

# 目 录

API 对接

  认证方式

  域名入口

  模型列表

  Chat Completions

  文生图

  文生视频/图生视频

第三方工具集成

  Claude Code 接入指南

  OpenAI Codex 接入指南

常见问题

  Q&A

  错误码

# API 对接

## 认证方式

## 认证方式

AI Hub 默认采用 OpenAI 认证方式 (Bearer Token) 。

所有请求 (包含 POST、GET) 都必须带上 `Authorization` Header。

```
Authorization: Bearer <your-api-key>
```

## 注意事项

- 请求认证使用 Bearer Token 方式。
- API Key 格式为平台分配的令牌字符串。
- 请妥善保管您的 API Key，不要在客户端代码中暴露。

## 域名入口

## 域名入口

AI Hub 所有 API 接口统一通过以下域名访问：

## 统一域名

```
https://ai-api.mandao.com
```

所有模型（包括 OpenAI、Gemini、Claude、MiniMAX、Kimi、Grok、Deepseek 等）均通过该统一域名访问，无需区分不同的子域名。

## 支持的接口协议

协议格式	基础路径	说明
OpenAI 兼容	<code>/v1</code>	Chat/Completions、Responses、Images、Embeddings、Audio、Rerank
Claude 原生 (Anthropic)	<code>/v1/messages</code>	Claude Messages API 原生格式

# 模型列表

## 模型列表接口

```
GET https://ai-api.mandao.com/v1/models
```

### 示例

```
curl -X GET 'https://ai-api.mandao.com/v1/models' \  
-H 'Authorization: sk-XXX' \  
-H 'Content-Type: application/json'
```

返回所有模型列表。

# Chat Completions

## API 参考

### 认证方式

所有 API 请求需要在 Header 中包含你的 API 密钥:

Authorization: Bearer YOUR\_API\_KEY

### 端点:

POST <https://ai-api.mandao.com/v1/chat/completions>

### 请求体:

```
{  
  "model": "gpt-4o",  
  "messages": [  
    {"role": "system", "content": "You are a helpful assistant."},  
    {"role": "user", "content": "Hello!"}  
  ],  
  "temperature": 0.7,  
  "max_tokens": 1000  
}
```

## 参数说明

参数	类型	描述
model	string	模型 ID
messages	array	消息数组
temperature	float	采样温度 (0-2)
max_tokens	integer	最大输出 tokens

# 文生图

## 文生图 API

本文介绍 `wan2.7-image`、`wan2.7-image-pro` 模型调用 API 的输入输出参数，供您使用接口时查阅字段含义。

### 1.1 部分模型列表

模型名	说明
<code>wan2.7-image</code>	Wan2.7 图片生成模型
<code>wan2.7-image-pro</code>	Wan2.7 图片生成 Pro 模型

### 接口

POST `https://ai-api.mandao.com/v1/images/generations`

### 1.2 请求头

参数名	是否必填	说明
<code>Authorization</code>	是	认证令牌，格式 <code>Bearer sk-xxxx</code> 或 <code>Access-Token: sk-xxxx</code>
<code>Content-Type</code>	是	<code>application/json</code> ; 图片编辑接口支持

参数名	是否必填	说明
		multipart/form-data

### 1.3 请求参数 (ImageRequest)

参数名	类型	是否必填	默认值	功能说明
model	string	是	-	模型名称，支持 pro、
prompt	string	是	-	图像生成提示词
n	uint	否	1	生成图片数量
size	string	条件必填	见说明	图片尺寸支持： <ul style="list-style-type: none"> <li>dall-e-2 支持：1024x1024</li> <li>dall-e-3 支持：1024x1024</li> <li>不可使用</li> </ul>
quality	string	否	见说明	图片质量支持： <ul style="list-style-type: none"> <li>dall-e-2</li> <li>gpt-4o</li> </ul>
response_format	string	否	-	响应格式

参数名	类型	是否必填	默认值	功能说明
<code>style</code>	<code>json</code>	否	-	风格参数
<code>user</code>	<code>json</code>	否	-	终端用户
<code>watermark</code>	<code>bool</code>	否	-	是否添加
<code>background</code>	<code>json</code>	否	-	背景参数
<code>moderation</code>	<code>json</code>	否	-	内容审核
<code>output_format</code>	<code>json</code>	否	-	输出格式
<code>output_compression</code>	<code>json</code>	否	-	输出压缩
<code>partial_images</code>	<code>json</code>	否	-	局部重绘
<code>extra_fields</code>	<code>json</code>	否	-	额外字段
<code>image</code>	<code>json</code>	否	-	图片数据
<code>watermark_enabled</code>	<code>json</code>	否	-	智谱 4V
<code>user_id</code>	<code>json</code>	否	-	智谱用户

特殊说明：

- `images/edits` 接口支持 `multipart/form-data` 提交，字段与 JSON 格式一致。

## 1.4 请求示例

```
{
  "model": "wan2.7-image",
  "prompt": "一只穿着宇航服的猫在月球上弹吉他",
  "n": 1,
  "size": "1024x1024",
  "quality": "standard",
  "response_format": "url",
  "style": "vivid"
}
```

## 1.5 响应参数

参数名	类型	说明
<code>created</code>	<code>int64</code>	创建时间 Unix 时间戳
<code>data</code>	<code>array</code>	生成的图片数据列表
<code>data[].url</code>	<code>string</code>	图片访问 URL ( <code>response_format=url</code> )
<code>data[].b64_json</code>	<code>string</code>	Base64 编码的图片数据 ( <code>response_format=b64_</code> 时返回)
<code>data[].revised_prompt</code>	<code>string</code>	系统基于安全策略修订后的

参数名	类型	说明
metadata	json	扩展元数据（可选，部分渠

## 1.6 响应示例

```
{
  "created": 1716883200,
  "data": [
    {
      "url": "https://cdn.example.com/images/gen_xxx.png",
      "b64_json": "",
      "revised_prompt": "A cat in an astronaut suit playing guitar on the moon, digital art"
    }
  ],
  "metadata": {}
}
```

## 1.7 CURL 调用示例

```
curl -X POST "https://ai-api.mandao.com/v1/images/generations" \
  -H "Authorization: Bearer sk-your-api-key" \
  -H "Content-Type: application/json" \
  -d '{
```

```
"model": "wan2.7-image",  
"prompt": "一只穿着宇航服的猫在月球上弹吉他",  
"n": 1,  
"size": "1024x1024",  
"quality": "hd"  
}'
```

---

# 文生视频/图生视频

# 文生视频/图生视频

接口

`https://ai-api.mandao.com/v1/video/generations`

## 2.1 部分模型列表

模型名	说明
<code>wan2.7-t2v</code>	Wan2.7 文生视频模型
<code>wan2.7-i2v</code>	Wan2.7 图生视频模型

## 2.2 请求头

参数名	是否必填	说明
<code>Authorization</code>	是	认证令牌，格式 <code>Bearer sk-xxxx</code> 或 <code>Access-Token: sk-xxxx</code>
<code>Content-Type</code>	是	<code>application/json</code>

## 2.3 请求参数

参数名	类型	是否必填	默认值	功能说明
<code>model</code>	<code>string</code>	否	-	模型/风格 <code>klings-v1</code> <code>v2-master</code>
<code>prompt</code>	<code>string</code>	否	-	文本提示词 描述要生成
<code>image</code>	<code>string</code>	否	-	图片输入 (Base64) 用于图生图
<code>duration</code>	<code>float64</code>	是	-	视频时长 <code>5.0</code>
<code>width</code>	<code>int</code>	是	-	视频宽度 <code>512</code> 、 <code>1</code>
<code>height</code>	<code>int</code>	是	-	视频高度 <code>512</code> 、 <code>1</code>
<code>fps</code>	<code>int</code>	否	-	视频帧率
<code>seed</code>	<code>int</code>	否	-	随机种子 固定种子可 如 <code>20231</code>
<code>n</code>	<code>int</code>	否	<code>1</code>	生成视频数
<code>response_format</code>	<code>string</code>	否	<code>url</code>	响应格式

参数名	类型	是否必填	默认值	功能说明
<code>user</code>	<code>string</code>	否	-	用户标识 用于请求鉴权
<code>metadata</code>	<code>object</code>	否	-	厂商特定/ 自定义参数 可包含： <ul style="list-style-type: none"> <li><code>negative</code> 负向提示词</li> <li><code>style</code></li> <li><code>quality</code> 质量等级</li> <li><code>camera</code> 镜头控制参数</li> <li>其他渠道</li> </ul>

特殊说明：

- `metadata` 为透传字段，不同上游渠道（可灵、即梦、Gemini 等）支持的子字段不同，请参考对应厂商文档。

## 2.4 请求示例（文生视频）

```
{
  "model": "wan2.7-t2v",
```

```
"prompt": "宇航员站起身走了",
"duration": 5.0,
"width": 512,
"height": 512,
"fps": 30,
"seed": 20231234,
"n": 1,
"response_format": "url",
"user": "user-1234",
"metadata": {
  "negative_prompt": "模糊, 低质量",
  "style": "cinematic"
}
```

## 2.5 请求示例 (图生视频)

```
{
  "model": "wan2.7-i2v",
  "image": "https://example.com/image.jpg",
  "prompt": "A cat playing piano in the garden",
  "duration": 5.0,
  "width": 1024,
  "height": 1024,
  "fps": 30,
  "n": 1,
  "metadata": {
    "camera_control": {
```

```
"type": "simple",
"config": {
  "horizontal": 2.5,
  "vertical": 0,
  "zoom": 0
}
}
```

## 2.6 提交成功响应参数

任务提交成功后，系统会返回任务基本信息。

### 2.6.1 标准响应

参数名	类型	说明
task_id	string	任务唯一标识
status	string	任务初始状态，通常为 <code>queued</code> 或 <code>submitted</code>

### 2.6.2 提交响应示例

```
{
  "task_id": "vid_abc123def456",
```

```
"status": "queued"
}
```

## 2.7 CURL 调用示例

提交视频生成任务：

```
curl -X POST "https://ai-api.mandao.com/v1/video/generatio
ns" \
-H "Authorization: Bearer sk-your-api-key" \
-H "Content-Type: application/json" \
-d '{
  "model": "wan2.7-t2v",
  "prompt": "宇航员站起身走了",
  "duration": 5.0,
  "width": 512,
  "height": 512,
  "n": 1
}'
```

## 2.8 查询任务状态

接口

```
https://ai-api.mandao.com/v1/video/generations/{taskId}
```

请求示例

```
curl -X GET "https://ai-api.mandao.com/v1/video/generations/vid_abc123def456" \
-H "Authorization: Bearer sk-your-api-key"
```

## 2.9 响应参数

参数名	类型	说明
code	string	状态码， <code>success</code> 表示成功
message	string	消息说明
data	object	任务详情对象 (TaskDto)
data.task_id	string	任务 ID
data.status	string	任务状态： <code>queued</code> (排队中)
data.result_url	string	生成结果的 URL (成功时返回)
data.fail_reason	string	失败原因 (失败时返回)
data.progress	string	任务进度百分比或描述
data.platform	string	上游平台标识
data.quota	int	消耗额度
data.created_at	int64	创建时间戳
data.updated_at	int64	更新时间戳

参数名	类型	说明
data.submit_time	int64	提交时间戳
data.start_time	int64	开始处理时间戳
data.finish_time	int64	完成时间戳
data.action	string	任务动作类型
data.properties	any	任务属性扩展字段

### 响应示例（成功）

```
{
  "code": "success",
  "message": "",
  "data": {
    "id": 12345,
    "task_id": "vid_abc123def456",
    "status": "succeeded",
    "result_url": "https://cdn.example.com/videos/result_xx.mp4",
    "progress": "100%",
    "platform": "kling",
    "quota": 5000,
    "created_at": 1716883200,
    "updated_at": 1716883500,
    "submit_time": 1716883200,
    "start_time": 1716883210,
```

```
"finish_time": 1716883500,  
"action": "generate",  
"properties": {},  
"fail_reason": ""  
}  
}
```

## 第三方工具集成

# Claude Code 接入指南

现在你可以通过 [漫道云AI](#) 使用 Claude 系列模型。

## 系统要求

平台	要求
Windows	Windows 10 或 Windows 11
macOS	macOS 10.15 (Catalina) 或更高版本
Linux	Ubuntu 18.04+, CentOS 7+, Debian 9+

所有平台均需要：

- Node.js 18+
- 网络连接

## 1. 安装 Node.js

请确保您已安装 Node.js 18+，请参考 [Node.js 官方网站](#)。

验证安装：

```
node --version
```

```
npm --version
```

提示：建议使用 LTS（长期支持）版本以获得最佳稳定性。

## 2. 安装 Claude Code CLI

打开终端/命令提示符，执行以下命令：

```
npm install -g @anthropic-ai/claude-code
```

验证安装：

```
claude --version
```

注意：Windows 用户如遇到权限问题，请确保以管理员身份运行命令提示符。

## 3. 配置 API

### 3.1 获取 API Key

访问 [漫道云AI](#) 获取您的 API 密钥。

### 3.2 配置环境变量

重要提示：请将下方的 `ANTHROPIC_AUTH_TOKEN` 替换为您在 UModelverse 平台获取的实际 API Key！

⚠ 注意：由于部分实验性功能存在 API 兼容性问题，建议通过 `CLAUDE_CODE_DISABLE_EXPERIMENTAL_BETAS` 参数禁用这些功能以确保稳定运行。

## Windows

配置位置：`%USERPROFILE%\.claude\settings.json`

```
{
  "env": {
    "ANTHROPIC_AUTH_TOKEN": "your-umodelverse-api-key",
    "ANTHROPIC_BASE_URL": "https://ai-api.mandao.com",
    "CLAUDE_CODE_DISABLE_EXPERIMENTAL_BETAS": "1"
  }
}
```

配置参数说明：

参数	说明
<code>ANTHROPIC_MODEL</code>	指定默认使用的模型
<code>ANTHROPIC_DEFAULT_*_MODEL</code>	将 Haiku、Sonnet 等模型统一指向平台

参数	说明
<code>CLAUDE_CODE_DISABLE_EXPERIMENTAL_BETAS</code>	禁用实验性功能，避免兼容性问题

注意：配置文件更加安全且便于管理，需要重启 Claude Code 才生效。

## 4. 启动 Claude Code

配置完成后，先进入到工程目录：

```
cd your-project-folder
```

然后，运行以下命令启动：

```
claude
```

首次启动后需要先进行主题的选择等操作：

1. 选择喜欢的主题（回车）
2. 确认安全须知（回车）
3. 使用默认 Terminal 配置（回车）
4. 信任工作目录（回车）
5. 开始编程！👉

# OpenAI Codex 接入指南

现在您可以通过 [漫道云AI](#)使用 `gpt-5.5` 等模型。

## 快速入门

### 1. 安装

1. 请确保您已安装 npm，请参考 [Node.js 官方网站](#)。
2. 安装 OpenAI Codex：

```
npm install -g @openai/codex@latest
```

### 2. 配置

#### 2.1 配置模型提供商

在 Mac 或 Linux 环境下创建并配置您的 `~/.codex/config.toml` 文件：

```
model_provider = "mandao"  
model_reasoning_effort = "high"  
model = "gpt-5.5"  
  
[model_providers.mandao]  
name = "mandao"
```

```
base_url = "https://ai-api.mandao.com/v1"
wire_api = "responses"
requires_openai_auth = true
```

## 2.2 配置 API Key

创建并配置您的 `~/.codex/auth.json` 文件：

```
{
  "OPENAI_API_KEY": "your-umodelverse-api-key"
}
```

注意: 请将 `your-umodelverse-api-key` 替换为您在 UModelverse 平台获取的实际 API Key。

## 3. 使用 Codex

配置完成后，您可以直接在终端中运行 Codex：

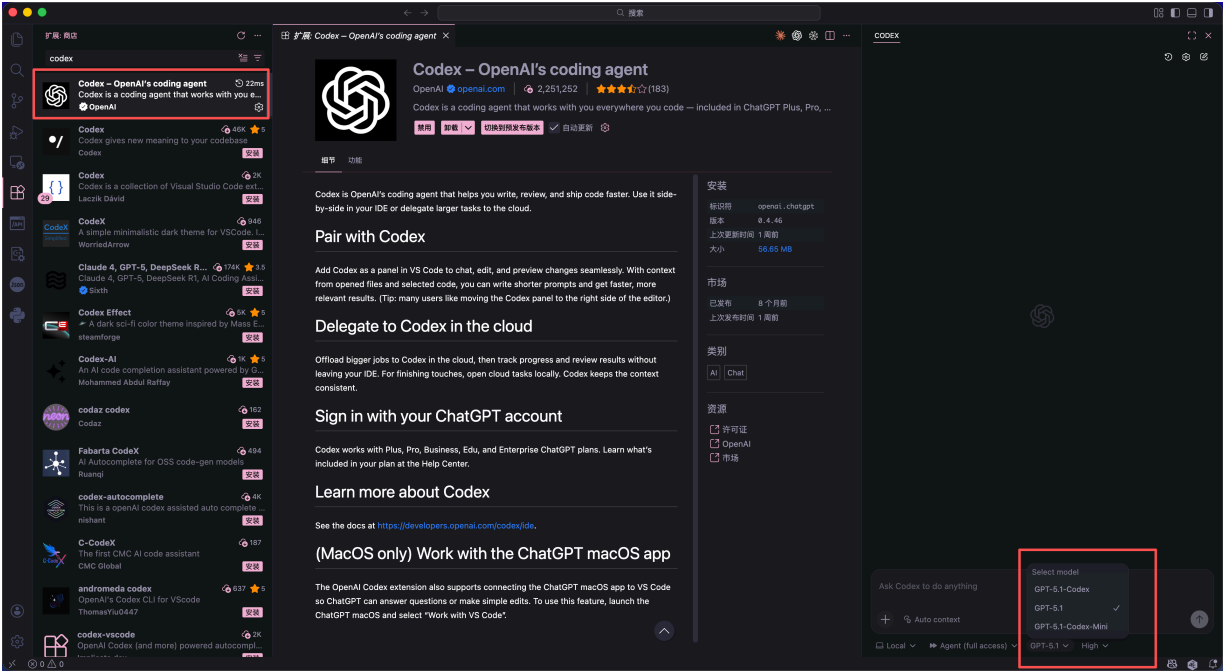
```
codex
```

您可以通过 `/model` 来切换模型，如图所示：

```
Select Model and Effort
Access legacy models by running codex -m <model_name> or in your config.toml

1. gpt-5.1-codex      Optimized for codex.
2. gpt-5.1-codex-mini Optimized for codex. Cheaper, faster, but less capable.
> 3. gpt-5.1 (current) Broad world knowledge with strong general reasoning.
```

您也可以在配置后直接安装codex VS Code 插件，并切换模型和推理强度进行使用。



## 常见问题

## Q&A

### ai-hub AQ

## 网络合规

我们采用的是 AWS 的网络，属于合规的。

## 大模型备案

目前国内都是禁止的，我们也没办法帮忙备案。

## 数据留存

我们只是类似网关，就和交换机一样，不会留存任何用户信息和数据，同时合同也会约束不会获取任何数据。

## 数据训练

同上，不留存就不会做训练。

## 传输加密与隔离

保证是标准协议接口与官方一致，基于大模型的要求，不做任何改变，加密是默认的；默认一个用户会创建一个容器组，保证完全隔离。

## 并发限制

我们是直连 API，企业账户官方没有并发限制，我们也不会做任何限制，目前接入 200 用户，暂时未出现性能瓶颈。

## TTFT

每个模型的测试，会有综合因素影响，您可以根据需要的模型自行体验测试，目前常用的都在 1s 以内。

## 模型能否配置到 Cursor 中？

可以是可以，但是有 2 个条件：

1. 用买 60 美元每月的账户
2. 配个中转，不能用原来的模型名称，要改名

普通 20 美元的包月账户你可以配置上去，但是无法编程，他禁止了 code agent，只能算是普通的问答。只有 60 这档，会把 code agent 带过去。

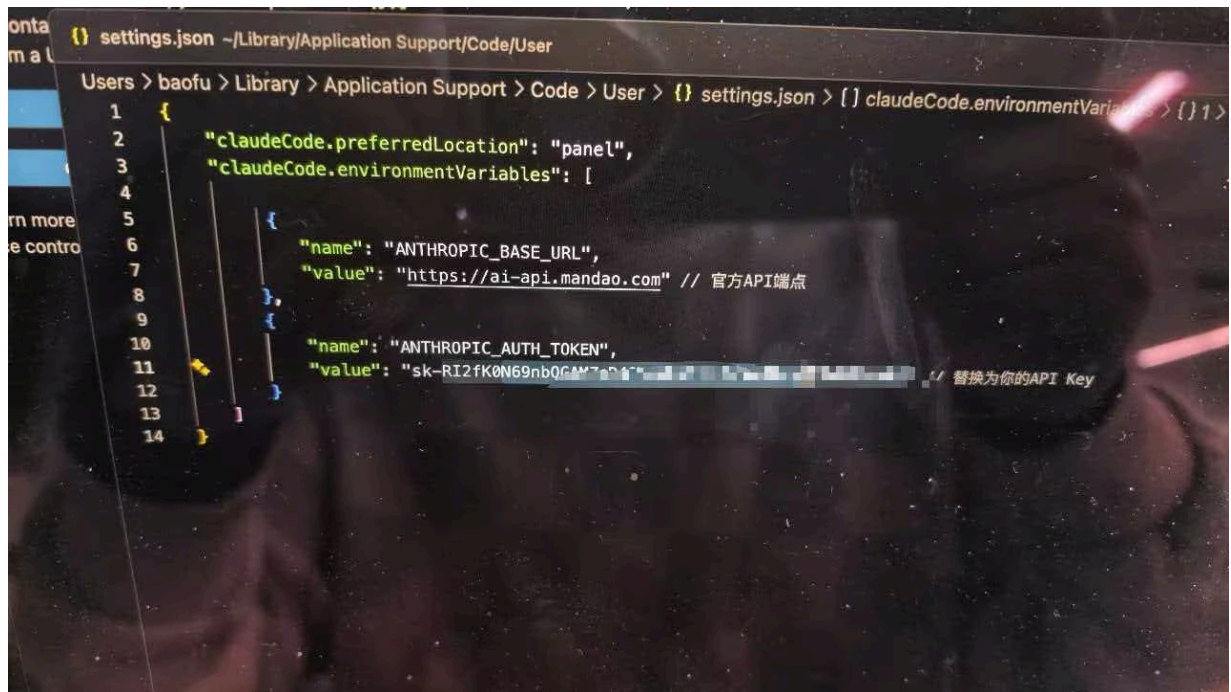
## 供应商切换与重复计费

问：如果在第一个供应商使用过程中，超时或者中断了，会不会切到新的供应商上？那如果当第一个供应商恢复了，会不会重复计算 token 量？也就是 2 个供应商都扣费的情况？如果不会这里的机制是怎样的？

答：换供应商我们对商户端也只记一次费，我们多次调用渠道是以渠道为准。一般失败也不会扣，会退款（用户无感知）。

## vs code 配置

需在 VS Code 的 settings.json 文件中（配置项 `claudeCode.environmentVariables`）设置



```
1 {
2   "claudeCode.preferredLocation": "panel",
3   "claudeCode.environmentVariables": [
4     {
5       "name": "ANTHROPIC_BASE_URL",
6       "value": "https://ai-api.mandao.com" // 官方API端点
7     },
8     {
9       "name": "ANTHROPIC_AUTH_TOKEN",
10      "value": "sk-RI2fK0N69nb0GAM7aP4t" // 替换为你的API Key
11    }
12  ]
13 }
14 }
```

## claude code 配置及切换模型



claudecode客户端调用openai协议报错There's an issue with the selected model (deepseek-v4-pro). It may not exist or you may not have access to it. Run /model to pick a different model.

先升级下claudecode 版本；

```
export CLAUDE_CODE_DISABLE_EXPERIMENTAL_BETAS=1
```

加上这个环境变量，然后重启claudecode

## 图片生成地址

<https://ai-api.mandao.com/v1/images/generations>

## 可用模型列表

```
curl -X GET 'https://ai-api.mandao.com/v1/models'
```

```
-H 'Authorization: sk-XXX'
```

```
-H 'Content-Type: application/json'
```

## trae中配置自定义模型

### 编辑模型 ✕

模型服务商 自定义配置

\* API 格式

OpenAI Chat Completions 格式 ▾

\* 自定义请求地址 ↔ 完整 URL

ℹ 请填写兼容 OpenAI API 的服务端点地址，不要以斜杠结尾。  
/chat/completions 将会被补充到你填写的地址末尾。

\* 模型 ID 多模态

\* API 密钥

> 高级配置  
包含模型系列（优化的 Prompt 和超参）、展示名称、上下文窗口等配置。

## 图片生成

<https://ai-api.mandao.com/v1/images/generations>

## 视频生成

<https://ai-api.mandao.com/v1/video/generations>

##gpt-5-codex 和 gpt-5.1-codex 系列模型说明  
这两个系列模型仅支持 /v1/response/ 接口。

## Gemini 系列模型如何开启/关闭思考过程

使用 Gemini 协议的 v1beta/models 接口，并配置以下参数。

```
“generationConfig”: {  
  “thinkingConfig”: {  
    “include_thoughts”: true,  
    “thinkingBudget”: 0  
  }  
}
```

参数说明：

include\_thoughts：设置为 true 开启思考过程输出，设置为 false 关闭

thinkingBudget：控制思考时间预算

# 错误码

## 错误码

如果请求错误，服务器返回的 JSON 文本包含以下参数。

HTTP 状态码	类型	错误码	错误信息
400	invalid_request_error	param_error	Invalid param
400	invalid_request_error	invalid_messages	Sensitive chat message
400	invalid_request_error	sensitive_check_error	Sensitive check error
400	invalid_request_error	websearch_error	Web search error
400	invalid_request_error	model_error	No permission to use the model
400	invalid_request_error	tokens_too_long	Prompt tokens too long

HTTP 状态码	类型	错误码	错误信息
401	invalid_request_error	auth_error	Validate Certificat failed
408	timeout	timeout	Request timeout, please tr again lat
429	rate_limit_error	rate_limit	Rate limi exceedec please tr again lat
500	internal_error	internal_error	Internal Server Er
500	internal_error	model_server_error	Request server Er
504	timeout	gateway_timeout_error	Gateway timeout, please tr again lat or use stream a