Swift+ APIs User Guide

Seamless Integration of Data from Magellan Directly into Your Systems

Overview

Welcome to Swift+ Open APIs. This document provides essential information for integrating and utilizing Swift APIs, including onboarding details for new API customers.

Thank you for your partnership. If you have any questions, please contact your ONEOK - Magellan representative.

Table of Contents

Overview ***2***

Customer Onboarding ***3***

Getting a Token ***3***

Token URLs **3**

Request Headers **3**

Request Body **3**

Client Id 3

Client Secret 4

Scope 4

Grant Type 4

Putting it all Together ***4***

CURL 5

Postman ***5***

Making an API Call ***7***

API Rate Limitation ***8***

API Versioning ***8***

Transaction Types: ***9***

Wrapping Up ***11***

Customer Onboarding

If you have been able to login, you likely have already created an account on the Swift+ Partners Developer Portal. If you need your own login, please do so at Swift+ Developer Portal - https://developers.magellanlp.com/. You will need to provide an email address with a Swift company domain.

Please note, the account creation process may take several days due to technical and security protocols. Plan accordingly to meet your development timelines.

Getting a Token

To call Swift+ APIs, you'll need a JSON Web Token (JWT) from ONEOK - Magellan. This token includes the necessary information to access and authorize your usage. To get a token, you'll need four pieces of information:

- The Token URL
- Your Client Id and Client Secret
- An API Scope
- The Grant Type

The rest of this section will describe how to use these four pieces in a token request.

Token URLs

In order to get a token for an API call, you'll first need to make a POST request to our token provider. The URL for the provider is as follows:
https://login.microsoftonline.com/d8088b60-8534-41bd-a682-1ad5ee71069a/oauth2/v2.0/token
The URL alone is not enough to get a token. Read on to understand what headers and body parameters you'll also need to provide.

Request Headers

The only header you need to include is a content-type header. Please use the following:

- *Content-Type: application/x-www-form-urlencoded*

Request Body

There are four parameters you'll need to include in the body of the request to get a valid token.

Client Id

- *client_id={ClientId}*

Replace {ClientId} with your company-specific client id. This id should have been provided to you by ONEOK - Magellan. If not, please contact your Swift+ representative.

Client Secret

- *client_secret={ClientSecret}*

Replace {ClientSecret} with your company-specific client secret provided by ONEOK - Magellan. If you haven't received it, please contact your Swift+ representative for assistance.

Scope

- *scope={ApiScope}*

Each APIs requires a unique token. The scope parameter is necessary to obtain a token for the specific API you intend to call. The currently available scopes are:

MOCK
api://swift-mock-inventory-api/.default
api://swift-mock-loading-api/.default
api://swift-mock-shipping-api/.default
api://swift-mock-billing-api/.default
api://swift-mock-locations-api/.default

OPEN
api://swift-open-inventory-api/.default
api://swift-open-loading-api/.default
api://swift-open-shipping-api/.default
api://swift-open-billing-api/.default
api://swift-open-locations-api/.default

Replace {ApiScope} with the scope for the desired API.

Grant Type

- *grant_type=client_credentials*

No changes are needed here. Including this indicates that the Client Credentials flow is being used in your token request.

Putting it all Together

To get a token, make a POST request to the token URL with the specified header and body parameters. The following examples illustrate common methods. Skip to the next section if you already know how to make this POST.

CURL

curl is a versatile command-line tool for transferring data with URLs. Below is an example of a token request using **curl**:

curl -H "Content-Type: application/x-www-form-urlencoded" -d "client_id=12345678-1234-1234-1234-123456789123&client_secret=123456789&scope=api://swift-open-inventory-api/.default&grant_type=client_credentials" -X POST https://login.microsoftonline.com/d8088b60-8534-41bd-a682-1ad5ee71069a/oauth2/v2.0/token

This request leverages three options: -H, -d, and -X.

-H specifies the header values.
-d specifies the data or body values.
-X specifies the request type.

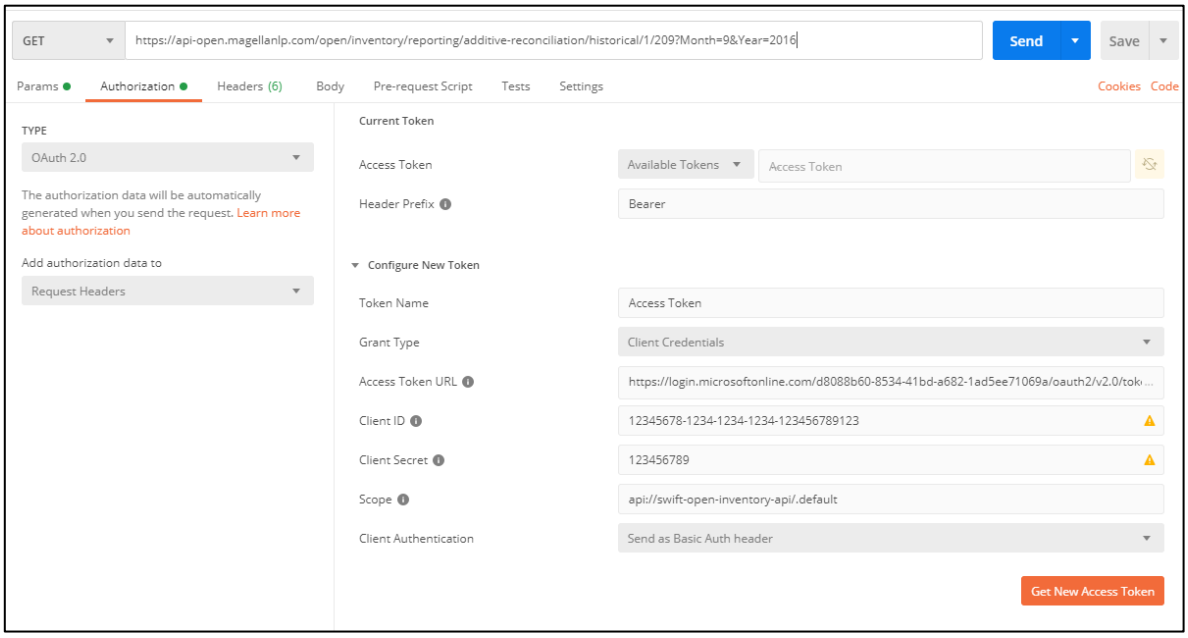
Replace the client_id and client_secret with your company's unique values.
You can read more about curl on their website: https://curl.se/.

Postman

Postman is a widely-used API development tool, particularly valued for its built-in support for OAuth 2.0 authorization.

To use OAuth 2.0 in Postman, select the Authorization tab in your request and set the type to OAuth 2.0. Postman will then provide a form to fill in the necessary fields to obtain a token.

Ensure you replace client_id and client_secret with your company's unique values.



After entering these values, click "Get New Access Token." If successful, a dialog box will confirm the success.

Click Proceed. Postman will then display the token.

Note: By clicking "Use Token" on this screen, Postman will automatically populate the Access Token field for the request, allowing you to simply click "Send."

Postman has some more detailed information on their website: https://learning.postman.com/docs/sending-requests/authorization/.

Making an API Call

With a valid token and API endpoint, the only additional information required is a subscription key. Obtain this by visiting your profile on our developer website: https://developers.magellanlp.com/profile.

For each subscription, you will find Primary and Secondary key fields. Click "show" to reveal one of these keys and use it as your subscription key.

You are now ready to make a call. A valid API call consists of an HTTP GET request to your chosen endpoint with two header values: the subscription key and the authorization token.

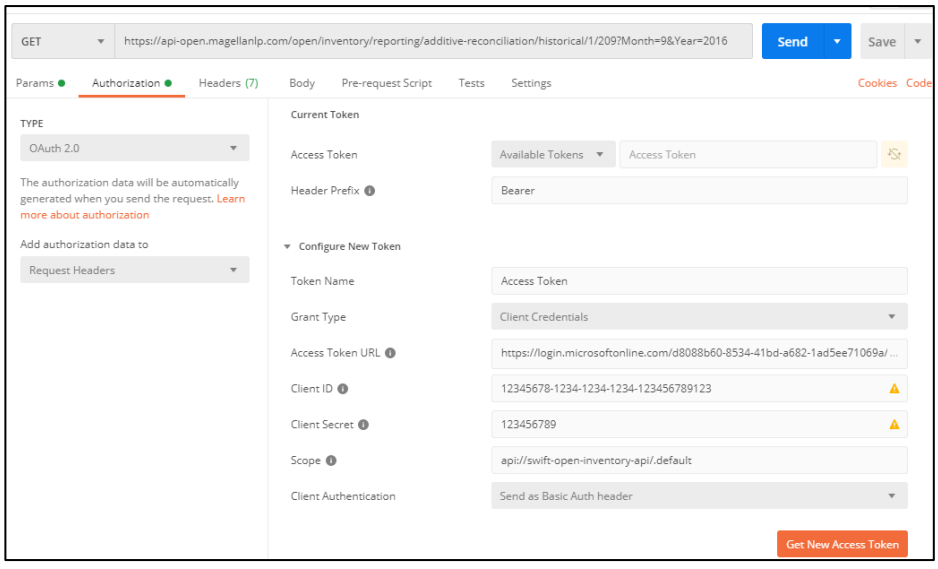
These headers are named "Ocp-Apim-Subscription-Key" for the subscription key and "Authorization" for the authorization token.

For the authorization token, include the word "Bearer" followed by a space before your token.

You can use any software capable of making HTTP GET requests to make API calls. Below is a template for a valid API request using curl:

- *curl -X GET https://api-open.magellanlp.com/open/inventory/reporting/additive-reconciliation/historical/1/209 -H "Authorization: bearer your-token-here" -H "Ocp-Apim-Subscription-Key: your-subscription-key-here"*

Below is an example of a request using Postman's built-in authorization feature. Ensure you replace the client information with your company's unique values and add the subscription key in the request headers.



As previously mentioned, any software capable of making HTTP GET requests can be used to make API calls. Choose the method that best suits your company's needs.

API Rate Limitation

The primary purpose of API rate limiting is to protect server infrastructure and ensure fair usage of API resources among all clients.

A "call" represents a single API request made by a client associated with a particular key, with each call consuming a portion of the rate limit quota assigned to that key.

The "Renewal Period" defines the duration after which the rate limit quota is reset or replenished for a specific key.

Currently the rate limit for the API is set as:

- Call: 10
- Renewal Period: 60 seconds

If the key has no remaining quota, the API may respond with an error such as "429 Too Many Requests", indicating that the rate limit has been exceeded.

API Versioning

The primary purpose of API versioning is to enable the adoption of new versions at your convenience, without causing disruptions to your current production applications. Versioning facilitates smooth transitions between versions, thereby enhancing your Swift API experience.

Swift API versioning is implemented using Semantic Versioning (SemVer). For more information on semantic versioning, please refer to Semantic Versioning 2.0.0. SemVer is beneficial for tracking backward-compatible changes, new functionalities, and bug fixes. Major breaking changes are indicated by incrementing the major version number, while backward-compatible additions and bug fixes are reflected in minor version numbers.

Additionally, Swift APIs utilize URI versioning, incorporating version numbers into the URL path.

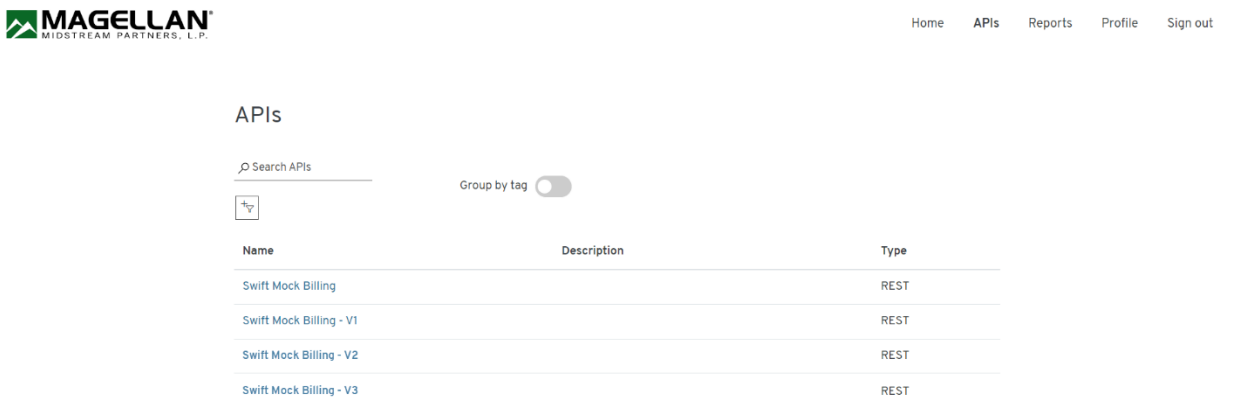
Example of Original Format

- *Https://api-mock.qa.magellanlp.com/mock/billing/invoices/additive/{Parameter1}/{Parameter2}/{Parameter3}*

Example of Version 1 Format

- *Https://api-mock.qa.magellanlp.com/mock/billing/V1/invoices/additive/{Parameter1}/{Parameter2}/{Parameter3} [?api-version]*

Example of how APIs will be presented in the Developer Portal



Transaction Types:

- BOL: BILL OF LADING
  - **Bill of Lading** is a legally binding document that serves as a receipt and contract for fuel loaded or unloaded, detailing product type, volumes, destinations or source, carrier name, and all regulatory compliance language. It facilitates inventory tracking, invoicing, and ownership transfer.
- RCN: RECONSIGNMENT
  - **Reconsignment** allow Inventory Owners to move their refined fuels from one location to another in the ONEOK refined fuel system. It also allows Inventory Owners to manage inventory volumes and meet their supplier's demands at specific locations.
- STATCHANGE: STATUS CHANGE
- RCP: RECEIPT
  - An **RCP** is a receipt is used to verify and document receipt of refined fuel products. It ensures accurate tracking of inbound shipments, confirming product type, volumes, and origin.

Receipts play a crucial role in inventory management, regulatory compliance, and operational efficiency, helping shippers and terminal operators maintain precise records of fuel movements.

- GT: GRADE TRANSFER
  - ***Grade transfer*** *refers to the movement of fuel from one product grade to another within storage or distribution networks. This can involve reclassifying inventory, blending fuels to meet specifications, or shifting product between tanks to optimize supply.*
- PTO-TO: PRODUCT TRANSFER ORDER - TO
  - **Product Transfer Order (PTO) - TO** refers to the process of transferring ownership of fuel barrels **to** another inventory owner at a specific location within the system. It ensures accurate tracking of product movements and facilitates seamless transactions between parties.
- PTO-FROM: PRODUCT TRANSFER ORDER - FROM
  - **Product Transfer Order (PTO) - FROM** refers to the process of transferring ownership of fuel barrels **from** one inventory owner to another at a specific location within the system. It ensures accurate tracking of product movements and facilitates seamless transactions between parties.
- ADJ: ADJUSTMENT
  - **Adjustment** refers to a modification in inventory records due to discrepancies, reclassifications, or corrections in product volumes. Adjustments can occur for various reasons, such as measurement variances, accounting reconciliations, or operational changes.
- OFL: OFFLINE
  - ***Offline*** *transactions originate or terminate at a location that is not a ONEOK owned and operated facility, like an airport or another pipeline provider.*
- TND: TENDER DEDUCTION - TIME BASED
  - **Tender Deduction – Time Based** refers to a volume deduction applied to product shipments based on the time the product remains in the system. This deduction accounts for **inherent losses** such as evaporation, shrinkage, or interface mixing that can occur over time during transport.
- SHP: SHIPMENT
  - A **shipment transaction** is a movement refined or crude product between an origin and a destination point.
- DLV: DELIVERY
  - A **delivery** is the physical transfer of product from ONEOK's system to a shipper's designated destination point
- STL: SETTLEMENT
  - A **Settlement** is the process of reconciling the actual volume and quality of product delivered versus what was tendered or scheduled.

- TND-AUTO: TENDER DEDUCTION - AUTOMATI
    - **Tender Deduction – Automatic** is a fixed volume deduction applied automatically to all products tendered for transportation.

      A tender deduction of one-tenth of one percent (0.1%) by volume is made on the products transported from all Origins.
- TND-MANUAL: TENDER DEDUCTION - MANUAL
    - **Tender Deduction – Manual** is a "manual" tender deduction that refers to a non-standard or route-specific volume deduction that's explicitly defined in a particular tariff items — rather than applied automatically across all shipments.
- SET OFF: STOF
    - **Set-off** is a financial or volumetric adjustment where one party's obligation is offset by a corresponding credit or claim—like deducting over-delivered volumes from a future shipment or reconciling charges across accounts.

Wrapping Up

At this point, you should have all the necessary information to start calling and utilizing the Swift+ APIs, making them an asset. If you have any questions, please contact your ONEOK - Magellan representative.

Thank you again for your patronage and happy coding!

The Swift+ API Team
https://developers.magellanlp.com
CTTeam@magellanlp.com