A MITEL
PRODUCT
GUIDE

OpenScape Business V3

Microsoft Exchange Online / Office 365
configure OAuth 2.0

Release Number 09/2024

Contents

History of Changes

Date	Issue	Summary
02.12.2022	1.0	release version for OpenScape Business V3R2.1
23.09.2024	1.1	editorial changes

Note: The basis for this document is the current OpenScape Business at the time of certification. Since OpenScape Business is constantly developed, input masks and interfaces as well as requirements may change in the future. The settings and entries described here then apply accordingly.

Comments and corrections are welcome, please contact: osbiz-certification@mitel.com

1 E-Mail Forwarding

The instructions below will cover the necessary steps to configure E-Mail Forwarding using the OAuth 2.0 authentication method with Microsoft Office 365.

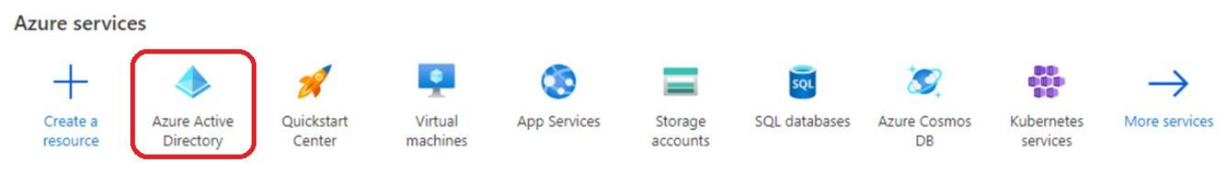
OpenScape Business uses postfix to connect to an exchange server (relay server) to send email notifications to required recipients. Till now this was done using the basic authentication scheme. Microsoft will obsolete the basic authentication method to all its **Exchange Online servers** in favor of the OAuth 2.0 authentication scheme.

The configuration will be split in two parts. The 1st part covers the registration of the application in the Microsoft Azure Active Directory portal; the 2nd part will show how to create the E-mail forwarding in WBM using the information from the previous parts.

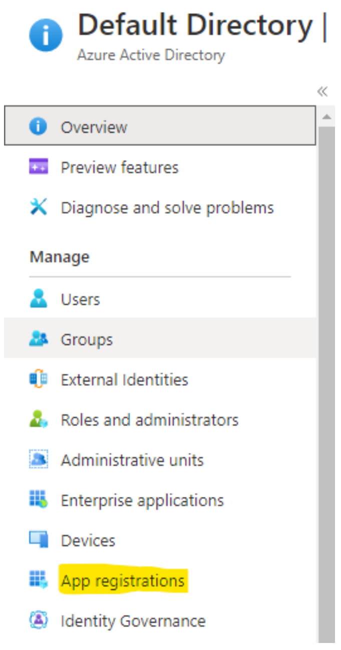
Hint: SMTP AUTH must be enabled at least for the used service account (this is often disabled organization-wide). Refer to: Enable or disable SMTP AUTH in Exchange Online | Microsoft Learn

1.1 Register the Application in Microsoft Azure AD

Using Microsoft OAuth 2.0 authentication to send emails from OpenScape Business requires a Microsoft Azure Active Directory Application defined in the Azure Active Directory the user you are trying to send email as belongs to. Open the browser and go to the Azure Active Directory admin center and login with your Administrator account: https://aad.portal.azure.com/ go to **Azure Active Directory**:

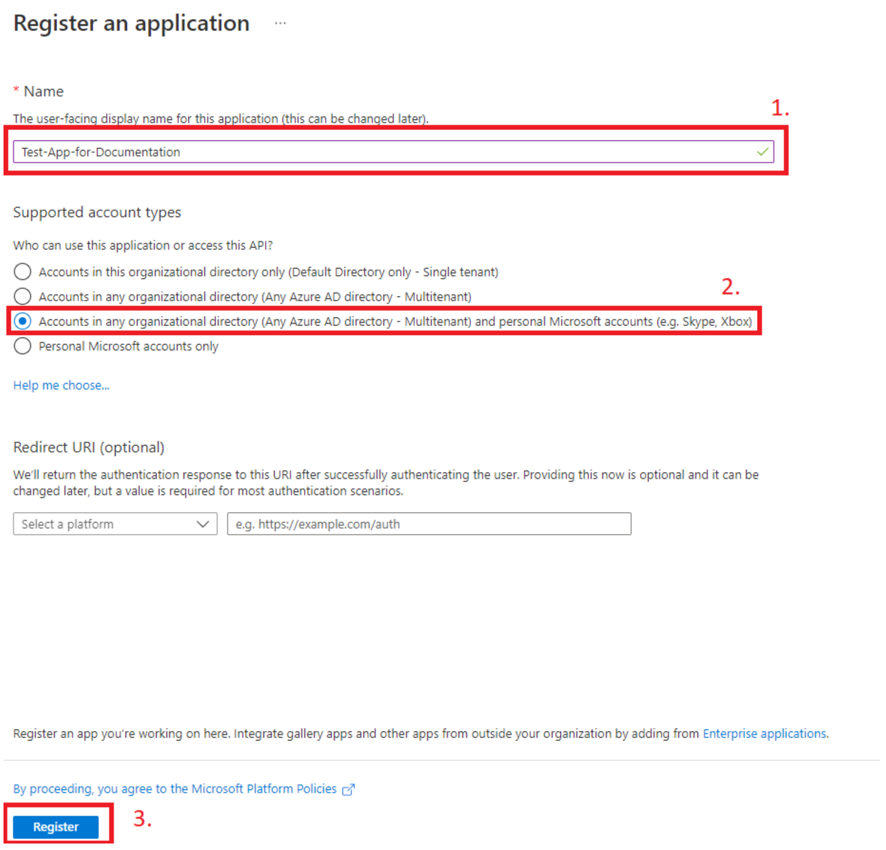


and select "**App Registrations**":



On the top, select "**New Registration**" and on the new application registration page

- Give it a Name
- On the "**Supported account types**" section, select whatever option suits your use case. This option restricts who can authenticate using this application. In the example below, the most permissive option is selected (depends on customer needs).
- Leave Redirect URI empty


Register an application ···

* Name

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

Test-App-for-Documentation

Supported account types

Who can use this application or access this API?

- ○ Accounts in this organizational directory only (Default Directory 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.

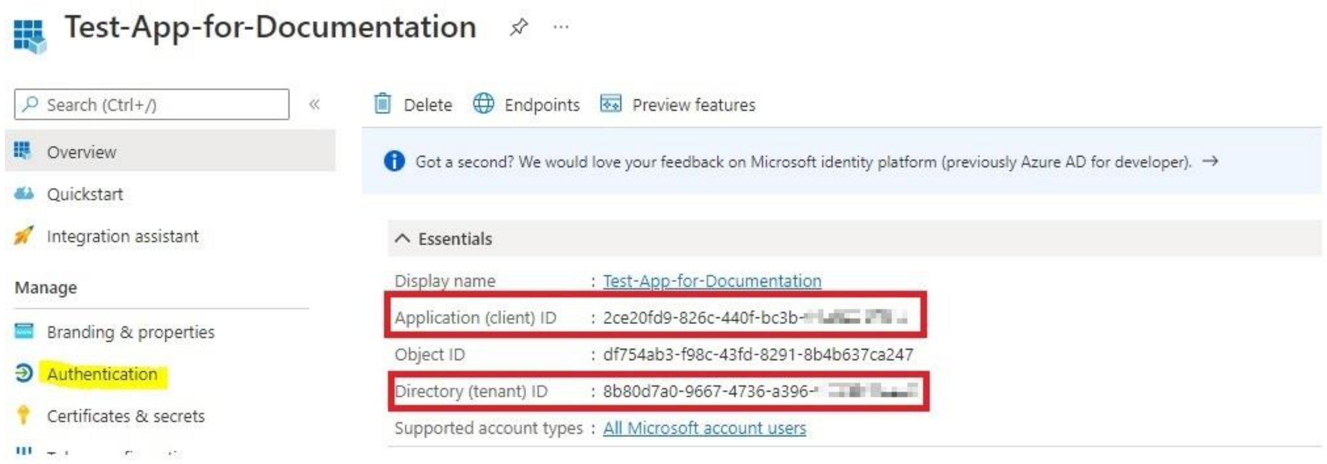
Select a platform | e.g. https://example.com/auth

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from Enterprise applications.

By proceeding, you agree to the Microsoft Platform Policies

	The same application can be used for all services if the respective API permissions are assigned and granted. For documentation purpose we choose a separate application with the example name: `Test-App-for-Documentation`

At the end of the process, Azure AD will present some information about the newly registered application. At this point, it's a good idea to take note of the "**Application ID**" (also known as client ID) and "**Directory (Tenant) ID**", as that information will be used to double-check the second parts of this configuration:

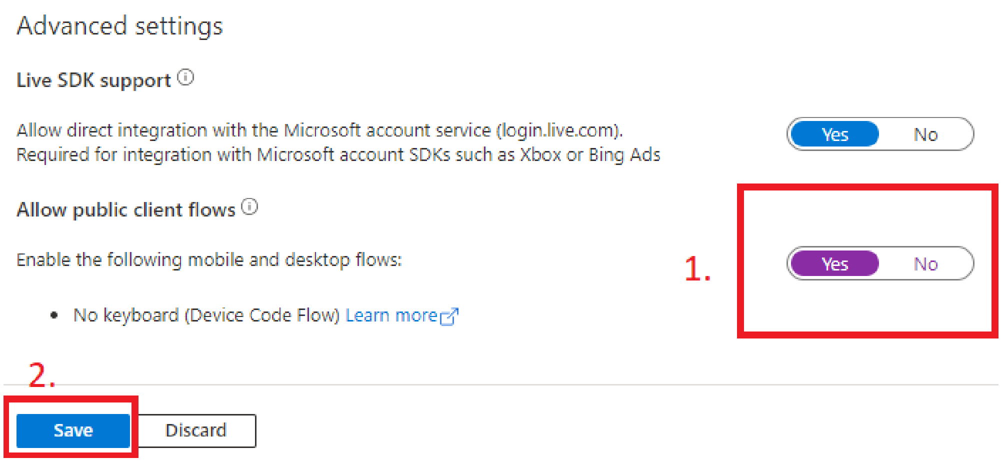


Got a second? We would love your feedback on Microsoft identity platform (previously Azure AD for developer). →

Essentials

Display name	: Test-App-for-Documentation
Application (client) ID	: 2ce20fd9-826c-440f-bc3b-[redacted]
Object ID	: df754ab3-f98c-43fd-8291-8b4b637ca247
Directory (tenant) ID	: 8b80d7a0-9667-4736-a396-[redacted]
Supported account types	: All Microsoft account users

In the menu above on the left Sidebar under the Manage subsection, select "**Authentication**". In the Authentication menu, under "**Advanced Settings**" enable the **Device Code Flow**. Don't forget to save your changes.



Advanced settings

Live SDK support ⓘ

Allow direct integration with the Microsoft account service (login.live.com).
Required for integration with Microsoft account SDKs such as Xbox or Bing Ads [✓ Yes] No

Allow public client flows ⓘ

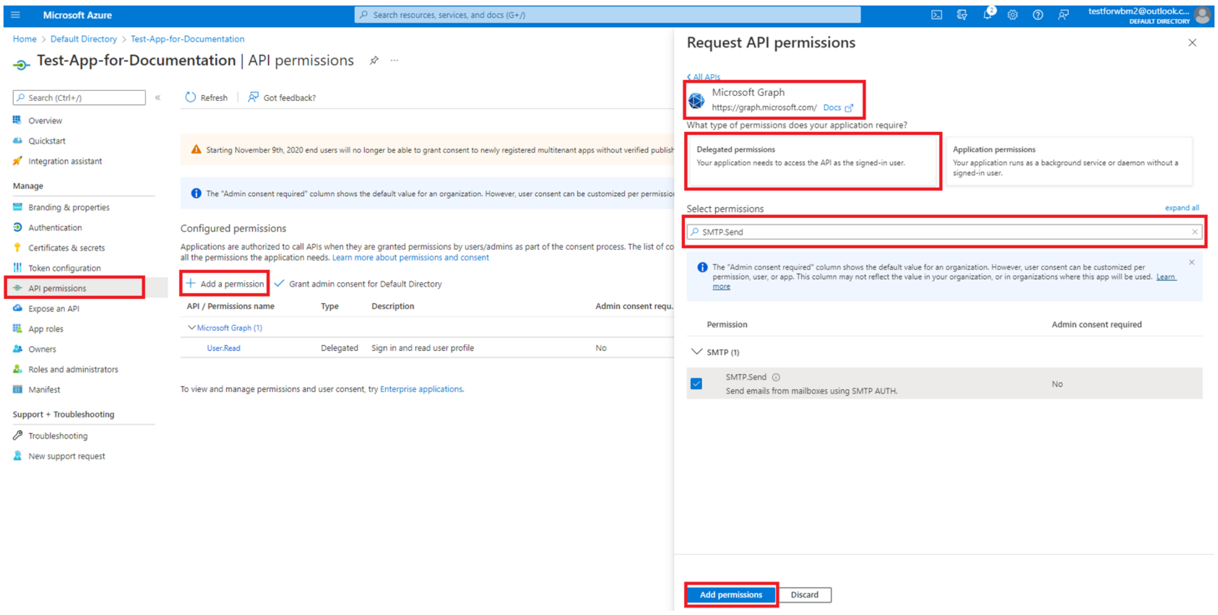
Enable the following mobile and desktop flows:

- No keyboard (Device Code Flow) Learn more

1. [✓ Yes] No

2\.

In the sidebar, select "**API Permissions**". To add permissions for your app, select **Add a permission** → Microsoft Graph API → Delegated Permissions → Search for the required permissions:



Starting November 9th, 2020 end users will no longer be able to grant consent to newly registered multitenant apps without verified publish

The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permissio

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 co all the permissions the application needs. Learn more about permissions and consent

API / Permissions name	Type	Description	Admin consent requ.
Microsoft Graph (1)			
User.Read	Delegated	Sign in and read user profile	No

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

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

SMTP.Send

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
SMTP (1)	
✓ SMTP.Send Send emails from mailboxes using SMTP AUTH.	No

The required permissions for OpenScape Business E-mail forwarding are:

- SMTP.Send
- offline_access
- openid

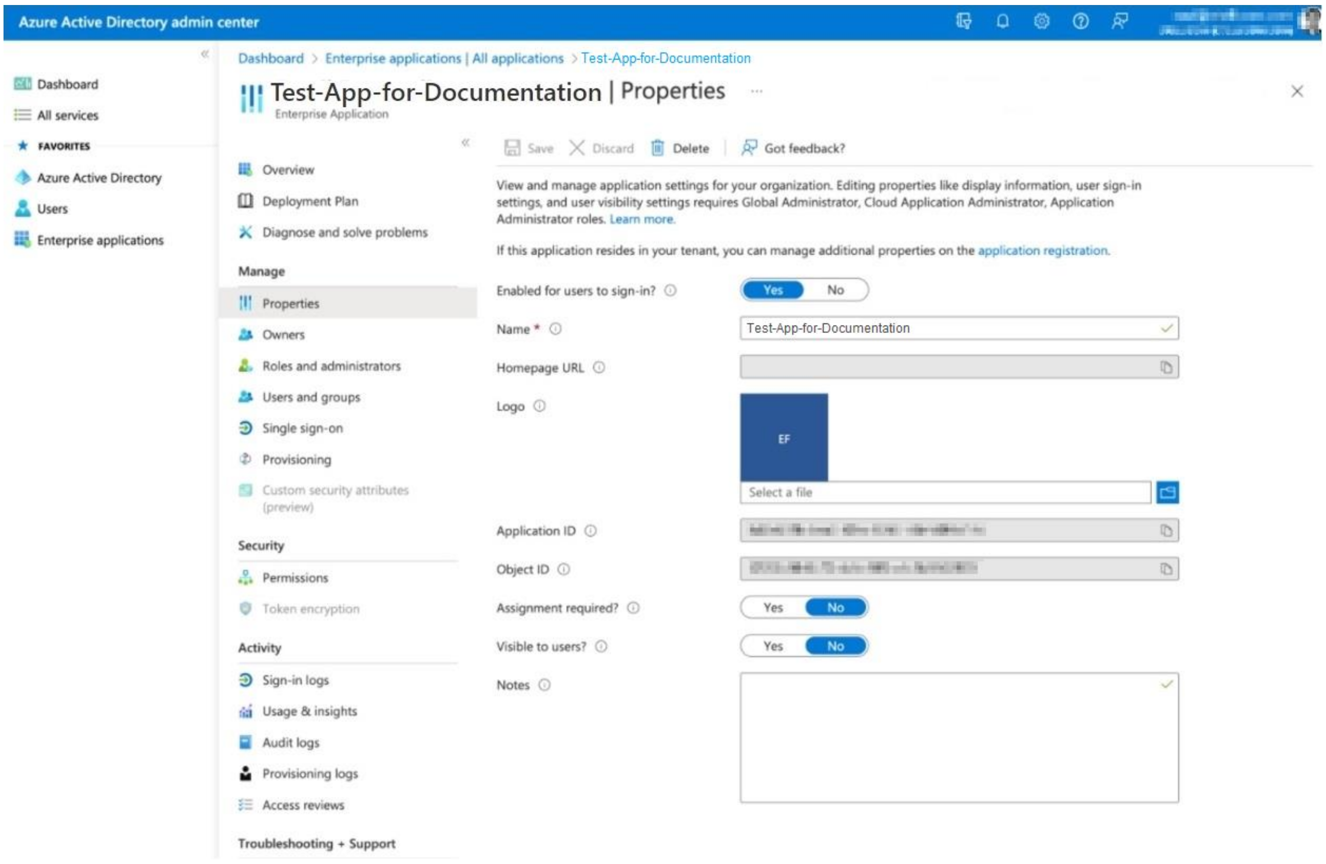
Additional configuration: If you are a subscriber of Office 365 Exchange Online and want to enable user sign-in for Exchange Online, please follow the next steps mentioned in Microsoft's Documentation:

In Azure AD, navigate to **Enterprise Applications**.

- Search for the previous created **"Test-App-for-Documentation"** application. Note, you may need to change the filter to **All Applications**. If it does not exist in that list, add it.

- Select **"Test-App-for-Documentation"** application and in the **Properties** for this app, ensure that the **Enable for users to sign-in?** setting is set to ***Yes***.

You should now be able to use Microsoft OAuth 2.0 authentication for E-mail forwarding in OpenScape Business by using the **Application ID** and the **Tenant ID** (requires enabled user sign-in for Exchange Online in tenant settings).



Test-App-for-Documentation | Properties

Enterprise Application

View and manage application settings for your organization. Editing properties like display information, user sign-in settings, and user visibility settings requires Global Administrator, Cloud Application Administrator, Application Administrator roles. Learn more.

If this application resides in your tenant, you can manage additional properties on the application registration.

Enabled for users to sign-in? [✓ Yes] No

Name * Test-App-for-Documentation

Homepage URL

Logo

Application ID [redacted]

Object ID [redacted]

Assignment required? Yes [✓ No]

Visible to users? Yes [✓ No]

Notes

After this 1st part, you should have the following application details available, which will be used to configure the remaining parts of these instructions:

Display name, which in our example is:

`Test-App-for-Documentation`

Tenant ID (also called Directory ID), which in our example is:

`8b80d7a0-9667-4736-a396-xxxxxxxxxx`

Application (or client) ID, which in our example is:

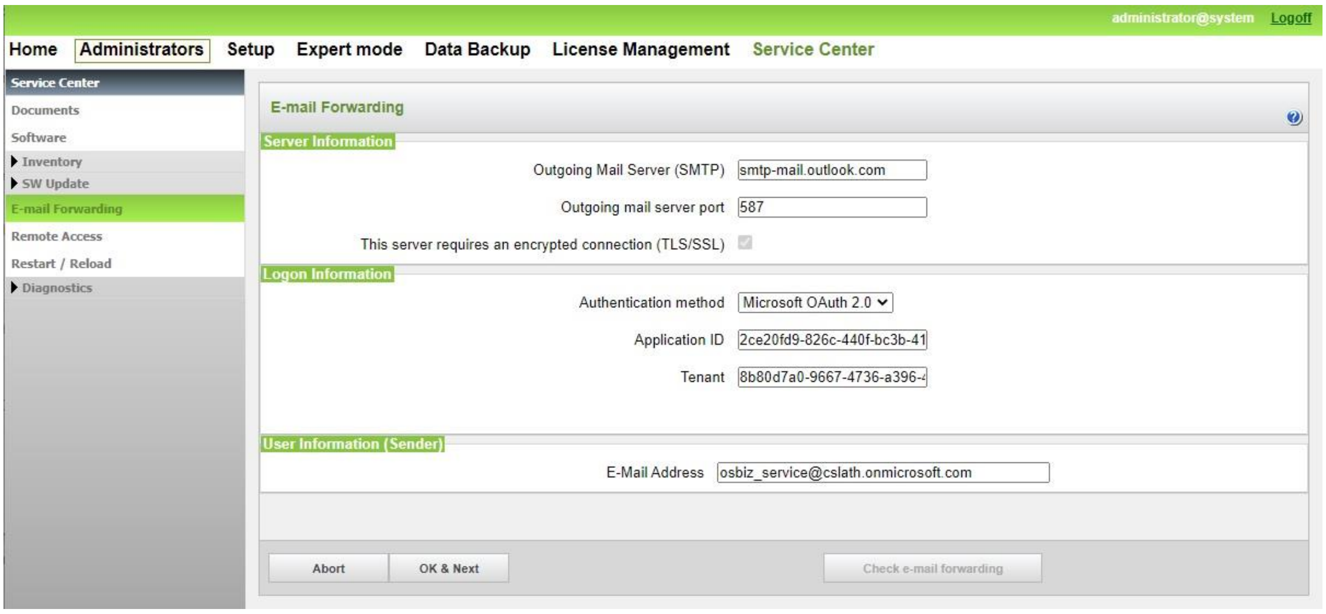
`2ce20fd9-826c-440f-bc3b-xxxxxxxxxx`

ℹ	Please notice that this are all example values and should be replaced with the actual details from the newly registered application above.

ℹ	Using a free Microsoft account, please use "`common`" in the **Tenant ID** field in OpenScape Business.

1.2 Create the E-mail Forwarding in WBM

The existing WBM page under **Service Center -> E-mail forwarding** is adapted to support the Microsoft OAuth 2.0 authentication. The "Microsoft OAuth 2.0" is the second drop down menu option. Instead of username and password the user must configure the **Application ID** and **Tenant** fields which will be used during the authentication process. Sign in with a valid email address for this application that is registered through the Azure configuration.



E-mail Forwarding

Server Information

Outgoing Mail Server (SMTP): smtp-mail.outlook.com
Outgoing mail server port: 587
This server requires an encrypted connection (TLS/SSL): ✓

Logon Information

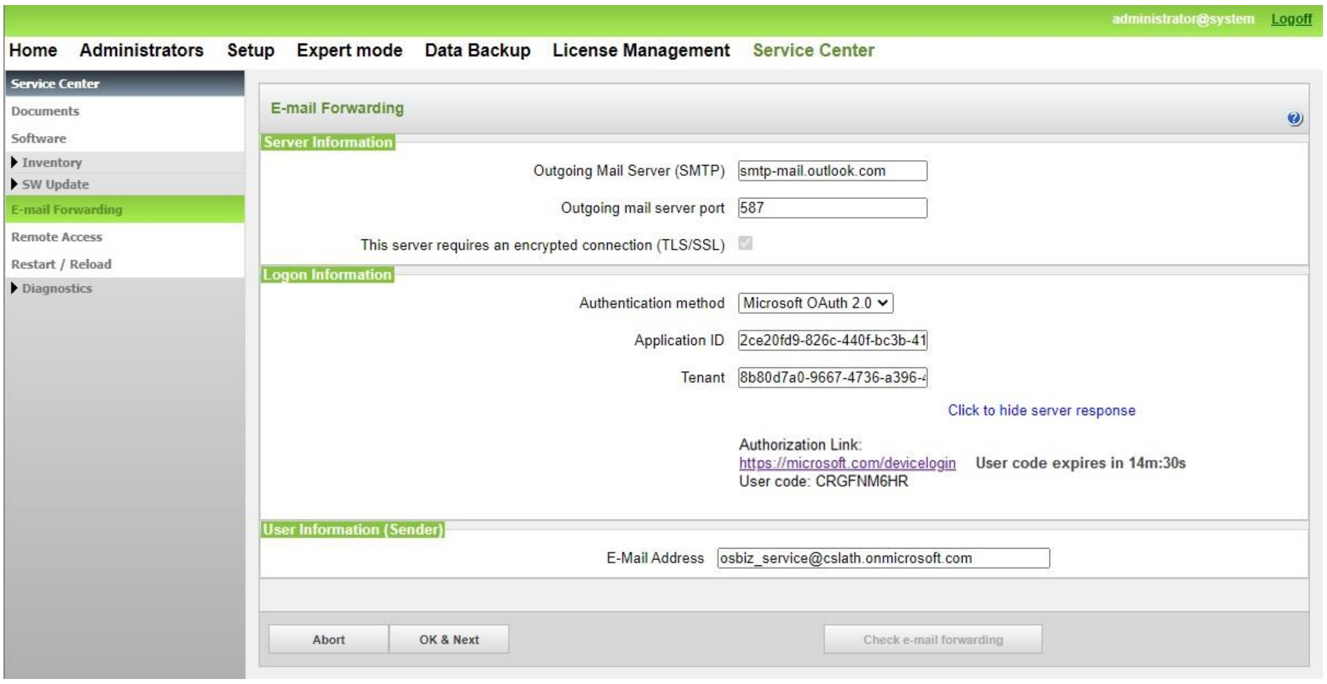
Authentication method: Microsoft OAuth 2.0
Application ID: 2ce20fd9-826c-440f-bc3b-41
Tenant: 8b80d7a0-9667-4736-a396-

User Information (Sender)

E-Mail Address: osbiz_service@cslath.onmicrosoft.com

Click [**OK & Next]** to proceed with the authentication process.

The OpenScape Business will send the configuration to the Azure server and waits until the Authorization Link is provided:



E-mail Forwarding

Server Information

Outgoing Mail Server (SMTP): smtp-mail.outlook.com
Outgoing mail server port: 587
This server requires an encrypted connection (TLS/SSL): ✓

Logon Information

Authentication method: Microsoft OAuth 2.0
Application ID: 2ce20fd9-826c-440f-bc3b-41
Tenant: 8b80d7a0-9667-4736-a396-

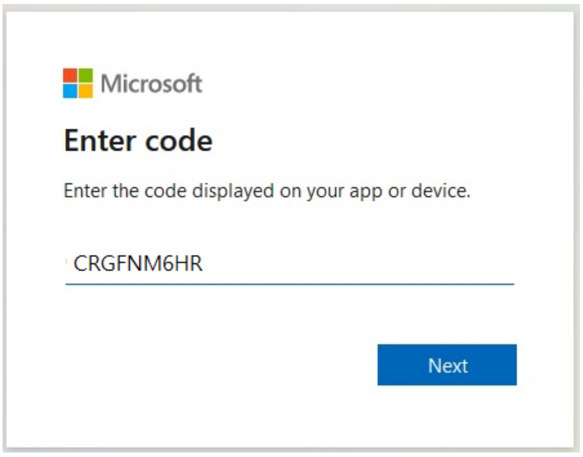
Click to hide server response

Authorization Link:
https://microsoft.com/devicelogin User code expires in 14m:30s
User code: CRGFNM6HR

User Information (Sender)

E-Mail Address: osbiz_service@cslath.onmicrosoft.com

Use the Authorization Link (**click on the link**) to complete the authentication within Azure by using the provided User code (**copy & paste**).

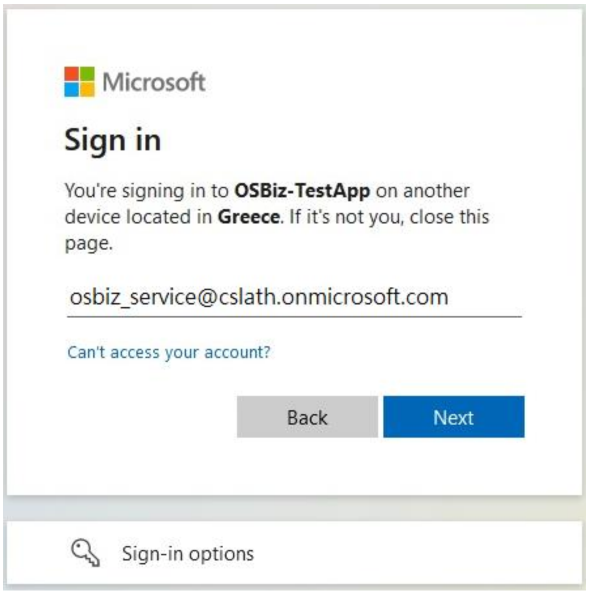


Enter code

Enter the code displayed on your app or device.

CRGFNM6HR

Sign in with the same email address for this application that is used within the E-mail Forwarding settings.

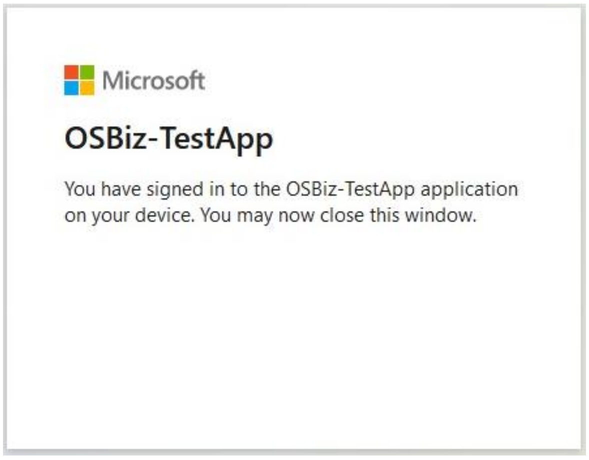

Microsoft

Sign in

You're signing in to **OSBiz-TestApp** on another device located in **Greece**. If it's not you, close this page.

osbiz_service@cslath.onmicrosoft.com

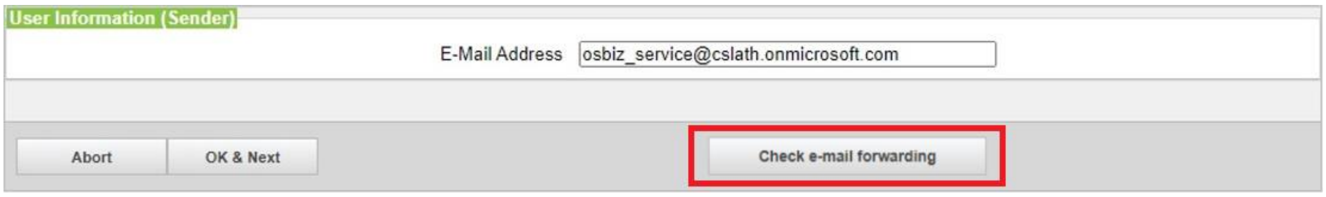
Successful sign in is confirmed:


Microsoft

OSBiz-TestApp

You have signed in to the OSBiz-TestApp application on your device. You may now close this window.

Verify the successful configuration and click [**Check e-mail forwarding**]:


User Information (Sender)

E-Mail Address: osbiz_service@cslath.onmicrosoft.com

1.3 FAQ

"SASL authentication failed" "Authentication unsuccessful, SmtpClientAuthentication is disabled for the Tenant"	• please check for following article https://answers.microsoft.com/en-us/msoffice/forum/all/unable-to-send-email-via-3rd-part-app-i-get-the/b29a7cd0-e696-4b0e-b66e-523be784a85a

2 Calendar Integration

The instructions below will cover the necessary steps to configure Calendar Integration using the OAuth 2.0 authentication method with Microsoft Office 365.

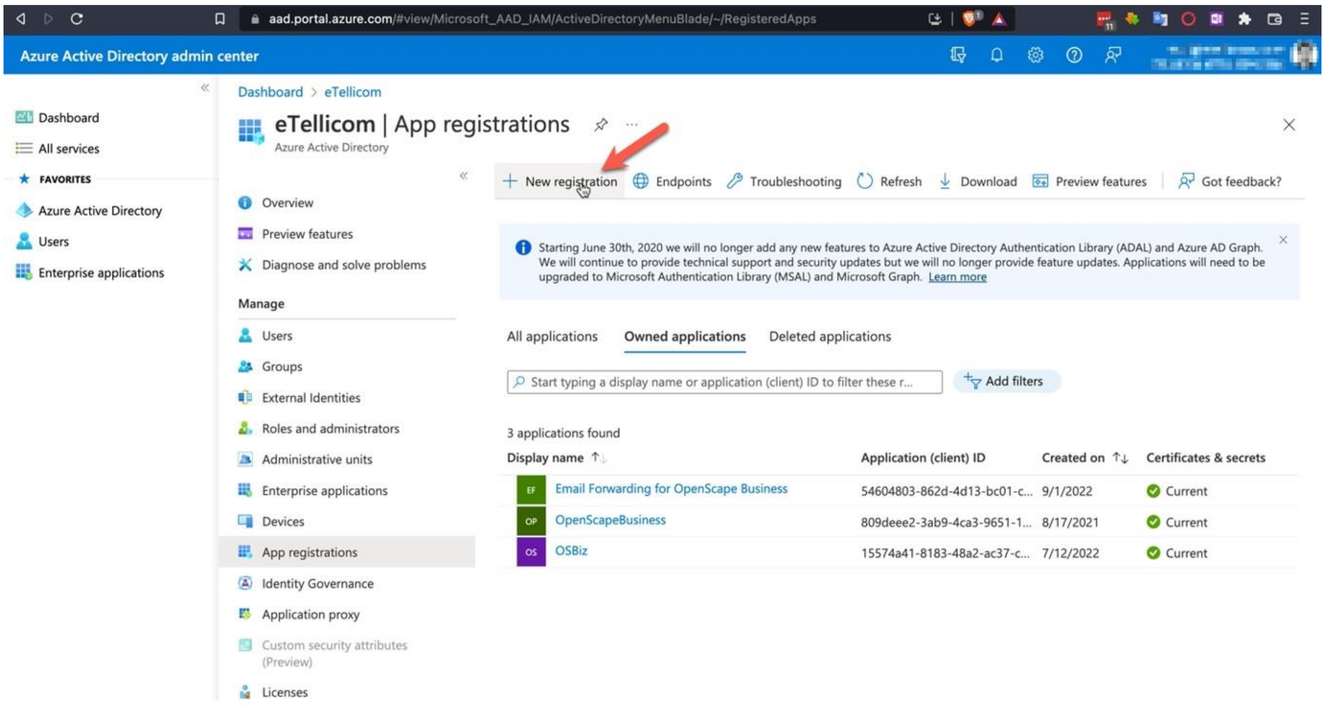
The configuration will be split in two parts. The 1st part covers the registration of the application in the Microsoft Azure Active Directory portal and the 2nd part will show how to create the configure Calendar Integration in UC Suite WBM using the information from the 1st part.

2.1 Register the Application in Microsoft Azure AD

Open the browser and go to the Azure Active Directory admin center and login with your Administrator account: https://aad.portal.azure.com/.

The instructions assume that an application with Administrator rights is being used; if that's not possible, you can register the application and request the Administrator to grant the required API permissions later, please check this article for more details: https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-permissions-and-consent#requesting-consent-for-an-entire-tenant.

Select "**Azure Active Directory**" in the menu on the top=left corner, click on "**App registrations**", and then click "**New registration**":

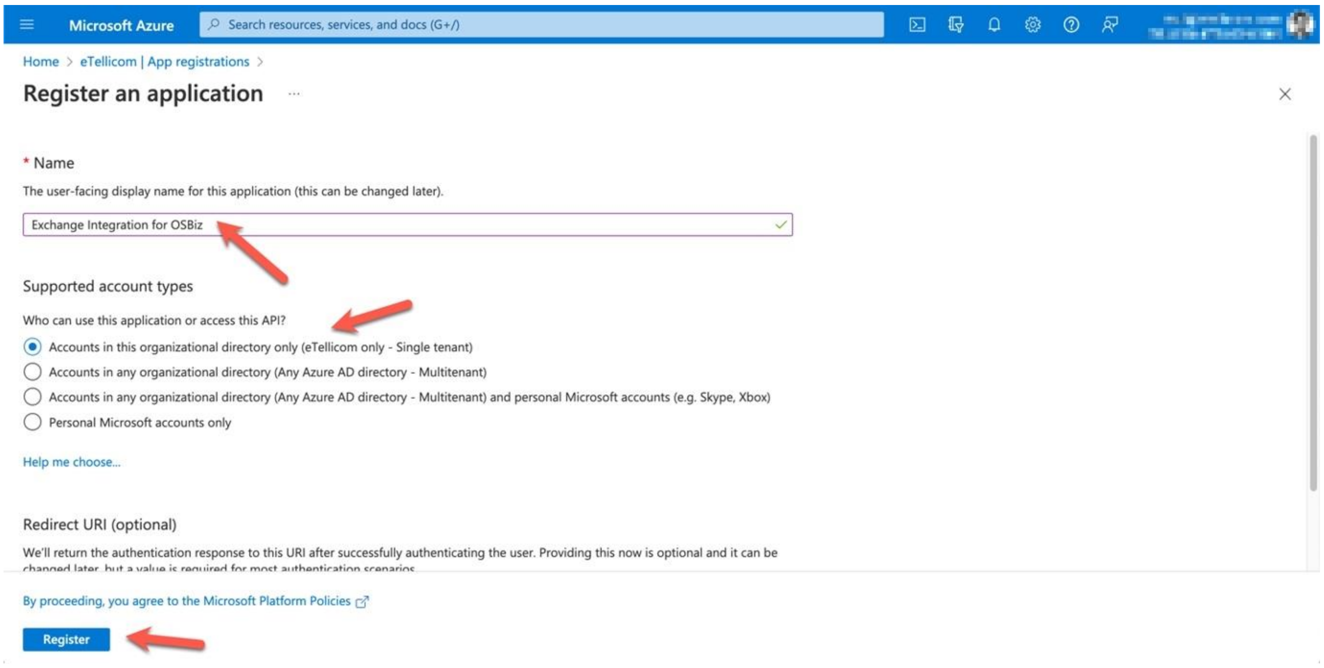


Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more

3 applications found

Display name	Application (client) ID	Created on	Certificates & secrets
EF Email Forwarding for OpenScape Business	54604803-862d-4d13-bc01-c...	9/1/2022	✓ Current
OP OpenScapeBusiness	809deee2-3ab9-4ca3-9651-1...	8/17/2021	✓ Current
OS OSBiz	15574a41-8183-48a2-ac37-c...	7/12/2022	✓ Current

Enter a meaningful name for the application (this will be later used to query the application when creating the Service Principal for Exchange Online), and for the "**Supported account type**", select the option which fits to customer installation (example: **Single tenant**). Then click the **Register** button to complete the registration:



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

Register an application

* Name

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

Exchange Integration for OSBiz

Supported account types

Who can use this application or access this API?

- [✓ Accounts in this organizational directory only (eTellicom 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)

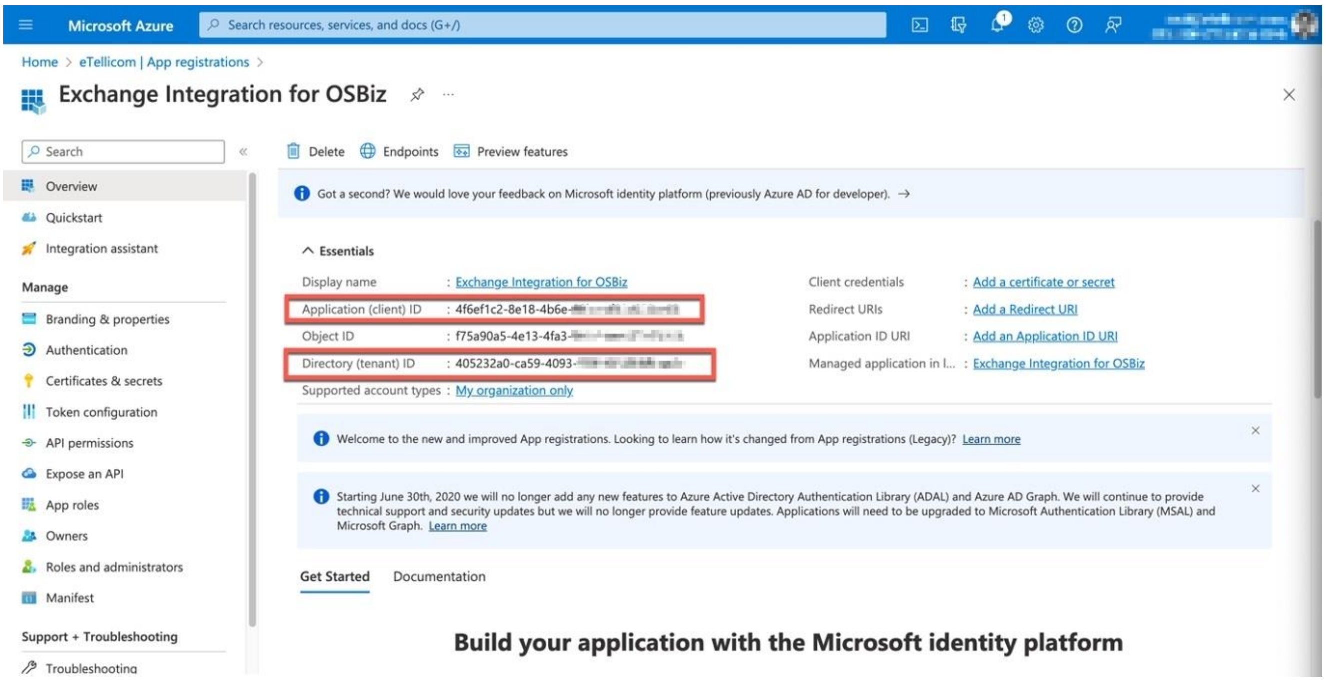
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

By proceeding, you agree to the Microsoft Platform Policies

Register

	The same application can be used for all services if the respective API permissions are assigned and granted. For documentation purpose we choose a separate application with the example name: `Exchange Integration for OSBiz`

At the end of the process, Azure AD will present some information about the newly registered application. At this point, it's a good idea to take note of the "**Application ID**" (also known as client ID) and "**Directory (Tenant) ID**", as that information will be used to double-check the second part of this configuration:



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

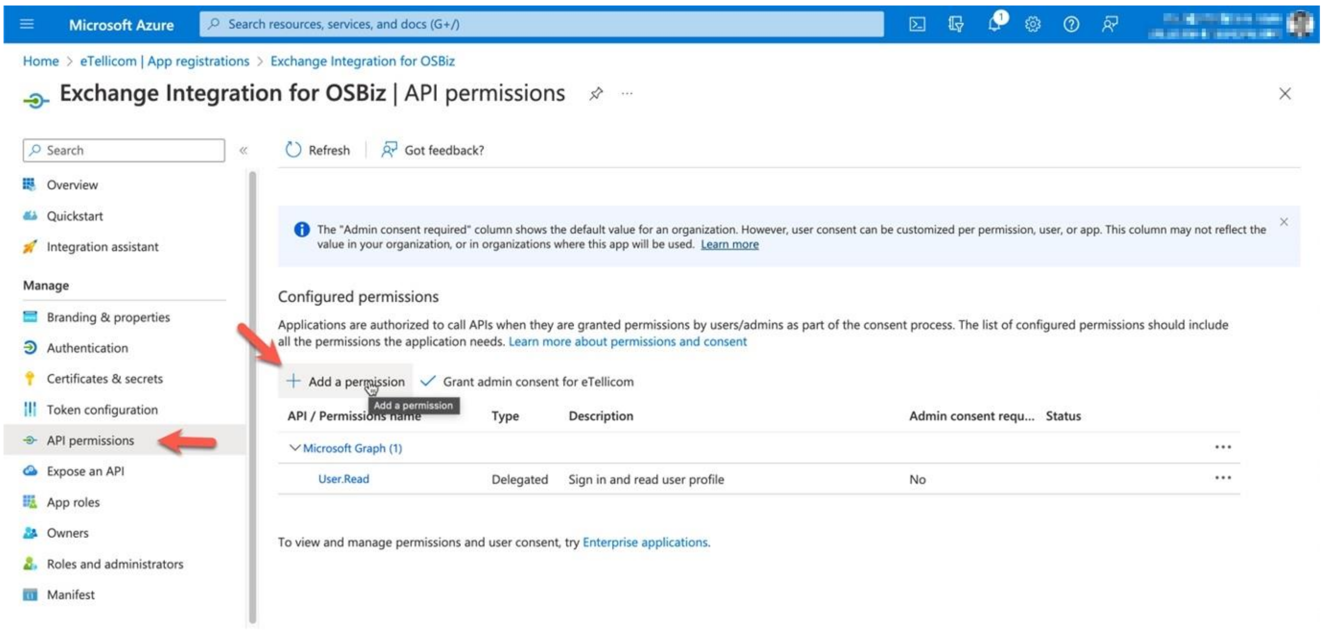
Display name	: Exchange Integration for OSBiz	Client credentials	: Add a certificate or secret
Application (client) ID	: 4f6ef1c2-8e18-4b6e-[redacted]	Redirect URIs	: Add a Redirect URI
Object ID	: f75a90a5-4e13-4fa3-[redacted]	Application ID URI	: Add an Application ID URI
Directory (tenant) ID	: 405232a0-ca59-4093-[redacted]	Managed application in l...	: Exchange Integration for OSBiz
Supported account types	: My organization only		

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? Learn more

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more

Build your application with the Microsoft identity platform

Now we need to assign the required permissions for the application. Click "**API permissions**" on the left, then click the "**Add a permission**" option to start adding the required API permission:



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

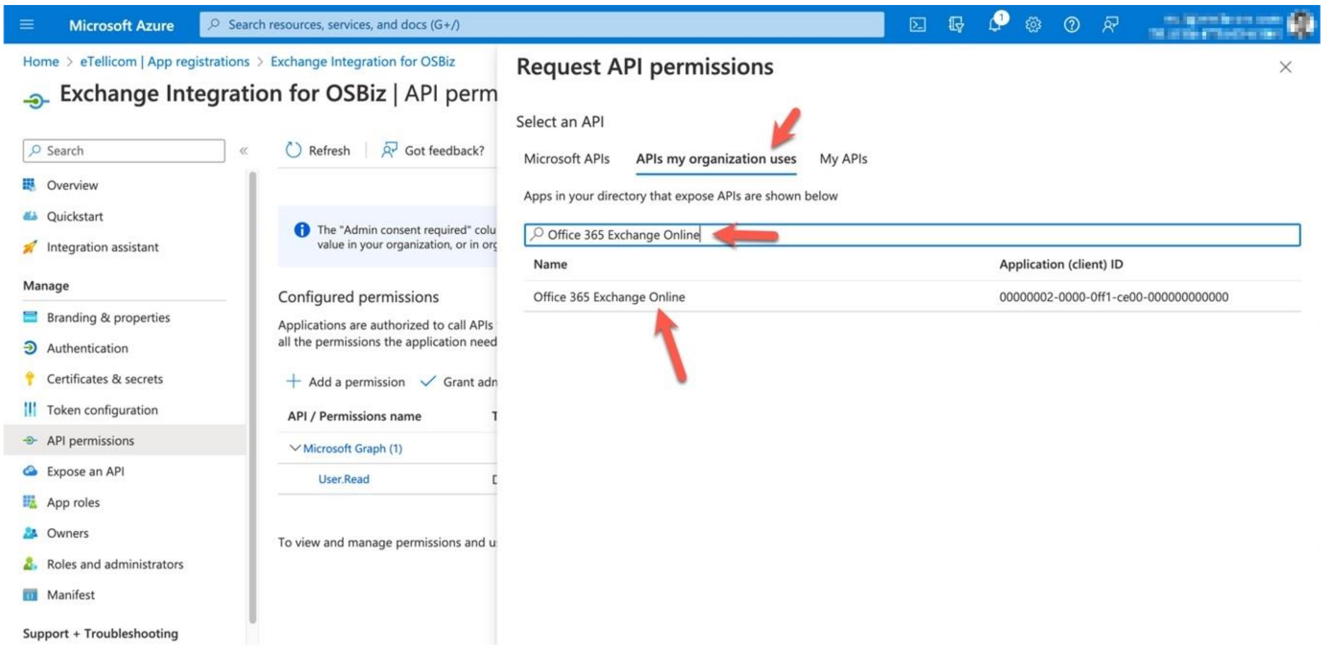
Configured permissions

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

API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	

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

Select the "**APIs my organization uses**" and type "**Office 365 Exchange Online**" (without the double quotes) entry, then click the row to select it:



The "Admin consent required" colu value in your organization, or in org

Applications are authorized to call APIs all the permissions the application need

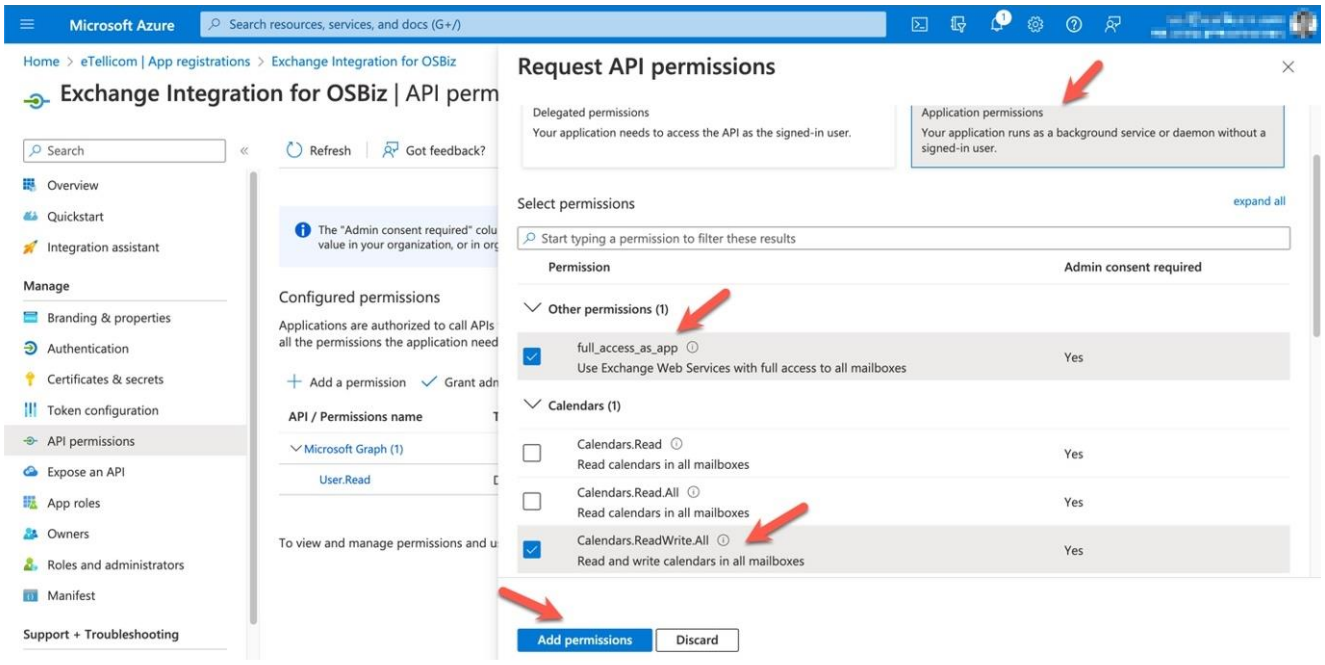
To view and manage permissions and u

Apps in your directory that expose APIs are shown below

Office 365 Exchange Online

Name	Application (client) ID
Office 365 Exchange Online	00000002-0000-0ff1-ce00-000000000000

Select the "**full_access_as_app**" and "**Calendars.ReadWriteAll**" permissions, then click the "**Add permissions**" button:



The "Admin consent required" colu value in your organization, or in org

Applications are authorized to call APIs all the permissions the application need

To view and manage permissions and u

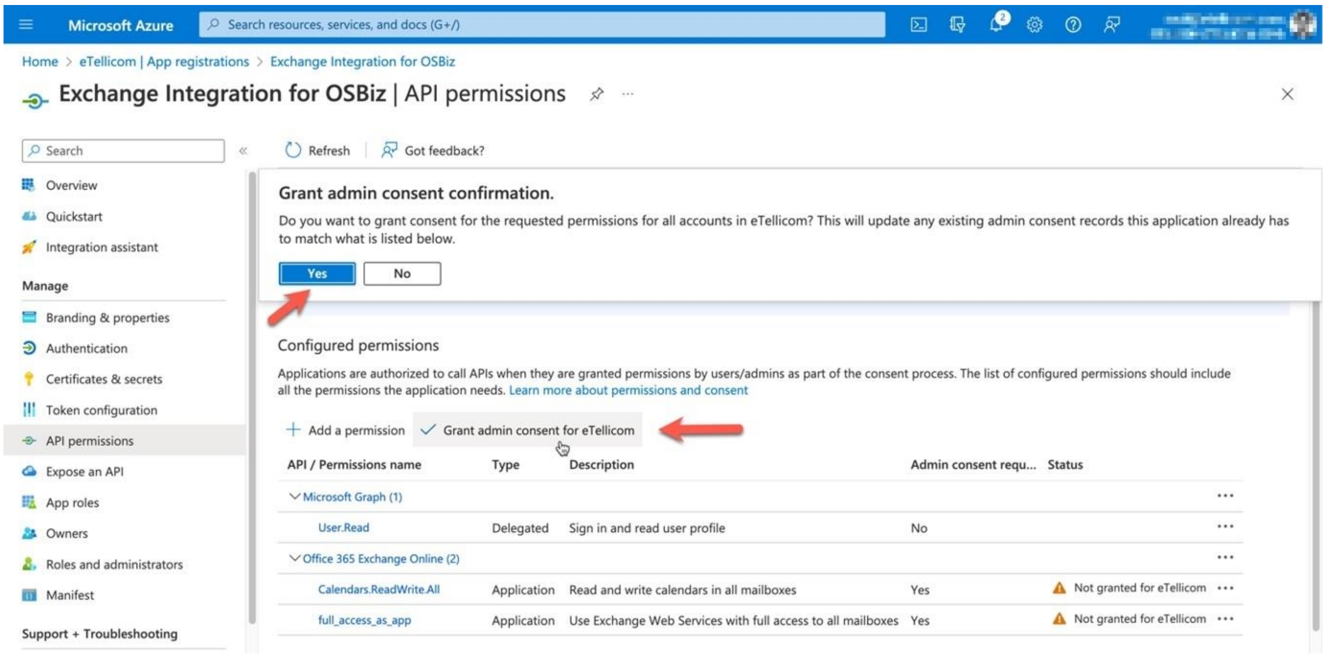
Delegated permissions
Your application needs to access the API as the signed-in user.

Application permissions
Your application runs as a background service or daemon without a signed-in user.

Start typing a permission to filter these results

	Permission	Admin consent required
	Other permissions (1)	
✓	full_access_as_app Use Exchange Web Services with full access to all mailboxes	Yes
	Calendars (1)	
☐	Calendars.Read Read calendars in all mailboxes	Yes
☐	Calendars.Read.All Read calendars in all mailboxes	Yes
✓	Calendars.ReadWrite.All Read and write calendars in all mailboxes	Yes

We now need to grant the admin consent permission for our tenant to use the API we selected above, that's done by clicking the "**Grant admin consent for <Tenant>**", then clicking **Yes** in the dialog box that will appear:



Microsoft Azure | Search resources, services, and docs (G+/)

Home > eTellicom | App registrations > Exchange Integration for OSBiz

Exchange Integration for OSBiz | API permissions

Search | 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

Grant admin consent confirmation.

Do you want to grant consent for the requested permissions for all accounts in eTellicom? This will update any existing admin consent records this application already has to match what is listed below.

Yes | No

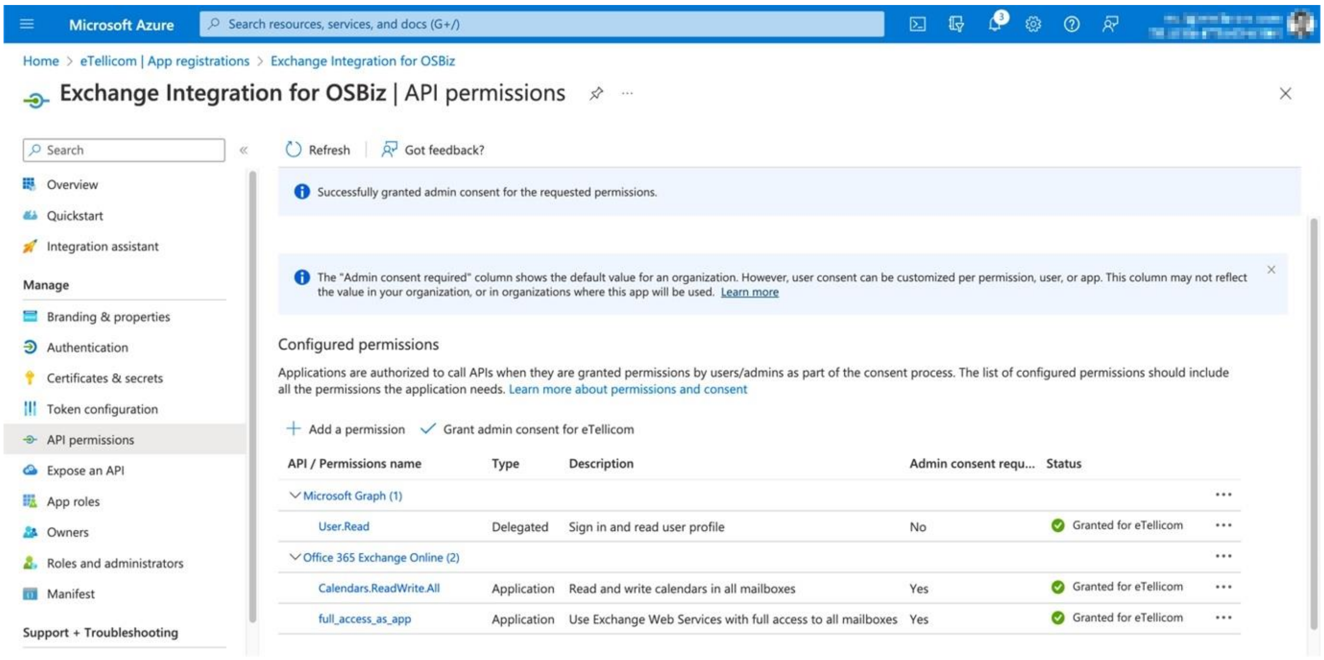
Configured permissions

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 eTellicom

API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	
Office 365 Exchange Online (2)				
Calendars.ReadWrite.All	Application	Read and write calendars in all mailboxes	Yes	⚠ Not granted for eTellicom
full_access_as_app	Application	Use Exchange Web Services with full access to all mailboxes	Yes	⚠ Not granted for eTellicom

Once it's completed, the Status column should display that the API is correctly granted for the tenant:



Microsoft Azure | Search resources, services, and docs (G+/)

Home > eTellicom | App registrations > Exchange Integration for OSBiz

Exchange Integration for OSBiz | API permissions

Search | 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

Successfully granted admin consent for the requested permissions.

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

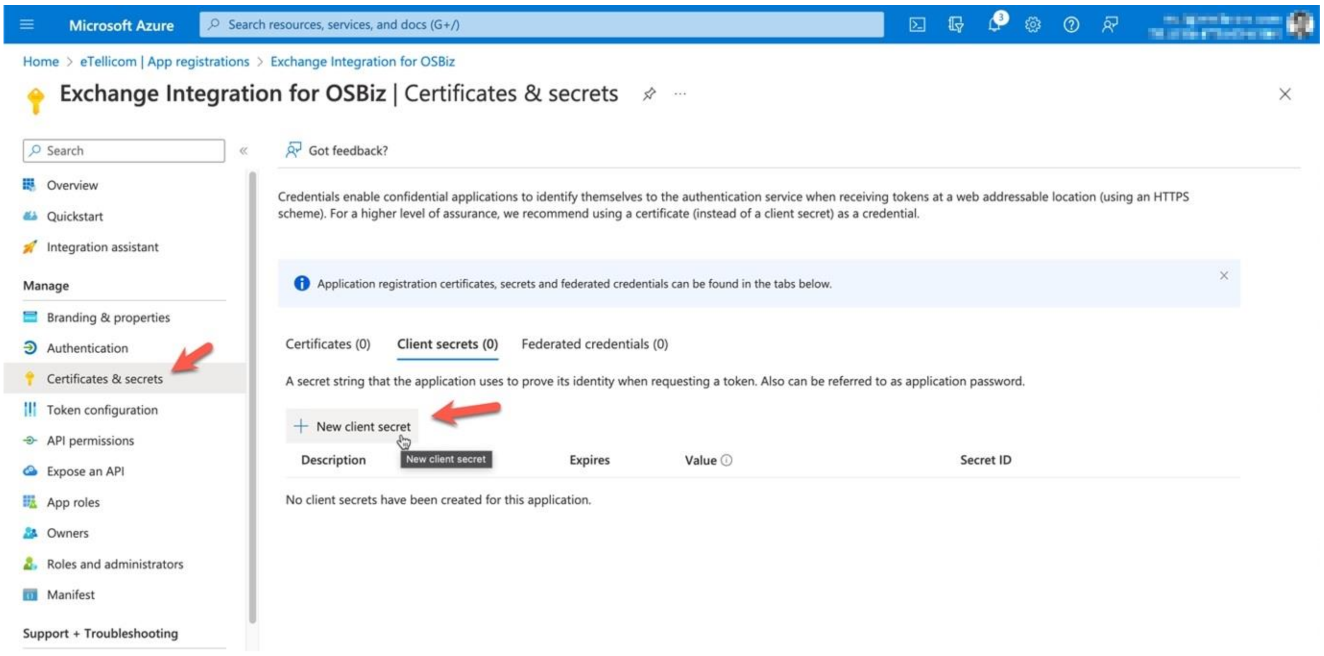
Configured permissions

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 eTellicom

API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	✓ Granted for eTellicom
Office 365 Exchange Online (2)				
Calendars.ReadWrite.All	Application	Read and write calendars in all mailboxes	Yes	✓ Granted for eTellicom
full_access_as_app	Application	Use Exchange Web Services with full access to all mailboxes	Yes	✓ Granted for eTellicom

Now we need to create a client secret for the application. Select the "**Client & secrets**" menu and then click "**New client secret**":



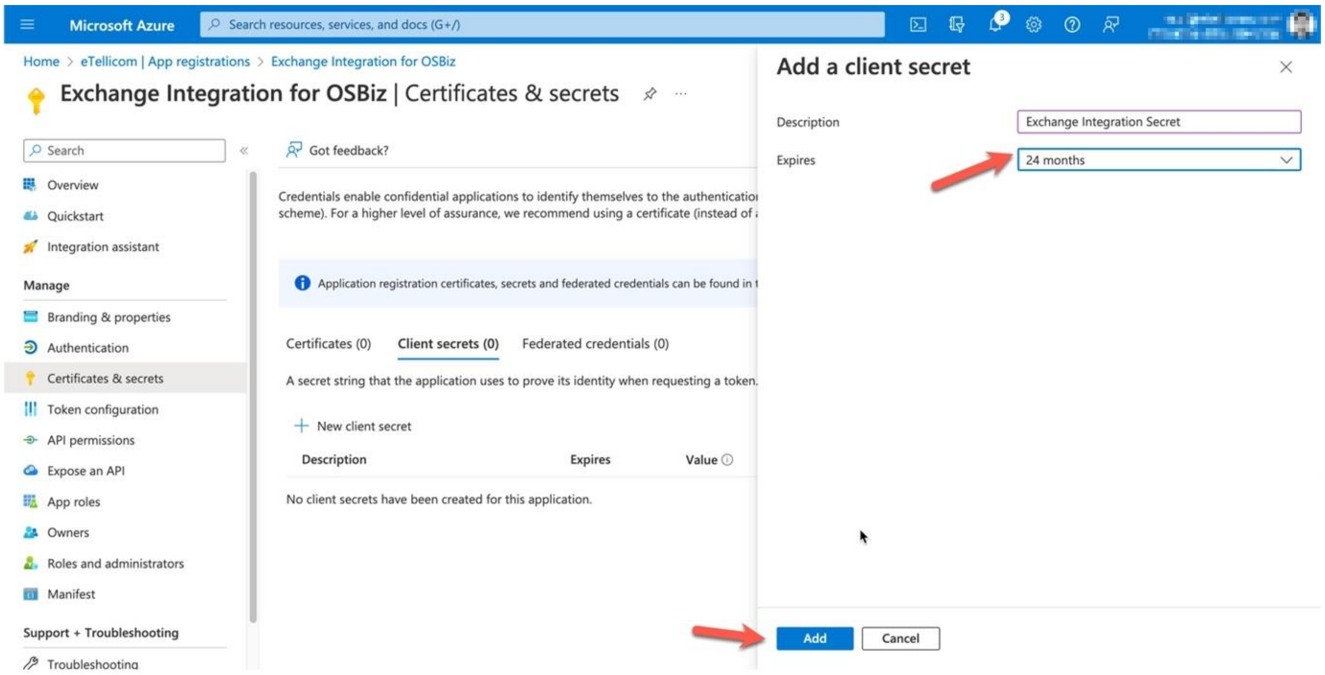
Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

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.

No client secrets have been created for this application.

Enter a description for the client secret (it's just for informational purposes, as multiple secrets can be created for the same application) and select the expiration date option which fits to customer installation (example: maximum, 24 months), then click "**Add**":

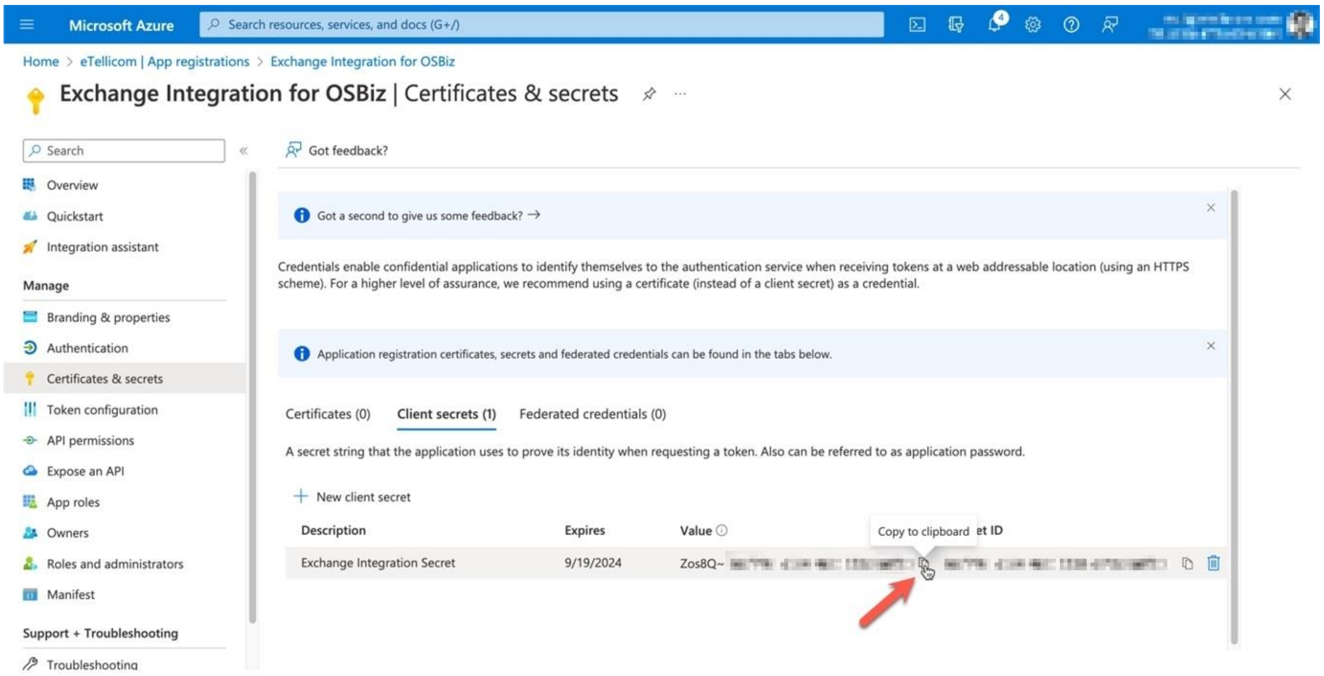


Description: Exchange Integration Secret

Expires: 24 months

	The maximum runtime of the client secret is 24 months. To avoid reconfiguration, the secret value must be renewed before expiration.

IMPORTANT: at this point, you HAVE TO COPY the Value for the client secret by clicking the button next to it, as this the only time the Value will be ever visible and available for copying. If you don’t do that at this point, that Client Secret Value cannot be used later when it’s entered on WBM:


Microsoft Azure | Search resources, services, and docs (G+/) | [redacted]

Home > eTellicom | App registrations > Exchange Integration for OSBiz

Exchange Integration for OSBiz | Certificates & secrets

Got a second to give us some feedback? →

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.

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	Copy to clipboard et ID
Exchange Integration Secret	9/19/2024	Zos8Q~ [redacted]	[redacted]

After this 1st part, you should have the following application details available, which will be used to configure the remaining parts of these instructions:

Display name, which in our example is:

`Exchange Integration for OSBiz`

Tenant ID (also called Directory ID), which in our example is:

`405232a0-ca59-4093-959f-xxxxxxxxxx`

Application (or client) ID, which in our example is:

`4f6ef1c2-8e18-4b6e-89bc-xxxxxxxxxx`

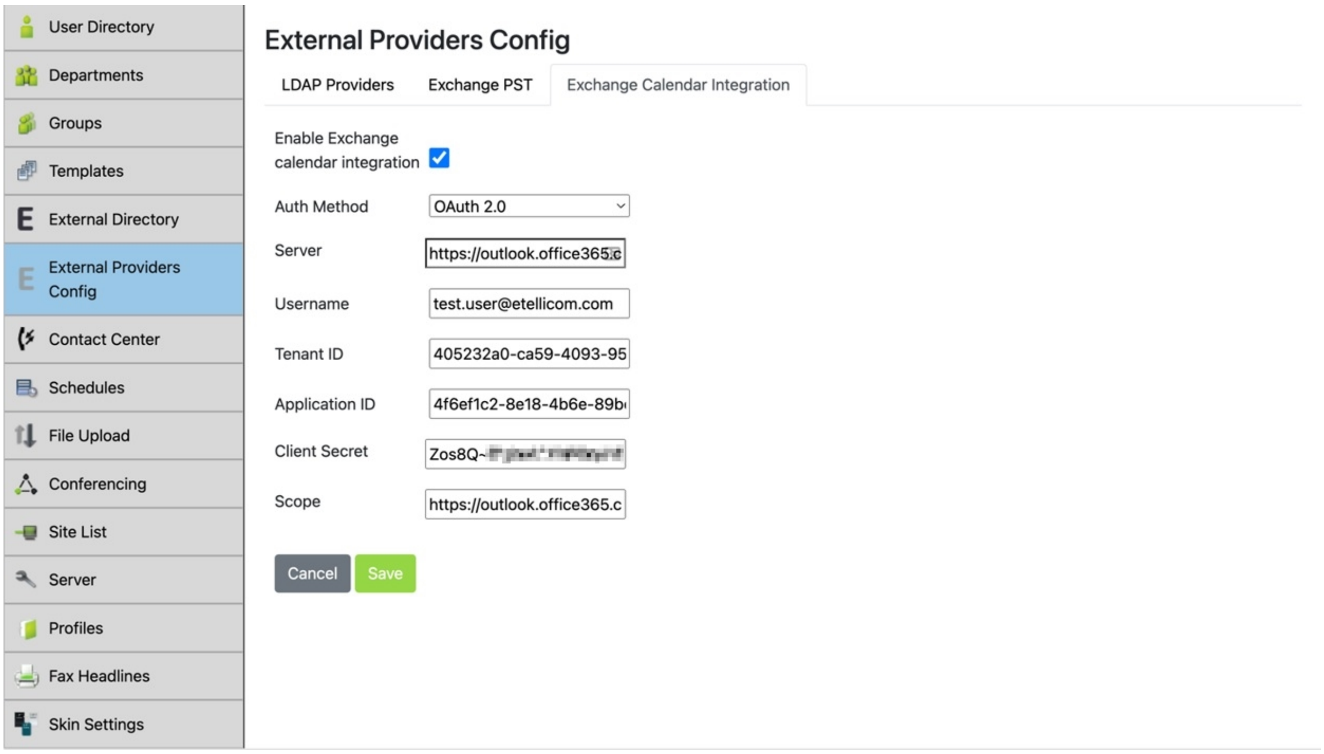
Client Secret Value, which in our example is:

`Zos8Q~…`

	Please notice that this are all example values and should be replaced with the actual details from the newly registered application above.

2.2 Exchange Calendar Integration in UC Suite WBM

Now that the application is registered with Azure, we can go to UC Suite WBM and open "**External Providers Config**", then select the "**Exchange Calendar Integration**" tab. Change the Auth Method to "**Microsoft OAuth 2.0**" and the following fields will appear. Use a valid email address for this application that is registered through the Azure configuration within the Username field:



External Providers Config

Enable Exchange calendar integration	✓
Auth Method	OAuth 2.0
Server	https://outlook.office365.c
Username	test.user@etellicom.com
Tenant ID	405232a0-ca59-4093-95
Application ID	4f6ef1c2-8e18-4b6e-89b
Client Secret	Zos8Q~
Scope	https://outlook.office365.c

Please enter the authentication details using the information that we collected from the first part of these instructions.

<u>Server</u>: `https://outlook.office365.com/EWS/Exchange.asmx`

<u>Username</u>: enter the email for your Microsoft Office 365 account

<u>Tenant ID, Application ID, Client Secret</u>: these are the same values that you took note from the last step in the 1st part of the configuration

<u>Scope</u>: `https://outlook.office365.com/.default`

Click [**Save**] and the configuration should now be completed.

Exchange Calendar integration (notes and hints):

- Understanding the function: When a user set appointments in his Microsoft Outlook calendar, the OpenScape Office Business system will automatically check for keywords in the calendar appointment subject like: "Meeting", "Sick", "Break", "Out of Office", "Holiday", "Lunch" or "Home". If such a keyword is found, the OpenScape Office Business will set the user's presence status automatically when the appointment time is reached, even if the Microsoft Outlook session of the specific user is not active anymore. The user simply configures his appointments with these keywords and OpenScape Office Business will automatically set his telephone presence status and as a result all calls will be routed to his voicemail or to his cell phone, depending on the configured forwarding destinations. The user can also configure in miscellaneous settings, whether his presence status should be switched back to "Office" presence status when the appointment time has ended. Appointments can also be set from Outlook Web Access (OWA) instead.

- A valid e-mail address for every user must be configured in all applications, as described in aforementioned steps respectively. o The UC Suite users Voicemail language settings is used by OpenScape Business to understand the keywords like: "Meeting", "Sick", "Break", "Out of Office", "Holiday", "Lunch" or "Home", so for example, German set Voicemail language can be used with the German keywords in the appointments subject like "Besprechung", "Krank", "Pause", "Außer Haus", "Urlaub", "Mittag" or "Zu Hause" only.

- In case "Auto back to office" option is not used, the user's presence status will be kept to the configured appointment status even if the appointment time elapses. As a result, the return time shown to other users will be increased every 15 minutes. If the "Auto back to office" option is used and the appointment time ends, the user's presence status is automatically set to "Office" again.

- Important note for testing: Appointments set into a future date and time with less than 3 minutes are ignored, because the system expects that the user creating an appointment can alter any option within 3 minutes. The OpenScape Business checks for appointments in the user's calendar every 30 seconds. For this reason and for any possible delays impaired by the network between the OpenScape Business and the Exchange Server, please configure test appointments starting at minimum after 3 and a half minutes or more.

- If you want to test the Exchange Calendar Integration functionality when user is not logged in either in myPortal for Outlook or in myPortal for Desktop, you can simply use the Outlook webmail. Usually, you can access outlook webmail by using this URL: "https://OWA". Login with the user's credentials and add the appointment with the appropriate keyword to the calendar. Check in another UC Suite user's client that the presence status changes when the appointment time has reached

3 Exchange Directory Integration

The instructions below will cover the necessary steps to configure Directory Integration using the OAuth 2.0 authentication method with Microsoft Office 365.

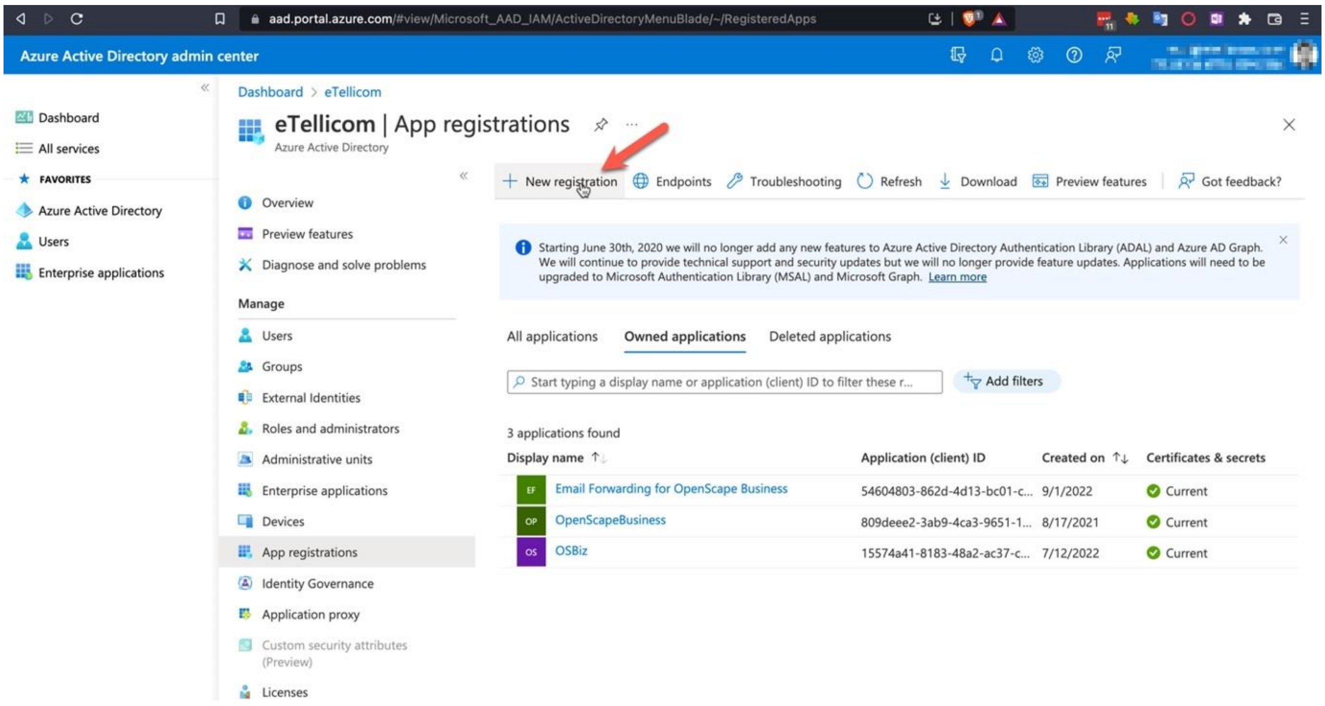
The configuration will be split in two parts. The 1st part covers the registration of the application in the Microsoft Azure Active Directory portal and the 2nd part will show how to configure Exchange Directory Integration in UC Suite WBM using the information from the first part.

3.1 Register the Application in Microsoft Azure AD

Open the browser and go to the Azure Active Directory admin center and login with your Administrator account: https://aad.portal.azure.com/.

The instructions assume that an application with Administrator rights is being used; if that's not possible, you can register the application and request the Administrator to grant the required API permissions later, please check this article for more details: https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-permissions-and-consent#requesting-consent-for-an-entire-tenant.

Select "**Azure Active Directory**" in the menu on the top=left corner, click on "**App registrations**", and then click "**New registration**":

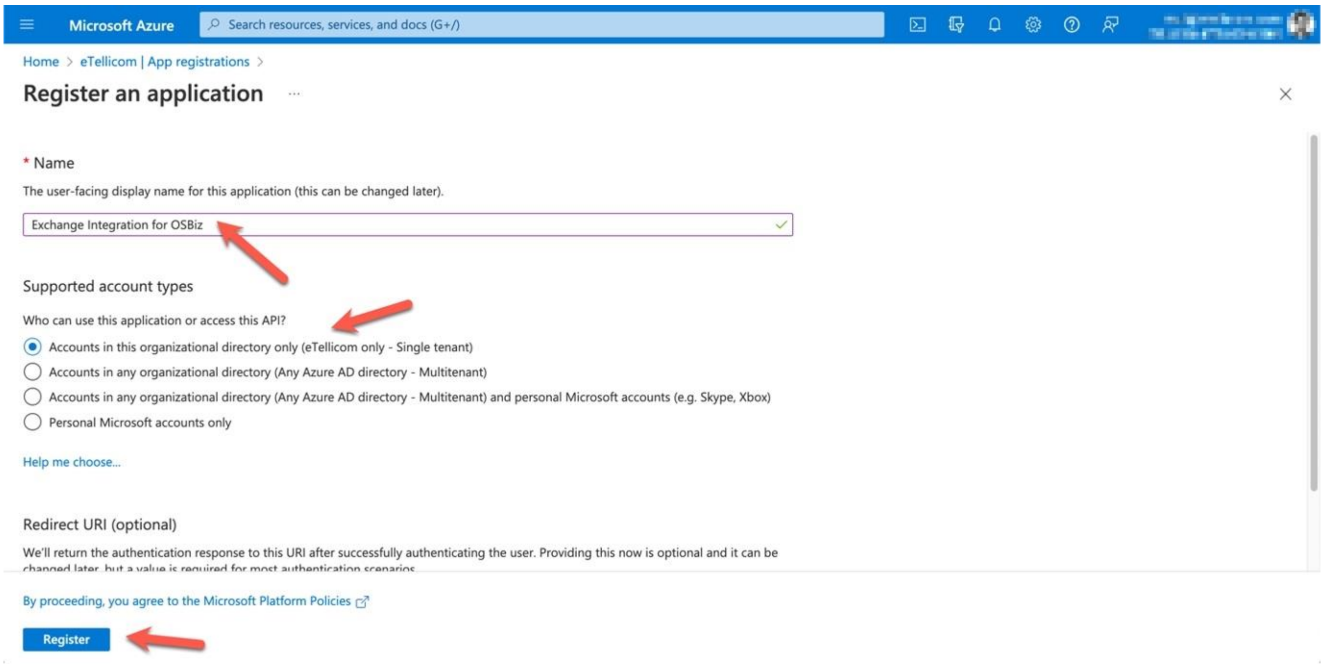


Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more

3 applications found

Display name	Application (client) ID	Created on	Certificates & secrets
EF Email Forwarding for OpenScape Business	54604803-862d-4d13-bc01-c...	9/1/2022	✓ Current
OP OpenScapeBusiness	809deee2-3ab9-4ca3-9651-1...	8/17/2021	✓ Current
OS OSBiz	15574a41-8183-48a2-ac37-c...	7/12/2022	✓ Current

Enter a meaningful name for the application (this will be later used to query the application when creating the Service Principal for Exchange Online), and for the "**Supported account type**", select the option which fits to customer installation (example: **Single tenant**). Then click the **Register** button to complete the registration:


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

Register an application

* Name

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

Exchange Integration for OSBiz

Supported account types

Who can use this application or access this API?

- [✓] Accounts in this organizational directory only (eTellicom 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...

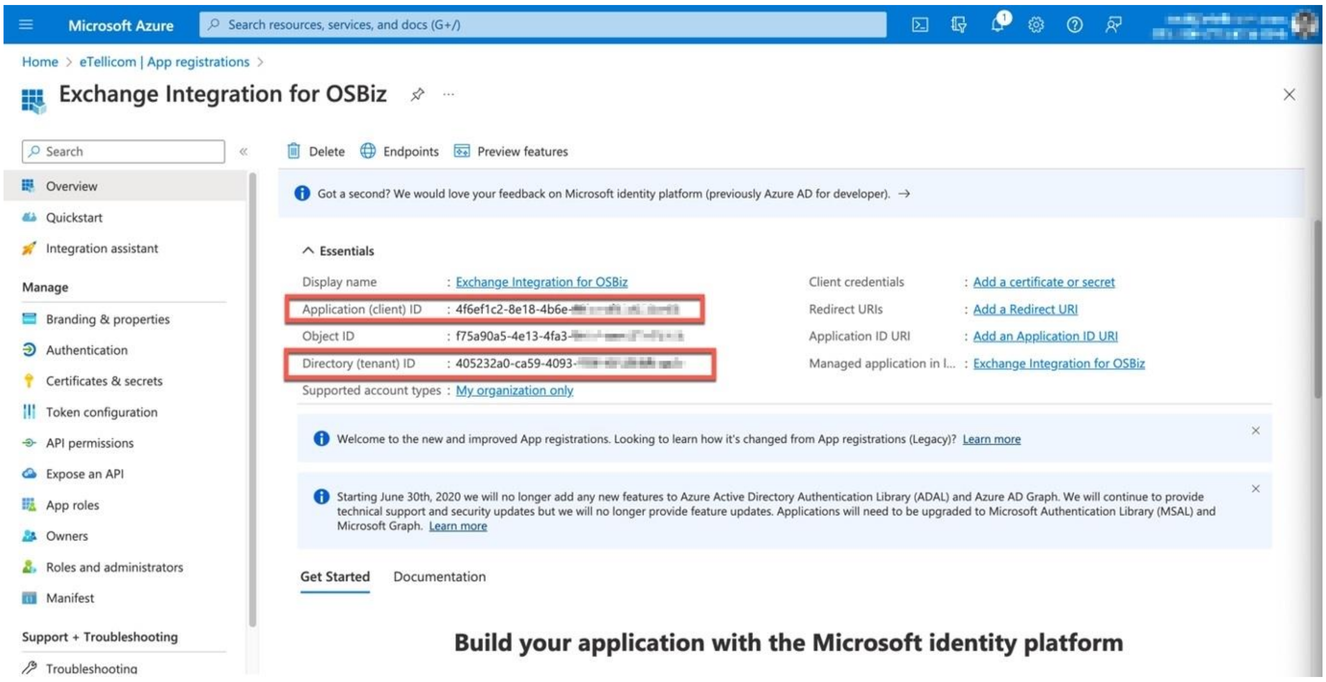
Redirect URI (optional)

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.

By proceeding, you agree to the Microsoft Platform Policies

	The same application can be used for all services if the respective API permissions are assigned and granted. For documentation purpose we choose a separate application with the example name: `Exchange Integration for OSBiz`

At the end of the process, Azure AD will present some information about the newly registered application. At this point, it's a good idea to take note of the "**Application ID**" (also known as client ID) and "**Directory (Tenant) ID**", as that information will be used to double-check the second and third parts of this configuration:


Microsoft Azure | Search resources, services, and docs (G+/) | [redacted]

Home > eTellicom | App registrations >

Exchange Integration for OSBiz

Search | 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

Delete | Endpoints | Preview features

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

Essentials

Display name	: Exchange Integration for OSBiz	Client credentials	: Add a certificate or secret
Application (client) ID	: 4f6ef1c2-8e18-4b6e-[redacted]	Redirect URIs	: Add a Redirect URI
Object ID	: f75a90a5-4e13-4fa3-[redacted]	Application ID URI	: Add an Application ID URI
Directory (tenant) ID	: 405232a0-ca59-4093-[redacted]	Managed application in l...	: Exchange Integration for OSBiz
Supported account types	: My organization only		

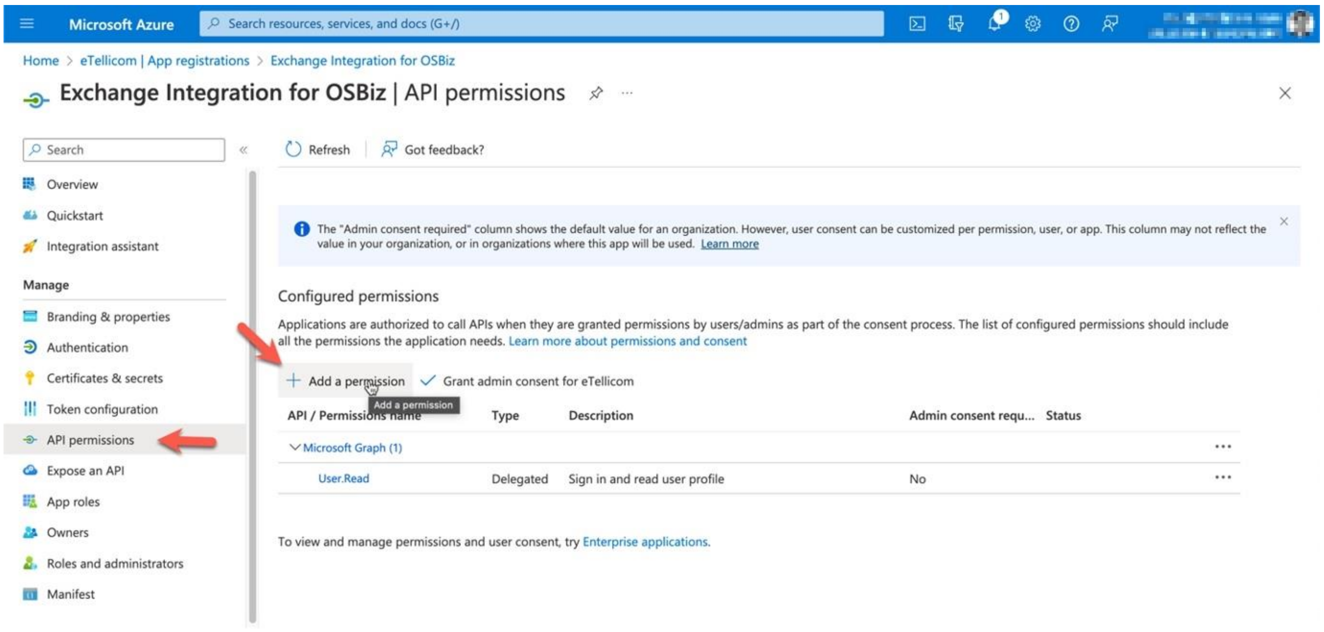
Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? Learn more

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more

Get Started | Documentation

Build your application with the Microsoft identity platform

Now we need to assign the required permissions for the application. Click "**API permissions**" on the left, then click the "**Add a permission**" option to start adding the required API permission:


Microsoft Azure | Search resources, services, and docs (G+/) | [redacted]

Home > eTellicom | App registrations > Exchange Integration for OSBiz

Exchange Integration for OSBiz | API permissions

Search | Overview | Quickstart | Integration assistant | Manage | Branding & properties | Authentication | Certificates & secrets | Token configuration | API permissions | Expose an API | App roles | Owners | Roles and administrators | Manifest

Refresh | Got feedback?

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

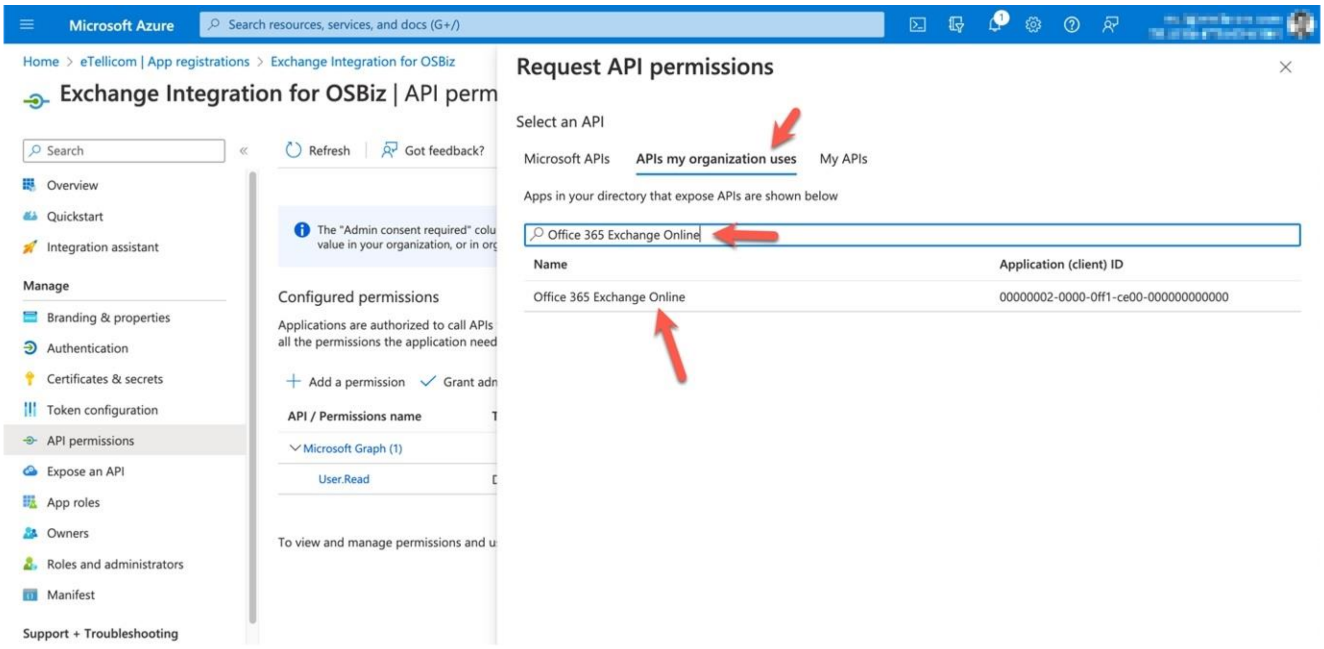
Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. Learn more about permissions and consent

+ Add a permission | ✓ Grant admin consent for eTellicom

API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	

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

Select the "**APIs my organization uses**" and type "**Office 365 Exchange Online**" (without the double quotes) entry, then click the row to select it:



The "Admin consent required" colu
value in your organization, or in org

Configured permissions

Applications are authorized to call APIs
all the permissions the application need

API / Permissions name	T
Microsoft Graph (1)	
User.Read	D

To view and manage permissions and u

Request API permissions

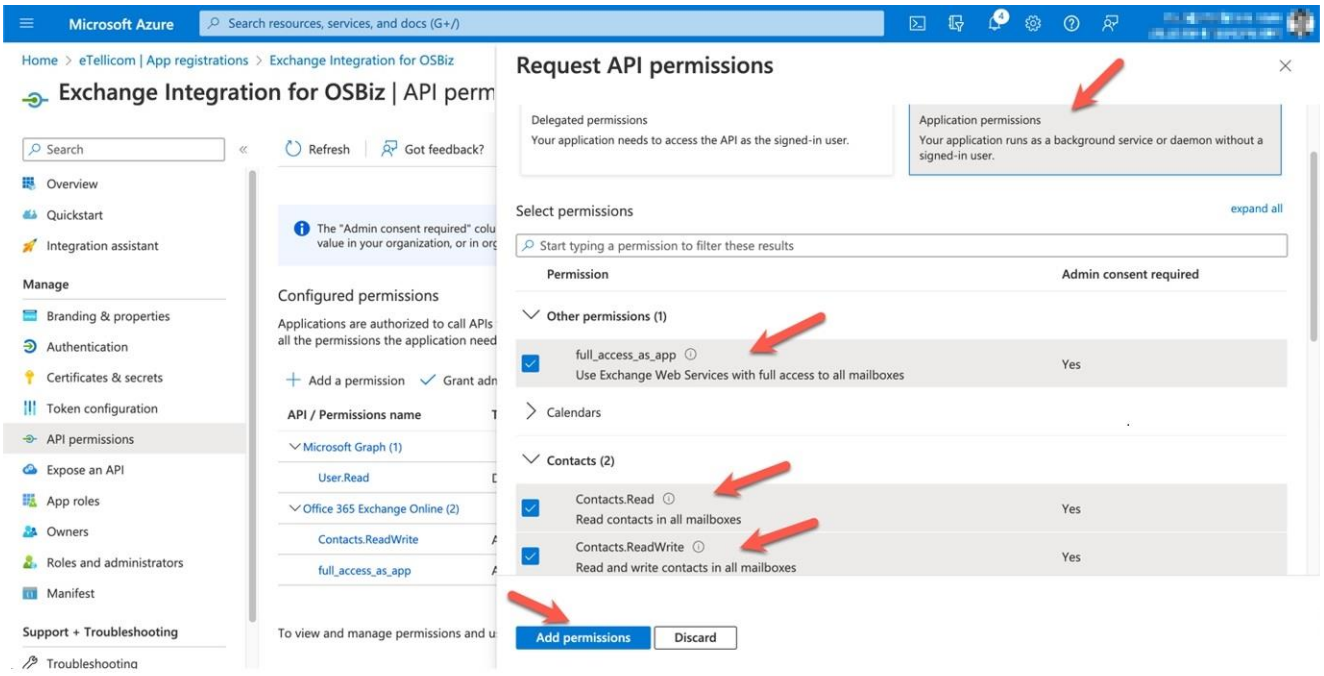
Select an API

Apps in your directory that expose APIs are shown below

Office 365 Exchange Online

Name	Application (client) ID
Office 365 Exchange Online	00000002-0000-0ff1-ce00-000000000000

Select the "**full_access_as_app**", "**Contacts.Read**" and "**Contact.ReadWrite**" permissions, then click the "**Add permissions**" button:



The "Admin consent required" colu
value in your organization, or in org

Configured permissions

Applications are authorized to call APIs
all the permissions the application need

API / Permissions name	T
Microsoft Graph (1)	
User.Read	D
Office 365 Exchange Online (2)	
Contacts.ReadWrite	A
full_access_as_app	A

To view and manage permissions and u

Request API permissions

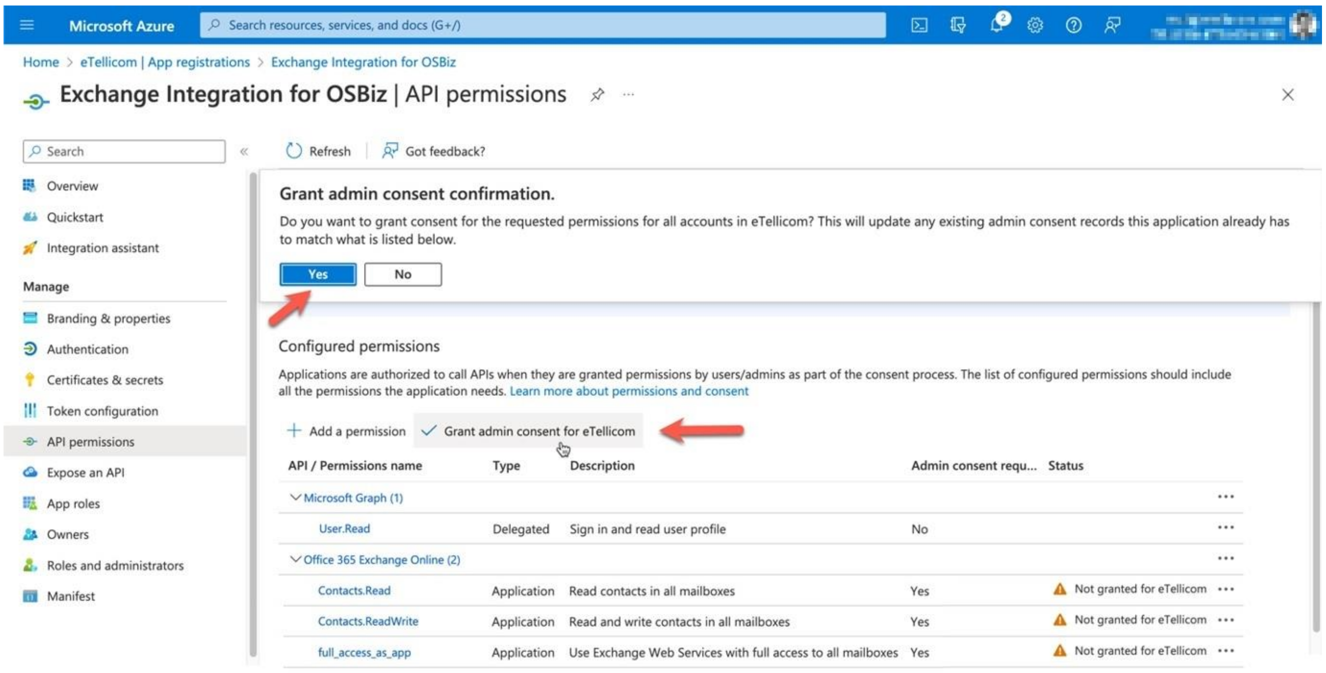
Delegated permissions
Your application needs to access the API as the signed-in user.

Application permissions
Your application runs as a background service or daemon without a signed-in user.

Select permissions

	Permission	Admin consent required
	Other permissions (1)	
✓	full_access_as_app Use Exchange Web Services with full access to all mailboxes	Yes
	Calendars	
	Contacts (2)	
✓	Contacts.Read Read contacts in all mailboxes	Yes
✓	Contacts.ReadWrite Read and write contacts in all mailboxes	Yes

We now need to grant the admin consent permission for our tenant to use the API we selected above, that's done by clicking the "**Grant admin consent for <Tenant>**", then clicking **Yes** in the dialog box that will appear:


Home > eTellicom | App registrations > Exchange Integration for OSBiz

Exchange Integration for OSBiz | API permissions

Grant admin consent confirmation.

Do you want to grant consent for the requested permissions for all accounts in eTellicom? This will update any existing admin consent records this application already has to match what is listed below.

Yes No

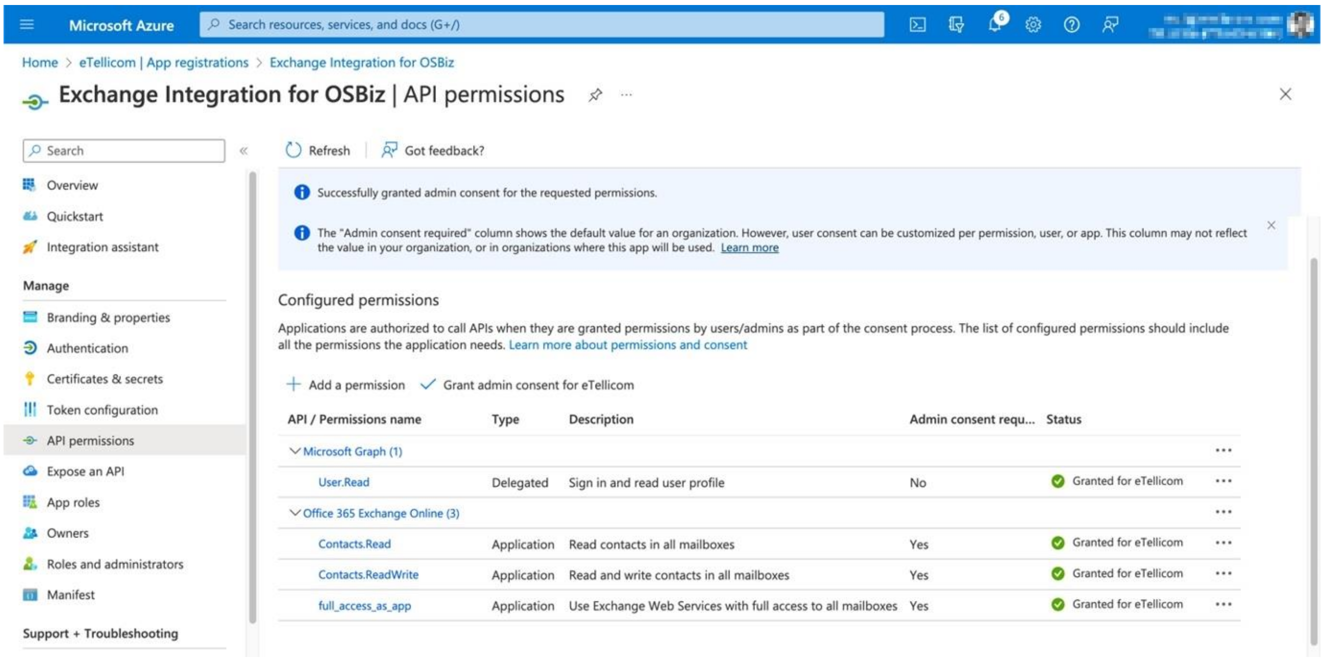
Configured permissions

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 eTellicom

API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	
Office 365 Exchange Online (2)				
Contacts.Read	Application	Read contacts in all mailboxes	Yes	Not granted for eTellicom
Contacts.ReadWrite	Application	Read and write contacts in all mailboxes	Yes	Not granted for eTellicom
full_access_as_app	Application	Use Exchange Web Services with full access to all mailboxes	Yes	Not granted for eTellicom

Once it's completed, the Status column should display that the API is correctly granted for the tenant:


Home > eTellicom | App registrations > Exchange Integration for OSBiz

Exchange Integration for OSBiz | API permissions

Successfully granted admin consent for the requested permissions.

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

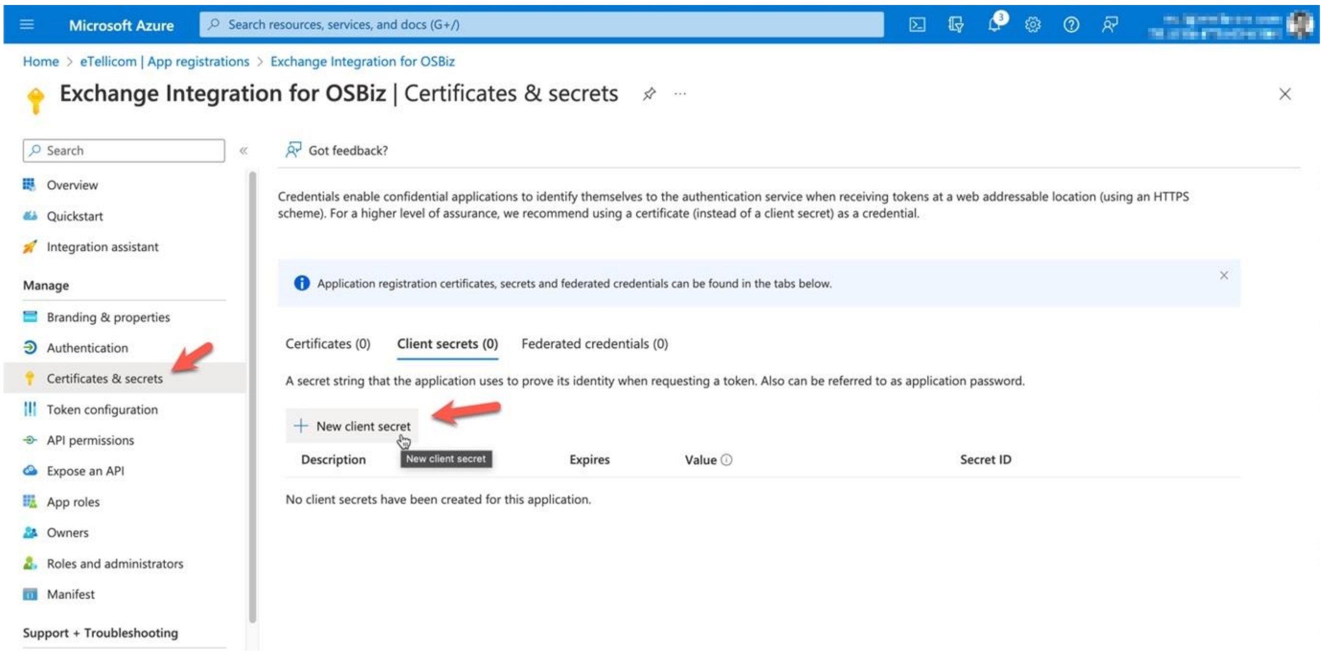
Configured permissions

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 eTellicom

API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	Granted for eTellicom
Office 365 Exchange Online (3)				
Contacts.Read	Application	Read contacts in all mailboxes	Yes	Granted for eTellicom
Contacts.ReadWrite	Application	Read and write contacts in all mailboxes	Yes	Granted for eTellicom
full_access_as_app	Application	Use Exchange Web Services with full access to all mailboxes	Yes	Granted for eTellicom

Now we need to create a client secret for the application. Select the “**Client & secrets**” menu and then click “**New client secret**”:



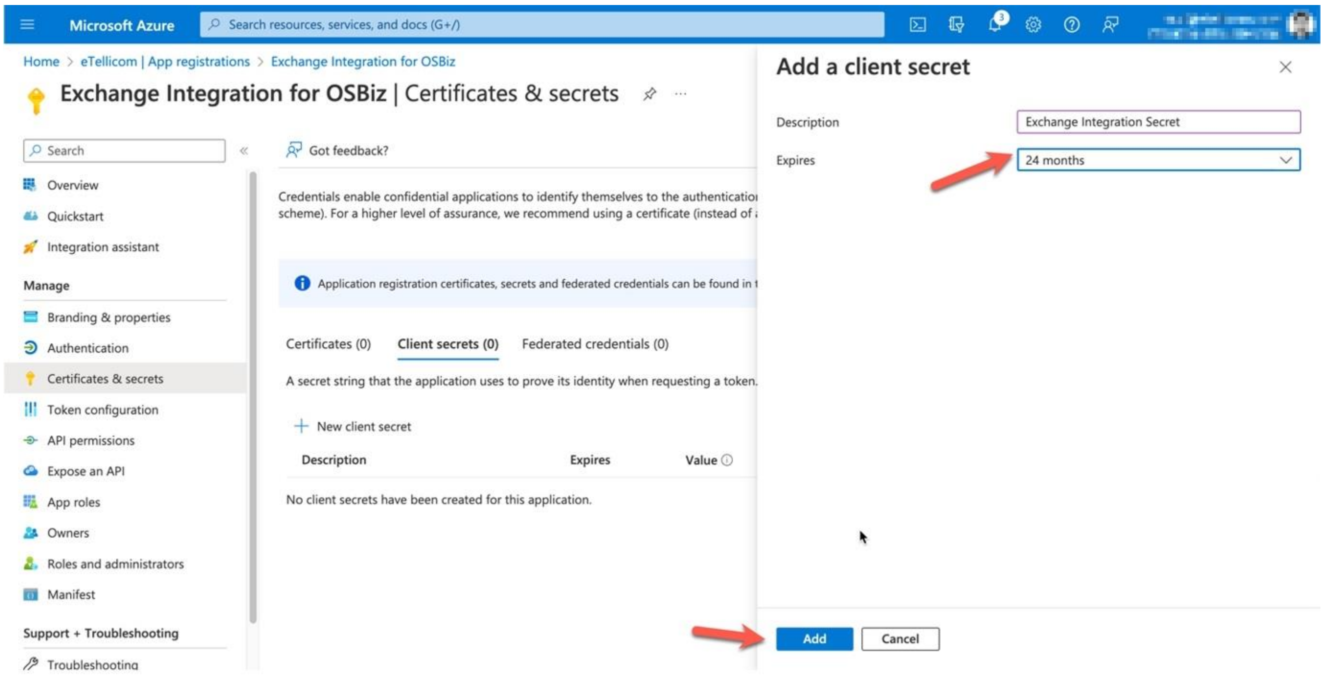
Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

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.

No client secrets have been created for this application.

Enter a description for the client secret (it’s just for informational purposes, as multiple secrets can be created for the same application) and select the expiration date option which fits to customer installation (example: maximum, 24 months), then click “**Add**”:

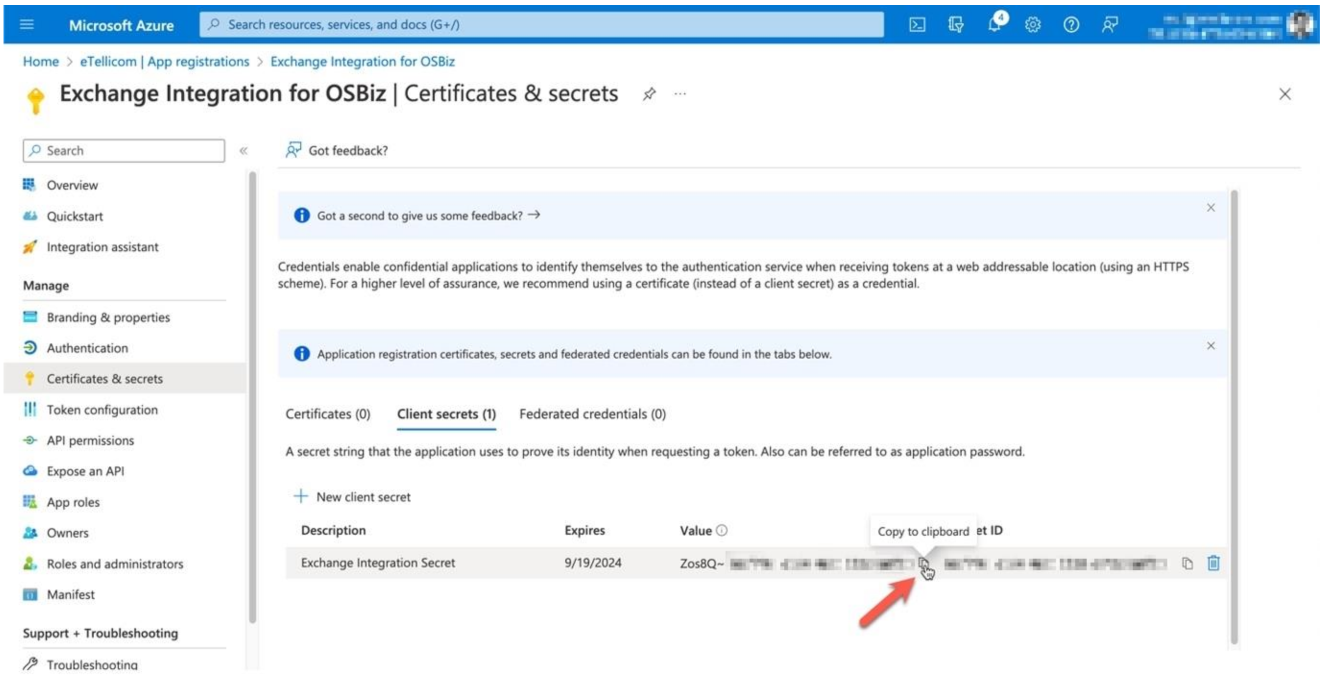

Microsoft Azure
Search resources, services, and docs (G+/)
Home > eTellicom | App registrations > Exchange Integration for OSBiz
Exchange Integration for OSBiz | Certificates & secrets
Add a client secret
Add
Cancel
Certificates (0)
Client secrets (0)
Federated credentials (0)
New client secret

Description: Exchange Integration Secret

Expires: 24 months

	The maximum runtime of the client secret is 24 months. To avoid reconfiguration, the secret value must be renewed before expiration.

IMPORTANT: at this point, you HAVE TO COPY the Value for the client secret by clicking the button next to it, as this the only time the Value will be ever visible and available for copying. If you don't do that at this point, that Client Secret Value cannot be used later when it's entered on WBM:


Microsoft Azure | Search resources, services, and docs (G+/) | [redacted]

Home > eTellicom | App registrations > Exchange Integration for OSBiz

Exchange Integration for OSBiz | Certificates & secrets

Search | 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

Got a second to give us some feedback? →

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.

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	Copy to clipboard et ID
Exchange Integration Secret	9/19/2024	Zos8Q~[redacted]	[redacted]

After this 1st part, you should have the following application details available, which will be used to configure the remaining parts of these instructions:

Display name, which in our example is:
`Exchange Integration for OSBiz`

Tenant ID (also called Directory ID), which in our example is:
`405232a0-ca59-4093-959f-xxxxxxxxxx`

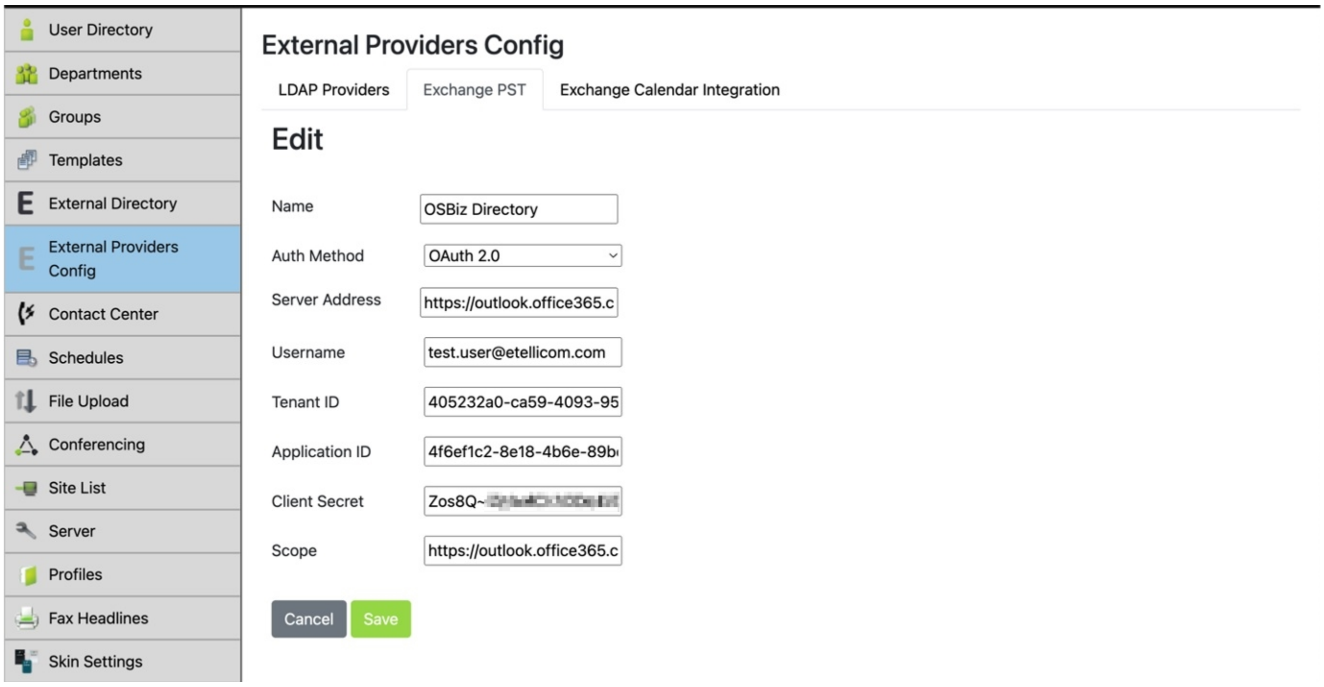
Application (or client) ID, which in our example is:
`4f6ef1c2-8e18-4b6e-89bc-xxxxxxxxxx`

Client Secret Value, which in our example is:
`Zos8Q~…`

ℹ	Please notice that this are all example values and should be replaced with the actual details from the newly registered application above.

3.2 Exchange Directory Integration in UC Suite WBM

Now that the application is registered with Azure, we can go to UC Suite WBM and open "**External Providers Config**", then select the "**Exchange PST**" tab. Change the Auth Method to "**Microsoft OAuth 2.0**" and the following fields will appear. Use a valid email address for this application that is registered through the Azure configuration within the Username field:



External Providers Config

Edit

Name	OSBiz Directory
Auth Method	OAuth 2.0
Server Address	https://outlook.office365.c
Username	test.user@etellicom.com
Tenant ID	405232a0-ca59-4093-95
Application ID	4f6ef1c2-8e18-4b6e-89b
Client Secret	Zos8Q~[redacted]
Scope	https://outlook.office365.c

Please enter the following values for Office 365 Exchange Online:

Server: `https://outlook.office365.com/EWS/Exchange.asmx`

Username: enter the user for your Microsoft Office 365 account

Tenant ID, Application ID, Client Secret: these are the same values that you took note from the last step in the 1st part of the configuration

Scope: `https://outlook.office365.com/.default`

Click [**Save**] and the configuration should now be completed.

ℹ	Public folder contacts search (notes and hints): • The public folder contacts search is restricted to the root level of the public folder structure only. Subfolders with contacts below the root level in the public folder structure are not queried. • Hence private contacts are not retrieved unless the user has explicitly marked the contact as public. • To test the public folder contact search functionality, use the search feature in myPortal for Desktop or myPortal for Outlook and search for a contact.

4 E-mail Queues with OAuth 2.0

The instructions below will cover the necessary steps to configure E-mail Queues using the OAuth 2.0 authentication method with Microsoft Office 365.

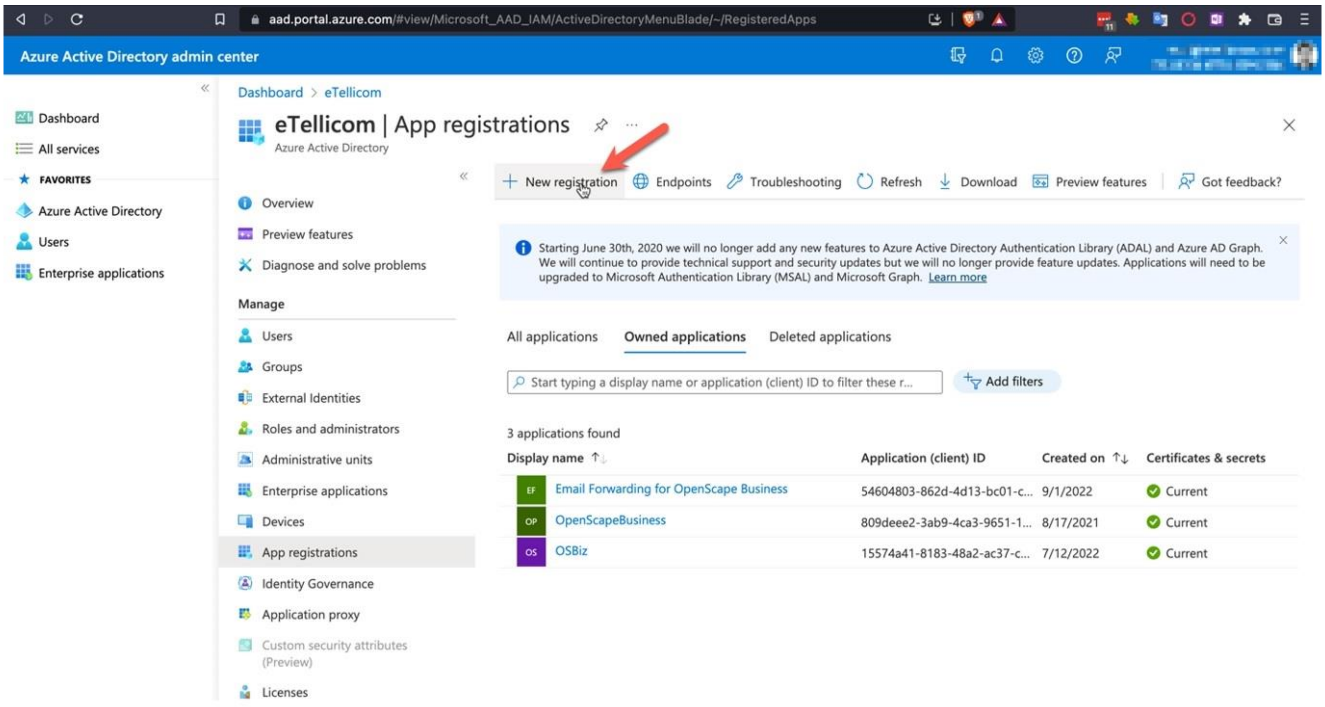
The configuration will be split in three parts. The 1st part covers the registration of the application in the Microsoft Azure Active Directory portal; the 2nd part covers the additional steps to be performed using the Windows PowerShell console tool to create and register a Service Principal for the application registered in the 1st part and finally assign the required permissions for the target mailbox to be used in the Contact Centre Queue; finally, the 3rd part will show how to create the E-mail Queue in WBM using the information from the previous parts.

4.1 Register the Application in Microsoft Azure AD

Open the browser and go to the Azure Active Directory admin center and login with your Administrator account: https://aad.portal.azure.com/.

The instructions assume that an application with Administrator rights is being used; if that's not possible, you can register the application and request the Administrator to grant the required API permissions later, please check this article for more details: https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-permissions-and-consent#requesting-consent-for-an-entire-tenant.

Select "**Azure Active Directory**" in the menu on the top=left corner, click on "**App registrations**", and then click "**New registration**":

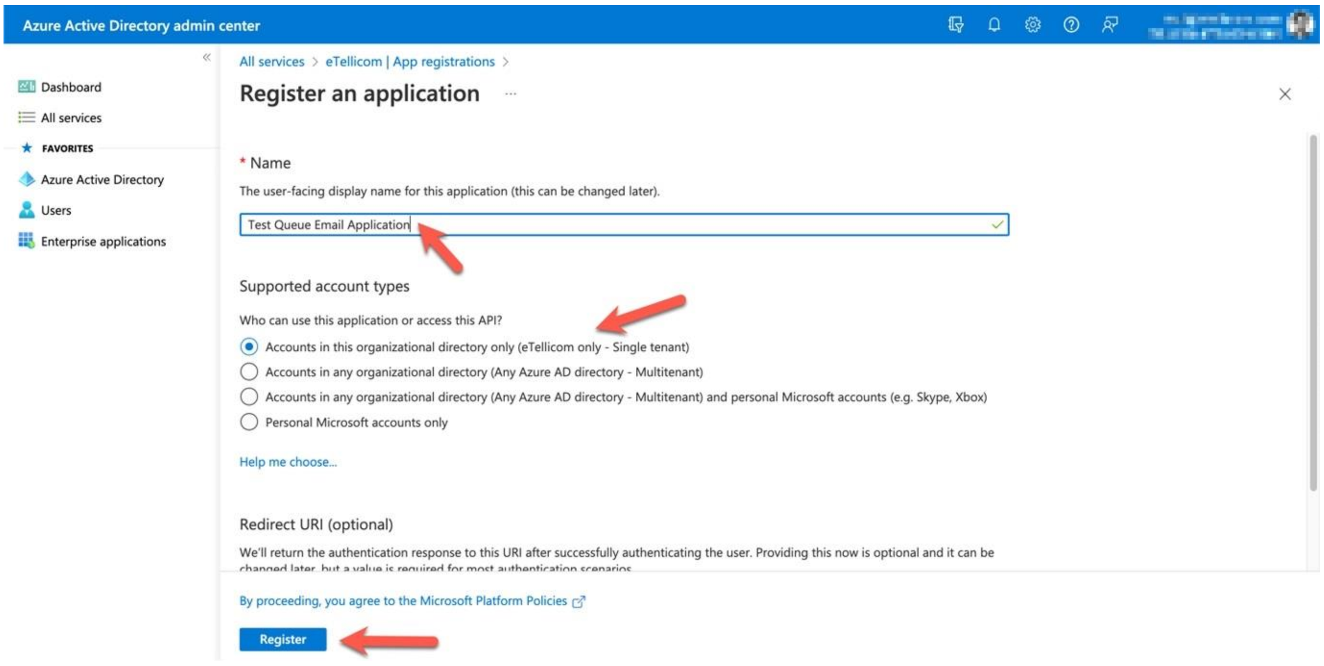


Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more

3 applications found

Display name	Application (client) ID	Created on	Certificates & secrets
EF Email Forwarding for OpenScape Business	54604803-862d-4d13-bc01-c...	9/1/2022	✓ Current
OP OpenScapeBusiness	809deee2-3ab9-4ca3-9651-1...	8/17/2021	✓ Current
OS OSBiz	15574a41-8183-48a2-ac37-c...	7/12/2022	✓ Current

Enter a meaningful name for the application (this will be later used to query the application when creating the Service Principal for Exchange Online), and for the “**Supported account type**”, select the option which fits to customer installation (example: **Single tenant**). Then click the **Register** button to complete the registration:


Azure Active Directory admin center

All services > eTellicom | App registrations >

Register an application

* Name

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

Test Queue Email Application

Supported account types

Who can use this application or access this API?

- [✓ Accounts in this organizational directory only (eTellicom 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...

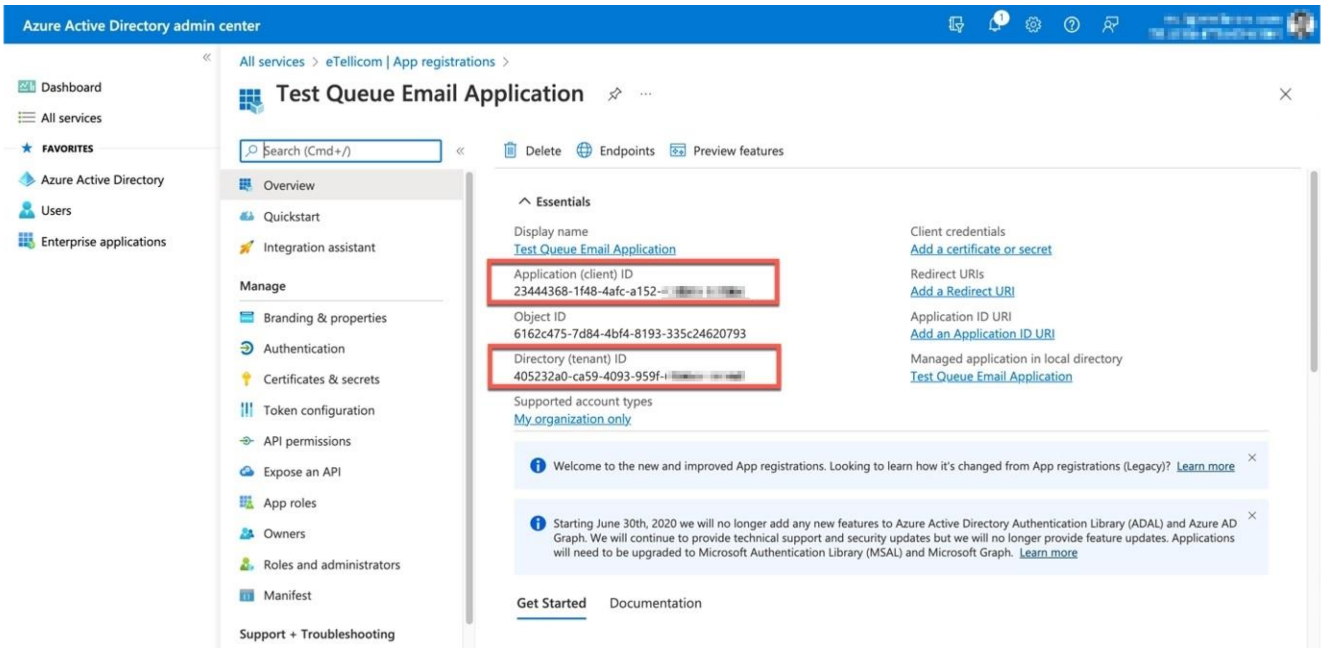
Redirect URI (optional)

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.

By proceeding, you agree to the Microsoft Platform Policies

	The same application can be used for all services, if the respective API permissions are assigned and granted, as well as the PowerShell commands (in the case of Email Queues only). For documentation purpose we choose a separate application with the example name: `Test Queue Email Application`

At the end of the process, Azure AD will present some information about the newly registered application. At this point, it's a good idea to take note of the "**Application ID**" (also known as client ID) and "**Directory (Tenant) ID**", as that information will be used to double-check the 2nd and 3rd part of this configuration:


Azure Active Directory admin center

Dashboard
All services
FAVORITES
Azure Active Directory
Users
Enterprise applications

All services > eTellicom | App registrations >

Test Queue Email Application

Search (Cmd+/)

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

Delete Endpoints Preview features

Essentials

Display name
Test Queue Email Application

Application (client) ID
23444368-1f48-4afc-a152-[redacted]

Object ID
6162c475-7d84-4bf4-8193-335c24620793

Directory (tenant) ID
405232a0-ca59-4093-959f-[redacted]

Supported account types
My organization only

Client credentials
Add a certificate or secret

Redirect URIs
Add a Redirect URI

Application ID URI
Add an Application ID URI

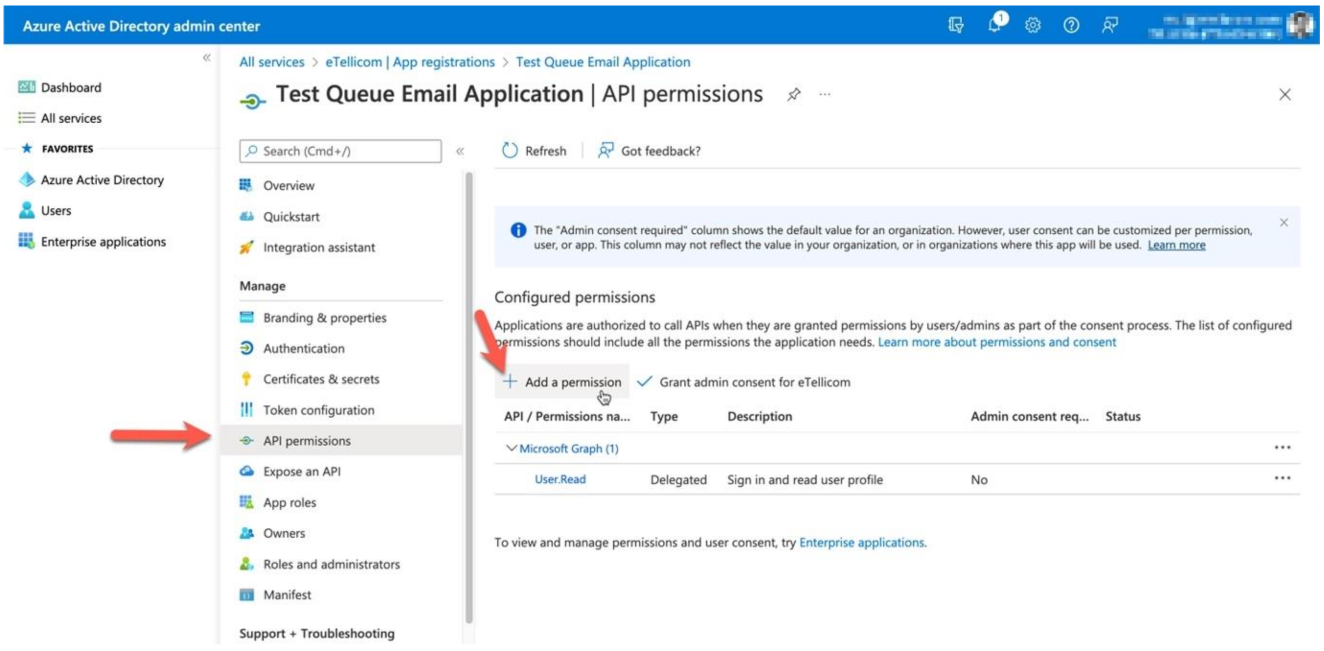
Managed application in local directory
Test Queue Email Application

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? Learn more

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more

Get Started Documentation

Now we need to assign the required permissions to be able to access the Email mailbox using the POP3 protocol for the Contact Centre. Click "**API permissions**" on the left, then click the "**Add a permission**" option to start adding the required API permission:



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

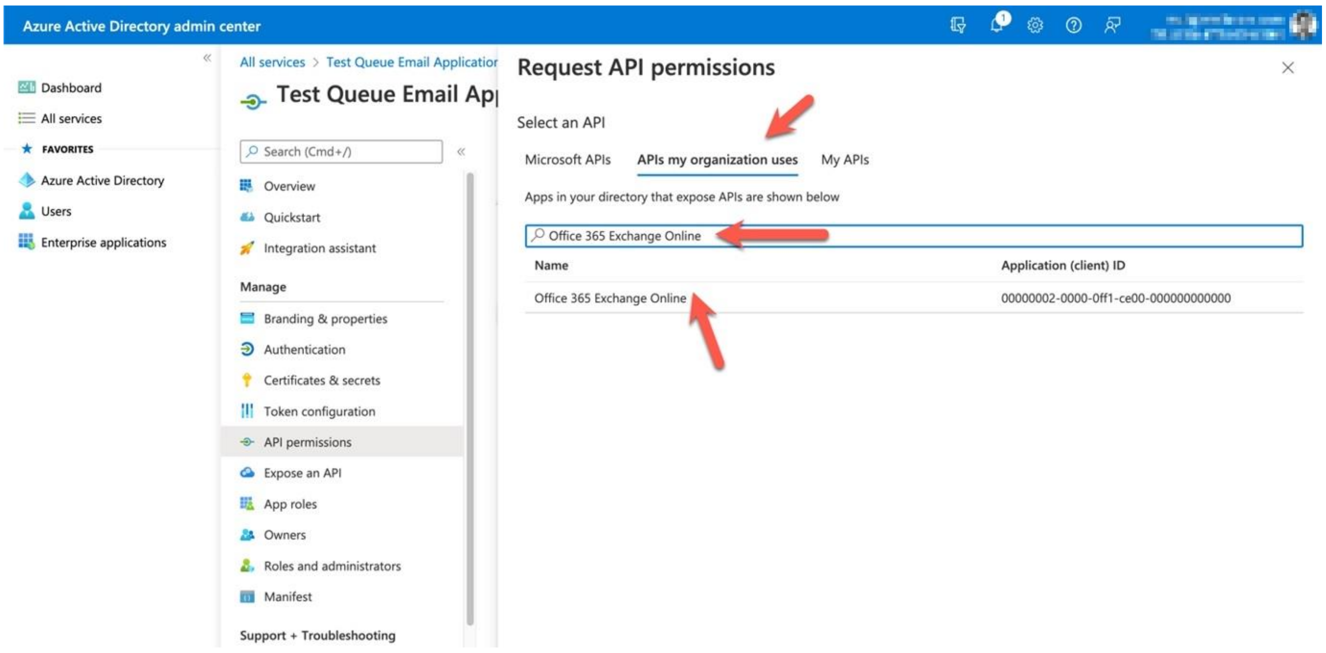
Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. Learn more about permissions and consent

+ Add a permission ✓ Grant admin consent for eTellicom

API / Permissions na...	Type	Description	Admin consent req...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	

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

Select the "**APIs my organization uses**" and type "**Office 365 Exchange Online**" (without the double quotes) entry, then click the row to select it:



Request API permissions

Select an API

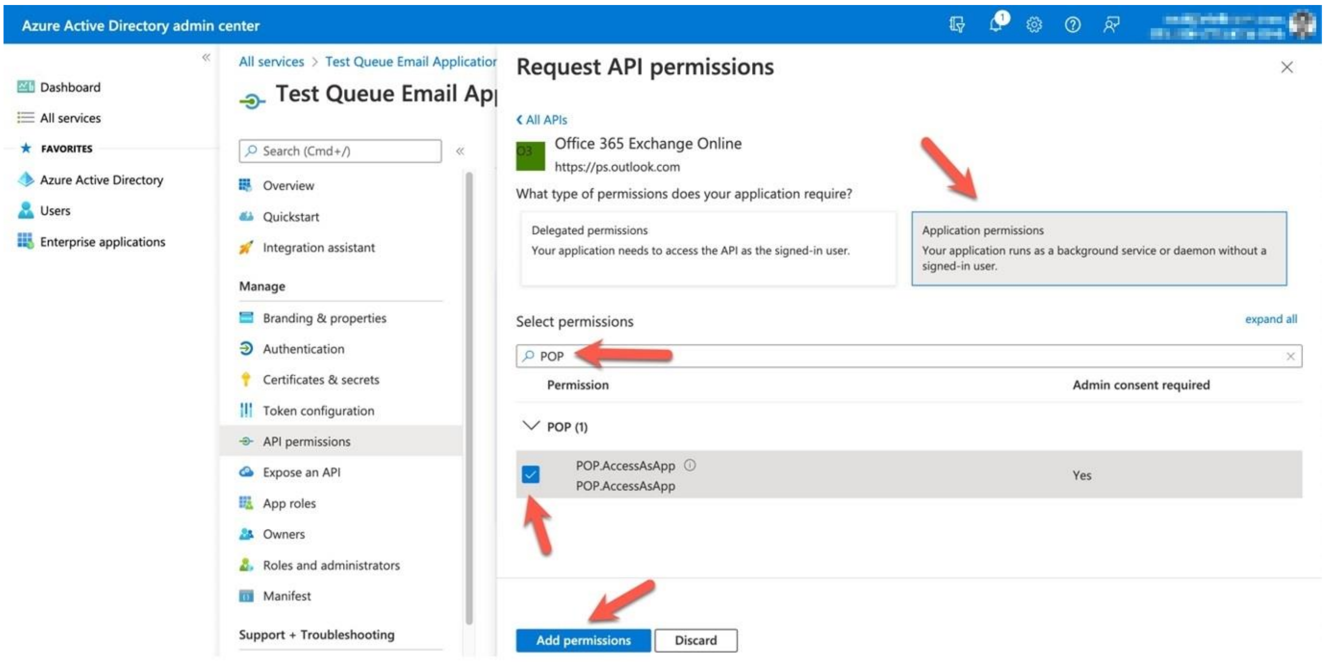
Microsoft APIs | APIs my organization uses | My APIs

Apps in your directory that expose APIs are shown below

Office 365 Exchange Online

Name	Application (client) ID
Office 365 Exchange Online	00000002-0000-0ff1-ce00-000000000000

Select the "**Application permissions**" box, and then type "**POP**" (without the double-quotes) in the selection box, expand the POP menu option and select the checkbox for the "**POP.AccessAsApp**" permission, then click the "**Add permissions**" button:

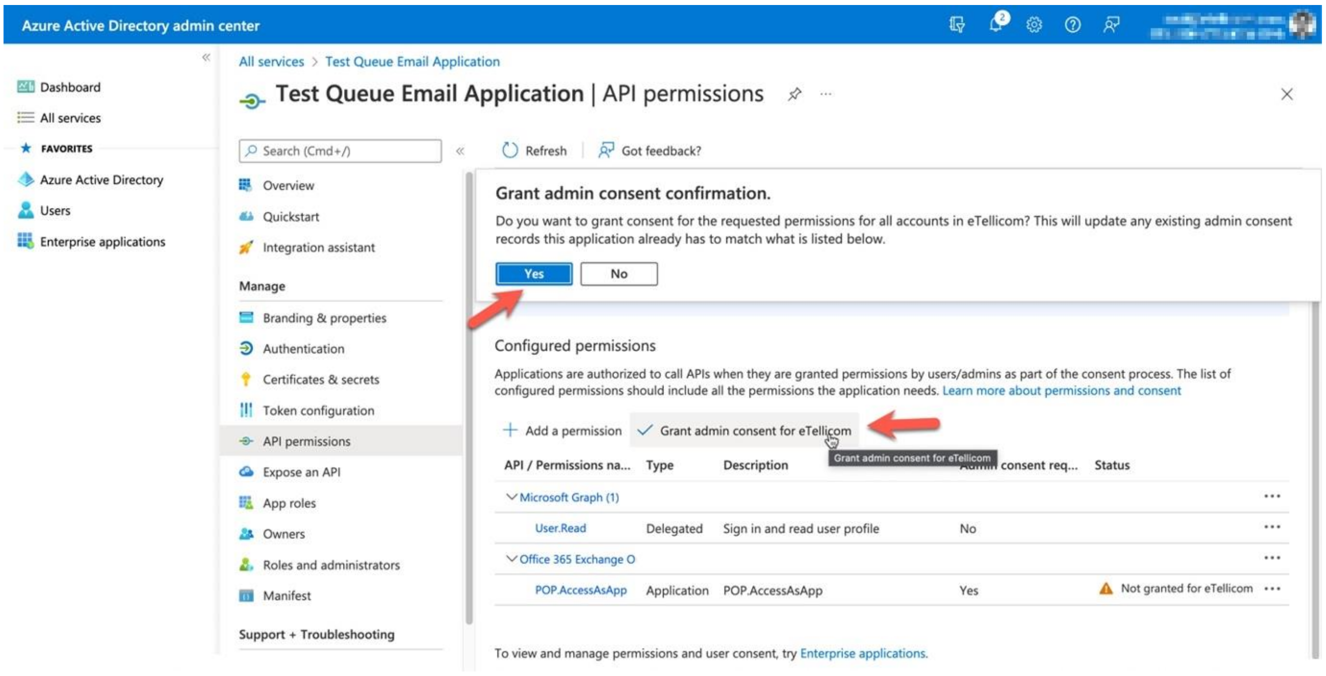


What type of permissions does your application require?

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.

We now need to grant the admin consent permission for our tenant to use the API we selected above, that's done by clicking the "Grant admin consent for <Tenant>", then clicking Yes in the dialog box that will appear:



Grant admin consent confirmation.

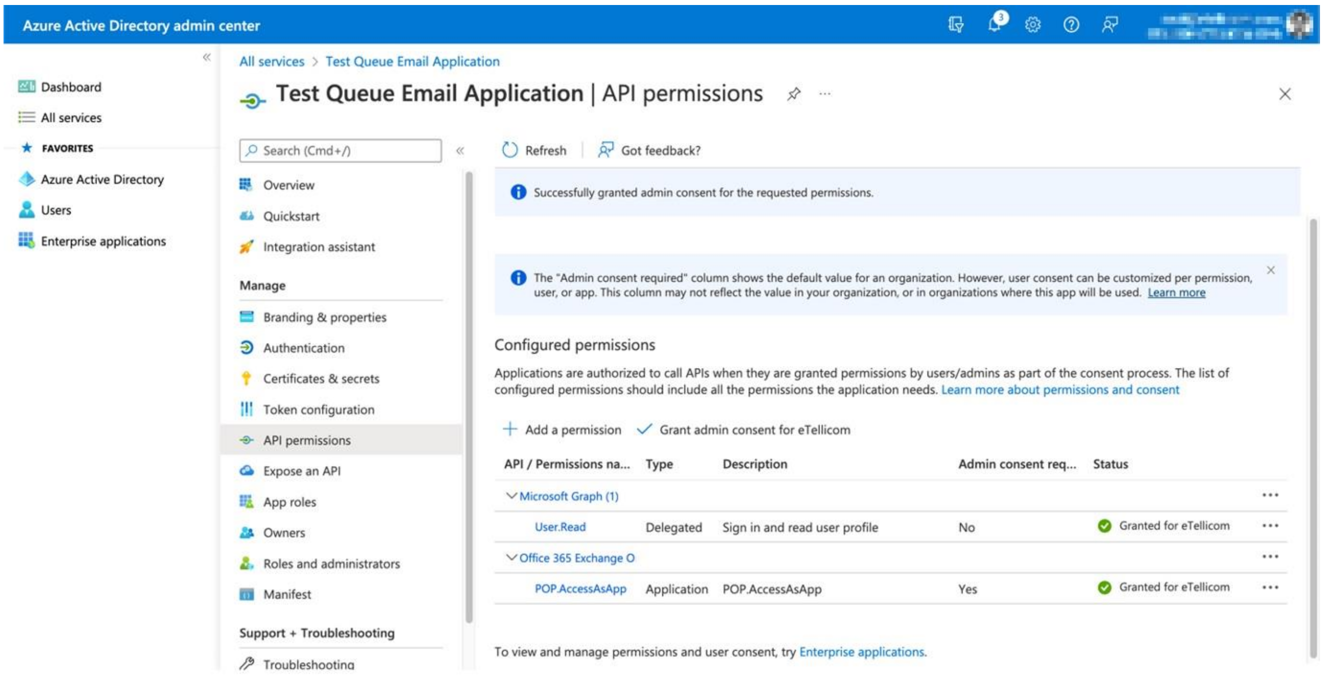
Do you want to grant consent for the requested permissions for all accounts in eTellicom? This will update any existing admin consent records this application already has to match what is listed below.

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 na...	Type	Description	Admin consent req...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	
Office 365 Exchange O				
POP.AccessAsApp	Application	POP.AccessAsApp	Yes	Not granted for eTellicom

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

Once it's completed, the Status column should display that the API is correctly granted for the tenant:


Test Queue Email Application | API permissions

Successfully granted admin consent for the requested permissions.

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

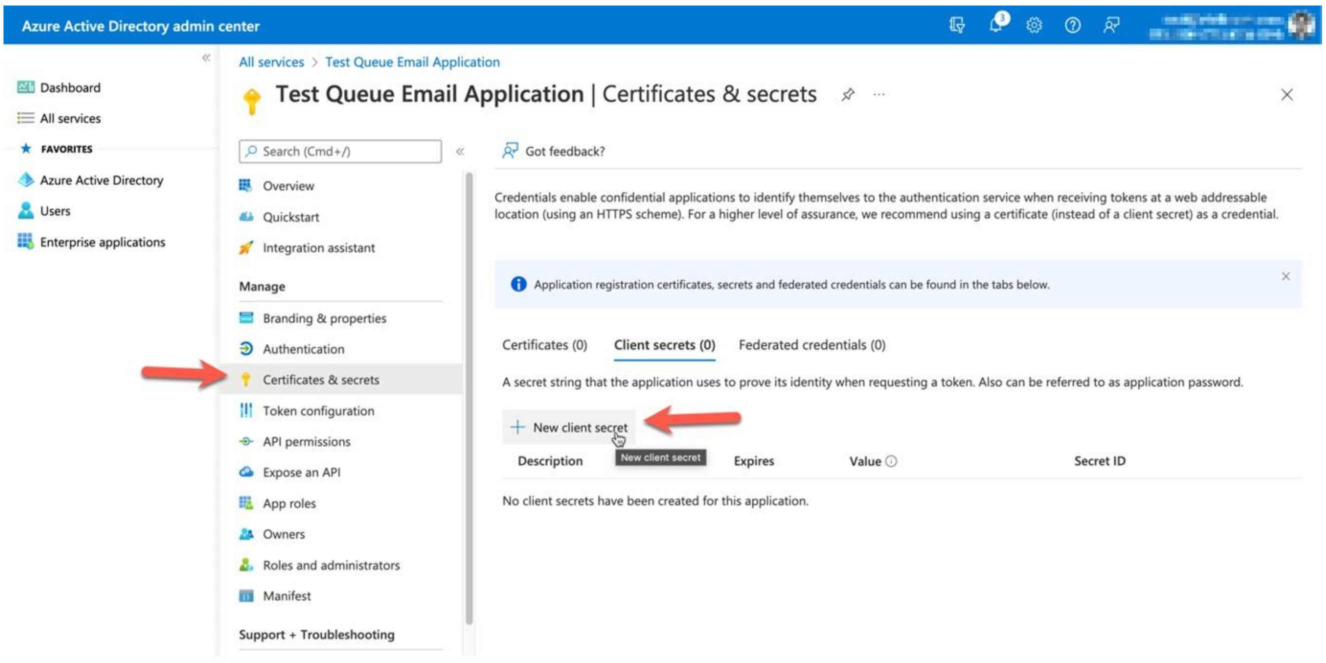
Configured permissions

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

API / Permissions na...	Type	Description	Admin consent req...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	Granted for eTellicom
Office 365 Exchange O				
POP.AccessAsApp	Application	POP.AccessAsApp	Yes	Granted for eTellicom

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

Now we need to create a client secret for the application. Select the "**Client & secrets**" menu and then click "**New client secret**":


Test Queue Email Application | Certificates & secrets

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.

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

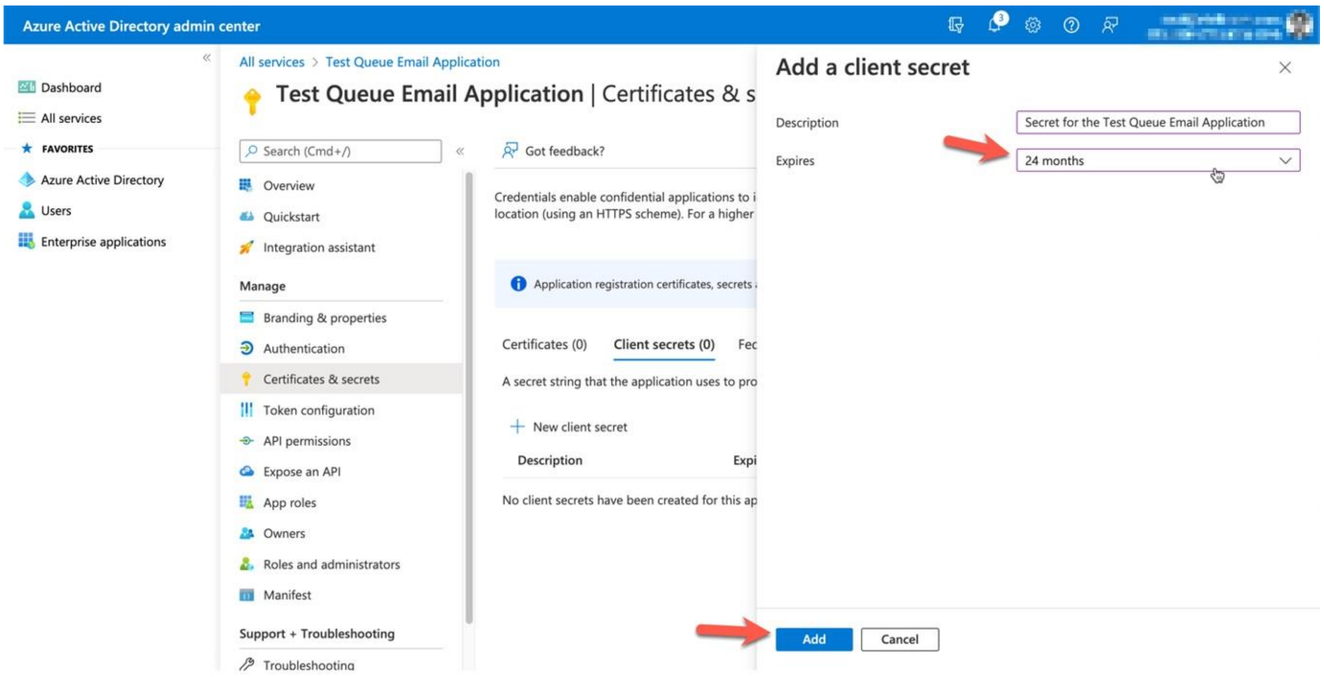
Certificates (0) | Client secrets (0) | Federated credentials (0)

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

Description	Expires	Value	Secret ID

No client secrets have been created for this application.

Enter a description for the client secret (it's just for informational purposes, as multiple secrets can be created for the same application) and select the expiration date option which fits to customer installation (example: maximum, 24 months), then click "**Add**":



Credentials enable confidential applications to i location (using an HTTPS scheme). For a higher

Application registration certificates, secrets

A secret string that the application uses to pro

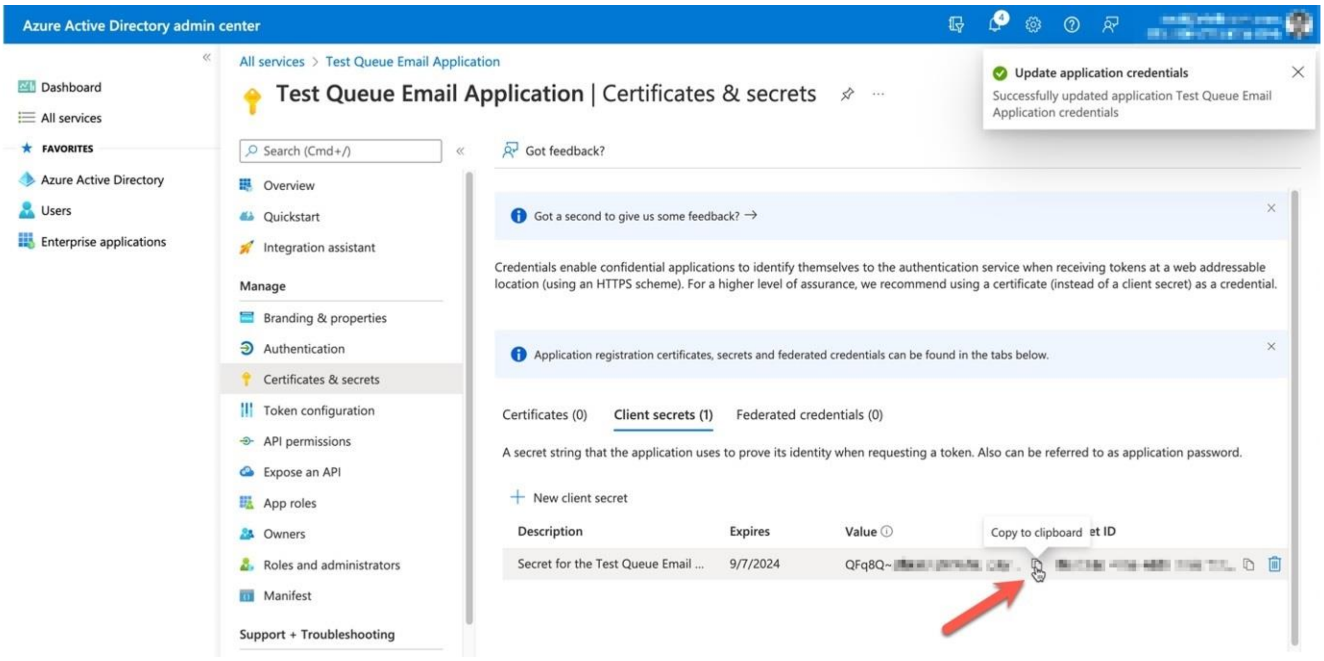
No client secrets have been created for this ap

Description: Secret for the Test Queue Email Application

Expires: 24 months

	The maximum runtime of the client secret is 24 months. To avoid reconfiguration, the secret value must be renewed before expiration.

IMPORTANT: at this point, you HAVE TO COPY the Value for the client secret by clicking the button next to it, as this the only time the Value will be ever visible and available for copying. If you don't do that at this point, that Client Secret Value cannot be used later when it's entered on WBM:



Azure Active Directory admin center [redacted]

Dashboard
All services
FAVORITES
Azure Active Directory
Users
Enterprise applications

All services > Test Queue Email Application

Test Queue Email Application | Certificates & secrets

Update application credentials
Successfully updated application Test Queue Email Application credentials

Search (Cmd+/)
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

Got feedback?

Got a second to give us some feedback? →

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.

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 for the Test Queue Email ...	9/7/2024	QFq8Q~[redacted]	[redacted]

Copy to clipboard

After this first part, you should have the following application details available, which will be used to configure the remaining parts of these instructions:

Display name, which in our example is:

`Test Queue Email Application`

Tenant ID (also called Directory ID), which in our example is:

`405232a0-ca59-4093-959f-xxxxxxxxxx`

Application (or client) ID, which in our example is:

`23444368-1f48-4afc-a152-xxxxxxxxxx`

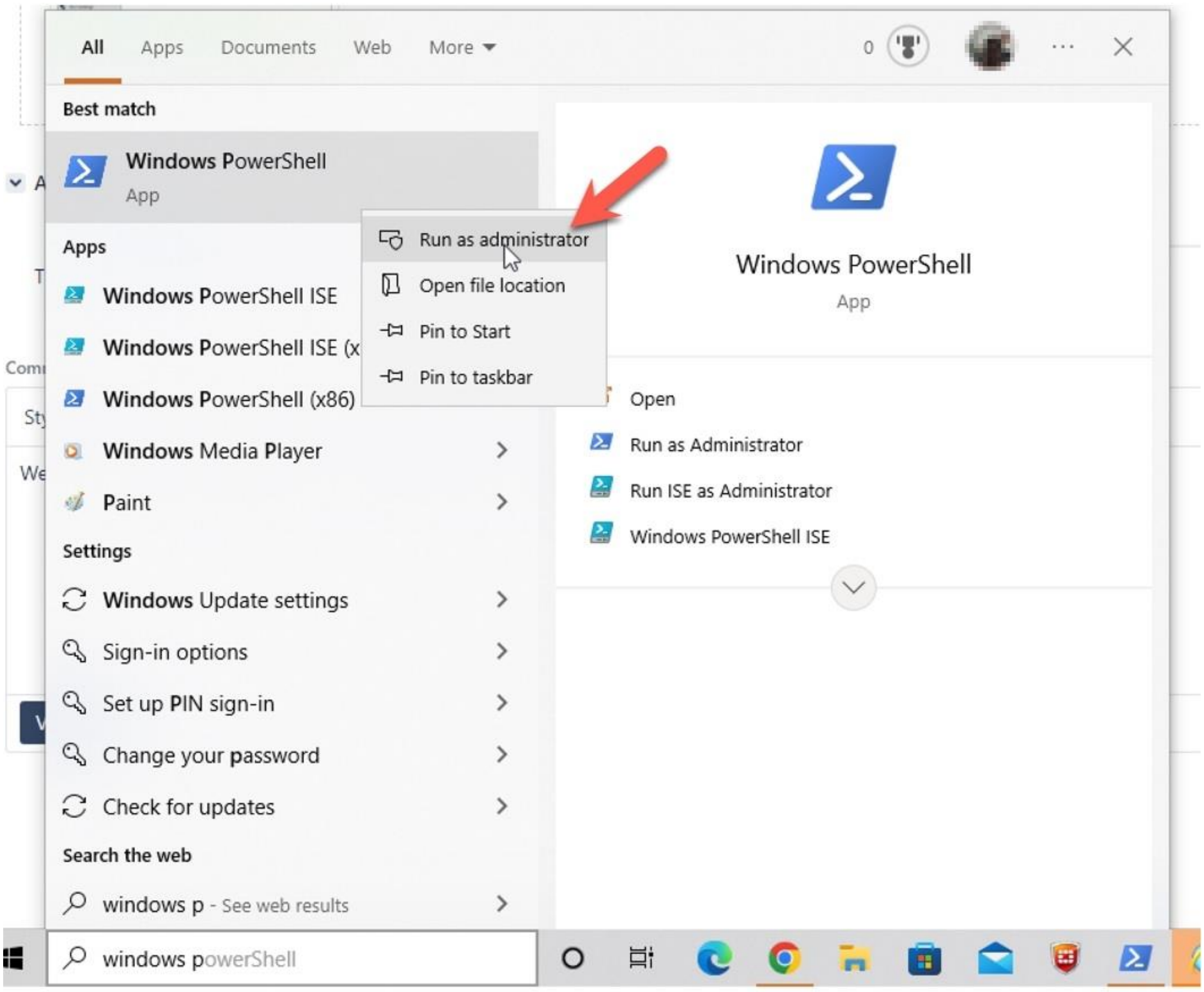
Client Secret Value, which in our example is:

`QFq8Q-…`

ℹ	Please notice that this are all example values and should be replaced with the actual details from the newly registered application above.

4.2 Service Principal and Mailbox permissions

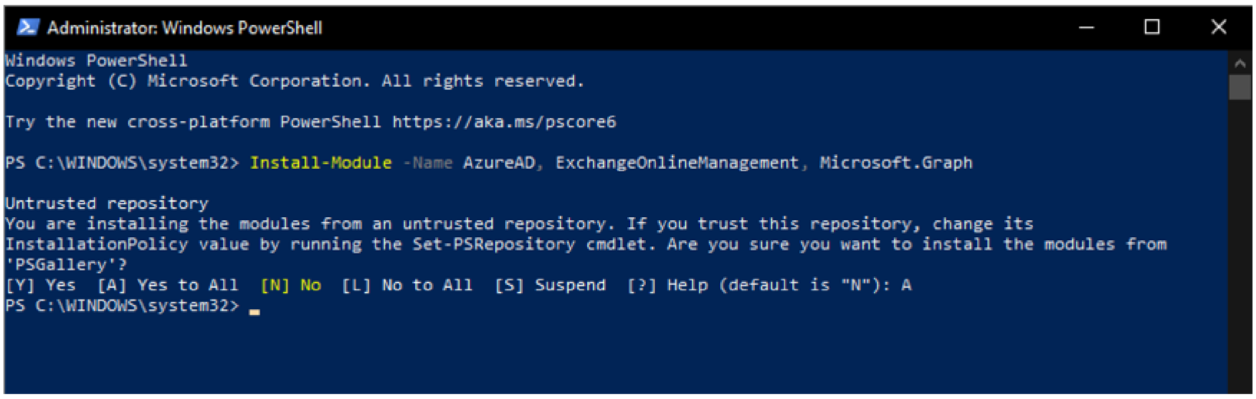
Now we need to use the **Windows PowerShell** to connect to the Azure AD service and the Exchange Online account for our tenant. Before we can do that, it's necessary to install the required PowerShell modules for those tasks. As that requires Administrator rights on the Windows PC being used for this task, you will have to right-click it and select "**Run as Administrator**":



When the console is open, enter the following command to install the three required modules for PowerShell:

```
Install-Module -Name AzureAD, ExchangeOnlineManagement, Microsoft.Graph
```

Then press Enter. Press “**A**” (for All) when it asks for confirmation. Wait for it to complete the installation:

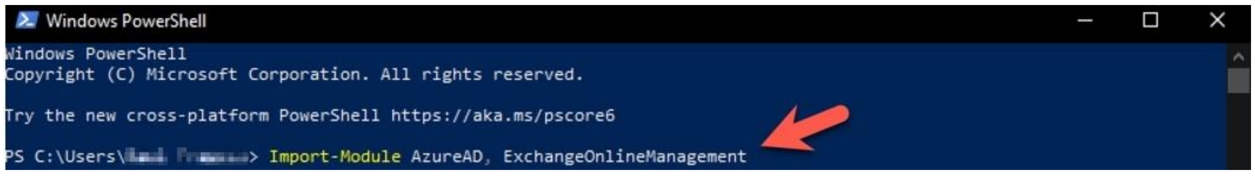


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> Install-Module -Name AzureAD, ExchangeOnlineManagement, Microsoft.Graph

Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its
InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from
'PSGallery'?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): A
PS C:\WINDOWS\system32> _
```

After that’s completed, type exit and press enter to close PowerShell.

We now need to open PowerShell again, but without running it as Administrator, so simply select it from the Windows applications menu. When it’s open, the first step is to import the modules we have installed in the previous step, we do that by typing the following command and pressing Enter:

```
Import-Module AzureAD, ExchangeOnlineManagement
```

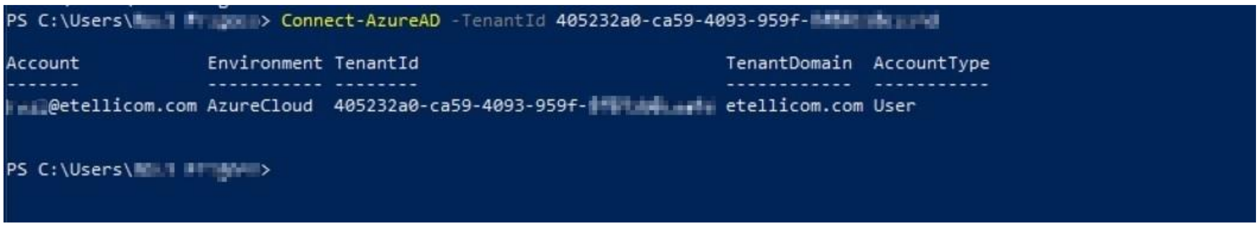


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\[redacted]> Import-Module AzureAD, ExchangeOnlineManagement
```

The next step is to connect to the Azure AD service by using the following command, please replace the Tenant ID [405232…] with the unique identifier for your tenant that you have from the first part of the instructions:

```
Connect-AzureAD -TenantId 405232a0-ca59-4093-959f-xxxxxxxxxx
```

Once you press Enter, a browser dialog will appear where you will need to authenticate with your Microsoft account for Azure AD. After that, it will show some information about the tenant you just connected to and return to the PowerShell console:

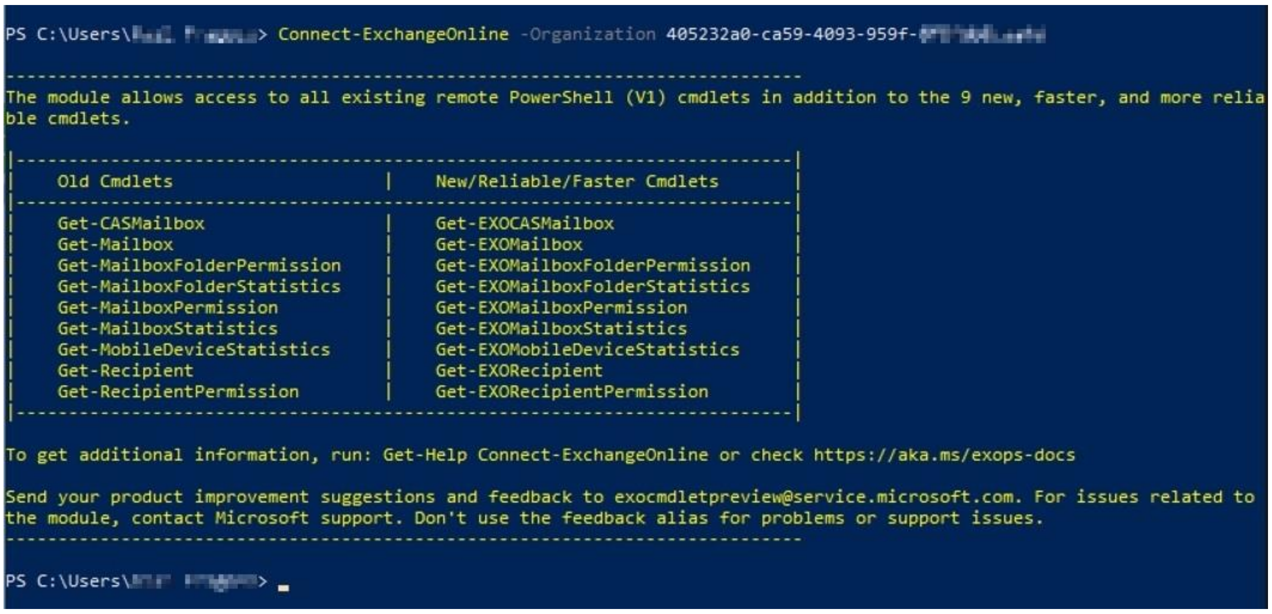


```
PS C:\Users\[redacted]> Connect-AzureAD -TenantId 405232a0-ca59-4093-959f-[redacted]

Account            Environment TenantId                              TenantDomain  AccountType
-------            ----------- --------                              ------------  -----------
[redacted]@etellicom.com AzureCloud  405232a0-ca59-4093-959f-[redacted] etellicom.com User

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

Now it’s time to connect to the Exchange Online service by using the following command (again, replace the Tenant ID with your own):

```
Connect-ExchangeOnline -Organization 405232a0-ca59-4093-959f-xxxxxxxxxx
```

Once again, a new browser window will appear to authenticate with the Microsoft account, just complete the authentication and it will return to the console:



```
PS C:\Users\[redacted]> Connect-ExchangeOnline -Organization 405232a0-ca59-4093-959f-[redacted]

----------------------------------------------------------------------------
The module allows access to all existing remote PowerShell (V1) cmdlets in addition to the 9 new, faster, and more relia
ble cmdlets.

|--------------------------------------------------------------------------|
|    Old Cmdlets                    |    New/Reliable/Faster Cmdlets       |
|--------------------------------------------------------------------------|
|    Get-CASMailbox                 |    Get-EXOCASMailbox                 |
|    Get-Mailbox                    |    Get-EXOMailbox                    |
|    Get-MailboxFolderPermission    |    Get-EXOMailboxFolderPermission    |
|    Get-MailboxFolderStatistics    |    Get-EXOMailboxFolderStatistics    |
|    Get-MailboxPermission          |    Get-EXOMailboxPermission          |
|    Get-MailboxStatistics          |    Get-EXOMailboxStatistics          |
|    Get-MobileDeviceStatistics     |    Get-EXOMobileDeviceStatistics     |
|    Get-Recipient                  |    Get-EXORecipient                  |
|    Get-RecipientPermission        |    Get-EXORecipientPermission        |
|--------------------------------------------------------------------------|

To get additional information, run: Get-Help Connect-ExchangeOnline or check https://aka.ms/exops-docs

Send your product improvement suggestions and feedback to exocmdletpreview@service.microsoft.com. For issues related to
the module, contact Microsoft support. Don't use the feedback alias for problems or support issues.
----------------------------------------------------------------------------

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

To make things easier for the next steps, we will create a variable named `$MyApp` to store the application details that we need, so we don’t need to enter them manually. We do that by entering the following command in the PowerShell console and pressing enter:

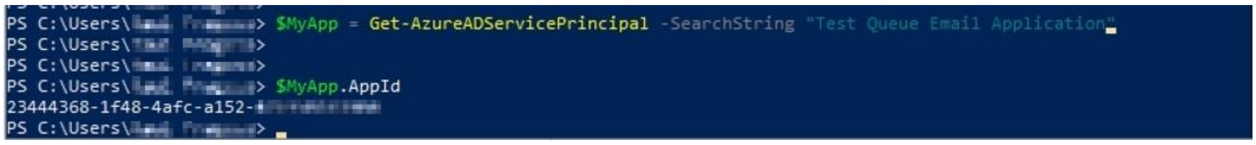
```
$MyApp = Get-AzureADServicePrincipal -SearchString "Test Queue Email
Application"
```

Remember to replace “**Test Queue Email Application**” with the Application Name that you have given to the application in the first part of these instructions.

You can optionally check that it worked by returning the `AppId` value for the variable with this command:

```
$MyApp.AppId
```

The displayed value should be the same as the Application ID assigned to the application in the first part of these instructions.



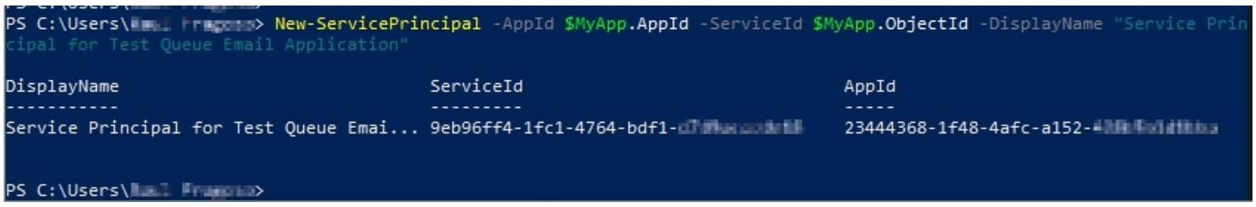
```
PS C:\Users\[redacted]> $MyApp = Get-AzureADServicePrincipal -SearchString "Test Queue Email Application"
PS C:\Users\[redacted]>
PS C:\Users\[redacted]>
PS C:\Users\[redacted]> $MyApp.AppId
23444368-1f48-4afc-a152-[redacted]
PS C:\Users\[redacted]> _
```

Now will be able to create the Service Principal for our application, by running the following command:

```
New-ServicePrincipal -AppId $MyApp.AppId -ServiceId $MyApp.ObjectId -
DisplayName "Service Principal for Test Queue Email Application"
```

Notice how we are using the properties of the `$MyApp` variable instead of entering the Application ID or other values manually, that helps with the readability of the command and is also less error prone.

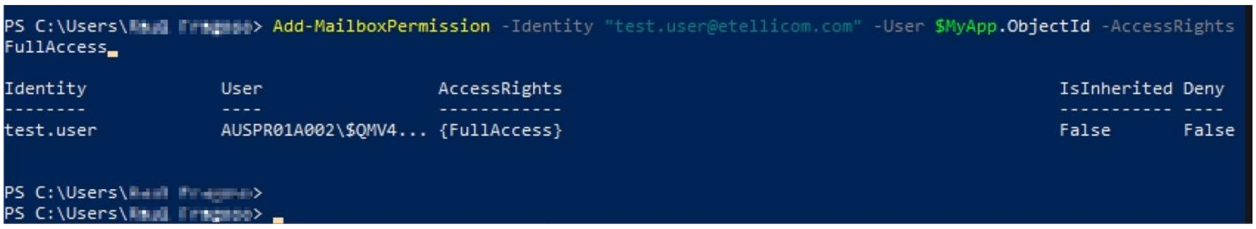
Please remember to replace the text for the `–DisplayName` with a better description for your application, but as this is for informational purposes only, just enter some meaningful text and press enter:



```
PS C:\Users\[redacted]> New-ServicePrincipal -AppId $MyApp.AppId -ServiceId $MyApp.ObjectId -DisplayName "Service Prin
cipal for Test Queue Email Application"

DisplayName                         ServiceId                              AppId
-----------                         ---------                              -----
Service Principal for Test Queue Emai... 9eb96ff4-1fc1-4764-bdf1-[redacted]    23444368-1f48-4afc-a152-[redacted]


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

The final step is to give the proper permission to the Mailbox user that will be used with the Email account for the Call Centre queue. That's done with the following command, you will have to replace the -Identity value with the actual email account that you want to assign the permission to:

```
Add-MailboxPermission -Identity "user@domain.com" -User $MyApp.ObjectId -
AccessRights FullAccess
```



```
PS C:\Users\[redacted]> Add-MailboxPermission -Identity "test.user@etellicom.com" -User $MyApp.ObjectId -AccessRights
FullAccess_

Identity             User                 AccessRights                                                IsInherited Deny
--------             ----                 ------------                                                ----------- ----
test.user            AUSPR01A002\$QMV4... {FullAccess}                                                False       False


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

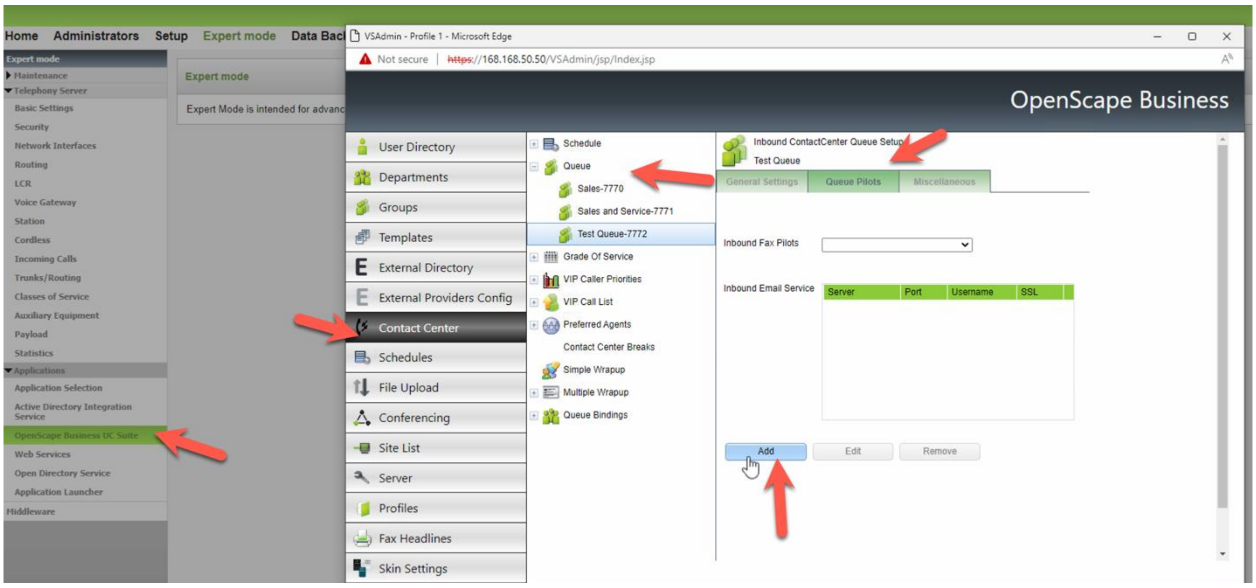
It's also important to notice that you can add multiple mailbox accounts to the same application, so if you have more accounts to be used with multiple queues, you can reuse the same application by simply adding the additional identities with the command above.

This is all that need to be done in PowerShell, so you can now close it and proceed with the 3rd part of the configuration.

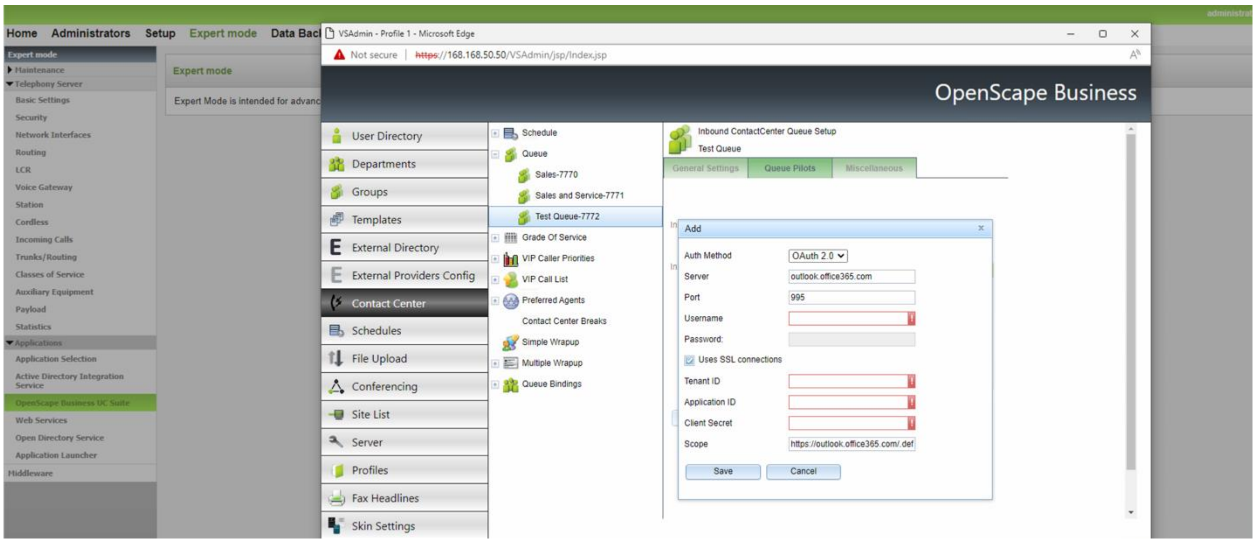
4.3 Create the E-mail for the Contact Centre Queue

Now that the application is registered with Azure AD and the Service Principal is created with the correct mailbox permissions for the Email account with Exchange Online, we can create the Email Queue pilot for the Contact Centre using the information that we collected from the 1st part of these instructions.

Go to WBM and open OpenScape Business UC Suite, select "**Contact Center**", expand the Queue menu, and select the respective Queue that you want to create the Email pilot for, then in the Queue Pilots tab, click "**Add**" to create a new Email pilot for the queue.



Change the Auth Method to "**OAuth 2.0**" and the following fields will appear:



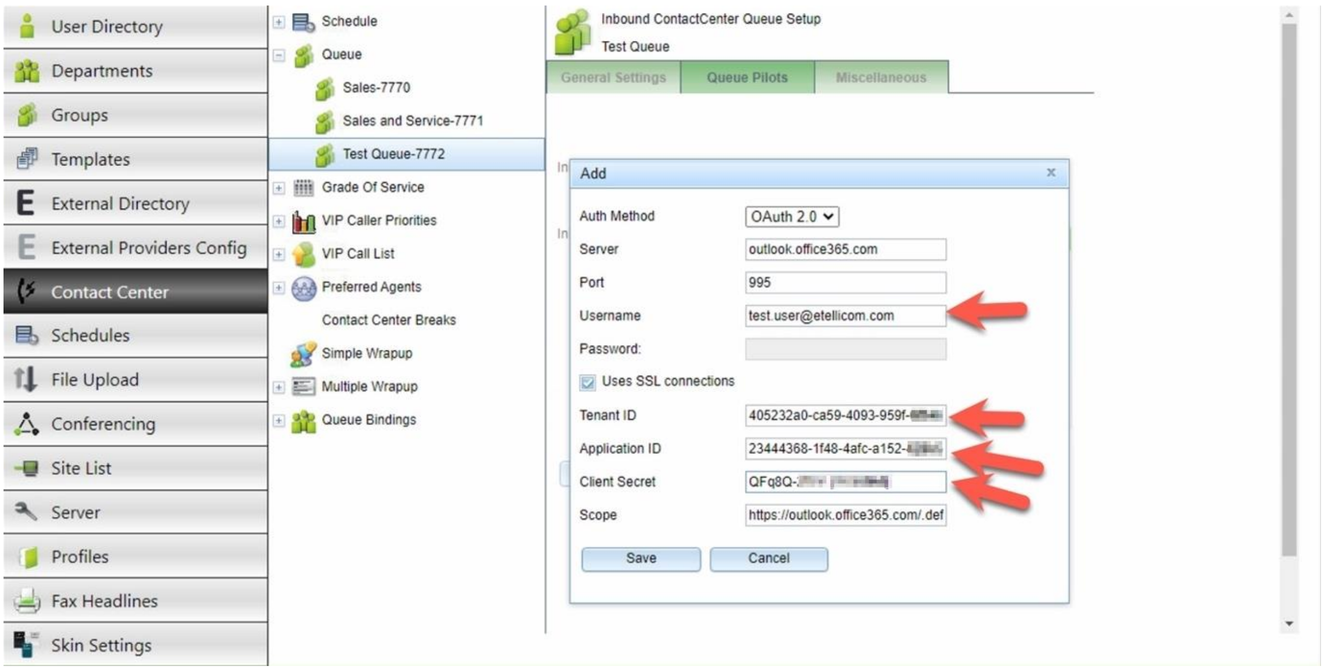
Auth Method	OAuth 2.0
Server	outlook.office365.com
Port	995
Username	
Password:	
☑ Uses SSL connections	
Tenant ID	
Application ID	
Client Secret	
Scope	https://outlook.office365.com/.def

All the default values can be left as they are, but you will have to enter the following fields:

Username: this is the same as the mailbox identity that you used in the last step for the 2nd part of the configuration

Tenant ID, Application ID, Client Secret value: these are the same values that you took note from the last step in the 1st part of the configuration

Taking our example from this document, we would have the following entered:



Auth Method	OAuth 2.0
Server	outlook.office365.com
Port	995
Username	test.user@etellicom.com
Password:	
Uses SSL connections	✓
Tenant ID	405232a0-ca59-4093-959f-[redacted]
Application ID	23444368-1f48-4afc-a152-[redacted]
Client Secret	QFq8Q-[redacted]
Scope	https://outlook.office365.com/.def

Click [**Save**] and the configuration should now be completed.

5 myReports with OAuth 2.0

ℹ	myReports uses the E-mail forwarding configuration, please refer to chapter 1.

6 Support & Serviceability

6.1 Required trace files for error analysis

- OpenScape Business Diagnosis Logs including:
  - Event Log
  - Framework Protocol
  - UC Suite Protocols (if activated)

7 Best Practice

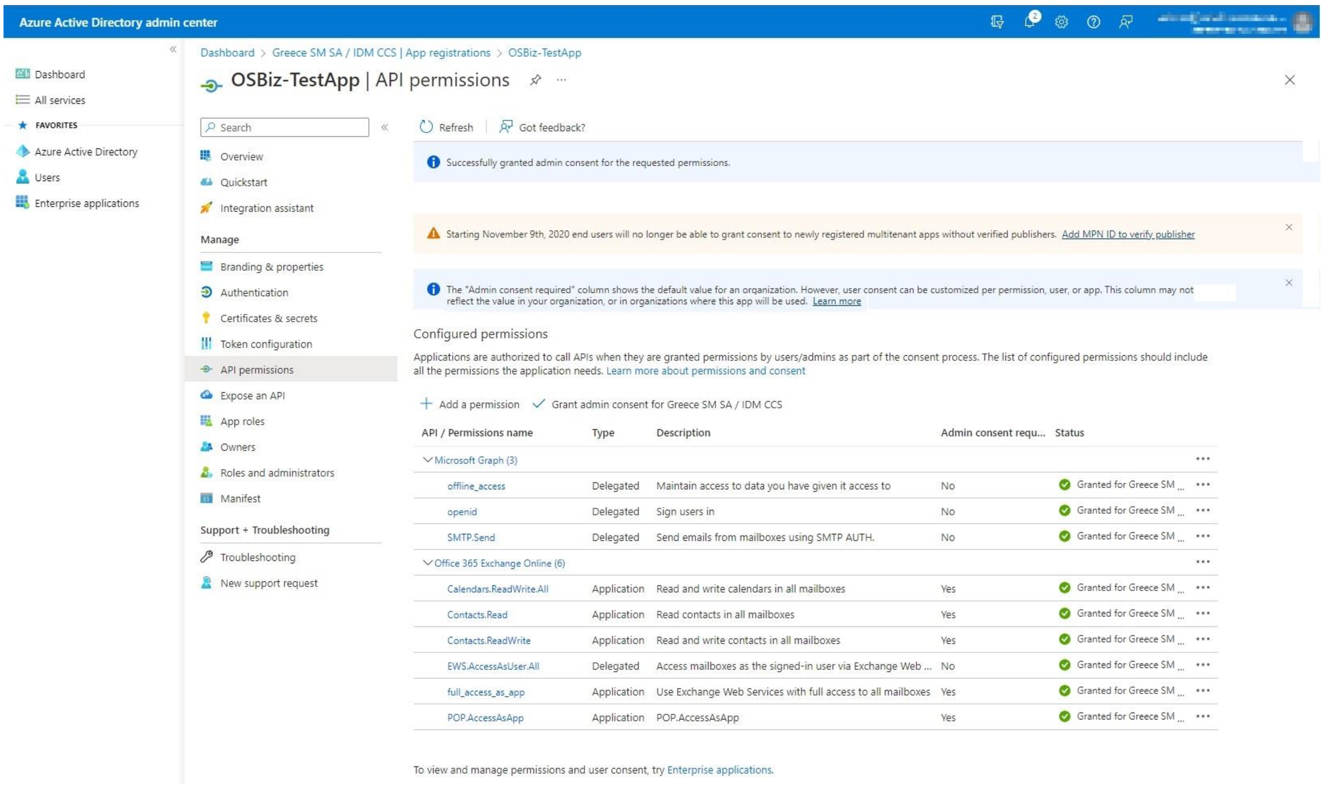
Information and useful hints from our lab and customer installations.

7.1 use the same application for all functions

The “**OSBiz-TestApp**” in this example is merging the API permissions of:

- E-mail forwarding
- Calendar Integration
- Directory Integration
- E-mail Queues

For details of permissions please refer to the previous chapters and combine the permissions as needed. A partial usage is possible as well:



Successfully granted admin consent for the requested permissions.

Starting November 9th, 2020 end users will no longer be able to grant consent to newly registered multitenant apps without verified publishers. Add MPN ID to verify publisher

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

API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (3)				
offline_access	Delegated	Maintain access to data you have given it access to	No	✓ Granted for Greece SM ...
openid	Delegated	Sign users in	No	✓ Granted for Greece SM ...
SMTP.Send	Delegated	Send emails from mailboxes using SMTP AUTH.	No	✓ Granted for Greece SM ...
Office 365 Exchange Online (6)				
Calendars.ReadWrite.All	Application	Read and write calendars in all mailboxes	Yes	✓ Granted for Greece SM ...
Contacts.Read	Application	Read contacts in all mailboxes	Yes	✓ Granted for Greece SM ...
Contacts.ReadWrite	Application	Read and write contacts in all mailboxes	Yes	✓ Granted for Greece SM ...
EWS.AccessAsUser.All	Delegated	Access mailboxes as the signed-in user via Exchange Web ...	No	✓ Granted for Greece SM ...
full_access_as_app	Application	Use Exchange Web Services with full access to all mailboxes	Yes	✓ Granted for Greece SM ...
POP.AccessAsApp	Application	POP.AccessAsApp	Yes	✓ Granted for Greece SM ...

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

7.2 Testing POP3 access

Script for testing POP3 access using the OAuth 2.0 authentication with Microsoft Exchange Online.

	Please notice that below are example values and should be replaced with the details from the registered application.

```
#!/bin/sh

######################################################################
#
# Script for testing POP3 access using the OAuth 2.0 authentication
# with Microsoft Exchange Online
#
######################################################################

# Replace the variables below with their appropriate values

OAUTH_TENANT_ID=REPLACE_WITH_YOUR_TENANT_ID_HERE
OAUTH_CLIENT_ID=ADD_YOUR_APPLICATION_ID_HERE
OAUTH_CLIENT_SECRET=PUT_YOUR_CLIENT_SECRET_VALUE_HERE
OAUTH_EMAIL_USER=your.user@yourcompany.com

######################################################################

# Do NOT change these variables, they are put  here just for better
# readability of the commands

OAUTH_GRANT_TYPE=client_credentials
OAUTH_SCOPE=https://outlook.office365.com/.default

######################################################################

echo "################################################################"
echo "OAUTH_TENANT_ID     = $OAUTH_TENANT_ID"
echo "OAUTH_CLIENT_ID     = $OAUTH_CLIENT_ID"
echo "OAUTH_CLIENT_SECRET = $OAUTH_CLIENT_SECRET"
echo "OAUTH_EMAIL_USER    = $OAUTH_EMAIL_USER"
echo "OAUTH_SCOPE         = $OAUTH_SCOPE"
echo "OAUTH_GRANT_TYPE    = $OAUTH_GRANT_TYPE"
echo "################################################################"

# Send the request to authenticate the client credentials and obtain
# the Bearer token which will be used with the next API requests

echo "Sending token request ..."

AUTHORIZATION_RESULT=$(curl --location --request POST
"https://login.microsoftonline.com/$OAUTH_TENANT_ID/oauth2/v2.0/token" \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode "grant_type=$OAUTH_GRANT_TYPE" \
--data-urlencode "client_id=$OAUTH_CLIENT_ID" \
--data-urlencode "client_secret=$OAUTH_CLIENT_SECRET" \
--data-urlencode "scope=$OAUTH_SCOPE" -s)
```

```
OAUTH_BEARER_TOKEN=$(echo $AUTHORIZATION_RESULT | grep -o '"access_token": *"[^"]*' | grep
-o '[^"]*$')

if [ -z "$OAUTH_BEARER_TOKEN" ]
then
       echo "Bearer token is empty, check for errors in the output below:"
       echo "$AUTHORIZATION_RESULT"
       exit 1
else
       echo "Returned token: $OAUTH_BEARER_TOKEN"
fi

echo "################################################################"

# Send the POP3 request to obtain the first email in the list

echo "Sending POP3 request to read the first email from the mailbox ..."

curl --user $OAUTH_EMAIL_USER --oauth2-bearer $OAUTH_BEARER_TOKEN --ssl --verbose
'pop3s://outlook.office365.com:995/1'

exit 0
```

7.3 Further

ℹ	Please send us important hints of your customer installation which are missed in the general description and worth to add here and share with the community. Thank you !!!

mitel.com

Powering connections