

# 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 authent

## 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

The best way to find the required permissions is to visit the Microsoft Docs page for Graph API: [Microsoft Graph REST API v1.0 endpoint reference - Microsoft Graph v1.0 | Microsoft Learn](#)

Version: Microsoft Graph REST API v1.0

Namespace: microsoft.graph

Create a new **user**. The request body contains the user to create. At a minimum, you must specify the required properties for the user. You can optionally specify any other writable properties.

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 request

```
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.

Home > B2B | App registrations > AAP-6-ALL-SSO-AdventskalenderJumpNRun-NONPROD | API permissions

Search: Refresh Get feedback?

Integration assistant

Configure permissions

Applications are authorized to call APIs when they are granted permissions by administrators as part of the consent process. The list of configured permissions should include all the permissions the application needs to run from local permissions and consent.

Add a permission: Grant admin consent for API

API / Permission name	Type	Description	Admin consent req.	Status
*Microsoft Graph (1)	Delegated	Sign-in and read user profile	No	Consent for API

To view and manage consented permissions for individual apps, as well as your tenant's consent settings, go to [API permissions](#).

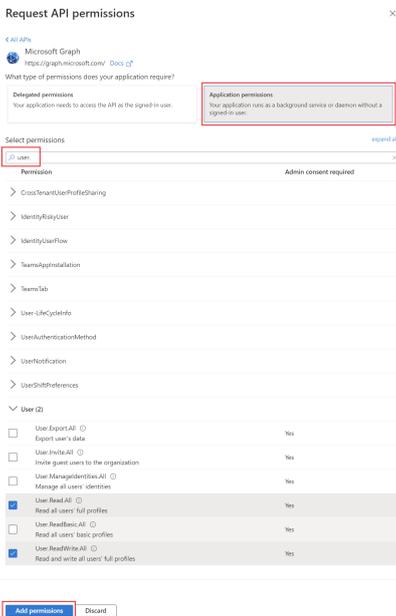
### Request API permissions

Select an API: Microsoft APIs (10) My organization users (1) My APIs

Commonly used Microsoft APIs

- Microsoft Graph**: Access to a wide range of information about users in Office 365, Enterprise Mobility + Security, and Windows 10. Includes contacts, tasks, items, OutlookExchange, OneDrive, OneNote, Planner, and more through a single endpoint.
- Access Bank**: Schedule large-scale transfer and iFCI applications to the bank.
- Access Communication Services**: Rich communication experiences with the same space, Chat, and more used by Microsoft Teams.
- Access Content DB**: For Azure Content DB, for any user.
- Access Data Catalog**: Provides a central location to store, catalog, and manage data across the organization.
- Access Data Explorer**: Query all the data in your organization's data stores and analyze it with a single interface.
- Access Data Lake**: Access to storage and compute for big data available via REST.
- Access DevOps**: Integrate with Azure DevOps and Azure DevOps API.
- Access IncidentResponse**: Programmatic control of incident response.
- Access Key Vault**: Manage keys, secrets, and certificates with the Azure Key Vault.
- Access Maps**: Create location-based apps and mobile applications using geospatial information, geospatial services, APIs, and SDKs in Azure.
- Access Partner**: Unified data governance service that helps manage and govern data on-premise, multi-cloud, and software or services-based data.
- Access Quantum**: Programmable access to create and manage qubits in Azure Quantum.
- Access Rights Management Service**: Allow content owners to restrict and protect content.
- Access Service Management**: Programmatic access to manage and dynamically enable through the Microsoft Graph.
- Access Storage**: Manage, monitor, analyze, report and data lake storage for structured and unstructured data.

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".



## Consent assigned permissions

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.

### 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 **User.Read.All**

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

Afterwards, the app can be used with the assigned permissions.

### Revision #5

Created 2 December 2022 11:41:39 by Luca Noah Caprez

Updated 6 December 2022 22:41:28 by Luca Noah Caprez