api0

GUIDE 06

# Connecter des services Azure à api0 : Azure DevOps, SharePoint, Azure SQL et Logic Apps

Comment exposer des données métier, des contenus et des processus Azure via api0, uniquement avec des endpoints d'API : au nom de qui part chaque appel, comment régler chaque composant Microsoft, et les erreurs que vous rencontrerez.

Mis à jour le 11 novembre 2026 · Lire en ligne : api0.ai/fr/guides/connecter-services-azure

---

api0 appelle des endpoints HTTP. Il n'y a aucun connecteur Azure à installer : pour chaque service, vous déclarez ses endpoints dans api0 et vous indiquez comment api0 obtient le jeton qui les signe.

La première décision est de savoir au nom de qui part chaque appel. Ce guide recommande le compte Microsoft de chaque personne partout où le service connaît les personnes, et une identité de service seulement là où il ne les connaît pas.

| Repère | Ce que c'est | Où le trouver |
|---|---|---|
| `<tenant-id>` | L'ID de votre annuaire Microsoft | Centre d'administration Entra → **Vue d'ensemble** → *ID de locataire* |
| `<client-id>` | L'ID de votre application Entra | **Inscriptions d'applications** → votre application → *ID d'application (client)* |
| `<site-id>` | L'ID Graph du site SharePoint | `GET https://graph.microsoft.com/v1.0/sites/<hôte>:/sites/<nom>?$select=id` |

## 1. Au nom de qui part chaque appel ?

| | Chaque personne se connecte | Une identité de service |
|---|---|---|
| Mode api0 | `per_user_oauth` | `client_credentials` |
| Type de permission Microsoft | Déléguée | Application |
| Ce que voit le service | La personne qui demande | L'application |
| Ce que la personne peut atteindre | Seulement ce que son compte peut atteindre | Tout ce que l'application peut atteindre |
| Traces dans Azure | Le nom de chaque personne | Le nom de l'application |
| Première utilisation | Chacun clique une fois sur **Connect** | Rien |

Utilisez la connexion de chaque personne pour Azure DevOps, SharePoint, OneDrive, Outlook et tout ce qui passe par Microsoft Graph. Avec une identité de service, toute personne qui parle à l'assistant agit avec les droits de l'application, et un nom écrit dans une demande n'est que du texte.

Gardez une identité de service seulement quand le service ne connaît pas les personnes, ou quand tout le monde doit voir la même chose : une base de reporting en lecture seule, ou un Logic App qui exécute un processus avec sa propre identité.

Dans Teams, chaque personne a toujours sa propre conversation individuelle et son propre historique avec le bot, quel que soit le mode. Le mode décide seulement avec quels droits partent les appels.

## 2. Le service a-t-il une API HTTP ?

| Service | API HTTP | Ce qu'il faut ajouter |
|---|---|---|
| Azure DevOps | Oui | Rien |
| SharePoint | Oui, Microsoft Graph | Rien |
| Logic App | Oui, son déclencheur Request | Rien |
| Azure SQL | Non, il parle TDS | Une API devant, comme Data API builder |

Si un service n'a pas d'API HTTP, api0 ne peut pas l'atteindre directement : mettez d'abord une API devant.

## 3. Créer une connexion api0 par audience de jeton

Un jeton Microsoft n'est valable que pour une seule API, son audience. Un jeton pour Microsoft Graph est refusé par votre propre API, et inversement. Il faut donc une connexion par audience, même avec une seule application Entra.

Sur app.api0.ai, ouvrez la carte **Connections** de la page **API**, puis cliquez sur **Add a connection**, ou sur **Edit** pour une connexion existante.

### 3.1 Chaque personne se connecte

| Champ | Valeur |
|---|---|
| **Name** | Un nom court, par exemple `graph` |
| **Sign in as** | Each person signs in (OAuth) |
| **Authorize URL** | `https://login.microsoftonline.com/<tenant-id>/oauth2/v2.0/authorize` |
| **Token URL** | `https://login.microsoftonline.com/<tenant-id>/oauth2/v2.0/token` |
| **Client ID** | `<client-id>` |
| **Client secret** | La Valeur du secret |
| **Scope** | Voir le tableau ci-dessous |

| Service appelé | Scope |
|---|---|
| Azure DevOps | `499b84ac-1321-427f-aa17-267ca6975798/user_impersonation offline_access` |
| Microsoft Graph, SharePoint en lecture et écriture | `https://graph.microsoft.com/Sites.ReadWrite.All offline_access` |
| Microsoft Graph, SharePoint en lecture seule | `https://graph.microsoft.com/Sites.Read.All offline_access` |
| Votre propre API ou Logic App | `api://<client-id>/access_as_user offline_access` |

Nommez chaque permission déléguée et ajoutez `offline_access`, pour qu'api0 renouvelle l'accès sans redemander de connexion. `https://graph.microsoft.com/.default offline_access` fonctionne aussi : il demande toutes les permissions Graph déléguées accordées à l'application.

L'enregistrement remplace toute la connexion, et api0 ne renvoie jamais le secret au navigateur : saisissez de nouveau le secret à chaque modification.

### 3.2 Une identité de service

| Champ | Valeur |
|---|---|
| **Name** | Un nom court, par exemple `demo-app` |
| **Sign in as** | One service identity (client credentials) |
| **Token URL** | `https://login.microsoftonline.com/<tenant-id>/oauth2/v2.0/token` |
| **Client ID** | `<client-id>` |
| **Client secret** | La Valeur du secret |
| **Scope** | Voir le tableau ci-dessous |

| Service appelé | Scope |
|---|---|
| Microsoft Graph | `https://graph.microsoft.com/.default` |
| Votre propre API ou Logic App | `api://<client-id>/.default` |
| Azure Resource Manager | `https://management.azure.com/.default` |

Le scope doit être une seule valeur se terminant par `/.default`. Le flux client credentials utilise les permissions d'application déjà accordées à l'application.

## 4. Les composants Microsoft, un par un

Chaque section dit à quoi sert le composant, comment le régler et comment le vérifier.

### 4.1 L'application Entra

Une seule application peut servir toutes les connexions. Elle porte les permissions, le secret et, pour la connexion par personne, l'adresse vers laquelle Microsoft renvoie les personnes.

1. Dans le centre d'administration Entra, ouvrez **Inscriptions d'applications** → **Nouvelle inscription**, nommez l'application et inscrivez-la.
2. Ouvrez **Certificats et secrets** → **Nouveau secret client**. Copiez la *Valeur* du secret, pas son *ID de secret*. Un nouveau secret peut mettre quelques minutes à être accepté. Saisissez-le uniquement sur app.api0.ai.
3. Pour la connexion par personne, ouvrez **Authentification** → **Ajouter une plateforme** → **Web** et ajoutez l'URI de redirection `https://gateway.api0.ai/oauth/downstream/callback`. Laissez *Jetons d'accès* et *Jetons d'ID* décochés.
4. Ouvrez **Autorisations de l'API** → **Ajouter une autorisation** et ajoutez ce dont chaque service a besoin : des **Autorisations déléguées** pour la connexion par personne, des **Autorisations d'application** pour une identité de service. Les sections 4.2 à 4.5 les listent.
5. Cliquez sur **Accorder un consentement d'administrateur pour** . Chaque ligne doit afficher un état vert *Accordé*.

Le consentement administrateur compte aussi pour les permissions déléguées. Beaucoup d'annuaires ne laissent pas les personnes consentir elles-mêmes aux applications : sans le consentement de l'administrateur, chacun voit *Need admin approval* au moment de Connect.

Vérification : dans **Autorisations de l'API**, chaque permission affiche *Accordé*, avec le type voulu, *Déléguée* ou *Application*.

Le secret expire au bout d'un ou deux ans. Créez alors un nouveau secret et mettez à jour chaque connexion api0 qui l'utilise.

### 4.2 Azure DevOps

Azure DevOps connaît chaque personne : utilisez la connexion par personne, pour que chaque work item soit créé à son nom et avec ses droits.

1. Dans l'application Entra, ajoutez la permission **Azure DevOps** → `user_impersonation`, en **Autorisations déléguées**, et accordez le consentement administrateur.
2. L'organisation Azure DevOps doit être rattachée au même annuaire que l'application : **Organization settings** → **Microsoft Entra**.
3. Dans api0, importez les endpoints Azure DevOps après avoir remplacé chaque `YOUR-ORGANIZATION` par le nom de votre organisation, et reliez-les à une connexion de la section 3.1 avec le scope `499b84ac-1321-427f-aa17-267ca6975798/user_impersonation offline_access`.
4. Chaque personne ouvre une fois `https://dev.azure.com/<organisation>` dans un navigateur : Azure DevOps refuse les appels d'API d'un compte qui ne s'y est jamais connecté.

Ne cliquez jamais sur **Connect** avec un compte qui a un rôle d'administrateur : Microsoft exige alors une authentification multifacteur que le renouvellement en arrière-plan ne peut pas fournir.

Le pas-à-pas complet, avec l'assistant d'api0 et chaque erreur rencontrée en chemin, est dans le guide 05 : connexion Microsoft de chacun.

Vérification : une personne clique sur **Connect**, puis demande à créer une issue dans Claude ou Teams. Dans Azure DevOps, le work item porte son nom dans **Created By**.

### 4.3 SharePoint via Microsoft Graph

1. Dans l'application Entra, ajoutez la permission Microsoft Graph `Sites.ReadWrite.All`, ou `Sites.Read.All` pour la lecture seule. Choisissez **Autorisations déléguées** pour la connexion par personne, le choix recommandé, ou **Autorisations d'application** pour une identité de service. Avec des permissions d'application, `Sites.Selected` limite l'accès à certains sites.
2. Accordez le consentement administrateur.
3. Dans api0, utilisez l'URL de base `https://graph.microsoft.com/v1.0/sites/<site-id>` et la connexion de la section 3.1.

Avec la connexion par personne, Graph applique les droits SharePoint de chacun. Donnez aux personnes l'accès au site, et limitez qui peut modifier une liste quand une modification vaut décision, comme une colonne d'approbation : dans la liste, ouvrez **Paramètres** → **Paramètres de la liste** → **Autorisations pour cette liste**, arrêtez l'héritage des autorisations et ne donnez *Modification* qu'aux personnes qui décident.

Exemples d'endpoints :

| Usage | Verbe | Chemin |
|---|---|---|
| Lire une liste | GET | `/lists/<liste>/items?$expand=fields&$select=id` |
| Modifier un élément | PATCH | `/lists/<liste>/items/{item_id}/fields` |
| Chercher des documents | GET | `/drive/root/search(q='{query}')` |
| Lire un document texte | GET | `/drive/items/{item_id}/content` |

Nommez un paramètre `$filter` pour que l'agent puisse filtrer une liste. Seules les colonnes indexées se filtrent de façon fiable : dans **Paramètres de la liste** → **Colonnes indexées**, indexez les colonnes sur lesquelles on filtre.

Vérification : une personne qui a accès au site clique sur **Connect**, puis demande une liste dans Teams ou Claude.

### 4.4 Azure SQL via Data API builder

Data API builder transforme des tables et des vues en endpoints REST avec `$filter`, `$orderby` et `$first`.

1. Créez la base. L'offre gratuite d'Azure SQL suffit pour une démo : réglez-la pour qu'elle s'arrête à sa limite gratuite, pas pour facturer au-delà.
2. Exécutez Data API builder là où il peut joindre la base, par exemple dans une Azure Container App, avec la chaîne de connexion de la base en secret. Autorisez les services Azure dans le pare-feu SQL.
3. Dans l'application Entra, ouvrez **Exposer une API** et définissez l'URI d'ID d'application sur `api://<client-id>`.
4. Pour une identité de service, ouvrez **Rôles d'application** et créez un rôle dont le type de membre autorisé est *Applications*, puis dans **Autorisations de l'API** → **Mes API**, ajoutez ce rôle à

l'application et accordez le consentement administrateur. Pour la connexion par personne, ajoutez plutôt une étendue nommée `access_as_user` dans **Exposer une API**.

5. Dans **Manifeste**, réglez la version du jeton d'accès sur 2.
6. Dans `dab-config.json`, utilisez le fournisseur `EntraId`, l'audience `<client-id>` et l'émetteur `https://login.microsoftonline.com/<tenant-id>/v2.0`.
7. Ne donnez au rôle `authenticated` que les actions voulues. La lecture seule est un bon choix par défaut.

Une base de reporting partagée et en lecture seule est un bon usage d'une identité de service.

La base, et une Container App réduite à zéro, peuvent s'endormir sans activité. Attendez-vous à un premier appel lent.

Vérification : demandez les lignes d'une table dans Claude ou Teams. Si Data API builder refuse, voyez le récapitulatif des erreurs.

### 4.5 Un Logic App via son déclencheur Request

1. Utilisez un Logic App Consumption avec un déclencheur **When an HTTP request is received**, et terminez-le par une action **Response**.
2. Dans **Paramètres** → **Autorisation**, ajoutez une stratégie avec l'émetteur `https://login.microsoftonline.com/<tenant-id>/v2.0` et l'audience `<client-id>`.
3. Désactivez SAS sur le déclencheur, pour que l'URL ne porte aucune signature.
4. Dans api0, utilisez l'URL du déclencheur jusqu'à `/paths`, avec le chemin `/invoke?api-version=2016-10-01`.
5. Si le Logic App appelle une autre API, donnez-lui une identité managée et accordez à cette identité la permission de l'API.

L'application demande les mêmes réglages qu'aux étapes 3 à 5 de la section 4.4, sans Data API builder.

Un Logic App agit avec sa propre identité managée, quelle que soit la personne qui l'appelle. Si la personne compte, par exemple pour savoir qui a fait une demande, envoyez-la dans la requête et vérifiez-la dans le workflow : ne vous fiez pas à un nom que la personne tape.

Vérification : ouvrez l'**historique des exécutions** du Logic App après un appel. Chaque appel y est une exécution, avec son entrée et le résultat de chaque action.

### 4.6 Le bot Teams

Le guide Teams (guide 07) couvre la licence Teams, l'Azure Bot, sa connexion dans api0 et l'installation de l'application. Une fois qu'il fonctionne pour une API, il fonctionne pour toutes les API du tenant : rien ne change dans Teams quand vous ajoutez Azure DevOps, SharePoint, SQL ou un Logic App.

Chaque personne lie une fois son compte Teams à son compte api0, en envoyant au bot le code affiché sur la page de liaison. Le bot ne répond que dans les conversations individuelles.

## 5. Relier chaque endpoint à sa connexion

Importez vos endpoints sur la page **API**. Puis, dans la carte **Connections**, utilisez **Which connection each tool uses** pour choisir la bonne connexion de chacun. Un endpoint qui ne désigne aucune connexion utilise `default`.

Réimporter un groupe peut remettre ce choix à zéro : vérifiez-le après chaque import.

## 6. Vérifier et essayer

Demandez `setup_status` à l'assistant de configuration avec le nom de la connexion. Il signale les réglages manquants, un scope qui ne correspond pas à l'API appelée ou au mode, et les connexions qu'aucun endpoint n'utilise.

Pour une connexion où chaque personne se connecte, chacun ouvre ensuite **Settings → My access tokens** sur app.api0.ai et clique sur **Connect** à côté d'elle. Il faut être connecté à app.api0.ai avec le même compte api0 que celui lié à Teams : l'accès est rangé sous ce compte.

Posez ensuite une question en lecture seule dans Claude ou Teams. api0 ne voit pas les réglages Azure : si un appel échoue, donnez le message exact à `explain_error`.

## Récapitulatif des erreurs

| Message | Cause | Correctif |
|---|---|---|
| `Need admin approval` au moment de Connect, ou `AADSTS90094` / `AADSTS65001` | Pas de consentement administrateur pour les permissions déléguées, et les personnes ne peuvent pas consentir elles-mêmes | Un administrateur ouvre **Autorisations de l'API** et clique sur **Accorder un consentement d'administrateur**, jusqu'à ce que chaque ligne affiche *Accordé*, puis la personne clique de nouveau sur Connect |
| `your access is not set up, or has expired` renvoyé par un outil | La personne n'a pas cliqué sur Connect, ou s'est connectée depuis un autre compte api0 que celui lié à Teams | Ouvrez Settings → My access tokens avec le compte api0 lié à Teams, et cliquez sur Connect |

| Message | Cause | Correctif |
| --- | --- | --- |
| `AADSTS50011` redirect URI mismatch | L'application n'a pas l'URI de redirection d'api0 | Ajoutez `https://gateway.api0.ai/oauth/downstream/callback` dans **Authentification** → **Web** |
| `AADSTS7000215` invalid client secret | L'ID du secret a été collé, ou le secret est tout récent | Attendez quelques minutes, puis recollez la Valeur du secret |
| `AADSTS7000222` client secret keys are expired | Le secret a expiré | Créez un nouveau secret et mettez à jour chaque connexion qui l'utilise |
| `AADSTS1002012` client credential flows must have a scope value with `/.default` | Le scope d'une identité de service ne se termine pas par `/.default` | Utilisez un scope de la section 3.2 |
| `AADSTS500011` resource principal not found | Le scope désigne une API inconnue de cet annuaire | Définissez l'URI d'ID d'application et vérifiez le `<client-id>` du scope |
| `Either scp or roles claim need to be present in the token` | L'application n'a aucune permission sur Microsoft Graph | Ajoutez la permission et accordez le consentement administrateur |
| `accessDenied` ou `403` de Microsoft Graph avec la connexion par personne | La personne n'a pas accès à ce site ou à cette liste | Donnez-lui l'accès dans SharePoint |
| `Access Not Allowed` `AuthorizationCheckFailed` de Data API builder | Pas de jeton valide, ou un rôle sans cette action | Reliez l'endpoint à la bonne connexion, puis vérifiez l'audience, l'émetteur et la version 2 du jeton |
| `Field cannot be referenced in filter or orderby as it is not indexed` | La colonne SharePoint n'a pas d'index | Indexez la colonne, ou filtrez sur une colonne indexée |

| Message | Cause | Correctif |
| --- | --- | --- |
| `The server did not receive a response from an upstream server` d'un Logic App | Le workflow a échoué avant son action Response | Ouvrez l'**historique des exécutions** du Logic App et lisez l'action en échec |
| Logic App qui refuse un appel avec à la fois une signature et un jeton bearer | SAS et OAuth ont été utilisés ensemble | Désactivez SAS sur le déclencheur et retirez `sp`, `sv` et `sig` de l'URL |
| `The selected region is currently not accepting new customers` | Cette région est fermée à votre abonnement | Choisissez une autre région. Un groupe de ressources reste dans sa région, créez-en donc un nouveau |
| `Client with IP address is not allowed to access the server` | Le pare-feu SQL bloque cette adresse | Ajoutez une règle de pare-feu. Derrière un proxy, l'adresse peut varier : autorisez temporairement sa plage |
| `Database is not currently available` retry the connection later | La base serverless se réveille | Réessayez après une minute |