

*Note: This document refers to Microsoft-specific OAuth 2.0 settings. The end-to-end OAuth 2.0 process for Mailit is described in the **OAuth 2.0 General.pdf** document that references this document.*

Using Microsoft Hotmail/LIVE OAUTH + IMAP

Microsoft Live IMAP4 servers (Hotmail, Outlook personal account) have been extended to support authorization via the industry-standard OAuth 2.0 protocol. Using OAUTH protocol, users can perform authentication by Microsoft Web OAuth instead of user and password. This way is more secure but also a little more complex.

Create your application in Azure Portal

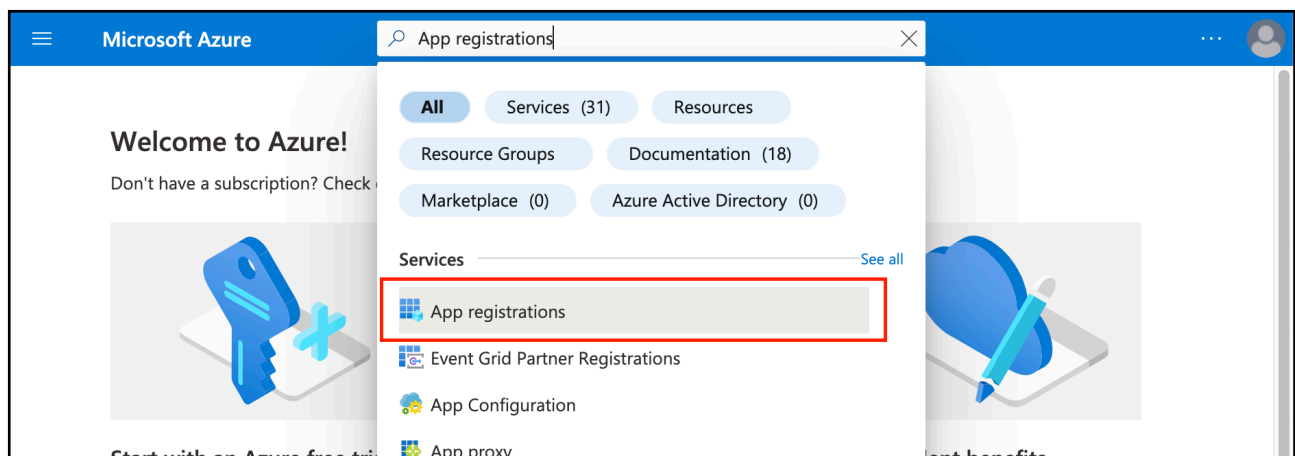
To use Microsoft/Office365 OAUTH in your solution, you must create an “application” here: <https://portal.azure.com>

Sign in to the Azure portal using either a work or school account or a personal Microsoft account.

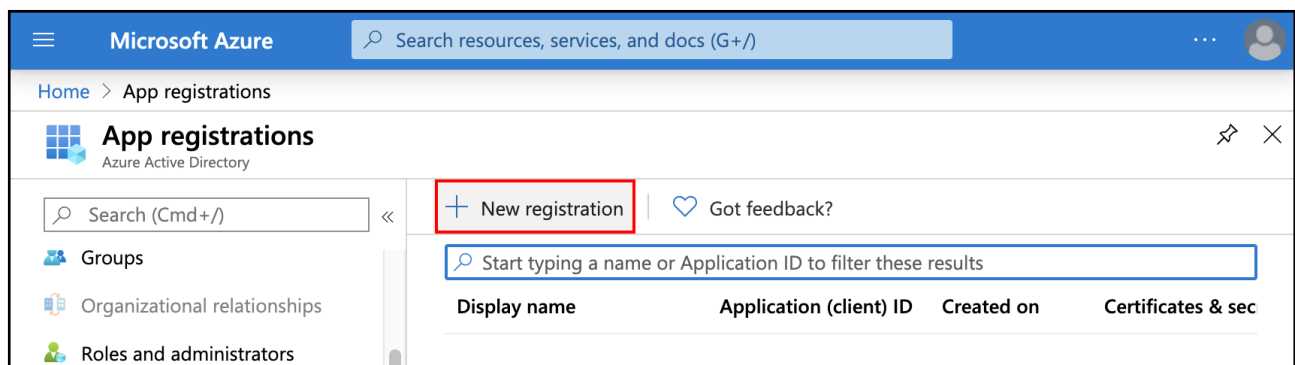
If you have access to more than one tenant, select your account in the top right corner, and set your portal session to the Azure AD tenant that you want.

To create an application, you need to go to the “App registrations” section in the Azure portal. This can be done in two ways:

1. In the left-hand navigation pane, select Azure Active Directory service, and then select App registrations. If the page did not open or opened but with a permissions error, use the second option.
2. In the search bar, start typing “App registrations”, in the drop-down search hint, select the “App registrations” service.



In the “App registrations” section, click the “New registration” button to create a new application.



When the Register an application page appears, enter your application's registration information:

Name - Enter a meaningful application name that will be displayed to users of the app.

Supported account types - Select which accounts you would like your application to support. To learn more about the types, click the "Help me choose ..." button. For the test we choose option 2 "Accounts in any organizational directory and personal Microsoft accounts" to support hotmail and outlook personal account.

Redirect URI (optional) - Select the type of app "Public client (mobile & desktop)", and then enter the redirect URI (<http://localhost:1234/oauth/>) for your application. The redirect address "<http://localhost:1234/oauth/>" associated with the Maillt plugin examples. In the examples, in the account settings, there are two fields "Local WebServer Host" and "Local WebServer Port". Adjust these depending on host name and port of the local Maillt web server.

When finished, select Register.

Microsoft Azure Search resources, services, and docs (G+/)

Home > App registrations > Register an application

Register an application

This application will not be associated with any directory and will be subject to limitations. You should not create production apps outside of a directory.

*** Name**
The user-facing display name for this application (this can be changed later).

Maillt OAuth ✓

Supported account types
Who can use this application or access this API?

☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant)

☒ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

☐ Personal Microsoft accounts only

[Help me choose...](#)

Redirect URI (optional)
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Public client/native (mobile & desktop) ✓

[By proceeding, you agree to the Microsoft Platform Policies](#)

Register

Azure AD assigns a unique application (client) ID to your app, and you're taken to your application's Overview page. Copy the "**Application (client) ID**" and "**Directory (tenant) ID**" to some place, you will need it for authorization.

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the 'Microsoft Azure' logo and a search bar. The left sidebar contains a 'Manage' section with various options like 'Branding & properties', 'Authentication', 'Certificates & secrets', etc. The main content area displays the 'Overview' page for an application named 'MailIt'. Under the 'Essentials' section, the 'Application (client) ID' and 'Directory (tenant) ID' are listed and highlighted with red boxes. The 'Application (client) ID' is 6b356da7-bdde-4122-9ddd-4f92d270... and the 'Directory (tenant) ID' is f8cdef31-a31e-4b4a-93e4-5f571e9125...

Property	Value
Display name	: MailIt
Application (client) ID	: 6b356da7-bdde-4122-9ddd-4f92d270...
Object ID	: 73f21747-f7a1-4510-a1d0-17e60511a...
Directory (tenant) ID	: f8cdef31-a31e-4b4a-93e4-5f571e9125...
Supported account types	: All Microsoft account users
Client credentials	: Add a certificate or secret
Redirect URIs	: 0 web, 0 spa, 1 public client
Application ID URI	: Add an Application ID URI

Next, you need to create a Client Secret, to do this you need to go to the "Certificates and secrets" section, to do this, click on the "Certificates and secrets" item in the left menu bar.

The screenshot shows the 'Certificates & secrets' page for the 'MailIt' application. The left sidebar has 'Certificates & secrets' selected. The main content area shows a message about application registration certificates and secrets. Below this, there are two tabs: 'Certificates (0)' and 'Client secrets (0)'. The 'Client secrets (0)' tab is highlighted with a red box. Below the tabs, there is a description of a client secret and a button labeled '+ New client secret', which is also highlighted with a red box. At the bottom, there is a table with columns: Description, Expires, Value, and Secret ID. The table is currently empty, with a message stating 'No client secrets have been created for this application.'

Description	Expires	Value	Secret ID
No client secrets have been created for this application.			

Click New Client Secret and fill in the form.

The screenshot shows the 'Add a client secret' form in the Microsoft Azure portal. The form is titled 'Add a client secret' and has a close button (X) in the top right corner. It contains two input fields: 'Description' with the value 'Maillt Secret' and 'Expires' with the value '24 months'. The left sidebar shows the navigation menu with 'Home > Maillt' and 'Maillt | Certificates & secrets' selected. The top bar shows the Microsoft Azure logo and a search bar.

Field	Value
Description	Maillt Secret
Expires	24 months

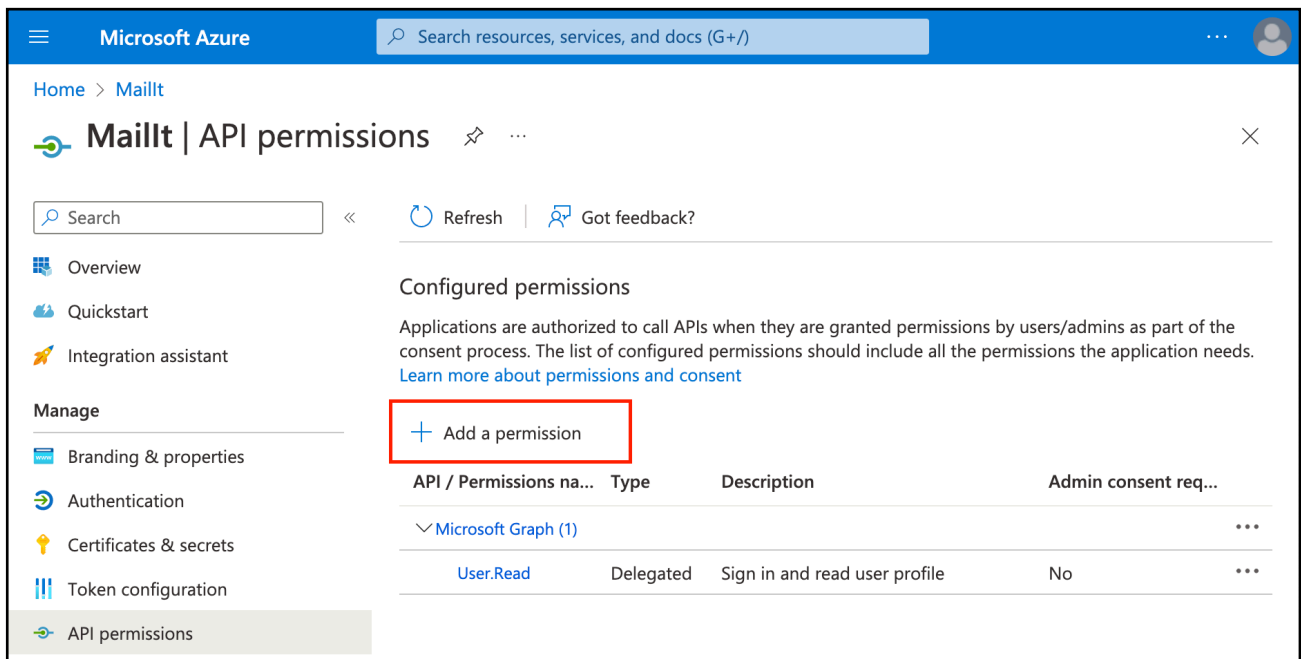
Remember to copy the client secret as soon as you create it. You will not be able to retrieve it after you perform another operation or leave the page.

The screenshot shows the 'Certificates & secrets' page in the Microsoft Azure portal. The page is titled 'Maillt | Certificates & secrets' and has a close button (X) in the top right corner. It contains a search bar, a 'Got feedback?' link, and a list of tabs: 'Certificates (0)' and 'Client secrets (1)'. The 'Client secrets (1)' tab is selected. Below the tabs, there is a table with columns: 'Description', 'Expires', 'Value', and 'Secret ID'. The table contains one row with the following data:

Description	Expires	Value	Secret ID
Maillt Secret	10/4/2024	Rux8Q~SUhggNL4...	3766ee44-22ca-41...

The 'Value' column is highlighted with a red box. The 'Secret ID' column contains a truncated ID and a copy icon.

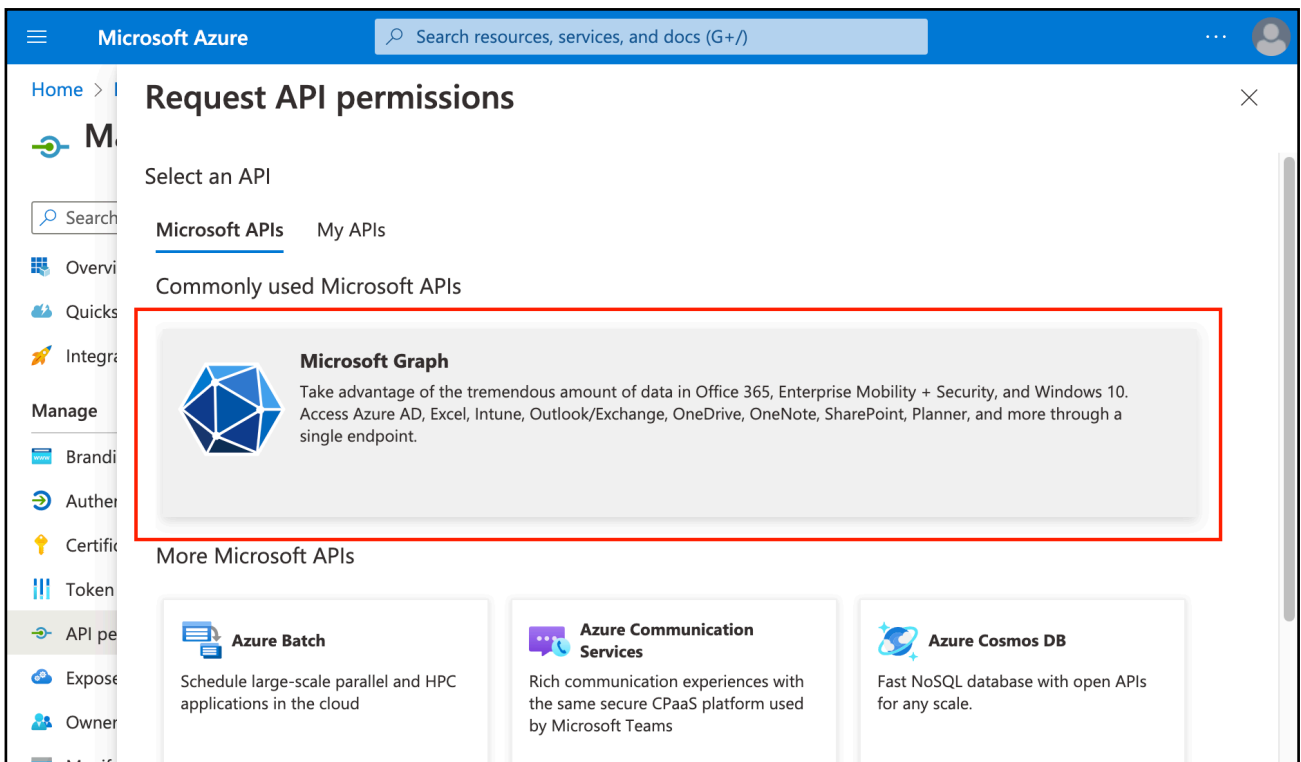
Next, define the rights that the application will be able to grant.
Go to "API Permission" and click "Add a permission".



The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and a user profile icon. The left sidebar contains a navigation menu with options like Home, Overview, Quickstart, Integration assistant, and a Manage section with sub-options like Branding & properties, Authentication, Certificates & secrets, Token configuration, and API permissions. The main content area is titled 'MailIt | API permissions' and includes a search bar, a Refresh button, and a Got feedback? link. Below this, there's a section for 'Configured permissions' with a description and a link to 'Learn more about permissions and consent'. A red box highlights the '+ Add a permission' button. Below the button is a table of configured permissions.

API / Permissions na...	Type	Description	Admin consent req...
Microsoft Graph (1)			
User.Read	Delegated	Sign in and read user profile	No

Choose Microsoft Graph



The screenshot shows the 'Request API permissions' page in the Microsoft Azure portal. The page has a title 'Request API permissions' and a subtitle 'Select an API'. There are two tabs: 'Microsoft APIs' and 'My APIs'. Under 'Microsoft APIs', there's a section for 'Commonly used Microsoft APIs' which includes a card for 'Microsoft Graph'. This card is highlighted with a red box. Below this, there's a section for 'More Microsoft APIs' which includes cards for 'Azure Batch', 'Azure Communication Services', and 'Azure Cosmos DB'.

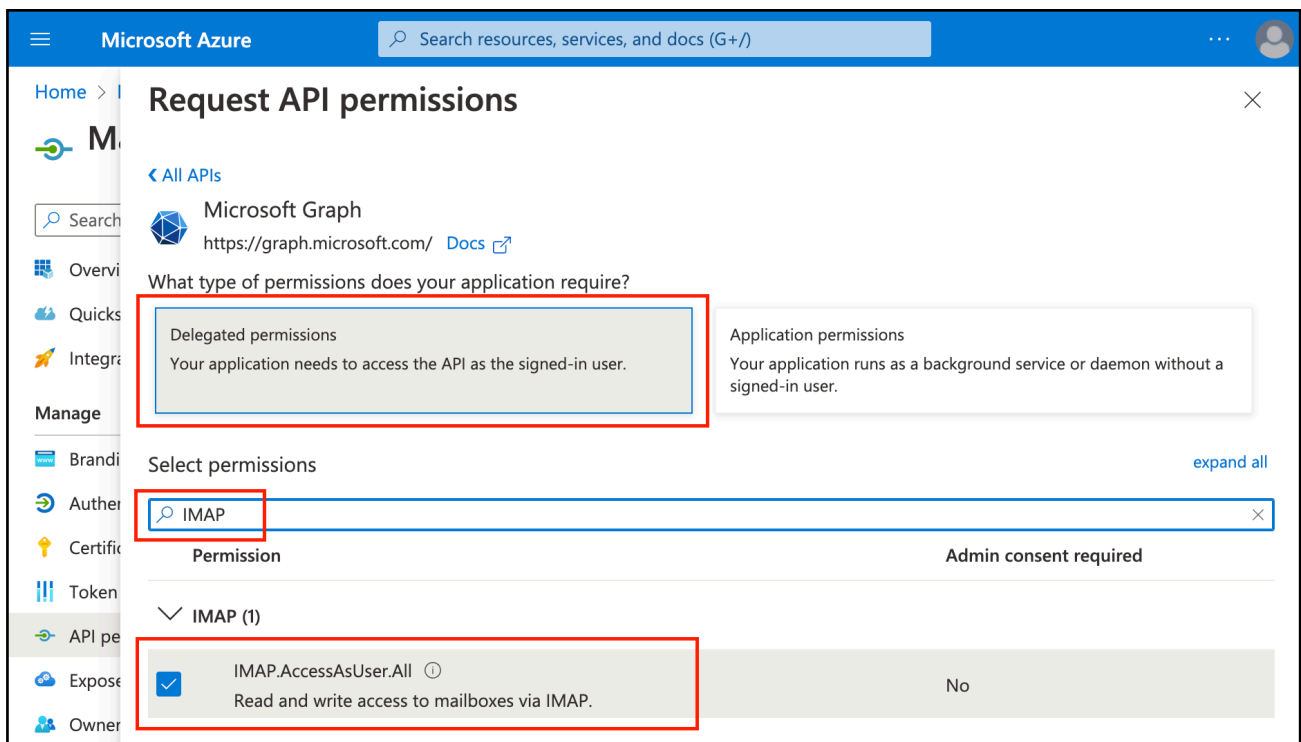
Microsoft Graph

Take advantage of the tremendous amount of data in Office 365, Enterprise Mobility + Security, and Windows 10. Access Azure AD, Excel, Intune, Outlook/Exchange, OneDrive, OneNote, SharePoint, Planner, and more through a single endpoint.

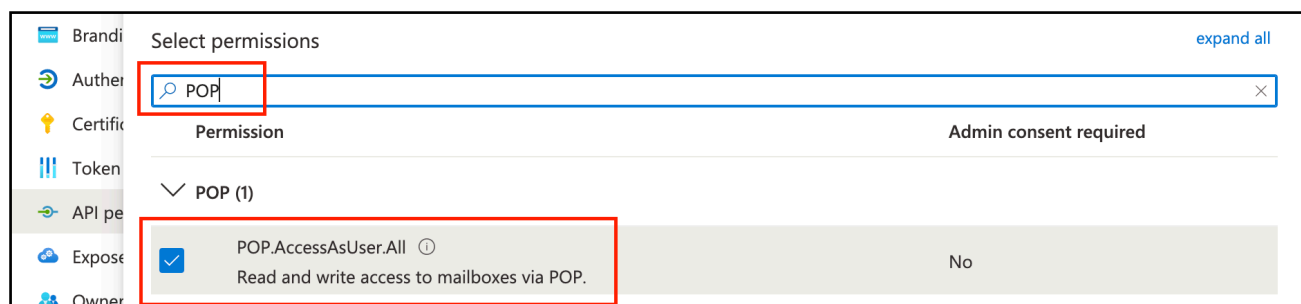
More Microsoft APIs

- Azure Batch**: Schedule large-scale parallel and HPC applications in the cloud
- Azure Communication Services**: Rich communication experiences with the same secure CPaaS platform used by Microsoft Teams
- Azure Cosmos DB**: Fast NoSQL database with open APIs for any scale.

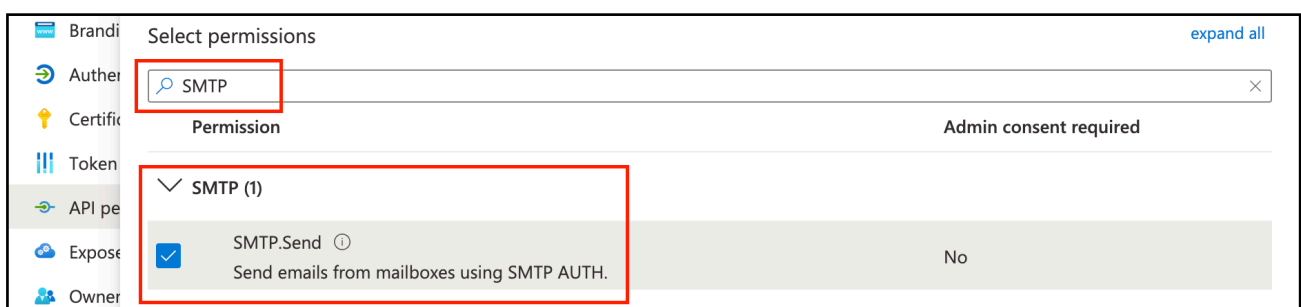
Select Delegated permissions and type IMAP in the search bar, IMAP.AccessAsUser.All permissions will appear under the search bar, select it.



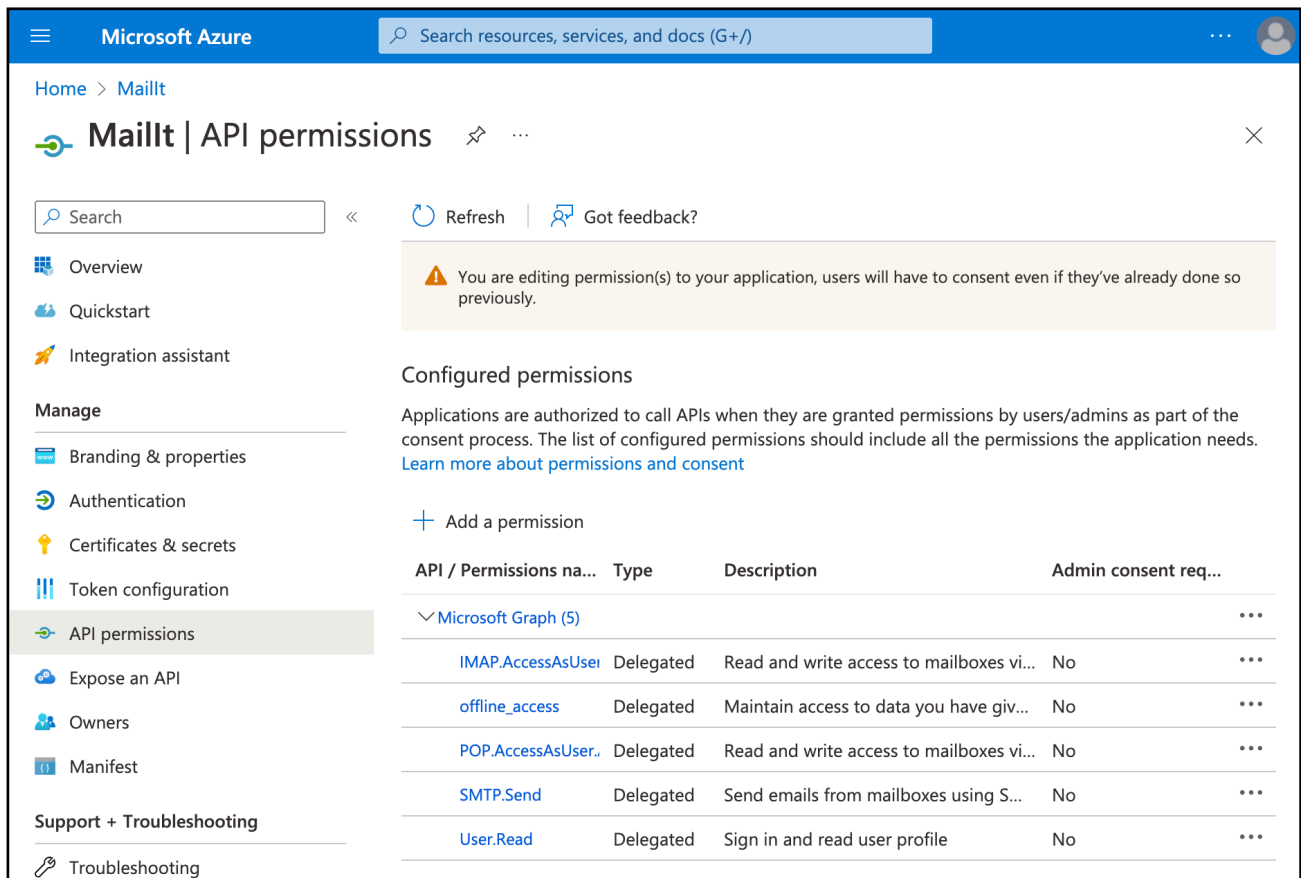
Next, type POP in the search bar, POP.AccessAsUser.All permissions will appear under the search bar, select it.



Next, type SMTP in the search bar, SMTP.Send permissions will appear under the search bar, select it.



Next in the search bar and start typing offline_access, when offline_access permissions appear under the search bar, select it.



That's all, click the add permission button. On the API Permissions page, the list of permissions should look like the screenshot below.

Above, client_id and secret support both "Office365 + Exchange Web Services (EWS)" and "Live (hotmail, outlook personal account) + SMTP/IMAP4", Office365 OAuth doesn't support SMTP/IMAP4, only EWS is supported.

Now, please open the app **Simple Receiving IMAP.fmp12**, go to connection settings by clicking on the gear icon located at the top right and follow these steps:

1. In the "IMAP Server" field, enter "outlook.office365.com".
1. In the "IMAP User Name" field, enter your email address.
2. In the field "Pref. IMAP Auth." select authorization type "OAUTH".
3. Click on the "OAuth 2.0 Client Credentials" tab.

4. In the "OAuth Provider" drop-down list, select "microsoftonline.com".
5. Previously, you copied the "Application (client) ID" and "client secret", paste the data into the corresponding fields "OAuth Client ID" and "OAuth Client Secret".

IMAP Connection Settings

IMAP Server

outlook.office365.com

IMAP Server Port

993

SSL/TLS

On

POP Server Timeout

60

IMAP Authentication only

Pref. IMAP Auth.

OAuth

IMAP User Name

dacons2020@outlook.com

IMAP Password

••••••••

Leave Messages on Server

On

Delete

Off

After (days)

7

Test Connection

Clear Settings

Last Session Transcript

OAuth 2.0 Client Credentials

OAuth 2.0 Additional settings

Log file

OAuth Provider

microsoftonline.com

OAuth Client ID

6b356da7-bdde-4122-9ddd-4f92d270a9c8

OAuth Client Secret

Rux8Q~SjHqgNL4Yuib2KT_67tQD2TwQIEQsvcaXF

Use Secret in Request

Off

Open OAuth2 Window in FileMaker

Open OAuth2 in External browser

Hint: In order to use OAuth2 authorization you need to create an application.

For the GMail service:
<https://console.developers.google.com/>

For the Outlook or Live service:
<https://portal.azure.com/>

After creating the application, you will be available to "Client ID" and "Client Secret"

6. If you want to specify a tenant, you can do so by replacing "common" with your tenant ID in the Authorization URL and Token URL fields, on the "OAuth 2.0 Additional settings" tab:

Authorization URL: <https://login.microsoftonline.com/{tenant}/oauth2/v2.0/authorize>

Token URL: <https://login.microsoftonline.com/{tenant}/oauth2/v2.0/token>

Last Session Transcript

OAuth 2.0 Client Credentials

OAuth 2.0 Additional settings

Log file

Local WebServer Host

localhost

Local WebServer Port

1234

Start

Stop

OAuth Refresh Token

Authorization URL

<https://login.microsoftonline.com/common/oauth2/v2.0/authorize>

Token URL

<https://login.microsoftonline.com/common/oauth2/v2.0/token>

Scope

offline_access https://outlook.office.com/POP.AccessAsUser.All https://outlook.office.com/IMAP.AccessAsUser.All

Token Expires In

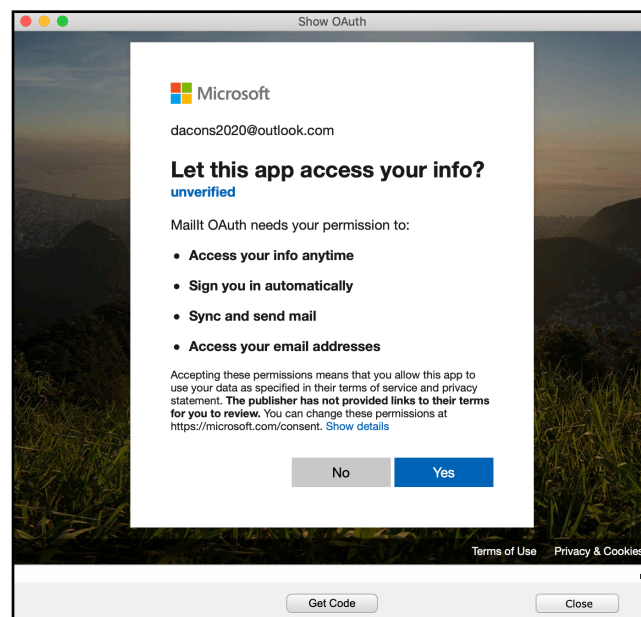
©2026 Dacons UG (haftungsbeschränkt)

The {tenant} value in the path of the request can be used to control who can sign into the application. Valid values are common, organizations, consumers, and tenant identifiers. For guest scenarios where you sign a user from one tenant into another tenant, you must provide the tenant identifier to sign them into the resource tenant. For more information, see Endpoints.
(<https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-auth-code-flow>)

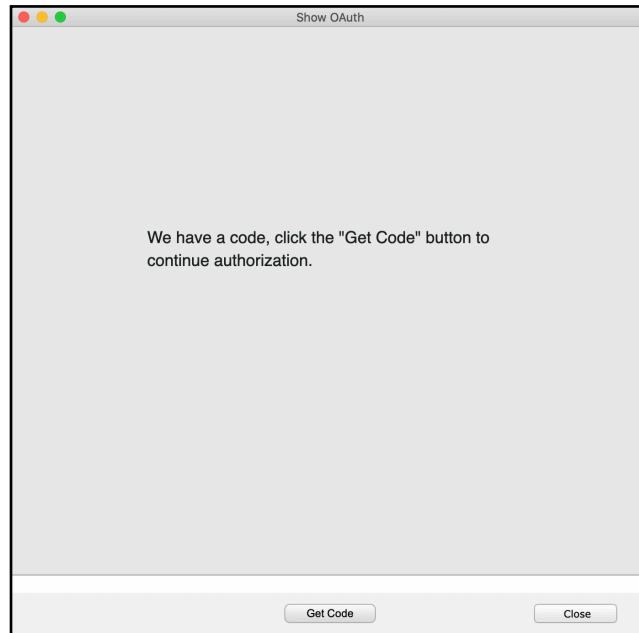
7. Click the “Open OAuth2 Wndow in FileMaker” or “Open OAuth2 in External browser” button to complete web authorization, as a result of which you can get a “Token”.

OAuth 2.0 Client Credentials configuration window. Fields include OAuth Provider (microsoftonline.com), OAuth Client ID (6b356da7-bdde-4122-9ddd-4f92d270a9c8), OAuth Client Secret (Rux8Q~SUh9gNL4Yujb2KT_67tQD2TwQIEQsvcaXF), and Use Secret in Request (Off). Buttons at the bottom: Open OAuth2 Wndow in FileMaker, Open OAuth2 in External browser.

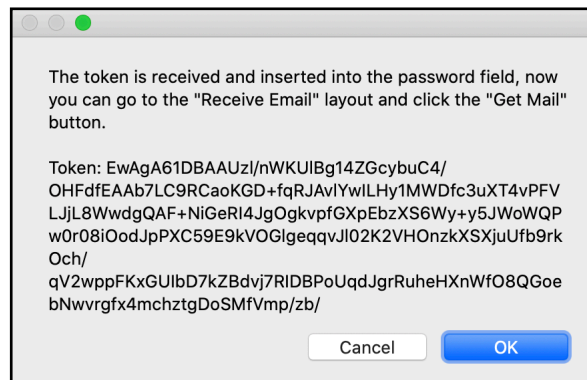
After clicking the "Open OAuth2 Window" button, a window with web authorization will open. Log in to your account and allow access to your application.



As a result, Microsoft will pass the string called “code” which is needed to get the “token”. Click the “Get Code” button to pick up the line from the WebView and complete the request for a “token”.



If everything went well you will see a message:



After receiving Token, click the “Test Connection” button to test the connection.

Leave Messages on Server	On
Delete	Off
After (days)	7
Test Connection Clear Settings	
Last Session TranscriptOAuth 2.0 Client CredentialsOAuth 2.0 Additional settingsLog file	
OAuth Provider	microsoftonline.com
OAuth Client ID	6b356da7-bdde-4122-9ddd-4f92d270a9c8
Hint: In order to use OAuth2 authorization you need to create an application. For the Gmail service: https://console.developers.google.com/ For the Outlook or Live service:	

After a successful connection, you can go to the Receive Email layout and click the Get Mail button to download messages from the server.



In response to the “Token” request, two additional properties are transferred: “Refresh Token” and “Token Expires In”. “Token” is substituted in the password field. “Refresh Token” and “Token Expires In” can be found in the “OAuth 2.0 Additional settings” tab.

In this example, the “Refresh Token” field displays the value in clear text so that authorization can be debugged, in real life, save the “Refresh Token” in a safe place.

The lifetime of “Token” is 1 hour from the moment it was received. After 1 hour, you need to request a new “Token” using “Refresh Token”. The implementation of the “Token” update process can be viewed in a script called “Test Connection” or “Connect to the mail server”.

Last Session Transcript	OAuth 2.0 Client Credentials	OAuth 2.0 Additional settings	Log file
Local WebServer Host	localhost	Local WebServer Port	1234
OAuth Refresh Token	M.C545_BL2.0.U.-CIBOKwvm!Jcn3czskmW!		
Authorization URL	https://login.microsoftonline.com/common/oauth2/v2.0/authorize		
Token URL	https://login.microsoftonline.com/common/oauth2/v2.0/token		
Scope	offline_access https://outlook.office.com/POP.AccessAsUser.All https://outlook.office.com/IMAP.AccessAsUser.All		
Token Expires In	Tuesday, 10.09.2024 21:11:24 EEST		

On the “OAuth 2.0 Additional settings” tab, there are additional settings such as “Local WebServer Host” and “Local WebServer Port.” These fields are needed to configure the local web server. The local web server is needed to display the html file (../Mailit Assets/oauth/index.html) in FileMaker WebView, the file is accessible by address (http://localhost:1234/oauth/index.html) for web authorization. The local web server starts when the “Open OAuth2 Window” button is clicked and stops after clicking the “Get Code.” If for some reason the local server cannot start, then check that the plugin has permissions read / write to a local drive, and that the port specified in the settings “1234” is not being used by another application or service. If the port is busy. Try to specify any other port that is available on your system.

Additional Information

Microsoft identity platform developer glossary

<https://docs.microsoft.com/en-us/azure/active-directory/develop/developer-glossary>

Authentication with the Live Connect endpoint

<https://docs.microsoft.com/en-us/advertising/guides/authentication-oauth-live-connect?view=bingads-13>