

Build your AI harness.

One shared assistant. Many real estate jobs.

A practical Homies setup guide · Daniel Foch · September 2026

You are building an assistant for your business. Start by assembling its working setup. Then teach it the jobs you want help with.

A **model** provides the intelligence. A **harness** is the working system around it: the app, connected accounts, business context, saved skills, approval controls, channels and somewhere to run. A **workflow** is one job that uses that system. Seller updates are a workflow; they are not the whole harness.

Start with one sentence

Help me set up my real estate AI harness.

Send this in the assistant you choose. It can ask which app you use, learn about your business and guide you one step at a time. You do not need to complete a form, edit a prompt or collect every file before starting.

The seven pieces you are assembling

PIECE	WHAT YOU ARE GIVING YOUR ASSISTANT
1. App + model	A place to work and the intelligence to do it
2. Connected accounts	Access to selected files, email, calendar or CRM
3. Context + memory	Your business, preferences and useful background
4. Saved skills	Reusable instructions for the jobs you want done
5. Approval controls	A clear boundary between preparing and acting
6. Phone + channels	A way to assign work and review it wherever you are
7. A place to run	Your computer or a supported cloud service

Modern apps supply many of these parts. You are usually **configuring a system**, not coding an application. Pick one capable model, then spend your attention making the setup useful. You can reconsider the model after testing real work.

Your first milestone: an app with your business context, one working connection and one useful saved skill. Add more jobs as you learn.

Install it. Connect it.

1. Choose a task-capable app

Two recognizable starting points are **ChatGPT Work** and **Claude desktop**. Their available features depend on your plan, organization and rollout. Use official downloads and check for file access, connectors, reusable instructions, task execution and approval controls. A plain chat can draft an answer; that alone does not supply the entire harness. [1][2]

ChatGPT: install the official desktop app and sign in. Choose **ChatGPT**, then **Work**. Existing Codex users can stay in Codex. In Projects, create **My Real Estate Assistant**; add the context files from page 3 and start Work tasks inside that project. [1][3]

Claude: install the official desktop app and sign in. Select **Cowork** if shown; on accounts with the newer combined experience, describe the task in a conversation. Create a project for your real estate work and add its instructions and files. Claude's guided **/setup-claude** command can help configure your role and tools where available. [2]

You do not have to adopt both. Start with the one you already have, provided it supports what you need. If a capability is missing, your first version can use approved file uploads and manual review.

2. Connect one account, then prove it works

A connector lets the assistant work with another service you authorize. Open the app's **Apps, Plugins, Connectors or Customize** area; the label varies. Select an official connection for a service you use: Google Drive, Microsoft 365, email, calendar or your supported CRM.

Sign in through that service's authorization screen. Check the account and permissions before approving access. Start with reading the relevant records where possible. Then ask: "**Find one recent file in the folder I just connected and tell me its title.**" Compare the answer with the real file. [4]

START WITH	ADD WHEN THE JOB NEEDS IT
One approved folder	Additional relevant files
Reading selected records	Draft creation
One successful retrieval	Controlled writes or sends

If your CRM has no official connector, use an approved export while you test. A technical provider can assess its official API later. Do not paste passwords or secret keys into chat. Connect only information you are entitled to use; an API does not create new privacy or MLS rights. [5]

Give it a useful memory.

3. Put your business context in one place

In **My Real Estate Assistant**, add three short documents. A Word file, Google Doc or plain text file is enough if the app supports it. Put reusable background in project files; put standing behavior in project instructions. Keep client-specific material scoped to the appropriate job or client rather than making every client's information universal context. [3]

DOCUMENT	INCLUDE
About my business	Market, services, target clients, brokerage, tools and what you want help with
How I work	Tone, preferred formats, working hours, review rules and how you organize information
Examples I like	A few approved emails, reports or listing materials, with a sentence explaining what makes each good

You can ask the assistant to interview you and write these. It should reuse relevant context you already gave it and ask one simple question at a time. Review the documents and remove unnecessary sensitive details before saving.

A useful starting instruction

Act as my real estate assistant. Use the approved project files to understand my business and style. Ask one short question at a time when an essential detail is missing. Check current facts against the authorized source. Prepare work for review; ask before sending, publishing or making a commitment. When I correct you, suggest what should be saved as a preference or reusable instruction, and show me the proposed change.

This is a starting rule you can save without filling any blanks. Use the app's actual approval controls as well; instructions alone do not restrict a tool's technical access.

Remember preferences. Refresh facts.

Memory is information saved for later use. It is not the same as retraining the underlying model, and it is not automatically current. [6]

Remember: preferred tone, a report format, an approved procedure, how you like recommendations presented.

Refresh: listing status, price, current activity, appointment availability and the client's latest instructions.

Say **"Save this as a standing preference"**, then ask where it was saved. In a new task inside the project, ask the assistant to summarize your preferences. Correct what it missed. If the app cannot reliably retrieve the saved context, attach the document when starting the job.

Teach it a skill.

4. Turn the way you work into a reusable recipe

This is the biggest part of the work. A **skill** is a saved way to do a job: what starts it, what information it needs, the steps, what good looks like and when to ask you. Your app may call it a skill, playbook, template or instructions.

Choose one task you already understand. Show a good example, explain how you produced it, and have the assistant write the procedure. Correct it before saving. A short voice explanation or walkthrough can be enough to start.

Say this after working through the task

Turn the way we just did that into a reusable skill. Include when to use it, the information needed, the steps, the output format, what to do when information is missing, and what needs my approval. Use the example we just reviewed. Ask one question at a time if an important detail is missing. Show me the instructions before saving them. Once approved, save them in the supported skills area or as a named playbook in this project, and tell me how to run it again.

In ChatGPT Work, the skill-creator interface may be available as **@skill-creator**; in Codex, **\$skill-creator**. You can also ask naturally to save a reusable skill. Where a native skill feature is unavailable, save the approved playbook as a project file and tell the assistant to follow it. Availability and names vary. [7]

Small example: a weekly seller-update skill

Trigger: you request an update for a listing. **Inputs:** current listing facts, this week's activity, the previous period, feedback and your approved example. **Steps:** check sources and dates; compare like periods; explain supported findings; draft the update.

Output: client draft, source notes and three call talking points. **Boundary:** ask about missing facts; wait before sending.

Fictional practice input: four completed showings this week versus six last week; seven inquiries versus five; three feedback responses; no market-comparison data supplied.

Expected draft excerpt: "We had four completed showings, down from six, and seven inquiries, up from five. We received three feedback responses. Let's review the feedback and next week's activity before deciding whether to change our approach." It should not invent feedback, a market trend or a price recommendation.

Save this as **Weekly seller update**, then try a second case. Once it works, add another skill: follow-up preparation, a listing-launch package or showing coordination. They share the same harness.

Set the controls. Take it with you.

5. Make the decision boundary simple

YOUR ASSISTANT MAY...	YOUR ASSISTANT SHOULD ASK BEFORE...
Find, organize, calculate and draft	Sending, publishing or making commitments
Point out missing information	Guessing a required fact
Suggest a next step	Changing a listing or client record

Configure the platform's approval settings or restrict the available actions where supported. A connector that can create email drafts may also technically be able to send; check the actual permissions. If sending cannot be controlled, have the assistant leave its draft in chat and send it yourself.

Try the skill before adding ten more

You are checking that the assistant understood your teaching. This can be simple.

TRY	IT SHOULD...
One ordinary case	Produce the expected work and wait at the agreed boundary
One missing-information case	Name the missing fact instead of inventing it
A second run after correction	Use the improved instructions without repeating the same mistake

Check both the output and the action log. Fix the saved skill, then repeat the failed case. One impressive answer is a beginning; repeatable useful work is the goal.

6. Put the assistant on your phone

Install the matching official mobile app. Sign in with the same account, open the same supported project or cloud task, and send a small request. Review the result there. ChatGPT Work and Claude support cloud-task experiences across devices, with availability depending on the plan and current rollout. [1][2]

Your first channel can simply be the phone app. A channel is how you reach the assistant; it is not where the work necessarily runs. Local files and computer actions may still need the desktop app open.

CRM messaging, email-triggered work or a custom phone integration can come later if there is a specific need and a supported connection. You do not need to build a webhook just to get started.

Quick check: assign a harmless file-summary task from your phone. Confirm that it used the expected source and that you can find the result again on your desktop.

Give it a place to run.

7. Decide what should happen while you are away

For work you start and supervise at your desk, your existing computer may be enough. For work that should happen at a set time, check the app's supported scheduled-task feature and where that task executes.

OPTION	WHAT YOU NEED TO KNOW
Supported cloud task	Runs on the provider's servers. Confirm the job, connected tools and scheduling are supported.
Local task	Your computer needs to stay awake, online and running the app. You handle availability.
Cloud task using local resources	The main task may run remotely, but local files, browser or computer access can still require your desktop to remain open.

A Mac mini is one possible always-on computer, not a prerequisite. Start with the hosting your chosen app already supports. Do not buy hardware before you know what the job needs. [1][2]

For a schedule, specify a time zone, output location and reviewer. For example: prepare the Monday seller-update drafts at 9 a.m. America/Toronto; notify me when ready; wait before sending. Test the actual job with the laptop closed rather than assuming a web interface makes everything unattended.

Keep failures visible

Ask for a status when work finishes or fails. Keep a record of what was created and where. A temporary information-retrieval failure can be retried; an uncertain send must be checked before attempting it again. Check for an existing draft or completed action so a rerun does not create duplicates. If the app cannot reliably schedule, log or control the action, keep that step manual until it can.

Improve the harness as you use it

After a correction, say: **"Should that go in my memory or in this skill? Show me the change."** Save approved lessons in the appropriate place. Refresh changing facts each run. Add another workflow only after the first is useful.

Measure what matters: time saved after input preparation, review and fixes. For a fictional example, 35 minutes manually minus 5 preparing inputs, 8 reviewing and 2 correcting leaves **20 minutes saved per run**, before recovering setup time. Your results will vary.

Your first-week recipe

Install → connect one account → add your context → teach one skill → set approvals → try the phone → schedule only when useful.

You have built a harness when those parts work together. Then you can give it more jobs.

Keep the language simple.

TERM	WHAT IT MEANS
Model	The intelligence generating or reasoning over an answer
Harness	The shared working setup around the model
Workflow / skill	A workflow is a job; a skill is a reusable recipe for doing a job
Context / memory	Context is information for the work; memory is selected information saved for later
API / connector	An API lets software request data or actions; a connector packages an integration you can use
OAuth	A sign-in authorization process that grants specified access; review the permissions requested
MCP	A standard for connecting AI applications to compatible tools and context; it is not hosting
Webhook	A notification that something happened; it can trigger work but does not perform the work itself
Channel / runtime	The channel is how you reach the assistant. The runtime is where the task executes.

Official starting points and further reading

[1] **ChatGPT Work:** Get started with Work and Work and Codex across devices. Use the official setup guide for download and current interface instructions.

[2] **Claude:** Get started in three steps and current Cowork / task experience. Some accounts still show Cowork; others are receiving the combined experience.

[3] **Where to save context:** ChatGPT Projects and Claude projects.

[4] **Permissions:** Google's OAuth scope reference illustrates that access can range from reading to editing or sending.

[5] **Data use:** OPC generative AI principles and CREA's public DDF rules. Check your current brokerage, board and provider terms. Training controls and APIs do not establish permission by themselves.

[6] **Context and memory:** Anthropic's context-engineering guide.

[7] **Saving skills:** ChatGPT skills and plugins. For technical terms: official MCP overview and GitHub's webhook explanation.

Start the setup, not another research project.

Help me set up my real estate AI harness.

Homies · Educational setup guide · Official sources reviewed September 17, 2026. Interfaces and plan availability change; use the linked guides for current details.