investigate

investigate

> *Catalogue généré le 2026-05-11*

En une phrase

Une méthode rigoureuse pour débugger en quatre phases : Claude refuse de réparer un bug sans avoir d'abord trouvé sa cause profonde.

Quand l'utiliser

- Quand un truc « marchait hier et ne marche plus » et que tu ne comprends pas pourquoi
- Quand tu vois une erreur 500, un message d'erreur ou un comportement inattendu
- Quand tu as déjà essayé deux ou trois réparations rapides et que ça ne tient pas
- Quand le bug semble revenir à chaque fois, comme un jeu de taupes
- Avant de fermer un ticket de bug, pour t'assurer que la vraie cause a été traitée

Comment l'invoquer

- **Slash command** : `/investigate` (à taper dans Claude Code)
- **Phrases déclencheurs (texte)** : "debug this" / "fix this bug" / "why is this broken" / "investigate this error" / "root cause analysis"
- **Auto-invocation** : □ Oui — Claude invoque ce skill automatiquement quand tu signales une erreur, une stack trace, un comportement bizarre, ou un dysfonctionnement.

Description détaillée

`/investigate` applique une règle stricte appelée « Iron Law » : pas de correctif sans cause profonde identifiée. C'est la différence entre éteindre la flamme et trouver la fuite de gaz. Claude suit quatre phases : enquête (lire les symptômes, les logs, les changements récents), analyse (matcher avec des patterns connus comme race condition, cache obsolète, null propagé), hypothèse (vérifier l'idée avant d'écrire du code), implémentation (le correctif minimal + un test de régression).

Le skill verrouille aussi automatiquement le périmètre de modification au dossier concerné par le bug, pour t'éviter de changer du code non lié. Il vérifie les enseignements (« learnings ») accumulés par Claude sur tes projets précédents — si tu as déjà rencontré un bug similaire, il s'en sert.

Règle des trois échecs : si trois hypothèses successives sont fausses, Claude s'arrête et te demande s'il faut continuer, escalader, ou ajouter du logging et observer. À la fin tu obtiens un rapport structuré (symptôme, cause, correctif, preuve que ça marche, test de non-régression). C'est plus lent que « répare-moi ça vite » mais ça évite les bugs qui reviennent.

Source

- **Plugin** : `gstack`
- **Nom interne** : `investigate`
- **Fichier** : `/home/thymon/.claude/skills/gstack/investigate/SKILL.md`

Revision #2
Created 2026-05-11 21:18:35 UTC by thymon
Updated 2026-05-11 21:36:21 UTC by thymon