

API Token Manual

Fundamentals of Generative AI for Item Development

Authors: Henry Makinde, Hope Adegoke, Mubarak Mojoyinola

For this training session, we will be using **Google Gemini (Prompting)**, **Hugging Face (Fine-Tuning)**, **Groq (RAG & Agentic AI)**, and **Tavily (Agentic AI)** API keys. All other API key providers are for your information only.

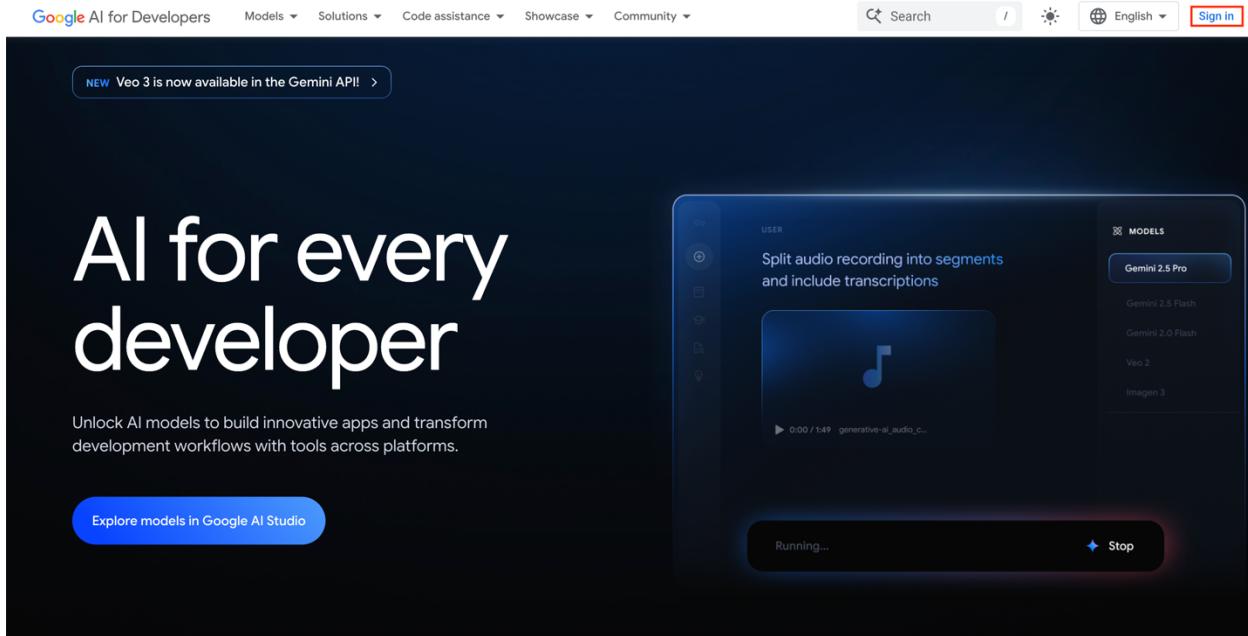
API Providers and Models

Provider	Access Type	Free Tier	Model Examples	Key Features
Google Gemini	Free Trial	Yes	Gemini 1.5 Flash/Pro	Google's foundation models
Hugging Face	Free/Paid	Yes (limited)	Falcon, Mistral, LLaMA	Wide range of open models
Groq	Free	Yes	LLaMA 3, Mixtral	Very fast inference API
Tavily	API-based (Search + RAG)	Yes — limited free usage	Tavily Search API	Real-time web search and retrieval for LLMs; integrates with LangChain and Groq
OpenAI	Paid	\$5 credits	GPT-4, GPT-3.5	High performance, versatile
Ollama (Local)	Free	N/A	LLaMA 3, Phi3, Mistral	Runs models locally, no key needed

Getting Google Gemini API Key

1. Sign Up

- Visit <https://ai.google.dev>



2. Get API Key

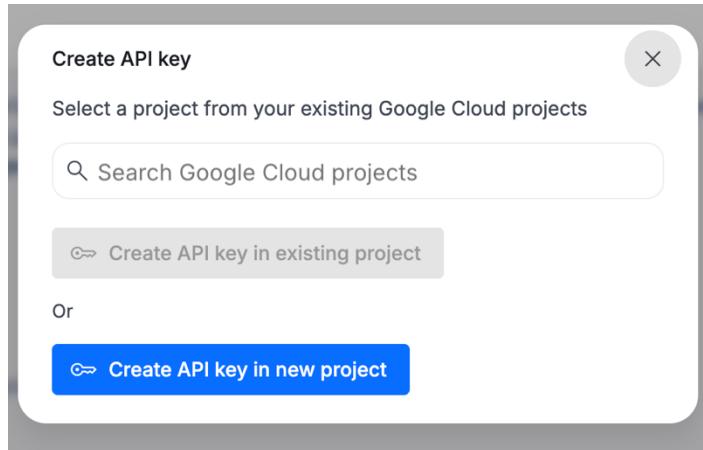
- Click Models. Under Gemini, choose docs

This screenshot shows the 'Models' section of the Google AI for Developers website. The 'Models' tab is selected and highlighted with a red box. Other tabs include 'Solutions', 'Code assistance', 'Showcase', and 'Community'. Below the tabs, there are four main categories: 'Gemini' (with 'About', 'Docs' [highlighted with a red box], and 'API reference'), 'Imagen' (with 'About', 'Docs', and 'Pricing'), 'Veo' (with 'About', 'Docs', and 'Pricing'), and 'Gemma' (with 'About', 'Docs', and 'Gemmaverse'). A large 'AI for every developer' banner is present, along with a subtext about building innovative apps. A blue 'Explore models in Google AI Studio' button is shown. On the right, a preview of the AI Studio interface displays a running audio recording task.

- Click “Get a Gemini API Key”

The screenshot shows the Gemini Developer API homepage. On the left, there's a sidebar with sections like 'Get started', 'Overview', 'Quickstart', 'API keys', 'Libraries', 'OpenAI compatibility', 'Models' (with sub-options like Gemini, Imagen, Veo, Lyria, Embeddings, Pricing, Rate limits, Billing info), 'Core Capabilities' (Text generation, Image generation, Speech generation, Long context, Structured output, Thinking), and a 'Meet the models' section. The main content area has a banner for 'Veo 3 is now available in the Gemini API!' with a 'Learn more' button. Below it, there's a navigation bar: Home > Gemini API > Models. The central part features the 'Gemini Developer API' heading and a code snippet for generating text in Python. A prominent blue button labeled 'Get a Gemini API Key' is highlighted with a red box. At the bottom right of the main content area is a 'Use Gemini in Google AI Studio' button.

- Click “Create an API Key”. A box will pop up then you choose “Create API key in new project”.



Copy the provided key to a secure location.

Use the Token

- Add to .env:

```
GOOGLE_API_KEY=your_google_key_here
```

Getting a Hugging Face API Key

1. Sign Up

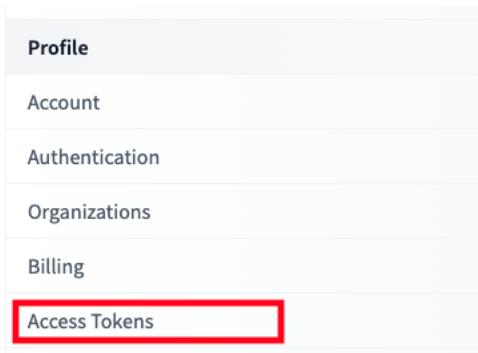
- Visit <https://huggingface.co/> and sign up

The screenshot shows the Hugging Face website. On the left, there's a dark-themed sidebar with a yellow emoji icon, the text "The AI community building the future.", and links to "Explore AI Apps" and "Browse 1M+ models". The main area features a search bar at the top right and a large list of models on the right side. The models are categorized and listed with their names, descriptions, and metrics like "Text Generation", "Updated", and "Downloads". A red box highlights the "Sign Up" button in the top right corner.

2. Generate Token

- Go to Settings > Access Tokens

The screenshot shows the user profile settings page for "BishopGreg". The sidebar on the left includes a "New" button, sections for "Profile", "Inbox (0)", "Settings" (which is highlighted with a red box), "Billing", and "Get PRO". Below these are sections for "Organizations" (with a "Create New" button) and "Resources" (with links to "Getting Started", "Documentation", "Forum", "Tasks", and "Learn"). The main content area is currently empty.



Access Tokens

User Access Tokens

+ Create new token

Access tokens authenticate your identity to the Hugging Face Hub and allow applications to perform actions based on token permissions.

ⓘ Do not share your **Access Tokens** with anyone; we regularly check for leaked Access Tokens and remove them immediately.

- Click “Create new token”, choose “read” access, give it a token name, and click “Create token”. Copy the provided key to a secure location.

Create new Access Token

Token type

Fine-grained **Read** Write

ⓘ This cannot be changed after token creation.

Token name

AIME key

This token has read-only access to all your and your orgs resources and can make calls to Inference Providers on your behalf. It can also be used to open pull requests and comment on discussions.

Create token

3. Use the Token

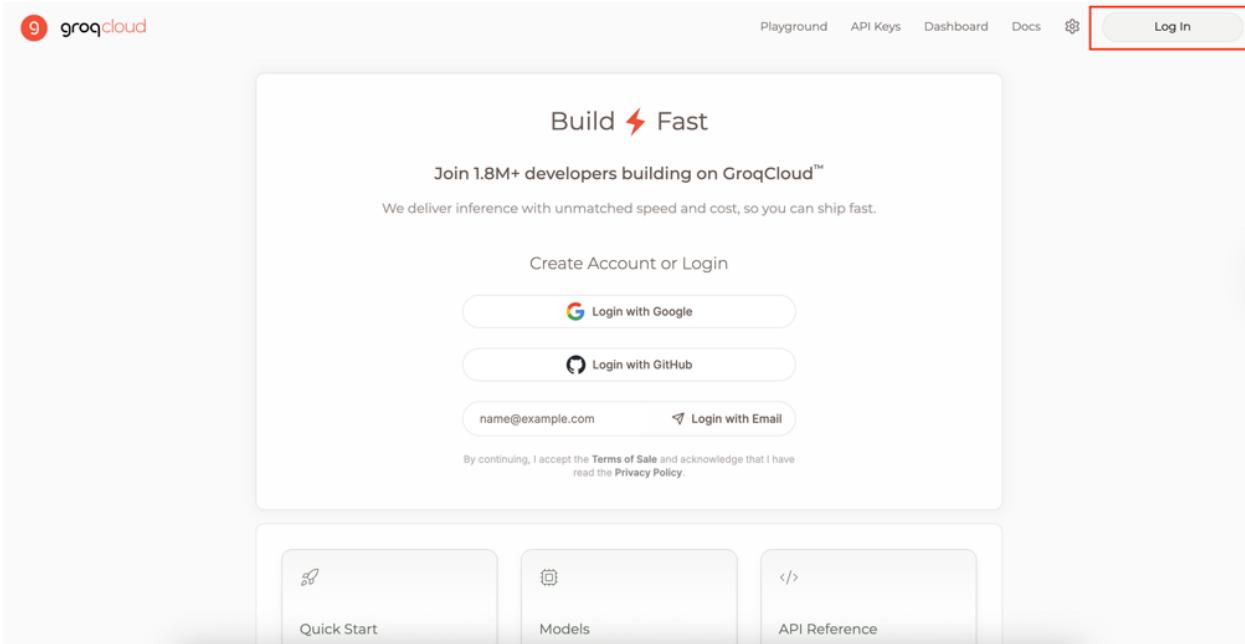
- Add to .env:

HF_API_KEY=your_huggingface_key_here

Using Groq API (Free LLaMA API)

1. Sign Up

- Go to <https://console.groq.com>
- Sign up with using Google or Github



2. Generate API Key

o > API Keys > Create new key

This screenshot shows the "API Keys" page within the GroqCloud interface. The top navigation bar has a red border around the "API Keys" tab. The main content area is titled "API Keys" and contains the sub-instruction: "Manage your project API keys. Remember to keep your API keys safe to prevent unauthorized access." To the right is a prominent red-bordered "Create API Key" button. Below this, there is a table header with columns: NAME, SECRET KEY, CREATED, LAST USED, and USAGE (24HRS). A note "(No keys)" is displayed below the table.

Give it a name and store the provided API key in a safe location.

3. Supported Models

The Models >



Remote Model Context Protocol (MCP) support is now available on Grov
Remote MCP enables AI applications to connect with external systems like databases, APIs, and tools.

[Learn More >](#)

REASONING

- GPT OSS 120B
- GPT OSS 20B
- Qwen 3 32B

FUNCTION CALLING / TOOL USE

- GPT OSS 120B
- GPT OSS 20B
- Llama 4 Scout
- Qwen 3 32B
- Kimi K2

TEXT TO SPEECH

- PlayAI TTS

SPEECH TO TEXT

- Whisper Large v3
- Whisper Large v3 Turbo

TEXT TO TEXT

- GPT OSS 120B
- GPT OSS 20B
- Kimi K2
- Llama 4 Scout
- Llama 3.3 70B

VISION

- Llama 4 Scout
- Llama 4 Maverick

MULTILINGUAL

- GPT OSS 120B
- GPT OSS 20B
- Kimi K2
- Llama 4 Scout
- Llama 3.3 70B
- Whisper Large v3

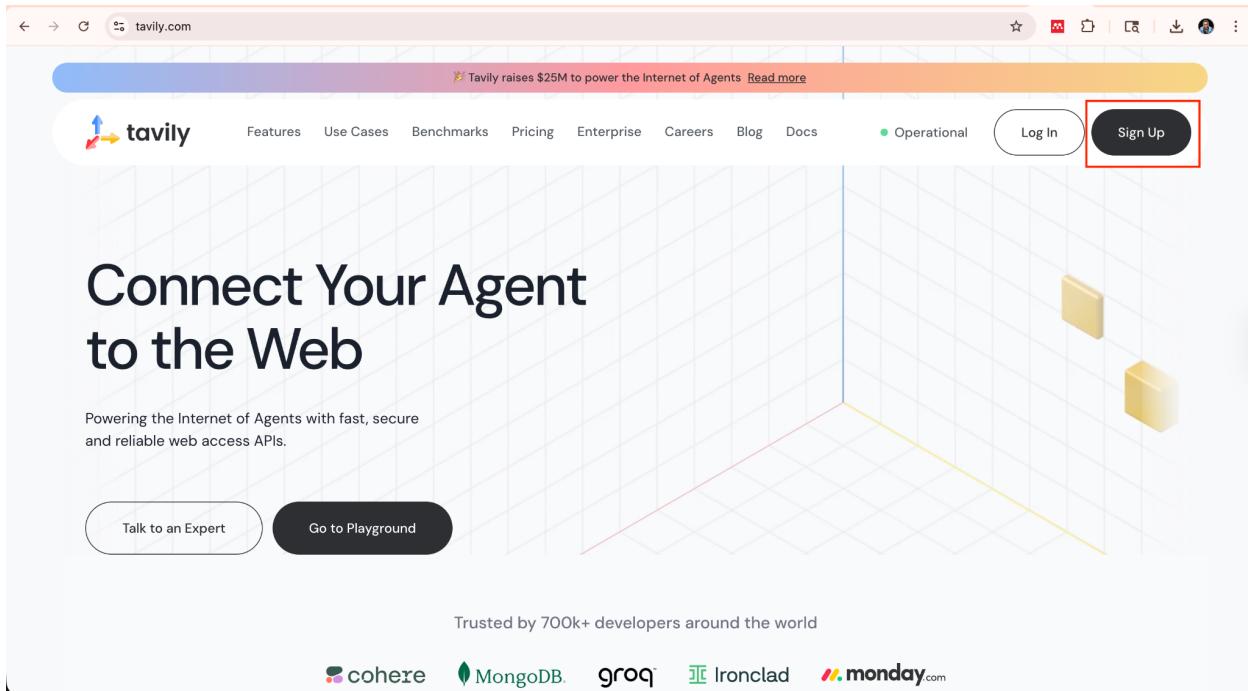
SAFETY / CONTENT MODERATION

- Llama Guard

Getting a Tavily API Key

4. Sign Up

- o Visit <https://tavily.com/> and sign up



5. Generate Token

- Click on the “+” sign in front of the API Key. A page will pop up to name and create the API key.

This screenshot shows the 'Overview' page in the Tavily dashboard. On the left, a sidebar has 'Personal' selected. The main area shows a 'CURRENT PLAN' card for 'Researcher' with an 'API Usage' section and a 'Manage Plan' button. Below this is an 'API Keys' section with a '+ Add' button highlighted with a red box. A table lists one API key: 'default' (Type: dev, Usage: 0, Key: tvly-dev-*****). There are 'OPTIONS' icons for each row. At the bottom, there's a 'Remote MCP' button and a blue message icon.

Create a new API key

Enter a name and limit for the new API key.

Key Name — A unique name to identify this key



Key Type — Choose the environment for this key

 **Development**
Rate limited to 100 requests/minute

 **Production**
Rate limited to 1,000 requests/minute

Limit monthly usage*

* If the combined usage of all your keys exceeds your plan's limit, all requests will be rejected.

Create  **Cancel**

- After creating, copy the provided key to a secure location..

6. Use the Token

- Add to .env:

```
tavily_API_KEY=your_tavily_key_here
```

Getting an OpenAI API Key

1. Sign Up

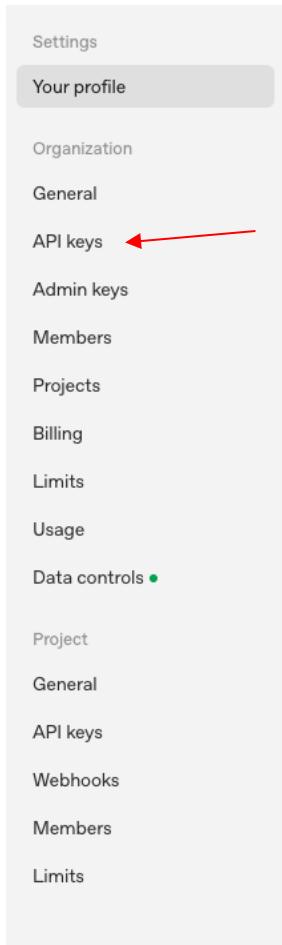
- Go to <https://platform.openai.com/>
 - Sign up using an email address or Google login

The screenshot shows the OpenAI developer platform interface. On the left is a sidebar with navigation links like 'Get started', 'Overview', 'Developer quickstart', 'Core concepts', 'Tools', and 'Agents'. The main content area features a 'Developer quickstart' section with sample code in JavaScript, a 'Browse models' section showing cards for 'GPT-4.1', 'o4-mini', and 'o3', and a 'Start building' section with buttons for 'Read and generate text' and 'Use a model's vision capabilities'.

2. Create API Key

- Click on your avatar > “Your profile” > “API Keys”

This screenshot shows the same OpenAI developer platform interface as above, but with a user profile overlay. The profile card for 'Hope Adegoke' is displayed, with the 'Your profile' button highlighted by a red box. Other options in the menu include 'Terms & policies', 'Help', and 'Log out'.



- Click “Create new secret key”. Name the key, select “default project”, click on Create secret key”, and copy the key to a safe location.

Create new secret key

Owned by

You Service account

This API key is tied to your user and can make requests against the selected project. If you are removed from the organization or project, this key will be disabled.

Name Optional

AIME key

Project

Default project

Permissions

All Restricted Read only

Cancel Create secret key

3. Store Securely

- Save in a .env file:

OPENAI_API_KEY=your_api_key_here

4. Note on Pricing

- New users get \$5 in free credits (valid 3 months)
- Paid usage starts afterward

Ollama: Local Inference (No API Key Needed)

1. Install Ollama

- Download from <https://ollama.com/download>

2. Run Model Locally

- Use terminal:

```
ollama run mistral
```

3. Benefits

- Completely free
- Ideal for private, offline use

Best Practices for Managing API Keys

• Security:

- Never hardcode keys in your scripts
- Use .env + python-dotenv to manage tokens securely

■ Load in Python:

Example:

```
from dotenv import load_dotenv
import os

load_dotenv()
api_key = os.getenv("OPENAI_API_KEY")
```

- **Rotation:** Delete old keys and rotate them periodically
- **Version Control:** Never commit .env files to GitHub