● GITHUB AGENT · RAECHAL AI

Raechal reads your repo and opens SEO pull requests.

Connect a repository and talk to it in plain English. It reads your actual source, finds the SEO problems in your templates and content, and proposes the fix as a pull request you review and merge. Nothing lands on main without you.

19	PR	5cr	∞
AGENT TOOLS	EVERY CHANGE	PER MESSAGE	LIFETIME ACCESS

GETTING STARTED

Three steps to a connected repo

Unlock once, install the GitHub App, then chat. Your first two repositories are included in the unlock.

STEP 01

Unlock the agent

A one-time activation that never expires. Includes your first two repositories.

`75 credits · lifetime` `2 repos included`

STEP 02

Install the GitHub App

Authorise the Raechal GitHub App on the repositories you choose. You pick exactly which repos it can see — nothing else is visible to it.

`You choose the repos`

STEP 03

Pick a repo and chat

Select the active repository, then ask for an audit or a fix. Add more repositories any time.

`+60 / extra repo`

THE CONNECTION

Your branch protection stays in charge

The agent never commits to your default branch. Every change it proposes is staged on a single dedicated working branch and raised as a pull request, so your existing review process is what decides whether it ships.

How changes reach you

One branch, one PR per cycle — not a swarm of commits.

- All edits collect on a dedicated Raechal SEO branch
- One pull request per working cycle, kept up to date
- Full diff to review, file by file, before anything merges

What it reads

Only the repositories you granted, and only what it needs.

- The file tree, narrowed to SEO-relevant files on request
- Individual files, or a specific region of a large file
- Framework detection so edits match your stack's

CAPABILITIES

What you can ask it to do

Talk to it like a teammate who knows your codebase. Reads run instantly; anything that writes is staged for your review.

Explore the repository

Find the files that actually matter for search, without you hunting for them.

- List accessible repositories and default branches
- Fetch the file tree, filtered to SEO-relevant files
- Read a whole file, or just the region you care about
- Search inside a file for a specific pattern

Audit page SEO

Check what your rendered pages expose to search engines and AI crawlers.

- Title, meta description, canonical and robots directives
- Structured data and schema coverage
- Sitemap presence and correctness
- Gaps flagged in plain language, ranked by impact

Propose code changes

Real edits to real files, staged as a pull request.

- Propose changes across one or many files
- Edit a precise snippet rather than rewriting a file
- Rewrite a page's content around target keywords
- Every proposal arrives as a reviewable diff

Machine-readable files

The files AI crawlers and search engines look for first.

- Add or update llms.txt for AI crawlers
- Add or update robots.txt directives
- Add or check a sitemap
- Generate JSON-LD schema markup

Publish content

Ship a Blog Studio article into your repo as a real post.

- Create a blog post in your framework's expected format
- Front matter written to match your existing posts
- Place generated images into the post
- Set the canonical URL on a published post

Post-publish SEO

The steps most teams forget once the article is live.

- Submit new URLs to IndexNow for faster discovery
- Confirm the canonical resolves as intended
- Check the sitemap picked the new page up
- Set your site domain so URLs are never guessed

AUTOBLOG POSTING

Blog Studio → pull request

Write an article in Blog Studio, send it to the GitHub Agent, and it arrives as a post in your repository — as a pull request, in your framework's format.

1 · Hand off from Blog Studio

Send a finished article to the GitHub Agent. The handoff is detected server-side, so a post can never be raised by accident.

2 · Written to your conventions

The agent detects your framework and writes the file where your project expects posts to live, with front matter matching your existing entries.

3 · Staged on the working branch

The new post joins the dedicated Raechal SEO branch alongside any other pending changes, rather than committing straight to main.

4 · Raised as a pull request

You get a PR with the full diff. Review it like any other contribution, then merge when you're ready — or close it and nothing happened.

BUILT TO BE SAFE ON A LIVE SITE

Nothing happens you didn't approve

It writes to a real codebase — so it's wired to propose rather than impose, and your existing review gates stay in force.

Nothing merges without you

Every change is a pull request against a dedicated branch. Your default branch, branch protection rules and required reviews all stay exactly as they are.

Every edit is reversible

Changes are snapshotted before they're written, so any edit can be rolled back in one click — and an unmerged PR can simply be closed.

You choose what it can see

The GitHub App only has access to the repositories you explicitly grant during install. Revoke access at any time from your GitHub settings.

Guardrails on risky files

The agent is careful with entry points and layout files, and asks before touching anything structural rather than quietly rewriting it.

CREDITS

What it costs

Pay with credits as you go. Access is lifetime once unlocked — no subscription on the GitHub Agent itself.

ACTION	COST
Unlock the GitHub Agent One time. Lifetime access. Your first two repositories are included.	**75** credits
Connect an additional repository Per extra repo, one time. Reconnecting an existing repo is free.	**60** credits
Chat message Per prompt — audits, reads, proposed fixes, questions.	**5** credits
Autopost a Blog Studio article Full publish flow, raised as a pull request.	**20** credits

GOOD TO KNOW

Where it draws the line

Honest boundaries, so nothing surprises you.

- The agent proposes; it never merges. Shipping is always your decision.
- Long-form writing lives in Blog Studio. The agent handles SEO, metadata and placement.
- Canonical URLs and sitemaps need your site domain set, so nothing is guessed.
- It only sees repositories you granted during GitHub App install.
- Entry-point and layout files are treated cautiously — expect it to ask before structural edits.
- Framework detection covers common static and JS frameworks; unusual setups may need a nudge.