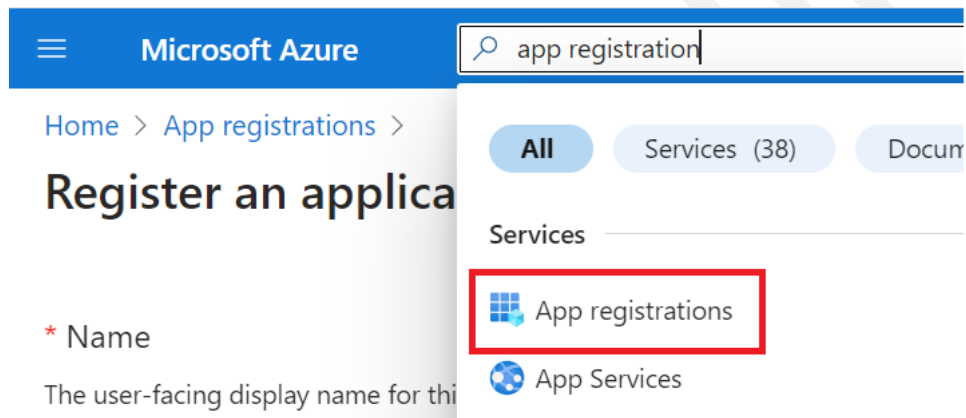


Get all Active Directory users in your organization

Go to app registrations in Azure

Create a new app



The screenshot shows the Microsoft Azure portal interface. At the top, there is a blue header with the Microsoft Azure logo and a search bar containing the text 'app registration'. Below the header, the breadcrumb navigation shows 'Home > App registrations >'. The main heading is 'Register an applica'. Below this, there is a form with a label '* Name' and a description 'The user-facing display name for thi'. On the right side, there is a search results dropdown menu. The menu has three tabs: 'All', 'Services (38)', and 'Docum'. Under the 'Services' section, there are two items: 'App registrations' (highlighted with a red box) and 'App Services'.

Go to tab “Manage”

Go to subtab “API permissions”

Click on “Add a permission”

Click on “Microsoft Graph”

Add the Application permission “User.Read.All”

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and the Copilot icon. The breadcrumb trail is Home > App registrations > app_entra_users. The main heading is app_entra_users | API permissions. The left-hand navigation pane is expanded to the 'Manage' section, with 'API permissions' highlighted. The main content area displays a warning about granting tenant-wide consent and a message about admin consent. Below this, the 'Configured permissions' section shows a table with one entry: 'Microsoft Graph (1)' containing the 'User.Read' permission. A '+ Add a permission' button is visible. On the right, the 'Request API permissions' pane is open, showing a list of permissions under the 'User (1)' category. The 'User.Read.All' permission, described as 'Read all users' full profiles', is selected with a blue checkmark. At the bottom of this pane are 'Add permissions' and 'Discard' buttons.

API / Permissions name	Type	Description
Microsoft Graph (1)		
User.Read	Delegated	Sign in and read user profile information for signed-in users.

Permission	Description
User-Phone	
UserAuthenticationMethod	
UserNotification	
UserShiftPreferences	
UserTeamwork	
User (1)	
<input type="checkbox"/> User.DeleteRestore.All	Delete and restore all users
<input type="checkbox"/> User.EnableDisableAccount.All	Enable and disable user accounts
<input type="checkbox"/> User.Export.All	Export user's data
<input type="checkbox"/> User.Invite.All	Invite guest users to the organization
<input type="checkbox"/> User.ManageIdentities.All	Manage all users' identities
<input checked="" type="checkbox"/> User.Read.All	Read all users' full profiles
<input type="checkbox"/> User.ReadBasic.All	Read all users' basic profiles

Grant admin consent

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

API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (2)				...
User.Read	Delegated	Sign in and read user profile	No	...
User.Read.All	Application	Read all users' full profiles	Yes	⚠ Not granted for ...

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

Go to Certificates and secrets

Click on New client secret

The screenshot shows the Microsoft Azure portal interface. At the top, there is a blue header with the Microsoft Azure logo. Below the header, the breadcrumb navigation reads 'Home > App registrations > app_entra_users'. The main heading is 'app_entra_users | Certificates & secrets'. A search bar and a 'Got feedback?' link are visible. On the left, a navigation menu lists various options: Overview, Quickstart, Integration assistant, Diagnose and solve problems, Manage, Branding & properties, Authentication, Certificates & secrets (highlighted), Token configuration, API permissions, Expose an API, and App roles. The main content area shows a message: 'Application registration certificates, secrets and federated credentials can be found in the application registration page.' Below this, there are three tabs: 'Certificates (0)', 'Client secrets (0)' (selected), and 'Federated credentials (0)'. A description states: 'A secret string that the application uses to prove its identity when requesting a token.' A red box highlights the '+ New client secret' button. Below the button is a table with columns for 'Description', 'Expires', and 'Value'. The table is currently empty, with a message below it: 'No client secrets have been created for this application.'

Copy the value of the secret right after creation

Save it somewhere safe

Branding & properties

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

App roles

Owners

Roles and administrators

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
secret_app_entra_users	3/26/2025		

Copy the Application (client) ID


Copy the Directory (tenant) ID


[Home](#) > [App registrations](#) >


 **app_entra_users**  

 Delete  Endpoints  Preview features


 Overview

 Quickstart

 Integration assistant

 Diagnose and solve problems

▼ Manage

 Branding & properties

^ Essentials

Display name : [app_entra_users](#)

Application (client) ID :

Object ID :

Directory (tenant) ID :

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

For testing purposes create auth.py and create the following variables with the values that you copied earlier

Save auth.py in the same folder where the “API call” script (defined later in this document) is saved

```
tenant_id = "your tenant id"  
client_id = "your client id"  
client_secret = "your client secret"
```

Note: this is not a safe way to save authentication data so be sure to use a service that is designed to save this type of data like Azure Key Vault or AWS Key Management Service

Use the script below to pick up the AD users (script below is not an image and can be copied)

```
import requests
import json
import auth

# Get an access token from Microsoft Identity platform
tenant_id = auth.tenant_id
client_id = auth.client_id
client_secret = auth.client_secret
authority_url = f"https://login.microsoftonline.com/{tenant_id}/oauth2/v2.0/token"

body = {
    "grant_type": "client_credentials",
    "client_id": client_id,
    "client_secret": client_secret,
    "scope": "https://graph.microsoft.com/.default",
}

token_response = requests.post(authority_url, data=body)
token = token_response.json().get('access_token')

# Use the token to make an API request
headers = {
    "Authorization": f"Bearer {token}",
    "Content-Type": "application/json"
}

# API call to get all users
url = "https://graph.microsoft.com/v1.0/users"
response = requests.get(url, headers=headers)

# Print the result
users = response.json()
print(json.dumps(users, indent=4))
```