

BTS SIO — Option SLAM

API ODOO -> M365

Extraction de données Odoo ERP vers l'environnement Microsoft 365 via Power Automate

Candidat	COUR Cyrille
Technologies	Odoo ERP / Power Automate / Office Scripts (TS) / Excel Online
Etablissement	Aristée
Période	Janvier 2026

1. Contexte et Objectifs du Projet

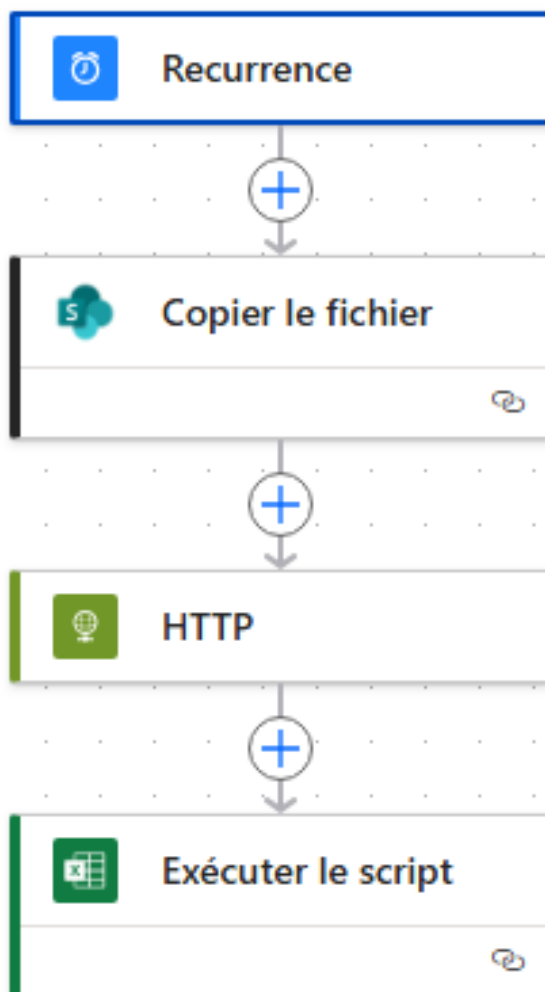
Dès mon intégration au sein de l'organisation, un besoin critique m'a été confié : concevoir un mécanisme d'interopérabilité pour extraire de manière automatisée la liste des adhérents enregistrés dans l'ERP Odoo et la synchroniser vers un espace de stockage SharePoint au format Excel Online.

Cette interconnexion répond à deux enjeux business majeurs :

- Garantir l'accès en lecture à la liste des adhérents pour les collaborateurs ne disposant pas de licence ou d'accès direct à l'ERP Odoo.
- Mettre à disposition une source de données persistante et structurée pour alimenter des applications métiers transverses développées sous Microsoft Power Apps, notamment l'application « PAPIE ».

□ **Pivot Architectural** : Initialement envisagée sous forme d'une API sur-mesure développée en Python avec le framework FastAPI, la solution a été réorientée après étude d'opportunité vers un flux Microsoft Power Automate. Cette approche réduit drastiquement la complexité d'infrastructure et s'affranchit de la gestion contraignante des tokens d'authentification Microsoft.

Ci-dessous, l'architecture globale et séquentielle du flux mis en œuvre :



2. Analyse Technique Détaillée des Modules

2.1 Module 1 : Planification de la tâche (Récurrence)

L'extraction doit s'exécuter de façon transparente en tâche de fond. Le premier module définit un déclencheur temporel (« Récurrence ») configuré pour lancer l'ensemble du traitement automatiquement tous les jours à 20h00, assurant la fraîcheur quotidienne des données pour le lendemain.

 Recurrence
 ⋮ ⏪

Paramètres Configuration Vue Code À propos

ScheduleConnector.RecurrenceParameterLabel *

Intervalle *

1

Fréquence *

Jour

Fuseau horaire

(UTC+01:00) Amsterdam, Berlin, Berne, Rome, Stockholm, Vienne

Heure de début

2026-02-11T21:00:00.000Z

À ces heures-ci

20

À ces minutes-ci

Entrez les valeurs de minute valides (de 0 à 59) séparées par des virgules, par exemple, 15,30

Version préliminaire

Exécute à 20:00 tous les jours

2.2 Module 2 : Initialisation du fichier de destination (Copier le fichier)

Afin d'éviter une maintenance lourde consistant à purger ou remplacer les lignes du fichier Excel existant une par une, une stratégie de réinitialisation par modèle a été choisie. Le flux va chercher un fichier « MODÈLE » propre et structurellement sain, puis crée une copie neuve à chaque exécution. Cela simplifie grandement la mise à jour des structures de données futures.

 Copier le fichier

Paramètres Configuration Vue Code Test À propos

Adresse du site actuel *
 UPV_INFORMATIQUE - https://e2cvarfr.sharepoint.com/sites/ServiceInformatique

Fichier à copier *
 /Documents partages/MODELE_Extract_Odoo.xlsx

Adresse du site de destination *
 UPV_INFORMATIQUE - https://e2cvarfr.sharepoint.com/sites/ServiceInformatique

Dossier de destination *
 /Documents partages/Extract

Si un autre fichier existe déjà *
 Replace

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

2.3 Module 3 : Interrogations de l'ERP via protocole HTTP

Il s'agit du cœur applicatif du flux. À l'aide du connecteur générique « HTTP », le flux émet une requête avec la méthode POST à destination de l'endpoint d'Odoo.

L'atout majeur de cette architecture est que l'appel est initié depuis Odoo vers l'environnement M365 (Odoo « pousse » l'information). Cela évite d'exposer ou de manipuler des jetons Microsoft hautement sécurisés et restrictifs.

Le corps de la requête HTTP intègre un schéma JSON contenant :

- Les identifiants de sécurité masqués : nom de la base de données, ID utilisateur et jeton d'API d'Odoo (récupérés dans les droits d'accès en mode développeur sur l'ERP).
- Un filtre logique 'is_company' pour extraire uniquement les entités de type « Société » et exclure les fiches « Personnes » de l'export.
- La déclaration des champs cibles en utilisant exclusivement leurs NOMS TECHNIQUES systèmes d'Odoo et non leurs labels d'affichage.



HTTP



Paramètres Configuration Vue Code À propos

URI *

Method *

Headers



❏ **Astuce Technique (Reverse Engineering Odoo) :** Pour identifier le nom technique exact d'une colonne sur Odoo (ex: pour le champ 'Nom'), il suffit de passer l'interface en mode développeur et de survoler le libellé avec la souris. Une info-bulle système apparaît alors, révélant la propriété 'champ' à reporter textuellement dans le JSON du flux.

```

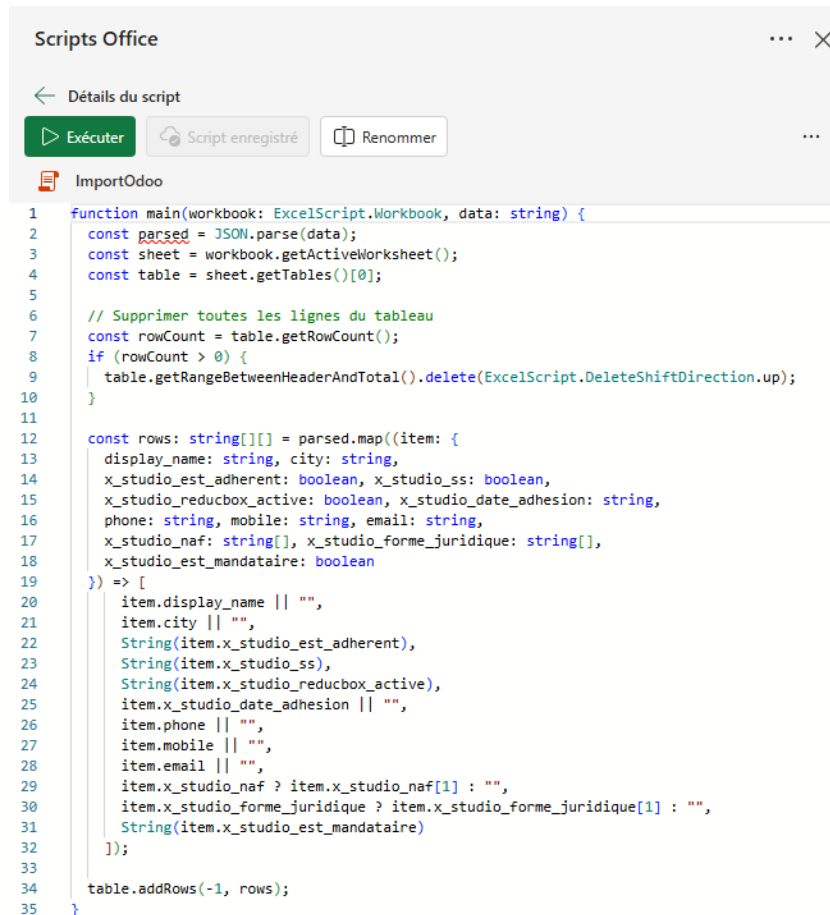
1 {
2   "jsonrpc": "2.0",
3   "method": "call",
4   "params": {
5     "service": "object",
6     "method": "execute_kw",
7     "args": [
8       "NOM DE DB ODOO",
9       "ID USER ODOO",
10      "TOKEN API ODOO",
11      "res.partner",
12      "search_read",
13      [
14        [
15          [
16            "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 }

```

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

3. Traitement de données et Injection Excel (Office Scripts)

Une fois le payload JSON réceptionné depuis Odoo, les données brutes doivent être écrites dans la copie du classeur Excel générée au module 2. Pour ce faire, le quatrième module du flux (« Exécuter le script ») fait appel à l'outil d'automatisation interne d'Excel Online : Office Scripts.



```

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     table.addRows(-1, rows);
34 }
35

```

Un script spécifique a été codé en TypeScript pour réaliser les opérations atomiques suivantes :

- Réceptionner l'objet JSON passé en argument par le module Power Automate.
- Purger l'intégralité du tableau Excel existant pour éviter les doublons ou résidus de la veille.
- Parser et transformer la collection d'objets JSON en une matrice de lignes et cellules standardisées.
- Injecter dynamiquement l'ensemble des nouvelles lignes au sein du tableau Excel structuré.

Emplacement *

SharePoint Site - UPV_INFORMATIQUE



Bibliothèque de documents *

Documents



Fichier *

/Extract/Extract_Odoo.xlsx



Script *

ImportOdoo



ScriptParameters/data *

 string(...) ×

4. Conclusion et Validation

Grâce à cette synergie entre Power Automate et Office Scripts, l'organisation dispose d'un pipeline de données robuste, asynchrone et totalement transparent. L'application Power Apps « PAPIE » consomme désormais une source Excel de confiance, mise à jour à J+1 sans aucune intervention humaine, garantissant ainsi l'intégrité des processus métiers.