

# TokenHub Token Plan



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

## Token Plan

Token Plan 套餐概览

Token Plan 快速入门

接入龙虾工具

OpenClaw

AutoClaw

WorkBuddy

CoPaw

Lighthouse OpenClaw

接入编程工具

CodeBuddy Code

OpenCode

Claude Code

Codex

Cline

Cursor

Kilo CLI

Kilo Code

Token Plan 常见问题

# Token Plan

## Token Plan 套餐概览

最近更新时间：2026-04-03 21:02:21

腾讯云大模型 Token Plan 是面向龙虾和编程场景设计的专属订阅套餐，覆盖腾讯混元、MiniMax、GLM、Kimi 等国产主流模型，兼容热门龙虾工具和主流编程工具，内设更多套餐档位可供选择。具备如下优势：

- **集合主流国产模型，可按需切换：**一次订阅即可自由切换 Tencent HY 2.0 Instruct、Tencent HY 2.0 Think、MiniMax-M2.5、Kimi-K2.5、GLM-5、Hunyuan-T1、Hunyuan-TurboS 等模型，更多更优模型在持续接入中。
- **热门龙虾工具全适配、主流 AI 编码工具全兼容：**支持 OpenClaw、Claude Code、OpenCode、Cline、Cursor、Roo Code、Kilo Code、Codex CLI 等 AI 工具，共享套餐额度，让龙虾随时随地干活，体验 AI 编码自由。
- **多种套餐适配不同场景：**提供 Lite、Standard、Pro、Max 四档套餐，满足从新手体验龙虾到高强度编程不同场景选择。
- **同等用量费用节省超过 50%：**同模型套餐价相比于直接调用 [文本生成服务](#) 低 50% 以上，套餐分阶定价，等级越高，单价越低，用量越多，省得越多。

### 套餐详情

访问 [Token Plan 购买页](#)，根据实际需求选择并购买套餐，套餐说明如下：

套餐	用量限制	价格	适用场景
个人版-体验套餐 (Lite)	每订阅月 3500万 Tokens	39元/月	<p>新手尝鲜，入门首选。 适用于首次体验龙虾能力，可实现网页操作、文件处理、数据分析、定时任务等。</p> <div><p><b>⚠ 注意：</b> 按龙虾基础使用场景预估可支持约 70 轮问答式交互（实际轮次受单轮输入输出内容长度、任务复杂度、代码量等因素影响，具体以实际使用为准）。</p></div>
个人版-基础套餐 (Standard)	每订阅月 1亿 Tokens	99元/月	<p>日常使用，高性价比。 适合日常用龙虾办公和轻量开发，可实现批量文件处理、Demo 制作、自动化 workflow 等。</p> <div><p><b>⚠ 注意：</b></p></div>

			按龙虾基础使用场景预估可支持约 200 轮问答式交互（实际轮次受单轮输入输出内容长度、任务复杂度、代码量等因素影响，具体以实际使用为准）。
个人版-进阶套餐（Pro）	每订阅月 3.2亿 Tokens	299 元/月	高频 AI 开发，Token 配额相比基础版提升至 3 倍。适合每天高频使用 AI 的开发者和效率达人，多仓库并行、复杂逻辑生成、代码重构、Agent 编排等。
个人版-专业套餐（Max）	每订阅月 6.5亿 Tokens	599 元/月	更多额度加持，重度 AI 开发首选。适合把 AI 当核心生产力工具的重度用户，全栈 AI 生成、多 Agent 协同、CI 自动化等。

## 订阅须知

Token Plan 不支持退款。因此在订阅前请知悉以下重要内容：

- 订阅账号规范：**为订阅人专享使用，**严禁账号共享**。若存在账号共享行为，可能导致**订阅权益受限**，腾讯不承担赔偿责任，敬请知悉。
- 严禁 API 调用：**仅限在 AI 工具（例如：Claude Code、CodeBuddy Code、OpenClaw 等）中使用，禁止以 API 调用的形式用于自动化脚本、自定义应用程序后端或任何非交互式批量调用场景。将套餐 API Key 用于允许范围之外的调用将被视为违规或滥用，可能会导致订阅被暂停或 API Key 被封禁。对于违规而导致服务受损，腾讯不承担赔偿责任。
- 速率限制：**并发速率与您的套餐等级相关，平台会根据资源进行动态调整，基本原则 Max > Pro > Standard > Lite。

## 可用模型

所有套餐均支持以下模型：

- Auto 模型（系统智能路由）
- Tencent HY 2.0 Instruct
- Tencent HY 2.0 Think
- Kimi-K2.5
- MiniMax-M2.5
- GLM-5
- Hunyuan-T1（即将下线）
- Hunyuan-TurboS（即将下线）

更多模型将持续接入中。

### 说明：

- Token Plan 可用模型暂不支持多模态能力。

- Kimi-K2.5 当前资源负载较高，高峰时段可能触发请求限频机制，为保障使用体验，请优先选用其他模型。

## 支持 AI 工具

龙虾工具		
<p><b>OpenClaw</b> 开源、自托管个人 AI 助手</p>	<p><b>WorkBuddy</b> 腾讯云全场景 AI 桌面智能体</p>	<p><b>Lighthouse OpenClaw</b> 腾讯云轻量应用服务器部署 OpenClaw</p>
<p><b>AutoClaw</b> 本地 AI 智能体</p>	<p><b>CoPaw</b> 开源 AI 智能体</p>	<p>持续接入中</p>

编程工具		
<p><b>CodeBuddy Code</b> 基于腾讯云 AI 技术的智能编程工具</p>	<p><b>OpenCode</b> 开源 AI 编程代理工具</p>	<p><b>Kilo Code</b> 轻量高性能编程工具</p>
<p><b>Cline</b> VSCode 扩展，代码补全和调试</p>	<p><b>Cursor</b> AI 原生代码编辑器</p>	<p><b>Kilo CLI</b> 轻量高性能编程工具命令行</p>
<p><b>Claude Code</b> AI 终端编程助手，支持自然语言编程</p>	<p><b>Codex</b> OpenAI 推出的智能编程代理</p>	<p>持续接入中</p>

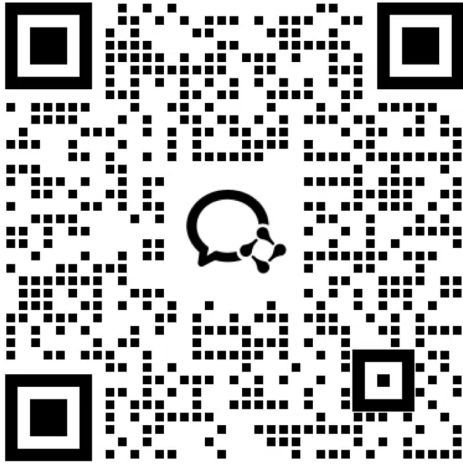
## 套餐有效期

套餐以自然月为单位，自购买成功当日开始计算。1 个月套餐有效期示例如下：

购买日期	购买时长	结束日期
01.04	1 个月	02.04 23:59
01.31	1 个月	02.28 23:59
02.01	1 个月	03.01 23:59

## Token Plan 体验交流群

为方便您更好地使用 Token Plan 服务，我们设立了专属体验交流群。您可通过下方二维码入群：



## 快速开始

详情请参见 [Token Plan 快速入门](#)。

## 常见问题

详情请参见 [Token Plan 常见问题](#)。

# Token Plan 快速入门

最近更新时间：2026-04-03 21:02:21

本文介绍如何在龙虾或者编程工具中快速使用 Token Plan 套餐所支持的模型进行办公或者代码开发。

## 步骤一：订阅 Token Plan 套餐

前往 [Token Plan 购买页](#)，选择适合您的订阅套餐。套餐介绍请参见 [Token Plan 套餐概览](#)。

腾讯云 Token Plan

面向龙虾和编程场景设计的专属订阅套餐，覆盖最新国产主流模型、完全兼容热门龙虾工具和主流编程工具，支持多种套餐选择。

套餐名称	适用人群	价格 (元/月)	核心配置
Lite	新手尝鲜, 入门首选	¥39.00	3500万 Tokens, 适用于首次体验龙虾能力, 可执行约 70 轮问答, 集成主流国产模型, 兼容主流龙虾和 AI 编程工具
Standard	日常使用, 高性价比	¥99.00	1 亿 Tokens, 适用于首次体验龙虾能力, 可执行约 200 轮问答, 集成主流国产模型, 兼容主流龙虾和 AI 编程工具
Pro	高频 AI 开发, Token 配额提升至 3 倍	¥299.00	3.2 亿 Tokens, 适合每天高频使用 AI 的开发者 and 效率达人, 集成主流国产模型, 兼容主流龙虾和 AI 编程工具
Max	极致额度加持, 重度 AI 开发首选	¥599.00	6.5 亿 Tokens, 适合把 AI 当核心生产力工具的重度用户, 集成主流国产模型, 兼容主流龙虾和 AI 编程工具

## 步骤二：获取核心配置

您可以获取 Model Name、Base URL、API Key，用于工具中。

### 获取 Model Name

在工具配置文件中指定 Model Name，可实时切换模型。支持配置的 Model Name：

- tc-code-latest
- hunyuan-2.0-instruct
- hunyuan-2.0-thinking
- minimax-m2.5 (兼容别名 minimax-m-2-5)
- kimi-k2.5 (兼容别名 kimi-k-2-5)
- glm-5 (兼容别名 glm-5-0)
- hunyuan-t1 (即将下线)
- hunyuan-turbo (即将下线)

#### 说明：

- 配置 Model Name 时，支持使用全小写格式。
- 设置 Model Name 为 tc-code-latest 值，平台会通过算法自动匹配最优模型。
- kimi-k2.5 当前资源负载较高，高峰时段可能触发请求限频机制，为保障使用体验，请优先选用其他模型。

- Token Plan 可用模型暂不支持多模态能力。

## 获取 Base URL 和完整 URL

不同的工具配置的 URL 根据兼容的协议会有不同：

### 1. Base URL

- 兼容 OpenAI 接口协议工具：`https://api.lkeap.cloud.tencent.com/plan/v3`。
- 兼容 Anthropic 接口协议工具：`https://api.lkeap.cloud.tencent.com/plan/anthropic`。

### 2. 完整 URL

- 兼容 OpenAI 接口协议工具：`https://api.lkeap.cloud.tencent.com/plan/v3/chat/completions`。
- 兼容 Anthropic 接口协议工具：`https://api.lkeap.cloud.tencent.com/plan/anthropic/v1/messages`。

## 获取 API Key

1. 生成 API Key: 在 TokenHub > [Token Plan](#) 页面，单击**生成密钥**，生成您的专属 API Key。



2. 获取 API Key: 密钥生成成功后，即可单击**复制**，获取套餐专属 API Key（格式为 `sk-tp-xxx`）。



## 步骤三：快速验证模型调用

获取 API Key 后您可在命令行终端通过 cURL 简单验证连通性，请将下方 `$your_api_key` 替换为您的实际 API Key，`$model_name` 可以对照 [Model Name](#) 换成您要测试的模型：

```
curl --location
'https://api.lkeap.cloud.tencent.com/plan/v3/chat/completions' \
--header 'Authorization: Bearer $your_api_key' \
--header 'Content-Type: application/json' \
--data '{
  "model": "$model_name",
  "messages": [
    {
      "role": "user",
      "content": "介绍下你自己"
    }
  ],
  "stream": true
}'
```

## 步骤四：配置 AI 工具

Token Plan 兼容热门的 [龙虾工具和 AI 编码工具](#)，您可以根据自己的偏好选择，将 URL、API Key 和具体的模型（Model Name）配置到工具中，就能使用大模型服务。

## 步骤五（可选）：管理套餐用量

您可以在控制台查看您套餐的 Tokens 消耗情况，也可以根据需要对套餐进行升级。

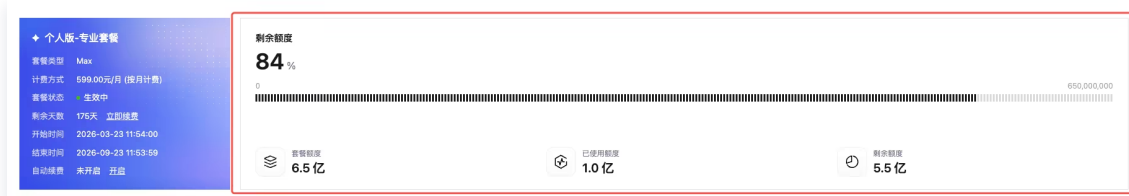
## 1. 查看套餐用量消耗

可以前往 TokenHub > [Token Plan](#) 页面查看本周期套餐额度消耗情况。

### 说明：

**额度刷新规则：**套餐额度在订阅月的周期内耗尽后，不会消耗其他资源包或账户余额，需要等待下一个订阅月自动恢复额度。您也可以选择升级到更高等级套餐享有更充裕的 Token 配额。

1.1 在**剩余额度**板块查看套餐额度整体消耗情况。



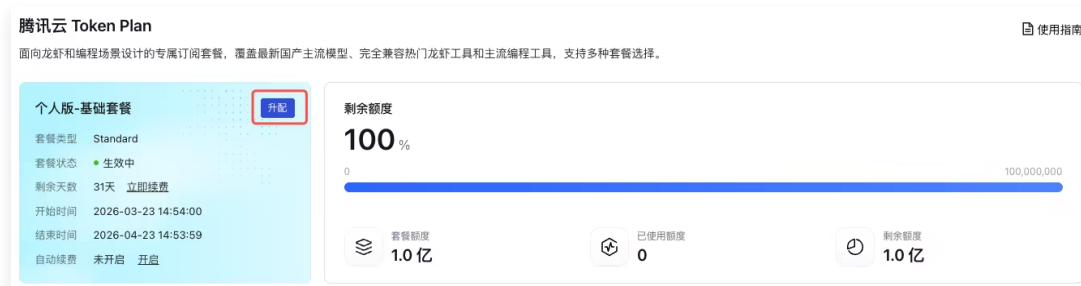
1.2 在页面下方的用量详情板块查看**每日 Token 总用量、分钟级输入 / 输出 Token 用量、请求维度 Token 用量**。

## 2. 升级套餐用量

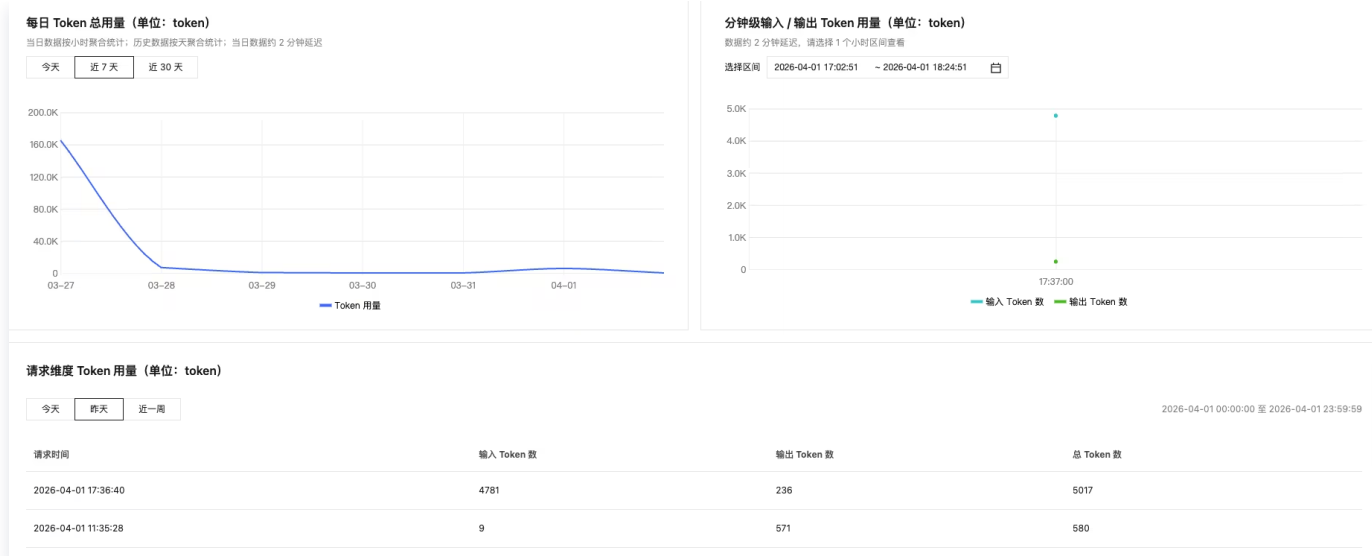
### 说明：

- Token Plan 套餐仅支持套餐升级，不支持套餐降级操作。
- 升级套餐的费用 = 新套餐费用 - 原套餐费用（补差价）。
- 升级后套餐 Token 剩余量 = 新套餐总 Token 量 - 原套餐 Token 使用量。
- 套餐升级成功后：
  - 套餐的 API Key 保持一样。
  - 套餐到期时间跟原套餐保持一致，例如：Lite 套餐是 2026-04-23 到期，升级到 Standard 套餐也是 2026-04-23 到期。

2.1 在 TokenHub > [Token Plan](#) 页面，单击**升配**，进行套餐升级。



2.2 选择要升配的目标套餐名称，升级到更高等级套餐以获得更多 Token 量。



## 步骤六 (可选) : 套餐续费

### 警告:

请您在 Token Plan 套餐到期前完成续费操作, 套餐到期后将无法进行续费, 套餐将失效, API Key 也会失效。

## 续费规则

套餐续费是指续订下一订阅月的套餐, 续费完成后, 新的额度将在下个订阅月生效:

- 若本订阅月额度未用尽: 剩余额度可继续使用至本订阅月结束, 下个订阅月自动重置为新的额度。
- 若本订阅月额度已用尽: 续费操作不会重置本订阅月的额度, 新的额度需等待下个订阅月生效。

示例如下: 用户购买 Standard 套餐, 本订阅月 (2026-03-23~2026-04-23) Token 总额度 1 亿。用户为下个订阅月 (2026-04-23~2026-05-23) 续费, 不同场景可用额度说明如下:

场景	本月额度消耗情况	可用额度说明
本订阅月额度未用尽	假如本订阅月 (2026-03-23~2026-04-23) 已用 0.5 亿。	<ul style="list-style-type: none"> <li>● 本订阅月 (2026-03-23~2026-04-23) 剩余可用额度 0.5 亿。</li> <li>● 下个订阅月 (2026-04-23~2026-05-23) 可用额度 1 亿。</li> </ul>
若本订阅月额度已用尽	假如本订阅月 (2026-03-23~2026-04-23) 已用 1 亿。	<ul style="list-style-type: none"> <li>● 本订阅月 (2026-03-23~2026-04-23) 剩余可用额度 0。</li> <li>● 下个订阅月 (2026-04-23~2026-05-23) 可用额度 1 亿。</li> </ul>

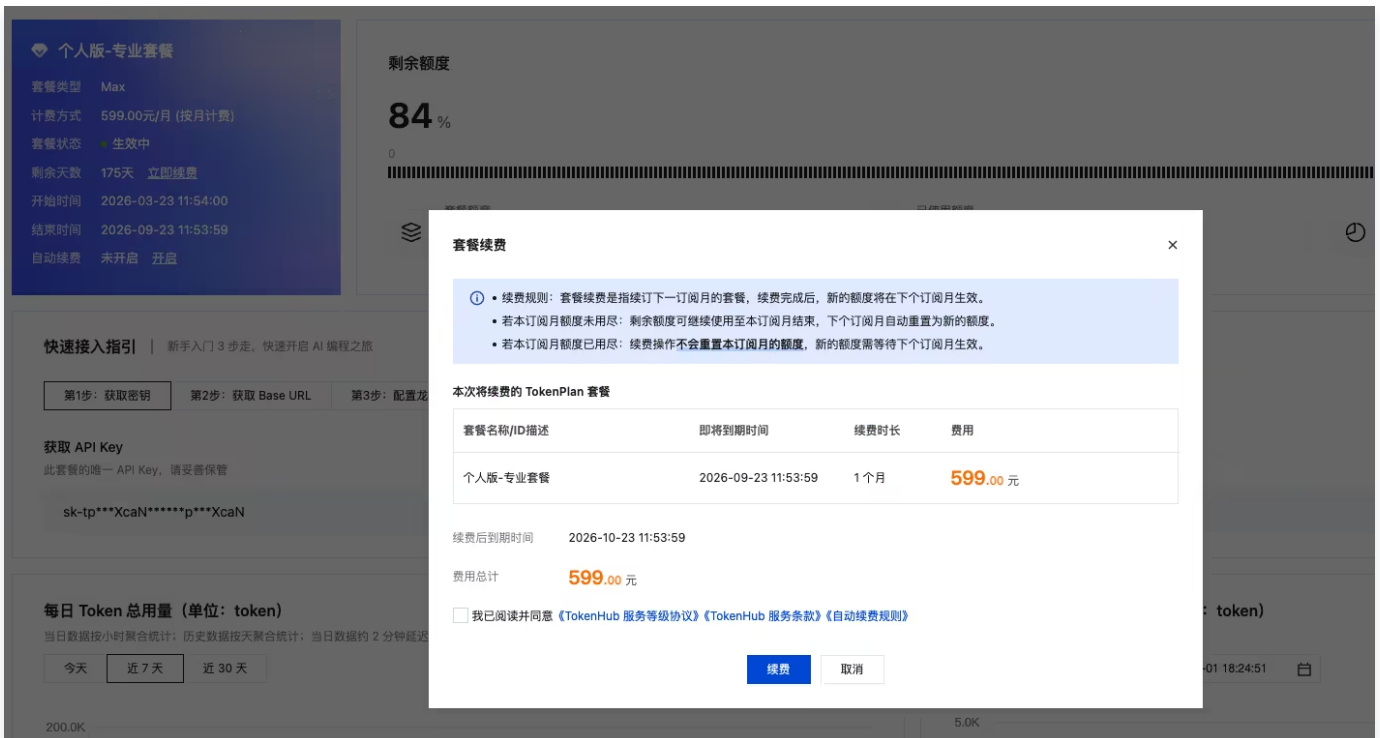
## 续费方式

Token Plan 支持手动续费和自动续费两种方式。具体说明如下:

- **手动续费：**套餐自购买到到期前，您可以在任意时间为套餐手动续费，以延长套餐的使用时间。  
在 TokenHub > [Token Plan](#) 页面，单击**立即续费**。



确认续费周期，单击**续费**，即可完成操作。



- **自动续费：**在 TokenHub > [Token Plan](#) 页面，单击**开启**，开启自动续费。套餐开启自动续费后，在账户余额充足的情况下，套餐到期前一天会自动执行续费。自动续费可以减少手动续费的管理成本，避免因忘记手动续费而导致 API Key 失效。



## 常见错误码

错误码	错误信息	可能原因	解决方法
401	unauthorized	URL 或者 API Key 配置错误	<ol style="list-style-type: none"><li>1. 确认 URL 是 <a href="#">Token Plan 的 URL</a>。</li><li>2. 确认 API Key 根据 <a href="#">API Key 获取指引</a> 正确获取。</li><li>3. 参考 <a href="#">龙虾工具</a>、<a href="#">编程工具</a> 指引，确认 URL 和 API Key 配置正确。</li></ol>
429	tpm rate limit exceeded	系统负载高触发限流	请重试 1-2 次或者切换其他模型。
20033	invalid model	模型名称配置错误	检查配置的模型名称（Model Name）是否是 <a href="#">Token Plan 支持的 Model Name</a> 。
20059	Input length too long	输入长度超出了允许范围	可通过新建会话解决报错。
20097	month allocated quota exceeded	每个订阅月请求额度已用完	等待至下个订阅月的同一时刻（订阅时刻）额度重置，或升级至更高级套餐。

## 常见问题

详情请参见 [Token Plan 常见问题](#)。

# 接入龙虾工具 OpenClaw

最近更新时间：2026-04-03 16:27:22

OpenClaw 是一款开源的本地 AI 代理框架，能让 AI 从“回答问题”进化为“动手执行任务”。它支持在 Windows、macOS 和 Linux 上自主运行，通过内置工具和可扩展的插件体系，自动完成文件整理、邮件处理、代码编写等操作。

腾讯云 Token Plan 支持在 OpenClaw 中使用，可参考本文进行配置及使用。

## 前置条件

在开始配置前，请确保您已完成以下准备工作：

1. [订阅 Token Plan 套餐](#)。
2. [获取 API Key 和 模型列表](#)。

## 安装 OpenClaw

您需要根据实际环境类型，执行对应命令安装 OpenClaw。

### macOS/Linux

```
curl -fsSL https://openclaw.ai/install.sh | bash
```

### Windows

在 PowerShell 中执行以下命令。

```
iwr -useb https://openclaw.ai/install.ps1 | iex
```

根据提示信息完成 OpenClaw 配置，参考配置如下。

向导配置项	配置内容
I understand this is powerful and inherently risky.	选择 <b>Yes</b> 。

Continue?	
Onboarding mode	选择 <b>QuickStart</b> 。
Model/auth provider	选择 <b>Skip for now</b> ，后续可以配置。
Filter models by provider	选择 <b>All providers</b> 。
Default model	使用默认配置。
Select channel (QuickStart)	选择 <b>Skip for now</b> ，后续可以配置。
Configure skills now? (recommended)	选择 <b>No</b> ，后续可以配置。
Enable hooks?	按空格键选中选项，按回车键进入下一步。
How do you want to hatch your bot?	选择 <b>Hatch in TUI</b> 。

## 配置 Token Plan

### ⚠ 注意:

如果出现配置正常但是访问404，401等问题是因为 OpenClaw 的缓存问题导致，您可以直接修改本地配置文件，重启 OpenClaw gateway，配置文件路径如下：

- Windows: `C:\Users\<<用户名>\.openclaw\openclaw.json`
- macOS/Linux: `~/\.openclaw/openclaw.json`

1. 执行以下命令打开 OpenClaw 的 WebUI，然后在 WebUI 的左侧菜单栏中选择 **Config > Raw**。

```
openclaw dashboard
```

2. 在 `openclaw.json` 配置项中增加 `models.providers` 的配置。

- `baseUrl`: 参考样例，使用 Token Plan 兼容 OpenAI 接口协议的 Base URL ( `https://api.lkeap.cloud.tencent.com/plan/v3` ) 配置。
- `<USER_API_KEY>`: 替换成您自己的 API Key。

```
"models": {
  "mode": "merge",
  "providers": {
    "tencent-token-plan": {
      "baseUrl": "https://api.lkeap.cloud.tencent.com/plan/v3",
      "apiKey": "<USER_API_KEY>",
      "api": "openai-completions",
```

```
"models": [  
  {  
    "id": "tc-code-latest",  
    "name": "Auto",  
    "reasoning": false,  
    "input": ["text"],  
    "cost": { "input": 0, "output": 0, "cacheRead": 0,  
"cacheWrite": 0 },  
    "contextWindow": 196608,  
    "maxTokens": 32768  
  },  
  {  
    "id": "hunyuan-2.0-instruct",  
    "name": "Tencent HY 2.0 Instruct",  
    "reasoning": false,  
    "input": ["text"],  
    "cost": { "input": 0, "output": 0, "cacheRead": 0,  
"cacheWrite": 0 },  
    "contextWindow": 128000,  
    "maxTokens": 16000  
  },  
  {  
    "id": "hunyuan-2.0-thinking",  
    "name": "Tencent HY 2.0 Think",  
    "reasoning": false,  
    "input": ["text"],  
    "cost": { "input": 0, "output": 0, "cacheRead": 0,  
"cacheWrite": 0 },  
    "contextWindow": 128000,  
    "maxTokens": 32000  
  },  
  {  
    "id": "hunyuan-t1",  
    "name": "Hunyuan-T1",  
    "reasoning": false,  
    "input": ["text"],  
    "cost": { "input": 0, "output": 0, "cacheRead": 0,  
"cacheWrite": 0 },  
    "contextWindow": 64000,  
    "maxTokens": 32000  
  }  
]
```

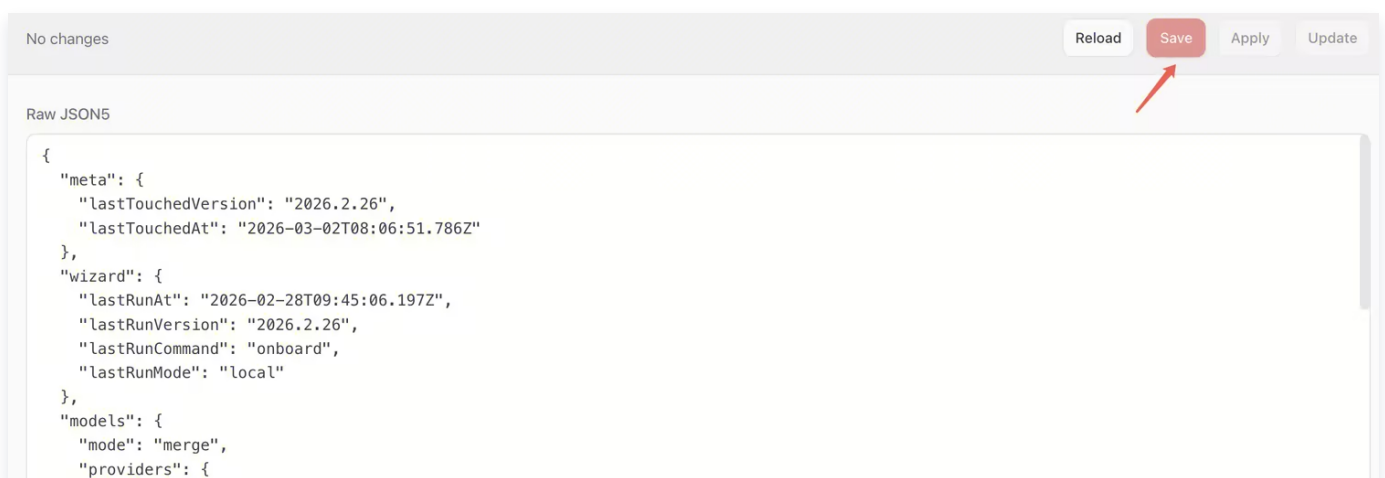
```
},
{
  "id": "hunyuan-turbos",
  "name": "hunyuan-turbos",
  "reasoning": false,
  "input": ["text"],
  "cost": { "input": 0, "output": 0, "cacheRead": 0,
"cacheWrite": 0 },
  "contextWindow": 32000,
  "maxTokens": 16000
},
{
  "id": "minimax-m2.5",
  "name": "MiniMax-M2.5",
  "reasoning": false,
  "input": ["text"],
  "cost": { "input": 0, "output": 0, "cacheRead": 0,
"cacheWrite": 0 },
  "contextWindow": 196608,
  "maxTokens": 32768
},
  {
    "id": "kimi-k2.5",
    "name": "Kimi-K2.5",
    "reasoning": false,
    "input": ["text"],
    "cost": { "input": 0, "output": 0, "cacheRead": 0,
"cacheWrite": 0 },
    "contextWindow": 262144,
    "maxTokens": 32768
  },
{
  "id": "glm-5",
  "name": "GLM-5",
  "reasoning": false,
  "input": ["text"],
  "cost": { "input": 0, "output": 0, "cacheRead": 0,
"cacheWrite": 0 },
  "contextWindow": 202752,
  "maxTokens": 16384
}
```

```
}  
]  
}  
}  
}
```

3. 在 `openclaw.json` 配置项中修改 `agents.default`，增加模型列表信息，其中：**primary** 字段指定的模型可以根据您的偏好进行配置。

```
"agents": {  
  "defaults": {  
    "model": {  
      "primary": "tencent-token-plan/glm-5"  
    },  
    "models": {  
      "tencent-token-plan/tc-code-latest": {},  
      "tencent-token-plan/hunyuan-2.0-instruct": {},  
      "tencent-token-plan/hunyuan-2.0-thinking": {},  
      "tencent-token-plan/hunyuan-t1": {},  
      "tencent-token-plan/hunyuan-turbos": {},  
      "tencent-token-plan/minimax-m2.5": {},  
      "tencent-token-plan/kimi-k2.5": {},  
      "tencent-token-plan/glm-5": {}  
    }  
  }  
}
```

4. 在界面上单击 **Save** 保存配置，保存完成后，单击 **Update** 使配置生效。

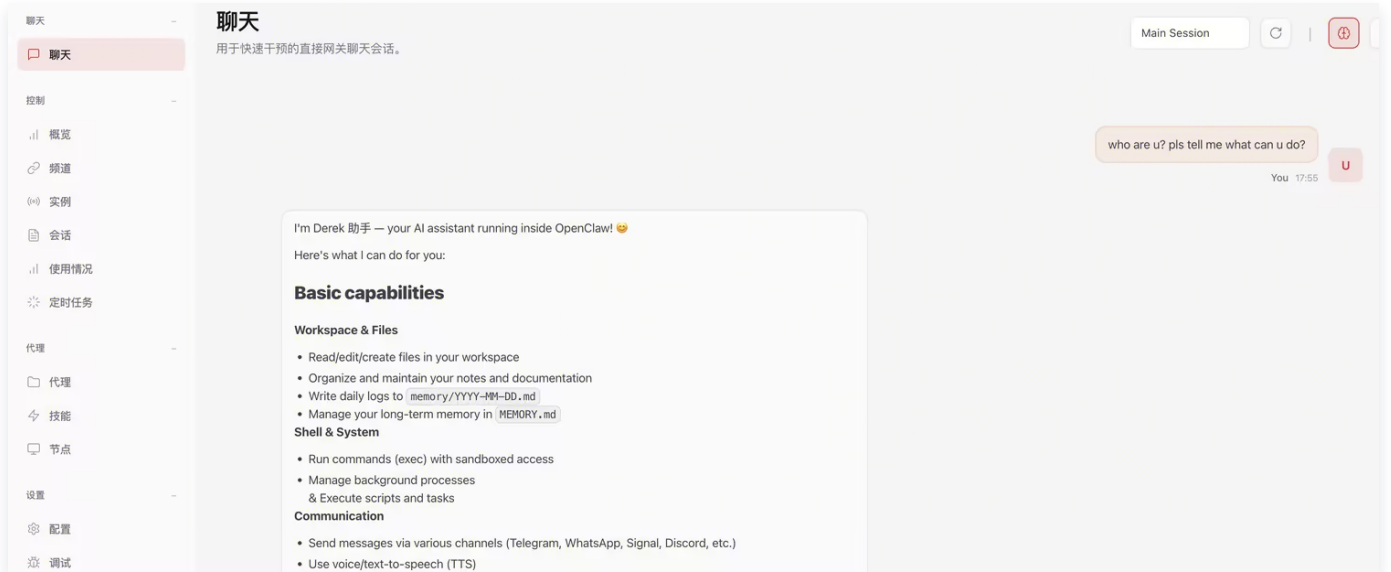


# 使用 OpenClaw

OpenClaw 支持通过 WebUI 和 TUI 的形式进行交互使用。

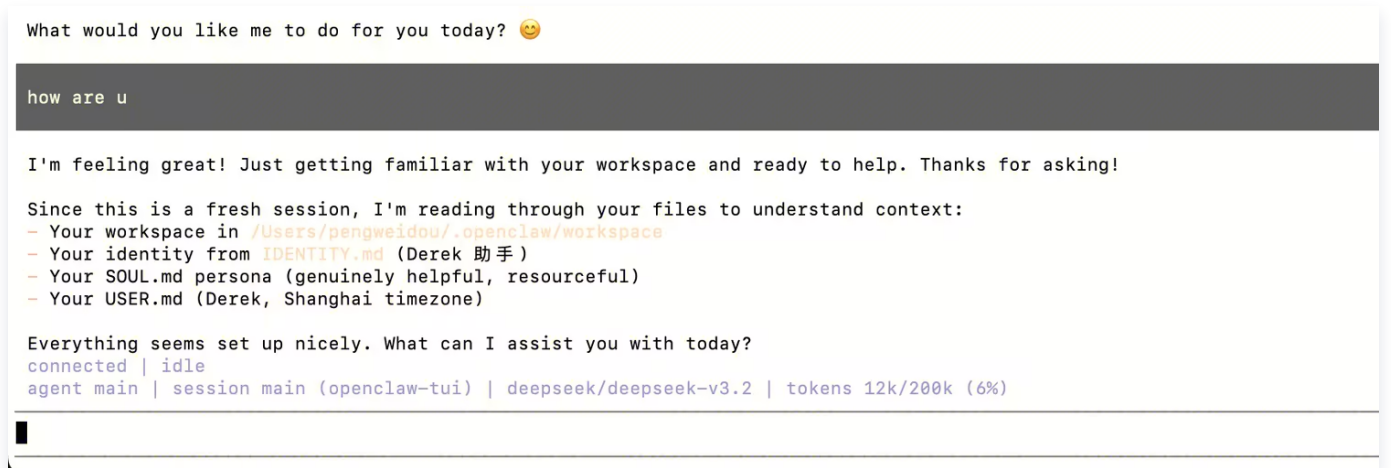
## • WebUI

您可以通过 `openclaw dashboard` 命令打开 WebUI，单击 WebUI 左侧的聊天，即可进行交互。



## • TUI

您可以通过 `openclaw tui` 命令打开 TUI 进行交互。



# AutoClaw

最近更新时间：2026-04-03 16:27:22

AutoClaw 是智谱 AI 推出的零门槛本地 AI 智能体，支持 OpenClaw 核心执行能力，主打普通用户轻松上手。本文介绍如何在 AutoClaw 中配置与使用腾讯云大模型 Token Plan。

AutoClaw 支持通过“自定义模型”功能接入任何兼容 OpenAI 协议的第三方模型服务。通过以下配置，可将腾讯云 Token Plan 的模型接入 AutoClaw。

## 前置条件

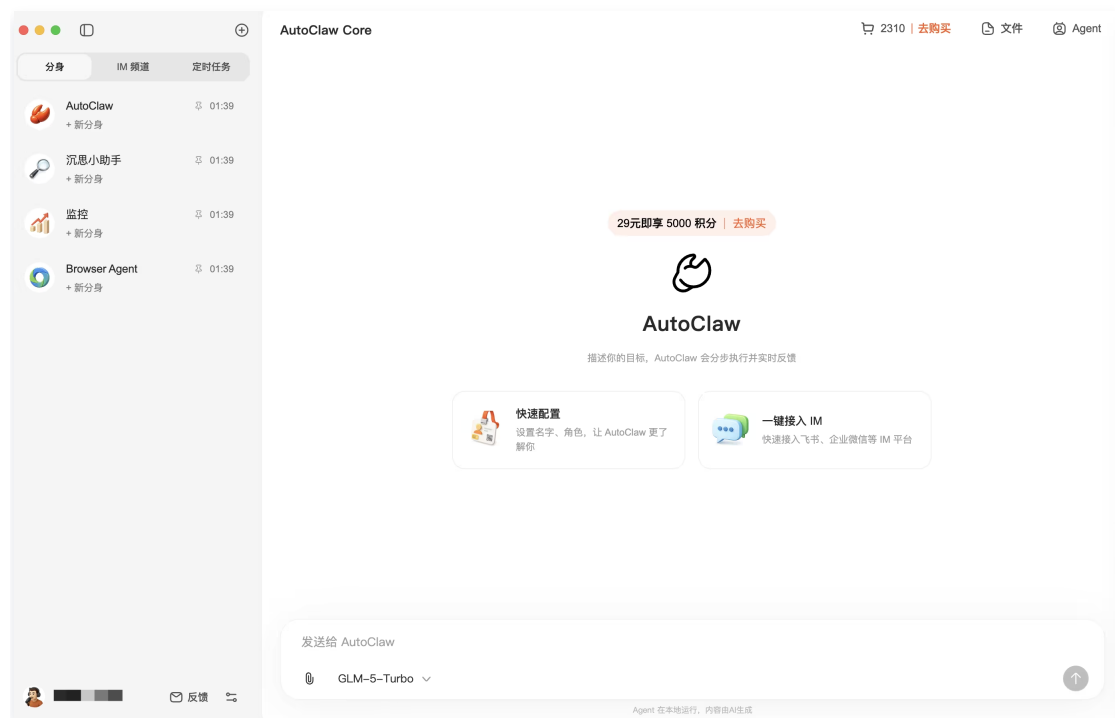
在开始配置前，请确保您已完成以下准备工作：

1. [订阅 Token Plan 套餐](#)。
2. [获取 API Key 和 模型列表](#)。

## 操作步骤

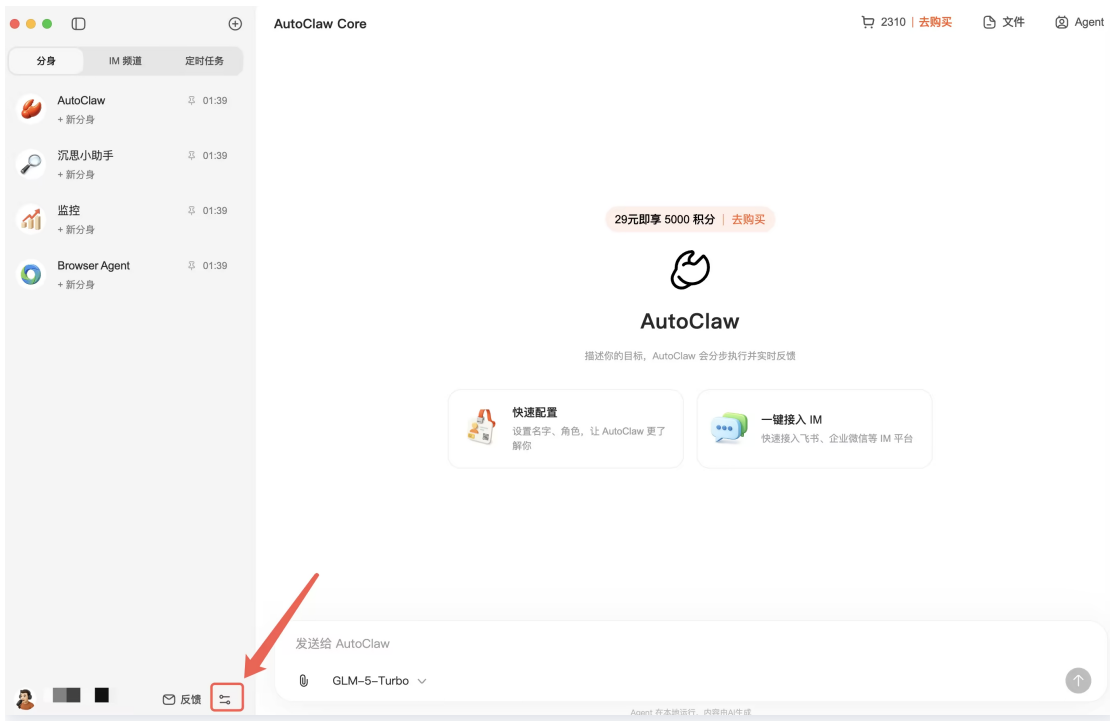
### 步骤一：打开 AutoClaw 应用

启动 AutoClaw 应用，进入主界面。



### 步骤二：进入设置界面

单击左下角反馈按钮旁的设置，进入设置界面。

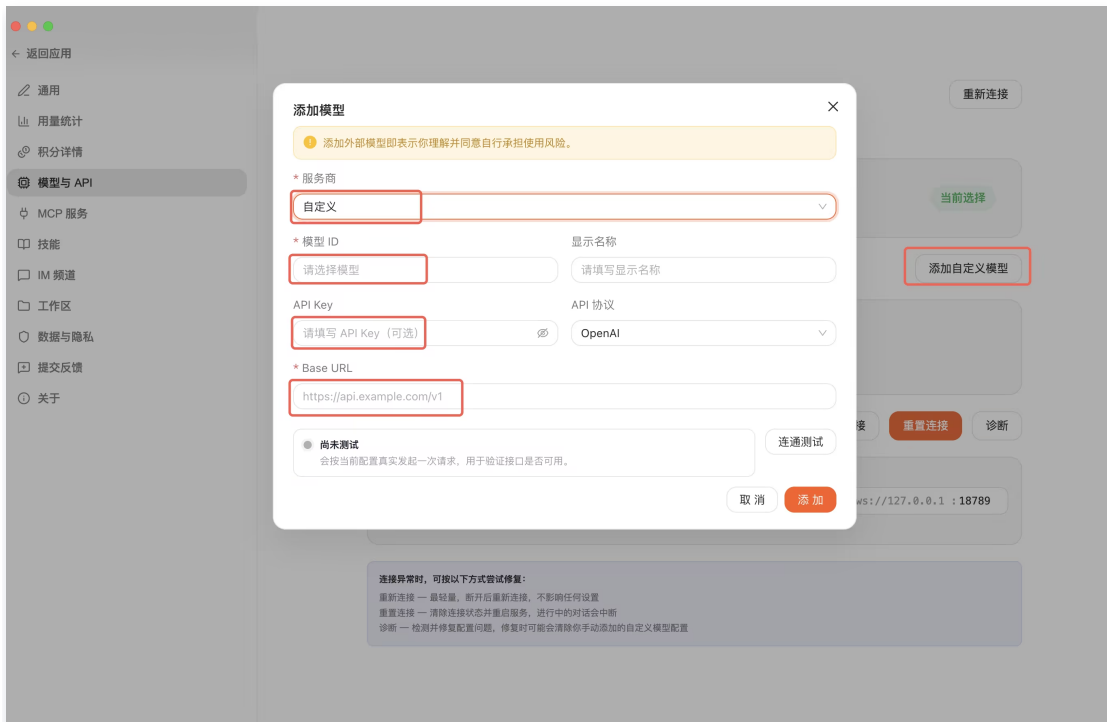


### 步骤三：进入模型配置页面

在设置界面的左侧导航栏中，单击**模型与 API**，进入模型配置页面。页面会显示**内置模型**和**自定义模型**两个区域。

### 步骤四：添加自定义模型

单击右侧**添加自定义模型**，在弹出的**添加模型**对话框中，选择**服务商**为**自定义**，依次填写**模型 ID**、**API Key**、**Base URL** 等信息：



## 配置 Auto

参数	值
服务商	选择自定义
模型 ID	<code>tc-code-latest</code>
显示名称	Auto
API Key	购买 Token Plan 后获取的专属 API Key
API 协议	选择 OpenAI
Base URL	<code>https://api.lkeap.cloud.tencent.com/plan/v3</code>

### 配置 Tencent HY 2.0 Instruct

参数	值
服务商	选择自定义
模型 ID	<code>hunyuan-2.0-instruct</code>
显示名称	Tencent HY 2.0 Instruct
API Key	购买 Token Plan 后获取的专属 API Key
API 协议	选择 OpenAI
Base URL	<code>https://api.lkeap.cloud.tencent.com/plan/v3</code>

### 配置 Tencent HY 2.0 Think

参数	值
服务商	选择自定义
模型 ID	<code>hunyuan-2.0-thinking</code>
显示名称	Tencent HY 2.0 Think
API Key	购买 Token Plan 后获取的专属 API Key
API 协议	选择 OpenAI
Base URL	<code>https://api.lkeap.cloud.tencent.com/plan/v3</code>

### 配置 MiniMax-M2.5

参数	值
服务商	选择自定义
模型 ID	<code>minimax-m2.5</code>
显示名称	MiniMax-M2.5
API Key	购买 Token Plan 后获取的专属 API Key
API 协议	选择 OpenAI
Base URL	<code>https://api.lkeap.cloud.tencent.com/plan/v3</code>

### 配置 Kimi-K2.5

参数	值
服务商	选择自定义
模型 ID	<code>kimi-k2.5</code>
显示名称	Kimi-K2.5
API Key	购买 Token Plan 后获取的专属 API Key
API 协议	选择 OpenAI
Base URL	<code>https://api.lkeap.cloud.tencent.com/plan/v3</code>

### 配置 GLM-5

参数	值
服务商	选择自定义
模型 ID	<code>glm-5</code>
显示名称	GLM-5
API Key	购买 Token Plan 后获取的专属 API Key
API 协议	选择 OpenAI
Base URL	<code>https://api.lkeap.cloud.tencent.com/plan/v3</code>

### 配置 Hunyuan-T1

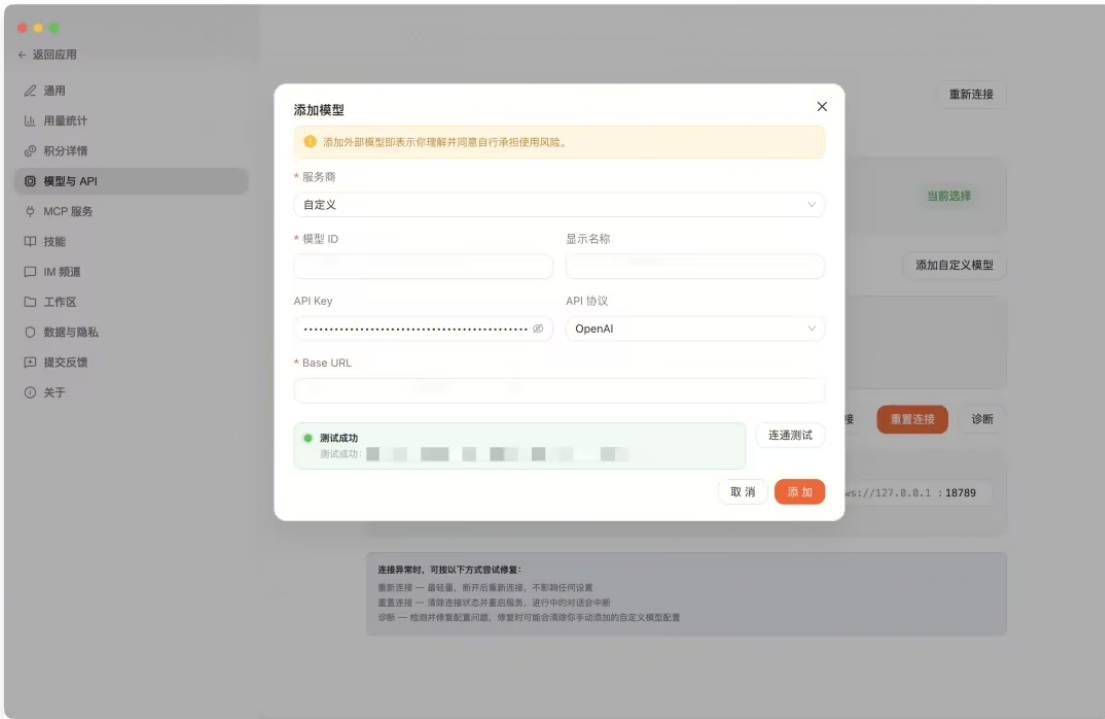
参数	值
服务商	选择自定义
模型 ID	hunyuan-t1
显示名称	Hunyuan-T1
API Key	购买 Token Plan 后获取的专属 API Key
API 协议	选择 OpenAI
Base URL	https://api.lkeap.cloud.tencent.com/plan/v3

## 配置 Hunyuan-TurboS

参数	值
服务商	选择自定义
模型 ID	hunyuan-turbos
显示名称	Hunyuan-TurboS
API Key	购买 Token Plan 后获取的专属 API Key
API 协议	选择 OpenAI
Base URL	https://api.lkeap.cloud.tencent.com/plan/v3

## 步骤五：连通测试

填写完成后，单击**连通测试**，系统会按当前配置真实发起一次请求，验证接口是否可用。当状态显示为**测试成功**时，说明配置正确。

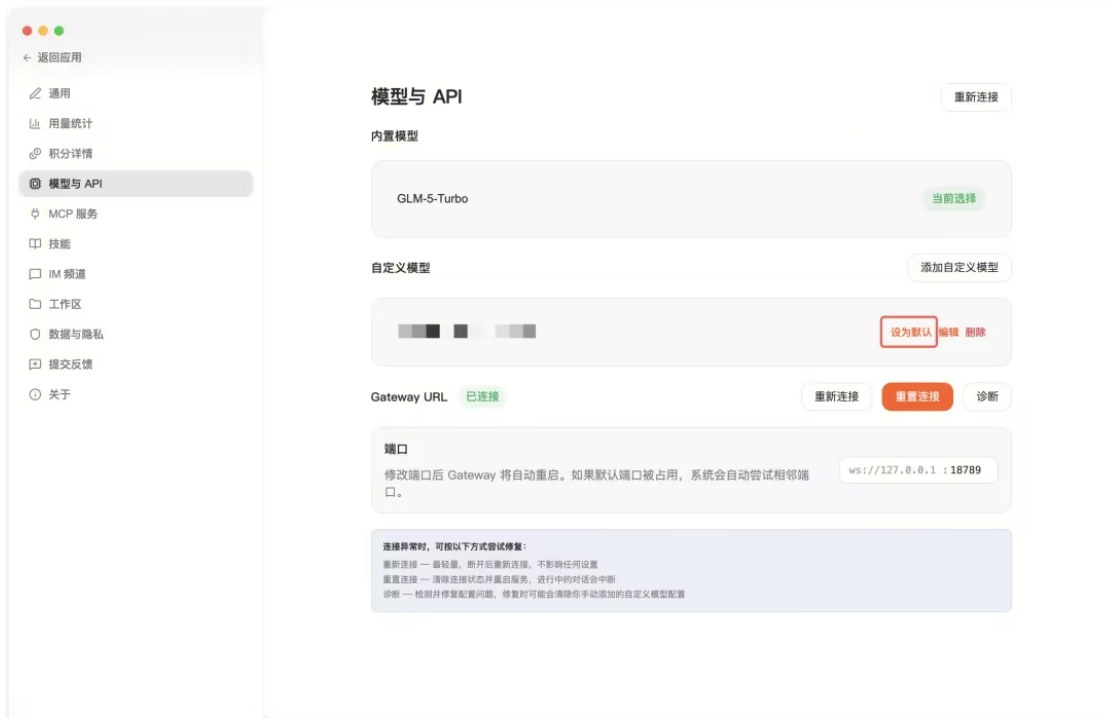


## 步骤六：添加模型

连通测试成功后，单击**添加**，完成模型的添加。重复第四步至第六步，逐一添加其余模型。

## 步骤七：设为默认模型

返回**模型与 API** 页面，在**自定义模型列表**中找到刚添加的模型，单击**设为默认**，即可将该模型设为默认使用的模型。



# WorkBuddy

最近更新时间：2026-04-03 16:27:22

WorkBuddy 是腾讯云推出的全场景桌面 AI 智能体，适配 QQ / 企微生态，支持本地操作电脑、多模型切换，以 Token 配额套餐模式提供商业化服务，兼顾个人与团队使用。本文介绍如何在 WorkBuddy 中配置与使用腾讯云大模型 Token Plan。

WorkBuddy 支持通过本地配置文件接入自定义模型。通过编辑 `models.json` 配置文件，可将腾讯云 Token Plan 的模型接入 WorkBuddy。

## 前置条件

在开始配置前，请确保您已完成以下准备工作：

1. [订阅 Token Plan 套餐](#)。
2. [获取 API Key 和 模型列表](#)。

## 操作步骤

以下步骤以 macOS 系统为例：

### 步骤一：打开终端

使用快捷键 `Cmd + Space` 打开 Spotlight，输入 `Terminal` 并回车。

### 步骤二：创建配置目录（如不存在）

```
mkdir -p ~/.workbuddy
```

### 步骤三：修改 `models.json` 配置文件

使用您喜欢的编辑器打开/新建该文件，例如：

```
# 使用 VS Code
code ~/.workbuddy/models.json

# 或使用 vim
vim ~/.workbuddy/models.json

# 或使用 nano
nano ~/.workbuddy/models.json
```

### 步骤四：写入配置内容

将以下内容粘贴到 `models.json` 文件中:

**注意:**

请您将 `your-token-plan-api-key` 字段替换为 Token Plan 专属的 API Key。

```
{
  "models": [
    {
      "id": "tc-code-latest",
      "name": "Auto",
      "vendor": "OpenAI",
      "url":
"https://api.lkeap.cloud.tencent.com/plan/v3/chat/completions",
      "apiKey": "your-token-plan-api-key",
      "maxInputTokens": 100000,
      "maxOutputTokens": 4096
    },
    {
      "id": "hunyuan-2.0-instruct",
      "name": "Tencent HY 2.0 Instruct",
      "vendor": "OpenAI",
      "url":
"https://api.lkeap.cloud.tencent.com/plan/v3/chat/completions",
      "apiKey": "your-token-plan-api-key",
      "maxInputTokens": 100000,
      "maxOutputTokens": 4096
    },
    {
      "id": "hunyuan-2.0-thinking",
      "name": "Tencent HY 2.0 Thinking",
      "vendor": "OpenAI",
      "url":
"https://api.lkeap.cloud.tencent.com/plan/v3/chat/completions",
      "apiKey": "your-token-plan-api-key",
      "maxInputTokens": 100000,
      "maxOutputTokens": 4096
    },
    {
      "id": "minimax-m2.5",
```

```
    "name": "MiniMax-M2.5",
    "vendor": "OpenAI",
    "url":
"https://api.lkeap.cloud.tencent.com/plan/v3/chat/completions",
    "apiKey": "your-token-plan-api-key",
    "maxInputTokens": 100000,
    "maxOutputTokens": 4096
  },
  {
    "id": "kimi-k2.5",
    "name": "Kimi-K2.5",
    "vendor": "OpenAI",
    "url":
"https://api.lkeap.cloud.tencent.com/plan/v3/chat/completions",
    "apiKey": "your-token-plan-api-key",
    "maxInputTokens": 100000,
    "maxOutputTokens": 4096
  },
  {
    "id": "glm-5",
    "name": "GLM-5",
    "vendor": "OpenAI",
    "url":
"https://api.lkeap.cloud.tencent.com/plan/v3/chat/completions",
    "apiKey": "your-token-plan-api-key",
    "maxInputTokens": 100000,
    "maxOutputTokens": 4096
  },
  {
    "id": "hunyuan-t1",
    "name": "Hunyuan-T1",
    "vendor": "OpenAI",
    "url":
"https://api.lkeap.cloud.tencent.com/plan/v3/chat/completions",
    "apiKey": "your-token-plan-api-key",
    "maxInputTokens": 100000,
    "maxOutputTokens": 4096
  },
  {
    "id": "hunyuan-turbos",
```

```
      "name": "Hunyuan-TurboS",
      "vendor": "OpenAI",
      "url":
"https://api.lkeap.cloud.tencent.com/plan/v3/chat/completions",
      "apiKey": "your-token-plan-api-key",
      "maxInputTokens": 100000,
      "maxOutputTokens": 4096
    }
  ],
  "availableModels": [
    "tc-code-latest",
    "hunyuan-2.0-instruct",
    "hunyuan-2.0-thinking",
    "minimax-m2.5",
    "kimi-k2.5",
    "glm-5",
    "hunyuan-t1",
    "hunyuan-turbos"
  ]
}
```

## 步骤五：保存并重启 WorkBuddy

保存文件后，重启 WorkBuddy 应用使配置生效。

# CoPaw

最近更新时间：2026-04-03 16:27:22

CoPaw 是阿里云通义实验室推出的开源国产 AI 智能体。CoPaw 支持通过“自定义提供商”功能接入任何兼容 OpenAI 协议的第三方模型服务。通过以下配置，可将腾讯云 Token Plan 的模型接入 CoPaw。本文介绍如何在 CoPaw 中配置与使用腾讯云大模型 Token Plan。

## 前置条件

在开始配置前，请确保您已完成以下准备工作：

1. [订阅 Token Plan 套餐](#)。
2. [获取 API Key 和 模型列表](#)。

## 操作步骤

### 步骤一：启动 CoPaw 并进入模型配置页面

启动 CoPaw 应用后，在浏览器中访问 `http://127.0.0.1:8088`（默认端口），单击左侧导航栏的 **设置 > 模型** 进入模型配置页面。

### 步骤二：添加自定义提供商

在模型配置页面中，单击 **添加提供商 > 设置**，填写以下信息，然后单击 **创建**。

参数	值
提供商 ID	<code>tencent-token-plan</code>
显示名称	腾讯云 Token Plan

创建成功后，您可以在页面上看到新添加的提供商卡片。

### 步骤三：配置提供商

单击新添加的提供商卡片上的 **设置**，填写以下连接参数：

参数	值
Base URL	<code>https://api.lkeap.cloud.tencent.com/plan/v3</code>
API Key	购买 Token Plan 后获取的专属 API Key

填写完成后保存设置。

## 步骤四：添加模型

单击自定义提供商卡片中的模型，在模型 ID 输入框中填写模型 ID，单击添加模型。支持如下模型：

模型显示名称	模型 ID
Auto	tc-code-latest
Tencent HY 2.0 Instruct	hunyuan-2.0-instruct
Tencent HY 2.0 Think	hunyuan-2.0-thinking
MiniMax-M2.5	minimax-m2.5
Kimi-K2.5	kimi-k2.5
GLM-5	glm-5
Hunyuan-T1	hunyuan-t1
Hunyuan-TurboS	hunyuan-turbos

逐一添加需要使用的模型。

## 步骤五：选择模型并使用

在页面上方的 LLM 配置区域，将提供商选择为刚添加的自定义提供商，选择对应的模型，单击保存即可开始对话。

# Lighthouse OpenClaw

最近更新时间：2026-04-08 15:34:51

腾讯云轻量应用服务器（Lighthouse）上部署的 OpenClaw（Clawdbot）官方一键部署镜像。本文介绍如何在 Lighthouse OpenClaw 中配置与使用腾讯云大模型 Token Plan。

## 前置条件

在开始配置前，请确保您已完成以下准备工作：

1. 订阅 [Token Plan 套餐](#)。
2. 获取 [API Key](#) 和 [模型列表](#)。

## 操作步骤

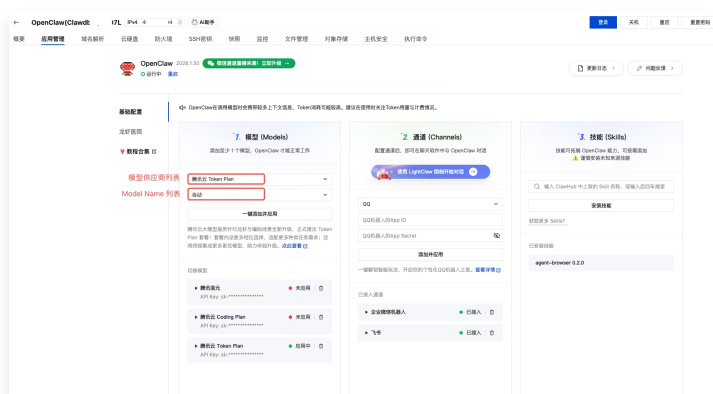
参考 [腾讯云轻量应用服务器 OpenClaw 部署](#) 指引，完成服务器的购买、通道和技能的配置。其中模型配置可以参考下文指引，选用 [腾讯云 Token Plan](#) 的模型服务：

### 步骤一：进入应用管理

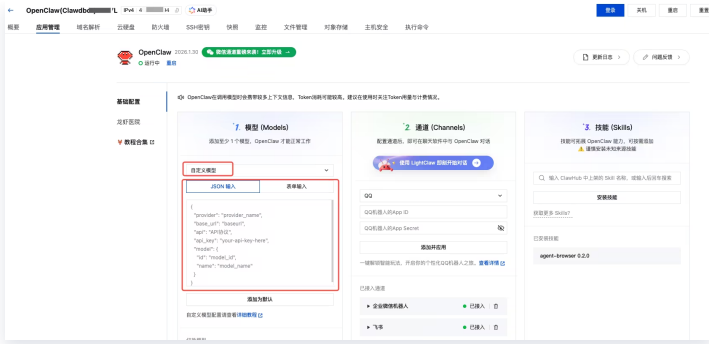
登录 [腾讯云轻量应用服务器控制台](#)，找到已部署 OpenClaw 的实例，单击实例名称进入实例详情页面后，选择应用管理页签。

### 步骤二：配置模型

- 如果您的账号同时购买 [Token Plan 套餐](#)和[腾讯云轻量应用服务器](#)：可以参考下图配置 [Token Plan](#) 模型。在应用管理界面中找到模型（Models）配置区域：
  - 模型供应商列表选择[腾讯云 Token Plan](#)。
  - Model Name 列表选择具体的模型。



- 如果您的 [Token Plan 套餐](#)和[腾讯云轻量应用服务器](#)不在同一个账号：可以通过自定义模型配置 [Token Plan](#) 模型。在应用管理界面中找到模型（Models）配置区域：
  - 模型供应商列表选择[自定义模型](#)。
  - [JSON](#) 输入区域填入不同模型的配置。[JSON](#) 配置信息详情请参见 [各模型 JSON 配置信息](#)。



**注意:**

请您将 `your-token-plan-api-key` 字段替换为您购买 Token Plan 后获取的专属 API Key。每个模型需单独添加一次。

## 各模型 JSON 配置信息

### 配置 Auto

```
{
  "provider": "tencent-token-plan",
  "base_url": "https://api.lkeap.cloud.tencent.com/plan/v3",
  "api": "openai-completions",
  "api_key": "your-token-plan-api-key",
  "model": {
    "id": "tc-code-latest",
    "name": "Auto"
  }
}
```

### 配置 Tencent HY 2.0 Instruct

```
{
  "provider": "tencent-token-plan",
  "base_url": "https://api.lkeap.cloud.tencent.com/plan/v3",
  "api": "openai-completions",
  "api_key": "your-token-plan-api-key",
  "model": {
    "id": "hunyuan-2.0-instruct",
    "name": "Tencent HY 2.0 Instruct"
  }
}
```

```
}
```

## 配置 Tencent HY 2.0 Think

```
{
  "provider": "tencent-token-plan",
  "base_url": "https://api.lkeap.cloud.tencent.com/plan/v3",
  "api": "openai-completions",
  "api_key": "your-token-plan-api-key",
  "model": {
    "id": "hunyuan-2.0-thinking",
    "name": "Tencent HY 2.0 Think"
  }
}
```

## 配置 MiniMax-M2.5

```
{
  "provider": "tencent-token-plan",
  "base_url": "https://api.lkeap.cloud.tencent.com/plan/v3",
  "api": "openai-completions",
  "api_key": "your-token-plan-api-key",
  "model": {
    "id": "minimax-m2.5",
    "name": "MiniMax-M2.5"
  }
}
```

## 配置 Kimi-K2.5

```
{
  "provider": "tencent-token-plan",
  "base_url": "https://api.lkeap.cloud.tencent.com/plan/v3",
  "api": "openai-completions",
  "api_key": "your-token-plan-api-key",
  "model": {
    "id": "kimi-k2.5",
    "name": "Kimi-K2.5"
  }
}
```

```
}  
}
```

## 配置 GLM-5

```
{  
  "provider": "tencent-token-plan",  
  "base_url": "https://api.lkeap.cloud.tencent.com/plan/v3",  
  "api": "openai-completions",  
  "api_key": "your-token-plan-api-key",  
  "model": {  
    "id": "glm-5",  
    "name": "GLM-5"  
  }  
}
```

## 配置 Hunyuan-T1

```
{  
  "provider": "tencent-token-plan",  
  "base_url": "https://api.lkeap.cloud.tencent.com/plan/v3",  
  "api": "openai-completions",  
  "api_key": "your-token-plan-api-key",  
  "model": {  
    "id": "hunyuan-t1",  
    "name": "Hunyuan-T1"  
  }  
}
```

## 配置 Hunyuan-TurboS

```
{  
  "provider": "tencent-token-plan",  
  "base_url": "https://api.lkeap.cloud.tencent.com/plan/v3",  
  "api": "openai-completions",  
  "api_key": "your-token-plan-api-key",  
  "model": {  
    "id": "hunyuan-turbos",  
  }  
}
```

```
"name": "Hunyuan-TurboS"  
}  
}
```

# 接入编程工具

## CodeBuddy Code

最近更新时间：2026-04-03 16:27:22

CodeBuddy Code 是基于腾讯云 AI 技术的智能编程工具，深度集成腾讯云生态，提供从代码编写到项目部署的全链路 AI 辅助。本文介绍如何在 CodeBuddy Code 中配置与使用腾讯云大模型 Token Plan。

### 前置条件

在开始配置前，请确保您已完成以下准备工作：

1. [订阅 Token Plan 套餐](#)。
2. [获取 API Key 和 模型列表](#)。

### 安装 CodeBuddy Code

请先确保本地已安装 Node.js 18.0 及以上版本，再执行以下命令安装 CodeBuddy Code。

```
npm install -g @tencent-ai/codebuddy-code
```

安装完成后，您可以执行如下命令检查，输出版本号则表示安装成功。

```
codebuddy --version
```

### 配置 Token Plan

1. 找到 CodeBuddy Code 的配置文件，配置文件路径如下，配置文件不存在时新建即可。

- Windows: `C:\Users\<用户名>\.codebuddy\models.json`。
- macOS/Linux: `~/codebuddy/models.json`。

2. 复制粘贴如下配置，并保存。

#### ⓘ 说明：

请您将代码中的 `<USER_API_KEY>` 替换为 Token Plan 专属的 API Key。

```
{  
  "models": [  
    {  
      "id": "tc-code-latest",
```

```
"name": "Auto",
"vendor": "Tencent Cloud",
"apiKey": "<USER_API_KEY>",
"url": "https://api.lkeap.cloud.tencent.com/plan/v3"
},
{
  "id": "hunyuan-2.0-instruct",
  "name": "Tencent HY 2.0 Instruct",
  "vendor": "Tencent Cloud",
  "apiKey": "<USER_API_KEY>",
  "url": "https://api.lkeap.cloud.tencent.com/plan/v3"
},
{
  "id": "hunyuan-2.0-thinking",
  "name": "Tencent HY 2.0 Think",
  "vendor": "Tencent Cloud",
  "apiKey": "<USER_API_KEY>",
  "url": "https://api.lkeap.cloud.tencent.com/plan/v3"
},
{
  "id": "hunyuan-t1",
  "name": "Hunyuan-T1",
  "vendor": "Tencent Cloud",
  "apiKey": "<USER_API_KEY>",
  "url": "https://api.lkeap.cloud.tencent.com/plan/v3"
},
{
  "id": "hunyuan-turbos",
  "name": "Hunyuan-TurboS",
  "vendor": "Tencent Cloud",
  "apiKey": "<USER_API_KEY>",
  "url": "https://api.lkeap.cloud.tencent.com/plan/v3"
},
{
  "id": "minimax-m2.5",
  "name": "MiniMax-M2.5",
  "vendor": "Tencent Cloud",
  "apiKey": "<USER_API_KEY>",
  "url": "https://api.lkeap.cloud.tencent.com/plan/v3"
},
```

```
{
  "id": "kimi-k2.5",
  "name": "Kimi-K2.5",
  "vendor": "Tencent Cloud",
  "apiKey": "<USER_API_KEY>",
  "url": "https://api.lkeap.cloud.tencent.com/plan/v3"
},
{
  "id": "glm-5",
  "name": "GLM-5",
  "vendor": "Tencent Cloud",
  "apiKey": "<USER_API_KEY>",
  "url": "https://api.lkeap.cloud.tencent.com/plan/v3"
}
],
"availableModels": ["tc-code-latest", "hunyuan-2.0-instruct", "hunyuan-2.0-thinking", "hunyuan-t1", "hunyuan-turbos", "minimax-m2.5", "kimi-k2.5", "glm-5"]
}
```

## 开始使用

配置完成后执行如下命令启动 CodeBuddy Code。

```
codebuddy
```

首次使用 CodeBuddy Code 时，您需要完成登录认证。启动后会显示登录方式选择界面：

```
Select login method:
> Log in via Chinese Site
  Log in via International Site
  Log in via Enterprise Domain
  Log in via iOA (Tencent only)
```

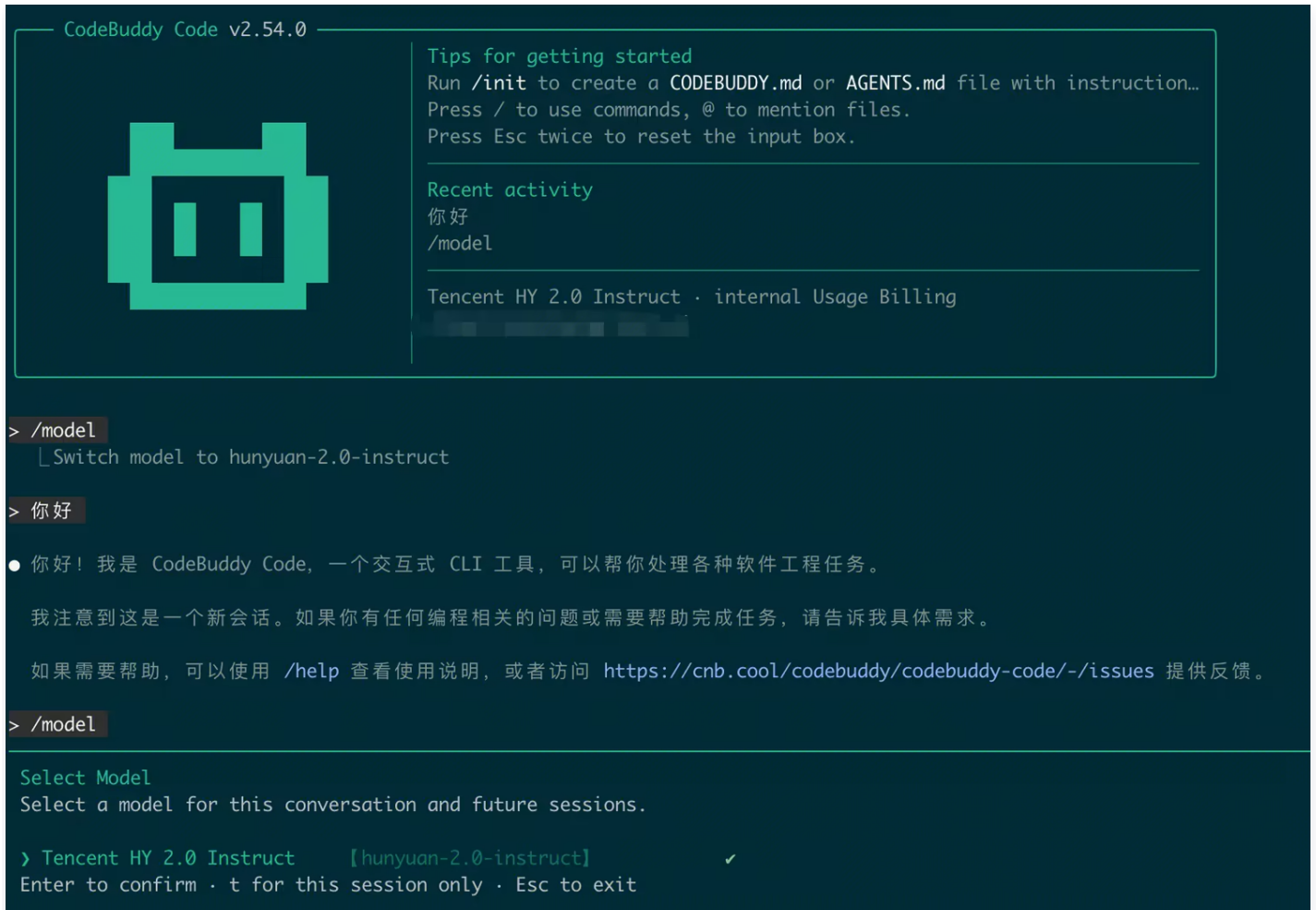
登录方式说明如下：

登录方式	适用场景	说明
Chinese Site	中国站用户	通过腾讯云中国站 (copilot.tencent.com) 进行认证，支持境内主流模型。

International Site	国际站用户	通过腾讯云国际站 (codebuddy.ai) 进行认证, 支持境外主流模型。
Enterprise Domain	专享版/私有化部署	连接企业专享版或自建的 CodeBuddy 服务, 需要输入企业提供的服务地址。
iOA	腾讯内部员工	通过腾讯 iOA 零信任系统进行认证, 仅限腾讯内部员工使用。

使用 ↑↓ 键选择登录方式, 按 Enter 确认后会自动打开浏览器完成认证。

输入 `/model`, 选择模型, 下图以 hunyuan-2.0-instruct 为例。



```
CodeBuddy Code v2.54.0

Tips for getting started
Run /init to create a CODEBUDDY.md or AGENTS.md file with instruction..
Press / to use commands, @ to mention files.
Press Esc twice to reset the input box.

Recent activity
你好
/model

Tencent HY 2.0 Instruct · internal Usage Billing

> /model
  Switch model to hunyuan-2.0-instruct

> 你好

● 你好! 我是 CodeBuddy Code, 一个交互式 CLI 工具, 可以帮你处理各种软件工程任务。

我注意到这是一个新会话。如果你有任何编程相关的问题或需要帮助完成任务, 请告诉我具体需求。

如果需要帮助, 可以使用 /help 查看使用说明, 或者访问 https://cnb.cool/codebuddy/codebuddy-code/-/issues 提供反馈。

> /model

Select Model
Select a model for this conversation and future sessions.

> Tencent HY 2.0 Instruct [hunyuan-2.0-instruct] ✓
Enter to confirm · t for this session only · Esc to exit
```

# OpenCode

最近更新时间：2026-04-03 16:27:22

OpenCode 是一款开源的 AI 编程代理工具，提供终端界面、桌面应用和 IDE 扩展等多种使用方式。本文介绍如何在 OpenCode 中配置与使用腾讯云大模型 Token Plan。

## 前置条件

在开始配置前，请确保您已完成以下准备工作：

1. [订阅 Token Plan 套餐](#)。
2. [获取 API Key 和 模型列表](#)。

## 安装 OpenCode

执行如下命令安装 OpenCode，更多操作可参考 [OpenCode 官方配置指引](#)。

```
npm install -g opencode-ai
```

安装完成后，您可以执行如下命令检查，输出版本号则表示安装成功。

```
opencode -v
```

## 配置 Token Plan

您需要按照下面的示例修改 OpenCode 的配置文件，配置文件路径如下，配置文件不存在时新建即可。

- macOS / Linux: `~/.config/opencode/opencode.json`。
- Windows: `C:\Users\<用户名>\.config\opencode\opencode.json`。

### 说明：

请您将 `<USER_API_KEY>` 替换为 Token Plan 专属的 API Key。

```
{
  "$schema": "https://opencode.ai/config.json",
  "provider": {
    "myprovider": {
      "npm": "@ai-sdk/openai-compatible",
      "name": "lkeap",
      "options": {
        "baseUrl": "https://api.lkeap.cloud.tencent.com/plan/v3",
```

```
"apiKey": "<USER_API_KEY>"
},
"models": {
  "tc-code-latest": {
    "name": "Auto",
    "modalities": {
      "input": [
        "text"
      ],
      "output": [
        "text"
      ]
    },
    "options": {
      "thinking": {
        "type": "enabled",
        "budgetTokens": 8192
      }
    },
    "limit": {
      "context": 196608,
      "output": 32768
    }
  },
  "hunyuan-2.0-instruct": {
    "name": "Tencent HY 2.0 Instruct",
    "modalities": {
      "input": [
        "text"
      ],
      "output": [
        "text"
      ]
    },
    "options": {
      "thinking": {
        "type": "enabled",
        "budgetTokens": 8192
      }
    }
  },
}
```

```
"limit": {
  "context": 128000,
  "output": 16000
},
"hunyuan-2.0-thinking": {
  "name": "Tencent HY 2.0 Think",
  "modalities": {
    "input": [
      "text"
    ],
    "output": [
      "text"
    ]
  },
  "options": {
    "thinking": {
      "type": "enabled",
      "budgetTokens": 8192
    }
  },
  "limit": {
    "context": 128000,
    "output": 32000
  }
},
"hunyuan-t1": {
  "name": "Hunyuan-T1",
  "modalities": {
    "input": [
      "text"
    ],
    "output": [
      "text"
    ]
  },
  "options": {
    "thinking": {
      "type": "enabled",
      "budgetTokens": 8192
    }
  }
}
```

```
    }
  },
  "limit": {
    "context": 64000,
    "output": 32000
  }
},
"hunyuan-turbos": {
  "name": "hunyuan-turbos",
  "modalities": {
    "input": [
      "text"
    ],
    "output": [
      "text"
    ]
  },
  "options": {
    "thinking": {
      "type": "enabled",
      "budgetTokens": 8192
    }
  },
  "limit": {
    "context": 32000,
    "output": 16000
  }
},
"minimax-m2.5": {
  "name": "MiniMax-M2.5",
  "modalities": {
    "input": [
      "text"
    ],
    "output": [
      "text"
    ]
  },
  "options": {
    "thinking": {
```

```
        "type": "enabled",
        "budgetTokens": 8192
      }
    },
    "limit": {
      "context": 196608,
      "output": 32768
    }
  },
  "kimi-k2.5": {
    "name": "Kimi-K2.5",
    "modalities": {
      "input": [
        "text"
      ],
      "output": [
        "text"
      ]
    },
    "options": {
      "thinking": {
        "type": "enabled",
        "budgetTokens": 8192
      }
    },
    "limit": {
      "context": 262144,
      "output": 32768
    }
  },
  "glm-5": {
    "name": "GLM-5",
    "modalities": {
      "input": [
        "text"
      ],
      "output": [
        "text"
      ]
    },
  },
```

```
"options": {
  "thinking": {
    "type": "enabled",
    "budgetTokens": 8192
  }
},
"limit": {
  "context": 202752,
  "output": 16384
}
}
}
}
}
```

## 使用 OpenCode

执行如下命令启动 OpenCode:

```
opencode
```

输入 `/models`，选择配置的模型并在 OpenCode 中使用。



完成上述配置后，即可开始使用 OpenCode。

你好

*Thinking:* 用户用中文说"你好", 这是一个简单的问候。  
根据我的指令, 我应该简洁地回应, 不要有多余的开场白或结束语。  
我应该直接回应问候。

由于用户用中文打招呼, 我也应该用中文回应。  
但根据指令, 我应该保持简洁, 不超过4行文字 (不包括工具使用)。

对于简单的问候, 我不需要使用任何工具, 只需要简单地回应即可。

你好! 有什么我可以帮你的吗?

■ Build · glm-5 · 6.2s

Greeting in Chinese conversation

Context

10,625 tokens

0% used

\$0.00 spent

LSP

LSPs will activate as files are read

█

Build GLM-5 lkeap

/~

tab agents ctrl+p commands

• OpenCode 1.2.15

# Claude Code

最近更新时间：2026-04-03 16:27:22

Claude Code 是一款开源的 AI 编程助手，支持通过 Anthropic API 接入模型，帮助开发者快速编写、调试和管理代码。

Token Plan 中的模型支持 Anthropic API 兼容接口，可以通过 Claude Code 调用。

## 前置条件

在开始配置前，请确保您已完成以下准备工作：

1. [订阅 Token Plan 套餐](#)。
2. [获取 API Key 和 模型列表](#)。

## 安装 Claude Code

### ⚠ 注意：

在 Windows 上使用 Claude Code，需要提前安装 [WSL](#) 或 [Git for Windows](#)。

执行如下命令安装 Claude Code，更多操作可参考 [Claude Code 安装指引](#)。

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

安装结束后，执行以下命令查看安装结果，若显示版本号则安装成功。

```
claude --version
```

## 配置 Token Plan

在 Claude Code 中接入 Token Plan，需要配置以下信息：

- ANTHROPIC\_BASE\_URL：设置为 `https://api.lkeap.cloud.tencent.com/plan/anthropic`。
- ANTHROPIC\_AUTH\_TOKEN：设置为 Token Plan 专属 API Key。
- ANTHROPIC\_MODEL：设置为 Token Plan 支持的模型。

配置步骤如下：

1. 编辑或新增 settings.json 文件，需要修改或添加的配置信息如下：

- <USER\_API\_KEY>：替换为您自己的 API Key。
- <Model\_Name>：更新为需要使用的模型信息，例如 glm-5。模型信息可参见 [Token Plan 支持的 Model Name](#)。

**⚠ 注意:**

不同系统配置文件路径不同，具体如下：

- MacOS / Linux: `~/.claude/settings.json`。
- Windows: `C:\Users\<<用户名>\.claude\settings.json`。

```
{
  "env": {
    "ANTHROPIC_AUTH_TOKEN": "<USER_API_KEY>",
    "ANTHROPIC_BASE_URL":
"https://api.lkeap.cloud.tencent.com/plan/anthropic",
    "ANTHROPIC_MODEL": "<Model_Name>"
  }
}
```

2. 编辑或新增 `.claude.json` 文件，修改或新增 `hasCompletedOnboarding` 字段值为 `true`。

**⚠ 注意:**

不同系统配置文件路径不同，具体如下：

- MacOS / Linux: `~/.claude.json`
- Windows: `C:\Users\<<用户名>\.claude.json`

```
{
  "hasCompletedOnboarding": true
}
```

3. 保存配置文件后，在新的终端窗口执行后续命令。

## 在命令行中使用 Claude Code

通过命令行进入项目目录后，执行 `claude` 命令启动 Claude Code 即可开始使用。

```
cd my-project
claude
```

选择 **Yes, I trust this folder** 信任当前目录。

```
Accessing workspace:
/Users/pengweidou/CodeProjects/atomic

Quick safety check: Is this a project you created or one you trust? (Like your
own code, a well-known open source project, or work from your team). If not,
take a moment to review what's in this folder first.

Claude Code'll be able to read, edit, and execute files here.

Security guide
> 1. Yes, I trust this folder
   2. No, exit

Enter to confirm · Esc to cancel
```

```
Claude Code v2.1.63

Welcome back!



glm-5 · API Usage Billing
~/CodeProjects/atomic

Tips for getting started
Run /init to create a CLAUDE.md file with i...

Recent activity
No recent activity

/model to try Opus 4.6

> try "refactor <filepath>"

? for short... Claude Code has switched from npm to native installer. Run `c...
Auto-updating...
```

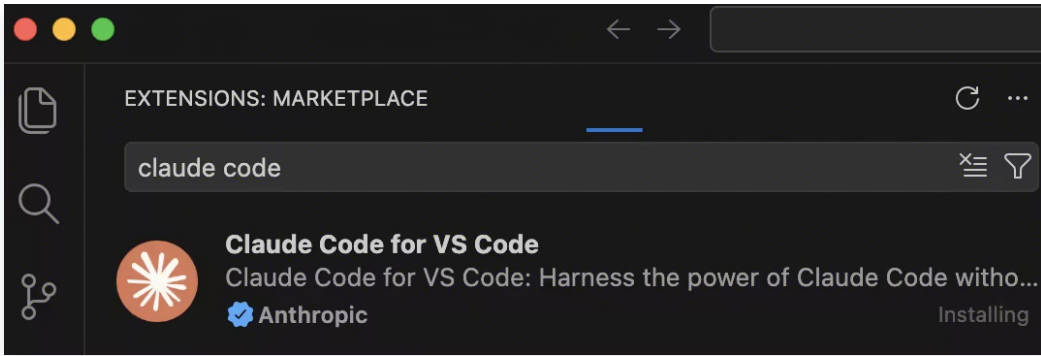
## 在 IDE 中使用 Claude Code

### ⚠ 注意:

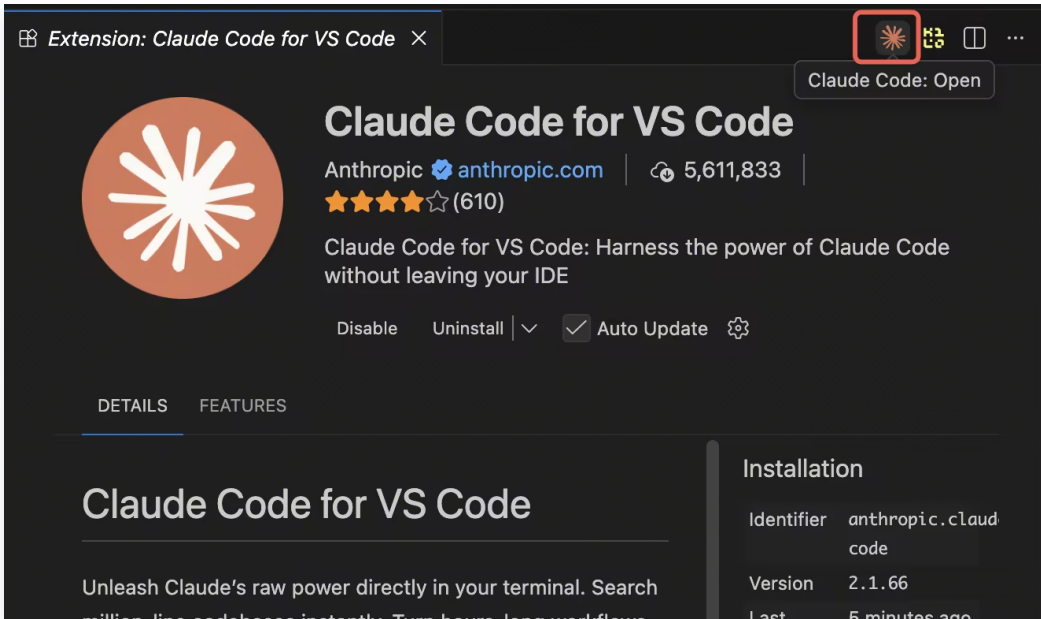
Claude Code IDE 插件依赖 Claude Code CLI 工具，需先完成 Claude Code 的安装及配置。

### VS Code

1. 在扩展市场搜索 Claude Code 进行安装。



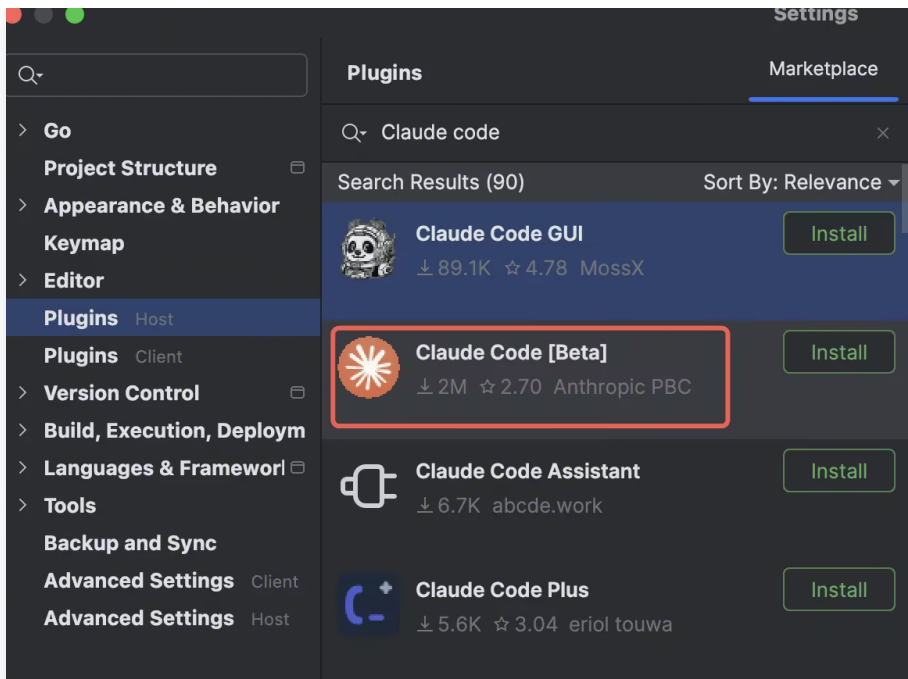
2. 安装成功后，单击 VS Code 右上角的 Claude Code 图标，进入 Claude Code 页面。



3. 切换模型：在对话框中输入 `/`，选择 **General config** 进入设置页面，在 **Selected Model** 中选择支持的模型，新建一个新窗口开始对话。

## JetBrains

1. 在扩展市场中搜索 Claude Code 并安装。



2. 安装后重启 IDE，单击右上角图标即可使用，您可以通过 `/model <模型名称>` 命令切换模型。



## 更多信息

如需进一步了解 Claude Code 的更多功能，请参考 [Claude Code 官方文档](#)。

# Codex

最近更新时间：2026-04-03 16:27:22

Codex 是由 OpenAI 开发的新一代 AI 智能编程助手，基于 GPT 架构打造强大代码生成引擎。Codex 通过深度学习技术，能够理解自然语言并转换为高质量代码，支持 Python、JavaScript、Java 等多种编程语言。本文介绍如何在 Codex 中配置与使用腾讯云大模型 Token Plan。

## 前置条件

在开始配置前，请确保您已完成以下准备工作：

1. [订阅 Token Plan 套餐](#)。
2. [获取 API Key 和 模型列表](#)。

## 安装 Codex

### ⚠ 注意：

新版本 Codex 采用 Responses API 接口，暂不支持 Chat/Completions API，只有 **0.80.0 及以下** 版本才支持（详情请参见 [Codex 官方说明](#)）。请安装支持 Chat/Completions API 的旧版本 Codex，例如：0.80.0 版本。

在终端中执行以下命令安装 Codex。

```
npm install -g @openai/codex@0.80.0
```

在终端中执行以下命令，若输出版本号，则表示安装成功。

```
codex --version
```

## 配置 Token Plan

完整复制以下内容并粘贴到 Codex 配置文件 `~/.codex/config.toml` 中。

1. 找到 Codex 的配置文件，配置文件路径如下，配置文件不存在时新建即可。
  - macOS/Linux: `~/.codex/config.toml`。
  - Windows: `C:\Users\<用户名>\.codex\config.toml`。
2. 复制以下内容粘贴到 config.toml 配置文件中，将 `<MODEL_NAME>` 替换为 Token Plan 支持的 [Model Name](#)。

```
model_provider = "Lkeap-Token-Plan"
```

```
model = "<MODEL_NAME>"
[model_providers.Lkeap_Token_Plan]
name = "Lkeap_Token_Plan"
base_url = "https://api.lkeap.cloud.tencent.com/plan/v3"
env_key = "TP_API_KEY"
wire_api = "chat"
```

## 配置环境变量

在命令行中将 TP\_API\_KEY 环境变量设置为 Token Plan 专属的 API Key。

### macOS/Linux

在终端中执行以下命令，查看默认 Shell 类型。

```
echo $SHELL
```

根据 Shell 类型设置环境变量，命令如下：

#### Zsh

```
echo 'export TP_API_KEY="USER_API_KEY"' >> ~/.zshrc
```

#配置生效

```
source ~/.zshrc
```

#### Bash

```
echo 'export TP_API_KEY="USER_API_KEY"' >> ~/.bash_profile
```

#配置生效

```
source ~/.bash_profile
```

## Windows

### CMD

```
# 用您的Token Plan API Key 替换 USER_API_KEY
setx TP_API_KEY "USER_API_KEY"

#新开 CMD 窗口检查是否生效
echo %TP_API_KEY%
```

### PowerShell

```
# 用 Token Plan 的 API Key 替换 USER_API_KEY
[Environment]::SetEnvironmentVariable("TP_API_KEY",
"USER_API_KEY", [EnvironmentVariableTarget]::User)

#新开一个窗口验证是否生效
echo $env:TP_API_KEY
```

## 使用 Codex

1. 在终端执行如下命令并回车，启动 Codex 工具。

```
codex
```

## 2. 选择 Skip 跳过。

```
🌟 Update available! 0.80.0 -> 0.107.0
Release notes: https://github.com/openai/codex/releases/latest
1. Update now (runs 'npm install -g @openai/codex')
2. Skip
3. Skip until next version
Press enter to continue
```

## 3. 选择采用 API Key 的方式使用 Codex，输入您套餐的 API Key。

## 4. 您可以输入问题或指令，开始对话。

```
> OpenAI Codex (v0.80.0)
model: gpt-5 /model to change
directory: -

Tip: NEW! Try shell snapshotting to make your Codex faster. Enable in /experimental!
Support for the "chat" size API is deprecated and will soon be removed. Update your model provide "response".

你好

你好！有什么我可以帮你的吗？我可以协助你处理代码、调试问题、解释项目结构等编程相关的任务。
```

# Cline

最近更新时间：2026-04-03 16:27:22

Cline 是一款内置于编辑器和终端的 AI 编码助手。它可以读写文件、运行终端命令、使用浏览器，并通过自然对话帮助您构建功能。本文介绍如何在 Cline 中配置与使用腾讯云大模型 Token Plan。

## 前置条件

在开始配置前，请确保您已完成以下准备工作：

1. [订阅 Token Plan 套餐](#)。
2. [获取 API Key 和 模型列表](#)。

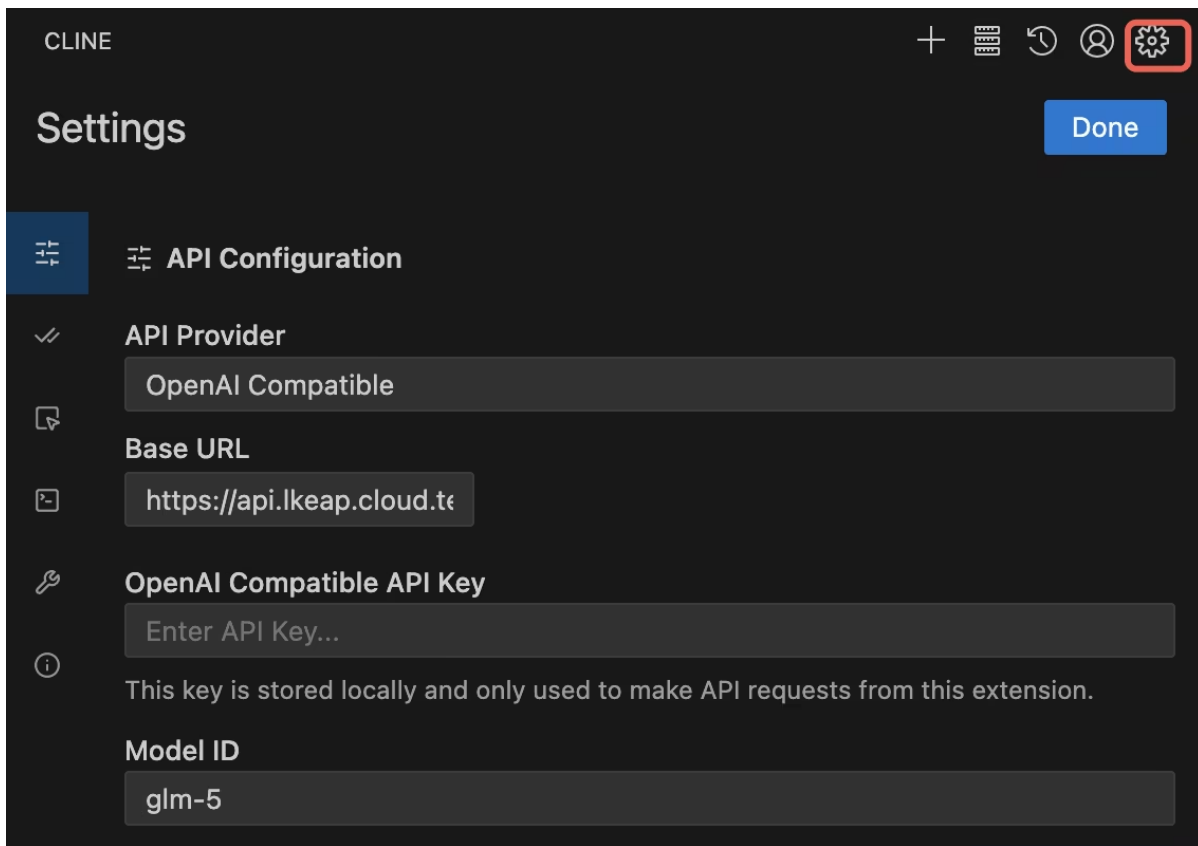
## 安装 Cline

打开 VSCode，在扩展市场搜索并安装 Cline，详情可参考 [Cline 官方安装指引](#)。

## 配置 Token Plan

Cline 插件安装完成后，单击右上角的设置按钮，配置以下信息：

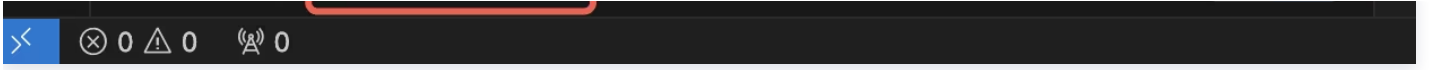
- API Provider：选择 **OpenAI Compatible**（Token Plan 接口兼容 OpenAI 标准）。
- Base URL：`https://api.lkeap.cloud.tencent.com/plan/v3`。
- API Key：填写 Token Plan 专属的 API Key。
- Model ID：填写 Token Plan 支持的 **Model Name**。



## 使用 Cline

配置完成后，您可以看到对话框下方显示对应的模型配置，并可以通过对话框与模型进行交互。

The screenshot shows the Cline AI interface. At the top, there's a header with 'CLINE' and various icons. Below that, a notification bar shows '\$0.0000' and a progress indicator from '5.0k' to '128.0k'. A green notification box states '1/1 All tasks have been completed!'. Below this, the AI is in a 'Thinking' state. A large green checkmark and the text 'Task Completed' are displayed. The main content area contains a friendly greeting: '你好! 我是 Cline, 一位技术精湛的软件工程师助手。我具备以下能力:' followed by a bulleted list of skills: '编程语言: 熟悉多种编程语言, 包括 Python、JavaScript、TypeScript、Java、C++、Go、Rust 等', '框架与技术: 了解主流框架如 React、Vue、Node.js、Django、Spring Boot 等', '开发工具: 可以使用终端命令、读写文件、搜索代码等工具帮助你完成任务', and '最佳实践: 熟悉设计模式、代码规范和软件工程最佳实践'. Below the list, it says '我可以帮你:' followed by another bulleted list: '编写和修改代码', '调试和解决问题', '重构和优化代码', '创建新项目', and '解释技术概念'. The interface ends with the question '有什么我可以帮助你的吗?' and a 'Start New Task' button. At the bottom, there's a text input field with a placeholder 'Type a message...' and a dropdown menu showing 'openai-compat:glm-5'.



# Cursor

最近更新时间：2026-04-03 16:27:22

Cursor 是基于 VSCode 深度改造、AI 原生的代码编辑器，内置 GPT-4/Claude 等模型，支持自然语言生成、代码理解、跨文件重构、智能调试与自主代理（Agent）。本文介绍如何在 Cursor 中配置与使用腾讯云大模型 Token Plan。

## 前置条件

在开始配置前，请确保您已完成以下准备工作：

1. [订阅 Token Plan 套餐](#)。
2. [获取 API Key 和 模型列表](#)。

## 安装 Cursor

官网下载安装包：通过 [Cursor 官网](#) 下载并安装 Cursor。

### ⚠ 注意：

由于 Cursor 的限制，只有订阅了 Cursor Pro 及以上版本套餐的用户才支持自定义配置模型。

## 配置 Token Plan

Cursor 安装完成后，在 Cursor 中，单击右上角的 **Cursor Settings**，选择 **Models** 模块并进行配置。

**Models** 模块的具体配置如下：

- OpenAI API Key：填写 Token Plan 的专属 API Key。
- Override OpenAI Base URL： `https://api.lkeap.cloud.tencent.com/plan/v3`。
- Add Custom Model：填写 Token Plan 支持的 [Model Name](#)。

## 使用 Cursor

配置完成后，即可在聊天面板中选择配置的模型进行交互。

# Kilo CLI

最近更新时间：2026-04-03 16:27:22

Kilo CLI 是 Kilo Code 全栈式 AI 编程代理平台的命令行形态。本文介绍如何在 Kilo CLI 中配置与使用腾讯云大模型 Token Plan。

## 前置条件

在开始配置前，请确保您已完成以下准备工作：

1. [订阅 Token Plan 套餐](#)。
2. [获取 API Key 和 模型列表](#)。

## 安装 Kilo CLI

在终端执行如下命令安装 Kilo CLI。

```
npm install -g @kilocode/cli
```

安装完成后，您可以执行如下命令检查，输出版本号则表示安装成功。

```
kilo --version
```

## 配置 Token Plan

1. 打开 Kilo CLI 的配置文件，配置文件路径如下：

- macOS/Linux: `~/.config/kilo/config.json`。
- Windows: `C:\Users\<用户名>\.config\kilo\config.json`。

2. 在配置文件中，复制粘贴如下配置，并保存。

### ⚠ 注意：

请您将代码中的 `<USER_API_KEY>` 替换为 Token Plan 专属的 API Key，将 `<MODEL_NAME>` 替换为 Token Plan 支持的 [Model Name](#)。

```
{
  "$schema": "https://kilo.ai/config.json",
  "provider": {
    "lkeap": {
      "npm": "@ai-sdk/anthropic",
```

```
"name": "Lkeap Token Plan",
"options": {
  "baseUrl":
"https://api.lkeap.cloud.tencent.com/plan/anthropic/v1",
  "apiKey": "<USER_API_KEY>"
},
"models": {
  "glm-5": {
    "name": "<MODEL_NAME>",
    "options": {
      "thinking": {
        "type": "enabled",
        "budgetTokens": 1024
      }
    }
  }
}
}
```

## 使用 Kilo CLI

配置完成后执行如下命令启动 Kilo CLI。

```
kilo
```

输入 `/models`，选择 Lkeap Token Plan 分组下的模型。

# Kilo Code

最近更新时间：2026-04-03 16:27:22

Kilo Code 是一款开源 AI 编程助手，全场景适配您的开发工作流，无论在 IDE、终端、浏览器中，还是移动办公场景均可使用。借助深度理解代码库的 AI 能力，高效生成代码、自动化代码评审、调试问题，加速研发交付。本文介绍如何在 Kilo Code 中配置与使用腾讯云大模型 Token Plan。

## 前置条件

在开始配置前，请确保您已完成以下准备工作：

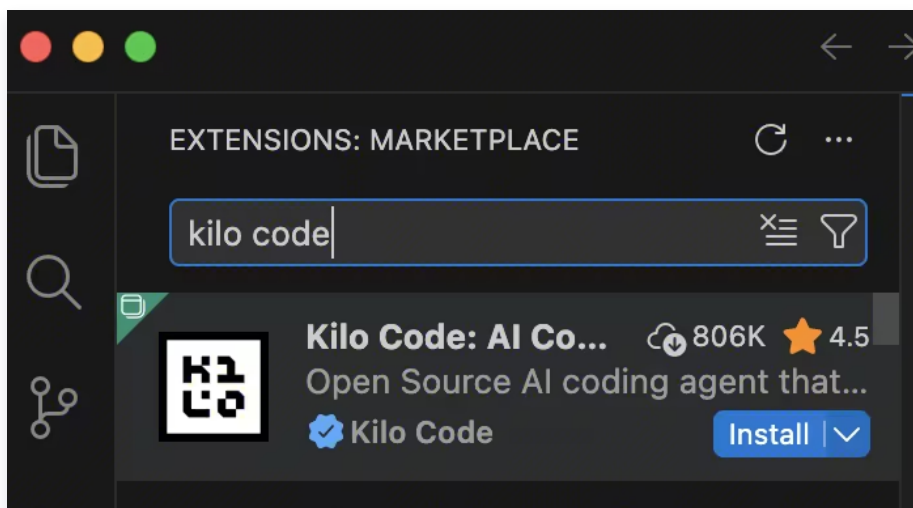
1. [订阅 Token Plan 套餐](#)。
2. [获取 API Key 和 模型列表](#)。

## 安装 Kilo Code 并使用 Token Plan

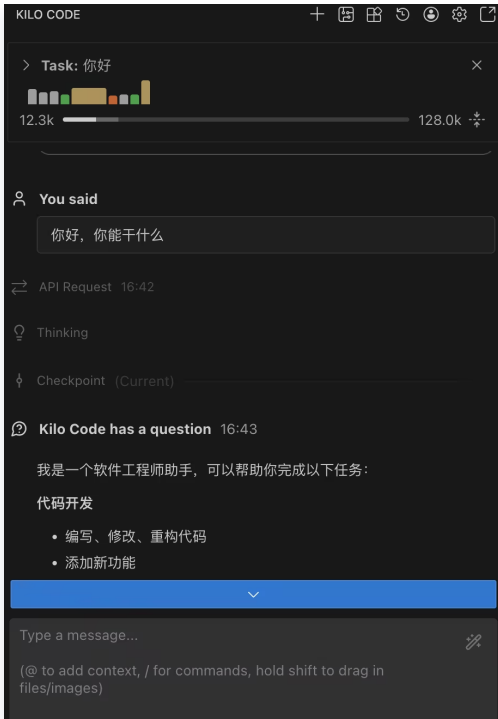
在 IDE 中搜索安装 Kilo Code 插件，下面分别以 JetBrains 和 VS Code 为例展示安装配置过程。

### VSCode

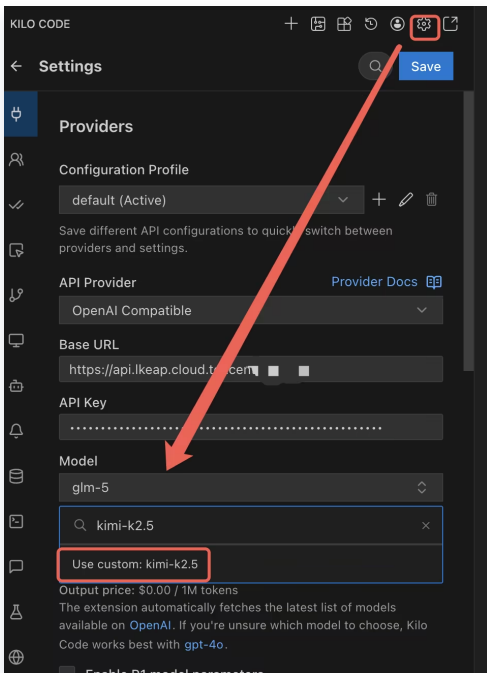
1. 打开 VSCode，在扩展市场中搜索 Kilo Code 并安装。



2. 单击 Kilo Code 右上角的**设置按钮**，选择 **Use your own API key**，并配置以下信息：
  - API Provider: **OpenAI Compatible** (Token Plan 接口兼容 OpenAI 标准)。
  - Base URL: `https://api.lkeap.cloud.tencent.com/plan/v3`。
  - API Key: Token Plan 专属的 API Key。
  - Model: 填写 Token Plan 支持的 **Model Name**，单击 **Use custom**。
3. 配置完成后，您可以在输入框中输入需求，与模型进行交互。

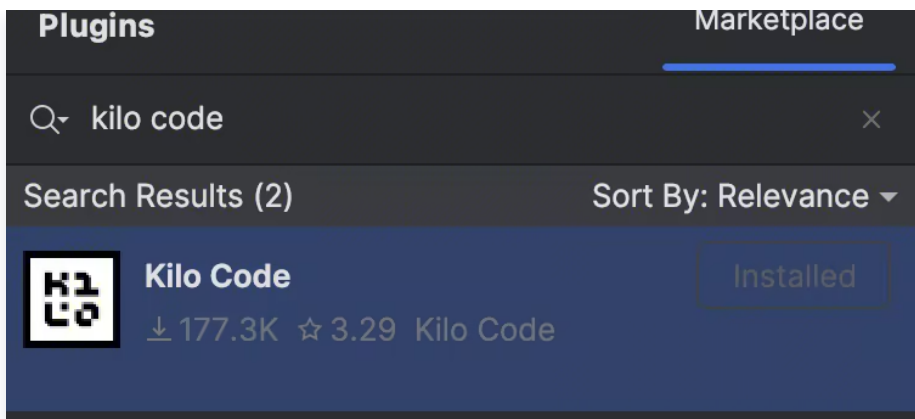


4. **模型切换**：您可以通过单击右上角的设置，在 Model 输入框搜索模型，选择 **Use custom** 进行切换。



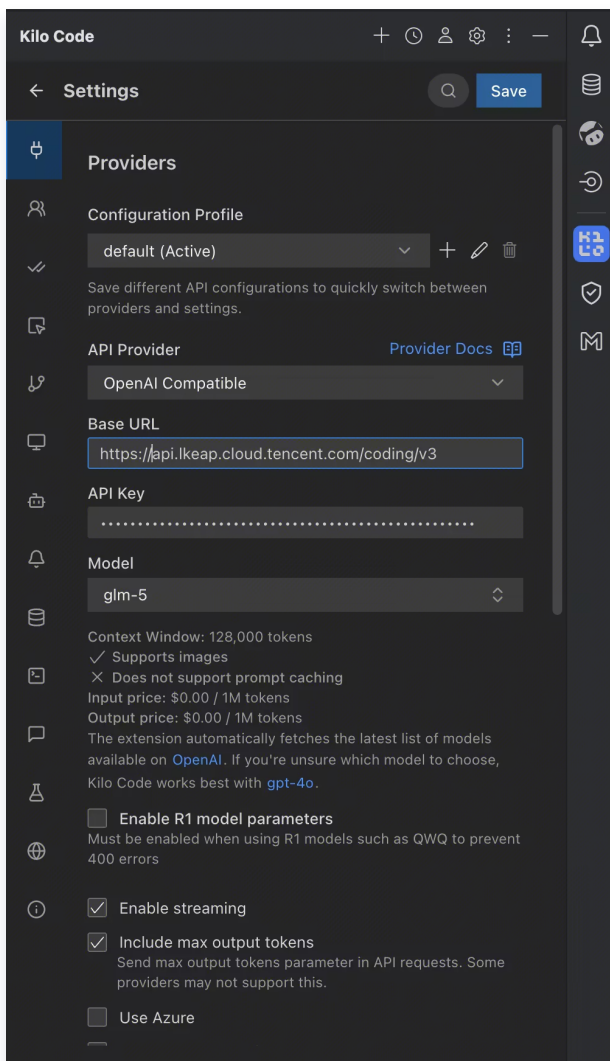
## JetBrains

1. 打开 JetBrains ，在扩展市场中搜索 Kilo Code 并安装，安装完成后重启 IDE 生效。

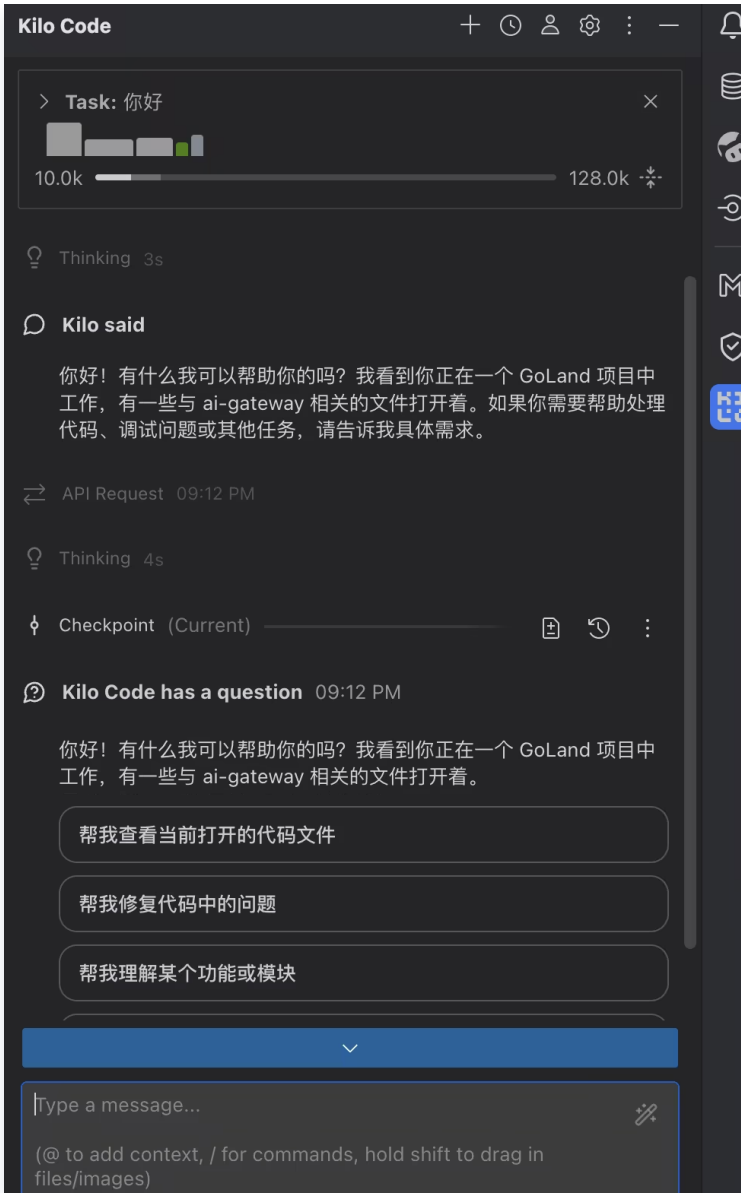


2. 安装成功后，单击 Kilo Code 右侧图标，选择 **Bring my own Key**，并配置以下信息：

- API Provider: **OpenAI Compatible** (Token Plan 接口兼容 OpenAI 标准)。
- Base URL: `https://api.lkeap.cloud.tencent.com/plan/v3`。
- API Key: Token Plan 专属的 API Key。
- Model: 填写 Token Plan 支持的 **Model Name**，单击 **Use custom**。



3. 在 Kilo Code 中使用，单击右上角的设置按钮，修改 Model 即可切换模型。



# Token Plan 常见问题

最近更新时间：2026-04-03 16:27:22

## 购买及用量额度问题

### 可以购买几个 Token Plan ？

同一主账号同时只能购买一个 Token Plan 套餐（不区分 Lite、Standard、Pro、Max 套餐）。如果一个主账号下有多个子账号，则主账号及其子账号合计只能购买一个套餐，且共用该套餐用量。

### 未开通自动续费，Token Plan 套餐有效期多长？

Token Plan 套餐自开通时起生效，有效期至次月对应日的同一时间结束。若次月没有对应日期，则有效期至次月最后一日 23:59:59（UTC+8）结束。

例如：2026 年 1 月 30 日 19:02:03 购买 Token Plan，有效期至 2026 年 2 月 28 日 23:59:59。

### Token Plan 用量额度消耗完了怎么办？

等待下个订阅月用量刷新，或者升级套餐到更高的等级，享用更充裕的配额。

### Token Plan 用量额度消耗完后会转为按量计费吗？

不会。Token Plan 额度耗尽后，继续调用将会失败报错，不会自动转为按量付费。如需继续使用，可升级至更高等级版本获取更多额度，或等待下一订阅周期额度刷新。

### Token Plan 可以使用腾讯云模型的免费额度吗？

不可以。Token Plan 为独立订阅产品，其计费体系不参与腾讯云 TokenHub 免费额度。

### Token Plan 套餐是否支持团队协作使用？

目前 Token Plan 主要面向个人开发者。团队协作使用，请通过 [文本生成服务](#) 调用模型服务，采用后付费模式，详情请参见 [模型价格](#)。

## 使用相关问题

### 购买后如何获取 API Key ？

在控制台 [Token Plan](#) 页面，单击生成密钥，获取 Token Plan 专属 API Key（格式为 `sk-tp-xxxx`）。

### Token Plan 可以生成多个 API Key 吗？

仅支持生成一个 API Key。

### 如何重置 API Key？

在控制台 [Token Plan](#) 页面的 API Key 右侧单击重置。

**⚠ 注意:**

- 此操作将立即生成新的 API Key，旧 API Key 预计 5 分钟内失效并且无法恢复，请谨慎操作。
- 旧 API Key 失效后，所有使用旧 API Key 的工具/应用/服务将立即无法调用 API，请确保更新对应的配置。

## Token Plan 到期不续费，重新开通后 API Key 会重置吗？

Token Plan 到期不续费，则套餐失效，原有的 API Key 也会失效。重新购买套餐后，会生成新的 API Key。

## 是否可以使用模型列表以外的模型？

Token Plan 当前仅支持使用 [模型列表](#) 内的模型，使用其他模型将会报错。

## Token Plan 有并发请求数限制吗？

Token Plan 并发速率与您的套餐等级相关，平台会根据资源进行动态调整，基本原则 Max > Pro > Standard > Lite。

## 不同套餐的模型性能一样吗？

所有套餐的模型性能一致。

## 各模型的上下文长度是多少？最大输出长度是多少？

模型	model 参数值	上下文长度 (Token)	最大输出长度 (Token)
Tencent HY 2.0 Instruct	hunyuan-2.0-instruct	144k	16k
Tencent HY 2.0 Think	hunyuan-2.0-thinking	192k	64k
MiniMax-M2.5	minimax-m2.5	200k	192k
Kimi-K2.5	kimi-k2.5	256k	256k
GLM-5	glm-5	200k	128k
Hunyuan-T1	hunyuan-t1	96k	64k
Hunyuan-TurboS	hunyuan-turbos	48k	16k