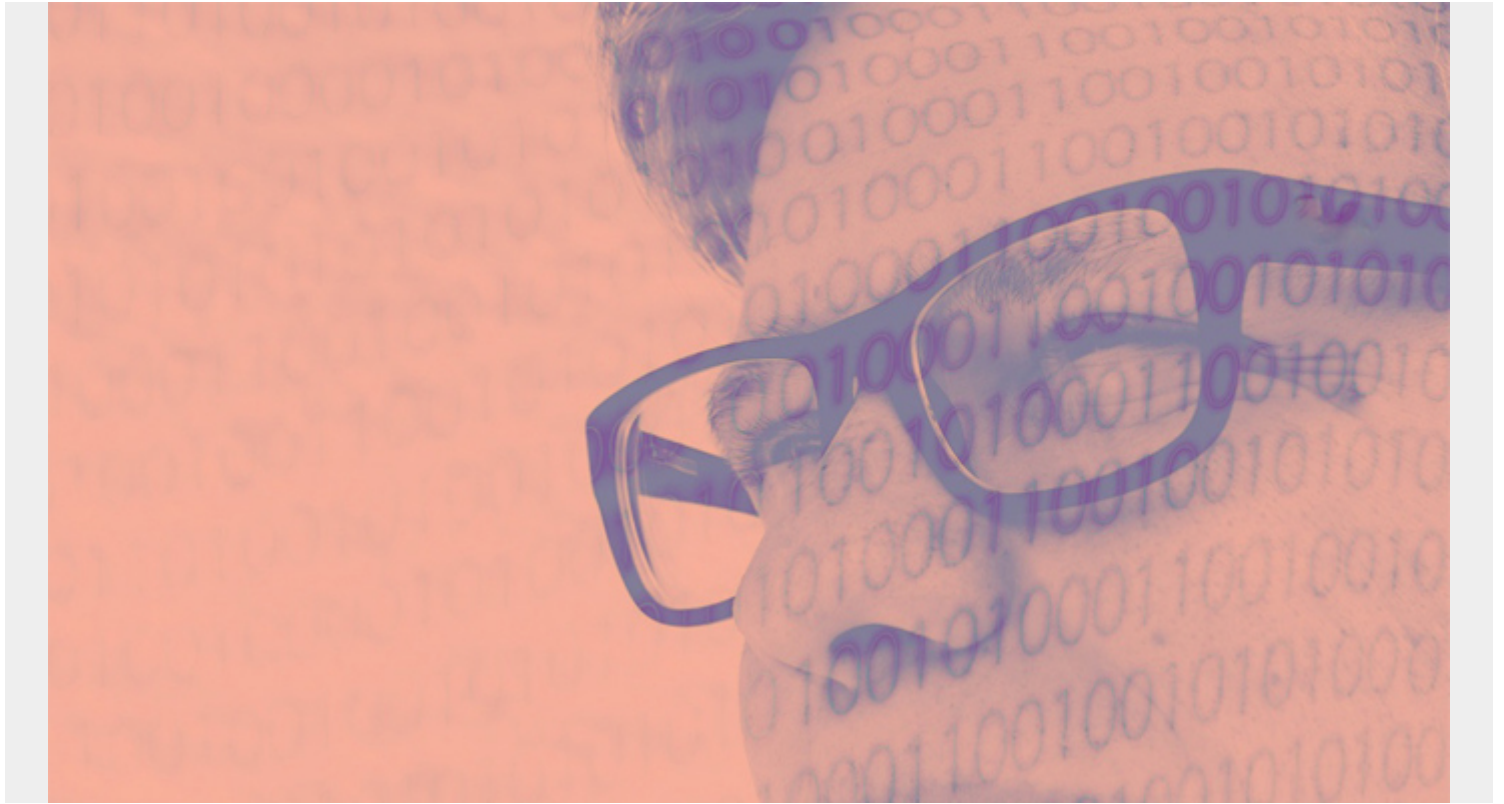


GETTING AUTHENTICATION ACCESS TOKENS FOR MICROSOFT APIS



In order to use Microsoft Power BI or other Microsoft APIs, you have to obtain an access token, also known as a bearer token. This is because Microsoft uses OAuth2, an industry standard protocol, for authentication. (In other words, a simple API key or username with a password is not enough.)

In this tutorial, we explain how to do that.

(This article is part of our [Power BI Guide](#). Use the right-hand menu to navigate.)

Note: We use curl to post data to Microsoft endpoints. That's like the command line version of Postman. On Mac and Ubuntu, curl is already there. You might have to install on Windows.

Registering Power BI

If you're doing all this for the very first time, in order to perform both steps of OAuth2 authentication, there's a Step 0.


You first have to register your application as a means of getting credentials. You do that one time. This generates an application ID and secret key. For Microsoft Power BI, you do it like this:

First, log into the embedding tool at <https://app.powerbi.com/embedsetup/UserOwnsData>

This is not the same as logging into Azure and creating an application in Active Directory there. You are creating an application on Power BI's Azure account (if you want to think of it that way).

Next, fill out the screens below. Note that:

- For the URL, you can use any web page. You will look at the parameters passed to this web page as we show below.
- Skip the screen that says **import content**.
- For API access, click **select all**.
- At the end, copy and save the **Application ID** and **Application Secret**.

 | **Power BI**

Developer | Embedding setup

Then, when you're ready to build a production-ready solution, we recommend setting up service principal. Need help? Check out our [troubleshooting guide](#)

✓ STEP 1

Sign in to Power BI

● STEP 2

Register your application

Register your application with Azure AD to allow your application to access the Power BI REST APIs and to set resource permissions for your application. You can change this later in the Microsoft Azure portal. [Learn more](#)

Application Name

Enter a display name to identify your application in Azure

Application Type

Server-side web application (for web apps or web APIs)

Home Page URL

☒ Use a default URL (Choose this if you'd like download a pre-configured sample app when you finish this process)

☐ Use a custom URL (Choose this if you know the target application's URL)

API access

Select the APIs and the level of access your application needs. You can change these settings later in portal.

[Learn more](#)

☒ Select all

Read only APIs ⓘ

- ☒ Read all datasets
- ☒ Read all dashboards
- ☒ Read all reports
- ☒ Read all workspaces
- ☒ Read all capacities
- ☒ Read all storage accounts
- ☒ Read all dataflows
- ☒ Read all gateways

Read and write APIs ⓘ

- ☒ Read and write all datasets
- ☒ Read and write all dashboards
- ☒ Read and write all reports
- ☒ Read and write all workspaces
- ☒ Read and write all capacities
- ☒ Read and write all storage accounts
- ☒ Read and write all dataflows
- ☒ Read and write all gateways

Create APIs ⓘ

- ☒ Create APIs

Summary:

User: Walker Rowe

Email: werowe@walkerrowe.com

Application: Expenses

Application ID:

[Redacted]

Application secret:

[Redacted]

Using OAuth2 for rest APIs

Once you've registered, you can move to this step.

Basic authentication is when you need only a user ID and password for access to something.

But Microsoft uses OAuth2 authentication. Microsoft APIs require that you present an Authorization header in order to use the API. Basically, OAuth2 is a two-step process:

1. Do a POST to login.microsoftonline.com

2. Take the access/bearer token from Step 1 and pass that to the API in a header called Authorization for whatever API you are calling.

Getting a token (code)


To get the authorization code, click on this URL to open a browser:

[https://login.microsoftonline.com/common/oauth2/authorize?client_id=\(appid\)&response_type=code&response_mode=query&redirect_uri=\(url you put when you registered app\)&scope=openid&state=foo](https://login.microsoftonline.com/common/oauth2/authorize?client_id=(appid)&response_type=code&response_mode=query&redirect_uri=(url you put when you registered app)&scope=openid&state=foo)

Basically, it will take you to the URL you put when you registered the application. But a screen will pop up asking you to grant certain permissions:

- **response_type**: code
- **response_mode**: query
- **state**: foo (Sny value will work here, it's just a place for free form-data.)
- **scope**: openid (You could also add offline_access.)
- **url**: We use the same URL throughout but change the URI to **authorize** and then **token** later to call different Microsoft endpoints.

Note: Here, the **tenant ID** is common, not a multi-tenant ID. Common means to retrieve the tenant ID associated with your Azure account.

 Microsoft

werowe@walkerrowe.com

Permissions requested

Review for your organization

expenses

unverified

This application is not published by Microsoft.

This app would like to:

- View all datasets
- View all dashboards
- View all reports
- View all workspaces
- View all capacities
- View all storage accounts
- View all dataflows
- View all gateways
- View all Power BI apps
- Read and write all datasets
- Read and write all dashboards
- Read and write all reports

- Sign in and read user profile

☒ Consent on behalf of your organization

If you accept, this app will get access to the specified resources for all users in your organization. No one else will be prompted to review these permissions.

Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. **The publisher has not provided links to their terms for you to review.** You can change these permissions at <https://myapps.microsoft.com>. [Show details](#)

Does this app look suspicious? [Report it here](#)

Cancel

Accept

Now,

you certainly could have written some kind of web listener to retrieve the code that Microsoft created. But we will just use the debugger in a Chrome browser to see the query parameter that Microsoft passed to our web page.

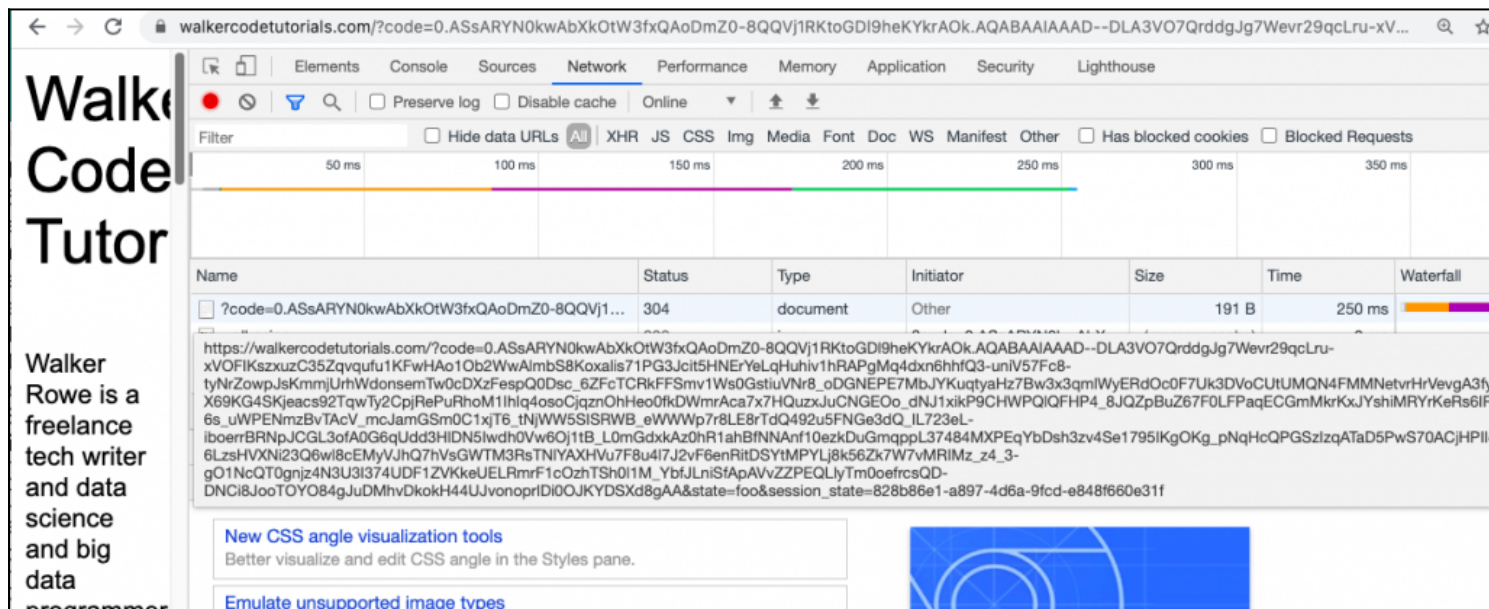
When Microsoft redirects you to the web page you indicated, go to the **network** tab in the browser and click the refresh button on the browser.

Then click on the **code** field and press **Copy as cURL**. The code (token) appears as the query parameter code as shown below.

`https://walkercodetutorials.com/?code=0.ASsARY...`

If you are wondering at this point why the URL is not some URL in Power BI, that's because you registered the application in Power BI. So, Microsoft knows that Power BI is what you want to use. The redirect URL serves merely as a place to retrieve this code.

Going forward, you would not want to click on the browser every time—this is not how a batch program would work. So look at the **prompt** setting in the [Microsoft Identity Platform reference guide](#) to see how to change that.



Getting an access token

We use curl to illustrate the next steps. Get the access token (bearer token) this way.

The values are:

- **grant_type**: Put "authorization_code"
- **client_id**: Application ID from above (The dots above hide my actual ID.)
- **client_secret**: Application Secret from above
- **redirect_uri**: Same as above
- **scope**: Same as above
- **url**: Note that the endpoint has changed to **token**

```
curl -X POST --form 'grant_type=authorization_code' --form 'client_id=7...5'
--form 'client_secret=21dVzEgtjUhfyzS3AJDaH0eMYB0q0ovYeH4YUoa//FM' --form
'scope=openid%20offline_access' --form 'response_type=code' --form
'redirect_uri=https://walkercodetutorials.com/' --form 'code=0.AS...AA'
https://login.microsoftonline.com/common/oauth2/token
```

Returns:

```
{"token_type":"Bearer","expires_in":"3599","ext_expires_in":"3599","expires_o
n":"1614591204","not_before":"1614587304","resource":"https://analysis.window
s.net/powerbi/api","access_token":"eyJ...G8CYZQT6t2p5IC1r3E7D_koNqc6h_-
f3918o_BP2N0Y0weCKKZ7WCw"}
```

Testing your Microsoft API access

Take the **access_token** value from the previous step and add it as an **Authorization** header value as shown below. (You have one hour before it expires.)

This, for example, is how you return a list of datasets in Power BI in My workspace. (That's the default workspace for free Power BI accounts, meaning for one individual's use only, as opposed to, for example, an enterprise account.)

Note: **myorg** does not mean your org. It's just a placeholder required by Microsoft.

```
curl -X GET -H "Authorization: Bearer ey...W_A" -H "Content-Type: application/json" https://api.powerbi.com/v1.0/myorg/datasets
```

That concludes this tutorial.

Related reading

- [BMC Machine Learning & Big Data Blog](#)
- [Power BI Basics: Creating a Pie Chart with Power Query Editor](#)
- [Data Visualization Guide](#), a series of tutorials
- [Data Storage Explained: Data Lake vs Warehouse vs Database](#)
- [Enabling the Citizen Data Scientists](#)
- [MySQL vs MongoDB: Comparing Databases](#)