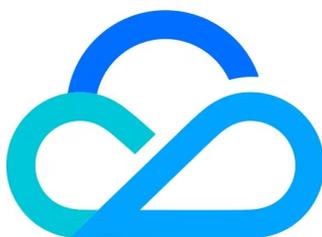


# 知识引擎原子能力 操作指南



腾讯云

## 【 版权声明 】

©2013–2026 腾讯云版权所有

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

## 【 商标声明 】



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

## 【 服务声明 】

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

## 【 联系我们 】

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

# 文档目录

## 操作指南

RAG 操作指南

Coding Plan

Coding Plan 概述

Coding Plan 常见问题

Coding Plan 接入 AI 工具

OpenClaw

CodeBuddy Code

OpenCode

Claude Code

CodeX

Cline

Cursor

Kilo CLI

Kilo Code

# 操作指南

## RAG 操作指南

最近更新时间：2026-03-22 21:49:32

### 概述

RAG ( Retrieval-Augmented Generation ) 是一种结合文档检索和生成模型的技术，用于回答复杂的问题。它结合了基于检索的方法和基于生成的方法的优点，在处理长文本和提供详细回答方面表现出色。

### 工作原理

RAG 通过以下几个步骤实现文档检索和问答：

- **文档加载和解析**

首先，系统从本地或网络源加载文档。文档可以是任何格式，如 PDF、DOCX、TXT 等。文档内容被解析并转换为纯文本格式。

- **文本拆分**

解析后的文本被分割成较小的文本块 ( chunks )。这种分割有助于系统更精细地处理和检索文档内容。

- **向量化**

每个文本块被转换为嵌入向量。嵌入向量是文本内容的向量表示，用于计算文本之间的相似度。

- **向量数据库**

嵌入向量存储在向量数据库 ( VectorStore ) 中。向量数据库用于高效地存储和检索嵌入向量。

- **查询处理**

用户输入查询。系统将查询转换为嵌入向量。

- **相似度计算**

系统计算查询向量与文档内容向量之间的相似度。通过相似度计算，系统可以找到与查询最相关的文本块。

- **重排序**

系统可以对多路召回的结果进行打分，输出 query 与切片内容的相关性分数。

- **提示模板生成**

系统将相关的文本块与查询合并，生成一个提示模板。这个模板将用于生成最终的回答。

- **生成回答**

提示模板被输入到大型语言模型 ( 如混元大模型 ) 中，语言模型根据提示模板生成最终的回答。

### 流程图

- **文档解析功能 ( 步骤1-2 )**

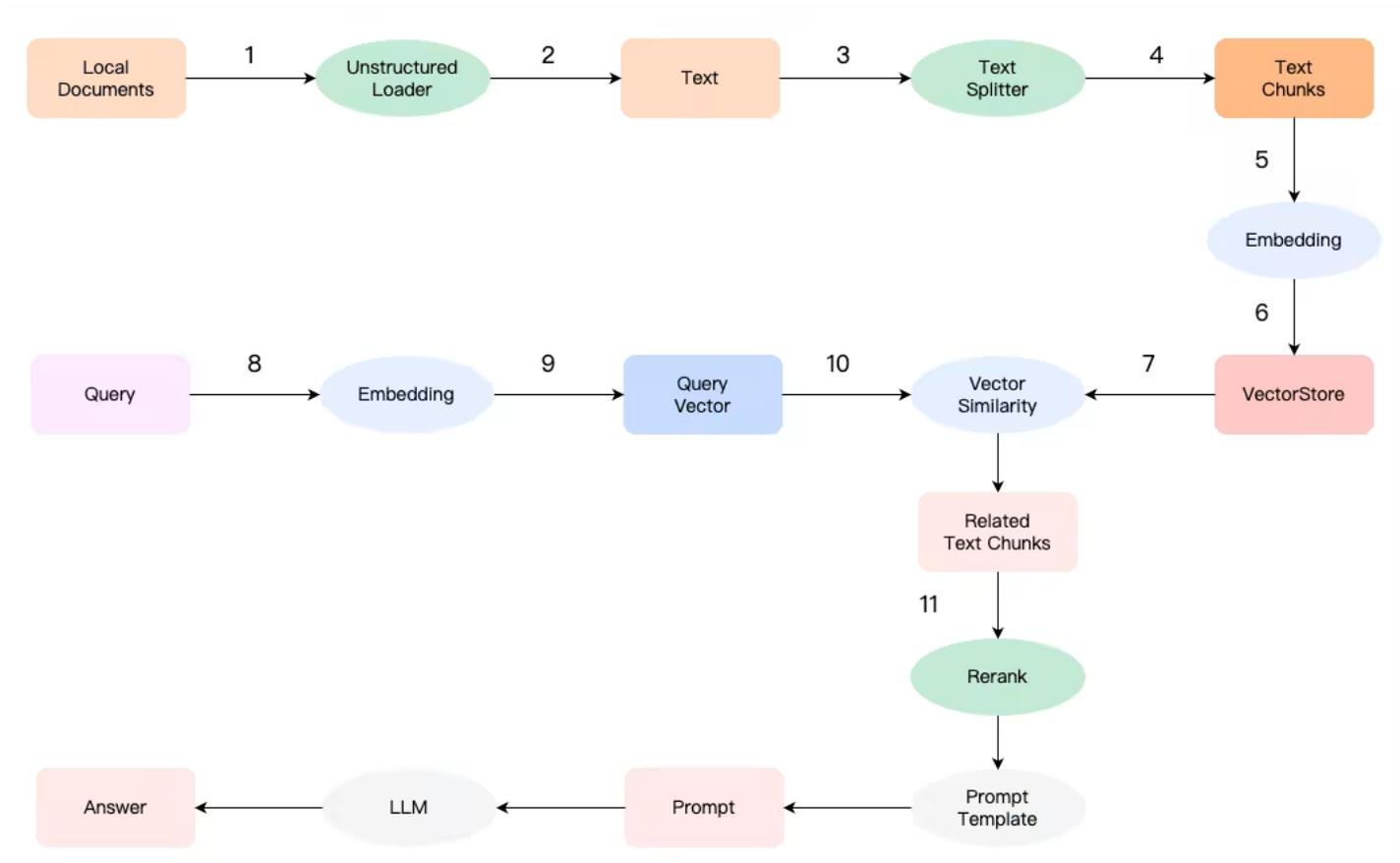
涵盖从本地文档的加载到文本内容的提取。这一阶段的目的是将非结构化的文档内容转换为纯文本形式，便于后续处理。

- **文档拆分功能 ( 步骤1-4 )**

包括文档解析功能，并进一步将提取的文本内容分割成较小的文本块。这种分割能够更精细地处理和组织文档内容，提升处理效率和检索效果。

• RAG 综合能力套件（步骤1-11）

从文档加载、内容提取、文本分割到嵌入向量的生成和存储，涵盖整个流程。通过结合检索和生成模型的优势，实现高效的文档内容检索和精确的问答生成。



## 如何开通功能

知识引擎原子能力为腾讯云智能体开发平台子产品，需开通腾讯云智能体开发平台体验权限后进行使用。腾讯云智能体开发平台的开通使用需要先通过 [腾讯云企业实名认证](#) 或者 [腾讯云个人实名认证](#)。通过实名认证后，首次在 [腾讯云智能体开发平台产品介绍页](#) 单击产品体验，即可开通知识引擎原子能力使用权限。详情请查看 [快速入门](#)。

## 文档解析功能流程说明

### 1. 提交文档

首先，您需要将需要解析的文档提交到系统中(CreateReconstructDocumentFlow)。支持的文档格式包括 PDF、DOCX、TXT 等，具体文档格式请参考接口文档说明。

### 2. 查询解析任务

提交文档后，您可以通过查询接口(GetReconstructDocumentResult)检查文档解析任务的状态。系统将会解析文档内容并生成相应的结构化数据。

## 文档拆分功能流程说明

## 1. 提交文档

首先，您需要将需要拆分的文档提交到系统中(CreateSplitDocumentFlow)。支持的文档格式包括 PDF、DOCX、TXT 等，具体文档格式请参考接口文档说明。

## 2. 查询拆分任务

提交文档后，您可以通过查询接口检查文档拆分任务的状态(GetSplitDocumentResult)。系统将会解析并拆分文档内容并生成相应的结构化数据。

# RAG 综合能力套件功能流程说明

## 1. 创建知识库

首先，您需要创建一个知识库(CreateKnowledgeBase)，用于存储问答对或文档。

## 2. 上传问答对或文档

接下来，将问答对(CreateQA)或文档(UploadDoc)上传到知识库中。系统会对这些内容进行处理和索引。

## 3. 查询文档状态

上传完成后，您可以通过查询接口(DescribeDoc)检查文档处理的状态。系统将解析和索引文档内容。

## 4. 进行检索

一旦文档处理成功，您即可通过检索接口(SearchKnowledge)进行内容检索，获取相关的问答或文档内容。

# Coding Plan

## Coding Plan 概述

最近更新时间：2026-03-25 20:49:12

腾讯云 Coding Plan，是为 AI Coding 场景推出的专属订阅套餐。

- 支持模型：选用 Tencent HY 2.0 Instruct、GLM-5、Kimi-K2.5、MiniMax-M2.5 等多个模型。
- 适配工具：支持使用 OpenClaw、CodeBuddy、Claude Code、Cline、Cursor 等主流编程工具。

### 适用场景

个人开发场景：助力开发者完成个人项目、学习实践、工具搭建等编码任务。

**说明：**  
企业级开发需求，请通过 [大模型 API](#) 调用模型服务。

### 套餐介绍

访问 [Coding Plan 购买页](#)，根据实际需求选择并购买套餐。

### 套餐详情

套餐内容	Lite 套餐	Pro 套餐
原价	40元/月	200元/月
用量限制	<ul style="list-style-type: none"><li>每 5 小时：最多约 1,200 次请求</li><li>每周：最多约 9,000 次请求</li><li>每订阅月：最多约 18,000 次请求</li></ul>	<ul style="list-style-type: none"><li>每 5 小时：最多约 6,000 次请求</li><li>每周：最多约 45,000 次请求</li><li>每订阅月：最多约 90,000 次请求</li></ul>
新客首购特惠活动	<ul style="list-style-type: none"><li>活动时间：2026.03.05 16:00 – 2026.04.19 00:00 (UTC+8)</li><li>发放规则：为保障订阅用户在调用高峰期的使用体验，特惠活动的参与名额每日限量，先到先得，售完即止</li><li>购买资格：仅限从未订阅过任何腾讯云大模型 Coding Plan 套餐的腾讯云新用户参与，同一用户仅可享受一次优惠</li></ul>	
	<b>Lite 套餐特惠价</b> <ul style="list-style-type: none"><li>首月7.9元/月</li><li>在活动时间内，购买特惠套餐并开启自动续费次月20元/月</li></ul>	<b>Pro 套餐特惠价</b> <ul style="list-style-type: none"><li>首月39.9元/月</li><li>在活动时间内，购买特惠套餐并开启自动续费次月100元/月</li></ul>

- 第三月起恢复原价40元/月

- 第三月起恢复原价200元/月

- **额度消耗:**

- 请求次数为模型调用的预估数值。通常一次用户提问会触发多次模型调用，且每次调用均会消耗 1 次用量额度，因此实际消耗的请求次数一般多于用户提问次数。
- 简单任务单次提问约消耗 5~15 次模型调用，复杂任务单次提问约消耗 15~30次或更多模型调用。用量信息可在 [Coding Plan 页面](#) 查看。

## 订阅前须知

**Coding Plan 不支持退款。**因此在订阅前请知悉以下重要内容：

- 1. 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**，敬请知悉。
- 2. 严禁 API 调用：**仅限在编程工具（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。
- 3. 数据使用授权：**您在使用 Coding Plan 期间，模型输入及模型生成的内容将用于服务改进与模型优化。若您停止使用 Coding Plan，将自动终止后续数据授权，终止授权的范围不涵盖此前已授权使用的相关数据。

## 使用说明

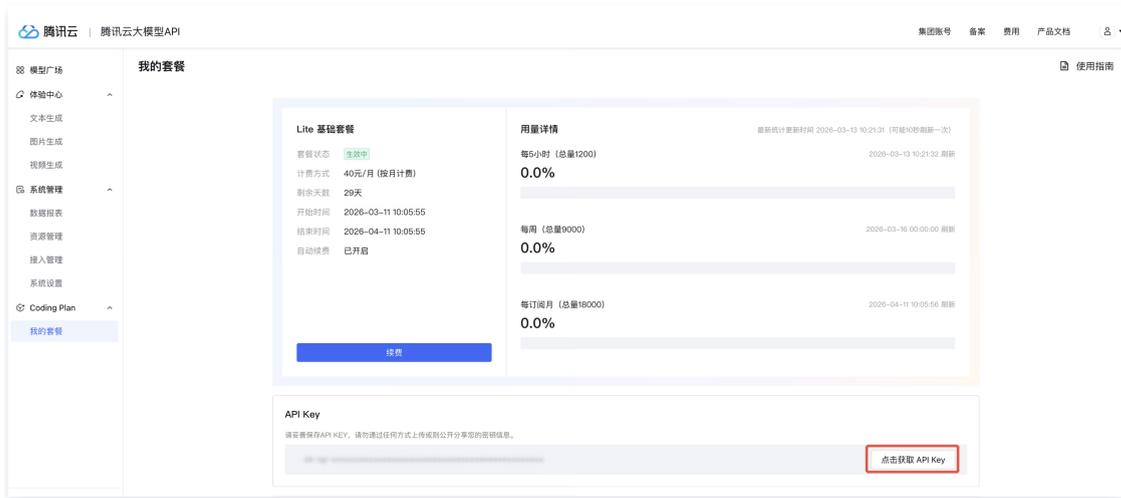
### 获取专属 API Key 和 Base URL

如果您已经购买了套餐，您需要获取并在 AI 工具中配置套餐专属的 API Key 和 Base URL，才能正确使用并抵扣套餐额度。

- **API Key:** 在 [Coding Plan 页面](#)，获取 Coding Plan 专属 API Key（格式为 `sk-sp-xxxx`）。
- **Base URL:** 后续需在 AI 工具中配置以下其中一个 Base URL（因工具而异），具体操作请参见对应的 AI 工具文档。
  - **OpenAI 兼容协议:** `https://api.lkeap.cloud.tencent.com/coding/v3`。
  - **Anthropic 兼容协议:** `https://api.lkeap.cloud.tencent.com/coding/anthropic`。

#### ⓘ 说明:

Coding Plan 专属的 API Key 和 Base URL 与腾讯云预付费及后付费的 API Key（`sk-xxxx`）和 Base URL（`https://api.lkeap.cloud.tencent.com/xxxxx/`）不互通，请勿混用。



## 套餐支持的模型

通过配置 Model Name，切换模型。

选择 Auto 模型会通过算法自动匹配最优模型，请通过 model 参数值配置。

### 说明：

- 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。
- Coding Plan 可用模型暂不支持多模态能力。

模型	model 参数值
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 (兼容别名 minimax-m-2-5)
Kimi-K2.5	kimi-k2.5 (兼容别名 kimi-k-2-5)
GLM-5	glm-5 (兼容别名 glm-5-0)
Hunyuan-T1	hunyuan-t1
Hunyuan-TurboS	hunyuan-turbos

## 支持的 AI 工具

Coding Plan 支持主流的 AI 工具，可按需选择。

AI 工具		
<p><b>OpenClaw</b></p> <p>开源、自托管个人 AI 助手</p>	<p><b>CodeBuddy Code</b></p> <p>腾讯云 AI 技术的智能编程工具</p>	<p><b>Claude Code</b></p> <p>AI 终端编程助手，支持自然语言编程</p>
<p><b>OpenCode</b></p> <p>开源 AI 编程代码工具</p>	<p><b>Cline</b></p> <p>VSCode 扩展，代码补全和调试</p>	<p><b>Cursor</b></p> <p>AI 原生代码编辑器</p>
<p><b>CodeX</b></p> <p>OpenAI 推出的命令行编程工具</p>	<p><b>Kilo CLI</b></p> <p>轻量高性能命令行编程工具</p>	<p><b>Kilo Code</b></p> <p>高效编程助手</p>

# Coding Plan 常见问题

最近更新时间：2026-03-30 15:42:02

## 购买及用量额度问题

### 可以购买几个 CodingPlan ？

同一主账号同时只能购买一个 Coding Plan 套餐（不区分 Lite 与 Pro 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，CodingPlan 套餐有效期多长？

CodingPlan 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 Coding Plan，有效期至 2026 年 2 月 28 日 23:59:59。

### Coding Plan 用量额度消耗完了怎么办？

每 5 小时、每周、每订阅月的额度用完后，可等待用量限额刷新。

### Coding Plan 用量额度消耗完后会转为按量计费吗？

不会。Coding 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 Pro 版本获取更多额度，或等待下一订阅周期额度刷新。

### Coding Plan 套餐用量额度刷新规则是什么？

- 每 5 小时：根据实际请求发生时间，滑动 5 小时窗口动态刷新。
- 每周：每周一 00:00:00 重置周限额。
- 每订阅月：每订阅月到期时间的下一秒重置订阅月限额。如：2026 年 3 月 6 日 19:02:03 购买 Coding Plan，在 2026 年 4 月 6 日 19:02:03 刷新订阅月限额。

### Coding Plan 可以使用腾讯云模型的免费额度吗？

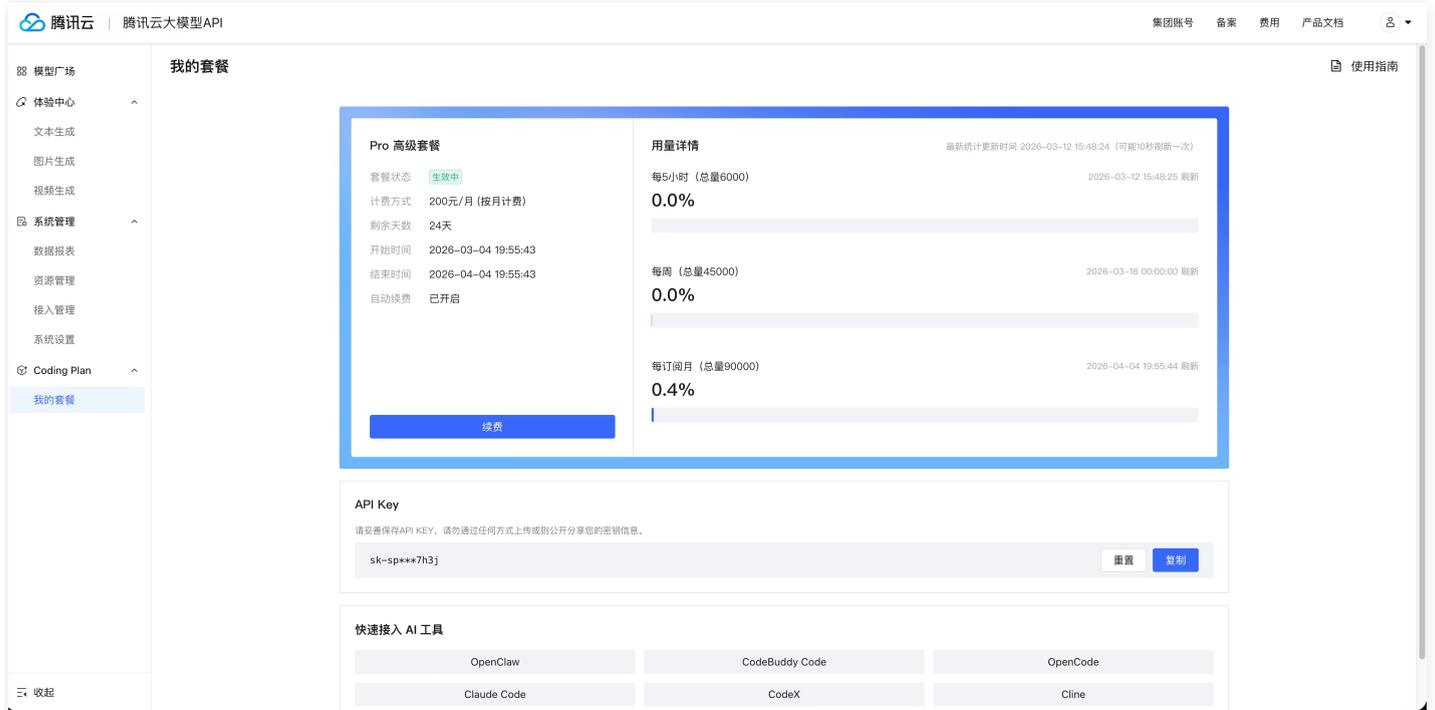
不可以。Coding Plan 为独立订阅产品，其计费体系不参与腾讯云通用混元大模型免费额度。

### Coding Plan 套餐是否支持团队协作使用？

目前 Coding Plan 主要面向个人开发者。团队协作使用，请通过 [第三方大模型 OpenAI 兼容接口](#) 调用模型服务，采用后付费模式，详情请参见 [计费概述](#)。

### 如何查看 CodingPlan 套餐用量消耗？

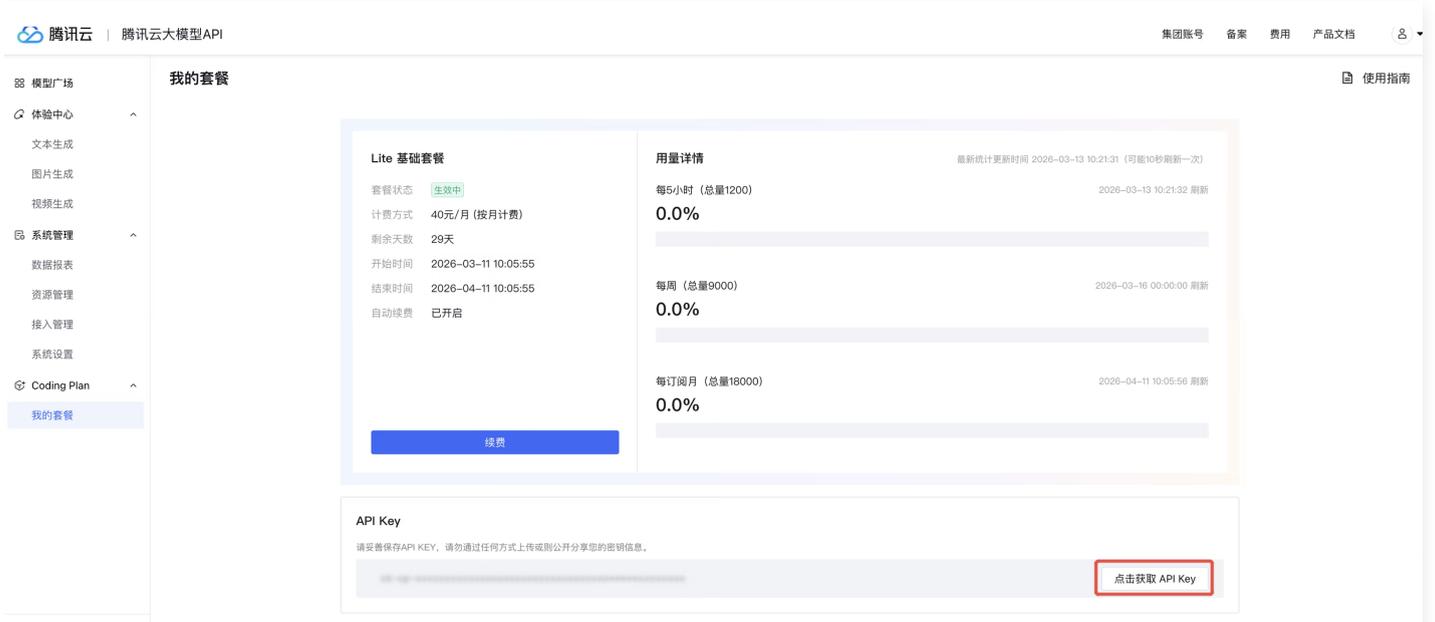
可在控制台 [CodingPlan > 我的套餐](#) 页面，查看套餐用量统计数据和使用率。



## 使用相关问题

### 购买后如何获取 API Key ?

在控制台 [CodingPlan > 我的套餐](#) 页面，点击获取 Coding Plan 专属 API Key ( 格式为 `sk-sp-xxxx` ) 。



### Coding 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?

可在控制台 [CodingPlan](#) > [我的套餐](#) 页面，API Key 点击重置按钮。

### ⚠ 注意：

- 此操作将立即生成新的 API Key，并使旧 API Key 失效（5 min 左右）。
- 所有使用旧 API Key 的应用 / 服务将立即无法调用 API，请确保已准备好更新配置。
- 重置后无法恢复旧 API Key，请谨慎操作。

The screenshot shows the '我的套餐' (My Plan) page in the Tencent Cloud console. The main content area displays the 'Pro 高级套餐' (Pro Advanced Plan) details. On the right, there are usage statistics for three periods: 5-hour (0.0%), weekly (0.0%), and monthly (0.4%). Below the usage statistics, the API Key is shown as 'sk-sp\*\*\*7h3j', with a red box highlighting the '重置' (Reset) button. At the bottom, there are buttons for quick integration with various AI tools like OpenClaw, Claude Code, etc.

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

不会重置。

## 是否可以使用模型列表以外的模型？

Coding 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。

## Coding Plan 有并发请求数限制吗？

Coding Plan 没有额外的并发请求数限制。使用时受套餐用量限制，包括每 5 小时、每周、每订阅月的请求次数上限，具体请参见 [Coding Plan 概述](#)。

## Lite 和 Pro 套餐的模型性能一样吗？

Lite 基础套餐与 Pro 高级套餐的模型性能一致，2 种套餐仅用量限制不同。

## 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 (Tokens)	最大输出长度 (Tokens)
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

## 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	根据接入工具章节和 Coding Plan 概述章节中的指导再次确认 URL 和 API Key 配置正确。
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	跟 Coding Plan 概述中支持的模型比对, 检查配置的模型名称是否正确。
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	hour allocated quota exceeded	每5小时请求额度已用完	每 5 小时是滑动窗口限制, 遇到此问题可以稍后重试, 或升级至 Pro 套餐。
20097	week allocated quota exceeded	每周请求额度已用完	等待至每周一 00:00:00 (UTC+8) 额度重置, 或升级至 Pro 套餐。
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 Pro 套餐。

# Coding Plan 接入 AI 工具 OpenClaw

最近更新时间：2026-03-10 11:02:01

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。

腾讯知识引擎 Coding Plan 支持在 OpenClaw 中使用，可参考本文进行配置及使用。

## 安装步骤

根据实际情况选择命令进行安装。

### 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. Continue?	选择 "Yes"。
Onboarding mode	选择 "QuickStart"。
Model/auth provider	选择 "Skip for now", 后续可以配置。
Filter models by provider	选择 "All providers"。

Default model	使用默认配置。
Select channel (QuickStart)	选择 "Skip for now", 后续可以配置。
Configure skills now? (recommended)	选择 "No", 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 "Hatch in TUI"。

## 配置 Coding 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, 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

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

```
openclaw dashboard
```

在 JSON 配置项中增加 models.providers 的配置。

- baseUrl: 是 Coding Plan 的接口 URL。
- <USER\_API\_KEY>: 替换成您自己的 API key。

```
"models": {
  "mode": "merge",
  "providers": {
    "tencent-coding-plan": {
      "baseUrl": "https://api.lkeap.cloud.tencent.com/coding/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"],
    
```

```

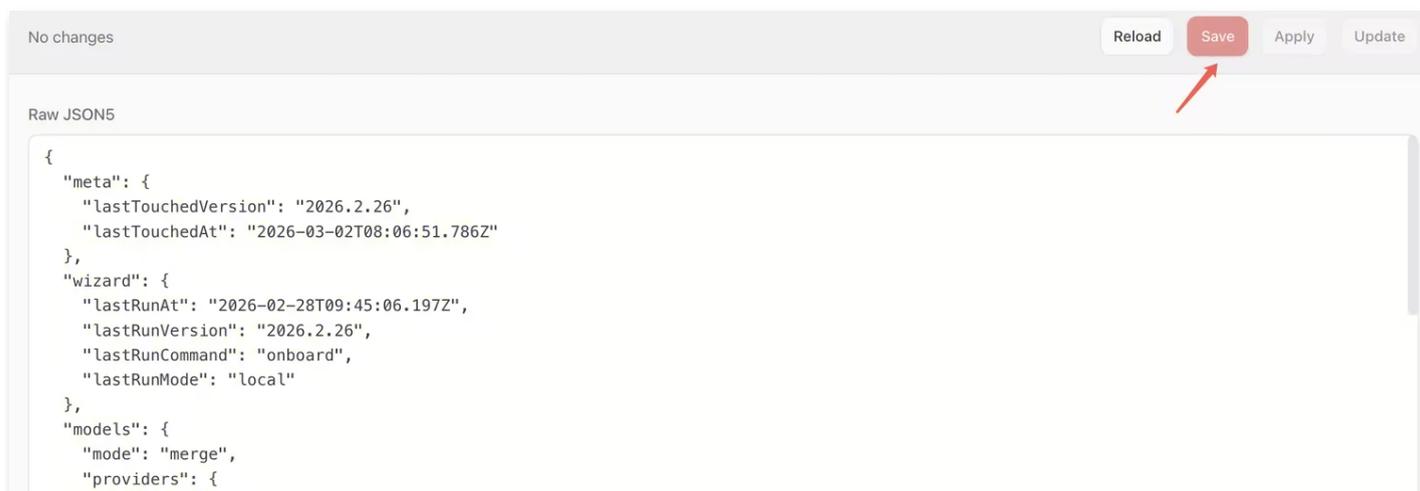
        "cost": { "input": 0, "output": 0, "cacheRead": 0,
"cacheWrite": 0 },
        "contextWindow": 32000,
        "maxTokens": 16000
    },
    {
        "id": "minimax-m2.5",
        "name": "MiniMax-M2.5",
        "reasoning": false,
        "input": ["text"],
        "cost": { "input": 0, "output": 0, "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
    }
]
}
}
}
}

```

修改 agents.default 增加模型列表模型信息：

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

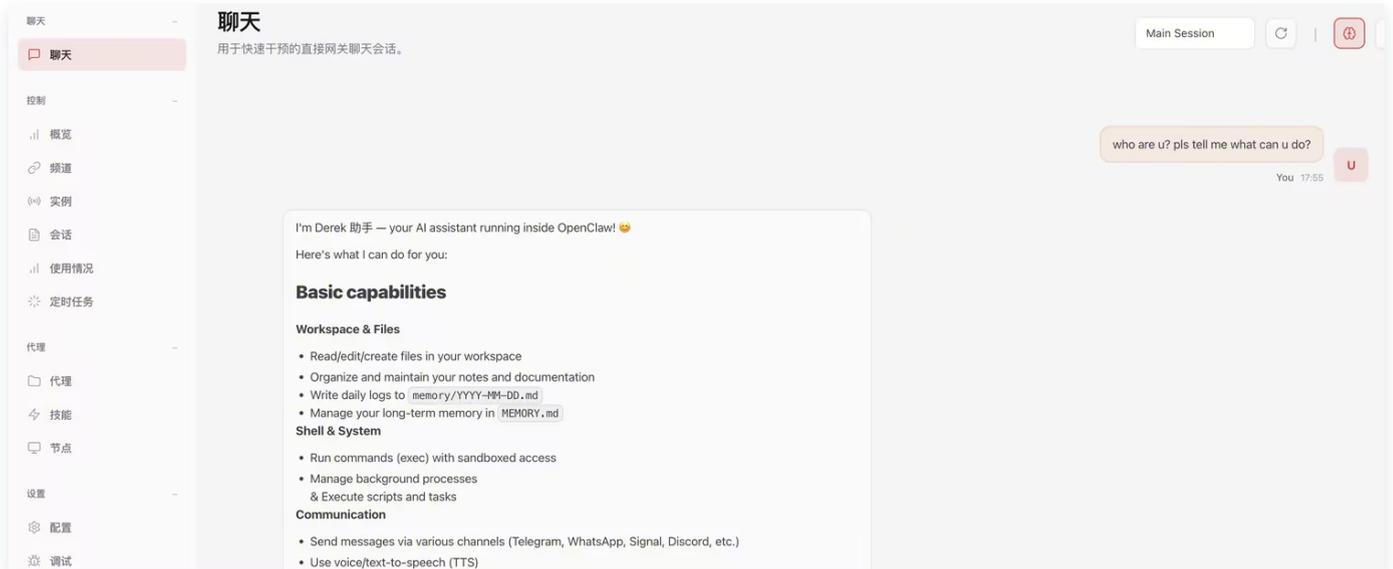
界面点击 Save 保存，保存完成后点击 Update 配置生效。



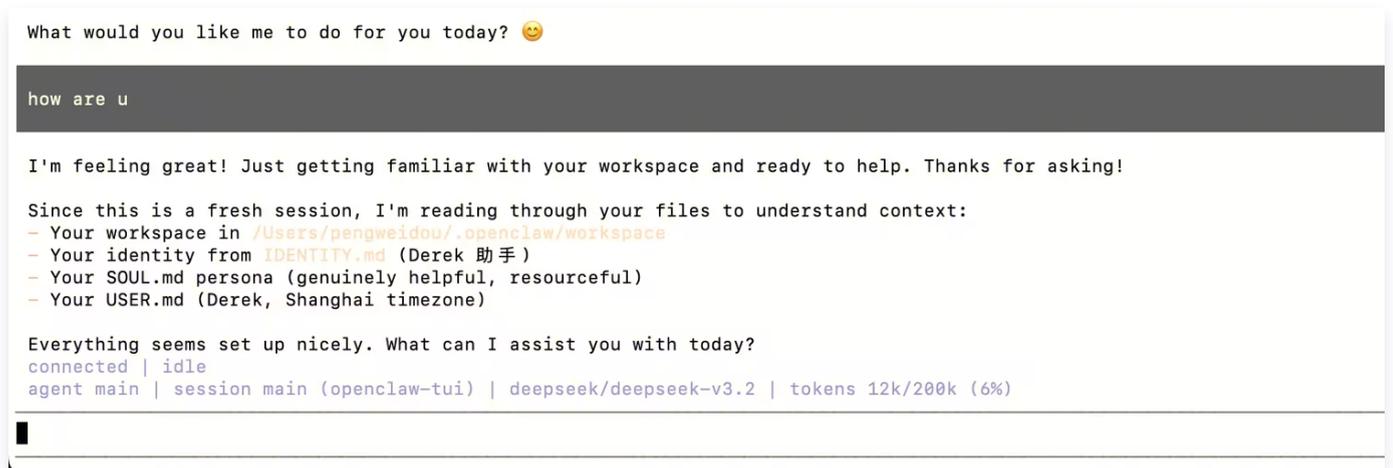
## 使用 OpenClaw

OpenClaw 支持通过 WebUI 和 TUI 的形式进行交互使用。

- 通过命令 `openclaw dashboard` 打开 WebUI，点击 Web UI 的左侧聊天，即可进行交互。



- 通过命令 `openclaw tui` 打开TUI进行交互。



# CodeBuddy Code

最近更新时间：2026-03-31 17:11:01

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。如需了解更多 CodeBuddy 相关信息，您可以参见 [CodeBuddy 产品概述](#)。

## 环境要求

- Node.js 18.0+

## 一键安装

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

## 验证安装

```
codebuddy --version
```

## 配置 Coding Plan

打开 CodeBuddy Code 的配置文件，复制粘贴如下配置。

### ⓘ 注意：

- 将 `<USER_API_KEY>` 替换为 Coding Plan 专属的 API Key。
- 配置文件路径如下，若文件不存在，直接新建即可：
  - Windows: `C:\Users\<用户名>\.codebuddy\models.json`。
  - macOS/Linux: `~/.codebuddy/models.json`。
- 如果遇到参数错误，可以关闭模型选择里的自动模式，将 `"name": "Auto"` 中的 `Auto` 换成支持的模型列表中任一模型的模型名，如换成 `"name": "Tencent HY 2.0 Instruct"`。

```
{
  "models": [
    {
      "id": "tc-code-latest",
      "name": "Auto",
      "vendor": "Tencent Cloud",
      "apiKey": "<USER_API_KEY>",
    }
  ]
}
```

```
    "url": "https://api.lkeap.cloud.tencent.com/coding/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/coding/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/coding/v3"
  },
  {
    "id": "hunyuan-t1",
    "name": "Hunyuan-T1",
    "vendor": "Tencent Cloud",
    "apiKey": "<USER_API_KEY>",
    "url": "https://api.lkeap.cloud.tencent.com/coding/v3"
  },
  {
    "id": "hunyuan-turbos",
    "name": "Hunyuan-TurboS",
    "vendor": "Tencent Cloud",
    "apiKey": "<USER_API_KEY>",
    "url": "https://api.lkeap.cloud.tencent.com/coding/v3"
  },
  {
    "id": "minimax-m2.5",
    "name": "MiniMax-M2.5",
    "vendor": "Tencent Cloud",
    "apiKey": "<USER_API_KEY>",
    "url": "https://api.lkeap.cloud.tencent.com/coding/v3"
  },
  {
    "id": "kimi-k2.5",
    "name": "Kimi-K2.5",
```

```

    "vendor": "Tencent Cloud",
    "apiKey": "<USER_API_KEY>",
    "url": "https://api.lkeap.cloud.tencent.com/coding/v3"
  },
  {
    "id": "glm-5",
    "name": "GLM-5",
    "vendor": "Tencent Cloud",
    "apiKey": "<USER_API_KEY>",
    "url": "https://api.lkeap.cloud.tencent.com/coding/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

配置完成后使用如下命令启动。

```
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

登录方式	适用场景	说明
Chinese Site	境内用户	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	境外用户	通过腾讯云国际站 (codebuddy.ai) 进行认证，支持境外主流模型。



# OpenCode

最近更新时间：2026-03-24 14:50:02

OpenCode 是一款开源 AI 编程工具，本文介绍如何在 OpenCode 中配置与使用腾讯知识引擎 Coding Plan。

## 安装步骤

执行如下命令安装。

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

安装完成后执行如下命令检查，输出版本号则表示安装成功。

```
opencode -v
```

## 配置 Coding Plan

在下面的路径中创建 OpenCode 的配置文件，并复制示例JSON代码粘贴到配置文件中，注意替换 <USER\_API\_KEY>，注意示例配置中以 glm-5 为例。

### 说明：

配置文件的路径如下：

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

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



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：2026-03-31 15:32:12

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。

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

## 安装步骤

### ⚠ 注意：

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

执行安装命令。

```
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 Coding Plan

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

- ANTHROPIC\_BASE\_URL，设置为：`https://api.lkeap.cloud.tencent.com/coding/anthropic`。
- ANTHROPIC\_AUTH\_TOKEN，设置为：Coding Plan 专属 API Key。
- ANTHROPIC\_MODEL，设置为：Coding 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。

### ⚠ 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/coding/anthropic",
    "ANTHROPIC_MODEL": "<Model_Name>"
  }
}
```

2. 在配置完 settings.json 文件后，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
}
```

保存配置文件后，在新的终端窗口执行后续命令。

## 使用 Claude Code

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



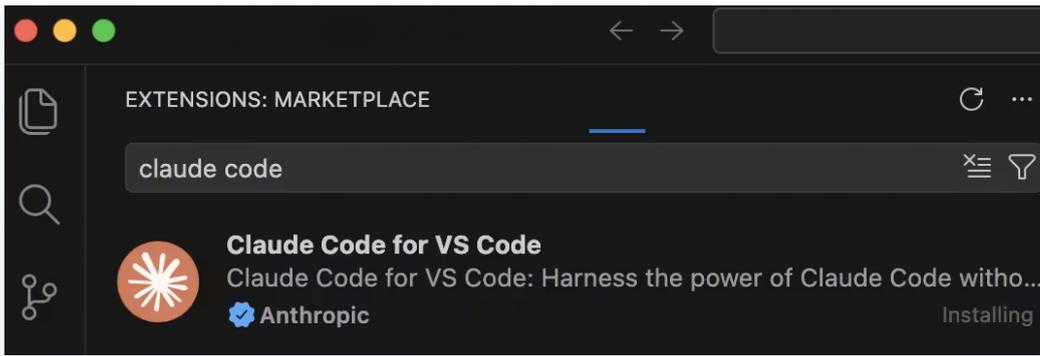
## 在 IDE 中使用 Claude Code

**⚠ 注意:**

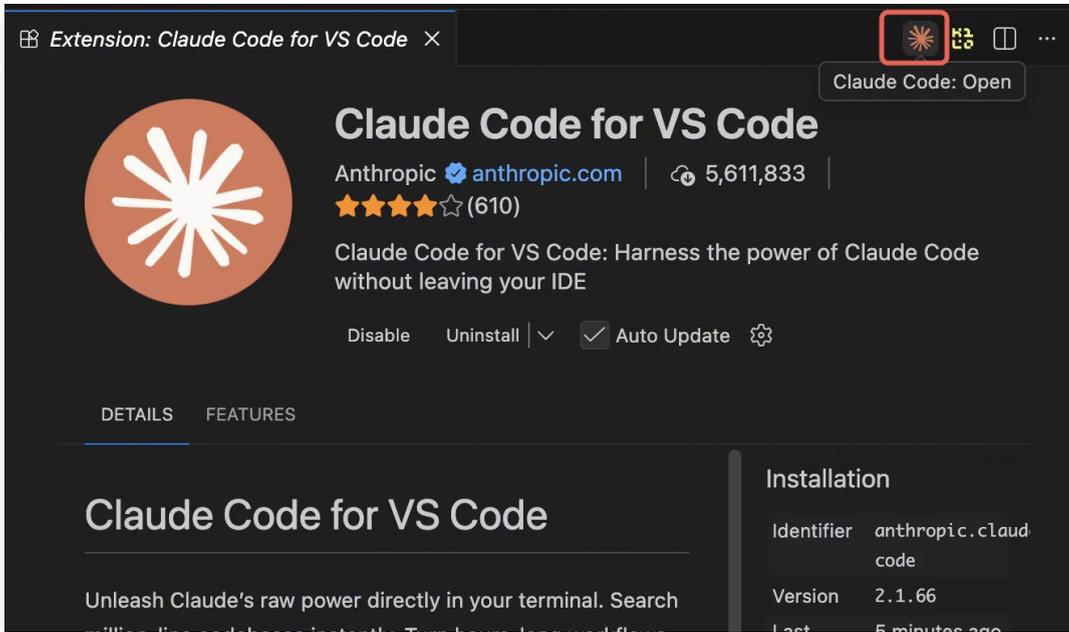
Claude Code IDE 插件依赖 Claude Code CLI 工具，需先完成 Claude Code 的安装及配置。

### VS Code

在扩展市场搜索 Claude Code 进行安装。



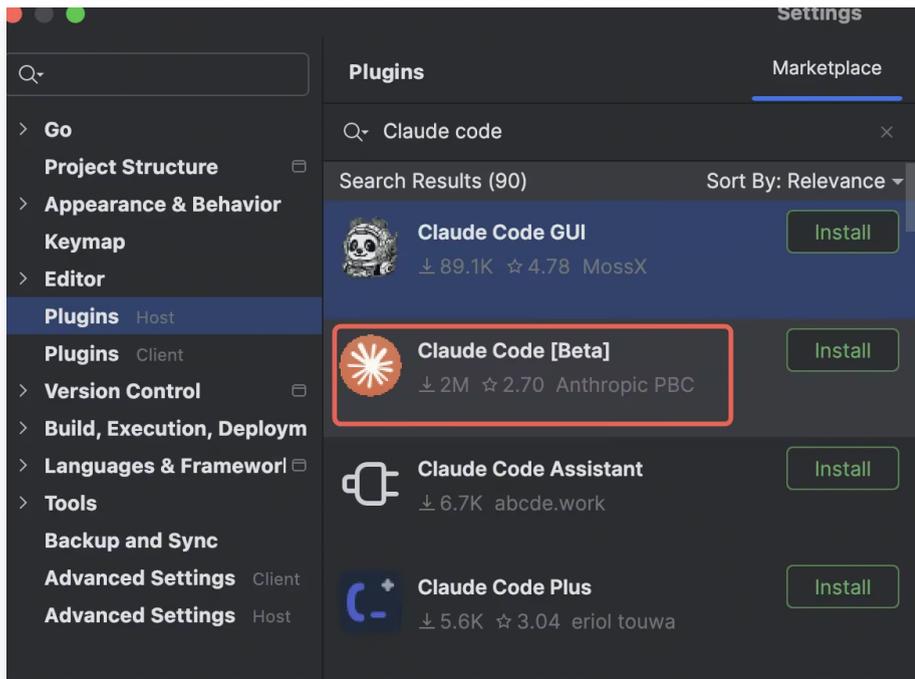
安装成功后，点击 VSCode 右上角的 Claude Code 图标，进入 Claude Code 页面。



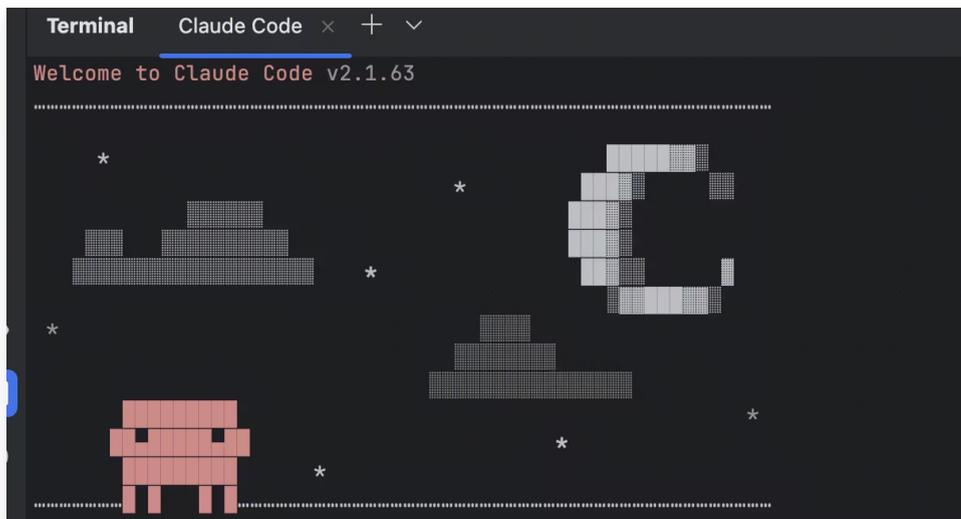
切换模型：在对话框中输入 /，选择 General config 进入设置页面，在 Selected Model 中选择支持的模型，新建一个窗口开始对话。

## JetBrains

在扩展市场中搜索 Claude Code 并安装。



安装后重启 IDE，点击右上角图标即可使用，可通过 `/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：2026-03-31 15:32:12

Codex 是由 OpenAI 推出的智能编程代理。

## 安装 CodeX

### ⚠ 注意：

- 请安装支持 Chat/Completions API 的旧版 Codex（例如：0.80.0 版本）。
- 新版本 Codex 采用 Responses API 接口，暂不支持 Chat/Completions API。

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 Coding Plan

编辑或新增 config.toml 文件，修改或新增的配置信息如下：

### ⚠ 注意：

不同系统配置文件路径不同，具体如下：

- macOS & Linux: `~/.codex/config.toml`。
- Windows: `C:\Users\<用户名>\.codex\config.toml`。

```
model_provider = "Lkeap_Coding_Plan"
model = "glm-5"
[model_providers.Lkeap_Coding_Plan]
name = "Lkeap_Coding_Plan"
base_url = "https://api.lkeap.cloud.tencent.com/coding/v3"
env_key = "CODING_API_KEY"
wire_api = "chat"
```

## 配置环境变量

在命令行中将 CODING\_API\_KEY 环境变量设置为 Coding Plan 专属 API Key。

### macOS

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 CODING_API_KEY="USER_API_KEY"' >> ~/.zshrc

#配置生效
source ~/.zshrc
```

#### Bash

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

#配置生效
source ~/.bash_profile
```

### Windows

#### CMD

```
# 用您的Coding Plan API Key 替换 USER_API_KEY
setx CODING_API_KEY "USER_API_KEY"

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

## PowerShell

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

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

## 使用 CodeX

使用如下命令打开 CodeX，选择跳过更新使用。

```
codex
```

```
🌟 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
```

开始对话使用。

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

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

> 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：2026-03-23 17:11:22

Cline 是一款 VSCode 智能编程的插件。

## 安装步骤

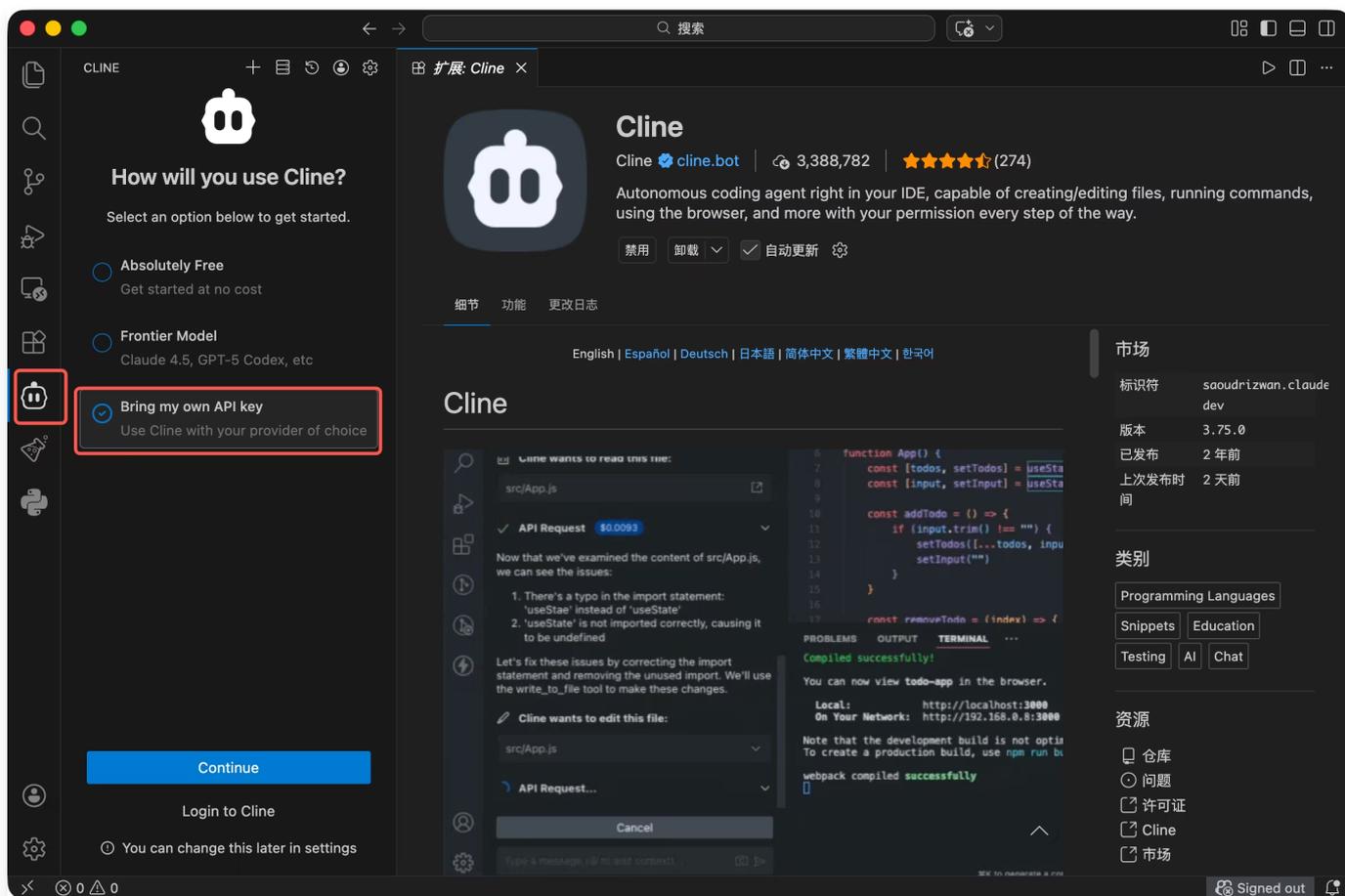
打开 VSCode，在扩展市场搜索 Cline 安装。

## 配置 Coding Plan

### 进入配置界面

安装完成后：

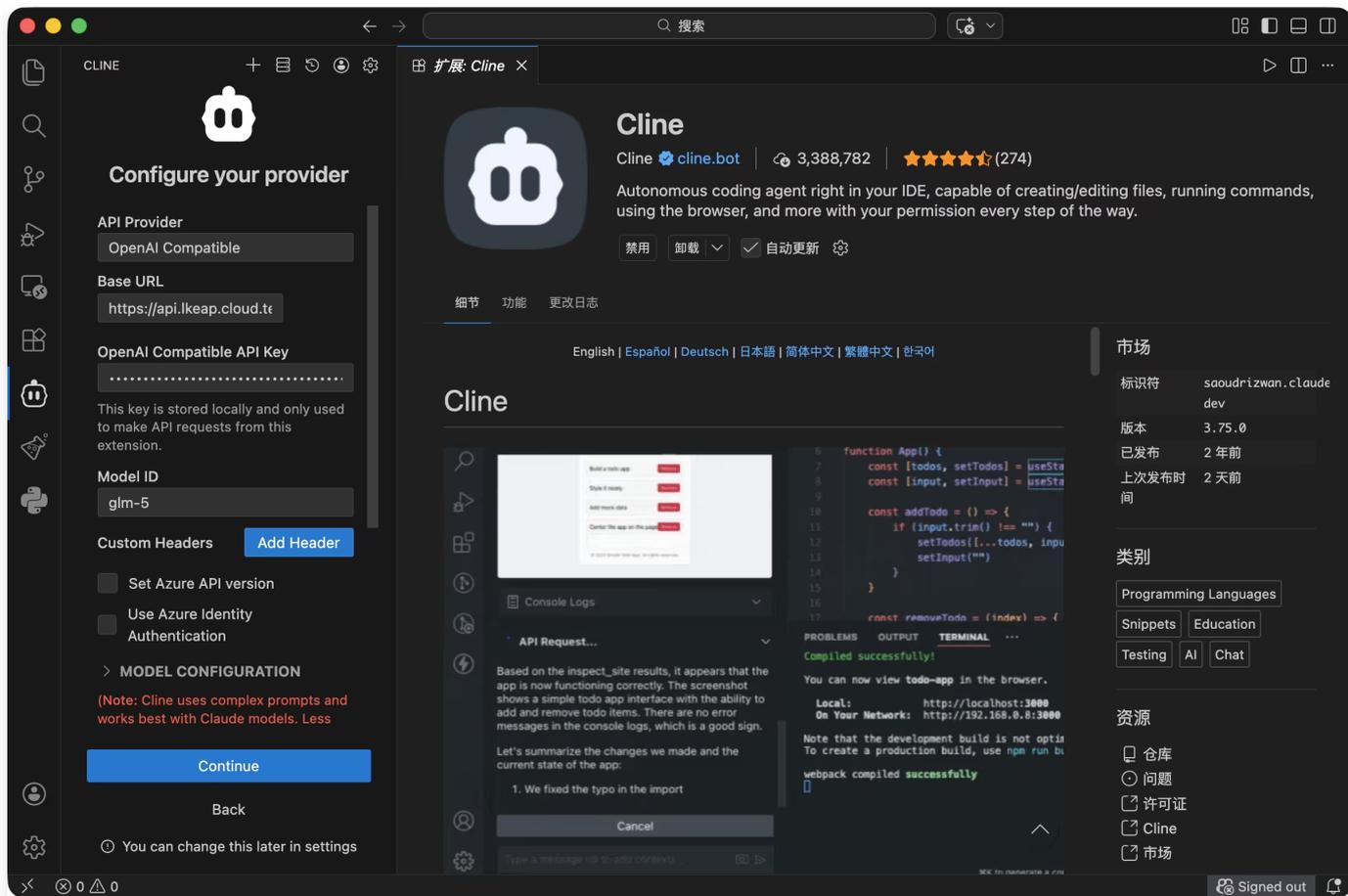
1. 点击左侧活动栏中的 **Cline** 图标。
2. 在欢迎界面中，选择 "Bring my own API Key"（使用自己的 API Key）。
3. 单击 **Continue** 进入 Provider 配置界面。



## 选择 API Provider

在 Provider 配置界面：

1. 点击 API Provider 下拉菜单。
2. 选择 OpenAI Compatible (OpenAI 兼容模式)。



## 配置连接参数

完成以下必填配置：

参数	说明	示例
Base URL	Coding Plan 的 API 服务的基础地址	https://api.lkeap.cloud.tencent.com/coding/v3
API Key	Coding Plan 的 API Key	sk-xxxxxxxxxxxxxxxxxxxx
Model ID	使用支持 Coding Plan 的模型	glm-5, kimi-k2.5 等

## 验证配置

配置完成后：

1. 单击 **Continue** 保存配置。

- 
2. 在 Cline 输入框中输入测试问题，如："你好，请介绍一下你自己"。
  3. 如果配置正确，Cline 将使用您配置的模型进行响应。

# Cursor

最近更新时间：2026-03-05 17:31:41

## 安装步骤

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 及以上套餐的用户才支持自定义配置模型。

Cursor 安装完成后，在 Cursor 中，点击右上角的 Cursor Settings，选择 Models 页面。Models 模块的具体配置如下：

- OpenAI API Key：使用 Coding Plan 的 API Key。
- Override OpenAI Base URL： `https://api.lkeap.cloud.tencent.com/coding/v3`。
- Add Custom Model：使用支持 Coding Plan 的模型，例如：glm-5。

## 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：2026-03-23 14:39:22

## 安装步骤

执行如下命令进行安装。

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

执行如下命令检查安装，输出版本号则表示安装成功。

```
kilo --version
```

## 配置 Coding Plan

编辑或新增 Kilo CLI 的配置文件 config.json，复制粘贴如下配置。

### ⚠ 注意：

- 注意将 <USER\_API\_KEY> 替换为 Coding Plan 专属的 API Key，配置中以 glm-5 模型为例。
- 不同系统配置文件路径不同，具体如下：
  - macOS & Linux: `~/.config/kilo/config.json`。
  - Windows: `C:\Users\<<用户名>\.config\kilo\config.json`。

```
{
  "$schema": "https://kilo.ai/config.json",
  "provider": {
    "lkeap": {
      "npm": "@ai-sdk/anthropic",
      "name": "Lkeap Coding Plan",
      "options": {
        "baseUrl":
"https://api.lkeap.cloud.tencent.com/coding/anthropic/v1",
        "apiKey": "<USER_API_KEY>"
      },
    },
    "models": {
      "glm-5": {
```

```
"name": "GLM-5",
"options": {
  "thinking": {
    "type": "enabled",
    "budgetTokens": 1024
  }
}
```

## 使用 Kilo CLI

配置完成后使用如下命令启动。

```
kilo
```

输入/models，选择 Lkeap Coding Plan 分组下的模型，示例中以 glm-5 为例。



# Kilo Code

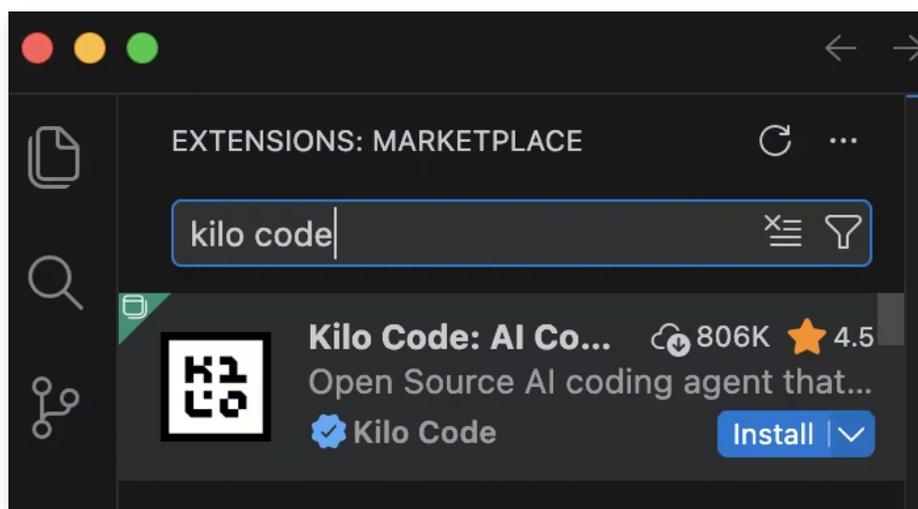
最近更新时间：2026-03-09 17:06:11

## 安装使用

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。

### VS Code

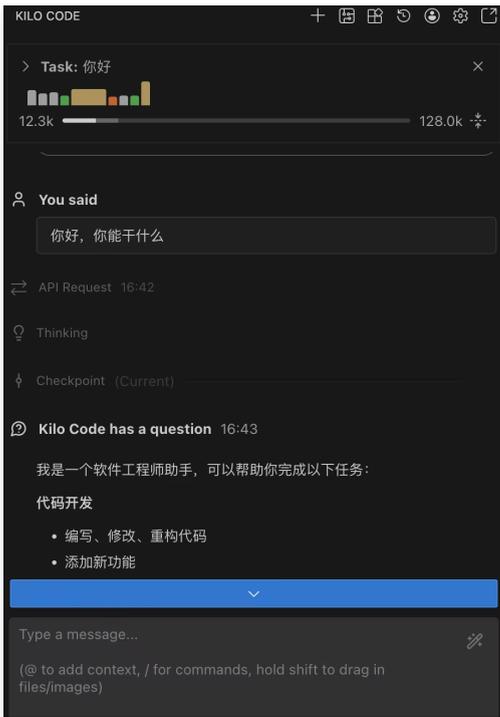
打开 VSCode，在扩展市场中搜索 Kilo Code 并安装。



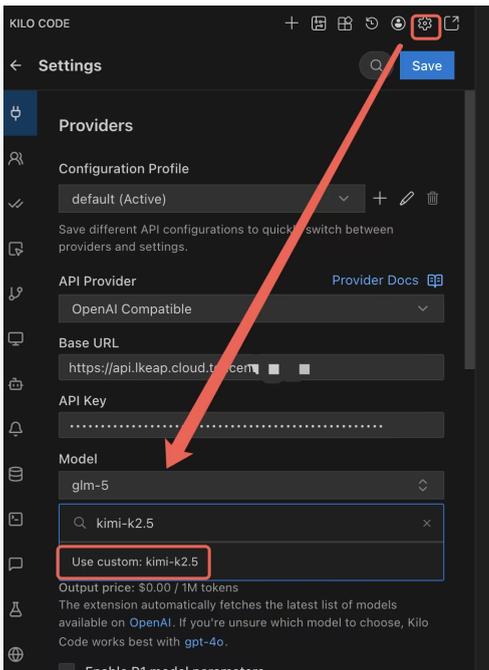
选择 Use your own API key，然后配置以下信息。

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

配置完成后，您可以在输入框中输入需求，与模型进行交互。

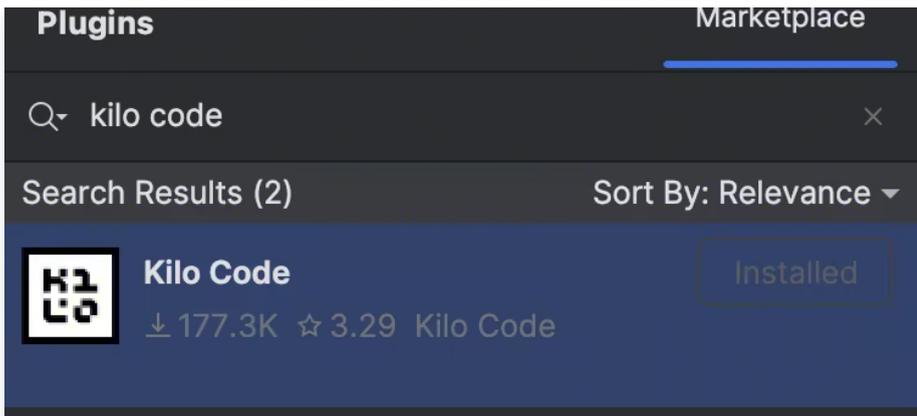


模型切换，可以通过单击右上角的设置，在 Model 输入框搜索模型，选择 Use custom 进行切换。



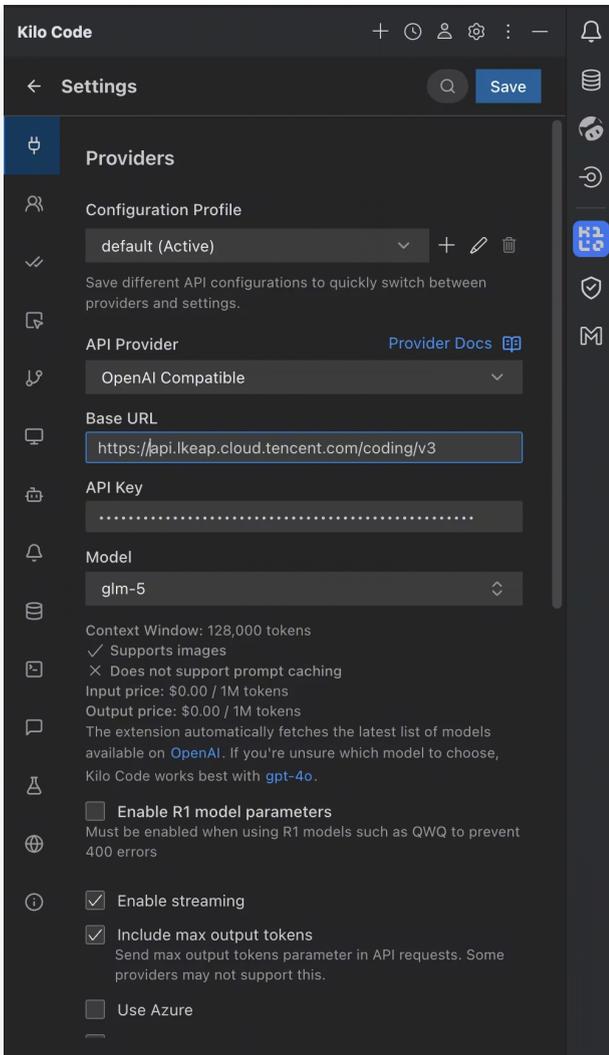
## JetBrains

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。



点击右侧图标，选择 Bring my own Key 进行配置，主要配置如下参数。

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



在 Kilo Code 中使用，单击右上角齿轮设置按钮，修改 Model 即可切换模型。

