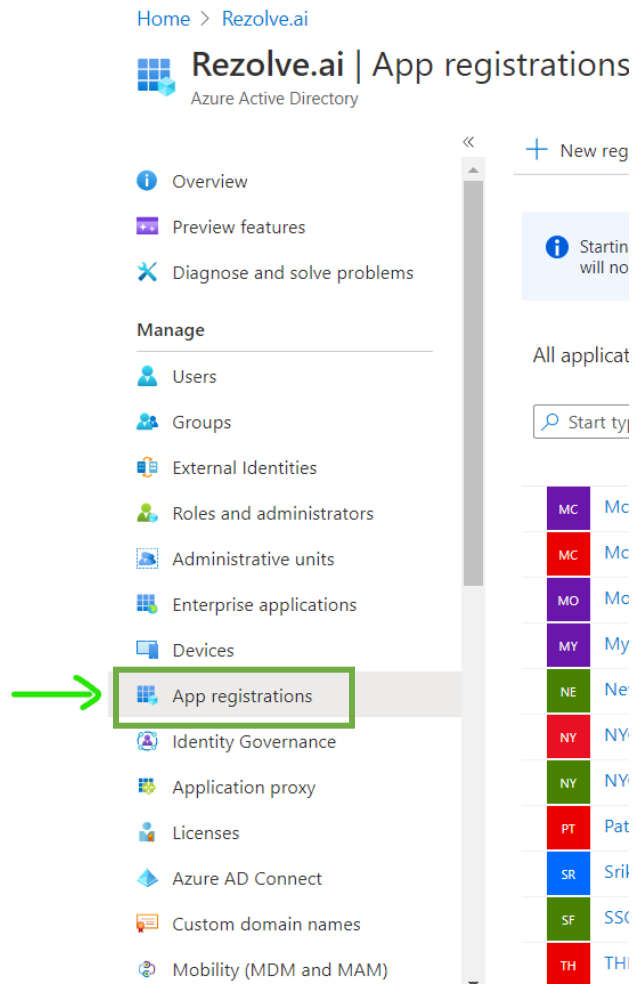


Enabling Rezolve.ai Agent in Teams App

Prerequisite

Part A: Granting API Permissions

1. Go to Azure AD Admin Portal and select *App Registration*



2. Select your existing Rezolve app from the list if you have one already setup. If you don't have one, create a new APP called Rezolve or the Bot name.
3. Once the app configuration screen opens, select *API permissions* from the menu on the left

Search (Ctrl+/) Refresh Got feedback?

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

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 con... all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission ✓ Grant admin consent for Rezolve.ai

API / Permissions name	Type	Description	Admin consent requ...
Microsoft Graph (10)			
ChannelMember.ReadWrite.All	Application	Add and remove members from all channels	Yes
Directory.Read.All	Application	Read directory data	Yes
email	Delegated	View users' email address	No
GroupMember.ReadWrite.All	Application	Read and write all group memberships	Yes
offline_access	Delegated	Maintain access to data you have given it access to	No
openid	Delegated	Sign users in	No
profile	Delegated	View users' basic profile	No
Team.Create	Application	Create teams	Yes
TeamMember.ReadWrite.All	Application	Add and remove members from all teams	Yes
User.Read.All	Application	Read all users' full profiles	Yes

4. Click on *Add a permission* and then select *Microsoft Graph*

LTDSTAGINGREZIDP

GREZIDP | API permissions

Refresh Got feedback?

The "Admin consent required" column shows the default value for organizations where this app will be used. [Learn more](#)

Configured permissions

Applications are authorized to call APIs when they are granted permis... all the permissions the application needs. [Learn more about permis...](#)

+ Add a permission ✓ Grant admin consent for Default Direct...

API / Permissions name	Type	Description
Microsoft Graph (2)		
ChannelMember.ReadWrite.All	Application	Add and remove...
User.Read	Delegated	Sign in and read...

To view and manage permissions and user consent, try [Enterprise app...](#)

Request API permissions

Select an API

Microsoft APIs | APIs my organization uses | My APIs

Commonly used Microsoft APIs

- Microsoft Graph**
Take advantage of the tremendous amount of data in Office 365, Enterprise Mob... Access Azure AD, Excel, Intune, Outlook/Exchange, OneDrive, OneNote, SharePoi... single endpoint.
- Azure Cosmos DB**
Fast NoSQL database with open APIs for any scale.
- Azure DevOps**
Integrate with Azure DevOps and Azure DevOps server
- Azure Service Management**
Programmatic access to much of the functionality available through the Azure portal.
- Azure Storage**
Secure, massively scalable object and data lake storage for unstructured and semi-structured data.

5. Now you will need to give the app 4 Delegate permissions and 6 Application permissions

Request API permissions



[< All APIs](#)



Microsoft Graph

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

What type of permissions does your application require?

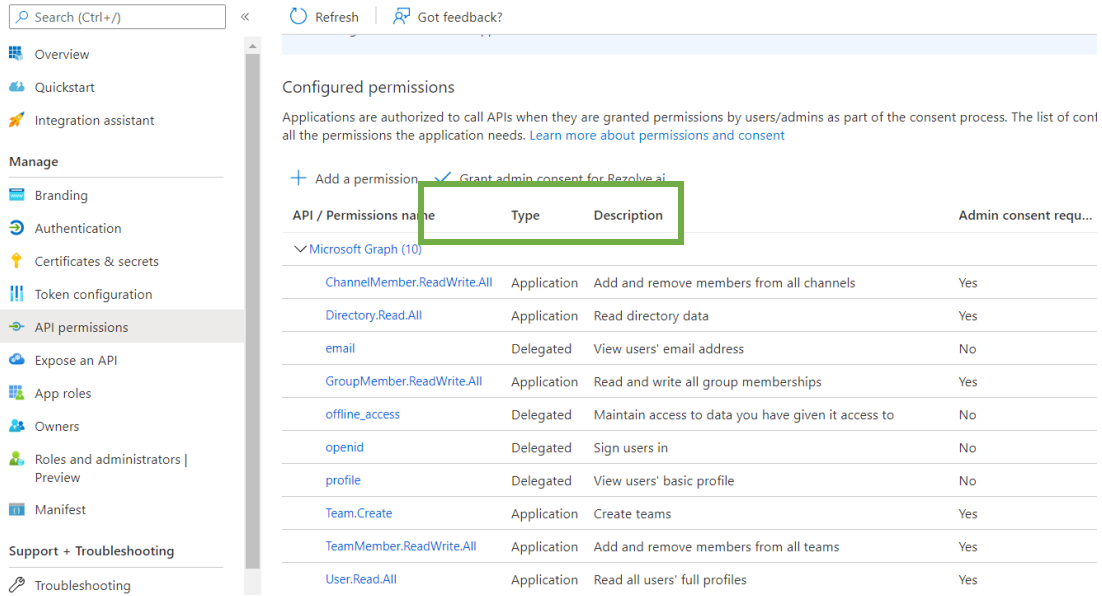
Delegated permissions

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.

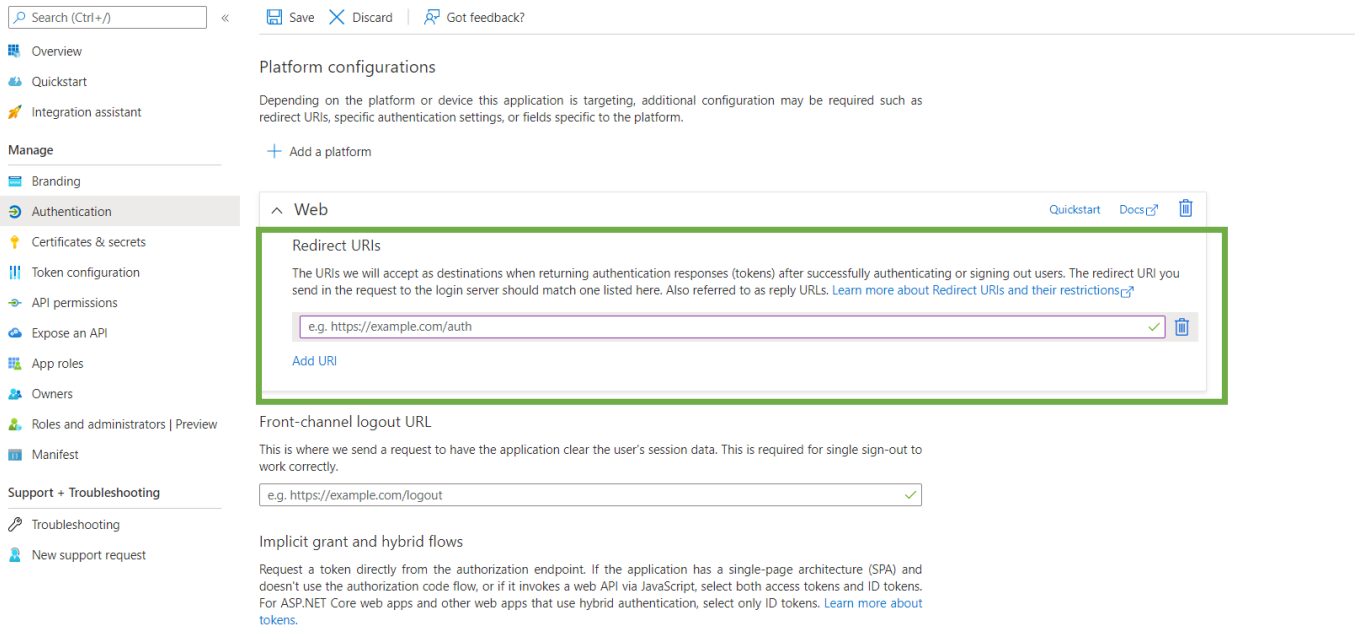
- a. Select *Delegate permissions*
- b. Scroll through the list, select the 4 items listed below and click Add
 - View users' email address
 - offline_access
 - openid
 - profile
- c. Click *Save* (found at the top)
- d. Once added, you will be taken back to the API permissions screen and will need to click *Add a permission* and then select *Microsoft Graph* again
- e. This time select *Application permissions*
- f. Scroll through the list, select the 6 items listed below and click Add.
 - ChannelMember.ReadWrite.All
 - Directory.Read.All
 - GroupMember.ReadWrite.All
 - Team.Create
 - TeamMember.ReadWrite.All
 - User.Read.All
- g. Click *Save* (found at the top)
- h. Click on the *Grant admin consent for...*



Your App Permission screen should look like this:

Part B: Configuring Authentication

1. With the app configuration screen already open, click on *Authentication* from the left-hand menu
2. On the Platform configurations page, add the 2 URLs provided by the ACS team under “Redirect URIs”



3. Check both boxes as shown under “Implicit grant and hybrid flows”

Search (Ctrl+/) << Save Discard Got feedback?

Front-channel logout URL
 This is where we send a request to have the application clear the user's session data. This is required for single sign-out to work correctly.
 e.g. https://example.com/logout

Manage

- Branding
- Authentication**
- Certificates & secrets
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators | Preview
- Manifest

Support + Troubleshooting

- Troubleshooting

Implicit grant and hybrid flows
 Request a token directly from the authorization endpoint. If the application has a single-page architecture (SPA) and doesn't use the authorization code flow, or if it invokes a web API via JavaScript, select both access tokens and ID tokens. For ASP.NET Core web apps and other web apps that use hybrid authentication, select only ID tokens. [Learn more about tokens.](#)

Select the tokens you would like to be issued by the authorization endpoint:

- Access tokens (used for implicit flows)
- ID tokens (used for implicit and hybrid flows)

Supported account types
 Who can use this application or access this API?

- Accounts in this organizational directory only (Default Directory only - Single tenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)

[Help me decide...](#)

Save

4. Click Save

Part C: Generate New Client Secret.

1. With the app configuration screen already open, select *Certificates and Secrets* from the left-hand menu

Search (Ctrl+/) << Got feedback?

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

Got a second to give us some feedback? →

Credentials enable confidential 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.

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

Certificates (0) **Client secrets (1)** 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
ResolveBot	6/14/2022	sq.7Q~2z6p19SWjhbE79zam8UQv3aOS...	fa61a94d-3302-404d-b48e-b44e9120e7

2. Click + *New client secret*

Search (Ctrl+/) << Got feedback?

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

Got a second to give us some feedback? →

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

Certificates (0) **Client secrets (1)** 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
ResolveBot	6/14/2022	sq.7Q~2z6p19SWjBhE79zam8UQv3aOS...	fa61a94d-3302-404d-b48e-b44e9120ebe7

3. Enter a description i.e. *ResolveAIBot* and click Add

Add a client secret

Description:

Expires:

4. Copy the *Value* and *Secret ID*, then paste it somewhere safe

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	Secret ID
ResolveAIBot	2/19/2022	5-bT3VNF.DUw34vOZdkU.mRXi_rZT-8o.5	ffb53ef2-c357-4bd8-b1fd-c9f9bb75416d

Note: In the original image, the Value and Secret ID cells are highlighted with green boxes and green arrows pointing to them, indicating they should be copied.

5. Select Overview in the left hand side menu
6. Copy the *Application (client) ID*

The screenshot shows the Azure portal interface for an application named "GraphAPITestSP". On the left-hand side, there is a navigation menu with "Overview" selected and highlighted by a green box and a red arrow. The main content area displays the "Essentials" section for the application. The "Application (client) ID" is highlighted with a green box and is e26b18c7-1b5a-4ba9-beef-1d4b231e9177. Other visible details include Object ID (d3f6bb25-85b4-402c-9e93-18b249fcb71) and Directory (tenant) ID (613cca5d-dd2e-4473-ab01-e1193e798037). The right-hand side shows client credentials (0 certificate, 1 secret), redirect URIs, and application ID URIs. Below the essentials, there are informational messages and links for "Get Started" and "Documentation".

Build your application with the Microsoft identity platform

The Microsoft identity platform is an authentication service, open-source libraries, and application management tools. You can create or standard-based authentication solutions, access and protect APIs, and add sign-in for your users and customers. [Learn more?](#)

Share the Value, Client Secret and Client ID, and SAML-P sign-on endpoint (Directory (tenant) ID) with ACS.