

Importer ses notes existantes

Migrer tes notes depuis Notion, Evernote, Google Keep, Bear ou Apple Notes dans ton nouveau vault.

Source : <https://prisme.one/plateforme/ia-juillet-2026/semaine-1-importer-notes>

Cette leçon est optionnelle si tu pars de zéro. Si tu as déjà des notes ou des chats Claude / ChatGPT que tu utilises encore, lis-la avant de lancer `/import`. Elle t'évitera d'importer n'importe quoi n'importe comment.

Pourquoi importer ?

L'IA est aussi puissante que ce qu'elle sait de toi. Si tu as passé des années à accumuler des notes, des process, des réflexions, des conversations avec Claude ou ChatGPT : tout ça est du contexte dormant. L'objectif de l'import, c'est de le **réveiller** : transformer ce passé en contexte actif que l'IA peut utiliser pour travailler avec toi.

Mais attention : on n'importe pas tout, et on n'importe pas tout d'un coup.

Étape 1 : As-tu vraiment envie d'importer ?

Avant de tout télécharger, pose-toi la question honnêtement. Importer prend du temps et de l'attention. Si tu importes des piles de notes qui ne te servent plus, tu pollues ton coffre et ton IA passera son temps à lire des trucs morts.

Importe si...

- Tu as un corpus de notes ou de chats vraiment utiles que tu consultes encore aujourd'hui
- Tu veux que l'IA connaisse ton passé pour mieux travailler sur tes projets actuels
- Tu as des process documentés que tu réutilises régulièrement

N'importe pas si...

- Tu n'as pas regardé ces notes depuis 2 ans
- Ce sont des tâches, du Kanban, du PM (laisse-les dans Notion / Linear)
- C'est juste pour "ne rien perdre" : laisse en archive externe, c'est très bien

Contexte ou project management ?

C'est la première vraie question. Pour chaque source de notes, demande-toi :

C'est du project management ?

- Tâches à faire, deadlines, tableaux Kanban
- Bases de données partagées en équipe
- Tickets, sprints, roadmaps

Laisse où c'est

Notion, Airtable, Linear restent les meilleurs outils. L'IA s'y connecte via MCP.

C'est du contexte ?

- Notes de réunion, décisions, bilans
- Process, méthodes, façons de travailler
- Réflexions, journaling, notes de lecture
- Conversations Claude / ChatGPT sur tes sujets

Importe dans le coffre

C'est ça qui enrichit l'IA.

Étape 2 : Qu'est-ce que tu vas importer ?

Si tu décides d'importer, identifie tes sources. La grande majorité des gens ont deux familles principales :

1. Tes chats historiques Claude / ChatGPT

Souvent des années de réflexion partagée avec ces modèles. La plupart des gens oublient cette source : c'est pourtant la plus dense en contexte sur toi, tes projets, ta façon de penser.

2. Tes anciennes notes

Notion, Evernote, Apple Notes, fichiers Markdown, Bear, Google Keep, dossiers de fichiers Word / PDF / PowerPoint... Tout ce que tu as accumulé ailleurs.

Et aussi : actif ou archivé ?

Pour chaque source, deuxième question rapide : c'est actif ou c'est terminé ?

Actif ou de référence

- Projets en cours
- Process que tu utilises encore
- Casquettes actuelles
- Références consultées régulièrement
- PROJETS / CASQUETTES / RESSOURCES

Terminé ou obsolète

- Projets terminés
- Anciennes entreprises
- Process abandonnés
- Notes de 2019 sur sujets dépassés
- ARCHIVES (ou supprimer)

Si tu doutes, mets dans `_Import/` et laisse l'IA proposer. Tu valides après.

Étape 3 (préalable) : Télécharger et préparer tes sources

Cette étape se fait **en amont, avant de lancer /import**. Tu télécharges tout ce que tu veux importer et tu le déposes dans `Second Cerveau/_Import/`. Plus tu prépares proprement, mieux la commande travaille ensuite.

Comment exporter chaque source

Tip : lance les 4 exports IA en début de session. Ils arrivent par email pendant que tu fais autre chose (Claude/ChatGPT mettent souvent moins de 10 min, Gemini quelques minutes à plusieurs heures selon le volume). Les liens email Claude et ChatGPT sont valides **24h seulement** : télécharge dès que tu reçois.

Conversations ChatGPT (OpenAI) : icône profil en bas à gauche → **Settings** → **Data Controls** → **Export Data** → **Export** → confirmer. Tu reçois un email avec un lien (valide 24h). L'export peut prendre **jusqu'à 7 jours** mais arrive souvent en quelques minutes. Tu obtiens un ZIP avec `conversations.json` (historique complet, structure `mapping{}` avec arbre des messages), `chat.html` (lisible), `user.json`, fichiers uploadés. Dépose le ZIP entier (ou juste `conversations.json`) dans `_Import/`.

Conversations Claude (Anthropic) : dans Claude web ou Claude Desktop, initiales en bas à gauche → **Settings** → **Privacy** → **Export data**. Tu reçois un email avec un JSON (valide 24h).

Structure : array de conversations, chacune avec `uuid`, `name`, `chat_messages[]` (sender human/assistant, text, attachments). **Pas d'export possible depuis l'app iOS/Android**, fais-le depuis le web/desktop. Dépose le JSON dans `_Import/`.

Conversations Gemini (Google) : va sur takeout.google.com → **Deselect all** → coche uniquement **My Activity** → clique "All activity data included" → **Deselect all** → coche uniquement **Gemini Apps** → OK → suis le wizard. Tu reçois un email Google avec lien Drive ou direct. Tu obtiens un ZIP avec HTML lisible **et** JSON brut sous `Takeout/My Activity/Gemini Apps/`. **Piège fréquent** : sans le double filtre "My Activity → Gemini Apps", tu télécharges tout ton historique Google entier. Bien insister sur cette étape.

Conversations Mistral Le Chat : Le Chat → Settings → **Export data** (téléchargement direct, pas d'email asynchrone). Format JSON, bulk export en beta en 2026. Bonus : dans Mistral Code, commande `/cmd` pour exporter une conversation isolée au format portable.

Notion : Settings → Export all workspace content → format **Markdown & CSV**. Décompresse le ZIP, copie dans `_Import/`.

Evernote : Fichier → Exporter → format **ENEX**. Dépose dans `_Import/`.

Apple Notes : pas d'export natif. Utilise [Exporter](#) (Mac, gratuit) → export Markdown.

Bear : export Markdown depuis l'app.

Google Keep : takeout.google.com.

Fichiers .md / .txt : copie directe.

Limites à connaître

- **Volume ChatGPT** : un export de gros utilisateur peut faire 50 à 200 Mo de JSON. La commande `/import` le découpe en notes individuelles, donc OK, mais sois patient au premier scan.
- **Auto-parsing** : la commande reconnaît automatiquement les structures Claude (`chat_messages`) et ChatGPT (`mapping`). Pour Gemini et Mistral, l'agent va devoir improviser ou tu peux convertir manuellement. Prévois un round dédié si tu as beaucoup de ces deux-là.
- **Conversations courtes (moins de 5 échanges)** : sont mises en `ARCHIVE` par défaut. Bonne pratique pour ne pas polluer le coffre avec des micro-conversations sans valeur.
- **Conversations entières en `KEEP_NOTE`** : mauvaise idée. La valeur est diffuse dans un chat brut. Préfère toujours `EXTRACT_CONTEXT` ou `SUMMARIZE`.

Formats reconnus dans `_Import/`

Format

Compatibilité

Que faire

Markdown (.md) / Texte (.txt)

Direct

Copie dans `_Import/` tel quel

Export Notion / Evernote (ENEX)

Direct

Décompresse si ZIP, copie dans `_Import/`

Conversations Claude / ChatGPT (JSON)

Auto-parsé

Dépose le JSON, `/import` le découpe en une note par conversation

PDF, Word, PowerPoint, Excel

Conversion auto

`/import` installe **MarkItDown** si besoin et convertit en .md

Quelques images d'illustration

OK

Si elles illustrent des notes précises, dans `_Import/`, l'IA les rattachera

Bibliothèque d'images / vidéos

À côté

Le coffre n'est pas une médiathèque. Garde ta base dans Drive, Dropbox ou un dossier dédié.

Images : tu peux glisser **quelques images** pour illustrer une note (un schéma, une capture). Ne rapatrie pas ta bibliothèque entière. Le coffre est un cerveau textuel, pas une photothèque. Pour une vraie médiathèque, garde un dossier `Médias/` à côté du Workspace, ou un Dropbox dédié.

MarkItDown : installé par `/import`

Tu n'as **rien à installer manuellement** : à l'étape de préparation, `/import` détecte les PDF/Word/PowerPoint, vérifie si MarkItDown (Microsoft) est dispo sur ta machine, et

l'installe à ta place s'il manque (avec confirmation). Tu n'as donc pas besoin de Python ou de bricoler en amont.

Si tu veux quand même l'installer toi-même avant : colle ce prompt dans OpenCode Desktop avant de lancer `/import` :

```
Installe MarkItDown sur ma machine. Source officielle :  
https://github.com/microsoft/markitdown
```

```
Détecte mon OS et ce que j'ai déjà (Python, pip, uv, brew...),  
choisis la meilleure méthode et fais-la tourner. Vérifie ensuite  
que `markitdown --help` fonctionne. Guide-moi si tu rencontres un  
blocage plutôt que de pousser un workaround.
```

Étape 4 : Importer progressivement, pas tout d'un coup

C'est **le principe le plus important** de toute la leçon. Plus tu donnes peu à la fois à l'IA, mieux elle travaille.

Pourquoi pas tout d'un coup ?

→ Le contexte de l'IA est limité. Plus tu lui balances de notes en même temps, plus elle perd en finesse et en cohérence.

→ Tu ne peux pas vraiment valider 800 propositions de routage. Tu vas faire « OK » par fatigue.

→ En cas de gros loupé, tu reviens en arrière sur des centaines de fichiers déjà déplacés.

Le bon réflexe : rounds courts et thématiques

→ **Par source** : un round Claude, un round ChatGPT, un round Notion. Pas tout mélangé.

→ **Par batch de ~100 notes max** : déjà beaucoup. Au-delà, l'IA dérape ou tu décroches.

→ **Par projet ou casquette** si tu as un très gros corpus : « cette semaine, j'importe tout ce qui touche au projet X » suffit comme cadre.

Un workflow type

Semaine 1 :

Round 1 → Import des chats Claude (1 export JSON dans `_Import/`)

Round 2 → Import des chats ChatGPT (idem)

Semaine 2 :

Round 3 → Import Notion : tout ce qui touche au projet A

Round 4 → Import Notion : tout ce qui touche aux casquettes

Semaine 3 :

Round 5 → Reste des notes Notion

Round 6 → Apple Notes / Bear / fichiers vrac

Entre chaque round, tu peux relancer `/import`. Le dossier `_Import/` est conçu pour ça : tu vides, tu remplis, tu relances.

Étape 5 : Lancer `/import` (plan mode puis build mode)

Quand `_Import/` est prêt (avec les fichiers d'un round, pas tout en même temps), tu lances `/import`. La commande travaille en **deux temps clairs** : plan d'abord, exécution ensuite.

PLAN MODE

- Alignement profond (contenu détaillé, intention, ce que tu sais déjà devoir archiver, projet central, budget tokens)
- Préparation auto : Markdown installé si besoin, JSON Claude/ChatGPT parsés en notes individuelles
- Scan initial : l'agent principal fait un 1s du dossier et échantillonne pour saisir le style du corpus
- Table de routage : il lit `Moi.md`, projets, casquettes avec leurs mots-clés
- Délégation à des sous-agents briefés (contexte vault + intention + corpus + mots-clés)
- Plan global agrégé soumis à validation

Aucune écriture pendant cette phase. Tu peux ajuster note par note ou demander à voir le contenu d'un item avant de décider.

BUILD MODE

- Exécution par lots de 15 items
- Rapport intermédiaire après chaque lot
- Pause / stop possible à tout moment
- Récap final + feedback loop

À la fin, l'IA te demande comment elle aurait pu faire mieux et te propose d'éditer son skill.

Les 4 verdicts possibles par item

C'est ce qui rend `/import` puissant. Pour chaque note ou conversation, l'IA tranche entre **quatre actions**, pas une seule :

KEEP_NOTE

La note est utile telle quelle. Elle est rangée dans 1 PROJETS/, 2 CASQUETTES/ ou 3 RESSOURCES/.

EXTRACT_CONTEXT

L'info utile est extraite et ajoutée à Moi.md, à la note du projet ou de la casquette. L'original part en 5 ARCHIVE/_extraits/.

SUMMARIZE

Plusieurs notes redondantes sur un sujet → une seule note de synthèse. Les sources partent en 5 ARCHIVE/_resumes-sources/.

ARCHIVE

Bruce Lee : « discard what is useless ». La note part en 5 ARCHIVE/.

Comment lancer

```
/import
```

C'est tout. L'IA prend le relais et te guide depuis l'alignement jusqu'au feedback final.

Coût et clé OpenRouter (pour les gros rounds)

Si un round dépasse les 100 items, ou si tu importes plusieurs années de chats, les sous-agents de scan peuvent **consommer beaucoup de tokens**. Trois options pour ne pas dégommer ton abonnement principal :

Option 1 : clé OpenRouter avec DeepSeek V4 Flash

Tu fournis à `/import` une clé OpenRouter, et la commande route les sous-agents de scan/routing vers **DeepSeek V4 Flash** (10x moins cher qu'un modèle frontier, largement suffisant pour du routing de notes). Ton abonnement ChatGPT Plus ou Claude Max reste libre pour le reste de ta journée.

À l'alignement initial (plan mode), indique que tu as une clé OpenRouter. L'IA te confirme et l'utilise pour les sous-agents.

Option 2 : plan Max à 100 €/mois minimum

Si tu ne veux pas gérer une clé séparée, **assure-toi d'avoir un plan Max** (Claude Max à 100 €/mois ou ChatGPT Pro). Les plans basiques ne tiennent pas un gros round.

Option 3 : juste faire des rounds plus petits

C'est l'esprit de l'étape 4 : si tu fais des rounds de 100 notes ou 1 export Claude à la fois, le coût reste raisonnable même sur un plan classique.

- ☐ Je sais distinguer contexte (à importer) et project management (à laisser dans Notion / Linear)
 - ☐ J'ai identifié mes deux familles de sources : chats Claude/ChatGPT + anciennes notes
 - ☐ J'ai exporté en JSON les conversations Claude (Settings → Privacy → Export data)
 - ☐ J'ai exporté en JSON les conversations ChatGPT (Settings → Data Controls → Export)
 - ☐ J'ai exporté mes notes Notion / Evernote / Apple Notes / autres dans les bons formats
 - ☐ Je sais que je n'importe pas tout d'un coup : rounds courts (~100 notes max) et séparés par source
 - ☐ Je sais que /import installe Markdown lui-même si j'ai des PDF/Word/PowerPoint
 - ☐ J'ai choisi mon budget : clé OpenRouter DeepSeek, plan Max suffisant, ou rounds courts
 - ☐ Premier round préparé dans _Import/ : 1 source à la fois
 - ☐ Lancé /import → conversation d'alignement faite (intention claire)
 - ☐ PLAN MODE : plan de routing validé avant exécution
 - ☐ BUILD MODE : exécution par lots, sources rangées dans les bons dossiers IPCRA
 - ☐ Feedback donné à la commande à la fin
-

Extraction verbatim depuis Prisme One · <https://prisme.one/plateforme/ia-juillet-2026/semaine-1-importer-notes>