



USING PYTHON AND OPENAI TO IMPLEMENT AI IN SOFTWARE PROJECTS

Eric Greene
eric@cloudcontraptions.com



Learn More About OpenAI



OpenAI: <https://openai.com/about/>

OpenAI API Docs: <https://platform.openai.com/docs/overview>

Goals for this Session

- Introduction to OpenAI
- Getting Started with OpenAI API
- Incorporate Large Language Models into Python Code



What is OpenAI?

- OpenAI is an artificial intelligence research and deployment company that develops advanced AI technologies, including large language models like ChatGPT, image generation models like DALL·E, and code generation tools like GitHub Copilot.
- OpenAI was founded in 2015 as a research lab focused on artificial intelligence and has since grown into a company that provides AI-based products and services.
- OpenAI's mission is to ensure that artificial general intelligence (AGI) benefits all of humanity, and it aims to develop safe and beneficial AI technologies.

OpenAI Purpose and Mission

- **Research & Development:** OpenAI conducts cutting-edge research in machine learning, deep learning, and AI safety.
- **Language Models:** OpenAI created GPT (Generative Pre-trained Transformer) models, including GPT-3.5 and GPT-4, which power applications like ChatGPT.
- **AI Safety & Ethics:** OpenAI prioritizes AI safety and responsible deployment to ensure AI benefits society.
- **Partnerships & Products:** OpenAI collaborates with companies like Microsoft, integrating AI into tools like Azure OpenAI Service and GitHub Copilot.
- **Mission:** OpenAI's goal is to ensure that artificial general intelligence (AGI) benefits all of humanity.

Key Features of the OpenAI API

- **API Key & Authentication** – To use the OpenAI API, you must sign up on the OpenAI platform, generate an API key, and include it in your requests via the `Authorization` header (e.g., `Authorization: Bearer YOUR_API_KEY`).
- **Basic API Usage** – The API is accessed via HTTP requests (usually `POST` requests) to endpoints such as `/v1/chat/completions` (for ChatGPT) and `/v1/completions` (for text generation). Requests include parameters like `model`, `messages` (for chat), and `temperature` (for randomness control).
- **Models & Their Capabilities** – OpenAI provides different models (e.g., `gpt-4`, `gpt-3.5-turbo`) for tasks like text generation, code completion, and embeddings. Some models are optimized for cost and speed (`turbo` versions).

Key Features of the OpenAI API

- **Rate Limits & Pricing** – API usage is billed based on token consumption (input + output tokens). Each model has different rate limits, and exceeding them may result in temporary request blocking.
- **SDKs & Libraries** – OpenAI provides official SDKs (`openai` Python library, `openai` Node.js package) that simplify making API requests. Alternatively, raw HTTP requests using tools like `requests` (Python) or `fetch` (JavaScript) can be used.

OpenAI Models

- **GPT Models (Text & Chat Generation)** – These models are used for natural language processing, chat-based AI, and general text generation. Examples are `gpt-3.5-turbo` and `gpt-4`.
- **Codex Models (Code Generation & Assistance)** – These models specialize in code completion and programming assistance. OpenAI deprecated Codex-specific models, but GPT-4 & GPT-3.5 are still excellent at writing, completing, and debugging code.
- **DALL-E (Image Generation)** – Creating high-quality images from text prompts. Inpainting (modifying images) and various levels of detail control. Examples include `dall-e-3`.

OpenAI Models

- **Whisper (Speech-to-Text)** – For transcribing audio into text. Converting spoken audio into text, including multilingual support. Example is `whisper-1`.
- **Embedding Models (Text Understanding & Search)** – For vectorizing text to be used in similarity search, recommendations, or AI memory systems. Examples include `text-embedding-ada-002`.

OpenAI API and Python

- **Installation:** Install the OpenAI Python package using pip:

```
pip install openai
```

- **Authentication:** Use your OpenAI API key to authenticate requests. Set the API key in your environment or pass it directly

```
import openai  
openai_service = OpenAI(api_key=OPENAI_API_KEY)
```

- **Making API Calls:** Use the OpenAI Python SDK to make API calls to various models. For example, to generate text using GPT-3.5:

```
response = openai.ChatCompletion.create(  
    model="gpt-3.5-turbo",  
    messages=[{"role": "user", "content": "Hello, how are you?"}]  
)
```



OpenAI API and Python Demo



Let's Explore OpenAI API and Python!



OpenAI API and Python Programming Next Steps

- Explore OpenAI API and Python
- Read the OpenAI API Documentation
- Try out the OpenAI API Examples
- Run the code from the webinar and explore it on your own
- Incorporate it into your next project!



Download the Code



github.com/cc-xebia-webinars/python-openai-webinar_02182025

slides and source code available

Q&A



Questions?

Thank you!



Eric Greene

eric@cloudcontraptions.com

