Azure App Registration for Active Directory User Import

In order to be able to import users into Novus from an Azure Active Directory an Application must be created in Azure.

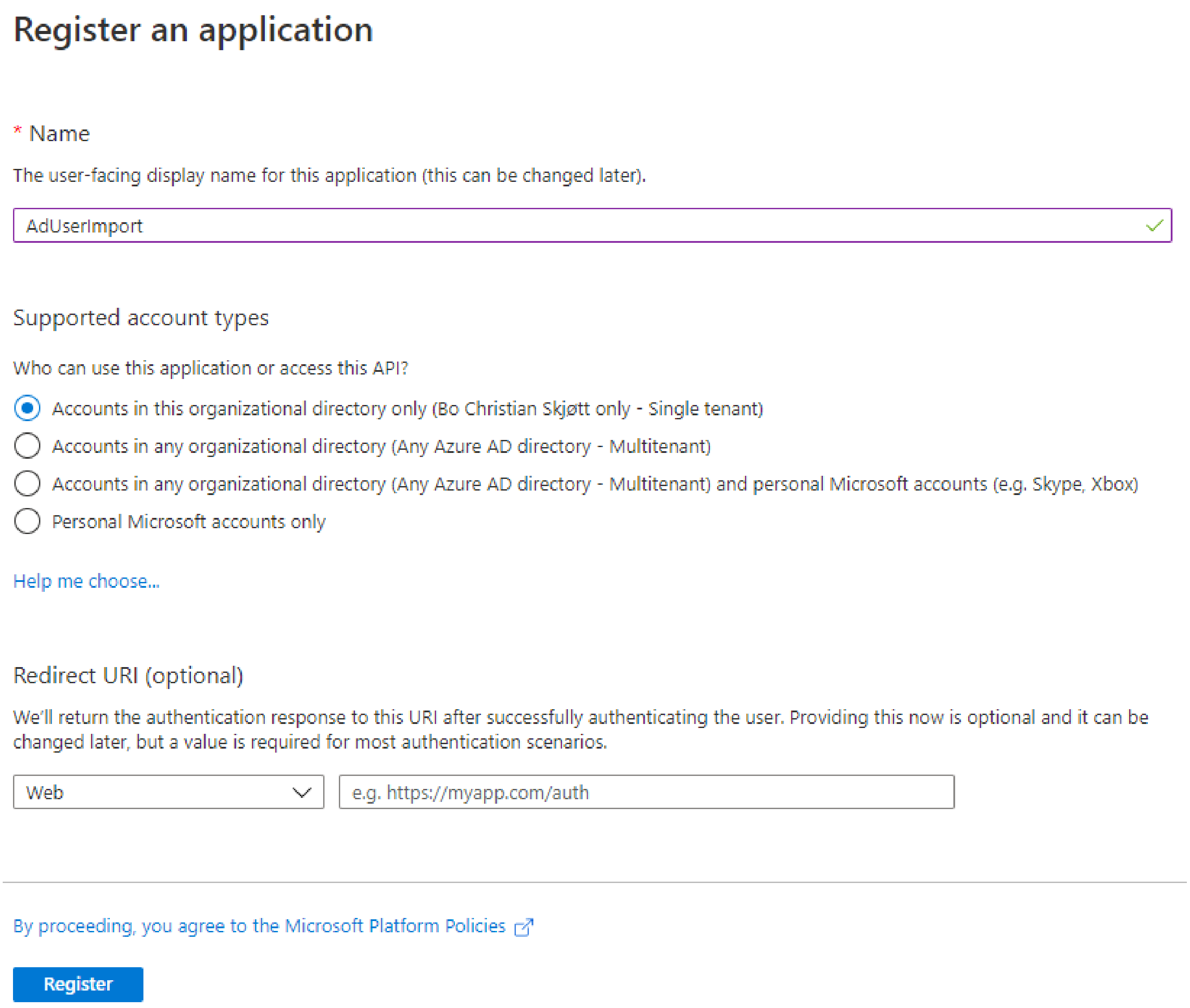
App creation in Azure portal

Go to your Active Directory in the Azure portal and follow the steps below.

Step 1. Register an App

Click App registrations, New registration.

Azure Screenshot



Register an application

* Name

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

AdUserImport

Supported account types

Who can use this application or access this API?

● Accounts in this organizational directory only (Bo Christian Skjøtt 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...

Redirect URI (optional)

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

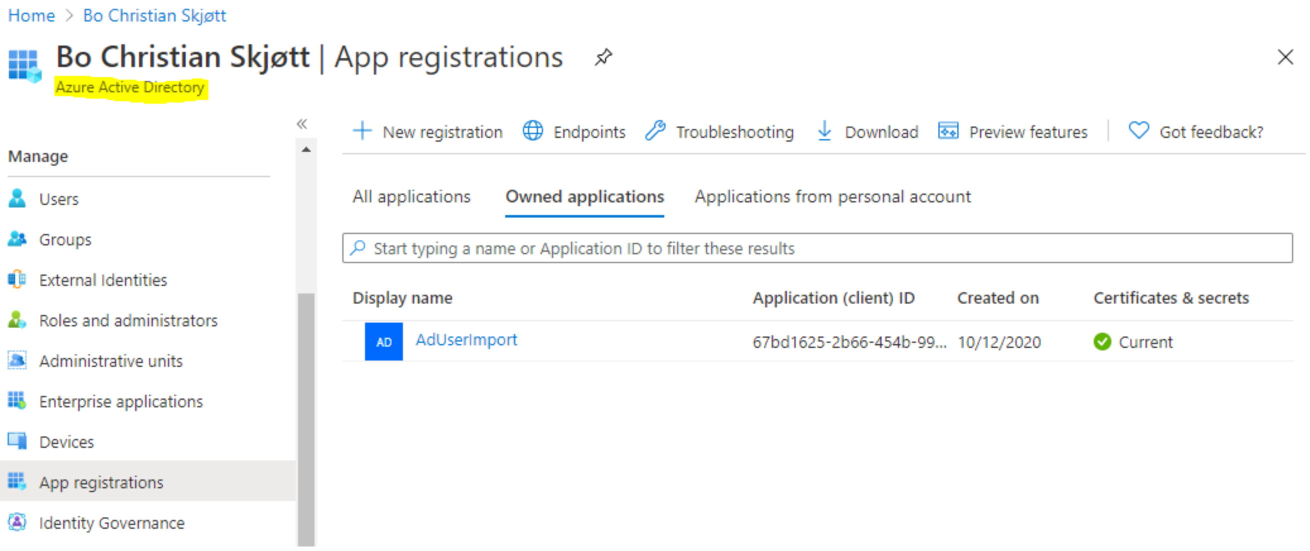
Web | e.g. https://myapp.com/auth

By proceeding, you agree to the Microsoft Platform Policies

Click the Register button

Azure Screenshot

Azure Screenshot



Display name	Application (client) ID	Created on	Certificates & secrets
AD AdUserImport	67bd1625-2b66-454b-99...	10/12/2020	✓ Current

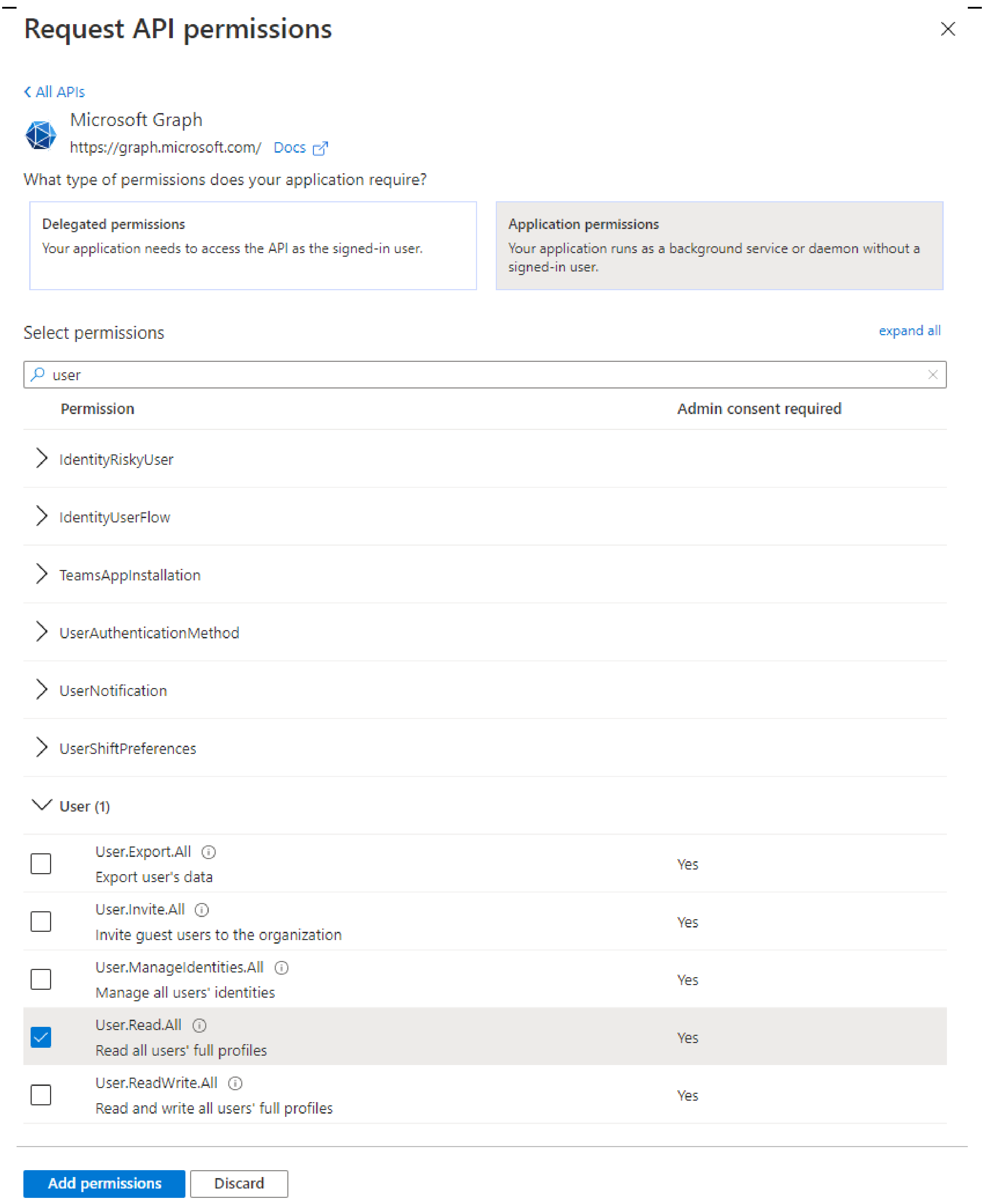
Step 2. Add API permissions

Select **API permissions** and click on **Add a permission**.
Select **Microsoft Graph** and then **Application permissions**.
Add these permissions:

- Group.Read.All
- GroupMember.Read.All
- User.Read.All

Azure Screenshot

Azure Screenshot



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

user

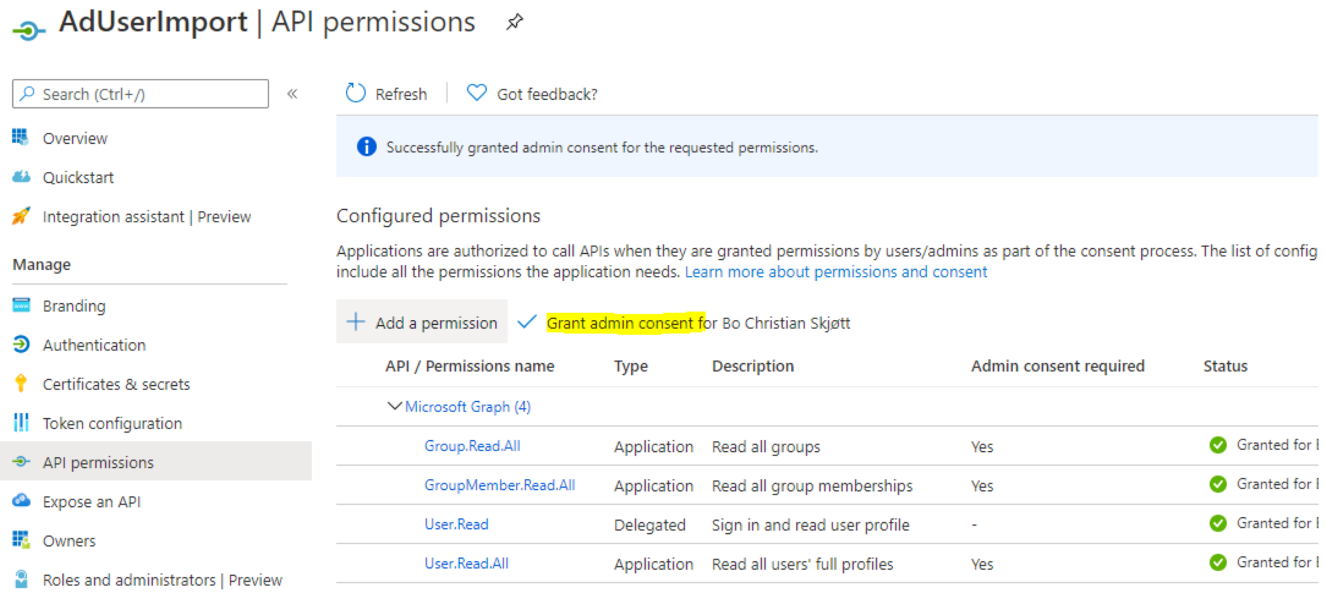
	Permission	Admin consent required
>	IdentityRiskyUser	
>	IdentityUserFlow	
>	TeamsAppInstallation	
>	UserAuthenticationMethod	
>	UserNotification	
>	UserShiftPreferences	
∨	User (1)	
☐	User.Export.All ⓘ Export user's data	Yes
☐	User.Invite.All ⓘ Invite guest users to the organization	Yes
☐	User.ManageIdentities.All ⓘ Manage all users' identities	Yes
☑	User.Read.All ⓘ Read all users' full profiles	Yes
☐	User.ReadWrite.All ⓘ Read and write all users' full profiles	Yes

Click on the **"Grant admin consent for ..."** button.

The permissions are then as shown below

Azure Screenshot

Azure Screenshot



Successfully granted admin consent for the requested permissions.

Configured permissions

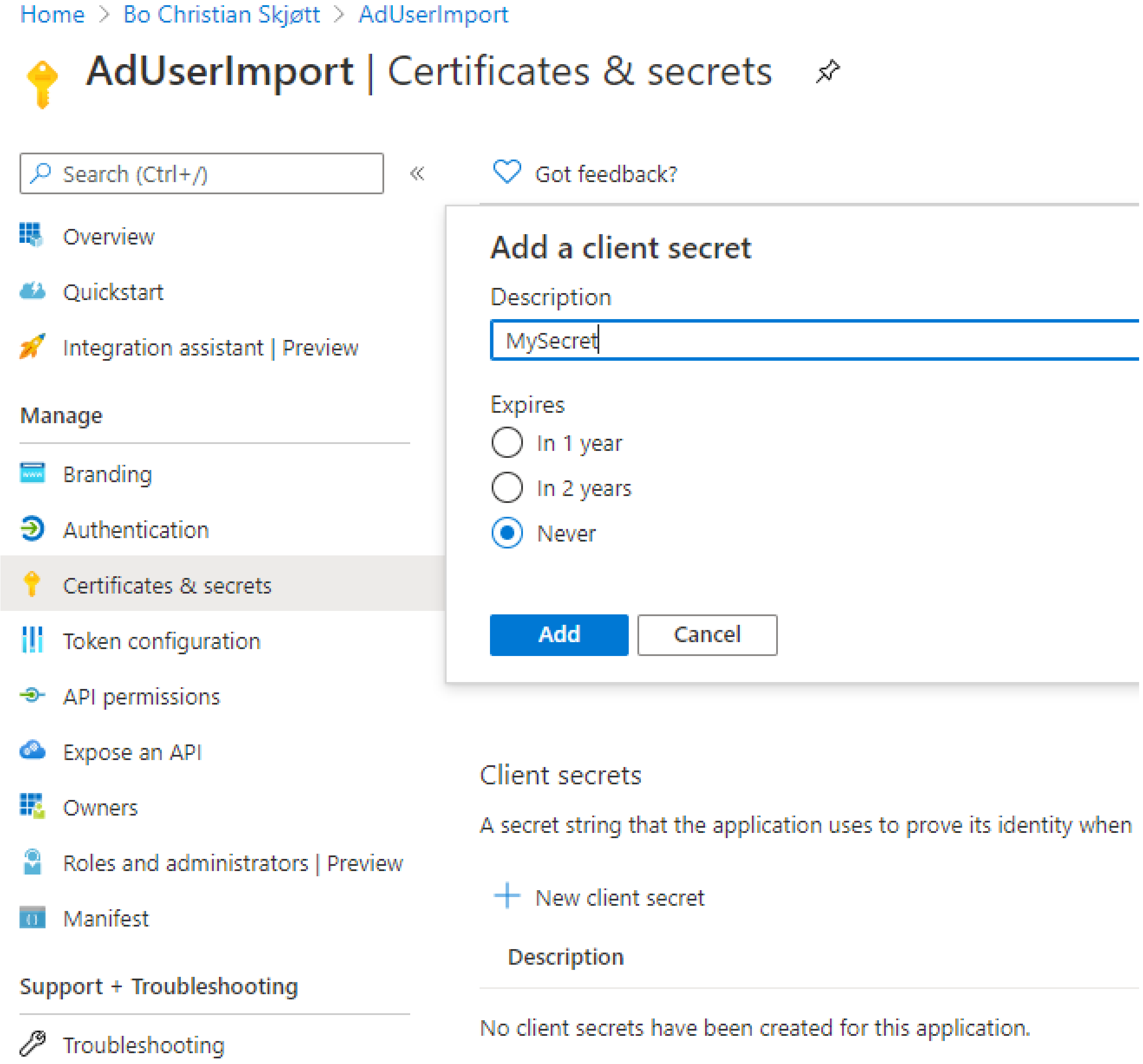
Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of config include all the permissions the application needs. Learn more about permissions and consent

API / Permissions name	Type	Description	Admin consent required	Status
Microsoft Graph (4)				
Group.Read.All	Application	Read all groups	Yes	✓ Granted for I
GroupMember.Read.All	Application	Read all group memberships	Yes	✓ Granted for I
User.Read	Delegated	Sign in and read user profile	-	✓ Granted for I
User.Read.All	Application	Read all users' full profiles	Yes	✓ Granted for I

Step 3. Add a Client Secret

Azure Screenshot

Azure Screenshot



Add a client secret

Description: MySecret

Expires
○ In 1 year
○ In 2 years
● Never

Client secrets

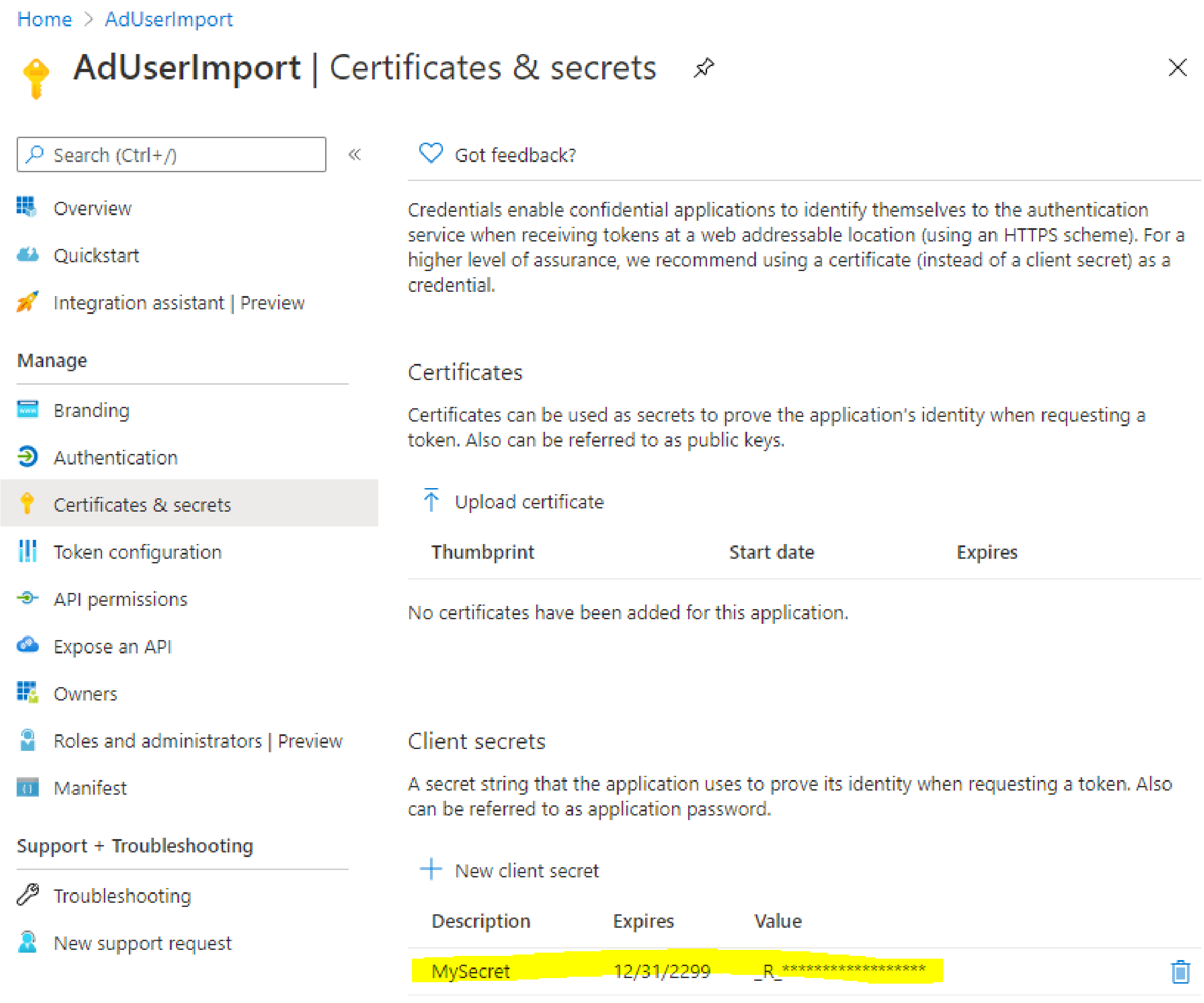
A secret string that the application uses to prove its identity when

Description
No client secrets have been created for this application.

Take a copy of the generated secret. It is only shown during creation.

Azure Screenshot

Azure Screenshot



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.

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.

Thumbprint	Start date	Expires

No certificates have been added for this application.

Client secrets

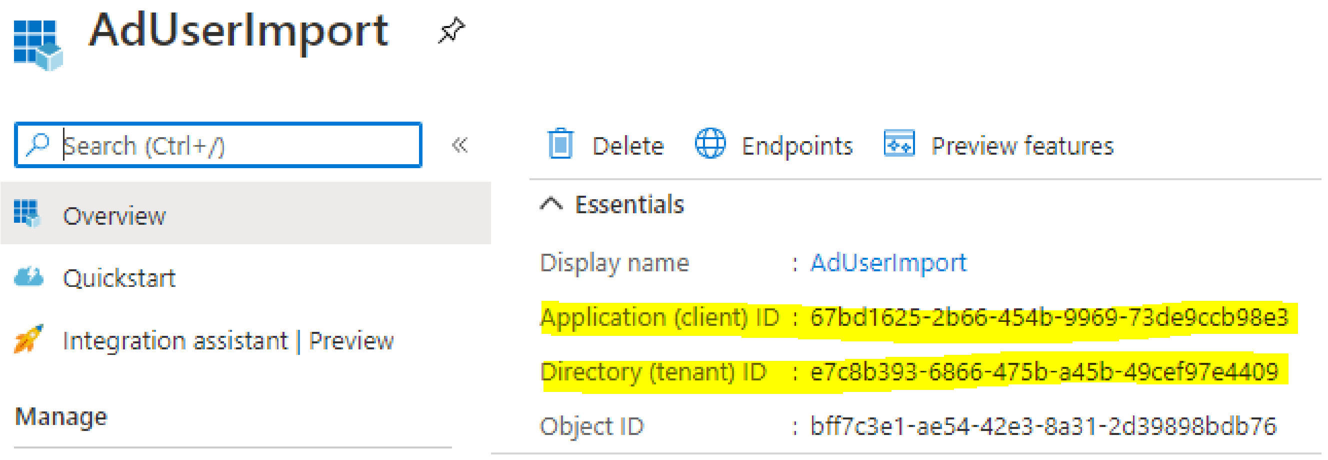
A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

Description	Expires	Value
MySecret	12/31/2299	_R_********************

Step 4. Get the Client and Tenant IDs for the application

Go to the Application Overview page and copy the Client ID and Tenant ID. You need to enter these in the Novus Configuration UI along with the Client Secret.

Azure Screenshot



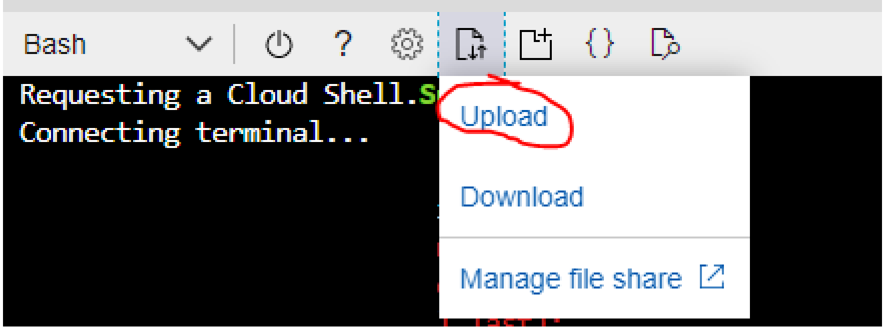
Display name : AdUserImport
Application (client) ID : 67bd1625-2b66-454b-9969-73de9ccb98e3
Directory (tenant) ID : e7c8b393-6866-475b-a45b-49cef97e4409
Object ID : bff7c3e1-ae54-42e3-8a31-2d39898bdb76

App creation using Azure CLI commands

The steps above can also be done with the Azure CLI commands below.

Copy the **requiredResourceAccess.json** file to the Azure storage (clouddrive) used by Azure CLI.

If you are using the **Cloud Shell** in the Azure Portal then you can click on the Upload File button in its menubar as shown below



```
Requesting a Cloud Shell.S
Connecting terminal...
```

Create an application with this command

```
az ad app create \
--display-name AdUserImport \
--password VerySecretWord#1234 \
--end-date 2100-12-31 \
--required-resource-accesses requiredResourceAccess.json
```

Replace the password with your choice.

Grant admin consent for the requested API permissions with this command

```
az ad app permission admin-consent --id 00000000-0000-0000-0000-000000000000
```

where 00000000-0000-0000-0000-000000000000 must be replaced with the actual ID of the application created above.

(Note: the **az ad app permission admin-consent** fails with an exception)