

# TokenHub 调用指南



腾讯云

## 【 版权声明 】

©2013–2026 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

## 【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

## 【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

## 【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

# 文档目录

## 调用指南

API 协议说明

文本生成

图像生成

视频生成

3D 生成

# 调用指南

## API 协议说明

最近更新时间：2026-04-03 16:27:22

### 概述

TokenHub 支持 **OpenAI 兼容协议**，兼容 OpenAI Chat Completions 接口规范，可直接使用 OpenAI SDK 接入，迁移成本极低。

### 模型与协议支持总览

模型名称	model 参数值	API 调用
HY 2.0 Think	hunyuan-2.0-thinking-20251109	详细调用参数可参见 <a href="#">文本生成</a> 。
HY 2.0 Instruct	hunyuan-2.0-instruct-20251111	
Hunyuan-role	hunyuan-role-latest	
Deepseek-v3.2	deepseek-v3.2	
Deepseek-v3.1	deepseek-v3.1-terminus	
Deepseek-r1-0528	deepseek-r1-0528	
Deepseek-v3-0324	deepseek-v3-0324	
GLM-5	glm-5	
kimi-k2.5	kimi-k2.5	
kimi-k2-thinking-turbo	kimi-k2-thinking-turbo	
kimi-k2-turbo-preview	kimi-k2-turbo-preview	
MiniMax-M2.5	minimax-m2.5	

MiniMax-M2.7	<code>minimax-m2.7</code>	
HY-Image-V3.0	<code>HY-Image-V3.0</code>	详细调用参数可参见 <a href="#">图像生成</a> 。
HY-Image-Lite	<code>HY-Image-Lite</code>	
HY-Video-1.5	<code>HY-Video-1.5</code>	详细调用参数可参见 <a href="#">视频生成</a> 。
YT-Video-2.0	<code>YT-Video-2.0</code>	
YT-Video-HumanActor	<code>YT-Video-HumanActor</code>	
YT-Video-FX	<code>YT-Video-FX</code>	
HY-3D-3.0	<code>HY-3D-3.0</code>	详细调用参数可参见 <a href="#">3D 生成</a> 。
HY-3D-3.1	<code>HY-3D-3.1</code>	
HY-3D-Express	<code>HY-3D-Express</code>	

## OpenAI 兼容协议

TokenHub 兼容了 OpenAI 的 Chat Completions 接口规范，开发者可直接使用 OpenAI 官方 SDK (Python / Node.js)，仅需替换 `base_url` 和 `api_key` 即可接入，无需对应用做额外修改。更多关于 API 使用的介绍，请参考：[OpenAI Completions API](#)。

# 文本生成

最近更新时间：2026-04-03 16:27:22

## 概述

文本生成是 TokenHub 最核心的能力之一，支持多种大语言模型，覆盖对话交互、内容创作、代码生成、推理分析等场景。

TokenHub 采用标准 **OpenAI Chat Completions API 协议**，您可以直接使用 OpenAI SDK 或任何兼容客户端接入。

## 接口地址

针对不同的地域，平台提供了不同的接入地址，以便于为您提供最稳定的接入。默认接入地址如下：

站点	接口地址
境内	<code>https://tokenhub.tencentmaas.com</code>
境外	<code>https://tokenhub-intl.tencentmaas.com</code>

同时平台提供了备用接入地址以确保异常情况下用户能顺利接入，当默认地址不可用时，您可以尝试切换为以下接入地址：

站点	接口地址
境内	<code>https://tokenhub.tencentmaas.cn</code>
境外	<code>https://tokenhub-intl.tencentmaas.cn</code>

## 鉴权方式

使用 API KEY 通过 `Authorization: Bearer` Header 鉴权。

## 请求参数

参数名	必选	类型	描述
model	是	String	服务 ID，可统一从在线推理服务、服务 ID 字段查看。 <ul style="list-style-type: none"><li>对于平台默认创建的服务，服务 ID 与模型名字相同，例如：<code>deepseek-v3</code>、<code>hunyuan-turboS-1</code>atest。</li></ul>

			<ul style="list-style-type: none"> <li>对于用户创建的自定义服务，服务 ID 格式为：<code>ep-xxxxxxxx</code>，可在 <a href="#">在线推理服务</a> 页面查看。</li> </ul>
messages	是	Array	聊天上下文消息数组，详细信息请参见 <a href="#">messages 参数说明</a> 。
stream	否	Boolean	是否启用流式输出。 取值范围： <code>true / false</code> ，默认值为 <code>false</code> 。
temperature	否	Float	输出随机性。 取值范围： <code>[0.0, 2.0]</code> 。
top_p	否	Float	输出多样性（核采样）。 取值范围： <code>[0.0, 1.0]</code> 。
max_tokens	否	Integer	限制最大输出 Token 数。
stop	否	Array of String	指定模型输出的停止序列。当生成结果命中任一指定序列时，模型将停止输出，且响应内容中不包含该停止序列。支持传入单个字符串或字符串数组，最多 4 个。 例如：让模型生成一个 10 条的清单，不希望它继续往下写第 11 条，此处可填写为： <code>["11."]</code> 。
tools	否	Array	Function Calling 工具定义列表。
tool_choice	否	String	工具调用策略： <code>none</code> （关闭） / <code>auto</code> （自动选择） / <code>required</code> （强制调用）。
seed	否	Integer	随机种子，用于结果复现。在多次请求中使用相同的 <code>seed</code> 值，并且其他参数也保持一致时，模型更有可能返回一致或非常接近的结果。

## messages 参数说明

消息数组中的每个对象包含以下字段：

字段	类型	描述
role	String	角色： <code>system</code> （系统提示）、 <code>user</code> （用户）、 <code>assistant</code> （助手）、 <code>tool</code> （工具返回）
content	String	消息文本内容。

**消息顺序规则：** `[system(可选) → user → assistant → user → ...]`，必须以 `user` 角色结尾。

## 返回参数

参数名	类型	描述
id	String	请求唯一标识。
object	String	对象类型，固定 <code>chat.completion</code> 。
created	Integer	创建时间（Unix 时间戳）。
model	String	实际使用的模型名称。
choices	Array	模型针对同一次请求返回的候选结果列表，详情请参见 <a href="#">choices 数组元素</a> 。
usage	Object	Token 消耗统计。

## choices 数组元素

字段	类型	描述
index	Integer	选项索引。
message	Object	回复消息，包含 <code>role</code> 和 <code>content</code> 。
finish_reason	String	结束原因： <code>stop</code> （正常结束）、 <code>length</code> （达到最大长度）、 <code>tool_calls</code> （需要调用工具）

## usage 对象

字段	类型	描述
prompt_tokens	Integer	输入 Token 数
completion_tokens	Integer	输出 Token 数
total_tokens	Integer	总 Token 数（按此计费）

## 示例

### 示例 1: 基础对话

```
# 请将 YOUR_API_KEY 替换为您在前面步骤创建的 API KEY
# 请在 model 字段中更换您需要体验的服务 ID
curl -X POST 'https://tokenhub.tencentmaas.com/v1/chat/completions' \
```

```
-H 'Authorization: Bearer YOUR_API_KEY' \  
-H 'Content-Type: application/json' \  
-d '{  
  "model": "deepseek-v3.1-terminus",  
  "messages": [  
    {"role": "user", "content": "你好,请介绍一下你自己"}  
  ]  
'
```

## 返回示例

```
{  
  "id": "5e9c7ae9-e0e4-4ec1-bbd0-  
22bcfda61e45", "object": "chat.completion", "model": "deepseek-v3.1-  
terminus", "choices": [{"index": 0, "message":  
  {"role": "assistant", "content": "你好!很高兴见到你! 🎉\n\n我是DeepSeek,由深度求  
索公司创造的AI助手。让我简单介绍一下自己:\n\n**我的特点:**\n- 📖 知识截止到2024年7  
月,是DeepSeek的最新版模型\n- 🗨️ 纯文本对话模型,专注于理解和生成文字内容\n- 📎 支持  
文件上传功能——可以处理图像、txt、pdf、ppt、word、excel等文件,并从中读取文字信息  
\n- 🔍 支持联网搜索(需要你在Web/App中手动开启)\n- 🧠 拥有128K的上下文长度,能记住  
我们较长的对话内容\n\n**我能帮你做什么:**\n- 回答各种问题,进行深入讨论\n- 协助写  
作、翻译、分析\n- 处理上传的文档内容\n- 提供学习、工作、生活方面的建议\n\n**重要提  
醒:**\n- 我完全免费使用,没有任何收费计划\n- 目前不支持语音功能\n- 你可以通过官方应  
用商店下载App使用\n\n我的回复风格比较热情细腻,希望能给你带来温暖的交流体验!有什么想  
聊的或需要帮助的,尽管告诉我吧! 🌟"}, "finish_reason": "stop"}], "usage":  
  {"prompt_tokens": 10, "completion_tokens": 244, "total_tokens": 254, "prompt_t  
okens_details": {"cached_token": 0}, "completion_tokens_details":  
  {"reasoning_tokens": 0}}}
```

## 示例 2: 流式对话

```
# 请将 YOUR_API_KEY 替换为您在前面步骤创建的 API KEY  
# 请在 model 字段中更换您需要体验的服务 ID  
curl -X POST 'https://tokenhub.tencentmaas.com/v1/chat/completions' \  
-H 'Authorization: Bearer YOUR_API_KEY' \  
-H 'Content-Type: application/json' \  
-d '{  
  "model": "hunyuan-2.0-instruct-20251111",  
  "messages": [  
    {"role": "system", "content": "你是一个有帮助的 AI 助手。"},
```

```
    {"role": "user", "content": "计算 1+1"}
  ],
  "stream": true
}'
```

流式返回采用服务器发送事件 **SSE (Server-Sent Events)** 格式:

```
data: {"id":"chatcpl-abc123","choices":[{"index":0,"delta":{"role":"assistant","content":"1"},"finish_reason":null}]}
```

```
data: {"id":"chatcpl-abc123","choices":[{"index":0,"delta":{"content":"+"},"finish_reason":null}]}
```

```
data: {"id":"chatcpl-abc123","choices":[{"index":0,"delta":{"content":"1"},"finish_reason":null}]}
```

```
data: {"id":"chatcpl-abc123","choices":[{"index":0,"delta":{"content":"="},"finish_reason":null}]}
```

```
data: {"id":"chatcpl-abc123","choices":[{"index":0,"delta":{"content":"2"},"finish_reason":null}]}
```

```
data: {"id":"chatcpl-abc123","choices":[{"index":0,"delta":{"},"finish_reason":"stop"}]}
```

```
data: [DONE]
```

### 示例 3: System Prompt

```
# 请将 YOUR_API_KEY 替换为您在前面步骤创建的 API KEY
# 请在 model 字段中更换您需要体验的服务 ID
curl -X POST 'https://tokenhub.tencentmaas.com/v1/chat/completions' \
  -H 'Authorization: Bearer YOUR_API_KEY' \
  -H 'Content-Type: application/json' \
  -d '{
    "model": "deepseek-v3.1-terminus",
    "messages": [
      {
        "role": "system",
```

```
"content": "你是一个专业的英语翻译助手。将用户输入的中文翻译为英文，将英文翻译为中文。只返回翻译结果，不做解释。"
```

```
},  
{  
  "role": "user",  
  "content": "今天天气真好"  
}  
]  
'
```

## 返回示例

```
{"id": "5d42fea3-413e-42ce-99b2-0d1595dae996", "object": "chat.completion", "model": "deepseek-v3.1-terminus", "choices": [{"index": 0, "message": {"role": "assistant", "content": "The weather is really nice today."}, "finish_reason": "stop"}], "usage": {"prompt_tokens": 38, "completion_tokens": 7, "total_tokens": 45, "prompt_tokens_details": {"cached_token": 0}, "completion_tokens_details": {"reasoning_tokens": 0}}}
```

## 示例 4: 多轮对话

```
# 请将 YOUR_API_KEY 替换为您在前面步骤创建的 API KEY  
# 请在 model 字段中更换您需要体验的服务 ID  
curl -X POST 'https://tokenhub.tencentmaas.com/v1/chat/completions' \  
-H 'Authorization: Bearer YOUR_API_KEY' \  
-H 'Content-Type: application/json' \  
-d '{  
  "model": "deepseek-v3.1-terminus",  
  "messages": [  
    {"role": "user", "content": "请介绍一下量子计算"},  
    {"role": "assistant", "content": "量子计算是一种利用量子力学原理进行信息处理的计算方式..."},  
    {"role": "user", "content": "它和传统计算有什么区别? "  
  ]  
}'
```

## 返回示例

```
{
  "id": "fda59c08-6a85-4514-bdbf-d77a8d68e018",
  "object": "chat.completion",
  "model": "deepseek-v3.1-terminus",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "好的，这是一个非常核心的问题。量子计算和传统计算的根区别在于它们处理信息的基本单位和工作原理。\\n\\n我们可以用一个非常经典的比喻来开始：\\n\\n*   **传统计算机**   像是在一个巨大的**图书馆**里，一个**图书管理员**（CPU）在一条很长的走廊（总线）上奔跑，一次只能打开一个房间（内存地址），查看一本书（一个比特的数据），然后做出决定。\\n*   **量子计算机**   则像是让**所有图书管理员**（量子比特）同时进入**所有房间**，并在一瞬间阅读**所有书籍的每一种可能的组合**，然后告诉你最终的结果。\\n\\n下面我们从几个关键维度进行详细对比：\\n\\n### 1. 基本信息单位：比特 vs. 量子比特\\n\\n| 特征 | 传统计算（比特） | 量子计算（量子比特） |\\n| :--- | :--- | :--- |\\n| **状态** | **二进制**：只能是 **0** 或 **1**。就像一盏灯，要么开，要么关。非常确定。 | **叠加态**：可以**同时**是0和1，或者说是0和1的任意概率组合。就像一盏同时处于开和关状态的“量子灯”。 |\\n| **表示方式** | 一个明确的、离散的值。 | 一个状态向量，用狄拉克符号表示为： $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ ，其中 $\alpha$ 和 $\beta$ 是复数，且 $|\alpha|^2 + |\beta|^2 = 1$ 。 |\\n| **核心差异** | **确定性**：每个比特在任何时刻都有明确的值。 | **概率性**：测量量子比特时，它会以  $|\alpha|^2$  的概率坍缩为0，以  $|\beta|^2$  的概率坍缩为1。 |\\n\\n### 2. 工作原理：逻辑门 vs. 量子特性\\n\\n| 特征 | 传统计算 | 量子计算 |\\n| :--- | :--- | :--- |\\n| **操作方式** | 使用**逻辑门**（如与门、或门、非门）对比特进行运算。一次操作改变一个或一组比特的状态。 | 使用**量子逻辑门**对量子比特进行操作。这些操作是**可逆的**，并能利用叠加态进行**并行计算**。 |\\n| **核心优势** | **串行处理**：任务被分解为一系列步骤，按顺序执行。对于简单、逻辑清晰的任务效率极高。 | **量子并行性**：由于量子比特处于叠加态，一次量子操作可以**同时作用于所有可能的输入**。这是量子加速的根源。 |\\n| **独特现象** | 无 | **量子纠缠**：两个或多个量子比特可以形成一种神秘的关联，无论它们相距多远，对一个量子比特的测量会瞬间决定另一个的状态。这允许量子计算机将不同量子比特的状态紧密联系起来，进行高度协同的计算。 |\\n\\n### 3. 性能与适用领域\\n\\n| 特征 | 传统计算 | 量子计算 |\\n| :--- | :--- | :--- |\\n| **擅长任务** | - **通用计算**：办公软件、网页浏览、游戏 - **逻辑控制**：操作系统、应用程序逻辑 - **大部分数据处理**：数据库管理、电子表格 - **特定领域的指数级加速**： - **密码学**：破解RSA等加密算法（Shor算法） - **材料模拟**：精确模拟分子和材料的量子性质 - **优化问题**：物流路线规划、金融投资组合优化 - **人工智能**：加速机器学习训练 |\\n| **计算复杂度** | 对于某些复杂问题（如大数分解），传统算法需要**指数级**增长的时间。 | 对于特定问题，量子算法可将复杂度降至**多项式**级别，实现“量子优越性”。 |\\n| **输出结果** | 精确、确定的结果。 | 通常是**概率性**的结果。由于需要测量，我们得到的是一个可能正确的答案，因此算法通常需要多次运行以提高置信度。 |\\n\\n### 4. 物理实现与挑战\\n\\n| 特征 | 传统计算机 | 量子计算机 |\\n| :--- | :--- | :--- |\\n| **硬件基础** | 基于**晶体管**（半导体），技术成熟，可大规模集成（如CPU有数十亿晶体管）。 | 需要能保持量子态的物理系统，如：超导电路、离子阱、
```

光子等。技术尚在早期。 | \n| **主要挑战** | 功耗、散热、晶体管尺寸接近物理极限（摩尔定律放缓）。 | **量子退相干**：量子态极其脆弱，极易受环境（如热、振动）干扰而失去量子特性。需要极低温（接近绝对零度）和高度隔离的环境。 | \n| **错误纠正** | 错误率极低，纠错相对简单（如奇偶校验）。 | 错误率很高，需要复杂的**量子纠错码**，用多个物理量子比特来编码一个逻辑量子比特，开销巨大。 | \n\n### 总结表格\n\n| 对比维度 | 传统计算 | 量子计算 | \n| :--- | :--- | :--- | \n| **基本单位** | 比特（0 或 1） | 量子比特（叠加态：0和1的叠加） | \n| **操作方式** | 逻辑门（串行） | 量子门（并行） | \n| **核心原理** | 布尔逻辑 | 叠加、纠缠、干涉 | \n| **结果输出** | 确定性 | 概率性 | \n| **擅长领域** | 通用任务、逻辑控制 | 特定复杂问题（如模拟、优化、密码破译） | \n| **技术成熟度** | 非常成熟，广泛应用 | 早期阶段，主要用于研究和特定计算 | \n| **与用户关系** | **替代关系**：量子计算机**不是**用来取代你的手机或笔记本电脑的。它更像一个**专用加速器**，用于解决传统计算机在可预见未来内都无法解决的特定难题。未来，我们可能通过云端访问量子计算机，让它处理最复杂的部分，而传统计算机负责日常任务和用户交互。 | \n\n简单来说，传统计算机是“精准的快枪手”，而量子计算机是“能同时探索所有可能性的先知”。它们各有千秋，将在未来很长一段时期内协同工作。”}, "finish\_reason": "stop"}], "usage": {"prompt\_tokens": 32, "completion\_tokens": 1321, "total\_tokens": 1353, "prompt\_tokens\_details": {"cached\_token": 0}, "completion\_tokens\_details": {"reasoning\_tokens": 0}}}

## 示例 5: Function Calling (工具调用)

```
# 请将 YOUR_API_KEY 替换为您在前面步骤创建的 API KEY
# 请在 model 字段中更换您需要体验的服务 ID
curl -X POST 'https://tokenhub.tencentmaas.com/v1/chat/completions' \
  -H 'Authorization: Bearer YOUR_API_KEY' \
  -H 'Content-Type: application/json' \
  -d '{
    "model": "deepseek-v3.1-terminus",
    "messages": [
      {"role": "user", "content": "北京今天天气怎么样?"}
    ],
    "tools": [
      {
        "type": "function",
        "function": {
          "name": "get_weather",
          "description": "获取指定城市的天气信息",
          "parameters": {
```

```
"type": "object",
  "properties": {
    "city": {
      "type": "string",
      "description": "城市名称, 如: 北京"
    }
  },
  "required": ["city"]
}
],
"tool_choice": "auto"
}'
```

当模型决定调用工具时，返回：

```
{
  "choices": [
    {
      "message": {
        "role": "assistant",
        "content": null,
        "tool_calls": [
          {
            "id": "call_abc123",
            "type": "function",
            "function": {
              "name": "get_weather",
              "arguments": "{\"city\": \"北京\"}"
            }
          }
        ]
      },
      "finish_reason": "tool_calls"
    }
  ]
}
```

将工具执行结果返回模型，继续对话：

```
{
  "model": "deepseek-v3",
  "messages": [
    {"role": "user", "content": "北京今天天气怎么样? "},
    {"role": "assistant", "content": null, "tool_calls": [{"id":
"call_abc123", "type": "function", "function": {"name": "get_weather",
"arguments": "{\\"city\\": \\"北京\\"}"}}]},
    {"role": "tool", "tool_call_id": "call_abc123", "content": "
{\\"temperature\\": 22, \\"weather\\": \\"晴\\", \\"humidity\\": 45}"}
```

## 示例 6: 使用 Python SDK

```
from openai import OpenAI

client = OpenAI(
# 请将 YOUR_API_KEY 替换为您在前面步骤创建的 API KEY
  api_key="YOUR_API_KEY",
  base_url="https://tokenhub.tencentmaas.com/v1"
)

# 基础对话
response = client.chat.completions.create(
# 请在 model 字段中更换您需要体验的服务 ID
  model="deepseek-v3.1-terminus",
  messages=[
    {"role": "system", "content": "你是一个有帮助的助手。"},
    {"role": "user", "content": "请用一句话解释什么是大语言模型"}
  ]
)

print(response.choices[0].message.content)
```

## 示例 7: 使用 Node.js SDK

```
import OpenAI from 'openai';
```

```
const client = new OpenAI({
// 请将 YOUR_API_KEY 替换为您在前面步骤创建的 API KEY
  apiKey: 'YOUR_API_KEY',
  baseURL: 'https://tokenhub.tencentmaas.com/v1',
});

async function main() {
  const response = await client.chat.completions.create({
// 请在 model 字段中更换您需要体验的服务 ID
    model: 'deepseek-v3.1-terminus',
    messages: [
      { role: 'system', content: '你是一个有帮助的助手。' },
      { role: 'user', content: '请用一句话解释什么是大语言模型' },
    ],
  });
  console.log(response.choices[0].message.content);
}
main();
```

## 错误码

HTTP 状态码	错误码	错误说明
400	invalid_request_error	请求参数有误
401	authentication_error	API Key 无效
403	model_not_available	模型不存在或未开通
429	rate_limit_exceeded	请求频率超限
500	engine_server_error	引擎内部错误
503	service_unavailable	服务暂时不可用

## 支持的模型

文本生成支持的完整模型列表请参见 [模型列表](#)，常用模型如：

模型	model 值	特点
DeepSeek-V3.2	deepseek-v3.2	第三代通用模型，128K 上下文

DeepSeek-R1	<code>deepseek-r1-0528</code>	深度推理模型
HY 2.0 Think	<code>hunyuan-2.0-thinking-20251109</code>	旗舰思考模型
GLM-5	<code>glm-5</code>	智谱旗舰模型，Agent 任务能力出色
Kimi-K2.5	<code>kimi-k2.5</code>	万亿参数 MoE 模型

## 下一步

- 图像生成接口，请参见 [图像生成](#)。
- 视频生成接口，请参见 [视频生成](#)。
- 3D 生成接口，请参见 [3D 生成](#)。

# 图像生成

最近更新时间：2026-04-03 16:27:22

## 概述

腾讯混元生图是一款提供 AI 图像生成与处理能力的 API 技术服务，可以结合输入的图片或文本智能创作出与输入相关的图像内容，具有更强大的中文理解能力、更多样化的风格选择，以及更偏东方审美的绘画创作能力，能更好地支持中文场景下的建筑风景生成、古诗词理解、水墨剪纸等中国元素风格生成，以及各种动漫、游戏风格的高精度图像生成和风格转换，为高质量的内容创作、内容运营提供技术支持。

## 产品功能

### 混元生图（3.0）

混元生图3.0（即 HY-Image-V3.0），基于混元大模型，能够思考图像的布局、构图、笔触，利用世界知识去推理常识性的画面。同时可以解析千字级别的复杂语义，生成包含文本的图片、复杂漫画、表情包，还能生成生动有趣的科普插画。

### 混元生图（极速版）

混元生图极速版（即 HY-Image-Lite），基于混元大模型，采用超高压压缩编解码器，实现图像生成快速响应与高品质输出。支持电商商品图美化、设计工具素材生成、游戏场景迭代等场景。

## 立即使用

TokenHub 采用标准 **OpenAI Chat Completions API 协议**，您可以直接使用 OpenAI SDK 或任何兼容客户端接入。通过调用 API 对腾讯混元生图模型进行操作，例如 HY-Image-V3.0、HY-Image-Lite。

具体使用方式请参见对应接口文档：

模型	接口名称	接口文档	接口功能
HY-Image-Lite	<a href="#">TextToImageLite</a>	<a href="#">混元生图（极速版）</a>	混元文生图接口，基于混元大模型，根据输入的文本描述快速生成图片。
HY-Image-V3.0	<a href="#">SubmitTextToImageJob</a>	<a href="#">提交混元生图（3.0）任务</a>	混元生图接口，基于混元大模型，根据输入的文本/参考图描述智能生成图片。
	<a href="#">QueryTextToImageJob</a>	<a href="#">查询混元生图（3.0）任务</a>	

## OpenAI 兼容接口调用示例

### 混元生图（极速版）接口

混元生图（极速版）API 兼容了 OpenAI 的接口规范，这意味着您可以在 OpenAI 协议中使用 cURL 来调用混元生图大模型。您仅需要将 `YOUR_API_KEY` 替换成混元的相关配置，即可将应用切换到混元生图。

**base\_url:** `https://tokenhub.tencentmaas.com/v1/api/image/lite`。

请求示例代码如下，其中：请求的具体参数与 [混元生图（极速版）](#) 接口一致，但参数统一以小写驼峰形式呈现，例如 `LogoAdd` 需要写为 `logo_add`。

### cURL

```
curl --location 'https://tokenhub.tencentmaas.com/v1/api/image/lite'\
--header 'Authorization: Bearer YOUR_API_KEY' \
--header 'Content-Type: application/json' \
--data '{
  "model": "hy-image-lite",
  "prompt": "雨中, 竹林, 小路",
  "rsp_img_type": "url"
}'
```

响应示例代码如下：

### cURL

```
{
  "created": 1774806537,
  "request_id": "ce*****c3",
  "data": [
    {
      "url": "https://hyimg*****9c81de85"
    }
  ]
}
```

## 混元生图（3.0）接口

### 提交混元生图（3.0）任务

混元生图（3.0）API 兼容了 OpenAI 的接口规范，这意味着您可以在 OpenAI 协议中使用 cURL 来调用混元生图大模型。您仅需要将 `YOUR_API_KEY` 替换成混元的相关配置，即可将您的应用切换到混元生图。

**base\_url:** `https://tokenhub.tencentmaas.com/v1/api/image/submit`。

请求示例代码如下，其中：请求的具体的参数与 [提交混元生图（3.0）任务](#) 一致，但参数统一以小写驼峰形式呈现，如 `LogoAdd` 需要写为 `logo_add`。

📌 **说明:**  
images 可以为任意可访问的图片地址。

### cURL

```
curl -X POST 'https://tokenhub.tencentmaas.com/v1/api/image/submit' \  
-H 'Authorization: Bearer YOUR_API_KEY' \  
-H 'Content-Type: application/json' \  
-d '{  
  "model": "hy-image-v3.0",  
  "prompt": "在图片中增加一个橘猫",  
  "images": [  
    "https://*****.jpeg"  
  ]  
}'
```

返回结果示例代码如下:

### cURL

```
{  
  "id": "251*****0",  
  "request_id": "96*****1d4",  
  "object": "image_job",  
  "created_at": 1774806585,  
  "status": "queued"  
}
```

### 查询混元生图（3.0）任务

将 `YOUR_API_KEY` 替换成混元的相关配置，发送查询相关请求，即可获取混元生图（3.0）的图片生成结果。其中：`id` 为提交混元生图（3.0）任务时响应代码中的 `id`。

**base\_url:** `https://tokenhub.tencentmaas.com/v1/api/image/query`。

请求示例代码如下，其中：请求的具体的参数与 [查询混元生图（3.0）任务](#) 一致，但参数统一以小写驼峰形式呈现。

### cURL

```
curl -X POST 'https://tokenhub.tencentmaas.com/v1/api/image/query' \  
-H 'Authorization: Bearer YOUR_API_KEY' \  
-H 'Content-Type: application/json' \  
-d '{  
  "model": "hy-image-v3.0",  
  "id": "251*****0"  
}'
```

返回结果示例代码如下：

## cURL

```
{  
  "request_id": "5785****da7f2af",  
  "object": "image_job",  
  "created_at": 1774806639,  
  "completed_at": 1774806639,  
  "status": "completed",  
  "data": [  
    {  
      "url": "https://****965e",  
      "revised_prompt": "\u003cthink\u003e用户希望在图片中增加一*****  
纹理。  
\u003c/recaption\u003e\u003canswer\u003e\u003cboi\u003e\u003cimg_size_10  
24\u003e\u003cimg_ratio_34\u003e"  
    }  
  ]  
}
```

# 视频生成

最近更新时间：2026-04-03 17:20:21

## 概述

腾讯混元生视频（Tencent HY Video），是一款提供视频生成和视频处理能力的 API 技术服务。该服务基于腾讯视频生成大模型等一系列领先的音视频 AI 技术，支持高质量地生成或处理视频内容。既能帮助专业视频创作者降低制作成本、发现视频创意，又能提升视频社交娱乐的趣味性。可广泛应用于短视频平台、影视制作、广告营销、社交媒体、游戏等领域。

## 产品功能

### 文/图生视频

支持文本和图片输入生成动态连贯性高的高清视频，可实现场景切换，简化制作流程、降低成本，应用于高要求的广告、影视片段与产品展示视频场景。

### 视频特效

通过上传图片 and 选择特效模板，生成一段特效视频，将静态图像转化为充满活力、动感、有趣的视频画面。

### 人像驱动

单张参考照片即可驱动生成动态人像视频，精准还原表情、姿态，支持写实、二次元等多风格切换。

## 立即使用

TokenHub 采用标准 OpenAI Chat Completions API 协议，您可以直接使用 OpenAI SDK 或任何兼容客户端接入。您可以调用 API 对生视频模型进行操作，具体使用方式详见对应接口文档。

## 接口概览

模型	接口名称	接口文档	接口功能
HY-Video-1.5	<a href="#">SubmitHunyuanToVideoJob</a>	<a href="#">提交混元生视频任务</a>	混元生视频接口，基于混元大模型，根据输入的文本或图片智能生成视频。
	<a href="#">DescribeHunyuanToVideoJob</a>	<a href="#">查询混元生视频任务</a>	
YT-Video-2.0	<a href="#">DescribeImageToVideoGenerationJob</a>	<a href="#">查询图生视频通用能力任务</a>	图生视频通用能力接口。

	<a href="#">SubmitImageToVideoGeneralJob</a>	<a href="#">提交图生视频通用能力任务</a>	
YT-Video-FX	<a href="#">SubmitTemplateToVideoJob</a>	<a href="#">提交视频特效任务</a>	通过上传图片 and 选择特效模板，生成一段特效视频。 特效模板详情请参见 <a href="#">视频特效模板列表</a> 。
	<a href="#">DescribeTemplateToVideoJob</a>	<a href="#">查询视频特效任务</a>	
YT-Video-HumanActor	<a href="#">DescribeHumanActorJob</a>	<a href="#">查询人像驱动任务</a>	支持提交音频和图文来生成对应视频，满足动态交互、内容生产等场景需求。
	<a href="#">SubmitHumanActorJob</a>	<a href="#">提交人像驱动任务</a>	

## OpenAI 兼容接口调用示例

### HY-Video-1.5

#### 提交混元生视频任务

混元生视频 API 兼容了 OpenAI 的接口规范，这意味着您可以在 OpenAI 协议中使用 cURL 来调用混元生视频大模型。您仅需要将 `base_url` 和 `YOUR_API_KEY` 替换成混元的相关配置，即可将您的应用切换到混元生视频。

`base_url`: `https://tokenhub.tencentmaas.com/v1/api/video/submit`。

请求示例代码如下，其中：请求的具体参数与 [提交混元生视频任务](#) 一致，但参数统一以小写驼峰形式呈现，例如 `LogoAdd` 需要写为 `logo_add`。

#### cURL

```
curl --location 'https://tokenhub.tencentmaas.com/v1/api/video/submit' \
--header 'Authorization: Bearer YOUR_API_KEY' \
--header 'Content-Type: application/json' \
--data '{
  "model": "hy-video-1.5",
  "prompt": "小河流水"
}'
```

响应示例代码如下：

#### cURL

```
{
  "id": "143*****128",
  "request_id": "d5*****_*****_*****_*****c506",
  "object": "video",
  "created_at": 1775196710,
  "status": "queued"
}
```

### 查询混元生视频任务

将 `base_url` 和 `YOUR_API_KEY` 替换成混元的相关配置，即可获得混元生视频的生成结果。其中：`id` 为提交混元生视频任务时响应代码中的 `id`。

`base_url`: `https://tokenhub.tencentmaas.com/v1/api/video/query`

请求示例代码如下，其中：请求的具体参数与 [查询混元生视频任务](#) 一致，但参数统一以小写驼峰形式呈现，例如 `LogoAdd` 需要写为 `logo_add`。

请求示例代码：

### cURL

```
curl --location 'https://tokenhub.tencentmaas.com/v1/api/video/query' \
--header 'Authorization: Bearer YOUR_API_KEY' \
--header 'Content-Type: application/json' \
--data '{
  "model": "hy-video-1.5",
  "id": "1429*****0"
}'
```

返回响应示例代码：

### cURL

```
{
  "request_id": "bb*****93",
  "object": "video",
  "created_at": 1774807628,
```

```
"completed_at": 1774807628,
"status": "completed",
"progress": 100,
"data": {
  "url": "https://vc*****e6a6"
}
}
```

## YT-Video-2.0/YT-Video-FX/YT-Video-HumanActor

### 提交优图生视频任务

优图生视频系列模型 API 兼容了 OpenAI 的接口规范，这意味着您可以在 OpenAI 协议中使用 cURL 来调用优图生视频模型。您仅需要将 `base_url` 和 `YOUR_API_KEY` 替换成相关配置，即可将您的应用切换到优图生视频。

`base_url`: `https://tokenhub.tencentmaas.com/v1/api/video/submit`。

请求示例代码如下，其中，请求的具体参数与原接口一致（YT-Video-2.0为 [图生视频接口](#)，YT-Video-FX为 [视频特效接口](#)，YT-Video-HumanActor 为 [人像驱动接口](#)），但参数统一以小写驼峰形式呈现，例如 `LogoAdd` 需要写为 `logo_add`。

以下以 YT-Video-2.0模型为例（注：image 可以为任意可访问的图片地址）：

### cURL

```
curl --location 'https://tokenhub.tencentmaas.com/v1/api/video/submit' \
--header 'Authorization: Bearer YOUR_API_KEY' \
--header 'Content-Type: application/json' \
--data '{
  "model": "yt-video-2.0",
  "image": {"url": "https://obj1-*****.jpeg"}
}'
```

响应示例代码如下：

### cURL

```
{
  "id": "*****143680",
  "request_id": "d0e*****a05",
```

```
"object": "video",
"created_at": 1774808129,
"status": "queued"
}
```

### 查询优图生视频任务

将 `base_url` 和 `YOUR_API_KEY` 替换成相关配置，即可获得优图生视频的生成结果。其中：`id` 为提交 `YT-Video-2.0` 任务时响应代码中的 `id`。

`base_url`: `https://tokenhub.tencentmaas.com/v1/api/video/query`

请求示例代码如下，其中：请求的具体参数与原接口一致（`YT-Video-2.0` 为 [查询图生视频通用能力任务](#)，`YT-Video-FX` 为 [查询视频特效任务](#)，`YT-Video-HumanActor` 为 [查询人像驱动任务](#)），但参数统一以小写驼峰形式呈现，例如 `LogoAdd` 需要写为 `logo_add`。

以下以 `YT-Video-2.0` 模型为例：

#### cURL

```
curl --location 'https://tokenhub.tencentmaas.com/v1/api/video/query' \
--header 'Authorization: Bearer YOUR_API_KEY' \
--header 'Content-Type: application/json' \
--data '{
  "model": "yt-video-2.0",
  "id": "1429*****0"
}'
```

返回响应示例代码：

#### cURL

```
{
  "request_id": "bb*****93",
  "object": "video",
  "created_at": 1774807628,
  "completed_at": 1774807628,
  "status": "completed",
  "progress": 100,
  "data": {
```

```
"url": "https://vc*****e6a6"  
}  
}
```

# 3D 生成

最近更新时间：2026-04-03 16:27:22

## 概述

腾讯混元生3D（Tencent HY 3D）基于腾讯自研生成式 AI 大模型，提供3D内容生成的 API 技术服务。API 支持通过文本描述或上传2D 图像自动创建高精度3D 模型，可广泛应用于3D 打印、家装家具、产品设计等领域。混元生3D API 显著降低了专业级建模的技术门槛，提升了创作效率及设计灵活性，赋能多行业高效完成3D 内容生产。

## 产品功能

### 混元生3D（专业版）

可选用混元生3D 3.0、3.1 模型，生成更高精度以及更高质量的3D 模型，支持文生3D、图生3D、多视图生3D、单几何生成（白模）、草图生3D、智能拓扑生3D 功能；在此基础上，3.1 版本模型在几何、纹理质量上实现进一步提升，支持八视图多角度输入。

### 混元生3D（极速版）

即 HY-3D-Express，采用混元生3D 极速版模型，可将生成模型时间缩短至1分30秒内，可在较短时间内生成3D 模型文件。

## 立即使用

TokenHub 采用标准 OpenAI Chat Completions API 协议，您可以直接使用 OpenAI SDK 或任何兼容客户端接入。您可以调用 API 对腾讯混元生3D 模型进行操作，具体使用方式详见对应接口文档。

模型	接口名称	Model name	接口文档	接口功能
HY-3D-3.0	<a href="#">SubmitHun yuanTo3DProJob</a>	hy-3d-3.0	<a href="#">提交混元生3D 专业版任务</a>	生成更高精度以及更高质量的3D 模型，支持文生3D、图生3D、多视图生3D、单几何生成（白模）、草图生3D、智能拓扑生3D 功能。
	<a href="#">QueryHun yuanTo3D ProJob</a>	hy-3d-3.0	<a href="#">查询混元生3D 专业版任务</a>	
HY-3D-3.1	<a href="#">SubmitHun yuanTo3DProJob</a>	hy-3d-3.1	<a href="#">提交混元生3D 专业版任务</a>	3.1 版本模型在几何、纹理质量上实现进一步提升，支持八视图多角度输入、文生3D、图生3D、单几何生成（白模）功能。
	<a href="#">QueryHun yuanTo3D</a>	hy-3d-3.1	<a href="#">查询混元生3D 专业版任务</a>	

	ProJob		务	
HY-3D-Express	SubmitHun nyuanTo3 DRapidJob	-	提交混元生 3D 极速版任 务	可在较短时间内生成3D 模型文件。
	QueryHun yuanTo3D RapidJob	-	查询混元生 3D 极速版任 务	

## OpenAI 兼容接口调用示例

混元生3D API 兼容了 OpenAI 的接口规范，这意味着您可以在 OpenAI 协议中使用 cURL 来调用混元生3D 大模型。您仅需要将 YOUR\_API\_KEY 替换成混元的相关配置，即可将您的应用切换到混元生3D。

### 提交混元生3D 任务

**base\_url:** `https://tokenhub.tencentmaas.com/v1/api/3d/submit`。

请求示例代码如下，其中：HY-3D-3.0、HY-3D-3.1请求的具体参数与 [提交混元生3D 专业版任务](#) 接口一致；HY-3D-Express 请求的具体参数与 [提交混元生3D 极速版任务](#) 接口一致。但参数统一以小写驼峰形式呈现，如 `ResultFormat` 需要写为 `result_format`。

以下以 HY-3D-3.0模型为例：

#### cURL

```
curl --location 'https://tokenhub.tencentmaas.com/v1/api/3d/submit' \
--header 'Authorization: Bearer YOUR_API_KEY' \
--header 'Content-Type: application/json' \
--data '{
  "model": "hy-3d-3.0",
  "prompt": "一只小狗"
}'
```

响应示例代码如下，请记录响应 id，作为查询混元生3D 任务的输入参数：

#### cURL

```
{
  "id": "14*****984",
  "request_id": "75*****33",
```

```
"object": "3d_job",
"created_at": 1774806931,
"status": "queued"
}
```

## 查询混元生3D 任务

**base\_url:** `https://tokenhub.tencentmaas.com/v1/api/3d/query`。

请求示例代码如下，其中：`id` 为提交混元生3D 任务时响应代码中的 `id`；HY-3D-3.0、HY-3D-3.1请求的具体参数与 [查询混元生3D 专业版任务](#) 接口一致；HY-3D-Express 请求的具体参数与 [查询混元生3D 极速版任务](#) 接口一致。但参数统一以小写驼峰形式呈现，如 `ResultFormat` 需要写为 `result_format`。

以下以 HY-3D-3.0模型为例：

### cURL

```
curl --location 'https://tokenhub.tencentmaas.com/v1/api/3d/query' \
--header 'Authorization: Bearer YOUR_API_KEY' \
--header 'Content-Type: application/json' \
--data '{
  "model": "hy-3d-3.0",
  "id": "1429890795996585984"
}'
```

响应（生成中）：

### cURL

```
{
  "request_id": "c41*****bf",
  "object": "3d_job",
  "created_at": 1774806968,
  "status": "in_progress"
}
```

响应（已完成）：

## cURL

```
{
  "request_id": "7824*****50",
  "object": "3d_job",
  "created_at": 1774807344,
  "completed_at": 1774807344,
  "status": "completed",
  "data": [
    {
      "type": "obj",
      "url": "https://hunyuan-test*****552a3",
      "preview_image_url": "https://hunyuan-
test*****125a56c0081cc404"
    },
    {
      "type": "glb",
      "url": "https://hunyuan*****-b668642f9ae",
      "preview_image_url": "https://hunyuan***1deca56c0081cc404"
    }
  ]
}
```