

Configurer Playwright MCP (contrôle navigateur)

Donner à l'IA un vrai navigateur via Playwright MCP (le standard officiel Microsoft 2026) : naviguer, cliquer, remplir des formulaires, faire des screenshots — sur les pages avec login ou les apps JS-heavy.

Configurer Playwright MCP (contrôle du navigateur)

Playwright MCP donne à ton IA un **vrai navigateur web**. Contrairement à Firecrawl qui lit le HTML d'une page, Playwright peut naviguer, cliquer, remplir des formulaires, se connecter avec ton login, prendre des screenshots — exactement comme toi devant ton écran.

C'est le **standard de fait** pour le contrôle navigateur via MCP en 2026 : maintenu par Microsoft (qui maintient aussi Playwright depuis des années), local, sécurisé, et il « voit » les pages via l'arbre d'accessibilité plutôt que via screenshots — ce qui est beaucoup plus économe en tokens (3 à 4x moins cher qu'une approche screenshot-only).

Pourquoi en plus de Firecrawl ?

Firecrawl est excellent pour **lire** des pages publiques. Mais certains usages nécessitent un vrai navigateur :

- **Pages avec login** — Firecrawl ne peut pas se connecter à ton compte
- **Applications web dynamiques** — Certaines apps chargent tout en JavaScript après login
- **Workflows multi-étapes** — Cliquer, remplir, valider, télécharger
- **Screenshots** — Capturer visuellement une page (utile pour de la veille concurrentielle)
- **Tester un site qu'on est en train de développer** — Naviguer dedans, signaler des bugs

En pratique : installe les deux. L'IA choisit automatiquement le bon outil selon ta demande.

Firecrawl

- ✓ Lire le contenu d'une page
- ✓ Rechercher sur le web
- ✓ Extraire des données structurées
- ✓ Crawler un site entier

Rapide, orienté lecture/extraction

Playwright MCP

- ✓ Naviguer comme un humain
- ✓ Se connecter avec login
- ✓ Cliquer, remplir des formulaires
- ✓ Screenshots + accessibility tree

Interactif, orienté actions

Installation via /connect-mcp

Le plus simple : utiliser le skill /connect-mcp fourni dans ton vault. Lance-le dans OpenCode (ou Claude Code) :

```
/connect-mcp playwright
```

Le skill détecte tes outils installés (OpenCode, Claude Code, Claude Desktop), patche le bon fichier de config (opencode.json ou ~/.claude.json), et configure Playwright MCP via npx (rien à installer manuellement).

Le package officiel utilisé est @playwright/mcp (Microsoft).

Coût : Gratuit. Open-source, tourne 100 % en local sur ta machine.

Si /connect-mcp n'est pas dispo chez toi, demande à OpenCode :

```
Installe le MCP officiel Playwright (@playwright/mcp). Patche mon  
opencode.json (ou ~/.claude.json) pour l'activer. Pas besoin de  
clé API.
```

Tester

Demande à ton IA :

```
Ouvre le site https://example.com dans le navigateur,  
prends un screenshot, et décris-moi ce que tu vois.
```

Tu peux aussi tester des interactions plus avancées :

```
Va sur https://news.ycombinator.com, lis les 5 premiers titres,  
et résume-les.
```

Sur ton tout premier lancement, Playwright va télécharger les binaires du navigateur (~150 Mo) en arrière-plan. C'est normal et ça ne se reproduira plus.

Cas d'usage concrets

Extraire des données d'une app avec login

```
Connecte-toi à mon compte [service] avec l'email xxx et le mot de  
passe xxx  
(que tu peux récupérer via `bw get password "service"`).  
Va dans la section "Rapports".  
Télécharge le rapport du mois dernier et range-le dans 0 INBOX/.
```

Astuce sécurité : plutôt que de coller tes credentials, branche le CLI Bitwarden (cf. la leçon [Zoom sur la sécurité](#)). L'IA récupère le mot de passe à la volée, sans qu'il transite dans ton prompt.

Documenter un process visuel

```
Ouvre [url de mon app].  
Prends un screenshot de chaque étape du process d'inscription.
```

Crée une note dans mon vault avec les screenshots et un descriptif.

Veille visuelle / concurrentielle

Ouvre la page d'accueil de [concurrent].
Prends un screenshot et range-le dans 3 RESSOURCES/Veille/.
Compare avec le dernier screenshot que tu as de ce site.
Dis-moi ce qui a changé.

Tester un site qu'on développe

Le site que je suis en train de coder tourne sur
`http://localhost:3000`.
Va dessus, navigue dans le menu principal, signale-moi tout ce qui
semble cassé ou bizarre visuellement.

Alternatives à connaître

Playwright MCP couvre 95 % des cas. Si tu as un besoin spécifique :

Browserbase MCP

Cloud hosted (pas besoin d'install locale), stealth mode, proxies, session persistante. Payant.
Utile si tu veux exécuter du scraping depuis ton serveur sans installer Chromium dessus.

Chrome DevTools MCP

Pour debug / performance audits / inspection réseau d'un site existant. Pas du tout pour
piloter un navigateur — c'est l'inverse, ça inspecte l'état.

Stagehand

Surcouche au-dessus de Playwright avec des instructions en langage naturel. Intéressant
pour des automatisations longues et stables où tu veux mixer scripting et IA.

mcp-chrome

Se connecte à TON Chrome déjà ouvert (avec tes sessions et cookies). Pratique pour bosser
sur des apps SSO/2FA où relogin à chaque session est lourd.

Migré depuis l'ancien Browser MCP ? L'ancien projet `browser-mcp` (basé sur Puppeteer) est moins maintenu et plus coûteux en tokens. Bascule sur `@playwright/mcp` — la migration prend 30 secondes via `/connect-mcp`.

Sécurité

Playwright MCP tourne **en local sur ta machine**. Tes identifiants ne sont pas envoyés à un serveur tiers. Le navigateur se ferme après chaque session. Tu peux le lancer en mode `--headless` (sans fenêtre visible) ou avec fenêtre pour suivre ce que l'IA fait en temps réel.

Pour les workflows sensibles, combine avec Bitwarden CLI (cf. [Zoom sur la sécurité](#)) : l'IA pioche les credentials au moment d'exécuter, jamais en clair dans ton prompt.

- ☐ J'ai installé Playwright MCP via `/connect-mcp` (ou demande à OpenCode)
- ☐ J'ai testé une navigation simple + un screenshot
- ☐ Je comprends quand utiliser Firecrawl (lecture) vs Playwright (actions)
- ☐ Je sais que je peux mixer Bitwarden CLI + Playwright pour les pages avec login