

Connecter Microsoft 365 (Outlook, OneDrive, Teams, Excel)

Donner à ton IA un accès complet à toute la suite Microsoft (mail Outlook, calendrier, OneDrive, SharePoint, Teams, Excel, OneNote, Tasks) via le MCP `ms-365-mcp-server` (90+ tools sur la Graph API officielle). Setup en 2 minutes avec `/connect-mcp`, compte perso ou Entra ID d'entreprise.

Objectif

Si ton tenant exige une app interne, ton admin doit créer une app registration Entra ID (type "Mobile and desktop applications"), te fournir `CLIENT_ID` et `TENANT_ID`, et accorder les permissions Graph dont tu as besoin (`Mail.ReadWrite`, `Calendars.ReadWrite`, `Files.ReadWrite`, etc.). Tu les passes en variables d'env au serveur.

Connecter Microsoft 365 (Outlook, OneDrive, Teams, Excel, SharePoint)

Si ton travail tourne autour de la suite Microsoft (mail Outlook, fichiers OneDrive, réunions Teams, fichiers SharePoint, Excel, OneNote, Tasks/Planner), tu veux que ton IA puisse y accéder directement. Pas via du copier-coller, pas via des exports manuels. Directement, comme elle accède à ton coffre Obsidian.

C'est exactement ce que fait le MCP **Microsoft 365 (ms-365-mcp-server)** : un MCP open source qui parle à toute la suite Microsoft via la Graph API officielle. 90+ tools, un seul plugin, et ton IA gère tes mails, tes events, tes fichiers et tes feuilles Excel pour toi.

`ms-365-mcp-server` (Softeria)

A Model Context Protocol server for Microsoft 365 and Office services

Open source, gratuit, 90+ tools sur toute la suite Microsoft 365 via la Graph API officielle.
Compte perso (outlook.com, hotmail) ou compte d'entreprise (Entra ID).

github.com/Softeria/ms-365-mcp-server

Ce que ça débloquent concrètement

OpenCode (ou Claude Code)

Ton IA, dans son interface habituelle

ms-365-mcp-server

Le pont vers Microsoft Graph API

Outlook

Mail + Calendar

OneDrive

Fichiers

Teams

Chats + meetings

SharePoint

Sites + drives

Excel

Lecture + écriture

OneNote

Pages + sections

Tasks

To Do + Planner

Contacts

Annuaire

C'est exactement la contrepartie Microsoft de la leçon [Connecter Google Workspace](#). Si tu vis dans l'écosystème Microsoft, c'est ce MCP que tu installes.

Cas d'usage typiques

Une fois branché, ton IA peut faire ce genre de choses sans que tu sortes d'OpenCode :

Mail Outlook

"Résume mes 20 derniers mails non lus, regroupe par sujet, et propose un brouillon de réponse pour les 3 urgents."

L'IA lit ton inbox, classifie, drafte des réponses, tu valides.

OneDrive / SharePoint

"Trouve le dernier brief client X dans OneDrive, lis-le, et rédige un suivi pour notre call de demain."

L'IA cherche le fichier, l'ouvre, en tire la synthèse, prépare le call.

Calendar + Teams

"Cale une réunion Teams de 30 min avec Sophie et Marc cette semaine, créneau qui marche pour tous, et envoie l'invitation."

L'IA croise les calendriers, crée le meeting, envoie les invits.

Excel

"Ouvre mon Excel de pipeline commercial, ajoute les 5 nouveaux leads de cette semaine, et envoie-moi la nouvelle prévision de CA."

Lecture + écriture dans tes feuilles, recalcul, restitution.

Installation : 3 chemins, du plus simple au plus contrôlé

Option A : via le skill /connect-mcp (recommandé)

C'est le chemin le plus simple, parce que le skill du bootcamp connaît déjà ce MCP.

Dans OpenCode (ou Claude Code), tape simplement :

```
/connect-mcp microsoft 365
```

Le skill va :

1. Ajouter @softeria/ms-365-mcp-server à ton opencode.json
2. Te demander si tu utilises ton compte Microsoft perso ou un compte Entra ID d'entreprise
3. Lancer l'auth Device Code Flow : un code à coller sur microsoft.com/devicelogin depuis ton navigateur
4. Vérifier que les tools sont bien chargés

C'est plié en 2 minutes. Passe à la section "Tester" plus bas.

Option B : commande Claude / OpenCode directe

Si tu veux le faire à la main sans passer par le skill, une seule commande dans ton terminal :

```
# macOS / Linux
claude mcp add ms365 -- npx -y @softeria/ms-365-mcp-server

# Windows
claude mcp add ms365 -s user -- cmd /c "npx -y @softeria/ms-365-
mcp-server"
```

Puis dans Claude Code (ou OpenCode), tape `/mcp` pour vérifier que `ms365` est listé. Au premier appel d'un tool, tu seras invité à te connecter via Device Code Flow.

Option C : configuration manuelle dans `opencode.json`

Pour ceux qui veulent maîtriser exactement le contenu de leur config :

```
{
  "mcp": {
    "ms365": {
      "type": "local",
      "command": ["npx", "-y", "@softeria/ms-365-mcp-server"],
      "enabled": true
    }
  }
}
```

Pour limiter à certains scopes (ex : juste mail + calendar) :

```
{
  "mcp": {
    "ms365": {
      "type": "local",
      "command": ["npx", "-y", "@softeria/ms-365-mcp-server", "--
enabled-tools", "mail,calendar"],
      "enabled": true
    }
  }
}
```

Presets disponibles : mail, calendar, files, personal, work, excel, contacts, tasks, onenote, search, users, all.

Compte Microsoft perso ou Entra ID d'entreprise ?

Compte perso (outlook.com, hotmail, live)

Aucune config admin

L'app cliente du MCP est déjà configurée côté Softeria. Tu lances l'install, tu acceptes le consentement sur le device login, et c'est en place. Idéal pour les indépendants et les usages perso.

Compte d'entreprise (Microsoft 365 / Entra ID)

Selon la politique IT

Si ton tenant exige une app interne, ton admin doit créer une **app registration Entra ID** (type "Mobile and desktop applications"), te fournir `CLIENT_ID` et `TENANT_ID`, et accorder les permissions Graph dont tu as besoin (Mail.ReadWrite, Calendars.ReadWrite, Files.ReadWrite, etc.). Tu les passes en variables d'env au serveur.

Si tu es dans un contexte pro et que tu n'es pas sûr, demande à ton équipe IT avant l'installation. C'est une bonne pratique de sécurité, et beaucoup d'entreprises veulent garder le contrôle sur les apps qui touchent à leur tenant.

Tester en 1 minute

Une fois connecté, lance OpenCode dans ton vault et essaie :

```
Donne-moi un résumé de mes 5 derniers mails reçus en boîte de réception.
```

```
Liste mes prochains rendez-vous Outlook pour cette semaine.
```

```
Cherche dans mon OneDrive le dernier fichier modifié contenant
```

"budget" et résume-le.

Si tu vois des résultats réels (et pas une excuse du type "je n'ai pas accès à"), c'est gagné. Sinon, demande à l'IA elle-même de débbugger : pourquoi le MCP ms365 ne répond pas ? Elle ouvrira la config et te dira ce qui manque.

Alternative officielle : Microsoft Work IQ

Microsoft a sorti ses propres MCP servers sous le nom **Work IQ** (anciennement Agent 365). C'est de la solution officielle Microsoft, intégrée directement avec Copilot Studio et Azure AI Foundry.

Quand le choisir : tu as déjà une **licence Microsoft 365 Copilot** active sur ton compte (entreprise principalement), et tu veux le support et la garantie Microsoft sur le canal officiel.

Quand garder Softeria : tu n'as pas de licence Copilot (cas le plus courant pour les indépendants et la majorité des entreprises), tu veux 200+ tools couvrant tout le Graph, et tu veux rester open source.

Plus d'infos : github.com/microsoft/work-iq et [Microsoft Learn - Work IQ MCP overview](#).

Bonnes pratiques

Active le mode read-only au démarrage

Au premier setup, lance le serveur avec `--read-only` pour explorer sans risque d'écriture accidentelle. Tu désactives ce flag quand tu es confiant.

Limite les scopes à ce dont tu te sers

Si tu n'utilises jamais Excel ni OneNote via l'IA, ne charge pas ces tools (`--enabled-tools mail,calendar,files`). Ton IA est plus rapide et plus précise quand son menu d'outils est court.

Pense à la confidentialité

Le MCP donne à ton IA un accès à des données sensibles (mails clients, documents internes, calendrier perso). Avant de connecter, vérifie ton modèle : si tu passes par un modèle non ZDR (ChatGPT Plus, Claude.ai), les contenus de tes mails peuvent transiter par leur infra.

Pour ce type d'usage, route plutôt via OpenRouter en mode Privacy avec un modèle open source (DeepSeek V4 Flash, GLM 5.2). Cf. [Zoom sur la sécurité](#).

- ☐ J'ai installé le MCP via /connect-mcp microsoft 365 (ou la commande claude mcp add)
- ☐ Je me suis authentifié via Device Code Flow
- ☐ J'ai testé une lecture de mail et une requête calendrier dans OpenCode
- ☐ J'ai limité mes scopes aux services que j'utilise vraiment
- ☐ Je sais que pour des données sensibles, je route via OpenRouter ZDR plutôt que ChatGPT Plus