

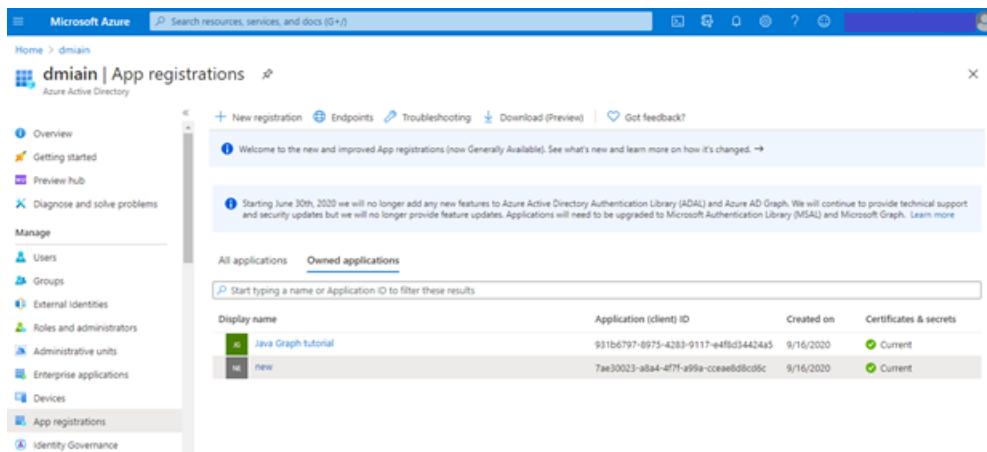
Get an Access Token for Microsoft Graph API

1. Create an **Azure AD Application** in your tenant.
2. Allow some permissions to the application for accessing **Microsoft Graph**.
3. Using an admin account **consent** on behalf of their organization.
4. Create a password (a key) for the app.

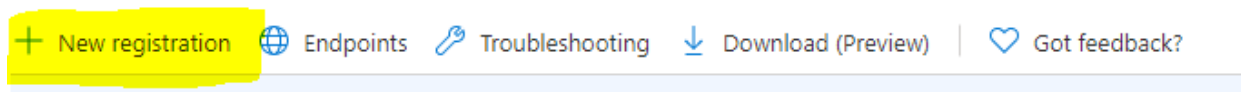
1.Create an Azure AD Application in your tenant.

1. Login to the <https://aad.portal.azure.com>
2. Registering the Application in the Azure Portal

In the left side pane click the label “Azure Active Directory” This will open up the blade for Azure Active Directory. In that screen should see a label “App registrations”. This is the starting point of a registering an Azure AD Application.



A button on the top “New registration”. Let’s click on that button to create a new application.



Now we will see the new application registration blade.

Microsoft Azure Search resources, services, and docs (G+)

Home > dmian >

Register an application

* Name

The user-facing display name for this application (this can be changed later).

Graph Teams

Supported account types

Who can use this application or access this API?

☐ Accounts in this organizational directory only (dmian only - Single tenant)
☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant)
☒ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
☐ Personal Microsoft accounts only

[Help me choose...](#)

By proceeding, you agree to the Microsoft Platform Policies

Register

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and changed later, but a value is required for most authentication scenarios.

Web http://localhost:8081

By proceeding, you agree to the Microsoft Platform Policies

Register

We have to note down few things here. We will be using these to build the application.

Application (client) ID → The id of your application
Directory (tenant) ID → The Azure AD tenant id

Home > dmian > new

new

Search (Ctrl+/)

Overview Quickstart Integration assistant | Preview

Manage Branding Authentication

Delete Endpoints

Got a second? We would love your feedback on Microsoft identity platform (previously Azur

Essentials

Display name : new

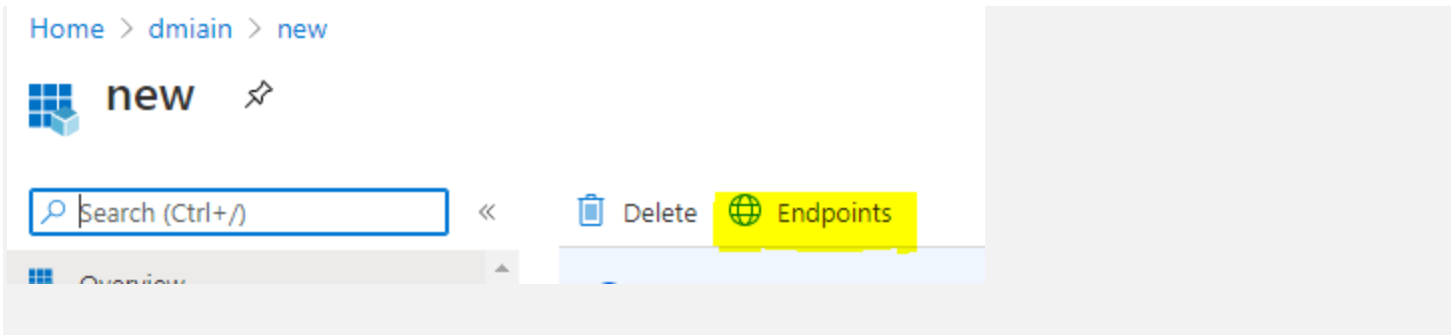
Application (client) ID : 7ae3002 -cceae8d8cd6c

Directory (tenant) ID : 1045962 8-cc750eeef4c8

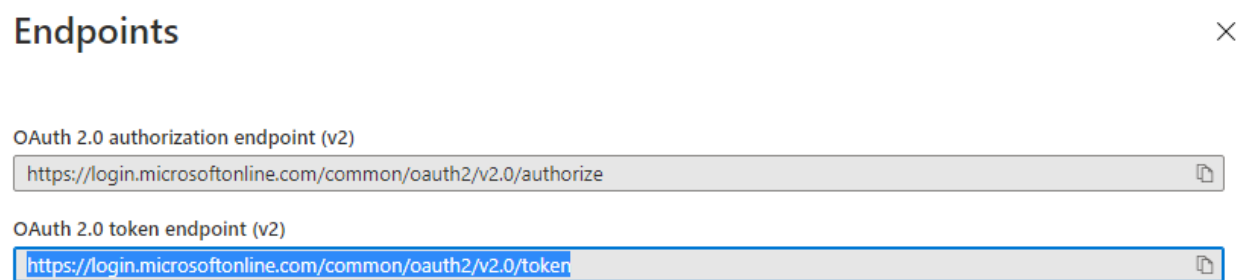
Object ID : 300fa26 7-91f3448a2e53

Next step is to get the token endpoint. This end point will generate the token for you. Generated token from this endpoint will be used to access Microsoft Graph API calls.

Click on the "Endpoints" button on the top of the screen.



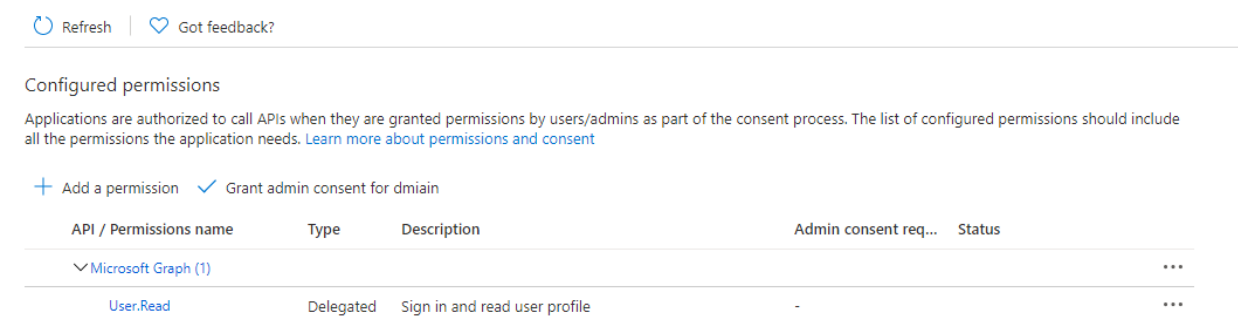
This will get all the endpoints for your application. Make sure you copy down the endpoint for **OAuth 2.0 token endpoint (v2)**



2. Allowing permissions for Microsoft Graph API

In the “Registered app” application blade, click on the “API permissions” label.

Azure has already given “User. Read” delegated permissions for the application. This permission will allow us to read user information for a logged in user. These are Microsoft Graph API permissions, in other hand we can call them as “Scopes”.



As mentioned before there are two methods of permission types can be used with an Azure AD application.

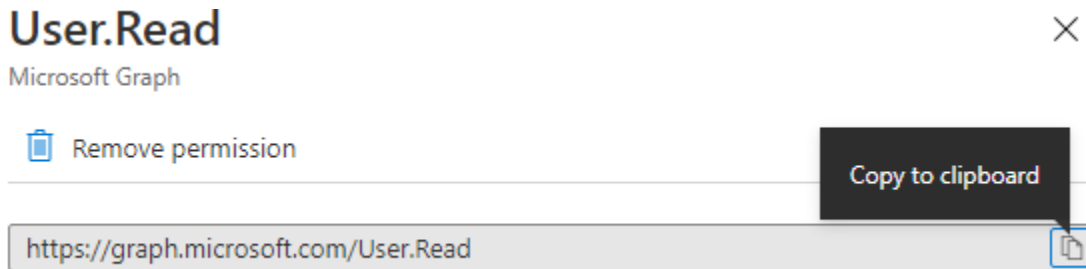
1. Delegated

we will use this application on behalf of a user. For an example, if we logged in using my Work or School account, we are allowing this application to use my credentials on behalf of user.

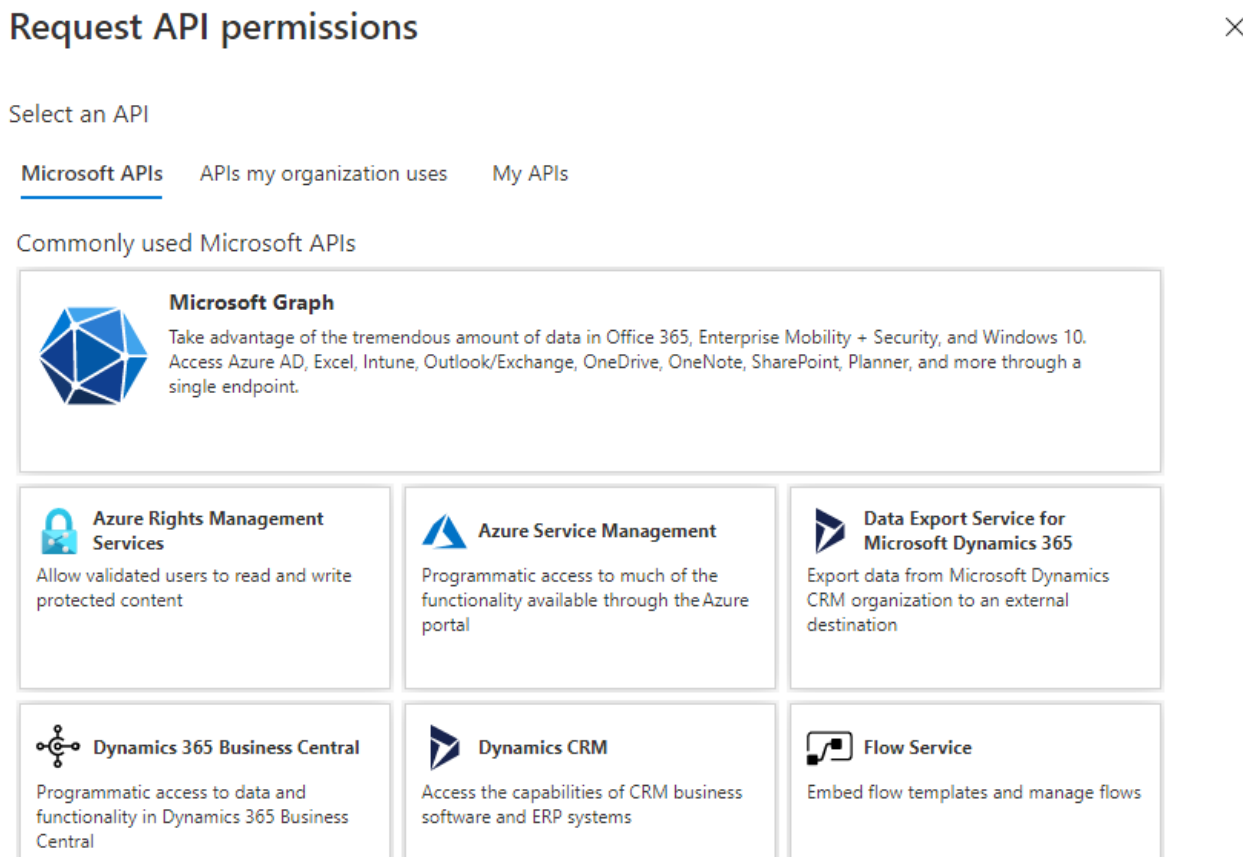
2. Application

We give the permission to this specific app. No user is required.

“Application” permissions are required for the app we registered. Select on the available “User. Read” permission and delete it.




Now, click on “+ Add a permission” button and select “Microsoft Graph”



Click on “Application permissions”. Now the see a list of permissions available for Microsoft Graph API.

< 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

Start typing a reply url to filter these results	
Permission	Admin consent required
> AccessReview	
> AdministrativeUnit	
> Application	

Select calendar permission & one or more (depending on need), click “Add permissions” button.

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 (2)	
<input checked="" type="checkbox"/> Calendars.Read ⓘ Read calendars in all mailboxes	Yes
<input checked="" type="checkbox"/> Calendars.ReadWrite ⓘ Read and write calendars in all mailboxes	Yes

Add permissions

Discard

3. Grant admin consent

Refresh

Got feedback?

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

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 dmiain

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (2)				...
Calendars.Read	Application	Read calendars in all mailboxes	Yes	⚠ Not granted for dmiain ...
Calendars.ReadWrite	Application	Read and write calendars in all mailboxes	Yes	⚠ Not granted for dmiain ...

an admin of the organization must allow this application to access the selected permission on behalf of the users.

Now click “Grant admin consent for <orgname>” button.

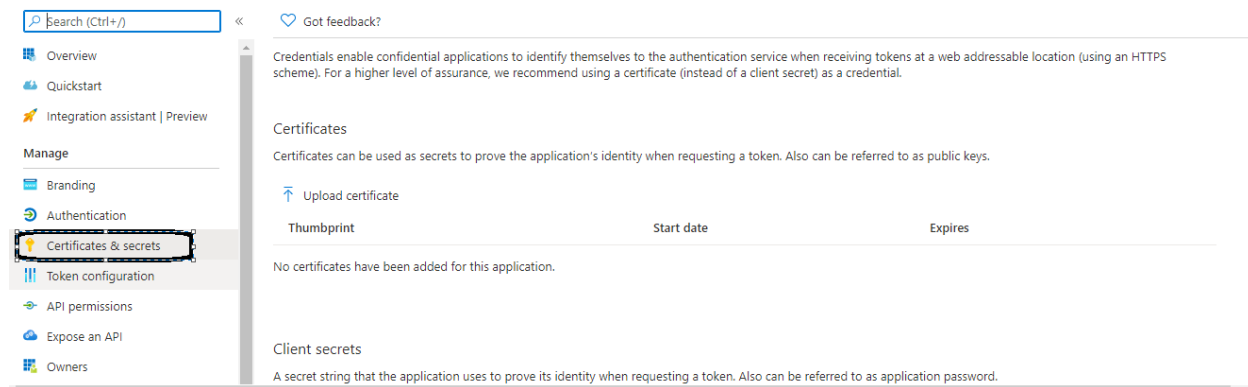
+ Add a permission

✓ Grant admin consent for dmiain

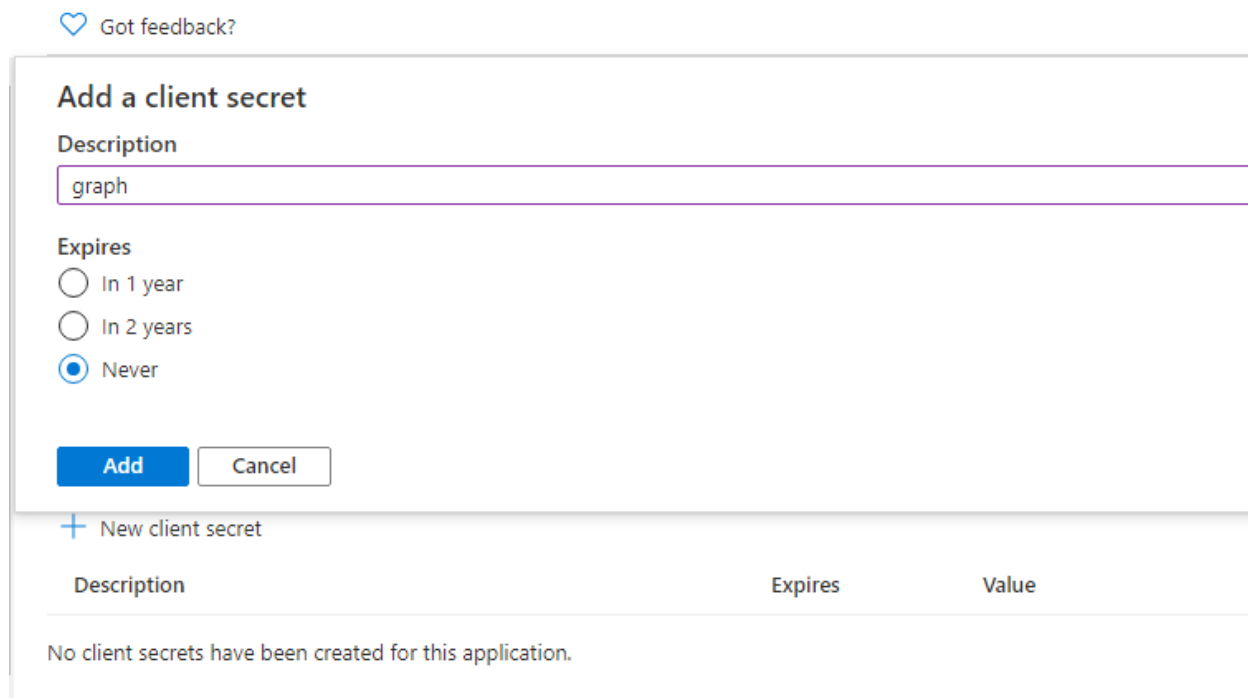
API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (2)				...
Calendars.Read	Application	Read calendars in all mailboxes	Yes	⚠ Not granted for dmiain ...
Calendars.ReadWrite	Application	Read and write calendars in all mailboxes	Yes	⚠ Not granted for dmiain ...

4. Create a key (password) for the application

Create a key for the application. Since we are not going to interact with any of the users. We need this key. Let's create one. Click on the "Certificates & secrets"



Now click on the "+ New client secret" and give a name and select an expiration period.



And click "Add", and make sure you have copied the key down. When we go away from this screen. Azure doesn't allow to see this key again.

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		
graph	12/31/2299	Q33C [REDACTED] YV5g4A~-~se4rr4N~D~f		

Spring Boot Configuration:

In Application.properties file replace the below value

Please refer 1st section to obtain this value

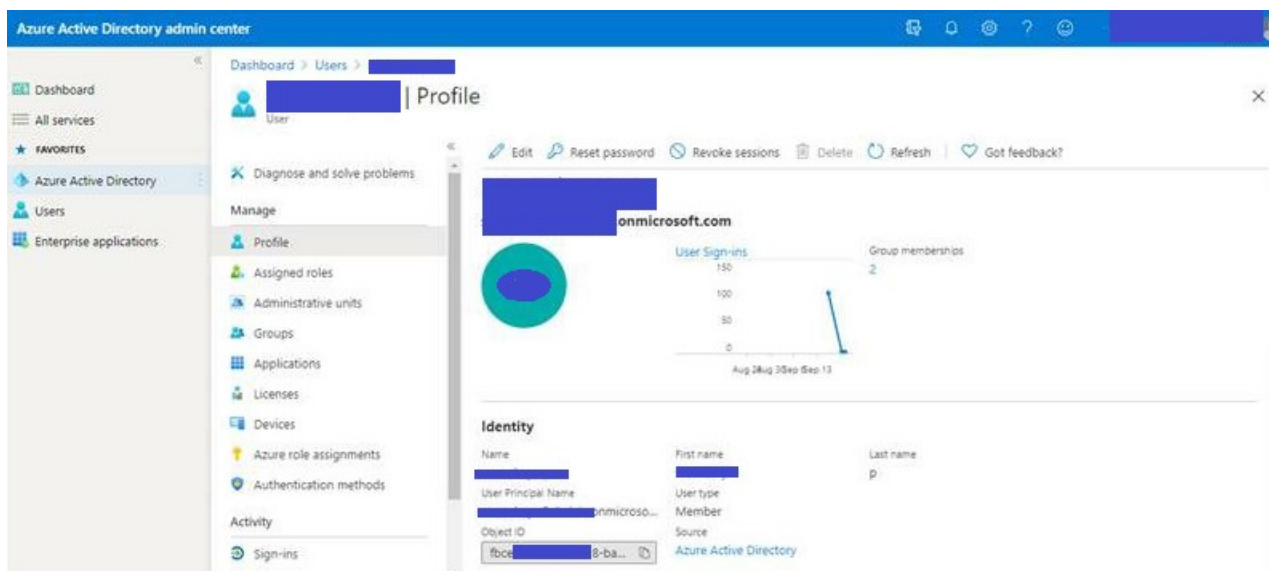
```
teams.client_id = Application (Client) id
teams.tenant_id = Directory (Tenant) id
```

Please refer 4th section to obtain this value

```
teams.client_secret = refer client secret image
teams.scope =https://graph.microsoft.com/.default
teams.grant_type =client_credentials
```

Graph URLs for applications:

All URLs will need to have endpoints like user/object id to access teams using application token and if you are using URL endpoints with /me then you will end up in some errors

The screenshot shows the Azure Active Directory admin center interface. On the left is a navigation pane with options like Dashboard, All services, Favorites, Azure Active Directory, Users, and Enterprise applications. The main area displays the 'Profile' of a user. At the top, there's a header with the user's name and a 'Profile' tab. Below this, there are action buttons: Edit, Reset password, Revoke sessions, Delete, Refresh, and Got feedback?. The profile section includes a circular profile picture, a domain 'onmicrosoft.com', and two charts: 'User Sign-ins' (a line graph showing activity from Aug 28 to Sep 13) and 'Group memberships' (a bar chart showing 2 memberships). Below the charts is an 'Identity' section with fields for Name, First name, Last name, User Principal Name, User type, Member, Object ID, Source, and Azure Active Directory.

<https://graph.microsoft.com/v1.0/users/{object id or userPrincipalName}/>

In Azure portal, if you open your profile you can find both object id and userPrincipalName. Use can use either of then to call your API