The 3 Claude Connectors Setup Playbook

The exact commands to give Claude live research, web scraping, and browser control with Perplexity, Firecrawl, and Playwright.

Why these three

Claude out of the box only knows what it learned during training and whatever you paste into the chat. That's fine for writing and reasoning, but it breaks down the moment you need something current, something from a specific website, or something done on a live page instead of just described.

I run three connectors that close those three gaps: Perplexity for research, Firecrawl for reading the web, and Playwright for controlling a browser. Set up together, Claude stops being a smart text box and starts being a system that looks something up, reads the page behind it, and acts on it, all in one conversation, without you copying and pasting between five browser tabs.

The one-line promise: give Claude these three and it can research a topic, pull the real content behind that research, and then go do something with it, on its own, from a single prompt.

This guide is the setup, not the theory. Skill folder locations, the exact install commands, the API keys each one needs, the actual prompts I run day to day, and what to do when one won't connect.

01

How it actually works

1. Perplexity plugs Claude into a live AI research model instead of its frozen training data. Ask a question that needs a current answer and Claude sends it to Perplexity, which searches the live web and comes back with a sourced answer Claude can quote and cite, not a guess dressed up to sound confident.
2. Firecrawl gives Claude a way to actually read the internet instead of just talking about it. Point it at one URL and it scrapes that single page into clean markdown, stripping out the navigation and ad clutter. Point it at a domain and it crawls the whole site, following internal links while respecting robots.txt, and hands back structured content instead of a wall of raw HTML.
3. Playwright hands Claude a real, controllable browser. It drives Chromium, Firefox, and WebKit through one API, the same engine a lot of QA teams already use for automated testing. With it wired into Claude, the model can click buttons, fill out multi-field forms, scroll, and pull data off pages that have no API and never will.

---

02 The one thing almost everyone gets wrong

The mistake is installing all three in one sitting and never testing them one at a time.

> **WATCH OUT**
>
> If you install Perplexity, Firecrawl, and Playwright back to back and something breaks, you won't know which one is the problem, because all three touched your shell config in the same few minutes. Add one, run a real command through it, confirm it actually responds with something useful, then add the next. Playwright first, since it needs no API key and is the fastest way to prove your Claude Code setup can talk to an MCP server at all.

The second most common mistake: pasting an API key straight into the chat so Claude can 'just use it.' Keys belong in your shell profile as environment variables, not in a message history that gets logged and synced. Every install step below routes the key through your shell instead.

---

03 The connectors

> **SOURCE CODE**
>
> Playwright's MCP server, the official docs and repo:
>
> **https://playwright.dev/**

INSTALL-PLAYWRIGHT.SH

```
claude mcp add playwright npx @playwright/mcp@latest
npx playwright install
```

INSTALL-FIRECRAWL.SH

```
export FIRECRAWL_API_KEY="your-key-from-firecrawl.dev"
claude mcp add firecrawl npx firecrawl-mcp
```

INSTALL-PERPLEXITY.SH

```
export PERPLEXITY_API_KEY="your-key-from-perplexity.ai"
claude mcp add perplexity npx server-perplexity-ask
```

VERIFY.SH

```
claude mcp list
# expect: playwright, firecrawl, perplexity all listed as connected
```

- Firecrawl's free tier is 1,000 monthly credits with no card required. A single page scrape costs 1 credit, a search costs 2 credits per 10 results, so the free tier covers a lot of real research before you'd ever hit a paywall.
- Playwright's connector is free and open source. The only cost is the one-time `npx playwright install` step that downloads the actual browser binaries.
- Perplexity needs a paid or free-tier API key depending on your usage; check perplexity.ai's current API pricing before wiring it into a high-volume workflow.

> **SITES SHOWN IN THE VIDEO**
>
> - Perplexity
> **https://www.perplexity.ai**
> - Firecrawl
> **https://www.firecrawl.dev/**
> - Playwright
> **https://playwright.dev/**

Set it up in Claude Code or Codex

1. Open a terminal in Claude Code (or Codex) and confirm you're on a current version, since MCP support and the `claude mcp` command need a recent build.
2. Run the Playwright install command first: `claude mcp add playwright npx @playwright/mcp@latest`, then `npx playwright install` to pull the actual browser binaries. It needs no API key, so this is your first real test that MCP servers work at all in your setup.
3. Sign up at firecrawl.dev, grab your API key from the dashboard under API Keys, and ask Claude to add `FIRECRAWL_API_KEY` to your shell profile (`~/.zshrc` or `~/.bashrc`), then open a brand new terminal window so the variable actually loads.
4. Run `claude mcp add firecrawl npx firecrawl-mcp` and test it with a real scrape before moving on.
5. Sign up for a Perplexity API key, add `PERPLEXITY_API_KEY` to your shell profile the same way, open a new terminal, then run `claude mcp add perplexity npx server-perplexity-ask`.
6. Run `claude mcp list` and confirm all three show as connected before you trust any of them in a real workflow.
7. A successful first run looks like this: ask Claude a dated question and watch it call Perplexity and cite a real source, then ask it to scrape a URL and watch Firecrawl return clean readable text instead of a dump of raw HTML tags.

> **💡 QUICK TIPS**
>
> - If you're on Codex instead of Claude Code, the connector setup lives in your project folder or under `.codex/skills//`, with a pointer added to `AGENTS.md` so the agent knows to load it on start.
> - Never paste an API key directly into a chat message. Add it to your shell profile so it stays local and Claude reads it from the environment instead.

---

04 The commands I actually run

RESEARCH BEFORE ANYTHING ELSE

```
Use Perplexity to find the current pricing and top 3 competitors for [PRODUCT OR NICHE]. Cite your sources.
```

PULL A COMPETITOR'S PAGE

```
Use Firecrawl to scrape [COMPETITOR URL] and give me their headline, subheadline, and every CTA on the page.
```

CRAWL A WHOLE SITE

```
Use Firecrawl to crawl [DOMAIN] and list every page under /blog.
```

FILL OUT A FORM WITH PLAYWRIGHT

```
Open [URL] with Playwright, fill in the contact form with [NAME], [EMAIL], and [MESSAGE], and confirm the submission went through.
```

CHAIN ALL THREE IN ONE SYSTEM

```
Use Perplexity to research the top 3 tools for [TASK]. Use Firecrawl to scrape each tool's pricing page. Then use Playwright to sign up for a free trial of whichever one has the best free tier.
```

When it breaks

ERROR	FIX
`claude mcp list` shows Firecrawl or Perplexity as disconnected	Confirm the API key is actually in your shell profile, then open a brand new terminal window. A reused terminal still has the old environment loaded.
Firecrawl scrape returns empty or partial content	The site is likely JS-heavy and blocking the crawl. Try scraping a single page instead of crawling the whole domain first.
Playwright opens a browser but nothing happens	Run `npx playwright install` once to make sure the actual browser binaries are downloaded, not just the MCP package.
Perplexity answers without a citation	Ask again and explicitly request sources. It always has them since it's searching live, but it doesn't always surface them unless you ask.
`claude mcp add` command not found	Update Claude Code to a current version. Older builds don't have the `mcp` subcommand at all.
A connector worked yesterday and doesn't today	API keys and free-tier limits both expire or reset. Check your Firecrawl or Perplexity dashboard for a rate limit hit before assuming the install itself broke.
Firecrawl crawl runs but stops after a handful of pages	You likely hit your monthly credit limit. Check usage in the Firecrawl dashboard, since a crawl burns credits per page, not per request.

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