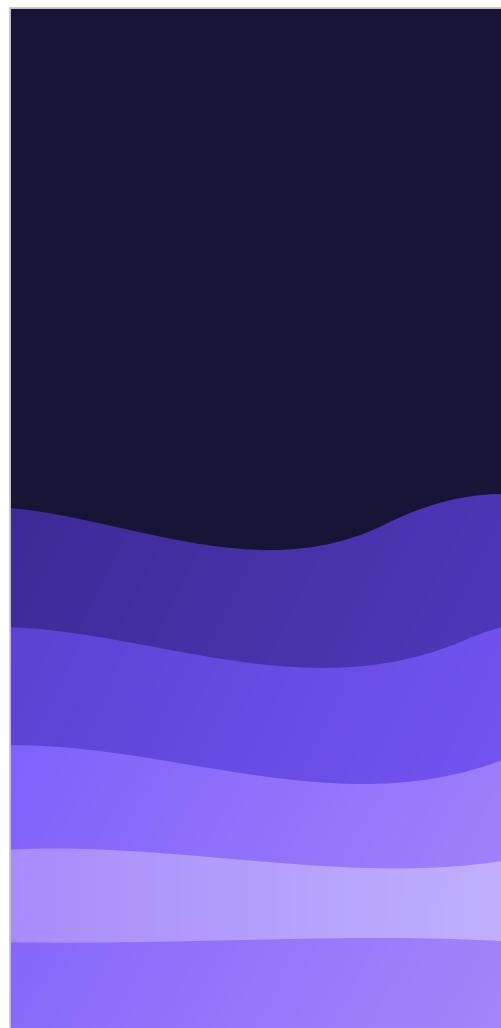


— 技术 · 实践



从长文本到一套 agent 栈

别再把它当长文本工具 —— K3 旗舰思考永远开, 一套整合的 agent 栈加双兼容 API, 国内可达、中文长材料顺手; 该用的那几面替你挑出来, 什么时候回 frontier 也说清。



目录.

01	引子 · 你印象里的 Kimi 是旧的	5 分钟
02	全景 · 一套对标前沿的栈	7 分钟
03	挑模型 · K3、K2.7-Code 与 K2.6	7 分钟
04	四模式 · 快答、深想、交办、集群	6 分钟
05	交办 · Agent 直接做成品	7 分钟
06	大活 · Agent Swarm 与 Kimi Claw	7 分钟
07	查证 · Deep Research 出带出处的报告	7 分钟
08	写码 & 接 API · 把 Kimi 配进你的 Agent	12 分钟
09	选档 & 取舍 · 买哪档, 什么时候回 frontier	6 分钟
10	实战速查 · 哪类活该用哪一面	5 分钟

引子 · 你印象里的 Kimi 是旧的。

你对 Kimi 的印象, 大概还停在「长文本」那一年。它早不是那个工具了——到 K3, 它长成一套对标 OpenAI / Anthropic 的全栈。这本书讲这套栈, 以及该用它的哪一面。

你大概是这样记住 Kimi 的: 2024 年那个能一口气读进几十万字的框——把一整本书、一份几百页的合同丢进去, 它真能读完、答得上。那一下确实惊艳, 也是它出圈的原因。问题是, 那是两年前的 Kimi。你的印象停在了「长文本」, 它却早走远了。到 K3, 它不再是一个特长生, 而是一套对标 OpenAI 和 Anthropic 的**全栈**: 2.8 万亿参数、1M 上下文的旗舰模型、能做出网站幻灯文档表格的 Agent、调度一大群子 agent 的 Agent Swarm、Deep Research、跑在终端的 Kimi Code, 底下还垫着一套 OpenAI 兼容、权重开源的 API。¹ 这本书讲的, 就是这套你大概率只用了一个角的栈。

- I

你记得的是 2024 年那个 Kimi。

先认下这一句: 你脑子里的 Kimi, 是它当年用来出圈的那张底牌, 不是它现在的样子。Kimi 出名靠的是长上下文: 一次吃下几万、几十万字, 别人要分段喂, 它一口吞; 再加上中文写得顺, 它成了很多人手里那个「读长文、写中文」的工具。这张底牌今天还在, 也仍是强项——这本书后面照样会用到它。但如果你对它的认识只停在这里, 你看到的是一张过期快照: 它早不是只会这两件事的特长生。判断它的标准, 也跟着变了。当年你问的是「它能不能读完这份长文」; 现在该问的是「这套栈里, 哪一面对得上我手头这类活」。问题从一道是非题, 变成了一张地图——而多数人手里还攥着两年前那道是非题。



一年六代 —— 你记得的 Kimi, 已经是好几代以前的事了。截至 2026-07。

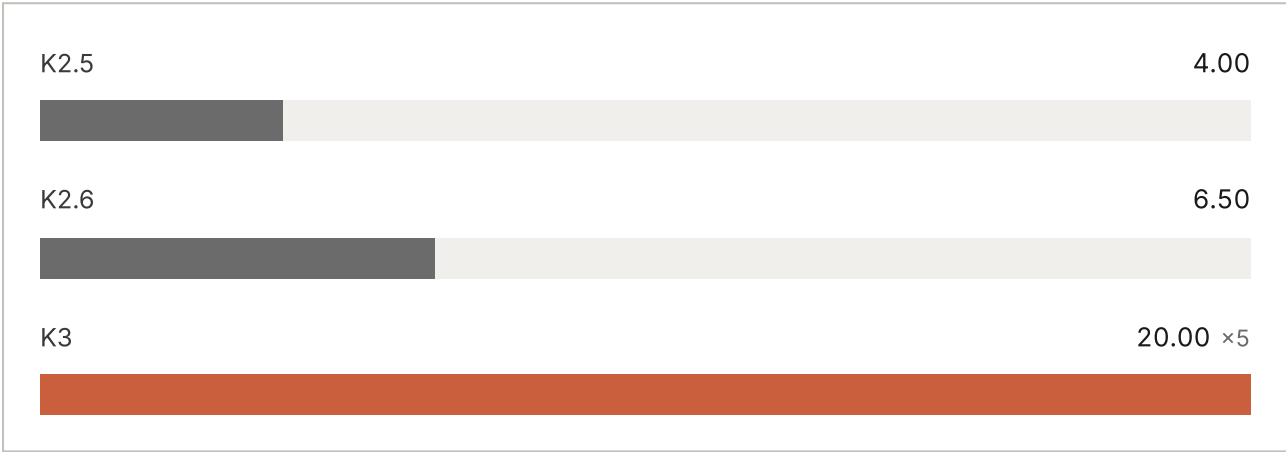
• • •

— II

它长成了一套对标前沿的栈。

把它现在的产品线摆出来, 你会发现几乎是照着 OpenAI 和 Anthropic 描的。旗舰模型 K3 是 2.8 万亿参数的 MoE、1M token 上下文, 原生多模态、思考永远开着; 权重官方承诺 7 月 27 日前放出, K2 系的早就以宽松协议挂在 Hugging Face 上。² 往上是一整套产品: 聊天加五档会员、Agent 直

接做成品、Agent Swarm、Deep Research、Kimi Code、Kimi Claw, 侧边栏还新添了桌面端的 Kimi Work, 底下垫一套 OpenAI 兼容的开放平台 API。¹ 这张表几乎能和 ChatGPT / Claude 那两套一对上 —— 它要的就是这个位置, 而不是「国产长文本工具」那个位置。最能说明它换了打法的是价格。两年三级跳: K2.5 输入每百万 token ¥4.00, K2.6 涨到 ¥6.50, K3 直接定到 ¥20.00、输出 ¥100.00。³ ⁴ 一个还在抢市场的便宜货不会这样定价; 敢定, 是因为它认定自己值这个钱。**抢量、拼便宜那条路, 现在是 DeepSeek 在走; Kimi 把自己搬到了对标前沿的货架上。**你该用什么眼光看它, 这个动作说得比任何 benchmark 都清楚。



¥ / 百万 token · 输入价 · K2.5 → K3, 输入价两年五倍 —— 它把自己挪出了「便宜」那一栏。

你印象里的 KIMI

一个读长文、写中文的框。一张底牌, 一个特长生 —— 拿它跟「国产某某」比。

现在的 KIMI

多模态旗舰 + 思考、Agent 做成品、Agent Swarm、Deep Research、Kimi Code、开源 API —— 一套对标前沿的栈。长文本只是其中一面。

• • •

这本书给这套栈画一张图。

能力早就在那里, 缺的是一张图: 哪一面对上你哪类活, 哪些根本不必学。这本书不列全部功能, 也不教你点哪个按钮(界面几个月一变, 判断不会)。它给你一张栈的地图: 每一面**对标前沿的哪一个、省你什么、什么时候用、什么时候别用**。后面的章节顺着这套栈走 —— 全景、挑模型、四种模式、Agent 交办、Agent Swarm 与 Claw、Deep Research、Kimi Code 与 API, 然后是选档与取舍, 末尾再附一页实战速查。也把话说在前头: 「长成全栈」不等于「样样第一」。最难的推理、最吃打磨的 agentic coding、一锤定音的输出, 仍是 Claude、GPT 这些前沿的地盘, Kimi 还差半步 —— 这本书会在每一面标出那条线, 该回 frontier 就回。还有一条最容易踩的: **会员订阅不含 API**, 开放平台 (platform.kimi.com) 是另一套账户、另一条账单。³ 它们重要, 只是别混成一笔账。

- 补充

立场说在前面: 我是付费用它的人, 不是 Moonshot 的产品经理。官网自己能写的话这里不写; 不接 affiliate, 推荐一样东西会告诉你为什么, 也会告诉你它的代价。模型版本、价格、上下文长度都标了日期 —— Kimi 迭代很快(这一年从 K2 一路走到 K3), 看到本书时请以官网当时的页面为准。

动手之前先照一次镜子 —— 看看这套栈, 你到底只用了哪几面:

01

翻记录, 挑 5 件

打开你的 Kimi 历史, 往回翻两周, 挑出 5 件你让它做过、且不止做过一次的事。

02

给每件标一面

对每件标上它属于这套栈的哪一面: 默认聊天、长文档、Deep Research、Agent 做成品、Agent Swarm、Kimi Code。先凭直觉填。

03

数一数有几件写着「默认聊天」

大概率五件里四件都是默认聊天。这张表就是你读这本书的起点, 也是读完之后的对照组 —— 后面每一章, 都在帮你把更多的活挪出那个框。

做完你会发现: 不是 Kimi 少了哪样能力, 是你一直只站在这套栈的一个入口。

参考.

- 01 Kimi 官网首页(kimi.com)——截至 2026-07 首屏标注「Kimi AI with K3 | Built for Agentic Coding & Knowledge Work」(中文站「K3 上线, 专为智能体编程与知识工作打造」), 产品入口排着 Chat / Slides / Websites / Docs / Sheets / Deep Research / Agent / Agent Swarm / Kimi Code / Kimi Claw(入口已迁 /bot), 侧边栏另有 Kimi Work(Beta)。这是「它早不只是聊天框」最直接的一手证据。[Kimi 官网](#)
- 02 Kimi K3 官方博客(2026-07-16)——K3 是 2.8 万亿参数的 MoE(896 专家、每 token 激活 16)、1M token 上下文、原生视觉、思考默认 max 档; 官方承诺「full model weights will be released by July 27, 2026」。K2 系权重(Kimi-K2.6 / Kimi-K2.7-Code)早已以 Modified MIT 挂在 Hugging Face。截至 2026-07-17, K3 权重尚未上架。[Kimi K3 官方博客](#)
- 03 Kimi 开放平台(platform.kimi.com, 原 platform.moonshot.cn)——API base_url 为 https://api.moonshot.cn/v1, 与 OpenAI Chat Completions 兼容; 旗舰 kimi-k3 每百万 token 输入 ¥20.00、缓存命中 ¥2.00、输出 ¥100.00, 1M 上下文。消费端会员与开放平台 API 是两套账户、两条计费。截至 2026-07。[Kimi 开放平台 · 定价](#)
- 04 钛媒体 · 「Kimi K2.6 涨 58% 的 API 定价」(2026-04)——独立报道, 指出 K2.6 输入价较上一代 K2.5 上涨约 58%、输出约三成, 解读为月之暗面不再以低价抢量、转向对标前沿。本书引它作外部参照, 不作权威终点。[钛媒体 · K2.6 定价](#)

你印象里的 Kimi 是张快照,
它早长成了一套栈.

全景 · 一套对标前沿的栈。

Kimi 现在的产品线几乎是照着 OpenAI 和 Anthropic 摆的：聊天加会员、多模态模型、Agent 与 Agent Swarm、Deep Research、Kimi Code、API、开源权重。这章给整套栈画张图，每块对标谁、缺了谁。

你天天用的是那个聊天框。但点开网页顶上的导航，是一排你大概没怎么碰过的入口：Slides、Websites、Docs、Sheets、Deep Research、Agent、Agent Swarm、Kimi Code、Kimi Claw，侧边栏还新添了桌面端的 Kimi Work。把这排入口和 ChatGPT、Claude 摆在一起，你会发现两边几乎能对上号。¹ Kimi 早不是「一个会读长文的聊天框」，是一整套栈——最上面是聊天与会员，底下垫着多模态旗舰模型、一群 agent、和一套开放平台 API。这章不教任何一面怎么用（那是后面每一章的事），只做一件事：给整套栈画张图，让你看清哪一面对着你哪类活、对标的又是前沿的哪一个。

- I

这套栈，几乎是照着前沿描的。

把 Kimi 的产品线一项项摆开，你能在 OpenAI / Anthropic 那边给几乎每一项找到对应。

聊天 + 五档会员	对标 ChatGPT / Claude 的消费端 App 与订阅梯度 —— 从免费的 Adagio 到 ¥699 的 Allegro, 一条按用量铺开的会员阶梯。
K3 旗舰 + K2.7-Code	对标 GPT / Claude 的「旗舰 + 编码」那一组。K3 是 2.8 万亿参数、1M 上下文、思考永远开的旗舰; K2.7-Code 仍是编码专线。
Agent(做 Slides / Websites / Docs / Sheets)	对标把答案直接变成成品的 agentic 能力(Canvas / Artifacts 那一路)—— 它交回来的是网页、幻灯、表格, 不只是一段文字。
Deep Research	对标 OpenAI / Gemini 的 Deep Research: 自己读几百个网页、交叉核对, 出一份带出处的报告。
Kimi Code	对标 Claude Code / Codex: 跑在终端、能改你仓库的编码 agent。
开放平台 API	对标 OpenAI / Anthropic API: base_url 兼容 OpenAI, 改一行就能接到现有工具上。

——对上之后, 结论很直白: 它要的是 ChatGPT / Claude 那个货架上的位置, 而不是「国产长文本工具」那个标签。这也是为什么用两年前的眼光看它, 会一直看走眼。

一套对标 OpenAI / Anthropic 的栈

聊天到 API, 一层层都能在前边找到对应 —— 再加一样它们不给的

聊天 + 五档会员

≈ ChatGPT / Claude 应用 + 订阅

K2.6 旗舰 · 思考 · K2.7-Code

≈ 旗舰 + 推理 + 编码模型

Agent · Agent Swarm

≈ 成品生成 / 多 agent

Deep Research

≈ Deep Research(同名一面)

Kimi Code · 开放平台 API

≈ Claude Code / Codex · API

开源权重 · Modified MIT

能下载、自托管、商用 —— OpenAI / Anthropic 不给的那一样

截至 2026-06 · 来源: kimi.com · platform.kimi.com · Hugging Face

从聊天到 API, Kimi 这套栈几乎层层对得上前沿; 底下垫的开源权重, 才是 OpenAI / Anthropic 不给的那一样。

• • •

- II

两样前沿给不了的。

对标归对标, 它手里还攥着两样 OpenAI、Anthropic 给不了的东西。第一样是开源。K2 系列的权重以宽松协议(Modified MIT)挂在 Hugging Face 上, 能下载、能自托管、能商用; K3 的权重官方承诺 7 月 27 日前放出, 截至发稿还没上架 —— 所以「搬回家」这句话, 现在只对 K2 系完全成立。

² Claude、GPT 你只能租它们的 API; 要数据不出门、要长期可控, 这一条压过任何能力对比。**开源是国产这一档共同的底牌** —— 这条线和 DeepSeek 那本书是同一条。第二样是它出身带的老本行: 一口气读几十万字的长上下文、和顺手的中文成稿。如今它们只是这套栈里的「其中一面」, 不再是它的全部身份 —— 但在中文长材料这类活上, 它仍比海外模型省心。旧定位没消失, 是被降级成了一项专长。认清这一点, 你才不会再拿「它中文好」当成选它的全部理由。

这是一张图，不是一份清单。

能力早摆在那，缺的是知道哪一面对你哪类活——以及哪两条账别混。用这张图，先记两条边界。一条是账：**消费端会员和开放平台 API 是两套账户、两条计费**，会员额度再多也不能拿去跑 API，反过来也一样。³ 另一条是能力的天花板：「样样对得上」不等于「样样追平」——最难的推理、最吃打磨的活，它还在前沿身后半步。这本书会在每一面标出那条线，该回 frontier 就回。所以后面的章节就是顺着这张图走：先挑模型(K3 / K2.7-Code / K2.6)，再分清四种模式，然后一面面过 Agent、Agent Swarm、Deep Research、Kimi Code，最后落到选档与取舍。你不必每一面都学——对上你手头的活，再翻那一章。

01

列出你在别处用的 3 个 AI 产品

比如 ChatGPT 聊天、Claude Code 写码、某个 Deep Research。写下你真在用的那几个。

02

给每个在 Kimi 这张图上找对应

聊天对 Chat、Claude Code 对 Kimi Code、Deep Research 对 Deep Research……一一连线。

03

圈出你在 Kimi 也付了钱、却没用过的那几面

对得上、你却从没在 Kimi 里点开过的那几面，就是这本书你最该先翻的章节。

参考.

- 01 Kimi 官网首页(kimi.com)——截至 2026-07 首屏标注「Kimi AI with K3 | Built for Agentic Coding & Knowledge Work」, 产品入口排着 Chat / Slides / Websites / Docs / Sheets / Deep Research / Agent / Agent Swarm / Kimi Code / Kimi Claw(入口已迁 /bot), 侧边栏另有 Kimi Work(Beta)。这排入口就是它「不只是聊天框」的产品全景。[Kimi 官网](#)
- 02 Hugging Face · moonshotai(Kimi-K2.6 / Kimi-K2.7-Code 模型卡)——K2 系权重以宽松(Modified MIT)协议开源, 可下载、自托管、商用; 1 万亿参数 MoE、激活约 320 亿, 原生多模态, 256K 上下文。K3 权重官方承诺 2026 年 7 月 27 日前放出, 截至 2026-07-17 尚未上架。开源是 OpenAI / Anthropic 都不给的一条。[Hugging Face · moonshotai](#)
- 03 Kimi 开放平台(platform.kimi.com) · API 总览 —— API base_url 为 <https://api.moonshot.cn/v1>, 在请求 / 响应格式上兼容 OpenAI Chat Completions, 可直接用 OpenAI SDK。消费端会员与开放平台 API 是两套账户、两条计费。截至 2026-07。[Kimi 开放平台 · API 总览](#)

它不是一个聊天框,
是一整套栈.

挑模型 · K3、K2.7-Code 与 K2.6.

K3 是 2.8 万亿参数、1M 上下文的新旗舰, 思考永远开着; K2.7-Code 仍是编码专线, K2.6 退为平价通用。这章讲三颗脑子怎么分工、什么时候值得付 K3 的价、什么时候回 frontier。

输入框那个模型名, 你大概从没点开换过。多数时候确实不用换 —— Kimi 默认就给你当前最强那颗脑子。但 2026 年 7 月 16 日之后, 那颗脑子换了: K3 发布, 旗舰易主。仍有三件事值得你知道: 默认这颗是谁、它的思考为什么再也关不掉、什么时候该用便宜的那两颗。顺带 —— 它这次的定价, 会替你回答「该用什么眼光看它」。

- I

默认已经是 K3, 思考再也关不掉.

不动设置时, 跑的就是 K3 —— 先知道默认是什么, 才谈得上改。K3 是 2026 年 7 月 16 日发布的旗舰: 2.8 万亿参数的 MoE(896 个专家、每 token 激活 16 个), 1M(1,048,576)token 上下文, 原生能看图看视频, API 里的 id 就叫 kimi-k3。^{1 2} 和 K2 时代最大的不同: **它的思考永远开着** —— 没有开关, 顶层参数 reasoning_effort 目前只有 max 一档, 低 / 高档位官方说后续上线。¹ 要记的还是那句: 这不是一堆各管一摊的模型, 你调的是「让哪颗脑子、想多久」。

2.8 万亿 总参数 · MoE	16 / 896 每 token 激活专家	1M 上下文窗口	永远开 思考 · 不可关
----------------------------	---------------------------------	--------------------	------------------------

K3 的硬指标 —— 据官方博客与开放平台, 截至 2026-07-17; 激活参数量官方未公布。

「开源」这一条要单独说。K2 系列的权重早就挂在 Hugging Face 上(Modified MIT, 可下载、可商用); K3 官方自称「全球首个开放 3T 级模型」, 但截至 2026-07-17 权重还没放出 —— 官方承诺 7

月 27 日前发布, 协议条款要随技术报告一起公布。^{1 5} 承诺不是现货: 现在就要数据不出门的读者, 能搬回家的仍是 K2.6 和 K2.7-Code 这两颗。何况官方建议的部署配置是 64 张以上加速卡的超节点——这颗脑子本来也不是给一张消费级显卡准备的。¹

. . .

- II

三颗在列的脑子, 各管一段.

K3 上架不代表 K2 下架。开放平台现在在列的, 是三颗正经可用的脑子, 加一颗正在落日的。重活归 **kimi-k3**: 长链路编码、百万 token 的长料、要反复推的知识工作——它也最贵, 下一节算账。编码专线归 **kimi-k2.7-code**: 1 万亿参数、256K, 思考强制开启(关不掉, 传关闭会直接报错); 它还有个高速变体 **kimi-k2.7-code-highspeed**, 输出约 180 tokens/s、价格翻倍, 适合交互式结对。^{2 5 6} 日常归 **kimi-k2.6**: 同样 256K、多模态, 思考可选可关, 还能走 Batch 批量(六折)——K3 反而不在 Batch 内。^{3 4} 至于 **kimi-k2.5**: 新用户已经用不了, 老用户也只能用到 2026 年 8 月 31 日, 别把新活再绑上去。² K2.6 时代, 这章教你「活简单就关思考」。这个开关在 K2.6 上还在, 写法没变:

```
from openai import OpenAI

client = OpenAI(api_key="YOUR_KIMI_API_KEY",
                base_url="https://api.moonshot.cn/v1") # 与 OpenAI 兼容

resp = client.chat.completions.create(
    model="kimi-k2.6",
    messages=[{"role": "user", "content": "把这段会议纪要按议题归类, 列出待办"}],
    extra_body={"thinking": {"type": "disabled"}}, # 关思考: 活简单、要秒回时
)
print(resp.choices[0].message.content)
```

关掉思考的最小调用(仅 KIMI-K2.6 支持; K3 与 K2.7-CODE 的思考关不掉, 传 DISABLED 会报错)

- 实践提示

给一条线：要快、要省、答错了一眼能看出来的活 → K2.6 关思考；交互式写码、要它盯着改 → K2.7-Code(赶时间换高速版)；长链路、大仓库、百万字长料、答错了代价大的活 → K3。选模型在 K3 之后变成了「选思考深度」——你不再开关它，你选一颗天生就想那么久的脑子。

• • •

- IIII

价格与跑分，都分两摞看。

第三条线：标价说明它把自己摆在哪儿；而跑分，你得先分成两摞，再决定信不信。先看价格。每百万 token，K3 输入 ¥20、输出 ¥100 —— 输出是 K2.6 的近四倍；国际站 \$3 / \$15。缓冲在缓存：命中只要 ¥2，官方称编码负载下缓存命中率超九成(厂商口径，你的里程以账单为准)，长链路 agent 的重复前缀大多走命中价。**3 1** 两年前 K2.5 卖 ¥4 / ¥21 的那个 Kimi 已经走远了：从 K2.6 涨价六成，到 K3 直接定在 frontier 价位，它要抢的从来不是「便宜」那张货架。冲省钱该看 DeepSeek 那本；选 Kimi，是冲这套栈和这颗脑子。

模型	输入	缓存命中	输出	上下文
kimi-k3 旗舰 · 思考永远开	20.00	2.00	100.00	1M
kimi-k2.7-code 编码专线 · 强制思考	6.50	1.30	27.00	256K
kimi-k2.7-code-highspeed 编码高速版 · 约 180 tokens/s	13.00	2.60	54.00	256K
kimi-k2.6 平价通用 · 思考可关 · Batch 六折	6.50	1.10	27.00	256K
kimi-k2.5 上代 · 2026-08-31 下线	4.00	0.70	21.00	256K

¥ / 百万 token · 官方 platform.kimi.com 定价; K3 不在 Batch 批量折扣内。

再看跑分 —— 先分两摞。**第一摞, 厂商自评:** K3 发布博客里的数字, 多数是官方用自家 harness 跑出、未经第三方榜单收录; 而官方在同一篇博客里写明, 整体表现仍「trails the most powerful proprietary models」, 点名 Claude Fable 5 和 GPT 5.6 Sol, 并在 Limitations 里承认与这两家有「a noticeable gap in user experience」, 还警告 K3 有过度自主倾向、可能替用户做意料外的决定。¹

第二摞, 独立实测: 发稿时刚开始落地 —— DeepSWE 官方榜(2026-07-17 更新, 全榜统一 mini-swe-agent harness)已收录 K3: 69%±5%、16 个模型第 4, 距榜首 73% 只差 4 个点、误差棒与二三位重叠, 单任务成本 \$4.65 在前五里第二低; 同一张榜上 K2.7-Code 是 31%±1、第 14。SWE-bench Verified 则仍无 K3 条目, Kimi 最好成绩还是 K2.5 的 70.8%。^{7 8}

第三摞, 偏好榜: 一手运营的 Code Arena(WebDev)榜上 K3 以 1679 分登顶(约 48 万票)—— 但它是人类盲投的 Elo 偏好榜, 比执行类基准软, 偏好不等于正确性。⁹ 所以「什么时候回 frontier」这条线没变, 只是更好说了: 一锤定音的最难活(架构级的编码、多跳推理、不能错的输出)回 Claude / GPT —— 这不是我们替它谦虚, 是 Moonshot 自己写在发布博客里的。¹ K3 值得付钱的场景同样清楚: 百万 token 一次吃完的长料、要跑很久的 agent 任务、国内直连的顺手, 以及(7 月 27 日之后)可自托管的开放权重。中间的日常活, K2.6 依旧够用。

— 注意

别在发布第一周拿跑分做决定。厂商自评用自家 harness, 独立执行类榜单刚收录一家, 偏好榜又偏软 —— 三摞证据现在都单薄。你自己最棘手的那件活, 才是唯一算数的基准; 下面三步就是帮你把它跑出来的。

01

挑一个你真卡住的长链路任务

跨多文件的功能改动、上百页的材料分析 —— 要有多步、要推理, 不是查个事实那种。

02

K2.6 和 K3 各跑一遍

同一个任务, 记下答案质量差、等待时间差, 以及开放平台后台的账单差 —— 缓存命中会让账单差小于标价差。

03

给自己定一条「什么活才配 K3」的线

按这一次的体感写下来, 贴在你看得到的地方。模型挡位是最后 10% 的微调, 不是把活说清楚的替代。

参考.

- 01 Kimi K3 官方博客(2026-07-16)—— 2.8 万亿参数 MoE(896 专家、每 token 激活 16)、1M token 上下文、原生视觉;思考固定 max 档,low / high 档后续上线;「full model weights will be released by July 27, 2026」,协议随技术报告公布;自评整体仍「trails the most powerful proprietary models」(点名 Claude Fable 5 与 GPT 5.6 Sol),Limitations 承认用户体验有明显差距、可能替用户做意料外的决定;建议 64 张以上加速卡部署;编码负载缓存命中率超九成(厂商口径)。激活参数量官方未公布。[Kimi K3 官方博客](#)
- 02 Kimi 开放平台 · 模型列表 —— kimi-k3 为「迄今能力最强的模型」;kimi-k2.7-code-highspeed 输出约 180 tokens/s(短上下文可达 260);kimi-k2 全系 2026-05-25 已下线;kimi-k2.5 与 moonshot-v1 系列停止向新用户开放,2026-08-31 全平台下线。截至 2026-07-17。[Kimi 开放平台 · 模型列表](#)
- 03 Kimi 开放平台 · 定价(chat-k3 及 chat-k27-code / chat-k26 / chat-k25 同路径各页)—— 每百万 token(缓存命中 / 缓存未命中输入 / 输出):kimi-k3 ¥2.00 / ¥20.00 / ¥100.00,国际站 \$0.30 / \$3.00 / \$15.00;kimi-k2.7-code ¥1.30 / ¥6.50 / ¥27.00,高速版 ¥2.60 / ¥13.00 / ¥54.00;kimi-k2.6 ¥1.10 / ¥6.50 / ¥27.00;kimi-k2.5 ¥0.70 / ¥4.00 / ¥21.00。截至 2026-07-17。[Kimi 开放平台 · 定价](#)
- 04 Kimi 开放平台 · Batch API 定价 —— 批量为标准价六折,仅支持 kimi-k2.7-code / kimi-k2.6 / kimi-k2.5;kimi-k3 不在 Batch 内。截至 2026-07-17。[Kimi 开放平台 · Batch 定价](#)
- 05 Hugging Face · moonshotai/Kimi-K2.7-Code 模型卡 —— 1 万亿参数 MoE、激活约 320 亿,256K 上下文,仅思考模式(不可关闭),权重 Modified MIT 开源。Kimi-K2.6 卡同规模同协议、思考可选。截至 2026-07-17,moonshotai.org 下无任何 K3 权重。[Hugging Face · Kimi-K2.7-Code](#)
- 06 Kimi 开放平台 · API 总览 —— 思考经 SDK 的 extra_body 传 thinking 参数:kimi-k2.6 可选关闭,kimi-k2.7-code 恒为 enabled(传 disabled 报错);kimi-k3 不使用 thinking 参数,走顶层 reasoning_effort(当前仅 max 一档)。截至 2026-07-17。[Kimi 开放平台 · API 总览](#)

- 07 DeepSWE 官方榜单(页面标注更新于 2026-07-17,113 个任务、16 个模型,统一 mini-swe-agent harness)—— kimi-k3[max] 69%±5%,第 4(榜首 gpt-5.6-sol[max] 73%±3%,单任务成本 \$4.65 为前五中第二低);kimi-k2.7-code 31%±1%,第 14。独立第三方实测,这是独立榜首个收录 K3 的成绩。[DeepSWE 官方榜](#)
- 08 SWE-bench Verified 官方榜单数据(官方 GitHub 原始 JSON)—— Kimi 最佳条目为 mini-SWE-agent + Kimi K2.5(high reasoning)70.8%(2026-02-17),同期榜首 79.2%;K2.6 与 K2.7-Code 从未提交;K3 发布不足 48 小时,截至 2026-07-17 无任何条目。[SWE-bench 官方榜](#)
- 09 Code Arena(WebDev)官方榜单(Arena Intelligence 运营,2026-07-17 亲见)—— Kimi K3 以 1679 分居首(约 48 万票)。人类盲投的 Elo 偏好榜,比执行类基准软;偏好不等于正确性,本书仅作参考。[Code Arena · WebDev 榜](#)
-

选模型,选的是思考深度
和账单厚度。

四模式 · 快答、深想、交办、集群。

输入框那几个模式——Instant / Thinking / Agent / Agent Swarm——是 Kimi 把「问答」和「做事」分开的旋钮。挑错了要么慢、要么不够使。这章给一条线。

Kimi 的输入框不只是个打字框——旁边有一个你大概没怎么动过的模式开关：Instant、Thinking、Agent、Agent Swarm。多数人一直留在默认那档，没意识到换一档，它干的活就完全是另一回事。这四个模式，是 Kimi 把「答你一句」和「替你做成」分开的旋钮。¹挑错了，要么简单问题让你干等、要么把一句话能答的事支起一整套 agent。这章给一张对照：哪一档对哪类活，以及一条贯穿始终的判断——挑能办成事的**最低**那一档。

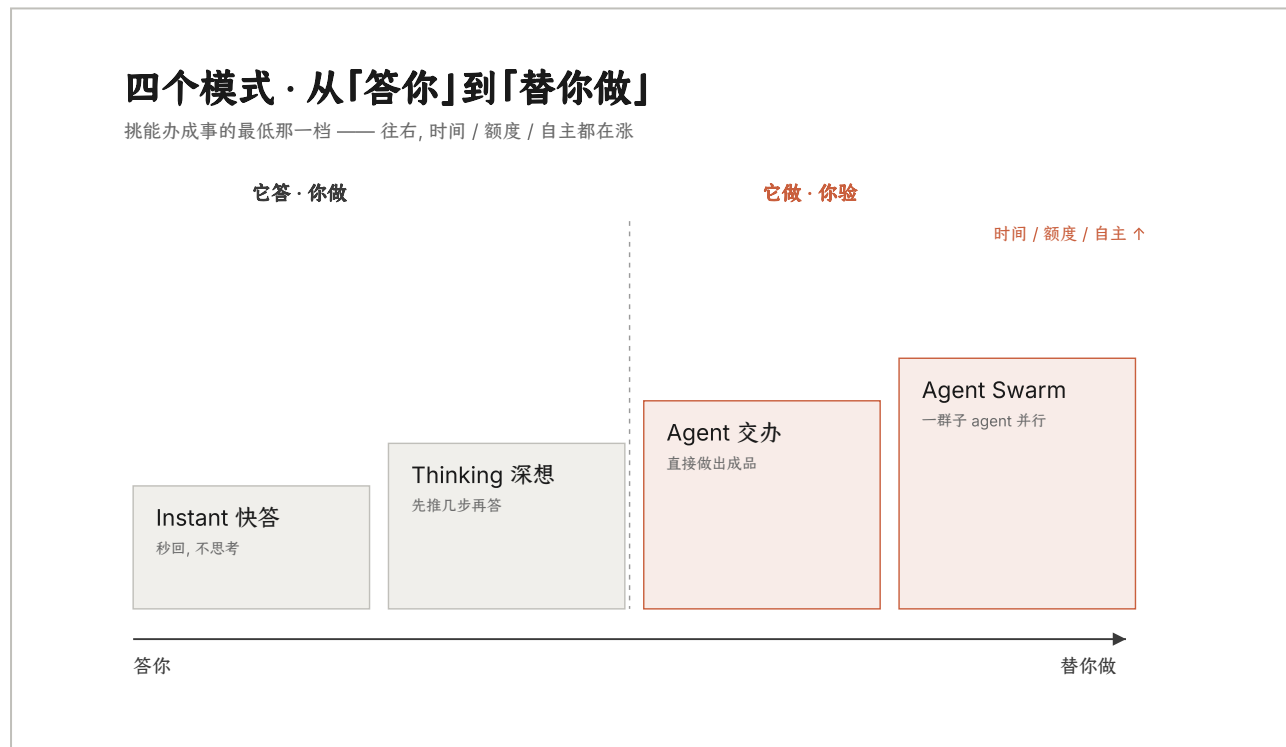
- I

四个模式，是从「答」到「做」的四档。

它们不是四个并列的功能，是一条从「给你一句话」到「替你干一整件事」的阶梯。

Instant · 快答	不开思考, 秒回。查事实、改错字、闲聊、要立刻拿到一句话 —— 这一档够。
Thinking · 深想	先在内部推几步再答。多步推理、算账、写码、要核对的活。第三章细讲这条线。
Agent · 交办	不只给答案, 直接做出网站 / 幻灯 / 文档 / 表格等成品。你要的是一份能交的件, 不是一段文字 —— 细看第五章。
Agent Swarm · 集群	派出一群子 agent 并行干, 扛大规模的查找、处理、多视角综合。大活才动用 —— 细看第六章。

一句话记住分界: 前两档(Instant / Thinking)是「它答、你做」, 后两档(Agent / Agent Swarm)是「它做、你验」。你这次到底要一段回答, 还是要一件成品, 决定了你该停在哪一档。



Instant → Thinking → Agent → Agent Swarm, 越往右它替你扛得越多, 时间、额度、自主也越大。

• • •

- II

往「做」那头走, 代价也一档档涨。

越往右, 它替你扛得越多, 你付出的时间、额度和风险也越大。Instant 几乎不花什么; Thinking 拿时间换深度(第三章那条线); Agent 一跑就吃掉共享额度池里的一份(各档约合 6-360 个)和并发额度, 还得等它把成品做完; Agent Swarm 更猛, 派一群子 agent, 烧的是按档限次的集群额度(Moderato 25 次 / Allegretto 50 / Allegro 120), 一次消耗约为普通 Agent 任务的数倍, 而且目前还是 Beta。² 代价不只是钱和时间——越自动的模式, 越可能**把一个偏掉的方向一口气做到底**。它做得越多, 你越要在它出发前就把要的东西说清。

• • •

- III

默认够用, 别为「更高级」上更高的档。

一个常见的错觉是「上更高级的模式 = 更好的结果」。多数日常活, 停在前两档才对。挑模式的标准只有一个: 这件活的**形状**。要一句答案, 用 Instant; 要一段讲得通的推理, 用 Thinking³; 要一份能直接交的成品(PPT、网页、表), 才上 Agent; 要并行处理一大批、或要多个独立视角, 才动 Agent Swarm。反过来硬套只会浪费: 拿 Agent 去问一个事实, 是等十分钟换一句本该秒回的话; 拿 Swarm 去做一件单线就能办的事, 是为不存在的规模付费。所以这章的判断就一句: 先看活的形状, 再挑能办成它的最低那一档。能用 Instant 解决的, 别开 Thinking; 一段文字够用的, 别叫 Agent。把模式当工具挑, 不当「等级」追。

01

挑最近 3 件你交给 Kimi 的活

翻记录, 挑 3 件不同类型的: 比如查个资料、改段代码、做份汇报。

02

给每件标一个「最低可行模式」

按活的形状判断: 它要的是一句话、一段推理、还是一份成品 / 一批处理? 标上 Instant / Thinking / Agent / Swarm。

03

对比你当时实际用的那档

多半你要么全停在默认、没用上该用的 Agent, 要么反过来给小事上了大模式。记下这个差, 就是你以后挑档的依据。

— 引用与参考

参考.

- 01 Kimi 帮助中心 · Agent 模式 —— 截至 2026-07, 输入框可在 Instant / Thinking / Agent / Agent Swarm 之间切换; Agent 模式不只回答, 直接产出网站 / 幻灯 / 文档 / 表格等成品。 [Kimi 帮助中心 · Agent](#)
- 02 Kimi 帮助中心 · Agent Swarm —— Agent Swarm 是多智能体协作模式, 派出一群子 agent 并行干活、各自独立再归并, 面向大规模查找 / 处理 / 多视角的任务; 现由 Kimi K3 驱动(K3 集群)。截至 2026-07 仍带 Beta 说明, Moderato 档起可用(25 次 / Allegretto 50 次 / Allegro 120 次)。 [Kimi 帮助中心 · Agent Swarm](#)
- 03 Kimi 开放平台 · API 总览 —— 思考(深想)在 kimi-k2.6 上可显式关闭(type: disabled, 经 extra_body 传); kimi-k3 与 kimi-k2.7-code 的思考永远开启。输入框里 Instant(关思考)与 Thinking(开思考)的切换对应这个开关。截至 2026-07。 [Kimi 开放平台 · API 总览](#)

四个模式, 从「答你」到「替你做」;
挑能办成事的最低那一档。

交办 · Agent 直接做成品.

Agent 模式不只给答案, 它做出网站、幻灯、文档、表格。这章讲什么时候该把整件活交给它, 什么时候它只是花架子。

网页对话是一问一答; Agent 模式是另一回事 —— 你说清要什么, 它直接交你一个能用的成品: 一份排好版的 PPT、一张算好的数据表、一页能上线的网页, 而不是一段还得你自己去做的文字。Agent 是这套栈里「替你做、不只答你」的那一面。¹ 这章讲什么时候该把整件活交给它, 什么时候它只是把活做快、做砸; 至于一次要派一群子 agent 的大场面 (Agent Swarm), 留到下一章。

- I

Agent 交的是成品, 不是文字.

它的产出不是一段话, 是一个文件 —— 这是它和默认对话最实的区别。默认对话给你一段文字, 你拿着文字自己去做 PPT、做表、建站。Agent 模式跳过这一步: 你描述要什么, 它直接交一份排好版的 PPT、一张算好的 Excel、一页能上线的网页。2026 年 4 月它又升级了全栈能力 —— 后端数据库、用户登录、版本管理、一键发布、代码导出, 做的网站不再只是静态页。¹ 给一个能截图的对照: 同一个需求, 默认对话给你一段文字大纲, Agent 模式给你一份能直接发出去的导出件。所以它真正替你省下的, 不是「想内容」那一步, 是「把内容落成一份像样文件」那几小时 —— 排版、套模板、把数据填进表格、把页面搭起来。内容你心里有数时, 它把这层体力活接过去; 这正是它和聊天框的分界。

• • •

它适合「结果说得清」的活。

你说得清，它做得成；你说不清，它做得快—— 飞快地做出一个不是你要的东西。Agent 模式吃「明确」。结果形态清楚（要一份 10 页的季度汇报 PPT）、素材现成（数据和要点你给齐）、能验收（做完你一眼看得出成没成）—— 这种活最适合交办，也最吃共享额度池里的量。² 反过来，开放式、要反复试方向的活（「帮我想想这个产品怎么定位」），留在对话里来回更省；交给 Agent，它只会一口气把一个还没定准的方向做到底。

— 实践提示

一个实用习惯：**把验收标准前置**。动手前先用一句话写清「做成什么样算成」，再放手。它跑得越自动，你越要在它出发前就把终点画好—— 否则你拿回来的，是一个做得很完整的错东西。

举个真实场景：季度汇报要一份十几页 PPT，数据和结论你都有，缺的只是排版那两小时。与其自己套模板，不如把要点喂给 Agent，用这样一段 brief 让它一次到位—— 注意验收标准写在了最前面：

提示词

Kimi · Agent

帮我做一份季度业务汇报 PPT。

【验收标准】15 页以内；每页一个要点；有数据的页必须配图表；风格简洁商务、不要花哨动画。

【素材】（在此粘贴你的数据 / 要点 / 上季度结论）

【结构】封面 → 业绩概览 → 三个重点项目 → 风险 → 下季度计划。

【硬约束】数字只用我给的、不要自己编；拿不准的地方留空并标「待补」。

先按以上出初稿，我看完再让你改。

帮我做一份 Q2 销售复盘 PPT。

【验收标准】12 页以内；每页一个要点；有数据的页必须配图表；风格简洁商务、不要花哨动画。

【素材】Q2 总营收 1,840 万(环比 +12%)、新签客户 37 家、华东大区超额 18%、华南未达标(完成 76%)、退订率从 5.1% 降到 3.4%。

【结构】封面 → 业绩概览 → 三个重点大区 → 风险(华南缺口) → Q3 计划。

【硬约束】数字只用我给的、不要自己编；拿不准的地方留空并标「待补」。

先按以上出初稿，我看完再让你改。

• • •

- III

它把对的事做成，也把错的事做全。

自动化的好处和风险是同一件事：它会把你交代的方向一做到底——不管那个方向对不对。所以交办的边界很清楚。适合交给它的：形态确定、你验收得了的成品活。不该全交给它的：需要你边做边改判断的活、一锤定音不容错的活、以及你根本验收不了对错的活(你自己都判断不了成品好坏，就别让它替你做主)。它能替你把活落成文件，替不了你定这活该往哪做。一句话收口：Agent 是你的执行手，不是你的决策人。把判断留给自己，把落地交给它——这条线划对了，它替你省下的是实打实的几小时；划错了，你拿回来的是一份返工成本更高的半成品。

01

挑一件你常花两小时排版的活

选一件结果说得清的：一份汇报 PPT、一张要整理的数据表、一页小网站。

02

先写验收标准，再交给 Agent

用一句话写清「做成什么样算成」，再让它跑，别中途手痒接管。

03

记下它做到几成、你补了多少

对比成品与你的标准，记录一次到位的比例和你的返工量，给它定一条「值得交办 vs 自己来」的线。

参考.

- 01 Kimi 帮助中心 · Agent 模式 —— 截至 2026-07, 在输入框切到 Agent, 它不只回答, 而是直接产出网站 / 幻灯 / 文档 / 表格等成品; 2026 年 4 月升级全栈能力(后端数据库、用户登录、版本管理、一键发布、代码导出); 单次任务约 5-20 分钟, 上下文 128K token(约 10 万汉字)。 [Kimi 帮助中心 · Agent](#)
 - 02 Kimi 帮助中心 · 会员套餐价格 —— Agent 用量与并行任务数随档位上升(Adagio 约 6、Andante 约 30、Moderato 约 60、Allegretto 约 150、Allegro 约 360); 该次数为共享 token 额度池按常见任务折算的估值, Kimi Code 为单独额度池。截至 2026-07。 [Kimi 帮助中心 · 会员价格](#)
-

你说得清, 它做得成;
你说不清, 它做得快.

大活 · Agent Swarm 与 Kimi Claw.

一个 agent 不够, 就放一群。Agent Swarm 能调度上百个子 agent 并行干大活; Kimi Claw 是更新、更险的一条线。这章讲什么活值得动用这么大的阵仗。

一个 agent 一次只能顺着一条线干。可有些活不是「难」, 是「多」—— 要翻一百个网页、要把四十份 PDF 读成一篇综述、要同时从十个角度看一个问题。这种活, 一个 agent 排队干到天黑, Kimi 干脆放一群上去。这一章讲两条「大活」的线: **Agent Swarm** (用规模换时间, 一次派出一群子 agent 并行) 和 **Kimi Claw** (用自主换托管, 一个 7×24 替你动手的常驻 agent)。两条都猛, 也都各有一笔你必须先认清的代价。

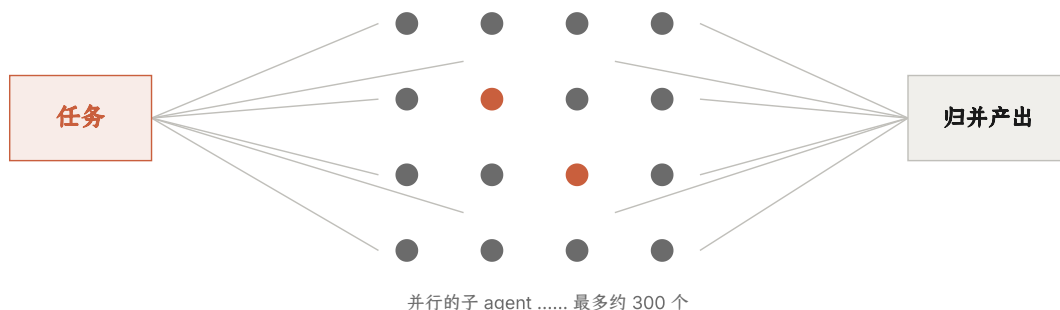
- I

Agent Swarm 换的是「串行排队」的时间。

它的本事不是更聪明, 是更多、更并行 —— 把一件能拆开的大活, 同时摊给几百个分身去干。到 K3 时代, Agent Swarm 改由 K3 驱动 (官方叫「K3 集群」), 一次任务最多能派出约 **300 个子智能体** 同时工作、跑超过 **4,000 次工具调用**, 比单个 agent 一步步串着干约快 4.5 倍 —— 这组数字还是 K2.6 时代那组, 官方没随 K3 更新。¹ 它扛的是「规模型」的活: 大规模检索、批量处理上百份文档、产出十万字以上的报告、复杂编程。给个对照: 单 Agent 是一个人加班到天亮, Agent Swarm 是一支临时组起的小队, 分头干完再归并。

Agent Swarm · 一个任务, 一群分身

拆得开的活才值 —— K2.6 最多约 300 个子 agent、4000+ 次工具调用、约 4.5× 快



适合: 大规模检索 · 上百份文档 · 多视角综合 —— 一步接一步的线性活别用。

截至 2026-06 · Beta · Allegretto / Allegro 档 · 来源: kimi.com 帮助中心

一个任务拆给最多约 300 个子 agent 并行, 再归并成产出。只对「拆得开」的大活值, 线性活别用。

但规模是有标价的。Agent 集群按档限次、且消耗约为普通 Agent 任务的数倍, 目前仍带 Beta 说明, Moderato(¥99/月)档起有 25 次, Allegretto 50 次、Allegro 120 次 —— 门槛比本书上一版写的时候低了一档。^{1 3} 所以它不是「更高级的默认」, 是一件你确认值得才动用的重型工具。

· · ·

— II

它只对「拆得开、并得了」的活值。

判断要不要上 Swarm, 只看一件事: 这活能不能拆成一堆互不依赖、可以同时干的小块。Swarm 的甜区是三种「规模」: 一是**查得广**(把散在上百处的信息一次捞齐), 二是**产得多**(一口气读完几十份材料、写成一份长报告), 三是**看得全**(同时让多个分身从不同角度看一个问题, 逼出分歧而不是抱团附和)。这三种活的共同点是: 能切成并行的块。¹ 反过来, 一件必须一步接一步、后一步等

前一步结果的线性活，给它一支 swarm 毫无意义——分身们彼此干等，你却付着数倍的额度。一个人能顺手办完的事，别叫一支小队来开会。先问「这活拆得开吗」，拆不开，留在单 Agent 甚至对话里更省。举个值得动用 swarm 的真实场景：一周内摸清一个新赛道——几十家公司的定价、定位、最近融资。一个 agent 串着查要查到天黑，而这种「查得广 + 产得多」正好拆得开。可以这样交代它：

提示词

Kimi · Agent Swarm

任务：盘点「国内 AI 编程工具」这个赛道。

【并行去做】先列出 ≥ 30 家，对每一家分别查：定价档位、目标用户、最近一轮融资、一句话差异点。
【归并产出】汇成一张表（公司 / 定价 / 用户 / 融资 / 差异），再写一页总览：谁在拼低价、谁在拼集成、空白点在哪。
【要求】每个数据点标来源链接和日期；查不到就写「未公开」，不要猜。

任务：盘点「国内面向中小商家的 SaaS 客服系统」这个赛道。

【并行去做】先列出 ≥ 30 家，对每一家分别查：定价档位、目标客群（行业 / 规模）、是否支持全渠道接入、一句话差异点。
【归并产出】汇成一张表（公司 / 定价 / 客群 / 全渠道 / 差异），再写一页总览：谁在拼低价、谁在拼 AI 自动回复、空白点在哪。
【要求】每个数据点标来源链接和日期；查不到就写「未公开」，不要猜。

· · ·

- III

Kimi Claw: 能力更大, 那条线也更紧.

Agent Swarm 是「一次干一票大的」, Kimi Claw 是「一直替你干」——自主度高一截, 你要守的线也紧一截。Kimi Claw 没有死, 是搬了家: 入口从 /claw 迁到了 kimi.com/bot, 形态也从「云端常驻 agent」一个长成了一族——云端 7×24、桌面端、安卓端三种部署, 加上微信 / 飞书 / 企微 / 钉钉 / 微博的 IM 机器人, 还有一个 Preview 状态的多 agent 协作空间「Claw 群聊」。7×24 常驻、40GB 云存储、ClawHub 5,000+ 技能、定时任务这些老本事都在, 内置 Kimi 模型、免你配 API。² 它能干的事, 比「做一份 PPT」大得多——因为它不只动脑, 还常驻、联网、能替你动手。两处细节

该知道：一，Claw 本体的 Beta 角标摘了，但新形态「Claw 群聊」仍标着 Preview —— 预发布的帽子是换了地方，不是摘完了，别当整条线已转正；二，默认模型官方两页说法不一（/bot 写 K2.6 Thinking、官方资源页写 K3），以你部署时页面实际为准。档位要求没变，仍是 Allegretto（¥199/月）起。2 3

— 注意

但「常驻 + 联网 + 能动手」三样叠在一起，意味着授权给到哪、数据落在哪、出了错谁兜底，你都得先想清楚。截至 2026-07，它的 Beta 角标摘了、新形态群聊仍标 Preview，档位也仍要 Allegretto（¥199/月）以上。3 这是典型的「能用」不等于「现在该押上去」：普通用户先把前面的 Agent 交办用熟，这条线等你想清边界再碰也不迟。能替你 24 小时动手的 agent，也能 24 小时替你闯祸。

01 找一件「多而不难」的活

挑一件量大、且能拆成并行小块的：比如把几十份资料读成一份综述、跨上百个来源做一次盘点。

02 用 Agent Swarm 跑一遍，盯住额度

交给它并行干，记下它比你预期快了多少、又吃掉了普通 Agent 几倍的额度。

03 给自己定一条「值得动用 swarm」的线

写下「什么样的活才值得付这几倍额度」。Kimi Claw 这种常驻 agent，先只拿无关紧要的定时任务试手，别一上来就交要紧事。

参考.

- 01 Kimi 帮助中心 · Agent Swarm —— 截至 2026-07, Agent Swarm 由 Kimi K3 驱动 (K3 集群); 单次任务最多部署约 300 个子智能体同时工作、超过 4,000 次工具调用, 较单 agent 串行约快 4.5 倍 (该组数字为 K2.6 时代沿用, 未随 K3 更新); 适合大规模检索、批量处理 100+ 文档、十万字以上产出、复杂编程。仍带 Beta 说明, Moderato 档起可用 (25 次 / Allegretto 50 次 / Allegro 120 次), 消耗约为普通 Agent 任务的数倍额度。[Kimi 帮助中心 · Agent Swarm](#)
- 02 Kimi Claw 落地页 (kimi.com/bot) 与帮助中心 kimi-claw 专栏 —— 7×24 常驻 agent, 云端 / 桌面 / 安卓三种部署形态, 另支持微信 / 飞书 / 企微 / 钉钉 / 微博机器人与多 agent 协作空间「Claw 群聊」(Preview); 40GB 云存储、ClawHub 上 5,000+ 技能、定时任务, Allegretto (¥199/月) 档起; Claw 本体页面已无 Beta 标记, 新形态 Claw 群聊标 Preview (预发布); 默认模型两页表述不一 (/bot 写 Kimi K2.6 Thinking、官方资源页写 Kimi K3)。旧入口 /claw 现 302 回首页。截至 2026-07-17。[Kimi Claw 落地页](#)
- 03 Kimi 帮助中心 · 会员套餐价格 —— Agent 集群 (Swarm) 按档限次: Moderato 25 次 / Allegretto 50 次 / Allegro 120 次 (门槛已自 Allegretto 下放到 Moderato), 消耗高; 「一键部署 Kimi Claw」自 Allegretto (¥199/月) 档起。截至 2026-07。[Kimi 帮助中心 · 会员价格](#)

能力越大,
你要守的那条线越紧.

查证 · Deep Research 出带出处的报告。

有些问题要的不是一段回答，是一份每条结论都带出处的报告。Kimi 的 Deep Research 会自己读几百个网页、交叉核对。这章讲什么活值得等这一趟。

有些问题你要的不是一段聊天回答，是一份能拿去用、每条结论都带出处的报告——选品、政策口径、一个行业的现状。Kimi 的 Deep Research 会自己开一堆网页、交叉核对、写成结构化文档——这一面对标的是 OpenAI、Gemini 的 Deep Research。¹ 代价是它要跑几分钟。这章讲什么活值得等这一趟，什么活普通对话三秒就够。

- I

深度研究换的是「自己开 20 个标签页」那一下午。

它替你省下的，是多来源检索、逐条核对、再整理成文的那一整个下午。深度研究和「联网搜一下」不是一回事。普通搜索给你几个链接和一段综述；深度研究由 Kimi 自研的 Kimi-Researcher 驱动，会把一个复杂问题拆开，自己一轮轮地搜、读、调整方向——官方给的画像是：每个任务平均 23 步推理、规划 74 个关键词、找到 206 个网址，再筛出信息质量最高的前 3.2%，最后交一份带目录、分章节、每条都能溯源的长报告，平均万字以上、引用约 26 个信源，还能导出 PDF / Word。¹ 这趟不是免费的：官方口径要跑 10-25 分钟，一次约吃掉月度额度的 5-10%（以 Moderato 档为参考，失败会自动返还）。¹ 所以它适合那种「你本来就要花一下午、开二十个标签页」的活，不适合随手一问。普通搜索给你链接，深度研究给你一份能交出去的报告。

• • •

带出处, 才是它和普通搜索的分水岭.

一份没有出处的报告只是更长的猜测; 出处, 才是你敢把它交出去的底气。深度研究最值钱的不是它写得长, 是它每条结论都挂着来源。¹ 但「挂着来源」不等于「来源支持这句话」—— 它偶尔会把相关当因果、引到一篇营销稿、或拿一个过时数据当现行。用法是: 别只读它的结论, 抽查它引的两三个链接, 看是否真撑得起那句话。让它把每条关键结论都标上来源和日期, 你复核起来最快。它的上下文是 128K token(官方折算约 6-10 万汉字), ² 够读一大批材料, 但也意味着来源太多太杂时它一样会取舍。出处是它给你的把手, 不是替你免掉核对 —— 它替你读了一堆来源, 替不了你点开核对那一下。把「让它标来源和日期」做成一段可复制的开场, 比你临时想怎么问省心 —— 一个选品调研的真实例子:

提示词

Kimi · Deep Research

帮我调研: 2026 年国内家用储能电池的主流品牌与选购要点。

【输出】一份分章节报告: 市场概况 / 主流品牌对比(≥5 家: 容量、价格区间、质保、口碑) / 选购避坑 / 结论建议。

【硬要求】每条关键结论后用方括号标来源和发布日期, 如 [来源名, 2026-05]; 区分「厂商宣传」与「第三方实测」。

【边界】信息有冲突时并列呈现、别替我下单一结论; 价格这种时效性强的只给区间并注明可能变动。

帮我调研: 2026 年国产开源大模型在企业私有化部署上的主流选型与落地要点。

【输出】一份分章节报告: 整体格局 / 主流模型对比(≥5 家: 参数规模、开源协议、显存需求、推理成本、社区活跃度) / 私有化避坑 / 结论建议。

【硬要求】每条关键结论后用方括号标来源和发布日期, 如 [Hugging Face 模型卡, 2026-04]; 区分「官方宣传指标」与「第三方实测榜单」。

【边界】信息有冲突时并列呈现、别替我下单一结论; 推理成本这种时效性强的只给区间并注明随硬件与版本变动。

• • •

不是每个问题都值得等这一趟。

查个事实、要个定义，普通对话三秒就给你；动用深度研究只是让你白等几分钟。给选择标准。值得走这一趟：多来源、要交叉对照、结论你要拿去做决定、你愿意为准确多等几分钟。不值得：单点事实、定义性问题、你其实只想要个起点——这些普通对话或一次搜索就够，开深度研究是高射炮打蚊子。还有一类它救不了的：时效性极强的问题。它的来源可能本身就旧，报告再工整也跑不赢现实。要最新行情、今天的政策口径，别指望它——它给的是「截至它检索那一刻能搜到的」，不是「此刻正在发生的」。

01

挑一个你最近真要查的问题

选一个你本来就要花时间查的中文问题：选品、政策口径、一个行业的现状。

02

让深度研究出一份报告

要求它对每条关键结论标来源和日期，方便你回头核。

03

点开 3 条出处核对

抽查它引的三处，看是否真支持那句话，记录它帮你省下的查证时间和它编错的地方。

参考.

- 01 Kimi 帮助中心 · 深度研究介绍 —— 由 Moonshot 自研的 Kimi-Researcher 驱动: 每个任务平均 23 步推理、规划 74 个关键词、找到 206 个网址并筛出前 3.2%; 报告平均万字以上、平均引用约 26 个可溯源信源, 附可交互 HTML, 可导出 PDF / Word; 通常需 10-25 分钟, 一次约消耗 5-10% 月度额度(以 Moderato 档为参考), 失败自动返还。对标 OpenAI / Gemini 的 Deep Research。截至 2026-07。 [Kimi 帮助中心 · 深度研究](#)
 - 02 Kimi 帮助中心 · 深度研究上下文长度 —— 截至 2026-07, 深度研究(Kimi-Researcher)的上下文为 128K token(官方折算约 6-10 万汉字, 单次输出约 8K-16K token); 够它读一大批材料, 但来源太多太杂时它一样要取舍。 [Kimi 帮助中心 · 深度研究上下文](#)
-

省得了通读的力气,
省不了核对的责任.

写码 & 接 API · 把 Kimi 配进你的 Agent.

Kimi Code 是跑在终端、对标 Claude Code 的第一方编码 agent; 开放平台有 OpenAI 与 Anthropic 两套兼容协议, Claude Code、OpenCode、Pi 改几行配置就能接, Codex 要借 CC-Switch 的本地路由。这章给每个 agent 的最小配置, 也说清哪里接不了。

如果你完全不碰代码, 这章可以跳过 —— 这正是把它单拎出来的原因。这章给开发者三道门: 一道是 **Kimi Code**, 跑在终端、对标 Claude Code 的第一方编码 agent; 一道是**开放平台 API** —— OpenAI 与 Anthropic 两套兼容协议, 改几行配置就能把你已经在用的 agent 接过来; 还有一道是 **CC-Switch**, 把多套供应商配置收进一个面板、一键切换。便宜不是它的卖点了(第三章那次定价说过), 但「国内可达 + 两套协议 + 开放权重」这三样, 对国内开发者仍是少有的顺。

- I

Kimi Code 是第一方的终端 agent.

它不是在聊天里贴代码, 是直接进你的仓库改 —— 读文件、跑命令、提改动。Kimi Code 是跑在终端里的命令行工具, 对标 Claude Code: 它在你的项目目录里直接读多个文件、跑命令、改文件, 一个完整的「改、跑、看、再改」循环, 不用你在浏览器和编辑器之间来回搬。¹ 后端模型现在是 K3 与编码专线的 kimi-for-coding(即 K2.7-Code 那条线, 第三章)—— 配上长程编码的底子, 它扛得住「改一个跨十几个文件的功能」, 而不只是补全一个函数。

```
# 官方脚本
curl -fsSL https://code.kimi.com/kimi-code/install.sh | bash

# 或 npm(需 Node.js ≥ 22.19.0)
npm install -g @moonshot-ai/kimi-code

kimi --version    # 验证
```

安装 KIMI CODE CLI(MACOS / LINUX 二选一;WINDOWS 用官方 POWERSHELL 脚本, 见文档)

装完用 /login 登录, 两条路任选: **会员 OAuth**(走你的 Kimi 会员订阅额度)或 **开放平台 API Key**(走按量计费)。¹ 这两条路的账完全分开: 会员那条的模型与上下文还按档位解锁——k3 要 Moderato 档及以上, 1M 上下文要 Allegretto 档及以上; 会员额度约为每 5 小时 300–1200 次请求, 并发上限 30。¹ ² 要记牢的是: 会员的 coding 端点(api.kimi.com/coding)与开放平台(api.moonshot.cn)是**两个完全独立的账号体系, Key 互不通用**——加上开放平台内部还分国内 / 国际两套(下一节), Kimi 一侧其实有三套账户, 配错哪一个都是 401。²

— 补充

Kimi Code 还能反向接别家: 它的 provider 配置支持 Anthropic Messages 协议, 你哪天想把同一个终端工具接到 Claude 上, 改配置即可。工具是中立的, 选模型是你的自由。

• • •

— III

拿 Key 与两套协议(地基).

所有接法都从同一块地基开始: 一个 Key、两套协议。先把地基打对, 后面每个 agent 都只是填空。先去控制台建 Key: 国内在 platform.kimi.com(人民币, 微信 / 支付宝充值), 国际在 platform.kimi.ai(美元)。**两套账户与 Key 完全独立, 混用直接 401**——官方 FAQ 的原话是

「不能混用」,这也是接入失败最常见的原因。³ 同一个 Key, 对应两套兼容协议。一是 **OpenAI 兼容**(Chat Completions): base_url 国内 `https://api.moonshot.cn/v1`、国际 `https://api.moonshot.ai/v1`, 鉴权 `Authorization: Bearer`, 官方惯用环境变量 `MOONSHOT_API_KEY`。⁴ 二是 **Anthropic 兼容**: 国内 `https://api.moonshot.cn/anthropic`、国际 `https://api.moonshot.ai/anthropic` —— 这是让 Claude Code 系工具指向 Kimi 的关键, 下一节第一个就用它。⁶

国内站 (¥)

```
# 控制台: https://platform.kimi.com/console/api-keys
export MOONSHOT_API_KEY="YOUR_KIMI_API_KEY"
# OpenAI 兼容
export OPENAI_BASE_URL="https://api.moonshot.cn/v1"
# Anthropic 兼容(给 Claude Code 系用)
export ANTHROPIC_BASE_URL="https://api.moonshot.cn/anthropic"
```

国际站 (\$)

```
# 控制台: https://platform.kimi.ai/console/api-keys
export MOONSHOT_API_KEY="YOUR_KIMI_API_KEY"
# OpenAI 兼容
export OPENAI_BASE_URL="https://api.moonshot.ai/v1"
# Anthropic 兼容(给 Claude Code 系用)
export ANTHROPIC_BASE_URL="https://api.moonshot.ai/anthropic"
```

两套账户速查: KEY 在哪建, 就用哪套端点(占位符 `YOUR_KIMI_API_KEY` 换成你的 KEY)

地基里还埋着一个限速坑: 速率按账户累计充值分档, 不充值的 tier0 只有 3 RPM、并发 1 —— 而 OpenAI SDK 默认会自动重试 2 次, 官方 FAQ 的原话是, tier0 用户「一次错误的请求就会消耗完所有的 RPM 额度」。充到 ¥50 的 tier1 才有 200 RPM 的可用体验; 注意代金券不计入累计充值。⁵

- 注意

接入报错先查这三样, 按命中率排: ① 国内 Key 配了国际端点(或反过来)→ 401; ② 拿会员订阅的 Key 去调开放平台(或反过来)→ 401; ③ tier0 限速 → 429。三样都不是配置写错了, 是账户对错了。

. . .

- III

把 Key 配进你已经在用的 agent.

按「官方文档写得有多明白」排序。每个给最小可用配置, 注意事项跟在配置后面。

Claude Code(Anthropic 兼容, 官方有指南).

这是官方文档写得最全的一条: 把 ANTHROPIC_BASE_URL 指到 Kimi 的 Anthropic 兼容端点、用 ANTHROPIC_AUTH_TOKEN 鉴权, Claude Code 就把 Kimi 当后端。⁶ 两个变量名别搞混—— ANTHROPIC_AUTH_TOKEN 走 Authorization: Bearer 头, ANTHROPIC_API_KEY 走 X-Api-Key 头; 两者同时存在会冲突, 之前配过 ANTHROPIC_API_KEY 的先删掉。⁷ 写进 ~/.claude/settings.json 长期生效:

```
{
  "env": {
    "ANTHROPIC_BASE_URL": "https://api.moonshot.cn/anthropic",
    "ANTHROPIC_AUTH_TOKEN": "YOUR_KIMI_API_KEY",
    "ANTHROPIC_MODEL": "kimi-k3[1m]",
    "ANTHROPIC_DEFAULT_OPUS_MODEL": "kimi-k3[1m]",
    "ANTHROPIC_DEFAULT_SONNET_MODEL": "kimi-k3[1m]",
    "ANTHROPIC_DEFAULT_HAIKU_MODEL": "kimi-k3[1m]",
    "ANTHROPIC_DEFAULT_FABLE_MODEL": "kimi-k3[1m]",
    "CLAUDE_CODE_SUBAGENT_MODEL": "kimi-k3[1m]",
    "ENABLE_TOOL_SEARCH": "false",
    "CLAUDE_CODE_AUTO_COMPACT_WINDOW": "1048576",
    "CLAUDE_CODE_EFFORT_LEVEL": "max"
  }
}
```

~/CLAUDE/SETTINGS.JSON -- 照官方指南的最小全套(国内站;国际站把 BASE_URL 换成
HTTPS://API.MOONSHOT.AI/ANTHROPIC)

四条注意事项,都来自官方指南:①各档模型变量要配全,缺了对应场景(后台摘要、子 agent)会静默失败;②/model 菜单是内置别名表、不会显示 Kimi 模型,配置是否生效以/status 为准;③ENABLE_TOOL_SEARCH 必须为 false, Kimi 端点暂不支持该特性;④换用 kimi-k2.7-code 时它强制思考,要在 Claude Code 里按 Tab 开 Thinking on,否则请求被 400 拒绝;另外 WebFetch 在该端点暂不可用。⁶

OpenCode(有内置 Moonshot, 最省事).

OpenCode 内置了 Moonshot AI(国际)与 Moonshot AI (China)(中国区)两个供应商:运行 /connect 选对应那个、粘贴 Key 即可,不用手写配置。¹⁰ 想自己管配置(比如团队共享),就在 opencode.json 里声明一个 OpenAI 兼容供应商:

```
{
  "$schema": "https://opencode.ai/config.json",
  "provider": {
    "kimi": {
      "npm": "@ai-sdk/openai-compatible",
      "name": "Kimi (Moonshot CN)",
      "options": {
        "baseUrl": "https://api.moonshot.cn/v1",
        "apiKey": "{env:MOONSHOT_API_KEY}"
      },
      "models": {
        "kimi-k3": { "name": "Kimi K3", "limit": { "context": 1048576, "output": 65536 } },
        "kimi-k2.7-code": { "name": "Kimi K2.7 Code", "limit": { "context": 262144, "output": 65536 } }
      }
    },
    "model": "kimi/kimi-k3"
  }
}
```

OPENCODE.JSON -- 自定义供应商(国内站;国际站换 BASEURL 与内置供应商 MOONSHOT AI)

Codex CLI(直连不行, 要借一层协议转换).

先把墙说清楚: 当前版本的 Codex, 自定义供应商的 wire_api 只支持 responses 一档(官方文档原文: responses is the only supported value), 而 Kimi 开放平台提供的是 Chat Completions —— 协议对不上, **直连不行**。⁸ 官方路径是借 CC-Switch 的本地路由做转换: 路由把 Codex 的 Responses 请求转成 Chat Completions 再发给 Kimi, Moonshot 官方的 Codex 接入指南就是照这条写的(配置在 CC-Switch 界面里完成, 不用手改 config.toml)。⁹ 完整步骤见下一节 —— 对 Codex 用户来说, CC-Switch 不是可选项, 是目前唯一官方写明的桥。

Pi(终端 agent, 配置文件驱动).

Pi 是 Mario Zechner 写的终端编码 agent(仓库现为 earendil-works/pi, 原名 pi-mono), 全部供应商配置收在一个 ~/.pi/agent/models.json 里。¹¹ 装好后把 Kimi 声明成一个 OpenAI 兼容供应商:

```
{
  "providers": {
    "moonshot": {
      "baseUrl": "https://api.moonshot.cn/v1",
      "api": "openai-completions",
      "apiKey": "MOONSHOT_API_KEY",
      "models": [
        { "id": "kimi-k3", "name": "Kimi K3", "reasoning": true, "contextWindow": 1048576, "maxTokens": 65536 },
        { "id": "kimi-k2.7-code", "name": "Kimi K2.7 Code", "reasoning": true, "contextWindow": 262144, "maxTokens": 65536 }
      ]
    }
  }
}
```

~/.PI/AGENT/MODELS.JSON -- APIKEY 支持环境变量名 / 字面值 / !命令三种写法; 建议像下面这样填环境变量名, 别把 KEY 本体写进文件

保存后在会话里 /model 选择即可, 改配置不用重启。¹¹ 走会员订阅那条线的读者: Pi 还内置了 Kimi For Coding 供应商, 设环境变量 KIMI_API_KEY 就能用 —— 注意那套 Key 与开放平台互不通用, 别混。^{2 11}

Cline / Roo Code / Aider(都有现成路径).

Cline 和 Roo Code 都在供应商下拉里内置了 Moonshot: 直选、粘贴 Key、选模型, 三个动作; 也可以选 OpenAI Compatible, 填 Base URL / API Key / Model ID 三个字段。^{12 13} Cline 有一条社区反馈值得知道: kimi-k2.6 走 OpenAI Compatible 会因 temperature 校验报 400, 优先用内置 Moonshot 供应商(社区 issue, 非官方结论)。¹² Aider 走通用的 OpenAI 兼容环境变量:

```
export OPENAI_API_BASE="https://api.moonshot.cn/v1"
export OPENAI_API_KEY="YOUR_KIMI_API_KEY"
aider --model openai/kimi-k3
```

AIDER 接 KIMI(也可写进 .AIDER.CONF.YML: OPENAI-API-BASE / OPENAI-API-KEY / MODEL)

Cursor(明说: 官方接不了).

Cursor 的自带 Key(BYOK)只支持 OpenAI / Anthropic / Google / Azure OpenAI / AWS Bedrock 五家, 文档里没有任意第三方 OpenAI 兼容端点的入口; 自定义 Key 还只覆盖标准聊天模型, Tab 补全照样走 Cursor 内置模型。¹⁵ 设置里那个 OpenAI base URL override 开关, 即便你的版本里还在, 也不受官方支持、随时可能移除, 别把 workflow 押在上面。要在 Cursor 里用 Kimi 模型, 目前诚实的答案是: 不能, 换上面任何一家 agent。

• • •

— IV

CC-Switch: 一个面板管所有切换.

当你只有一套配置, 手动改 env 就够了; 当你在 Kimi、Claude 官方、其他国产模型之间来回切, 手动改就变成了体力活加残留地狱。CC-Switch(farion1231/cc-switch, 官网 ccswitch.io)就是为这件事生的: 一个桌面面板, 管理 Claude Code、Codex、Gemini CLI、OpenCode 等 8 个工具的供应商配置, v3.17.0、约 12 万 star。¹⁶ 它不是命令行工具, 是 Tauri 桌面应用:

```
brew install --cask cc-switch
```

安装 CC-SWITCH(MACOS;WINDOWS / LINUX 在 CCSWITCH.IO 下安装包)

它的机制值得花三十秒搞懂: 你的每套供应商配置(profile)存在它自己的数据库(~/.cc-switch/cc-switch.db); 点「启用」时, 它把对应配置**原子写入**工具真正的配置文件——Claude Code 是 ~/.claude/settings.json 的 env 块, Codex 是 ~/.codex/auth.json 加 config.toml。Claude Code 支持热切换不用重启, Codex 要重开终端。手动改 live 配置文件也没关系, 会被它回填进数据库; 它还会检测 shell 里残留的 ANTHROPIC_API_KEY 之类的环境变量并警告——因为环境变量优先级高于配置文件, 那正是「改了不生效」的头号来源。¹⁶ 接 Kimi 几乎不用动手: Claude Code 侧内置了「Kimi」预设(端点 api.moonshot.cn/anthropic),

填 Key 即用 —— 预设默认模型目前还是 kimi-k2.7-code, 想用 K3 就把它改成 kimi-k3, 改完对照官方指南核一遍变量(ENABLE_TOOL_SEARCH=false、压缩窗口 1048576); 另有「Kimi For Coding」预设走会员订阅线(api.kimi.com/coding/). **6 16** Codex 侧就是第三节那堵墙的答案: 开启路由总开关与 Codex 路由(默认监听 127.0.0.1:15721), 添加 Kimi 供应商(Base URL https://api.moonshot.cn/v1、默认模型 kimi-k3、上游格式 Chat Completions), 启用后重启 Codex —— 路由运行时它替你完成 Responses 与 Chat Completions 的转换。 **9**

- 补充

两句实话。一, CC-Switch 是社区第三方工具, 不是 Kimi 官方维护 —— Moonshot 官方指南自己提醒你, 它的预设值可能与官方推荐值有差异, 启用后用 /status 核一遍。二, 你的 Key 和请求会经过它的本地路由, 公司环境先过一遍安全评估; 下载只认官网 ccs witch.io, 官方声明过任何收费的「CC Switch」都是假冒。

和手动改 env 怎么选? 一套配置走天下 → 手动就够, 第二节都给了。多套之间频繁切、或要给团队演示 → CC-Switch 省的是切换成本和「不残留」。两者不冲突 —— 它写入的正是你手动改的那两个文件。 **16**

• • •

- V

终端循环不是给所有人的。

它替你写, 替不了你审 —— 你看不懂它的 diff, 它的速度只会让你更快地把错误合进去。划三条边界。完全不写代码 → 跳过这章, 用 Agent 模式做表、做站就够。会写、但只改一两行 → 对话框里贴代码更快, 不必请出终端。值得请出终端 agent: 多文件、长链条、要反复跑测试的改动 —— 投入越重, 终端循环越划算。一个值得请出它的真实场景: 给一个跨十几个文件的功能加开关。与其一句「加个 feature flag」就放手, 不如先把范围、测试和安全边界交代清楚, 让它先报方案再动手:

任务：给现有的导出功能加一个 feature flag「EXPORT_V2」。

【先报方案】先列出你打算改哪些文件、为什么，等我确认再动手，别直接改。

【做法】flag 默认关闭；关闭时行为和现在完全一致。

【测试】改完跑一遍现有测试；为新分支补至少两条（开 / 关各一）。

【安全】不要改数据库 schema、不要删文件、不要碰 CI 配置；每步说清要执行什么命令。

任务：给现有的登录接口加一个限流中间件「RATE_LIMIT_LOGIN」，每 IP 每分钟最多 10 次。

【先报方案】先列出你打算改哪些文件、为什么，等我确认再动手，别直接改。

【做法】用配置开关控制，默认关闭；关闭时行为和现在完全一致。

【测试】改完跑一遍现有测试；为限流逻辑补至少两条（未超限放行 / 超限返回 429）。

【安全】不要改数据库 schema、不要删文件、不要碰 CI 配置；每步说清要执行什么命令。

— 注意

一条安全线：放它自动跑命令前，你得能看懂它每一步要做什么、并且能回滚。它跑得快是优点，也是风险——一个你没看懂就合进去的改动，出问题时它跑得有多快，你回起来就有多狼狈。另外，K3 发布第一周，所有跑分都只有厂商自评和一个盲测榜（第三章），别拿它们当选型的理由；选型的理由该是「够用 + 兼容 + 国内顺手」。

01

挑一个真实的小仓库改动

选一件你本来就要做的事：修个 bug、加个脚本、补几条测试。

02

用你刚配好的那条线跑一遍，全程盯着

Kimi Code、Claude Code 接 Kimi、Pi —— 哪条都行，看它的每一步，该拦就拦。

03

记下改对几成、回滚几次

对比它的产出与你自己动手的时间，记录一次过的比例和你回滚的次数，给它定一条「值得请出它」的线。

参考.

- 01 Kimi Code CLI 官方文档 —— 安装: `curl -fsSL https://code.kimi.com/kimi-code/install.sh` | `bash`(macOS / Linux)或 `npm install -g @moonshot-ai/kimi-code`(需 Node.js $\geq 22.19.0$); /login 支持「Kimi Code(OAuth)」会员授权与「Kimi Platform API 密钥」两种方式; 可用模型 k3 / kimi-for-coding / kimi-for-coding-highspeed, k3 需 Moderato 档及以上会员、1M 上下文需 Allegretto 档及以上; 会员额度约 300-1200 次请求 / 5 小时窗口、并发上限 30。截至 2026-07-17。 [Kimi Code CLI 官方文档](#)
- 02 Kimi 帮助中心 · Kimi Code CLI 常见问题 —— 会员专用端点: OpenAI 兼容 <https://api.kimi.com/coding/v1>, Anthropic 兼容 <https://api.kimi.com/coding/>, 计费走 Kimi 会员订阅额度; 官方逐字声明「api.kimi.com 和 api.moonshot.cn 是两个完全独立的账号体系, API Key 互不通用」。截至 2026-07-17。 [Kimi 帮助中心 · Kimi Code](#)
- 03 Kimi 开放平台 · 快速开始与常见问题 —— 国内控制台 platform.kimi.com(人民币、微信 / 支付宝充值), 国际控制台 platform.kimi.ai(美元); 官方 FAQ 逐字:「两个平台的账户和 key 完全独立, 不能混用」, 混用报 401 `invalid_authentication_error`; 「中国境内建议使用 platform.kimi.com 平台, 境外建议使用 platform.kimi.ai 平台」。截至 2026-07-17。 [Kimi 开放平台 · 快速开始](#)
- 04 Kimi 开放平台 · API 总览 —— 兼容 OpenAI Chat Completions, `base_url` 国内 <https://api.moonshot.cn/v1>、国际 <https://api.moonshot.ai/v1>; 鉴权 `Authorization: Bearer`, 环境变量 `MOONSHOT_API_KEY`; `max_tokens` 已弃用, 改用 `max_completion_tokens`。截至 2026-07-17。 [Kimi 开放平台 · API 总览](#)
- 05 Kimi 开放平台 · 充值与限速 —— 按账户累计充值分档: tier0(¥0)并发 1、3 RPM、50 万 TPM; tier1(¥50)并发 50、200 RPM; 最高 tier5(¥20,000)1000 并发、10,000 RPM。官方 FAQ 提醒: OpenAI SDK 默认重试 2 次会放大 RPM 消耗, tier0 一次错误请求即可耗尽全部 RPM; 代金券不计入累计充值。国际站档位 \$1-\$3,000。截至 2026-07-17。 [Kimi 开放平台 · 充值与限速](#)
- 06 Kimi 开放平台 · 在 Claude Code 中使用 Kimi —— Anthropic 兼容端点 <https://api.moonshot.cn/anthropic>(国际站 <https://api.moonshot.ai/anthropic>); 官方环境变量: `ANTHROPIC_BASE_URL` / `ANTHROPIC_AUTH_TOKEN` /

ANTHROPIC_MODEL=kimi-k3[1m] 及 OPUS / SONNET / HAIKU / FABLE 各档模型变量、CLAUDE_CODE_SUBAGENT_MODEL、ENABLE_TOOL_SEARCH=false、CLAUDE_CODE_AUTO_COMPACT_WINDOW=1048576、CLAUDE_CODE_EFFORT_LEVEL=max; 须删除旧 ANTHROPIC_API_KEY 防冲突;/model 菜单不显示 Kimi 模型,以 /status 为准;kimi-k2.7-code 需按 Tab 开启 Thinking,否则 400 invalid thinking;WebFetch 暂不可用。截至 2026-07-17。 [Kimi 开放平台 · Claude Code 指南](#)

- 07 Anthropic 官方文档 · Claude Code 环境变量 —— ANTHROPIC_AUTH_TOKEN 走 Authorization: Bearer 头,ANTHROPIC_API_KEY 走 X-API-Key 头,二者同时存在会冲突;ANTHROPIC_SMALL_FAST_MODEL 已废弃,由 ANTHROPIC_DEFAULT_HAIKU_MODEL 等档位变量取代。截至 2026-07-17。 [Claude Code · 环境变量](#)
- 08 OpenAI 官方文档 · Codex 配置参考 —— 自定义 model_providers 支持 name / base_url / env_key / wire_api 等键;wire_api 原文「responses is the only supported value」,即只支持 Responses 一档。Kimi 开放平台提供的是 Chat Completions,协议接不上,故 Codex 无法直连,需协议转换层。截至 2026-07-17(该页自 developers.openai.com 迁至 learn.chatgpt.com)。 [Codex · 配置参考](#)
- 09 Kimi 开放平台 · 在 Codex CLI 中使用 Kimi K3 —— 官方路径:经 CC Switch 本地路由把 Codex 的 Responses API 请求转换为 Kimi 的 Chat Completions;步骤为开启路由总开关与 Codex 路由、添加 Kimi 供应商(Base URL https://api.moonshot.cn/v1、默认模型 kimi-k3、上游格式 Chat Completions、上下文窗口 1048576)、启用并重启 Codex。官方提示 CC Switch 为第三方工具,API Key 与请求响应由其本地路由处理。截至 2026-07-17。 [Kimi 开放平台 · Codex 指南](#)
- 10 OpenCode 官方文档与 Kimi 官方 OpenCode 指南 —— 内置 Moonshot AI(国际)与 Moonshot AI (China)(中国区)两个供应商,/connect 粘贴 Key 即可;自定义供应商走 opencode.json,npm 包 @ai-sdk/openai-compatible,options.baseURL / options.apiKey(支持 {env:VAR} 引用)/ models 逐字键名。截至 2026-07-17。 [Open Code · Providers](#)
- 11 Pi 官方文档(earendil-works/pi,原 badlogic/pi-mono)—— 终端编码 agent; 安装 npm install -g @earendil-works/pi-coding-agent; 自定义供应商写在 ~/.pi/agent/models.json,支持 api: openai-completions / anthropic-messages

等,apiKey 可填环境变量名;另内置 Kimi For Coding 供应商(环境变量 KIMI_API_KEY)。截至 2026-07-17。 [Pi · GitHub](#)

- 12 Cline 官方文档 —— 内置 Moonshot 供应商(下拉直选、粘贴 Key);或选 OpenAI Compatible,逐字段 Base URL / API Key / Model ID。社区 issue(cline#10544)报告:kimi-k2.6 走 OpenAI Compatible 会因 temperature 校验报 400,建议优先用内置 Moonshot 供应商(社区反馈,非官方结论)。截至 2026-07-17。 [Cline · Moonshot](#)
- 13 Roo Code 官方文档 —— 内置 Moonshot 供应商(下拉直选);或选 OpenAI Compatible(Base URL / API Key / Model);Roo Code 仅使用原生 tool calling,模型须支持 OpenAI 风格工具调用。截至 2026-07-17。 [Roo Code · Moonshot](#)
- 14 Aider 官方文档 · OpenAI 兼容端点 —— export OPENAI_API_BASE 与 OPENAI_API_KEY 后以 aider --model openai/模型名 启动;.aider.conf.yml 对应键 openai-api-base / openai-api-key / model。无 Moonshot 专页,走通用 OpenAI 兼容路径。截至 2026-07-17。 [Aider · OpenAI Compatible](#)
- 15 Cursor 官方文档 · API Keys —— 自带 Key(BYOK)仅支持 OpenAI / Anthropic / Google / Azure OpenAI / AWS Bedrock 五家,无任意第三方 OpenAI 兼容端点;自定义 Key 只覆盖标准聊天模型,Tab 补全等仍走 Cursor 内置模型。截至 2026-07-17,Kimi 无法经官方支持的方式接入 Cursor。 [Cursor · API Keys](#)
- 16 CC-Switch(farion1231/cc-switch,官网 ccs witch.io)—— v3.17.0(2026-07-13),约 12 万 star(GitHub API 118,315),Tauri 桌面应用(macOS 可 brew install --cask cc-switch);管理 Claude Code / Codex / Gemini CLI / OpenCode 等 8 个工具的供应商配置;配置存于 ~/.cc-switch/cc-switch.db(SQLite),切换时原子写入 ~/.claude/settings.json 的 env 块与 ~/.codex/auth.json + config.toml;Claude Code 支持热切换,Codex 需重开终端;自动检测 shell 环境变量冲突;内置 Kimi(api.moonshot.cn/anthropic,预设模型 kimi-k2.7-code)与 Kimi For Coding(api.kimi.com/coding/)预设;Codex 侧本地路由默认 127.0.0.1:15721。截至 2026-07-17。 [CC-Switch · GitHub](#)

它替你写,
替不了你审。

选档 & 取舍 · 买哪档, 什么时候回 frontier.

选档不是看价格, 是看你缺 Agent 次数、Agent 集群, 还是 Kimi Code 额度。这章把五档会员摆开, 说清订阅不含 API, 以及什么活该回 Claude / GPT、什么活交给 DeepSeek。

一本书读到这里, 该落到两个决定: 这套栈你买哪一档, 以及哪些活根本不该交给它。选档不是看价格, 是看你缺的是哪样分项额度; 取舍不是看谁口号大, 是看你最常干的活谁做得最顺。

- I

五档差的是分项额度, 不是模型.

你多付的钱, 买的是各项用量, 不是一个更聪明的模型 —— 五档跑的是同一套模型。截至 2026-07, Kimi 会员五档全用音乐术语命名: Adagio(免费)、Andante ¥49、Moderato ¥99、Allegretto ¥199、Allegro ¥699。¹ 它们之间差的不是模型, 而是**分项额度**: Agent 用量(约 6 → 360 个)、Kimi Code 倍率(1x → 60x)、Agent 集群(Moderato 25 次 / Allegretto 50 / Allegro 120 —— 门槛比本书上一版低了一档)、专业数据库调用(200 → 12,000 次), 以及 Kimi Claw 从 Allegretto(¥199)才解锁。还要注意额度结构变了: Agent、深度研究、PPT、Kimi Claw 这些**共享一个 token 额度池**(次数只是折算估值), 只有 Kimi Code 是单独的池 —— 你 Swarm 跑得猛, 是真的会把 Agent 用量也吃紧的。¹ 所以选档先问: 你最吃的是哪一样? 重度写码的人盯 Kimi Code 倍率, 常交办成品的人盯 Agent 次数和集群, 多数轻度用户免费的 Adagio 或 ¥49 的 Andante 就够。别照着价目表挑「功能最全」的那档 —— 挑「补上你那块短缺」的那档。

• • •

订阅不含 API —— 自动化是另一条账。

想把 Kimi 接进脚本和系统, 订阅那点额度不算数; API 是另一套账户、另一条账单。这条线最容易踩。消费端会员和开放平台(platform.kimi.com)的 API 是两回事: 两套账户、两条计费, 会员买得再高也不附带 API 额度。² API 按 token 计价(旗舰 kimi-k3 每百万 ¥20 输入 / ¥100 输出, 缓存命中 ¥2), base_url 是 <https://api.moonshot.cn/v1>, 兼容 OpenAI(第八章那条)。另外第八章还讲过第三条账: 会员自带的 coding 端点(api.kimi.com/coding)—— 三套账户各算各的, 谁的 Key 也别串。判断很简单: 偶尔手动用 → 会员; 要稳定、批量、嵌进产品自动跑 → 走 API 单独算账。会员买的是你自己用, API 买的是让它替系统跑。顺带辟一个传言: 市面上流传的「Kimi Code 独立订阅 \$19 / \$39」是把国际版会员价张冠李戴。截至 2026-07-17, Kimi Code 仍随会员提供、不是独立商品; 官方只预告了「会员权益将与 Kimi Code 权益拆分、可按需分别购买」, 新套餐的档名和价格还没公布 —— 别按传言里的数字做预算。⁴

. . .

取舍: 它的位置, 和什么时候别用它。

Kimi 不是「最便宜」, 也不是「样样第一」, 更不是「国产版某家」。把它当谁的平替, 你只会拿它的短处比别人的长处。认清它的独一份: 不是单项最强, 是**一整套整合好的 agent 栈 + 国内可达 + 中文长材料顺手 + 开源权重打底**叠在一起(现货是 K2 系, K3 权重官方承诺 7 月 27 日前放出)。³ 选它, 是因为这一整套对上了你的活 —— 要的就是「聊天 + 交办 + 研究 + 写码」一站式、还在国内直连。单拎一项去比, 它可能不占上风。给一把按场景的尺子(只讲适配, 不踩同行): 要一站式的 agent 栈、中文长材料、国内顺手的编码 → Kimi; 最难的推理、最吃打磨的 agentic coding、一锤定音的输出 → 回 Claude / GPT 这些前沿; 极致性价比的推理、或想拿开源权重自建省钱 → DeepSeek 更划算(它专打便宜这条, 而 Kimi 已经涨价离场); 多模态创作、绑字节生态 → 豆包更全; 阿里生态、企业接入合规 → 通义更稳。选主力的标准不是谁 benchmark 高, 是「你最常干的

那类活,谁做得最顺」。多数人需要一两个主力加一个备胎,不是五个都开着。这一节最该被你截图存下来。

☐ 列出你上个月真花掉额度的 5 类活。

☐ 对照五档的 Agent 次数 / Kimi Code 倍率 / 集群,看你其实卡在哪一样。

☐ 挑出其中该走 API、或换前沿 / DeepSeek / 豆包 / 通义 更顺的活。

☐ 定下你的主力档位与「什么时候升 / 降档」的触发条件。

— 要点

收束就一句: Kimi 这一年从长文本工具长成了一套栈。它不靠便宜取胜,靠的是这一整套对上你的活。选对那一面、买对那一档,比追更高的档位重要——它对不上的活,也大方交给更合适的那一个。

参考.

- 01 Kimi 帮助中心 · 会员套餐价格 —— 截至 2026-07 五档: Adagio(免费)、Andante ¥49、Moderato ¥99、Allegretto ¥199、Allegro ¥699。分项额度: Agent 用量约 6 / 30 / 60 / 150 / 360 个(共享 token 额度池按常见任务折算的估值), Kimi Code 倍率 1x / 4x / 20x / 60x(单独额度池), Agent 集群 Moderato 25 次 / Allegretto 50 次 / Allegro 120 次, 专业数据库 200 / 1,000 / 2,000 / 5,000 / 12,000 次, Kimi Claw 自 Allegretto 起; 连续包年最高省 ¥1,680。国际版为另一套档位(\$19-\$199)。 [Kimi 帮助中心 · 会员价格](#)
- 02 Kimi 开放平台(platform.kimi.com) · 定价 —— 消费端会员与 API 是两套账户、两条计费; 订阅不含 API 额度。API 按 token 计价(旗舰 kimi-k3 每百万 ¥20.00 输入 / ¥2.00 缓存命中 / ¥100.00 输出, 1M 上下文), base_url <https://api.moonshot.cn/v1>。截至 2026-07。 [Kimi 开放平台 · 定价](#)
- 03 Hugging Face · moonshotai(Kimi-K2.6 / Kimi-K2.7-Code 模型卡) —— K2 系权重 Modified MIT 开源, 可自托管; K3 权重官方承诺 2026 年 7 月 27 日前放出。Kimi 的独一份不是「最便宜」也不是「样样第一」, 而是「整合好的 agent 栈 + 国内可达 + 中文长材料 + 开源权重」叠在一起。截至 2026-07。 [Hugging Face · Kimi-K2.7-Code](#)
- 04 Kimi Code 官方文档 · 会员权益 —— Kimi Code 随 Kimi 会员订阅一同提供、并非独立商品; k3 需 Moderato 档及以上、1M 上下文需 Allegretto 档及以上, 高速版需 Allegretto; 每 5 小时约 300-1,200 次请求、并发 30。官方预告「会员权益将与 Kimi Code 权益拆分、可按需分别购买」, 截至 2026-07-17 未上线、新套餐档名与价格未公布。 [Kimi Code · 会员权益](#)

便宜是 DeepSeek 的活,
Kimi 卖的是整套栈.

实战速查 · 哪类活该用哪一面。

前面把每一面单独讲透了,这一页把判断压成一张表:常见的活该走哪一面、怎么用、注意什么。对上哪一行,就用那一面。

前面把每一面单独讲透了。这一页把那些判断压成一张表——你不必每次重读一章,对着手头的活,在表里找到那一行就行。用法只有一句: **对上哪一行,就走那一面**;对不上的,别硬留在 Kimi 这,回上一章「选型取舍」看该交给谁。

- I

场景 → 用哪一面 → 怎么用。

一张表覆盖八成日常。先认你这次的活属于哪一类,再照「怎么用」那一列去交代它。

这类活	走哪一面	怎么用(一句最佳实践)
查事实、要定义	默认聊天 · Instant	直接问, 别开思考、别上 Agent
多步推理、算账、写逻辑	思考 · Thinking	把背景和判断标准一次给够
读懂几万字长材料	长上下文(贴文件)	让它先定位原文再答, 抽查中段
中文成稿 / 改稿	默认聊天	给素材和立场让它成稿; 改稿优于从零写
做 PPT / 网页 / 表格成品	Agent	验收标准前置, 数字只用你给的
大规模盘点 / 几十份材料综述	Agent Swarm	框成可并行、每点标来源 (Moderato+ 25 次起)
带出处的调研报告	Deep Research	每条结论标来源 + 日期, 抽查 2-3 条
多文件、长链条代码改动	Kimi Code · K3	先报方案再动手, 限范围 + 测试 + 可回滚
接进脚本 / 产品自动跑	开放平台 API	改 base_url + model; 走 API 单独计费
7×24 常驻、定时任务	Kimi Claw	先拿无关紧要的定时任务试手

表里的「面」就是前面各章: 聊天与思考(第三、四章)、Agent(第五章)、Agent Swarm(第六章)、Deep Research(第七章)、Kimi Code 与 API(第八章)。¹ 要看具体怎么做, 翻回那一章。

. . .

- II

三条贯穿所有场景的最佳实践。

不管走哪一面, 这三条都成立 —— 记住它们, 比记住任何一个入口都管用。

先把活说清, 再挑模型 / 模式	缺背景、没判断标准的提问, 开思考、换 Agent 都救不回来。提问质量是前 90%, 挡位是最后 10%。
越自动, 越要前置验收和红线	Agent / Swarm / Kimi Code / Claw 这些「替你做」的, 出发前就把「做成什么样算成」和「不许碰什么」写清楚 —— 否则它把错事也做得很完整。
要事实, 就要出处	Deep Research 和 Swarm 的产出都让它标来源 + 日期, 你抽查两三条再用。挂着来源 ≠ 来源支持那句话。

还有一条账上的铁律: 以上都是消费端会员的事; 一旦要稳定、批量、嵌进系统自动跑, 走开放平台 API(api.moonshot.cn/v1), 那是另一套账户、另一条计费, 订阅额度不算数。²

. . .

- III

反面速查: 这几样别这么用。

省力的另一半, 是不把劲使错地方。最常见的几个错配:

- 拿 Agent / Swarm 去问一个事实 —— 等十分钟换一句本该秒回的话。
- 把开放式构思(「帮我想想方向」)交给 Agent —— 它会把没定准的方向一口气做到底。
- 冲着「便宜」选 Kimi —— 它已涨价离场, 省钱该看 DeepSeek。
- 把会员额度当 API 用 —— 两套账, 订阅不含 API。 **2**
- 拿 Kimi Claw 押要紧事 —— 要 Allegretto 以上才有, 常驻 agent 先拿无关紧要的定时任务试手, 边界想清再交。 **3**

把这页截下来贴在手边。它不替你做判断, 替你少走那十次最常见的弯路 —— 剩下的, 交给前面九章那把尺。

- 引用与参考

参考.

- 01 Kimi 官网首页(kimi.com)—— 截至 2026-07 首屏标注「Kimi AI with K3 | Built for Agentic Coding & Knowledge Work」, 产品入口涵盖 Chat / Slides / Websites / Docs / Sheets / Deep Research / Agent / Agent Swarm / Kimi Code / Kimi Claw(/bot) / Kimi Work(Beta), 即本页表格里那几「面」的来源。 [Kimi 官网](#)
- 02 Kimi 开放平台(platform.kimi.com) · API 总览 —— base_url
https://api.moonshot.cn/v1, 兼容 OpenAI; 消费端会员与 API 是两套账户、两条计费, 订阅不含 API 额度。截至 2026-07。 [Kimi 开放平台 · API 总览](#)
- 03 Kimi 帮助中心 · 会员价格 —— Agent 集群(Swarm)自 Moderato(¥99/月)起(25 / 50 / 120 次, 仍带 Beta 说明); Kimi Claw 自 Allegretto(¥199/月)起, 本体页面已无 Beta 标记(新形态 Claw 群聊标 Preview); 会员功能共享 token 额度池(Kimi Code 为单独池)。截至 2026-07。 [Kimi 帮助中心 · 会员价格](#)

对上哪一行, 就走哪一面;
对不上, 别硬留在它这。



从长文本到一套 agent 栈。

别再把它当长文本工具 —— K3 旗舰思考永远开, 一套整合的 agent 栈加双兼容 API, 国内可达、中文长材料顺手; 该用的那几面替你挑出来, 什么时候回 frontier 也说清。



扫码在线阅读

作者 · AKLMAN

Aklman · 文库 —— 写给已经订阅 AI 工具、却只用到一小部分的人; 每本短书讲清一份订阅里该学、该跳过、该换入口的部分。写代码, 也写字。

这是静态 PDF 快照; 网页版阅读体验更佳, 内容持续更新、数据更及时准确。

library.aklman.com · x.com/aklman2018 · github.com/aklmans

章节	字数	更新	版本
10	1.4 万字	2026.06	PDF 1.0

© 2026 · AKLMAN.LIBRARY

本书仅供学习交流, 内容基于公开资料整理, 不构成商业建议。