Installer tes skills de SEO technique

Où déposer les six skills, comment vérifier que Claude Code les voit, comment sortir un export Search Console propre, et la première demande à écrire.

Timothée Boussardon · Organikk.co · Août 2026

Sommaire

À lire en premier

Cette installation pose les six skills et te fait sortir ton premier rapport. Compte dix minutes, dont huit d'installation d'outils si tu ne les as pas déjà.

- Les skills sont des dossiers. Tu les déposes dans `~/.claude/skills/`.
- Claude Code les détecte au démarrage suivant. Il n'y a rien à configurer.
- Un skill a besoin de deux outils en plus : Lighthouse et jq.
- Deux skills ont besoin d'un export Search Console.
- Les trois autres tournent avec une URL, un sitemap ou ton code.
- Tu déclenches un skill en écrivant ta demande, pas avec une commande.
- Chaque skill rend un rapport en markdown.
- Aucun skill ne publie ni ne modifie ton site en ligne.

Les six skills

Indexation, Core Web Vitals, données structurées, maillage par la structure du site, maillage par la Search Console, cannibalisation.

Chacun prend une entrée précise, applique la même suite d'étapes à chaque fois et rend un rapport.

Le contenu complet des six fichiers `SKILL.md` est publié dans l'article 6 tâches de SEO technique déléguées à Claude.

1. Ce qu'il te faut

Quatre prérequis, dont un seul est bloquant.

Claude Code installé sur ton poste.

Un terminal ouvert dans le dossier de ton projet, ou dans n'importe quel dossier si tu audites un site en ligne.

Node.js, pour installer Lighthouse.

Un accès Search Console au site, pour deux des six skills.

Sans Search Console, quatre skills sur six tournent quand même.

2. Déposer les six skills

Un skill est un dossier qui contient un fichier SKILL.md. C'est ce fichier que Claude Code lit.

Crée le dossier des skills s'il n'existe pas encore.

```
mkdir -p ~/.claude/skills/
```

Crée un dossier par skill.

```
cd ~/.claude/skills/
mkdir -p indexation-check
mkdir -p seo-core-web-vitals
mkdir -p seo-donnees-structurees
mkdir -p maillage-systeme
mkdir -p maillage-interne-gsc
mkdir -p seo-cannibalisation
```

Ouvre l'article 6 tâches de SEO technique déléguées à Claude, qui publie les six fichiers en entier.

Copie le contenu de chaque `SKILL.md` et colle-le dans un fichier `SKILL.md` à l'intérieur du dossier qui porte le même nom.

Ne renomme pas les dossiers. Le nom du dossier est le nom du skill.

Un dossier vide ou un fichier nommé autrement que SKILL.md ne sera jamais lu.

3. Vérifier que Claude Code les voit

Deux contrôles avant de lancer quoi que ce soit.

Liste le dossier.

```
ls ~/.claude/skills/
```

Tu dois voir les six noms.

```
indexation-check
seo-core-web-vitals
seo-donnees-structurees
maillage-systeme
maillage-interne-gsc
seo-cannibalisation
```

Ferme Claude Code, puis rouvre-le. Il lit les skills au démarrage.

Un skill qui n'apparaît pas dans cette liste ne se déclenchera jamais, quoi que tu écrives.

4. Installer Lighthouse et jq

Le skill Core Web Vitals appelle Lighthouse en local et découpe ses résultats avec jq.

```
npm install -g lighthouse
brew install jq
```

Vérifie que les deux répondent.

```
lighthouse --version
jq --version
```

Tu ne veux rien installer en global ? Remplace `lighthouse` par `npx lighthouse` dans les commandes.

Le premier lancement est plus lent, le résultat est identique.

Les cinq autres skills n'ont besoin d'aucun de ces deux outils.

5. Sortir un export Search Console propre

Deux skills lisent tes données de clics et d'impressions. Ils lisent le CSV d'origine, pas une version retravaillée.

Les deux skills qui lisent ta Search Console

Le maillage sort la hiérarchie des pages mères, filles et petites-filles, puis la liste des liens à poser avec leur ancre et le contexte de la phrase.

La cannibalisation classe les conflits par type et tranche : redirection 301, fusion, différenciation, ou rien du tout.

Ouvre la Search Console, onglet Performances.

Choisis la période. Trois mois pour la cannibalisation, six mois pour le maillage.

Clique sur Exporter, puis sur CSV.

L'archive contient un fichier par onglet. Tu as besoin de Pages et de Requêtes.

Dézippe-la dans le dossier de ton projet.

Ne réenregistre pas les fichiers depuis un tableur. Les skills lisent les colonnes d'origine.

6. La première demande

Il n'y a pas de commande à retenir. Tu écris ce que tu veux.

> audit indexation de mon-site.fr, la liste des URLs est dans sitemap.xml

Claude Code reconnaît la demande et charge le skill correspondant.

Les cinq autres se déclenchent de la même façon.

> audit Core Web Vitals de https://mon-site.fr/sitemap.xml

> audit de cannibalisation, l'export GSC est dans ./gsc-requetes.csv

> plan de maillage interne à partir de ./gsc-pages.csv

> audit du maillage interne du site, sans passer par la Search Console

> mets en place les données structurées JSON-LD sur ce projet

Commence par l'indexation. Tant qu'une page est bloquée, ce que mesurent les autres skills ne sert à rien.

7. Lire le rapport

Chaque skill rend un fichier markdown, construit toujours de la même façon.

Les anomalies critiques sont en tête, les mineures ensuite, les corrections classées par priorité.

Le skill d'indexation distingue « non indexée » et « non testable ». Une page non testable n'est pas un problème, c'est une limite de mesure.

Le skill de cannibalisation ne recommande jamais une redirection 301 sans avoir regardé les chiffres des deux pages.

Le seul skill qui écrit dans ton code est celui des données structurées. Claude Code te demande l'autorisation avant chaque écriture.

8. Les pannes fréquentes

Six erreurs reviennent à l'installation. Chacune a une cause unique.

Claude ne trouve pas le skill.

Le dossier est ailleurs que dans `~/.claude/skills/`, ou Claude Code n'a pas été relancé depuis la copie.

`lighthouse: command not found`

Node n'est pas installé, ou l'installation globale a échoué. Passe par `npx lighthouse`.

Le sitemap ressort vide.

L'URL pointe sur un index de sitemaps et pas sur un sitemap d'URLs. Donne le sitemap enfant.

Le rapport dit « non testable » partout.

Google limite les requêtes automatisées. Relance plus tard, ou branche l'API Search Console officielle.

Le skill ne trouve pas les colonnes du CSV.

Le fichier a été ouvert puis réenregistré par un tableur. Repars du CSV d'origine.

Le skill tourne mais ne trouve rien.

Vérifie la période de l'export. Trois mois de données sur un site neuf ne suffisent pas à faire ressortir un conflit.

Les six skills installés, tu as le socle technique. La suite est éditoriale : mots-clés, modèles de pages, rédaction.

Les autres guides sont sur organikk.co/guides.

Si tu veux qu'on installe ce système ensemble sur ton site, on peut se faire un call : cal.com/tim-boussardon-yzrrb1/30min

---

Une question sur l'installation : contact@organikk.co