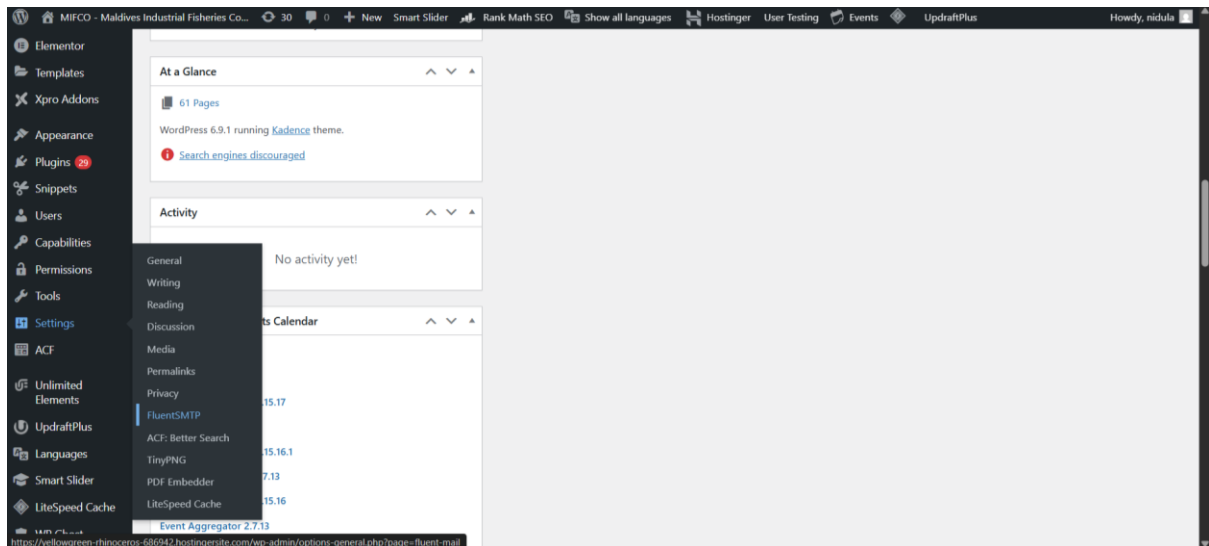


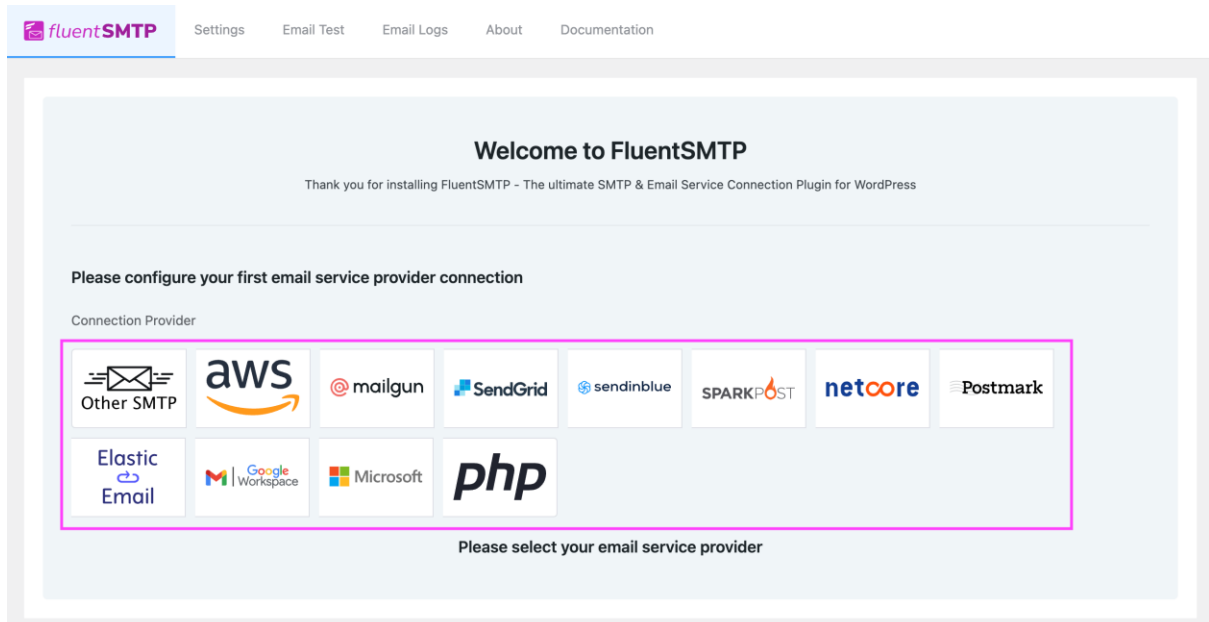
## Configuring FluentSMTP with Outlook or Office365 Emails

Please follow the below sections to configure Outlook or Office365 Emails with FluentSMTP.

Please go to **WordPress Settings → FluentSMTP**



Now from your **WordPress Dashboard** go to **Settings → FluentSMTP**

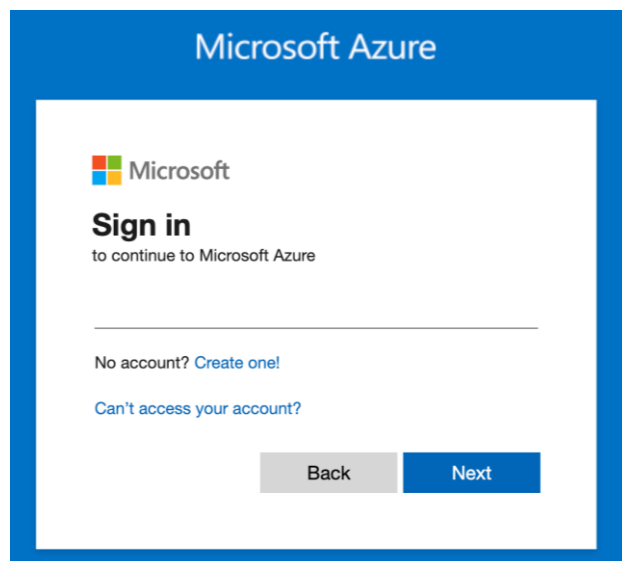


After selecting Microsoft as the mailer, a very simple settings window will open as shown in the screenshot below.

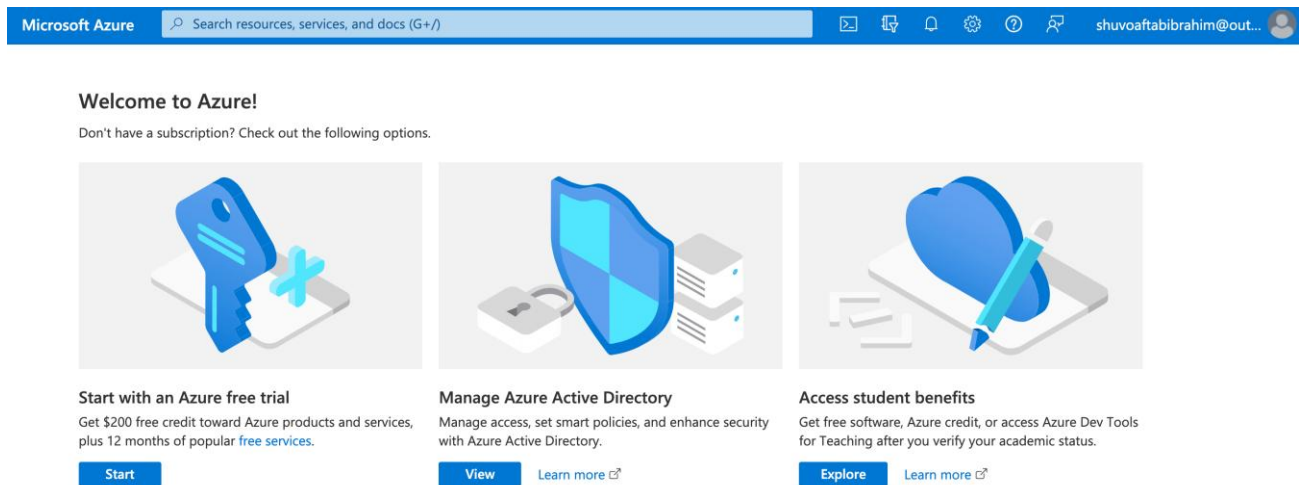
The screenshot shows the 'Add Connection' settings window in fluentSMTP. At the top, there is a navigation bar with links for 'Settings', 'Email Test', 'Email Logs', 'About', and 'Documentation'. The main content area is titled 'Add Connection' and features a 'Connection Provider' section where 'Microsoft' is selected. Below this, there are 'Sender Settings' and 'Outlook/Office365 API Settings' sections. The 'Sender Settings' section includes fields for 'From Email' and 'From Name', with checkboxes for 'Set the return-path to match the From Email' and 'Force Sender Name'. The 'Outlook/Office365 API Settings' section includes a note to check documentation for API keys, buttons for 'Store Application Keys In DB' and 'Application Keys in Config File', and fields for 'Application Client ID' and 'Application Client Secret'. A 'App Callback URL' field is also present with a pre-filled URL. A red button labeled 'Authenticate with Office365 & Get Access Token' is located below the API settings. At the bottom, there is a green 'Save Connection Settings' button and a red warning message: 'Outlook/Office365 is not recommended for sending mass marketing emails.'

The Application Client ID and Client Secrets are parts of the Microsoft Azure Cloud Application. We need to:

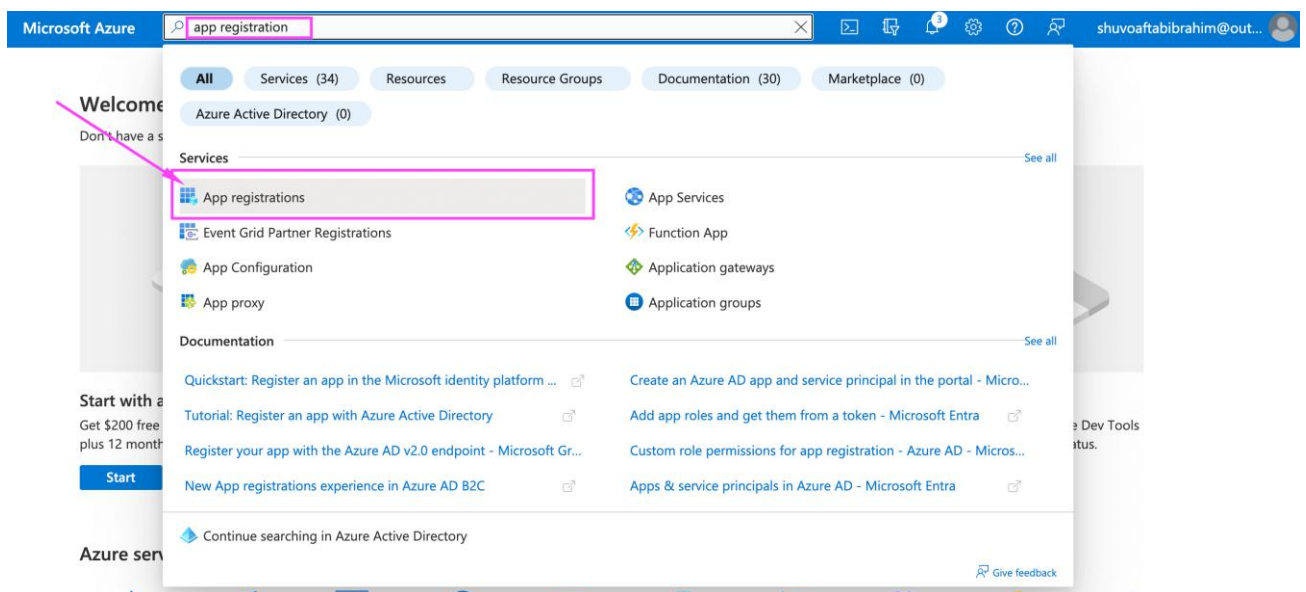
Please go to Microsoft Azure Portal from Here: [Home – Microsoft Azure](#)



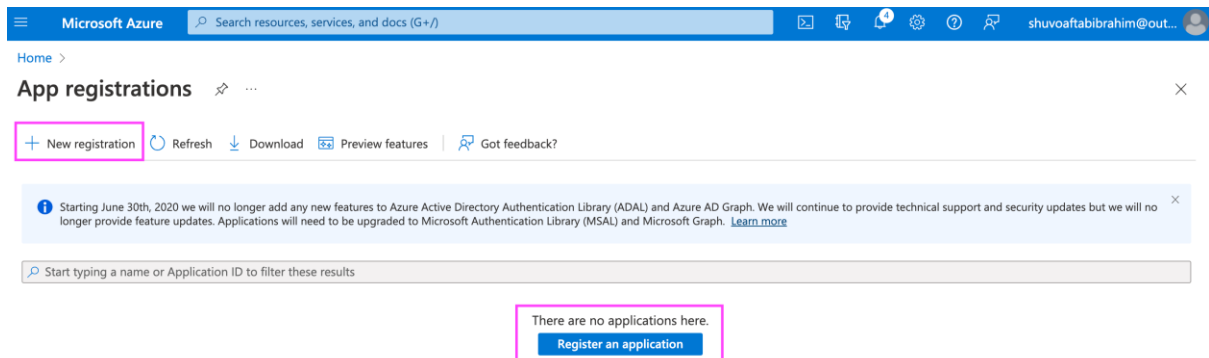
You may be asked to log in to your account using your Outlook or Office365 Account Email as the above screenshot. Once you are logged in you will be redirected to the Azure Portal Home as the below screenshot.



Now from the top search bar, Search for “**app registration**” and it will automatically suggest a few services and look for the **App Registrations** as below. Then click on this to go to the **App Registration Dashboard**.




You can also directly go to New APP Registration from here: [Register an application – Microsoft Azure](#) and you will be redirected to the below dashboard once you are logged in as per the previous step.



Now, if you do not have any existing Apps Registered then you will see the same dashboard as the above screenshot and can click on the **Register an application** button from the center.

If you already have an existing App registered, then you will have to click on the **+ New Registration** button on the top section to **Register an Azure Application**.

## Register an application ...

 This application will not be associated with any directory and will be subject to limitations. You should not create production apps outside of a directory.

### \* Name

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

### Supported account types

Who can use this application or access this API?

- Accounts in 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.

Web

By proceeding, you agree to the [Microsoft Platform Policies](#)

Register

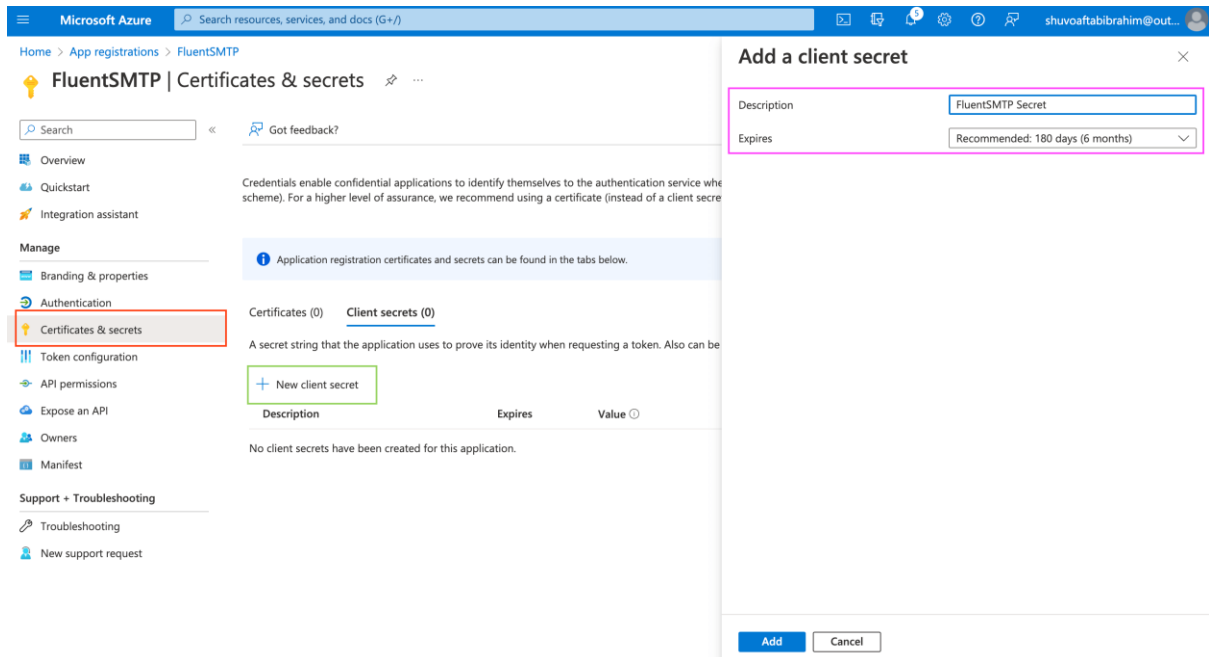
1. **Name:** In the next section, Provide a Name for the Application as an Identifier.
2. **Supported Application Type:** This must be “Accounts in any organizational directory (Any Azure AD directory – Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)” to let FluentSMTP get authorized in the later steps and send emails from your Microsoft Account.
3. **Redirect URI:** The Platform should be of **Web** type and the Redirect URI is the **App Callback URL** collected from the earlier step provided by the FluentSMTP plugin on your website.

The Redirect URI is in the following format:

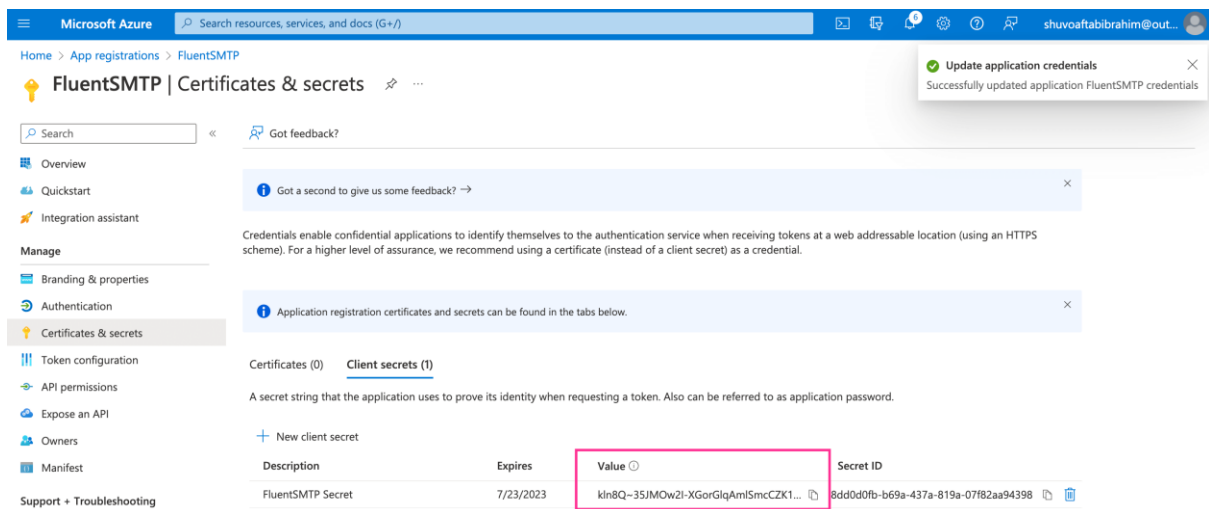
*https://YourDomain.com/wp-json/fluent-smtp/outlook\_callback*

The screenshot shows the Microsoft Azure portal interface. At the top, there's a search bar and a notification: "Create application Successfully created application FluentSMTP." Below that, a navigation pane on the left lists options like "Overview", "Quickstart", "Integration assistant", and "Manage". The main content area shows the "Essentials" section for the application "FluentSMTP". Key details include: Display name: FluentSMTP; Application (client) ID: 34fe4654-7134-4faa-97a6-8629c9673abb (highlighted with a red box); Object ID: 4d9b6306-4c97-4e7b-85e3-b19bb9523ace; Directory (tenant) ID: f8cdef31-a31e-4b4a-93e4-5f571e91255a; Supported account types: All Microsoft account users. On the right, there are links for "Client credentials", "Redirect URIs", and "Application ID URI".

Now we need to create and store the Client Secret Value. To do this, please go to Certificates & Secrets → + New Client Secret



Once you are done please click on the Add button from the bottom and it will redirect you to Client Secret Overview and a Client Secret Value will be generated for you. **Please copy and store this Client Secret Value as this will not be visible again once you leave this page.**



## Add API Permissions (Mail.Send)

After generating the Client Secret, follow the steps below:

### Step 1 – Go to API Permissions

1. In the left menu, click on **API permissions**
2. Click on **+ Add a permission**

The screenshot shows the Azure portal interface for 'Email Configuration MIFCO | API permissions'. The left-hand navigation menu has 'API permissions' highlighted with a red box. The main content area shows a table of configured permissions for 'X Soft Solution Pvt Ltd'. The table has columns for 'API / Permissions name', 'Type', 'Description', 'Admin consent required', and 'Status'. Three permissions are listed: 'email', 'Mail.Send', and 'User.Read', all with a status of 'Granted for X Soft Solu...'. A '+ Add a permission' button is visible at the top of the table.

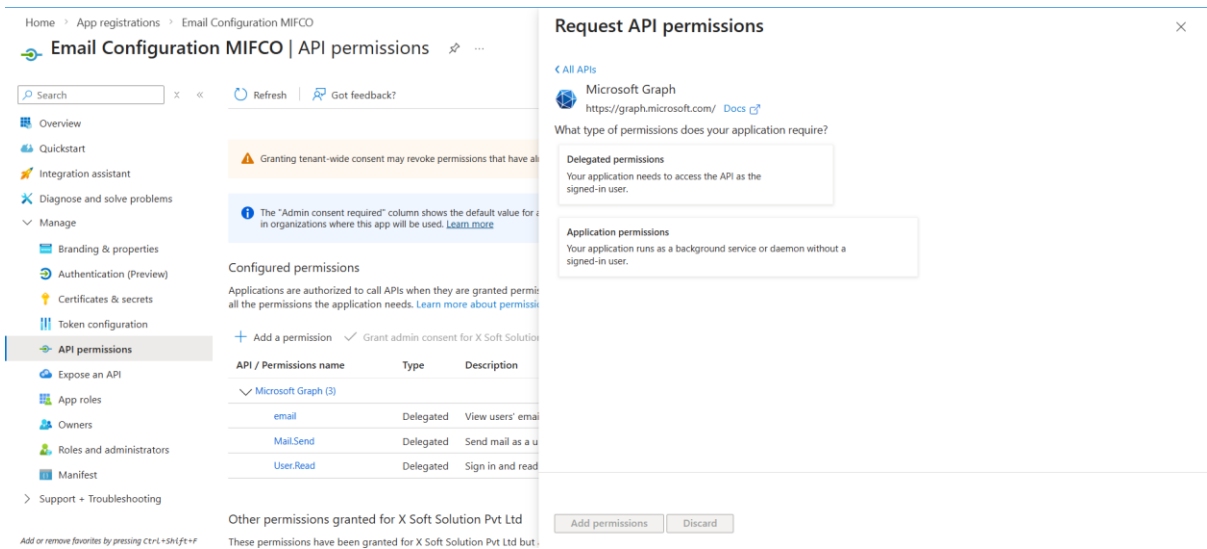
API / Permissions name	Type	Description	Admin consent required	Status
Microsoft Graph (3)				
email	Delegated	View users' email address	No	Granted for X Soft Solu...
Mail.Send	Delegated	Send mail as a user	No	Granted for X Soft Solu...
User.Read	Delegated	Sign in and read user profile	No	Granted for X Soft Solu...

### Step 2 – Select Microsoft Graph

1. Click on **Microsoft Graph**
2. Select **Delegated permissions**

Delegated permissions allow the application to send email as the signed-in user.

The screenshot shows the 'Request API permissions' dialog box. It is titled 'Request API permissions' and has a sub-header 'Select an API'. Underneath, there are three tabs: 'Microsoft APIs', 'APIs my organization uses', and 'My APIs'. The 'Microsoft APIs' tab is selected, showing a list of 'Commonly used Microsoft APIs'. The 'Microsoft Graph' API is highlighted with a grey background. Below it, several other APIs are listed in a grid, including Azure Cosmos DB, Azure DevOps, Azure Key Vault, Azure Rights Management Services, Azure Service Management, Azure Storage, Dynamics 365 Business Central, Dynamics CRM, and Dynamics ERP.

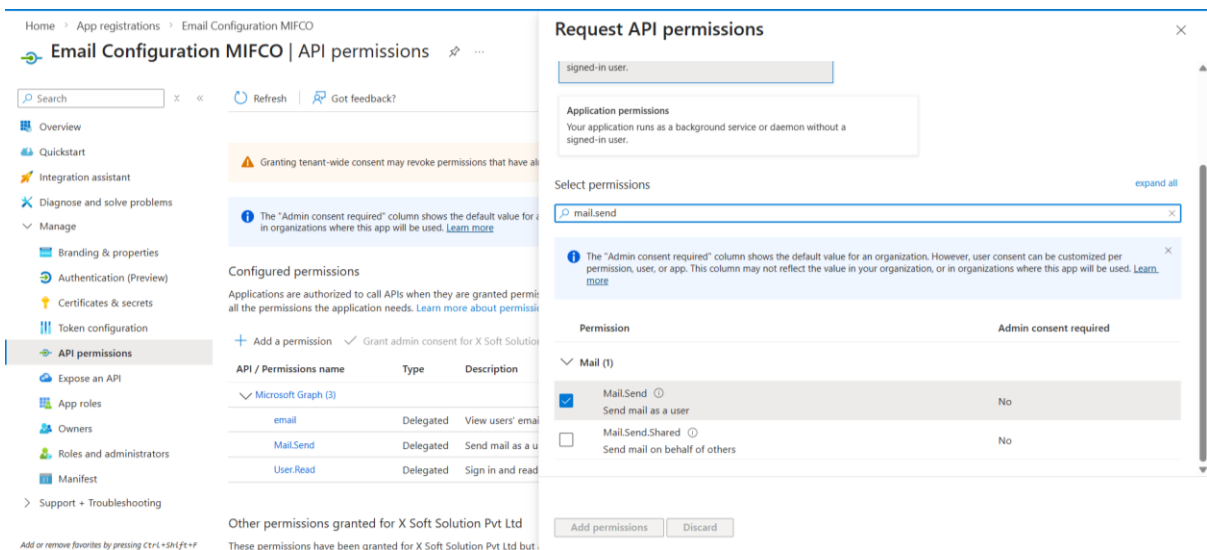


### Step 3 – Add Mail.Send Permission

1. In the search box, type:  
*mail.send*
2. Under the Mail section, select:

#### Mail.Send – Send mail as a user

3. Click **Add permissions**



## Step 4 – Grant Admin Consent (Important)

After adding the permission:

1. Click on **Grant admin consent for [Your Organization Name]**
2. Click **Yes** to confirm

After successful approval, the Status should show:

**Granted**

### ⚠ If You Are Not an Admin

If you are **not a Microsoft 365 Global Admin or Application Admin**, you will not be able to grant admin consent.

In this case:

1. Contact your organization's **Microsoft 365 Global Administrator**
2. Ask the admin to:
  - Go to **Azure Portal → App registrations**
  - Open the application
  - Go to **API permissions**
  - Click **Grant admin consent**
  - Click **Yes** to approve

It is recommended to use an **Admin account** when setting up API permissions to avoid permission errors such as:

```
error=access_denied  
error_subcode=cancel
```

Once the admin grants consent, the Status should change to:

**Granted**

Once this is completed, your Azure App Registration is properly configured to send emails using Microsoft 365.

Now you can proceed back to FluentSMTP and complete the authentication process.

## FluentSMTP Microsoft Settings

We are done obtaining the **Application Client ID** and **Client Secret Value**. After providing them in the **FluentSMTP Settings**, please click on the **“Authenticate with Office365 & Get Access Token”** button.

## Application Details

Edit Connection

Connection Provider

Microsoft  [change](#)

**Sender Settings**

From Email:

From Name:

Set the return-path to match the From Email [?](#)  Force Sender Name [?](#)

**Outlook/Office365 API Settings**

Please [check the documentation first to create API keys at Microsoft](#)

Application Client ID:

Application Client Secret:

App Callback URL (Use this URL to your APP):

Please authenticate with Office365 to get Access Token

Outlook/Office365 is not recommended for sending mass marketing emails.

## Token Generation

Then you will be redirected to the **Token Generation Page** from where we need to copy and store the generated Token to use in the FluentSMTP Settings as below:

Your Access Code

Copy the following code and paste in the fluentSMTP settings

## Save Microsoft Connection Settings

Now provide the earlier generated Token in the FluentSMTP Settings and finally click on the **Save Connection Settings** button.

### Outlook/Office365 API Settings

Please [check the documentation first to create API keys at Microsoft](#)

Application Client ID

Application Client Secret

App Callback URL (Use this URL to your APP)

**Please authenticate with Office365 to get Access Token**

Access Token

Please send test email to confirm if the connection is working or not.

**Outlook/Office365 is not recommended for sending mass marketing emails.**