

Speaker availability - o365

To set up the "Speaker availability" feature, the following data points need to be provided:

```
clientId  
tenantId  
clientSecret
```

Authentication and authorization steps

The basic steps required to configure a service and get a token from the Microsoft identity platform endpoint that your service can use to call Microsoft Graph under its own identity are:

1. Register your app.
2. Configure permissions for Microsoft Graph on your app.
3. Get administrator consent.
4. Create client secret

1. Register your app

To authenticate with the Microsoft identity platform endpoint, you must first register your app at the [Azure app registration portal](#). You can use either a Microsoft account or a work ~~or school~~ account to register your app.

For a service that will call Microsoft Graph under its own identity, you need to register your app for the Web platform and copy the following values:

- The Application ID assigned by the Azure app registration portal.
- A Client (application) Secret, either a password or a public/private key pair (certificate).
- A Redirect URL for your service to receive token responses (not necessarily)
- A Redirect URL for your service to receive admin consent responses if your app implements functionality to request administrator consent. (not necessarily)

For steps on how to configure an app using the Azure app registration portal, see [Register your app](#).

Search (Ctrl+/)

Delete

Endpoints

Preview features

Overview

Quickstart

Integration assistant

Manage

Branding

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

App roles

Owners

Roles and administrators | Preview

Manifest

Support + Troubleshooting

Troubleshooting

New support request

Essentials

Display name : postman - Microsoft graph app

Application (client) ID : 269a4c98-ec7a-466c-a2ab-

Object ID : 488e524a-2cce-4cec-8188-4db1a373c75a

Directory (tenant) ID : 5f15a603-f962-433d-a4b0-

Supported account types : My organization only

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will upgrade to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)

Get Started

Documentation

Build your application with the M

The Microsoft identity platform is an authentication service, open-source libraries, and application mar and protect APIs, and add sign-in for your use

Cloud

Database

N

X

S

O

Call APIs

Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources.

View API permissions

Sign in users in 5 minutes

Use our SDKs to sign in users and steps. Use the quickstarts to start app, SPA, or daemon app.

View all quickstart guides

2. Configure permissions for Microsoft Graph

To configure application permissions for your app in the [Azure app registrations portal](#): under an application's **API permissions** page, choose **Add a permission**, select **Microsoft Graph**, and then choose the permissions your app requires under **Application permissions**.

The following screenshot shows the **Select Permissions** dialog box for Microsoft Graph application permissions.

Select "Calendars.Read"

Request API permissions

[← All APIs](#)

Microsoft Graph

<https://graph.microsoft.com/> [Docs](#)

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.

Select permissions

[expand all](#)

Permission

Admin consent required

✓ Calendars (1)



Calendars.Read ⓘ

Read calendars in all mailboxes

Yes



Calendars.ReadWrite ⓘ

Read and write calendars in all mailboxes

Yes

[Add permissions](#)[Discard](#)

3. Administrator consent experience

You can rely on an administrator to grant the permissions your app needs at the [Azure portal](#); however, often, a better option is to provide a sign-up experience for administrators by using the Microsoft identity platform `/adminconsent` endpoint.

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 Signet

API / Permissions name	Type	Description	Admin consent requ...	Status
▼ Microsoft Graph (1) ...				
Calendars.Read	Application	Read calendars in all mailboxes	Yes	✓ Granted for Signet ...

Other permissions granted for Signet

These permissions have been granted for Signet but aren't in the configured permissions list. If your application requires these permissions, you should consider adding them to the configured permissions list. [Learn more](#)

API / Permissions name	Type	Description	Admin consent requ...	Status
▼ Microsoft Graph (2) ...				
User.Read	Delegated	Sign in and read user profile	No	✓ Granted for Signet ...
User.Read.All	Delegated	Read all users' full profiles	Yes	✓ Granted for Signet ...

4. Create client secret

1. From **App registrations** in Azure AD, select your application.
2. Select **Certificates & secrets**.
3. Select **Client secrets -> New client secret**.
4. Provide a description of the secret, and a duration. When done, select **Add**.

After saving the client secret, the value of the client secret is displayed. Copy this value because you won't be able to retrieve the key later. You will provide the key value with the application ID to sign in as the application. Store the key value where your application can retrieve it.

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

DESCRIPTION	EXPIRES	VALUE
demo secret	5/14/2020	nWu9HVZ7Rnj.2y7XSkVyUngZ][x9Z:e 