OpenAI

Building with the GPT-5 model series

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

Meet the GPT-5 series: our most powerful, most steerable models yet.

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

Every new model "thinks" a little differently. Prompts that worked with GPT-4.1 or other models won't always translate perfectly. To unlock the full potential of GPT-5 models, you'll need to refine your prompts and tailor them to its unique behaviors and personality.

GPT-5 and GPT-5.1 represented major leaps forward in what startups can accomplish, both due to their state-of-the-art performance, controls developers have to steer and shape behavior, and ability to take on complex, multi-step reasoning tasks where reliability, depth, and control matter. GPT-5.2 builds on that momentum even further.

GPT-5.2 excels at agentic and multi-step reasoning tasks and can handle long context with ease. It's also our strongest vision model yet, excels at tool calling, and is very strong at coding. All of this combined makes it a powerful model across a range of industries and use cases: building unique frontend experiences, shipping fixes end-to-end, analyzing financial dashboards, interpreting technical diagrams, creating presentations, or automating complex business processes. GPT-5.2 delivers higher intelligence, faster results, better consistency, and more predictable behavior than any previous model.

We've introduced reasoning_level and verbosity parameters to the GPT-5 model series that give you more control over the model's behavior. With GPT-5.1 and GPT-5.2, you'll also see powerful adaptive reasoning capabilities that allow the model to right size its reasoning effort to the task at hand.

What we'll cover in this guide

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

01 Migrating to the Responses API

02 Designing and building agents

03 Prompt tuning

04 Steering the model

05 Eval loops and troubleshooting

By the end of this guide, you should understand how to leverage these models to their full potential to unlock more consistent, predictable, and accurate behavior while optimizing costs.

Foundations: Migrate to the Responses API

Your first step to unlocking full intelligence of any model in the GPT-5 series is to build on the infrastructure designed for it. The Responses API allows the model to persist its chain of thought (reasoning items) across turns and tool calls, either with OpenAI managing the state or by passing back encrypted reasoning items when building with Zero Data Retention (ZDR).

This means every request to the model has access to its complete internal context, significantly boosting performance and improving caching to lower cost. While we'll continue to support the Chat Completions API, we strongly recommend all agentic and reasoning workflows to be on the Responses API to capture these benefits.

Velocity

More natively supported tools, smarter tool use, and built-in state management reduce glue code and orchestration. You'll ship faster with fewer engineers and focus more time on your product and customers.

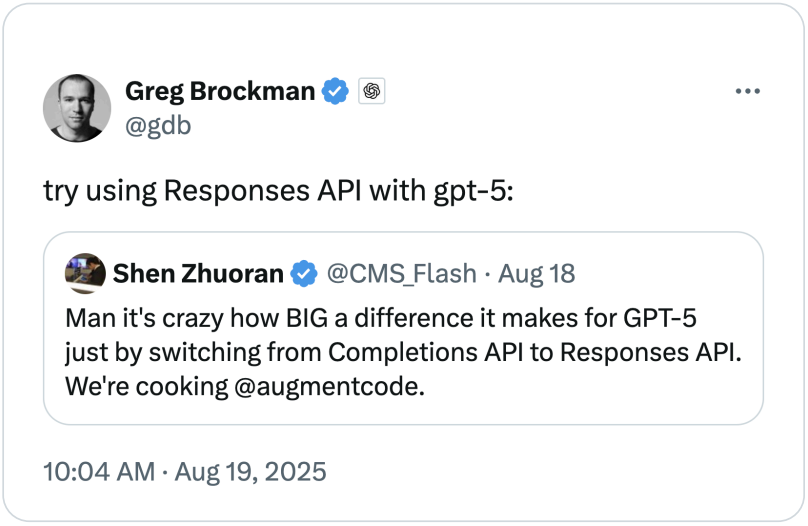
Scale without drag

Full context reasoning plus faster performance and higher cache hit rates lower costs and improve latency as you grow. With Zero Data Retention (ZDR) compatibility, you can unlock full intelligence for your agentic workflows while building for the enterprise.

Future-proofing

The Responses API is the path forward for working with the GPT-5 series and agentic workflows. Building here keeps you off legacy APIs when the most powerful features ship, and aligns your direction with where OpenAI is investing most heavily, giving you long-term stability as the platform grows.

To maximize performance and future-proof your startup, we highly recommend moving workflows to the Responses API today.



Step 01: Migrating to the Responses API

Getting started with Responses API

Responses API	Why we built 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
Responses Starter App	Starter app to build with the Responses API
Build Hour Demo	See GPT-5 in action on the Responses API

Design and build your agent

GPT-5.2 is our most powerful general purpose model designed for complex reasoning, multi step workflows, and intelligent tool use – making it an excellent foundation for building agents. As you build agents with GPT-5.2, it's helpful to ground your approach in a few foundational principles.

Build with first class primitives

OpenAI provides a full ecosystem to help you build, deploy, and monitor agents end-to-end. Models like GPT-5.2 serve as the agent's core engine, tools allow agents to interact with external systems, and OpenAI's platform gives you visibility into agent loops and evals – helping you iterate quickly as you scale.

Design your workflow

Map out the workflow your agent should handle. GPT-5.2 can power the full range of tasks – from interpreting ambiguous requests, deciding the next action, selecting the correct tool, handling exceptions, or determining when the workflow is complete.

Agent Builder lets you design workflows visually – ideal for rapid prototyping. For full control, use the Agents SDK and write your own code

Equip your agent

Agents become powerful when you give them the right resources. Well designed and documented tools extend the universe of what GPT-5.2 is capable of. Equip your agents with knowledge from databases, vector stores, file search and more to enrich the model's intelligence.

Optimize performance

Once your agent is working end-to-end, shift focus to reliability, evaluation, and optimization. Strong operational discipline dramatically improves user experience and reduces unexpected behavior.

Use Agent evals, a full eval platform that supports trace grading, datasets, and prompt optimization all in one place.

Step 02: Designing and building agents

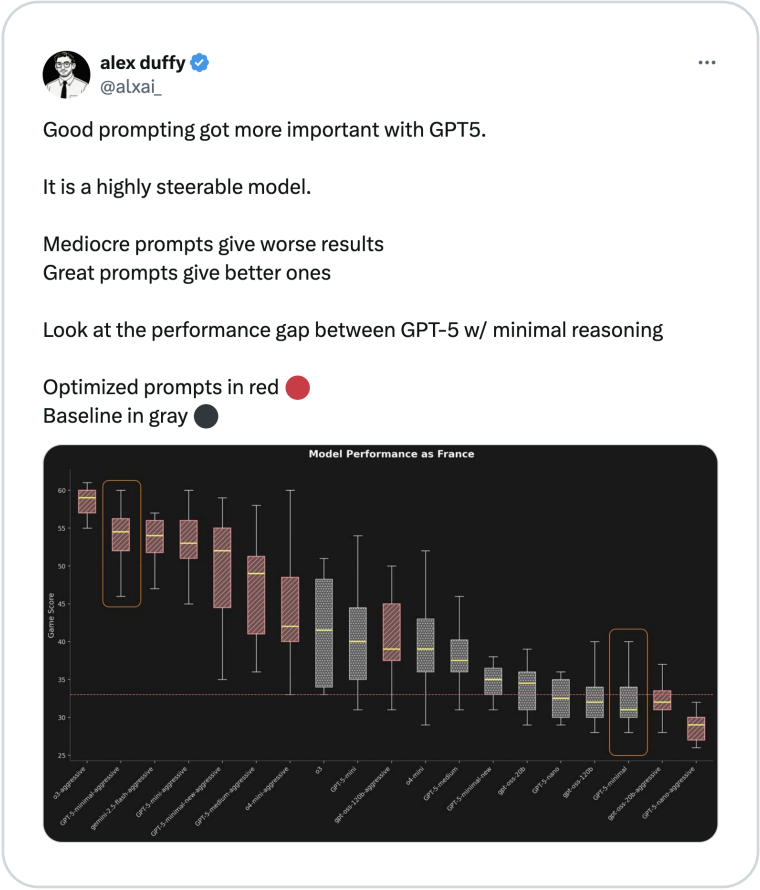
Getting started with agents and GPT-5.2

A Practical Guide to Building Agents	In depth guide to design and build agents
Building Agents	Foundational concepts to practical implementation
AgentKit Guide	Build, deploy, and optimize agentic workflows with AgentKit
Example agents	Support Agent Customer Service Agent Frontend Testing Agent

Prompt tuning

Moving to GPT-5.2 isn't just about adopting a new model – it's about mastering how to optimize and steer it. Builders that develop strong prompting practices move faster, allow their engineers to scale, and create products that feel meaningfully better to users.

GPT-5.2 follows many of the same prompting strategies as GPT-5, but may still need prompt adjustments to unlock full performance. Updating your prompt from a different model family or LLM provider may require more hands-on work. We recommend starting with our GPT-5.2 prompting guide and prompt optimization tool.



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 the model 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.

Step 03: Prompt tuning

Metaprompt and simplify

The GPT-5 model series is skilled at metaprompting—use the model to improve its own prompts as you iterate. Metaprompting proves useful in scenarios like prompt optimization, debugging, and scaling unique instructions. Rather than manually guessing which lines of a prompt cause certain behaviors, you can metaprompt the models to inspect their own instructions and logs.

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

Prompt Optimization Tool	Interactive playground to refine prompts
GPT-5.2 Prompting Guide	Best practices for crafting powerful prompts for GPT-5.2
GPT-5.1 Prompting Guide	Best practices for crafting powerful prompts for GPT-5.1
GPT-5 Prompting Guide	Best practices for crafting powerful prompts for GPT-5
GPT-5 Build Hour Demo	Live example of GPT-5 in action

Steer with reasoning, verbosity, and new capabilities

The GPT-5 model series introduced new controls that let you shape how the model reasons and communicates. These capabilities help builders match model effort and output to the unique complexity of their products.

Reasoning effort and adaptive reasoning

`reasoning_effort` controls how much the model thinks. Options for the model are:

- **GPT-5:** `minimal`, `low`, `medium`, and `high`. Defaults to `medium`.
- **GPT-5.1:** `none`, `low`, `medium`, and `high`. Defaults to `none`.
- **GPT-5.2:** `none`, `low`, `medium`, `high`, and `xhigh`. Defaults to `none`.

With GPT-5.1, we introduced stronger adaptive reasoning capabilities, which allows the model to dynamically right-size effort within the set range to the complexity of the task at hand. GPT-5.2 introduced improved adaptive reasoning capabilities, making the model more efficient than ever.

Both GPT-5.1 and GPT-5.2 have a new reasoning mode, "none", which produces no reasoning tokens in the response. Much of our guidance on prompting for GPT-4.1 applies for this effort level.

To find the right `reasoning_effort`, experiment and measure against your evals to find the right balance of reasoning and time for your product.

Verbosity

`verbosity` influences the length of the model's output. Options for all models in the series are `low`, `medium`, and `high`. You can also add prompt instructions for scenarios where you want to add more precise guidance on how verbose the desired output should be.

Apply_patch and shell tools

We've introduced two new tools for GPT-5.1 and GPT-5.2 in the Responses API. `apply_patch` allows you to easily create, update, and delete files in your codebase using structured diffs. `Shell` allows you to interact with your local computer through a controlled command-line interface.

Step 04: Steering the model

New & enhanced capabilities

GPT-5.2 Prompting Guide	Best practices for crafting powerful prompts for GPT-5.2
GPT-5.1 Prompting Guide	Best practices for crafting powerful prompts for GPT-5.1
Shell and Apply_patch Tools	API docs for new tools
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

Eval loops and troubleshooting

Because the GPT-5 model series is highly steerable and eager to follow instructions, careful prompt tuning – paired with solid evals and metaprompting – can resolve a lot of issues. Along with our troubleshooting guide, we recommend metaprompting, eval loops, and other latency optimization strategies.

Build an eval loop

As you iterate, having a strong eval loop is essential for catching regressions and ensuring the model consistently meets your quality bar. Treat prompt changes, parameter tweaks, and model upgrades as experiments—validate each one against your eval suite before shipping.

Latency optimization

Latency depends on more than just the model—prompt size, tool calls, context assembly, and workflow design all matter. Shorten prompts, delegate simple tasks to smaller models, and reduce unnecessary steps in agentic loops to improve speed. For user-facing experiences, use Priority Processing, leverage preambles, stream responses and minimize round-trips; for backend workloads, batch or parallelize work whenever possible.

Models in the GPT-5 series

Building the best models in the world is a primary focus for OpenAI. We're continuously improving the models based on customer feedback and research breakthroughs.

GPT-5.2	GPT-5
GPT-5.2-Chat	GPT-5-Mini
GPT-5.2-Pro	GPT-5-Nano
GPT-5.1	GPT-5-Pro
GPT-5.1-Chat	GPT-5-Chat

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 their 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