

# Managely API Quick Start

12/29/2023 4:41 pm EST

## Introduction

The Managely API is built upon well-established web standards and is straightforward to start using. Requests are issued using GraphQL queries and mutations, and responses are delivered in JSON. Authentication is handled using a simple OAUTH2 password flow. If you are unfamiliar with any of these technologies, you can learn more at the links below. The rest of this document assumes familiarity with these technologies.

### GraphQL

<https://graphql.org/learn/>

### JSON

[https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp)

### OAUTH2

<https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2>

## Necessary Tools

As the API is built using a REST-based protocol, any tool that performs REST requests will work. However, some tools have GraphQL support built-in and make discovery easier. Managely has one of these tools built in, and you can access it directly in your web browser. This is the tool that used for the rest of this document, but if you would like to consider an external tool, look at Insomnia (<https://insomnia.rest/>).

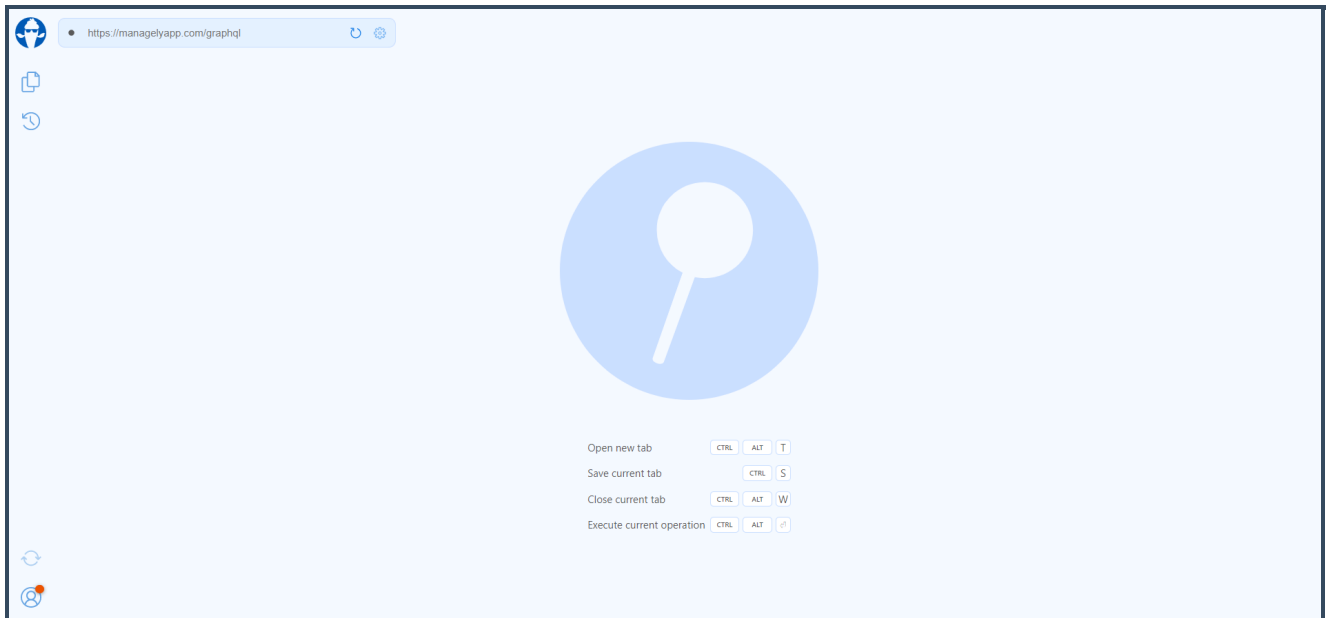
To integrate the API with your own code, any library that can make REST requests, along with a library that can handle OAUTH2 will work. For a GraphQL specific client library, Apollo (<https://www.apollographql.com/apollo-client>) is a popular choice.

## Exploring the API

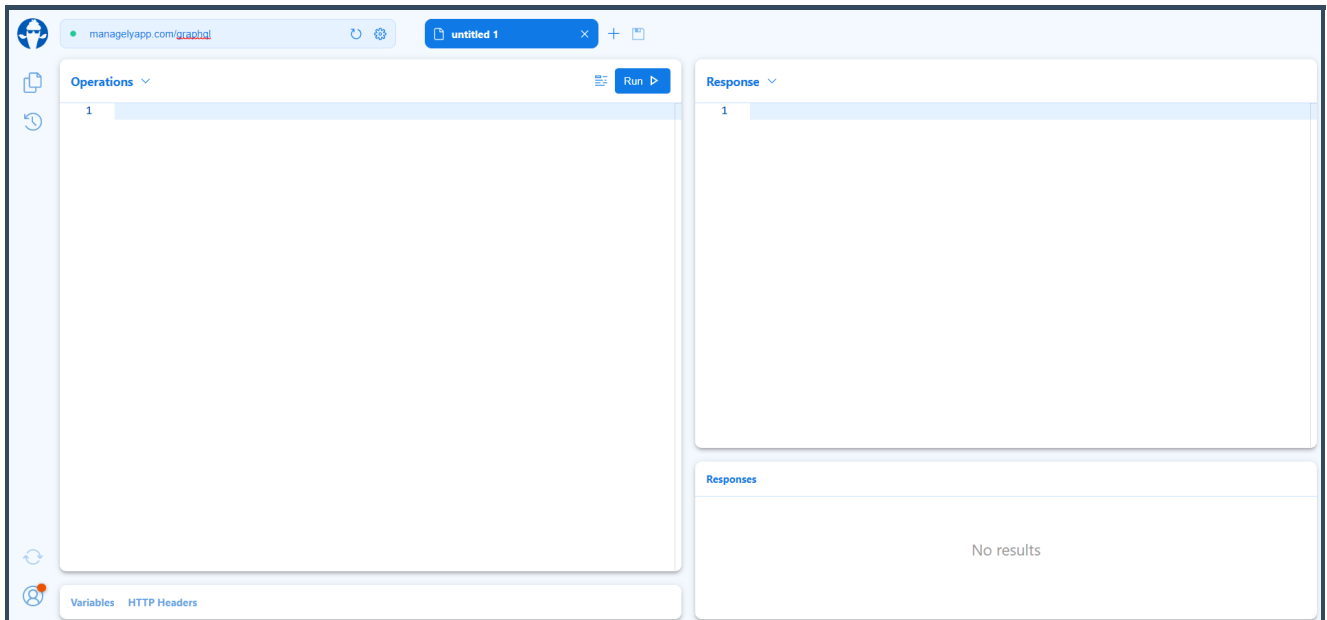
### Getting set up and signed in

Use the instructions in this section to explore the API.

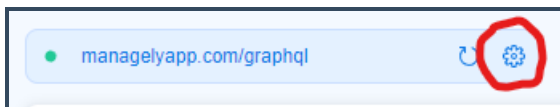
1. Browse to the Managely API endpoint at <https://managelyapp.com/graphql/> in a web browser. A screen opens that looks like this:



2. Open a new query tab by pressing the 'Open new tab' key combination listed. This varies depending on your operating system. For Windows this is CTRL-ALT-T; for Mac this is Command-Option-T. Use this window to send queries or mutations.



3. Configure your credentials to log into the server. Click the gear icon next to the API URL:



This opens the Connection Settings dialog:

## Connection Settings ✕

General
Authorization
HTTP Headers

**Type**

▼
OAuth 2

**Grant Type**

▼
Resource Owner Password Credentials

**Username**

👤

**Password**

🔒

**Access Token URL**

https://managelyapp.com/connect/token

Apply
Cancel

4. Click the Authorization tab, and complete these fields:

Field	Type this text or select this:
Type	Select <i>OAuth2</i>
Grant Type	Select <i>Resource Owner Password Credentials</i>
Username	<Your username>
Password	<Your password>
Access Token URL	https://managelyapp.com/connect/token
Client ID	Type <i>managely</i>
Credentials	Select <i>As Basic Auth Header (default)</i>

5. At the bottom of the dialog, click the Fetch Token button.

A screenshot of a web interface showing a text input field labeled "Audience" with a light blue border. Below the input field is a blue button with the text "Fetch Token".

If you set it up correctly, your Access Token and Refresh Token appear:

A screenshot of a web interface showing two text input fields. The top field is labeled "Access Token" and contains a long alphanumeric string. The bottom field is labeled "Refresh Token" and also contains a long alphanumeric string. Below the "Refresh Token" field are three blue buttons: "Clear Token", "Refresh Token", and "Fetch Token".

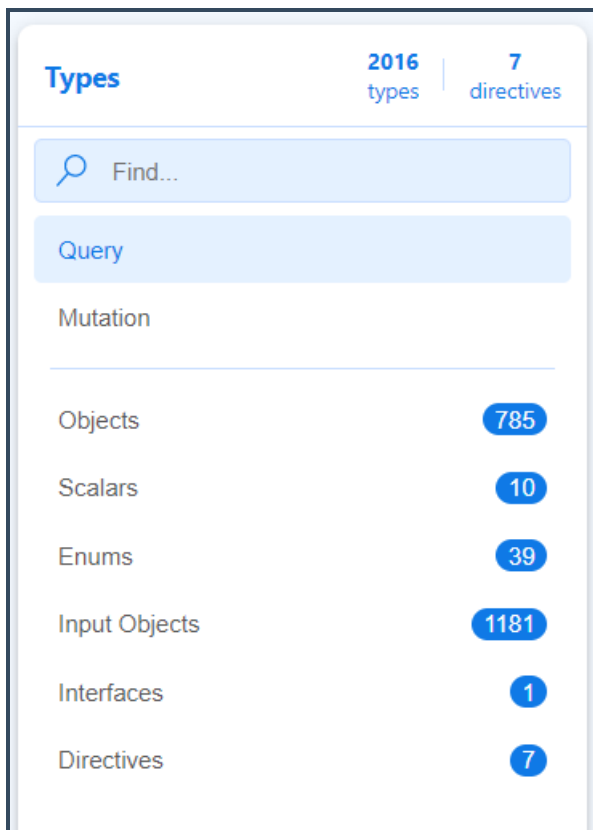
6. Click Apply to close the dialog.

## Looking at the API definition

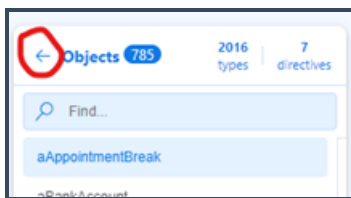
In the upper-left corner of the first panel, there is a drop-down menu labeled Operations.

A screenshot of a drop-down menu with the text "Operations" in blue and a small downward-pointing chevron icon to its right.

In Operations mode, you can issue queries or mutations to the API. However, if you want to look at the API definition, change this to Schema Reference, which shows a list of queries. You start in the Query object, but you can use the Types panel on the right to switch to the Mutation object.



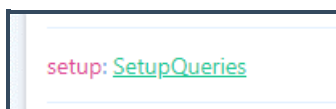
Note that once you drill into an object on the left, it is not possible to jump back out. You need to use the Types panel on the right to navigate. If you get lost, click the Back Arrow, and then click Query to get back to the Query object.



## Making a test API request

Use the API to get some information about the current user.

1. In the Types panel on the right, click Query. This shows a list of queries in the left panel.
2. Find setup in the list, and drill into SetupQueries:



Schema Reference ▾

**SetupQueries** No description  
Kind of type: Object

Fields

accountingPeriod: <a href="#">AccountingPeriodQueries</a>	No description
accountingPeriodSetup: <a href="#">AccountingPeriodSetupQueries</a>	No description
agingBuckets: <a href="#">AgingBucketsQueries</a>	No description
apiLog: <a href="#">ApiLogQueries</a>	No description
attachment: <a href="#">AttachmentQueries</a>	No description
authority: <a href="#">AuthorityQueries</a>	No description
batchAutomation: <a href="#">BatchAutomationQueries</a>	No description
billingCycle: <a href="#">BillingCycleQueries</a>	No description

3. Find userManager in the list and drill into UserManagerQueries:

userManager: [UserManagerQueries](#)

Schema Reference ▾

**UserManagerQueries** Queries to get the users of the signed-in dealer.  
Kind of type: Object

Fields

allDealerAndHostUserProfiles: <a href="#">[UserProfileModel]</a>	Gets the set of available dealer/host profiles for the currently signed-in dealer.
availableProductPortalsWithClients(...): <a href="#">[CheckboxViewModel]</a>	Gets the list of available portals. where: <a href="#">[CheckboxViewModelFilterInput]</a> order: <a href="#">[CheckboxViewModelSortInput]</a>
availableUsers(...): <a href="#">[UserProfile]</a>	Gets the set of available users for the currently signed-in dealer. userid: <a href="#">int!</a> where: <a href="#">[UserProfileFilterInput]</a> order: <a href="#">[UserProfileSortInput]</a>
currentUser: <a href="#">UserEmployeeModel</a>	Gets details about the currently signed-in user
currentUserPicture: <a href="#">String</a>	Gets details about the currently signed-in user
currentUserReadAuthentication(...):	Gets the specified service authentication records for the currently signed-in user.

4. Find currentUser in the list, and drill into UserEmployeeModel:

currentUser: [UserEmployeeModel](#)

Schema Reference

UserEmployeeModel  
Kind of type: Object

No description

Fields

address1: String	No description
address2: String	No description
cellCarrierId: Int	No description
cellPhone: String	No description
city: String	No description
description: String	No description
email: String	No description
employeeNumber: String	No description

5. This shows the fields that make up the UserEmployeeModel object. Create a query that returns some of these fields.
6. Return to the Operations view by pulling down the menu that says Schema Reference and changing it to Operations. Optionally, you could open a second tab in the tool (CTRL-ALT-T for Windows; Command-Option-T for Mac). This way, the first tab is on the schema reference and the second tab is on the operations.
7. Type your desired GraphQL query. Notice that the tool helps with auto-complete functionality:

Operations

Execute

```
1 query {  
2   setup  
3  
4  
5 }
```

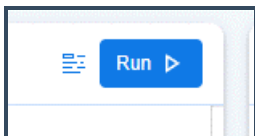
userManager UserManagerQueries

The completed query should look like this:

```
Operations
Execute
1 query {
2   setup {
3     userManager {
4       currentUser {
5         userName
6         description
7         jobTitle
8       }
9     }
10  }
11 }
```

This query retrieves only the userName, description, and jobTitle fields.

8. Click the Run button in the upper-right corner of the Operations panel:



The results appear in the Response panel:

```
Response 200 status
1 {
2   "data": {
3     "setup": {
4       "userManager": {
5         "currentUser": {
6           "userName": "dealer",
7           "description": "Is a programmer",
8           "jobTitle": "Not important"
9         }
10      }
11    }
12  }
13 }
```

Note that if you get an error at this point about not being authorized, one of two things has probably happened: either you did not follow the procedure in Getting set up and signed in or the token that you retrieved has expired. Follow the instructions in Getting set up and signed in to get a token, or if your token expired, return to the Connection Settings dialog and at the bottom of the Authorization tab, click Fetch Token and try again.

## Next steps

Familiarize yourself with the API and its structure. If you are unfamiliar with GraphQL or JSON, use the links in the Introduction to learn about those technologies.

Once you are familiar with these concepts, branch out to other tools, or use the information in this document to start

using a GraphQL client library to begin integrating this API into your own project.