Guide d'installation Beta

Opquast MCP Server

Version 1.0.0 — 09/04/2026

1. Prérequis

Avant de commencer, assurez-vous de disposer des éléments suivants :

☐ **Claude Code CLI** — npm install -g @anthropic-ai/claude-code
`https://docs.anthropic.com/en/docs/claude-code`

☐ **Node.js 18+** — Pour l'outil d'inspection des pages (Playwright ou Chrome DevTools). Recommandé : v20 ou v22.
`https://nodejs.org`

☐ **Système compatible** — macOS 14+, Windows 11+, ou Linux (Debian 12+, Ubuntu 22.04+). Requis pour Playwright. Sur Linux, le script installe automatiquement les dépendances système.

☐ **Google Chrome** — Uniquement si vous choisissez Chrome DevTools (optionnel avec Playwright)
`https://www.google.com/chrome/`

Obtenez votre token personnel

Le token commence par `oqs_mcp_` et n'est affiché **qu'une seule fois** — copiez-le immédiatement.

Demander un token

En version bêta, le token doit être activé par l'équipe Opquast avant utilisation.
Gérez vos tokens sur login.opquast.com/mcp/tokens/

2. Installation

Installation automatique (recommandé)

Exécutez cette commande dans votre terminal en remplaçant `<TOKEN>` par votre token :

```
# Installation globale (disponible dans tous vos projets)
curl -sSL https://mcp.opquast.com/install.sh | bash -s -- <TOKEN>

# Ou installation locale (projet courant uniquement)
curl -sSL https://mcp.opquast.com/install.sh | bash -s -- <TOKEN> --local
```

Ce script configure automatiquement :

- Le serveur **Opquast MCP** (accès aux 245 règles du référentiel)
- Un **outil d'inspection des pages** au choix :
  - **Playwright** (recommandé) — installe son propre navigateur Chromium
  - **Chrome DevTools** — utilise Google Chrome installé sur la machine
- Les **autorisations d'outils** (pré-approbation des appels MCP, avec votre accord)
- L'envoi optionnel de **statistiques anonymes** (scope, outil choisi, autorisations — aucune donnée personnelle)

> **Autorisations :** Lors de l'installation, le script vous propose de pré-autoriser les outils MCP. Si vous acceptez, un fichier `.claude/settings.json` est créé dans le répertoire courant avec les permissions appropriées. Les appels aux outils seront alors automatiquement approuvés dans ce projet. Vous pouvez aussi utiliser les flags `--authorize`, `--stats`, `--playwright` ou `--chrome-devtools` pour un mode non-interactif.

> **Linux :** Si vous choisissez Playwright, le script installe automatiquement les dépendances système nécessaires (`--with-deps`). Cette étape peut demander des droits administrateur (`sudo`).

Désinstallation

```
curl -sSL https://mcp.opquast.com/install.sh | bash -s -- --uninstall
```

Le script détecte automatiquement les serveurs installés et leur scope (global ou projet). Le serveur Opquast est toujours désinstallé. Pour l'outil d'inspection (Playwright ou Chrome DevTools), une confirmation vous sera demandée.

Installation manuelle

Si vous préférez configurer manuellement (ajoutez `-s local` pour le projet courant uniquement) :

```
# Serveur Opquast MCP
claude mcp add opquast https://mcp.opquast.com/mcp \
    -t http -s user \
    -H "Authorization: Bearer <TOKEN>"
```

```
# Option A : Playwright (recommandé — installe Chromium)
claude mcp add playwright \
    -t stdio -s user \
    -- npx -y @playwright/mcp@latest --headless
npx -y playwright install chromium

# Option B : Chrome DevTools (utilise Chrome installé)
claude mcp add chrome-devtools \
    -t stdio -s user \
    -- npx -y chrome-devtools-mcp@latest --isolated
```

Pour pré-autoriser les outils MCP, ajoutez dans `.claude/settings.json` (à la racine de votre projet) :

```
{
  "permissions": {
    "allow": [
      "mcp__opquast__*",
      "mcp__playwright__*"
    ]
  }
}
```

Remplacez `mcp__playwright__*` par `mcp__chrome-devtools__*` si vous avez choisi Chrome DevTools.

3. Premiers pas

Lancer Claude Code

Ouvrez votre terminal et lancez Claude Code :

```
claude
```

> **Astuce :** Le navigateur est lancé automatiquement par l'outil d'inspection (Playwright ou Chrome DevTools) avec un profil isolé. Vous n'avez pas besoin de l'ouvrir manuellement.

Exemples de prompts

Voici quelques exemples de ce que vous pouvez demander :

- *« Audite https://example.com selon les règles Opquast »*
- *« Quelles sont les règles Opquast sur les formulaires ? »*
- *« Donne-moi la règle n°42 avec les instructions pour la vérifier »*
- *« Audite cette page sur le thème Sécurité »*
- *« Quels sont les thèmes du référentiel Opquast ? »*

4. Outils disponibles

Le serveur MCP Opquast expose 7 outils :

Outil	Description
`get_rule`	Récupérer une règle par son numéro (1–245). Formats : résumé, détaillé ou actionnable.
`search_rules`	Rechercher des règles par mot-clé, avec filtrage optionnel par thème ou tag.
`get_rules_by_theme`	Obtenir toutes les règles d'un thème donné.
`list_themes`	Lister les 14 thèmes disponibles avec le nombre de règles par thème.
`list_versions`	Lister les versions du référentiel disponibles.
`get_audit_checklist`	Obtenir le protocole d'audit structuré pour une URL.
`generate_audit_report`	Générer un rapport d'audit HTML ou JSON avec synthèse et recommandations.

5. Lancer un audit

Workflow type

1. **Lancez Claude Code** dans votre terminal
2. **Demandez l'audit** : *« Audite https://example.com selon les règles Opquast »*

3. **Attendez** — Claude évalue les 245 règles (environ 5 à 15 minutes)
4. **Consultez le rapport** — un lien vers le rapport HTML est fourni

> **Astuce :** Pour un audit ciblé, précisez un thème : *« Audite cette page sur le thème Formulaires »*

Thèmes du référentiel

Le référentiel Opquast *Qualité numérique* comporte 245 règles réparties en 14 thèmes :

- Contenus
- Données personnelles
- E-Commerce
- Formulaires
- Identification et contact
- Images et médias
- Internationalisation
- Liens
- Navigation
- Newsletter
- Présentation
- Serveur et performances
- Structure et code
- Sécurité

6. Consommation de tokens

Chaque interaction avec Claude Code consomme des tokens (unités de traitement du modèle IA). Voici les ordres de grandeur selon le type d'utilisation :

Action	Estimation	Détail
Consulter une règle	~1 000 tokens	Requête simple, réponse courte.
Rechercher des règles par mot-clé	~2 000 tokens	Dépend du nombre de résultats.

Action	Estimation	Détail
Audit thématique (1 thème)	~30 000 tokens	Inspection de la page + évaluation de 10 à 30 règles.
Audit complet (245 règles)	~200 000 tokens	Inspection approfondie + évaluation de toutes les règles + génération du rapport.

Note : Ces estimations sont indicatives. La consommation réelle dépend de la complexité de la page auditée et du nombre d'échanges avec le modèle.

Astuce : Pour réduire la consommation, auditez un thème à la fois plutôt que les 245 règles d'un coup.

7. Dépannage

Erreur 401 — Invalid, expired, or inactive token

Vérifiez que votre token commence par oqs_mcp_ et qu'il a été activé par l'équipe Opquast. Vous pouvez vérifier le statut de vos tokens sur login.opquast.com/mcp/tokens/. Relancez l'installation avec le bon token : curl -sSL https://mcp.opquast.com/install.sh | bash -s -- <VOTRE_TOKEN>

L'outil d'inspection ne se connecte pas

Si vous utilisez Playwright, relancez npx -y playwright install chromium. Si vous utilisez Chrome DevTools, vérifiez que Google Chrome est installé. Le serveur lance automatiquement un navigateur avec un profil isolé.

Claude ne trouve pas les outils Opquast

Vérifiez la configuration MCP : claude mcp list. Le serveur 'opquast' doit apparaître dans la liste.

Timeout lors de l'audit

Certaines pages lourdes peuvent dépasser le timeout par défaut. Essayez d'auditer un thème spécifique plutôt que l'audit complet.

Le rapport ne s'affiche pas

Les rapports expirent après 1 heure. Si le lien ne fonctionne plus, relancez la génération du rapport.

Playwright : erreur de dépendances système (Linux)

Sur Linux, relancez l'installation avec : npx -y playwright install --with-deps chromium. Cela installe automatiquement les bibliothèques système nécessaires (libgbm, libatk, etc.).

Session perdue après mise à jour du serveur

C'est normal : lors d'un redéploiement, le serveur redémarre et les sessions en cours sont interrompues. Le client se reconnecte automatiquement avec une nouvelle session. Si ce n'est pas le cas, relancez simplement votre conversation.

> **Note :** Pendant la phase beta, les rapports générés expirent après 1 heure. Téléchargez-les dès réception si vous souhaitez les conserver.

8. Compatibilité avec d'autres clients IA

Le serveur Opquast MCP utilise le protocole standard **MCP (Model Context Protocol)** via transport HTTP. Il est donc compatible avec **tout client supportant MCP**, pas uniquement Claude Code.

Clients testés ou compatibles

Client	Type	Notes
Claude Code	CLI	**Client testé** pendant la beta
Claude Desktop	App	Support natif MCP

Client	Type	Notes
Claude.ai	Web	Via les intégrations MCP (settings)
VS Code + Copilot	IDE	Agent mode avec serveurs MCP
Cursor	IDE	Support MCP via Composer
Windsurf	IDE	Support Streamable HTTP
Cline	Extension VS Code	Support Streamable HTTP
ChatGPT	Web/App	Support MCP (OpenAI)
JetBrains AI	IDE	Plugin AI Assistant ou Junie
Gemini CLI	CLI	Google, support MCP
OpenAI Codex CLI	CLI	Via `~/.codex/config.toml`

> **Configuration :** Quel que soit le client, la connexion nécessite : l'URL du serveur (`https://mcp.opquast.com/mcp`), le transport **HTTP**, et le header `Authorization: Bearer <TOKEN>` . Consultez la documentation de votre client pour la syntaxe exacte.

Exemple : OpenAI Codex CLI

Ajoutez votre token en variable d'environnement puis configurez `~/.codex/config.toml` :

```
# Dans ~/.bashrc ou ~/.zshrc
export OPQUAST_MCP_TOKEN="oqs_mcp_VOTRE_TOKEN"
```

```
# Dans ~/.codex/config.toml
[mcp_servers.opquast]
url = "https://mcp.opquast.com/mcp"
bearer_token_env_var = "OPQUAST_MCP_TOKEN"
enabled = true
startup_timeout_sec = 20
tool_timeout_sec = 120
```

> **Note :** Seul Claude Code a été testé en profondeur pendant la beta. Les autres clients devraient fonctionner mais peuvent présenter des différences dans la gestion des outils MCP. N'hésitez pas à nous signaler tout problème.

La liste complète des clients MCP est disponible sur modelcontextprotocol.io/clients.

9. Contact et support

Email : support@opquast.com

Référentiel : checklists.opquast.com

Ce guide en ligne : https://mcp.opquast.com/guide

Ce guide en PDF : https://mcp.opquast.com/guide.pdf