

Sending SMTP email with Microsoft Graph and OAUTH authentication using C#



Elie Karkafy

Microsoft MVP | MCT | Senior Solutions Architect at ampiO Solutions

13 articles

+ Follow

April 10, 2023

Open Immersive Reader

In my previous article i posted how to send SMTP email with MG and OAUTH using VB.NET (1) [Sending SMTP email with Microsoft Graph and OAUTH authentication using VB.net | LinkedIn](#) . In this post, we will check out a use case we had with one of our customers for an easy way to setup the basic configuration and code for allowing a custom developed application to send emails with Microsoft Graph and OAUTH authentication using C#.

Whether you are building a fully customized app, low app code, or integrations using existing plus and play solutions, Microsoft always recommends you to use a third party or a SendGrid, for sending emails in Azure. This is the way forward to do it but sometimes it's not!

However, before we dive deeper into the config and the code. Let's look when this solution matches the business requirements in some situations.

Sometimes, you may like to send an e-mail and keep the history of the entire conversation at your fingertips - in

your Microsoft 365 mailbox. Using the Microsoft Graph APIs to send e-mails as a user, you can decide whether to keep the sent e-mails in your "Sent" folder. This setup helps a lot with conversations that have sparked from your automated e-mails. For me, this creates a seamless experience.

Simply, it's your code, so you set the rules. Logically integrate the e-mail solution you build into your existing applications and workflows, allowing a fully customized experience.

Sometimes, working with these requirements can be challenging. Using the Microsoft Graph APIs allows you to use the service and mailbox accounts already part of and approved by your organization.

Setup

To allow our applications to send e-mails as a given user or service account, we need to configure an Azure AD application with the appropriate permission. Additionally, we need to ensure that the user or service account has a license assigned for sending e-mails.

The process of configuring our Azure AD apps and users is very straightforward.

Create an Azure AD App with Graph permissions

To allow our app to send e-mails using our user or service account, we need to configure an Azure AD application with the appropriate permissions.

Create a new App Registration

Head over to Azure AD and create a new App Registration.

- Azure Portal > Active Directory > App registrations > New registration
- Name: Whatever you want.
- Type: Accounts in this organizational directory only (Single tenant)
- Redirect URI: Not required.

Register an application ...

* Name

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

test app ✓

Supported account types

Who can use this application or access this API?

- ☒ Accounts in this organizational directory only (Ampio Solutions, Inc. 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](#) ☞

Register

Click on Register

Set up the application permissions

From the test app page in the Azure Portal navigate to:

- API permissions > Add a permission
- Microsoft Graph > Application Permissions > Mail.Send > click Add Permission

Request API permissions

[← All APIs](#)

 Microsoft Graph
<https://graph.microsoft.com/> [Docs](#) ☞

What type of permissions does your application require?

Delegated permissions

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

Application permissions

Your application runs as a background service or daemon signed-in user.

Select permissions

Mail.		Admin consent required
Permission		
▼ Mail (1)		
<input type="checkbox"/>	Mail.Read ⓘ Read mail in all mailboxes	Yes
<input type="checkbox"/>	Mail.ReadBasic ⓘ Read basic mail in all mailboxes	Yes
<input type="checkbox"/>	Mail.ReadBasic.All ⓘ Read basic mail in all mailboxes	Yes
<input type="checkbox"/>	Mail.ReadWrite ⓘ Read and write mail in all mailboxes	Yes
<input checked="" type="checkbox"/>	Mail.Send ⓘ Send mail as any user	Yes

Your configured permissions should look like this:

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins. List of configured permissions should include all the permissions the application needs. [Learn more](#)

[+ Add a permission](#) [✓ Grant admin consent for \[redacted\]](#)

API / Permissions n...	Type	Description	Admin consent req...
Microsoft Graph (1)			
Mail.Send	Application	Send mail as any user	Yes

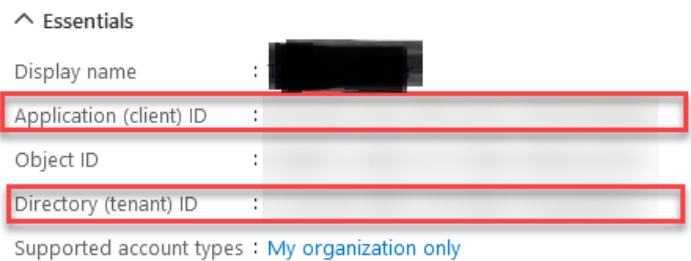
Using a Client ID and Client Secret

We need to create a new secret and securely store the value of the said secret, along with the Tenant ID and the app's Client ID

From the App page navigate to:

- Certificates & Secrets > New client secret
- Copy the secret and store it in a safe location

Next, make sure you copy the Application ID (Client ID) and the Tenant ID for your application. You can find these on the Overview page of your app.



To recap, you should at this point have stored these values for use later in our code:

- Application (client) ID.
- Directory (tenant) ID.
- The value of the secret you created for this app.

Ensure your user or service account has a license to send e-mails

Go to the Azure Portal:

- Active Directory > Licenses > All Products
- (Any license that has a mailbox) - For me, it's Office 365 E5.
- Assign it to the user (For me, again, this is already done).

We are done with the preparations. At this point, we should have:

- New Azure AD application.
- Configured the appropriate permissions for sending e-mails.
- Ensured there was a license assigned to the user account.

The Code

Install the required dependencies

To allow successful communication with Microsoft Graph, we need to make use of a few NuGet libraries.

- [Azure.Identity](#) (NuGet)
- [Microsoft.Graph](#) (NuGet)

We use the `Azure.Identity` to retrieve our `ClientSecretCredential` object, and we use the `Microsoft.Graph` to instantiate a new `GraphServiceClient` and eventually, send e-mails with the `Users.SendMail` endpoint.

Here's the C# code

Replace the `tenantId` with the Tenant ID, replace the `clientId` with the Application ID, replace the secret from the step above.

Replace the UPN on this line `'await graphClient.Users["myuser@mydomain.com"]'` with a user that has a mailbox in your tenant.

I am using Visual Studio with C# to write my code.

```
public static async void SendEmail()
{
    #region Obtain Authentication Token

    var scopes = new[] { "https://graph.microsoft.com/.default"
};

    var tenantId = "1f63282c-f731-4437-8d64-
f8180fbb4676";    // Replace with your Tenant ID
```

```
var clientId = "r1ct9e4e-5a26-4073-ac9c-  
fd07744c5fcy"; // Replace with Application ID from  
Overview tab
```

```
var clientSecret =  
"8tk2Q~DmhAz.lGbgcIVN.vcTIVRNYMW3672f45dIJ"; //  
Replace with the Client Secret
```

```
// using Azure.Identity;
```

```
var options = new TokenCredentialOptions
```

```
{
```

```
AuthorityHost = AzureAuthorityHosts.AzurePublicCloud
```

```
};
```

```
var clientSecretCredential = new  
ClientSecretCredential(tenantId, clientId, clientSecret,  
options);
```

```
#endregion
```

```
#region Send Email Using MS Graph
```

```
GraphServiceClient graphClient = new  
GraphServiceClient(clientSecretCredential, scopes);
```

```
var message = new Microsoft.Graph.Message()
```

```
{
```

```
Subject = "This is the subject line",
```

```
Body = new ItemBody
```

```
{
```

```
ContentType = BodyType.Text,
```

```
Content = "This is the body of the email message."
```

```
},
```

```
ToRecipients = new List<Recipient>()
```

```
{
```

```
new Recipient
```

```

{
    EmailAddress = new EmailAddress
    {
        Address = "user@somedomain.com" // This is the
recipient of the message
    }
}
};

await graphClient.Users["myuser@mydomain.com"] //
This is the user account to send from. An account in your
tenant

.SendMail(message, null)

.Request()

.PostAsync();
}

```

#endregion

That's it !

Your custom application is now ready to send emails in Azure with Microsoft Graph and OAUTH Authentication using C#.

[Report this](#)

Published by



Elie Karkafy

Microsoft MVP | MCT | Senior Solutions Architect at ampiO Solutions
Published • 8mo

13
articles

[+ Follow](#)



In this article, we will check out an easy way to setup the basic configuration and code for allowing a custom developed application to send emails with Microsoft Graph and OAUTH authentication using C#. [#azureactivedirectory](#) [#microsoftgraph](#) [#azuretips](#)



Like



Comment

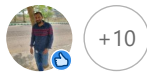


Share

22 17 comments

Reactions





17 Comments

Most relevant ▾



Add a comment...



rachel dsouza • 3rd+
Team Lead at Idealake | .Net , SQL & Sitefinity

5mo ...

Hi Elie, the code is giving following error. Please let me know if there is way to fix it.



Like | Reply · 3 Replies

Load previous replies



Elie Karkafy • 3rd+
Microsoft MVP | MCT | Senior Solutions Architect at ampiO Solutions
[rachel dsouza](#) can show me please the complete error

5mo ...

Like | Reply



Brice Gnoboa • 3rd+
Développeur Fullstack Dotnet chez SOTEL'S

5mo ...

This code doesn't work anymore



Like | Reply · 2 Replies



Elie Karkafy • 3rd+
Microsoft MVP | MCT | Senior Solutions Architect at ampiO Solutions
Hi [Brice Gnoboa](#) seems with the new version of Graph SDK , the .sendemail isnt available anymore . Did you try to use the graph sdk v1.0 ?

5mo ...

Like | Reply



Brice Gnoboa • 3rd+
Développeur Fullstack Dotnet chez SOTEL'S
Ok, i'll try next time. Thanks a lot !

5mo ...

Like · 1 | Reply

Load more comments

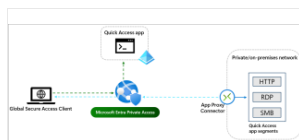


Elie Karkafy

Microsoft MVP | MCT | Senior Solutions Architect at ampiO Solutions

+ Follow

More from Elie Karkafy



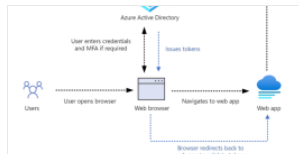
RDP Private Resources using Microsoft Entra Private Access - Quick Access

Elie Karkafy on LinkedIn



What's new and What's learning period in Microsoft Defender for Identity.

Elie Karkafy on LinkedIn



Protect your identities from a Token theft using Token Protection in Conditional...

Elie Karkafy on LinkedIn

[See all 13 articles](#)