HOW SOFTWARE WORKS · BUILDER · 14 MIN READ

Put AI inside your app: AI APIs

How apps call models like GPT, Claude and Gemini through an API: keys, costs, system prompts, streaming and keeping it safe.

Pass the check (4 of 5) online to earn the “AI Integrator” badge:
https://digitalbuilders757.com/university/study/ai-apis-in-your-app

Chat app vs API

When you use ChatGPT or Claude in the app, a person types and reads. When your own app needs AI (summarise a form, draft a reply, tag a support ticket), it talks to the model through an API instead: your code sends a request with instructions and text, and gets text back.

OpenAI, Anthropic and Google each offer APIs for their models, usually through an SDK (a ready-made code library) for JavaScript or Python. Platforms like the Vercel AI SDK or OpenRouter let one app switch between providers.

The parts of a request

- Model: which model to use. Bigger models are smarter and slower and cost more; smaller ones are cheap and fast.
- System prompt: your app's standing instructions ('You tag support emails as billing, bug or other. Reply with one word.').
- Messages: the user's text and any context you add (the email, the product info).
- Settings: maximum length of the reply, and sometimes temperature (how varied the wording is).
- Tools or structured output: ask for JSON in a fixed shape so your code can use the answer reliably.

Costs and limits

APIs charge per token, counted separately for what you send (input) and what you get back (output). Long documents and long replies cost more. Providers also have rate limits: how many requests or tokens per minute your key may use.

Keep costs predictable: set a spending limit in the provider's dashboard, cap reply length, send only the text the task needs, use a smaller model where it works, and cache answers you reuse.

Keep the key on the server

An AI API key is a credit card for your account. It must only live on the server (an API route, a server action or an edge function), stored as an environment variable. The browser calls your server; your server calls the AI. Never put the key in front-end code, where anyone can copy it.

Also check who is allowed to call your AI route (signed-in users only? how many times per day?) so strangers can't run up your bill.

Make it good: test with real examples

Collect 10 to 20 real examples with the answers you'd want, and run them every time you change the prompt or model. That small test set is called an eval. It turns 'seems better?' into a score.

Stream long replies (show words as they arrive) so the app feels fast, and show the user that AI wrote it, with a way to edit or report a bad answer.

Key terms

API
An agreed way for one piece of software to ask another for something.

API key
A secret code that identifies your app to a service and is often tied to billing.

Token
A small chunk of text (a word, part of a word or punctuation) that models read and count usage in.

Streaming
Showing an AI reply word by word as it's generated instead of waiting for the end.

Eval
A set of real examples with expected answers, run to measure an AI feature's quality.

SDK
A ready-made code library that makes a service's API easy to use from your language.

System prompt
Hidden standing instructions that shape every answer an AI app gives.

Rate limit
The most requests or tokens a key may use in a period, such as per minute.

Structured output
Asking a model to answer in a fixed shape, like JSON, so code can read it reliably.

Environment variable
A setting or secret the app reads at run time, kept outside the code.

Try it: Design an AI feature on one page

1. Pick one job for AI in an app you'd build (for example: tag incoming messages).
2. Write the system prompt and the exact JSON shape you want back.
3. Write five example inputs with the answers you'd want: your first eval.
4. Ask a coding agent to build a server route that calls the model, keeps the key in an environment variable, and runs your five examples.

KNOWLEDGE CHECK

Put AI inside your app: AI APIs

5 questions. Circle one answer for each; 4 or more correct passes. Answers are at the back.

1. Where should the code that calls an AI API with your secret key run?

A. In the browser, so the AI answers faster without a server round trip
B. On the server, with the key in an environment variable
C. In a public GitHub README
D. Inside an image

2. How do AI APIs usually charge?

A. A flat monthly fee per app, however many people use it
B. Per user who signs in to your app
C. Per token, for input and output
D. Per page view

3. What is an SDK?

A. A kind of database
B. A web host
C. A password
D. A ready-made code library for using a service

4. Why ask the model for structured output like JSON?

A. So your code can reliably read the answer
B. It makes the answer look more professional to users
C. JSON answers use fewer tokens, so they always cost less
D. Models can only speak JSON when they run inside an app

5. What is an eval?

A. A pricing plan
B. A set of real examples with expected answers you run to measure quality
C. A type of error
D. An AI company

Official sites

OpenAI API docs: https://developers.openai.com/api/docs
Claude API docs: https://platform.claude.com/docs
Gemini API docs: https://ai.google.dev/gemini-api/docs
Vercel AI SDK: https://ai-sdk.dev

Last reviewed 2026-11-25. AI tools change fast; check each tool's own site for current features, plans and prices.

ANSWER KEY

Put AI inside your app: AI APIs

1. B: On the server, with the key in an environment variable
The browser calls your server; only the server holds the key.

2. C: Per token, for input and output
You pay for tokens sent and received, so long inputs and replies cost more.

3. D: A ready-made code library for using a service
SDKs wrap an API in easy functions for your programming language.

4. A: So your code can reliably read the answer
A fixed shape lets your app use the result without guessing.

5. B: A set of real examples with expected answers you run to measure quality
Evals turn 'feels better' into a score you can track as you change prompts or models.