

Application registration and setup

Some of the features in Puzzel's Agent Application requires users to authenticate themselves in Azure AD. Follow the steps below which takes you through the process of configuring Puzzel Office 365 Connector application in Azure AD. This is done as a two-step process as explained below.

Application registration

Follow the procedure below to register the Office 365 connector application in Azure AD:

1. Go to <https://portal.azure.com/>, and open Azure Active Directory and select App registrations and click on New registration button

The screenshot shows the Azure AD App Registrations console. The left-hand navigation pane is visible, with 'App registrations' selected. The main content area displays a list of applications under the 'Owned applications' tab. The list includes the following entries:

Application Name	Created Date	Status
CS Catalog Sync	2/28/2018	Expired
MT MSAL Testapp	7/10/2018	-
DU Dummy	10/9/2019	-
PC Puzzel Catalog Sync	5/24/2018	-
AZ AzureNewinterface-Test	10/22/2019	Current
MN My Node.js App	7/6/2018	Current
DS Devpuzzel SSO	2/12/2018	Current
PS Puzzel Skype	10/11/2018	-
PC Puzzel catalog sync	10/23/2019	-

2. Enter the following three parameters in the Register an application window and click **Register**
 - o Name - Choose a name for the application (e.g. O365 Email Connector)
 - o Supported account types – Choose the option that is most relevant to you. The recommended option is “Accounts in this organizational directory only “
 - o Redirect URI (optional) – Choose Web and enter <https://localhost>

Register an application

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

O365 Email Connector ✓

Supported account types
Who can use this application or access this API?

Accounts in this organizational directory only (Development Puzzel 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)

[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.

Web | https://localhost ✓

[By proceeding, you agree to the Microsoft Platform Policies](#)

[Register](#)

3. You will see the screen below after the application has been registered. Use the value of Application ID to fill in for the **Clientid** field in the service config file.

The screenshot shows the 'O365 Email Connector' application registration page in the Azure portal. The left sidebar contains navigation options: Overview, Quickstart, Manage (Branding, Authentication, Certificates & secrets, API permissions, Expose an API, Owners, Roles and administrators, Manifest), and Manifest. The main content area displays the application details:

- Display name: O365 Email Connector
- Application (client ID): [Redacted]
- Directory (tenant) ID: [Redacted]
- Object ID: [Redacted]
- Supported account types: My organization only
- Redirect URIs: 1 web, 0 public client
- Application ID URI: Add an Application ID URI
- Managed application in: O365 Email Connector

Below the details, there are sections for 'Call APIs' (with icons for various services) and 'Documentation' (with links to Microsoft identity platform, authentication scenarios, authentication libraries, code samples, Microsoft Graph, glossary, and help and support).

4. Click on Manifest option to edit the application Manifest You need to edit the manifest of the Azure application by adding/editing an object in the current JSON

O365 Email Connector - Manifest

Save Discard Upload Download

The editor below allows you to update this application by directly modifying its JS manifest.

```

1  {
2    "id": "8e0751e7-bad6-4249-84df-ea77c5a3d534",
3    "acceptMappedClaims": null,
4    "accessTokenAcceptedVersion": null,
5    "addIns": [],
6    "allowPublicClient": null,
7    "appId": "becbb7df-2c38-46ca-96c9-51fd983a3f1",
8    "appRoles": [],
9    "oauth2AllowUrlPathMatching": false,
10   "createdDateTime": "2019-11-01T09:58:58Z",
11   "groupMembershipClaims": null,
12   "identifierUri": [],
13   "informationalUrls": {
14     "termsOfService": null,
15     "support": null,
16     "privacy": null,
17     "marketing": null
18   },
19   "keyCredentials": [
20     {
21       "customKeyIdentifier": "<base64Thumbprint >",
22       "keyId": "<keyid>",
23       "type": "AsymmetricX509Cert",
24       "usage": "Verify",
25       "value": "<base64Value>"
26     }
27   ]
28 }

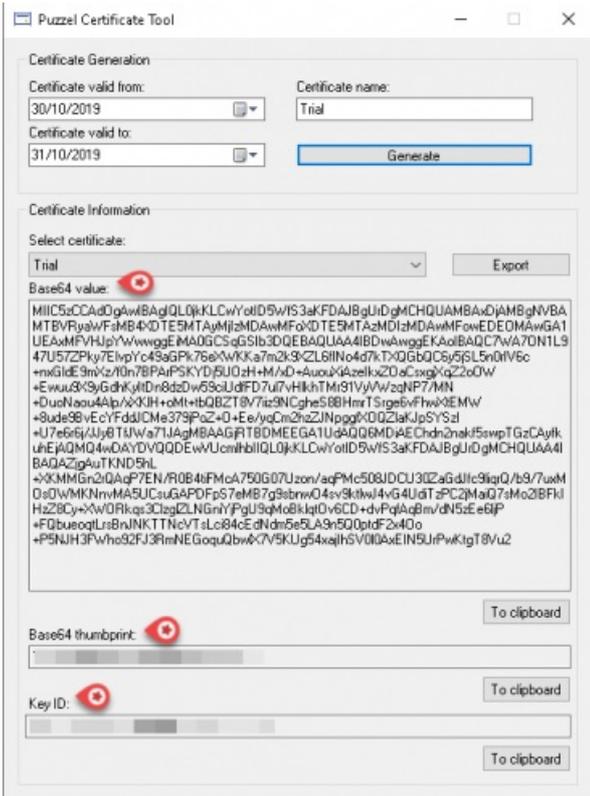
```

```

keyCredentials": [
  {
    "customKeyIdentifier": "<Base64Thumbprint >",
    "keyId": "<keyid>",
    "type": "AsymmetricX509Cert",
    "usage": "Verify",
    "value": "<base64Value>"
  }
]

```

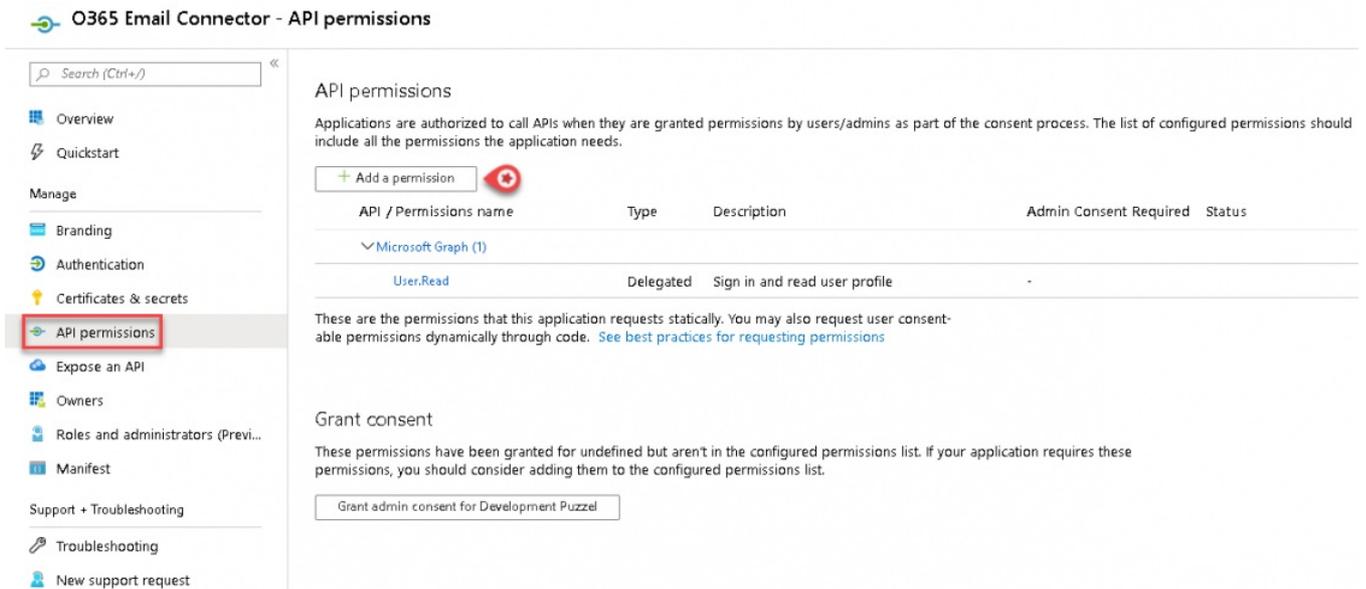
The values to the "customKeyIdentifier", "keyId", "value" can be found in the certificate generated using the Puzzel certificate tool.



Set up permissions

To set up the permissions for syncing the emails, do the following:

1. Go to <https://portal.azure.com/>, and open Azure Active Directory -> App registrations and select the application you just created.
2. Select **API permissions** and click on **Add a permission** button



3. Select **Office 365 exchange Online** option under Api's my organization uses tab and select Application permissions

Request API permissions

Select an API

[Microsoft APIs](#)
APIs my organization uses
[My APIs](#)

Apps in your directory that expose APIs are shown below

Name	Application (client) ID
Microsoft Forms	c9a559d2-7aab-4f13-a6ed-e7e9c52aec87
Microsoft.MileIQ.Dashboard	f7069a8d-9edc-4300-b365-ae53c9627fc4
Skype Presence Service	1e70cd27-4707-4589-8ec5-9bd20c472a46
Azure DevOps	499b84ac-1321-427f-aa17-267ca6975798
Windows Notification Service	04436913-cf0d-4d2a-9cc6-2ffe7f1d3d1c
DeploymentScheduler	8bbf8725-b3ca-4468-a217-7c8da873186e
Skype for Business	7557eb47-c689-4224-abc-f-ae9bd7573df
Microsoft Service Trust	d6fdaa33-e821-4211-83d0-cf74736489e1
Office 365 Exchange Online 	00000002-0000-0ff1-ce00-000000000000
Office 365 Management APIs	c5393580-f805-4401-95e8-94b7a6ef2fc2
Microsoft StaffHub	aa580612-c342-4ace-9055-8edee43ccb89

4. Select **Mail.Read Option** and Click on **Add permission.**

Request API permissions

[< All APIs](#)

 Exchange
<https://outlook.office365.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 without a signed-in user.

Select permissions [expand all](#)

Type to search

Permission	Admin Consent Required
<input type="checkbox"/> full_access_as_app Use Exchange Web Services with full access to all mailboxes ⓘ	Yes
<ul style="list-style-type: none"> > Calendars > Contacts > Mailbox > MailboxSettings 	
<ul style="list-style-type: none"> <div style="border: 1px solid #ccc; padding: 5px; display: inline-block;"> * <input checked="" type="checkbox"/> Mail.Read Read mail in all mailboxes ⓘ </div> Yes <input type="checkbox"/> Mail.ReadWrite Read and write mail in all mailboxes ⓘ Yes <input type="checkbox"/> Mail.Send Send mail as any user ⓘ Yes 	

* Add permissions Discard

5. The following screen will appear prompting you to grant consent as an Admin. Click on the button shown in the picture below and select **Yes** to confirm

 You are adding permission(s) to your application, users will have to consent even if they've already done so previously.

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)



API / Permissions name	Type	Description	Admin Consent Requir...	Status
<input type="button" value="Exchange (1)"/>				...
Mail.Read	Application	Read mail in all mailboxes	Yes	 Not granted for Develop... ...
<input type="button" value="Microsoft Graph (1)"/>				...
User.Read	Delegated	Sign in and read user profile	-	...

6. The application will now have the permissions configured.

 Successfully granted admin consent for the requested permissions.

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

API / Permissions name	Type	Description	Admin Consent Requir...	Status
<input type="button" value="Exchange (1)"/>				...
Mail.Read	Application	Read mail in all mailboxes	Yes	 Granted for Developme... ...
<input type="button" value="Microsoft Graph (1)"/>				...
User.Read	Delegated	Sign in and read user profile	-	 Granted for Developme... ...