

Claude Code + GHL

Talk to your CRM in plain English. The operating layer for the leads everything else generates.

Built by [@brycenwood.ai](#) · October 2026

System 6 gives you a morning lead-recovery cron. System 7 gives you the full bridge. Once Claude Code is connected to GHL at the API level, you can ask it to do anything you would normally have to click through menus for — not just morning lead drafts. Pull lists, audit broken workflows, build new ones, draft 50 personalized follow-ups in one prompt, build dashboards on demand.

Here is the thing about GHL. Most local service businesses pay for it and use about 10% of what it does. The UI is brutal. Building a workflow takes 30 clicks. Pulling a list of leads who haven't been followed up with takes you through three different screens. Most owners never crack the magic.

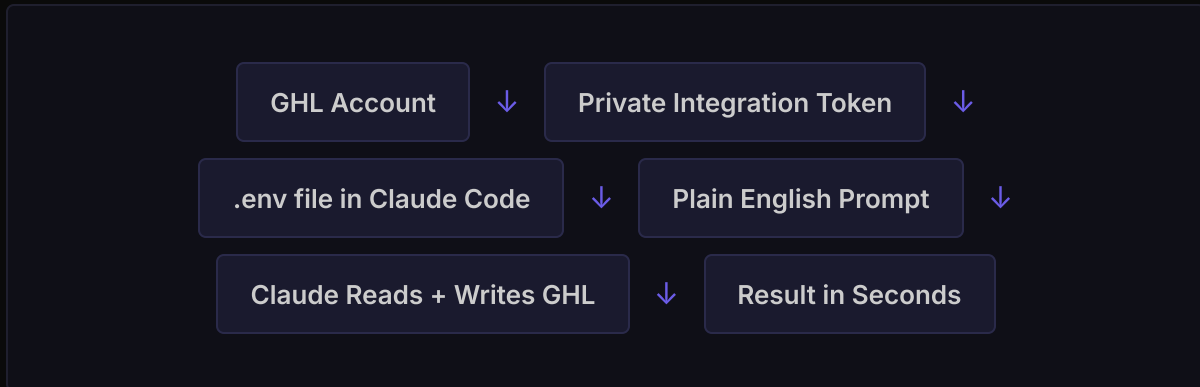
This system is the operating layer for the other six in this playbook. Your llms.txt brings AI traffic. Your sitemap brings Google traffic. Your website turns visits into leads. GHL captures them. System 6 catches the overnight stragglers. And System 7 is how you actually run the whole thing without a team.

OVERVIEW

What You're Building

A direct connection between Claude Code (the AI you talk to like a coworker) and GHL (the CRM running your business). Once it is set up, you can ask Claude Code to

read from your GHL, write to your GHL, draft messages, find broken automations, build dashboards, walk you through workflow setup — anything you would normally have to click through the UI for.



HOW THIS STACKS WITH SYSTEM 6

System 6 uses a small config file (`_config/ghl.json`) to drive a single morning cron. System 7 sets up a full `.env` + helper file pattern that any prompt or script in your project can reuse. Build System 6 first if you only want overnight lead recovery. Build System 7 when you want Claude Code to operate every part of GHL in plain English.

BEFORE YOU START

What You Need

- **Claude Code** installed (the CLI tool, not `claude.ai`)
- **A GHL sub-account**. Any tier works. You do not need agency-level access.
- **5 minutes** to grab a Private Integration Token from GHL settings
- **Python 3** on your machine. Claude Code installs any packages it needs.

TIME + COST

End to end this took me about 10 minutes the first time. Cost: \$0. Once it is set up, every prompt you run after that is just talking to Claude Code like any other conversation.

STEP 1

Generate a Private Integration Token in GHL

GHL needs a way to know that Claude Code is allowed to talk to it. That is what a Private Integration Token (PIT) does. Think of it as a key that only your account holds.

1. Log into your GHL sub-account
2. Click your profile (bottom-left) then **Settings**
3. Scroll the left sidebar and click **Private Integrations**
4. Click **Create New Integration**
5. Name it: Claude Code
6. Select scopes: check every contact, opportunity, workflow, conversation, and custom field scope. You can always tighten later.
7. Click **Create**
8. Copy the token that appears. You will only see it once.

TREAT THIS TOKEN LIKE YOUR PASSWORD

Do not share it. Do not commit it to GitHub. We will store it in a .env file (and Claude Code will set up your project to ignore that file automatically).

If GHL has moved a menu since this was written, paste this into Claude Code and it will walk you through the current path:

PASTE THIS INTO CLAUDE CODE

```
Walk me through getting a Private Integration Token from my
GoHighLevel sub-account, with current 2026 UI screenshots if needed. I
need a token with full access to contacts, opportunities, workflows,
conversations, and custom fields. Tell me exactly which menus to
click. If you are unsure of the current path, tell me what to search
for inside GHL settings.
```

STEP 2

Connect Claude Code to Your GHL

Now we tell Claude Code where to find your GHL. Open Claude Code in any project folder, or make a new one called ghl-ops just for this. Paste the prompt below. Claude Code will set up your environment, build a small Python helper, and run a test to confirm the connection works.

PASTE THIS INTO CLAUDE CODE

Set up a connection between this project and my GoHighLevel sub-account.

1. Create a .env file with these two variables:
GHL_API_KEY=[I will paste my Private Integration Token here]
GHL_LOCATION_ID=[my sub-account location ID]
2. Add .env to .gitignore so the token never gets pushed to GitHub.
3. Build a small Python helper file called ghl_client.py with:
 - A function that loads the .env values
 - A function that lists contacts (with filters for tag, last activity, opportunity status, and date range)
 - A function that reads a contact's full conversation history
 - A function that sends a contact an SMS (but only when I explicitly ask)
 - A function that adds or removes a tag on a contact
 - A function that lists all workflows and pipelines in the sub-account
4. Use the GHL API v2 with the Authorization: Bearer header and Version: 2021-07-28 header.
5. After you build it, run a test that pulls 5 contacts from my GHL and prints their names. Confirm the connection works. If it fails, debug and fix it before stopping.

WHERE TO FIND YOUR LOCATION ID

In GHL: profile menu, then Settings, then Business Profile. The Location ID is in the URL of that page (the long string after /location/). Or just ask Claude Code: "Help me find my GHL Location ID."

GHL API GOTCHA TO KNOW ABOUT

The contacts endpoint sortBy parameter has to be snake_case (date_added) not camelCase (dateAdded). It is not in the GHL changelog. If your test fails with a 422, that is probably why. Tell Claude Code and it will fix it.

STEP 3

Pull Your Stale Leads in One Sentence

This is the moment most service biz owners realize what is actually possible. With the connection from Step 2 working, paste this prompt:

PASTE THIS INTO CLAUDE CODE

Pull every contact in my GHL that:

- Was created in the last 90 days
- Has no opportunity attached, OR has an opportunity stuck in the first stage
- Has not been messaged by me or a teammate in the last 30 days

Sort by created date (oldest first). Output as a clean table with: name, phone, original first message, days since last activity, and any tags on the contact.

Do not message anyone yet. Just show me the list.

WHAT JUST HAPPENED

This list normally takes 20+ clicks across three GHL screens to assemble — and most owners never figure out how to filter for it. Now it takes one sentence. Welcome to the new layer.

STEP 4

Draft a Personalized Follow-Up for Every Stale Lead

System 6's morning cron drafts emails for overnight leads. Here is how you do the same thing on demand for any list of contacts — pulled from any filter, with SMS or email, all in one prompt.

PASTE THIS INTO CLAUDE CODE

For each contact in the stale leads list from Step 3, draft a follow-up SMS that:

- Sounds casual and like a real person, not a sales bot
- References what they originally asked about (use their first message in GHL conversations)
- Does not pressure them
- Ends with a soft question to restart the conversation
- Is under 160 characters

Show me the drafts in a table next to each contact. I will review before anything sends. After I approve, send the approved batch with a 30-second delay between each so it does not look automated.

ALWAYS REVIEW BEFORE SENDING

Do not auto-send AI-drafted messages without a human eyeball, especially the first few times. The 30 seconds you spend reviewing is worth it. After a few rounds, you will trust the patterns and you can let Claude Code send the approved batch on its own.

STEP 5

Audit Your GHL Top to Bottom

This is the prompt I run on every account I touch. It is how I found 5 critical bugs in my own sales pipeline that had been silently dropping leads for weeks.

PASTE THIS INTO CLAUDE CODE

Audit my GHL sub-account end to end. Look for:

1. Workflows that are "Published" but have not fired in 30+ days (probably broken or no triggers matching)
2. Contacts created in the last 90 days with no opportunity attached and no follow-up activity
3. Opportunities sitting in any stage longer than 21 days with no notes or activity
4. Custom fields that exist but have no data populated for more than 50% of contacts (probably orphaned)
5. Pipelines with stages that have no contacts, ever
6. Tags applied to fewer than 3 contacts (probably one-off mistakes)

7. Contacts with no first name or no phone (data hygiene issues)

Output a prioritized list of issues, sorted by likely impact (lost revenue first). Do not fix anything yet. Just show me what is broken.

REAL RESULT

I ran this exact audit on my Summit Wraps GHL on May 4, 2026. It found 5 critical bugs in our sales pipeline that had been silently dropping leads for weeks. Two were dead automations (workflows that were "Published" but pointing to deleted custom fields). Three were stalled opportunities no one had touched. Total recovery: 47 leads moved back into active follow-up. None of it would have been caught by clicking around in the UI.

STEP 6

Build a Speed-to-Lead Workflow With Claude Code as Your Guide

Speed to lead — texting a new lead within 30 seconds of them filling out a form — is the single highest-impact automation a service business can run. Most never set it up because the GHL workflow builder is exhausting. Have Claude Code walk you through the exact clicks.

PASTE THIS INTO CLAUDE CODE

I want to build a speed-to-lead workflow inside the GHL workflow builder UI. Walk me through every click step by step.

Workflow:

- Trigger: form submission on any inbound web form
- Action 1: send the contact an SMS within 30 seconds: "Hey [first name], got your message. What's the best number to reach you at?"
- Action 2: wait 30 minutes
- Action 3: if they have not replied, send a second SMS: "Just making sure this came through. When's a good time to call?"
- Action 4: add the tag "speed-to-lead" so I can track it

Tell me exactly which menu to click first, what to name each step, what fields to fill, and which dropdowns to choose. Account for any

GHL UI quirks you know about. After I confirm it is built, ask me to test it by submitting my own form and tell me what to look for.

WHY THIS IS THE HIGHEST ROI STEP IN THIS WHOLE PLAYBOOK

Studies show that texting a lead within 5 minutes vs 30 minutes of inquiry increases the chance of converting them by up to 21x. Most service businesses miss this because the manual setup is hard. With Claude Code guiding you, you set it up once and never think about it again.

STARTER PROMPT LIBRARY

Three More Prompts to Run Today

Once the connection works, you will start asking Claude Code to do things you never realized were possible. Here are three more starter prompts to get the muscle memory going.

PASTE THIS INTO CLAUDE CODE

Find every customer who paid me money in the last 12 months but has not booked again. Draft each one a personalized re-engagement text that references the service they paid for last time. Send 5 a day so it does not look spammy.

PASTE THIS INTO CLAUDE CODE

Read every conversation in my GHL from the last 14 days. Tell me which inbound customer messages I never responded to. Sort by deal size if I have it on the contact, otherwise by most recent first.

PASTE THIS INTO CLAUDE CODE

Build me a daily revenue dashboard from my GHL. Pull total opportunities by stage, total opportunities won this week vs last week, average days in pipeline, and top 5 lead sources by conversion rate. Email it to me every morning at 7am.

WHERE THIS GOES NEXT

This is the bridge. Every system in this playbook gets more powerful the moment it is connected to GHL through Claude Code. Your llms.txt knows which services convert. Your YouTube cross-poster tags new viewers as leads. Your website's contact form fires a workflow before the page even reloads. The next chapter is System 8 — once you have one Claude Code talking to your CRM, you can have a whole team of them, each with a defined role, all working in parallel.

That is the whole system.

Zero ad spend. Zero coding experience. The other playbooks live at brycenwood.com/playbook, one download each.

Built by Brycen Wood · [@brycenwood.ai](https://twitter.com/brycenwood.ai) · brycenwood.com