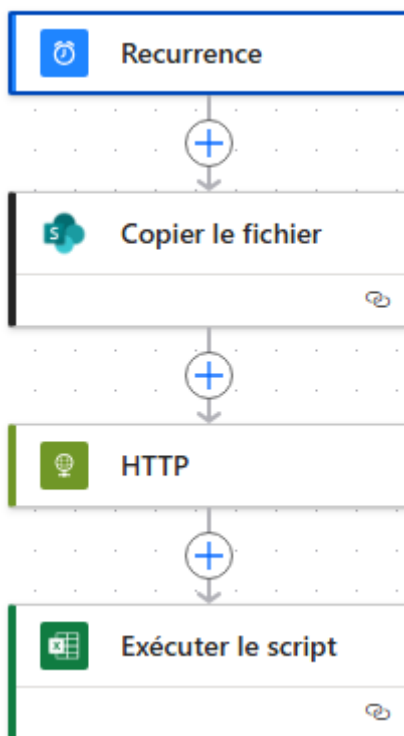


Contexte : Lors de mon entretien d'embauche on m'a tout de suite dit que je serai amené à créer une API pour récupérer des données de Odoo et les envoyer dans Microsoft 365, dans un Excel dans un SharePoint précisément. On souhaite récupérer la liste de nos adhérents inscrites dans Odoo pour les envoyer dans l'environnement 365 pour plusieurs raisons. La première y avoir accès même si on n'est pas sur Odoo (pas tout le monde n'y a accès) mais aussi pour l'utiliser pour des applis powerapps notamment PAPIE !


Après plusieurs semaines de recherches / travail j'ai trouvé une manière plus simple qu'une API avec FASTAPI Python comme je penser faire au début, un powerautomate.

Voici le power automate complet



Analysons en détail chacun des modules.

Le premier module 'Récurrence'

 Recurrence ⋮ <<

Paramètres Configuration Vue Code À propos

ScheduleConnector.RecurrenceParameterLabel *

Intervalle * * Fréquence * *

Fuseau horaire

Heure de début


À ces heures-ci

À ces minutes-ci

Version préliminaire
Exécute à 20:00 tous les jours

Il permet simplement d'effectuer toutes les actions à venir automatiquement à 20h00 tous les jours.

Le deuxième module 'Copier le fichier'

 Copier le fichier ⋮ ⏪

Paramètres Configuration Vue Code Test À propos


Adresse du site actuel *

Fichier à copier *

Adresse du site de destination *

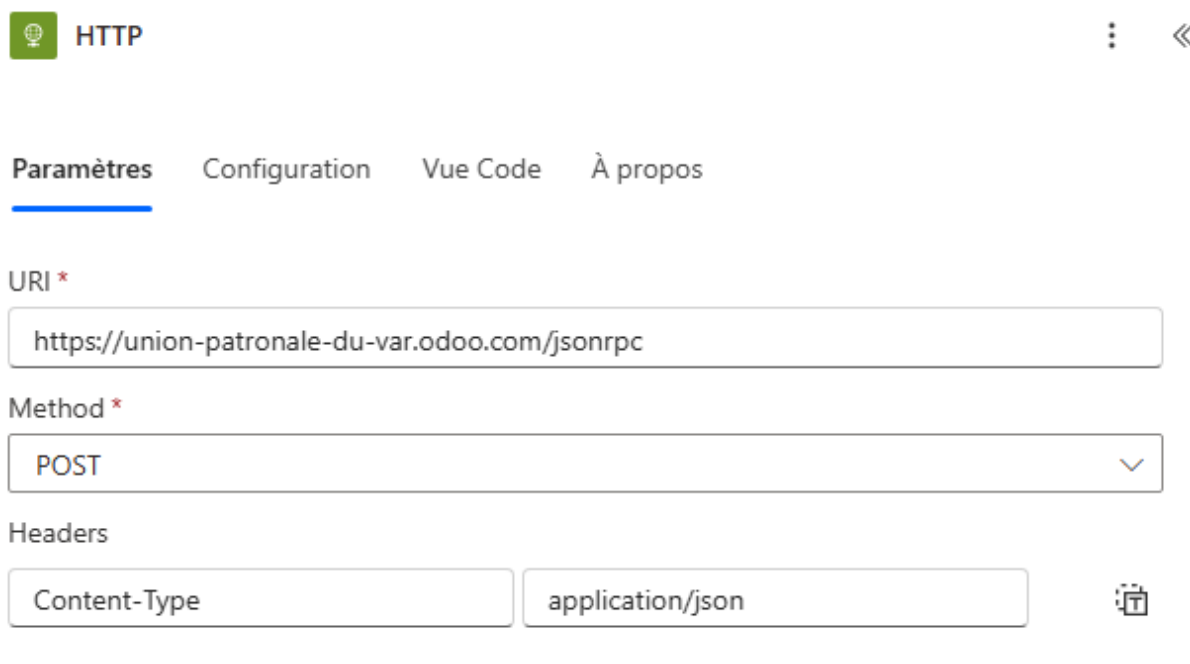
Dossier de destination *

Si un autre fichier existe déjà *

 Connecté à c.cour@upv.org. [Modifier la connexion](#)

Ce module récupère le fichier MODELE de base qui sera copier et re créer, ça permet de plus facilement mettre à jour les infos plutôt que remplacer lignes par lignes le fichier.

Le troisième module 'http'



The screenshot shows the configuration for an HTTP action in Power Automate. The URI is `https://union-patronale-du-var.odoo.com/jsonrpc`, the Method is `POST`, and the Content-Type header is `application/json`.

C'est LE module le plus important du power automate, on met l'URL de Odoo, et on utilise la méthode POST, contrairement à l'API qu'on devait faire le faire d'utiliser POST DEPUIS Odoo permet de ne pas avoir besoin de token Microsoft qui sont bien + exigeant que ceux de Odoo ! C'est le point fort qui m'a fais passer de fastAPI à powerautomate. M365 se connecte à Odoo et « s'envoie » les infos au lieu de récupérer !

```

1  {
2  "jsonrpc": "2.0",
3  "method": "call",
4  "params": {
5    "service": "object",
6    "method": "execute_kw",
7    "args": [
8      "NOM DE DB ODOO",
9      "ID USER ODOO",
10     "TOKEN API ODOO",
11     "res.partner",
12     "search_read",
13     [
14       [
15         [
16           "is_company",
17           "=",
18           true
19         ]
20       ]
21     ],
22     {
23       "fields": [
24         "display_name",
25         "city",
26         "x_studio_est_adherent",
27         "x_studio_ss",
28         "x_studio_reducbox_active",
29         "x_studio_date_adhesion",
30         "phone",
31         "mobile",
32         "email",
33         "x_studio_naf",
34         "x_studio_forme_juridique",
35         "x_studio_est_mandataire",
36       ],
37       "limit": 2000
38     }
39   ]
40 }
41 ]

```

Voici le JSON dans http ; dans un premier temps on récupère les infos de COUR Cyrille – cyrillecour11@gmail.com – Aristée 2025/2027

Odoo que j'ai masqué nom de DB id user et token api. (Pour récupérer ces infos sur Odoo il faut aller dans les droits d'accès en mode développeur d'Odoo) ensuite un met un filtre « is company » car sur Odoo nous avons des Sociétés et des Personnes, seules les sociétés nous intéressent. Pour finir les champs correspondant à ce que nous avons dans Odoo (les noms TECHNIQUES pas ceux des colonnes)

Nom Nom commercial Ville Adh. S. Soc... ReducBox Date d'adhési... Téléphone Mobile E-mail Territoire N... Forme juridiq... Mand.

Voici les noms des colonnes dans notre 'app' Contact sur Odoo, en mettant la souris sur 'nom'

- Libellé : Nom d'affichage
- Champ : display_name
- Modèle : res.partner
- Type : char
- Contexte : {}
- Lecture seule : True

On obtiens plusieurs infos dont 'champ' le nom technique à mettre dans le JSON

Enfin sur le Excel on créer un script office dans automatisaion

```

Scripts Office
← Détails du script
▶ Exécuter  Script enregistré  Renommer
ImportOdoo
1 function main(workbook: ExcelScript.Workbook, data: string) {
2   const parsed = JSON.parse(data);
3   const sheet = workbook.getActiveWorksheet();
4   const table = sheet.getTables()[0];
5
6   // Supprimer toutes les lignes du tableau
7   const rowCount = table.getRowCount();
8   if (rowCount > 0) {
9     table.getRangeBetweenHeaderAndTotal().delete(ExcelScript.DeleteShiftDirection.up);
10  }
11
12  const rows: string[][] = parsed.map((item: {
13    display_name: string, city: string,
14    x_studio_est_adherent: boolean, x_studio_ss: boolean,
15    x_studio_reducbox_active: boolean, x_studio_date_adhesion: string,
16    phone: string, mobile: string, email: string,
17    x_studio_naf: string[], x_studio_forme_juridique: string[],
18    x_studio_est_mandataire: boolean
19  }) => [
20    item.display_name || "",
21    item.city || "",
22    String(item.x_studio_est_adherent),
23    String(item.x_studio_ss),
24    String(item.x_studio_reducbox_active),
25    item.x_studio_date_adhesion || "",
26    item.phone || "",
27    item.mobile || "",
28    item.email || "",
29    item.x_studio_naf ? item.x_studio_naf[1] : "",
30    item.x_studio_forme_juridique ? item.x_studio_forme_juridique[1] : "",
31    String(item.x_studio_est_mandataire)
32  ]);
33
34  table.addRows(-1, rows);
35 }

```

On écrit un petit script en TypeScript, Ce script : Reçoit un JSON puis vide le tableau Excel existant puis transforme les objets JSON en lignes pour enfin insérer les données dans le tableau.

Pour finir le quatrième et dernier module 'Executer le script' :

Emplacement *

SharePoint Site - UPV_INFORMATIQUE

Bibliothèque de documents *

Documents

Fichier *

/Extract/Extract_Odoo.xlsx

Script *

ImportOdoo

ScriptParameters/data *

`string(...)`

Est le module qui permet d'utiliser le script du Excel.

ScriptParameters/data `string(body('HTTP')?['result'])`.

ET VOILA, notre powerautomate récupère nos infos sur Odoo en JSON, les convertis et les écrits dans un Excel dans un Sharepoint seul chaque soir !