



// SECTION 01 • FIELD GUIDE

3 CHATGPT CONNECTORS

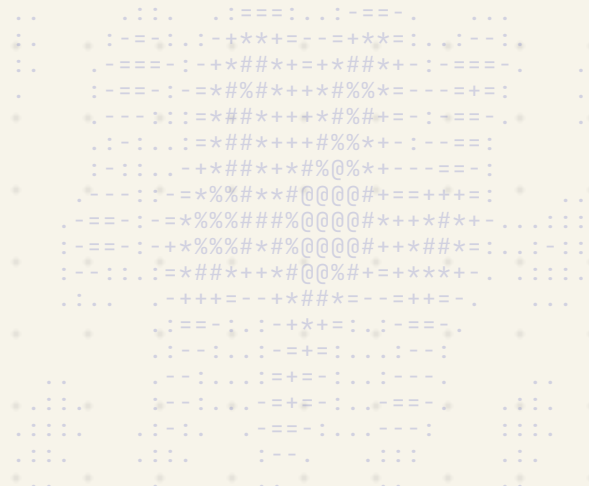


The three most powerful connectors you can plug into ChatGPT. PostHog, Perplexity, and Context.dev. What each one unlocks, the exact install in ChatGPT and in Codex, and the prompt to run first.

```
> boot opusjake_os
> resource: chatgpt-connectors
> version: v1.0 • 2026
> status: ready_
```

► START HERE: CONNECT POSTHOG

ChatGPT on its own answers from memory. Plug in these three and it reads your product data, researches the live internet with citations, and watches any website for you. Here is the install for each, in ChatGPT and in Codex, plus the first prompt to run.



THE STACK

#	CONNECTOR	WHAT IT GIVES CHATGPT
01	PostHog	Your product data: analytics, errors, replays, flags
02	Perplexity	Live, cited answers from the web
03	Context.dev	Radar for the web: monitors that report when a page changes

// TWO WAYS TO INSTALL

- ChatGPT app (web + desktop): sidebar **Plugins** → search → + for official plugins. For any other MCP server: **Plugins** → **Add** → **Create MCP App** and paste the server URL. If ChatGPT asks, turn on **Developer mode** in settings first.
- Codex: one `codex mcp add` line in the terminal, then `codex mcp login` to sign in.

Every install below gives you the fastest path first.

01 / POSTHOG

The one that changes how you run a product. PostHog holds your analytics, session replays, error tracking, feature flags, experiments, and surveys. Connect it and ChatGPT stops guessing about your users and starts reading what they actually did.

Get it here: posthog.com/mcp

// INSTALL

PostHog is an official ChatGPT plugin ("Analyze your product data"). In ChatGPT: **Plugins** → search **PostHog** → +, then sign in to PostHog. Done.

Codex, two lines:

```
codex mcp add posthog --url https://mcp.posthog.com/mcp
codex mcp login posthog
```

Or let the wizard wire it into every AI tool you use at once:

```
npx @posthog/wizard mcp add
```

Connecting is free. A few tools run AI on PostHog's side and bill as PostHog AI usage.

// WHAT IT CAN DO

The tools you will use first:

- `query-trends`, `query-funnel`, `query-retention` for any analytics question in plain English.
- `query-error-tracking-issues-list` and `query-error-tracking-issue` to find and read crashes.
- `query-session-recordings-list` to pull the replays behind a problem.
- `create-feature-flag` and `feature-flag-roll-out-to-everyone` to ship and roll back.
- `experiment-create` and `experiment-results-get` to run A/B tests from chat.
- `execute-sql` when you want the raw numbers.

// RUN THIS FIRST

```
@PostHog Look at my project for the last 30 days.
```

1. Where is the biggest drop-off in my signup funnel?
2. What are the top 5 errors this week, and how many users did each one hit?
3. Pull 3 session recordings of users who hit the worst error and tell me what they were trying to do.

```
Then give me the one fix that would move signups the most.
```

In Codex it goes one step further: it opens the file behind the stack trace, writes the fix, and rolls it out behind a feature flag.

02 / PERPLEXITY

The answer layer. It searches, reads, and hands ChatGPT a written answer with citations attached, so ChatGPT works from today's internet instead of its training data.

// INSTALL

Perplexity has no ChatGPT plugin, so it goes in as a custom MCP app:

1. **Plugins** → **Add** → **Create MCP App**.
2. Name: `Perplexity`.
3. Server URL: `https://api.perplexity.ai/mcp`
4. Authentication: **OAuth**. Tick **I understand and want to continue**, click **Create**, then sign in to Perplexity.

Codex, two lines:

```
codex mcp add perplexity --url https://api.perplexity.ai/mcp
codex mcp login perplexity
```

Heads up: Perplexity bills tool calls to your Perplexity API account at standard API pricing.

// PICK THE RIGHT TOOL

- `perplexity_search` returns ranked links. Cheapest.
- `perplexity_ask` is a fast answer with live search. The default.
- `perplexity_reason` for problems that need logic more than lookups.
- `perplexity_research` runs deep and returns a cited report. Slow, worth it.

Name the tool in your prompt, or ChatGPT reaches for `ask` every time.

// RUN THIS FIRST

Use perplexity_research on [TOPIC].

What changed in the last 90 days, who are the three most credible sources, and what is still unsettled? Cite every claim. If sources disagree, show me the disagreement.

03 / CONTEXT.DEV

Radar for the internet. Context.dev scrapes, crawls, and searches the web, and its monitors re-check any page on a schedule and tell you when something changes: a price drop, a new job listing, a competitor's update.

Plugin page: [Context.dev in ChatGPT](#)

// INSTALL

Context.dev is an official ChatGPT plugin. In ChatGPT: **Plugins** → search **Context.dev** → +, then approve the sign-in in your browser. Start a new chat after installing.

Codex, two lines:

```
codex mcp add context --url https://mcp.context.dev/mcp
codex mcp login context
```

The free plan includes 1,000 credits a month.

// HOW THE RADAR WORKS

- A **monitor** captures a baseline of a page, re-checks it on a schedule, and records a change when it finds one.
- **Page** monitors watch one URL. Exact mode reports every text change. Semantic mode reports only the changes you describe, like "tell me when the price drops".
- **Sitemap** monitors report URLs added or removed: new job posts, new blog posts, new product pages.
- Schedules run from every 10 minutes to once a year. The default is once a day.

// RUN THIS FIRST

@Context.dev Set up three monitors and check them every 6 hours:

1. [PRODUCT PAGE URL]: tell me when the price drops.
2. [COMPANY]/careers: tell me about any new job listings.
3. [COMPETITOR]/changelog: summarize anything new they ship.

Report changes only. Ignore dates and typo fixes.

STACK THEM

Each one alone is useful. Together they close the loop from your data to the market around it.

1. @PostHog: find the step in my signup funnel with the worst drop-off this month, and pull 3 replays of users who quit there.
2. Use perplexity_research: how do the top 3 products in my category handle that same step? Cite sources.
3. @Context.dev: put monitors on those 3 competitors' signup and pricing pages so I hear the moment they change.

// BEFORE YOU CONNECT ANYTHING

- **Start with PostHog.** Your own data beats anything on the internet. [Connect it here.](#)
- **Name the plugin or tool in the prompt.** It is faster and cheaper than letting ChatGPT guess.
- **Never put API keys in URLs.** OAuth or headers only.
- **Only add MCP servers you trust.** ChatGPT warns you for a reason: a server can feed instructions to the model through what it returns.

Follow opusjake.ai for the rest of the stack.