OPERATION MANUAL

PURPOSE OF THE DOCUMENT

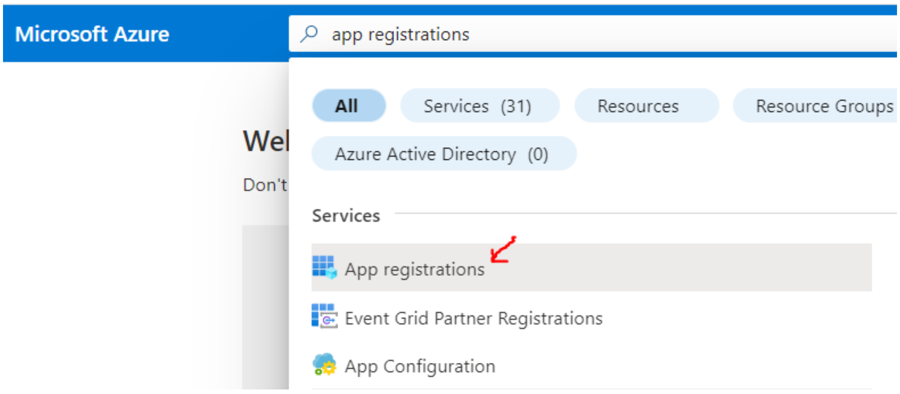
The purpose of this document is to describe the procedure to follow to enable the access of an application to the Office365 platform using the oAuth authentication method.

It also describes how to limit the application's access permissions to the desired services (mail, OneDrive...) as well as the users it can access within an organization's Azure AD.

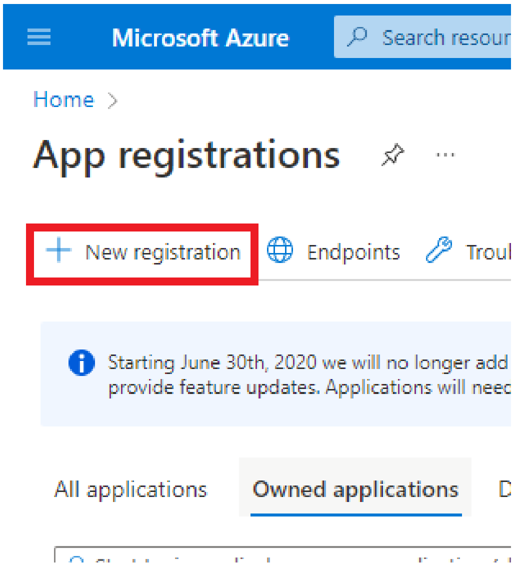
In no case is the interactive authentication method documented, as it is a method that requires the validation of a user every time a connection is started.

APPLICATION CREATION

1. Navigate to : portal.azure.com
2. Login with a user that has admin privileges in the Microsoft tenant
3. Use the search bar and type **App Registrations**

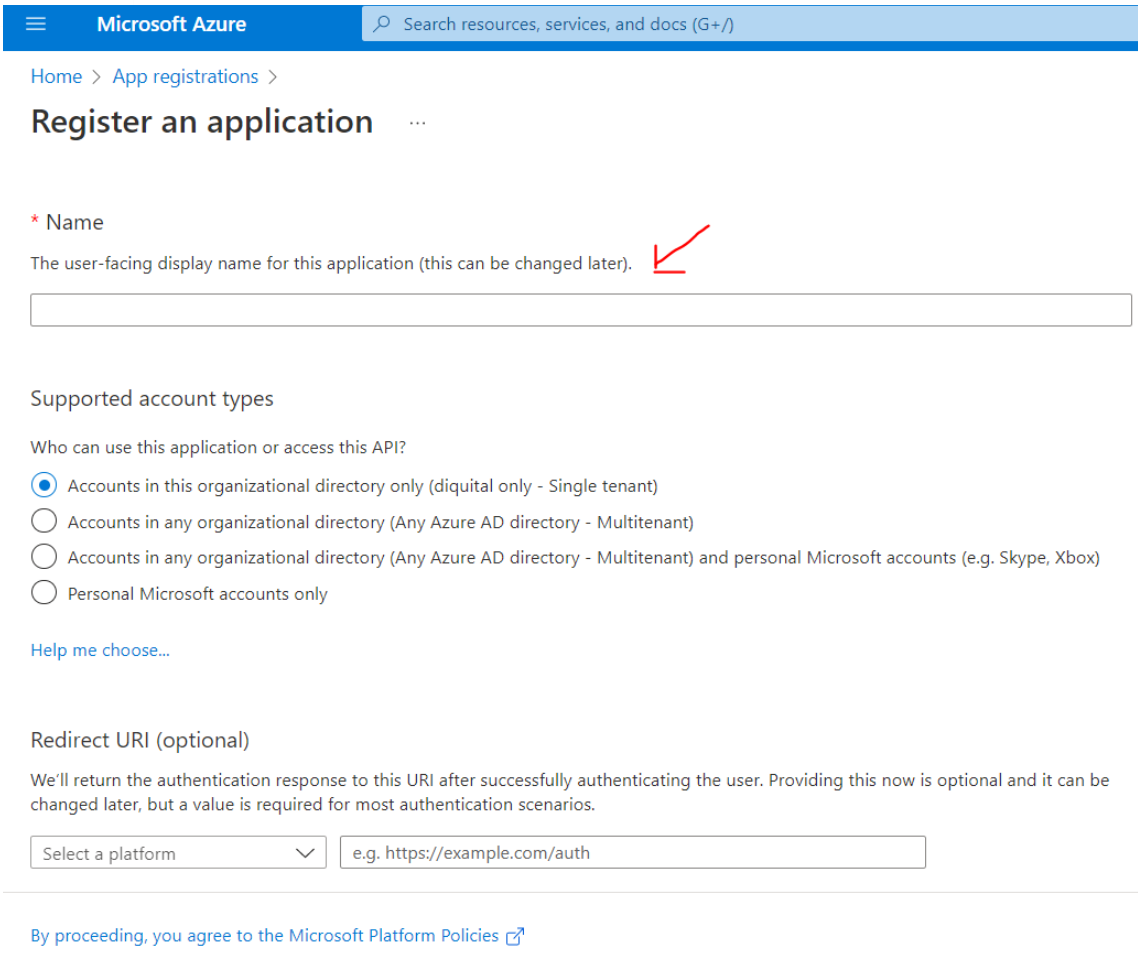


4. Once inside App registrations we click on the button **New registration**



Starting June 30th, 2020 we will no longer add provide feature updates. Applications will neec

5. Enter a desired name



Microsoft Azure
Search resources, services, and docs (G+/)
Home > App registrations >

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 (diquital 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...

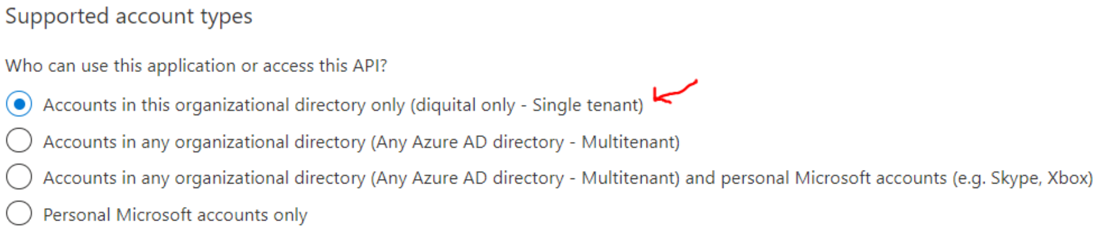
Redirect URI (optional)

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.

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

By proceeding, you agree to the Microsoft Platform Policies

6. From the options shown in **Supported account types**, select the option shown in the screenshot.



Supported account types

Who can use this application or access this API?

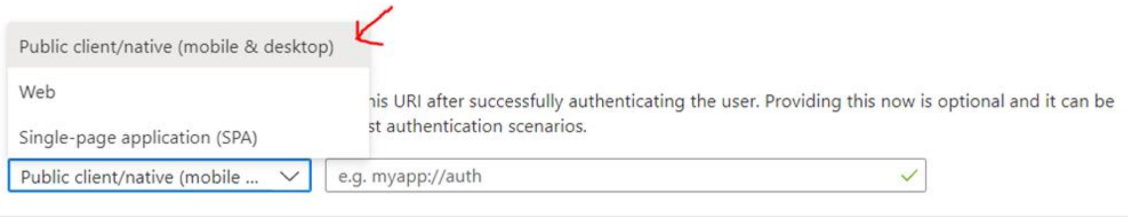
● Accounts in this organizational directory only (diquital only - Single tenant)

○ 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

7. In the **Redirect URI (optional)** section, select the **Public client/native**.

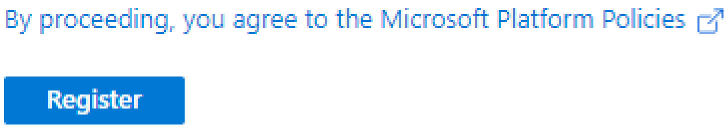


Public client/native (mobile & desktop)
Web
Single-page application (SPA)

his URI after successfully authenticating the user. Providing this now is optional and it can be st authentication scenarios.

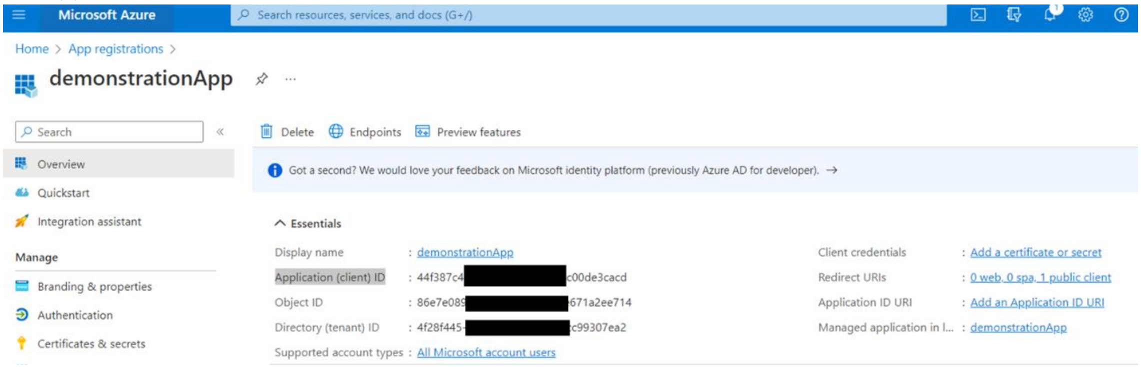
Public client/native (mobile ... | e.g. myapp://auth

8. Once all the values have been entered in the form, click on the Register button (at the end of the screen):



By proceeding, you agree to the Microsoft Platform Policies

Once the application is registered, a window will open with information that you will need to connect to the application

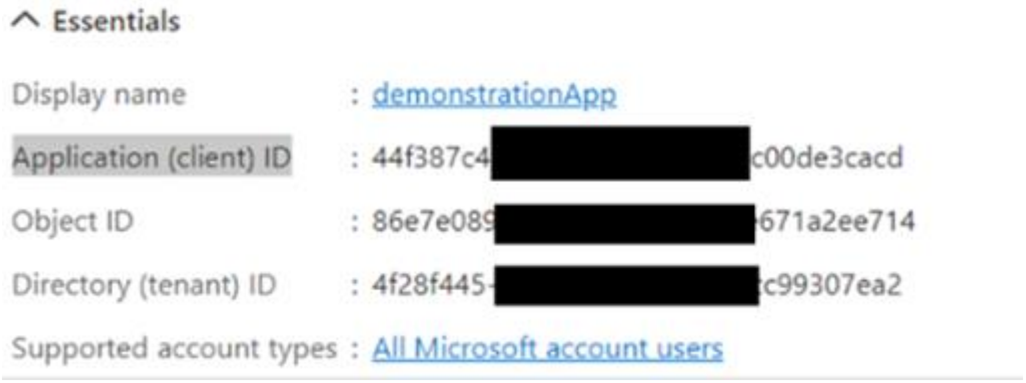


Got a second? We would love your feedback on Microsoft identity platform (previously Azure AD for developer). →

Essentials

Display name	: demonstrationApp	Client credentials	: Add a certificate or secret
Application (client) ID	: 44f387c4[redacted]c00de3cacd	Redirect URIs	: 0 web, 0 spa, 1 public client
Object ID	: 86e7e089[redacted]671a2ee714	Application ID URI	: Add an Application ID URI
Directory (tenant) ID	: 4f28f445-[redacted]c99307ea2	Managed application in l...	: demonstrationApp
Supported account types	: All Microsoft account users		

The fields that you will need are: **Application (client) ID** and **Directory (tenant) ID**



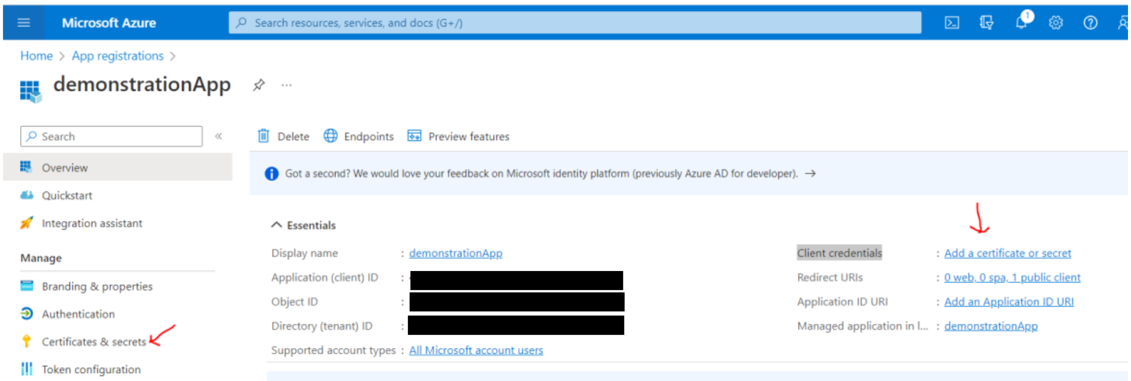
Essentials

Display name	: demonstrationApp
Application (client) ID	: 44f387c4[redacted]c00de3cacd
Object ID	: 86e7e089[redacted]671a2ee714
Directory (tenant) ID	: 4f28f445-[redacted]c99307ea2
Supported account types	: All Microsoft account users

APPLICATION ACCESS CONFIGURATION

Once the application is registered, we will generate the application access key by following the steps:

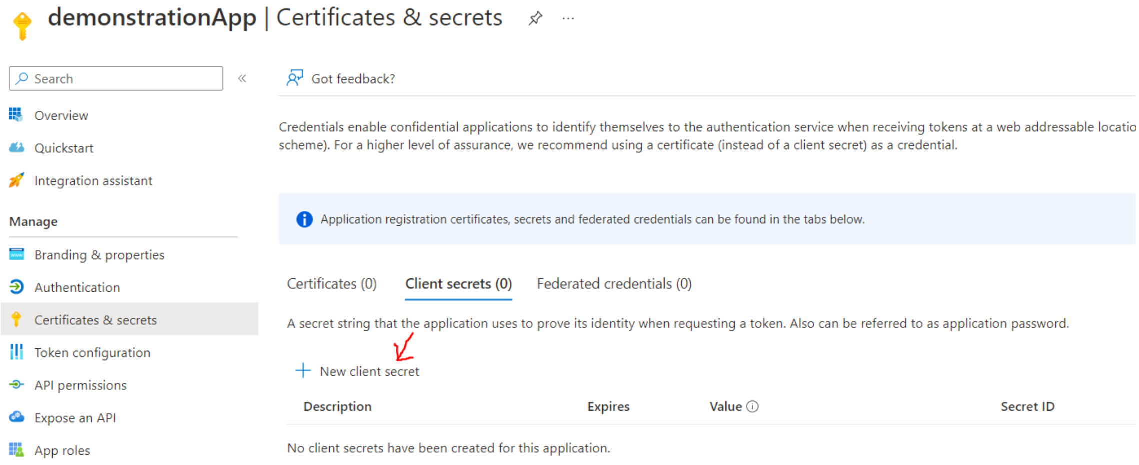
1. Go to the **Certificates & secrets** console for the specific application.



Got a second? We would love your feedback on Microsoft identity platform (previously Azure AD for developer). →

Display name	: demonstrationApp	Client credentials	: Add a certificate or secret
Application (client) ID	: [redacted]	Redirect URIs	: 0 web, 0 spa, 1 public client
Object ID	: [redacted]	Application ID URI	: Add an Application ID URI
Directory (tenant) ID	: [redacted]	Managed application in l...	: demonstrationApp
Supported account types	: All Microsoft account users		

2. We will create a new Secret client by clicking the button **New client secret**



Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable locatio scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

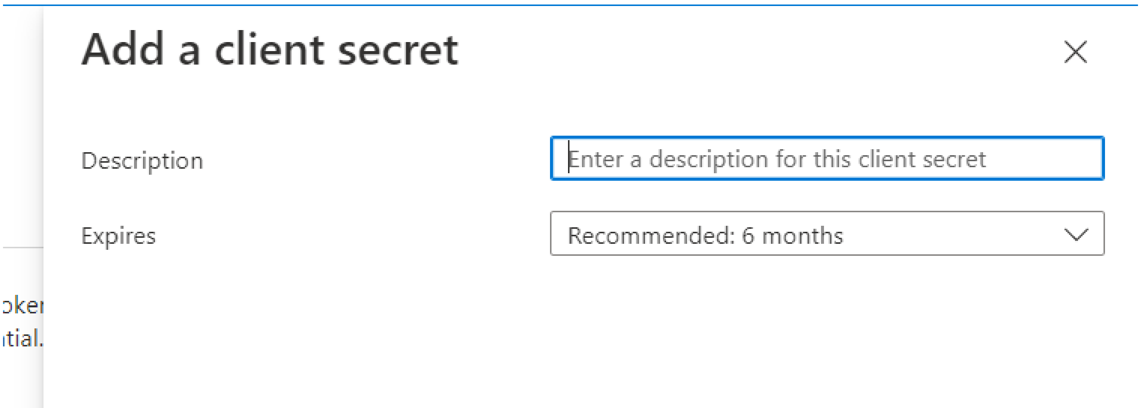
Application registration certificates, secrets and federated credentials can be found in the tabs below.

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	Secret ID

No client secrets have been created for this application.

3. We will write a descriptive name, given that its periodic renewal will have to be administered.

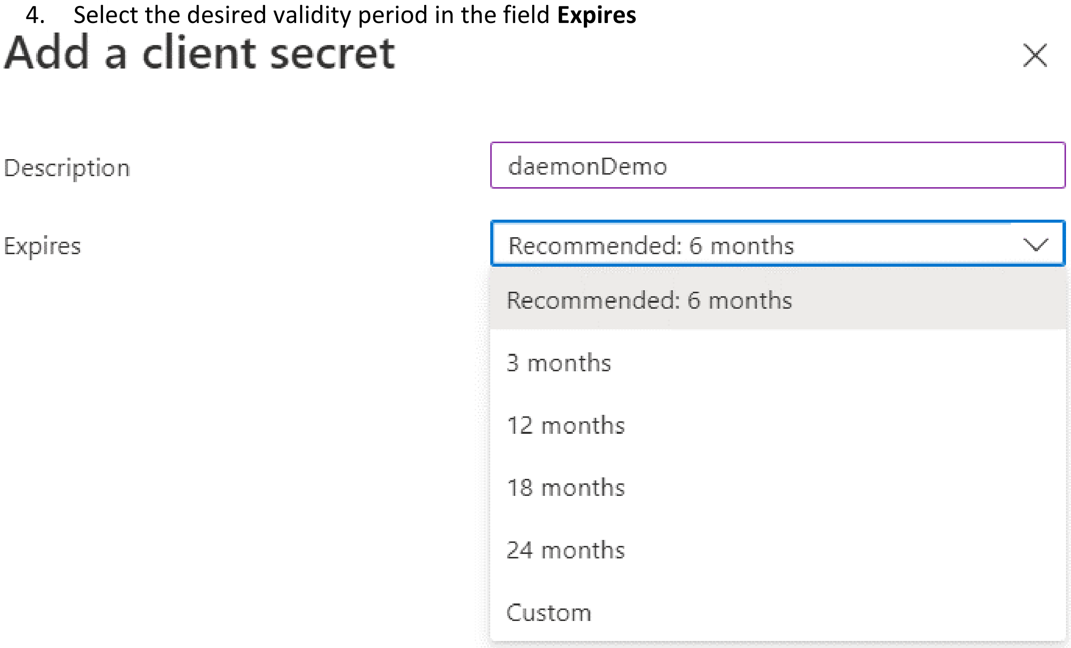


Add a client secret

Description: Enter a description for this client secret

Expires: Recommended: 6 months

4. Select the desired validity period in the field **Expires**

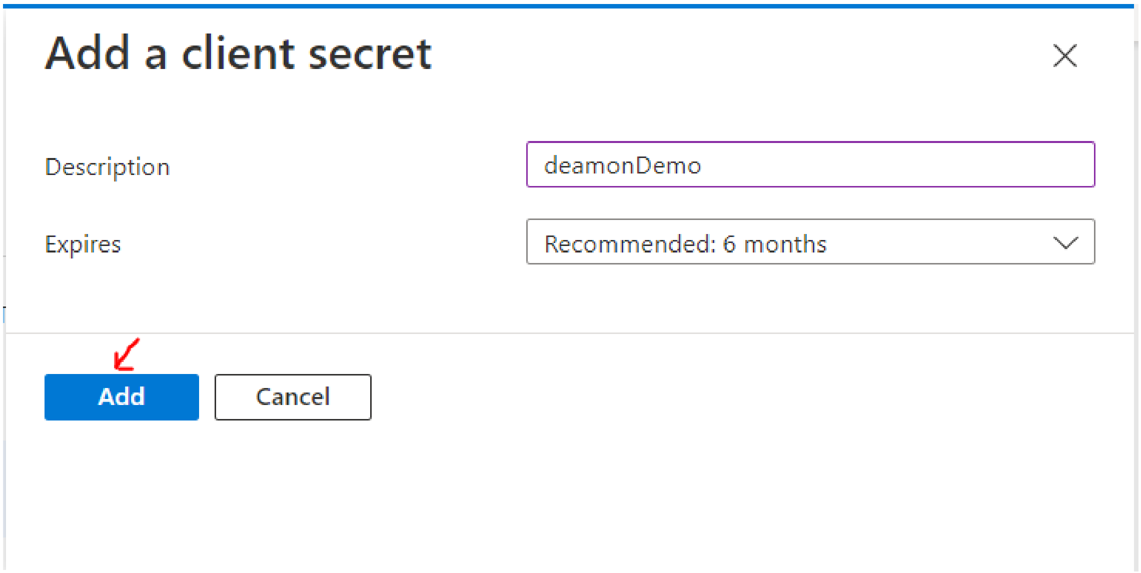


Add a client secret

Description: daemonDemo

Expires: Recommended: 6 months

5. We will create the secret by clicking on the **Add** button

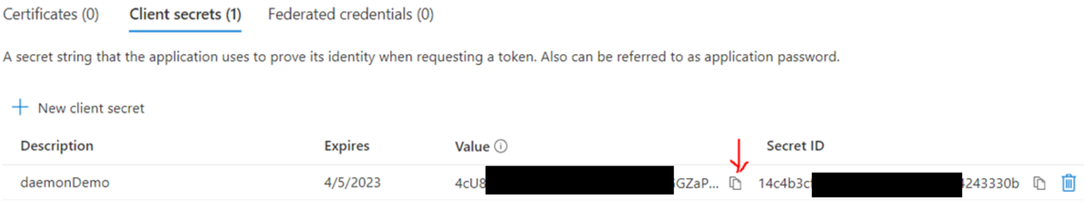


Add a client secret

Description: deamonDemo

Expires: Recommended: 6 months

6. Save the value of secrecy. **Important**, once you refresh the page, this value will no longer be visible. (If this value cannot be extracted in this step you can always re-create a new secret and delete the existing one.) This secret should be treated as sensitive information.



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	Secret ID
daemonDemo	4/5/2023	4cU8[redacted]GZaP...	14c4b3c[redacted]243330b

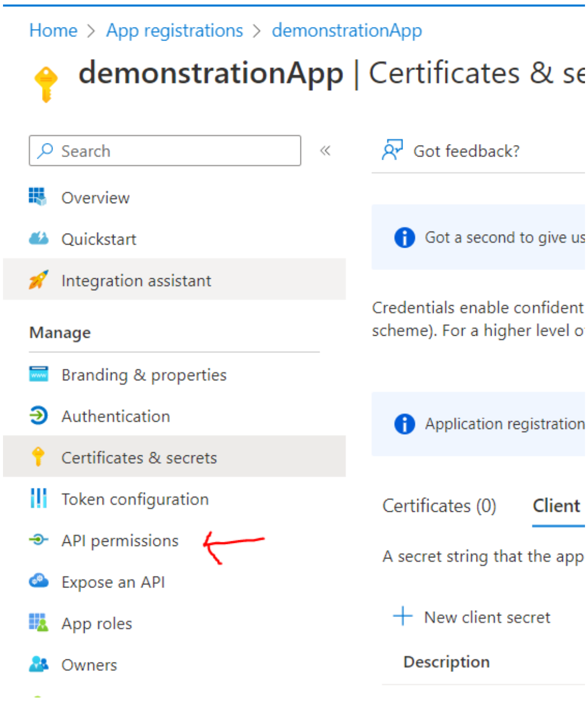
7. Store the “Client Secret Value” in a secure place

CONFIGURATION OF APPLICATION PERMISSIONS

Once our application can access the Office365 platform, we need to define which services it can access and which operations it can perform on them. This guide will follow a read write email use cases using the graph API in an application daemon scenario.

To better understand what permissions, you want to give for you specific use case consult the Graph API documentation

1. Navigate to the “API permissions” console located on the left side of the screen



Home > App registrations > demonstrationApp

demonstrationApp | Certificates & se

Search

Overview
Quickstart
Integration assistant

Manage

Branding & properties
Authentication
Certificates & secrets
Token configuration
API permissions
Expose an API
App roles
Owners

Got feedback?

Got a second to give us

Credentials enable confident
scheme). For a higher level o

Application registration

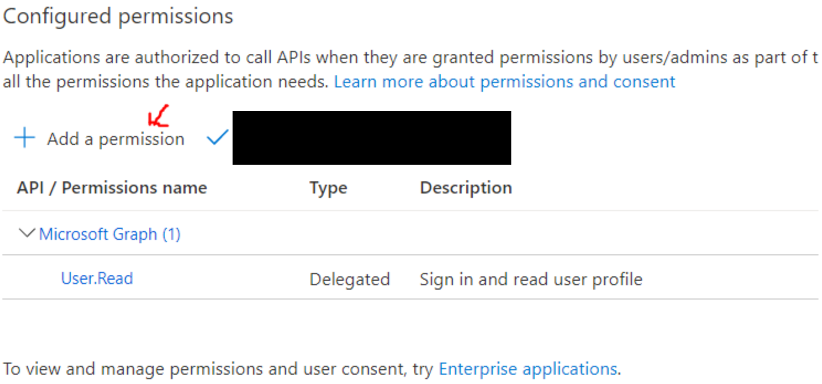
Certificates (0) Client

A secret string that the app

New client secret

Description

2. Add the necessary permissions by clicking the button **Add a permission**



Configured permissions

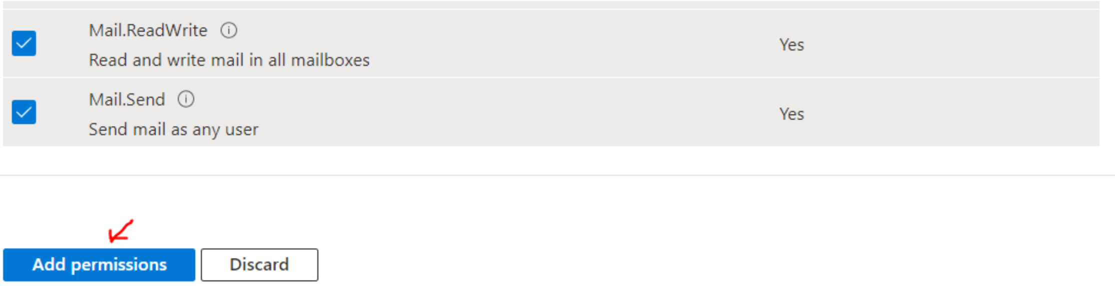
Applications are authorized to call APIs when they are granted permissions by users/admins as part of t
all the permissions the application needs. Learn more about permissions and consent

+ Add a permission ✓ [redacted]

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

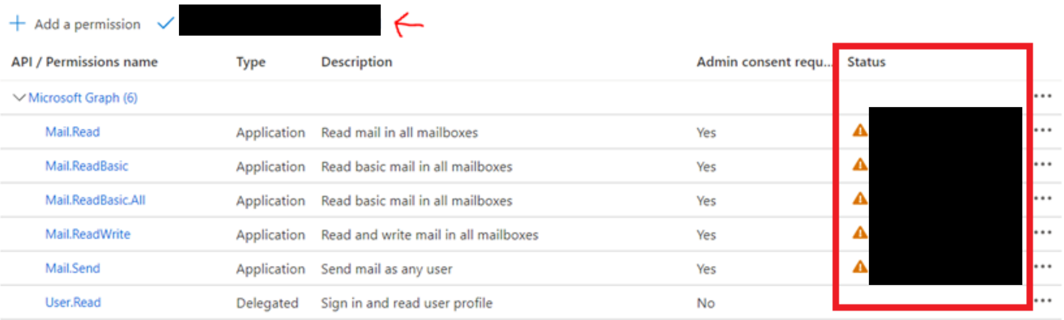
To view and manage permissions and user consent, try Enterprise applications.

3. We will finish assigning permissions by clicking the button **Add permissions**



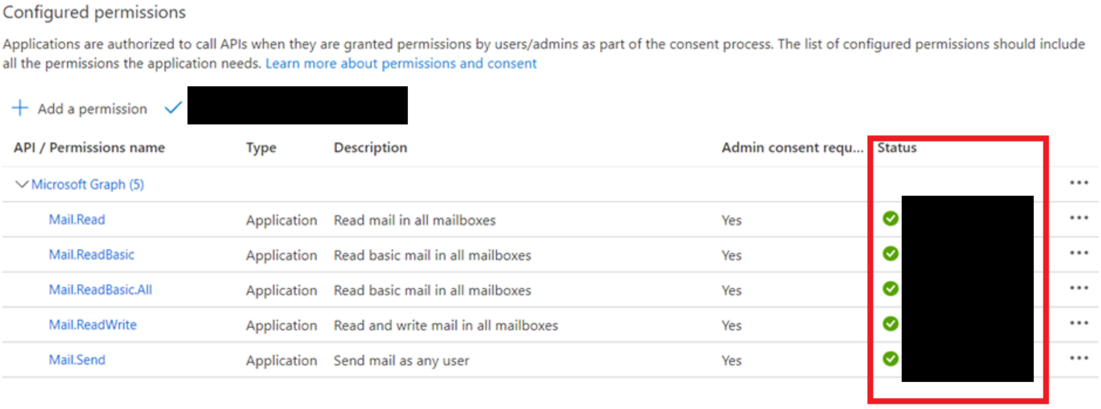
☑	Mail.ReadWrite ⓘ Read and write mail in all mailboxes	Yes
☑	Mail.Send ⓘ Send mail as any user	Yes

4. On the next screen we will have to verify in the status column if consent is required at the level of the organization's administrator.
   This is necessary so that the application can access user data (the "Restrict application access" explains how to limit which users the application can access)



API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (6)				
Mail.Read	Application	Read mail in all mailboxes	Yes	⚠ [redacted]
Mail.ReadBasic	Application	Read basic mail in all mailboxes	Yes	⚠ [redacted]
Mail.ReadBasic.All	Application	Read basic mail in all mailboxes	Yes	⚠ [redacted]
Mail.ReadWrite	Application	Read and write mail in all mailboxes	Yes	⚠ [redacted]
Mail.Send	Application	Send mail as any user	Yes	⚠ [redacted]
User.Read	Delegated	Sign in and read user profile	No	

5. We will give consent to all the permissions by clicking on the **Grant admin consent for...** button and confirm that we no longer have exclamation points in the 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 configured permissions should include all the permissions the application needs. Learn more about permissions and consent

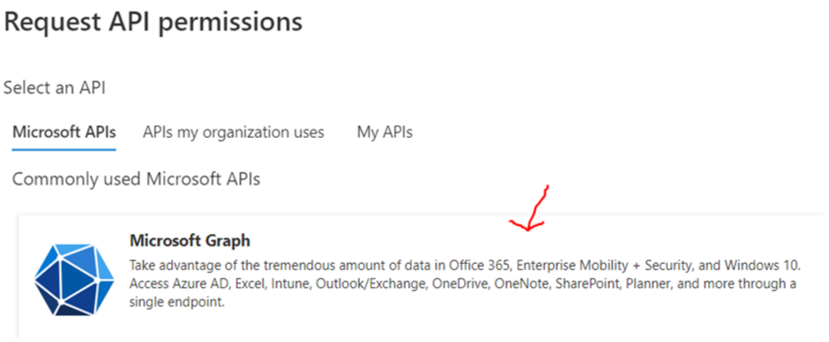
API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (5)				
Mail.Read	Application	Read mail in all mailboxes	Yes	✓ [redacted]
Mail.ReadBasic	Application	Read basic mail in all mailboxes	Yes	✓ [redacted]
Mail.ReadBasic.All	Application	Read basic mail in all mailboxes	Yes	✓ [redacted]
Mail.ReadWrite	Application	Read and write mail in all mailboxes	Yes	✓ [redacted]
Mail.Send	Application	Send mail as any user	Yes	✓ [redacted]

NOTE: It is recommended to remove the default **User.Read** permission for security reasons.

API GRAPH

The Graph API provides most of the functionality of the Office365 platform and therefore most of the services and operations that we can perform with our application will be under this umbrella.

From the add permissions screen, select the Microsoft Graph API



Request API permissions

Select an API

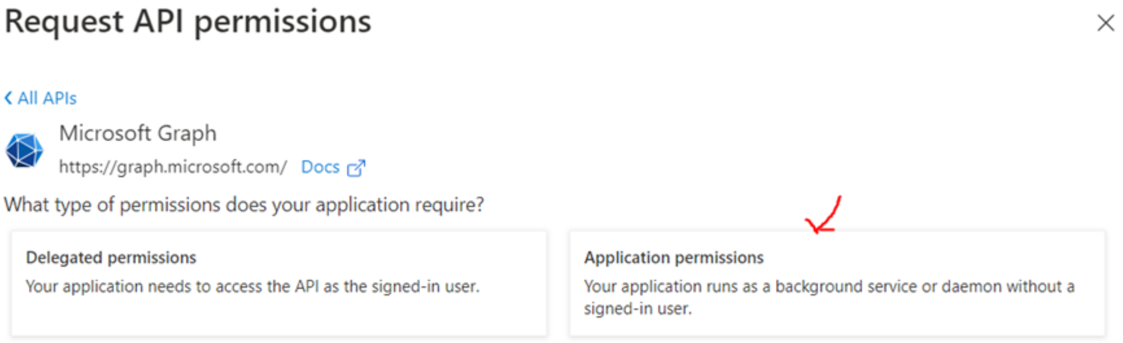
Microsoft APIs | APIs my organization uses | My APIs

Commonly used Microsoft APIs

Microsoft Graph
Take advantage of the tremendous amount of data in Office 365, Enterprise Mobility + Security, and Windows 10. Access Azure AD, Excel, Intune, Outlook/Exchange, OneDrive, OneNote, SharePoint, Planner, and more through a single endpoint.

In the next screen you can select 2 different types of permissions

- **Delegated**: which require user authentication
- **Application:** which do not, (ideal for our application daemon use case)



Request API permissions

< All APIs

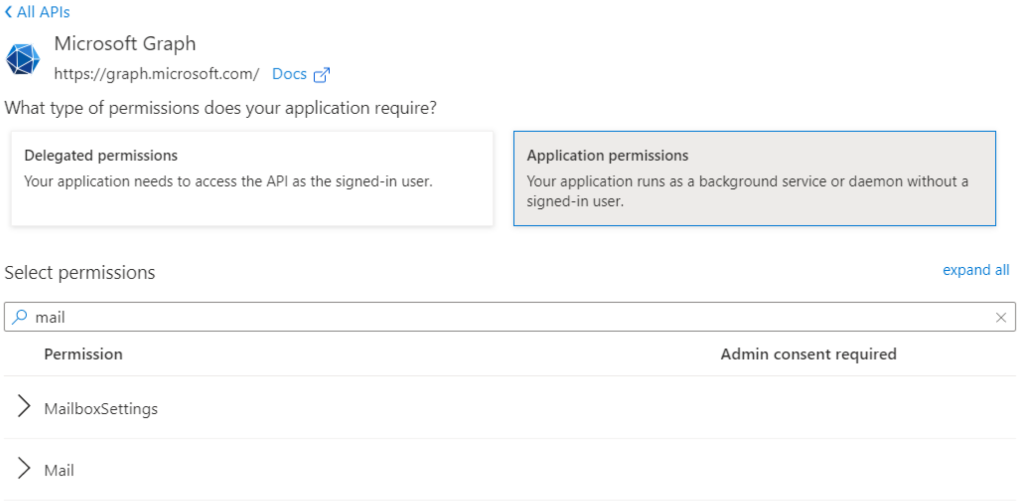
Microsoft Graph
https://graph.microsoft.com/ Docs

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.

The next screen lists all the operations that can be done with this API. They are shown categorized, but it also allows searches to make it easier to find what we need.



< 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 | expand all

mail

Permission	Admin consent required
> MailboxSettings	
> Mail	

READ EMAILS

From the Graph API permissions screen, we will add the following permissions (if we write it in the search bar, it will appear on the screen without having to search for it):

- Mail.Read

SEND EMAILS

- Mail.ReadWrite
- Mail.Send

RESTRICT APPLICATION ACCESS

To limit the level of access (user data) to which the application can access, an application policy must be defined that restricts set access.

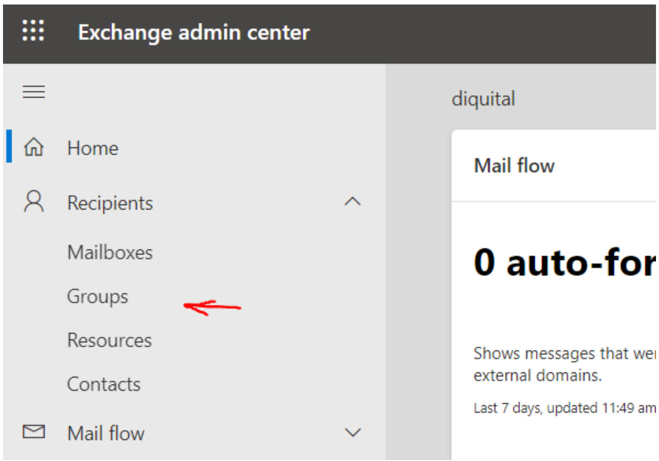
In the case of Exchange, it is also necessary to create a security group that must be added to the application's policy.

Other services may require additional measures that should be carefully assessed.

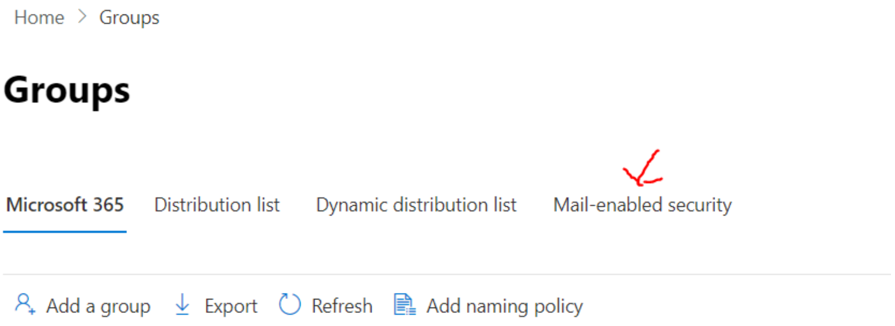
CREATE A MAIL-ENABLED SECURITY GROUP

In order to define which users/mailboxes the application can access; we will create security groups and add only the essential users for the application.

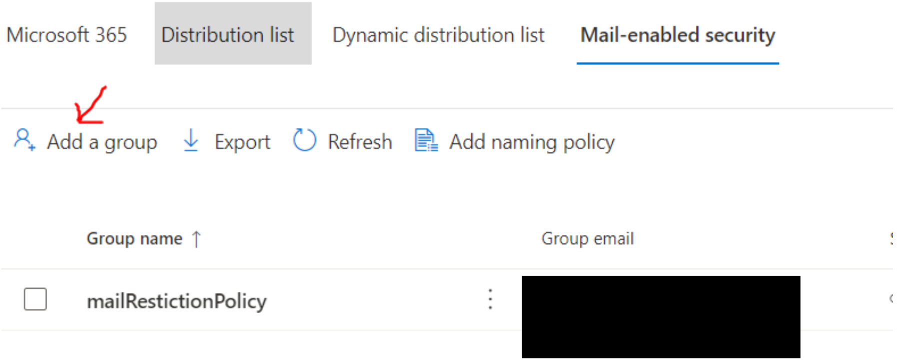
1. Access admin.exchange.microsoft.com and log in as an administrator.
2. Navigate to the "Groups" screen



3. Within the Groups screen, click on **Mail-enabled security**

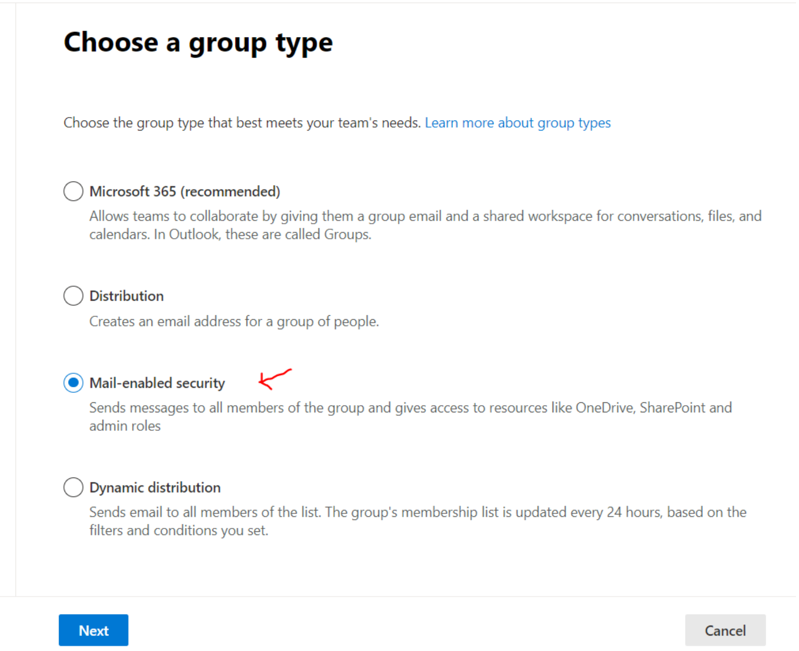


4. Create a new group by clicking the button **Add a group**



	Group name ↑	Group email
□	mailRestictionPolicy	[redacted]

5. Select the **Mail-enabled security** group type and press the **Next** button



Choose a group type

Choose the group type that best meets your team's needs. Learn more about group types

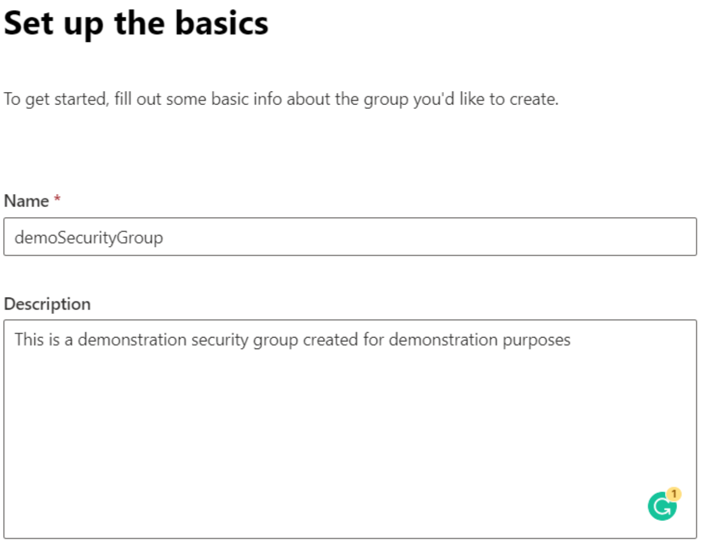
○ Microsoft 365 (recommended)
Allows teams to collaborate by giving them a group email and a shared workspace for conversations, files, and calendars. In Outlook, these are called Groups.

○ Distribution
Creates an email address for a group of people.

● Mail-enabled security
Sends messages to all members of the group and gives access to resources like OneDrive, SharePoint and admin roles

○ Dynamic distribution
Sends email to all members of the list. The group's membership list is updated every 24 hours, based on the filters and conditions you set.

6. Define the desired name and a description (provide meaningful values) for the group and click **Next**



Set up the basics

To get started, fill out some basic info about the group you'd like to create.

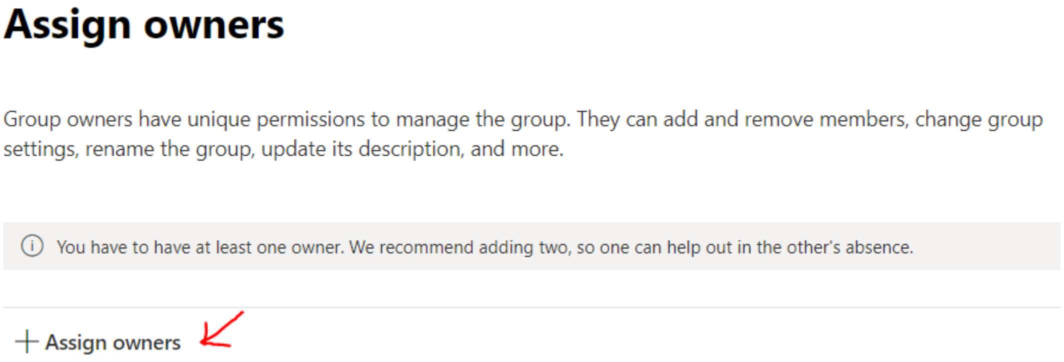
Name *

demoSecurityGroup

Description

This is a demonstration security group created for demonstration purposes

7. Press the **Assign owners** button to select the users who will be the owner of this group, this will allow them to add and administer the members of this group.

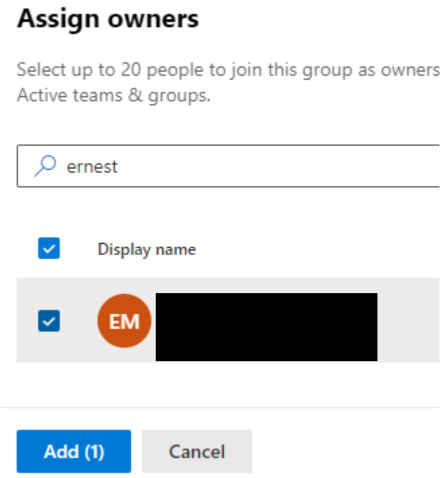


Assign owners

Group owners have unique permissions to manage the group. They can add and remove members, change group settings, rename the group, update its description, and more.

You have to have at least one owner. We recommend adding two, so one can help out in the other's absence.

8. Select the users that we want to assign as owners and click the **Add** button (it is recommend adding only users with an administrator profile)



Assign owners

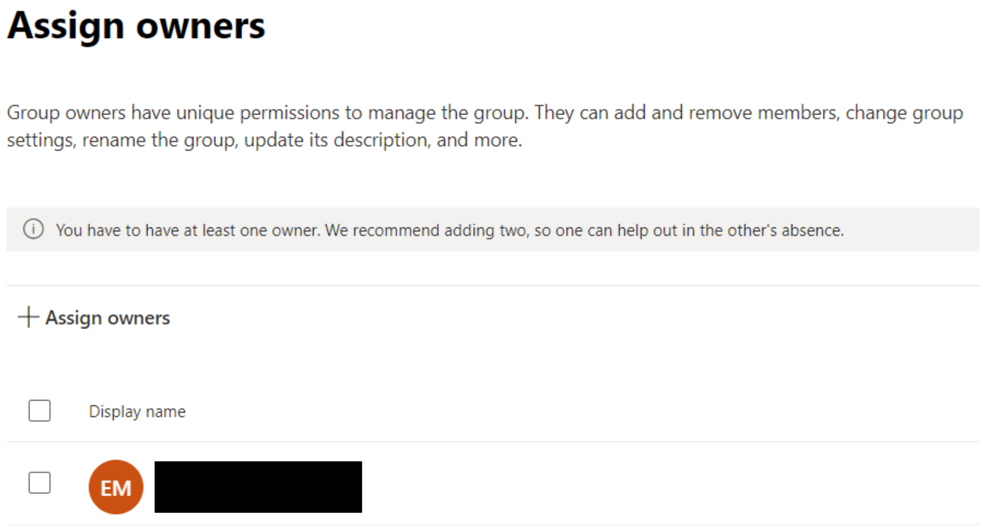
Select up to 20 people to join this group as owners Active teams & groups.

ernest

☑ Display name

☑ EM [redacted]

9. Once the owners have been selected, select **Next**.

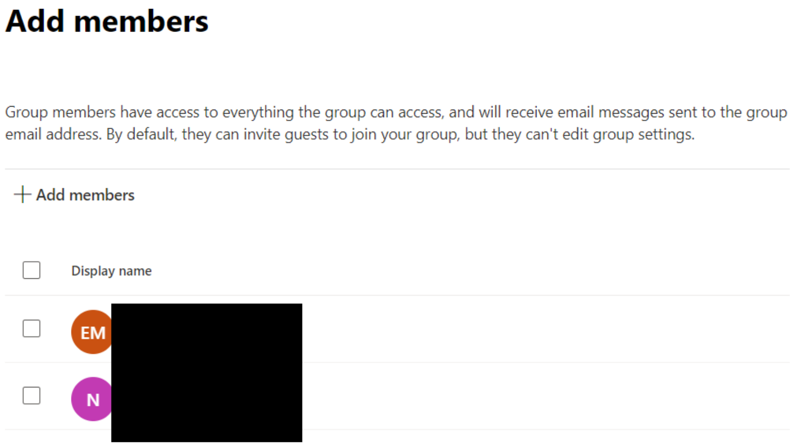


Assign owners

Group owners have unique permissions to manage the group. They can add and remove members, change group settings, rename the group, update its description, and more.

ⓘ You have to have at least one owner. We recommend adding two, so one can help out in the other's absence.

10. On this screen we can add the users that will be members of the security group by simply clicking the **Add members** button

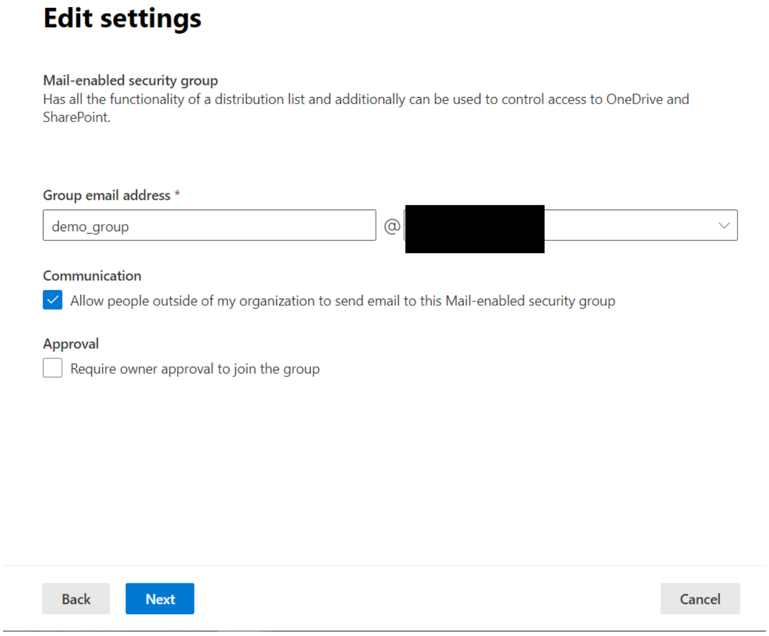


Add members

Group members have access to everything the group can access, and will receive email messages sent to the group email address. By default, they can invite guests to join your group, but they can't edit group settings.

NOTE: When adding members to the group it is important to know that when the new security policies are applied, we will have to select whether to enable or reject the list of members.

11. Create an email address for the group. (**Importantly** we will use it when associating the group with the security policies)



Edit settings

Mail-enabled security group
Has all the functionality of a distribution list and additionally can be used to control access to OneDrive and SharePoint.

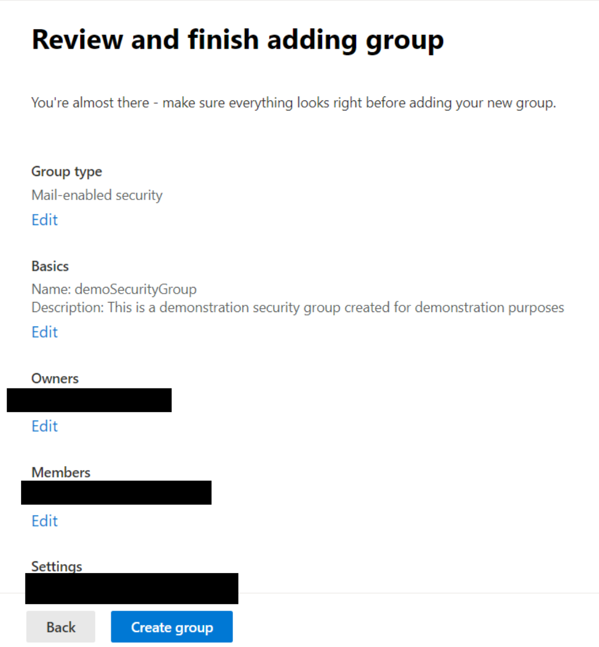
Group email address *
demo_group @ [redacted]

Communication
☑ Allow people outside of my organization to send email to this Mail-enabled security group

Approval
□ Require owner approval to join the group

NOTE: The Communication and Approval options can be checked depending on the organization's needs.

12. Check that all the data has been filled in according to our needs and create the group by clicking on **Create group**.



Review and finish adding group

You're almost there - make sure everything looks right before adding your new group.

Group type
Mail-enabled security
Edit

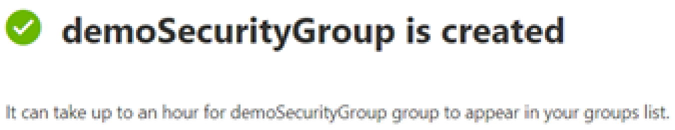
Basics
Name: demoSecurityGroup
Description: This is a demonstration security group created for demonstration purposes
Edit

Owners
[redacted]
Edit

Members
[redacted]
Edit

Settings
[redacted]

13. We get a message confirming that the group has been successfully created..



✓ demoSecurityGroup is created

It can take up to an hour for demoSecurityGroup group to appear in your groups list.

UPDATING THE APPLICATION POLICY

Once the **Mail-enabled security group** is defined, the application policy must be updated to restrict access to the desired mailboxes.

To do this, it is required to have the **ExchangeOnlineManagment** module installed, if you do not have it you can follow the instructions in the link: [About the Exchange Online PowerShell V2 module and V3 module]

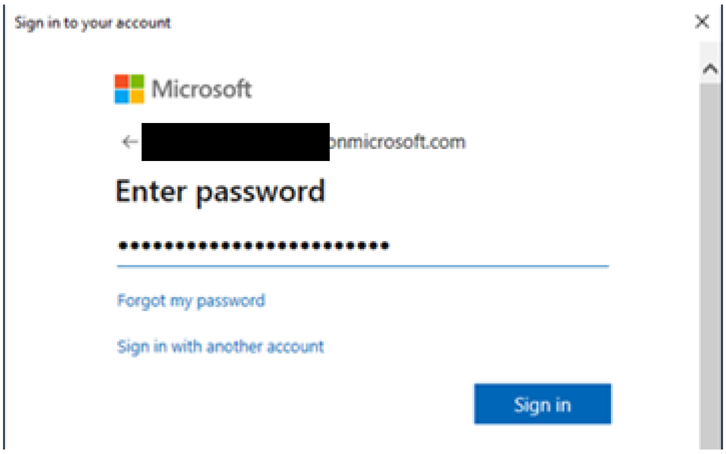
You can also check this other link for information about the **Exchange Online** PowerShell module: [Connect to Exchange Online PowerShell]

Following the instructions, we can assign the Mail-enabled security group to the security policy of our application:

1. Open powershell
2. Connect to the online exchange (with a Microsoft tenant administrator user) using the command:

```
Connect-ExchangeOnline -UserPrincipalName admin@contoso.onmicrosoft.com
```

3. After executing the command, a screen will open asking for credentials.



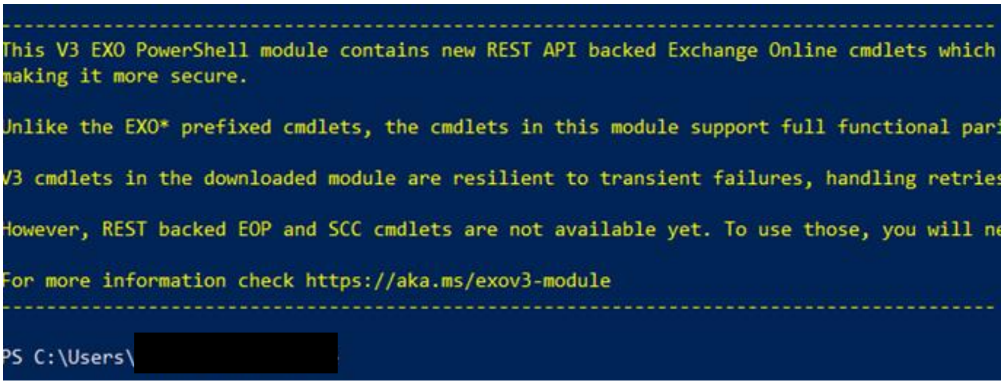
Sign in to your account

← [redacted]onmicrosoft.com

Enter password

••••••••••••••••••••••••

4. Once successfully connected you will see the message below.



```
----------------------------------------------------------------------------------------
This V3 EXO PowerShell module contains new REST API backed Exchange Online cmdlets which
making it more secure.

Unlike the EXO* prefixed cmdlets, the cmdlets in this module support full functional par

V3 cmdlets in the downloaded module are resilient to transient failures, handling retrie

However, REST backed EOP and SCC cmdlets are not available yet. To use those, you will ne

For more information check https://aka.ms/exov3-module
----------------------------------------------------------------------------------------

PS C:\Users\[redacted]
```

5. At this point we must decide whether to allow or deny access to the members of the **Mail-enabled security group** that we created previously.

   Allow access: Apply the RestrictAccess value to the -AccessRight flag. This restriction means that only mailboxes associated with the group can be accessed.

```
New-ApplicationAccessPolicy -AppId e7e4dbfc-XXXX-YYYY-ZZZZ-2ae8f144f59b -
PolicyScopeGroupId EvenUsers@contoso.com -AccessRight RestrictAccess -Description
"data"
```

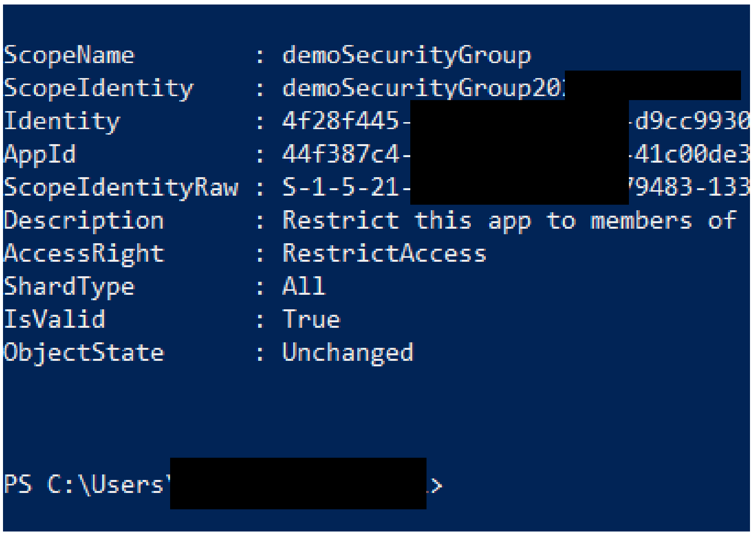
   Deny access: Apply the DenyAccess value to the -AccessRight flag. This restriction will grant access to all mailboxes that are not within this access group. But will block access to those that are.

```
New-ApplicationAccessPolicy -AppId e7e4dbfc-XXXX-YYYY-ZZZZ-2ae8f144f59b -
PolicyScopeGroupId EvenUsers@contoso.com -AccessRight DenyAccess -Description "data"
```

   For more policy creation information see New-ApplicationAccessPolicy

6. Run the command, duly constructed in the previous step, using the AppId of our application.

7. If everything worked correctly, we will get the following message



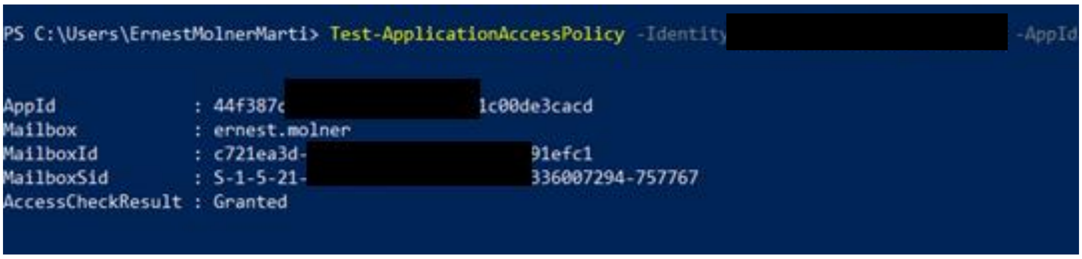
```
ScopeName          : demoSecurityGroup
ScopeIdentity      : demoSecurityGroup20[redacted]
Identity           : 4f28f445-[redacted]-d9cc9930
AppId              : 44f387c4-          -41c00de3
ScopeIdentityRaw   : S-1-5-21-          9483-133
Description        : Restrict this app to members of
AccessRight        : RestrictAccess
ShardType          : All
IsValid            : True
ObjectState        : Unchanged


PS C:\Users[redacted]>
```

To verify that our application has access to the selected mailboxes, we can execute the following command:

```
Test-ApplicationAccessPolicy -Identity ernest.molner@contoso.com -AppId
```

Using the mailbox you want to test in “-Identity” and “-AppId” with the application id



```
PS C:\Users\ErnestMolnerMarti> Test-ApplicationAccessPolicy -Identity [redacted] -AppId

AppId             : 44f387c[redacted]1c00de3cacd
Mailbox           : ernest.molner
MailboxId         : c721ea3d-[redacted]91efc1
MailboxSid        : S-1-5-21-[redacted]336007294-757767
AccessCheckResult : Granted
```

By validating the result obtained in AccessCheckResult we will know if the application has access or not:

- Granted: The application can access the user's mailbox
- Denied: The application cannot access

Important: Changes to groups and security policies may take up to an hour to update.

We recommend not handing in any Application Id or Secret before this time, otherwise developers could access potentially sensitive information.

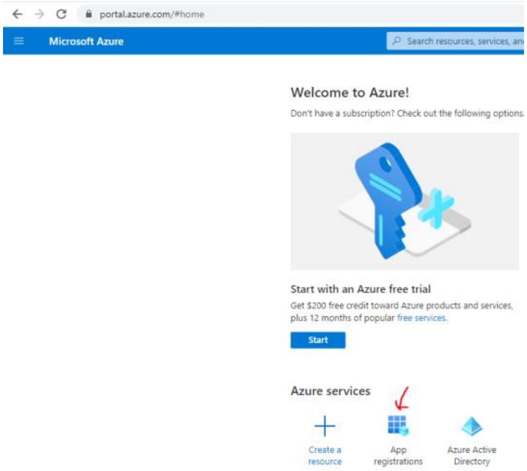
RENEWAL OF APPLICATION SECRETS

Application secrets have a maximum validity period of 2 years, so they must be renewed and updated.

If they were not renewed within the validity period, the applications would no longer have access to the Office365 platform, effectively generating a loss of functionality for our applications

In order not to interrupt access to the applications we must carry out the following procedure a few weeks before the expiry of the Secrets of the applications.

1. Access the platform: portal.azure.com (with an administrator user) and navigate to "App Registrations"



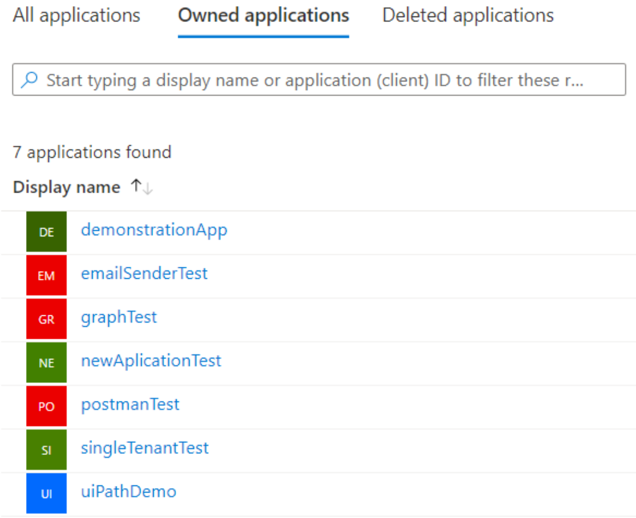
Welcome to Azure!

Don't have a subscription? Check out the following options.

Start with an Azure free trial

Get $200 free credit toward Azure products and services, plus 12 months of popular free services.

2. Click on the application where the Secret to be renewed is located



Start typing a display name or application (client) ID to filter these r...

7 applications found

	Display name ↑↓
DE	demonstrationApp
EM	emailSenderTest
GR	graphTest
NE	newAplicationTest
PO	postmanTest
SI	singleTenantTest
UI	uiPathDemo

1. Once inside the application follow steps 1 to 7 in the section APPLICATION ACCESS CONFIGURATION