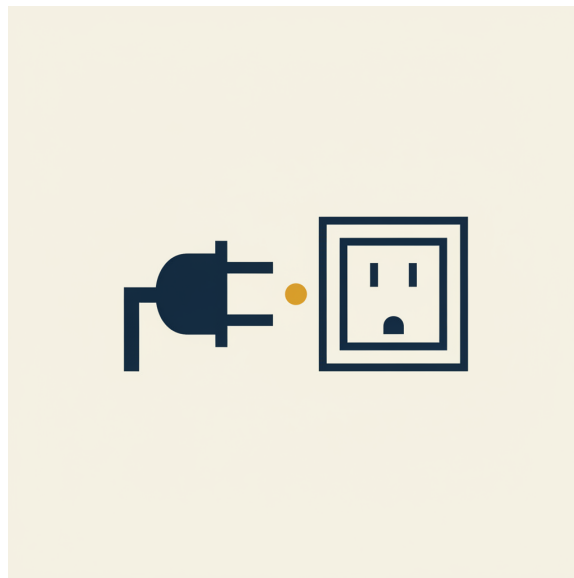




TIER 3 - EMPLOYABLE * V1.0 -- MAY 2026

API BASICS -- ROLLING YOUR OWN

The Anthropic, OpenAI, and Google APIs in plain language. The simplest possible API call you can actually run. When direct-API beats consumer tools, when it doesn't, and the honest math on whether the upgrade pays back.



BY

Alex Jahn / Agent Logic

v1.0 -- May 2026

The person on the team who's about to graduate from consumer-tier AI tools -- when ChatGPT and Claude's web apps stop being enough, what comes next, and whether the API path is worth it for your situation

15-20 minutes

Free. Forever.

EDITION

AUDIENCE

READ TIME

COST

Prepared by Agent Logic / alexanderjahn79@icloud.com / theaiguysi.com

CONTENTS

What's in here

- 1 The moment consumer tools aren't enough** **3**
You've been using ChatGPT or Claude through the web app for months. It works. Then one of these happens:
- 2 What "API" actually means here** **4**
An API (application programming interface) is a way for one program to talk to another. For AI specifically: a way for YOUR code to send...
- 3 The three big APIs** **4**
Snapshot mid-2026. Pricing changes; the framework holds.
- 4 The simplest possible call** **5**
You can run an API call without writing any code. The literal minimum is `curl`, a command-line tool that comes pre-installed on Mac.
- 5 When the API path pays back** **6**
The API is worth the lift when at least TWO of these are true:
- 6 Worked decisions** **7**
Your situation: you're a contractor who runs proposal drafting via Claude's web app. 5-10 proposals per week. Voice template + scope...
- 7 The honest limit** **8**
Three categories where "rolling your own" stops paying back:
- 8 Where to go from here** **8**
Eleven more modules in the Tier 3 expansion ahead:

SECTION 1

The moment consumer tools aren't enough

When the chat window becomes the bottleneck

You've been using ChatGPT or Claude through the web app for months. It works. Then one of these happens:

- You're running the same prompt 50 times a week and the copy-paste is the bottleneck
- You want AI INSIDE another tool you use (a Google Sheet, an internal app, a custom workflow)
- You need consistency across runs that the web tool's UI doesn't give you
- The cost or the rate limits or the privacy story stops working at the volume you're at

That's the moment. Up to here, consumer tools cover 90% of needs. Past here, you start looking at the API path -- the direct programmatic access to the same models, via code.

This module is the bridge. Plain-language API basics. When to use them, when not to, the simplest call you can actually run, and the honest math on whether it's worth the lift.

What you'll have by page 12

By the end of this primer:

- A
- The
how to get a key.
- The
- The
consumer tools win.
- The

plain-lang

three big

simplest

when-it-p

honest lin

The API path is one notch toward "developer." Most people who reach for it should turn around. The ones who keep going have a real reason -- usually volume, integration, or consistency. Know which side you're on before you start.

SECTION 2

What "API" actually means here

The plain-language version

An API (application programming interface) is a way for one program to talk to another. For AI specifically: a way for YOUR code to send a prompt to Anthropic's or OpenAI's or Google's servers, get a response back, and do something with it -- all without a human typing into a chat window.

The web app is a human-friendly wrapper around the API. ChatGPT (the web app) sends your typed message to OpenAI's API behind the scenes, gets a response, and shows it in a chat bubble. The API is the same underlying thing -- minus the bubble, minus the human-friendly UI, plus full programmatic control.

What you GET from the API path:

- **Run it from anywhere code runs** -- a Python script, a Google Sheet via Apps Script, a Make/Zapier workflow, a custom app
- **Pay per-use** instead of subscription tier limits
- **More control** -- temperature, max tokens, system prompts, structured output formats
- **Better privacy** -- most API plans don't train on your data by default

What you GIVE UP:

- The handholding of the consumer UI
- The convenience of voice mode, image upload, file analysis (some of these are also available via API but require more setup)
- The "I can just type and see what happens" exploration mode

The trade is: more control, less ease. If you don't need the control, don't take the trade.

SECTION 3

The three big APIs

Snapshot mid-2026. Pricing changes; the framework holds.

Anthropic API (Claude) -- the writer's API. Best for long-form drafting, reasoning, document

analysis. Claude 4 family is the strongest for any task involving nuanced text.

\$3-\$15 per million input tokens, \$15-\$75 per million output tokens depending on model tier.

a key: console.anthropic.com -> API Keys -> Create. Add a payment method; you start with a small credit.

OpenAI API (GPT, DALL-E, Whisper) -- the broadest API. Models for text, image, voice, code, embeddings -- all under one account. Best ecosystem if you need MORE than just chat (TTS, transcription, vision, function calling, image gen).

usually slightly cheaper for the small/fast tier.

Create.

Google AI API (Gemini) -- the integration-friendly API. Strong if your code already lives in Google Cloud, Google Workspace, or you're using BigQuery / Vertex AI.

the three at comparable tiers; the free tier is genuinely generous.

aistudio.google.com -> Get API Key.

cases, prototypes.

Pricing: r

Getting

Best fit: a

Pricing: o

Getting a

Best fit: n

Pricing: t

Getting a

Best fit: G

One-line picker:

- **Best text quality** -> Anthropic (Claude)
- **Multi-modal needs** -> OpenAI
- **Cost-sensitive or Google-ecosystem** -> Google (Gemini)
- **Just starting** -> Try all three free tiers; commit later

You can have keys for all three. Most working professionals who hit the API path end up with at least two.

SECTION 4

The simplest possible call

You can run an API call without writing any code. The literal minimum is `curl`, a command-line tool that comes pre-installed on Mac.

The 30-second version (curl, no code)

Open a terminal. Paste this (replace `YOUR_KEY` with your actual API key):

```
`` curl https://api.anthropic.com/v1/messages \ -H "x-api-key: YOUR_KEY" \ -H "anthropic-version: 2023-06-01" \ -H "content-type: application/json" \ -d '{ "model": "claude-sonnet-4-6", "max_tokens": 200, "messages": [{"role": "user", "content": "Hello in three words"}]}' ``
```

Hit return. You'll see Claude's response in your terminal. That's an API call. Cost: a fraction of a cent.

The 5-minute version (Python)

If you have Python installed:

```
`` pip install anthropic
```

```
from anthropic import Anthropic client = Anthropic(api_key="YOUR_KEY") response = client.messages.create( model="claude-sonnet-4-6", max_tokens=200, messages=[{"role": "user", "content": "Hello in three words"}] ) print(response.content[0].text) ``
```

Run it. Same thing happens, more cleanly. From here, you can plug this into a script, a cron job, a Make workflow that runs every hour, whatever.

What you have now

A working API call. From here, you can wrap it in real workflows. Most of the production AI work done in small businesses starts from exactly this -- a 10-line script that calls the API on a schedule or in response to a trigger.

Twenty lines of code separates "I use ChatGPT" from "I have AI running in my workflows." The lift isn't huge. The capability gain is.

SECTION 5

When the API path pays back

The API-pays-back test:

The API is worth the lift when at least TWO of these are true:

- **Volume:** you're running 50+ prompts per week of the same shape
- **Integration:** you need AI inside another tool (Sheet, app, workflow) where copy-paste is the friction
- **Consistency:** you need the same exact prompt to run reliably each time without UI variation
- **Privacy:** you want data-handling guarantees that exceed the consumer tier
- **Specialized models:** you need vision, embeddings, structured output, function calling that the chat UI doesn't expose well

One of those = consider; two or more = worth doing; four or more = you're past "consider," start building.

The API is NOT worth the lift when:

- You use AI casually, 5-10 conversations a week, all exploratory
- You need the hand-holding of the chat UI
- You're not comfortable with code or with debugging when something breaks
- The use case is one-off (a single big project; not a recurring workflow)

Most people who think "I should learn the API" are in the second list. Save yourself the lift.

SECTION 6

Worked decisions

Worked decision 1 -- Should I move to the API?

Your situation: you're a contractor who runs proposal drafting via Claude's web app. 5-10 proposals per week. Voice template + scope library + the workflow from Tier 2.

Test:

(kinda, but tolerable).

One borderline yes; not enough.

genuinely hits 50+/week or when you need to embed proposal-drafting into a custom tool.

Worked decision 2 -- Should I move to the API?

Your situation: you've built a Google Sheet that has 200 customer reviews. You want to run sentiment + theme analysis on each one. Current workflow: copy each review, paste into Claude, get the analysis back, paste it into the Sheet. Takes a full day.

Test:

Integration

Specializ

Verdict: s

Integration

(yes).

Three solid yeses.

Anthropic or Google Gemini API. The script runs through all 200 rows in ~20 minutes for a couple dollars. You ran a full day's work into half an hour of setup + 20 minutes of compute.

SECTION 7

The honest limit

Three categories where "rolling your own" stops paying back:

- **Anything that requires a real backend or a real UI.** Once you need user auth, a database, a hosted app -- you're not "rolling your own with the API," you're building a software product. That's a different skill set and probably the wrong scope.
- **Anything where the AI is the easy part.** Most production AI applications fail on the boring parts -- data quality, integration plumbing, error handling, observability. The API itself is 10% of the work. The other 90% is real software engineering.
- **Anything regulated.** Healthcare, finance, government, education -- "I built a quick AI tool" can run into compliance issues fast. Talk to your legal/compliance team before scaling beyond a personal-use script.

Within those limits, the API path is one of the highest-impact transitions an AI-fluent professional can make. Once you can wire an API call into your existing tools, every workflow with a recurring AI step becomes a candidate for automation.

SECTION 8

Where to go from here

Eleven more modules in the Tier 3 expansion ahead:

- **Evaluation and testing AI outputs** -- the eval framework, golden test sets, regression testing.

After that: RAG depth, AI in spreadsheets, internal tools, cost engineering, security/compliance, voice agents, multi-agent orchestration, content at scale, evaluating AI vendors, the AI-fluent leader capstone.

Get the next module the day it drops: theaiguywi.com/training

One email per release. No drip. No spam. Opt out anytime.

If you want this same API-readiness assessment done for your specific situation -- whether moving to the API path makes sense, what the first script should be, how to integrate it into your existing tools -- that's the consulting offer.

Reach out: alexanderjahn79@icloud.com

A short call. Honest scope. We figure out together if it's a fit.

Closing -- the lock-in line

The API isn't magic -- it's the same models you've been using, with the chat window removed. The capability gain is real for the right use cases. Most people don't have those use cases. The ones who do are the ones who should reach for the API path. Run the pays-back test honestly before you start.

20

Twenty lines of code.

That's the lift between "I use AI" and "I have AI in my workflows." For the right use case, it's the highest-leverage 20 lines you'll ever write.

-- Alex

Agent Logic --

Lac, WI. This is module 7 of 18 in Tier 3 (Employable).

© 2026 Agent Logic. Share freely.

theaiguyw