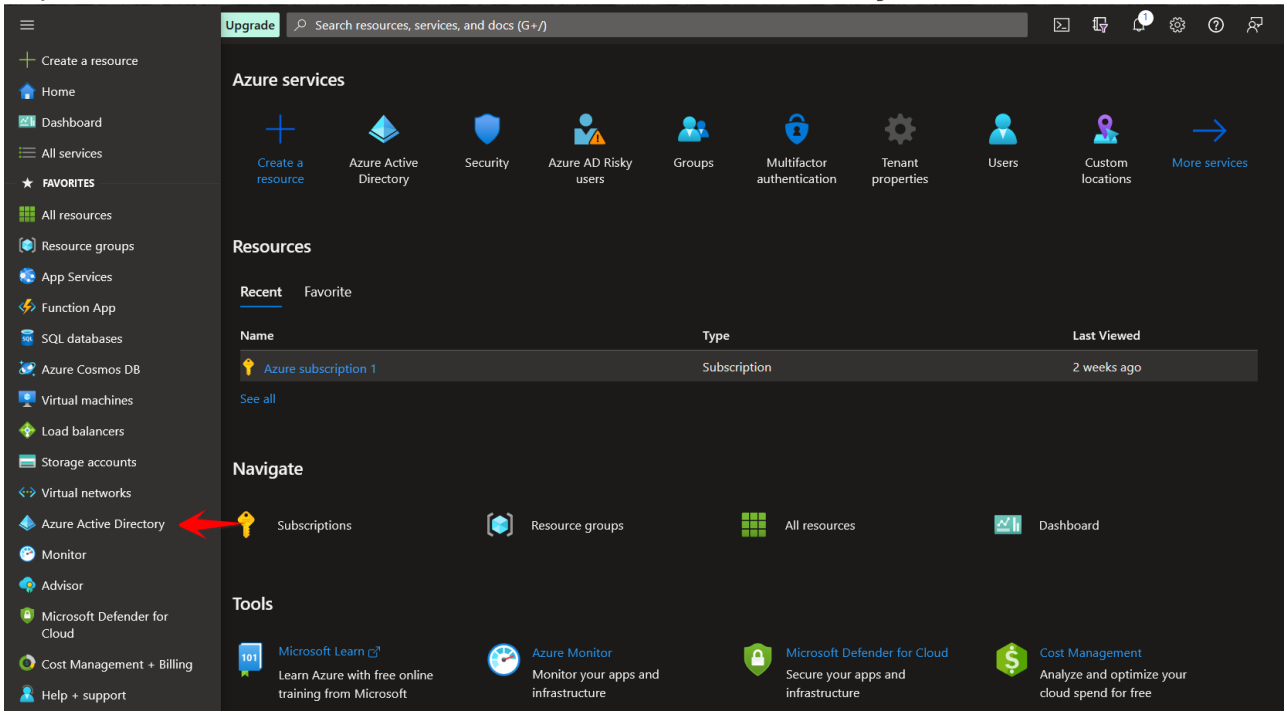
Configure Outlook 365 Email Clients with OAuth 2.0

Oauth2 support has been added for email clients such as Outlook that have deprecated basic authentication for IMAP email and moved to Oauth.

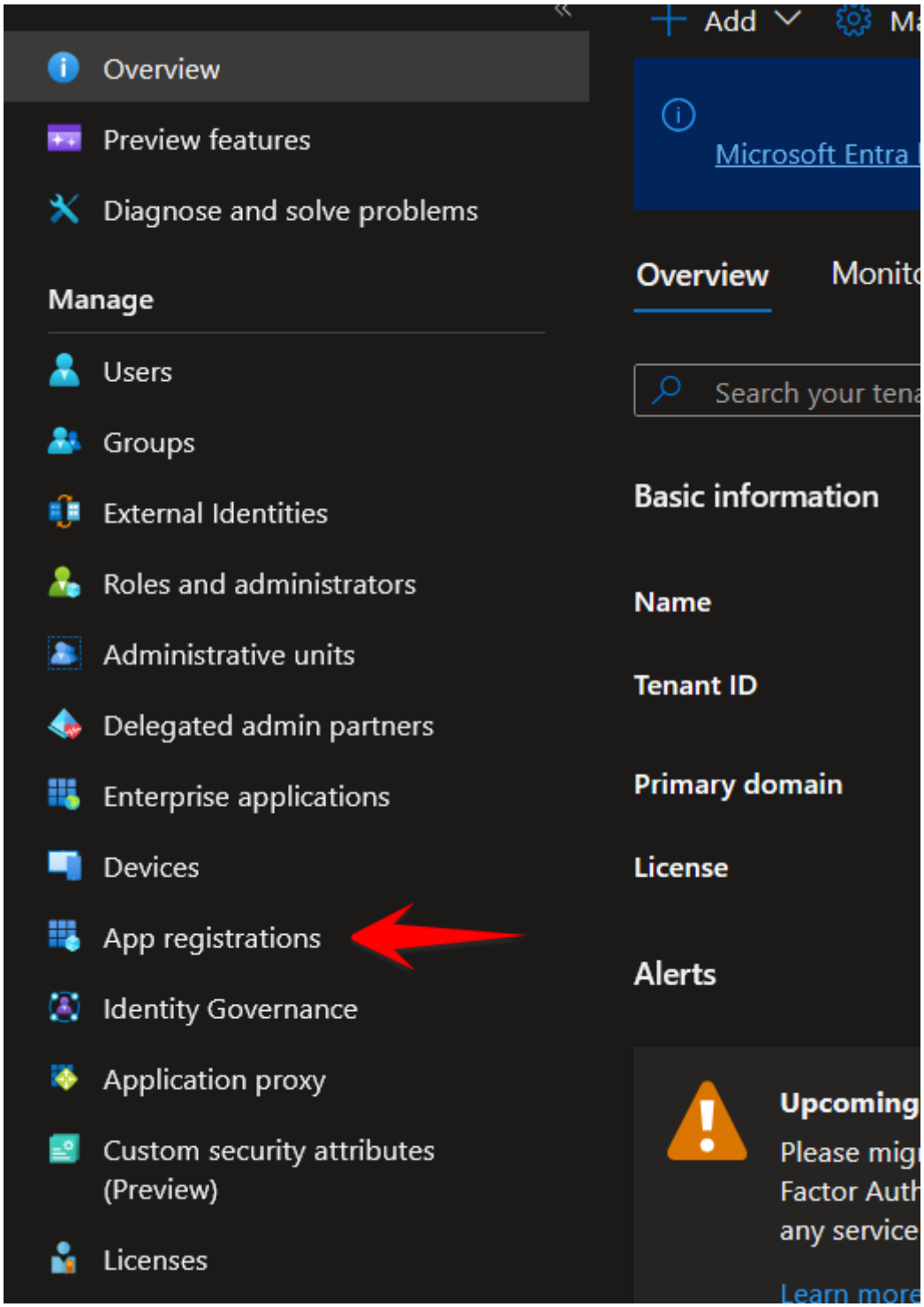
Support for this was added in the AIQ 4.8.8 and 4.9.0 releases.

Setting up Azure

1. Go to >Azure (https://portal.azure.com/) and sign in with your credentials.
2. Expand the left side menu and select **Azure Active Directory**.



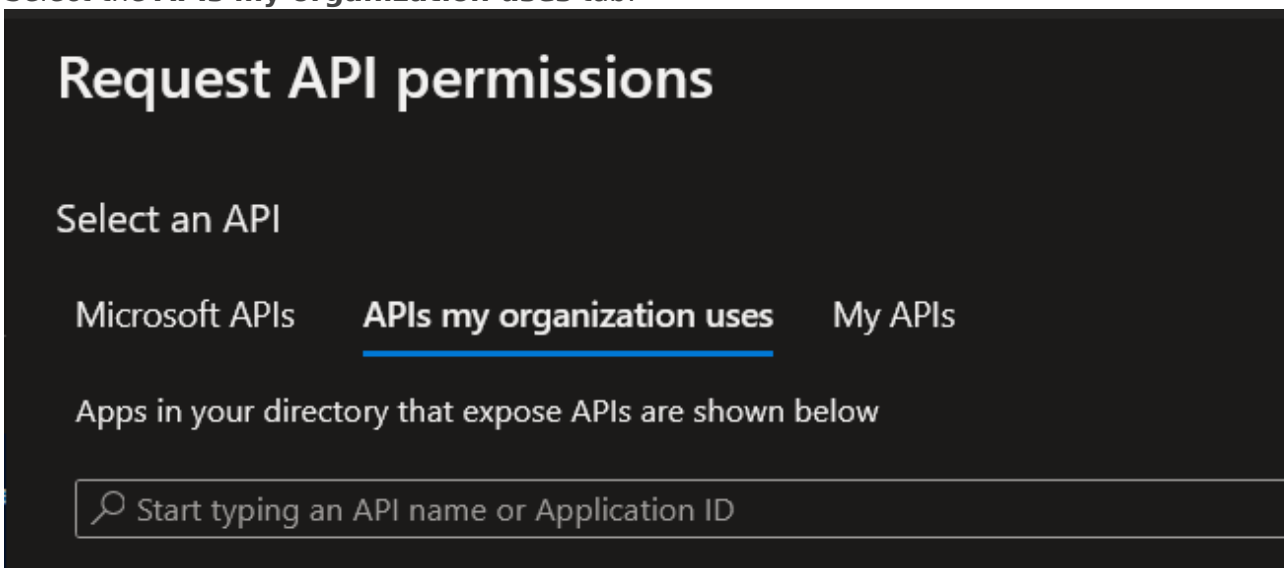
3. In the left side menu, select **App registrations**.



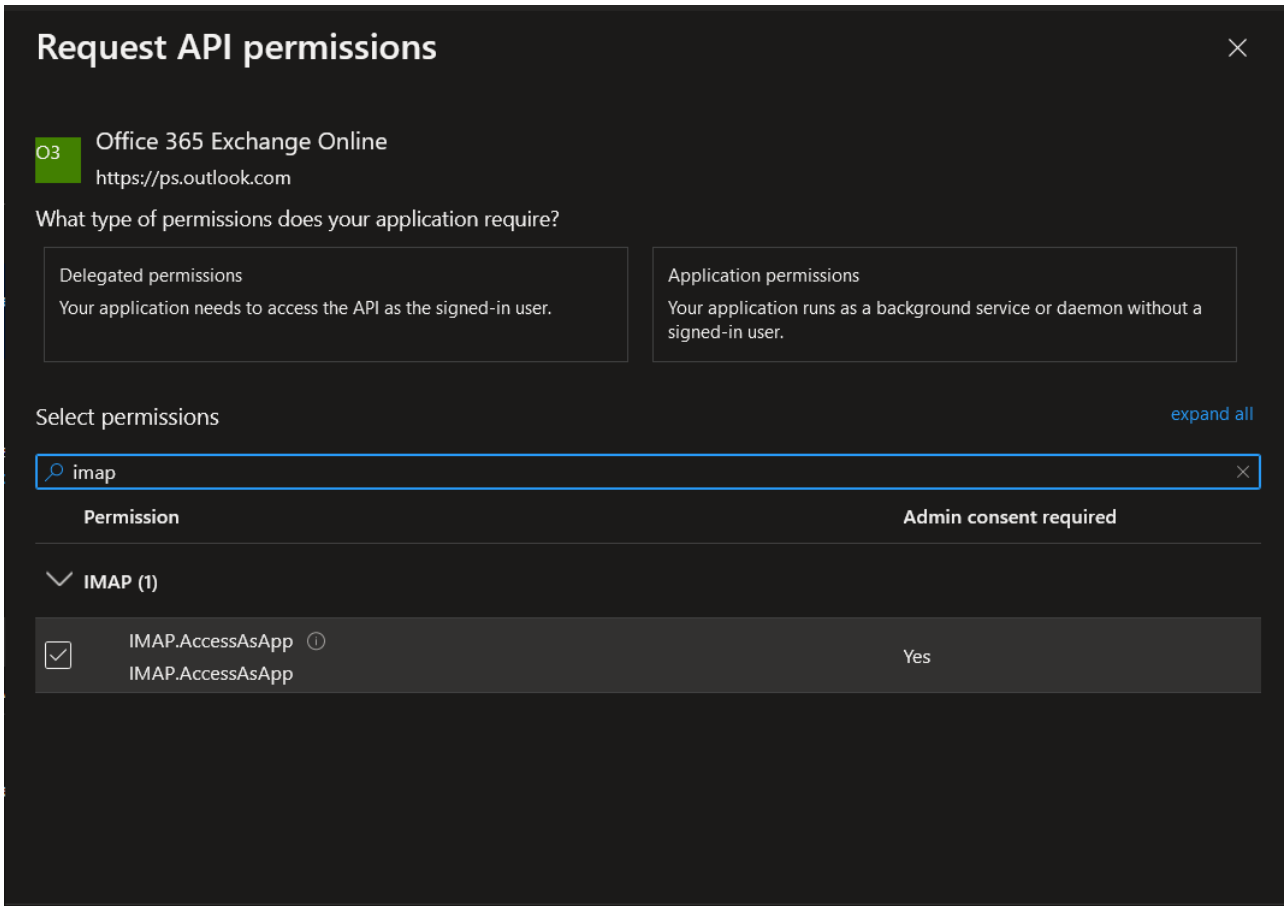
4. Click **New Registration**.
5. Enter a name to identify your application.
6. In **Supported account types**, select **Accounts in this organizational directory only (Single tenant)**.



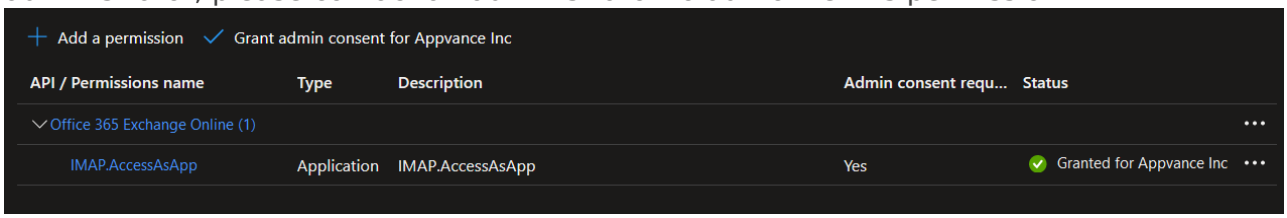
7. Click **Register**.
8. In the left side menu, click **API Permissions**.
9. Delete any existing permissions.
10. Click **Add a permission**.
11. Select the **APIs my organization uses** tab.



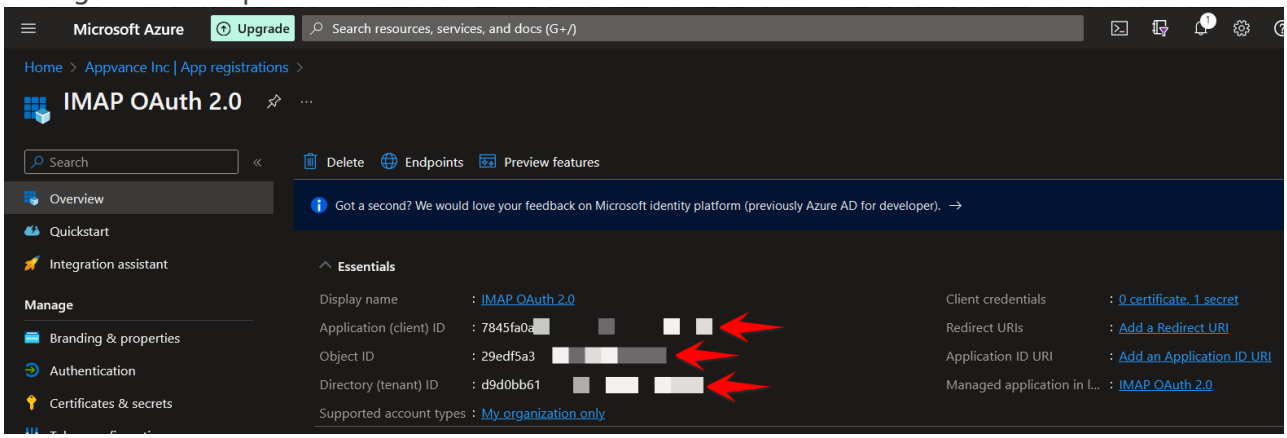
12. Search for “Office 365 Exchange Online“. In the select permissions search box type “IMAP“ and select the results.



13. Click **Update permissions**.
14. This permission needs to be authorized by an administrator of your organization. If your are the administrator, click **Grant admin consent for {Organization}**. If you're not an administrator, please contact an administrator to authorize this permission.

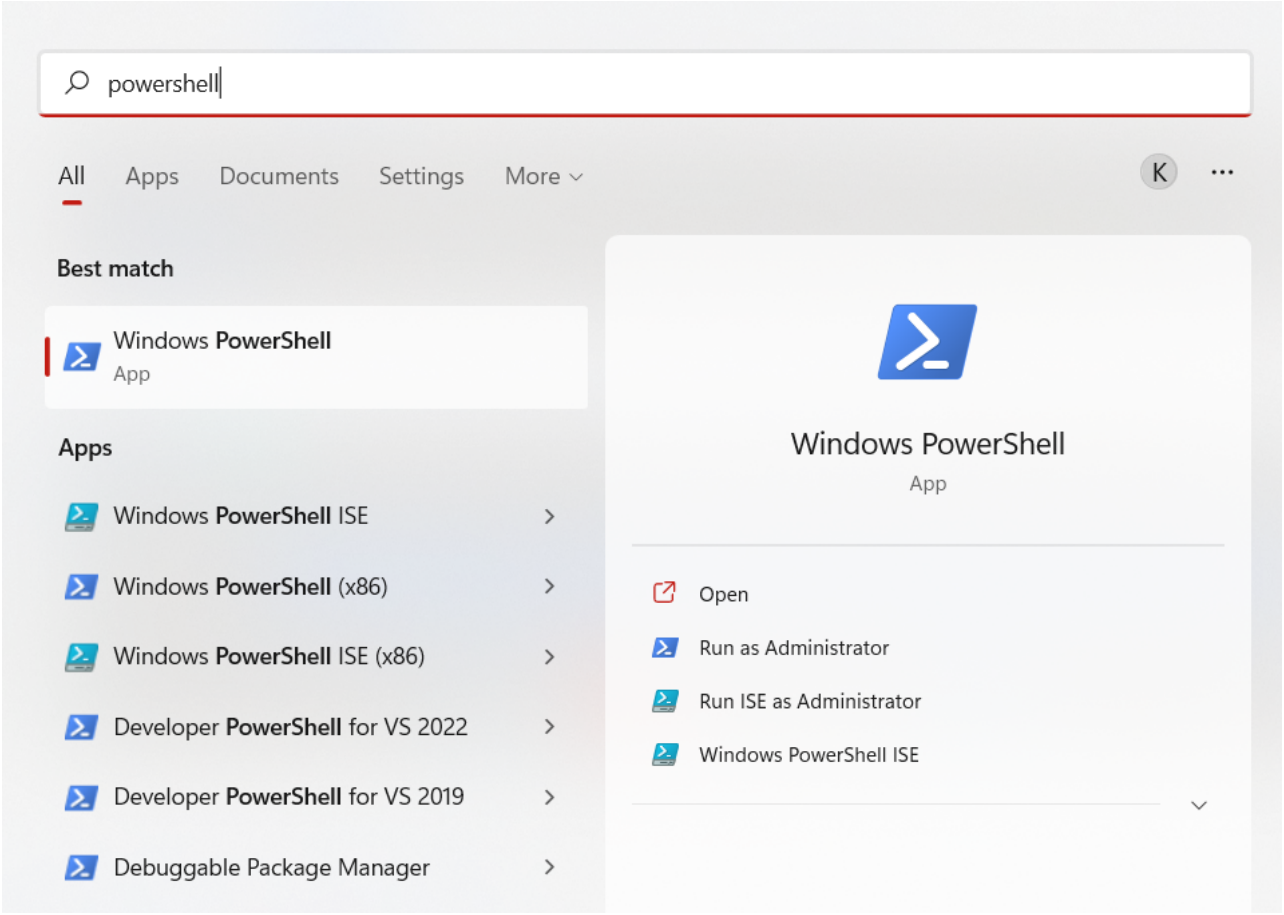


15. Save the values in the Overview page. You will need them in the command line configuration steps.



Command Line Configurations

1. Run Windows PowerShell as an Administrator.



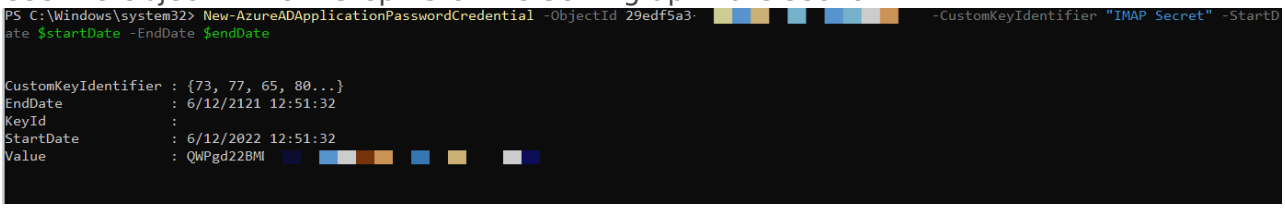
2. Run the command: `Install-Module AzureAD`
   If it requests a confirmation, press Y and then Enter.
3. Run the command: `Install-Module ExchangeOnlineManagement`
   If it requests a confirmation, press Y and then Enter.
4. Run the command: `set-executionpolicy unrestricted`
   If it requests a confirmation, press Y and then Enter.
5. Run the command: `Connect-ExchangeOnline --Credential $Credential`
   The output should be something similar to this:

```
PS C:\Windows\system32> Connect-ExchangeOnline -Credential $Credential

------------------------------------------------------------------------------
This V3 EXO PowerShell module contains new REST API backed Exchange Online cmdlets which doesn't require WinRM for Client-Server communication. You can no
w run these cmdlets after turning off WinRM Basic Auth in your client machine thus making it more secure.

Unlike the EXO* prefixed cmdlets, the cmdlets in this module support full functional parity with the RPS (V1) cmdlets.

V3 cmdlets in the downloaded module are resilient to transient failures, handling retries and throttling errors inherently.

However, REST backed EOP and SCC cmdlets are not available yet. To use those, you will need to enable WinRM Basic Auth.

For more information check https://aka.ms/exov3-module
------------------------------------------------------------------------------
```

6. Run the command: `$app = Get-AzureADServicePrincipal -SearchString "{Application Name}"`
   .
   Use the application name from Step 15 of the Setting up Azure section.
7. Run the command: `New-ServicePrincipal -AppId $app.AppId -ServiceId $app.ObjectId -DisplayName "{Application Name} Service Principal"`
   The output should be something similar to this:

```
PS C:\Windows\system32> New-ServicePrincipal -AppId $app.AppId -ServiceId $app.ObjectId -DisplayName "IMAP OAuth Service Principal"

DisplayName                     ServiceId                            AppId
-----------                     ---------                            -----
IMAP OAuth Service Principal    0ff8586f-                            7845fa0a-
```

8. Run the command: `Add-MailboxPermission -Identity "<USER ID>" -User $app.ObjectId -AccessRights FullAccess`
   The output should be something similar to this:
   

9. Run the command: `$startDate = Get-Date`
10. Run the command: `$endDate = $startDate.AddYears(99)`
11. Run the command: `New-AzureADApplicationPasswordCredential -ObjectId {Object ID of Application} -CustomKeyIdentifier "Application Secret" -StartDate $startDate -EndDate $endDate`
    Use the Object ID from Step 15 of the Setting up Azure section.
    

Example Test Script

You can use the following example as a test Service Suite script. You must perform the above configuration steps first, so you can collect the required parameters: secret key, application id and tenant id.

```
var mailClient = createOAuthMailClient( "mail@outlook.com", "secretKey", "applicationID",
"tenantid");
mailClient.loginOAuth();
var Mails = mailClient.getOAuthEmails();
log(" " + Mails.toString());
mailClient.logoutOAuth();
```

See List of Services Workbench Calls for more details.

Last updated: August 02, 2023

If you have any feedback on this topic, please contact us at documentation@appvance.ai

---

Revision #10
Created 2026-07-31 15:38:59 UTC by Admin
Updated 2026-11-17 22:01:00 UTC by AI_Mananger