A MITEL
PRODUCT
GUIDE

Mitel OpenScape Business

Microsoft Exchange Online / Office 365
Register in Microsoft Entra ID

Release Number 08/2026

Contents

History of Changes

Date	Issue	Summary
04.08.2026	1.0	initial version for OpenScape Business V4

Note: The basis for this document is the current OpenScape Business. 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

***Disclaimer**:*

Microsoft Branding, Pictures and Icons in this document might be under copyright of Microsoft.

The Microsoft examples in this document give a rough overview of needed components in a basic setup and require individual verification for customers need.

Settings and configuration might change due to different Software versions.

Please note:

Microsoft is a 3rd party product. Mitel offers data interworking capabilities for Microsoft with a technical description of how to configure the OpenScape Business.

Mitel doesn´t deliver any administration services for Microsoft.

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.

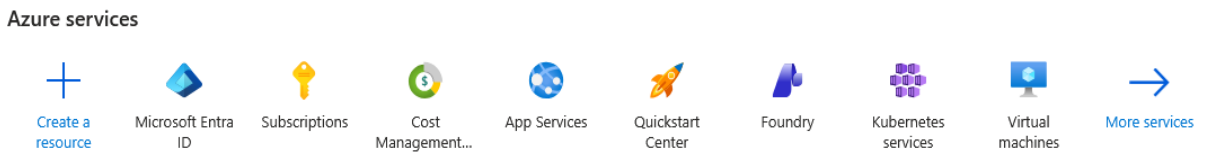
OpenScape Business uses postfix to connect to an exchange server (relay server) to send email notifications to required recipients. Till now this has been done using the basic authentication scheme. Microsoft will obsolete the basic authentication method to all its **Exchange Online Servers** in favor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

The configuration is split into two parts. The 1st part covers the registration of the application in the Microsoft Entra ID Directory portal; the 2nd part will show how to create the E-mail forwarding in Admin portal (WBM) using the information from the previous part.

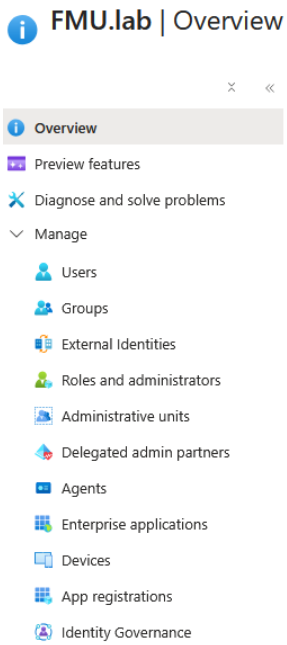
Hint: SMTP AUTH must be enabled at least for the used service account (this is often disabled organization-wide). Refer to: [Enable or disable SMTP AUTH in Exchange Online | Microsoft Learn]

1.1 Register the Application in Microsoft Entra ID

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

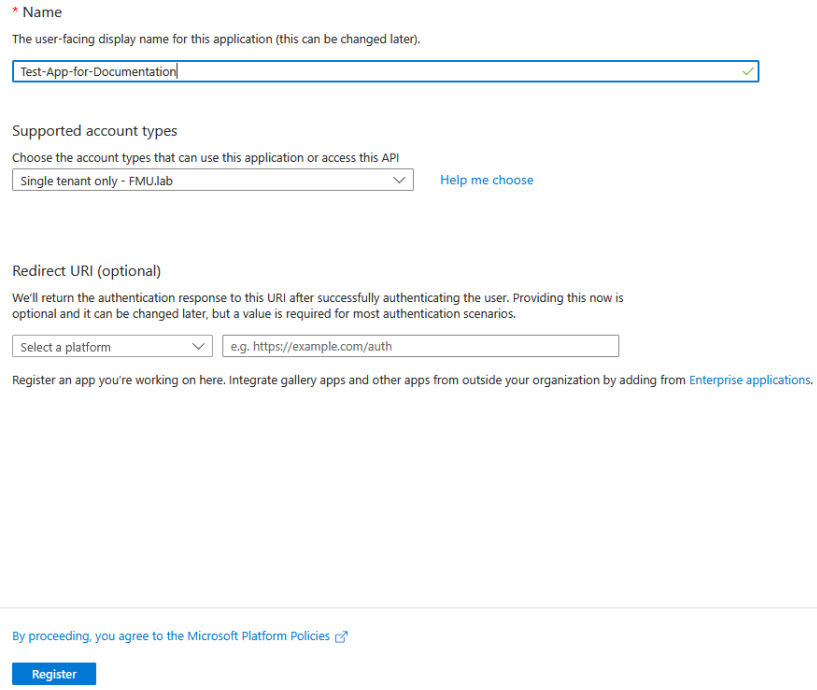


and select "**App Registrations**":



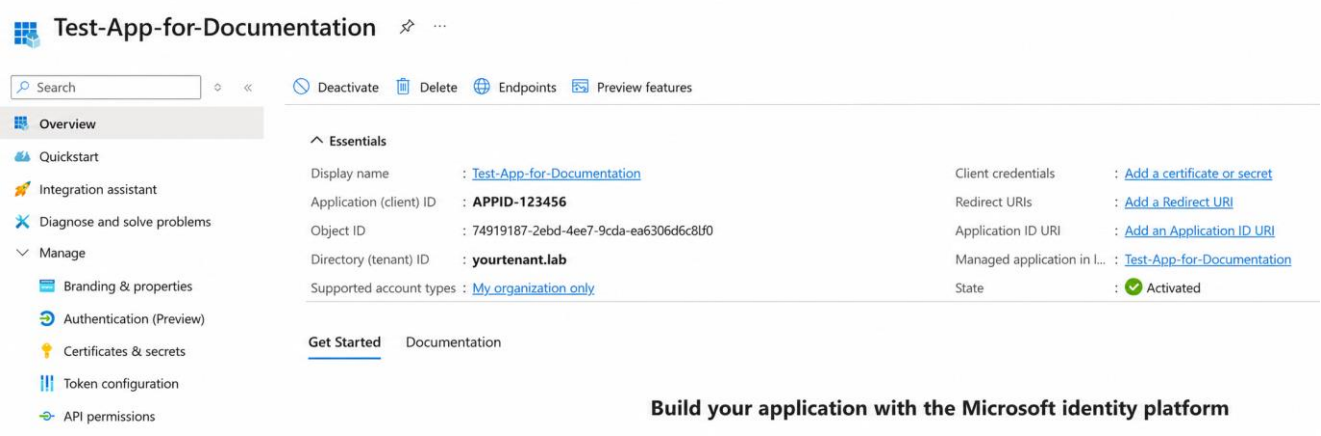
On the top, select "**New Registration**" and on the new application registration page

- 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

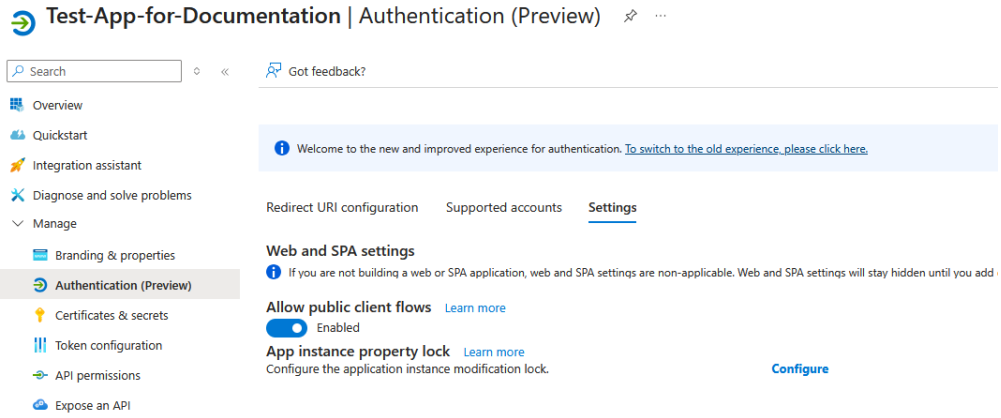


	The same application can be used for all services if the respective Microsoft API permissions are assigned and granted.

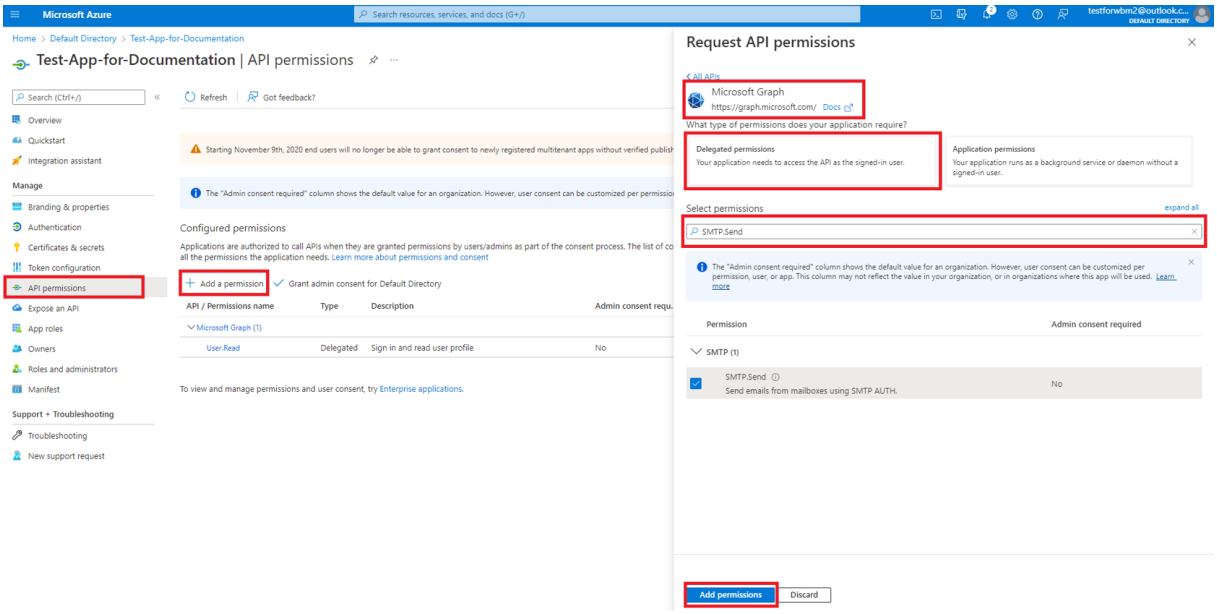
At the end of the process, Microsoft Entra ID will present some information about the newly registered application. At this point take note of the "**Application ID**" (also known as client ID) and "**Directory (Tenant) ID**", as that information will be used to double-check the second part of this configuration:



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.



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:



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

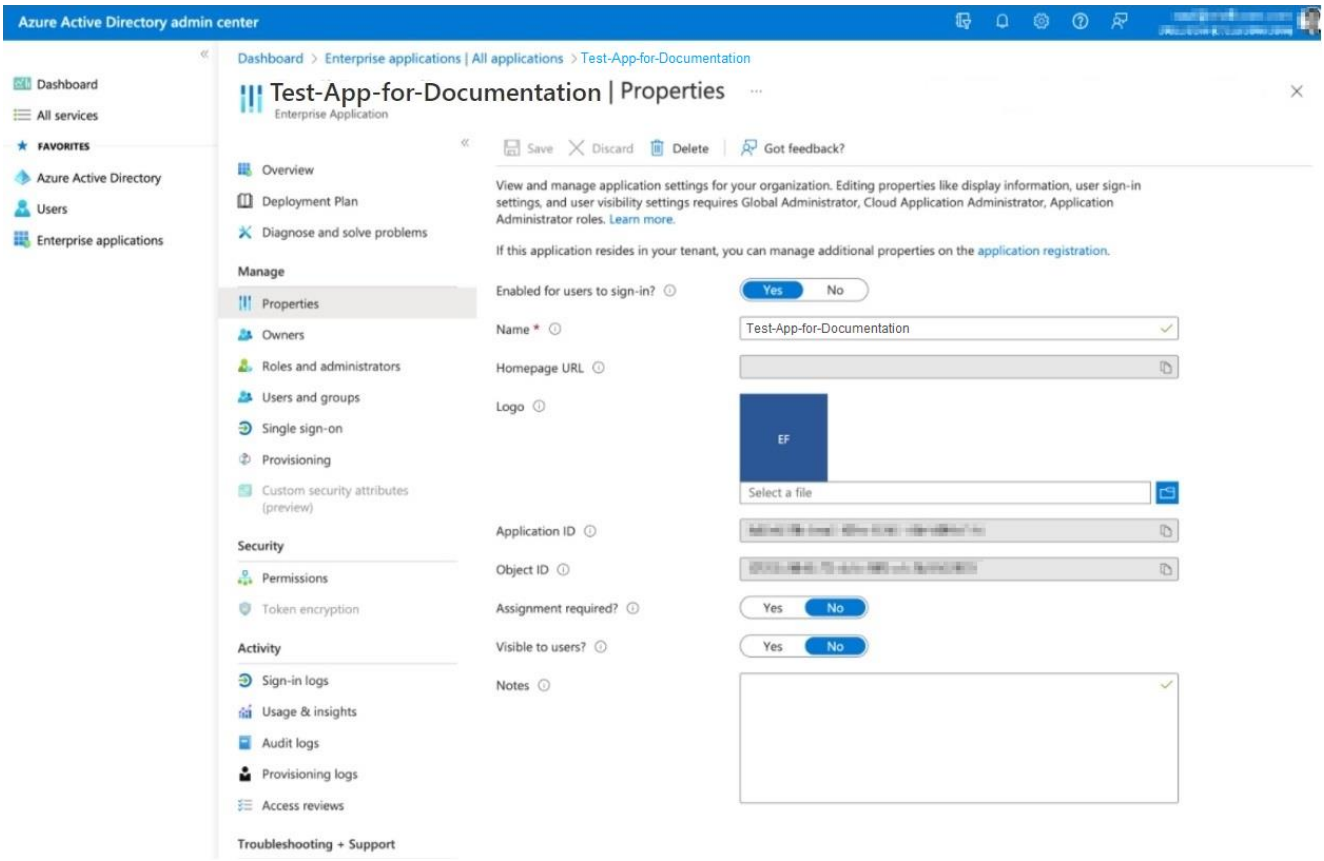
- SMTP.Send
- offline_access
- openid

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 as mentioned in Microsoft's Documentation:

	Microsoft Entra ID Directory, 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).



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:

`yourtenant.lab`

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

`APPID-123456`

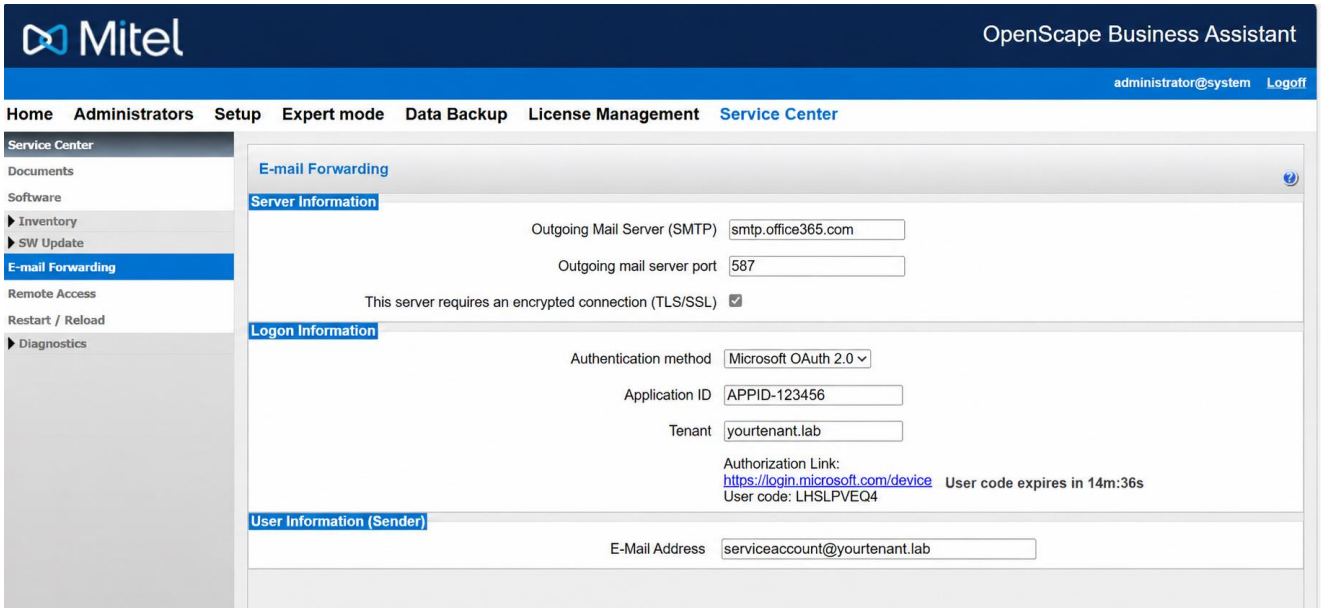
	Please notice that this are all example values and should be replaced with the actual details from the newly registered application above.

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

	Please note that SmtpClientAuthentication is disabled for the Tenant **`SmtpClientAuthenticationDisabled : False`** and follow https://aka.ms/smtp_auth_disabled.

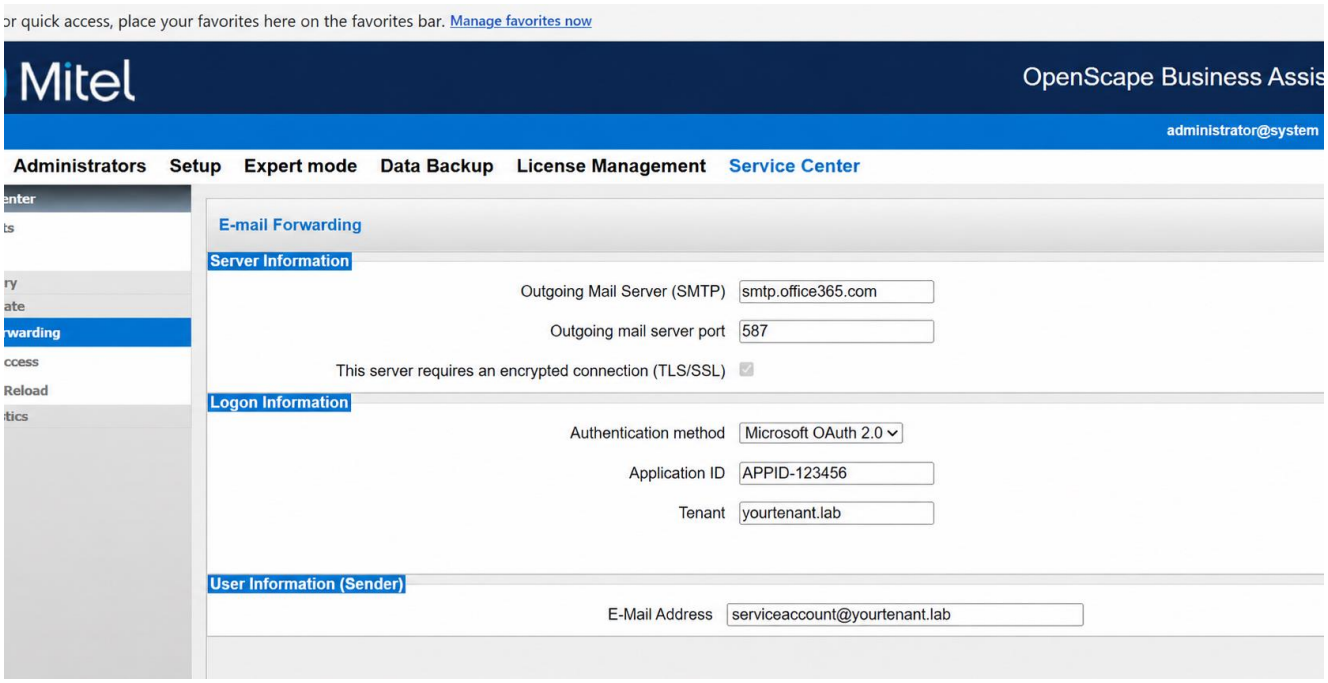
1.2 Create the E-mail Forwarding in WBM

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 Microsoft Entra ID configuration.

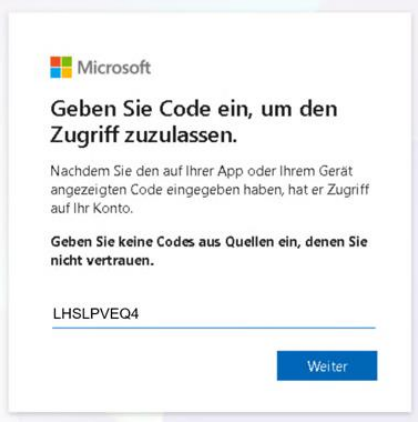


Click [**OK & Next]** to proceed with the authentication process.

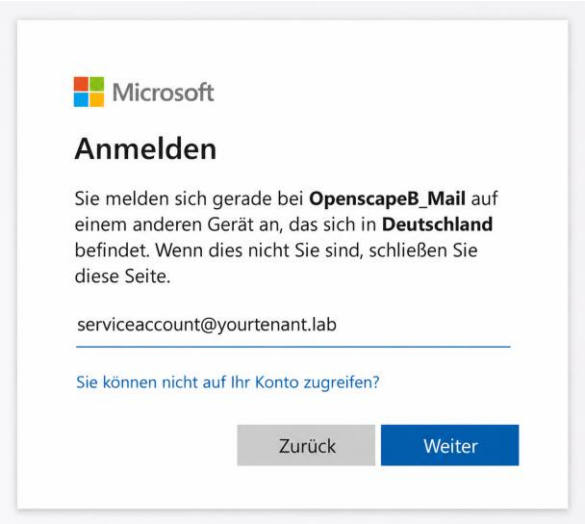
The OpenScape Business will send the authentication data to the Microsoft Graph and wait until the Authorization Link is provided:



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



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



Successful sign in is confirmed:

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

1.3 FAQ

"SASL authentication failed” “Authentication unsuccessful, SmtpClientAuthentication is disabled for the Tenant"	• please check for following articles 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 https://aka.ms/smtp_auth_disabled SmtpClientAuthenticationDisabled : False

2 Exchange Calendar and Directory integration

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

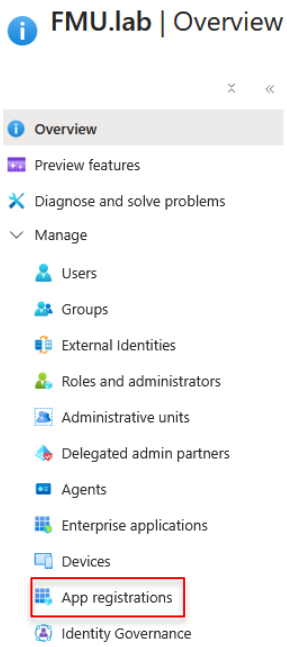
The configuration is split into two parts. The 1st part covers the registration of the application in the Microsoft Entra ID portal, and the 2nd part will show how to create the configure Calendar and Directory Integration in UC Suite Admin portal using the information from the 1st part.

2.1 Register the Application in Microsoft Entra ID

Using Microsoft OAuth 2.0 authentication requires a Microsoft Entra ID Application.

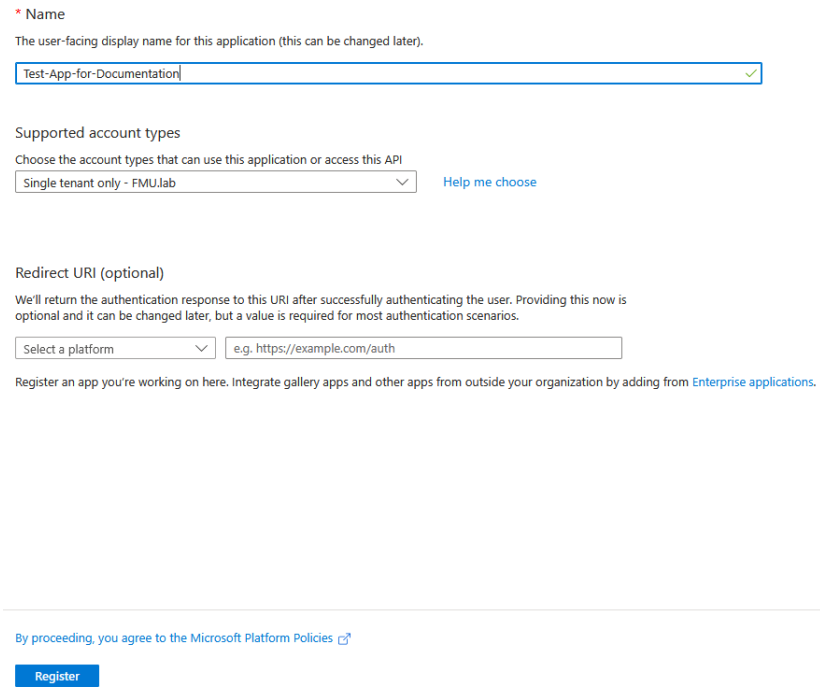
Open the browser and go to the Microsoft Entra ID Directory admin center and login with your Administrator account: https://portal.azure.com/ than go to Microsoft Entra ID:

and select "**App Registrations**":



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

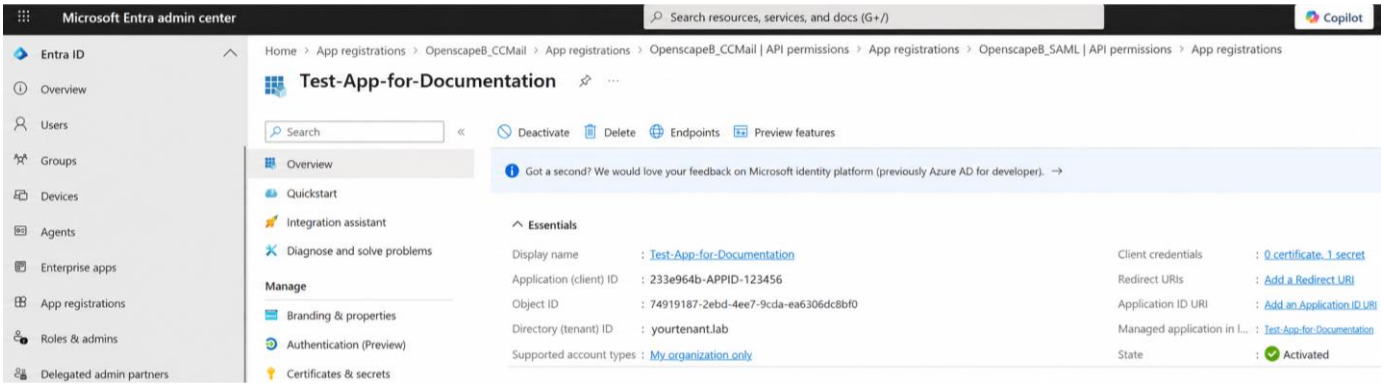
- Give it a Name
- On the "**Supported account types**" section, select whatever option suits your use case. This option restricts who can authenticate using this application.
  In the example below, the most permissive option is selected (depends on customer needs).
  Leave Redirect URI empty



At the end of the process, Microsoft Entra ID will present information about the newly registered application.

At this point take note of the "**Application ID**" (also known as client ID) and "**Directory (Tenant) ID**",

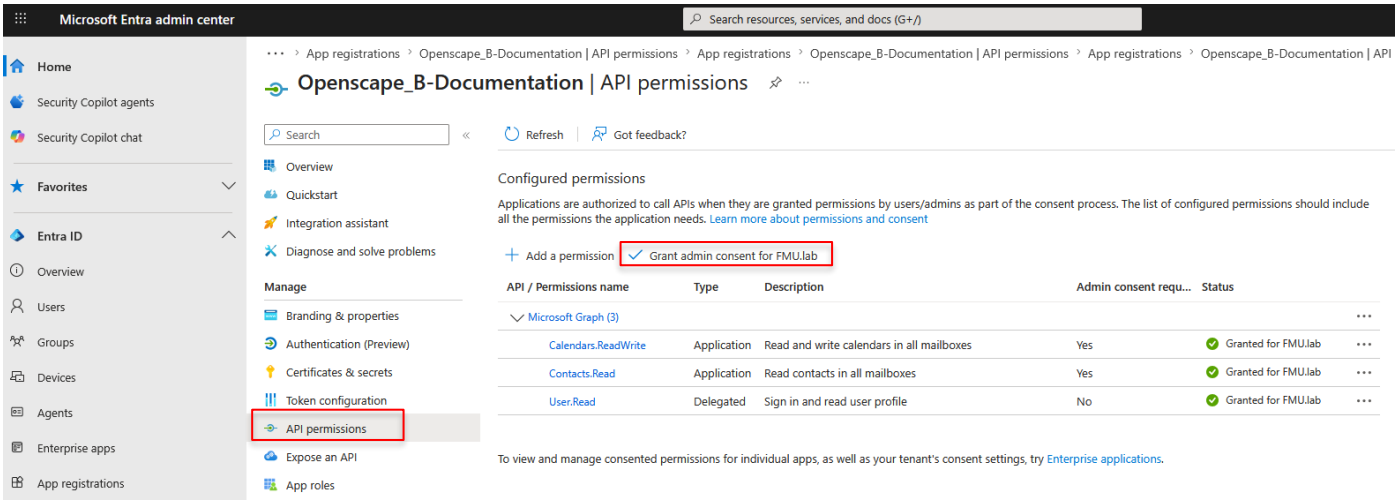
as that information will be used to double-check the second parts of this configuration:



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

The required permissions for OpenScape Business Calendar and Directory Integration are:

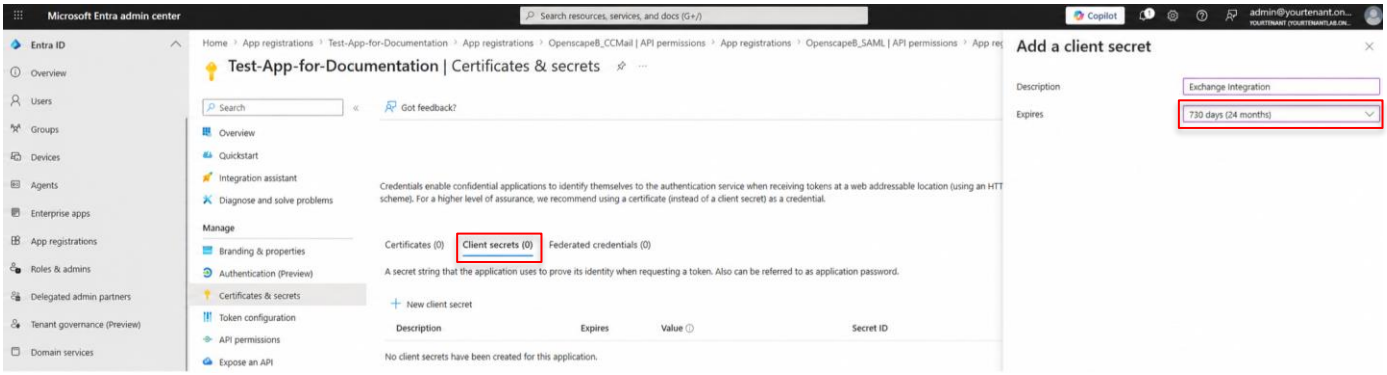
- Calendars.ReadWrite
- Contacts.Read



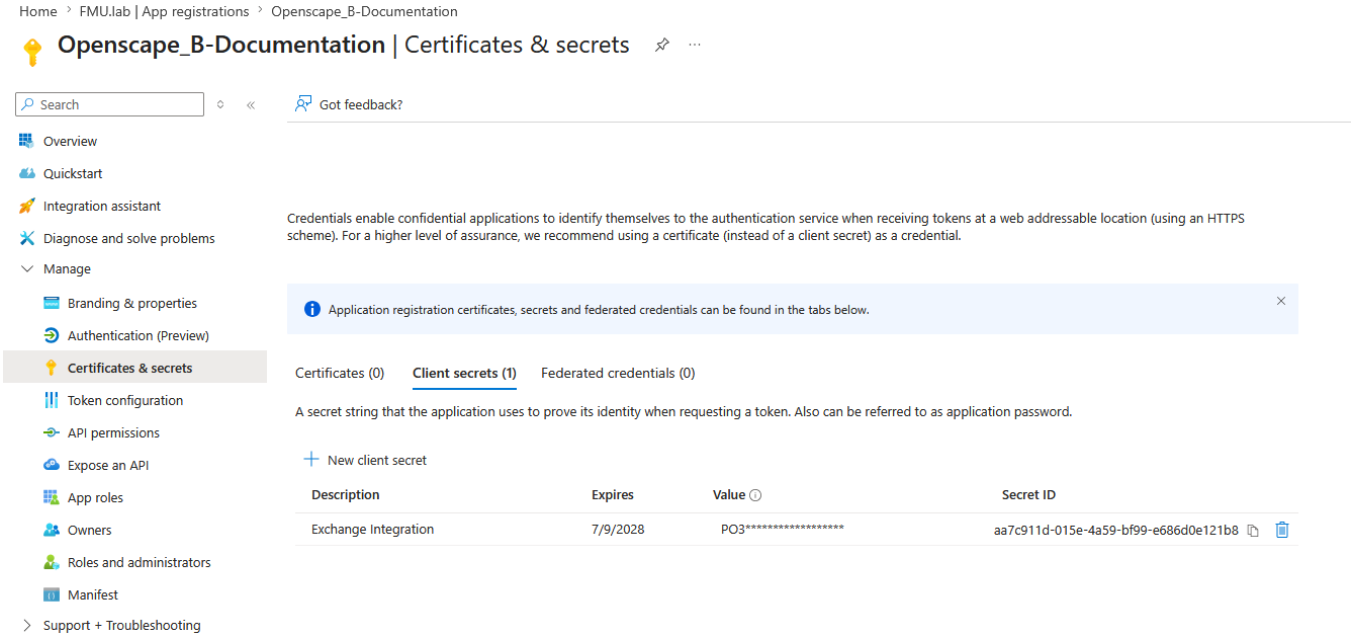
We now need to grant the admin consent permission for our tenant to use the API we selected above, that's done by clicking the "**Grant admin consent for <Tenant>**",

then clicking **Yes** in the dialog box that will appear.

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



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**":



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

Display name, which in our example is:

Openscape_B-Documentation

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

bb5dc110-yourtenant.lab

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

233e964b-AppId-123456

Client Secret Value, which in our example is:

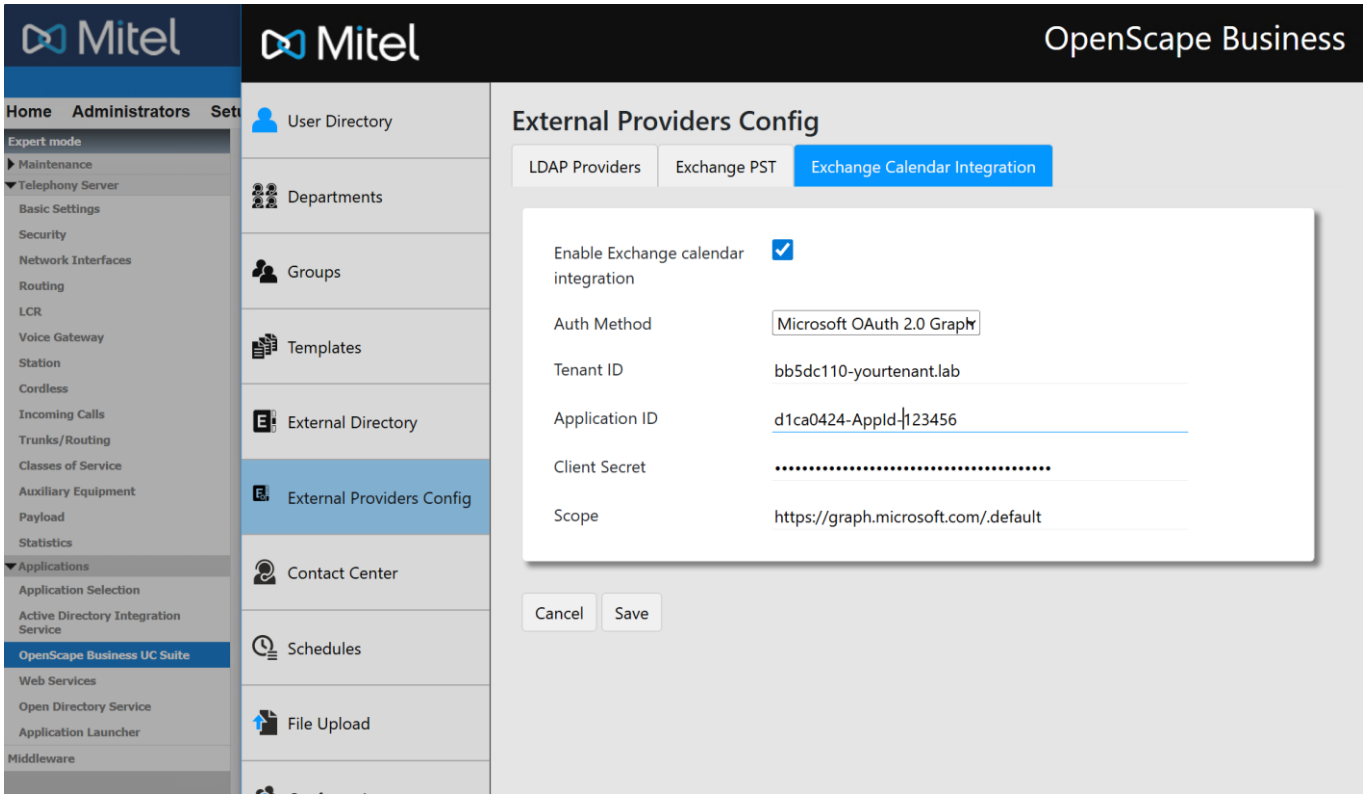
PO3****************

	Please note that these 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 Microsoft Entra ID, we can go to UC Suite Admin Portal, open "**External Providers Config**" and select the "**Exchange Calendar Integration**" tab.

Change the Auth Method to "**Microsoft OAuth 2.0 Graph"** 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:



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

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://graph.microsoft.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.

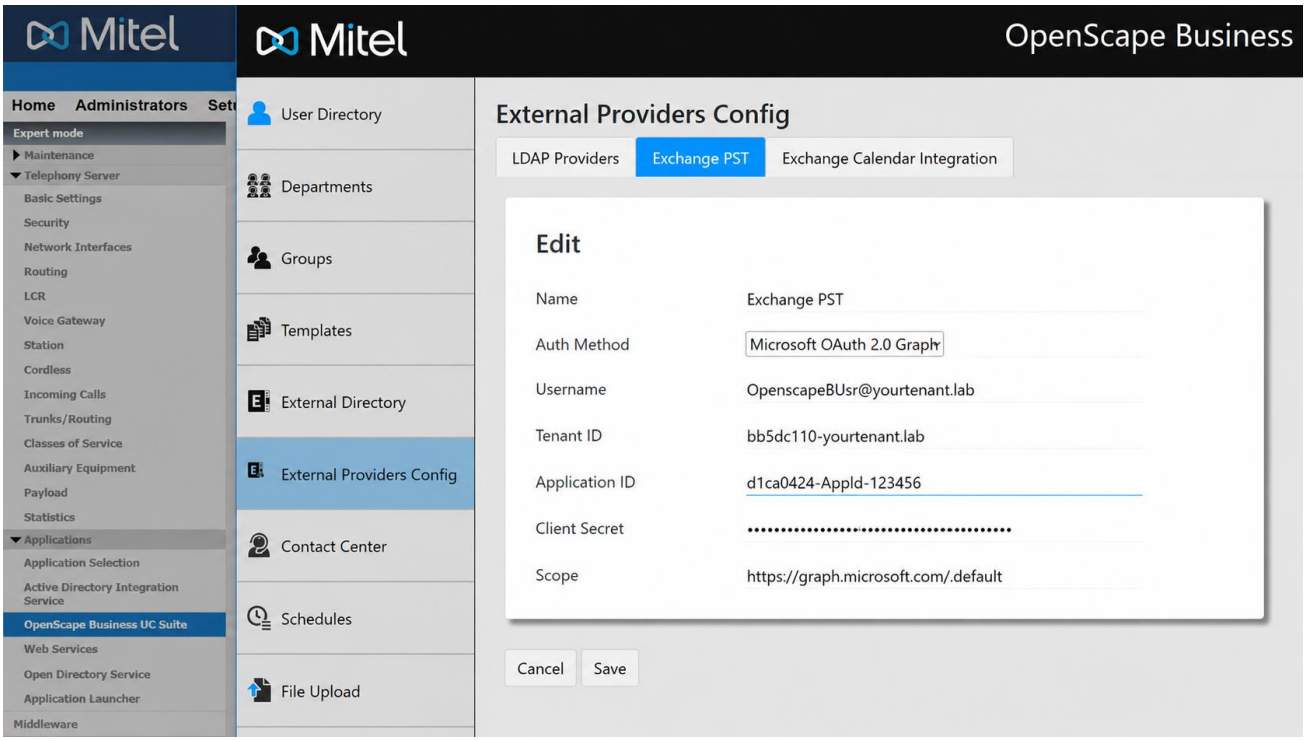
- Important note for testing: Appointments set into a future date and time with less than 3 minutes are ignored, because the system expects that the user creating an appointment can alter any option within 3 minutes. The OpenScape Business checks for appointments in the user’s calendar every 30 seconds. 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://outlook.office365.com/". 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

2.3 Exchange Directory Integration in UC Suite

Now that the application is registered with Microsoft Entra ID, we can go to UC Suite Admin Portal, open "**External Providers Config**" and select the "**Exchange PST**" tab.

Change the Auth Method to "**Microsoft OAuth 2.0 Graph**" and the following fields will appear. Use a valid email address for this application that is registered through the Microsoft Entra ID configuration within the Username field:



Tenant ID, Username, 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://graph.microsoft.com/.default

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

	Public folder contacts search (notes and hints): • Access to Microsoft Exchange Public Folders – the programmatic access to public folders is restricted to supported Outlook clients only, and for bulk import/export purposes by Microsoft. Microsoft will not provide APIs for programmatically creating, reading, updating and deleting public folders after October 2026 (https://techcommunity.microsoft.com/blog/exchange/retirement-of-exchange-web-services-in-exchange-online/3924440).

3 E-mail Queues with OAuth 2.0

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

The configuration will be split into three parts. The 1st part covers the registration of the application in the Microsoft Entra ID Directory. 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 UC Suite Admin Portal using the information from the previous parts.

3.1 Register the Application in Microsoft Entra ID Directory

Open the browser and go to **Microsoft Entra admin center**.

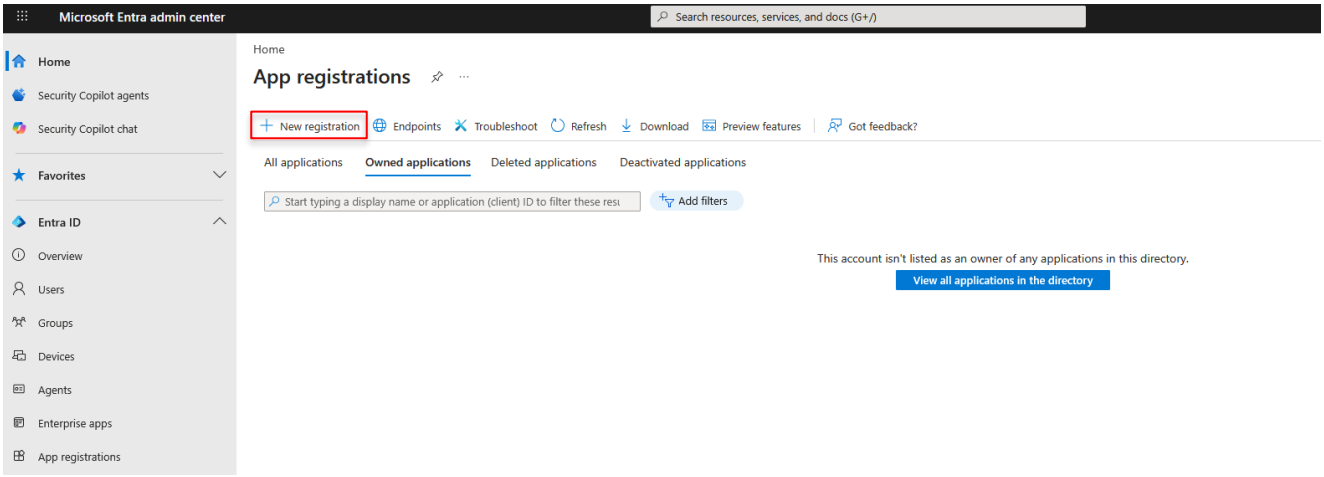
Select "**App Registrations**" in the left menu. Then proceed with the app registration as per the previous instructions for Azure Active Directory. Start by clicking "**New registration**".

Give it a name and click Register. Take note of the "**Application (client) ID**" and "**Directory (tenant) ID**", as they will be required later.

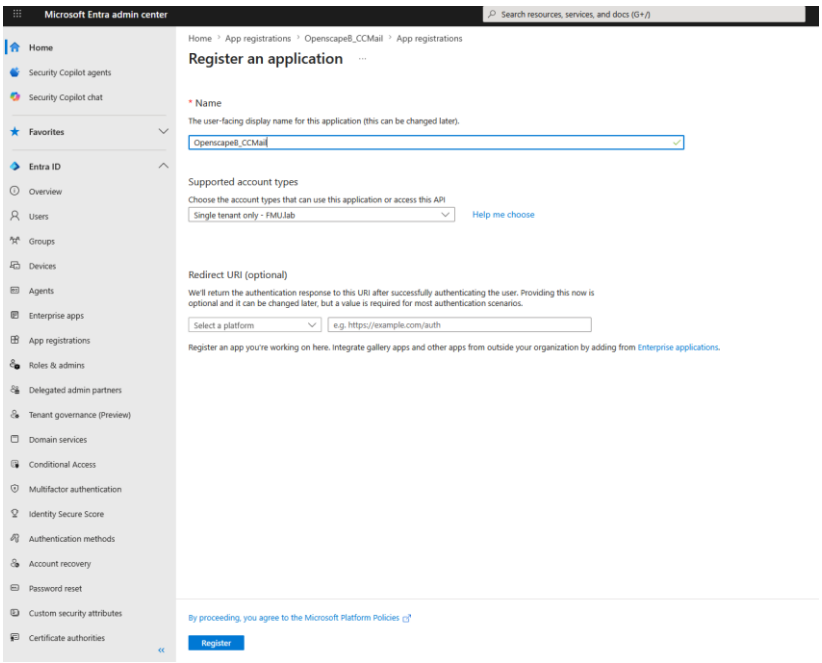
Goto the Microsoft Entra ID Directory admin center and login with your Administrator account: https://aad.portal.azure.com/.

The instructions assume that an account 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 " **Microsoft Entra ID** " in the menu on the top left corner, click on "**App registrations**", and then click "**New registration**":

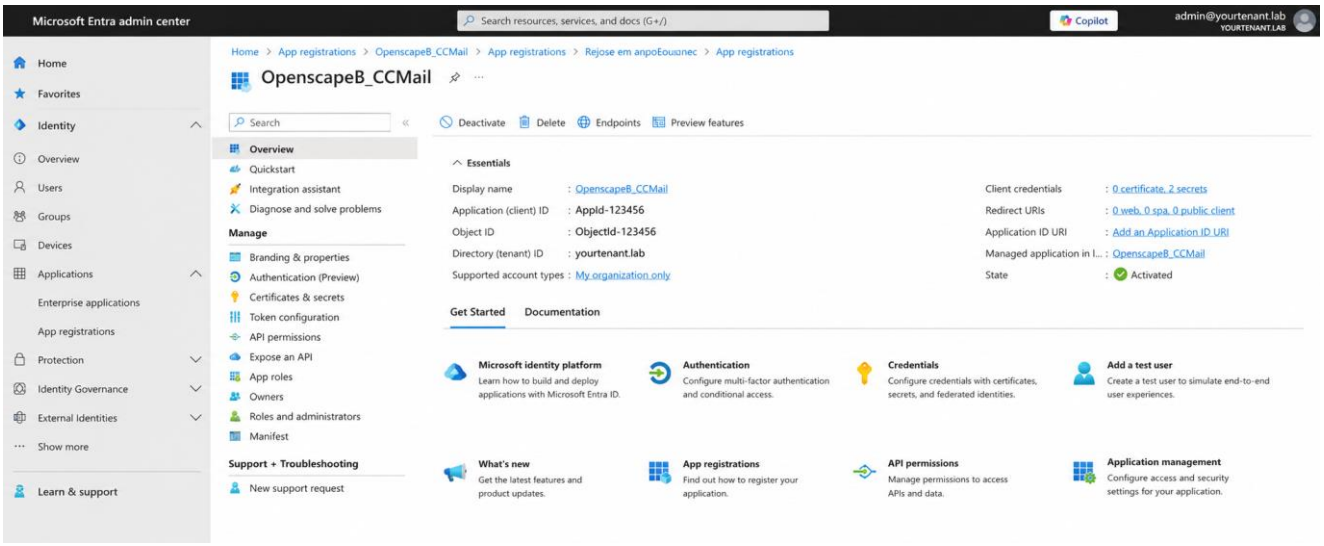


Enter a meaningful name for the application (this will be later used to query the application when creating the Service Principal for Exchange Online), and for the "**Supported account type**", select the option which fits to customer installation (example: **Single tenant**). Then click the **Register**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



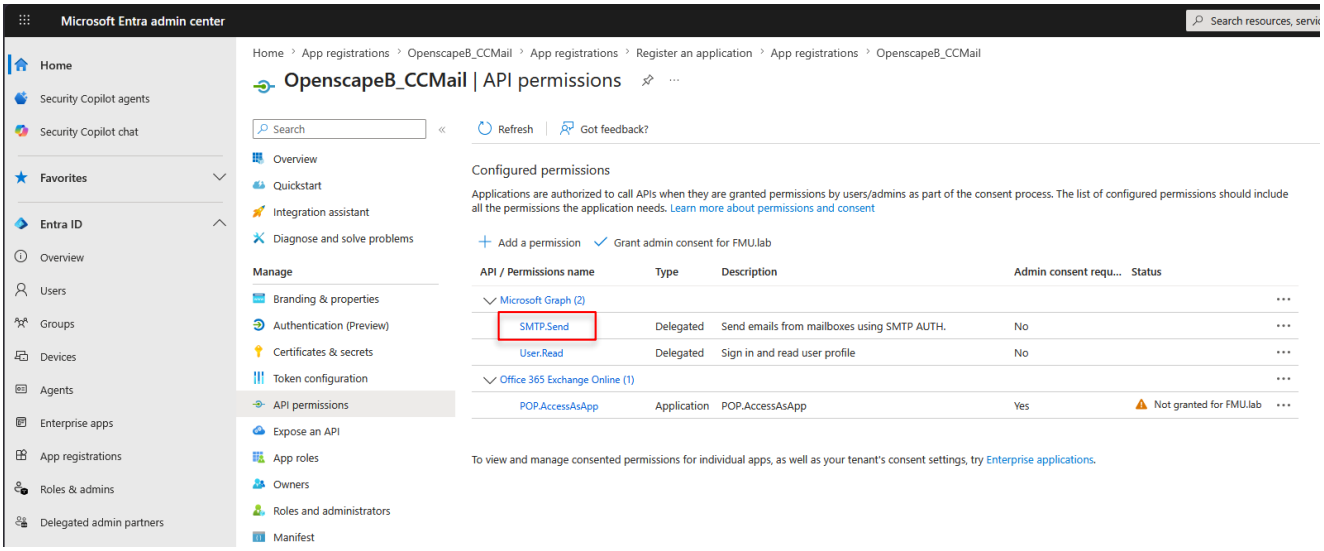
	The same application can be used for all services, if the respective API permissions are assigned and granted, as well as the PowerShell commands (in the case of Email Queues only).

At the end of the process, Microsoft Entra ID 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:

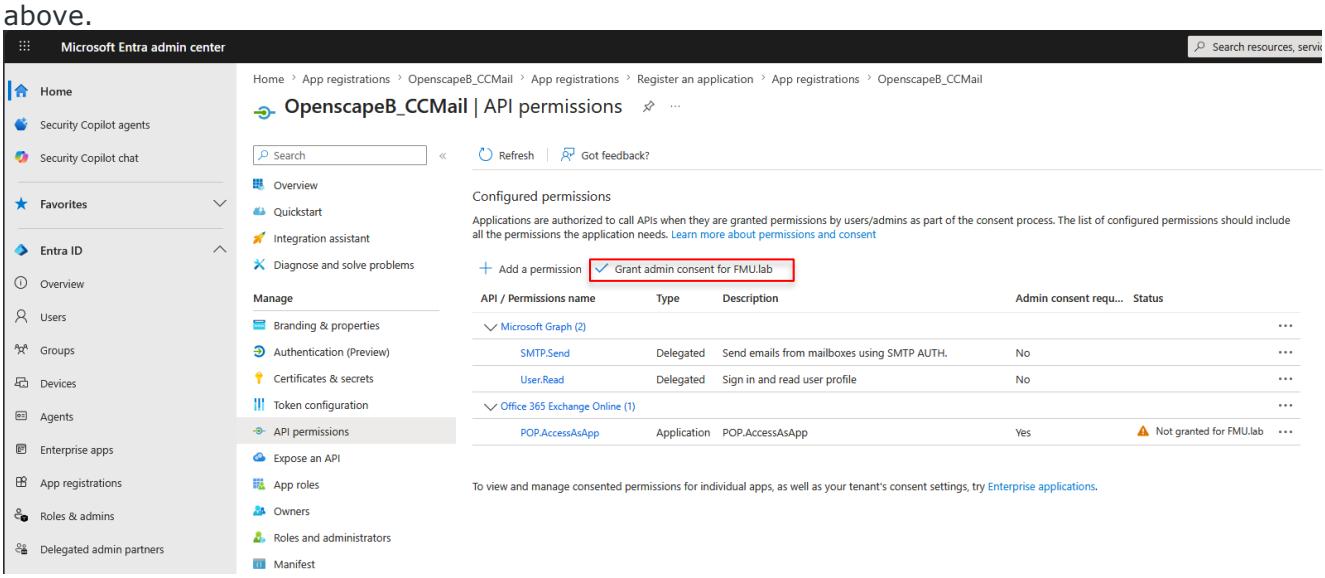


Proceed by giving the proper "**API permissions**" to the application. In the following screen, click "**Add a permission**", select the tab "**APIs my organization uses**", type "**Office 365 Exchange Online**" and select it "**Application permissions**". Scroll down to "**POP**", expand it and select "**POP.AccessAsApp**", then click "**Add permissions**".

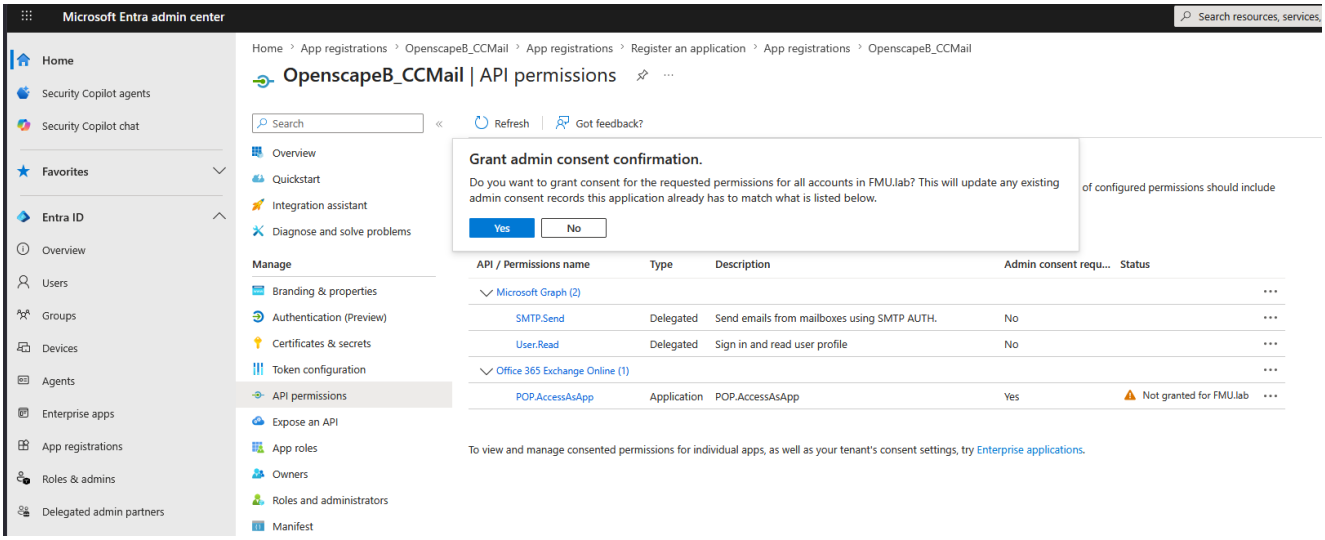
In addition, the **SMTP.Send** permission could be set as well like in OpenScape Business E-mail forwarding (for usage with **Inbound Email Queue Options/Return Email Address).**



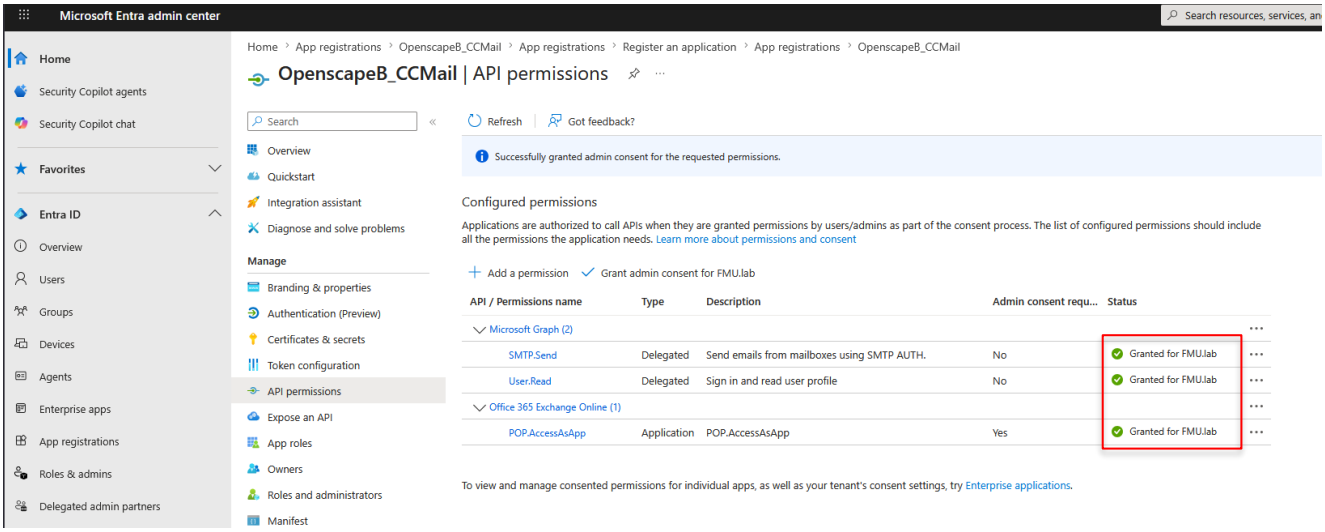
We now need to grant the admin consent permission for our tenant to use the API we selected above.



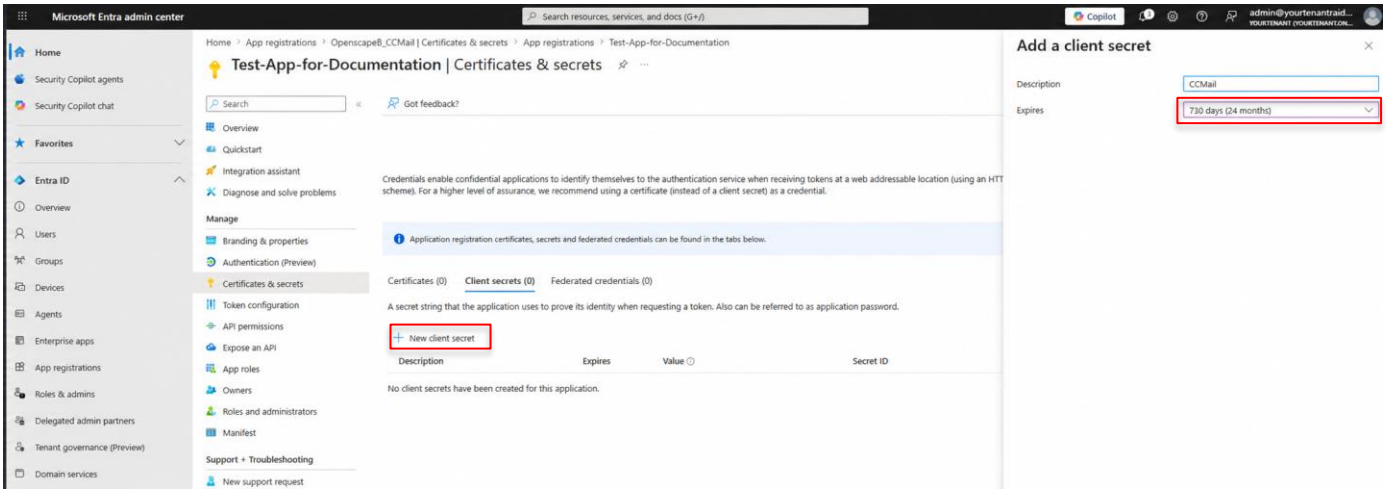
That’s done by clicking the “**Grant admin consent for <Tenant>**”, then clicking **Yes** in the dialog box that will appear:



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

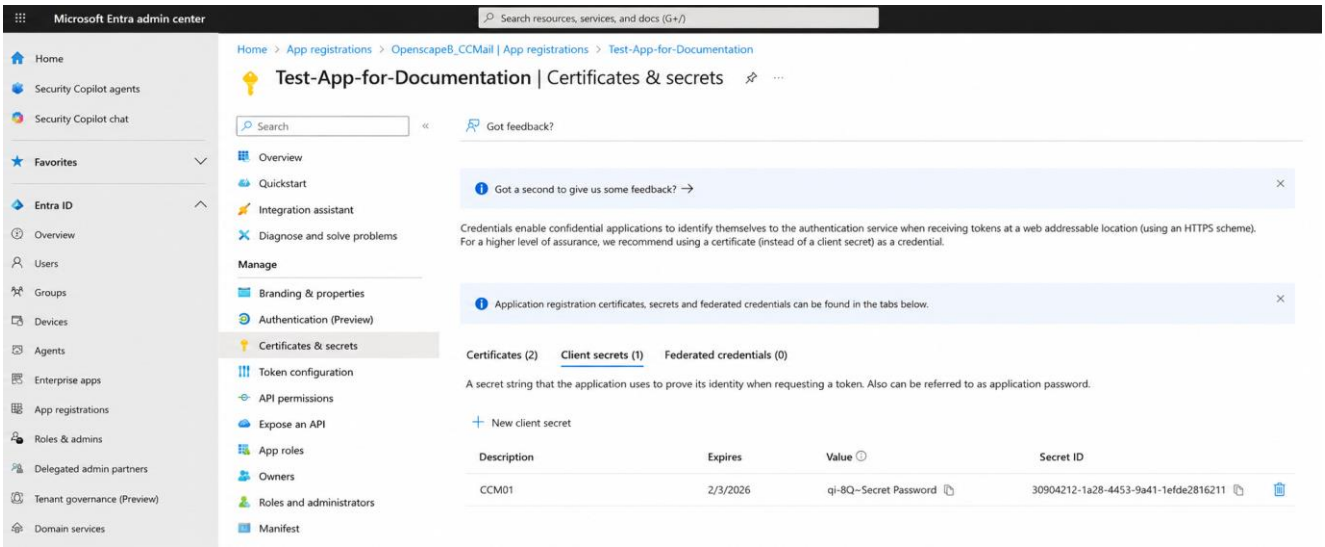


Now we need to create a client secret for the application. Select the "**Client & secrets**" menu and then click "**New client secret**":



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**":

	The maximum running time 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 is the only time the Value will be ever visible and available for copying. If you don't do that at this point, the Client Secret Value cannot be used later when it has to be entered on OpenScape Business Admin Portal.

Display name, which in our example is:

OpenscapeB_CCMail

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

bb5dc110-yourtenant.lab

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

dea68964-AppId-123456

Client Secret Value, which in our example is:

****

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

3.2 Service Principal and Mailbox permissions

Now we need to use **Windows PowerShell** to connect to the Microsoft Entra ID 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.

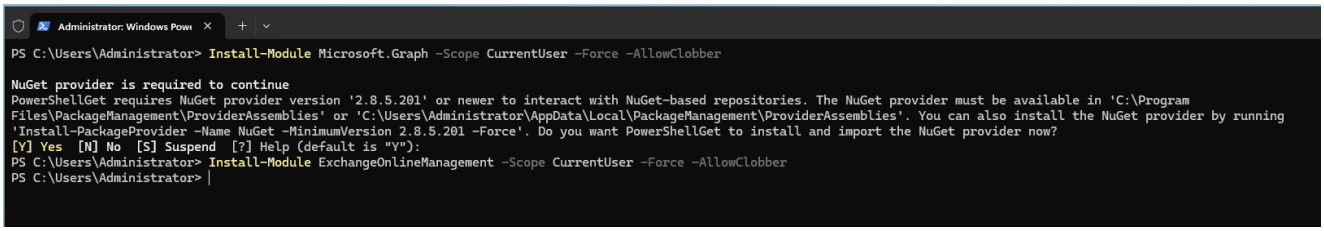
Additionally a Microsoft Entra ID Account with administrator permissions is mandatory.

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

```
Install-Module Microsoft.Graph -Scope CurrentUser -Force -AllowClobber
```

```
Install-Module ExchangeOnlineManagement -Scope CurrentUser -Force -AllowClobber
```

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

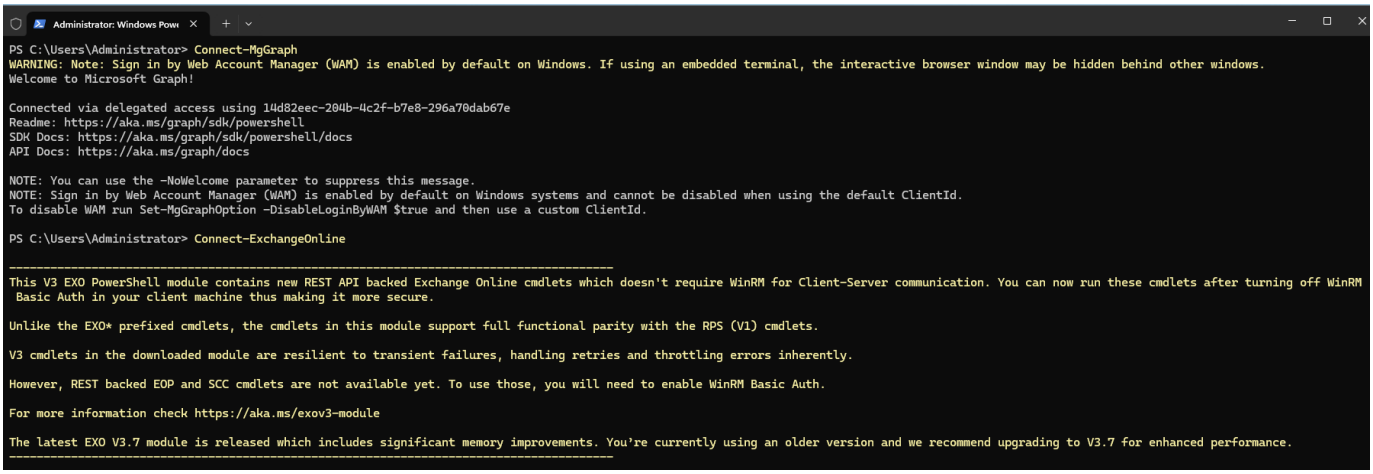


Next, connect Microsoft Graph following any requested prompts to authenticate your account:

Connect-MgGraph

Then connect to Exchange Online and also follow the required steps to authenticate with the respective Tenant:

Connect-ExchangeOnline

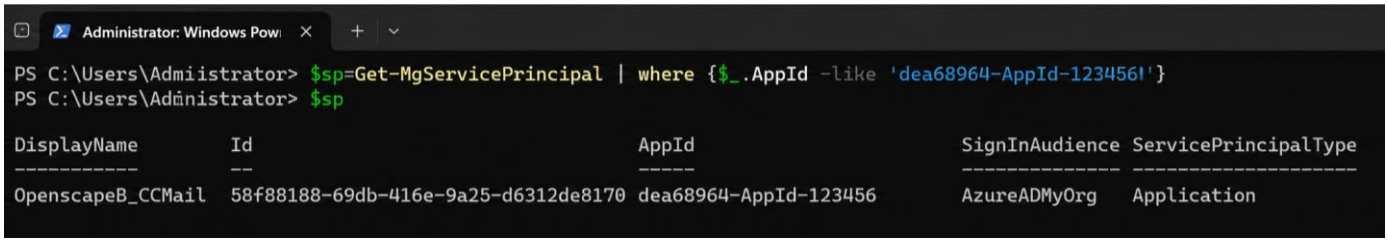


Now create a variable to assign the object for the Application so it can be used to create the Service Principal, using the Application (client)

ID copied from the first step (in our example it's 'dea68964-AppId-123456', you should add your own Application ID):

```
$sp=Get-MgServicePrincipal | where {$_.AppId -like 'dea68964-AppId-123456'}

$sp
```

Notice that we enter `$sp` in the PowerShell console just to make sure the correct application is assigned to the variable.

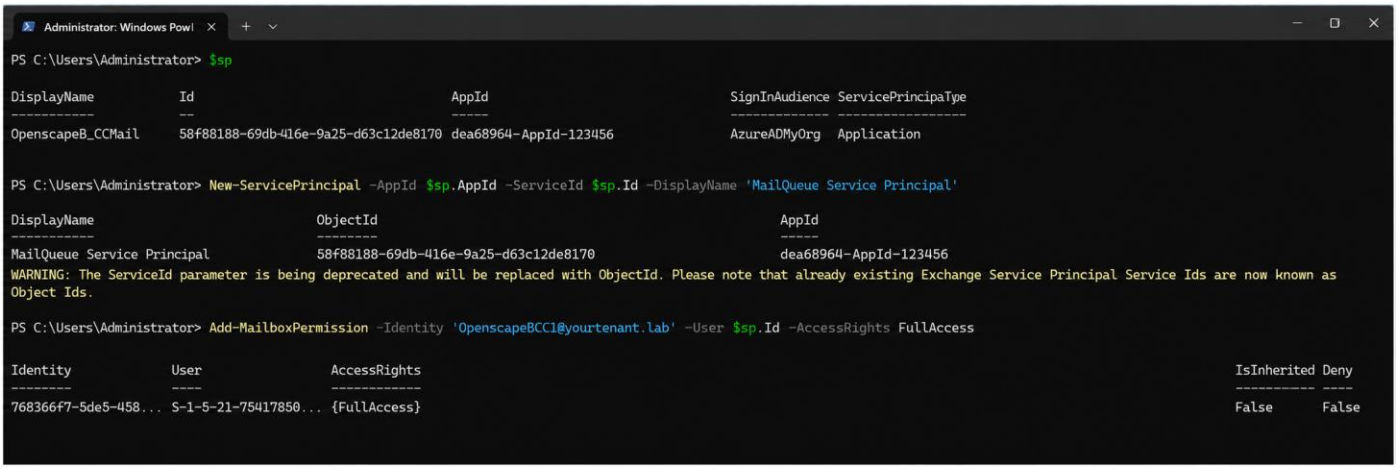


Now we create the Service Principal using the Service ID and Application ID of the application:

```
New-ServicePrincipal -AppId $sp.AppId -ServiceId $sp.Id -DisplayName
'OpenscapeB_CCMail'
```

Finally, we add the access rights to the mailbox that will be using the service (the actual email account, in our example it's test. 'OpenscapeBCC1@yourtenant.lab)

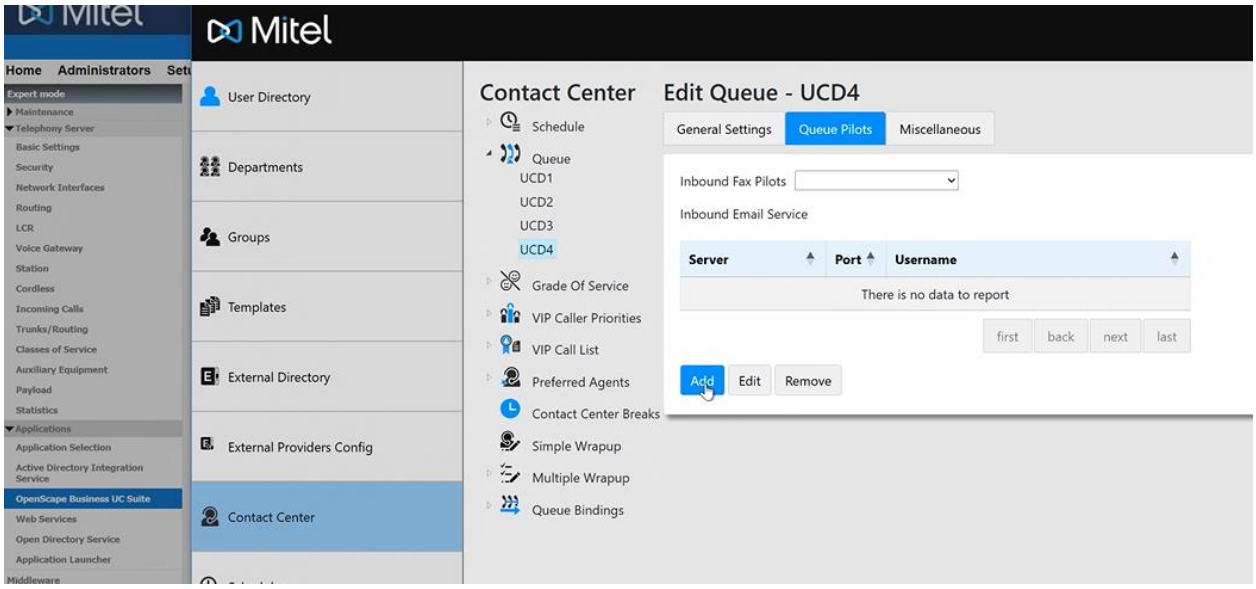
Add-MailboxPermission -Identity 'OpenscapeBCC1@yourtenant.lab' -User $sp.Id -AccessRights FullAccess



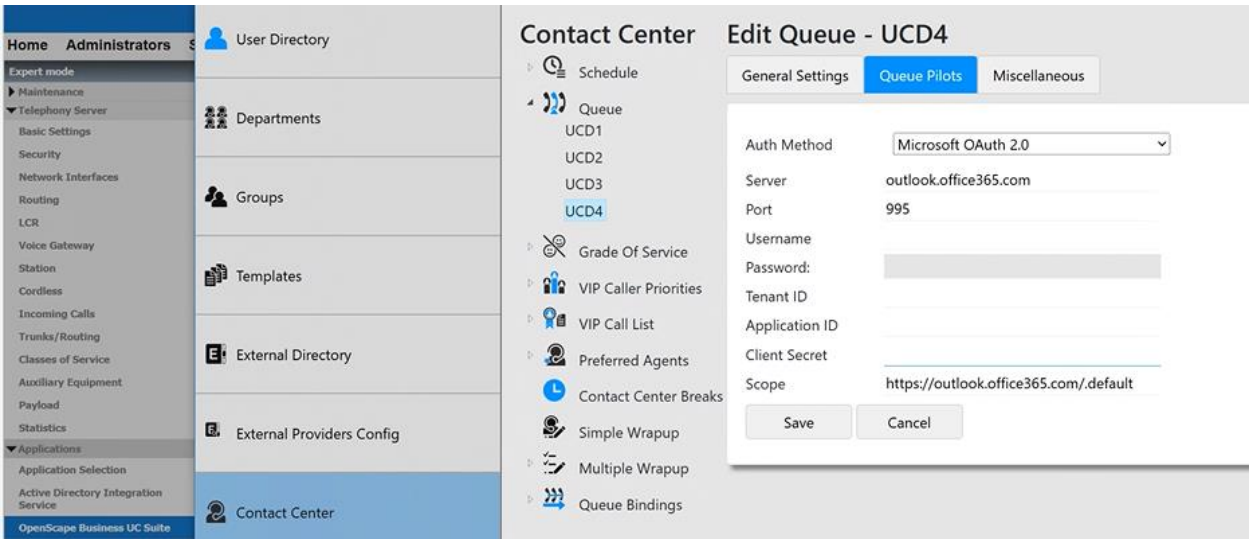
3.3 Create the E-mail for the Contact Centre Queue

Now that the application is registered with Microsoft Entra ID 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 UC Suite Admin Portal, 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 "Microsoft **OAuth 2.0**" and the following fields will appear:

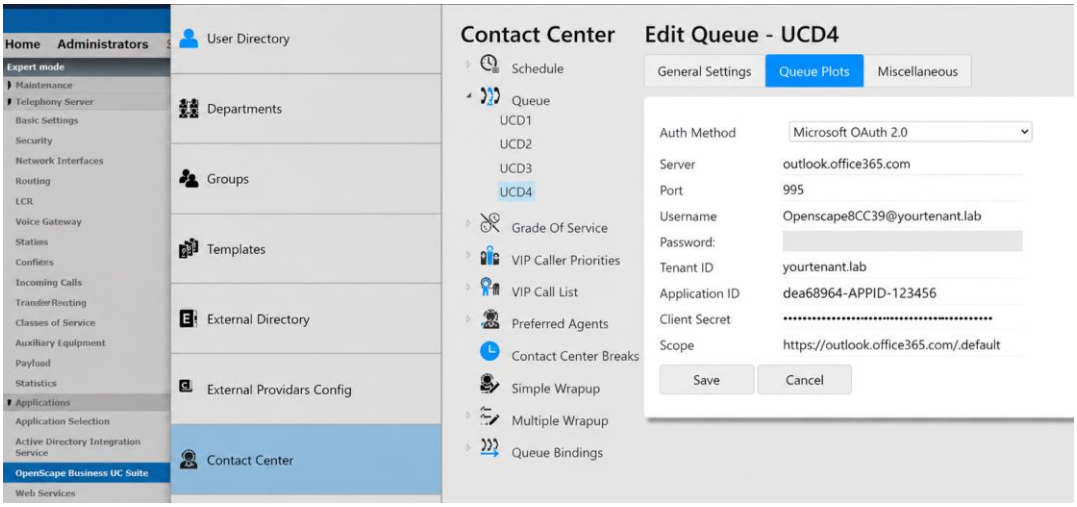


All the default values can be left as they are, but you will have to enter the following fields:

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

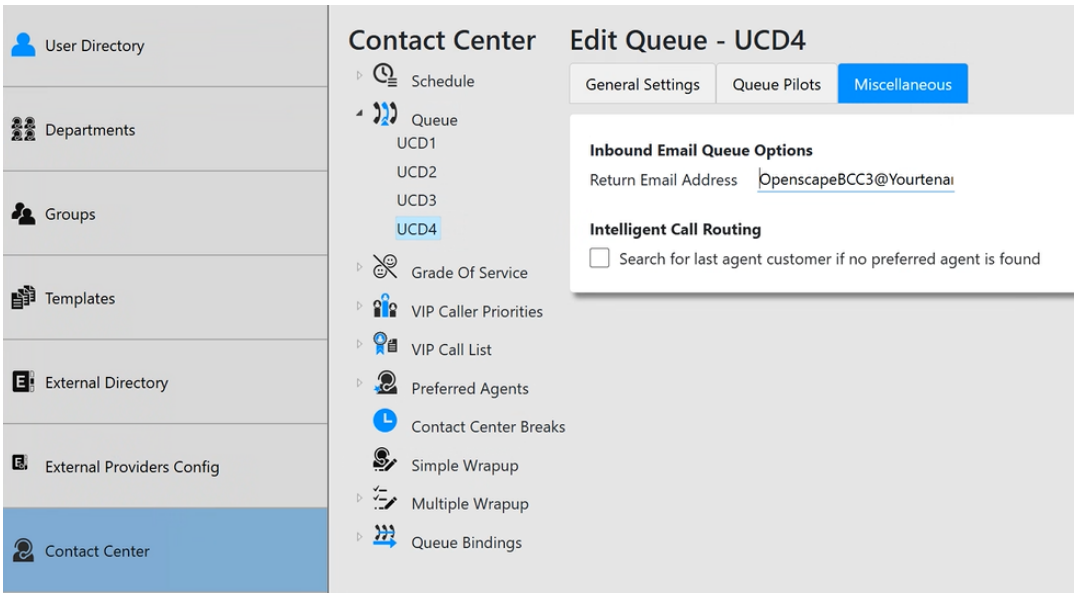
Tenant ID, Application ID, Client Secret value: these are the same values that you took note of from the last step in the 1st part of the configuration

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



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

In case the feature 'Return Email Address' should be used, it's necessary provide 'Mailbox Delegation-Send As' permission to the account defined in'E-Mail Forwarding'.



4 myReports with OAuth 2.0

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

5 Support & Serviceability

5.1 Required trace files for error analysis

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

6 Best Practice

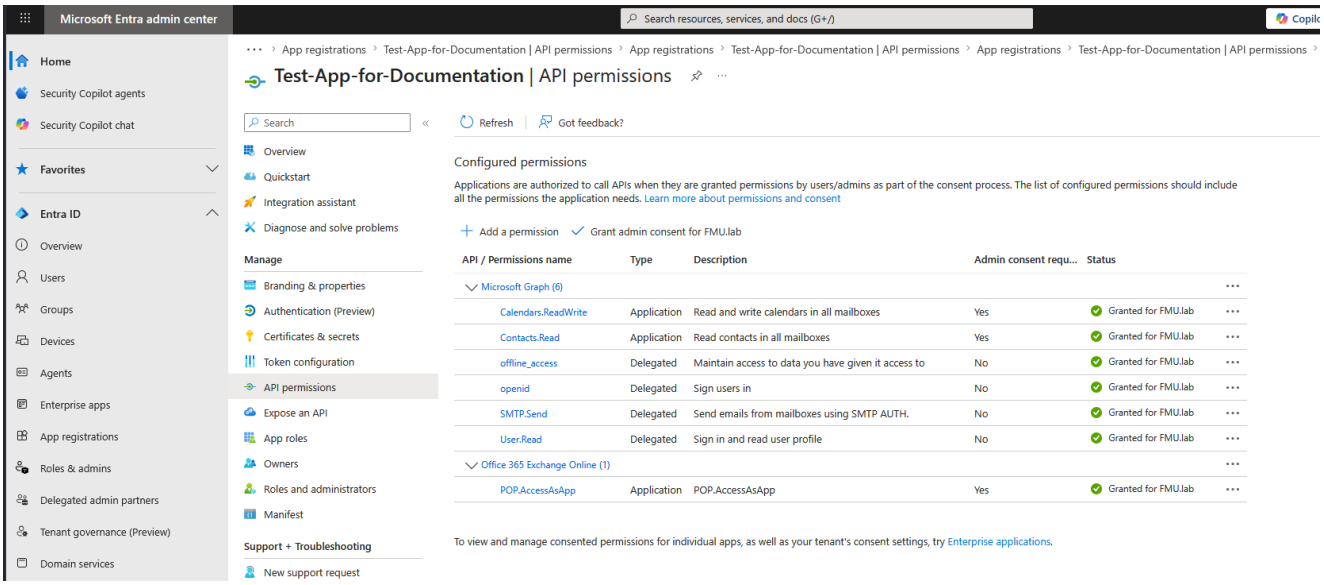
Information and useful hints from our lab and customer installations.

6.1 Use the same application for all functions

The "**Test-App-for-Documentation**" 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. Partial usage is possible as well:



6.2 Testing POP3 access

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

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

test_pop.sh

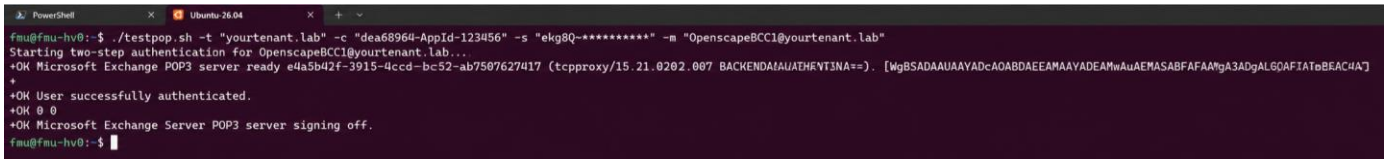
```
#!/bin/bash

# --- Function to show usage ---
usage() {
    echo "Usage: $0 -t <tenant_id> -c <client_id> -s <client_secret> -m <mailbox_email>"
    exit 1
}

# --- Parse Command Line Arguments ---
while getopts "t:c:s:m:" opt; do
    case $opt in
        t) TENANT_ID="$OPTARG" ;;
        c) CLIENT_ID="$OPTARG" ;;
        s) CLIENT_SECRET="$OPTARG" ;;
        m) MAILBOX="$OPTARG" ;;
        *) usage ;;
    esac
done

if [ -z "$TENANT_ID" ] || [ -z "$CLIENT_ID" ] || [ -z "$CLIENT_SECRET" ] || [ -z "$MAILBOX" ]; then
    usage
fi

# 1. Get OAuth Access Token
TOKEN_RESPONSE=$(curl -s -X POST "https://login.microsoftonline.com/$TENANT_ID/oauth2/v2.0/token" \
    -H "Content-Type: application/x-www-form-urlencoded" \
    --data-urlencode "client_id=$CLIENT_ID" \
    --data-urlencode "scope=https://outlook.office365.com/.default" \
    --data-urlencode "client_secret=$CLIENT_SECRET" \
```

```
    --data-urlencode "grant_type=client_credentials")

ACCESS_TOKEN=$(echo "$TOKEN_RESPONSE" | jq -r '.access_token')

if [ "$ACCESS_TOKEN" == "null" ] || [ -z "$ACCESS_TOKEN" ]; then
    echo "Error: Token retrieval failed."
    exit 1
fi

# 2. Construct the SASL XOAUTH2 string
# \x01 is the Null separator required by Microsoft
AUTH_STRING=$(printf "user=%s\x01auth=Bearer %s\x01\x01" "$MAILBOX"
"$ACCESS_TOKEN" | base64 -w 0)

# 3. Two-Step Authentication Flow
echo "Starting two-step authentication for $MAILBOX..."
{
    sleep 2
    echo "AUTH XOAUTH2"  # Step 1: Tell server we want to use XOAUTH2
    sleep 2
    echo "$AUTH_STRING"  # Step 2: Send the actual token
    sleep 2
    echo "STAT"          # Step 3: Check mailbox status
    sleep 1
    echo "QUIT"
} | openssl s_client -connect outlook.office365.com:995 -crlf -quiet 2>/dev/null
```

Usage:

	It might be necessary to save the script with EOL Conversion(Linux) by an appropriate editor.

```
./testpop.sh -t <tenant_id> -c <client_id> -s <client_secret> -m <mail address >
```



6.3 Further

	Please send us important hints of your customer installation which are missing in the general description and worth adding here and sharing with the community. Thank you !!!