

Traiter les fichiers PowerPoint, Word et Excel

Comment lire et convertir les fichiers Office avec l'IA — et pourquoi migrer vers des fichiers libres.

Traiter les fichiers PowerPoint, Word et Excel avec l'IA

OpenCode ne peut pas lire nativement les fichiers `.pptx`, `.docx` ou `.xlsx`. Ce sont des formats binaires compressés (du XML dans un ZIP) : quand l'IA essaie de les ouvrir directement, elle obtient du charabia. C'est une limite de **tous** les agents code, pas juste d'OpenCode.

Il y a deux limites à bien distinguer, parce que beaucoup s'y cassent les dents :

- **L'agent lit du texte.** Les fichiers Office (`.pptx`, `.docx`, `.xlsx`) et les PDF sont des binaires. Sans conversion, l'IA voit du charabia. C'est vrai pour tous les agents code.
- **Le modèle derrière compte aussi.** La plupart des modèles open source que tu utiliseras via OpenRouter (DeepSeek, GLM...) sont *text-only* : ils ne lisent ni les binaires (PDF, Word, Excel, PowerPoint) ni les **images**. Les modèles multimodaux (GPT, Gemini, Claude) voient les images, mais ont quand même besoin d'une conversion pour les fichiers Office.

Bonne nouvelle : il existe une solution simple pour chaque cas. Et surtout, une philosophie : **aller vers des fichiers libres** que l'IA peut éditer directement.

Quel outil pour quel cas

- **PDF, Word, Excel, PowerPoint** : Markdown (Solution 1). Ta solution par défaut.
- **EPUB et formats exotiques** : pandoc.
- **PDF complexe** (colonnes, tableaux denses, scans) : pdfjs-dist.

- **Images dans le document** (schéma, capture, photo) : bascule sur un modèle multimodal (GPT, Gemini) qui voit les images.
-

Solution 1 : Markdown MCP (par Microsoft), pour lire les fichiers Office

Microsoft a créé un outil open-source (160k+ stars sur GitHub) qui convertit automatiquement les fichiers Office en Markdown lisible par l'IA. C'est **la solution recommandée** pour importer du contenu existant.

Formats supportés : PDF, PowerPoint, Word, Excel, images, audio, HTML, CSV, JSON, XML, ZIP, YouTube...

Installation

Ouvre OpenCode et demande-lui :

```
Installe le MCP markdown pour que tu puisses lire les fichiers
PowerPoint, Word et Excel.
Voici la config :
```

```
{
  "mcpServers": {
    "markdown": {
      "command": "uvx",
      "args": ["markdown-mcp"]
    }
  }
}
```

Installe la version à jour. Markdown évolue vite, et une vieille version alpha (type 0.0.1a1) qui traîne dans certains tutos plante ou rate des formats. `uvx markdown-mcp` récupère normalement la dernière version. Si tu l'avais installé il y a longtemps, demande à OpenCode de forcer la mise à jour (`uvx markdown-mcp@latest`).

Utilisation

Une fois installé, tu peux simplement dire :

```
Lis le fichier presentation.pptx et résume-le en notes dans mon vault
```

```
Extrais le contenu de rapport.docx et crée une note de synthèse
```

```
Convertis le tableau budget.xlsx en fichier CSV lisible
```

L'IA convertira automatiquement le fichier en Markdown avant de le traiter.

Solution 2 : Convertir en PDF (zéro installation)

Si tu ne veux rien installer, le plus simple :

1. Ouvre ton fichier dans PowerPoint, Word ou Excel
 2. **Fichier → Enregistrer sous → PDF**
 3. Place le PDF dans ton vault ou dossier de travail
 4. OpenCode sait lire les PDF nativement
-

Solution 3 : l'app Claude sur claude.ai

Sur claude.ai (le chat web), tu peux directement **glisser-déposer** tes fichiers PowerPoint, Word et Excel en pièce jointe. L'IA les lit et les comprend sans aucune configuration.

C'est la méthode la plus rapide pour une analyse ponctuelle.

Cas particuliers : EPUB, PDF complexes et images

MarkItDown couvre l'immense majorité des besoins. Pour le reste, voici les bons outils.

EPUB et livres numériques

MarkItDown (comme docling) gère mal les EPUB : le résultat est souvent incomplet ou cassé, ne t'y fie pas. **Pour les EPUB, la solution fiable c'est pandoc.** C'est l'outil de référence pour

convertir un livre numérique en Markdown propre.

Demande à OpenCode :

```
Installe pandoc et convertis mon-livre.epub en Markdown
```

PDF complexes

Certains PDF passent mal avec MarkItDown : colonnes multiples, tableaux denses, mises en page tordues, scans. Dans ce cas, **pdfjs-dist** (la librairie PDF de Mozilla, celle qui fait tourner Firefox) extrait le texte de façon plus robuste.

Demande à OpenCode d'extraire le texte de ton PDF avec pdfjs-dist.

Images dans un document

Si l'information que tu cherches est **dans une image** (schéma, capture d'écran, photo, graphique), aucune conversion en texte ne la fera apparaître. Il te faut un **modèle qui voit les images**, donc un modèle multimodal : bascule sur GPT ou Gemini. Un modèle text-only (DeepSeek, GLM...) restera aveugle à l'image, même convertie.

Le plus simple : glisse l'image (ou le PDF qui la contient) directement dans claude.ai, ChatGPT ou Gemini. Sinon, configure OpenCode avec un modèle multimodal.

La vraie philosophie : aller vers des fichiers libres

L'objectif à moyen terme, c'est de **sortir petit à petit des fichiers propriétaires**. Pourquoi ?

- Les fichiers Markdown, HTML et CSV sont **directement éditables** par l'IA, sans conversion ni MCP supplémentaire
- Tu n'as plus besoin d'un abonnement Microsoft Office ou Google Workspace
- L'IA peut modifier le contenu **en direct**, pas juste le lire
- Tes fichiers sont légers, versionnables (Git) et pérennes

Au lieu de...	Utilise...	Pourquoi c'est mieux
PowerPoint (.pptx)	Présentations HTML	L'IA crée des slides animées bien plus belles que PowerPoint, éditables en direct

Au lieu de...	Utilise...	Pourquoi c'est mieux
Word (.docx)	Fichiers Markdown (.md)	Lisibles partout, éditables par l'IA en direct, convertibles en PDF
Excel (.xlsx)	Fichiers CSV ou Markdown	L'IA les lit et les modifie nativement, pas de logiciel requis
Google Docs / Sheets	MCP Google Workspace (GOG)	Connecte directement OpenCode à Google Drive, Docs, Sheets, Calendar

Exemple : remplacer PowerPoint par des slides HTML

Avec le skill **frontend-slides**, l'IA peut créer des présentations HTML animées directement depuis OpenCode. Le résultat est souvent **bien plus beau** qu'un PowerPoint classique :

- Animations fluides et personnalisées
- Design responsive (fonctionne sur mobile)
- Zéro logiciel requis : s'ouvre dans n'importe quel navigateur
- L'IA peut les modifier en direct, slide par slide

Pour l'installer, demande simplement à OpenCode :

```
Installe le skill frontend-slides depuis ce repo :
https://github.com/zarazhangrui/frontend-slides
```

Puis pour créer une présentation :

```
Crée une présentation de 10 slides sur [ton sujet]
```

Et la commande /done dans tout ça ?

La commande `/done` met à jour le **contexte de ton projet** en fonction de ce que l'IA a fait pendant ta session de travail. Elle ne "lit" pas tes fichiers PowerPoint ou Word : elle résume les **actions effectuées** durant la session (fichiers créés, modifiés, décisions prises, etc.).

Si tu as travaillé avec l'IA sur un fichier Office (via Markdown ou une conversion PDF), alors oui, ce travail sera reflété dans le contexte mis à jour par /done.

- ☐ Je comprends pourquoi OpenCode ne lit pas les fichiers Office nativement
- ☐ J'ai installé le MCP Markdown (ou je sais convertir en PDF)
- ☐ J'ai testé la lecture d'un fichier Office dans OpenCode
- ☐ Je comprends l'intérêt de migrer vers des fichiers libres (Markdown, HTML, CSV)