# Automated Email Sending API Documentation

Contents

[Automated Email Sending API Documentation 1](#_Toc194915806)

[Introduction 1](#_Toc194915807)

[Authentication/Authorisation 2](#_Toc194915808)

[Authentication Methods 2](#_Toc194915809)

[Authorisation scopes 2](#_Toc194915810)

[Scope: 2](#_Toc194915811)

[Token Management 2](#_Toc194915812)

[Libraries Required 2](#_Toc194915813)

[Identities required 2](#_Toc194915814)

[Permissions/Roles 2](#_Toc194915815)

[API Endpoints/URL/URIs 2](#_Toc194915816)

[References 3](#_Toc194915817)

## Introduction

The Automated Email sending API allows developers to send emails with or without attachments for the reporting purposes. This API connects with the Microsoft Exchange server to be able to fulfil the posting request of the mail to recipients. A challenge encountered during this is that due to the way personal 80hg.io accounts are set up with their hybrid accounts, it does not allows emails to be sent using this type of accounts only accounts set up with Cloud accounts.

This document only contains information regarding the Mail.Send API of this and not the Group.Read.All or User.Read.All. These are separate to minimise security and only user within a certain user group can send emails with Cloud accounts. The additional function for these are simply obtaining user IDs or group IDs from the specified email name or group name and ensuring the user is part of the group.

Use-case:

* PAFA Reporting
* Finance Project
  + Cost Center Reporting
  + Vertical Profit & Loss
* ADD others…

## Authentication/Authorisation

Authentication Methods

OAuth

Authorisation scopes

Scope: https://graph.microsoft.com/.default

Token Management:

Access token

### Libraries Required

MSAL – Microsoft Authentication Library

OpenPyxl – Open Python Excel

Identities required

Tenant ID

Client ID

Client Secret

Permissions/Roles

App Registration Permission

* Application
  + Mail.send
  + Group.Read.All
  + User.Read.All

No Microsoft Graph Permissions needed – Even though it uses Microsoft Graph, it uses the permissions on the App registration.

## API Endpoints/URL/URIs

**Method**: POST

**URL**: [https://graph.microsoft.com/v1.0/users/{from\_email}/sendMail](https://graph.microsoft.com/v1.0/users/%7bfrom_email%7d/sendMail)

**Auth**: ConfidentialClientApplication(Client ID, Authority (URL), Client Secret)

**Request headers**:

|  |  |
| --- | --- |
| Authorization | Bearer {Access token} |
| Content-Type | application/json |
| Accept | application/json |

**Success response**: 202

**Error response**: 400\*

\* Any 400, 401, 402, 403, 404 error codes

## References

<https://readmedium.com/python-read-and-send-outlook-mail-using-oauth2-token-and-graph-api-53de606ecfa1>

<https://learn.microsoft.com/en-us/graph/api/user-sendmail?view=graph-rest-1.0&tabs=http>

<https://msal-python.readthedocs.io/en/latest/>