Eigenflow

GUIDE SABLE · MCP

Le Code Mode de Cloudflare

La nouvelle façon de se servir de MCP, comment brancher le serveur de Cloudflare, et notre test du 17 novembre 2026 : 308 outils MCP pesaient 260 000 tokens chargés d'avance.

PAR

Abyane Nabeebaccus

INSTAGRAM

@abyane_nb

ÉDITION

novembre 2026

Tu as commenté SABLE, voici le guide. Le principe : au lieu d'appeler les outils d'un agent un par un, le modèle écrit un petit programme qui les appelle, et ce programme tourne dans un bac à sable. Cloudflare a appelé ça Code Mode. Ici : le problème, ce que fait Code Mode, comment brancher le serveur de Cloudflare, qui d'autre fait pareil, et ce que tu peux régler dès aujourd'hui dans ton Claude Code, avec notre test du 17 novembre 2026.

Billets de Cloudflare, d'Anthropic et de Goose, docs de Claude Code et de l'API d'Anthropic relus le 17 novembre 2026. On n'a aucun lien avec Cloudflare, Anthropic ni Block.

À savoir avant de lire

- MCP reste. Le modèle s'en sert autrement : il écrit du code qui appelle les outils, au lieu de les appeler un par un.
- Chez nous, les définitions de 308 outils MCP pesaient environ 260 000 tokens avant le premier mot, quand Claude Code les chargeait toutes d'avance. Avec son réglage par défaut, qui les charge à la demande : environ 12 000.
- Le serveur MCP de Cloudflare donne accès à toute son API, plus de 2 500 endpoints, avec deux outils et environ 1 000 tokens. Sans Code Mode, Cloudflare estime le même serveur à 1,17 million de tokens.
- Ces chiffres de Cloudflare sont comptés avec tiktoken, le tokeniseur d'OpenAI. Claude compte autrement.
- Code Mode est jeune : Cloudflare le dit expérimental, son kit en est à la version 0.5.6.
- Ça ne paie pas à tous les coups. Sur des appels qui dépendent chacun du précédent, Anthropic a mesuré un coût 8 % plus élevé.
- Notre test mesure ce que pèsent les outils dans Claude Code. On n'a pas branché le serveur de Cloudflare : ses chiffres sont les siens.

1. MCP en deux phrases

MCP (Model Context Protocol) est le protocole ouvert qui branche un agent IA sur des outils : ton agenda, ton CRM, ta boutique, un navigateur. Le serveur MCP publie la liste de ses outils (requête `tools/list`), l'agent les découvre et les appelle (`tools/call`).

Anthropic l'a lancé en novembre 2024. Depuis le 9 décembre 2025, il appartient à l'Agentic AI Foundation, créée par la Linux Foundation.

2. Le problème : les allers-retours et la liste d'outils

Les allers-retours

Tu demandes à ton agent : « trouve Léa et sors ses commandes ». En appel classique :

1. Le modèle appelle l'outil qui cherche une cliente. Ton application exécute l'appel et renvoie toute la fiche au modèle.

2. Le modèle lit la fiche, recopie l'identifiant de Léa et appelle l'outil des commandes. Toute la liste repart au modèle.

3. Le modèle lit la liste et te répond.

Trois passages du modèle, et chaque résultat intermédiaire s'ajoute à la conversation, relue en entier à chaque passage. Chez Anthropic, contrôler les notes de frais de 20 employés demande 20 allers-retours, et faire passer la transcription d'une réunion de deux heures d'un outil à l'autre peut ajouter 50 000 tokens.

La liste d'outils

Avant même ta question, le modèle reçoit la définition de chaque outil branché : nom, description, paramètres. Anthropic a fait le compte pour cinq serveurs courants, dont GitHub (35 outils, environ 26 000 tokens) et Slack (11 outils, environ 21 000) : 58 outils, environ 55 000 tokens avant le début de la conversation. Chez Anthropic, les définitions d'outils sont déjà montées à 134 000 tokens avant optimisation (billet du 24 novembre 2025).

Notre test du 17 novembre 2026

On a mesuré la même chose sur notre Claude Code (version 2.1.325, Opus 5.5), dans un dossier vide, avec la phrase « Réponds seulement : ok ». Trois passes :

- Recherche d'outils coupée (`ENABLE_TOOL_SEARCH=false`), les 308 outils MCP de nos serveurs chargés d'avance : 297 334 tokens d'entrée.
- Réglage par défaut de Claude Code, outils chargés à la demande : 49 191 tokens.
- Aucun serveur MCP : 37 060 tokens.

Les définitions des 308 outils pèsent donc environ 260 000 tokens, à peu près 845 par outil, avant le premier mot. Un seul connecteur, Higgsfield (génération d'images et de vidéos), en apporte 175.

Au prix de liste de l'API, ce premier message coûte 2,38 $ tout chargé, contre 0,24 $ à la demande : c'est la mise en cache des 297 000 tokens. Ensuite, chaque requête les relit depuis le cache, environ 0,06 $ contre 0,01 $ (notre calcul, pas une mesure). Nous sommes sur abonnement : rien n'a été facturé, ce sont les montants que calcule Claude Code (champ `costUSD`). Et quelle que soit l'offre, 297 000 tokens occupent près de 30 % de la fenêtre d'Opus 5.5, et dépassent celle d'un modèle à 200 000 tokens.

---

3. Ce que fait Code Mode

Cloudflare l'a présenté le 26 septembre 2025, dans un billet titré « Code Mode: the better way to use MCP ». Le principe, en trois temps :

1. L'agent lit le schéma de chaque serveur MCP et le transforme en API TypeScript, avec la documentation en commentaires.

2. Le modèle ne voit plus la liste des outils. Il a un seul outil, qui exécute du code, et il écrit un programme contre cette API.

3. Le programme tourne dans un bac à sable : un isolate V8 de Cloudflare, sans accès à internet, qui ne peut parler qu'aux serveurs MCP qu'on lui a branchés. Seul ce que le programme affiche ou renvoie revient au modèle.

Dans cette première version, toute l'API TypeScript est encore chargée dans le contexte. La recherche à la demande arrive avec le serveur de février 2026 (partie 4).

L'exemple de Léa

Prends un serveur de boutique avec deux outils, `find_customers` et `list_orders` (noms inventés pour l'exemple). Le modèle reçoit leurs déclarations TypeScript, puis écrit ceci, dans la forme que montre la doc de Cloudflare :

```
À COLLER

async () => {
  const clientes = await codemode.find_customers({ name: "Léa" });
  const lea = clientes[0];
  if (!lea) return null;
  const commandes = await codemode.list_orders({ customerId: lea.id });
  return commandes.map((c) => ({
    numero: c.number, date: c.date, total: c.total,
  }));
};
```

Ce qui revient au modèle : trois champs par commande. Ni la fiche complète de Léa, ni ses adresses, ni le détail des lignes. Il a écrit le code en un passage, l'a lancé en un seul appel d'outil, puis te répond : deux passages au lieu de trois. Dans le bac à sable, le code a quand même appelé les deux outils : le gain se fait côté modèle, le serveur reçoit autant de requêtes.

Pourquoi ça marche mieux

L'explication de Cloudflare : « LLMs have seen a lot of code. They have not seen a lot of tool calls. » Les modèles ont appris sur le code de millions de projets open source ; les appels d'outils, ils ne les ont vus que dans des exemples fabriqués par leurs développeurs. Cloudflare l'avance comme explication, personne ne l'a mesurée. La doc d'Anthropic dit la même chose de Claude.

Plus terre à terre : une boucle, un filtre ou un tri ne coûtent aucun token quand c'est le code qui les fait.

4. Le serveur MCP de Cloudflare

Le 20 février 2026, Cloudflare a appliqué l'idée à sa propre API. Elle compte plus de 2 500 endpoints (3 773 dans le schéma public le 17 novembre, d'après notre comptage). Ses serveurs MCP par produit ne suivaient plus.

Le nouveau serveur n'expose que deux outils pour l'API :

- `search` : le modèle écrit du JavaScript qui fouille la spécification OpenAPI, côté serveur, et ne récupère que les endpoints utiles. La spécification entière n'entre jamais dans le contexte.
- `execute` : le modèle écrit du JavaScript qui appelle l'API, enchaîne les requêtes, gère la pagination et renvoie le résultat.

Le tableau de Cloudflare, compté avec tiktoken, pour une fenêtre de 200 000 tokens :

- Spécification OpenAPI brute dans le prompt : environ 2 000 000 tokens.
- MCP classique, 2 594 outils avec schémas complets : 1 170 523 tokens, 585 % de la fenêtre.
- MCP classique, paramètres obligatoires seulement : 244 047 tokens, 122 %.
- Code Mode, 2 outils : 1 069 tokens, 0,5 %.

Le 1,17 million correspond à un serveur que Cloudflare n'a jamais publié : c'est une estimation. Aujourd'hui, le serveur expose aussi `docs` (recherche dans la doc de Cloudflare) et `whoami` (le compte connecté), et son README annonce environ 1 100 tokens. Chaque résultat est coupé vers 6 000 tokens. Et le modèle cherche avant d'agir : dans l'exemple du billet, protéger un site contre les attaques DDoS a pris quatre appels d'outils.

Le brancher

L'adresse : https://mcp.cloudflare.com/mcp ; à la connexion, Cloudflare te demande de te connecter et de choisir les permissions que tu donnes à l'agent. Pour un script ou une CI, un jeton API Cloudflare passé en en-tête Authorization marche aussi.

Dans Claude Code, la commande générique de la doc de Claude Code, avec l'adresse de Cloudflare :

```
À COLLER
claude mcp add --transport http cloudflare-api https://mcp.cloudflare.com/mcp
```

Puis, dans une session, ouvre le panneau des serveurs pour faire la connexion (ou `claude mcp login cloudflare-api` dans ton terminal) :

```
À COLLER
/mcp
```

La doc de Cloudflare propose aussi, pour Claude Code, un plugin qui regroupe ses serveurs MCP et ses skills : `/plugin marketplace add cloudflare/skills`.

Pour coller la configuration JSON dans un fichier `.mcp.json`, garde la ligne `"type": "http"` : le billet de Cloudflare l'omet, et Claude Code refuse une entrée qui a une adresse sans type. La version du README de Cloudflare :

```
À COLLER
{
  "mcpServers": {
    "cloudflare-api": {
      "type": "http",
      "url": "https://mcp.cloudflare.com/mcp"
    }
  }
}
```

Dans l'appli Claude ou sur claude.ai : Customize, Connectors, Add, Add custom connector. Un nom, l'adresse, Add, puis Connect. L'offre Free n'accepte qu'une connexion personnalisée.

Ce serveur ne sert que si ton site, ton DNS ou tes Workers sont chez Cloudflare. Commence par des permissions en lecture seule : l'agent n'aura que ce que tu coches.

Les portails MCP

Sur Cloudflare One, les portails MCP regroupent plusieurs serveurs derrière une adresse et proposent Code Mode. Depuis le 30 juillet 2026, un nouveau portail démarre en Opt-in : le client ajoute `?codemode=search_and_execute` à l'adresse, sauf si l'administrateur l'a mis en On by default ou Enforced.

Pour les développeurs

Le kit est open source : `@cloudflare/codemode` sur npm (licence MIT), version 0.5.6 du 29 octobre 2026. `codeMcpServer()` place un serveur MCP existant derrière un seul outil `code`, `openApiMcpServer()` publie une grosse API en `search` et `execute`. Le code tourne dans des Dynamic Workers, réservés au forfait payant Workers.

---

5. Qui d'autre le fait

Anthropic

Le 4 novembre 2025, Anthropic a publié « Code execution with MCP », qui cite le Code Mode de Cloudflare. Son exemple : des outils présentés comme des fichiers de code que l'agent lit au besoin, et 150 000 tokens ramenés à 2 000, soit 98,7 % de moins.

Le 24 novembre 2025, deux fonctions de l'API qui nous intéressent ici :

- Tool Search Tool : les outils marqués `defer_loading` ne sont chargés que si Claude les cherche. Sur leur exemple, 85 % de tokens en moins.
- Programmatic Tool Calling (PTC) : Claude écrit du Python qui appelle tes outils dans un conteneur d'exécution. Les résultats intermédiaires restent dans le code. Sur des tâches de recherche complexes, 43 588 tokens en moyenne ramenés à 27 297.

PTC est stable aujourd'hui, et Cloudflare le range lui-même avec Code Mode. Les différences comptent :

- Le code est en Python (JavaScript ou TypeScript chez Cloudflare et Goose).
- Il tourne dans le conteneur d'Anthropic, dont les données sont gardées jusqu'à 30 jours.
- Les outils appelables sont ceux que tu déclares dans ta requête, avec `allowed_callers`. Ceux du connecteur MCP de l'API en sont exclus.
- À chaque appel d'outil, le code se met en pause et l'API te rend la main : ton application exécute l'outil et renvoie le résultat. Le modèle, lui, n'est pas relancé entre deux appels.

Chiffres de la doc : environ 38 % de tokens d'entrée facturés en moins sur un banc de 75 outils, et 20 à 40 % d'économie typique entre 10 et 49 outils.

Goose

Goose, l'agent open source né chez Block, a ajouté une extension Code Mode dans sa version 1.17.0, le 15 décembre 2025. Le modèle y a trois outils : `list_functions` pour lister, `get_function_details` pour lire un schéma, `execute_javascript` pour lancer le code, dans un moteur JavaScript embarqué (Boa) sans accès aux fichiers, au shell ni au réseau. Sa doc conseille l'appel classique pour 1 à 3 extensions, Code Mode à partir de 5.

FastMCP

Si tu écris tes serveurs en Python : FastMCP propose un transform CodeMode depuis sa version 3.1.0 (mars 2026), marqué expérimental.

Claude Code

Claude Code règle déjà la liste d'outils. Par défaut, il ne charge que les noms des outils et les instructions des serveurs, et la définition complète arrive quand Claude cherche l'outil. C'est notre passe à 49 191 tokens. Les allers-retours, eux, restent : les outils sont toujours appelés un par un.

---

6. Ce que tu peux faire dès aujourd'hui

Regarder ce qui remplit ta fenêtre

Dans Claude Code :

```
À COLLER

/context
```

La commande affiche l'occupation du contexte en grille, avec des suggestions pour les outils lourds. Puis :

```
À COLLER
/mcp
```

pour voir chaque serveur et son nombre d'outils. Un serveur qui ne sert pas se coupe dans ce panneau sans perdre sa configuration, ou s'enlève avec `claude mcp remove` suivi de son nom.

Laisser la recherche d'outils active

Elle est active par défaut. Ne mets `ENABLE_TOOL_SEARCH` à `false` que pour mesurer. Les valeurs documentées :

- non défini ou `true` : outils chargés à la demande ;
- `auto` : chargés d'avance tant qu'ils pèsent moins de 10 % de la fenêtre, à la demande au-delà ; `auto:5` pour un seuil de 5 % ;
- `false` : tout chargé d'avance.

Deux cas la coupent sans bruit : la variable `CLAUDE_CODE_DISABLE_EXPERIMENTAL_BETAS`, et une passerelle (`ANTHROPIC_BASE_URL`) qui refuse ses champs. Claude Code charge alors tout d'avance.

Limiter les gros résultats

`MAX_MCP_OUTPUT_TOKENS` plafonne un résultat d'outil MCP : 25 000 tokens par défaut, avertissement dès 10 000.

Refaire notre mesure

Dans un dossier vide. La question part 60 secondes après le lancement, pour laisser les serveurs se connecter : notre premier essai, lancé sans attendre, n'avait chargé que 57 outils sur 308.

```
À COLLER
MSG='{"type":"user","message":{"role":"user","content":"Réponds seulement :
ok"}}'
(sleep 60; echo "$MSG") | ENABLE_TOOL_SEARCH=false claude -p --input-format
stream-json --output-format stream-json --verbose --max-turns 1 > tout-
charge.jsonl
(sleep 60; echo "$MSG") | claude -p --input-format stream-json --output-format
stream-json --verbose --max-turns 1 > a-la-demande.jsonl
```

Dans la dernière ligne de chaque fichier (type result), additionne `input_tokens`, `cache_creation_input_tokens` et `cache_read_input_tokens`. Sur une clé API, la passe tout chargé se paie : 2,38 $ chez nous.

7. Limites et pièges

- **Le bac à sable isole le code, et seulement le code.** Le programme peut faire tout ce que permettent les outils et le jeton qu'on lui confie. Un jeton qui peut supprimer une zone DNS laisse le code la supprimer. Donne le minimum.
- **Le code généré tourne sans relecture.** Anthropic le rappelle : il faut un environnement isolé, des limites de ressources et une surveillance. Sa doc PTC signale aussi un risque d'injection si un résultat d'outil contient du texte que le programme exécute.
- **tiktoken n'est pas le tokeniseur de Claude.** Les 1 069 et 1 170 523 tokens de Cloudflare viennent de l'outil d'OpenAI. Même entre ses propres modèles, Anthropic note environ 30 % d'écart depuis Opus 4.7. Pour un chiffre fiable sur Claude, l'API a un endpoint de comptage, `count_tokens`.
- **Code Mode est jeune.** La doc de Cloudflare, mise à jour le 27 octobre 2026 : « Code Mode is experimental and may have breaking changes. Use caution in production. » Le kit est en 0.5.6, FastMCP le marque expérimental.
- **Il ne paie pas toujours.** D'après la doc d'Anthropic, sur τ²-bench, où chaque étape dépend du résultat précédent, le score ne bouge pas et le coût monte d'environ 8 %. Peu d'appels, petits résultats, premier tour d'une conversation : le surcoût du conteneur et du script peut dépasser le gain.
- **Deux Code Mode l'un dans l'autre.** Si ton client fait déjà du Code Mode, Cloudflare conseille d'ajouter `?codemode=false` à l'adresse de son serveur. Tu retombes alors à environ 3 000 outils et 244 000 tokens.

Sources

- Code Mode: the better way to use MCP, Cloudflare, 26 septembre 2025.
- Code Mode: give agents an entire API in 1,000 tokens, Cloudflare, 20 février 2026.
- Le dépôt du serveur MCP de Cloudflare, README du 6 novembre 2026.
- Cloudflare's own MCP servers et Build a Code Mode MCP server, doc Cloudflare, 27 octobre 2026.
- Code Mode MCP in goose, 15 décembre 2025, et la doc Code Mode de Goose.
- Programmatic tool calling, doc de l'API d'Anthropic.
- Introducing advanced tool use on the Claude Developer Platform, Anthropic, 24 novembre 2025.
- Code execution with MCP, Anthropic, 4 novembre 2025.
- Connect Claude Code to tools via MCP, doc de Claude Code, partie sur la recherche d'outils.