Authenticate an IMAP, POP or SMTP connection using OAuth

Article • 10/25/2022 • 7 minutes to read

Learn how to use OAuth authentication to connect with IMAP, POP or SMTP protocols and access email data for Office 365 users.

> OAuth2 support for IMAP, POP, SMTP protocols as described below is supported for both Microsoft 365 (which includes Office on the web) and Outlook.com users.

If you're not familiar with the OAuth 2.0 protocol, start by reading the OAuth 2.0 protocol on Microsoft identity platform overview. To learn more about the Microsoft Authentication Libraries (MSAL), which implement the OAuth 2.0 protocol to authenticate users and access secure APIs, read the MSAL overview.

You can use the OAuth authentication service provided by Azure Active Directory (Azure AD) to enable your application to connect with IMAP, POP or SMTP protocols to access Exchange Online in Office 365. To use OAuth with your application, you need to:

1. Register your application with Azure AD.
2. Get an access token from a token server.
3. Authenticate connection requests with an access token.

Register your application

To use OAuth, an application must be registered with Azure Active Directory.

Follow the instructions listed in Register an application with the Microsoft identity platform to create a new application.

Get an access token

You can use one of our MSAL client libraries to fetch an access token from your client application.

Alternatively, you can select an appropriate flow from the following list and follow the corresponding steps to call the underlying identity platform REST APIs and retrieve an access token.

1. OAuth2 authorization code flow
2. OAuth2 device authorization grant flow
3. OAuth2 client credentials grant flow

Make sure to specify the full scopes, including Outlook resource URLs, when authorizing your application and requesting an access token.

Protocol	Permission scope string
IMAP	`https://outlook.office.com/IMAP.AccessAsUser.All`
POP	`https://outlook.office.com/POP.AccessAsUser.All`
SMTP AUTH	`https://outlook.office.com/SMTP.Send`

In addition, you can request for offline_access scope. When a user approves the offline_access scope, your app can receive refresh tokens from the Microsoft identity platform token endpoint. Refresh tokens are long-lived. Your app can get new access tokens as older ones expire.

Authenticate connection requests

You can initiate a connection to Office 365 mail servers using the IMAP and POP email settings for Office 365.

SASL XOAUTH2

OAuth integration requires your application to use SASL XOAUTH2 format to encode and transmit the access token. SASL XOAUTH2 encodes the username, access token together in the following format:

```text
base64("user=" + userName + "^Aauth=Bearer " + accessToken + "^A^A")
```

`^A` represents a **Control** + **A** (`%x01`).

For example, the SASL XOAUTH2 format to access `test@contoso.onmicrosoft.com` with access token `EwBAAl3BAAUFFpUAo7J3Ve0bjLBWZWCclRC3EoAA` is:

```text
base64("user=test@contoso.onmicrosoft.com^Aauth=Bearer
EwBAAl3BAAUFFpUAo7J3Ve0bjLBWZWCclRC3EoAA^A^A")
```

After base64 encoding, this translates to the following string. Note that line breaks are inserted for readability.

```text
dXNlcj10ZXN0QGNvbnRvc28ub25taWNyb3NvZnQuY29tAWF1dGg9QmVhcmVy
IEV3QkFBbDNCQUFVRkZwVUFvN0ozVmUwYmpMQldaV0NjbFJDM0VvQUEBAQ==
```

SASL XOAUTH2 authentication for shared mailboxes in Office 365

In case of shared mailbox access using OAuth, application needs to obtain the access token on behalf of a user but replace the userName field in the SASL XOAUTH2 encoded string with the email address of the shared mailbox.

IMAP Protocol Exchange

To authenticate an IMAP server connection, the client must respond with an `AUTHENTICATE` command in the following format:

```text
AUTHENTICATE XOAUTH2 <base64 string in XOAUTH2 format>
```

Sample client-server message exchange that results in an authentication success:

```text
[connection begins]
C: C01 CAPABILITY
S: * CAPABILITY … AUTH=XOAUTH2
S: C01 OK Completed
```

```
C: A01 AUTHENTICATE XOAUTH2
dXNlcj1zb21ldXNlckBleGFtcGxlLmNvbQFhdXRoPUJlYXJlciB5YTI5LnZGOWRmdDRxbVRjMk52YjN
SbGNrQmhkSFJoZG1semRHRXVZMjl0Q2cBAQ==
S: A01 OK AUTHENTICATE completed.
```

Sample client-server message exchange that results in an authentication failure:

text

```
[connection begins]
S: * CAPABILITY … AUTH=XOAUTH2
S: C01 OK Completed
C: A01 AUTHENTICATE XOAUTH2
dXNlcj1zb21ldXNlckBleGFtcGxlLmNvbQFhdXRoPUJlYXJlciB5YTI5LnZGOWRmdDRxbVRjMk52YjN
SbGNrQmhkSFJoZG1semRHRXVZMjl0Q2cBAQ==
S: A01 NO AUTHENTICATE failed.
```

POP Protocol Exchange

To authenticate a POP server connection, the client will have to respond with an `AUTH` command split into two lines in the following format:

text

```
AUTH XOAUTH2
<base64 string in XOAUTH2 format>
```

Sample client-server message exchange that results in an authentication success:

text

```
[connection begins]
C: AUTH XOAUTH2
S: +
C: dXNlcj1zb21ldXNlckBleGFtcGxlLmNvbQFhdXRoPUJlYXJlciB5YX
JlciB5YTI5LnZGOWRmdDRxbVRjMk52YjNSbGNrQmhkSFJoZG1semRHRXVZMjl0
Q2cBAQ==
S: +OK User successfully authenticated.
[connection continues...]
```

Sample client-server message exchange that results in an authentication failure:

text

```
[connection begins]
C: AUTH XOAUTH2
S: +
C: dXNlcj1zb21ldXNlckBleGFtcGxlLmNvbQFhdXRoPUJlY
XJlciB5YTI5LnZGOWRmdDRxbVRjMk52YjNSbGNrQmhkSFJoZG1semRHRXVZMj
l0Q2cBAQ=
S: -ERR Authentication failure: unknown user name or bad password.
```

SMTP Protocol Exchange

Note As per the current test with SMTP Oauth 2.0 client credential flow with non-interactive sign in is not supported.

To authenticate an SMTP server connection, the client must respond with an `AUTH` command in the following format:

text

```
AUTH XOAUTH2 <base64 string in XOAUTH2 format>
```

Sample client-server message exchange that results in an authentication success:

text

```
[connection begins]
C: auth xoauth2
S: 334
C: dXNlcj1zb21ldXNlckBleGFtcGxlLmNvbQFhdXRoPUJlY
XJlciB5YTI5LnZGOWRmdDRxbVRjMk52YjNSbGNrQmhkSFJoZG1semRHRXVZMj
l0Q2cBAQ==
S: 235 2.7.0 Authentication successful
[connection continues...]
```

Sample client-server message exchange that results in an authentication failure:

text

```
[connection begins]
C: auth xoauth2
S: 334
C: dXNlcj1zb21ldXNlckBleGFtcGxlLmNvbQFhdXRoPUJlY
XJlciB5YTI5LnZGOWRmdDRxbVRjMk52YjNSbGNrQmhkSFJoZG1semRHRXVZMj
l0Q2cBAQ==
S: 535 5.7.3 Authentication unsuccessful
[SN2PR00CA0018.namprd00.prod.outlook.com]
```

Use client credentials grant flow to authenticate IMAP and POP connections

Service principals in Exchange are used to enable applications to access Exchange mailboxes via client credentials flow with the POP and IMAP protocols.

Add the POP and IMAP permissions to your AAD application

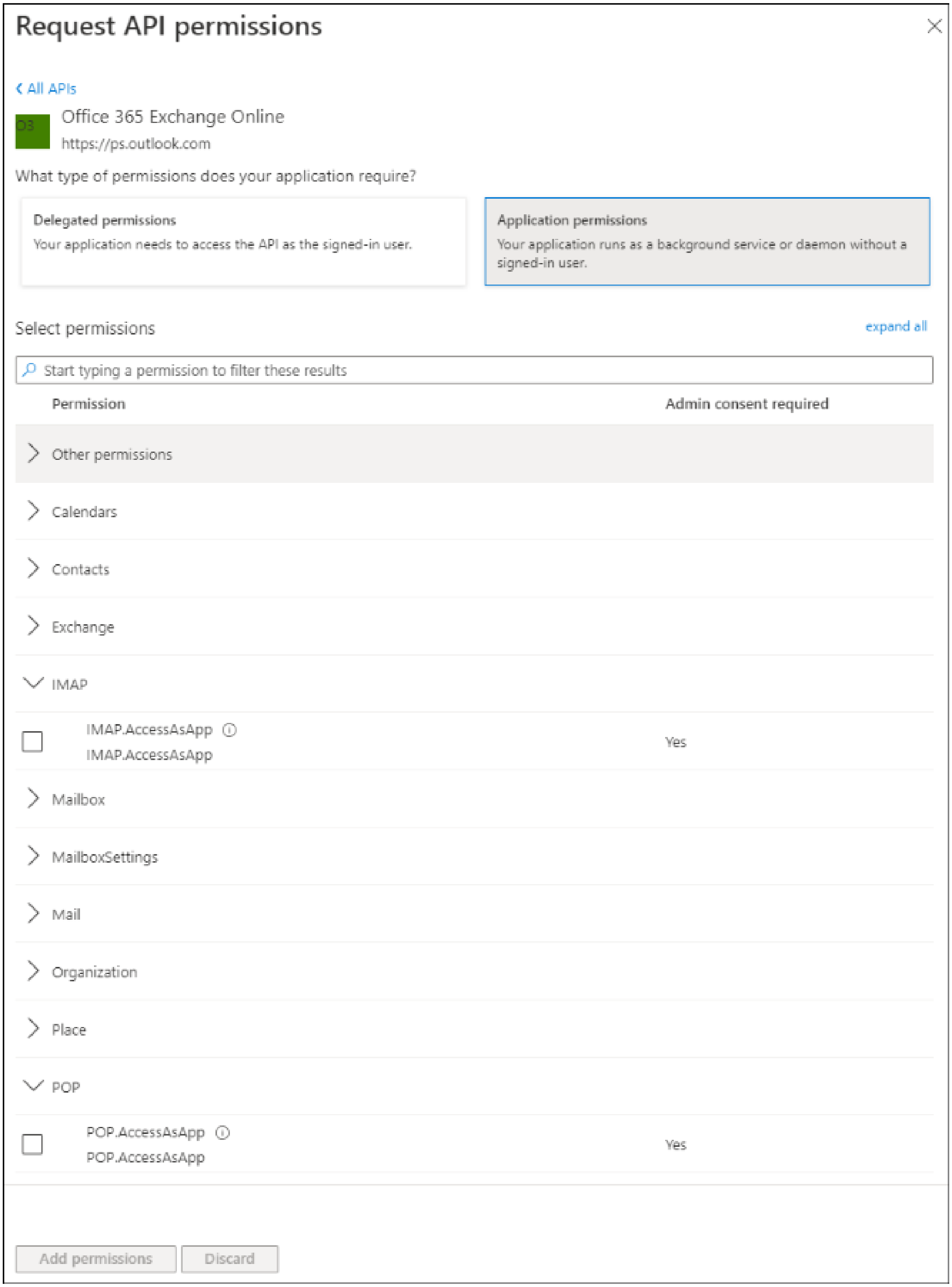
1. In the Azure portal, choose the **API Permissions** blade in your Azure AD application's management view.

2. Select **Add permission**.

3. Select the **APIs my organization uses** tab and search for "*Office 365 Exchange Online*".

4. Click **Application permissions**.

5. For POP access, choose the **POP.AccessAsApp** permission. For IMAP access, choose the **IMAP.AccessAsApp** permission.



Office 365 Exchange Online
https://ps.outlook.com

What type of permissions does your application require?

Delegated permissions
Your application needs to access the API as the signed-in user.

Application permissions
Your application runs as a background service or daemon without a signed-in user.

	Permission	Admin consent required
	› Other permissions	
	› Calendars	
	› Contacts	
	› Exchange	
	⌄ IMAP	
□	IMAP.AccessAsApp ⓘ IMAP.AccessAsApp	Yes
	› Mailbox	
	› MailboxSettings	
	› Mail	
	› Organization	
	› Place	
	⌄ POP	
□	POP.AccessAsApp ⓘ POP.AccessAsApp	Yes

6. Once you've chosen which type of permission, select **Add permissions**.

You should now have the POP or IMAP application permissions added to your AAD application's permissions.

Get tenant admin consent

To access Exchange mailboxes via POP or IMAP, your AAD application must get tenant admin consent for each tenant. To learn more, see tenant admin consent process.

How to grant consent if the application is registered/configured for multiple tenant usage e.g. for Partner/ISV developed centraly registered application

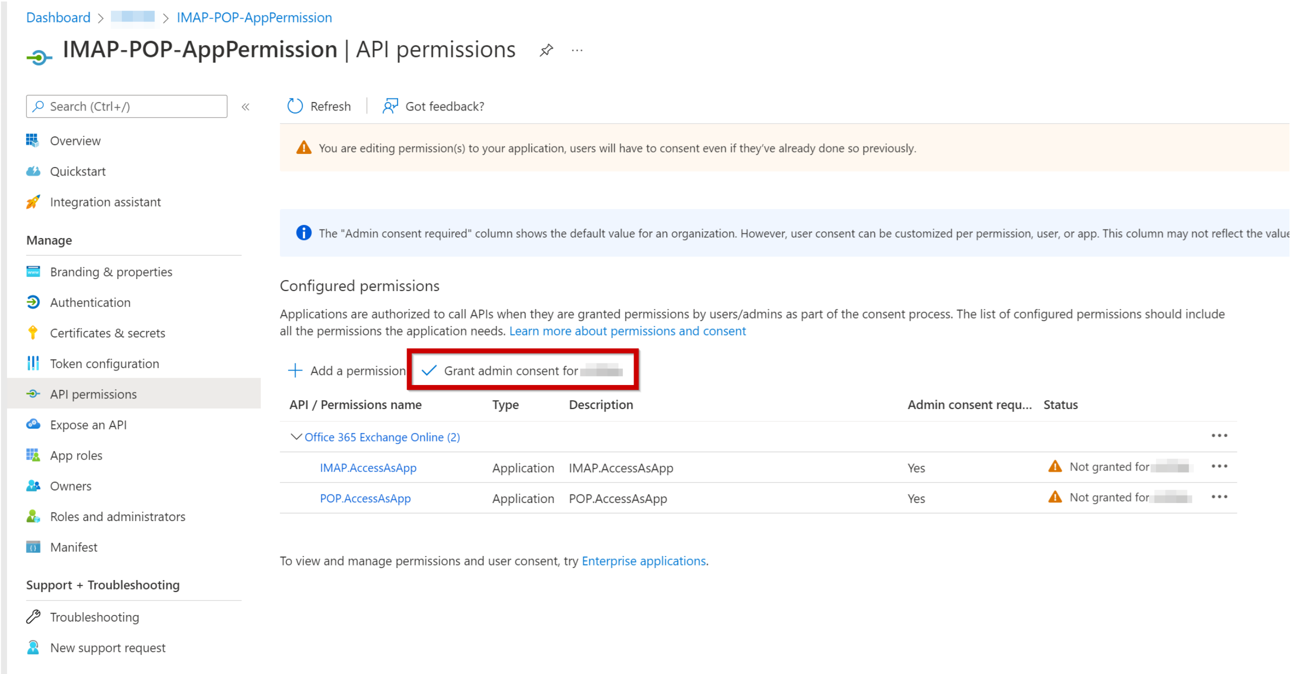
If your ISV/partner registered the Azure AD Appliacation with the option "Accounts in any organizational directory", you need to add this application and consent it using the following steps by leveraging the authorization request URL.

In your OAuth 2.0 tenant authorization request, the `scope` query parameter should be `https://ps.outlook.com/.default` for both the POP and IMAP application scopes. The following is an example of the OAuth 2.0 authorization request URL:

```text
https://login.microsoftonline.com/{tenant}/v2.0/adminconsent?client_id=
<CLIENT_ID>&redirect_uri=<REDIRECT_URI>&scope=https://ps.outlook.com/.default
```

How to grant consent if you registered the application for your own tenent

If you registered your application in your own tenant using "Accounts in this organizational directory only", you can simply go forward and use the application configuration page within the Azure AD admin center to grant the admin consent, and don´t need to use the authorization request URL approch.



Dashboard > [redacted] > IMAP-POP-AppPermission

IMAP-POP-AppPermission | API permissions

Search (Ctrl+/)

- Overview
- Quickstart
- Integration assistant

Manage

- Branding & properties
- Authentication
- Certificates & secrets
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators
- Manifest

Support + Troubleshooting

- Troubleshooting
- New support request

Refresh | Got feedback?

You are editing permission(s) to your application, users will have to consent even if they've already done so previously.

The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. Learn more about permissions and consent

+ Add a permission ✓ Grant admin consent for [redacted]

API / Permissions name	Type	Description	Admin consent requ...	Status
Office 365 Exchange Online (2)				
IMAP.AccessAsApp	Application	IMAP.AccessAsApp	Yes	Not granted for [redacted]
POP.AccessAsApp	Application	POP.AccessAsApp	Yes	Not granted for [redacted]

To view and manage permissions and user consent, try Enterprise applications.

Register service principals in Exchange

Once your Azure AD application is consented to by a tenant admin, the tenant admin must register your AAD application's service principal in Exchange via Exchange Online PowerShell. This is enabled by the New-ServicePrincipal cmdlet.

To use the New-ServicePrincipal cmdlet, install the ExchangeOnlineManagement and connect to your tenant as shown in the following snippet.

text

```
Install-Module -Name ExchangeOnlineManagement -allowprerelease
Import-module ExchangeOnlineManagement
Connect-ExchangeOnline -Organization <tenantId>
```

If you still get an error running the New-ServicePrincipal Cmdlet after you perform these steps, it is likely due to the fact that the user does'nt have enough permissions in Exchange online to perform the operation.

The following is an example of registering an Azure AD application's service principal in Exchange:

text

```
New-ServicePrincipal -AppId <APPLICATION_ID> -ServiceId <OBJECT_ID> [-
Organization <ORGANIZATION_ID>]
```

The tenant admin can find the service principal identifiers referenced above in your AAD application's enterprise application instance on the tenant. You can find the list of the enterprise application instances on the tenant in the **Enterprise applications** blade in the Azure Active Directory view in Azure Portal.

You can get your registered service principal's identifier using the Get-ServicePrincipal cmdlet.

text

```
Get-ServicePrincipal | fl
```

The OBJECT_ID is the Object ID from the Overview page of the Enterprise Application node (Azure Portal) for the application registration. It is **not** the Object ID from the Overview of the App Registrations node. Using the incorrect Object ID will cause an authentication failure.

The tenant admin can now add the specific mailboxes in the tenant that will be allowed to be access by your application. This is done with the Add-MailboxPermission cmdlet.

The following is an example of how to give your application's service principal access to one mailbox:

text

```
Add-MailboxPermission -Identity "john.smith@contoso.com" -User
<SERVICE_PRINCIPAL_ID> -AccessRights FullAccess
```

Your Azure AD application can now access the allowed mailboxes via the POP or IMAP protocols using the OAuth 2.0 client credentials grant flow. For more information, see the instructions in Permissions and consent in the Microsoft identity platform.

You must use `https://outlook.office365.com/.default` in the `scope` property in the body payload for the access token request.

The access tokens generated can be used as tokens to authenticate POP and IMAP connections via SASL XOAUTH2 format as described previously.

See also

- IMAP OAuth testing using powershell script
- Authentication and EWS in Exchange
- IMAP, POP Connection settings
- Internet Message Access Protocol
- Post Office Protocol
- SMTP Service extension for Authentication