

# Get app details and grant permissions to app registration

## Provision App Registration

App Registrations are containers that allow user-independent permission assignment and are therefore ideally suited for automation. App Registrations can be created in Azure Portal → Microsoft Entra ID → App Registrations. App Registrations should be preferred to service accounts whenever possible.

The following variables are used to authenticate to the Graph API using application permissions. The ClientSecret must not be stored as clear text in scripts or applications under any circumstances, but must be stored in designated containers (e.g. Azure Runbook Credential Store, Azure Key Vault or Windows Credential Store).

## Variables used

**\$TenantID:** This is the identity of the tenant, which is unique.

**\$ClientID:** The ClientID can be used to uniquely identify the App Registration.

**\$ClientSecret:** The ClientSecret expires every max. 24 months (2 years) and is like the password for the App Registration.

## Read ClientID & TenantID

The ClientID & TenantID can be read out on the start page of the App Registration itself.

Search << Delete Endpoints Preview features

**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

**Essentials**

Display name : AAP-RB-ALL-PERMISSION-ChangeHostnameByUPN-PROD

Application (client) ID : [REDACTED]

Object ID : [REDACTED]

Directory (tenant) ID : [REDACTED]

Supported account types : [My organization only](#)

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication

[Get Started](#) Documentation

The Microsoft identity platform is an authentic

## Create client secret

For the ClientSecret you have to switch to the "Certificates & Secrets" tab. A ClientSecret can be added via "New client secret". Then a name can be given there. **Attention:** Afterwards the ClientSecret is valid for 24 months (2 years) and expires after a certain time. In addition, the value is only displayed once. Save the ClientSecret as the first step in your password storage solution and note the expirationdate.

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

scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

Certificates (0) **Client secrets (2)** Federated credentials (0)

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	Secret ID
Test for Screenshot	12/2/2024	tr:8Q~mNaCIpaWDirwBSy6D0gbPNdmd...	e82c8ae9-60ba-4d32-9f57-0bf53b160d0f

## Grant permissions

As soon as the authentication via the app registration with the three values works, the permissions have to be assigned. This is a very tricky step, as an app could practically become a global administrator and overwrite or delete any tenant configurations.

The permissions on App Registrations for PowerShell Scripts must always be set to Application-Permissions, because the actions should be executed as App-Context.

## Find the required permissions

Microsoft Graph REST API v1.0 endpoint reference - Microsoft Graph v1.0 | Microsoft Learn

Version

Microsoft Graph REST API v1.0

Filter by title

API v1.0 reference

Overview

Users

Overview

User

User

List

Create

Get

Update

Delete

Change password

Get delta

App role assignment

Calendar

Delegated permission grant

Directory object

Drive

Group

Insights

Mail

Namespace: microsoft.graph

Create a new user.

Note

To create external users, use the invitation API.

Permissions

One of the following permissions is required to call this API. To learn more, including how to choose permissions, see [Permissions](#).

Permission type	Permissions (from least to most privileged)
Delegated (work or school account)	User.ReadWrite.All, Directory.ReadWrite.All
Delegated (personal Microsoft account)	Not supported.
Application	User.ReadWrite.All, Directory.ReadWrite.All

HTTP

POST /users

Copy

## Assign permissions

Once it has been determined which permissions need to be assigned, this can be done in Azure Portal → Microsoft Entra ID → App Registration → API permissions.

First, "Add a permission" must be selected and then Microsoft Graph as an API resource.

[illegible]

Then it is important that "Application permissions" is selected. There you can then search for the corresponding permission from the Microsoft Docs page and add it using "Add permissions".

## Request API permissions

[Add apps](#)
Microsoft Graph  
<https://graph.microsoft.com/> [docs](#) [github](#)

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](#)

**user** [expand all](#)

Permission	Admin consent required
<a href="#">Cross-TenantUserProfileViewing</a>	
<a href="#">IdentityTokenUser</a>	
<a href="#">IdentityUserFlow</a>	
<a href="#">TeamsAppInstallation</a>	
<a href="#">TeamsTab</a>	
<a href="#">User-LifeCycleInfo</a>	
<a href="#">UserAuthenticationMethod</a>	
<a href="#">UserNotification</a>	
<a href="#">UserShiftPreferences</a>	
<b>User (2)</b>	
<input type="checkbox"/> <a href="#">User.Export.All</a> <a href="#">?</a> Export user's data	Yes
<input type="checkbox"/> <a href="#">User.Invite.All</a> <a href="#">?</a> Invite guest users to the organization	Yes
<input type="checkbox"/> <a href="#">User.Manage.All</a> <a href="#">?</a> Manage all users' identities	Yes
<input checked="" type="checkbox"/> <a href="#">User.Read.All</a> <a href="#">?</a> Read all users' full profiles	Yes
<input type="checkbox"/> <a href="#">User.ReadBasic.All</a> <a href="#">?</a> Read all users' basic profiles	Yes
<input checked="" type="checkbox"/> <a href="#">User.ReadWrite.All</a> <a href="#">?</a> Read and write all users' full profiles	Yes

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

Now that the permission has been added thanks to the previous step, the last thing to do is to check it and then approve it. This can only be done with the "Global Administrator" role, as the app will then receive this permission forever and this is like assigning a role. The person who has Global Administrator must click on "Grand admin consent for <companyname>" in the app so that the permissions also become active.

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](#)

API / Permissions name	Type	Description	Admin consent requ...	Status	
▼ Microsoft Graph (3)					...
User.Read	Delegated	Sign in and read user profile	No	✔ Granted for 300	...
User.Read.All	Application	Read all users' full profiles	Yes	⚠ Not granted for 850	...
User.ReadWrite.All	Application	Read and write all users' full profiles	Yes	⚠ Not granted for 110	...

Revision #5

Updated 21 July 2024 15:11:16