

# Structured Outputs

## Introduction

JSON is one of the most widely used formats in the world for applications to exchange data.

Structured Outputs is a feature that ensures the model will always generate responses that adhere to your supplied [JSON Schema](#), so you don't need to worry about the model omitting a required key, or hallucinating an invalid enum value.

Some benefits of Structed Outputs include:

- 1 **Reliable type-safety:** No need to validate or retry incorrectly formatted responses
- 2 **Explicit refusals:** Safety-based model refusals are now programmatically detectable
- 3 **Simpler prompting:** No need for strongly worded prompts to achieve consistent formatting

In addition to supporting JSON Schema in the REST API, the OpenAI SDKs for [Python](#) and [JavaScript](#) also make it easy to define object schemas using [Pydantic](#) and [Zod](#) respectively. Below, you can see how to extract information from unstructured text that conforms to a schema defined in code.

Getting a structured response

python ▾



```
1 from pydantic import BaseModel
2 from openai import OpenAI
3
4 client = OpenAI()
5
6 class CalendarEvent(BaseModel):
7     name: str
8     date: str
9     participants: list[str]
10
11 completion = client.beta.chat.completions.create(
12     model="gpt-4o-2024-08-06",
13     messages=[
14         {"role": "system", "content": "You are a helpful assistant."},
15         {"role": "user", "content": "What are the dates for the upcoming conference?"},
16     ],
17     response_format=CalendarEvent,
18 )
```

We use cookies and similar technologies to deliver, maintain, improve our services and for security purposes. Check our [Cookie Policy](#) for details. Click 'Accept all' to let OpenAI and partners use cookies for these purposes. Click 'Reject all' to say no to cookies, except those that are strictly necessary. Choose [Manage Cookies](#) to pick specific cookies you're okay with or to change your preferences.

Reject all

Accept all