AI TOOLS WIKI · MÉMO

Hermes Agent *mémo*

Les commandes de **Hermes Agent** à connaître : installation, configuration du modèle, passerelle de messagerie, tâches planifiées et contrôle de la mémoire.

Famille Agents de développement · **Éditeur** Nous Research · **Licence** MIT
Note du wiki 11/15 · **Commandes vérifiées le** 11 octobre 2026
Fiche, notes et avis aitoolswiki.viellard.fr/outils/agents-de-developpement/hermes-agent

Installation et premier lancement

`curl -fsSL https://hermes-agent.nousresearch.com/install.sh | bash`	Linux, macOS, WSL2
`source ~/.zshrc`	ou ~/.bashrc
`hermes setup`	assistant de configuration complet
`hermes model`	choisir fournisseur et modèle
`hermes --tui`	interface moderne (recommandée) ; `hermes` : CLI classique
`hermes --continue`	reprendre la session

Windows (PowerShell) : `iex (irm https://hermes-agent.nousresearch.com/install.ps1)`.

Commandes du terminal

`hermes chat`	Démarre une conversation interactive ou ponctuelle.
`hermes gateway setup` / `hermes gateway` / `hermes gateway status`	Configure, démarre et vérifie la passerelle de messagerie.
`hermes cron`	Inspecte et gère les tâches planifiées.
`hermes pause` / `hermes resume`	Arrête / relance globalement les tâches planifiées, kanban et passerelle.
`hermes profile`	Gère des profils (instances isolées).

Commandes du terminal (suite)

`hermes skills`	Parcourt, installe, publie et audite les skills.
`hermes memory`	Configure un fournisseur de mémoire externe.
`hermes mcp`	Gère les serveurs MCP, ou lance Hermes comme serveur MCP.
`hermes sessions`	Parcourt, exporte, supprime les sessions.
`hermes status` / `hermes usage` / `hermes insights`	État de l'agent / limites du compte / coûts et activité.
`hermes doctor`	Diagnostique la configuration et les dépendances.
`hermes security audit`	Audit de vulnérabilités de la chaîne d'approvisionnement.
`hermes backup` / `hermes import`	Sauvegarde le dossier Hermes en archive / la restaure.
`hermes update`	Met à jour Hermes et réinstalle les dépendances.
`hermes claw`	Aides à la migration depuis OpenClaw.

Commandes dans une session

`/new` (alias `/reset`)	Nouvelle session ; à utiliser aux changements naturels de tâche pour garder la mémoire propre.
`/model`	Affiche ou change le modèle et le fournisseur.
`/memory`	Examine et approuve ou rejette les écritures de mémoire en attente.
`/learn`	Distille une skill réutilisable à partir d'une source décrite.
`/skills`	Recherche, installe et gère les skills.

Commandes dans une session (suite)

`/refine`	Lance maintenant la revue d'auto-amélioration (mémoire et skills).
`/compress`	Compresse le contexte.
`/context`	Répartition de l'usage de la fenêtre de contexte.
`/goal`, `/plan`, `/loop`	Objectif durable, plan Markdown sans exécution, relance périodique d'un prompt.
`/rollback`, `/diff`	Restaure des points de contrôle de fichiers, montre les changements Git.
`/approvals`, `/yolo`	Mode d'approbation des commandes dangereuses ; `/yolo` les désactive.
`/usage`, `/insights`	Tokens, coût et limites ; analytique des 30 derniers jours.

Conseils d'utilisation

- **Commencer par un profil, une tâche étroite et des permissions minimales**, puis élargir quand le workflow est fiable.
- **Vérifier ce qui est réellement mémorisé** (`MEMORY.md`, `USER.md`, `/memory`) : la confirmation textuelle du modèle ne prouve pas une écriture.
- **Maîtriser les coûts** : la revue d'arrière-plan peut être désactivée, plafonnée ou confiée à un modèle moins cher.
- **Isoler l'agent** : Docker ou autre sandbox, compte non-root, et éviter `/yolo` hors environnement jetable.
- **Créer de nouvelles sessions régulièrement** plutôt que de laisser un fil de messagerie grossir.
- **Contrôler les fournisseurs appelés** après une mise à jour (`hermes doctor`, journaux réseau).

Sources (documentation officielle)

- Hermes Agent – démarrage rapide
- Hermes Agent – commandes du terminal
- Hermes Agent – commandes slash