

[Documentation](#)[API reference](#)[Forum](#)[Help](#)

**Streaming is now available in the Assistants API.**

[Learn more](#)[Dismiss](#)

# Developer quickstart

## Get up and running with the OpenAI API



Looking for ChatGPT? Head to [chat.openai.com](https://chat.openai.com).

The OpenAI API provides a simple interface for developers to create an intelligence layer in their applications, powered by OpenAI's state of the art models. The Chat Completions endpoint powers ChatGPT and provides a simple way to take text as input and use a model like GPT-4 to generate an output.

### Want to jump straight to the code?

Skip the quickstart and dive into the [API reference](#).

This quickstart is designed to help get your local development environment setup and send your first API request. If you are an experienced developer or want to just dive into using the OpenAI API, the [API reference](#) of [GPT guide](#) are a great place to start. Throughout this quickstart, you will learn:

How to setup your development environment

How to install the latest SDKs

Some of the basic concepts of the OpenAI API

How to send your first API request

If you run into any challenges or have questions getting started, please join our [developer forum](#).

## Account setup





First, create an [OpenAI account](#) or [sign in](#). Next, navigate to the [API key page](#) and "Create new secret key", optionally naming the key. Make sure to save this somewhere safe and do not share it with anyone.

## Quickstart language selection

Select the tool or language you want to get started using the OpenAI API with.

[curl](#)[Python](#)[Node.js](#)

Python is a popular programming language that is commonly used for data applications, web development, and many other programming tasks due to its ease of use. OpenAI provides a custom [Python library](#) which makes working with the OpenAI API in Python simple and efficient.

## Step 1: Setting up Python

### ✓ Install Python

---

### ✓ Set up a virtual environment (optional)

---

### ✓ Install the OpenAI Python library

---

## Step 2: Set up your API key

### ✓ Set up your API key for all projects (recommended)

---

### ✓ Set up your API key for a single project

---

## Step 3: Sending your first API request

### Making an API request





## Making an API request

After you have Python configured and set up an API key, the final step is to send a request to the OpenAI API using the Python library. To do this, create a file named `openai-test.py` using the terminal or an IDE.

Inside the file, copy and paste one of the examples below:

### Chat Completions

```
1 from openai import OpenAI
2 client = OpenAI()
3
4 completion = client.chat.completions.create(
5     model="gpt-3.5-turbo",
6     messages=[
7         {"role": "system", "content": "You are a poetic assistant, skilled"},
8         {"role": "user", "content": "Compose a poem that explains the conc"},
9     ]
10 )
11
12 print(completion.choices[0].message)
```

To run the code, enter `python openai-test.py` into the terminal / command line.

The [Chat Completions](#) example highlights just one area of strength for our models: creative ability. Explaining recursion (the programming topic) in a well formatted poem is something both the best developers and best poets would struggle with. In this case, `gpt-3.5-turbo` does it effortlessly.

## Next steps

Now that you have made your first OpenAI API request, it is time to explore what else is possible:

For more detailed information on our models and the API, see our [GPT guide](#).

Visit the [OpenAI Cookbook](#) for in-depth example API use-cases, as well as code snippets for common tasks.

Wondering what OpenAI's models are capable of? Check out our library of [example prompts](#).

Want to try the API without writing any code? Start experimenting in the [Playground](#).

Keep our [usage policies](#) in mind as you start building.





Was this page useful?

