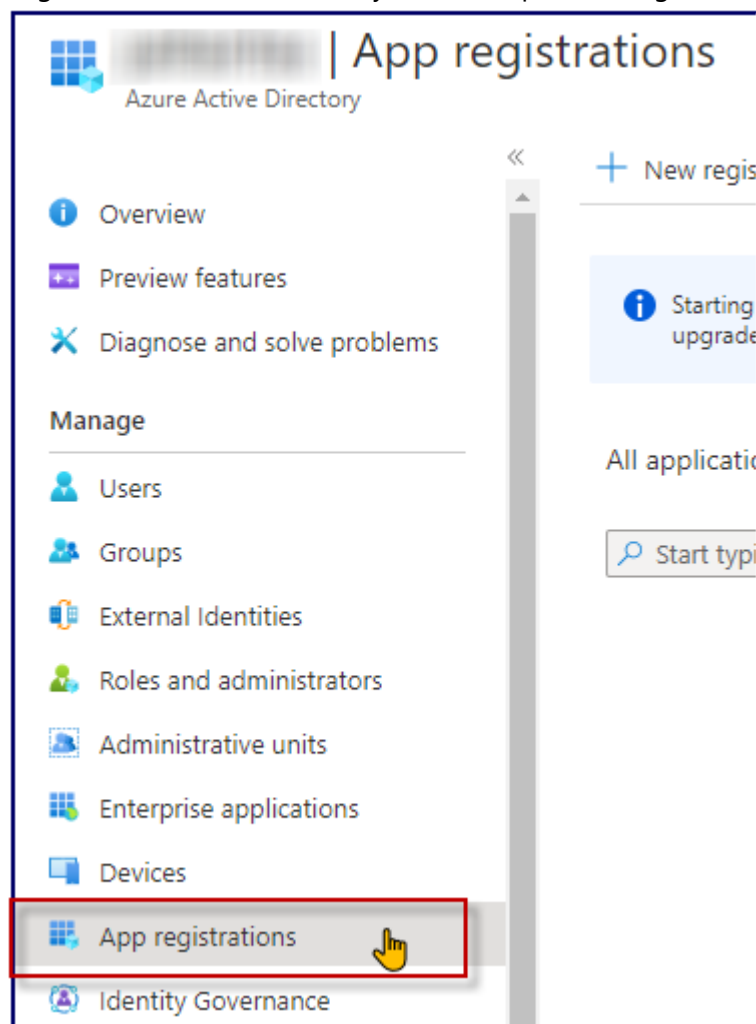


Deployment Guide

1. [Register Azure App Registration for the back-end API](#)
2. [Set Up the API App Registration's API Permissions](#)
3. [Set Up the API App Registration's scope Expose an API](#)
4. [Create the App Registration for the Web API](#)
5. [Set-up the WebClient's App Registration Authentication](#)
6. [Set up the WebClient App's API Permissions](#)
7. [Add API App Registration Consent Scope](#)
8. [Add Permissions to your WebClient app](#)
9. [Run Deployment Script](#)

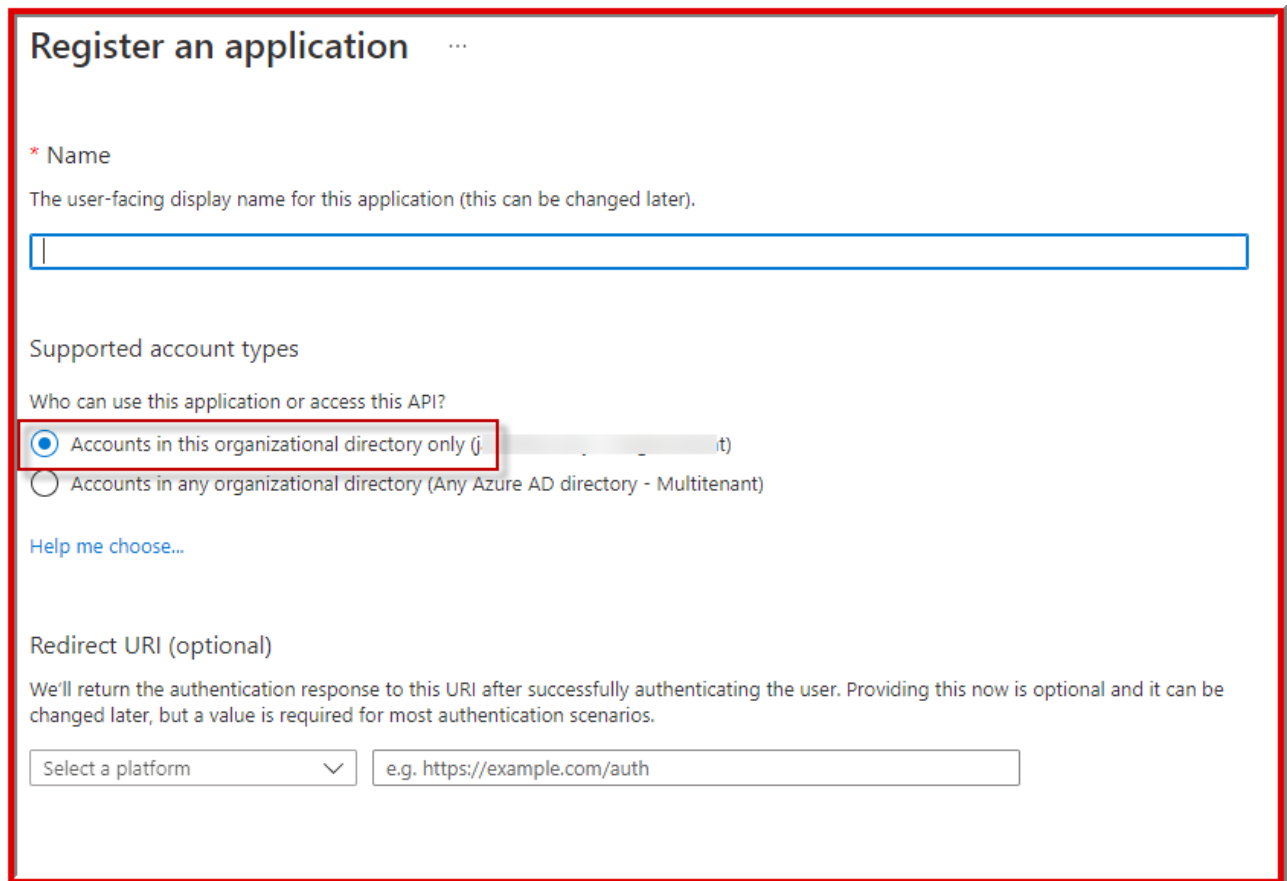
1. Register Azure App Registration for the back-end API

1. Log in to the Azure Portal for your subscription, and go to the [App registrations](#) blade.



2. Click **New registration** to create an Azure AD application.
 - **Name:** Name for your App: **Data Transfer Portal API** (You can name your app registration to any other meaningful value as well)
 - **Supported account types:** Select "Accounts in this organizational directory only(Default Directory only - Single tenant)". (*refer image below*).

- Leave the "Redirect URI" field blank for now.



Register an application ...

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

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

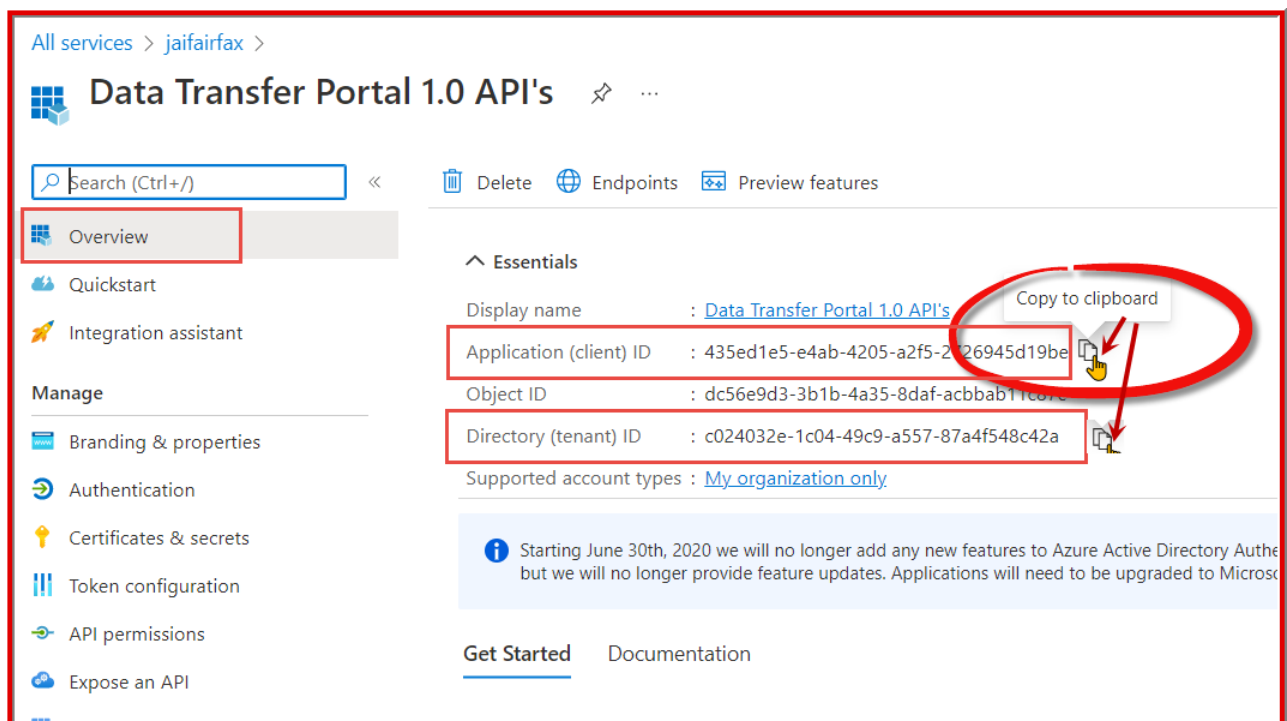
☒ Accounts in this organizational directory only (j... .it)
☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant)

[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.

Select a platform e.g. https://example.com/auth

3. Click **Register** to complete the registration.
4. When the app is registered, you'll be taken to the app's "Overview" page. Copy the **Application (client) ID**; we will need it later.



All services > jaifairfax >

Data Transfer Portal 1.0 API's ✎ ...

Search (Ctrl+/) << Delete Endpoints Preview features

Overview Quickstart Integration assistant

Manage
Branding & properties
Authentication
Certificates & secrets
Token configuration
API permissions
Expose an API

Essentials

Display name	: Data Transfer Portal 1.0 API's
Application (client) ID	: 435ed1e5-e4ab-4205-a2f5-2026945d19be
Object ID	: dc56e9d3-3b1b-4a35-8daf-acbbab11c07e
Directory (tenant) ID	: c024032e-1c04-49c9-a557-87a4f548c42a

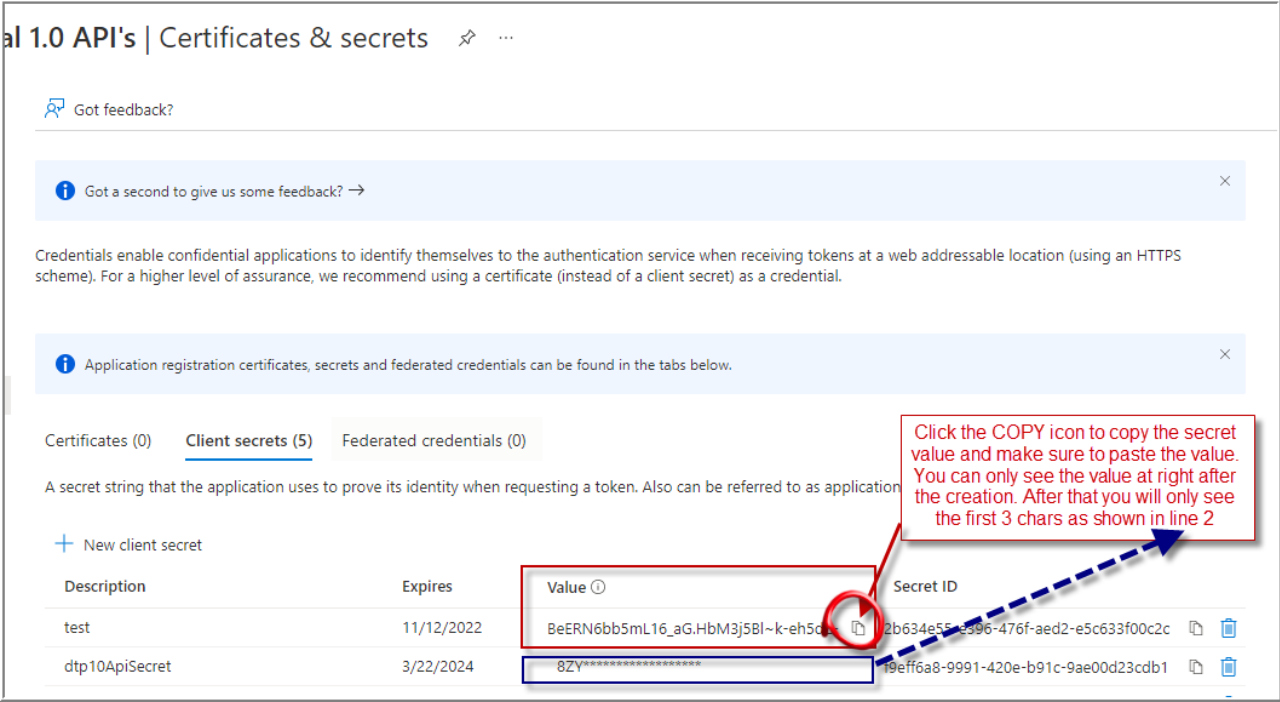
Supported account types : [My organization only](#)

Get Started Documentation

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Entra ID.

5. On the side rail in the Manage section, navigate to the "Certificates & secrets" section. In the Client secrets section, click on "+ New client secret". Add a description for the secret, and choose when the

secret will expire. Click "Add".



6. Once the client secret is created, copy and paste its **Value** (together with the app registration id and tenant id) in the table below; we will need it later.

Be aware that you can only see the secret value at the time of creation. You must copy and save the value at this time!

Data	Value
Tenant ID (Directory ID):	
API App Registration Display name:	
API App Registration ID (Client ID):	
API App Reg Secret Value:	

2. Set Up the API App Registration's API Permissions

Data Transfer Portal 1.0 API's | API permissions

Search (Ctrl+/) Refresh Got feedback?

Overview
Quickstart
Integration assistant

Manage

Branding & properties
Authentication
Certificates & secrets
Token configuration
API permissions
Expose an API
App roles
Owners
Roles and administrators
Manifest

Support + Troubleshooting

Troubleshooting
New support request

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 jaifairfax

API / Permissions name	Type	Description	Admin co
▼ Azure Storage (1)			
user_impersonation	Delegated	Access Azure Storage	No
▼ Microsoft Graph (5)			
Directory.AccessAsUser.All	Delegated	Access directory as the signed in user	Yes
Directory.Read.All	Delegated	Read directory data	Yes
IdentityUserFlow.Read.All	Delegated	Read all identity user flows	Yes
User.Read	Delegated	Sign in and read user profile	No
User.Read.All	Delegated	Read all users' full profiles	Yes

Other permissions granted for jaifairfax

These permissions have been granted for jaifairfax but aren't in the configured permissions list. If your application requires these permissions, you should consider adding them to the configured permissions list. [Learn more](#)

API / Permissions name	Type	Description	Admin co
▼ Azure Service Management			
user_impersonation	Delegated	Access Azure Service Management as organiza...	No

3. Set Up the API App Registration's scope [Expose an API]

Data Transfer Portal 1.0 API's | Expose an API

Search (Ctrl+/) << Got feedback?

Overview
Quickstart
Integration assistant

Manage

Branding & properties
Authentication
Certificates & secrets
Token configuration
API permissions
Expose an API
App roles
Owners
Roles and administrators
Manifest

Support + Troubleshooting
Troubleshooting
New support request

Application ID URI: `api://435ed1e5-e4ab-4205-a2f5-2726945d19be`

Scopes defined by this API

Define custom scopes to restrict access to data and functionality protected by the API. An application that requires access to parts of this API can request that a user or admin consent to one or more of these.

Adding a scope here creates only delegated permissions. If you are looking to create application-only scopes, use 'App roles' and define app roles assignable to application type. [Go to App roles.](#)

+ Add a scope

Scopes	Who can consent	Admin consent disp...	User coi
<code>api://435ed1e5-e4ab-4205-a2f5-2726...</code>	Admins and users	Permit use of DTP	User Per

Authorized client applications

Authorizing a client application indicates that this API trusts the application and users should not be asked to consent when the client calls this API.

+ Add a client application

Client Id	Scopes
No client applications have been authorized	

4. Create the App Registration for the Web Client

- Repeat steps 2-4 to create another Azure App Registration for the web site.

Note: The WebClient does not need a secret to be created

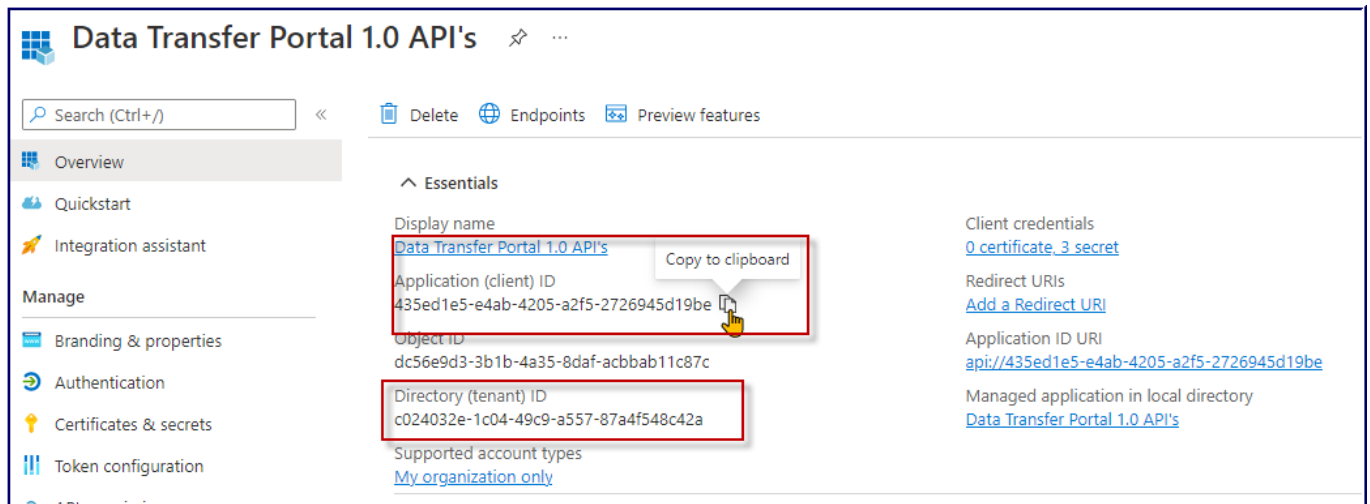
- Name:** Name of the API - **Data Transfer Portal** (or any other name meaningful to you)
- Supported account types:** Select "**Accounts in this organizational directory only**(Default Directory only - Single tenant)".
- Leave the "**Redirect URI**" field **blank**.

App Registrations Results

At this point you should have the following values:
(You can copy this table and paste it into OneNote/Word/Excel)
and populate the values for your records

Data	Value
Tenant ID:	

Data	Value
API App Registration Display name:	
API App Registration ID (Client ID):	
API App Reg Secret Value:	
Web Client App Registration Display Name	
Web Client App Registration ID	

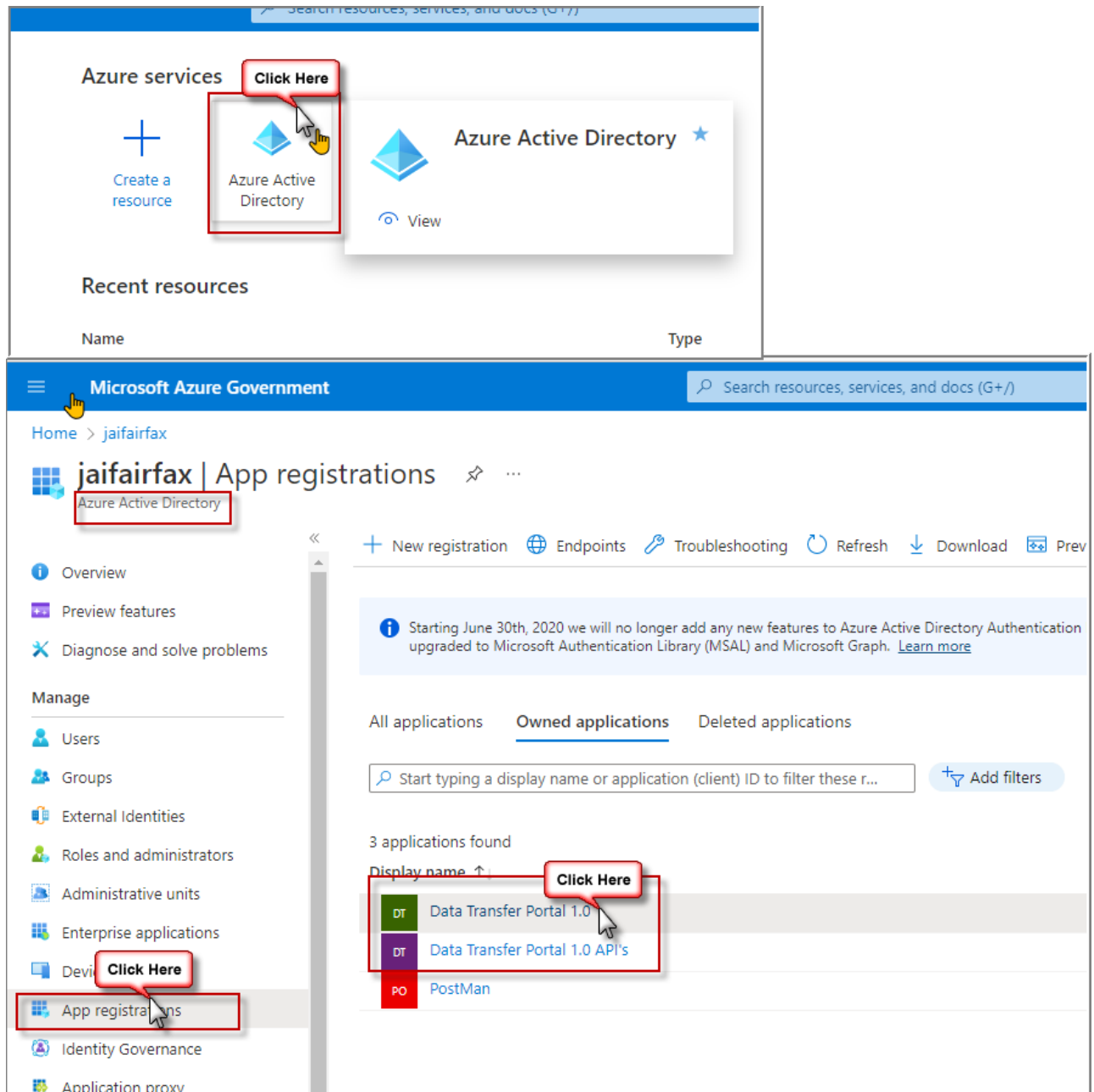


-[] Create App Registration (Service Principal) With Powershell

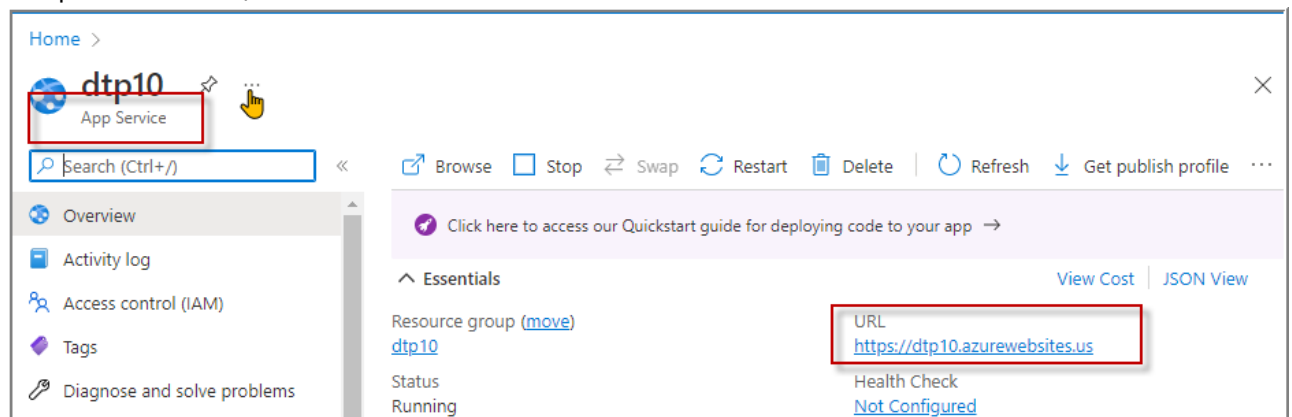
5. Set-up the WebClient's App Registration Authentication

NOTE: This step is to set-up authentication for Microsoft Graph Azure AD app. After the deployment script successfully creates the resources, navigate back to the Web Client App registration created in Step 7 to set up the authentication

1. Go to the **App Registrations** page in Azure Active Directory [here](#) and open the Microsoft Graph Azure AD app you created (in Step 1) from the application list.



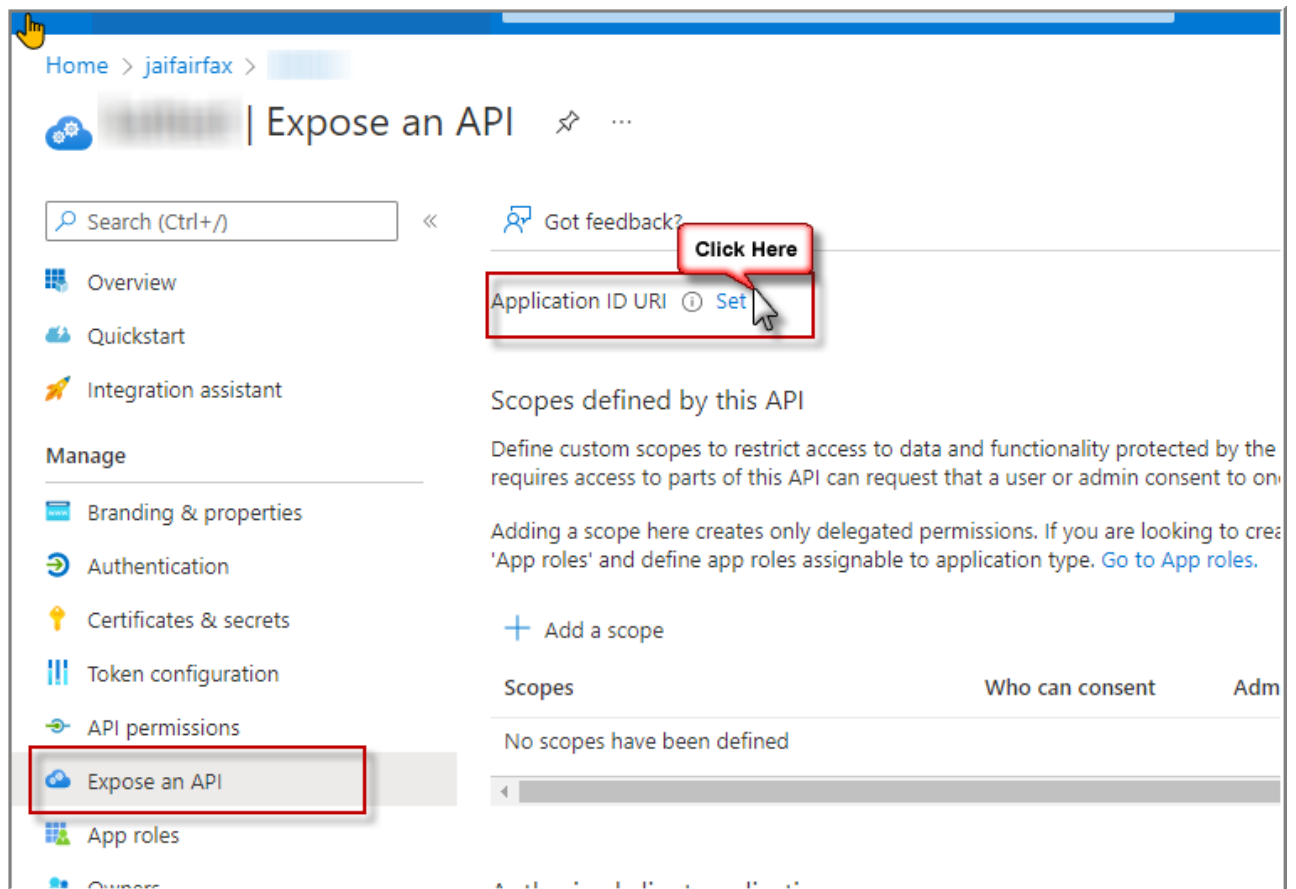
2. Select the **WebClient App Registration** (as shown in the image above) Under **Manage**, click on **Authentication** to bring up authentication settings.
3. Add a new entry to **Redirect URIs**: - **Type**: Web - **Redirect URI**: Enter `https://%appDomain%/` for the URL e.g. `https://appName.azurewebsites.us/` (This is the URL from the App Service created by Bicep shown below)



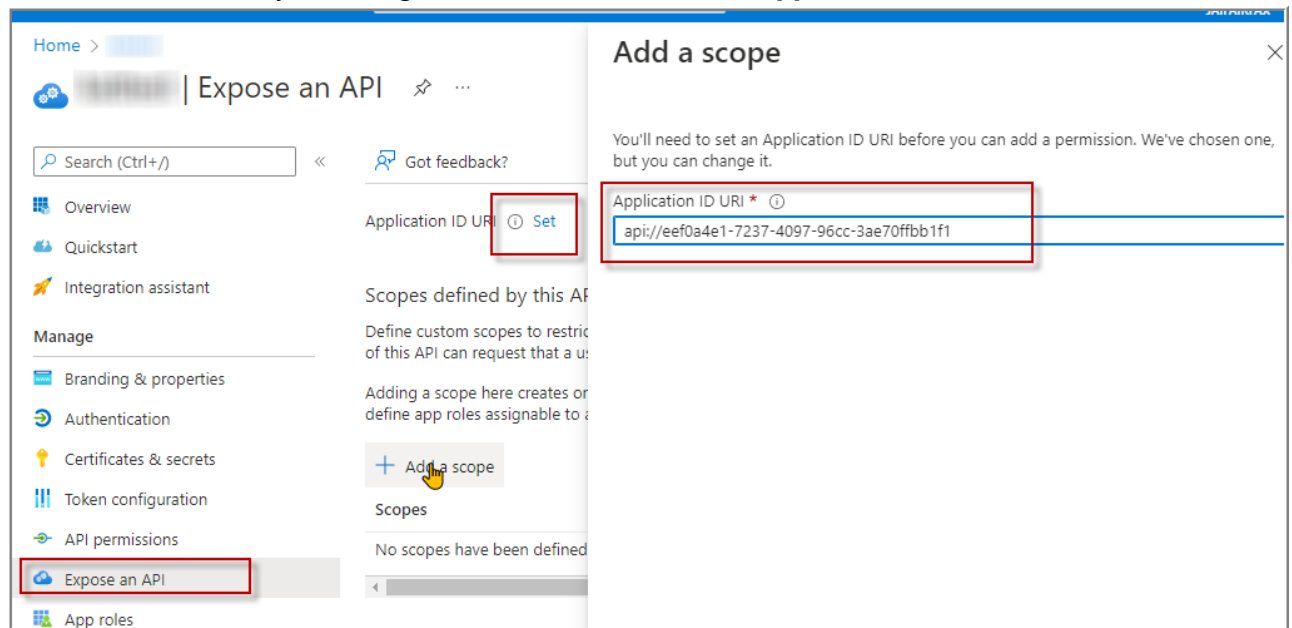
4. Under **Implicit grant**, leave the **ID tokens** and **Access tokens** unchecked.
5. Under **Advanced Settings**, make sure that **Enable the following mobile and desktop flows** slider has **No** selected
6. Click **Save** to commit your changes.

6. Set up the WebClient App's API Permissions

1. Back under **Manage**, click on **Expose an API**.
2. Click on the **Set** link next to **Application ID URI**, and change the value to `api://%appDomain%` e.g. `api://appName.azurewebsites.us`.



3. Click **Save** to commit your changes. You should now see the **Application ID URI** like below:



7. Add API App Registration Consent Scope

4. Under the section: **Scopes defined by this API**, Click on **Add a scope**, under **Scopes defined by this API**. In the flyout that appears, enter the following values:

Property	Value
Scope name:	access_as_user
Who can consent?:	Admins and users
Admin and user consent display name:	Access the API as the current logged-in user
Admin and user consent description:	Access the API as the current logged-in user

The screenshot shows the 'Add a scope' dialog in the Azure portal. The left sidebar has 'Expose an API' highlighted. The dialog fields are as follows:

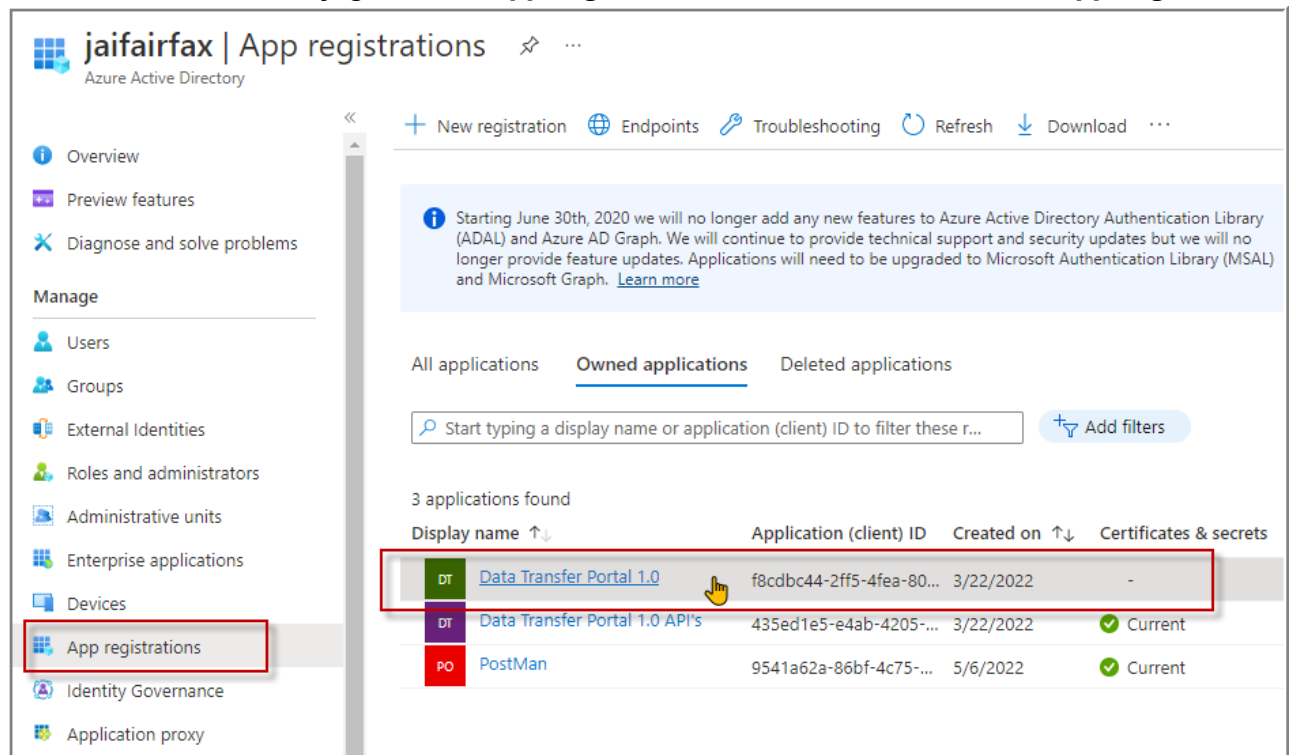
- Scope name ***: access_as_user
- Who can consent?**: Admins and users (selected), Admins only
- Admin consent display name ***: Access the API as the current logged-in user
- Admin consent description ***: Access the API as the current logged-in user
- User consent display name**: e.g. Read your files
- User consent description**: e.g. Allows the app to read your files.
- State**: Enabled (selected), Disabled

At the bottom of the dialog are 'Add scope' and 'Cancel' buttons.

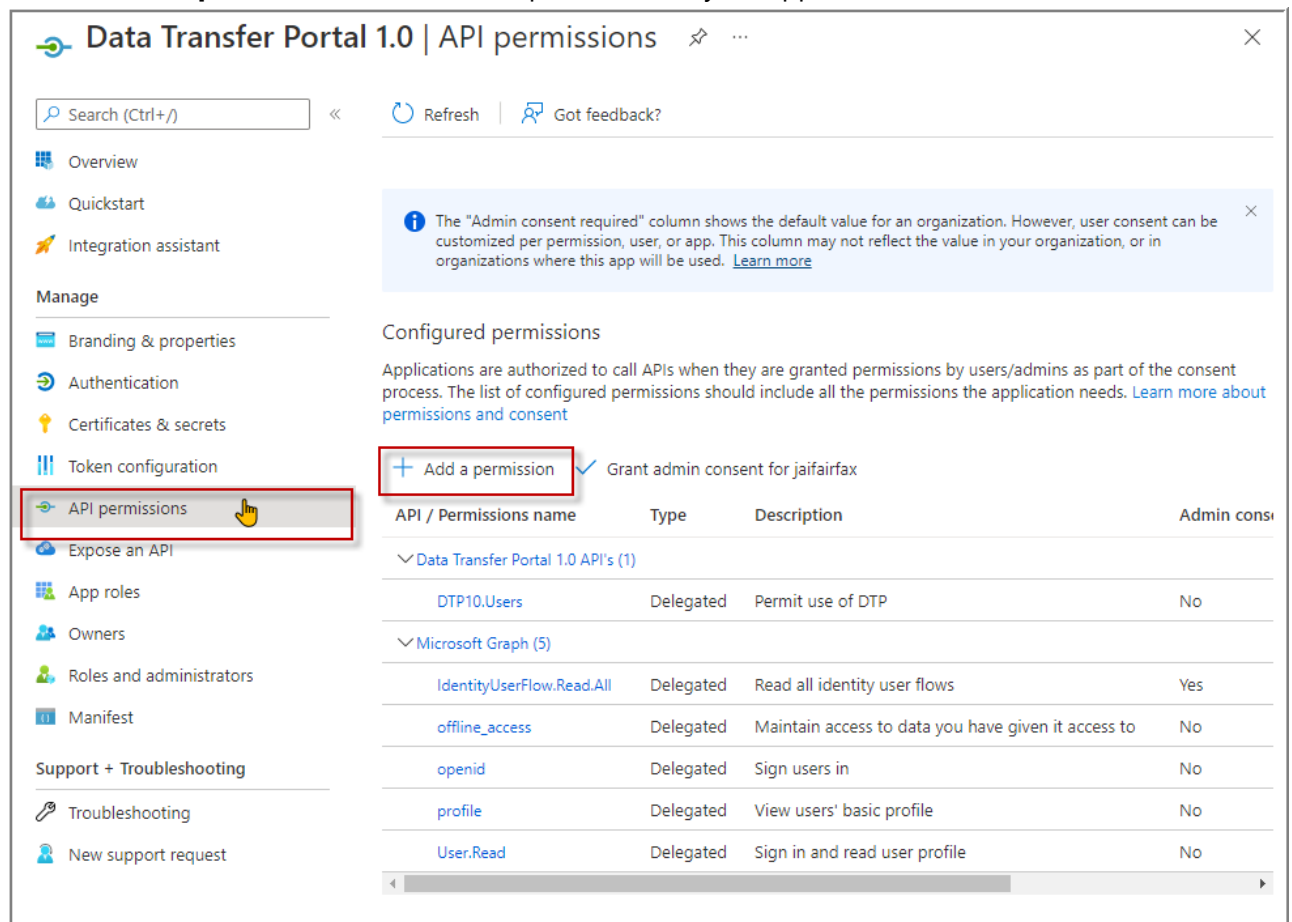
- While still in the **Expose and API** blade, click **Add scope** then click on the **Save and continue** to commit your changes.
- Click **Add a client application**, under **Authorized client applications**. In the flyout that appears, enter the following values: - **Client ID**: 5e3ce6c0-2b1f-4285-8d4b-75ee78787346 - **Authorized scopes**: Select the scope that ends with `access_as_user`. (There should only be 1 scope in this list.)
- Click **Add application** to commit your changes.

8. Add Permissions to your WebClient app

1. In **Azure Active Directory**, go back to **App Registrations** and select the **WebClient App Registration**



2. Select **API Permissions** blade from the left hand side.
3. Click on **Add a permission** button to add permission to your app.



4. In Microsoft APIs under Select an API label, select the particular service and give the following permissions:

Permissions to add:

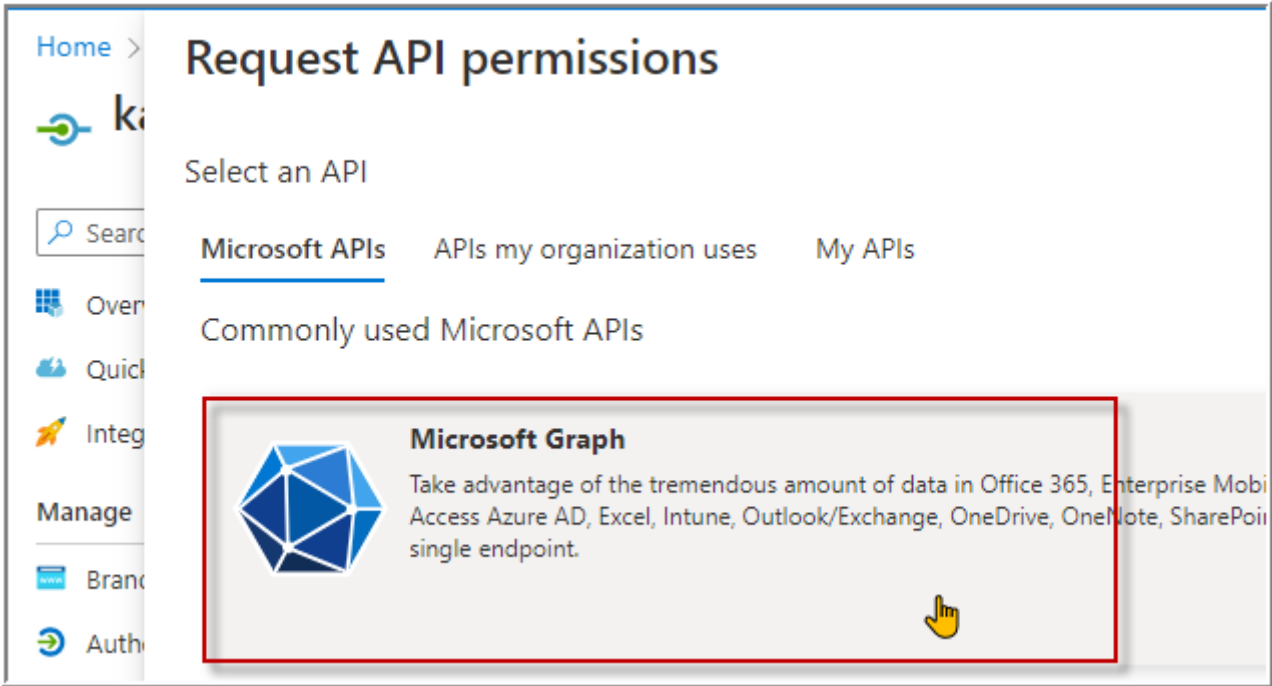
IdentityUserFlow.Read.All *start typing 'Identity' in the search field to show related permissions*

offline_access

openid

profile

User.Read



Home > Request API permissions

< All APIs

Microsoft Graph
<https://graph.microsoft.us/> 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 signed-in user.

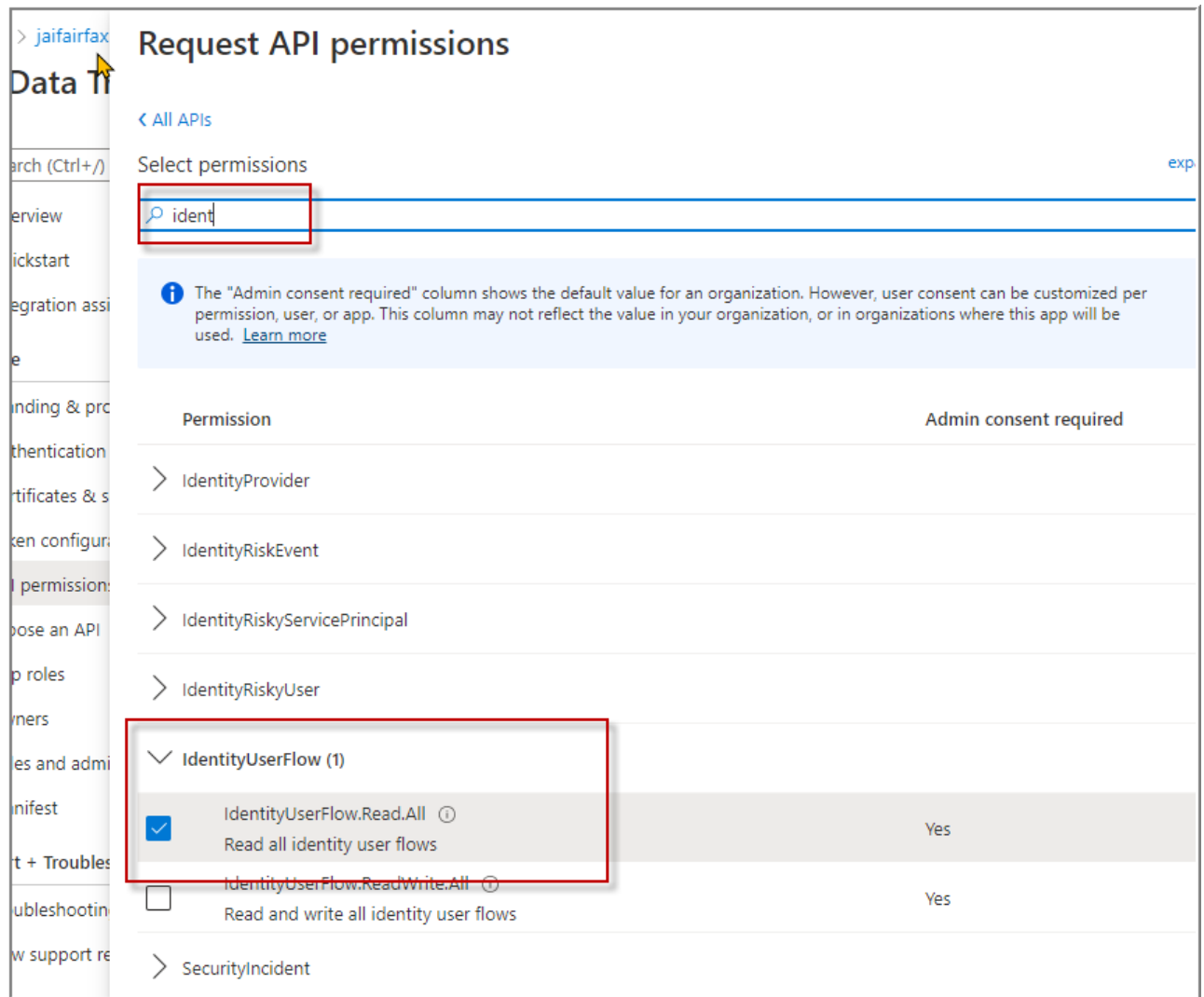
Select permissions

Start typing a permission to filter these results

i The "Admin consent required" column shows the default value for an organization. However, user consent can be customized for a permission, user, or app. This column may not reflect the value in your organization, or in organizations where this app is used. [Learn more](#)

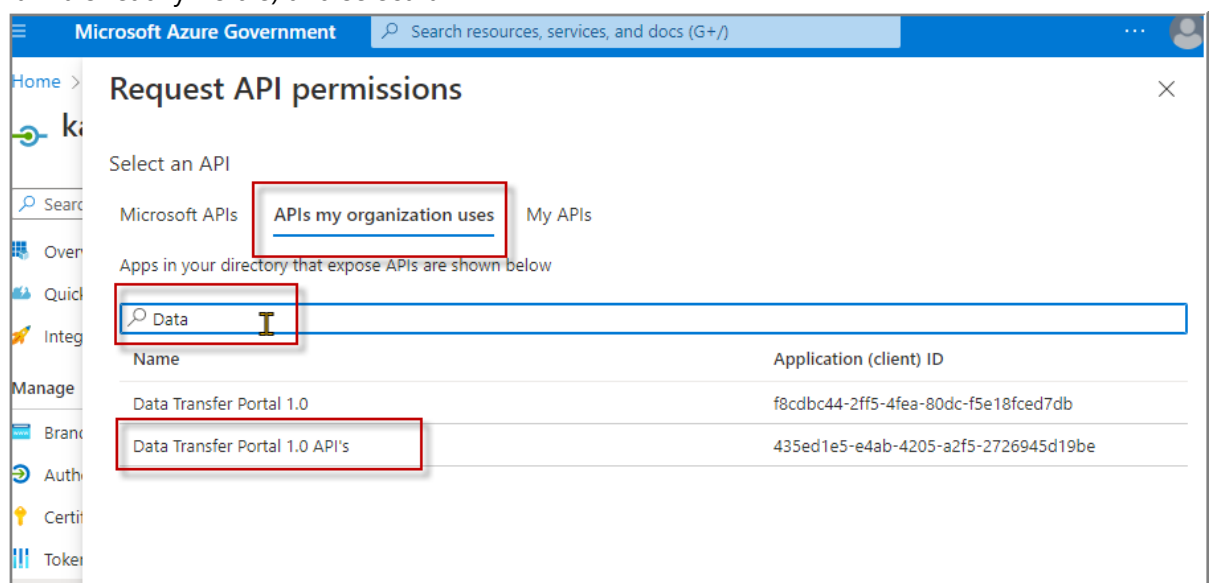
Permission	Admin consent required
<input type="checkbox"/> email i View users' email address	No
<input checked="" type="checkbox"/> offline_access i Maintain access to data you have given it access to	No
<input checked="" type="checkbox"/> openid i Sign users in	No
<input checked="" type="checkbox"/> profile i View users' basic profile	No

Add permissions Discard



5. Click on **Add a permission** again and select **APIs my organization uses**

- In the search field, start typing the name of your **API app registration name** (or simply click on it if it is readily visible) and select it



- Select **Delegated Permissions**
- Select the expanded permission: **XXX.Users**

- Click **Add Permissions**

Request API permissions

< All APIs

Data Transfer Portal 1.0 API's
api://435ed1e5-e4ab-4205-a2f5-2726945d19be

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 permission to filter these results

Information The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value in your organization, or in organizations where this app will be used. [Learn more](#)

Permission	Admin consent required
DTP10 (1) <input checked="" type="checkbox"/> DTP10.Users ⓘ Permit use of DTP	No

- Click on **Add Permissions** to commit your changes. You should see the results as shown below:

Home > jaifairfax > Data Transfer Portal 1.0 API's

Data Transfer Portal 1.0 API's | API permissions

Search (Ctrl+/) Refresh Got feedback?

Overview
Quickstart
Integration assistant

Manage
Branding & properties
Authentication
Certificates & secrets
Token configuration
API permissions
Expose an API
App roles
Owners
Roles and administrators
Manifest

Support + Troubleshooting
Troubleshooting
New support request

Information The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value in your organization, or in organizations where this app will be used. [Learn more](#)

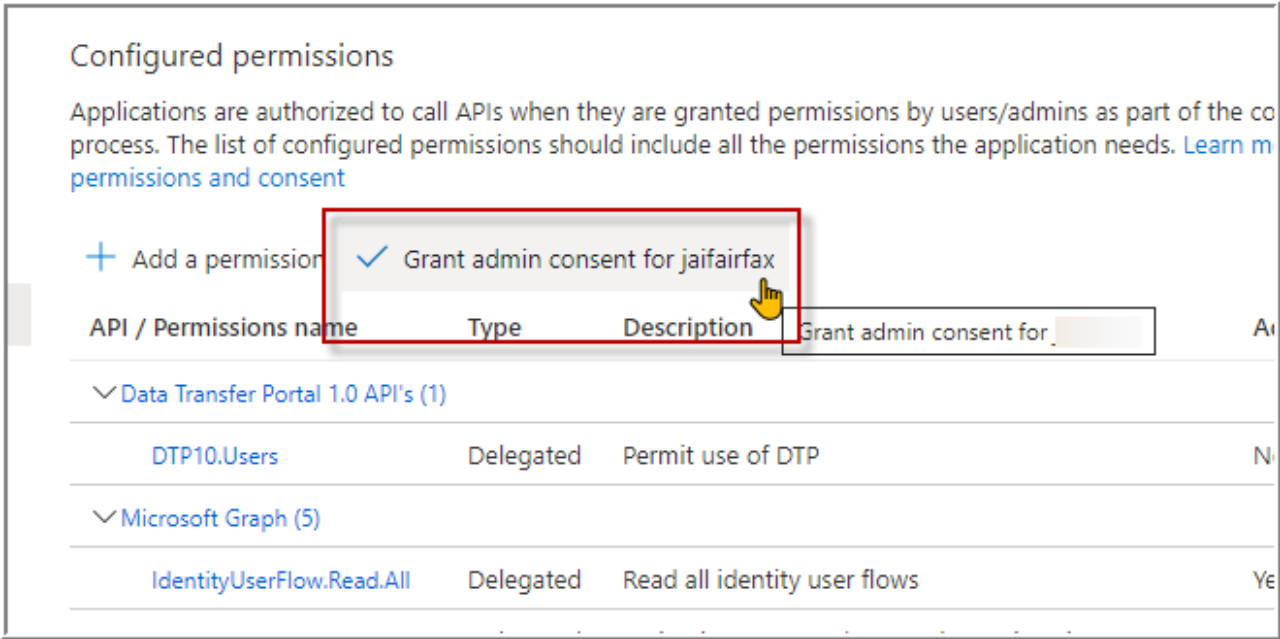
Configured permissions
Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission ✓ Grant admin consent for jaifairfax

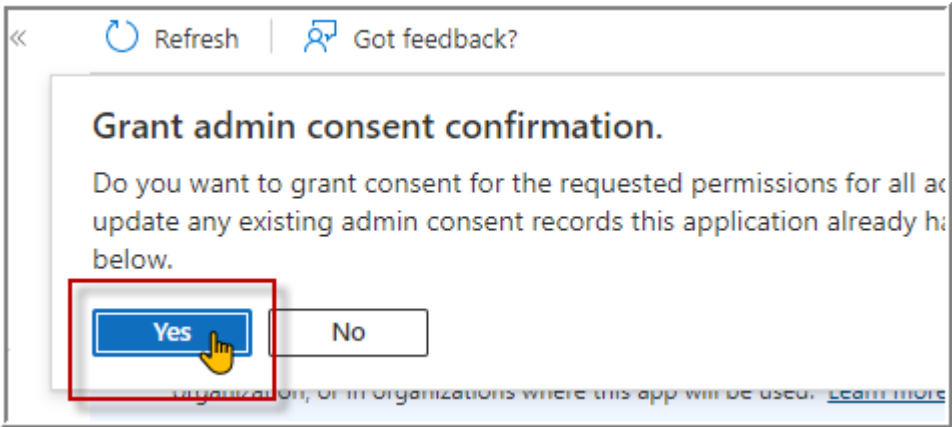
API / Permissions name	Type	Description	Admin consent req...	Status
Azure Storage (1)				...
user_impersonation	Delegated	Access Azure Storage	No	...
Microsoft Graph (5)				...
Directory.AccessAsUser.All	Delegated	Access directory as the signed in user	Yes	...
Directory.Read.All	Delegated	Read directory data	Yes	...
IdentityUserFlow.Read.All	Delegated	Read all identity user flows	Yes	...
User.Read	Delegated	Sign in and read user profile	No	...
User.Read.All	Delegated	Read all users' full profiles	Yes	...

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

- If you are logged in as the **Global Administrator**, click on the **"Grant admin consent for %tenant-name%"** button to grant admin consent, else inform your Admin to do the same through the portal.



Click **Yes** in the dialog:



9. Run Deployment Script

10. Troubleshooting

Please check the [Troubleshooting](#) guide.