AI TOOLS WIKI · MÉMO

Graphify *mémo*

Les commandes à connaître pour **Graphify**, qui transforme un codebase (et ses docs, schémas, PDF…) en graphe de connaissances interrogeable. Claude Code ou Codex consulte le graphe au lieu de relire et grepper des dizaines de fichiers, ce qui économise beaucoup de contexte sur les gros projets.

Famille Mémoire et contexte · **Éditeur** Graphify Labs · **Licence** Apache-2.0 / MIT
Note du wiki 9/15 · **Commandes vérifiées le** 25 septembre 2026
Fiche, notes et avis aitoolswiki.viellard.fr/outils/memoire-et-contexte/graphify

Installation

`uv tool install graphifyy`	attention : le paquet s'écrit avec deux « y »
`graphify install`	enregistre la skill auprès de l'assistant
Une fois le graphe construit, dans le projet : faire consulter le graphe systématiquement	
`graphify claude install`	Claude Code (hook avant les recherches et lectures de fichiers)
`graphify codex install`	Codex (via AGENTS.md)

Les autres paquets `graphify*` sur PyPI ne sont pas affiliés au projet.

Construire et mettre à jour le graphe

`/graphify .`	Construit le graphe du dossier courant (rapport, JSON et visualisation HTML dans `graphify-out/`).
`/graphify . --update`	Ne réextrait que les fichiers modifiés.
`graphify extract . --code-only`	Indexe seulement le code, en local, **sans clé API** (analyse syntaxique pure).
`/graphify . --mode deep`	Extraction plus poussée des relations (plus coûteuse).

Construire et mettre à jour le graphe (suite)

`/graphify . --no-viz`	Rapport + JSON, sans la visualisation HTML.
`/graphify . --watch`	Resynchronise à chaque modification de fichier.
`graphify update .`	Met le graphe à jour, par exemple après un `git pull`.
`graphify hook install`	Reconstruit le graphe à chaque commit et changement de branche (analyse du code seulement, sans coût API).

Interroger le graphe

`/graphify query "qu'est-ce qui relie l'auth à la base de données ?"`	Question en langage naturel sur le code.
`/graphify path "UserService" "DatabasePool"`	Chemin entre deux éléments du code.
`/graphify explain "RateLimiter"`	Explique un élément et ses liens.
`graphify query "…"`	Même chose depuis le terminal.
`/graphify add <url>`	Ajoute un article (arXiv…) ou une vidéo (transcrite) au graphe.

Le rapport `graphify-out/GRAPH_REPORT.md` liste les nœuds centraux (« god nodes »), les connexions surprenantes et des questions suggérées. Chaque relation est marquée `EXTRACTED` (trouvée), `INFERRED` (déduite) ou `AMBIGUOUS`.

Exporter et servir

`/graphify . --wiki`	Wiki Markdown parcourable par un agent.
`/graphify . --obsidian`	Coffre Obsidian (ne touche pas à tes notes existantes).
`python -m graphify.serve graphify-out/graph.json`	Serveur MCP (`query_graph`, `get_node`, `get_neighbors`, `shortest_path`…).

Exporter et servir (suite)

`graphify prs --triage`	Tableau des PR, classées par l'IA selon leur impact sur le graphe.

Travail en équipe

1. Committer `graphify-out/` : tout le monde démarre avec la carte du projet (garder `graphify-out/cost.json` hors de git).
2. Chaque développeur lance `graphify hook install` une fois après le clone.
3. Après chaque `git pull`, lancer `graphify update .`. Astuce : `git config --global alias.gpull '!git pull && graphify update .'`

Conseils d'utilisation

- **Réservé aux gros projets** : sous 30 fichiers, Claude peut tout lire directement ; le gain devient important au-delà de quelques centaines de fichiers.
- **Poser des questions ciblées** (`query`, `path`, `explain`) plutôt que de faire lire tout le rapport.
- **Exclure le bruit** avec un `.graphifyignore` (même syntaxe que `.gitignore`) : `node_modules/`, `dist/`, fichiers générés.
- **Commencer par** `--code-only` pour tester sans clé API ni coût ; ajouter les docs ensuite.
- **Si une réponse semble dépassée**, lancer `graphify update .` avant de reposer la question.
- **Tout retirer** : `graphify uninstall` (ajouter `--purge` pour supprimer aussi `graphify-out/`).

Sources (documentation officielle)

- README Graphify
- Graphify Tested – Engr Mejba Ahmed
- Token-Efficient Claude Code – Rukmal Dias
- Graphify Review – RoboRhythms