Setting Up Microsoft Graph API

Here’s a step-by-step guide to set up and use the **Microsoft Graph API** to send emails:

**Step 1: Register an Application in Azure Active Directory**

1. **Go to the Azure Portal**:
   * Open [Azure Portal](vscode-file://vscode-app/private/var/folders/yy/10g095k5669dkp3511kz6lxw0000gn/T/AppTranslocation/378E2F82-3275-46E8-8EC2-A9940E34FFD6/d/Visual%20Studio%20Code.app/Contents/Resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and log in with your Microsoft account.
2. **Navigate to App Registrations**:
   * In the left-hand menu, go to **Azure Active Directory** > **App registrations**.
3. **Register a New Application**:
   * Click **New registration**.
   * Provide a name for your app (e.g., "Email Sender App").
   * Choose **Accounts in any organizational directory and personal Microsoft accounts** (if you want to support both work/school and personal accounts).
   * Click **Register**.
4. **Copy the Application (Client) ID**:
   * After registering, copy the **Application (client) ID**. You’ll need this later.

**Step 2: Configure API Permissions**

1. **Go to API Permissions**:
   * In your app's overview page, click **API permissions** in the left-hand menu.
2. **Add Permissions**:
   * Click **Add a permission**.
   * Select **Microsoft Graph**.
   * Choose **Application permissions**.
   * Search for and select the following permission:
     + Mail.Send (required to send emails).
   * Click **Add permissions**.
3. **Grant Admin Consent** (if required):
   * If you're using a work/school account, you may need an administrator to grant consent for the permissions.

**Step 3: Generate a Client Secret**

1. **Go to Certificates & Secrets**:
   * In your app's overview page, click **Certificates & secrets** in the left-hand menu.
2. **Create a New Client Secret**:
   * Click **New client secret**.
   * Provide a description (e.g., "Email Sender Secret") and set an expiration period.
   * Click **Add**.
3. **Copy the Client Secret**:
   * Copy the generated client secret. You won’t be able to see it again, so save it securely.

**Step 4: Authenticate and Get an Access Token**

To use the Microsoft Graph API, you need an **access token**. This requires using the **OAuth2 Authorization Code Flow**.

1. **Install Required Libraries**: Install the [requests](vscode-file://vscode-app/private/var/folders/yy/10g095k5669dkp3511kz6lxw0000gn/T/AppTranslocation/378E2F82-3275-46E8-8EC2-A9940E34FFD6/d/Visual%20Studio%20Code.app/Contents/Resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) library if you haven’t already:

pip install requests

1. **Use the Following Code to Get an Access Token**: Replace the placeholders with your app's details.

import requests

def get\_access\_token(tenant\_id, client\_id, client\_secret):

url = f"https://login.microsoftonline.com/{tenant\_id}/oauth2/v2.0/token"

data = {

"grant\_type": "client\_credentials",

"client\_id": client\_id,

"client\_secret": client\_secret,

"scope": "https://graph.microsoft.com/.default"

}

response = requests.post(url, data=data)

if response.status\_code == 200:

return response.json()["access\_token"]

else:

print(f"Failed to get access token: {response.status\_code}, {response.text}")

return None

# Replace with your values

tenant\_id = "your\_tenant\_id"

client\_id = "your\_client\_id"

client\_secret = "your\_client\_secret"

access\_token = get\_access\_token(tenant\_id, client\_id, client\_secret)

print("Access Token:", access\_token)

* **Tenant ID**: Found in the Azure Active Directory overview page.
* **Client ID**: The Application (client) ID from Step 1.
* **Client Secret**: The secret you generated in Step 3.

**Step 5: Send an Email Using the Graph API**

Once you have the access token, use it to send an email.

1. **Use the Code You Already Have**: Replace your\_access\_token with the token obtained in Step 4.

send\_email\_via\_graph\_api(

access\_token="your\_access\_token",

subject="Test Email",

body="<h2>Test HTML Body</h2>",

to\_email="recipient@example.com"

)

1. **Verify the Email**: Check the recipient's inbox to confirm the email was sent successfully.

**Step 6: Handle Token Expiration**

Access tokens expire after 1 hour. To avoid interruptions:

* Use the **refresh token** (if applicable) to get a new access token.
* Alternatively, re-run the authentication process to get a new token.

**Summary of Required Information**

* **Tenant ID**: Found in Azure Active Directory.
* **Client ID**: Application (client) ID from Step 1.
* **Client Secret**: Generated in Step 3.
* **Access Token**: Obtained in Step 4.

**Additional Notes**

* **Testing**: Use a test account to avoid sending emails from your primary account during development.
* **Security**: Never hardcode sensitive information (like client secrets) in your code. Use environment variables or a secure secrets manager.
* **Rate Limits**: Be aware of Microsoft Graph API rate limits to avoid throttling.