

# Microsoft Office 365 API Provider Setup

Last Modified on 12/30/2019 8:49 pm EST

## Overview

In order to authenticate an instance of the Microsoft Dynamics 365 connector, you must have the following:

- A Microsoft Azure Active Directory account
- A Microsoft Office 365 account
- A web application registered and configured with Azure Active Directory

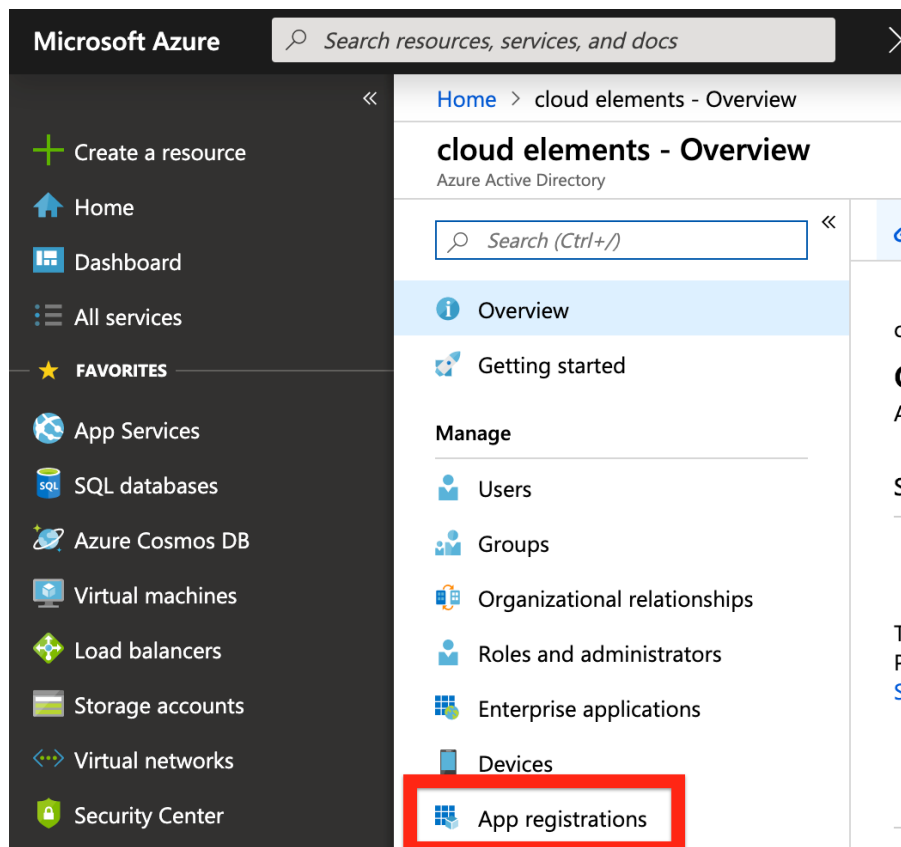
## Registering and Configuring a Web Application with Azure Active Directory

There are several required steps to registering and configuring a new web application in Azure AD. This section covers these steps, which include:

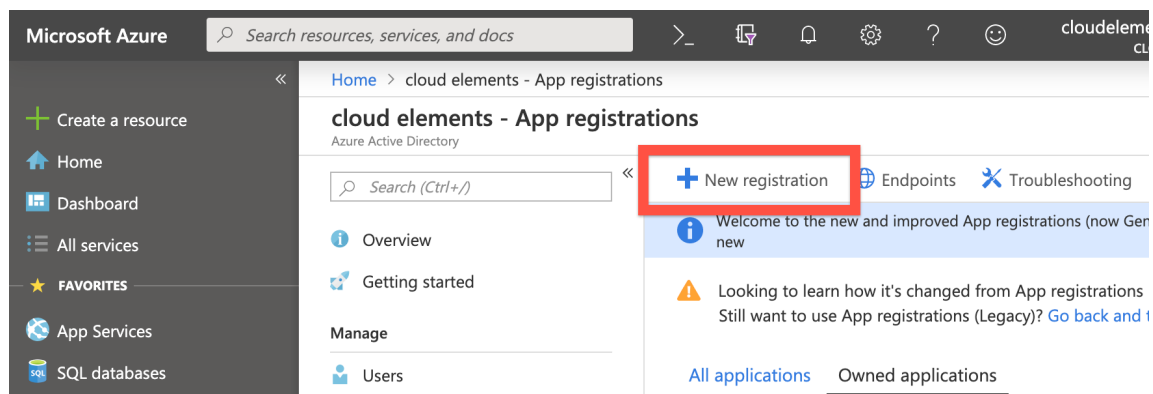
- Registering your application in the Azure Portal
- Recording the application ID
- Generating and recording the client secret
- Setting App permissions

## Registering your Application in the Azure Portal

1. In a web browser, navigate to the [Azure Portal](#) and sign in using your Azure Active Directory credentials.
2. From the left-hand navigation toolbar, select **Azure Active Directory** and then select **App registrations**.



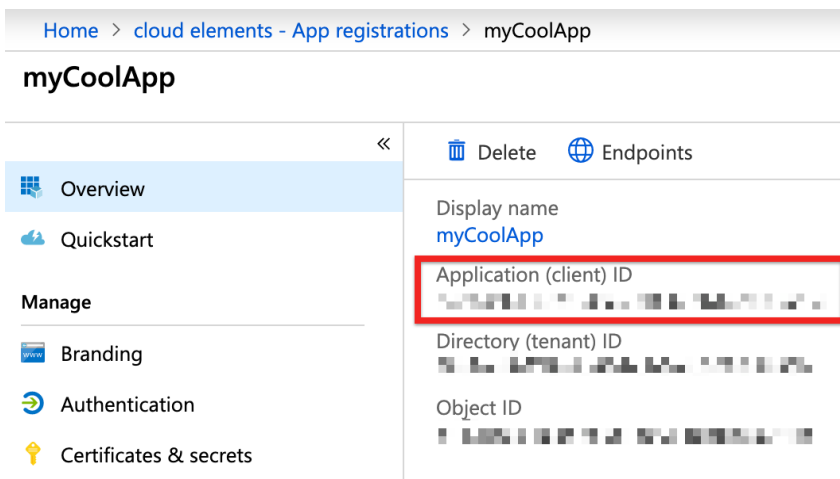
3. Select **New registration**.



4. On the Register an application screen, enter a name for your application in the Name field, and then click Register; the supported account types and redirect URI will be configured later during this process. For more information on the application registration process, see Microsoft's [documentation](#).

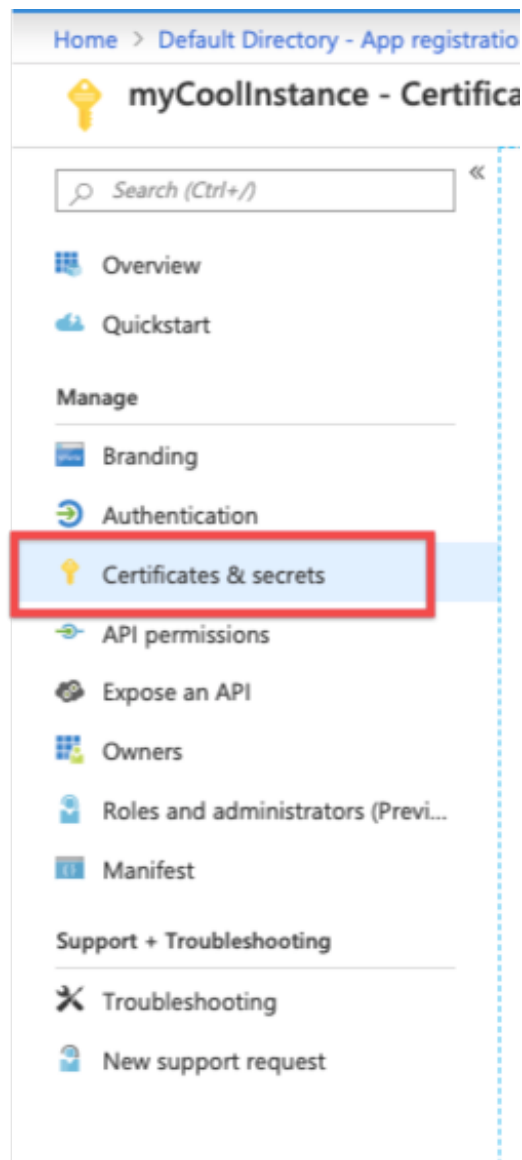
## Recording the Application ID

After registering your new application, record the Application (client) ID as displayed in your application's overview page. You will use this string as the API key when you authenticate a connector instance.



## Generate and Record the Client Secret

1. Click Certificates & secrets.



2. On the Certificates & secrets page, click the New client secret button.

Home > Default Directory - App registrations > myCoolInstance - Certificates & secrets

## myCoolInstance - Certificates & secrets

Search (Ctrl+ /)

- Overview
- Quickstart
- Manage
  - Branding
  - Authentication
  - Certificates & secrets**
  - API permissions
  - Expose an API
  - Owners
  - Roles and administrators (Previ...
  - Manifest
- Support + Troubleshooting
  - Troubleshooting
  - New support request

Credentials enable applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

### Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

[Upload certificate](#)

No certificates have been added for this application.

THUMBPRINT	START DATE	EXPIRES
------------	------------	---------

### 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
-------------	---------	-------

No client secrets have been created for this application.

3. In the Add a client secret window, add a name to the Description field, select Never, and then click Add.

### Add a client secret

Description

myCoolSecret

Expires

☐ In 1 year

☐ In 2 years

☒ Never

[Add](#) [Cancel](#)

4. Under Client secrets, record the value for your newly created client secret. This value is your client secret, which you will use as the API secret when you authenticate a connector instance.

myCoolInstance - Certificates & secrets

Search (Ctrl+*/*)

Overview

Quickstart

Manage

Branding

Authentication

**Certificates & secrets**

API permissions

Expose an API

Owners

Roles and administrators (Previ...

Manifest

Support + Troubleshooting

Troubleshooting

New support request

Copy the new client secret value. You won't be able to retrieve it after you leave this blade.

Credentials enable applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

**Certificates**

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Upload certificate

No certificates have been added for this application.

THUMBPRINT	START DATE	EXPIRES
------------	------------	---------

**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
myCoolSecret	12/31/2299	[REDACTED]

## Setting App Permissions

After exposing your APIs, you need to set app permissions depending on your requirements.

On the navigation panel to your left, click **API permissions** and then **Add a permission**, as shown in the picture below.

Home > App registrations > CloudElementsV2 - API permissions

CloudElementsV2 - API permissions

Search (Ctrl+*/*)

Overview

Quickstart

Manage

Branding

Authentication

**Certificates & secrets**

API permissions

Expose an API

Owners

Roles and administrators (Previ...

Manifest

Support + Troubleshooting

Troubleshooting

New support request

API permissions

Applications are authorized to use APIs by requesting permissions. These permissions show up during the consent process where users are given the opportunity to grant/deny access.

Add a permission

API / PERMISSIONS NAME	TYPE	DESCRIPTION	ADMIN CONSENT REQUIRED
Azure Active Directory Graph (8)			
Application.ReadWrite.All	Application	Read and write all applications	Yes <span>Granted for Default ...</span>
Device.ReadWrite.All	Application	Read and write devices	Yes <span>Granted for Default ...</span>
Directory.ReadWrite.All	Delegated	Read and write directory data	Yes <span>Granted for Default ...</span>
Directory.ReadWrite.All	Application	Read and write directory data	Yes <span>Granted for Default ...</span>
Domain.ReadWrite.All	Application	Read and write domains	Yes <span>Granted for Default ...</span>
Group.ReadWrite.All	Delegated	Read and write all groups	Yes <span>Granted for Default ...</span>
Member.Read.Hidden	Application	Read all hidden memberships	Yes <span>Granted for Default ...</span>
User.ReadBasic.All	Delegated	Read all users' basic profiles	- <span>Granted for Default ...</span>
Microsoft Graph (20)			
Calendars.ReadWrite	Delegated	Have full access to user calendars	- <span>Granted for Default ...</span>