Chrome DevTools MCP *mémo*

Les commandes à connaître pour **Chrome DevTools MCP**, le serveur MCP officiel de Chrome qui donne à Claude Code les outils de DevTools : traces de performance, réseau, console, mémoire, captures. Là où **Playwright** sert surtout à **piloter** une page, Chrome DevTools MCP sert à la **diagnostiquer**.

Famille Navigateur et tests · **Éditeur** Google (équipe Chrome DevTools) · **Licence** Apache-2.0
Note du wiki 12/15 · **Commandes vérifiées le** 25 septembre 2026
Fiche, notes et avis aitoolswiki.viellard.fr/outils/navigateur-et-tests/chrome-devtools-mcp

Installation dans Claude Code

`claude mcp add chrome-devtools --scope user npx chrome-devtools-mcp@latest`	Serveur MCP seul
`/plugin marketplace add ChromeDevTools/chrome-devtools-mcp` `/plugin install chrome-devtools-mcp@chrome-devtools-plugins`	Ou plugin (serveur MCP + skills)

Prérequis : Node.js LTS et Chrome stable. Le navigateur ne démarre qu'au premier outil qui en a besoin.
Premier test : demander « Check the performance of https://developers.chrome.com ». Claude doit ouvrir le navigateur et enregistrer une trace.

Options de lancement utiles

Option	Effet
`--slim`	Seulement 3 outils (navigation, script, capture). Pour les tâches simples, moins de contexte consommé.
`--headless`	Sans interface visible.
`--isolated`	Profil temporaire, effacé à la fermeture.
`--autoConnect`	Se connecte à ton Chrome déjà ouvert (Chrome 144+, débogage distant activé via `chrome://inspect/#remote-debugging`).

Options de lancement utiles (suite)

Option	Effet
`--browserUrl http://127.0.0.1:9222`	Se connecte à une instance Chrome en mode débogage.
`--no-usage-statistics`	Désactive l'envoi de statistiques d'usage à Google (activé par défaut).
`--no-performance-crux`	N'envoie pas les URL tracées à l'API CrUX de Google.

Outils MCP par catégorie

Claude les appelle lui-même ; les connaître aide à formuler les demandes.

Catégorie	Outils principaux	À demander par exemple
Performance	**`performance_start_trace`**, **`performance_stop_trace`**, **`performance_analyze_insight`**	« Pourquoi le LCP de cette page est lent ? »
Audit	**`lighthouse_audit`**	« Lance un audit Lighthouse et corrige les points critiques. »
Réseau	**`list_network_requests`**, **`get_network_request`**	« Quelles requêtes échouent au chargement ? »
Console	**`list_console_messages`**, **`get_console_message`**	« Lis les erreurs console et trouve leur source. » (traces avec source maps)
Mémoire	**`take_heapsnapshot`**, **`compare_heapsnapshots`**, **`get_heapsnapshot_retainers`**…	« Cherche une fuite mémoire après 10 navigations. »
Inspection	**`take_snapshot`**, **`take_screenshot`**, **`get_css_styles`**, **`evaluate_script`**	« Pourquoi ce bouton est mal aligné ? »
Navigation	**`new_page`**, **`navigate_page`**, **`list_pages`**, **`select_page`**, **`wait_for`**	–
Interaction	**`click`**, **`fill`**, **`fill_form`**, **`type_text`**, **`press_key`**, **`hover`**, **`upload_file`**	« Remplis le formulaire et soumets-le. »

Outils MCP par catégorie (suite)

Catégorie	Outils principaux	À demander par exemple
Émulation	**`emulate`**, **`resize_page`**	« Teste en mobile avec une connexion 3G lente. »

CLI (expérimental, sans MCP)

`npm i chrome-devtools-mcp@latest -g`	
`chrome-devtools status`	vérifie l'installation
`chrome-devtools new_page "https://example.com"`	
`chrome-devtools take_screenshot 1 --filePath capture.png`	1 = numéro de page
`chrome-devtools stop`	arrête le démon

Conseils d'utilisation

- **Combiner avec Playwright MCP** : Playwright pour les parcours répétables et les tests, Chrome DevTools MCP pour diagnostiquer performance, console, réseau et mémoire.
- **Demander des diagnostics, pas des clics** : sa valeur est dans les traces et les insights structurés, comme dans le panneau Performance de DevTools.
- **Activer `--slim`** quand tu n'as besoin que de naviguer et capturer : le jeu complet d'outils occupe beaucoup de contexte.
- **Données sensibles** : le serveur expose tout le contenu du navigateur au client MCP. Utiliser `--isolated` plutôt que ton profil personnel.
- **Profil persistant** : sans `--isolated`, le profil est gardé dans `~/.cache/chrome-devtools-mcp/`, ce qui garde les connexions entre sessions.

Sources (documentation officielle)

- README chrome-devtools-mcp
- Référence des outils
- Configuration par client
- Playwright vs Chrome DevTools MCP: Driving vs Debugging – Steve Kinney
- Chrome DevTools vs Playwright vs Puppeteer MCP (2026) – MCP.Directory