

OpenSea API

- [Overview](#)
- [Requesting API keys](#)
- [Analytics Endpoints](#)
- - [Get Collection Stats](#) `get`
- - [Get Events](#) `get`
- - [Get Events \(by account\)](#) `get`
- - [Get Events \(by collection\)](#) `get`
- - [Get Events \(by NFT\)](#) `get`
- [Embedded Wallet Endpoints](#)
- - [Get Embedded Wallet Address](#) `get`
- [NFT Endpoints](#)
- - [Get Account](#) `get`
- - [Get Collection](#) `get`
- - [Get Collections](#) `get`
- - [Get Contract](#) `get`
- - [Get NFT](#) `get`
- - [Get NFTs \(by account\)](#) `get`
- - [Get NFTs \(by collection\)](#) `get`
- - [Get NFTs \(by contract\)](#) `get`
- - [Get Payment Token](#) `get`
- - [Get Traits](#) `get`
- - [Refresh NFT Metadata](#) `post`
- [OpenAPI Definition](#)
- [OpenSea Marketplace Endpoints](#)
- - [Build Criteria Offer](#) `post`
- - [Create Criteria Offer](#) `post`
- - [Create Item Offer](#) `post`
- - [Create Listing](#) `post`
- - [Fulfill Listing](#) `post`
- - [Fulfill Offer](#) `post`
- - [Get All Listings \(by collection\)](#) `get`
- - [Get All Offers \(by collection\)](#) `get`
- - [Get Best Listing \(by NFT\)](#) `get`
- - [Get Best Listings \(by collection\)](#) `get`
- - [Get Best Offer \(by NFT\)](#) `get`
- - [Get Collection Offers](#) `get`
-

- [Get Item Offers.get](#)
-
- [Get Listings.get](#)
-
- [Get Order.get](#)
-
- [Get Trait Offers.get](#)

OpenSea Stream API

- [Stream API Overview](#)
- [Stream API Event Example Payloads](#)
- [Using Stream API without SDK](#)

Other

- [Supported Chains](#)

Using Stream API without SDK

Developers do not have to directly rely on the SDK to subscribe to the Stream API. Any language with a websocket client can directly subscribe to the Stream API and receive updates. The developer will need the following to subscribe:

- Base Endpoint:* Mainnet:
 - `wss://stream.openseabeta.com/socket/websocket`
-
- Testnet:
 - `wss://testnets-stream.openseabeta.com/socket/websocket`
- API Key
- Collection they'd like to subscribe to* Can use

-
- - to subscribe to all
 - Event types to subscribe to* `item_metadata_updated`
 -
 - `item_listed`
 -
 - `item_sold`
 -
 - `item_transferred`
 -
 - `item_received_offer`
 -
 - `item_received_bid`
 -
 - `item_cancelled`

• Step 1. Connecting

-
- Construct your full connection string as:
 - **`{base_endpoint}?token={api_key}`**
 - For example:
 - `wss://testnets-stream.openseabeta.com/socket/websocket?token=`
 - And set up your client to connect to this endpoint. You should ping the Server every 30 seconds with a heartbeat message that should look as follows:JSON
- {
 - "topic": "phoenix",
 - "event": "heartbeat",
 - "payload": {},
 - "ref": 0
- }

. Step 2: Subscribing to a collection

-
- Figure out the slug of the collection you'd like to join and the event type you'd like to subscribe to. One current limitation is that you will only be able to subscribe to all events for this collection, and will therefore need to filter client-side as needed on event_types.
- Send the following message to subscribe to a collection:JSON
- {
- "topic": "collection:boredapeyachtclub",
- "event": "phx_join",
- "payload": {},
- "ref": 0
- }
- Keep track of the
- ref
- number used here, as it can be used later when unsubscribing

. Step 3: Unsubscribing from a collection

-
- To unsubscribe, you'll need to just change the
- event
- field to
- phx_leave
- and pass in the same
- ref
- value used earlier.
- ref
- allows you to use multiple concurrent connection to different Collections as needed. You can also simply close your websocket client's connection to disconnect.

. Example: Firecamp

-
- [Firecamp](#)
- is a popular API GUI client that supports making websocket requests. We've prepared a ready-to-use example in testnets that can be imported directly to your Firecamp app to see how we are able to connect to the Stream API and make requests. It also automatically pings for a heartbeat every 30 seconds to ensure your client remains connected.
- [Stream API Example_firecamp.json](#)
- Simply import this into your Firecamp UI to get a working, ready to use example of connecting to the Stream API Websocket
- [Source for websocket connecting](#)