

The Agent Reach Setup Guide

Install the free, open-source CLI that gives Claude Code or Codex real access to LinkedIn, YouTube, Reddit, X and any web page, with no scraping subscription.

Why this exists

Agent Reach is a free command line tool that gives an AI agent real internet access. It is MIT licensed, published on GitHub by Panniantong, and it had passed 85,000 stars when I wrote this in September 2026. One install, and Claude Code or Codex can read LinkedIn profiles and job listings, YouTube transcripts, Reddit threads, Instagram profiles, X timelines, GitHub repos, RSS feeds and ordinary web pages.

The reason I bothered is that most agent setups only pretend to have web access. Either the model answers from training data and quietly invents the details, or you plug in a scraping API that charges per request and still fails on the sites you actually wanted. Both of those cost you something. One costs money, the other costs trust in the output.

Agent Reach takes a different route. It does not run a scraping service at all. It installs the open-source reader that already works for each platform, keeps your credentials on your own machine, and gives the agent one consistent command to call. Nothing to subscribe to, nothing metered. The promise is simple: your agent reads the real page instead of guessing at it.

01

How Agent Reach actually works

1. It is a wrapper, not a scraper. Each platform is handled by a program that already exists and already works: yt-dlp for YouTube transcripts, the gh CLI for GitHub, feedparser for RSS, Jina Reader for plain web pages, twitter-cli for X, OpenCLI for Reddit and Instagram and Facebook, bili-cli for Bilibili, mcp-server-linkedin for LinkedIn, and Exa through mcporter for global search. Agent Reach installs them and puts one door in front of all of them.
2. It splits channels into zero-config and login-required. Most of the list works the moment you install it. The ones that need you to be signed in are X, Reddit and XiaoHongShu, and for those you hand over a session cookie or an existing browser profile. Your credentials get written to ~/.agent-reach/config.yaml with file permissions set to 600, so only your user can read them.
3. Your agent calls it like any other CLI. Because it is a command line tool, anything that can run a shell command can use it, which covers Claude Code and OpenAI Codex equally. You do not wire up an MCP server for the whole thing or paste an API key into a config. The agent runs agent-reach, reads what comes back, and keeps going.

The one thing almost everyone gets wrong

People run the install command, watch it finish, and assume every channel on the list is now live. It is not. The default install is a read-only dependency check, and the optional channels are opt-in by name. So you ask your agent for Instagram data, it comes back empty, and you decide the tool is broken when you simply never turned that channel on.

WATCH OUT

Run agent-reach doctor before you trust anything. It reports the state of every channel one by one, and it is the difference between a setup that works and a setup you think works. The second trap is accounts: use a throwaway account for any platform that needs a login. The project says this plainly in its own docs, and it is good advice. Pulling data from the account you actually care about is how that account ends up restricted.

The install commands and the agent handoff

SOURCE CODE

The full repo, MIT licensed, issues and docs included:

<https://github.com/Panniantong/Agent-Reach>

INSTALL.SH

```
# Recommended path: pipx
pipx install https://github.com/Panniantong/agent-reach/archive/main.zip

# 1. Read-only dependency check. Changes nothing.
agent-reach install --env=auto

# 2. Preview exactly what it wants to change on your system.
agent-reach install --env=auto --dry-run

# 3. Actually install and configure. Asks before touching anything.
agent-reach install --env=auto --system

# 4. Confirm every channel.
agent-reach doctor
```

VENV-FALLBACK.SH

```
# Use this if your Python came from Homebrew, or if pip refuses
# with an "externally-managed-environment" (PEP 668) error.

python3 -m venv ~/.agent-reach-venv
source ~/.agent-reach-venv/bin/activate
pip install https://github.com/Panniantong/agent-reach/archive/main.zip
```

CHANNELS.SH

```
# Optional channels are opt-in. Add them by name, comma separated.
agent-reach install --env=auto --system --channels=opencli,xiaohongshu

# X / Twitter: export the cookie with Cookie-Editor on x.com first
agent-reach configure twitter-cookies
export TWITTER_AUTH_TOKEN="..."
export TWITTER_CT0="..."
twitter search "your query" -n 10

# LinkedIn: register the MCP server through mcporter
mcporter config add linkedin --command uvx --arg mcp-server-linkedin@latest --scope
home

# XiaoHongShu on a server (desktop uses OpenCLI instead)
agent-reach configure xhs-cookies

# Reddit on a server (desktop uses OpenCLI instead)
pipx install git+https://github.com/public-clis/rdt-cli.git@<pinned-version>

# Podcast transcription (Xiaoyuzhou) needs a free Groq key
agent-reach configure groq-key
bash ~/.agent-reach/tools/xiaoyuzhou/transcribe.sh <URL>

# Stock data from Xueqiu, pulled from your Chrome session
agent-reach configure --from-browser chrome --platform xueqiu

# Keep it current
agent-reach check-update
agent-reach watch
```

THE HANDOFF LINE: LET THE AGENT INSTALL IT

The repo is built to be installed by your agent rather than by hand. Its README gives the line in Chinese:

帮我安装 Agent Reach: <https://raw.githubusercontent.com/Panniantong/agent-reach/main/docs/install.md>

The part that matters is the doc URL, so an English sentence pointing at the same file does the same job:

Install Agent Reach on this machine by following
<https://raw.githubusercontent.com/Panniantong/agent-reach/main/docs/install.md> exactly.
Run the read-only check first, show me the dry run, and ask me before changing anything
on my system. I want these channels: [YOUR CHANNELS].

LINKS FROM THE VIDEO

- Agent Reach on GitHub
<https://github.com/Panniantong/Agent-Reach>
- The install doc you hand to your agent
<https://raw.githubusercontent.com/Panniantong/agent-reach/main/docs/install.md>
- The update doc
<https://raw.githubusercontent.com/Panniantong/agent-reach/main/docs/update.md>

Set it up in Claude Code or Codex

1. Open a terminal in any folder. Agent Reach is a global CLI, so it does not matter which project you are in. Nothing gets cloned into your working folder, and that is deliberate.
2. Install pipx if you do not have it, then run: `pipx install https://github.com/Panniantong/agent-reach/archive/main.zip`. If the terminal then tells you agent-reach is not a command, run `pipx ensurepath` and open a fresh terminal window so the new PATH takes effect.
3. If pip refuses with an externally-managed-environment error, which is normal on a Homebrew Python, use the venv fallback above instead. Create `~/.agent-reach-venv`, activate it, and install the same archive URL inside it.
4. Run `agent-reach install --env=auto`. This changes nothing. It reads your machine and reports what is missing, which is the report you want before you agree to anything.
5. Run `agent-reach install --env=auto --dry-run` and read the list of system changes. Thirty seconds here saves you from wondering later what it touched.
6. Run `agent-reach install --env=auto --system` to do the real install. It will ask for approval before changing your system. Do not let it run anything under `sudo` that you did not understand and agree to.
7. Add the optional channels you actually want, by name: `agent-reach install --env=auto --system --channels=opencli,xiaohongshu`. Start small. Two channels you use beat eleven you configured once.
8. Set up the login-based channels one at a time. For X, export the cookie with the Cookie-Editor browser extension on x.com, run `agent-reach configure twitter-cookies`, and set `TWITTER_AUTH_TOKEN` and `TWITTER_CT0`. Ask Claude Code to add those two exports to your shell profile, then open a new terminal so they load.
9. For LinkedIn, register the MCP server: `mcporter config add linkedin --command uvx --arg mcp-server-linked@latest --scope home`. That is what powers profile reads and job search.
10. Run `agent-reach doctor`. This is the real test. A good first run lists your channels with a working status next to the zero-config ones, and tells you exactly which login-based channels are still waiting on credentials.
11. Give your agent its first job: ask Claude Code or Codex to run `agent-reach doctor`, read the output, and pull the transcript of one YouTube video you already know. If the transcript comes back with real quotes from that video, the whole chain works.

QUICK TIPS

- Everything Agent Reach owns lives under ~/.agent-reach/, with cloned tools in ~/.agent-reach/tools/ and temporary files in /tmp/. If you ever see your agent cloning a repo into your project folder, stop it. That is not how this is meant to run.
- On a headless server, skip Instagram and Facebook. Both go through OpenCLI, which needs a desktop Chrome session, so they will never work there.
- GitHub works through the gh CLI, so if you are already logged into gh, that channel is live with no extra setup at all.

04

The commands I actually run

AUDIT WHAT IS CONNECTED

Run `agent-reach doctor` and show me the raw output. Then tell me, in a short table, which channels are working right now, which need a login I have not given yet, and which of them are worth setting up for [WHAT I DO]. Do not install anything yet.

MINE A YOUTUBE VIDEO FOR WHAT IT ACTUALLY SAID

Use Agent Reach to pull the transcript of [YOUTUBE URL]. Give me the claims the creator makes, the specific numbers and tool names they mention, and anything they state as fact that I should check before repeating. Quote the transcript directly for each point.

FIND COMPANIES WITH A BUYING SIGNAL

Use the LinkedIn channel to search job listings for [ROLE OR KEYWORD] in [LOCATION]. For each result, give me the company, the role, and one line on why a company hiring for that role would need [WHAT I SELL]. Skip anything posted more than [N] days ago.

READ THE ROOM BEFORE WRITING ANYTHING

Research [TOPIC] using Agent Reach. Pull the top Reddit threads and the transcripts of the two or three most relevant YouTube videos. I want the complaints people repeat, the words they use for the problem, and the advice that keeps getting contradicted. Cite the thread or the video for every point.

WATCH A COMPETITOR WITHOUT CHECKING MANUALLY

Set up a weekly check on [COMPETITOR] using `agent-reach watch`. I want their new posts, any YouTube uploads, and mentions of them in the places people discuss [MY NICHE]. Write me a short summary each time with what changed since the previous run, and say clearly when nothing changed.

When it breaks

ERROR	FIX
externally-managed-environment during pip install	PEP 668, usually a Homebrew Python. Build the venv instead: <code>python3 -m venv ~/.agent-reach-venv</code> , source it, then pip install the archive URL inside it.
command not found: agent-reach	pipx installed it somewhere not on your PATH. Run <code>pipx ensurepath</code> , then open a brand new terminal window. The old one keeps the old PATH.
A channel you expected is missing from agent-reach doctor	Optional channels are opt-in. Install it by name: <code>agent-reach install --env=auto --system --channels=opencli</code> . Re-run doctor to confirm.
X / Twitter returns nothing, or an auth error	The session cookie expired. Re-export it with Cookie-Editor on x.com, run <code>agent-reach configure twitter-cookies</code> again, and reset <code>TWITTER_AUTH_TOKEN</code> and <code>TWITTER_CT0</code> in a new terminal.
Instagram or Facebook fail on a server	Both run through OpenCLI, which needs a desktop Chrome session. Run those two from your own machine. They will not work headless.
LinkedIn calls fail or the tool is not found	The MCP server is not registered. Run <code>mcporter config add linkedin --command uvx --arg mcp-server-linkedin@latest --scope home</code> , then re-run doctor.
Your agent starts cloning repos into your project folder	Wrong location. Agent Reach keeps its tools in <code>~/.agent-reach/tools/</code> . Stop the agent, delete what it created in your project, and point it back at the install doc.
The installer asks for sudo and you are not sure why	Cancel it. Run <code>agent-reach install --env=auto --dry-run</code> , read the list of system changes, and only approve the ones you understand. It should not need to touch anything outside <code>~/.agent-reach/</code> .
A login-based channel worked yesterday and is silent today	Cookies expire quietly. Run <code>agent-reach doctor</code> first to see which one dropped, then re-run the matching <code>agent-reach configure</code> command for that platform.

Ready to go deeper?

This playbook is the short version. Inside the hub you get the full build walkthroughs, the prompt library, and the systems I use to ship this in a weekend.

Join the Build →

<https://hub.digicuratoragency.com/join>