Outlook Email with integr8tor

Overview

This guide explains how to use Microsoft Graph API with OAuth2 for Integr8tor email input. The Graph API provides a modern, secure way to access Microsoft 365 email services.

Account Requirements and Limitations

Supported Accounts

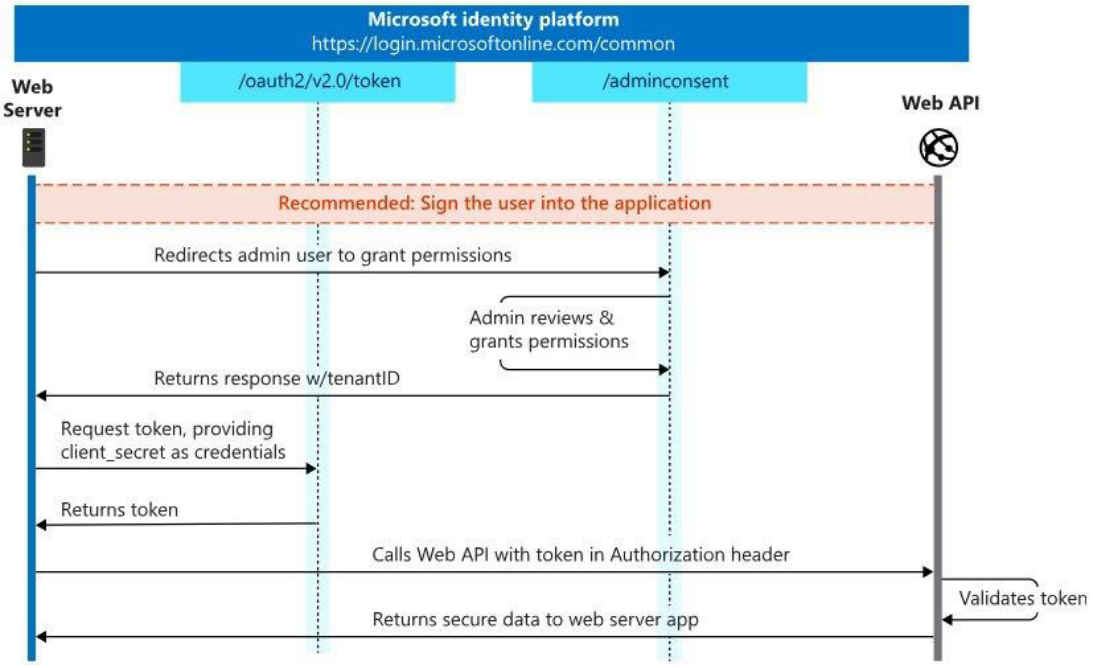
- Microsoft 365 Business Accounts Only
  - Enterprise E1, E3, E4 licenses
  - Business Basic, Standard, Premium licenses
  - Must have admin consent for application permissions

Unsupported Accounts

- Personal Microsoft accounts (required browser interaction and different configuration)
- Accounts requiring multi-factor authentication (MFA) for every login

Authentication Flow

[Protocol Diagram]



This implementation uses the OAuth2 Client Credentials Flow, which:

- Operates server-to-server without user interaction

- Does not require browser-based authentication
- Uses application permissions instead of delegated permissions
- Requires admin consent in Azure

Client Secret Management

Important: Client secret management is the user's responsibility. This includes:

- Monitoring secret expiration dates
- Rotating secrets before they expire
- Ensuring Integr8tor application configuration is updated with new secrets
- Recommended to set calendar reminders for secret rotation
- Back up secret values securely before rotation

Azure Application Setup and Permissions

1. Go to **Azure Portal** from https://portal.azure.com
2. Navigate to **Azure Home** → Click **App registrations**.
3. Register an application with the following details:
    a. Name: `OutlookEmailFetcher`
    b. Supported account types: `Accounts in the organization directory only`
    c. Redirect URI: Leave empty (not required for app-only authentication)
    d. Click Register.
4. Get Client Credentials
    a. Copy the following from Overview under **App registrations**
        i. Application (client) ID
        ii. Directory (tenant) ID
5. Create a client secret
    a. Go to **Certificates & secrets** and click + New client secret.
    b. Enter name (e.g. Integr8torSecret)
    c. Set expiration (180 days recommended)
    d. Copy the Client Secret ID and Value
        i. Note: Once a Secret is created and you navigate away from the Certificates & Secrets page, the Secret value is no longer available to be seen or copied.
6. Assign API Permissions
    a. Go to **API permissions** → Click **+ Add a permission**
    b. Select Microsoft Graph and Click **Application Permissions**.
    c. Search and add the following permissions:
        i. `Mail.Read` (Read user emails)
        ii. `Mail.ReadWrite` (Read & manage emails)
        iii. `Mail.Send` (Send email): Only required for sending confirmation emails.
        iv. `User.Read.All`
    d. Click **Add permissions**.

e. Grant Admin Consent
    i. Navigate to **Enterprise Applications**
    ii. Find our app named **OutlookEmailFetcher**
    iii. Go to **Security** section and click **Permissions**
    iv. Click the **Grant admin consent** button and accept the permissions (Note: only admin user can grant admin consent for permissions.)

Configuration Parameters

All configuration parameters must be stored in `webapps/ROOT/WEB-INF/prefs/UsersPreferences.xml` file within the `<node name="outlookserver">` element.

Required Parameters

- `active` : Boolean flag (true or false) to enable / disable retrieving Outlook emails.
- `clientId` : Application (client) ID from Azure portal
- `tenantId` : Directory (tenant) ID from Azure portal
- `clientSecret` : Generated secret value from Certificates & secrets
- `email` : Target email address for operations (only one email)

Optional Parameters

- `dbContext` : database context (default: *autoflow*)
- `confirmation` : Boolean flag (true or false) to enable / disable email confirmation after job submission (default: *false*)
- `debug` : Boolean flag (true or false) to enable / disable showing debugging message (default: *false*)
- `pollingTimeSeconds` : Time between email fetch operations in seconds (Default: 300 seconds, Min: 30 seconds)
- `maxAgeInMin` : Maximum age of emails to fetch in minutes (Default: 60 minutes, Min: 5 minutes)
- `maxMessagesPerRequest` : Number of results per API call (1-100) (Default: 15, Min: 1, Max: 100)

Example

```
<node name="outlookserver">
  <map>
    <entry key="active" value="true"/>
    <entry key="clientId" value="*****-****…….*****-*******"/>
    <entry key="tenantId" value="*****-****-****-*****-*******"/>
    <entry key="clientSecret" value="abc************"/>
    <entry key="email" value="email-input@ucamco.com"/>
    <entry key="pollingTimeSeconds" value="300"/>
    <entry key="dbcontext" value="autoflow"/>
    <entry key="confirmation" value="false"/>
    <entry key="debug" value="true"/>
    <entry key="maxAgeInMin" value="90000"/>
    <entry key="maxMessagesPerRequest" value="15"/>
  </map>
</node>
```

Graph API Limitations

Rate Limits

- Maximum 10,000 API requests per 10-minute period per user mailbox
- Maximum 4 concurrent requests per application per mailbox
- 150MB Upload Limit in a 5 minute-period per mailbox
- Maximum 150 MB email size
- Maximum 150 MB attachment size
- Mailbox Storage
    - Personal Accounts
        - 150GB (free account)
    - Business Accounts
        - Per mailbox
            - Business Basic/Standard: 50GB
            - Enterprise E3/E5: 100GB mailbox
            - Shared mailbox: 50GB
- More info
    - Personal / Business throttling: https://learn.microsoft.com/en-us/graph/throttling
    - Service limits: https://learn.microsoft.com/en-us/office365/servicedescriptions/exchange-online-service-description/exchange-online-limits

API Operations Breakdown

Email Operations and Request Counts

1. Fetching Emails
    a. Returns multiple emails based on `maxMessagesPerRequest`
    b. Counts as 1 API call per 1 fetch operation
2. Attachment Operations

   a. Each attachment requires a separate download request
   b. For N attachments in an email, requires N separate API calls
   c. Example: Email with 3 attachments = 3 API calls
3. Marking as Read
   a. Counts as 1 API call per email
4. Deleting emails
   a. Counts as 1 API call per email
5. Sending confirmation
   a. Counts as 1 API call per job (attachment)

Example Scenario

For processing one email with three attachments:

```
Total API Calls = 1 (fetch) + 3 (attachments) + 1 (mark as read)
+ 1 (delete) + 3 (confirmation email) = 9 API calls
```

Troubleshooting

Common issues and solutions:

1. Secret key expiration errors
2. Permission / Authentication issues: Verify Azure app permissions
3. Missing emails: Check `maxAgeInMin` and `pollingTimeSeconds` settings