# Generate JWT Token using **Microsoft Entra**

To generate a **JWT token** using **Microsoft Entra**, which can be used to authenticate API calls to a **Qualys application**, you need to follow the steps below.

This process involves registering your application in **Microsoft Entra**, obtaining the necessary credentials, and then using the **OAuth 2.0 Client Credentials flow** to get the token.

## 1. Set Up Microsoft Entra (Azure AD) Application Registration

You first need to register your application in **Microsoft Entra** (Azure Active Directory) to get credentials for generating a JWT token.

**Steps:**

1. **Sign into the Microsoft Entra Admin Center**:
   - Go to https://entra.microsoft.com or https://portal.azure.com.
2. **Register a New Application**:
   - Navigate to **Azure Active Directory** > **App registrations**.
   - Click on **New registration**.
   - Provide a **name** for your app.
   - Choose **Supported account types** (usually **Accounts in this organizational directory only** or **Accounts in any organizational directory**).
   - You can leave **Redirect URI** empty for now (or set it if required for user-based auth).
   - Click **Register** to complete the registration.
3. **Note the Application (Client) ID** and **Directory (Tenant) ID**:
   - After registering the app, make sure to note the **Application (client) ID** and **Directory (tenant) ID** for later use in generating the token.

## 2. Create Client Secret or Certificate

You need to create a **client secret** or use a **certificate** to authenticate the client when generating the JWT token.

**Steps:**

1. In your **App registration** screen, go to **Certificates & secrets**.
2. Under **Client secrets**, click **New client secret**.
3. Provide a **description** and set an expiration for the secret.
4. Click **Add**, and copy the **client secret** value (it will be shown only once).

## 3. Assign API Permissions

You need to grant the application permission to access the APIs that allow you to authenticate 3rd-party services. Depending on the 3rd-party application's API, you may need to use **OAuth 2.0** authorization.

For example, if you're accessing Microsoft Graph or any other external API via OAuth:

1. Go to **API Permissions** > **Add permission**.

2. Choose **APIs my organization uses** or **Microsoft Graph** (or any other 3rd-party API).
3. Select the permissions **delegated.**
   In Azure AD, delegated permissions are used when a signed-in user is present, and the app is acting on behalf of that user.
4. Grant user-specific delegated permissions like User.Read and User.ReadWrite.
5. If required, click **Grant admin consent** for the permissions.

## 4. Generate JWT Token Using OAuth 2.0 Client Credentials Flow

Now, you will generate the JWT token via the **Client Credentials Flow**. This flow allows you to authenticate an app without user interaction, making it suitable for backend-to-backend communication.

**Steps:**

Make a **POST** request to the Microsoft Entra token endpoint:

- **URL**: https://login.microsoftonline.com/{tenantId}/oauth2/v2.0/token
- **Headers**:
  - Content-Type: application/x-www-form-urlencoded
- **Request Body**:

- client_id={clientId}&
- client_secret={clientSecret}&
- scope={scope}&
- grant_type=client_credentials
    - Replace {tenantId}, {clientId}, and {clientSecret} with the actual values from your app registration.
    - **Scope**: This is the resource you want to access. Typically, it will be the URL of the 3rd-party API (for example, for Microsoft Graph: https://graph.microsoft.com/.default).
    - **grant_type**: Set to client_credentials.

**Example of cURL Command:**

```
curl -X POST https://login.microsoftonline.com/{tenantId}/oauth2/v2.0/token \
 -H "Content-Type: application/x-www-form-urlencoded" \
 -d "client_id={clientId}" \
 -d "client_secret={clientSecret}" \
 -d "scope={https://graph.microsoft.com/.default}" \
 -d "grant_type=client_credentials"
```

**Parameter Breakdown:**

| Parameter | Description |
|---|---|
| tenantId | Replace with your **Azure AD tenant ID** |
| client_id | The **Application (client) ID** of your registered app in Azure AD. |
| client_secret | The **client secret** you created under **Certificates & secrets** for the app. |
| scope | For more information, refer to https://learn.microsoft.com/en-us/entra/identity-platform/scopes-oidc (This is only for reference) |
| grant_type | Always set to `client_credentials` for this flow (no user context). |

**5. Response**

The API will return an access token (JWT token) in the response.

Example Response:

```
{
  "token_type": "Bearer",
  "expires_in": 3600,
  "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Ik..."
}
```

- The access_token is the **JWT** that you will use to authenticate API requests to the 3rd-party application.

**6. Use the JWT Token for API Calls to the 3rd-Party Application**

Once you have the JWT token, you can authenticate API requests to the **3rd-party application** by adding the token in the Authorization header as a **Bearer token**.

**Example of API Call to 3rd-Party Application:**

```
GET https://thirdpartyapi.example.com/endpoint

Authorization: Bearer {access_token}
```

**7. Additional Considerations**

- **Token Expiry**: JWT tokens usually expire after a certain period (e.g., one hour). You may need to refresh the token or request a new one periodically.
- **Scope**: Ensure that the correct scopes are assigned for the 3rd-party API you are integrating with.
- **Error Handling**: Handle token expiry or invalid tokens in your application to avoid failed authentication.

**Summary**

1. **Register an app** in Microsoft Entra.
2. **Create a client secret** or certificate.
3. **Assign permissions** for the app to access the required API.
4. **Use the OAuth 2.0 Client Credentials Flow** to generate a JWT token.

5. **Use the JWT token** in the Authorization header of API calls to the 3rd-party application.

By following these steps, you can generate a JWT token using Microsoft Entra and use it to authenticate API calls to 3rd-party applications.

**References:**

JWKS URI: https://login.microsoftonline.com/common/discovery/v2.0/keys

Access Tokens in MS Identity Platform: https://learn.microsoft.com/en-us/entra/identity-platform/access-tokens