1.5 (2022-02-17) . . . . .                     | 11        |
| 4.5      | 0.1.3 (2021-12-03) . . . . .                     | 12        |
| 4.6      | 0.1.0 (2021-11-07) . . . . .                     | 12        |



## OPENSEA NFT API PYTHON 3 WRAPPER

This an API wrapper library for the [OpenSea API](#) written in Python 3.

The library provides a simplified interface to fetch a diverse set of NFT data points from OpenSea.

### 1.1 Supported endpoints

The wrapper covers the following OpenSea API endpoints:

- Single asset ([/asset](#))
- Single asset contract ([/asset\\_contract](#))
- Single collection ([/collection](#))
- Collection stats ([/collection/{slug}/stats](#))
- Multiple assets] ([/assets](#))
- Multiple collections ([/collections](#))
- Multiple events ([/events](#))
- Multiple bundles ([/bundles](#))
- Asset listings ([/asset/{asset\\_contract\\_address}/{token\\_id}/listings](#))
- Asset offers ([/asset/{asset\\_contract\\_address}/{token\\_id}/offers](#))

### 1.2 Prerequisite

You need to have an **API key** to use the OpenSea API, and thus you need one to use this wrapper too. [You can request a key here](#). NOTE: The API key can take over 4-7 days to be delivered. It also requires you to show the project you are working on.

## 1.3 Installation

Install with pip:

```
virtualenv env && source env/bin/activate
pip install opensea-api
```

### 1.3.1 Upgrade

```
pip install opensea-api -U
```

## 1.4 Usage examples

```
# import the OpenseaAPI object from the opensea module
from opensea import OpenseaAPI

# create an object to interact with the Opensea API (need an api key)
api = OpenseaAPI(apikey="YOUR_APIKEY")

# fetch a single asset
contract_address = "0x495f947276749Ce646f68AC8c248420045cb7b5e"
token_id = "66406747123743156841746366950152533278033835913591691491127082341586364792833"
↪ "
result = api.asset(asset_contract_address=contract_address, token_id=token_id)

# fetch multiple assets
result = api.assets(owner="0xce90a7949bb78892f159f428d0dc23a8e3584d75", limit=3)

# fetch a single contract
result = api.contract(asset_contract_address="0x495f947276749Ce646f68AC8c248420045cb7b5e"
↪ ")

# fetch a single collection
result = api.collection(collection_slug="cryptopunks")

# fetch multiple collections
result = api.collections(asset_owner="0xce90a7949bb78892f159f428d0dc23a8e3584d75", ↪
↪ limit=3)

# fetch collection stats
result = api.collection_stats(collection_slug="cryptopunks")

# fetch multiple events
from opensea import utils as opensea_utils

period_end = opensea_utils.datetime_utc(2021, 11, 6, 14, 30)
result = api.events(
    occurred_before=period_end,
    limit=10,
```

(continues on next page)

(continued from previous page)

```
    export_file_name="events.json",
)

# fetch multiple bundles
result = api.bundles(limit=2)

# paginate the events endpoint (cursors are handled internally)
from datetime import datetime, timezone

start_at = datetime(2021, 10, 5, 3, 25, tzinfo=timezone.utc)
finish_at = datetime(2021, 10, 5, 3, 20, tzinfo=timezone.utc)

event_generator = api.events_backfill(start=start_at,
                                     until=finish_at,
                                     event_type="successful")

for event in event_generator:
    if event is not None:
        print(event) # or do other things with the event data
```

Here's a demo video showcasing the basics.

## 1.5 Documentation

- [Wrapper documentation](#)
- [OpenSea API documentation](#)





## USAGE EXAMPLES

## 2.1 Get data about a single asset

```
from opensea import OpenseaAPI

api = OpenseaAPI(apikey="<APIKEY>")
result = api.asset(asset_contract_address="0x495f947276749Ce646f68AC8c248420045cb7b5e",
                  token_id=
↳ "66406747123743156841746366950152533278033835913591691491127082341586364792833")
print(result)
```

## 2.2 Get data about a single asset contract

```
from opensea import OpenseaAPI

api = OpenseaAPI(apikey="<APIKEY>")
result = api.contract(asset_contract_address="0x495f947276749Ce646f68AC8c248420045cb7b5e
↳ ")
print(result)
```

## 2.3 Get data about a single collection

```
from opensea import OpenseaAPI

api = OpenseaAPI(apikey="<APIKEY>")
result = api.collection(collection_slug="cryptopunks")
print(result)
```

## 2.4 Get collection stats

```
from opensea import OpenseaAPI

api = OpenseaAPI(apikey="<APIKEY>")
result = api.collection_stats(collection_slug="cryptopunks")
print(result)
```

## 2.5 Get data about multiple assets

This example fetches three NFTs that Snoop Dogg owns:

```
from opensea import OpenseaAPI

api = OpenseaAPI(apikey="<APIKEY>")
result = api.assets(owner="0xce90a7949bb78892f159f428d0dc23a8e3584d75",
                    limit=3)
print(result)
```

## 2.6 Get data about multiple collections

This example creates a JSON file with 3 collections where Snoop Dogg is an owner:

```
from opensea import OpenseaAPI

api = OpenseaAPI(apikey="<APIKEY>")
result = api.collections(asset_owner="0xce90a7949bb78892f159f428d0dc23a8e3584d75",
                        limit=3,
                        export_file_name='snoop_collections.json')
print(result)
```

## 2.7 Get data about multiple events

This example creates a JSON file with 10 events that happened between the defined time period (UTC timezone) between 2021-11-06 14:25 and 2021-11-06 14:30

```
from opensea import OpenseaAPI
from opensea import utils

api = OpenseaAPI(apikey="<APIKEY>")
period_start = utils.datetime_utc(2021, 11, 6, 14, 25)
period_end = utils.datetime_utc(2021, 11, 6, 14, 30)
result = api.events(occurred_after=period_start,
                  occurred_before=period_end,
                  limit=10,
                  export_file_name='events.json')
print(result)
```

## 2.8 Get data about multiple bundles

```
from opensea import OpenseaAPI

api = OpenseaAPI(apikey="<APIKEY>")
result = api.bundles(limit=3)
print(result)
```



## CONTRIBUTING

- [Submit an issue](#) to report a bug or suggest a new feature
- Submit a PR with a bugfix or a new feature.
- Write a blog post or a tutorial about how you are using the OpenSea API and this wrapper



## HISTORY

### 4.1 0.1.8 (2022-xx-xx)

- Add `include_orders` in the assets api
- Add cursor-based pagination in `assets` endpoint

### 4.2 0.1.7 (2022-03-26)

- Add support for `asset listings` and `asset offers` endpoints
- Add `occured_after` and `collection_editor` arguments to events endpoint
- Handle SSL error when making requests
- Docs: add example to paginate the events endpoint (using `events_backfill()`)

### 4.3 0.1.6 (2022-02-25)

- Fix `/events` endpoint pagination (`events_backfill()` function) by passing only *the cursor hash* and not the full URL to the next request.

### 4.4 0.1.5 (2022-02-17)

- Ability to override `base_url` with any other URL
- Support for cursor-based pagination for `/events` endpoint (and removed deprecated arguments)
- New function to help paginate the `/events` endpoint
- Introducing a temporary function to fix the `next url` problem until OpenSea addresses this issue
- Minor docs updates and cleanup

## 4.5 0.1.3 (2021-12-03)

- Ability to reach all endpoints from one `OpenseaAPI` object
- API key support (Opensea requires it from now on)

## 4.6 0.1.0 (2021-11-07)

- First release on PyPI.