API Introduction

StockX provides a simple REST API platform to authenticate, make requests, and retrieve data with JSON formatted responses for catalog searching, selling, and order management functions. This document provides information to developers on how to integrate with the StockX API.

Accessing the API

To access StockX's developer platform, a simple authenticated request might look like:
https://api.stockx.com/v2/selling/listings

APIs Offered

StockX offers APIs for three general functions
- Catalog Search
- Listing on the StockX marketplace
- Order Management

StockX APIs support basic end-to-end selling activities. This includes searching our catalog of products, placing and managing listings that turn into asks (an offer to sell), and getting your order/active sales information.

Common uses for these APIs
- Placing and updating listings at a high volume.
- Executing sales quickly based on market movement.
- Assisting with inventory management - updating inventory levels and managing cross-platform sales.
- Using order and shipping information to ensure fulfillment of orders to StockX.
- Integrating with in-house systems/other software for finance, shipping, inventory, or other purposes.

What are Listings?

A listing is a feature rich wrapper over what most StockX sellers know as an ask. Note that listings have a 1:1 relationship with asks. When a seller applies a change to a listing, under the hood StockX manages that ask on your behalf.
Sellers benefit from using the Listing APIs as they are interfacing with a high scale and low latency environment. Going forward, listings will be the home to any new seller related features when it comes to managing their active/inactive inventory in our marketplace.

How it Works

The Listing API is entirely **_asynchronous_**. When a seller makes a write request to the API, we will synchronously validate the input and return a corresponding operation. API partners can use operations to track the progress of submitted changes. Note that listings have a 1:n relationship with operations.
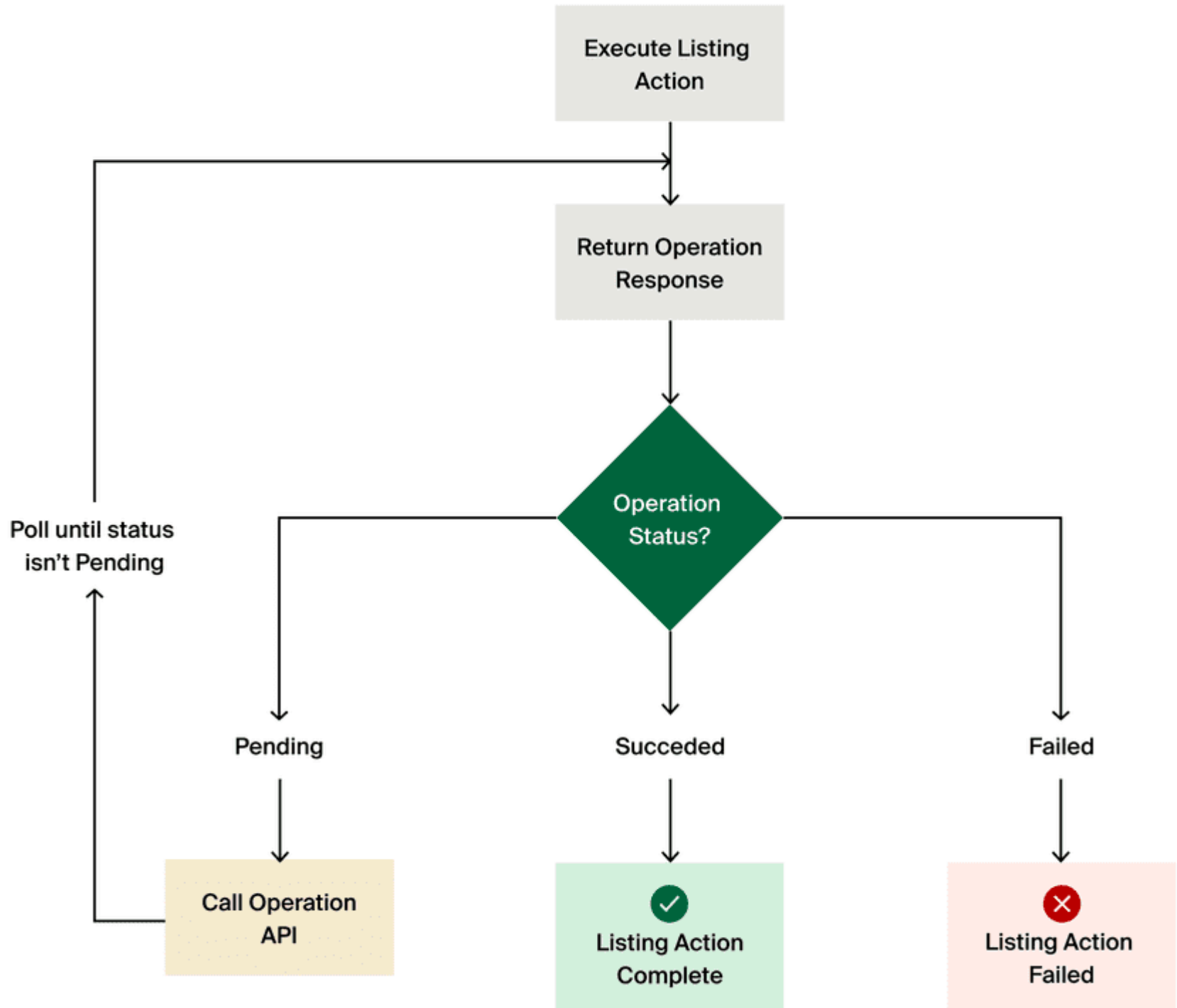
Operation

```
{
    "listingId": "1384fba7-abde-4986-b06c-2e9887ff0f33",
    "operationId": "218720db-afc9-41d6-9a76-d3993e2de8fb",
    "operationType": "ACTIVATE",
    "operationStatus": "PENDING",
    "operationUrl": "https://api.stockx.com/v2/selling/listings/1384fba7-abde-4986-b06c-2e98
    "operationInitiatedBy": "USER",
    "createdAt": "2022-10-19T20:27:55.323Z",
    "updatedAt": "2022-10-19T20:27:55.323Z",
    "error": null,
    "changes": {
        "additions": {},
        "updates": {
            "updatedAt": "2022-10-19T20:27:55.323Z"
        },
        "removals": {}
```

```
    }
  }
```

Sellers can check the progress of an inflight operation by polling the returned `operationUrl` and checking the `operationStatus`. The status will be reflected with one of three values, `PENDING`, `SUCCEEDED`, `FAILED`. `PENDING` means the request is still being processed. `SUCCEEDED` means the operation was accepted and is reflected in the StockX marketplace. `FAILED` means the operation was failed for various reasons. When the operation has failed, check the `error` attribute to get more details.

Operations are also a great reference point to remind sellers on what attributes they updated as a part of the request.

There are times when under the hood StockX will internally make a change to a listing and this will be communicated by looking at the `operationInitiatedBy` attribute. This attribute will be reflected with one of two values, USER or SYSTEM.

StockX will never update the amount of a listing.

Listing Action	Successful Outcomes
Create Inactive Listing	Listing created successfully, no ask created on StockX
Create Active Listing	Listing created successfully, the ask is live on StockX
Update Listing	Listing updated successfully, the changes are live on StockX
Delete Listing	Listing deleted successfully, the ask is no longer live on StockX
Activate Listing	Listing activated successfully, the ask is live on StockX
Deactivate Listing	Listing deactivated successfully, the ask is no longer live on StockX

Authentication

The StockX API requires an API key that is generated automatically after your StockX developer access is approved. The key identifies your application and is used to track call usage. The API key needs to be added to your request (as a request header) that will allow your application to make calls to StockX APIs.

In addition, the StockX API uses the OAuth 2.0 authorization framework for application and user authorization. OAuth is the industry standard to provide client applications with secure access. A valid access token is required for each request you make to the StockX APIs.

Rate Limiting

Clients are allowed a maximum quota of 25,000 requests per 24-hour period, with a limit of 1 request per second. All StockX APIs are governed by these rate limits. The daily quota limits reset at 12 AM UTC. When a rate limit has been reached, a return response code of 429 is returned. To maximize the use of the API use batch APIs where available.

Batch limits are throttled based on the number of items within a request. There is a 5-minute gap between 500 batch calls. If used at maximum we allow 500 items every 5 minutes with a 50,000 item limit per day.

These rate limits ensure that StockX can maintain the integrity of our systems and provide an optimal quality of service (QoS), If you find that these rate limits are inadequate for your use case, you can request an increase in the rate limits by reaching out to our developer support team at developersupport@stockx.com. StockX will evaluate each rate limit request on a case-by-case basis. StockX reserves the right to accept or deny the rate limit request.

API Requests and URLs

The StockX API uses REST as its calling style which works with standard HTTP calls.

The StockX API resources are identified by a unique URL, for example:

`https://api.stockx.com/v2/selling/batch/create-listing/:batchId`

V2 corresponds with the version of the API and must be present. The API version will change when updates are made that will break compatibility with existing applications.

Formats

StockX only supports outputs in JSON format. All requests and responses to the StockX API will be in JSON.

HTTP Methods

StockX RESTful APIs use standard HTTP methods to represent actions against a resource:
- GET - Reads a resource. HTTP 200 on success.
- POST - Creates a new resource. HTTP 201 on success.
- PATCH - Updates a resource. HTTP 200 on success.
- PUT - Updates a resource. HTTP 200 on success.
- DELETE - Deletes a resource. HTTP 200 on success.

POST will return 201 only if it's a true create scenario. POST will return a 200 when it's not truly creating a resource

Response Codes

StockX uses conventional HTTP response codes to indicate the success or failure of an API request. In general, codes in the 2xx range indicate success, codes in the 4xx range indicate an error that resulted from the provided information (e.g., a required parameter was missing, invalid syntax, etc.), and codes in the 5xx range indicate an error with StockX servers.

Code	Description
200	Success
201	Created
204	Success - No content to return.

Code	Description
400	Bad Request
401	Unauthorized request
403	Forbidden
404	Not Found - The resource doesn't exist.
413	Request too large
415	Unsupported media type
429	The request was not accepted because the application exceeded the rate limit.
500	Internal Server Error
503	Service Unavailable
504	Gateway Timeout - The request could not complete in time.

Get Started

Authentication

API Introduction

API Reference

Versioning and Support

FAQs

Contact Us

Terms of Use