

# Azure Application Permissions

Last Modified on 06/12/2019 7:09 pm AEDT

ReadiNow API Callouts can connect to Microsoft Azure APIs by using OAuth 2.0 authentication.

Azure offers two permission models:

- **Application** - where a software application such as ReadiNow connects to an API service on its own behalf
- **Delegated** - where a software application is connecting to an API service on behalf of a specific end user

This page describes how to configure API Callouts to connect using the **Application** model, which is the recommended approach. However, certain APIs may only be made accessible via the delegated model. See [Azure Delegated Permissions](#) for details on configuring delegated permissions.

Additional information can also be found in the Microsoft Azure reference at: [Configuring a client application to access web APIs](#)

## Overview

The following sample demonstrates how to:

- configure Azure to receive connections using the application permission model
- configure ReadiNow API Callouts to authenticate with Azure

Once complete, refer to [Connecting to Azure APIs](#) to extend the sample to set up specific API endpoints and call them using a workflow.

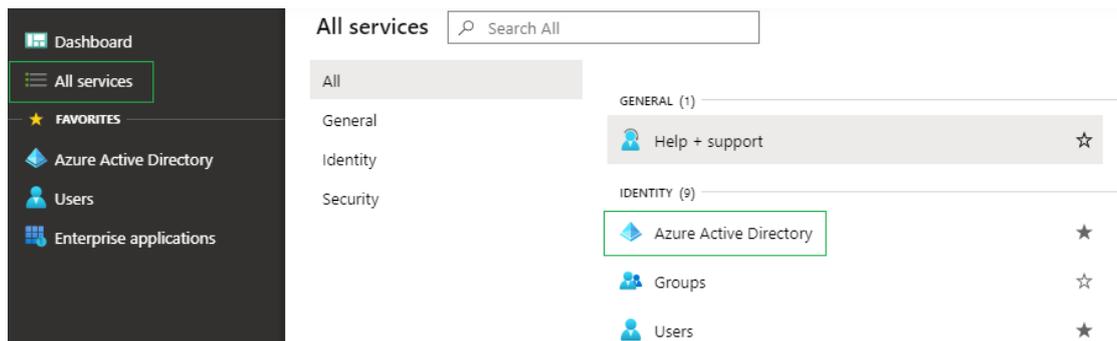
Note: the following is provided as an example to illustrate connecting to the Azure APIs in general, and use the Azure 'users' API as an example. If you wish to achieve automatic provisioning, then use the single-sign-on provisioning mechanism.

## Configure Azure

The following steps will configure Azure to receive a connection from the ReadNow platform, and grant sufficient permission for the ReadNow API Callout to request information about users.

### 1. Register an application

1. Log into the [Azure portal](#)
2. Select **Azure Active Directory** on left, or locate it under **All services**



### 3. Select **App Registrations**

### 4. Click the **New Application Registration** button

### 5. Enter a name for the application registration, such as "ReadNow - Application Permission Sample"

#### Register an application

##### \* Name

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

[1. Select an application name](#)

ReadNow - Application Permission Sample ✓

##### Supported account types

Who can use this application or access this API?

[2. Select the first, default, option](#)

- Accounts in this organizational directory only (ReadNow - 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)

[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.

[3. Leave blank. This is not required when using 'Application Permissions'.](#)

Web



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

6. Leave **Supported account types** and **Redirect URI** as their default and blank options respectively.

7. Click the **Register** button at the bottom of the screen
2. Configure the application
  1. An application information screen such as the following will be presented

Readiness - Application Permission Sample

Search (Ctrl+/) « Delete Endpoints

Overview

Quickstart

Manage

Branding

Authentication

Display name  
Readiness - Application Permission Sample

Application (client) ID  
d7ad578e-7dcb-4572-8158-a5735fb56076

Directory (tenant) ID  
466d4e93-3865-4736-bac9-a892913595b6

Object ID  
4dfe61a7-4f56-400f-b024-4fa774922bee

Supported account types  
My organization only

Redirect URIs  
Add a Redirect URI

Application ID URI  
Add an Application ID URI

Managed application in local directory  
Readiness - Application Permission Sample

2. Make a note of the **Application (client) ID** - you will need this in a later step
3. Configure a Client Secret
  1. A *client secret* can be thought of as a password for an application, such as the Readiness platform, rather than a person.
  2. Click **Certificates & secrets** on the left hand margin
  3. Click the **New Client secret** button
  4. Select an expiry date and click the **Add** button
  5. A new value will appear such as: LjVYHK9r0oCUCMutAN5QUU4vzgu@X= \_: in the client secrets table

Client secrets

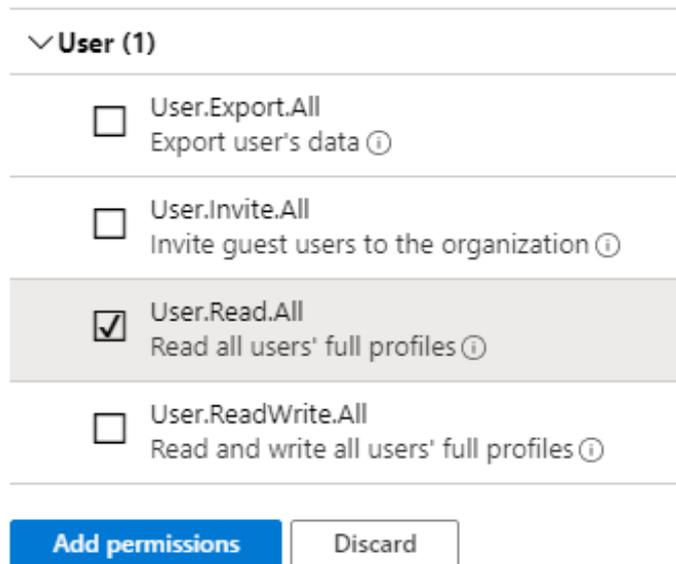
A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value
Password uploaded on Fri Dec 06 2019	12/6/2020	LjVYHK9r0oCUCMutAN5QUU4vzgu@X= _:

6. Immediately copy it to a notepad document, or similar. This is the *OAuth client secret*. It cannot be recovered later.
4. Configure permissions that are needed to access the API
  1. Click on **API permissions** in the left margin
  2. Click the **Add a permission** button
  3. Click on the **Microsoft Graph** tile (or whichever API service you wish to access via Readiness API Callouts)
  4. Select **Application permissions**
  5. Locate and enable the **User.Read.All** permission (or whichever permissions

are required for the API you intend to call)

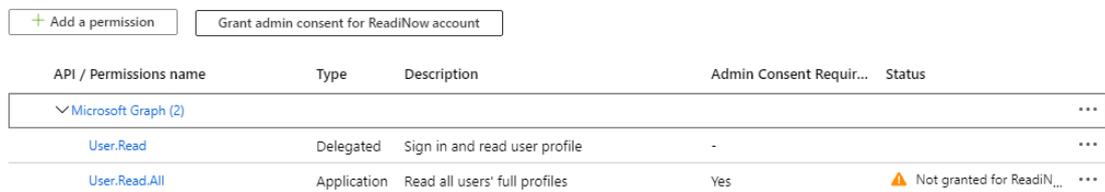


▼ User (1)

- User.Export.All  
Export user's data ⓘ
- User.Invite.All  
Invite guest users to the organization ⓘ
- User.Read.All  
Read all users' full profiles ⓘ
- User.ReadWrite.All  
Read and write all users' full profiles ⓘ

Add permissions Discard

6. Click the **Add permissions** button at the bottom of the panel
7. The new permission will appear in the permissions table
- 8.



API / Permissions name	Type	Description	Admin Consent Required	Status
▼ Microsoft Graph (2)				...
User.Read	Delegated	Sign in and read user profile	-	...
User.Read.All	Application	Read all users' full profiles	Yes	⚠ Not granted for ReadNow...

9. The new permission will have a status of not granted. It must be explicitly given consent because it is an Application permission granted to a software system (namely the ReadNow API Callouts) rather than a delegated permission acting on behalf of a person.
10. Click the **Grant admin consent for account** button
11. A Microsoft login window will appear
12. Login, review the permissions granted, and click the **Accept** button

## Configure ReadNow API Callouts

The following steps will start to prepare a new API Callout library in ReadNow to connect to Azure.

1. Log into ReadNow

2. Go to Administration / Integration / API Callouts
3. Create a new API Callout
4. Name it "Azure" or similar
5. Leave the **Base URL** blank
6. Set the message format to JSON
7. On the Authention tab, set the **Authentication method** to **OAuth 2.0**
8. Ensure that the **Grant Type** is set to **Client Credentials** - this corresponds to the Azure 'Application permission' type

#### BASIC SETTINGS

---

Base URL :

Message format :        JSON

Ignore certificate error :   

**APIs**   **API Categories**   **Authentication**   **Shared Headers**   **Shared Inputs**

---

Authentication method :    OAuth 2.0

Grant type :                 Client Credentials

Client Id :                  d7ad578e-7dcb-4572-8158-a573...

Client secret :             \*\*\*\*\*

Token URL :                 https://login.microsoftonline.com/**yourdomain.com**/oauth2/token

Scope (optional) :

Additional params :         resource:https://graph.microsoft.com/

9. Set the **Client ID** to the **Application (client) ID** value provided by Azure above
10. Set the **Client Secret** to the value provided by Azure above
11. Set the **Token URL** to:  
       https://login.microsoftonline.com/yourdomain.com/oauth2/token (where yourdomain.com is your ActiveDirectory domain, such as company.com)
12. Set the **Additional params** to: resource:https://graph.microsoft.com/    This indicates to Azure which Azure API service the authentication token will be allowed to access.

## Next Steps

Azure and REDINow are now both configured so that REDINow callouts can connect to Azure.

Continue with [Connecting to Azure APIs](#) to extend the sample to:

- create a API Callout endpoint to request user details
  - create a workflow that uses the API Callout and processes results
-