知识引擎原子能力

第三方大模型兼容接口（功能已迁移至TokenHub）

腾讯云

文档目录

第三方大模型兼容接口（功能已迁移至TokenHub）
腾讯云 DeepSeek OpenAI 对话接口
腾讯云 DeepSeek Anthropic 兼容接口
第三方大模型 OpenAI 兼容接口
第三方大模型 Anthropic 兼容接口
API KEY 管理

第三方大模型兼容接口（功能已迁移至 TokenHub）
腾讯云 DeepSeek OpenAI 对话接口

最近更新时间：2026-04-15 18:24:12

> **注意：**
> DeepSeek API 相关功能已转移至 TokenHub，后续请到 TokenHub 使用，此文档不再更新。

腾讯云知识引擎原子能力 DeepSeek OpenAI 对话接口兼容了 OpenAI 的接口规范，这意味着您可以直接使用 OpenAI 官方提供的 SDK 来调用。您仅需要将 base_url 和 api_key 替换成相关配置，不需要对应用做额外修改，即可无缝将您的应用切换到相应的大模型。

- base_url：`https://api.lkeap.cloud.tencent.com/v1`
- api_key：与混元大模型和其他第三方大模型的 API Key 均不共用，需在控制台 API key 页面进行创建，操作步骤请参见 API key 管理。
- 接口请求地址完整路径：`https://api.lkeap.cloud.tencent.com/v1/chat/completions`
- 调用情况可在 控制台 中查看。计费详情请参见 计费概述。

> **说明：**
> 默认单账号下的模型限制为：
> - QPM (Queries Per Minute): 15,000
> - TPM (Tokens Per Minute): 1,200,000

在线体验

如您希望在网页内直接体验 DeepSeek 模型对话，推荐您前往 腾讯云智能体开发平台，使用 DeepSeek 联网助手。

已支持的模型

DeepSeek R1-0528 模型

模型	model 参数值	参数量	最大上下文长度	最大输入长度	最大输出长度	思维链最大输出长度
DeepSeek-R1-0528	deepseek-r1-0528	671B	128k	96k	16k	32k

					（不含思维链长度） 默认4k	

DeepSeek V3-0324 模型

模型	model 参数值	参数量	最大上下文长度	最大输入长度	最大输出长度
DeepSeek-V3-0324	deepseek-v3-0324	671B	128k	128k	16k 默认4k

DeepSeek V3.1-Terminus 模型

模型	model 参数值	参数量	最大上下文长度	最大输入长度	最大输出长度	思维链最大输出长度
DeepSeek-V3.1-Terminus	deepseek-v3.1-terminus	685B	128k	96k	32k 默认4k	32k

DeepSeek-V3.2 模型

模型	model 参数值	参数量	最大上下文长度	最大输入长度	最大输出长度	思维链最大输出长度
DeepSeek-V3.2	deepseek-v3.2	685B	128k	96k	32k 默认4k	32k

说明：
model 参数值：调用模型时携带的“Model”字段，如 deepseek-v3.2。

DeepSeek-R1-0528（model 参数值为 deepseek-r1-0528）

DeepSeek-R1-0528为671B 参数模型，架构优化与训练策略升级后，相比上一版本在代码生成、长文本处理和复杂推理领域提升明显。

DeepSeek-V3-0324（model 参数值为 deepseek-v3-0324）

DeepSeek-V3-0324 为671B 参数 MoE 模型，在编程与技术能力、上下文理解与长文本处理等方面优势突出。

DeepSeek-V3.1-Terminus（model 参数值为 deepseek-v3.1-terminus）

DeepSeek-V3.1-Terminus 为685B 参数 MoE 模型，在保持模型原有能力的基础上，优化了语言一致性和 Agent 能力等问题，输出效果相比前一版本更加稳定。

DeepSeek-V3.2（model 参数值为 deepseek-v3.2）

DeepSeek-V3.2 为685B 参数 MoE 模型，其引入的稀疏注意力架构使长文本处理更高效，并在推理评测中达到 GPT-5水平。

> ⚠ **注意：**
> 默认单账号下 **DeepSeek-V3.2** 模型的限制为：
> - QPM：15,000
> - TPM：300,000

快速开始

API 使用前提：已在腾讯云控制台 API key 管理 开通知识引擎原子能力并创建 API Key。如果通过 SDK 调用，需要安装 OpenAI 。

- 如果您首次使用知识引擎原子能力，请参考 API key 管理 进行知识引擎原子能力的开通，并将示例代码中的 model 参数修改为上表中您需要调用的模型名称。
- 由于 deepseek-r1-0528 模型的思考过程可能较长，可能导致响应慢或超时，建议您优先使用**流式输出**方式调用。

安装 SDK

您需要确保已安装 Python 3.8或以上版本。
安装或更新 OpenAI Python SDK
运行以下命令：

```
pip install -U openai
```

如果运行失败，请将 pip 改为 pip3。

示例代码片段

非流式请求

Python

```
import os
from openai import OpenAI
```

```
client = OpenAI(
    # 请用知识引擎原子能力API Key将下行替换为：api_key="sk-xxx",
    api_key="LKEAP_API_KEY", # 如何获取API Key：
https://cloud.tencent.com/document/product/1772/115970
    base_url="https://api.lkeap.cloud.tencent.com/v1",
)

completion = client.chat.completions.create(
    model="deepseek-r1-0528",  # 此处以 deepseek-r1-0528 为例，可按需更换模型
名称。
    messages=[
        {'role': 'user', 'content': '请解释一下RESTful API的设计原则'}
        ]
)

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

NodeJS

```
import OpenAI from "openai";

const openai = new OpenAI(
    {
        // 请用知识引擎原子能力API Key将下行替换为：apiKey: "sk-xxx",
        apiKey: "LKEAP_API_KEY", // 如何获取API Key：
https://cloud.tencent.com/document/product/1772/115970
        baseURL: "https://api.lkeap.cloud.tencent.com/v1"
    }
);
const completion = await openai.chat.completions.create({
    model: "deepseek-r1-0528",  // 此处以 deepseek-r1-0528 为例，可按需更换
模型名称。
    messages: [
        { role: "user", content: "请解释一下RESTful API的设计原则" }
    ],
});

console.log(completion.choices[0].message.content)
```

cURL

```
curl https://api.lkeap.cloud.tencent.com/v1/chat/completions \
-H "Content-Type: application/json" \
-H "Authorization: Bearer sk-xxxxxxxxxxx" \
-d '{
  "model": "deepseek-r1-0528",
  "messages": [
        {
            "role": "user",
            "content": "请解释一下RESTful API的设计原则"
        }
    ],
  "stream": false
}'
```

多轮对话

腾讯云知识引擎原子能力 DeepSeek API 支持使用多轮对话功能，多轮对话功能可以满足如追问、采集等连续多轮对话才能完成交流的场景。服务端不记录用户请求的上下文，用户在每次请求时，需要将之前所有对话的历史拼接好之后，再传递到对话 API。

Python

```
from openai import OpenAI

client = OpenAI(
    # 请用知识引擎原子能力API Key将下行替换为：api_key="sk-xxx",
    api_key="LKEAP_API_KEY", # 如何获取API Key：
https://cloud.tencent.com/document/product/1772/115970
    base_url="https://api.lkeap.cloud.tencent.com/v1",
)

# 初始对话上下文 - 技术问答示例
messages = [
    {'role': 'user', 'content': '请解释一下RESTful API的设计原则'},
    {'role': 'assistant', 'content': 'RESTful API的核心原则包括：统一接口、无
状态、可缓存、分层系统等'},
    {'role': 'user', 'content': '能详细说明一下统一接口这个原则吗？'}
```

```
]

print("第一轮对话 - 技术概念探讨")

completion = client.chat.completions.create(
    model="deepseek-r1-0528", # 此处以 deepseek-r1-0528 为例，可按需更换模型名称。
    messages=messages
)

print(completion.choices[0].message.content)

# 更新对话上下文
messages.append({'role': 'assistant', 'content': completion.choices[0].message.content})
messages.append({'role': 'user', 'content': '这些原则在实际项目中如何应用？'})

print("第二轮对话 - 实践应用")

completion = client.chat.completions.create(
    model="deepseek-r1",
    messages=messages
)

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

NodeJS

```
import OpenAI from "openai";

const openai = new OpenAI(
    {
        //  请用知识引擎原子能力API Key将下行替换为：apiKey: "sk-xxx",
        apiKey: "LKEAP_API_KEY", // 如何获取API Key：https://cloud.tencent.com/document/product/1772/115970
        baseURL: "https://api.lkeap.cloud.tencent.com/v1"
    }
);
```

```
const completion = await openai.chat.completions.create({
    model: "deepseek-r1-0528",  // 此处以 deepseek-r1-0528 为例，可按需更换模型名称。
    messages: [
        { role: "user", content: "请解释一下RESTful API的设计原则"},
        { role: "assistant", content: "RESTful API的核心原则包括：统一接口、无状态、可缓存、分层系统等"},
        { role: "user", content: "能详细说明一下统一接口这个原则吗？" }
    ],
});

console.log(completion.choices[0].message.content)
```

cURL

```
curl https://api.lkeap.cloud.tencent.com/v1/chat/completions \
-H "Content-Type: application/json" \
-H "Authorization: Bearer sk-xxxxxxxxxxx" \
-d '{
  "model": "deepseek-r1-0528",
  "messages": [
        {
            "role": "user",
            "content": "请解释一下RESTful API的设计原则"
        },
        {
            "role": "assistant",
            "content": "RESTful API的核心原则包括：统一接口、无状态、可缓存、分层系统等"
        },
        {
            "role": "user",
            "content": "能详细说明一下统一接口这个原则吗？"
        }
    ],
  "stream": true
}'
```

流式输出

deepseek-v3-0324 和 deepseek-r1-0528 模型均支持流式输出；在输出内容比较长的场景下，为降低超时风险，推荐您使用流式输出方式。

Python

```
from openai import OpenAI
import os

# 初始化OpenAI客户端
client = OpenAI(
    # 请用知识引擎原子能力API Key将下行替换为：api_key="sk-xxx",
    api_key="LKEAP_API_KEY", # 如何获取API Key：
https://cloud.tencent.com/document/product/1772/115970
    base_url="https://api.lkeap.cloud.tencent.com/v1",
)

def main():
    reasoning_content = ""  # 思维链回答
    answer_content = ""     # 最终回答
    is_answering = False    #  是否思考中的标记符

    # 发送请求
    stream = client.chat.completions.create(
        model="deepseek-r1-0528",  # 此处以 deepseek-r1-0528 为例，可按需更
换模型名称
        messages=[
            {"role": "user", "content": "请解释一下RESTful API的设计原则"}
        ],
        stream=True
    )

    for chunk in stream:
        delta = chunk.choices[0].delta

        # 处理空内容情况
```

```
        if not getattr(delta, 'reasoning_content', None) and not
getattr(delta, 'content', None):
            continue

        # 处理开始回答的情况
        if not getattr(delta, 'reasoning_content', None) and not
is_answering:
            is_answering = True

        # 处理思维链回答
        if getattr(delta, 'reasoning_content', None):
            reasoning_content += delta.reasoning_content
        # 处理最终回答
        elif getattr(delta, 'content', None):
            print(delta.content, end='', flush=True)
            answer_content += delta.content

if __name__ == "__main__":
    try:
        main()
    except Exception as e:
        print(f"发生错误：{e}")
```

NodeJS

```
import OpenAI from "openai";

const openai = new OpenAI({
    // 请用知识引擎原子能力API Key将下行替换为：apiKey: "sk-xxx",
    apiKey: "LKEAP_API_KEY", //如何获取API Key：
https://cloud.tencent.com/document/product/1772/115970
    baseURL: "https://api.lkeap.cloud.tencent.com/v1"
});

async function main() {
    let reasoningContent = ""; // 思维链回答
    let answerContent = ""; // 最终回答
    let isAnswering = false; // 是否思考中的标记符
```

```
    const completion = await openai.chat.completions.create({
        model: "deepseek-r1-0528", // 此处以 deepseek-r1-0528 为例，可按需更换模型名称
        messages: [
            { role: 'user', content: '请解释一下RESTful API的设计原则' }
        ],
        stream: true,
    });

    for await (const chunk of completion) {
        const delta = chunk.choices[0].delta;

        // 处理空内容情况
        if (!delta.reasoning_content && !delta.content) {
            continue;
        }

        // 处理开始回答的情况
        if (!delta.reasoning_content && !isAnswering) {
            isAnswering = true;
        }

        // 处理思维链回答
        if (delta.reasoning_content) {
            reasoningContent += delta.reasoning_content;
        }
        // 处理最终内容
        else if (delta.content) {
            process.stdout.write(delta.content);
            answerContent += delta.content;
        }
    }
}

main().catch(console.error);
```

cURL

```
curl https://api.lkeap.cloud.tencent.com/v1/chat/completions \
```

```
-H "Content-Type: application/json" \
-H "Authorization: Bearer sk-xxxxxxxxxxx" \
-d '{
  "model": "deepseek-r1-0528",
  "messages": [
        {
            "role": "user",
            "content": "请解释一下RESTful API的设计原则"
        }
    ],
  "stream": true
}'
```

DeepSeek V3.1-Terminus/DeepSeek-V3.2

deepseek-v3.1-terminus 和 deepseek-v3.2 模型均支持思考模式与非思考模式，可通过参数开启或关闭思维链。

Python

```
import os
from openai import OpenAI

client = OpenAI(
    # 请用知识引擎原子能力API Key将下行替换为：api_key="sk-xxx",
    api_key="LKEAP_API_KEY", # 如何获取API Key：
https://cloud.tencent.com/document/product/1772/115970
    base_url="https://api.lkeap.cloud.tencent.com/v1",
)

completion = client.chat.completions.create(
    model="deepseek-v3.1-terminus",
    messages=[
        {
          'role': 'user',
          'content': '请解释一下RESTful API的设计原则'
        }
    ],
```

```
    extra_body={
        "thinking": {
            # "type": "disabled"  不带思维链
            # "type": "enabled"   包含思维链
            "type": "enabled"
        }
    }
)

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

NodeJS

```
import OpenAI from "openai";

const openai = new OpenAI(
    {
        // 请用知识引擎原子能力API Key将下行替换为：apiKey: "sk-xxx",
        apiKey: "LKEAP_API_KEY", // 如何获取API Key:
https://cloud.tencent.com/document/product/1772/115970
        baseURL: "https://api.lkeap.cloud.tencent.com/v1"
    }
);
const completion = await openai.chat.completions.create({
    model: "deepseek-v3.1-terminus",
    messages: [
        { role: "user", content: "请解释一下RESTful API的设计原则" }
    ],
    thinking: {type: "enabled"},
});

console.log(completion.choices[0].message.content)
```

cURL

```
curl https://api.lkeap.cloud.tencent.com/v1/chat/completions \
-H "Content-Type: application/json" \
-H "Authorization: Bearer sk-xxxxxxxxxx" \
```

```
-d '{
  "model": "deepseek-v3.1-terminus",
  "messages": [
        {
            "role": "user",
            "content": "请解释一下RESTful API的设计原则"
        }
    ],
  "thinking": {"type": "enabled"},
  "stream": false
}'
```

注意事项

稳定性

若执行后出现“concurrency exceeded”的响应，则表明您的请求遭遇了限流。这通常是由于服务器资源暂时不足所致。建议您稍后再试，届时服务器负载可能已得到缓解。

DeepSeek-R1-0528

不推荐设置 System Prompt。

参数配置说明	具体参数和功能
不支持设置的功能	Function Calling、对话前缀续写、上下文硬盘缓存
不支持的参数	logprobs、top_logprobs、stop
支持的参数	top_p、temperature、max_tokens、presence_penalty、frequency_penalty、json_object、json_schema
参数默认值	temperature：0.6（取值范围是[0:2]），top_p：0.6（取值范围是(0:1]）

DeepSeek-V3-0324

参数配置说明	具体参数和功能
不支持设置的功能	对话前缀续写、上下文硬盘缓存
不支持的参数	logprobs、top_logprobs
支持的参数	top_p、temperature、max_tokens、presence_penalty、frequency_penalty、stop、Function Calling、json_object

参数默认值	temperature：0.6（取值范围是[0:2]），top_p：0.6（取值范围是(0:1]）

DeepSeek-V3.1-Terminus

参数配置说明	具体参数和功能
不支持设置的功能	对话前缀续写、上下文硬盘缓存
不支持的参数	logprobs、top_logprobs
支持的参数	top_p、temperature、max_tokens、Function Calling、presence_penalty、frequency_penalty、stop、thinking、json_object
参数默认值	temperature：0.6（取值范围是[0:2]），top_p：0.6（取值范围是(0:1]）

DeepSeek-V3.2

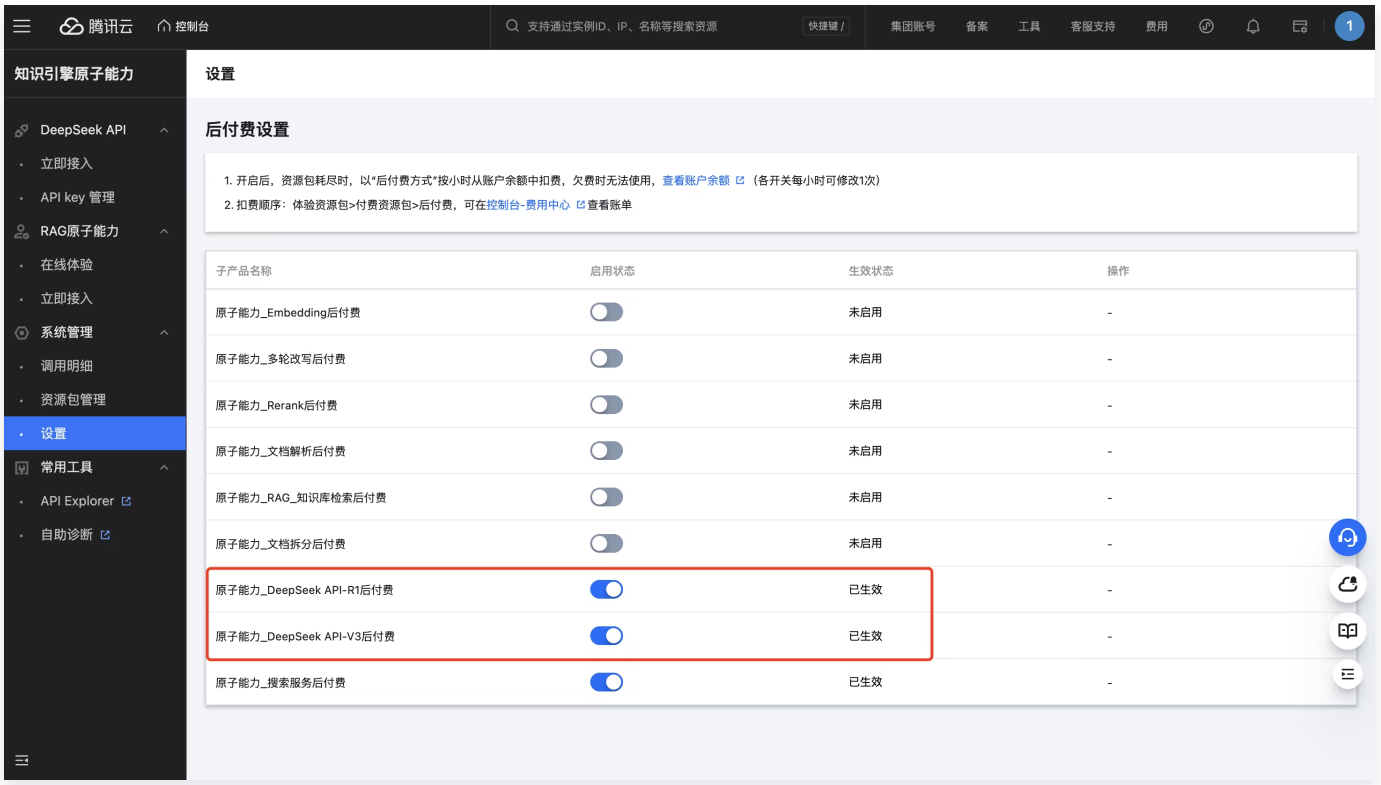
参数配置说明	具体参数和功能
不支持设置的功能	对话前缀续写、上下文硬盘缓存
不支持的参数	logprobs、top_logprobs
支持的参数	top_p、temperature、max_tokens、presence_penalty、frequency_penalty、stop、thinking、json_object、json_schema、Function Calling
参数默认值	temperature：0.6（取值范围是[0:2]），top_p：0.6（取值范围是(0:1]）

敬请关注后续动态。

联网搜索

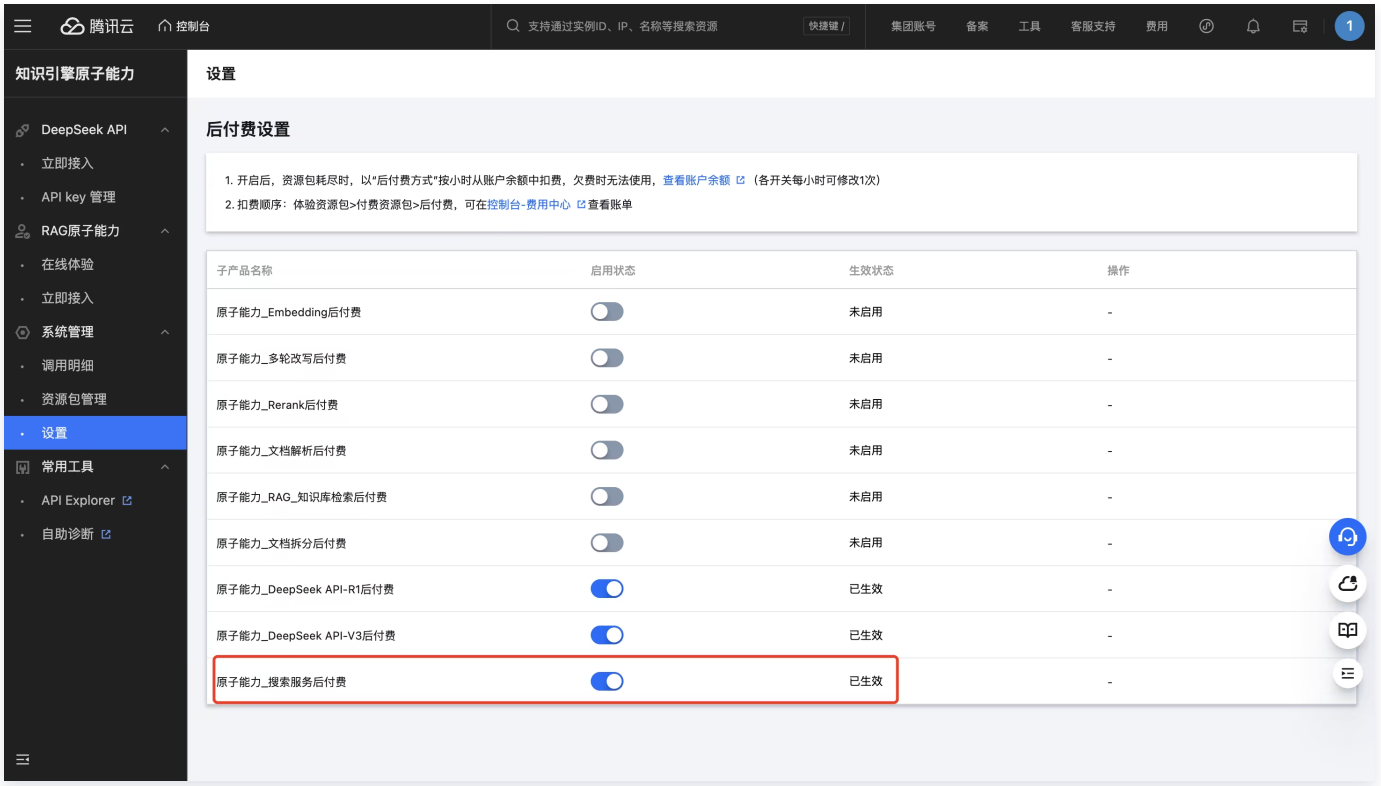
本接口支持联网搜索，前提是开启 DeepSeek API 的服务。若需要使用相关联网搜索功能，首先需要开启联网搜索后付费开关或者购买搜索服务的资源包，然后通过传入 enable_search 参数体验 DeepSeek 的联网搜索能力（仅支持流式输出场景）。

1. 首先开启 DeepSeek API 的服务，进入 原子能力控制台 开启 DeepSeek 的后付费开关。



2. 进入 原子能力控制台 打开联网搜索后付费开关或者购买搜索服务的资源包。

开启搜索服务的后付费开关：



购买搜索服务的资源包：

3. 调用 DeepSeek API 接口的时候，增加 enable_search 参数体验带联网搜索能力的 DeepSeek API。

输入示例：

python

```
import os
from openai import OpenAI

client = OpenAI(
    # 请用知识引擎原子能力API Key将下行替换为：api_key="sk-xxx",
    api_key="LKEAP_API_KEY", # 如何获取API Key：
https://cloud.tencent.com/document/product/1772/115970
    base_url="https://api.lkeap.cloud.tencent.com/v1",
)

completion = client.chat.completions.create(
    model="deepseek-v3-0324",  # 此处以deepseek-v3-0324为例，可按需更换模型名
称。
    messages=[
        {
            "role": "user",
            "content": "深圳今日天气"
```

```
        }
    ],
    extra_body={
        "enable_search": True,   # 开启联网搜索
    }
)

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

NodeJS

```
import OpenAI from "openai";

const openai = new OpenAI(
    {
        // 请用知识引擎原子能力API Key将下行替换为：apiKey: "sk-xxx",
        apiKey: "LKEAP_API_KEY", // 如何获取API Key：
https://cloud.tencent.com/document/product/1772/115970
        baseURL: "https://api.lkeap.cloud.tencent.com/v1"
    }
);
const completion = await openai.chat.completions.create({
    model: "deepseek-v3-0324",  // 此处以deepseek-v3-0324为例，可按需更换模型
名称。
    messages: [
        { role: "user", content: "深圳今日天气" }
    ],
    enable_search: true,  // 开启联网搜索
});

console.log(completion.choices[0].message.content)
```

cURL

```
curl https://api.lkeap.cloud.tencent.com/v1/chat/completions \
-H "Content-Type: application/json" \
-H "Authorization: Bearer sk-xxxxxxxxxx" \
```

```
-d '{
  "model": "deepseek-r1-0528",
  "messages": [
        {
            "role": "user",
            "content": "深圳今日天气"
        }
    ],
  "enable_search": true,
  "stream": false
}'
```

输出示例：

{'id': '0c9eba283a56d0add6084e24263056b7', 'choices': [{'finish_reason': 'stop', 'index': 0, 'logprobs': None, 'message': {'content': '根据深圳市气象局(台)的预报，2025年4月29日深圳的天气情况如下：\n\n今天深圳多云间阴天，早晚有轻雾，部分时间可见阳光，偏东风最大阵风6-7级，气温22-29℃。相对湿度在40%-70%之间。\n\n具体来说，今天的气温预计在22℃到29℃之间，风向为偏东风，风力为2-3级，沿海、高地和海区阵风可达5-6级。此外，今天日出时间为05:52，日落时间为18:49。\n\n需要注意的是，今天早晚有轻雾，部分时间可见阳光，建议关注清劲偏东风的影响。\n\n总结：今天深圳的天气以多云间阴天为主，气温适中，早晚有轻雾，风力较大，适合外出但需注意防风。', 'refusal': None, 'role': 'assistant', 'annotations': None, 'audio': None, 'function_call': None, 'tool_calls': None, 'search_results': [{'index': 1, 'url': 'http://weather.sz.gov.cn/?COLLCC=2354144265&', 'name': '深圳市气象局(台)', 'snippet': '深圳市气象局(台) 今日预报 实况 29.4℃ 12-20时 多云;气温27-30℃;东风2-3级,沿海、高地和海区阵风5-6级;相对湿度40%-70%。 05:52 日出 18:49 日落 6天 距立夏剩 06:40 月出 20:41 月落 4月29日12时56分 深圳福田国家基本气象站 东南偏东风 小于三级 相对湿度 0mm 24小时降雨量 十天预报 逐时预报 展开 【天气提示】 预计29日多云间阴天,早晚有轻雾,部分时间可见阳光,偏东风最大阵风6-7级,气温22-29℃;30日多云到阴天,局地有短时阵雨,早晚清凉;展望五一假期,初期和末期有(雷)阵雨,局地雨势较大,中期以多云为主,间中有短时阵雨,午间较热。建议关注29日清劲偏东风的影响。 明天 4-30 周四 5-1 周五 5-2 周六 5-3 周日 5-4 周一 5-5 周二 5-6 周三 5-7 周四 5-8 13时 多云 14时 多云 15时 多云 16时 多云 17时 多云 18时 多云 19时 多云 20时 多云 21时 少云 热 点 推 荐 工作动态 公告公示 重要资讯 公开目录 媒体聚焦 庆祝中华全国总工会成立100周年暨全国劳动模范和先进工作者表彰大会隆重举行 习近平发表重要讲话 04/29 广交会上看外贸新动能 04/29 新,高质量发展看动能 04/29 一季度规模以上工业企业利润由降转增 04/29 暴涨96%!一季度"中国游 中国购"持续升温 04/29 李强主持召开国务院常务会议 部署开展美丽河湖保护与建设行动 研究进一步加强困境儿童福利保障有关举措 讨论《中华人民共和国医疗保障法(草案)》 决

```
定核准浙江三门三期工程等核电项目 04/28 经济日报:消费市场保持升温势头|中国经济新看点
04/27 规范涉企执法,怎样防止问题反弹、提振企业信心?', 'icon': '', 'site':
'weather.sz.gov.cn', 'published_time': 1745856000}]}}], 'created':
1745929940, 'model': 'deepseek-v3-aisearch', 'object':
'chat.completion', 'service_tier': None, 'system_fingerprint': None,
'usage': {'completion_tokens': 189, 'prompt_tokens': 3366,
'total_tokens': 3555, 'completion_tokens_details': None,
'prompt_tokens_details': None}}
```

深度思考

只要调用 deepseek-r1-0528 模型即代表开启深度思考（深度思考过程通过 reasoning_content 返回）。

查看更多完整示例

Python
NodeJS
Go

错误码

错误码	错误信息	说明
20024	invalid params	参数信息有误，请查阅 API 文档 检查入参。
20031	not enough quota	您的账号目前没有可用资源。为了继续使用，请先 开通 并完成付费。
20033	invalid model	模型名称错误，请检查模型名称。
20034	concurrency rate limit exceeded	并发限流错误。这通常是由于服务器资源暂时不足所致。建议您稍后再试，届时服务器负载可能已得到缓解。
20052	concurrency exceeded	模型服务负载过高限流错误，请稍后重试。
20057	model engine error	模型引擎错误，请您稍后重试，或者联系平台技术同学处理。
20059	input length too long	输入长度超过上下文长度，请减小输入内容的长度。
20062	internet search not enough	联网搜索无可用资源，请查阅 API 文档，开通联网搜索服务。

	quota	
20072	tpm rate limit exceeded	TPM 限流错误。

错误示例

```
{"error":{"message":"not enough
quota","type":"runtime_error","param":null,"code":"20031"}}
```

安全审查示例

finish_reason = **content_filter** 表示输出内容触发了安全审核机制。这通常发生在系统检测到某些输入或输出内容可能包含敏感信息或不适当的语言，因此自动启动了审核流程以确保内容的安全性和适宜性。在这种情况下，系统会对相关内容进行仔细审查，以防止不当信息的传播。

非流式输出示例：

```
{"id":"26a58a8ab6e7712937ad542436b4b97a","object":"chat.completion","cre
ated":1740379897,"model":"deepseek-r1-0528","choices":
[{"index":0,"message":{"role":"assistant","content":"你好，我无法给到相关内
容。"},"finish_reason":"content_filter"}],"usage":
{"prompt_tokens":0,"completion_tokens":0,"total_tokens":0}}
```

流式输出示例：

```
data:
{"id":"d2d486bfdb31b1b6f55c8b5cbeb492d3","object":"chat.completion.chunk
","created":1740379627,"model":"deepseek-r1-0528","choices":
[{"index":0,"delta":{"role":"assistant","content":"你好，我无法给到相关内
容。"},"finish_reason":"content_filter"}],"usage":
{"prompt_tokens":0,"completion_tokens":0,"total_tokens":0}}
```

腾讯云 DeepSeek Anthropic 兼容接口

最近更新时间：2026-04-15 18:24:12

> **注意：**
> DeepSeek API 相关功能已转移至 TokenHub，后续请到 TokenHub 使用，此文档不再更新。

腾讯云知识引擎原子能力 DeepSeek Anthropic 接口兼容了 Anthropic 的接口规范，您仅需要将 base_url 和 api_key 替换成相关配置，不需要对应用做额外修改，即可将 DeepSeek 的能力，接入到 Anthropic API 生态中。

- base_url：`https://api.lkeap.cloud.tencent.com/anthropic`。
- api_key：需在控制台 API key 页面进行创建，操作步骤请参见 API key 管理。
- 接口请求地址完整路径：`https://api.lkeap.cloud.tencent.com/anthropic/v1/messages`。
- 调用情况可在 控制台 中查看。计费详情请参见 计费概述。

> **说明：**
> 默认单账号下的模型限制为：
> - QPM (Queries Per Minute): 15,000
> - TPM (Tokens Per Minute): 1,200,000

已支持的模型

DeepSeek V3.1-Terminus 模型

模型	model 参数值	参数量	最大上下文长度	最大输入长度	最大输出长度	思维链最大输出长度
DeepSeek-V3.1-Terminus	deepseek-v3.1-terminus	685B	128k	96k	32k 默认4k	32k

DeepSeek-V3.2 模型

模型	model 参数值	参数量	最大上下文长度	最大输入长度	最大输出长度	思维链最大输出长度
DeepSeek-V3.2	deepseek-v3.2	685B	128k	96k	32k 默认4k	32k

> **说明：**
> model 参数值：调用模型时携带的“Model”字段，例如 deepseek-v3.1-terminus。

DeepSeek-V3.1-Terminus（model 参数值为 deepseek-v3.1-terminus）

DeepSeek-V3.1-Terminus 为685B 参数 MoE 模型，在保持模型原有能力的基础上，优化了语言一致性和 Agent 能力等问题，输出效果相比前一版本更加稳定。

DeepSeek-V3.2（model 参数值为 deepseek-v3.2）

DeepSeek-V3.2 为685B 参数 MoE 模型，其引入的稀疏注意力架构使长文本处理更高效，并在推理评测中达到 GPT-5水平。

快速开始

- API 使用前提：已在腾讯云控制台 API key 管理 开通知识引擎原子能力并创建 API Key。
- 如果您首次使用知识引擎原子能力，请参考 API key 管理 进行知识引擎原子能力的开通，并将示例代码中的 model 参数修改为上表中您需要调用的模型名称。

将 DeepSeek 模型接入 Claude Code

安装 Claude Code

安装或更新 Anthropic Claude Code，运行以下命令：

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

配置环境变量

```
export ANTHROPIC_BASE_URL=https://api.lkeap.cloud.tencent.com/anthropic
export ANTHROPIC_AUTH_TOKEN=${API_KEY}
export API_TIMEOUT_MS=600000
export ANTHROPIC_MODEL=${MODEL_NAME}
export CLAUDE_CODE_DISABLE_NONESSENTIAL_TRAFFIC=1
```

Running Environment

Operating System: Ubuntu 24.04.3 LTS / x86_64

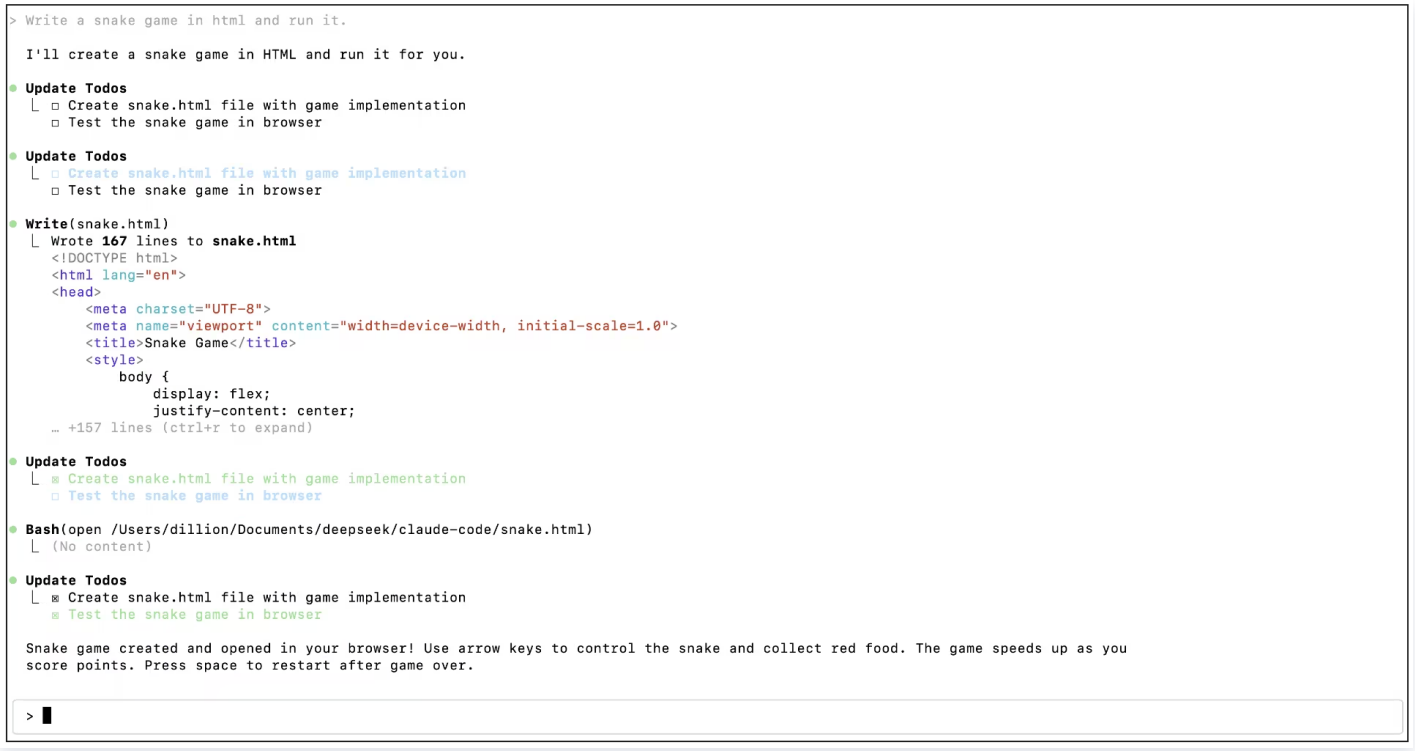
Runtime Version: GNU bash, version 5.2.21(1)-release (x86_64-pc-linux-gnu)

> **注意：**
> 设置 `API_TIMEOUT_MS` 是为了防止输出过长，触发 Claude Code 客户端超时，这里设置的超时时间为 10 分钟，使用者可自行设置。

执行 claude 命令

进入项目目录，执行 claude 命令，即可开始使用。

```
cd my-project
claude
```



通过 Anthropic API 调用 DeepSeek 模型

安装 SDK

安装或更新 Anthropic Python SDK，运行以下命令：

```
pip install anthropic
```

示例代码片段

Python

```python
import anthropic

client = anthropic.Anthropic(
    api_key=os.getenv("API_KEY"),
    base_url="https://api.lkeap.cloud.tencent.com/anthropic",
)

message = client.messages.create(
    model="deepseek-v3.1-terminus",
    max_tokens=1000,
    system="You are a helpful assistant.",
    messages=[
        {
            "role": "user",
            "content": [
                {
                    "type": "text",
                    "text": "Hi, how are you?"
                }
            ]
        }
    ]
)
print(message.content)
```

cURL

```bash
curl https://api.lkeap.cloud.tencent.com/anthropic/v1/messages \
-H "Content-Type: application/json" \
-H "x-api-key: sk-xxxxxxx" \
-d '{
    "model": "deepseek-v3.1-terminus",
    "max_tokens": 1000,
    "stream": true,
    "system": [
        {
            "type": "text",
```

```
            "text": "You are a helpful assistant."
        }
    ],
    "messages": [
        {
            "role": "user",
            "content": [
                {
                    "type": "text",
                    "text": "Hi, how are you?"
                }
            ]
        }
    ]
}'
```

Anthropic API 兼容性详情

HTTP Headers

字段	支持状态	说明
`anthropic-beta`	忽略	不处理此头部
`anthropic-version`	忽略	不处理此头部
`x-api-key`	完全支持	用于身份验证

基础字段

字段	支持状态	说明
`model`	支持	使用 DeepSeek 模型替代
`max_tokens`	完全支持	最大输出令牌数
`container`	忽略	不处理此字段
`mcp_servers`	忽略	不处理此字段
`metadata`	忽略	不处理此字段

service_tier	忽略	不处理此字段
stop_sequences	完全支持	停止序列
stream	完全支持	流式响应
system	完全支持	系统消息
temperature	完全支持	温度参数 (0.0-2.0)
thinking	忽略	不处理此字段
top_k	忽略	不处理此字段
top_p	完全支持	Top-p 采样

工具支持

tools

字段	支持状态	说明
name	完全支持	工具名称
input_schema	完全支持	输入参数模式
description	完全支持	工具描述
cache_control	忽略	不处理此字段
tool_choice	字符串格式	完全支持
tool_choice	对象格式	完全支持
tool_choice.disable_parallel_tool_use	忽略	不处理此字段

tool_choice

字段	支持状态
none	完全支持
auto	完全支持

`any`	完全支持
`tool`	完全支持
`disable_parallel_tool_use`	忽略

消息字段支持

字段类型	变体	子字段	支持状态
`content`	string	-	完全支持
`content`	array, type="text"	text	完全支持
`content`	array, type="text"	cache_control	忽略
`content`	array, type="text"	citations	忽略
`content`	array, type="image"	-	不支持
`content`	array, type="document"	-	不支持
`content`	array, type="search_result"	-	不支持
`content`	array, type="thinking"	-	忽略
`content`	array, type="redacted_thinking"	-	不支持
`content`	array, type="tool_use"	id	完全支持
`content`	array, type="tool_use"	input	完全支持
`content`	array, type="tool_use"	name	完全支持
`content`	array, type="tool_use"	cache_control	忽略
`content`	array, type="tool_result"	tool_use_id	完全支持
`content`	array, type="tool_result"	content	完全支持
`content`	array, type="tool_result"	cache_control	忽略
`content`	array, type="tool_result"	is_error	忽略

⚠ 注意：

1. **忽略的字段**：某些 Anthropic 特有的字段会被忽略，但不会报错。
2. **工具并行调用**：`disable_parallel_tool_use` 参数被忽略。
3. **缓存控制**：所有 `cache_control` 相关字段都被忽略。

第三方大模型 OpenAI 兼容接口

最近更新时间：2026-04-15 18:24:12

> **注意：**
> DeepSeek API 相关功能已转移至 TokenHub，后续请到 TokenHub 使用，此文档不再更新。

腾讯云第三方大模型 OpenAI 对话接口兼容了 OpenAI 的接口规范，这意味着您可以直接使用 OpenAI 官方提供的 SDK 来调用。您仅需要将 base_url 和 api_key 替换成相关配置，不需要对应用做额外修改，即可无缝将您的应用切换到相应的大模型。

- base_url：`https://api.lkeap.cloud.tencent.com/v3`
- api_key：**与 DeepSeek 、混元大模型的 API key 均不共用**，需在控制台 API key 页面进行创建。
- 接口请求地址完整路径：`https://api.lkeap.cloud.tencent.com/v3/chat/completions`
- 调用情况可在 控制台 中查看。计费详情请参见 计费概述 。

> **说明：**
> 默认单账号下的模型限制为：
> - QPM (Queries Per Minute): 60
> - TPM (Tokens Per Minute): 1,000,000

已支持的模型

GLM 系列模型

模型	model 参数值	最大上下文长度	最大输出长度
GLM-5	glm-5	200k	128k

MiniMax 系列模型

模型	model 参数值	最大上下文长度	最大输出长度
MiniMax M2.5	minimax-m2.5	200k	192k

Kimi 系列模型

模型	model 参数值	最大上下文长度	最大输出长度
Kimi K2.5	kimi-k2.5	256k	256k

Kimi K2-0905-preview	kimi-k2-0905-preview	256k	256k
Kimi K2-turbo-preview	kimi-k2-turbo-preview	256k	256k
Kimi K2-thinking-turbo	kimi-k2-thinking-turbo	256k	256k

说明：

- model 参数值：调用模型时携带的“Model”字段，例如 glm-5。
- Kimi K2.5 当前仅支持文本输入。

快速开始

- API 使用前提：已在腾讯云控制台 API key 管理 开通腾讯云大模型 API 能力并创建 API key。如果通过 SDK 调用，需要安装 OpenAI 。
- 如果您首次使用大模型 API 能力，请参考 API key 管理 进行大模型 API 能力的开通，并将示例代码中的 model 参数修改为上表中您需要调用的模型名称。

安装 SDK

您需要确保已安装 Python 3.8或以上版本。

安装或更新 OpenAI Python SDK

运行以下命令：

```
pip install -U openai
```

如果运行失败，请将 pip 改为 pip3。

示例代码片段

流式输出

第三方模型均支持流式输出；在输出内容比较长的场景下，为降低超时风险，推荐您使用流式输出方式。

Python

```
from openai import OpenAI
import os
```

```
# 初始化OpenAI客户端
client = OpenAI(
    # 请用大模型API能力API Key将下行替换为：api_key="sk-xxx",
    api_key="API_KEY",
    base_url="https://api.lkeap.cloud.tencent.com/v3",
)

def main():
    reasoning_content = ""  # 思维链回答
    answer_content = ""     # 最终回答
    is_answering = False    #  是否思考中的标记符

    # 发送请求
    stream = client.chat.completions.create(
        model="glm-5",  # 此处以 glm-5 为例，可按需更换模型名称
        messages=[
            {"role": "user", "content": "请解释一下RESTful API的设计原则"}
        ],
        stream=True
    )

    for chunk in stream:
        delta = chunk.choices[0].delta

        # 处理空内容情况
        if not getattr(delta, 'reasoning_content', None) and not
getattr(delta, 'content', None):
            continue

        # 处理开始回答的情况
        if not getattr(delta, 'reasoning_content', None) and not
is_answering:
            is_answering = True

        # 处理思维链回答
        if getattr(delta, 'reasoning_content', None):
            reasoning_content += delta.reasoning_content
        # 处理最终回答
        elif getattr(delta, 'content', None):
            print(delta.content, end='', flush=True)
```

```
            answer_content += delta.content

if __name__ == "__main__":
    try:
        main()
    except Exception as e:
        print(f"发生错误：{e}")
```

NodeJS

```
import OpenAI from "openai";

const openai = new OpenAI({
    // 请用腾讯云大模型能力API Key将下行替换为：apiKey: "sk-xxx",
    apiKey: "API_KEY",
    baseURL: "https://api.lkeap.cloud.tencent.com/v3"
});

async function main() {
    let reasoningContent = ""; // 思维链回答
    let answerContent = ""; // 最终回答
    let isAnswering = false; // 是否思考中的标记符

    const completion = await openai.chat.completions.create({
        model: "glm-5", // 此处以 glm-5 为例，可按需更换模型名称
        messages: [
            { role: 'user', content: '请解释一下RESTful API的设计原则' }
        ],
        stream: true,
    });

    for await (const chunk of completion) {
        const delta = chunk.choices[0].delta;

        // 处理空内容情况
        if (!delta.reasoning_content && !delta.content) {
            continue;
        }
```

```
        // 处理开始回答的情况
        if (!delta.reasoning_content && !isAnswering) {
            isAnswering = true;
        }

        // 处理思维链回答
        if (delta.reasoning_content) {
            reasoningContent += delta.reasoning_content;
        }
        // 处理最终内容
        else if (delta.content) {
            process.stdout.write(delta.content);
            answerContent += delta.content;
        }
    }
}

main().catch(console.error);
```

cURL

```
curl https://api.lkeap.cloud.tencent.com/v3/chat/completions \
-H "Content-Type: application/json" \
-H "Authorization: Bearer sk-xxxxxxxxxxx" \
-d '{
  "model": "glm-5",
  "messages": [
        {
            "role": "user",
            "content": "请解释一下RESTful API的设计原则"
        }
    ],
  "stream": true
}'
```

注意事项

稳定性

若执行后出现“concurrency exceeded”的响应，则表明您的请求触发限流。这通常是由于服务器资源暂时不足所致。建议您稍后再试，届时服务器负载可能已得到缓解。

GLM-5

支持通过 thinking 参数控制思考模式，默认开启思考。

参数配置说明	具体参数和功能
不支持设置的功能	对话前缀续写、上下文硬盘缓存
支持的参数	top_p、temperature、max_tokens、presence_penalty、frequency_penalty、stop、thinking、json_object、Function Calling
参数默认值	temperature：0.6（取值范围是(0:1)），top_p：0.95（取值范围是(0:1)）

MiniMax M2.5

不支持通过 thinking 参数控制思考模式。

参数配置说明	具体参数和功能
不支持设置的功能	对话前缀续写、上下文硬盘缓存
不支持的参数	json_object、json_schema
支持的参数	top_p、temperature、max_tokens、presence_penalty、frequency_penalty、stop、Function Calling
参数默认值	temperature：1.0（取值范围是(0:1]），top_p：0.95（取值范围是(0:1]）

Kimi K2.5

- 支持通过 thinking 参数控制思考模式，默认开启。
- 在思考模式下，不支持强制调用某个工具，tool_choice 仅支持设置为 auto（默认值）和 none。

参数配置说明	具体参数和功能
不支持设置的功能	对话前缀续写、上下文硬盘缓存
不支持的参数	top_p、temperature、presence_penalty、frequency_penalty
支持的参数	max_tokens、stop、thinking、json_object、json_schema、Function Calling

参数默认值	temperature：1.0，top_p：0.95，presence_penalty：0.0，frequency_penalty：0.0

Kimi K2-0905-preview

参数配置说明	具体参数和功能
不支持设置的功能	对话前缀续写、上下文硬盘缓存
支持的参数	top_p、temperature、max_tokens、presence_penalty、frequency_penalty、stop、json_object、json_schema、Function Calling
参数默认值	temperature：0.6（取值范围是[0:1]），top_p：1.0（取值范围是(0:1]）

Kimi K2-turbo-preview

参数配置说明	具体参数和功能
不支持设置的功能	对话前缀续写、上下文硬盘缓存
支持的参数	top_p、temperature、max_tokens、presence_penalty、frequency_penalty、stop、json_object、json_schema、Function Calling
参数默认值	temperature：0.6（取值范围是[0:1]），top_p：1.0（取值范围是(0:1]）

Kimi K2-thinking-turbo

参数配置说明	具体参数和功能
不支持设置的功能	对话前缀续写、上下文硬盘缓存
支持的参数	top_p、temperature、max_tokens、presence_penalty、frequency_penalty、stop、json_object、json_schema、Function Calling
参数默认值	temperature：1.0（取值范围是[0:1]），top_p：1.0（取值范围是(0:1]）

敬请关注后续动态。

查看更多完整示例

Python
NodeJS
Go

错误码

错误码	错误信息	说明
20024	invalid params	参数信息有误，请查阅 API 文档 检查入参。
20031	not enough quota	您的账号目前没有可用资源。为了继续使用，请先 开通 并完成付费。
20033	invalid model	模型名称错误，请检查模型名称。
20034	concurrency rate limit exceeded	并发限流错误。这通常是由于服务器资源暂时不足所致。建议您稍后再试，届时服务器负载可能已得到缓解。
20052	concurrency exceeded	模型服务负载过高限流错误，请稍后重试。
20057	model engine error	模型引擎错误，请您稍后重试，或者联系平台技术同学处理。
20059	input length too long	输入长度超过上下文长度，请减小输入内容的长度。
20072	tpm rate limit exceeded	TPM 限流错误。

错误示例

```
{"error":{"message":"not enough
quota","type":"runtime_error","param":null,"code":"20031"}}
```

安全审查示例

finish_reason = **content_filter** 表示输出内容触发了安全审核机制。这通常发生在系统检测到某些输入或输出内容可能包含敏感信息或不适当的语言，因此自动启动了审核流程以确保内容的安全性和适宜性。在这种情况下，系统会对相关内容进行仔细审查，以防止不当信息的传播。

非流式输出示例：

```
{"id":"26a58a8ab6e7712937ad542436b4b97a","object":"chat.completion","cre
ated":1740379897,"model":"glm-5","choices":[{"index":0,"message":
{"role":"assistant","content":"你好，我无法给到相关内
```

```
容。"},"finish_reason":"content_filter"}],"usage":
{"prompt_tokens":0,"completion_tokens":0,"total_tokens":0}}
```

流式输出示例：

```
data:
{"id":"d2d486bfdb31b1b6f55c8b5cbeb492d3","object":"chat.completion.chunk
","created":1740379627,"model":"glm-5","choices":[{"index":0,"delta":
{"role":"assistant","content":"你好，我无法给到相关内
容。"},"finish_reason":"content_filter"}],"usage":
{"prompt_tokens":0,"completion_tokens":0,"total_tokens":0}}
```

第三方大模型 Anthropic 兼容接口

最近更新时间：2026-04-15 18:24:12

> **注意：**
> DeepSeek API 相关功能已转移至 TokenHub，后续请到 TokenHub 使用，此文档不再更新。

腾讯云知识引擎原子能力 Anthropic 接口兼容了 Anthropic 的接口规范，您仅需要将 base_url 和 api_key 替换成相关配置，不需要对应用做额外修改，即可无缝将您的应用接入到 Anthropic API 生态中。

- base_url：`https://api.lkeap.cloud.tencent.com/api/anthropic`。
- api_key：**与 DeepSeek 、混元大模型的 API key 均不共用**，需在控制台 API key 页面进行创建。
- 接口请求地址完整路径：`https://api.lkeap.cloud.tencent.com/api/anthropic/v1/messages`。
- 调用情况可在 控制台 中查看。计费详情请参见 计费概述。

> **说明：**
> 默认单账号下的模型限制为：
> - QPM (Queries Per Minute): 60
> - TPM (Tokens Per Minute): 1,000,000

已支持的模型

GLM 系列模型

模型	model 参数值	最大上下文长度	最大输出长度
GLM-5	glm-5	200k	128k

MiniMax 系列模型

模型	model 参数值	最大上下文长度	最大输出长度
MiniMax M2.5	minimax-m2.5	200k	192k

Kimi 系列模型

模型	model 参数值	最大上下文长度	最大输出长度
Kimi K2.5	kimi-k2.5	256k	256k

快速开始

- API 使用前提：已在腾讯云控制台 API key 管理 开通腾讯云大模型 API 能力并创建 API key。
- 如果您首次使用知识引擎原子能力，请参考 API key 管理 进行知识引擎原子能力的开通，并将示例代码中的 model 参数修改为上表中您需要调用的模型名称。

将模型接入 Claude Code

安装 Claude Code

安装或更新 Anthropic Claude Code，运行以下命令：

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

配置环境变量

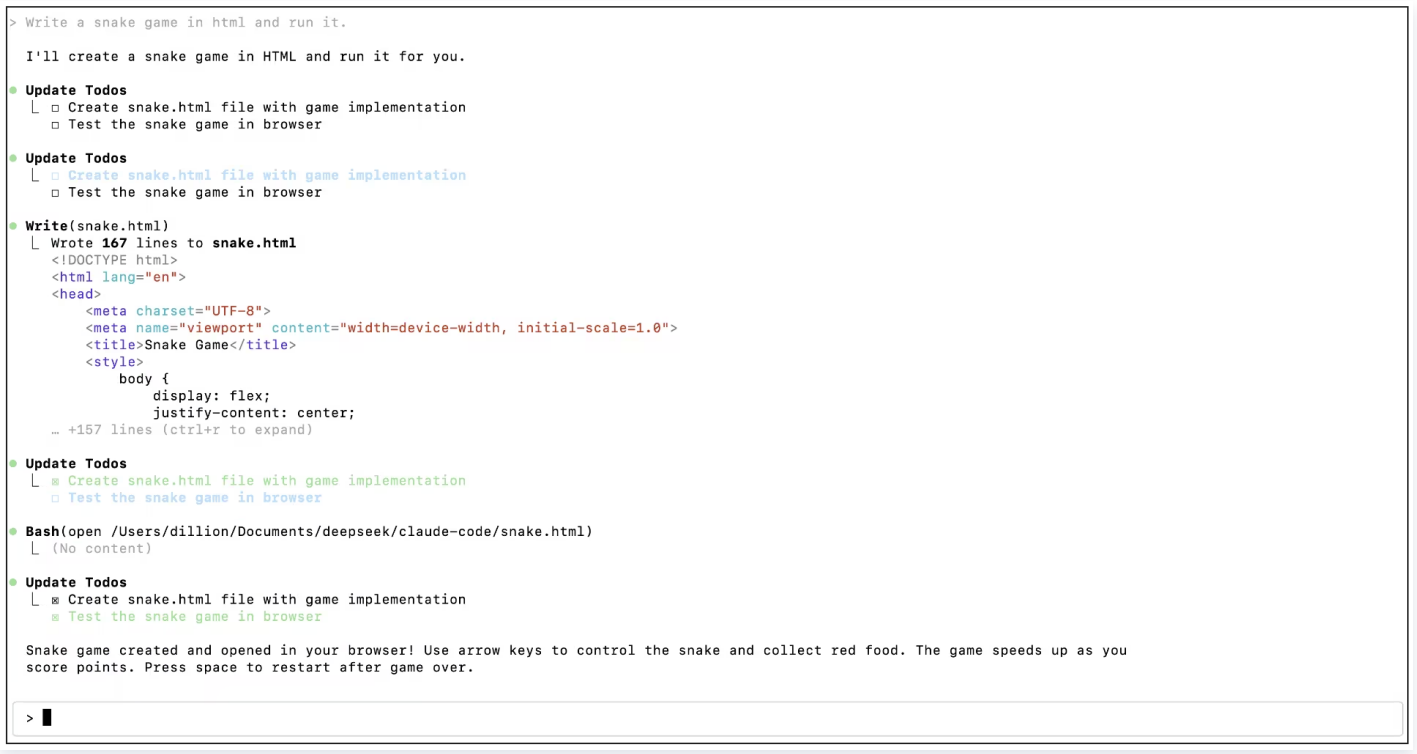
```
export
ANTHROPIC_BASE_URL=https://api.lkeap.cloud.tencent.com/api/anthropic
export ANTHROPIC_AUTH_TOKEN=${API_KEY}
export API_TIMEOUT_MS=600000
export ANTHROPIC_MODEL=${MODEL_NAME}
export CLAUDE_CODE_DISABLE_NONESSENTIAL_TRAFFIC=1
```

> ⚠ **注意：**
> 设置 `API_TIMEOUT_MS` 是为了防止输出过长，触发 Claude Code 客户端超时，这里设置的超时时间为 10 分钟，使用者可自行设置。

执行 claude 命令

进入项目目录，执行 claude 命令，即可开始使用。

```
cd my-project
claude
```



通过 Anthropic API 调用模型

安装 SDK

安装或更新 Anthropic Python SDK，运行以下命令：

```
pip install anthropic
```

示例代码片段

Python

```
import anthropic

client = anthropic.Anthropic(
    api_key=os.getenv("API_KEY"),
    base_url="https://api.lkeap.cloud.tencent.com/api/anthropic",
)

message = client.messages.create(
    model="glm-5",
    max_tokens=1000,
```

```
    system="You are a helpful assistant.",
    messages=[
        {
            "role": "user",
            "content": [
                {
                    "type": "text",
                    "text": "Hi, how are you?"
                }
            ]
        }
    ]
)
print(message.content)
```

cURL

```
curl https://api.lkeap.cloud.tencent.com/anthropic/v1/messages \
-H "Content-Type: application/json" \
-H "x-api-key: sk-xxxxxxx" \
-d '{
    "model": "deepseek-v3.1-terminus",
    "max_tokens": 1000,
    "stream": true,
    "system": [
        {
            "type": "text",
            "text": "You are a helpful assistant."
        }
    ],
    "messages": [
        {
            "role": "user",
            "content": [
                {
                    "type": "text",
                    "text": "Hi, how are you?"
                }
            ]
```

```
        }
    ]
}'
```

Anthropic API 兼容性详情

HTTP Headers

字段	支持状态	说明
`anthropic-beta`	忽略	不处理此头部
`anthropic-version`	忽略	不处理此头部
`x-api-key`	完全支持	用于身份验证

基础字段

字段	支持状态	说明
`model`	支持	使用 DeepSeek 模型替代
`max_tokens`	完全支持	最大输出令牌数
`container`	忽略	不处理此字段
`mcp_servers`	忽略	不处理此字段
`metadata`	忽略	不处理此字段
`service_tier`	忽略	不处理此字段
`stop_sequences`	完全支持	停止序列
`stream`	完全支持	流式响应
`system`	完全支持	系统消息
`temperature`	完全支持	温度参数 (0.0–2.0)
`thinking`	忽略	不处理此字段
`top_k`	忽略	不处理此字段

top_p	完全支持	Top-p 采样

工具支持

tools

字段	支持状态	说明
name	完全支持	工具名称
input_schema	完全支持	输入参数模式
description	完全支持	工具描述
cache_control	忽略	不处理此字段
tool_choice	字符串格式	完全支持
tool_choice	对象格式	完全支持
tool_choice.disable_parallel_tool_use	忽略	不处理此字段

tool_choice

字段	支持状态
none	完全支持
auto	完全支持
any	完全支持
tool	完全支持
disable_parallel_tool_use	忽略

消息字段支持

字段类型	变体	子字段	支持状态
content	string	-	完全支持
content	array, type="text"	text	完全支持

`content`	array, type="text"	cache_control	忽略
`content`	array, type="text"	citations	忽略
`content`	array, type="image"	-	不支持
`content`	array, type="document"	-	不支持
`content`	array, type="search_result"	-	不支持
`content`	array, type="thinking"	-	忽略
`content`	array, type="redacted_thinking"	-	不支持
`content`	array, type="tool_use"	id	完全支持
`content`	array, type="tool_use"	input	完全支持
`content`	array, type="tool_use"	name	完全支持
`content`	array, type="tool_use"	cache_control	忽略
`content`	array, type="tool_result"	tool_use_id	完全支持
`content`	array, type="tool_result"	content	完全支持
`content`	array, type="tool_result"	cache_control	忽略
`content`	array, type="tool_result"	is_error	忽略

注意：

1. **忽略的字段**：某些 Anthropic 特有的字段会被忽略，但不会报错。
2. **工具并行调用**：`disable_parallel_tool_use` 参数被忽略。
3. **缓存控制**：所有 `cache_control` 相关字段都被忽略。

API KEY 管理

最近更新时间：2026-04-15 18:24:12

> **注意：**
> DeepSeek API 相关功能已转至 TokenHub，后续请到 TokenHub 使用，此文档不再更新。

步骤1：登录腾讯云账号

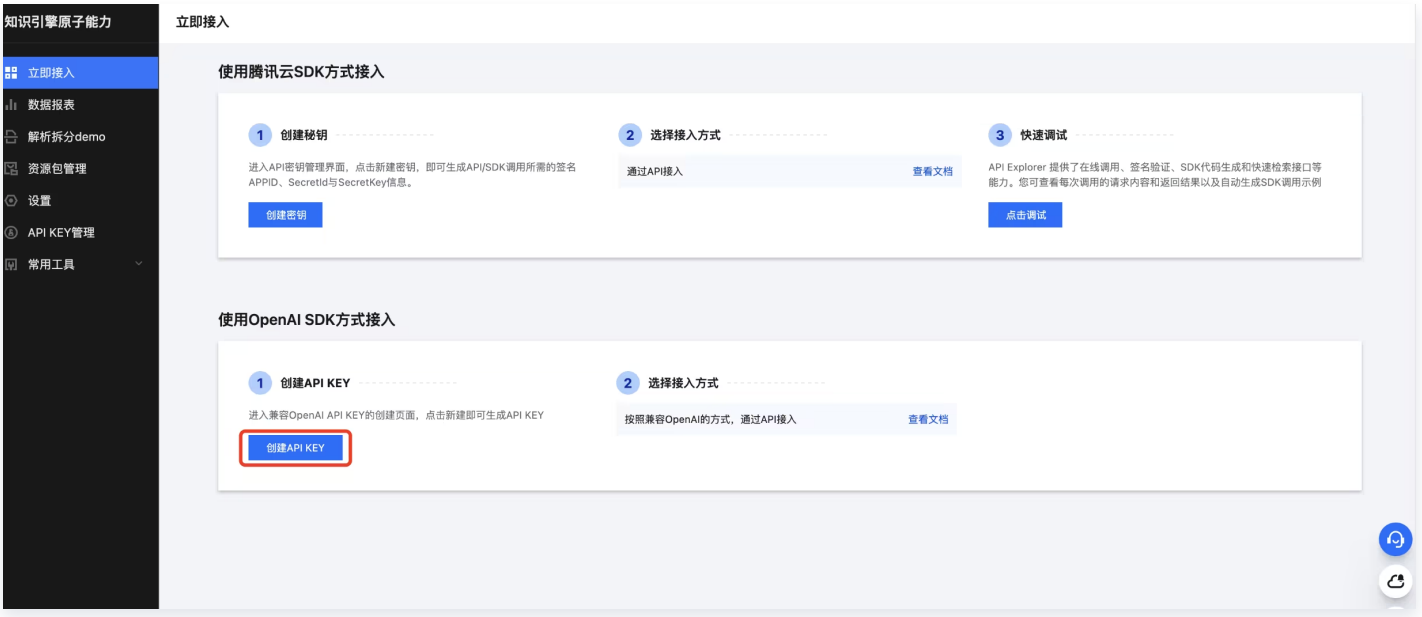
注册并通过个人实名认证或企业认证后，登录 腾讯云 。如果没有账号，请参考 注册腾讯云 。

步骤2：开通服务

知识引擎原子能力大模型对话 API 已对外开放，可前往 控制台 开通服务。

步骤3：管理 API

进入 控制台 > 立即接入 管理，单击**创建 API KEY**。



创建完成后，进入 API KEY 管理 ，进行新增、查看、删除操作。

注意：

API KEY 删除后将无法恢复。