Procédure pour l'utilisation des scripts Python et JavaScript dans le cadre de la migration des mots de passe de TPM vers Vaultwarden

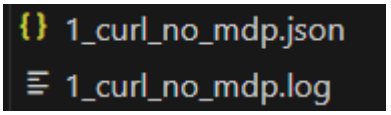
Contexte :

Dans le cadre de ma mission qui était de migrer une base de données entière de mots de passe de l'outil **TPM** (Team Password Manager) à l'outil **Vaultwarden**, j'ai dû m'aider de scripts **Python**, **JavaScript** et de l'**API** de TPM pour arriver à mes fins.

- ## Premier script : 1_curl_page.py

Ce script à pour objectif de répéter la commande **curl** récursivement afin de communiquer avec l'API de TPM et de récupérer les pages de TPM, il va également vous créer un fichier **.json** et un **fichier de log** pour voir le nombre de pages récupérer par le script.

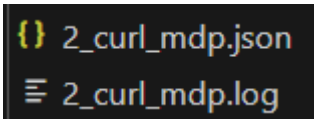
Sortie attendue :



- ## Deuxième script : 2_curl_mdp.py

Ce script là va reprendre le fichier json qui est sortie du premier script pour cette fois ci faire un **curl** récursif mais de toutes les entrées de mots de passe en prenant les ID qui sont sortie dans le premier fichier json.

Sortie attendue :



Cependant le problème de ce fichier json est qu'il va sortir avec ce champs la

```
"parents": [
    1180,
    1628,
    1646
],
```

et pour la futur migration des json vers les CSV KeePassXC pour importer les CSV dans Vaultwarden, pour la hiérarchie il nous faut des mots et non pas des numéro, ce que j'ai fait c'est un troisième script qui vient récupérer tous les projets avec leur noms et leurs ID pour pouvoir modifier les numéros avec le nom du projet en lien avec l'ID.

- **Troisième script : 3_curl_projects.py**

Ce script là va récupérer tous les projets de TPM avec leurs noms et leurs ID.

Sortie attendue :

Le json contient des entrées comme ça :

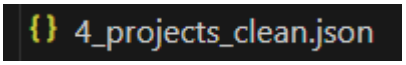

```
[
    {
        "id": 1743,
        "name": [redacted],
        "tags": "",
        "managed_by": {
            "id": 3,
            "username": '[redacted]',
            "email_address": [redacted],
            "name": '[redacted]',
            "role": "Project manager"
        },
        "archived": false,
        "favorite": false,
        "num_files": 0,
        "updated_on": "2025-11-26 13:51:44"
    },
```


Ce qui va être trop lourd et avec trop de détails pour le cinquième script, j'ai donc un quatrième script qui va nettoyer ce fichier json pour n'avoir que le champ du nom et de l'ID.

- **Quatrième script : 4_projects_clean.py**

Ce script là va prendre le fichier json précédent et va le nettoyer afin d'avoir que les noms et les ID des projets.

Sortie attendue :



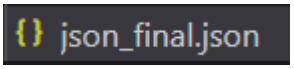
On a donc une sortie plus propre qu'avant :


```
[
    {
        "id": 1743,
        "name": [redacted]
    },
```

- **Cinquième script : 5_nouveau_champs_parents.py**

Ce script est très important car c'est celui qui va nous donner le fichier json final pour le convertir en CSV.

Sortie attendue :



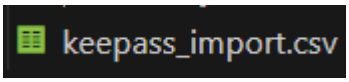
On a maintenant un fichier json avec tous les mots de passe en clair et des champs parents converti des ID aux noms des projets.

```
        "parents": [
            "INNLOG - Commun",
            "Environnement INTERNE",
            "Kubernetes"
        ],
```

- **Sixième script : 6_json_to_csv.py**

Ce script là va prendre le fichier json final est le convertir en fichier CSV compatible KeePassXC pour pouvoir ensuite l'importer dans Vaultwarden.

Sortie attendue :

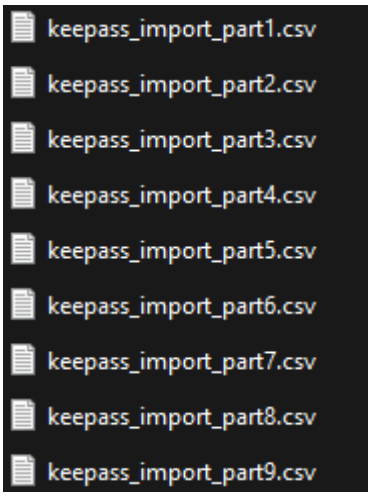


Cette fois ci nous avons un fichier CSV que l'on peut importer dans Vaultwarden, cependant Vaultwarden contient une protection anti-DoS qui ne me permet pas d'importer un fichier CSV avec des centaines voir des milliers de lignes il faut donc parser le fichier en plusieurs fichiers ce qui est le but du septième script.

- **Septième script : 7_parse_csv.py**

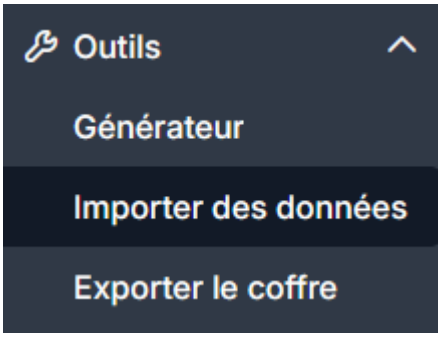
Ce script là va parser (séparer) le fichier CSV complet pour pouvoir l'importer en plusieurs fichiers.

Sortie attendue :

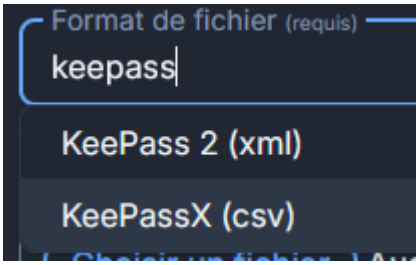


- **Import dans Vaultwarden (dossier personnel)**

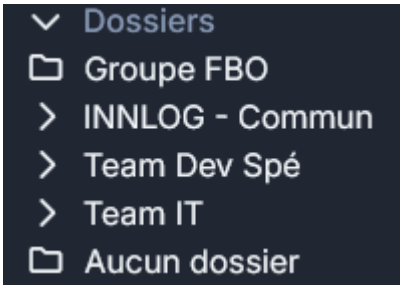
Une fois nos fichiers créés nous pouvons les importés dans Vaultwarden un par un. Il faudra aller dans l'interface web de Vaultwarden et une fois connecté allez dans le menu **Importer des données.**



Faire dérouler le menu **Format de fichier** et sélectionner **KeePassX (csv)**



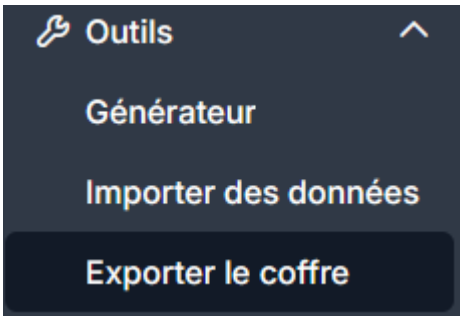
Après ça vous pourrez sélectionner vos fichier CSV et les importer vous finirez avec ça dans votre COFFRE PERSONNEL :

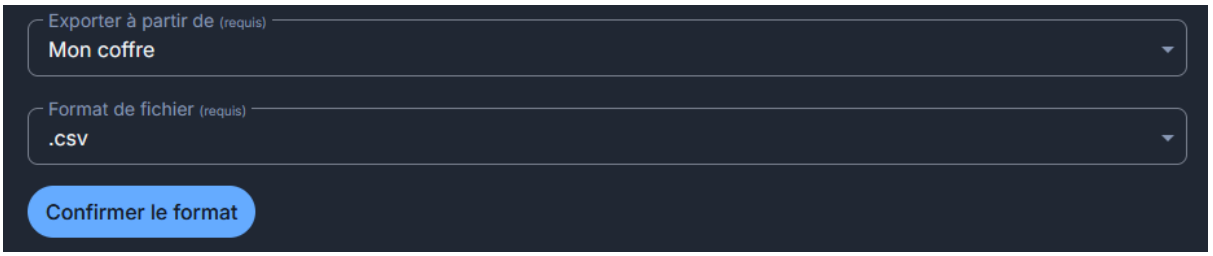


Une fois que vous avez ça dans votre coffre personnel, il faut le mettre à disposition de l'organisation Innlog.

- **Exportation du coffre Vaultwarden**

Le coffre personnel Vaultwarden doit être exporté pour la suite du processus





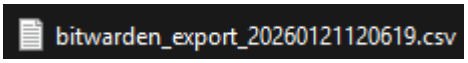
- **Création de l'organisation Innlog**

Il faudra créer l'organisation dans Innlog, vous devrez aller dans votre coffre personnel et cliquer sur **Nouvelle organisation.** Rentrez le nom de votre organisation et une adresse mail, ce mail sera le propriétaire de l'organisation, il pourra être modifié plus tard.

Une fois que je vous avez accès à votre organisation c'est là que le dernier script va nous être utile. Le fichier CSV exporté en amont de Vaultwarden va être le fichier d'entrée car la structure CSV n'est pas la même.

- **Huitième script : 8_CSV_dossier_to_CSV_collections.py**

Ce script là va prendre le fichier CSV qui a été exporté en amont. A l'export vous avez un fichier qui a cette allure là :



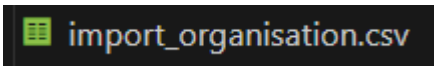
il faudra le renommer pour ensuite mettre le nouveau dans le script.

Comme ceci par exemple. Il faudra modifier le script ensuite pour qu'il prenne bien le

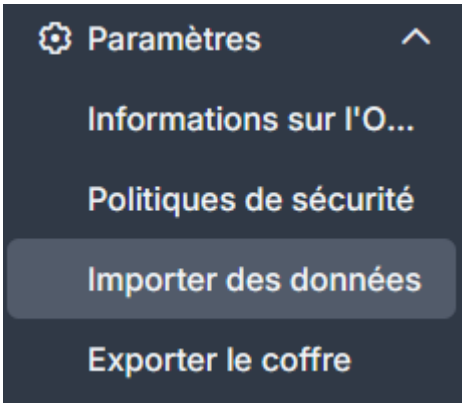
bon nom lors de son exécution.

```
input_file = 'export_vw.csv'        # To
output_file = 'import_organisation.csv'
```

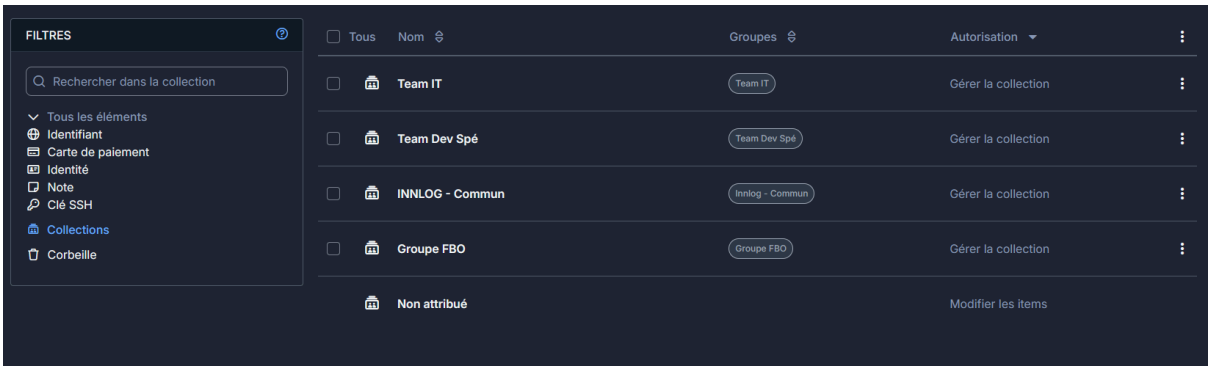
Sortie attendue :



Une fois ce fichier là dans votre dossier vous pourrez le réimporter dans votre organisation.



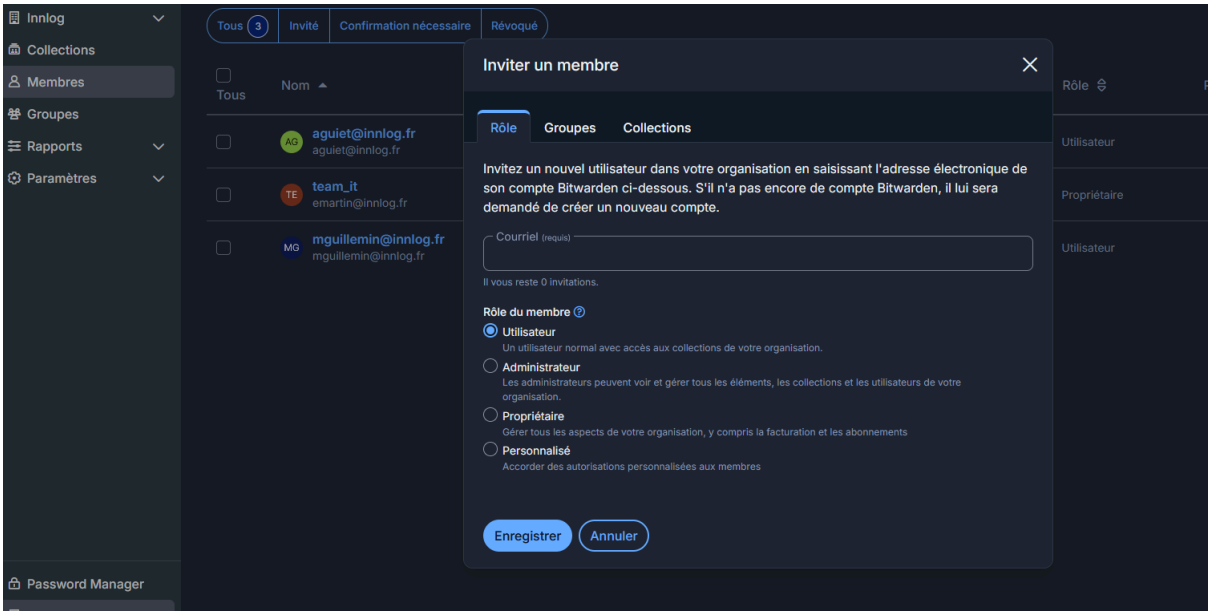
Si vous allez maintenant dans votre coffre d'organisation vous aurez quelque chose comme ça



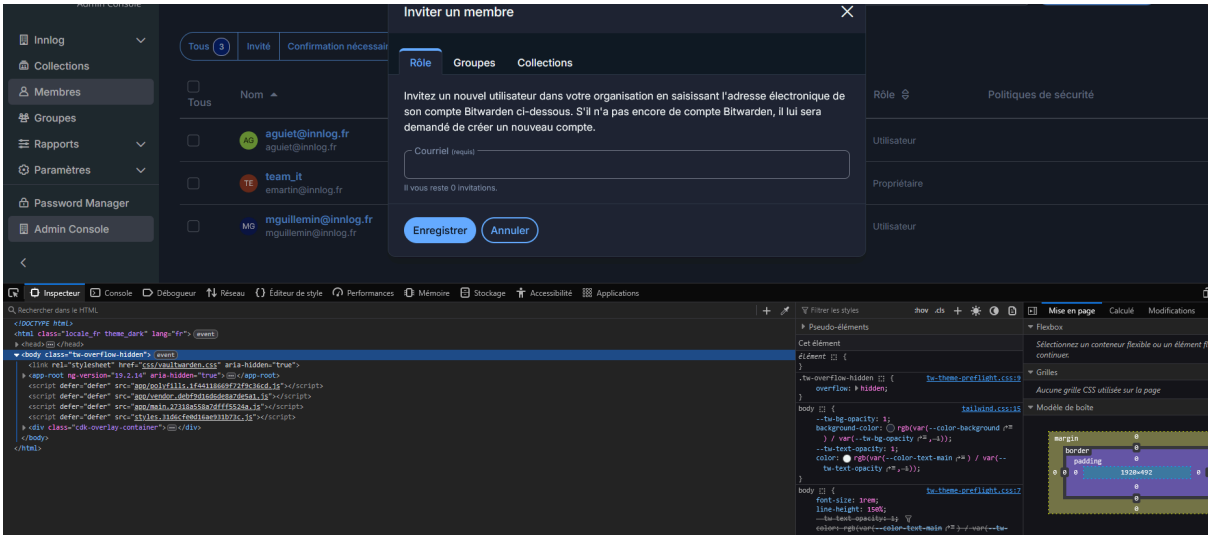
- **Inviter massivement des utilisateurs dans l'organisation**

Dans Vaultwarden pour que des personnes fassent partie de votre organisation, il faut les inviter et ce processus se fait par un email à chaque fois ce qui peut vite être chronophage si on a des dizaines de personnes à inviter. J'ai donc automatisé tout ça avec un bout de **JavaScript** à coller dans la console web afin d'inviter massivement des utilisateurs. C'est le script **massive_invite.js**

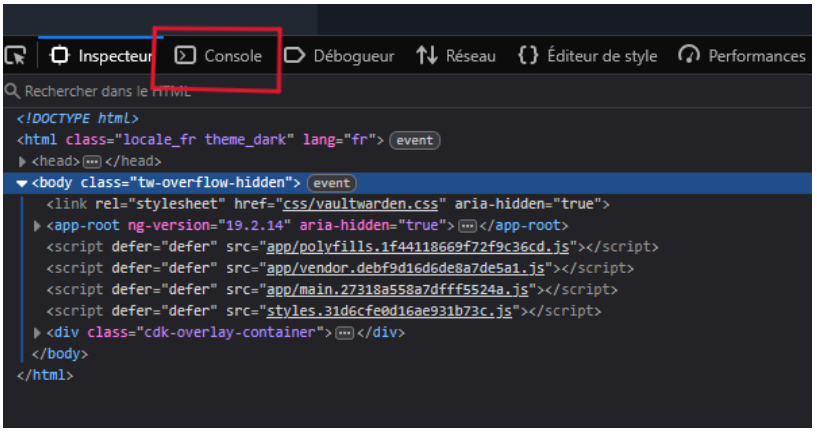
Quand vous serez dans votre organisation aller dans le menu **Membres** et **Inviter un membre**



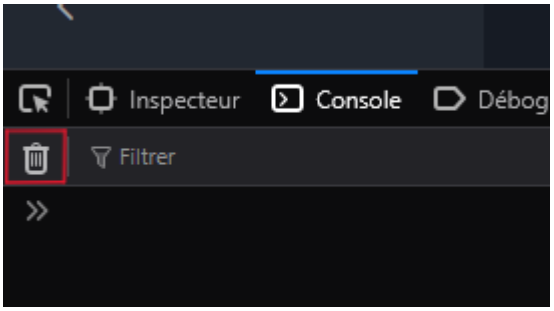
A ce moment là vous aurez une petite fenêtre qui s'ouvrira et la faites la touche **F12** de votre clavier et vous arriverez ici :



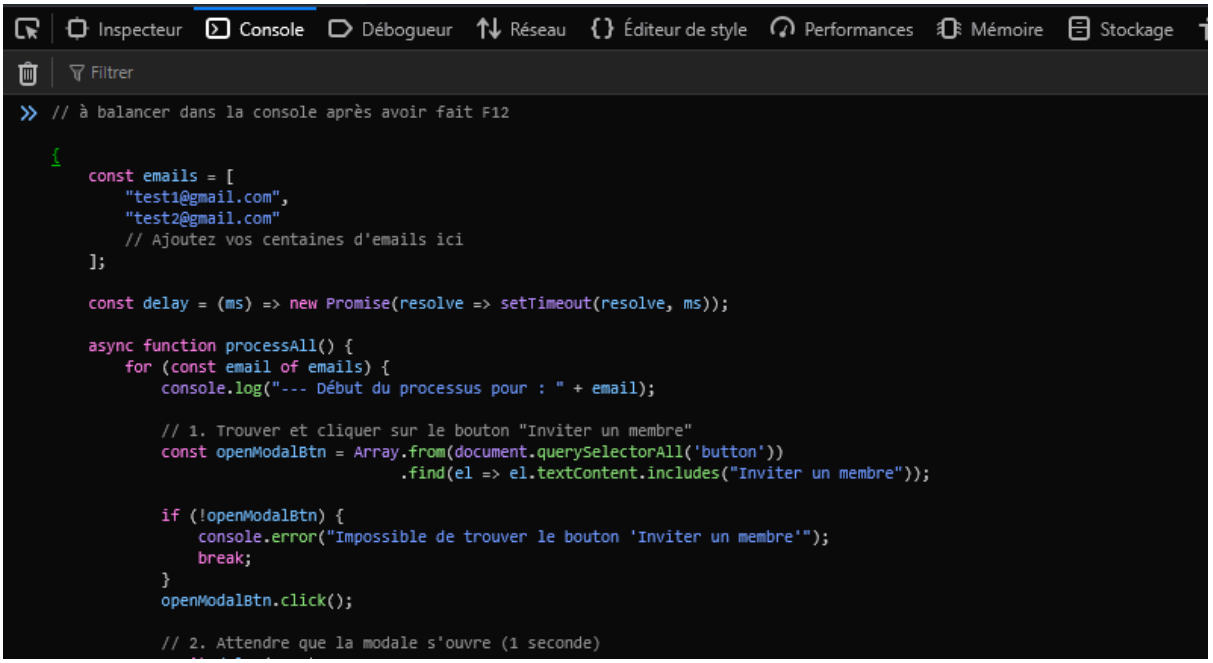
Il faudra aller dans la rubrique console :



Une fois arriver dans la console vous aurez certainement des lignes rouges d’erreur pour vous en débarrasser faites la poubelle juste à gauche :



Une fois que la consol est nettoyer copier coller le script avec tous les emails que vous voulez inviter :



Faites Entrer et laisser la magie opérer