OpenAI

A practical guide to building with GPT-5

Proven startup strategies to migrate, prompt, and scale with OpenAI's newest frontier model

Meet GPT-5: our most powerful, most steerable model yet.

Built for the full spectrum of coding and agentic tasks, GPT-5 is faster, smarter, and more adaptable than anything we've released before. Its greatest strength is how responsive it is to your direction, making it easier than ever to shape behavior for your specific use case.

But here's the catch: every new model 'thinks' a little differently. Prompts that worked with GPT-4.1 or other models won't always translate directly. To unlock GPT-5's full potential, you'll need to refine your prompts and tailor them to its unique behaviors and personality.

Our newest flagship model represents a major leap forward in what startups can accomplish, both due to its state-of-the-art performance (74.9% on SWE-bench Verified) and the controls developers have to steer and shape behavior. GPT-5 excels at agentic and multi-step reasoning tasks where reliability, depth, and control matter: parsing complex inputs, orchestrating tool use, or managing multi-stage workflows. Beyond agentic use cases, whether you're refining natural language interfaces, powering developer tools, generating structured outputs, or automating complex business processes, GPT-5 delivers higher accuracy, better consistency, and more predictable behavior than any previous model.

What we'll cover in this guide

In this guide, we'll share proven techniques to get the *most* out of GPT-5 based on our work with leading startups with technical resources and actionable steps to get started.

01	**Migrate**	Steps to migrate to the Responses API, designed for long-term scale, speed, and new reasoning capabilities
02	**Optimize**	Techniques to develop strong prompting that help you move faster and reduce engineering overhead
03	**Steer**	New controls let you guide how the model reasons and communicates to match effort and output based on task complexity
04	**Troubleshoot**	Resources to avoid common pitfalls like overthinking or overly verbose answers

By the end of this guide, you should understand how to leverage GPT-5 to its full potential to unlock more consistent, predictable, and accurate behavior while optimizing costs.

Migrate to the Responses API

Your first step to unlocking GPT-5's full intelligence is to build on the infrastructure designed for it. Only the Responses API allows the model to persist its chains of thought (reasoning items) across turns and tool calls, either with OpenAI managing state or by passing back encrypted reasoning items.

This means every request to the model has access to its complete internal context, significantly boosting performance and improving caching to lower costs—capabilities the Chat Completions API simply doesn't support.

Velocity

Smarter tool use and built-in state management reduce glue code and orchestration. You ship faster with fewer engineers and focus more time on your product and customers.

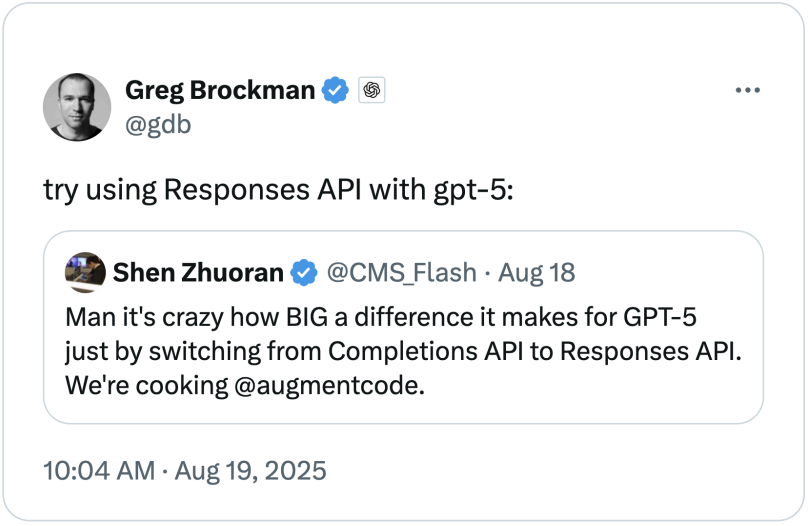
Scale without drag

Full-context reasoning plus faster performance and higher cache-hit rates lower infrastructure costs and latency as you grow. With zero-data retention (ZDR) compatibility, you're not locked into today's deployment pattern—you're ready for the agentic workflows that will define tomorrow's applications.

Future-proofing

The Responses API is the path forward for new reasoning capabilities. Building here keeps you off legacy APIs when the most powerful features ship and aligns your codebase with where OpenAI is investing most heavily, giving you long-term stability as the ecosystem evolves.

The Responses API is the unified surface for working with GPT-5. To maximize performance and future-proof your startup, we highly recommend moving workflows to the Responses API today.

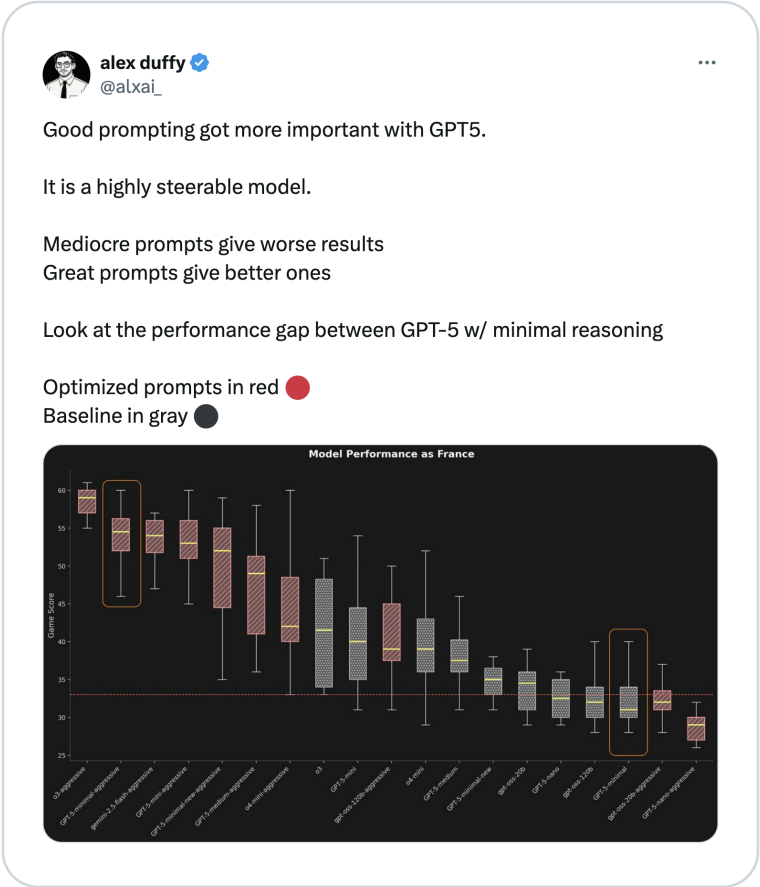


Getting started with Responses API

Responses API	Why we build the Responses API
Reasoning Models on the Responses API	Instructions on passing reasoning tokens, caching, and unlocking more intelligence
Migration Guide	Step-by-step instructions to move from Completions to Responses
Codex CLI Toolkit	Open-sourced tools to automate and streamline migration
Build Hour Demo	See GPT-5 in action on the Responses API

Optimize prompting

Moving to GPT-5 isn't just about adopting a new model – it's about mastering how to optimize it. Startups that develop strong prompting practices move faster, spend less on engineering overhead, and create products that feel meaningfully better to users.



Start with evals

Begin by running your existing prompts as is against your evals to establish a baseline and see where outputs diverge from expectations.

Inspect the model's reasoning

For specific failure cases, loop the eval again and stream reasoning summaries with GPT-5 in the Responses API. Watching the model reason helps you pinpoint where it needs more steering.

Iterate systematically

Change one variable at a time, continue hill-climbing, and document the impact. Adjust prompts to address concrete failure modes rather than making broad, simultaneous changes.

Metaprompt and simplify

GPT-5 is skilled at metaprompting—use the model to improve its own prompts as you iterate. Often, it requires less scaffolding than older models; shorter, clearer instructions can perform better.

Step 02: Optimize

Template and document

When prompts work reliably, lock them into reusable templates or a prompt library. Document what *good vs. bad* outputs look like so the team can build consistently, and revisit periodically as techniques evolve.

Getting started with prompt optimization

GPT-5 Prompting Guide	Best practices for crafting powerful prompts
Build Hour Demo	Live example of GPT-5 in action
Prompt Optimization Tool	Interactive playground to refine prompts

Steer GPT-5 with reasoning, verbosity, and new capabilities

GPT-5 introduces new controls that let you fine-tune how the model reasons and communicates. These capabilities help startups match model effort and output to the unique complexity of their products.

Reasoning effort

`reasoning_effort` controls how much the model thinks (and how readily it calls tools). The default is `medium`; options are `minimal`, `low`, `medium`, and `high`. Experiment to right-size effort to the complexity of your task and measure against your evals using the prompting guide.

Verbosity

`verbosity` influences the length of the model's output. Options are `low`, `medium`, and `high`. You can also add prompt instructions for scenarios where you want the model to override the default.

Experimentation guidance

GPT-5 is highly steerable. These parameters give you more control over model behavior. There's no single deterministic best configuration—systematically experiment and evaluate to identify what works best for your use case.

New & enhanced capabilities

GPT-5 Build Hour	Live coding session showcasing GPT-5 capabilities
Using GPT-5	Quick guide to get started and build effectively
New parameters and tools	Overview of the latest features and settings
Preambles	Tips for setting strong context in prompts
Latency optimization	Techniques to speed up responses
Cost optimization	Strategies to reduce usage costs

Troubleshoot using common patterns

From working closely with hundreds of startups, we see recurring issues such as overthinking, underthinking, over-deference, overly verbose outputs, latency problems (see Latency Optimization), tool overuse, and malformed tool calls. Because GPT-5 is highly steerable and eager to follow instructions, careful prompt tuning—paired with solid evals and metaprompting—resolves most of these quickly. For deeper guidance on diagnosing and correcting each pattern, explore the GPT-5 Troubleshooting Cookbook.

About the authors

This guide was developed by Hillary Bush, Startups Account Director, and Prashant Mital, Startup Solutions Architect, based on their experience working with top startups leveraging GPT-5.

They created this guide after helping dozens of early-stage and growth-stage startups adopt GPT-5 in production, seeing consistent patterns in how the most successful teams migrated APIs, tuned prompts, and used new reasoning controls to ship faster and build stronger products.

The goal of the OpenAI Startups Team is to share these best practices broadly so any startup, whether pre-seed or scaling globally, can accelerate its journey from idea to impact with GPT-5. We hope you found this guide useful – happy building!

Sources cited

Step 01 - Migrate	Greg Brockman @gdb, X
Step 02: Optimize	Alex Duffy @alxai_, X