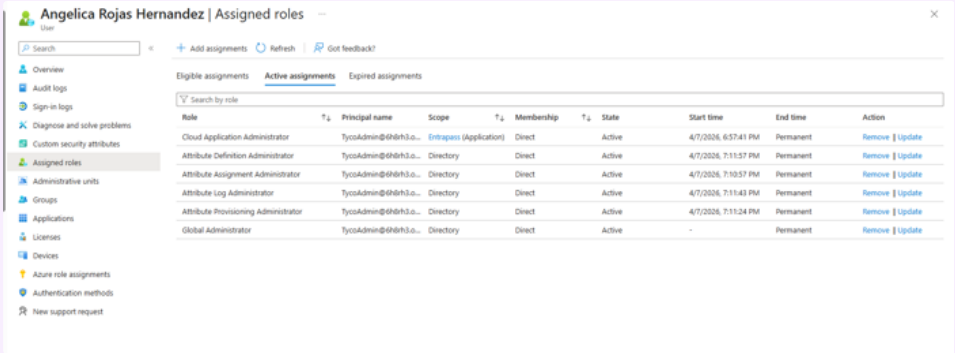
App Registration with Microsoft EntraID

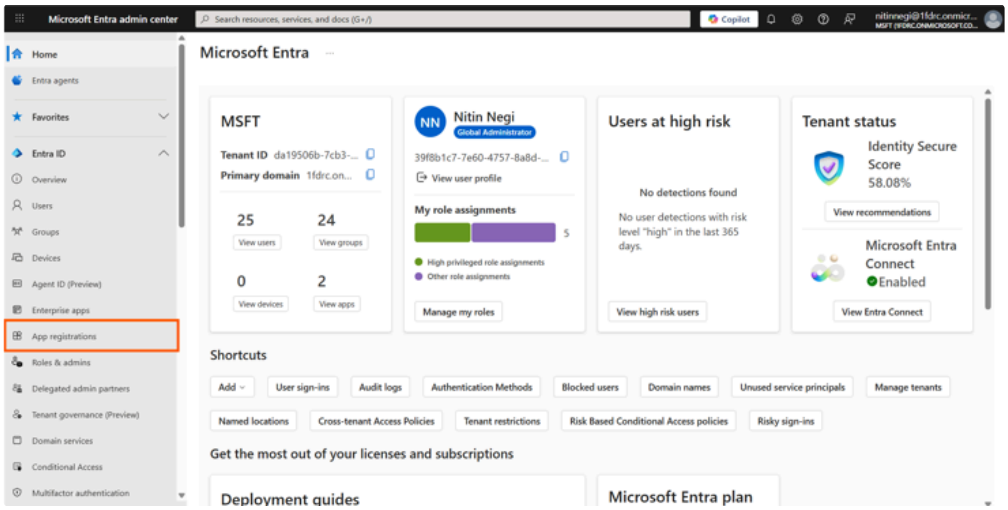
Before registering the app, ensure the EntraID user has these permissions



Permission
Attirbute Definition Administrator
Attribute Assignement Administrator
Attribute Log Administrator
Attribute Provisioning Administrator
Global Administrator
Cloud Application Administrator

✅ Step 1: Sign in to Microsoft Entra Admin Center

1. Open a browser and go to: `<https://entra.microsoft.com>`
2. Sign in with an account that has **Application Administrator** or **Global Administrator** role assigned.
   Refer to the images for the exact steps.



✅ **Step 2: Navigate to App registrations**

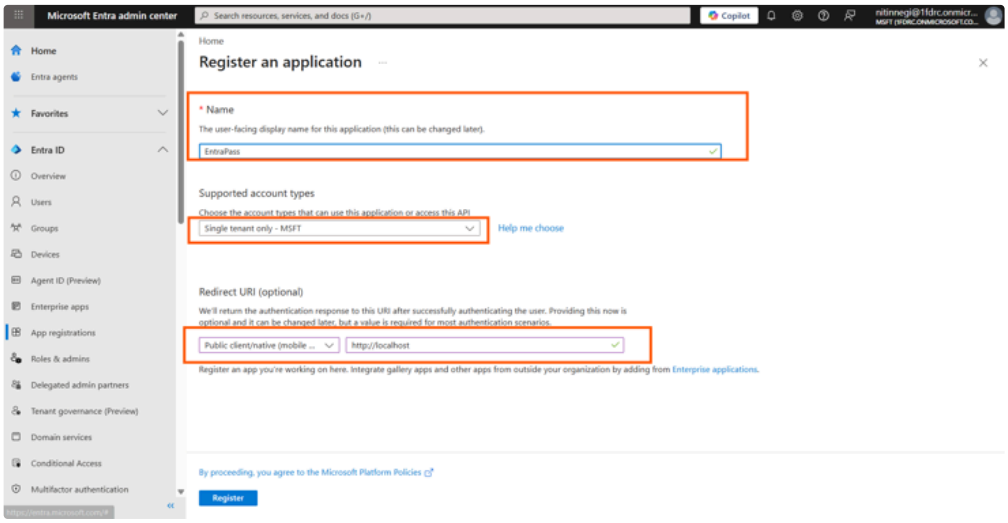
1. In the left navigation, select: **Identity → Applications → App registrations** *(this should match the left-menu screenshot on the page)*
2. On the **App registrations** screen, click **New registration**..Refer mentioned image in Step 1

✅ **Step 3: Configure application details**

On the **Register an application** blade:

1. **Name**
   - Enter a clear, descriptive name for the application (for example: `Entrapass Web - Prod`).
2. **Supported account types** – choose one based on your use case:
   - **Accounts in this organizational directory only (Single tenant)** – only users from your tenant can sign in.
   - **Accounts in any organizational directory (Multitenant)** – users from any Azure AD / Entra ID tenant.
   - **Personal Microsoft accounts** – outlook.com / hotmail.com / xbox Live, etc.
3. **Redirect URI**
   - For native/mobile apps: use the platform-specific custom URI (as in the screenshots, if provided).
4. Click **Register**.

   Refer to the images for the exact steps.



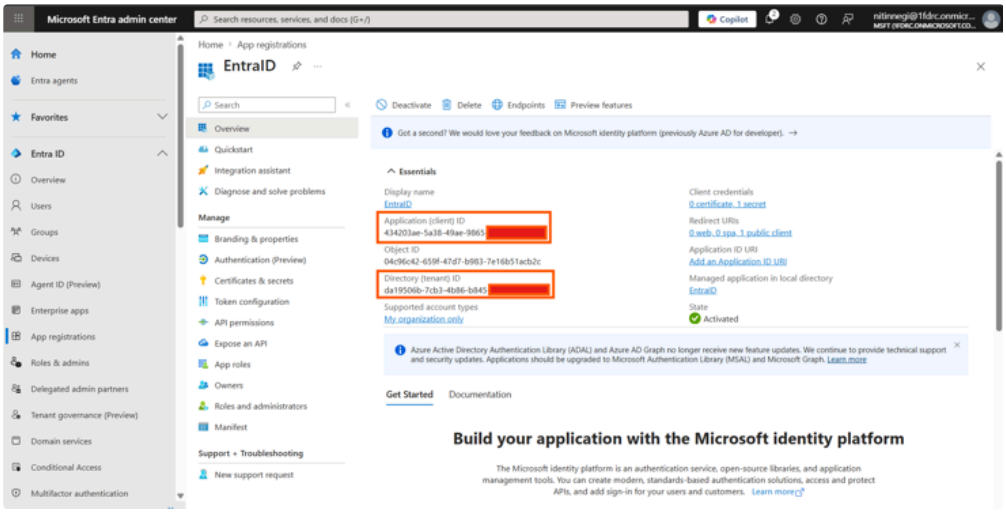
✅ **Step 4: Note down app details**

After registration, you'll land on the app's **Overview** page. From here, copy and keep the following values (they're used by your app and for API calls):

- **Application (client) ID**
- **Directory (tenant) ID**

Store them in a secure location (e.g., Key Vault, password manager, or your app configuration).

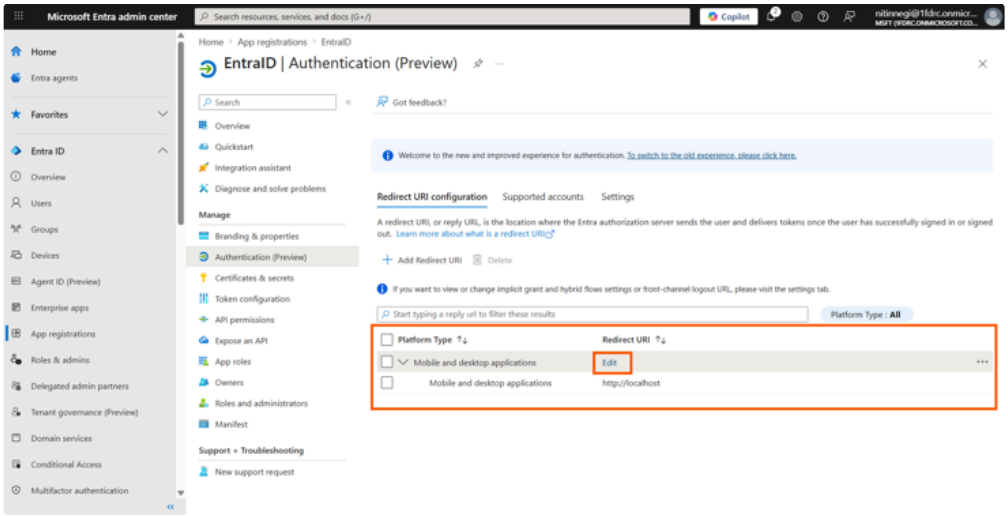
Refer to the images for the exact steps.

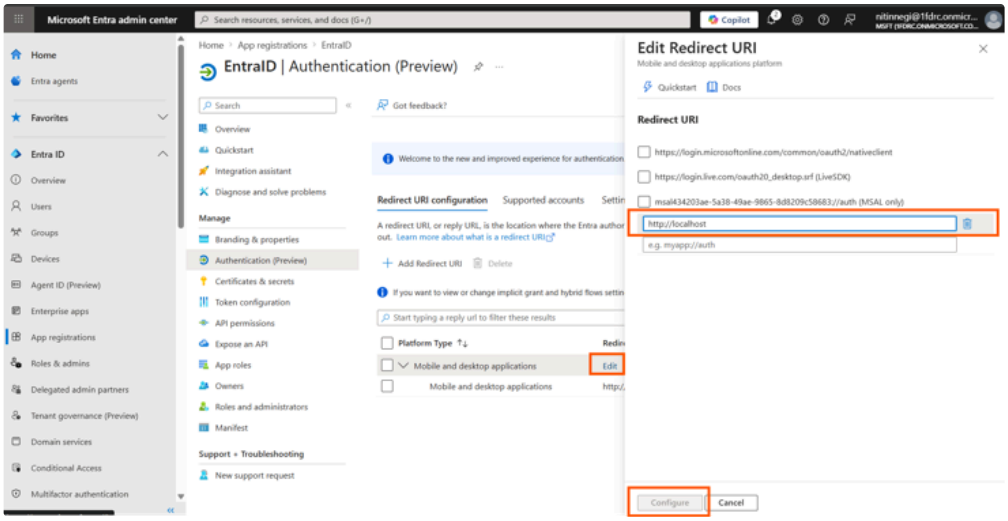


✅ **Step 5: Configure authentication**

1. In the left menu of the app, select **Authentication**.
2. Under **Platform configurations**, add or edit your platform:
   - **Mobile and desktop applications** – add the redirect URIs shown in the screenshots for Android/iOS if applicable.
3. Click **Save**.

   Refer to the images for the exact steps.





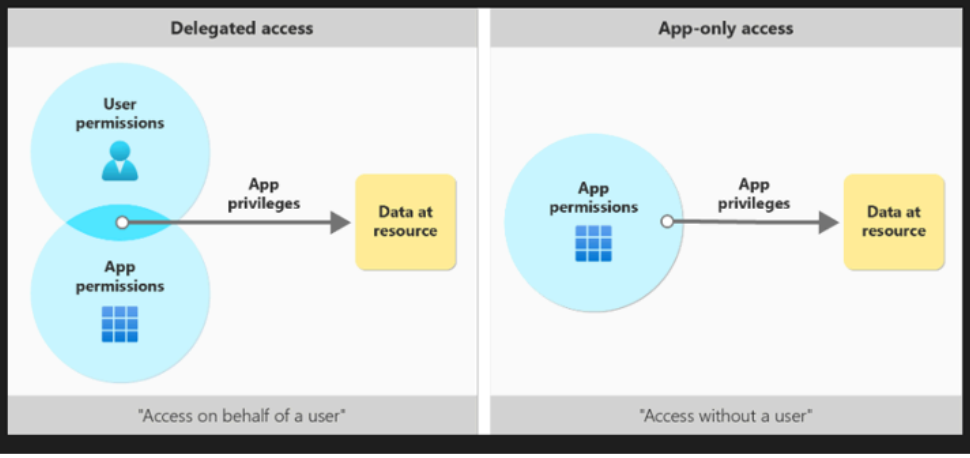
✅ **Step 6: Add API permissions**

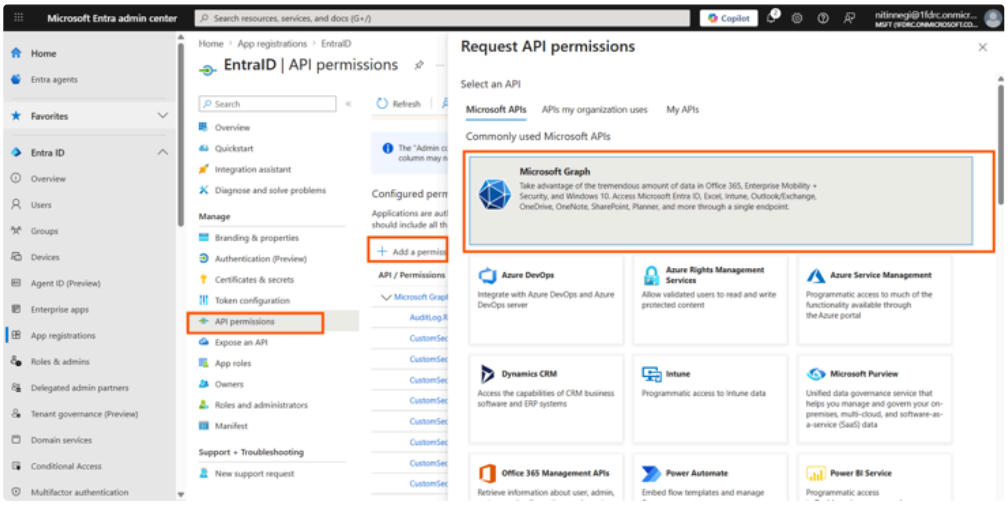
1. In the left menu, select **API permissions**.
2. Click **Add a permission**.
3. Choose **Microsoft Graph** (or another API as per your solution).
4. Select the permission type:
   - **Delegated permissions** – when a signed-in user is involved.
   - **Application permissions** – for daemon / background services. (Preferred)
5. From the list, select the required permissions (for example: `User.Read`, or others as per image below).
6. Click **Add permissions**.
7. The permissions require admin consent, click **Grant admin consent** for the tenant and confirm.

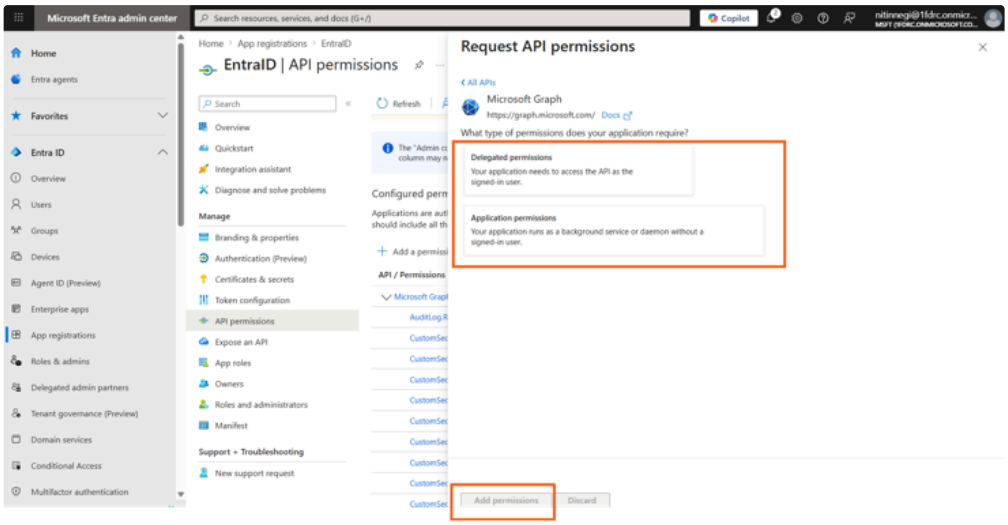
Refer to the images for the exact steps.

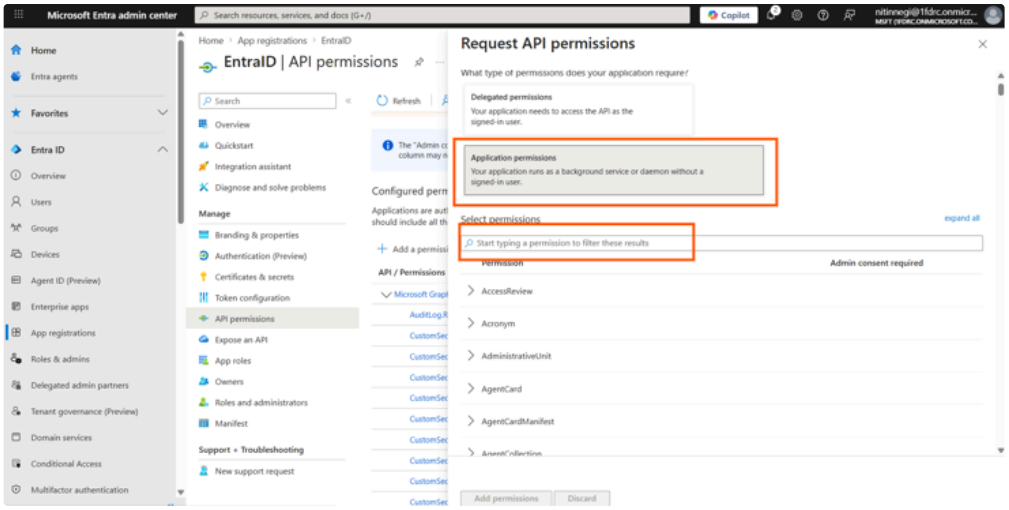
Access scenarios

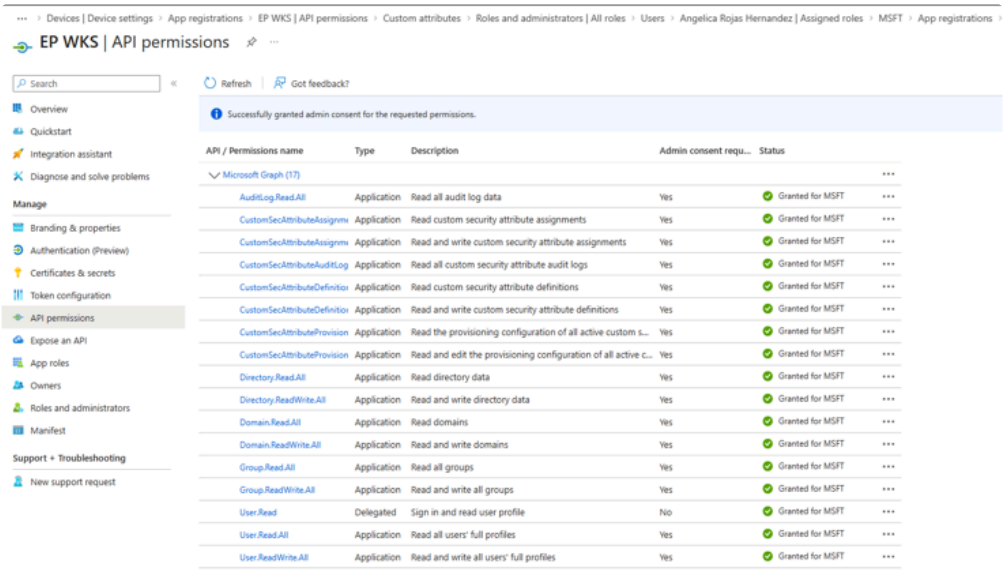
As an application developer, you must identify how your application accesses data. The application can use delegated access, acting on behalf of a signed-in user, or app-only access, acting only as the application's own identity.







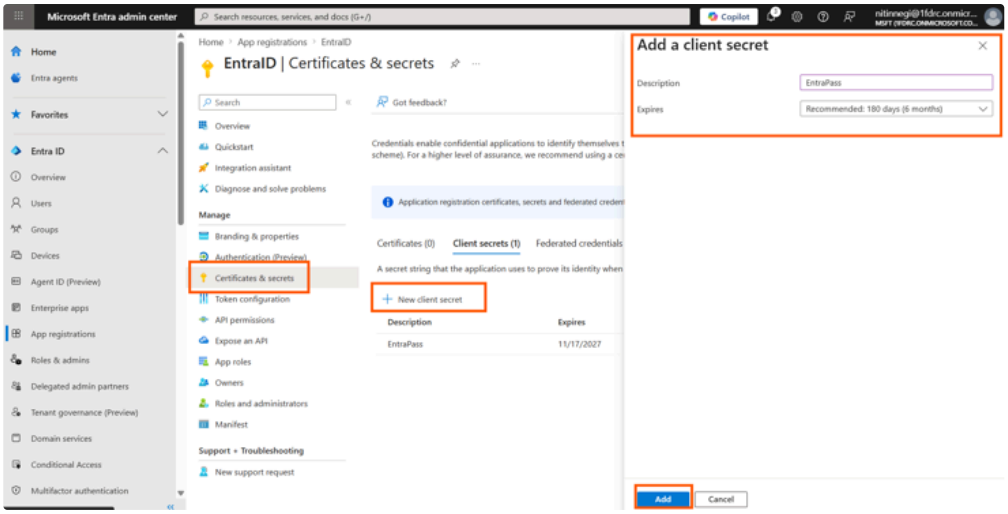


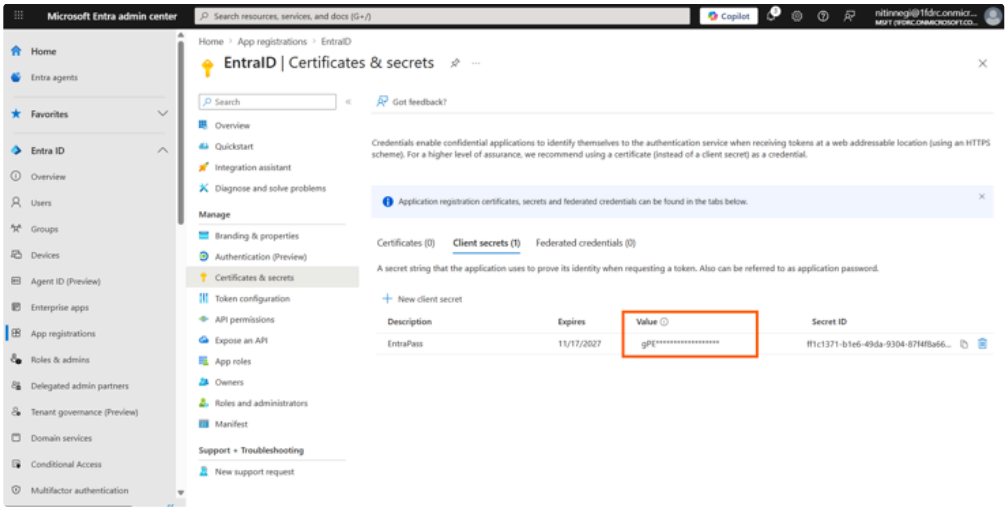


✅ **Step 7: Create client secret (or certificate)**

1. In the left menu, select **Certificates & secrets**.
2. Under **Client secrets**, click **New client secret**.
3. Enter a **Description** and choose an **Expires** duration.
4. Click **Add**.
5. Immediately copy the **Value** of the client secret and store it securely – you won't be able to see it again after you leave this page.

Refer to the images for the exact steps.





✅ **Step 8: Test the app registration**

Use Postman or your application code to validate the token flow. For a typical OAuth 2.0 client-credentials or authorization-code flow:

- **Token endpoint**: `<https://login.microsoftonline.com/{tenant-id}/oauth2/v2.0/token>`
- Use the following in your request:
  - **client_id**: Application (client) ID from Step 4
  - **client_secret**: The secret created in Step 7
  - **scope** or **resource**: Depends on flow/API (e.g. `<https://graph.microsoft.com/.default>`)
  - **redirect_uri**: One of the redirect URIs configured in Step 5 (for auth-code flow)

Verify that you receive a valid **access token** and that it works with the configured API permissions.