Vibe42

文档快照

Vibe42 文档

在电脑上安装 Vibe42，配对手机或浏览器，随时随地接着跑你的 coding agent。

10 个章节

58 篇文档

来源: vibe42.ai/zh-HANS/docs

2026-11-03

目录

01 **开始使用** 4

Vibe42 是什么 4

安装到电脑 6

配对设备并开始第一个会话 8

02 **工作原理** 11

电脑、设备与 Team 11

连接是怎么建立的 12

安全模型 14

会话、窗口与 最近 16

03 **终端** 20

终端引擎 20

在手机和平板上输入 22

窗口、分屏与终端列表 24

键盘快捷键 25

主题与字体 28

04 **Coding Agent** 30

支持的 Coding Agent 30
不注入、不锁定 31
启动 Agent 33
Agent 状态 35
通知 37
OSC 7501 状态协议 40
让 Agent 使用 Vibe42 42
在 Vibe42 中使用 Claude Code 44
在 Vibe42 中使用 OpenAI Codex CLI 45
在 Vibe42 中使用 OMP 46
在 Vibe42 中使用 Pi Coding Agent 48
在 Vibe42 中使用 OpenCode 49
在 Vibe42 中使用 Grok Build 50
在 Vibe42 中使用 Devin CLI 52
在 Vibe42 中使用 Antigravity CLI 53
在 Vibe42 中使用 Cursor Agent 54

05 **Workspace 与文件** 56

Workspace 56
文件 57
预览产物与端口转发 61
终端隔离 62
压缩与去重 65
备忘、休息提醒与清理 67

06 **远程桌面** 70

远程桌面：开始使用 70
操控、画质与剪贴板 72
解锁屏幕与打开 App 74

07 **账号、Team 与套餐** 76

账号与登录 76
Team、角色与共享电脑 77
套餐与名额 79
Credits 与用量 81
试用、兑换码与退款 82

08 **平台说明** 85

macOS App 85
Linux 桌面 App 87
Linux 服务器与无头机器 89
iPhone 与 iPad 91
Android 93
Web 应用与 PWA 95
Windows 96
更新与卸载 98

09 **v42 CLI** 101

v42 CLI 概览 101
命令参考 103
自动化示例 107

10 **故障排查** 111

连接问题 111
收不到通知 114
终端与输入问题 116
收集日志 118
常见问题 120

01 开始使用

Vibe42 是什么

用三分钟了解 Vibe42 如何让你不需要任何网络配置就能远程连接 Mac 或 Linux 上的终端、Agent、文件与桌面。

vibe42.ai/zh-HANS/docs/getting-started/what-is-vibe42

如果你正在评估 Vibe42 是否适合日常工作流，想用三分钟了解它能做什么、不能做什么以及从哪里开始，请先看这一页。

Vibe42 是不需要任何网络配置的远程终端与 AI 工作台。在你的 Mac 或 Linux 电脑上安装 Vibe42，再配对手机、平板、浏览器或另一台电脑，就能在任何地方继续使用那台电脑上的终端、Agent、文件和屏幕。

常见使用场景：

- **出门后继续跟进 Agent**：在工位或家里的电脑上启动 Claude Code，离开座位后在手机上收到推送通知，点开即可查看输出、批准权限或继续交代任务。
- **在外审阅代码**：在咖啡馆用浏览器打开 `https://app.vibe42.ai`，直接连接家里的机器查看 git diff、跑测试、预览或下载文件。
- **远程解锁 Mac 跑图形应用**：遇到 Mac 锁屏时远程输入系统密码解锁桌面，再通过远程桌面运行 Xcode 或查看模拟器。

它是怎么工作的

使用 Vibe42 不需要公网 IP、端口映射、防火墙入站规则，也不用手动分发密钥。

```
你的设备（原生 App / 浏览器 / CLI）
        │
        ├─ P2P 点对点直连（原生 App、后台服务与 CLI 之间优先自动建立）
        │
        └─ 经最近 AP 加密中转（随时兜底可用；浏览器永远走中转）
                │
                ▼
你的电脑（运行 Vibe42 后台服务的 macOS / Linux）
```

你的设备与电脑上的 Vibe42 后台服务都会主动向外连接距离最近的 AP（接入点）。每条连接都在设备与电脑之间直接完成端到端加密握手，AP 只负责转发密文，除基础连接元数据外看不到任何内容。

当你用原生 App 或 CLI 连接时，Vibe42 会在后台同时尝试建立点对点直连，一旦打通就会自动切走流量，不再经过 AP；浏览器（Web 应用）则始终通过 AP 中转。具体机制见连接是怎么建立的与安全模型。

三个核心概念

Vibe42 的日常使用围绕三个对象展开：

- **电脑（Computer）**：运行后台服务的 Mac、Linux 主机、虚拟机或容器，你的持久终端、Agent 和本地文件都在上面运行。
- **设备（Device）**：用来连接并操控电脑的一端，包括手机、平板、桌面 App 和网页浏览器。
- **Team**：管理套餐、名额与每月 Credits 的结算单元。每个账号注册后都会自动拥有一个个人 Team。

Vibe42 没有免费套餐。Team 的 Owner 可以开通 3 天免费试用（需绑卡）或直接订阅套餐；受邀加入他人 Team 的成员共享该 Team 的名额，不需要自己购买套餐。名额与角色的详细规则见电脑、设备与 Team。

它不是什么

- **不是云 IDE 或云主机**：你的代码库、编译工具链与本地环境始终留在自己的硬件上运行。
- **不是传统 SSH 客户端**：不需要配置固定 IP、动态域名（DDNS）、VPN 子网或维护 `authorized_keys`，所有设备通过 10 分钟有效的配对码授权。
- **不托管代码或 API key**：我们不保存你的终端输出、按键、源码文件或模型 API key。
- **不在云端托管运行 Agent**：Agent 以你自己的系统用户身份在本地电脑上运行；Vibe42 不会向 Agent 注入 hook、skill 或插件，详见零注入原则。
- **不支持远程桌面控制 Windows 屏幕**：Windows 可在 WSL 里安装 Server 版使用终端、Agent 和文件，但不能用远程桌面查看或操作 Windows 的屏幕（Windows 原生版本近期上线）。

平台支持矩阵

平台	作为电脑（被控端）	作为设备（操控端）
macOS	macOS 14+ 桌面 App（Apple Silicon 与 Intel 独立安装包，内置后台服务）；macOS 13 用一行命令安装 Server 版	macOS 14+ 桌面 App，或 Web 应用
Linux 桌面	x86_64 / arm64 AppImage（需 glibc 2.39+，如 Ubuntu 24.04 / Debian 13 / Fedora 40 及更新版本，内置后台服务）	x86_64 / arm64 AppImage，或 Web 应用
Linux 服务器 / 容器	一行命令安装 Server 版（支持 `x86_64` 与 `arm64`，静态链接程序）	CLI 或 Web 应用
iPhone / iPad	不支持	iOS / iPadOS 17+ 原生 App（App Store），或 Web 应用
Android	不支持	Android 11+ 原生 App（Google Play），或 Web 应用
Windows	Server 版装在 WSL 里：终端、Agent、文件；无远程桌面；原生版本近期上线	现代浏览器打开 `https://app.vibe42.ai`（可安装为 PWA，即添加到桌面或任务栏的网页应用）

下一步

- 安装到电脑：按你的机器类型安装 macOS App、Linux AppImage 或一行命令 Server 版。
- 配对设备并开始第一个会话：连接手机或浏览器，创建第一个终端并验证推送通知。
- 如果你手头正开着 Agent，也可以把 `https://vibe42.ai/install.md` 交给它自动完成安装，详见让你的 Agent 安装 Vibe42。

安装到电脑

根据电脑类型选择安装 macOS 桌面 App、Linux AppImage 或一行命令 Server 版，并完成配对审批。

vibe42.ai/zh-HANS/docs/getting-started/install

当你要在 Mac 或 Linux 电脑上安装 Vibe42 供远程连接时，按本页步骤选择对应的安装方式并完成配对审批。

先选安装路径

机器类型	安装方式
你的 Mac（macOS 14+ 桌面环境）	macOS 桌面 App（从 `https://vibe42.ai/#downloads` 下载 `.dmg`，内置 Vibe42 后台服务）
Linux 桌面（x86_64 或 arm64，glibc 2.39+）	Linux AppImage（从 `https://vibe42.ai/#downloads` 下载 `.AppImage`，内置后台服务）
Linux 服务器、容器、旧发行版、Windows（WSL）、macOS 13 或无桌面会话的 Mac	一行命令安装 Server 版（`get.vibe42.app/install.sh`，只有后台服务、没有桌面 App 的版本）
只用来操控其他电脑的机器	无需安装后台服务——直接在浏览器打开 `https://app.vibe42.ai`；或在装好桌面 App 后在启用询问页点 **暂不启用**

> **重要**
>
> 装了桌面 App（macOS App 或 Linux AppImage）的机器已内置后台服务，不要再跑一行命令安装（安装脚本会直接拒绝）。请始终以日常拥有终端的普通用户身份运行安装命令，除非你本来就以 `root` 身份工作，否则不要用 `root` 或 `sudo`。即使 Team 套餐尚未激活，也可以先完成安装与配对，套餐生效后电脑会自动连上。

macOS 桌面 App

要求 **macOS 14 及以上**，Apple Silicon 与 Intel 机型分别提供独立的 `.dmg` 安装包。如果机器上已经装有 `Vibe42.app` 或旧名称的 `Vibe X.app`，不要重新下载 `.dmg` 覆盖，直接打开现有 App 即可自动更新。

1. 前往 `https://vibe42.ai/#downloads` 下载对应芯片的 `.dmg`，打开后将 **Vibe42** 拖入 `Applications`。
2. 打开 Vibe42 并点 **登录**，系统默认浏览器会打开登录页；核对设备信息后点 **批准**。
3. 批准后，应用会根据账号状态依次显示最多五个引导页面：
   - **选择团队**：当你属于多个 Team 或有待处理的 Team 邀请时出现。
   - **订阅 Vibe42，继续你的工作**（非 Team Owner 看到的是 **当前团队尚未开通**）：当前 Team 尚未开通套餐时提示。
   - **要把这台电脑作为可远程控制的电脑吗？**：点 **启用这台电脑** 启动后台服务并允许远程连接这台 Mac，或点 **暂不启用** 仅把本机当作操控端。
   - **允许远程控制这台 Mac**：开启 **屏幕录制** 与 **辅助功能**。可以先跳过，之后随时在 **设置 → 远程桌面** 中授权。

- **后台运行权限**：在 **System Settings → General → Login Items & Extensions** 中保持开启，确保后台服务持续在线。

更多平台细节见 macOS。

Linux 桌面 AppImage

支持 **x86_64** 与 **arm64** 架构，要求 **glibc 2.39+**（即 Ubuntu 24.04、Debian 13、Fedora 40 及更新发行版），并需要 FUSE（Linux 挂载文件系统的组件）与 `systemd` 用户会话（user session）。

以下可选软件包可完善桌面集成：

- **系统托盘**：StatusNotifierItem 支持（GNOME 桌面需装 AppIndicator 扩展）。
- **文件沙盒预览**：安装 `bubblewrap` 与 `xdg-dbus-proxy` 以预览代码、Markdown、HTML 和 PDF。
- **X11 输入法**：安装 `fcitx5` 或 `ibus` 对应的 GTK4 模块。
- **视频解码**：安装 VA-API 或 NVIDIA 驱动用于远程桌面硬件解码（未检测到硬解时会按需自动下载 Cisco OpenH264）。

1. 从 `https://vibe42.ai/#downloads` 下载 `.AppImage` 与同名 `.sha256` 校验文件，校验通过后加上执行权限并启动：

```
sha256sum -c Vibe42-*.AppImage.sha256
chmod +x Vibe42-*.AppImage
./Vibe42-*.AppImage
```

若系统没有 FUSE，可先安装 FUSE 或加 `--appimage-extract-and-run` 参数启动。首次运行会自动注册 `.desktop` 快捷方式、图标及 `vibex://` 协议处理器；之后若移动了 AppImage 文件，重新启动一次即可自动更新路径。

2. 点 **登录**，并在浏览器中点 **批准**。如果浏览器没有自动跳回 App，把浏览器回调页显示的 `vibex://…` 链接复制并粘贴回 App 登录页即可。

后台服务安装在 `~/.v42/`，通过 systemd 用户服务 `v42-companion.service` 运行。若当前环境没有 `systemd` 用户会话，请改用一行命令安装并设置 `VIBEX_SERVICE_MODE=supervisor`。更多说明见 Linux 桌面。

一行命令安装（Server 版）

Server 版只有后台服务、没有桌面 App，支持 `x86_64` 与 `arm64` 的 Linux 服务器、无显示器主机、容器（Linux 下为静态链接程序）以及 macOS 13。在 Windows 上，于 WSL 的发行版终端里运行同一条命令。前置条件：`curl`、`tar` 和 `tmux` 3.0+。

1. 以日常使用终端的普通用户身份执行：

```
curl -fsSL https://get.vibe42.app/install.sh | sh
```

2. 脚本安装后台服务（`v42-companion`）后，会在终端打印 8 位配对码与审批链接：

```
Pairing code: ABCD-EFGH
https://vibe42.ai/console/pair?code=ABCD-EFGH
```

3. 在 **10 分钟**内，用浏览器打开 `https://vibe42.ai/console/pair?code=ABCD-EFGH` 批准，或在任一已登录的 Vibe42 App 中前往 **电脑与设备 → 连接新的电脑/服务器** 输入该配对码。

安装脚本会把文件放入 `~/.v42`（旧版目录为 `~/.vibex`），并把 `~/.v42/bin` 写入 `PATH`。有 systemd 的主机使用用户单元 `v42-companion.service` 加看门狗 timer 保活；无 systemd 的环境（如多数容器）自动切换为 supervisor 模式，也可传环境变量 `VIBEX_SERVICE_MODE=supervisor` 强制指定。重新运行同一条安装命令即可原地升级并保留配对。

后台服务常用命令

命令	用途
`~/.v42/bin/v42-companion status [--json]`	查看运行状态、配对信息与连接健康度
`~/.v42/bin/v42-companion logs -f`	实时查看后台服务日志
`~/.v42/bin/v42-companion start`	启动后台服务
`~/.v42/bin/v42-companion stop`	停止后台服务
`~/.v42/bin/v42-companion update --check`	检查更新（去掉 `--check` 执行升级）
`~/.v42/bin/v42-companion uninstall [--purge]`	卸载服务（加 `--purge` 一并清理本地状态与密钥）

服务器与容器环境细节见 Linux 服务器。

装完之后

审批通过后，这台电脑就会出现在 Team 的电脑列表里。接下来可以直接打开 `https://app.vibe42.ai`，或前往配对设备并开始第一个会话连接手机。

配对设备并开始第一个会话

开通 Team 套餐、免密码配对手机 App、启动终端或 Agent，并验证推送通知直达终端窗口。

vibe42.ai/zh-HANS/docs/getting-started/first-session

当你在至少一台电脑上装好 Vibe42 后（见安装到电脑），按本页步骤连接手机或浏览器，创建第一个会话，并验证从离开屏幕、收到推送到点开继续操作的完整流程。

1. 确认账号与套餐前置条件

在连接远端电脑之前，先确认两件事：

1. **验证账号邮箱**：前往 `https://vibe42.ai/register` 用邮箱和密码（不少于 10 位）注册，或选择 **使用 GitHub 继续** / **使用 Google 继续**。用邮箱注册必须在 24 小时内点击验证链接，未验证的账号在任何设备上都

无法登录。

2. **开通 Team 套餐**：
   - **Team Owner**：在浏览器打开 `https://vibe42.ai/console/checkout?resource=access` 开通 3 天免费试用（需绑卡）或直接订阅套餐。iPhone 与 Android App 内不展示价格、购买或订阅管理入口。
   - **受邀成员**：前往 `https://vibe42.ai/console/invites` 接受邀请即可共享该 Team 的名额，无需自己购买套餐。

2. 选择用来连接的设备

你可以用任意设备组合访问同一台电脑：

- **Web 应用**：在现代浏览器打开 `https://app.vibe42.ai`（也可安装为 PWA，即可以添加到桌面或主屏幕的网页应用），无需安装客户端。如果手边只有手机，可以先在手机浏览器登录 `https://app.vibe42.ai`，再用它配对手机原生 App。
- **手机与平板原生 App**：通过 `https://vibe42.ai/#downloads` 前往 App Store（iOS / iPadOS 17+）或 Google Play（Android 11+）下载。
- **桌面 App**：使用 Mac 或 Linux 上的 Vibe42 App。

3. 配对手机 App

iPhone、iPad 和 Android 原生 App **不使用密码登录**，而是从一台已登录的设备发起配对：

1. **生成配对码**：在已登录的桌面 App 或 Web 应用（`https://app.vibe42.ai`）中打开 **电脑与设备 → 下载与安装 → 配对手机/平板**，界面会显示二维码与 8 位配对码（格式 `XXXX-XXXX`）。
2. **在手机上扫码或输码**：打开手机 App，点击 **扫码配对** 扫描二维码，或手动输入 8 位配对码。
3. **回原设备核对并批准**：回到刚才生成配对码的设备，确认请求设备的型号信息与你的手机一致，点击批准。

> 说明
>
> 配对码有效期为 **10 分钟**，单次有效，一次只能配对一台设备，且只能由生成该配对码的原页面批准。

4. 启动第一个会话

1. 点击侧栏的 **+** 新建终端。
2. 选择目标电脑（或该电脑上的某个 Workspace）。
3. 在第三步选择要启动的程序：
   - **终端**（只打开 shell）：打开你的默认交互式 shell。
   - **Agent**：从列表中选择要启动的 Agent——默认包含 **Claude Code**、**OpenAI Codex CLI**、**OMP**、**Pi Coding Agent**、**OpenCode**、**Grok Build**、**Devin CLI**、**Antigravity CLI** 和 **Cursor Agent**。

这里的 Agent 列表来自 **设置 → Agent**。你可以从内置约 30 个 Agent 的目录中增删、调整顺序，也可以自定义命令。Vibe42 直接启动电脑上已安装的 Agent 程序，不会修改它的配置——详见启动 Agent。

5. 验证完整闭环

用一分钟确认终端常驻与推送通知都已正常工作：

1. 通过侧栏 **+** 启动 **Claude Code**（或其他 Agent），发给它一个需要运行一会儿或会请求权限确认的任务。
2. 将手机锁屏或切到其他应用。手机 App 不在后台常驻，推送通过系统级 APNs（iOS）或 FCM（Android）送达。
3. 当 Agent 状态**变为 需要处理**、**已完成** 或 **出错** 时，手机会收到推送通知。点击通知即可直接打开对应的终端窗口。
4. 在另一台设备上打开 `https://app.vibe42.ai` 并选择同一个窗口，你会看到两边实时同步——会话一直在电脑上运行，关掉 App 也不会停。

桌面 App 仅在运行期间显示系统通知，Web 应用显示页内提示，此外还可以绑定 Telegram。通知渠道与事件过滤规则按每台电脑独立配置，详见 Agent 通知。

常见卡点速查

遇到的提示或现象	排查与解决入口
设备数量已达套餐上限 或 **电脑数量已达套餐上限**	Team 的设备名额或电脑名额已满。前往 Console 登出闲置设备、删除旧电脑或升级套餐，详见套餐与名额。
手机收不到推送通知	检查系统通知权限、该电脑的通知开关以及 Agent 状态是否发生了变化，详见通知故障排查。
电脑显示离线	确认电脑未休眠、已联网且正在运行 Vibe42 后台服务（`v42-companion`），详见连接故障排查。

02 工作原理

电脑、设备与 Team

了解电脑、设备、Team 与套餐名额之间的对应关系，弄清什么情况占用名额以及如何释放名额。

vibe42.ai/zh-HANS/docs/how-it-works/computers-devices-teams

当你想弄清套餐名额是怎么计算的、为什么一台机器会同时占用电脑名额和设备名额，或者更换硬件时如何腾出名额，请查看本页说明。

四个核心对象

对象	定义
电脑（Computer）	运行 Vibe42 后台服务的 Mac、Linux 主机、虚拟机或容器，对外提供终端、Agent、文件与桌面访问。
设备（Device）	已登录的原生 App、已登录的 `app.vibe42.ai` 浏览器会话，以及一行命令安装的 Server 版（显示为 **服务器版**）。
Team	计费与权限的归属单元，名下持有套餐、电脑、设备、成员和 Credits。每个账号注册后自动拥有一个个人 Team。
套餐（Plan）	绑定在 Team 上的订阅方案。套餐决定电脑名额、设备名额、成员席位以及每月 Credits，具体数字见 Console 与定价页 `https://vibe42.ai/#pricing`。

什么会占用名额

Team 名下独立统计三类名额：

- **电脑名额**：每台**已启用**的电脑占 1 个电脑名额。
- **设备名额**：每个已登录的原生 App（macOS、Linux、iOS、Android）、每个登录了 Web 应用（`https://app.vibe42.ai`）的浏览器，以及每台通过一行命令安装的机器（在设备列表中显示为 **服务器版**），各占 1 个设备名额。
- **成员席位**：仅在受邀者**接受邀请**时检查并占用，发送邀请本身不会提前占用席位。

常见机器组合的占用方式：

- **装了桌面 App 的 Mac**：登录 App 占 **1 个设备名额**；点 **启用这台电脑** 启用被控后，再占 **1 个电脑名额**（合计 1 电脑 + 1 设备；若选 **暂不启用** 不启用被控，则只占 1 个设备名额）。
- **启用被控的 Linux 桌面 AppImage 或一行命令安装的 Linux 服务器**：每台机器各占 **1 个电脑名额 + 1 个设备名额**。

以下情况**不会**额外占用名额，也**不会**自动释放名额：

- 在浏览器中登录 Console（`vibe42.ai/console`）管理账单或账号设置**不占**设备名额。
- 在同一台设备上重复登录**不会**占用双份设备名额。

- 机器关机、离线，或在设置里关闭某项功能，**不会**自动释放已占用的电脑或设备名额。

名额满了会怎样

Console 的 **电脑**、**已授权设备** 和 **成员** 页面都会以 `x of y used` 显示当前用量。

- **电脑名额已满**：新电脑仍然可以安装、配对并上线，但会处于未启用状态（**超出套餐名额**），暂时无法连接；等你释放旧名额或升级套餐后会自动启用。
- **设备名额已满**：新设备尝试登录或配对时会停在 **设备数量已达套餐上限**，直到登出闲置设备或升级套餐后才能继续。

如何释放名额与更换机器

- **释放电脑名额**：前往 Console → **电脑** 点击 **删除**。删除电脑会永久清除该电脑保存在云端的 Workspace 与备忘元数据，但电脑本地磁盘上的所有文件与目录保持原样。
  - 删除一台一行命令安装的服务器时，若该安装没有托管其他电脑，会**自动释放**它对应的 **服务器版** 设备名额。
  - 删除一台装了桌面 App 的 Mac 或 Linux 电脑时，**只释放电脑名额**；桌面 App 仍保持设备登录状态，如需一并释放设备名额，请再到 **已授权设备** 页将它登出。
- **释放设备名额**：前往 Console → **已授权设备** 点击目标设备旁的 **解除配对**（登出后该设备需重新登录或配对才能再连）。也可以点击 **解除离线设备配对** 一键登出所有离线超过 3 天的设备。
- **更换电脑与原地升级**：换新硬件时，先在 Console 删除旧电脑，再在新机器上安装；在同一台机器上重新运行一行安装命令属于原地升级，不占用新名额。

谁能管理什么

- **所有者**：负责选择套餐、管理账单与付款，拥有 Team 内电脑、设备与成员的全部管理权限。
- **管理员**：可以删除任意电脑、登出任意设备、查看 Team 用量，但不能更改套餐或账单。
- **成员**：只能登出自己的设备。默认只有 Owner 和 Admin 可以添加新电脑；Member 除非获单独授权，否则不能添加电脑。

电脑在 Team 内的访问范围

每台电脑都可设为 **仅自己**（默认）或 **全团队**，用来控制 Team 内的 Member 能否看到并连接这台电脑。完整的角色权限与共享规则见 Team、角色与共享电脑以及套餐与名额。

连接是怎么建立的

了解 Vibe42 如何就近选择 AP、自动升级 P2P 直连、隔离终端与文件传输通道，以及在企业防火墙下的工

作方式。

vibe42.ai/zh-HANS/docs/how-it-works/connections

当你想了解设备与电脑之间的流量走向、什么时候走点对点直连、为什么传大文件不会卡住终端，或者公司防火墙需要放行哪些出站端口时，请参考本页。

连接路径总览

路径类型	适用端	是否消耗 Credits
Direct（直连）	原生 App（macOS、Linux、iOS、Android）、Vibe42 后台服务与 `v42` CLI 在直接拨号或打洞成功后自动建立	**否**——点对点直连流量完全不计费
Relayed（中转）	浏览器 Web 应用（`app.vibe42.ai`）始终走中转；原生 App 无法打通直连或开启 **强制中继** 时自动使用	**是**——经 AP 中转的业务流量按规则计入 Credits
Mixed（混合）	原生 App 与 CLI 在同一台电脑的三条通道中部分已建立直连、部分仍走中转	仅走中转的那条通道产生的业务流量计入 Credits

AP（接入点）的选择机制

你的设备和电脑都不需要公网 IP 或开放入站端口，双方各自主动向 Vibe42 的 AP 发起出站连接。

启动时，客户端会同时测试各区域节点（包括香港、洛杉矶、东京、法兰克福等）的 RTT（往返延迟），自动连接延迟最低的 AP。你可以在 **设置 → 网络诊断** 中手动指定某个 AP；若指定的节点暂时不可用，客户端会自动切换回其他可用 AP。

从中转平滑升级到直连

每次连接都会先经由 AP 建立，让你不用等待网络探测就能立即输入命令。在原生 App、后台服务与 CLI 之间，程序会在后台交换双方的候选地址（局域网网段、内网网卡、VPN 虚拟网卡以及公网地址），同时尝试直接连接与 NAT 打洞（穿透路由器建立直接通道）。

一旦直连建立成功，流量会自动切换到直连通道，不再经过 AP；如果路由器或防火墙阻断了直连，连接就继续保持在加密中转链路上，不会中断正在使用的终端。

三条相互独立的传输通道

针对每一台已连接的电脑，Vibe42 都会同时维护三条独立的连接：

- **control**：会话状态同步、窗口元数据与管理指令。
- **terminal**：终端输入输出数据流。
- **file**：文件浏览、预览、上传与下载。

由于 `file` 与 `terminal` 走各自独立的连接，上传或下载大文件时不会拖慢你在终端里的按键响应。

断线与自动重连

遇到 Wi-Fi 切换蜂窝数据、笔记本休眠唤醒或网络短暂波动时，连接会自动恢复：

- **快速恢复**：断线后 **30 秒**内重新连回同一 AP 时，会话就地快速恢复。
- **重新握手**：断线时间较长或切换到其他 AP 时，客户端会重新握手，重试间隔在 **1 到 60 秒**之间逐步拉长。
- **固定错误即停**：遇到设备被撤销、电脑或设备名额超限、凭据被拒等无法靠重试恢复的错误，客户端会立即停止重试并显示具体原因。
- **手机切后台重连**：iOS 与 Android App 退到系统后台后会主动断开活动连接，做到后台零耗电；切回前台时自动重新连接，通常几秒内即可恢复。

强制中继

设备端与电脑端都提供 **强制中继** 开关。只要任意一端开启 **强制中继**，Vibe42 就会跳过直连拨号与打洞，全程通过 AP 中转；浏览器 Web 应用始终走中转。

网络诊断

打开 **设置 → 网络诊断** 可查看每 2 秒刷新一次的实时连接状态：

- 当前连接的 AP 名称、区域、传输协议与实时 RTT。
- 每台电脑三条通道各自的路径状态：**直连 QUIC**、**直连 TCP**、**WebSocket**、**中继** 或 **未连接**。

在终端运行 `v42 network [computer]` 也可以输出同样的 JSON 诊断数据（见 CLI 命令参考）。

企业网络与防火墙要求

Vibe42 只需要出站访问权限，不需要配置任何入站规则：

- `ap-<region>-direct.vibe42.app`：UDP QUIC `6042` 端口、TCP `6042` 端口以及 WSS `443` 端口。
- `ap-<region>.vibe42.app`：仅 WSS `443` 端口（通过 Cloudflare 接入）。

客户端按 **QUIC → TCP → WSS** 的顺序依次尝试连接。在仅开放 TCP `443` 出站端口的公司内网或酒店网络中，Vibe42 仍能通过 WSS 正常连接，但严格的企业防火墙通常会拦截 P2P 打洞，因此连接会保持在 AP 中转并消耗 Credits——详见 Credits 与用量及连接故障排查。

安全模型

了解 Vibe42 的端到端 Noise 加密、本地 Ed25519 密钥、零内容留存、审计边界与 60 秒撤销机制。

`vibe42.ai/zh-HANS/docs/how-it-works/security`

当你想核实 Vibe42 如何保护终端与文件传输安全、密钥保存在哪里，以及服务端能看到什么、不能看到什么时，请阅读本页。

Vibe42 的安全设计遵循三条原则：**端到端加密**、**零内容留存**，以及**用短效配对代替机器密码**。

密钥存在哪里、连接如何加密

每个安装端——无论是电脑上的 Vibe42 后台服务还是手机、桌面客户端——都会在本地独立生成一对 **Ed25519 根密钥**。私钥保存在操作系统的安全存储区，**永不离开本机**，Vibe42 服务端不持有任何一端的私钥。

设备与电脑之间的每条连接都会执行端到端 **Noise** 加密握手：

- **P2P 直连**：加密数据直接在你的设备与目标电脑之间点对点传输。
- **经 AP 中转**：会话密钥同样由你的设备与电脑直接协商生成，AP 只负责转发密文，无法解密任何传输内容。

服务端能看到什么、不保存什么、不用作什么

AP / 服务端能看到的

- 已认证的设备与电脑身份标识
- 连接建立时间、所在 AP 区域及字节计数
- 端到端加密后的 Noise 密文
- 通知的标题与正文——唯一一类以可读形式经过并保存在 Vibe42 服务器上的内容，因为服务器必须把它交给 Apple、Google 或 Telegram；可以按电脑关闭，关闭后不会生成（见通知）

我们永远不保存、不读取的

- 终端会话、输入的命令、按键或屏幕输出
- 源代码、目录列表或文件内容
- 端口转发流量内容或会话加密密钥

我们永远不做的

- 用你的会话、代码、文件或元数据训练任何 AI 或机器学习模型
- 经手你的模型 API key：你在本机配置的 key 留在电脑本地，不经过 Vibe42

审计日志记录什么

为了执行名额检查、访问控制与安全审计，Vibe42 仅记录管理操作与路由元数据：

- 账号登录、敏感操作二次验证、设备与电脑的配对和撤销记录。
- 访问控制判断结果与 Team 管理操作。
- 连接路径类型（直连或中转）、直连建立结果、源端与目标电脑 ID、所在 AP 区域、传输总字节数与时间戳。

审计日志绝不包含终端文字、文件内容、端口转发数据或通知正文。

账号安全防护

- **密码存储**：账号密码只保存 **Argon2id** 哈希值。
- **Passkey 与两步验证**：支持在 Console 注册 Passkey（私钥留在设备本地）以及 TOTP 两步验证（验证器应用生成的动态验证码）与一次性恢复码。
- **敏感操作二次验证**：修改安全设置、绑定 GitHub/Google、在账号设备页撤销安装或注销账号时，必须完成二次验证，验证方式优先级为 **Passkey > TOTP > 密码**。

- **第三方登录校验**：**使用 GitHub 继续** 与 **使用 Google 继续** 只接受登录服务商返回且已验证的主邮箱。

具体设置方法见账号与登录。

设备授权与即时撤销

- **短效配对**：配对码有效期 **10 分钟**、单次有效，且只能由生成配对码的原页面批准。
- **60 秒内撤销生效**：你可以随时在 Console 或其他已授权设备上登出设备或删除电脑。撤销操作会在 **60 秒**内生效并强制断开当前连接（包括已建立的 P2P 直连）。
- **Team 严格隔离**：不同 Team 之间的电脑与设备无法互通，服务端也不存在任何可连入你后台服务的管理后门。

你这一侧的安全边界

- **本地系统用户权限**：后台服务以安装时的本地操作系统用户身份运行。任何获准连接该电脑的 Vibe42 成员，拥有的文件读写与命令执行权限都与该本地用户完全相同。
- **Team 访问范围控制**：Team 的 Owner 与 Admin 可访问 Team 内全部电脑；普通 Member 能否访问取决于你为电脑设置的 **仅自己** 或 **全团队**（见 Team、角色与共享电脑）。
- **只批准本人发起的请求**：不要把配对码发给不信任的人，也不要在非本人发起的浏览器登录确认页点击 **批准**。

完整的个人信息与数据处理条款请查阅隐私政策 `https://vibe42.ai/privacy`。

会话、窗口与 最近

了解 Vibe42 如何借助 tmux 保持终端不断线并在电脑重启后把它们重新建起来、通过 最近 追踪跨电脑窗口状态，以及支持多端同屏和多窗口并排查看。

`vibe42.ai/zh-HANS/docs/how-it-works/sessions`

当你想了解关掉 App 后终端里的程序为什么还能继续运行、**最近** 列表的五种状态分别代表什么，或者如何在平板和电脑上并排打开多个终端窗口时，请阅读本页。

基于 tmux 的持久化会话

在每台电脑上，Vibe42 的终端都运行在 tmux（让命令行程序在后台持续运行的终端复用工具）的窗口（window）与分屏（pane，即同一个窗口内切分出的区域）中。无论你是直接划掉手机 App、进电梯断网，还是从办公电脑切到手机，正在运行的任务和 Agent 都会留在原电脑上继续执行；从任意设备重新点开窗口就能接着操作。

> **重要**
> 会话不受网络断开、关闭 App、重启或更新后台服务影响。电脑重启时，正在运行的程序会停止；后台服务再次启动后，会按下一节的说明把窗口重新建起来。

电脑重启之后

后台服务会在本机持续记下每个终端窗口的样子。电脑重启后，后台服务再次启动时，会按这份记录把窗口重新建起来。

只有当这台电脑上由 Vibe42 管理的终端全部已经停掉时才会重建——最常见的情况是电脑重启，终端因为其他原因全部停掉也一样。重建发生在后台服务下一次启动时：后台服务运行在你的系统用户账号下，在 Mac 上会在你重新登录系统账号之后启动；在只用 SSH 的 Linux 服务器上，需要先开启 user lingering（见 Linux 服务器），后台服务才会在开机时自动启动。只重启或更新后台服务（电脑没有重启）时，终端本来就一直在运行，什么都不需要重建，窗口里的程序也不会被打断。

会恢复什么

- **窗口和分屏（pane）**：同样的分屏和布局。
- **窗口名**：有固定名字的窗口保持原名；其余窗口照旧跟着正在运行的命令自动改名。
- **自定义标题**：你用 **重命名窗口** 设置的自定义标题。
- **Workspace**：窗口所属的 Workspace。
- **终端隔离**：开启了终端隔离的窗口，只要那份隔离副本的文件夹还在，就会重新连回这份副本。
- **所在文件夹**：每个分屏所在的文件夹；那个文件夹已经不存在时，改用默认文件夹。
- **当前选中项**：重启前处于活动状态（当前选中）的窗口和分屏。

每个分屏重建后，顶部会先回放重启前保存的输出（带颜色），然后是一行颜色较暗的提示，说明会话已恢复、上方内容是什么时候保存的，之后才是新的 shell 或程序。全屏程序（例如编辑器、`htop` 这类占满整屏的程序）只保存当时那一屏。

Agent 与启动命令

- **通过启动命令打开的窗口**：通过新建终端流程选了 Agent、或用 `v42 term new --agent`、`--command` 启动的窗口，重建时会再次运行同一条启动命令。支持按会话重新打开的 Agent（包括全部 9 个默认 Agent，见支持的 Agent）会直接打开上一次的对话；对话内容本身保存在 Agent 自己的存储里，Vibe42 只负责把它重新打开。如果重启前 Agent 还没开始任何对话（例如还没发过第一条消息），就只是重新启动这个 Agent。重启那一刻正在进行的那一轮工作已经被中断，重新打开对话后不会自己继续，需要你再发一条消息让它接着做。不支持按会话重新打开的 Agent 以及自定义命令，会重新运行同一条命令，从头开始。
- **自己在普通 shell 里敲命令启动的 Agent**：只回放输出，然后在同一个文件夹里给你一个普通 shell，不会自动重新运行这个 Agent。
- **其他程序**：开发服务器、构建、测试等程序不会自动重新运行；它们的输出会回放，你需要手动再启动。

限制

- **保存容量**：每个分屏最多保存 1 MiB 输出，超出时丢掉最早的部分；整台电脑合计最多 64 MiB，超出时先丢不是 Agent 的分屏。
- **保存时机**：输出停下 5 秒后保存一次；一直有输出时最多每 30 秒保存一次；后台服务正常退出（包括正常关机）时再保存一次。突然断电时，最后大约 30 秒的输出可能没保存上。
- **仅保存在本机**：保存的输出只留在这台电脑上，不会上传到 Vibe42。

窗口与分屏操作

你可以通过窗口与分屏菜单随时调整布局（快捷键与手势说明见窗口与分屏）：

- **在此目录新建窗口**：在当前目录下新建一个 tmux 窗口。
- **向右分屏 / 向下分屏**：向右或向下切分当前终端，新建一个分屏。
- **重命名窗口**：设置自定义窗口名，覆盖终端程序自动更新的标题（后台服务运行期间一直保留，电脑重启后也会恢复；但只重启或更新后台服务时，自定义标题可能变回自动标题）。
- **移入此窗口**：把已有的分屏移入当前窗口。
- **拆为独立窗口**：把当前分屏拆分出来，变成独立的新窗口。
- **关闭窗口 / 关闭面板**：关闭窗口或分屏，执行前会弹窗提醒其中的运行进程将被终止。

最近

最近 汇总了你在各台电脑上最近打开过的窗口，上限为 **100 条**，支持把常用窗口固定（**置顶**）在顶部。

每条记录会显示以下五种状态之一：

状态	含义
可用	目标电脑在线且窗口存在，点击即可打开。
电脑离线	目标电脑当前离线或暂时无法连接。
正在同步…	已连上电脑，正在同步最新的窗口信息。
窗口已关闭	电脑端已确认该 tmux 窗口已关闭。
窗口已失效 · 可从最近使用中移除	来自早期不兼容会话格式的旧记录。

只有收到电脑端确认窗口已删除时，Vibe42 才会把条目标记为 **窗口已关闭**——单纯断网或离线绝不会被标为已关闭。此外，已关闭的条目**不会自动消失**，让你清楚看到窗口去向，而不是无声消失；确认不再需要后点 **从最近隐藏** 即可手动移除。电脑重启后重建出来的窗口和重启前是同一个窗口，**最近** 里的条目和置顶（**置顶**）会继续指向它，不会显示为已关闭；网页版打开着的终端链接也会自动跳到重建后的窗口。

多设备同时查看

多台设备（或共享电脑上的多位 Team 成员）可以同时打开同一个终端窗口查看并输入。界面上的 **正在访问的连接** 会显示当前同时连在该窗口上的其他连接。

在新窗口打开

需要并排查看多个终端时，打开侧栏条目的菜单并选择 **在新窗口打开**。关闭弹出的新界面窗口只会关掉本地视图，**不会终止**远端电脑里正在运行的程序。

平台	在新窗口打开 的行为
macOS App	弹出错开排列的原生 macOS 窗口
iPad App	打开 iPadOS 系统多窗口（分屏浏览 / 台前调度）

平台	在新窗口打开 的行为
iPhone App	不提供此选项
Android App	仅在 Android 桌面模式下提供
Web 应用	打开独立的浏览器弹出窗口

与你自己的 tmux 会话的关系

Vibe42 管理独立的 tmux 会话，不会改动你自己平时使用的默认 tmux 会话。需要注意一点区别：通过 Vibe42 界面或 `v42` CLI 创建的分屏，会通过后台服务自动响应 OSC 7501 能力探测（程序向 Vibe42 报告状态用的一串控制序列），从而让支持该协议的 Agent 完整上报状态；而如果你在 shell 里直接手动运行 `tmux` 命令新建分屏，则不会挂载后台服务的探测响应，详见 OSC 7501 协议。

03 终端

终端引擎

了解 Vibe42 如何基于 Ghostty 渲染终端、编码 TUI 组合键、显示终端内图片，以及跨设备处理剪贴板与文件附件。

vibe42.ai/zh-HANS/docs/terminal/engine

Vibe42 在所有客户端上统一使用 Ghostty 终端内核，让命令输出、全屏 TUI（在终端里画界面的程序，例如 Claude Code 或 Neovim）和 Agent 在浏览器、手机、平板与桌面 App 上表现一致。想了解各端渲染方式、组合键支持、终端内图片配额、跨设备复制或粘贴附件限制时，可以查阅本页。

Ghostty 内核与各端渲染路径

所有 Vibe42 客户端共用来自 Ghostty 的 VT 解析器（负责解析终端文字与控制序列的组件），并调用当前设备的原生图形接口由 GPU 渲染。画面刷新率自动跟随屏幕，在高刷新率屏幕上同样流畅。

客户端平台	仿真内核	渲染路径
macOS 与 iOS / iPadOS	`libghostty-vt`	Metal
Linux 桌面 App	`libghostty-vt`	wgpu
Android	`libghostty-vt`	TextureView（自动跟随高刷屏）
Web 应用（`app.vibe42.ai`）	`ghostty-vt`（WASM）	WebGL，不可用时自动回退至 Canvas 2D

终端内的文字颜色和背景色由正在运行的程序决定，整体配色方案可随时在主题与字体中切换。

高速输出与平滑回滚

文字绘制和画面合成都在 GPU 上完成，不依赖浏览器 DOM 节点排版，因此快速滚动的编译日志和频繁重绘的 TUI 界面都能保持流畅响应。

- **本地回滚缓冲**：每个分屏（终端窗口切分出的一个区域）在当前设备上保留 10,000 行本地历史记录，上下翻看时无需等待网络加载。
- **同步输出（DEC 2026 模式）**：当 TUI 程序或 Agent 用成对标记包裹一整屏更新时，Vibe42 会等整批内容收齐后一次性画出，避免画面闪烁或撕裂。
- **跟随屏幕刷新率**：渲染节奏自动匹配当前屏幕，无需手动调节帧率参数。

键盘协议与 TUI 组合键

Claude Code、终端编辑器等现代终端程序依赖扩展键盘协议来区分 `Shift+Enter`、`Ctrl+;`、`Alt+方向键` 与普通按键。

电脑上的 Vibe42 后台服务会实时记录当前程序开启了哪些终端输入模式：

- **Kitty 键盘协议**（Kitty keyboard protocol）
- **xterm** `modifyOtherKeys`
- **应用光标键模式**（Application cursor keys）
- **括号粘贴模式**（Bracketed paste）

无论你在哪台设备上按键，后台服务都会在电脑上按当前终端所处的输入模式完成编码。TUI 程序里的组合键表现与本地终端一致，不需要自己配置转义序列。由客户端优先处理的全局快捷键请参见键盘快捷键。

终端内图片（Kitty 图形协议）

Vibe42 支持 Kitty 图形协议，命令行工具可以直接在终端内显示图表、截图和预览图。后台服务会在电脑上解码图片数据，再发给正在查看该终端的设备。

限制项	配额
单张图片大小上限	16 MiB
每个分屏的图片存储上限	64 MiB
每个分屏同时显示张数（Web 应用）	最多 5 张

OSC 52 跨设备复制

当电脑上的程序发出 OSC 52 序列（一种用于复制文本的终端控制指令，例如在 Neovim 里复制或在 tmux 内选中复制）时，Vibe42 会把这段文本写入**当前正在查看该终端的设备**的系统剪贴板。这是终端内程序主动向设备剪贴板写入内容的唯一方式。

> 说明
>
> 在桌面 App 上，收到 OSC 52 序列时 Vibe42 窗口必须处于前台焦点状态，才能写入系统剪贴板。每次复制成功后，界面都会弹出提示。

粘贴与附件限制

你可以直接向终端粘贴文本、粘贴图片或拖入文件。Vibe42 会把图片和文件上传到目标电脑，并在终端光标处插入对应的电脑文件路径。

输入类型	上限与处理方式
纯文本粘贴	单次不超过 1 MiB
图片粘贴或上传	4 MiB 以内原样上传；超出 4 MiB 的图片及 HEIC 照片会自动转为 JPEG（最长边不超过 4096 像素）
拖放或粘贴文件	自动上传至电脑，并将电脑上的文件路径插入当前终端
手机与平板附件	支持直接拍照、从相册选图或从文件选择器上传（详见在手机和平板上输入）
插入远端路径	不上传，只把电脑上已有文件或文件夹的路径（单引号转义）插到光标处，不按回车（详见文件）

在手机和平板上输入

在 iPhone、iPad 和 Android 上使用命令栏、可拖动的 方向键盘、修饰键锁定、触控手势、软键盘避让模式与拍照附件操作终端。

vibe42.ai/zh-HANS/docs/terminal/mobile-input

手机和平板的软键盘缺少实体修饰键与 `Esc` 键，频繁弹出收起还容易打乱全屏 TUI（在终端里画界面的程序）的排版。Vibe42 为 iPhone、iPad 和 Android 提供了触控输入控件，并支持平板外接实体键盘，方便你在移动设备上操作命令行与 Agent。需要设置移动端按键、触控手势或输入法行为时，可以参考本页。

命令栏（Composer）

直接在实时终端里用触屏打字容易被系统自动纠错干扰。要给 Agent 写多行指令、在运行前核对命令参数，或者粘贴并修改一段长命令时，可以先展开命令栏编辑再发送：

- 点击 **发送到终端** 将写好的内容发往终端。
- 选择 **发送附带回车** 立即提交执行，或选择 **逐行发送** 把多行内容逐行发给交互式程序或 shell。

可拖动的 方向键盘 与修饰键

当 Agent 或 TUI 程序弹出列表等你选择时，唤起系统软键盘会挡住大半个屏幕。这时可以直接使用悬浮的 **方向键盘**：

- 提供上下左右四个方向键和 `Space`，操作时不会弹出系统软键盘。
- 按住即可拖到屏幕任意位置，避免挡住当前提示内容。

辅助工具栏上的修饰键（如 `Ctrl`、`Alt`）支持单次触发与锁定：

- **点一下**：只对接下来按下的那一个键生效，按完自动松开。
- **双击**：保持锁定状态，适合连续按多组组合键；再点一次即可解除锁定。

自定义快捷按钮

你可以把常用的按键或组合键固定到移动端快捷栏上：

- 自定义快捷按钮保存在云端账号中，登录同一账号的多台移动设备会自动同步。
- 输入标准键名时，按钮会自动显示为对应的按键图标，节省横向空间。

触控手势

在终端画面上使用不同手势，可以分别执行历史翻阅、文本选择和 TUI 鼠标操作：

手势	作用
单指上下滑动	翻看当前分屏（终端窗口切分出的一个区域）的本地历史记录
长按（500 ms）	选中词语，拖动选区手柄即可复制文本

手势	作用
双指拖拽	当终端内的程序开启鼠标模式时，向程序发送滚动或拖拽操作
双指捏合	放大或缩小当前设备上的终端字号

软键盘避让模式

如果每次软键盘弹出或收起都重新调整 PTY（终端和程序之间的底层通道）的行列数，全屏 TUI 界面就会反复重排闪烁。Vibe42 提供三种软键盘避让模式，可按习惯切换：

模式	软键盘弹出时的行为
页面平移	将终端画面整体往上推，不改变 PTY 的行数和列数
终端缩放	按软键盘上方的剩余可视高度缩小终端行数
光标对齐	保持 PTY 尺寸不变，只在光标会被软键盘挡住时平移画面跟随光标

拍照、相册与文件附件

在手机上想把界面截图发给 Agent 排查问题，或者传入手机里的文件时，可以直接从终端添加附件：

1. 打开附件菜单，选择 **拍照**、**从图库选择** 或 **选择文件**。
2. Vibe42 会把文件上传到电脑，并在当前终端的光标处插入电脑上的文件路径。

菜单里还有 **插入远端路径**，用来把那台电脑上已有文件或文件夹的路径插进终端：在弹出的文件浏览窗口里选中文件或文件夹，路径就会插到光标处，不上传，也不自动按回车（详见文件）。

超过 4 MiB 的图片和 HEIC 格式照片会在上传前自动转为 JPEG（最长边不超过 4096 像素）。完整的粘贴与附件配额见终端引擎。

iPad 外接键盘与输入法

给 iPad 连上外接实体键盘后，你可以直接使用桌面端风格的快捷键。随时长按 `⌘` 键即可在屏幕上弹出快捷键提示表；完整键位列表见键盘快捷键。

针对中文等需要候选词组字的输入法（IME）：

- **iOS 与 iPadOS**：直接调用系统原生输入通道，候选词栏和拼音组字都能正常使用。
- **Web 应用**：先在终端内显示正在输入的拼音或组字状态，确认上屏后再把文字发往终端。
- **Linux 桌面 App**：在 Wayland 下开箱即用；如果在 X11 会话下使用，请安装 `fcitx5` 或 `ibus` 对应的 GTK4 输入法模块。

窗口、分屏与终端列表

管理一直在电脑上运行的终端窗口与分屏布局，浏览和排序首页终端列表，跨电脑全局搜索，并查看同一终端的实时在线连接。

vibe42.ai/zh-HANS/docs/terminal/windows-and-panes

Vibe42 里的每个终端窗口都会一直在你的电脑上运行，你还可以把一个窗口切分成多个分屏（同一个窗口里并排或上下排列的独立区域）。关掉客户端或切换设备都不会打断正在运行的程序，会话保持机制详见会话工作原理。需要调整分屏布局、浏览终端列表或快速跳转到某个会话时，可以参考本页。

终端窗口与分屏操作

窗口和分屏由电脑上的 Vibe42 后台服务统一维护，因此你在桌面 App 上调整的分屏布局会立即同步到手机或浏览器上。

打开窗口与分屏菜单即可调整布局：

菜单项	说明
在此目录新建窗口	在同一台电脑（以及当前所在的 Workspace）里新建一个终端窗口
向右分屏	将当前分屏左右切分，在右侧打开新的终端分屏
向下分屏	将当前分屏上下切分，在下方打开新的终端分屏
重命名窗口	修改终端窗口标题，修改后在你的所有设备上同步显示
移入此窗口	将另一个终端移入当前窗口，合并成一个分屏
拆为独立窗口	将当前分屏拆分出来，变成一个独立的终端窗口
关闭面板	确认后关闭当前分屏
关闭窗口	确认后关闭整个终端窗口以及里面的所有分屏

重命名与重组分屏

你可以用 **重命名窗口** 给长时间运行的任务起个好认的名字（比如 `api-server` 或 `claude-refactor`），在手机小屏幕上也能一眼找到。如果左右分屏在手机上显得太挤，选择 **拆为独立窗口** 就能把当前分屏拆成独立窗口；之后需要并排看日志和代码时，再用 **移入此窗口** 把两个终端合并到同一个窗口里。

关闭终端与断开连接的区别

退出 Vibe42、手机锁屏或断网都只是关掉当前设备上的画面，电脑上的程序会继续运行。而选择 **关闭窗口** 或 **关闭面板** 则会真正关掉电脑上的终端窗口或分屏——Vibe42 会先弹出确认框，提醒你里面正在运行的程序将被停止。

在新应用窗口中打开

在桌面 App 和 Web 应用上，你可以打开多个独立的应用窗口（**新建窗口**：macOS 为 `⌘N`，Apple 浏览器为 `⌥⌘N`；Linux 桌面为 `Ctrl+Shift+N`，其他平台浏览器为 `Alt+Shift+N`），在多块显示器上同时查看多个终端。

在 macOS 桌面 App 中关闭应用窗口（`⇧⌘W`）只会关掉本机屏幕上的这个窗口，不会停止电脑上正在运行的终端。不同操作系统与浏览器的多窗口支持有所不同，具体选项以 App 内显示为准。所有窗口与分屏快捷键请参见键盘快捷键。

首页终端列表

首页终端列表汇总展示了当前 Team 下各台在线电脑正在运行的终端：

- **按电脑分组**：把终端按所在的电脑分组显示。
- **按添加时间排序 / 按名称排序**：按创建时间或按名称字母顺序排序。
- **列表视图 / 网格视图**：在紧凑的列表视图与网格视图之间切换。

如果你按目录管理项目，也可以通过左侧栏的 **Agent**、**Workspace** 与 **最近** 标签页浏览终端，详见 Workspace。

全局搜索与 正在访问的连接

按下 `⇧⌘P`（Windows、Linux 与 Android 为 `Ctrl+Shift+P`，Firefox 浏览器下不可用）即可打开 **全局搜索**，搜索并直接跳转到：

- **Workspace**
- **终端**
- **远程桌面屏幕**
- **远程桌面窗口**

当有其他连接同时打开了这个终端时（例如你的手机、另一个浏览器标签页，或共用这台电脑的 Team 成员），界面上的 **正在访问的连接** 会实时显示当前正在查看该终端的所有连接。

键盘快捷键

Vibe42 在 macOS、iPadOS、Linux、Windows、Android 与各浏览器下的完整键盘快捷键参考，以及终端焦点与浏览器替代键规则。

`vibe42.ai/zh-HANS/docs/terminal/shortcuts`

Vibe42 提供了用于切换终端、切分分屏（终端窗口内的独立区域）、跳转 Workspace 和调节字号的键盘快捷键。在 macOS、Linux 桌面 App、连接外接键盘的 iPad 或 Web 应用上使用键盘操作时，可随时查阅本页。

快捷键触发规则

为避免与 shell、TUI（在终端里画界面的程序）以及浏览器快捷键冲突，Vibe42 按以下四条规则处理按键：

- **终端焦点优先传给程序**：当光标在终端内时，按键默认直接发给正在运行的程序（例如 `Ctrl+B` 或 Vim 快捷键），只有下表列出的快捷键才由 Vibe42 拦截。
- **弹窗打开时不触发**：界面上有弹窗、面板浮层或确认框打开时，全局快捷键暂停响应。
- **Web 浏览器替代键**：浏览器自身占用了 `⌘N`、`⌘T`、`⌘W`（或 `Ctrl+N`、`Ctrl+T`、`Ctrl+W`）等快捷键；在 `app.vibe42.ai` 中，这些操作改用带 `⌥` 或 `Alt` 的组合键（见下表括号标注）。
- **Firefox 浏览器限制**：Firefox 将 `Ctrl+Shift+P` 固定用于打开隐私浏览窗口，且不允许网页覆盖，因此在 Windows 与 Linux 的 Firefox 中无法通过快捷键打开 全局搜索。

> **提示**
> 在连接了外接键盘的 iPad 上，长按 `⌘` 键即可在屏幕上弹出快捷键速查表。更多平板输入说明见在手机和平板上输入。

完整快捷键列表

下表两列对应的平台范围如下：

- **mac**：macOS 桌面 App、连接外接键盘的 iPadOS，以及 Apple 设备上的 Web 应用（括号内为 Web 专用组合键）。
- **other**：Linux 桌面 App、连接外接键盘的 Android，以及 Windows 或 Linux 上的 Web 应用（括号内为 Web 专用组合键）。

动作	mac	other
设置	`⌘,`	`Ctrl+,`
新窗口	`⌘N`（Web `⌥⌘N`）	`Ctrl+Shift+N`（Web `Alt+Shift+N`）
新建 Workspace…	`⇧⌘N`（Web `⌥⇧⌘N`）	`Ctrl+Alt+N`
新建终端	`⌘T`（Web `⌥⌘T`）	`Ctrl+Shift+T`（Web `Alt+Shift+T`）
全局搜索	`⇧⌘P`	`Ctrl+Shift+P`（Firefox 不可用）
显示/隐藏侧栏	`⌘B`	`Ctrl+B`（终端外）/ `Ctrl+Shift+B`
切换全屏	`⌘↩` 或 `^⌘F`	`Ctrl+Enter`
关闭窗口	`⇧⌘W`（Web 无）	—
关闭所有窗口	`⌥⇧⌘W`（仅 macOS）	—
关闭终端	`⌥⌘W`（Web `⌥⇧⌘W`）	`Ctrl+Shift+W`（Web `Alt+Shift+W`）
关闭面板	`⌘W`（Web `⌥⌘W`）	—
上一个终端 / 下一个终端	`^⇧Tab` / `^Tab`；`⇧⌘[` / `⇧⌘]`（Web `⌥⇧⌘[` / `]`）	`^⇧Tab` / `^Tab`；`Ctrl+Shift+←` / `→`；`Ctrl+PgUp` / `PgDn`
Terminal 1–8 / 最后一个终端	`⌘1-8` / `⌘9`	`Alt+1-8` / `Alt+9`
Workspace Group 1–8 / 最后一个 Workspace 分组	`^⌘1-8` / `^⌘9`	`Alt+Shift+1-8` / `Alt+Shift+9`
向右分屏 / 向下分屏	`⌘D` / `⇧⌘D`	`Ctrl+Shift+O` / `Ctrl+Shift+E`

动作	mac	other
上一个面板 / 下一个面板	`⌘[` / `⌘]`	`Ctrl+Meta+[` / `]`
选择面板 ↑↓←→	`⌥⌘方向`（Web `⌥⇧⌘方向`）	`Ctrl+Alt+方向`
Move Divider ↑↓←→	`^⌘方向`	`Ctrl+Meta+Shift+方向`
放大字号 / 缩小字号 / 重置字号	`⌘=` 或 `⌘+` / `⌘-` / `⌘0`	`Ctrl+=` / `Ctrl+-` / `Ctrl+0`
滚动到顶部 / 滚动到底部	`⌘Home` / `⌘End`	`Shift+Home` / `Shift+End`
上一页 / 下一页	`⌘PgUp` / `⌘PgDn`	`Shift+PgUp` / `Shift+PgDn`
复制 / 粘贴 / 全选	`⌘C` / `⌘V` / `⌘A`	`Ctrl+Shift+C`（`Ctrl+Insert`）/ `Ctrl+Shift+V`（`Shift+Insert`）/ `Ctrl+Shift+A`
行首 / 行尾 / 删到行首；按词左右移	`⌘←` / `⌘→` / `⌘⌫`；`⌥←` / `⌥→`	—

有关终端内 TUI 与 Agent 程序使用的扩展键盘协议，请参见终端引擎。

没有默认键位的菜单项

以下功能可从应用菜单栏或界面按钮打开，没有默认快捷键：

- **显示/隐藏右栏**
- **电脑**
- **连接新的电脑/服务器…**
- **配对手机/平板…**
- **键盘快捷键**

暂不提供的功能

Vibe42 目前不提供以下操作及快捷键：

- 使用 `⌘K` 清屏（请在 shell 内运行 `clear` 或按 `Ctrl+L`）
- 临时放大单个分屏
- 自动均分各分屏尺寸

主题与字体

从 114 套深浅配色中分别配置 日间 与 夜间 主题，切换内置等宽编程字体，调节字号、行高与连字，并保持中日韩双宽字符严格对齐。

vibe42.ai/zh-HANS/docs/terminal/themes-and-fonts

在大屏显示器上看着舒服的字号和对比度，到了户外的手机屏幕上往往看不清。Vibe42 内置 114 套配色主题和 5 款等宽编程字体，并把终端显示设置单独保存在每台设备本地。想设置深浅色切换、调节字体与行高，或了解中文对齐方式时，可以阅读本页。

配色主题与 日间 / 夜间 切换

Vibe42 内置 **114 套配色方案**（**81 套深色**、**33 套浅色**），选自 `terminalcolors.com`。

- **日间与夜间分别设置**：你可以为 **日间**（浅色外观）和 **夜间**（深色外观）分别选一套配色。
- **跟随系统**：支持固定浅色、固定深色，或跟随系统外观自动切换日夜配色。
- **界面与终端同步换色**：切换主题时，侧栏、标签栏等外围界面会和终端画面一起换色，不会出现深色终端套在刺眼浅色边框里的情况。

程序在 shell 里输出的 ANSI 颜色会自动对应到当前主题的色板上；渲染细节见终端引擎。

内置字体与排版设置

所有客户端都自带以下 5 款常用等宽编程字体，不用在每台设备上手动装字体就能获得一致的显示效果：

- **Geist Mono**
- **JetBrains Mono**
- **Fira Code**
- **Blex Mono**
- **Noto Sans Mono**

在 **设置** 中可以调整 5 项文字排版参数：

设置项	作用
字体	从上述 5 款内置等宽字体中选择主字体
字号	设置终端的基础字号
行高	调节行间距，可根据习惯调紧凑看日志或调宽松阅读
连字	在支持连字的字体中开启或关闭编程连字（比如把 `->`、`===`、`!=` 连成单个符号）
字体加粗	略微加粗终端文字笔画，在普通分辨率屏幕或强光下更容易看清

随时缩放字号

临时投屏演示或换到手机查看时，不用进 **设置** 就能直接改字号：

- **键盘快捷键**：在 macOS 与 iPadOS 上按 `⌘=`（或 `⌘+`）放大、`⌘-` 缩小、`⌘0` 恢复默认字号；在 Linux、Windows 与 Android 上对应 `Ctrl+=`、`Ctrl+-` 与 `Ctrl+0`。完整键位见键盘快捷键。
- **触控手势**：在 iOS 或 Android 的终端画面上用双指捏合即可缩放字号，详见在手机和平板上输入。

CJK 双宽对齐与多级字体栈

中英文、终端图标与表格线混排时，如果备用字体的宽度不一致，很容易把 TUI（在终端里画界面的程序）的边框和分栏顶歪。Vibe42 按固定的四级顺序查找字形：

```
主字体 → 符号兜底 → CJK 兜底 → monospace
```

每个中日韩全角字符都固定占两个半角字符宽度，保证分屏分隔线、命令行表格和 Agent 输出的代码对比整齐对齐。

仅在当前设备本地保存

所有终端显示设置——包括外观模式、日夜主题、字体、字号、行高、连字开关和 **字体加粗**——都**只保存在当前设备本地**，不会跨设备同步。在手机上调大字号或换用浅色高对比主题，不会改变桌面电脑上的显示效果。

04 Coding Agent

支持的 Coding Agent

了解哪些 Coding Agent 能在 Vibe42 中运行、三档状态上报机制的区别，以及如何添加自定义 Agent 命令。

`vibe42.ai/zh-HANS/docs/agents/supported-agents`

任何在终端里运行的程序，都能在 Vibe42 中直接使用。Vibe42 连接的是你 Mac 或 Linux 电脑上一直在后台运行的终端，并由 Ghostty 终端引擎渲染画面，运行某个工具前不需要专门安装插件。你想了解不同 Agent 的状态指示点和推送通知如何工作时，可以查看本页。

三档状态支持

任何命令行工具都能在窗口里交互运行。而 **Agent** 面板、侧栏状态点以及通知依赖两套机制获取运行状态：OSC 7501（程序向 Vibe42 报告状态用的一串控制序列），以及 Vibe42 后台服务内置的包装引擎。按状态来源分为三档：

1. **原生完整支持**：Agent 自己输出完整的 OSC 7501 序列，直接上报状态、进度、本轮摘要与暂停原因，不经过包装引擎。目前支持 **Claude Code**（≥ 2.1.295）和 **Pi Coding Agent**（≥ 1.1.0）。
2. **部分自报 + 引擎补充**：Agent 上报部分状态信号，其余状态由后台服务的包装引擎根据屏幕规则与会话记录补齐。包括 **OpenAI Codex CLI**、**OMP**、**Grok Build**、**Antigravity CLI**、**Crush** 和 **Amp**。
3. **引擎代报**：Agent 本身尚未支持 OSC 7501。当你从 Vibe42 启动它时，后台服务的包装引擎会根据屏幕输出规则与本地会话记录判断状态并代为上报。内置目录中的其余 Agent 都属此类，包括 **OpenCode**、**Devin CLI**、**Cursor Agent**、**Gemini CLI**、**GitHub Copilot CLI**、**Cline**、**Kimi**、**Kiro**、**Droid**、**Aider**、**Goose** 等。

> **重要**
>
> 包装引擎只接管从新建终端流程或 `v42 term new --agent` 启动的 Agent。如果你在普通命令行里手动敲命令启动程序，只有原生完整支持 OSC 7501 的 Agent 会显示状态和触发推送。

默认列表与内置目录

在新建终端流程的第三步，第一项固定为 **终端**（只打开 shell），下方是你的 Agent 列表。默认列表预置了 9 个常用 Agent；内置目录共收录约 30 个 Agent，你可以在设置的 **Agent** 分区中增删或调整顺序。

Agent	如何启动	状态来源	备注
Claude Code	新建终端流程、`v42 term new --agent`，或在 Vibe42 创建的分屏中手动运行	原生 OSC 7501（≥ 2.1.295）；更旧版本由引擎代报	默认列表；支持完整状态、摘要与权限确认细分
OpenAI Codex CLI	新建终端流程或 `v42 term new --agent`	部分自报 + 引擎补充	默认列表

Agent	如何启动	状态来源	备注
OMP	新建终端流程或 `v42 term new --agent`	部分自报 + 引擎补充	默认列表
Pi Coding Agent	新建终端流程、`v42 term new --agent`，或在 Vibe42 创建的分屏中手动运行	原生 OSC 7501（≥ 1.1.0）	默认列表
OpenCode	新建终端流程或 `v42 term new --agent`	引擎代报	默认列表
Grok Build	新建终端流程或 `v42 term new --agent`	部分自报 + 引擎补充	默认列表
Devin CLI	新建终端流程或 `v42 term new --agent`	引擎代报	默认列表
Antigravity CLI	新建终端流程或 `v42 term new --agent`	部分自报 + 引擎补充	默认列表
Cursor Agent	新建终端流程或 `v42 term new --agent`	引擎代报	默认列表
Crush、Amp	新建终端流程或 `v42 term new --agent`	部分自报 + 引擎补充	位于内置目录
Gemini CLI、GitHub Copilot CLI、Cline、Kimi、Kiro、Droid、Aider、Goose 等	新建终端流程或 `v42 term new --agent`	引擎代报	位于内置目录（共约 30 个）

不在目录中的 Agent 怎么办

如果你使用刚发布的新工具、团队内部 Agent 或自定义启动脚本，可以在设置的 **Agent** 分区添加自定义项：

- 填写显示名称（≤ 64 字符）与单行启动命令（`<program> <args>`）。
- Agent 类型保持 **根据命令识别**；如果你的脚本底层调用了目录中已有的某个 Agent，也可以手动指定对应类型以启用它的识别规则。
- 如果工具既不在识别规则内，自己也不输出 OSC 7501，状态可能只有 **工作中** 和 **空闲**（具体以 App 内显示为准）。

如果你能修改该工具或外层脚本，只需按 OSC 7501 状态协议向终端打印几行控制序列，即可接入完整的 **工作中**、**已完成**、**需要处理** 与 **出错** 状态追踪和手机推送。

不注入、不锁定

Vibe42 不向 Agent 注入 hook、插件或 skill，不修改配置文件，也不代理 API 调用。代码与密钥始终留在你

的电脑上。

vibe42.ai/zh-HANS/docs/agents/zero-injection

Vibe42 把 Coding Agent 当作运行在你电脑上的普通终端程序，而不是托管在我们服务里的插件。如果你想确认 Vibe42 会不会改动本机文件、API key 与源代码流向哪里，或者想了解在不改动配置的前提下如何追踪状态，可以查看本页。

我们不做什么

无论你安装 Vibe42 还是从界面启动 Agent，Vibe42 都遵守以下边界：

- **不写 hook**：不会往你的代码仓库或 Agent 目录写入任何 hook（在特定事件时自动触发的脚本）。
- **不装 skill 或插件**：不会向任何 Agent 注入自定义 skill、扩展包或辅助组件。
- **不改配置文件**：不会修改 `~/.claude` 或其他任何 Agent 的本机配置。
- **不代理 API 调用**：模型请求直接从你的电脑发往对应的模型服务商。API key、登录凭据与源代码始终留在你的电脑上，直接使用你自己的订阅额度。

Vibe42 里的 Agent 列表，只是一组跨设备同步的“启动命令书签”。

这对你意味着什么

- **切换 Agent 零成本**：你可以随时试用新发布的命令行工具，或在同一个 Workspace 里同时跑三种不同的 Agent。停用某个工具时，不需要清理任何残留配置。
- **Agent 升级不受影响**：上游发布新版本、调整配置格式或新增交互功能时，不需要等 Vibe42 发版适配，打开终端就能直接用。
- **安全边界清晰**：Vibe42 只在你的电脑与已配对设备之间端到端加密传输终端画面和文件（详见安全架构），不介入电脑与大模型 API 之间的通信。

不注入，状态是怎么来的

Vibe42 不靠修改配置文件获取事件，而是通过两条非侵入路径获取 Agent 状态：

1. **OSC 7501 控制序列**：原生支持该协议的 Agent 像输出颜色代码或修改窗口标题一样，直接向终端打印状态序列；Vibe42 后台服务解析后将其剥离，屏幕上不会留下杂字符。
2. **后台服务包装引擎**：对于尚未完整支持 OSC 7501 的受支持 Agent，当你从 Vibe42 启动它时，后台服务的包装引擎会在进程外围根据屏幕输出规则和 Agent 自身的会话记录判断状态并代为上报。

与其他实现方式的取舍

市面上连接远程 Agent 的方案通常采用以下三种做法之一：

机制	集成方式	带来的取舍
注入 hook / 插件	改写 Agent 配置目录，通过回调脚本上报事件	改动本机配置，Agent 升级改格式后容易失效，且无法支持不提供 hook 机制的工具

机制	集成方式	带来的取舍
包装成专用聊天界面	用自定义表单或聊天框包裹底层 SDK / CLI	丢失原生终端交互、斜杠命令与快捷键，有时还要求把 API key 填给第三方
终端原生方案（Vibe42）	在 tmux 中直接运行原版 CLI，通过 OSC 7501 或外部观察分屏（窗口里的一个终端区域）提取状态	100% 保留原生终端体验且不改动配置文件，同时提供跨工具统一状态面板与手机推送

唯一的“接触”与绕过方式

Vibe42 与 Agent 进程唯一的接触点是包装引擎：当你通过新建终端流程或 `v42 term new --agent` 启动未原生完整支持 OSC 7501 的 Agent 时，后台服务会观察该分屏的屏幕输出以代报状态（原生完整支持 OSC 7501 的 Agent 则直接跳过包装引擎）。

如果你希望彻底绕过包装引擎，只需在新建窗口时选择第一项 **终端**（只打开 shell），再在普通命令行里手动敲命令启动 Agent。包装引擎不会介入你在终端里手动运行的任何命令。

启动 Agent

通过新建终端流程、快捷入口或 v42 CLI 启动 Agent，管理跨设备同步的 Agent 列表与自定义命令。

vibe42.ai/zh-HANS/docs/agents/launching-agents

通过 Vibe42 启动 Agent，会在你电脑上创建一个一直在后台运行的终端窗口并启动程序，同时自动接入状态追踪与推送通知。本页介绍启动 Agent 的各个入口、Agent 列表管理、自定义命令规范，以及登录 shell 和终端隔离的工作方式。

三步新建流程与快捷入口

完整的新建终端流程依次选择电脑、选择 Workspace 或目录，并在第三步选择要启动的程序：

- 第一项固定为 **终端**（只打开 shell）。
- 下方是你的 Agent 列表，默认选中你上次用过的项，直接按回车即可启动。
- 在这一步你可以输入关键词搜索、返回上一步修改位置，或点击 **管理 Agent…** 管理列表。

当目标位置已经确定（或账号下只有一台电脑而跳过电脑选择）时，从以下快捷入口会直接进入第三步：

- Workspace 菜单：**在此 Workspace 新建终端**
- 文件树菜单：**在这里创建终端**
- 窗口菜单：**在此目录新建窗口**
- 全局搜索与终端列表的 **+** 按钮
- 创建 Workspace 时勾选：**创建完成后在工作区中打开终端**

> **说明**
>
> 进程由电脑上的 Vibe42 后台服务直接启动。即使你在手机上点击启动后立刻断网或锁屏，Agent 也会在电脑上正常运行。

管理 Agent 列表

打开设置中的 **Agent** 分区即可定制启动列表。默认列表包含 Claude Code、OpenAI Codex CLI、OMP、Pi Coding Agent、OpenCode、Grok Build、Devin CLI、Antigravity CLI 和 Cursor Agent；内置目录共提供约 30 个 Agent。

在 **Agent** 设置中，你可以：

- 从内置目录添加 Agent，或移除不常用的项
- 拖拽调整显示顺序
- 添加自定义名称与启动命令
- 随时恢复默认列表

你的 Agent 列表与上次选择的项会在账号下的所有已登录设备间自动同步。

自定义命令规则

在 **Agent** 设置中添加或编辑自定义项时，需要配置三项内容：

- **名称**：不超过 64 个字符。启动后新建窗口的名称会直接采用该 Agent 名称。
- **Agent 类型**：默认为 **根据命令识别**；你也可以手动指定某个已知类型，让包装引擎使用对应的状态识别规则。
- **命令**：单行“程序 + 参数”，参数会原样传给目标程序：

```
<program> <args>
```

启动命令需满足以下限制：

- 单行文本，总长度 ≤ 1024 字节
- 不能以 `-` 开头，不能包含控制字符
- 仅支持“程序 + 参数”形式，不支持 `&&`、`;`、重定向或管道（`|`）等 shell 组合语法

如果你在启动前需要执行多步准备或设置环境变量，请先在电脑上写成一个可执行脚本，再把脚本路径填为启动命令。

登录 shell、退出行为与隔离副本

启动命令会在你电脑的交互式登录 shell（`<shell> -l -i -c`，即平时打开本机终端时的完整命令行环境）中运行。你在 `.zshrc`、`.bashrc` 等启动文件里配置的 `PATH`、`nvm` 等环境都会生效，不需要在 Vibe42 里重复配置。

当 Agent 进程退出后，窗口不会关闭，而是回到普通命令行，方便你检查 `git status` 或运行测试。

如果电脑重启，后台服务再次启动时会重建这个窗口并再次运行同一条启动命令。支持按会话重新打开的 Agent 会重新打开上一次的对话（重启时中断的那一轮需要你再发一条消息让它接着做）；自己在 shell 里敲的命令则不会重新运行，详见会话与窗口管理。

如果你在开启了终端隔离的 Workspace 中启动 Agent，Vibe42 会先创建一份独立副本，再在该副本内启动 Agent，避免多个同时运行的 Agent 互相覆盖文件（详见终端隔离）。

通过 CLI 启动

你也可以使用 `v42` 命令行工具在终端或脚本中启动 Agent（详见 CLI 命令）：

```
v42 term new <computer> [--workspace <ws>] [--cwd <dir>] [--name <n>] [--agent <id>] [--command "<cmd>"] [--attach]
```

需要后台服务的包装引擎追踪 Agent 状态时，请加上 `--agent <id>`。这里的 id 是 Agent 目录里的标识，例如 `claude-code`、`codex`、`gemini`、`opencode`、`cursor`、`copilot`；传入未知 id 会报错并列出所有可用的 id。

电脑版本过旧时的提示

如果目标电脑上的 Vibe42 版本较旧、尚不支持直接启动 Agent，第三步列表中将只显示 **终端**，并提示：

```
Update Vibe42 on this computer to start agents from here.
```

此时通过 CLI 调用会返回 `launch_unsupported`。升级该电脑上的 Vibe42 后即可正常使用（参见更新与卸载）。

Agent 状态

了解五种 Agent 状态与三种输入请求细分、侧栏指示点、Agents 面板排序逻辑，以及哪些情况下不会显示状态。

vibe42.ai/zh-HANS/docs/agents/agent-status

当你在多个 Workspace 或多台电脑上同时运行 Coding Agent 时，Vibe42 会把不同工具的运行情况统一为五种状态，让你一眼看出哪些任务还在跑、哪些正停下来等你处理。阅读本页可了解各状态的含义、排序与已读复位规则，以及不显示状态的常见原因。

五种状态与侧栏指示点

带有活跃 Agent 状态的终端项会在侧栏显示一个机器人图标和对应的状态圆点：

状态	侧栏状态点	含义
空闲	不显示圆点	Agent 处于空闲状态，或上一轮 **已完成** 结果已被你打开查看过。

状态	侧栏状态点	含义
工作中	呼吸动画圆点	Agent 正在思考、修改文件或执行命令；若上报了进度，会同时显示百分比（`{{progress}}%`）。
已完成	绿色圆点	当前轮次已完成，有尚未查看的新结果。
需要处理	橙色圆点	Agent 暂停并等待你操作；若上报了进度也会显示 `{{progress}}%`，并细分为下方三种原因。
出错	红色圆点	Agent 遇到错误并停止。

当状态为 **需要处理** 时，若有具体分类信息，界面会显示为三种细分状态：

- **需要授权**：等待你批准工具调用、命令执行或文件写入。
- **需要回答**：向你提问，或等待你在多个方案之间做出选择。
- **需要登录**：需要登录或重新完成身份验证后才能继续。

Agent 面板与首页排序规则

手机、平板、浏览器与桌面 App 同时在线时，所有设备都会实时同步同一份状态，并在两处集中展示：

- **Agent** 标签页：按 Workspace 或电脑分组；组内按紧急程度排序：**出错** > **需要处理** > **已完成** > **工作中** > **空闲**，同状态内再按更新时间排序。
- **首页**（当前 Agents）：按状态分组展示所有正在追踪的 Agent，需要处理的暂停任务与刚完成的任务一目了然。

如果同一个分屏（窗口里的一个终端区域）通过 OSC 7501（程序向 Vibe42 报告状态的控制序列）上报了多条子任务记录，该分屏的汇总状态同样按 **需要处理**（`blocked`）> **出错** > **工作中** > **已完成** > **空闲** 的优先级决定。

看过后 已完成 变 空闲 的复位规则

绿色的 **已完成** 圆点相当于一条未读提醒：

- 当后台分屏进入 **已完成** 时，绿点会一直保留，直到你切过去查看。
- 一旦你打开该分屏，它的对外状态就会从 **已完成** 变为 **空闲**（表示已看过）。
- 如果这个分屏原本就开在你眼前，新出现的 **已完成** **不会**自动复位成 **空闲**，方便你看到当前轮次刚刚跑完。

状态由电脑上的 Vibe42 后台服务统一维护。你在手机上点开看过的 **已完成** 窗口，桌面端和网页端上的绿点也会同步消失。

状态来自哪里

不同 Agent 的状态通过三档机制接入（详见支持的 Coding Agent）：

1. **原生完整支持 OSC 7501**：Claude Code（≥ 2.1.295）与 Pi（≥ 1.1.0）直接上报状态、摘要与 **需要处理** 细分类型。
2. **部分自报 + 引擎补充**：Codex、OMP、Crush、Amp、Antigravity 与 Grok Build 自报部分信号，其余由后台服务的包装引擎补齐。

3. **包装引擎代报**：目录中的其余 Agent 由后台服务根据屏幕规则和本地会话记录识别并代报。

哪些情况看不到状态

如果某个窗口里的 Agent 正在运行，但侧栏没有机器人图标和状态点，通常属于以下四种情况：

- **在普通命令行里手动敲命令启动**：包装引擎只接管从新建终端流程或 `v42 term new --agent` 启动的进程；手动输入命令启动时，除非该 Agent 原生完整支持 OSC 7501，否则不会有状态和推送。
- **不是通过 Vibe42 打开的终端**：只有 Vibe42 创建的分屏才会由后台服务应答 OSC 7501 探测（`ESC ] 7501 ; ? ST`）。不是通过 Vibe42 打开的终端或分屏收不到应答，程序就不会上报状态。
- **升级前留下的旧分屏**：升级 Vibe42 之前就已经存在的旧分屏无法应答探测，在 Vibe42 中新建一个窗口即可解决。
- **启动超过 20 秒未识别到进程，或进程已退出**：如果启动脚本准备时间过长，启动后 20 秒内未出现目标 Agent 进程，包装引擎会放弃接管；当 Agent 退出回到普通命令行后，该分屏的状态记录也会清空。

通知

当 Agent 完成任务、等待确认或出错时通过手机、桌面、Web 或 Telegram 收到提醒，并了解通知文字如何经我们的服务器送到 Apple 与 Google。

`vibe42.ai/zh-HANS/docs/agents/notifications`

当 Coding Agent 完成当前轮次、停下来等待你确认或遇到报错时，以及终端或电脑发生关键事件时，Vibe42 会主动向你发送通知，让你离开电脑后也不必时刻盯着屏幕。阅读本页可了解支持的事件类型、各端投递行为、Telegram 绑定、按电脑独立配置以及去重折叠规则。

触发通知的事件类型

通知事件分为三大类，你可以在每台电脑的通知设置中按需单独开关：

类别	事件类型	触发时机
Agent 事件	**Agent 需要处理**、**Agent 已完成**、**Agent 出错**	默认均开启。当分屏（窗口里的一个终端区域）的汇总 Agent 状态**变为 需要处理**、**已完成** 或 **出错** 时各发一条。
终端事件	终端响铃、终端通知	程序输出标准终端响铃（BEL 字符）或终端通知序列时触发。
会话、设备与电脑事件	窗口或分屏关闭、设备断开、会话创建/关闭、电脑配对/解绑/上线/离线、频率过高折叠	窗口结构变化、设备连接变化、电脑上下线或触发高频折叠时发送。

Agent 通知的标题格式为 `状态：标题`（例如 `Done：Fix login bug`）；正文包含状态行与本轮摘要（`msg`），末尾附带三行上下文：Workspace 名（若不属于任何 Workspace 则显示目录路径）、电脑名与时间。

各端投递方式

- **手机推送（iPhone、iPad、Android）**：手机 App 不在后台常驻运行，也不保持后台长连接，不会额外耗电；所有推送都通过系统级推送服务（iOS 的 APNs 或 Android 的 FCM）送达。点击推送可直接打开 Vibe42 并跳转到对应电脑的目标窗口。
- **桌面 App（macOS、Linux）**：仅在桌面 App 运行期间接收系统通知；桌面 App 退出期间发生的事件不会在下次启动时补发。
- **Web 应用**：在浏览器打开 `app.vibe42.ai` 期间以页面内弹窗形式提示。
- **Telegram**：前往设置 **通知 → 消息通道**，通过二维码或绑定码即可绑定你的 Telegram 接收通知；随时可在同一页面解绑。

按电脑独立设置

通知策略按每台电脑分别配置，你可以让主力开发机的 Agent 提醒推送到手机，同时把测试服务器的通知限制在桌面端：

- **此电脑的通知**：开启或暂停该电脑的全部通知。
- **送达到**：分别勾选是否投递到 Web 与桌面 App、Telegram、手机推送。
- **事件类型**：逐项开关各类 Agent、终端、会话与电脑事件。

摘要等待、去重与高频折叠

为了避免重复提醒和消息刷屏，Vibe42 在发送前会做三项处理：

- **最多等待 15 秒收集摘要**：当状态刚切换到需通知的状态时，系统最多等待 15 秒以收集本轮摘要（`msg`），确保你收到的推送带有具体内容。
- **3 秒内响铃去重**：不少工具在上报状态的同时还会触发一次终端响铃。同一分屏通过 OSC 7501 上报状态后的 3 秒内，该分屏产生的终端响铃与终端通知会被自动合并去重。
- **高频自动折叠**：当通知频率达到约 20 条/分钟以上时，会自动折叠为摘要通知，防止循环报错或高频构建脚本刷屏。

通知文字是怎样送到你手机上的

通知的标题与正文是 Vibe42 里唯一一类以可读形式经过并保存在我们服务器上的内容。下面按先后顺序说明一条通知从生成到显示在手机上经过了谁：

1. **在被控电脑上先裁决、再生成**：触发通知的事件都发生在被控电脑上：Agent 状态变化（等待确认、完成、出错）、终端响铃、终端通知、窗口或分屏关闭、设备断开、会话创建或关闭、监视规则触发。电脑上的后台服务先按这台电脑的通知设置裁决：事件类型被关闭、或命中响铃/终端通知的节流，就在电脑上直接丢弃——不生成文字、不发送任何东西。只有通过裁决的事件才会被渲染成通知的标题和正文，使用的是你账号的界面语言，所以你收到的通知已经是你的语言。
2. **从被控电脑发到 Vibe42 服务器（传输加密，但不是端到端）**：渲染后的标题和正文通过后台服务与 Vibe42 控制服务器之间的加密控制会话上行；这条会话经 AP（接入点）转发，但终点是 Vibe42 的服务器。因为服务器必须能读到文字，才能把它交给 Apple、Google 或 Telegram，所以通知的标题与正文**不是**端到端加密——

这和终端、文件流量不同（那些只有你的设备和电脑能解密）。电脑离线或未配对时，事件直接丢弃，不会排队补发。

3. **Vibe42 服务器处理与入队**：
   - 把结尾的目录换成 Workspace 名（如果这个终端属于某个 Workspace），把电脑名换成你在 Vibe42 里给这台电脑起的名字。
   - 存一条通知记录（标题、正文、事件类型、电脑、时间、投递结果）作为你的通知历史，App 里能看到最近的通知。这条记录保存在 Vibe42 的服务器上，随账号删除一并删除。
   - 为每个目的地各建一个投递任务：每台已登录的手机（iPhone、iPad、Android 各算一台）、每个绑定的 Telegram、以及 Web/桌面通道。立刻尝试投递；失败按递增的间隔重试，最多 6 次；24 小时内仍未送达就作废；期间电脑被删除或那台手机已登出，则取消投递。
   - 推送的标题最多 200 个字符、正文最多 3500 个字符，超出的部分截断。
   - 服务器日志不写通知的标题与正文；审计日志只记身份、路由与管理元数据，**从不包含通知文字**（详见安全架构）。

4. **从 Vibe42 服务器发给 Apple 与 Google**：
   - **iPhone、iPad**：Vibe42 服务器向 Apple 的推送服务 APNs（`api.push.apple.com`）发一条 HTTPS 请求，用 Vibe42 的 APNs 密钥签名认证。请求里有：你这台设备的推送令牌（Apple 发给你手机上 Vibe42 App 的设备令牌，App 登录时注册到 Vibe42；它不是你的 Apple ID，也不含邮箱或手机号）、标题、正文（标准 `alert` 通知，带默认提示音），以及一个跳转信息 `vibexTarget`（电脑 ID、窗口 ID、分屏 ID、事件类型，只用于点按后打开正确的窗口）。
   - **Android**：向 Google 的 Firebase Cloud Messaging（FCM，`fcm.googleapis.com`）发一条 HTTPS 请求，用 Vibe42 的服务账号认证。请求里有：你这台设备的注册令牌、标题、正文（Android 上显示在 Vibe42 的事件通知渠道里），以及同样的 `vibexTarget`。
   - **Apple 与 Google 能读到什么**：这两条请求都走 TLS 加密的 HTTPS，但**文字在请求体里是明文**：Apple 和 Google 的服务器能读到通知的标题与正文，再转交到你的手机。Vibe42 没有在这之上再加一层只有手机能解的加密（iOS 端没有用通知服务扩展解密，Android 端也直接显示送达的内容）。Apple 与 Google 收到的就是：推送令牌、标题、正文、`vibexTarget`，没有你的账号信息，也没有终端或文件内容。它们之后如何处理，按各自的推送服务政策执行。

5. **在手机上显示**：系统按 Apple/Google 送达的内容显示通知，Vibe42 App 不需要在运行、也不保持后台连接；点按时 App 读取 `vibexTarget` 打开对应电脑与窗口。

通知里有什么、没有什么

- **标题和正文里有什么**：
  - Agent 事件：标题是“状态: 会话标题”，正文是状态行、本轮摘要（`msg`）或等待确认时的工具名与原因。
  - 终端通知：程序自己给出的标题与消息文本。终端响铃：终端标题、当前命令或“窗口 N 分屏 M”这样的位置，正文只是事件名。
  - 窗口或分屏关闭：被关闭的名字。监视规则：规则名与规则给出的摘要或匹配到的文字。
  - 结尾三行：终端所在目录、电脑名、发生时间。
- **标题和正文里没有什么**：屏幕上的其他输出、命令历史、按键、文件内容。只有程序主动放进状态上报或终端通知里的那几行文字。

Telegram、Web 与桌面 App

- **Telegram**：服务器把“标题 + 正文”通过 Telegram 的 Bot API（`api.telegram.org`，HTTPS）发到你绑定的聊天；Telegram 的服务器同样能读到文字。
- **Web 与桌面 App**：通过你已登录的、到 Vibe42 服务器的连接下发，只在在线时送达，不经过 Apple、Google 或 Telegram；但文字同样经过并保存在 Vibe42 服务器（进入通知历史）。

关闭通知后，没有任何一方会收到内容

- 在这台电脑的通知设置里关闭 **此电脑的通知**，或在 **事件类型** 里关掉某一类事件：裁决在被控电脑本机完成，被关闭的事件在电脑上直接丢弃——**不生成通知文字，不发往 Vibe42 服务器，因此也不会到 Apple、Google、Telegram 或你的任何设备，通知历史里也不会出现**。没有任何一方收到通知内容。
- 只在 **送达到** 里关掉手机推送（保留 Web/桌面或 Telegram）：文字仍会发到 Vibe42 服务器并进入通知历史，但不会发给 Apple 或 Google；关掉 Telegram 同理，不会发给 Telegram。
- 这些设置保存在你的账号里并同步到那台电脑，电脑按同步到的设置裁决。

排障

如果未收到预期推送，请参考通知故障排查检查系统通知权限、单机 **送达到** 开关与 Agent 启动方式。

OSC 7501 状态协议

OSC 7501 终端控制序列完整规范。了解探测握手、多记录表模型、汇总优先级与推送触发规则，让你的脚本或工具接入 Agent 面板。

vibe42.ai/zh-HANS/docs/agents/osc-7501

OSC 7501 是连接命令行程序与 Vibe42 **Agent** 面板、侧栏状态点及推送通知的终端控制序列协议（程序向终端打印的一串状态指令）。如果你正在开发 Coding Agent、维护团队内部命令行工具，或者想让自己写的长耗时构建脚本出现在 Agent 面板并在需要确认时推送到手机，可以按本规范直接接入。

完整语法与线格式

一条 OSC 7501 序列由冒号分隔的 `key=value` 键值对组成，以 `BEL`（`0x07`）或 `ST`（`ESC \`）结尾：

```
ESC ] 7501 ; key=value:key=value ... (BEL | ESC \)
state=<idle|working|done|blocked|error|clear>   必需；非法整条丢弃
id=<seg/seg/...>   记录路径，每段 [A-Za-z0-9_.+-]，≤8 层、≤128 字节；缺省为根记录
kind=<permission|question|auth>   仅 blocked 有效
progress=<0-100>   仅 working / blocked 有效
app=<可打印 ASCII，≤32 字节>   Agent 标识，例如 claude
title=<base64(UTF-8)>   解码后 ≤192 字节，不含控制字符
msg=<base64(UTF-8)>   解码后 ≤2048 字节，本轮摘要
探测：ESC ] 7501 ; ? ST   终端应答后程序才开始上报
```

解析与边界规则：

- **长度上限**：整条序列总长 ≤ 4096 字节。
- **键值容错**：键名只支持小写 ASCII；未知键会被忽略；单个非法的 `key=value` 对会被跳过，但若 `state` 缺失或取值非法，整条序列会直接丢弃。
- **序列剥离**：Vibe42 后台服务会在转发前剥离原始序列，客户端屏幕上不会显示任何控制字符。
- **不映射 OSC 9;4**：通用的 OSC `9;4` 终端进度序列不会映射为 OSC 7501 状态。

字段说明

字段	是否必需	取值约束与行为
`state`	必需	取值 `idle`、`working`、`done`、`blocked`、`error`、`clear`。其中 `blocked` 在界面显示为 **需要处理**。
`id`	可选	以 `/` 分隔的层级路径，每段仅限 `[A-Za-z0-9_.+-]`，最多 8 层、总长 ≤ 128 字节；省略时指向根记录。
`kind`	可选	仅 `state=blocked` 时有效：`permission` 对应 **需要授权**，`question` 对应 **需要回答**，`auth` 对应 **需要登录**。
`progress`	可选	`0` – `100` 的整数，仅 `state=working` 或 `state=blocked` 时有效，界面显示为 `{{progress}}%`。
`app`	可选	可打印 ASCII 组成的工具标识，≤ 32 字节（如 `claude`）。
`title`	可选	UTF-8 文本经 Base64 编码；解码后 ≤ 192 字节，不得含控制字符。
`msg`	可选	本轮摘要，UTF-8 文本经 Base64 编码；解码后 ≤ 2048 字节。

探测握手

规范接入的程序在开始发送状态前，应先输出探测序列：

```
ESC ] 7501 ; ? ST
```

收到终端返回的应答后，程序才开始上报 OSC 7501。原生 tmux 本身不会应答该探测；在通过 Vibe42 创建的分屏（窗口里的一个终端区域）中，电脑上的后台服务会代为应答（被包装引擎接管的进程也由引擎直接应答）。

> **说明**
>
> 你在 tmux 内手动切出的分屏以及升级 Vibe42 之前已有的旧分屏收不到探测应答，支持探测的程序会自动保持静默。

单分屏记录表模型与生命周期

每个分屏独立维护一张记录表，最多容纳 **256 条**记录，支持主任务与多个子任务用不同 `id` 路径同时上报：

- **同 `id` 整体替换**：向同一 `id` 发送新记录时，整条覆盖旧记录，不保留未重传的可选字段。
- **`clear` 清理**：带具体 `id` 发送 `state=clear` 会删除该记录及其所有子记录（`id/*`）；不带 `id` 的根 `clear` 会清空整张表。

- **复位与新提示符回收**：终端硬复位（RIS）会清空记录表；当 shell 输出 `OSC 133 ; A`（新提示符标记）时，表中所有 `working` 与 `blocked` 记录会被自动清除，避免程序异常退出后一直停在运行状态。

汇总优先级、已读复位与推送触发

- **汇总优先级**：当表中存在多条记录时，按 `blocked > error > working > done > idle` 取最高优先级作为该分屏的对外状态；同优先级下，根记录优先于子记录，其次取最近更新的一条。
- **已完成 复位规则**：当后台分屏汇总为 `done` 后，一旦用户打开查看该分屏，对外状态即变为 `idle`（表示已看过）；若该分屏一直处于打开状态，新产生的 `done` 不会自动复位。
- **推送触发**：仅当分屏的**汇总状态变为** `done`、`blocked` 或 `error` 时触发推送通知；系统最多等待 15 秒收集 `msg` 摘要，并在上报后的 3 秒内对同一分屏的普通终端响铃和终端通知自动去重。

Shell 示例

上报 `working` 进度并在完成后切换为 `done`：

```
printf '\e]7501;state=working:app=demo:progress=30:msg=%s\e\\' "$(printf 'Running tests' |
base64)"

# ... 执行构建或测试步骤 ...

printf '\e]7501;state=done:app=demo:title=%s:msg=%s\e\\' \
  "$(printf 'CI check' | base64)" \
  "$(printf 'All suites passed' | base64)"
```

在需要人工授权前上报 `blocked`（`kind=permission`）：

```
printf '\e]7501;state=blocked:kind=permission:app=demo:title=%s:msg=%s\e\\' \
  "$(printf 'Database migration' | base64)" \
  "$(printf 'Approve schema changes?' | base64)"
```

让 Agent 使用 Vibe42

把 install.md 交给 Agent 完成安装配对，用默认输出 JSON 的 v42 CLI 编排工作流，或通过 OSC 7501 上报自定义任务状态。

vibe42.ai/zh-HANS/docs/agents/for-agents

Vibe42 不仅方便你在手机或另一台电脑上操控 Coding Agent，也专门支持让 Agent 直接调用和阅读。当你想让正在运行的 Agent 替你装好 Vibe42、用 `v42` 命令行工具编排自动化任务、在隔离副本中安全协作，或是直接把文档发给 Agent 阅读时，可以参考本页。

1. 把 install.md 发给 Agent 完成安装

如果你手头的 Mac 或 Linux 机器上已经在运行某个 Coding Agent，只需把 `https://vibe42.ai/install.md` 发给它：

```
阅读 https://vibe42.ai/install.md 并在当前机器上安装 Vibe42。
```

`install.md` 为 Agent 提供了清晰的环境判断逻辑与执行步骤：Agent 会自动识别当前系统环境（macOS、Linux 桌面、无桌面 Linux 服务器或容器），选择合适的安装流程，跑完安装后把 8 位配对码或确认链接展示给你。

同时，`install.md` 明确要求 Agent 遵守两条安全规则：

- **不触碰密码**：Agent 不会向你索要账号密码、两步验证码或浏览器登录凭据，配对始终由你在已登录设备或 Console 上亲自批准。
- **一次只配一台**：每次只针对当前机器发起配对并等待你确认，不批量跨机器分发凭据。

完整操作步骤见指南让你的 Agent 安装 Vibe42。

2. 用 v42 CLI 编排自动化

`v42` 命令行工具专为非交互式自动化设计（详见 CLI 概览与 CLI 自动化）：

- **默认输出 JSON**：命令默认输出结构化 JSON，Agent 可以直接解析电脑信息、窗口标识与网络状态，不需要用正则表达式解析表格文本。
- **稳定的退出码（exit code）**：固定的退出码让脚本和 Agent 能准确区分网络波动、参数错误或配额限制。
- **`V42_AGENT_NAME` 审计标识**：由 Agent 调用 `v42` 时，可设置环境变量 `V42_AGENT_NAME`，在管理审计记录中标明操作来自哪个 Agent：

```
V42_AGENT_NAME=claude v42 term new my-mac --workspace api --agent claude-code
```

输出终端隔离副本的 SKILL.md

在开启了终端隔离的 Workspace 中同时运行多个 Agent 时，每个终端都在各自独立的副本里工作。运行以下命令可直接输出一份 `SKILL.md` 说明文档，教你的 Agent 理解隔离副本机制以及如何安全地检查和合并改动：

```
v42 copy skill
```

3. 让自研脚本通过 OSC 7501 上报状态

当你让 Agent 编写耗时较长的构建脚本、数据迁移工具或团队内部命令行工具时，可以让它按 OSC 7501 状态协议向终端输出控制序列（程序向 Vibe42 报告状态用的一串指令）。接入后，该脚本无需额外配置就会出现在 Vibe42 的 **Agent** 面板中，实时展示进度百分比，并在完成、报错或等待人工确认时向你的手机发送推送通知。

面向 Agent 阅读的文档格式

Vibe42 文档站提供标准的 `llms.txt` 索引，并且每个文档页面都提供对应的 `.md` 纯 Markdown 版本。你可以直接让 Agent 读取所需页面的 Markdown 原文，无需消耗上下文去解析网页 HTML 结构。

在 Vibe42 中使用 Claude Code

在 Vibe42 中运行 Claude Code，获得原生 OSC 7501 状态追踪、权限确认推送与手机端接续体验。

`vibe42.ai/zh-HANS/docs/agents/claude-code`

Claude Code 是 Anthropic 推出的终端 Coding Agent。本页介绍如何在 Vibe42 中启动 Claude Code、查看实时状态与通知，以及在手机上接手正在运行的会话。

在 Vibe42 中启动

Claude Code 排在 Vibe42 默认 Agent 列表的第一项。你可以在新建终端流程的第三步直接选中 **Claude Code** 回车启动，或通过命令行运行 `v42 term new <computer> --agent claude-code`。

Vibe42 会在电脑的交互登录 shell（`<shell> -l -i -c`）中直接运行你本机安装的程序，遵循不注入、不锁定原则——不修改 `~/.claude`、不写入 hook，也不代理你的账号凭据。如果需要传入自定义参数或调用外层脚本，请在设置 **Agent** 中添加单行 `<program>` 命令。Claude Code 退出后，窗口会保留并回到普通 shell。

状态支持与表现

Claude Code 2.1.295 及以上版本已原生支持 OSC 7501（程序向 Vibe42 报告状态用的一串控制序列），无需经过包装引擎；更旧版本则自动由 Vibe42 后台服务的包装引擎代报。

在侧栏与 **Agent** 面板中，你可以看到五种 Agent 状态（**空闲**、**工作中**、**已完成**、**需要处理**、**出错**）以及本轮摘要。由于是原生自报，≥ 2.1.295 的版本即使由你在 Vibe42 创建的普通终端分屏里手动输入命令运行，也能正常上报状态。进入 **需要处理** 时会区分三种原因：

- **需要授权**：等待你批准工具调用。
- **需要回答**：等待你回答问题。
- **需要登录**：需要登录。

通知

每当分屏状态变为 **已完成**、**需要处理** 或 **出错** 时，Vibe42 都会将状态与本轮摘要通过手机推送（APNs / FCM）、运行中的桌面 App、Web 应用或已绑定的 Telegram 发送给你（详见通知）。点击通知即可打开该 Claude Code 窗口。

从手机接续

会话一直在电脑上运行，关掉 App 也不会停。在工位发起任务后离开电脑，你可以随时在 iPhone、iPad 或 Android 上继续操作（参考出门前启动 Claude Code与移动端输入）：

- 收到权限确认提示或提问时，用可拖动方向键（D-pad）、回车、Escape 或自定义快捷按钮直接批准和选择。
- 较长的多行提示词先在底部命令栏（**发送到终端**）写好核对，再一次性发送。
- 用手机拍照或从相册选图上传到电脑，自动将本机文件路径插入输入框。

已知限制

- 你需要先在电脑上自行安装并登录 Claude Code；Vibe42 显示的是原生的终端界面，而不是单独改写的网页聊天框。
- 不是通过 Vibe42 打开的终端，以及升级 Vibe42 前遗留的旧分屏不会应答 OSC 7501 探测，因此不会显示状态。
- 低于 2.1.295 的版本依赖包装引擎代报，只有在通过新建终端流程或 `v42 term new --agent` 启动时才会显示状态。

在 Vibe42 中使用 OpenAI Codex CLI

在 Vibe42 中运行 OpenAI Codex CLI，通过部分自报加包装引擎补充实现状态追踪、推送通知与手机端远程接续。

vibe42.ai/zh-HANS/docs/agents/codex

OpenAI Codex CLI 是 OpenAI 开源的终端 Coding Agent。本页介绍如何在 Vibe42 中启动 OpenAI Codex CLI、它的混合状态上报机制如何工作，以及如何在手机上接手运行中的会话。

在 Vibe42 中启动

OpenAI Codex CLI 预置在 Vibe42 的默认 Agent 列表中。你可以在新建终端流程的第三步直接选中 **OpenAI Codex CLI** 启动，或通过命令行执行 `v42 term new <computer> --agent codex`。

Vibe42 会在电脑的交互登录 shell（`<shell> -l -i -c`）中直接启动你本机安装的程序，遵循不注入、不锁定原则——不会改动任何本地配置，也不会代理你的 OpenAI API 请求或密钥。如需添加启动参数或外层脚本，请在设置 **Agent** 中配置单行 `<program>` 命令。在开启了终端隔离的 Workspace 中，它会在新建的独立副本内运行；进程退出后窗口保留并回到普通 shell。

状态支持与表现

OpenAI Codex CLI 采用**部分自报 + 引擎补充**机制：CLI 自身上报部分状态信号，Vibe42 后台服务的包装引擎结合屏幕输出规则与本地会话记录补齐其余状态，并统一汇总为 OSC 7501（程序向 Vibe42 报告状态用的一串控制序列）状态。

在侧栏和 **Agent** 面板中，你可以查看 **空闲**、**工作中**、**已完成**、**需要处理** 与 **出错** 五种 Agent 状态。停在 **需要处理** 时，细分标签对应以下场景：

- **需要授权**：等待你批准工具调用。
- **需要回答**：等待你回答问题。
- **需要登录**：需要登录。

通知

每当分屏状态变为 **已完成**、**需要处理** 或 **出错** 时，Vibe42 会通过系统级 APNs / FCM 手机推送（零后台耗电）、运行中的桌面 App、Web 应用或已绑定的 Telegram 发送提醒（详见通知）。点击通知即可跳进对应的终端窗口。

从手机接续

会话一直在电脑上运行，关掉 App 也不会停。离开工位后，你可以随时在 iPhone、iPad 或 Android 上继续操作（参见移动端输入）：

- 遇到工具授权或选项菜单时，用可拖动方向键（D-pad）、回车、Escape 与自定义快捷按钮快速确认。
- 在底部命令栏（**发送到终端**）先写好多行追加指令，核对无误后再发送。
- 拍照或从手机相册选择截图上传到电脑，自动将本机路径插入输入框。

已知限制

- 你需要先在电脑上自行安装并完成 OpenAI Codex CLI 的认证；Vibe42 显示的是原生的终端界面，而不是单独拆出的聊天框视图。
- 请从新建终端流程或 `v42 term new --agent` 启动以挂载包装引擎。若在普通 shell 中手动输入命令启动，将缺失包装引擎补充的状态；不是通过 Vibe42 打开的终端，或升级前遗留的旧分屏也无法应答 OSC 7501 探测。
- 如果启动后 20 秒内未检测到对应 Agent 进程，包装引擎会停止包装该分屏。

在 Vibe42 中使用 OMP

在 Vibe42 中运行 OMP，通过部分自报加包装引擎补充实现状态追踪、推送通知与手机端远程接续。

`vibe42.ai/zh-HANS/docs/agents/omp`

oh-my-pi（**OMP**）是以 MIT 许可证开源的终端 Coding Agent，源自 Mario Zechner 的 Pi 的分支版本，自述为“coding agent with the IDE wired in”（把 LSP、调试器等 IDE 能力接进终端 Agent）。本页介绍如何在 Vibe42 中启动 OMP、它的混合状态上报机制如何工作，以及如何在手机上接手运行中的会话。

在 Vibe42 中启动

OMP 预置在 Vibe42 的默认 Agent 列表中。你可以在新建终端流程的第三步直接选中 **OMP** 启动（运行 `omp` 命令），或通过命令行执行 `v42 term new <computer> --agent omp`。

Vibe42 会在电脑的交互登录 shell（`<shell> -l -i -c`）中直接启动你本机安装的程序，遵循不注入、不锁定原则——不会改动它的配置文件，也不会代理你的 API 请求或凭据。如需添加启动参数或外层脚本，请在设置 **Agent** 中配置单行 `<program>` 命令。在开启了终端隔离的 Workspace 中，它会在新建的独立副本内运行；进程退出后窗口保留并回到普通 shell。

状态支持与表现

OMP 采用**部分自报 + 引擎补充**机制：CLI 自身上报部分状态信号，Vibe42 后台服务的包装引擎结合屏幕输出规则与本地会话记录补齐其余状态，并统一汇总为 OSC 7501（程序向 Vibe42 报告状态用的一串控制序列）状态。

在侧栏和 **Agent** 面板中，你可以查看 **空闲**、**工作中**、**已完成**、**需要处理** 与 **出错** 五种 Agent 状态。停在 **需要处理** 时，细分标签对应以下场景：

- **需要授权**：等待你批准工具调用。
- **需要回答**：等待你回答问题。
- **需要登录**：需要登录。

通知

每当分屏状态变为 **已完成**、**需要处理** 或 **出错** 时，Vibe42 会通过系统级 APNs / FCM 手机推送（零后台耗电）、运行中的桌面 App、Web 应用或已绑定的 Telegram 发送提醒（详见通知）。点击通知即可跳进对应的终端窗口。

从手机接续

会话一直在电脑上运行，关掉 App 也不会停。离开工位后，你可以随时在 iPhone、iPad 或 Android 上继续操作（参见移动端输入）：

- 遇到工具授权或选项菜单时，用可拖动方向键（D-pad）、回车、Escape 与自定义快捷按钮快速确认。
- 在底部命令栏（**发送到终端**）先写好多行追加指令，核对无误后再发送。
- 拍照或从手机相册选择截图上传到电脑，自动将本机路径插入输入框。

已知限制

- 你需要先在电脑上自行安装并配置好 OMP；Vibe42 显示的是原生的终端界面，而不是单独拆出的聊天框视图。
- 请从新建终端流程或 `v42 term new --agent` 启动以挂载包装引擎。若在普通 shell 中手动输入 `omp` 命令启动，将缺失包装引擎补充的状态；不是通过 Vibe42 打开的终端，或升级前遗留的旧分屏也无法应答 OSC 7501 探测。
- 如果启动后 20 秒内未检测到对应 Agent 进程，包装引擎会停止包装该分屏。

在 Vibe42 中使用 Pi Coding Agent

在 Vibe42 中运行 Pi Coding Agent，获得原生 OSC 7501 状态追踪、权限确认推送与手机端接续体验。

vibe42.ai/zh-HANS/docs/agents/pi

Pi（**Pi Coding Agent**）是由 Mario Zechner 开发的开源终端 Coding Agent，自述为“a minimal agent harness”，强调极简并可按工作流扩展。本页介绍如何在 Vibe42 中启动 Pi、查看实时状态与通知，以及在手机上接手正在运行的会话。

在 Vibe42 中启动

Pi Coding Agent 预置在 Vibe42 的默认 Agent 列表中。你可以在新建终端流程的第三步直接选中 **Pi Coding Agent** 回车启动（运行 `pi` 命令），或通过命令行运行 `v42 term new <computer> --agent pi`。

Vibe42 会在电脑的交互登录 shell（`<shell> -l -i -c`）中直接运行你本机安装的程序，遵循不注入、不锁定原则——不会改动它的配置文件、不写入 hook，也不代理你的 API 请求或凭据。如果需要传入自定义参数或调用外层脚本，请在设置 **Agent** 中添加单行 `<program>` 命令。Pi 退出后，窗口会保留并回到普通 shell。

状态支持与表现

Pi 1.1.0 及以上版本已原生支持 OSC 7501（程序向 Vibe42 报告状态用的一串控制序列），自己输出完整状态、进度、本轮摘要与暂停原因，无需经过包装引擎；更旧版本则自动回落到 Vibe42 后台服务的包装引擎。

在侧栏与 **Agent** 面板中，你可以看到五种 Agent 状态（**空闲**、**工作中**、**已完成**、**需要处理**、**出错**）以及本轮摘要。由于是原生自报，≥ 1.1.0 的版本即使由你在 Vibe42 创建的任何终端分屏里手动输入 `pi` 运行，也能正常显示状态。进入 **需要处理** 时会区分三种原因：

- **需要授权**：等待你批准工具调用。
- **需要回答**：等待你回答问题。
- **需要登录**：需要登录。

通知

每当分屏状态变为 **已完成**、**需要处理** 或 **出错** 时，Vibe42 都会将状态与本轮摘要通过手机推送（APNs / FCM）、运行中的桌面 App、Web 应用或已绑定的 Telegram 发送给你（详见通知）。点击通知即可打开该 Pi 窗口。

从手机接续

会话一直在电脑上运行，关掉 App 也不会停。在工位发起任务后离开电脑，你可以随时在 iPhone、iPad 或 Android 上继续操作（参见移动端输入）：

- 收到权限确认提示或提问时，用可拖动方向键（D-pad）、回车、Escape 或自定义快捷按钮直接批准和选择。
- 较长的多行提示词先在底部命令栏（**发送到终端**）写好核对，再一次性发送。
- 用手机拍照或从相册选图上传到电脑，自动将本机文件路径插入输入框。

已知限制

- 你需要先在电脑上自行安装并配置好 Pi Coding Agent；Vibe42 显示的是原生的终端界面，而不是单独改写的网页聊天框。
- 不是通过 Vibe42 打开的终端，以及升级 Vibe42 前遗留的旧分屏不会应答 OSC 7501 探测，因此不会显示状态。
- 低于 1.1.0 的版本依赖包装引擎代报，只有在通过新建终端流程或 `v42 term new --agent` 启动时才会显示状态。

在 Vibe42 中使用 OpenCode

在 Vibe42 中运行 OpenCode，通过后台服务的包装引擎获得状态面板、手机推送通知与移动端 TUI 操控体验。

vibe42.ai/zh-HANS/docs/agents/opencode

OpenCode 是一款面向终端打造的开源 Coding Agent。本页介绍如何在 Vibe42 中启动 OpenCode、包装引擎如何追踪它的运行状态与发送通知，以及在手机上操控其终端界面的要点。

在 Vibe42 中启动

OpenCode 位于 Vibe42 的默认 Agent 列表中。你可以在新建终端流程的第三步直接选择 **OpenCode** 启动，或使用命令行执行 `v42 term new <computer> --agent opencode`。

Vibe42 会在电脑的交互登录 shell（`<shell> -l -i -c`）中直接启动你本机安装的程序，遵循不注入、不锁定原则——不会改动 OpenCode 的本地配置文件、不注入插件，也不经手你配置的模型服务商 API key。如需附带启动参数或外层脚本，请在设置 **Agent** 中配置单行 `<program>` 命令。若所在 Workspace 开启了终端隔离，OpenCode 会在新建的独立副本中运行；退出程序后窗口保留并回到普通 shell。

状态支持与表现

OpenCode 的状态由 **Vibe42 后台服务的包装引擎代报**。从 Vibe42 启动后，包装引擎会根据屏幕输出规则和会话记录识别当前运行阶段并上报 Agent 状态。

在侧栏与 **Agent** 面板中，你可以实时查看 **空闲**、**工作中**、**已完成**、**需要处理** 和 **出错** 状态（具体细分展示以 App 内显示为准）。停在 **需要处理** 时，三种细分标签对应以下场景：

- **需要授权**：等待你批准工具调用。
- **需要回答**：等待你回答问题。
- **需要登录**：需要登录。

通知

每当分屏状态变为 **已完成**、**需要处理** 或 **出错** 时，Vibe42 都会通过手机推送（APNs / FCM 零后台耗电）、运行中的桌面 App、Web 应用或已绑定的 Telegram 通知你（详见通知）。点击推送即可打开该 OpenCode 窗口。

从手机接续

OpenCode 会话一直在电脑上运行，关掉 App 也不会停。你可以随时在 iPhone、iPad 或 Android 上打开同一 TUI（在终端里画交互界面的程序）继续操作（参见移动端输入）：

- 使用可拖动方向键（D-pad）、修饰键、Escape 以及自定义快捷按钮操作 TUI 弹窗和权限批准提示。
- 在底部命令栏（**发送到终端**）先写好多行描述，核对后再发送到终端。
- 用手机拍照或从相册上传截图到电脑，自动将本机文件路径插入当前输入位置。

已知限制

- 你需要先在电脑上自行安装 OpenCode 并配置好模型凭据；Vibe42 显示的是原生的终端界面，而不是单独拆出的聊天框视图。
- 状态追踪与 Agent 推送依赖包装引擎，必须从新建终端流程或 `v42 term new --agent` 启动；在普通 shell 里手动输入命令启动时仍可正常使用 OpenCode，但不会显示状态点或触发 Agent 推送。
- 如果启动后 20 秒内未检测到 OpenCode 进程，包装引擎会停止包装该分屏。

在 Vibe42 中使用 Grok Build

在 Vibe42 中运行 Grok Build，通过部分自报加包装引擎补充实现状态追踪、推送通知与手机端远程接续。

vibe42.ai/zh-HANS/docs/agents/grok-build

Grok Build 是 xAI 的终端 Coding Agent，自述为“a powerful and extensible coding agent”，在项目目录里运行 `grok` 即可进入交互式终端界面。本页介绍如何在 Vibe42 中启动 Grok Build、它的混合状态上报机制如何工作，以及如何在手机上接手运行中的会话。

在 Vibe42 中启动

Grok Build 预置在 Vibe42 的默认 Agent 列表中。你可以在新建终端流程的第三步直接选中 **Grok Build** 启动（运行 `grok` 命令），或通过命令行执行 `v42 term new <computer> --agent grok-build`。

Vibe42 会在电脑的交互登录 shell（`<shell> -l -i -c`）中直接启动你本机安装的程序，遵循不注入、不锁定原则——不会改动它的配置文件，也不会代理你的 API 请求或凭据。如需添加启动参数或外层脚本，请在设置 **Agent** 中配置单行 `<program>` 命令。在开启了终端隔离的 Workspace 中，它会在新建的独立副本内运行；进程退出后窗口保留并回到普通 shell。

状态支持与表现

Grok Build 采用**部分自报 + 引擎补充**机制：CLI 自身上报部分状态信号，Vibe42 后台服务的包装引擎结合屏幕输出规则与本地会话记录补齐其余状态，并统一汇总为 OSC 7501（程序向 Vibe42 报告状态用的一串控制序列）状态。

在侧栏和 **Agent** 面板中，你可以查看 **空闲**、**工作中**、**已完成**、**需要处理** 与 **出错** 五种 Agent 状态。停在 **需要处理** 时，细分标签对应以下场景：

- **需要授权**：等待你批准工具调用。
- **需要回答**：等待你回答问题。
- **需要登录**：需要登录。

通知

每当分屏状态变为 **已完成**、**需要处理** 或 **出错** 时，Vibe42 会通过系统级 APNs / FCM 手机推送（零后台耗电）、运行中的桌面 App、Web 应用或已绑定的 Telegram 发送提醒（详见通知）。点击通知即可跳进对应的终端窗口。

从手机接续

会话一直在电脑上运行，关掉 App 也不会停。离开工位后，你可以随时在 iPhone、iPad 或 Android 上继续操作（参见移动端输入）：

- 遇到工具授权或选项菜单时，用可拖动方向键（D-pad）、回车、Escape 与自定义快捷按钮快速确认。
- 在底部命令栏（**发送到终端**）先写好多行追加指令，核对无误后再发送。
- 拍照或从手机相册选择截图上传到电脑，自动将本机路径插入输入框。

已知限制

- 你需要先在电脑上自行安装并完成 Grok Build 的登录认证；Vibe42 显示的是原生的终端界面，而不是单独拆出的聊天框视图。
- 请从新建终端流程或 `v42 term new --agent` 启动以挂载包装引擎。若在普通 shell 中手动输入 `grok` 命令启动，将缺失包装引擎补充的状态；不是通过 Vibe42 打开的终端，或升级前遗留的旧分屏也无法应答 OSC 7501 探测。
- 如果启动后 20 秒内未检测到对应 Agent 进程，包装引擎会停止包装该分屏。

在 Vibe42 中使用 Devin CLI

在 Vibe42 中运行 Devin CLI，通过后台服务的包装引擎获得状态面板、手机推送通知与移动端 TUI 操控体验。

vibe42.ai/zh-HANS/docs/agents/devin

Devin CLI 是 Cognition 的本地命令行 Coding Agent，自述为“a local command-line coding agent with deep Devin Cloud integration”，在项目目录里运行 `devin` 即可开始交互会话。本页介绍如何在 Vibe42 中启动 Devin CLI、包装引擎如何追踪它的运行状态与发送通知，以及在手机上操控其终端界面的要点。

在 Vibe42 中启动

Devin CLI 位于 Vibe42 的默认 Agent 列表中。你可以在新建终端流程的第三步直接选择 **Devin CLI** 启动（运行 `devin` 命令），或使用命令行执行 `v42 term new <computer> --agent devin`。

Vibe42 会在电脑的交互登录 shell（`<shell> -l -i -c`）中直接启动你本机安装的程序，遵循不注入、不锁定原则——不会改动它的配置文件、不注入插件，也不会代理你的 API 请求或凭据。如需附带启动参数或外层脚本，请在设置 **Agent** 中配置单行 `<program>` 命令。若所在 Workspace 开启了终端隔离，Devin CLI 会在新建的独立副本中运行；退出程序后窗口保留并回到普通 shell。

状态支持与表现

Devin CLI 的状态由 **Vibe42 后台服务的包装引擎代报**。从 Vibe42 启动后，包装引擎会根据屏幕输出规则和会话记录识别当前运行阶段并上报 Agent 状态。

在侧栏与 **Agent** 面板中，你可以实时查看 **空闲**、**工作中**、**已完成**、**需要处理** 和 **出错** 状态（具体细分展示以 App 内显示为准）。停在 **需要处理** 时，三种细分标签对应以下场景：

- **需要授权**：等待你批准工具调用。
- **需要回答**：等待你回答问题。
- **需要登录**：需要登录。

通知

每当分屏状态变为 **已完成**、**需要处理** 或 **出错** 时，Vibe42 都会通过手机推送（APNs / FCM 零后台耗电）、运行中的桌面 App、Web 应用或已绑定的 Telegram 通知你（详见通知）。点击推送即可打开该 Devin CLI 窗口。

从手机接续

Devin CLI 会话一直在电脑上运行，关掉 App 也不会停。你可以随时在 iPhone、iPad 或 Android 上打开同一 TUI（在终端里画交互界面的程序）继续操作（参见移动端输入）：

- 使用可拖动方向键（D-pad）、修饰键、Escape 以及自定义快捷按钮操作 TUI 弹窗和权限批准提示。
- 在底部命令栏（**发送到终端**）先写好多行描述，核对后再发送到终端。
- 用手机拍照或从相册上传截图到电脑，自动将本机文件路径插入当前输入位置。

已知限制

- 你需要先在电脑上安装 Devin CLI 并登录你的 Devin 账号；Vibe42 显示的是原生的终端界面，而不是单独拆出的聊天框视图。
- 状态追踪与 Agent 推送依赖包装引擎，必须从新建终端流程或 `v42 term new --agent` 启动；在普通 shell 里手动输入 `devin` 命令启动时仍可正常使用 Devin CLI，但不会显示状态点或触发 Agent 推送。
- 如果启动后 20 秒内未检测到 Devin CLI 进程，包装引擎会停止包装该分屏。

在 Vibe42 中使用 Antigravity CLI

在 Vibe42 中运行 Antigravity CLI，通过部分自报加包装引擎补充实现状态追踪、推送通知与手机端远程接续。

vibe42.ai/zh-HANS/docs/agents/antigravity

Antigravity CLI 是 Google 的终端 Coding Agent，是 Antigravity 的命令行形态，命令为 `agy`。本页介绍如何在 Vibe42 中启动 Antigravity CLI、它的混合状态上报机制如何工作，以及如何在手机上接手运行中的会话。

在 Vibe42 中启动

Antigravity CLI 预置在 Vibe42 的默认 Agent 列表中。你可以在新建终端流程的第三步直接选中 **Antigravity CLI** 启动（运行 `agy` 命令），或通过命令行执行 `v42 term new <computer> --agent antigravity`。

Vibe42 会在电脑的交互登录 shell（`<shell> -l -i -c`）中直接启动你本机安装的程序，遵循不注入、不锁定原则——不会改动它的配置文件，也不会代理你的 API 请求或凭据。如需添加启动参数或外层脚本，请在设置 **Agent** 中配置单行 `<program>` 命令。在开启了终端隔离的 Workspace 中，它会在新建的独立副本内运行；进程退出后窗口保留并回到普通 shell。

状态支持与表现

Antigravity CLI 采用**部分自报 + 引擎补充**机制：CLI 自身上报部分状态信号，Vibe42 后台服务的包装引擎结合屏幕输出规则与本地会话记录补齐其余状态，并统一汇总为 OSC 7501（程序向 Vibe42 报告状态用的一串控制序列）状态。

在侧栏和 **Agent** 面板中，你可以查看 **空闲**、**工作中**、**已完成**、**需要处理** 与 **出错** 五种 Agent 状态。停在 **需要处理** 时，细分标签对应以下场景：

- **需要授权**：等待你批准工具调用。
- **需要回答**：等待你回答问题。
- **需要登录**：需要登录。

通知

每当分屏状态变为 **已完成**、**需要处理** 或 **出错** 时，Vibe42 会通过系统级 APNs / FCM 手机推送（零后台耗电）、运行中的桌面 App、Web 应用或已绑定的 Telegram 发送提醒（详见通知）。点击通知即可跳进对应的终端窗口。

从手机接续

会话一直在电脑上运行，关掉 App 也不会停。离开工位后，你可以随时在 iPhone、iPad 或 Android 上继续操作（参见移动端输入）：

- 遇到工具授权或选项菜单时，用可拖动方向键（D-pad）、回车、Escape 与自定义快捷按钮快速确认。
- 在底部命令栏（**发送到终端**）先写好多行追加指令，核对无误后再发送。
- 拍照或从手机相册选择截图上传到电脑，自动将本机路径插入输入框。

已知限制

- 你需要先在电脑上自行安装并完成 Antigravity CLI 的登录认证；Vibe42 显示的是原生的终端界面，而不是单独拆出的聊天框视图。
- 请从新建终端流程或 `v42 term new --agent` 启动以挂载包装引擎。若在普通 shell 中手动输入 `agy` 命令启动，将缺失包装引擎补充的状态；不是通过 Vibe42 打开的终端，或升级前遗留的旧分屏也无法应答 OSC 7501 探测。
- 如果启动后 20 秒内未检测到对应 Agent 进程，包装引擎会停止包装该分屏。

在 Vibe42 中使用 Cursor Agent

在 Vibe42 中运行命令行版 Cursor Agent，借助后台服务的包装引擎获取状态指示、手机推送提醒与远程接续能力。

vibe42.ai/zh-HANS/docs/agents/cursor-agent

Cursor Agent 将 Cursor 的 Coding Agent 工作流带到了命令行终端中。本页介绍如何在 Vibe42 中启动 Cursor Agent、通过包装引擎追踪运行状态，以及在手机上响应确认提示与接续会话。

在 Vibe42 中启动

Cursor Agent 预置在 Vibe42 的默认 Agent 列表中。你可以在新建终端流程的第三步选中 **Cursor Agent** 直接启动，或通过命令行执行 `v42 term new <computer> --agent cursor`。

Vibe42 会在电脑的交互登录 shell（`<shell> -l -i -c`）中直接运行你本机安装的命令，遵循不注入、不锁定原则——不改动你的 Cursor 本地配置、不安装额外扩展，也不代理你的 Cursor 账号请求。如果需要传入自定义启动参数或调用外层脚本，请在设置 **Agent** 中配置单行 `<program>` 命令。Cursor Agent 退出后，窗口会保留在普通 shell 下，方便你直接检查代码变更。

状态支持与表现

Cursor Agent 的运行状态由 **Vibe42 后台服务的包装引擎代报**。通过 Vibe42 启动后，包装引擎会根据终端屏幕规则与会话记录判断状态并同步到你的所有设备。

侧栏和 **Agent** 面板会展示 **空闲**、**工作中**、**已完成**、**需要处理** 与 **出错** 五种 Agent 状态（具体细分展示以 App 内显示为准）。处于 **需要处理** 时，三种细分标签对应以下场景：

- **需要授权**：等待你批准工具调用。
- **需要回答**：等待你回答问题。
- **需要登录**：需要登录。

通知

每当分屏状态变为 **已完成**、**需要处理** 或 **出错** 时，Vibe42 会向你的手机（经 APNs / FCM）、运行中的桌面 App、Web 应用或已绑定的 Telegram 发送提醒（详见通知）。点击推送可直接打开该 Cursor Agent 窗口。

从手机接续

会话一直在电脑上运行，关掉 App 也不会停。离开桌面后，你可以随时在 iPhone、iPad 或 Android 上继续操作（参见移动端输入）：

- 遇到工具调用确认或选项提示时，通过可拖动方向键（D-pad）、回车、Escape 与跨设备同步的快捷按钮快速操作。
- 在底部命令栏（**发送到终端**）先写好多行指令，确认无误后再发送。
- 用手机拍照或选图上传到电脑，自动在终端里插入该文件的本机路径。

已知限制

- 你需要先在电脑上自行安装并登录 Cursor Agent；Vibe42 显示的是原生的终端界面，不提供单独拆出的聊天框视图。
- 由于状态由包装引擎代报，必须通过新建终端流程或 `v42 term new --agent` 启动才会显示状态并触发 Agent 推送；在普通 shell 里手动输入命令启动不会启用包装引擎。
- 若启动后 20 秒内未检测到 Cursor Agent 进程，包装引擎会停止包装该分屏。

05 Workspace 与文件

Workspace

把某台电脑上的目录绑定为 Workspace，让新终端自动进入项目目录、实时浏览文件树，并在 Team 内按项目整理终端会话。

vibe42.ai/zh-HANS/docs/workspaces/workspaces

Workspace 把某台电脑上的一个本地目录（从根目录写起的绝对路径）绑定为侧栏里的项目入口。当你希望新建终端自动进入项目目录、在终端旁边的右栏浏览实时目录树，或者和 Team 成员按项目整理多个 Agent 会话时，可以用它来管理。

Workspace 是什么、不是什么

一个 Workspace 只是一条目录记录：**某台电脑 + 一个目录绝对路径 + 一个名称**。

- **直接读取电脑磁盘**：在侧栏 Workspace 下点 **文件** 打开目录树时，直接从那台电脑上的 Vibe42 后台服务读取。
- **不上传、不存储、不同步文件**：Vibe42 服务端只保存 Workspace 的记录信息（名称、侧栏排序和归档状态），不保存项目源码、目录快照或隔离副本。所有文件都留在你自己的电脑上。
- **同一路径可建多个 Workspace**：同一台电脑上的同一个绝对路径可以创建多个 Workspace，方便你按不同任务拆分工作流。

侧栏标签页与快速启动

侧栏顶部有三个标签页：**Agent**、**Workspace** 和 **最近**。每台设备会单独记住你上次选中的标签页。

点击标签页行末尾的 **+**，会打开 **新终端/打开App** 窗口。里面的目标按 **Workspace**、**电脑** 和 **远程桌面** 分组，支持即时搜索，可以在一个地方启动普通终端、Agent 或远程桌面。

在 **Workspace** 标签页下展开一个 Workspace，第一行是 **文件**，下面是它的终端；点 **文件** 会在右栏打开以这个 Workspace 文件夹为根的文件夹 Tab，侧栏里不再显示目录树；电脑离线时这一行是灰色的。详见文件。

创建 Workspace 的三个步骤

按下面三步创建 Workspace：

1. 在 **选择电脑** 中选择目录所在的电脑。
2. 在 **文件位置** 中浏览目录，或直接输入该电脑上的目录绝对路径。
3. 在 **名称与启动** 中填写名称；如果想建好后马上打开终端，保持勾选 **创建完成后在工作区中打开终端**。

Workspace 菜单与归档

打开侧栏里任意 Workspace 的菜单，可以执行以下操作：

菜单项	说明
在此 Workspace 新建终端	以该 Workspace 目录为起始目录新建一个终端窗口
在文件页打开	在完整的文件视图中打开该目录
副本…	查看和管理隔离终端副本与改动历史（见终端隔离）
编辑…	修改名称、路径，或开启终端隔离与压缩与去重等目录选项
上移 / 下移	在侧栏中上下调整该 Workspace 的顺序
归档	把该 Workspace 移出侧栏，不改动电脑磁盘上的任何文件

> 说明
>
> 点击 **归档** 时，界面会提示：**归档后该 Workspace 从侧栏移除，其中的终端回到 Recents；文件不受影响。**。归档后，原本属于该 Workspace 的运行中终端会回到 **最近** 继续运行，详见会话工作原理。

终端归属、工作目录与 最近

终端与 Workspace 之间的关系遵循三条规则：

- **初始工作目录（`cwd`，即命令执行时所在的当前目录）**：在 Workspace 内新建的终端默认以该目录为 `cwd`；如果开启了终端隔离，则是新建副本中的对应路径。在这个 Workspace 的文件夹 Tab 里右键点一个子文件夹并选择 **在这里创建终端**，新终端会以该子目录为 `cwd`，同时归属这个 Workspace。
- **移动已有终端**：运行中的终端可以在**同一台电脑**的不同 Workspace 之间移动，也可以移出 Workspace。移动只改变侧栏里的分组，不会改变运行中 shell 的 `cwd`。
- **与 最近 的关系**：不在任何 Workspace 内创建的终端，以及移出 Workspace 或随 Workspace 归档留下的运行中终端，都会出现在 **最近** 标签页下。

Team 共享与角色权限

Workspace 归属于当前 Team，默认对 Team 内所有成员可见且可直接使用：

- **所有者** 和 **管理员**：可以创建、修改、调整排序和归档 Workspace。
- **成员**：可以进入并使用已有的 Workspace，浏览文件树和启动终端。

如果侧栏里同事创建的项目太多，可以切换 **显示团队共享的工作区 / 隐藏团队共享的工作区** 折叠它们。这里的“共享”指 Team 内其他人创建的 Workspace；隐藏状态只保存在当前设备上，不会影响其他成员的侧栏。

文件

在右栏打开文件夹或跟随终端当前目录，预览代码、Markdown、PDF 与音视频，通过独立通道上下传大文

件，并把转义后的文件路径插进终端。

vibe42.ai/zh-HANS/docs/workspaces/files

不用配置 SFTP 或 SCP（常用的 SSH 文件传输工具），你就可以在 Vibe42 里直接浏览在线电脑的文件、预览常见格式并上下传文件。想了解文件面板支持哪些操作、预览的体积上限、文件传输如何消耗 Credits，以及不同系统需要什么权限时，可以查看本页。

能做什么与不能做什么

文件视图展示运行 Vibe42 后台服务的系统账户有权访问的全部目录，默认打开该账户的主目录（Home）。

支持的操作：

- 用 **上一级**、**主目录** 或 **编辑路径**（输入任意绝对路径后按回车）切换目录
- 用 **筛选当前目录** 在本地按文件名过滤当前目录，或用 **显示隐藏文件 / 不显示隐藏文件** 显示、隐藏以 `.` 开头的隐藏文件
- 用 **新建文件夹** 新建文件夹
- 预览源码、Markdown、图片、PDF、音频和视频
- 点击上传按钮或拖放上传（**拖放文件到此处上传**）、不限体积流式下载，以及在手机上调用系统 **分享** 导出文件
- 复制文件的绝对路径或相对路径，或者把带引号转义的路径（自动处理空格和特殊字符，可直接用于命令行）插入当前终端，也可以从终端的附件菜单把电脑上的路径插进终端（见下文）

明确不提供的操作：

> **重要**
>
> Vibe42 文件界面**不提供**修改、删除、重命名或移动文件的功能。需要改内容、删文件、改名或移动位置时，请直接在终端里用命令行工具、编辑器或 Agent 操作。

访问入口与右键菜单

你可以在两个地方浏览文件：完整的 **文件** 页面（从 Workspace 菜单点击 **在文件页打开** 进入），以及终端右边的右栏。两者用的是同一个文件浏览器。

右栏里的文件夹 Tab

在侧栏展开一个 Workspace，下面第一行是 **文件**，再往下是这个 Workspace 里的终端。点 **文件** 这一行，会在右栏打开一个以这个 Workspace 文件夹为根的文件夹 Tab；侧栏本身不再内嵌显示目录树。电脑离线时，这一行显示为灰色，点不开。（Workspace 菜单里的 **在文件页打开** 仍然打开整页的 **文件** 页面。）

文件夹 Tab 显示以这个文件夹为根的目录树，顶部的路径导航（面包屑）只到这个根为止，不会往上走出去。顶部有四个工具按钮：是否显示隐藏文件（**显示隐藏文件 / 不显示隐藏文件**）、**刷新**、**新建文件夹** 和 **上传**（上传到当前选中的文件夹）。点 **新建文件夹** 后直接在树里输入名字，按 Enter 创建，按 Esc 取消；创建失败时错误会直接显示在输入框旁边，可以改了再试；创建成功后，新文件夹会成为当前选中的文件夹。

右栏有两个常驻 Tab：**便签** 和 **当前目录**，不能关闭。其余 Tab（文件预览 Tab 和文件夹 Tab）都可以关闭，**合计最多 9 个**。再次打开同一个文件，或同一台电脑上的同一个文件夹，会直接切到已有的 Tab，不会多占一个位置；9 个都占满时再打开新的，会自动关掉**最早打开**的那一个。手机上右栏默认是关着的，只有你明确点了才会打开（铺满屏幕，Tab 栏左边有 **返回** 按钮返回）：点右栏按钮、点开一个文件、点 Workspace 下面的 **文件**，或者点终端里的文件路径。

当前目录 Tab

当前目录 Tab 跟着**当前终端**所在的文件夹走（即终端的当前工作目录，`cwd`）：

- **树的根目录**：如果这个终端属于某个 Workspace，并且它所在的文件夹在 Workspace 文件夹里面，树的根就是 Workspace 文件夹；否则根是这台电脑文件系统的最顶层。
- **自动跟随终端**：切换到另一个终端，或在终端里运行 `cd` 换到别的文件夹，树会自动定位并展开到新的位置。
- **手动点选目录**：如果你在树里手动点了别的文件夹，它会一直停在你点的位置，直到终端下一次换文件夹。
- **无终端或电脑离线**：没有打开的终端、或者那台电脑离线时，Tab 里只显示一行说明：**"打开一个终端后，这里会跟随它的当前目录。"**。

在任何目录树里右键点击（触屏设备为长按）任意目录或文件，可以使用以下快捷操作：

菜单项	适用对象	行为说明
在这里创建终端	目录	以该目录为工作目录（`cwd`）新建终端，并自动归入所在 Workspace
在终端中打开	文件与目录	在终端中打开该条目所在目录
发送路径至当前窗口	文件与目录	把用单引号转义好的路径插入当前终端，不自动按回车
复制绝对位置 / 复制相对位置	文件与目录	复制完整绝对路径或相对于 Workspace 的路径到剪贴板
上传文件到这个文件夹	目录	把本机文件上传到该目录内

预览类型与渲染上限

Vibe42 根据文件扩展名（以及常见的无扩展名配置文件名）自动选择预览方式：

类别	支持格式
文本与代码	常见编程语言源码扩展名，以及 `Makefile`、`Dockerfile`、`.gitignore` 等配置文件
Markdown	`.md` 等 Markdown 文档
图片	`png`、`jpg`、`jpeg`、`gif`、`webp`、`bmp`、`ico`、`svg`、`avif`
文档	`pdf`
音频	`mp3`、`wav`、`ogg`、`flac`、`m4a`、`aac`
视频	`mp4`、`webm`、`mkv`、`mov`、`avi`、`m4v`
压缩包	不支持在线展开预览，只提供下载

为了避免大文件或超长日志卡住界面，预览有明确的体积上限：

- **2 MiB 文本上限**：超过 2 MiB 的文本文件会提示 **文件过大，无法预览**，无法识别的二进制文件会提示 **二进制文件无法预览**，两者都直接显示下载按钮。
- **语法高亮上限**：代码高亮和富文本排版最多处理约 26 万字符、3,000 行或单行 4,000 字符；不超过 2 MiB 但超出上述范围的文本会改为分页纯文本展示。
- **音视频拖动进度**：桌面和手机客户端按需分段读取音视频数据，可以随时拖动进度条；网页版则会先读取整个媒体文件。

上传、下载与 Credits 计费

进行中和已完成的传输任务都列在 **传输** 面板里：

- **防残缺上传**：上传文件时先写入同目录下的 `.part` 临时文件，全部传完后再改名为正式文件名，避免断网留下半截文件。
- **下载不限体积**：下载采用流式传输，没有单文件大小限制。
- **独立通道不卡终端**：每台电脑都使用各自独立的 `control`、`terminal` 和 `file` 三条传输通道，传几个 GB 的大文件也不会影响终端打字（详见连接工作原理）。
- **Credits 计费规则**：当连接经过 Vibe42 中转节点（AP，包括所有网页版连接）时，文件预览和上下传流量会计入中转流量并消耗 Credits；桌面或手机客户端与后台服务点对点直连传输时不消耗 Credits（详见 Credits 与用量）。

把电脑上的路径插进终端

终端的快捷键栏（桌面和手机都有）里有一个回形针按钮 **添加图片或文件**，菜单里有 **插入远端路径**。它用来把**已经在那台电脑上**的某个文件或文件夹的路径插进终端，不上传任何东西（和拍照、从图库选择、选择文件不同，那三项会把你手头设备上的文件上传到电脑）。

如果终端所在的电脑就是你正在用的这台设备，并且系统提供文件选择窗口，就会直接打开系统的选择窗口；否则会弹出 Vibe42 自己的文件浏览窗口，既可以选文件，也可以选文件夹（网页版和 iPhone / iPad 一律用 Vibe42 自己的窗口）。选中后，路径会加上单引号转义（这样带空格和特殊字符的路径在命令里也安全）插到光标处，**不会自动按回车**。浏览窗口会从这台电脑上次用过的文件夹开始（第一次用时从当前终端所在的文件夹开始）；新建 Workspace 时选择文件夹的那一步也共用这个“上次用过的文件夹”。

平台权限与依赖

- **目录读不出来时**：如果那台电脑拒绝访问某个文件夹，或者读取超过 8 秒还没结果，Vibe42 会在原处显示一段说明：**"读取目录耗时较长，该电脑可能正在等待你授权访问。"** 或 **"该电脑拒绝访问此位置。"**，后面跟着 **"请到远程桌面或终端中完成授权，然后刷新。"**。说明下面最多有两个按钮：
  - **打开远程桌面**：在那台电脑支持远程桌面、并且不是你正在用的这台设备时出现。用它连上去，直接在那台电脑的屏幕上点允许。
  - **在此目录新建终端**：在这个文件夹打开一个新终端。在终端里运行一次 `ls`，就会触发系统的授权弹窗。

  在 macOS 上，访问“桌面”“文稿”这类受保护的文件夹时，系统授权弹窗出现在**那台电脑自己的屏幕上**，要有人在那台电脑上点允许，Vibe42 才能读取。授权完成后点 **刷新**。

- **macOS Full Disk Access（完全磁盘访问权限，可选）**：在 macOS 上，如果想直接从微信、系统邮件（Mail）等受系统沙盒隔离限制的应用里把附件拖进 Vibe42 上传，可以在系统设置里授予 Vibe42 **Full Disk Access**。
- **Linux 桌面 App 预览依赖**：Linux 桌面 App 使用隔离的 WebKitGTK 视图渲染代码、Markdown、HTML 和 PDF 预览，电脑上需要安装 `bubblewrap`（`bwrap`）和 `xdg-dbus-proxy`。

预览产物与端口转发

通过文件面板预览静态文件、用 v42 forward 加密转发开发服务器端口，或打开远程桌面查看 Agent 的运行结果。

vibe42.ai/zh-HANS/docs/workspaces/preview-and-forwarding

当电脑上的 Agent 生成了报告、截图、构建产物，或者启动了本地开发服务器时，你有三种方式查看和验证结果。想选择合适的查看方式或使用 `v42 forward` 转发端口时，可以参考本页。

查看 Agent 产物的三种方式

方式	适用场景	可用设备
文件面板预览	静态文件：Markdown 报告、截图、PDF、录屏视频、日志与源码	手机、平板、网页版、桌面 App
`v42 forward`（CLI）	监听在远程电脑 `127.0.0.1` 上的交互式网页服务、前端开发服务器与本地 API	安装了 `v42` CLI 的笔记本或台式电脑
远程桌面	桌面图形界面程序、Xcode / 模拟器，或在手机、平板与网页版上直接看电脑屏幕	手机、平板、网页版、桌面 App

在文件面板中预览静态产物

如果 Agent 把结果直接存成了磁盘文件——比如测试截图、架构图、PDF 文档、测试录屏 `.mp4` 或 Markdown 总结——不需要启动网页服务器，也不需要转发端口，直接在文件页面或 Workspace 侧栏目录树中点击对应文件就能预览，在手机和浏览器上同样可以直接打开。

使用 `v42 forward` 转发端口

想在手边的笔记本或台式机上直接访问远端正在运行的开发服务器（比如 Vite、Next.js 或后端 API）时，可以用 Vibe42 CLI 转发端口：

```
v42 forward <computer>:<remote-port> [<local-addr>:]<local-port>
```

这条命令会在你手边的电脑上监听一个本地 TCP 端口，并把流量加密转发到目标电脑的 `127.0.0.1:<remote-port>`。

常用示例

把电脑 `my-mac` 上的 `3000` 端口转发到本机 `3000` 端口：

```
v42 forward my-mac:3000 3000
```

然后在手边电脑的浏览器里打开 `http://127.0.0.1:3000` 即可访问。

把本地端口设为 `0`，操作系统会自动挑一个空闲端口：

```
v42 forward my-mac:3000 0
```

安全边界、并发上限与计费

- **本地默认只允许本机访问**：省略 `<local-addr>` 时，`v42 forward` 只绑定本机回环地址 `127.0.0.1`（只有当前这台电脑自己能访问的地址），同一个 Wi-Fi 下的其他设备无法连入这条隧道；如果你手动指定了非 `127.0.0.1` 的监听地址，CLI 会打印安全警告。
- **远端默认只允许访问本机回环地址**：默认只能转发到目标电脑自己的回环地址（拥有终端访问权限即自动具备 `forward:loopback` 权限）；如果想通过这台电脑跳板访问它所在局域网里的其他机器，必须先在目标电脑上显式放行。
- **并发上限**：单个转发会话最多支持 **64 条并发 TCP 连接**。
- **端到端加密且不记录内容**：转发流量使用与终端相同的 Noise 端到端加密通道传输；审计日志只记录连接信息，绝不记录端口转发传输的具体内容（见安全架构）。
- **Credits 计费**：当 CLI 与目标电脑上的 Vibe42 后台服务通过中转节点（AP）连接时，端口转发流量计入中转流量并消耗 Credits；点对点直连隧道不消耗 Credits。

> **说明**
>
> 端口转发只能通过命令行 `v42 forward` 使用。Vibe42 客户端界面内没有端口转发开关，也不会把你的端口暴露成公网链接。

使用远程桌面兜底

如果 Agent 开发的是 macOS 或 Linux 桌面窗口程序、启动了 iOS Simulator，或者你手边只有手机、平板或浏览器（本地无法运行 `v42` CLI）却想直接操作远端的 `localhost:3000` 页面，直接打开远程桌面即可看到并操作目标电脑的屏幕。

终端隔离

为每个新终端窗口创建一份整目录写时复制独立副本，让多个并行运行的 Agent 各自改代码、编译和提交，

互不覆盖。

vibe42.ai/zh-HANS/docs/workspaces/terminal-isolation

当多个 Agent 同时在同一个项目目录里改代码时，很容易互相覆盖文件、搞坏编译缓存或打乱 git 暂存区。终端隔离会为每个新打开的终端窗口自动建一份写时复制（CoW，一种极快的文件系统克隆技术，只有被修改的数据块才额外占用磁盘）的整目录独立副本，让它们互不干扰。想了解文件系统要求、版本自动记录规则、**副本…** 面板用法、已知限制以及本机 `v42 copy` 命令时，可以查看本页。

与 `git worktree` 的区别

`git worktree` 只会检出已纳入 git 跟踪的文件，本地 `.env` 配置、未跟踪资源以及 `node_modules`、`target` 等依赖目录往往缺失，需要自己复制或建软链接。

终端隔离直接利用文件系统的写时复制能力，在几秒内把**整个 Workspace 目录连同 `.git`、未跟踪配置和装好的依赖**完整克隆出一份独立副本；不管项目多大都能快速完成，只有真正改动过的数据块才会多占磁盘空间，而且普通非 git 目录也能用。

终端隔离是单个 Workspace 的选项，**默认关闭**。你可以在创建 Workspace 时开启，也可以之后在 **编辑…** 里打开；**一旦开启就不能再关闭**。

前提条件与不支持的情况

如果当前目录不满足创建隔离副本的条件，Vibe42 会直接拒绝新建终端并提示具体原因，不会悄悄退回原目录：

不支持的情况	处理结果与提示文案
文件系统不支持写时复制（例如 `ext4`）	拒绝并提示：**该文件系统不支持写时复制（例如 ext4）。请使用 APFS、btrfs、XFS 或 ZFS。**
目录是 `/`、用户主目录（Home）或主目录的上级目录	拒绝并提示：**主目录、它的上级目录和 / 不能开启终端隔离。**
目录本身是一个 `git` linked worktree	拒绝创建
路径不存在或不是文件夹	拒绝创建
电脑上的 Vibe42 后台服务版本过旧	拒绝创建，需先升级后台服务
副本索引损坏	拒绝创建，需在电脑本机执行 `v42 copy rebuild-index --yes`
正在执行首次压缩与去重转换	暂停创建并提示正在转换，转换完成后恢复

副本位置、编号与分屏共享规则

每次新建终端窗口时，后台服务会在原目录同级建一个带编号的副本目录：

```
<父目录>/.<目录名>.v42-copies/<编号>/work
```

- **编号规则**：`<编号>` 总是取当前最小的可用正整数（`1`、`2`、`3`……）。
- **同窗口分屏共用**：在同一个终端窗口里向右或向下分屏（把一个终端窗口拆成多个并排区域），窗口内的所有分屏共用同一个副本。

- **子目录新建终端**：在副本里的某个子目录上点击 **在这里创建终端**，会再克隆出一份新编号副本，并把新终端的当前工作目录（`cwd`）设到新副本里相同的相对子目录。
- **同路径共享数据**：同一台电脑上指向同一个路径的多个 Workspace，共用同一套副本和版本数据。

版本自动记录

当副本里的文件有改动时，Vibe42 会在下面四个时机自动保存一个版本：

- Agent 完成一轮任务（通过 OSC 7501 状态控制序列把 Agent 状态更新为 **已完成**、**需要处理** 或 **出错**）
- 有未保存改动时，每 10 分钟记录一次
- 终端窗口关闭时
- 手动点击 **保存版本**（或运行 `v42 copy save`）

如果文件没有任何变化，就不会生成空版本。任何层级的 `.git` 目录以及被 git 忽略的文件都不计入版本历史。

在 副本… 面板审阅与写回

打开 Workspace 菜单里的 **副本…**，可以查看所有副本的状态、历史版本和改动文件：

- **副本状态**：**使用中**、**已关闭**、**准备中**、**文件夹缺失**、**已在 Vibe42 之外被改动**、**没有历史** 或 **起点缺失**。
- **文件改动状态**：每个改动过的文件会标为 **未应用**、**已应用** 或 **冲突**（冲突提示：**工作区文件夹里这个文件也被改过，请手动处理。**）。
- **应用所选 / 全部应用**：把还没写回的文件改动写回原 Workspace 目录。真正写入前会逐个再次核对文件，如果某个文件在列表生成后又被改过，会自动跳过。
- **推送为分支**：在 git 仓库中，把副本当前的 `HEAD` 提交作为新分支推送回原 Workspace 仓库。git 本身会拒绝推送到原仓库当前已检出的分支，也会拒绝非快进（non-fast-forward）更新。
- **删除副本**：从磁盘删除该副本；如果还有未写回或未保存的改动，会先弹窗提醒。

生命周期、自动清理与外部改动

副本保存在电脑本地磁盘上。关闭终端窗口时，Vibe42 会先保存最后一个版本。如果当前没有其他分屏在使用这份副本，并且副本是**干净**的（没有未写回的文件改动，且在 git 仓库中副本的 `HEAD` 提交已经存在于原仓库），Vibe42 会自动删掉这份副本；还有未处理改动的副本会留在磁盘上，直到你手动写回、推送或删除。

如果在 Vibe42 之外直接修改了副本或原目录里的文件、导致改动记录对不上，Vibe42 会明确报错或标为冲突，不会悄悄覆盖文件。

已知限制

- Python `venv` 脚本等内部写死了原目录绝对路径的文件，在副本里仍然指向原 Workspace 路径。
- 在新副本里第一次运行 `cargo build` 时，编译结果依赖绝对路径的 Rust crate 会重新编译。
- 被 git 忽略的文件不会记入版本历史，干净副本自动删除时这些文件也会一起删掉。
- 修改 Workspace 的目录路径后，旧路径下的副本仍留在原来的位置。

本机 `v42 copy` 命令与 Agent Skill

隔离终端里会自动设置环境变量 `V42_COPY` 指向当前副本。你可以在电脑本机直接运行以下命令管理副本，不需要登录账号：

```
v42 copy status
v42 copy save
v42 copy versions
v42 copy changes
v42 copy diff
v42 copy apply
v42 copy push
v42 copy rebuild-index --yes
v42 copy skill
```

运行 `v42 copy skill` 会打印一份标题为 **"Working in a Vibe42 isolated copy"** 的 `SKILL.md` 说明文档，你可以按需安装给自己的 Agent（详见零注入）。

同时开启终端隔离与压缩与去重时，`.v42-copies` 副本目录也会自动去重和压缩；只有在首次全目录转换期间，才会暂停新建隔离终端以及副本的写回、推送、删除和重建索引操作。

压缩与去重

在 APFS、btrfs、XFS 与 ZFS 文件系统上按文件内容去重并启用透明压缩，在不改变文件读取内容的前提下为 Workspace 与隔离副本节省磁盘空间。

`vibe42.ai/zh-HANS/docs/workspaces/storage-dedup`

项目里的依赖目录、编译缓存和多个并行 Agent 副本，常常会在磁盘上留下成千上万个内容完全相同或很容易压缩的文件。压缩与去重是一项按 Workspace 开启的目录选项，能在底层文件系统层面自动回收本地磁盘空间，同时完全不改变任何程序读到的文件内容。想查看支持的文件系统、首次转换期间的限制、安全跳过规则以及它如何与终端隔离配合时，可以参考本页。

做什么与不做什么

在某个 Workspace 上开启后，电脑上的 Vibe42 后台服务会让内容完全相同的文件共用同一份底层数据块（去重），并在文件系统支持时开启透明压缩（由操作系统在写入磁盘时自动压缩、读取时自动解压，对应用程序完全透明）：

- **读到的文件内容完全不变**：编译器、编辑器和命令行工具读到的文件内容、权限位（mode）和修改时间与处理前完全一样。
- **不触碰特殊文件**：绝不改动任何层级的 `.git` 目录、符号链接（软链接）、硬链接、属于其他系统用户的文件，以及带有特殊文件系统标记的文件。
- **开启后不可关闭**：压缩与去重**默认关闭**；一旦在某个 Workspace 上开启，**就不能再关闭**。

文件系统支持矩阵

只有支持安全数据块共享的文件系统才能使用此功能。注意 `ext4` 不在支持范围内：

操作系统与文件系统	按内容去重	文件系统透明压缩	支持状态
macOS **APFS**	支持	`decmpfs`（LZFSE）	支持
Linux **btrfs**	支持	`zstd`	支持
Linux **XFS**	支持	无（仅去重）	支持
Linux **ZFS**	支持（块克隆）	由 dataset 的 `compression` 属性决定	支持（需 OpenZFS 开启块克隆）
ext4 及其他	—	—	不支持

在不支持的文件系统上尝试开启时，界面会提示：

> **这个文件系统不支持。请使用 APFS、btrfs、XFS 或 ZFS。**

另外，当目录是 `/`、用户主目录（Home）或主目录的上级目录，或者电脑上的后台服务版本过旧、电脑当前离线时，也无法开启。

开启即转换与后台持续监听

开启选项后，会立刻对整个 Workspace 目录执行一次全量转换：

- **后台服务独立运行**：转换全程在电脑本地后台运行，不需要 Vibe42 客户端一直保持连接；就算电脑或后台服务中途重启，恢复后也会自动接着处理。
- **日常自动处理**：首次全量转换完成后，后台服务会持续监听该目录——文件停止修改约 **1 分钟**后自动在后台处理，并每 **30 分钟**做一次全目录例行检查。

首次转换期间的临时限制

在第一次全目录转换进行期间：

- 暂停新建隔离终端，也暂停对已有副本执行写回（apply）、推送分支（push）、删除副本（delete）和重建索引（`rebuild-index`），界面会提示正在转换。
- 该 Workspace 的 **编辑…** 和 **归档** 暂时禁用，等转换完成后恢复。
- 文件面板的浏览、上下传和普通（非隔离）终端**照常使用**，不受影响。

文件处理规则

每个文件在处理前和替换写入时都要通过以下安全检查：

规则项	具体行为
最小文件体积	只处理大小 **≥ 16 KiB** 的普通文件
静置时间要求	跳过最近 **1 分钟内改过**的文件
写入占用避让（macOS APFS、Linux ZFS）	自动跳过当前正被任何程序以写入方式打开的文件

规则项	具体行为
排除范围	跳过任何层级的 `.git`、符号链接、硬链接、其他用户的文件和带特殊标记的文件
替换前再次核对	真正替换前再次确认文件内容和状态没有变化，并在替换后原样恢复权限位和时间戳

状态进度与节省统计

你可以在 Workspace 设置里查看当前的转换阶段和省下的磁盘空间：

- **运行状态**：依次为 **等待开始…** → **正在扫描文件…** → **正在比对文件（…%）** → **正在转换文件（… / …）** → **正在保存…**。
- **空间节省统计**：完成后显示 **已节省 …**，并分别列出靠**去重**和靠**透明压缩**各自省下的磁盘空间。
- **错误提示**：如果最近一次处理遇到问题，界面会显示 **上次出错：…**。

与终端隔离的关系

压缩与去重和终端隔离是两个互不依赖的开关，可以只开其中一个，也可以同时开启。当两者都开启时，终端隔离生成的 `.v42-copies` 副本目录也会自动纳入去重和压缩范围，让多个副本并行编译产生的重复文件少占磁盘空间。

备忘、休息提醒与清理

在 Workspace 中用备忘保存常用提示词并一键送入终端，配置跨设备同步的每日休息提醒，管理本机后台服务开关，以及清理离线电脑。

vibe42.ai/zh-HANS/docs/workspaces/extras

除了终端、文件树和隔离副本，Vibe42 还提供了几项日常辅助功能：随项目保存常用指令的备忘、按每日时段锁定的休息提醒、控制当前 Mac 能否被远程访问的本机后台服务开关，以及离线机器清理。需要设置这些选项或了解清理影响时，可以查看本页。

Workspace 备忘（Memos）

备忘保存在具体的 Workspace 下，适合把常用的 Agent 提示词、测试命令或交接说明直接放在对应项目旁边。备忘在右栏的 **便签** Tab 里查看，这里显示的是你当前打开的终端或文件所属 Workspace 的备忘。

三档可见性

每条备忘都可以在当前 Team 内单独设置谁能看、谁能改：

可见性档位	谁可以查看	谁可以编辑
私有	仅创建者本人	仅创建者本人

可见性档位	谁可以查看	谁可以编辑
团队只读	Team 所有成员	仅创建者本人
团队可编辑	Team 所有成员	Team 所有成员

发送到终端与状态管理

- **发送到当前终端**：把备忘内容直接填入当前终端的光标位置，**不会自动按回车**，方便你在执行前检查参数或补充内容。
- **标记完成**、**恢复** 与 **删除便签**：处理完的条目可以标为完成，需要时可以恢复为未完成，也可以永久删除。
- **未完成数量上限**：每个 Workspace 里“未完成”状态的备忘条数有上限；达到上限后，把旧条目标为完成或删掉就能继续新建。

休息提醒

随时能用手机推进 Agent 任务，也很容易在深夜不知不觉工作太晚。你可以在 **设置 → 健康管理** 中设置每日休息提醒：

- **云端同步、按本机时间生效**：每天可以设一个固定的休息时段。时段设置保存在服务端，并在你登录的各台设备间同步，按每台**设备各自的本地系统时间**生效。
- **时段内只开放设置页**：进入休息时段后，客户端里除了 **设置** 之外的终端和工作面板都会锁定（保留设置页，方便遇到紧急情况时临时改时间或关掉提醒）。
- **通知照常推送**：休息时段内，推送通知仍然正常发送，不会漏掉重要告警。

本机后台服务开关

首次打开 macOS 桌面 App 时，界面会询问 **要把这台电脑作为可远程控制的电脑吗？**（提供 **启用这台电脑** 和 **暂不启用** 两个选项），之后也可以随时开关本机 Vibe42 后台服务：

- **占用一个电脑名额**：开启本机后台服务会把当前这台 Mac 注册为可远程控制的电脑，占用当前 Team 套餐里的一个电脑名额。
- **不开启也能用本地终端**：如果选择 **暂不启用**（或者当前 Team 的电脑名额已满），桌面 App 里的本地终端依然可以正常使用，你也照样可以用这台 Mac 远程连接 Team 里的其他电脑。
- **只对当前 Team 生效**：本机后台服务的开启和关闭状态只作用于当前选中的 Team。

日常清理：离线电脑与 最近

清理离线超过 3 天的电脑

临时创建的云服务器或重建过的容器，常会在电脑列表里留下不再使用的旧记录。你可以在 **电脑** 页面点击 **移除离线电脑**，一键移除离线超过 3 天的电脑。

> 注意
>
> 清理离线电脑会**永久删除**这些电脑在云端关联的所有 Workspace 记录以及里面的全部备忘。如果某台笔记本或台式机只是临时关机、休眠或出差没联网，请不要执行这项清理。

从最近隐藏

当侧栏 **最近** 标签页里攒了不再需要显示的条目时，可以点击 **从最近隐藏** 把它从列表中移除；具体可移除的条目与效果以 App 内显示为准。

06 远程桌面

远程桌面：开始使用

从任意已配对设备查看并操控你的 Mac 或 Linux 桌面，完成首次权限配置并了解平台支持范围。

vibe42.ai/zh-HANS/docs/remote-desktop/getting-started

远程桌面让你在已配对的手机、平板、电脑或浏览器上实时查看并操控 Mac 或 Linux 电脑的整个桌面，或者只打开某一个图形界面应用窗口。当你除了终端之外还需要操作 Xcode、浏览器、模拟器等桌面程序，或第一次为电脑开启屏幕录制与控制权限时，请阅读本页。

平台支持矩阵

远程桌面要求被控电脑已进入图形桌面会话，主控设备安装 Vibe42 客户端或使用支持 WebCodecs（浏览器原生视频编解码接口）的现代浏览器。你不能从当前设备向它自己发起远程桌面连接。

平台	作为被控电脑（Computer）	作为主控设备（Device）
macOS 14+（macOS App）	支持（内置后台服务）	支持
Linux 桌面（X11）	支持	支持（Linux App）
Linux 桌面（Wayland）	支持 KDE Plasma 6、GNOME 与 Hyprland	支持（Linux App）
iPhone 与 iPad（iOS 17+）	不支持	支持
Android（11+）	不支持	支持
Web 应用（`app.vibe42.ai`）	不支持	支持（浏览器需支持 WebCodecs）
Windows	不支持	支持（通过浏览器使用 Web 应用）

被控电脑上的 Vibe42 后台服务版本必须支持远程桌面。如果后台服务版本过旧，界面会显示 **这台电脑暂不支持远程桌面** 并提示更新。升级步骤见更新与卸载。

macOS 首次授权与控制开关

在其他设备查看或控制你的 Mac 之前，需要在 macOS 中开启两项系统权限：

1. **屏幕录制**：用于采集屏幕画面和窗口内容。未授权时其他设备无法看到画面。从 macOS 15 起，系统会定期在 Mac 屏幕上弹窗，请你在本机再次确认录屏权限。
2. **辅助功能**：用于向 Mac 发送远程鼠标移动、点击、滚动和键盘按键。如果只开启了 **屏幕录制** 而未开启 **辅助功能**，连上后只能看画面，不能操作鼠标和键盘。

macOS App 会在首次设置时引导你开启这两项权限；之后你也可以随时在 macOS App 的 **设置 → 远程桌面** 中查看或修改（该设置页仅在 macOS App 中提供）。

同一个设置页里还有 **允许其他设备远程控制这台 Mac** 开关，默认开启。关掉这个开关后，正在连接这台 Mac 的远程桌面会立即断开，其他设备的远程桌面列表里也不会再显示这台 Mac。

Linux 各桌面环境的首次准备

Linux 被控电脑的首次授权方式与视频编码设置取决于你使用的显示服务（X11 或 Wayland）和显卡驱动：

- **X11**：无需系统弹窗授权，直接连接即可。
- **KDE Plasma 6 与 GNOME（Wayland）**：第一次有设备连入时，被控电脑屏幕上会弹出系统授权窗口。请在电脑前点击同意并开启远程交互（remote interaction），系统会记住这次选择，以后重连不再弹窗。
- **Hyprland（Wayland）**：无需系统授权弹窗即可直接连接。

视频编码方面，Linux 上的后台服务优先调用系统的 `ffmpeg` 进行显卡硬件编码（VA-API、NVENC 或 V4L2）。如果没有可用的硬件编码，则会回退到 Cisco OpenH264 软件编码。OpenH264 不随安装包内置，你可以在 Linux App 设置中开启，或在被控电脑的终端里运行以下命令从 Cisco 官方下载并校验编解码器：

```
~/.v42/bin/v42-companion codec enable
```

以后如果配好了显卡硬件编码，也可以随时把它关掉。主控设备的视频解码要求见操控、画质与剪贴板。

如何进入远程桌面

你可以从以下几个入口打开远程桌面：

- 在左侧栏点击 **远程桌面**，选择在线的电脑。
- 打开新建终端弹窗，选择支持远程桌面的电脑（也可以直接打开单个应用窗口，详见解锁屏幕与打开 App）。
- 在全局搜索中查找对应电脑的桌面。
- 在 Web 应用中直接访问 `/computers/<id>/desktop`。

打开过的远程桌面会自动记录在左侧栏的 **最近** 中，你还可以点 **置顶** 把常用的桌面固定在侧栏，和常用终端、Workspace 放在一起随时切换。

> **重要**
>
> 远程桌面需要持续传输实时视频画面，流量消耗远高于普通终端。原生 App 之间建立点对点（P2P）直连时不消耗任何 Credits；而使用浏览器 Web 应用时流量始终经过 Vibe42 中转服务器转发，所有中转流量都会计入 Team 的月度 Credits。详见连接机制与 Credits 与用量。

不支持的功能清单

使用远程桌面时请注意以下限制：

- **开机后尚未登录的系统登录界面**：电脑刚开机或重启后，如果停留在还没有登录任何系统账号的登录界面，无法通过远程桌面登录（已登录账号后的锁屏界面支持远程解锁，见解锁屏幕与打开 App）。
- **虚拟显示器与防窥黑屏**：不支持创建虚拟显示器，也不支持把被控电脑的物理显示器变黑。
- **声音与麦克风**：不支持传输电脑声音或转发本地麦克风语音。
- **图片富文本剪贴板与拖拽传文件**：剪贴板只同步 60 KiB 以内的纯文本，不支持复制粘贴图片、带格式的富文本，也不支持直接把文件拖进远程桌面画面上传。传文件请使用文件面板或 `v42 cp` / `v42 sync`。

- **多人同时操作的键鼠锁定**：多台设备同时连入同一个远程桌面时，所有设备都可以操作鼠标和键盘，不做主从锁定。

操控、画质与剪贴板

调节远程桌面画质档位与画面缩放，了解编解码器要求、多显示器切换、桌面与移动端键鼠输入及纯文本剪贴板同步规则。

vibe42.ai/zh-HANS/docs/remote-desktop/controls

连上远程桌面后，你可以根据当前网络情况调节画质与帧率、在多块显示器之间切换、选择适合电脑或触屏的操控方式，并通过剪贴板同步纯文本。想降低延迟或节省中转 Credits、在手机平板上操作电脑桌面，或了解视频解码与复制粘贴规则时，请阅读本页。

画质五档与选档建议

顶部控制条提供五个画质档位：

档位	行为说明	适用场景
自动	默认档位，按实时网速和网络延迟自动调整分辨率与帧率	日常连接、手机移动网络或信号有波动的 Wi-Fi
流畅	优先保证低延迟和画面流畅，适当降低画面细节	跨国慢速网络或快速切换窗口
均衡	兼顾界面文字清晰度与流量消耗	经服务器中转时的日常桌面查看与操作
清晰	保留清晰的文字边缘和界面细节	阅读代码、日志或文字密集的调试面板
超清	以显示器原始分辨率、每秒 60 帧（60 fps）传输画面	局域网或点对点（P2P）直连下需要最高清晰度与流畅度的场景

> 提示
>
> 长时间开启 **超清** 前，建议先确认当前连接是否为点对点直连。如果流量正经过中转服务器转发（例如使用浏览器 Web 应用），原始分辨率 60 fps 的视频流会快速消耗套餐 Credits。详见 Credits 与用量。

画面缩放与编解码要求

画面提供两种缩放模式，决定远端屏幕如何显示在你的窗口里：

- **适应窗口**：按比例把整块远端屏幕（或单个窗口）完整缩放到当前窗口内，不用滚动就能看全。
- **原始尺寸**：按 1:1 原始像素大小显示画面，适合在小屏幕上平移查看高分辨率显示器的局部细节。

视频编码默认选 **自动**；如果你的主控设备能解码对端电脑支持的多种视频格式，也可以在控制条上手动切换。各平台主控设备的视频解码要求如下：

- **macOS、iOS 与 Android App**：直接调用系统自带的视频解码器。
- **Web 应用（`app.vibe42.ai`）**：浏览器必须支持 **WebCodecs** API。
- **Linux 桌面 App**：优先使用显卡硬件解码（VA-API 或 NVIDIA NVDEC）；如果没有可用的硬件解码，会按需下载并校验 Cisco OpenH264。

被控 Linux 电脑的视频编码设置方法见远程桌面：开始使用。

多显示器切换

当被控电脑连了多台物理显示器时：

- 打开 **显示器** 列表，或直接在顶部控制条的显示器菜单里切换屏幕。
- 在左侧栏展开这台电脑，可以直接点 `Screen 1`、`Screen 2`（`Screen N`）切换到对应屏幕，也可以直接点选某个单独的应用窗口（详见解锁屏幕与打开 App）。

键盘与指针操控

桌面端与 Web 应用

实体键盘的按键会直接发往被控电脑。使用中文或其他需要选词的输入法（IME）时，拼音和候选词在你的本地设备上完成输入，确认上屏后的文字会直接发到远端窗口的光标处。

当 macOS App 在前台且焦点在远程桌面画面内时，App 会临时接管系统快捷键，让切换应用等组合键发往被控电脑而不是你手头的 Mac；把鼠标点到远程桌面画面外即可恢复本机系统快捷键。

iPhone、iPad 与 Android 移动端

在手机和平板上，你可以通过工具栏在两种触控模式之间切换：

- **鼠标模式**：把屏幕当成虚拟触控板用，滑动手指移动鼠标指针、轻点代表左键点击，配合工具栏上的右键按钮打开右键菜单。适合点小按钮、菜单项或拖动窗口边缘。
- **触控模式**：手指点在画面的哪个位置，就直接点击远端屏幕的对应位置。适合点击大按钮和快速操作。

移动端工具栏还带有一排修饰键与功能键（用于组合 Command、Ctrl、Alt/Option、Shift、Esc 和 F 功能键）、独立的右键按钮，以及把手机剪贴板里的文字直接粘贴到远端的 **粘贴到远端** 按钮。

剪贴板同步规则

远程桌面剪贴板**只同步 60 KiB 以内的纯文本**。图片、带格式的富文本以及复制的文件不会通过剪贴板传输。

为了防止意外覆盖你剪贴板里原有的内容，同步遵循三条规则：

1. **连上时不自动推送旧内容**：建立远程桌面连接前，两边电脑或手机剪贴板里原本就有的文字，不会在连上的瞬间自动覆盖对方。

2. **远端 → 本机（自动写入）**：在连接期间，只要你在被控电脑上新复制了一段不超过 60 KiB 的纯文本，这段文字就会自动写入你当前主控设备的剪贴板。
3. **本机 → 远端（粘贴时发送）**：Vibe42 不会在后台持续读取你的本机剪贴板；只有当你在远程桌面里按下 `⌘V`（或点粘贴 / **粘贴到远端**）时，才会把本机剪贴板里的纯文本发到被控电脑并完成粘贴。

解锁屏幕与打开 App

远程解锁已锁屏的 Mac 或 Linux 用户会话，直接启动单个应用进入单窗口视图，并借助侧栏的 最近 列表与置顶 操作快速重连。

vibe42.ai/zh-HANS/docs/remote-desktop/unlock-and-apps

当你离开座位后电脑自动锁屏时，Vibe42 可以远程解锁当前已登录的桌面会话，还支持只打开或查看某一个应用窗口，不用在小屏幕上缩放整块大显示器。想了解各系统的远程解锁方式、解锁结果含义或单窗口视图用法时，请阅读本页。

远程解锁锁屏画面

当你连上一台处于锁屏状态的电脑，且电脑上的 Vibe42 后台服务检测到当前锁屏支持解锁并允许远程控制时，你手头的设备会自动弹出解锁密码框。输入这台电脑当前已登录系统账号的开机密码并提交，后台服务就会在电脑本机完成解锁。

提交密码后，后台服务会返回以下六种结果之一：

返回结果	含义说明
`Unlocked`	密码验证通过，屏幕已解锁并进入桌面画面。
`NotLocked`	请求到达时电脑本来就没有锁屏，无需解锁。
`WrongPassword`	系统拒绝了输入的密码，请核对当前登录账号的密码后重试。
`NotOnConsole`	当前用户会话不在物理显示器前台（例如系统通过快速用户切换切到了另一个系统账号）。
`Untypable`	密码里包含当前输入方式无法输入的字符，或操作系统拦截了模拟按键。
`Failed`	系统锁屏程序或会话管理器在解锁时遇到了其他错误。

各平台支持的锁屏器

- **macOS**：支持用户已登录系统后的 macOS 原生锁屏（需已开启 **辅助功能** 权限与 **允许其他设备远程控制这台 Mac** 开关，见远程桌面：开始使用）。
- **Linux — KDE Plasma 6 与 GNOME（Wayland）**：通过系统的 `logind` 会话接口完成解锁。
- **Linux — Hyprland（Wayland）**：支持 `hyprlock`。
- **Linux — X11**：支持实现了标准 freedesktop 屏幕保护 D-Bus 接口的锁屏程序（例如 `xfce4-screensaver`）。直接独占键盘输入但不提供 freedesktop 接口的独立 X11 锁屏程序无法被检测或远程解锁。

> **警告**
>
> 远程解锁只适用于系统账号**已经登录**之后的锁屏画面。如果电脑刚开机或重启，还停留在尚未登录任何账号的系统登录界面（login window），用户级后台服务此时还未启动，无法通过远程桌面完成首次登录。

密码安全机制

你输入的系统密码只在点击解锁的那一次请求中，通过端到端 Noise 加密连接直接发给被控电脑上的后台服务。你的主控设备、后台服务以及 Vibe42 服务器都不会保存或记录这个密码。加密细节见安全架构。

打开 App 与单窗口视图

在手机、iPad 分屏或小尺寸笔记本窗口里显示整块超宽显示器，往往字太小看不清。单窗口视图让你可以只打开或查看远端电脑上的某一个应用窗口：

- **从远程桌面侧栏树进入**：在左侧栏展开目标电脑，可以看到各块物理显示器（`Screen N`）、**打开 App** 入口，以及按运行中应用分组的已打开窗口列表。点任意窗口即可单独查看它；点 **打开 App** 可以选择新应用并启动。
- **从新建终端弹窗进入**：在新建终端弹窗里选择支持远程桌面的电脑，会打开 **在「…」上打开 App** 应用选择列表。点选应用后会在远端电脑上启动它，并直接进入单窗口视图。

在单窗口视图中，窗口菜单提供三项操作：

- 打开远端应用的 **App 菜单**
- 点击 **关闭窗口** 正常关闭远端电脑上的这个窗口
- 点击 **调整窗口到画布大小**，把远端电脑上该应用窗口的实际大小调整为匹配你当前 Vibe42 窗口的尺寸与长宽比

在 iPad 或小屏笔记本上用 IDE 和模拟器时，**调整窗口到画布大小** 能让远端窗口刚好铺满当前画面。完整操作步骤可参考指南解锁你的 Mac 并运行 Xcode。

最近 与 置顶 固定

每个远程桌面会话（路由为 `/computers/<id>/desktop`）打开后都会自动记入左侧栏的 **最近**。如果你经常在 Agent 终端和某个图形窗口（比如模拟器或本地预览浏览器）之间来回切换，可以用 **置顶** 把这个远程桌面会话固定在侧栏顶部，随时一键切回。

07 账号、Team 与套餐

账号与登录

了解账号注册、邮箱验证、GitHub 与 Google 关联、Passkey 与 TOTP 两步验证、各端登录方式及注销账号的影响。

vibe42.ai/zh-HANS/docs/account/account-and-login

当你要注册 Vibe42 账号、配置 Passkey 或两步验证、在桌面 App 或 CLI 中登录，或者确认注销账号的影响时，可以查看本页。

注册与邮箱验证

前往 `https://vibe42.ai/register`，可以用以下两种方式注册：

- **邮箱 + 密码**：密码不少于 **10 个字符**。
- **第三方账号登录（OAuth）**：点击 **使用 GitHub 继续** 或 **使用 Google 继续**。

用邮箱和密码注册后，系统会立即发送验证邮件。验证链接在 **24 小时**内有效，重发邮件有短暂冷却时间。**未验证邮箱的账号无法在任何设备上登录。** 如果没收到邮件，请按常见问题中的邮件排查步骤处理。

GitHub / Google 登录与自动关联规则

Vibe42 只接受 GitHub 或 Google 返回且已验证的主邮箱：

- 如果该主邮箱与已有的**已验证** Vibe42 账号一致，登录时会自动关联到该账号。
- 如果该主邮箱匹配到一个**未验证邮箱**的密码注册账号，系统会自动关联并将账号标为已验证，同时**清空之前设置的旧密码**，防止他人提前用你的邮箱抢注并预设密码。之后如果还需要用密码登录，可以去 `/password-reset` 重新设置密码。

你也可以随时在 **Console → 登录与安全** 手动绑定或解绑 GitHub 与 Google。

忘记密码

忘记密码时，前往 `/password-reset` 向注册邮箱发送重置邮件，即可设置不少于 10 个字符的新密码。

Passkey、TOTP 两步验证与恢复码

打开 **Console → 登录与安全** 可以为个人账号开启更强的登录保护：

- **通行密钥**：绑定硬件安全密钥或系统通行密钥（Touch ID、Windows Hello、密码管理器等），私钥始终留在你的设备里。
- **身份验证器**（基于时间的一次性动态码两步验证）：用身份验证器应用扫码绑定后，系统会生成一次性**恢复码**，请离线保存。不要把密码、动态码或恢复码发给任何人，包括官方客服。

敏感操作二次验证（step-up）何时触发

即使浏览器已经处于登录状态，执行以下敏感操作前也必须重新完成一次二次验证（按 **Passkey > TOTP > 密码** 的顺序验证）：

- 在 **Console → 登录与安全** 修改登录凭据或两步验证设置；
- 绑定或解绑 GitHub / Google；
- 在账号级设备管理页撤销已授权的安装；
- 注销（删除）Vibe42 账号。

不同客户端如何登录

- **Console 与网页端（`https://app.vibe42.ai`）**：直接在浏览器里登录。
- **桌面 App（macOS / Linux）与 `v42` CLI**：点击 **登录**（或运行 CLI 登录命令），系统浏览器会打开标题为 **登录 Vibe42 应用** 或 **授权设备登录** 的确认页。核对页面上的设备信息，**确认是你本人发起的请求**，再点击 **批准**。
- **iPhone、iPad 与 Android App**：移动端不用账号密码登录，统一用已登录设备生成的 8 位配对码或二维码配对（详见配对设备并开始第一个会话）。

会话列表与登出

你可以在 Console 查看当前活跃的网页会话与已授权的设备安装，随时单独登出某个会话或撤销遗失的设备。撤销后，设备会在 60 秒内断开连接。

注销账号的后果

> 注意
> 注销账号需要通过二次验证，确认后会立即解散你名下所有的 Team，且无法恢复。

确认注销账号后：

- 你作为 **所有者** 的所有 Team 会立即解散；
- 生效中的订阅立即终止，**不会自动退款**（退款条件见试用、兑换码与退款）；
- 这些 Team 下的所有成员、电脑与设备会立即失去访问权限；
- 个人账号数据会在 **30 天内**删除或匿名化，历史账单发票与税务记录按法律要求保留。

Team、角色与共享电脑

了解个人 Team 与多 Team 切换、所有者/管理员/成员 三种角色权限、成员邀请流程，以及用 仅自己 与 全

团队 共享电脑。

vibe42.ai/zh-HANS/docs/account/teams

当你要邀请同事加入 Team、核对 **所有者**、**管理员** 与 **成员** 三种角色的权限，或者把一台电脑共享给 Team 成员时，可以查看本页。

个人 Team 与同时加入多个 Team

每个 Vibe42 账号注册后都会自动创建一个个人 Team，你自己就是它的 **所有者**。你也可以接受邀请加入其他人的 Team，在同一个账号下切换多个 Team。

套餐、名额与 Credits 都归属于 Team，而不是个人。受邀加入已开通多人套餐的 Team 后，你可以直接使用该 Team 的名额，不需要自己另外买套餐（详见套餐与名额和与你的 Team 共享一台电脑）。

角色权限对照表

权限项	所有者	管理员	成员
选择套餐、付款与管理账单	允许	不允许	不允许
管理成员与发送邀请	允许	允许（不含变更套餐与 Owner）	不允许
删除电脑与启用/停用电脑	Team 内任意电脑	Team 内任意电脑	不允许
登出设备	Team 内任意设备	Team 内任意设备	仅限自己的设备
查看 Team Credits 与名额用量	允许	允许	不允许
创建（接入新）电脑	默认允许	默认允许	默认不允许（可逐人单独开启）

成员邀请流程

1. 在 Console 输入受邀人的邮箱地址发送邀请。如果对同一个邮箱再次发邀请，上一封未处理的邀请会自动失效。
2. 受邀人登录一个**已验证邮箱、且邮箱地址与受邀邮箱完全一致**的 Vibe42 账号。
3. 受邀人在 `https://vibe42.ai/console/invites`（或登录 App 时的 Team 选择页）查看并接受邀请。

系统在**受邀人点击接受时**才检查并占用成员席位，发送邀请时不占用。如果接受时席位已满，界面会提示席位不足，等腾出空位或由 Owner 升级套餐后即可加入。

电脑的访问范围：仅自己 与 全团队

每台电脑的访问范围默认为 **仅自己**，**只有这台电脑的创建者可以修改**：

- **仅自己**（默认）：Team 里的其他 **成员** 看不到也连不上这台电脑。
- **全团队**：Team 里的全体 **成员** 都能看到、连接并使用这台电脑上的终端、文件与远程桌面。**全团队** 只开放使用权限，**不开放管理权限**，普通成员不能删除或改动他人电脑的配置。

> **重要**
>
> Team 的 **所有者** 与 **管理员** 不受访问范围限制。无论电脑设为 **仅自己** 还是 **全团队**，Owner 和 Admin 始终能看到并连接 Team 里的所有电脑，并控制哪台电脑启用（占用电脑名额）或停用。

退出 Team 与被移出 Team

- 任何 **管理员** 或 **成员** 都可以随时退出 Team；**所有者** 不能退出自己的 Team。
- 成员退出或被移出后，他在该 Team 下的电脑与设备会在 60 秒内断开与该 Team 的连接，但电脑本地的文件和正在运行的程序不受影响。

切换 Team 与不可用拦截页

- **Team 选择页**：如果你属于多个 Team，或者有待处理的邀请，登录 App 后会先打开 Team 选择页，让你选择要进入哪个 Team。
- **Team 不可用拦截页**：如果当前 Team 暂时无法使用（比如当月 Credits 用完、Team 被冻结，或你已不在该 Team 中），App 会说明原因，并提供按钮直接切换到账号下的其他 Team。

套餐与名额

了解 Vibe42 套餐包含的三类名额、App 内四种拦截提示的处理方式，以及变更套餐、提前 7 天出账与取消订阅规则。

`vibe42.ai/zh-HANS/docs/account/plans-and-slots`

当你要在个人套餐与 Team 套餐之间做选择、在 App 里遇到名额已满提示，或者想确认升降级与取消订阅何时生效时，可以查看本页。

套餐包含什么

Vibe42 没有免费套餐。每份生效的套餐都归属于一个 **Team**，包含：

- **电脑名额**：供已启用的电脑使用；
- **设备名额**：供已登录的原生 App、已登录的网页浏览器，以及运行 Vibe42 后台服务的 Server 版（只有后台服务、没有桌面 App 的版本）使用；
- **成员席位**：供加入该 Team 的成员使用；
- **每月 Credits**：用于抵扣经由 Vibe42 接入点（AP）中转的流量（详见 Credits 与用量）。

个人套餐供单人使用；**Team 套餐支持多名成员在同一个 Team 内共享电脑**。各套餐的名额数量与价格见 `https://vibe42.ai/#pricing` 和 Console（`https://vibe42.ai/console/checkout?resource=access`）。在价格或方案调整前已订阅的老用户，只要不主动切换，就保留原有套餐。

> **说明**
>
> iPhone 与 Android App 内不显示价格、购买入口或订阅设置。请在浏览器中打开 Console 选购或管理套餐。

名额核算规则摘要

- 每台**已启用**的电脑占 1 个电脑名额。
- 每个已登录的原生 App、每个登录了 `app.vibe42.ai` 的浏览器，以及每台一行命令安装的服务器（**服务器版**），各占 1 个设备名额。
- 成员席位在受邀人接受邀请时检查并占用。
- 电脑关机或断网**不会**释放名额。要腾出名额，请前往 Console 删除不再使用的电脑，或登出闲置设备（可以点击 **解除离线设备配对** 一键登出离线超过 3 天的设备）。

想了解为什么一台装了桌面 App 的 Mac 会同时占用 1 个设备名额和 1 个电脑名额，请查看电脑、设备与 Team。

App 内的四种拦截提示与去处

当 Team 的名额用满或订阅状态不可用时，App 会展示以下四种拦截页之一，每个页面都有按钮直达 Console（或切换 Team）：

界面提示	触发原因与处理办法
设备数量已达套餐上限	当前 Team 的设备名额已满。前往 Console 登出闲置设备或升级套餐。
电脑数量已达套餐上限	当前 Team 的电脑名额已满（新配对的电脑显示为未启用的 **超出套餐名额**）。前往 Console 删除或停用旧电脑，或升级套餐。
成员名额已满	你接受邀请时，该 Team 的成员席位已满。请联系 Team Owner 在 Console 腾出席位或升级套餐。
暂时无法接入这个团队	该 Team 没有生效套餐、当月 Credits 已用完、处于冻结状态，或你已不在该 Team 中。前往 Console 检查账单，或切换到账号下的其他 Team。

变更套餐、出账与取消规则

只有 **Team Owner** 可以购买、变更或取消套餐。在 Console 账单页中，所有订阅调整都通过 **变更套餐** 和 **取消订阅** 两个按钮完成：

- **自动续费与单次付款**：套餐默认按月自动续费，结账时也可以选择 **单次付款 · 不自动续费**。
- **提前 7 天出账与取消截止期**：自动续费订阅会在**续费日前 7 天**生成下一期账单。如果不想续期，必须在**续费日前至少 7 天**取消订阅。
- **升档与降级的生效时间**：
  - 符合即时生效条件的升档：**立即生效**；
  - 其余套餐变更以及所有**降级**：按提前 7 天截止规则处理，在**当前计费周期结束时**生效。

关于 3 天免费试用、兑换码和退款政策，请查看试用、兑换码与退款。

Credits 与用量

了解 Vibe42 按 64 MiB 折算 1 Credit 的双侧中转计费规则、为什么 P2P 直连完全免费，以及 Credits 用完后的表现。

vibe42.ai/zh-HANS/docs/account/credits-and-usage

当你想了解套餐里的 Credits 怎么扣费、为什么点对点（P2P）直连不扣 Credits，或者怎么查看并节省每月用量时，可以阅读本页。

Credits 是什么

Credits 是 Vibe42 计算中转网络流量的唯一单位。套餐按**资源月**向 Team 发放额度，每月按订阅起始日重置，当月没用完的余额**不会结转**到下个月（各套餐的每月额度见 `https://vibe42.ai/#pricing` 与 Console）。

计费口径与免费清单

只有经过 Vibe42 接入点（AP）中转的数据才会消耗 Credits。

- **换算标准**：`64 MiB = 1 Credit`。
- **中转双侧计费**：数据经过 AP 中转时，进入 AP 和离开 AP 两侧会各计算一次。
  - *示例*：你从手机或浏览器经过 AP 向电脑上的 Vibe42 后台服务传输 **64 MiB** 的文件或终端数据（`App → AP → 电脑`），其中 `App → AP` 一侧计 `64 MiB`（1 Credit），`AP → 电脑` 一侧计 `64 MiB`（1 Credit），**共消耗 2 Credits**。

以下流量**完全免费（0 Credits）**：

- **P2P 点对点直连流量**（原生 App、后台服务与 `v42` CLI 之间一旦建立直连，传输多少数据都不扣 Credits）
- 加密与底层传输协议开销
- 连接协商与 NAT 打洞（直连探测）数据包
- 控制连接（`control` 状态同步通道）
- Console 管理后台的接口请求

哪些功能在中转时会消耗 Credits

当连接走 AP 中转时，以下四类数据都按同一规则计入用量：

1. **终端会话**（纯文本输入输出，日常消耗极小）
2. **文件预览、上传与下载**（文件管理）
3. **远程桌面画面与键鼠操作**（远程桌面操控）
4. **CLI 端口转发**（`v42 forward`）

因为连续的视频画面和几百 MB 的大文件体积远大于终端文字，走中转时**远程桌面**和**大文件传输**消耗 Credits 最快。

如何确认当前走的是直连还是中转

- **在 App 内查看**：打开 **设置 → 网络诊断**，查看目标电脑的各个通道显示的是 **直连 QUIC / 直连 TCP**（免费直连）还是 **中继**（计费中转）。
- **在命令行查看**：运行 `v42 network [computer]`，查看 JSON 格式的实时连接路径（详见连接是怎么建立的与 CLI 命令参考）。

为什么浏览器与公司网络总是走中转

- **浏览器始终走中转**：受浏览器安全限制，网页端（`https://app.vibe42.ai`）无法建立点对点直连，**始终经过 AP 中转**。
- **企业防火墙与严格路由器**：只放行网页端口（TCP `443`）或使用严格 NAT（网络地址转换）的公司网、校园网和酒店网络往往会拦截点对点直连，使原生 App 留在中转路径上。
- **开启了强制中转**：检查设备或电脑上是否手动开启了 **强制中继** 开关。

Credits 用完后会发生什么

当 Team 的当月额度用完时，App 会显示 **本月 Credits 已用完**：

- **所有走 AP 中转的数据连接立即暂停**（包括网页端会话，以及需要中转的原生 App 会话）。
- **保留基础控制连接**，以便套餐续费或升级后客户端能马上恢复。
- **P2P 直连不受影响**：只要原生 App 或 CLI 能与电脑直连，依然可以正常使用。
- 到了每月重置日，中转功能会自动恢复；Team Owner 在 Console 续费或升级套餐后也会立即恢复。

在哪里查看当前用量

- **App 内**：打开 **设置 → 用量**，查看当月 Credits 的 **已用 / 总额**、**剩余** 和 **重置时间**。
- **Console**：Team 的 Owner 与 Admin 也可以在 Console 查看。

节省 Credits 的建议

- **优先用原生 App 或 CLI 代替浏览器**：macOS、Linux、iOS、Android 原生 App 在网络条件允许时会自动切换到免费的 P2P 直连。
- **中转用远程桌面时调低画质**：无法直连时选择较低的画质档位，用完及时关掉远程桌面画面。
- **大文件留到直连时再传**：先确认 **网络诊断** 里的 `file` 通道显示为 **直连 QUIC** 或 **直连 TCP**，再传输大文件。

试用、兑换码与退款

了解 3 天免费试用的资格与取消规则、Console 兑换码的四种权益、退款政策摘要，以及常见结账提示的含

义。

vibe42.ai/zh-HANS/docs/account/trial-redeem-refunds

当你要开启 3 天免费试用、为 Team 使用兑换码、确认某笔账单能否退款，或者在结账页遇到报错提示时，可以查看本页。

3 天免费试用

符合条件的 Team Owner 可以前往 `https://vibe42.ai/console/checkout?resource=access` 开启 **3 天免费试用**：

- **需要绑卡**：开启试用必须绑定有效银行卡，**每张卡只能试用一次**。
- **每个 Team 与 Owner 限用一次**：只要当前 Team 或它的 Owner 之前用过免费试用，或者有过任何付费记录，就不能再次开启试用。
- **到期自动扣款与提前转正**：如果不取消，3 天试用结束时会自动按所选套餐扣除首期费用；你也可以在试用期内随时主动付款，立即转为正式订阅。
- **随时取消试用**：试用期间可以随时在 Console 账单页取消。取消后，试用仍可继续用到 3 天期满，到期自动结束，不会扣款。
- **扣款失败处理**：如果试用期结束时自动扣款失败，试用会立即停止，Team 回到未开通状态。

兑换码

兑换码格式为 `XXXX-XXXX-XXXX`，在 `https://vibe42.ai/console/redeem?code=XXXX-XXXX-XXXX` 兑换。

- **仅限 Team Owner 兑换**：只有 Team 的 **所有者** 才能使用兑换码。如果在桌面 App 或网页端输入兑换码，会自动跳转到浏览器的 Console 页面完成确认。
- **四种权益类型**：不同活动的兑换码提供以下四种权益之一：
  1. **优惠订阅**：按折扣开通或续订套餐；
  2. **延长订阅**：延长当前生效套餐的有效期；
  3. **赠送套餐**：为尚未开通套餐的 Team 赠送一段使用期；
  4. **延长订阅** 或 **赠送套餐**：已有生效订阅就延长有效期，还没开通就直接赠送一段使用期。

退款政策摘要

所有退款申请都由人工审核处理。请用你的注册邮箱发邮件联系 `support@vibe42.ai`。审核通过后，我们会在 **3–5 个工作日**内向支付渠道提交退款。

- **首购未使用全额退款**：首次直接购买套餐后，在**首个计费周期开始的 3 天内**，且累计计费流量**低于 1 MB**（视为未使用），可以申请全额退款。
- **试用转正不适用首购退款**：自 **2026-10-27** 起开启的免费试用，到期转正扣款后**不享受**上述首购 3 天未使用全额退款。
- **计费错误与即时升档结算**：重复扣款、我方计费错误，以及符合条件的即时升档产生的“收新退旧”差额，均全额退还。

- **其他审核通过的退款**：不属于上述全额退款范围的其他申请，经审核同意退款的，会扣除**支付渠道手续费与 3% 人工处理费**。

> **警告**
>
> 未联系客服直接向银行或发卡机构发起拒付（chargeback），会导致账号被立即封禁。完整法律条款请查看 `https://vibe42.ai/refunds` 与 `https://vibe42.ai/terms`。

常见结账与试用报错提示

界面提示文案	原因与解决办法
此团队无法开始免费试用，请直接订阅。	该 Team 或它的 Owner 之前已经用过免费试用，或有过付费记录。请直接付费订阅。
此付款方式已用于过免费试用	这张银行卡之前已经用于免费试用（每张卡限试用一次）。请直接付费订阅或换一张卡。
付款方式绑定已过期 / 已取消绑定付款方式	绑卡验证页面超时，或你在完成前关闭了页面。重新发起结账即可。
正在确认付款结果。你可以稍后回来查看，请先不要发起另一笔付款。	支付渠道正在确认扣款结果。请稍等片刻再刷新 Console 查看，不要重复付款。
这项操作需要 Team 所有者完成，请联系对方处理套餐或付款。	你当前在该 Team 中的角色是 **管理员** 或 **成员**。请让 Team **所有者** 开启试用、变更套餐或兑换码（见套餐与名额）。

08 平台说明

macOS App

Vibe42 macOS App 的系统要求、LaunchAgent 后台服务、系统权限、内置 CLI 与更新说明。

vibe42.ai/zh-HANS/docs/platforms/macos

Vibe42 macOS App 既可以作为操控端连接你的其他机器，也可以把当前 Mac 设为被控电脑，从别的设备访问终端、Agent、文件与远程桌面。本文说明 macOS 的系统要求、后台服务行为、系统权限设置及更新方式。

系统要求与获取方式

- **操作系统**：macOS 14（Sonoma）或更高版本。
- **芯片架构**：为 Apple Silicon（`arm64`）与 Intel（`x86_64`）提供独立的 `.dmg` 安装包。下载页 vibe42.ai/#downloads 会自动识别你的芯片型号。
- **macOS 13（Ventura）**：桌面 App 需要 macOS 14+。macOS 13 请改用一行命令安装 Server 版（只有后台服务、没有桌面 App 的版本），同样可以使用远程终端、Agent、文件传输和 CLI（见 Linux 服务器与无头机器）。

安装与首次启动要点

1. 下载 `.dmg`，将 **Vibe42.app** 拖入 `Applications`。
2. 打开 App，点击 **登录** 跳转到系统浏览器登录账号，再点击 **批准**。完整步骤见安装 Vibe42。
3. 首次引导询问是否将本机作为被控电脑时，选择 **启用这台电脑**（如果只用这台 Mac 控制其他机器，选 **暂不启用**）。
4. 在 **System Settings → General → Login Items & Extensions** 中保持 Vibe42 的后台权限开启，让 Vibe42 后台服务保持在线。

> **重要**
>
> 如果 Mac 上已经装了 `Vibe42.app` 或旧版 `Vibe X.app`，请直接用应用内置的更新功能，不要重新下载 `.dmg` 覆盖安装。

macOS 特有的行为与设置

内置后台服务与 LaunchAgent 生命周期

把当前 Mac 设为被控电脑后，Vibe42 会在 `~/.v42`（旧版原地升级可能位于 `~/.vibex`）部署后台服务，并注册用户级 LaunchAgent `ai.vibecafe.v42.companion`（macOS 的登录自启后台项）。

关闭主窗口或按 `⌘Q` 只会退出桌面窗口，**不会停止后台服务**。你的手机、平板或其他电脑依然可以随时连入终端与远程桌面。要同时停止桌面窗口和后台服务，请从状态栏图标选择 **停止服务并退出**，或在 App 的 **设置 → 网络诊断** 中点击 **完全退出**。你还可以在设置中开关 **登录时自动启动**，控制登录 macOS 时是否自动打开桌面窗口。

与内置 `v42` CLI 共用身份

macOS App 在 `~/.v42/bin` 中内置了 `v42` 与 `v42-companion` 命令行工具。内置 CLI 与桌面 App 共用同一个安装目录和本地 Ed25519 密钥身份。只要桌面 App 已登录，你在本机终端就能直接运行 `v42`，无需再次登录，也不额外占用设备名额。详见 v42 CLI 概览。

远程桌面专属设置页

设置 → 远程桌面 设置页仅在 macOS App 中提供。你可以在这里开关 **允许其他设备远程控制这台 Mac**，并按提示授予系统 **屏幕录制** 与 **辅助功能** 权限。详见远程桌面：开始使用。

桌面系统通知

macOS App 只在桌面窗口运行期间弹出系统通知；窗口退出期间发生的事件不会在下次打开时补发。如果需要在退出窗口后继续收到提醒，请配对 iPhone 或 Android 设备，或绑定 Telegram（见通知设置）。

更新方式

macOS App 会在后台自动检查更新，新版本就绪后提示 **重启以更新到 …**。

每次更新均通过 HTTPS 下载，安装前校验 SHA-256 哈希与 Ed25519 数字签名，并在更新后检查后台服务运行状态。完整的更新与卸载步骤（包括用 `v42-companion uninstall --purge` 清理 `~/.v42`）见更新与卸载。

限制与已知行为

- **屏幕录制定期复核（macOS 15+）**：从 macOS 15 起，系统会定期在 Mac 本机弹窗，要求再次确认后台录屏应用的 **屏幕录制** 权限。
- **重启后未登录界面**：`ai.vibecafe.v42.companion` 是用户级 LaunchAgent，Mac 完全重启后需要先登录一次系统账号，后台服务才会启动上线。已登录账号进入锁屏后可正常远程输入密码解锁（见解锁屏幕与打开 App）。
- **休眠与合盖断网**：Mac 进入系统休眠后网络连接会暂停，唤醒后自动恢复。若将笔记本合盖作为常驻被控机使用，请接通电源并在系统电池或节能设置中关闭自动休眠。

日志位置

查找本机 App 与后台服务日志文件及反馈问题前的脱敏方法，见日志与诊断。

Linux 桌面 App

Linux 桌面 AppImage 的系统要求、systemd 用户服务管理，以及托盘、文件预览、输入法和硬件编解码可选依赖说明。

vibe42.ai/zh-HANS/docs/platforms/linux-desktop

Vibe42 Linux 桌面 App 以 AppImage（单文件可执行包）形式分发，提供 `x86_64` 与 `arm64` 两种架构，既是带 GPU 加速的原生操控端，也会一并管理本机的 Vibe42 后台服务。本文说明系统 `glibc` 与 GTK 版本要求，配置系统托盘、沙箱文件预览、X11 输入法与硬件编解码等可选系统包的方法，以及更新与日志机制。

系统要求与获取方式

- **处理器架构**：`x86_64` 或 `arm64`（aarch64）。
- **发行版基线**：基于 Ubuntu 24.04 构建，要求 `glibc 2.39+`、`GTK 4.14+` 与 `libadwaita 1.5+`。支持 Ubuntu 24.04 LTS、Debian 13、Fedora 40 及更新版本。
- **运行环境要求**：需要 FUSE（`fuse3`，Linux 挂载文件系统的组件；若环境不支持 FUSE 可加 `--appimage-extract-and-run` 参数启动）以及正常运行的 `systemd` 用户会话（user session）。
- **下载地址**：前往 vibe42.ai/#downloads 下载 `Vibe42-<version>-x86_64-unknown-linux-gnu.AppImage`（arm64 机器为 `Vibe42-<version>-aarch64-unknown-linux-gnu.AppImage`）及同名的 `.sha256` 校验文件。

如果你的 Linux 机器运行 `glibc` 较旧的发行版（如 Ubuntu 22.04、Debian 12），无法运行桌面 AppImage。请改用一行命令安装 Server 版将其作为被控电脑接入（见 Linux 服务器与无头机器），并通过浏览器使用 Web 应用操控。

安装与首次启动要点

1. 下载后先校验 SHA-256，再加上可执行权限并启动：

```
sha256sum -c Vibe42-*.AppImage.sha256
chmod +x Vibe42-*.AppImage
./Vibe42-*.AppImage
```

2. 首次启动时，App 会在 `~/.local/share` 下自动注册 `.desktop` 桌面入口、应用图标以及 `vibex://` 协议处理器（用于接收浏览器登录回调）。如果之后把 AppImage 文件移到别的目录，在新位置启动一次就会自动修正快捷方式路径。完整的登录与引导步骤见安装 Vibe42。

Linux 桌面特有的行为与可选依赖

接管并同步后台服务

在 `systemd` 用户会话下启动时，Linux 桌面 App 会把后台服务部署到 `~/.v42/`，并通过 `v42-companion.service` 用户服务运行——路径和服务名与一行命令安装完全一致。如果这台机器此前已经跑过一行命令安装，桌面 App 启动后会直接接管并负责后续更新；若在非 `systemd` 环境下启动 AppImage，则仅作为操控

端运行，不管理后台服务。

可选系统依赖对照表

部分桌面功能依赖宿主系统安装的软件包：

功能模块	宿主系统依赖	缺失时的回退表现
系统托盘常驻	`StatusNotifierItem` 支持（GNOME 桌面需安装 AppIndicator 扩展）	没有托盘图标，关闭主窗口即直接退出桌面 App
富格式文件预览（代码高亮、Markdown、HTML、PDF）	`bubblewrap` 与 `xdg-dbus-proxy`（内置 WebKitGTK 沙箱必需）	代码、Markdown 与 HTML 回退为纯文本显示；PDF 仅提供下载
X11 下的中文 / CJK 输入法	宿主系统的 `fcitx5` 或 `ibus` GTK4 输入法模块（Wayland 原生支持，无需额外模块）	X11 会话下无法在窗口内呼出输入法组字
远程桌面解码（作为操控端）	VA-API 或 NVIDIA NVDEC 硬件解码驱动	按需从 Cisco 下载并校验 OpenH264 软件解码
远程桌面编码（作为被控端）	系统 `ffmpeg`（带 VA-API / NVENC / V4L2 硬编）	在 App 设置中开启或执行 `v42-companion codec enable` 下载 OpenH264

更新方式

Linux 桌面 App 会在后台自动检查更新；在支持系统托盘的桌面上，也可以点击托盘菜单中的 **检查更新** 或 **重启以更新到 …**。校验通过的新版 AppImage 会暂存在本地，下次启动时自动替换，并同步更新 `v42-companion.service`。完整的卸载流程（包括清理 `~/.local/share` 桌面文件与执行 `v42-companion uninstall`）见更新与卸载。

限制与已知问题

- **Wayland 远程桌面支持范围**：作为被控端时，Wayland 会话下支持 KDE Plasma 6、GNOME 与 Hyprland（KDE 和 GNOME 首次被连需在被控机本地弹窗中授权，见远程桌面：开始使用）；其他 Wayland 合成器暂不支持被控。
- **无 FUSE 环境启动**：若系统未安装 `fuse3` 或未开放 `/dev/fuse`，请执行 `./Vibe42-*.AppImage --appimage-extract-and-run` 启动。
- **无托盘时的关窗行为**：在未安装 AppIndicator 扩展的默认 GNOME 桌面上，关闭主窗口会直接退出桌面界面（但 `v42-companion.service` 仍会在 `systemd` 用户会话中继续后台运行）。

日志位置

使用 `v42-companion logs -f` 查看后台服务日志以及定位桌面客户端日志目录的方法，见日志与诊断。

Linux 服务器与无头机器

在 Linux 服务器、容器、ARM 开发板及 macOS 13 上通过一行命令安装与管理 Vibe42 Server 版。

vibe42.ai/zh-HANS/docs/platforms/linux-server

Vibe42 Server 版（只有后台服务、没有桌面 App 的版本）通过一行 shell 命令即可安装 Vibe42 后台服务与 `v42` CLI，不依赖任何图形桌面库或特定的 `glibc` 版本。当你要接入云服务器、家庭主机、Docker 或开发容器、ARM64 开发板、Windows 上的 WSL、macOS 13 机器，或配置 `systemd` 与 `supervisor` 启动模式时，请阅读本页。

系统要求与一行命令安装

- **支持平台**：
  - Linux `x86_64` 与 `arm64`：采用静态链接构建，可在任意 Linux 发行版（Ubuntu、Debian、Alpine、RHEL/Rocky、Arch、NixOS 及各类容器镜像）上直接运行。
  - macOS（`arm64` 与 `x86_64`），包含无法安装桌面 App 的 macOS 13。
- **前置工具**：`curl`、`tar` 以及 `tmux` 3.0 或更高版本。
- **安装命令**：

```
curl -fsSL https://get.vibe42.app/install.sh | sh
```

> **重要**
>
> 请用日常拥有代码目录与终端会话的操作系统普通用户执行该命令——除非你的工作环境本身就是纯 `root` 容器，否则不要加 `sudo` 或用 `root` 运行。已经安装了 Vibe42 桌面 App 的机器不要再运行一行安装脚本（脚本检测到桌面 App 后会拒绝执行）。

首次安装与配对要点

安装脚本会先校验下载归档包的 SHA-256 哈希，将程序安装到 `~/.v42`（旧版安装可能位于 `~/.vibex`），并自动把 `~/.v42/bin` 写入 shell `PATH`。目录下同时提供 `v42-companion`（兼容保留旧名 `vibex-companion`）与 `v42` CLI。

安装末尾终端会打印 8 位配对码（如 `Pairing code: ABCD-EFGH`）和审批链接（`https://vibe42.ai/console/pair?code=ABCD-EFGH`），并等待最多 10 分钟。你可以在浏览器打开该链接批准，也可以在任意已登录 Vibe42 的设备中打开 **电脑与设备 → 连接新的电脑/服务器** 输入配对码。完整引导流程见安装 Vibe42与连接一台 Linux 服务器。

服务守护模式与日常运维

`systemd` 模式与 `supervisor` 模式

安装脚本会自动探测宿主环境，你也可以通过环境变量 `VIBEX_SERVICE_MODE=systemd|supervisor|auto` 手动指定：

- **`systemd` 模式**（具备 systemd 用户会话的主机默认启用）：注册并启动用户级单元 `v42-companion.service` 以及健康检查定时器 `v42-watchdog.timer`。
- **`supervisor` 模式**（无 systemd 的环境自动启用，例如大多数容器）：通过轻量用户态守护进程运行，并在你的 shell rc 配置文件中写入自启动块，每次打开交互式 shell 时确保后台服务在线（卸载时会自动清理该配置块）。可传 `VIBEX_SERVICE_MODE=supervisor` 强制使用此模式。

让纯 SSH 服务器在退出登录后保持在线

在不少 Linux 服务器上，当用户的最后一个 SSH 连接断开时，`systemd` 默认会停止该用户的 `systemd --user` 实例，导致后台服务随之离线。执行一次以下命令开启用户驻留（linger），即可让服务在退出 SSH 后保持在线，并在机器重启后自动启动：

```
sudo loginctl enable-linger "$USER"
```

常用管理命令

```
~/.v42/bin/v42-companion status
~/.v42/bin/v42-companion status --json
~/.v42/bin/v42-companion logs -f
~/.v42/bin/v42-companion start
~/.v42/bin/v42-companion stop
~/.v42/bin/v42-companion update --check
```

Windows 上的 WSL

在 Windows PC 上，可以在 WSL 里运行同一条一行安装命令把机器接入为被控电脑（需 WSL 2 与一个 Linux 发行版，例如 Ubuntu，且发行版里装有 `curl`、`tar`、`tmux` 3.0+；如果发行版没有启用 systemd 用户会话，安装器会自动切换到 supervisor 模式）。这台电脑只在 WSL 运行期间在线：Windows 关机、休眠，或 WSL 被停止（例如 `wsl --shutdown`）时，它在 Vibe42 里显示离线；重新打开 WSL 终端后后台服务会自动恢复（systemd 用户会话或 supervisor 自启动）。详见 Windows。

更新机制与原地修复

自管理的 Server 版后台服务会在每天服务器本地时间 **07:00** 前后自动检查更新。你也可以随时执行 `v42-companion update [--check] [--json]` 手动更新，或者直接重新运行 `curl -fsSL https://get.vibe42.app/install.sh | sh` ——重跑安装脚本会原地升级或修复，完整保留当前机器身份、配对关系与电脑名额。

每次更新按六步事务执行：**下载 → 签名校验 → 预检（磁盘空间、终端运行环境与正在进行的传输）→ 等待空闲 → 原子切换 → 健康检查**。只要当前仍有终端连着或有文件正在传输，后台服务就不会自行重启打断你的任务。更新不提供自动回滚。关于 `uninstall` 与 `uninstall --purge` 的区别见更新与卸载。

限制与注意事项

- **无头环境与远程桌面**：没有运行 X11 或受支持 Wayland 桌面会话的无头服务器不支持远程桌面；远程终端、Agent 状态通知、文件传输、终端隔离副本以及 `v42 forward` 端口转发均正常可用。
- **容器冷启动**：在容器内使用 `supervisor` 模式时，如果容器启动后没有运行交互式登录 shell 触发 rc 自启动，请在容器的 entrypoint 启动脚本中调用 `~/.v42/bin/v42-companion start`。

日志位置

执行 `~/.v42/bin/v42-companion logs -f` 即可实时查看运行日志；日志文件路径与发送前的脱敏检查清单见日志与诊断。

---

iPhone 与 iPad

在 iOS 与 iPadOS 17+ 上配对使用 Vibe42，了解零后台驻留与 APNs 推送、iPad 多窗口与外接键盘支持及系统限制。

`vibe42.ai/zh-HANS/docs/platforms/ios`

iPhone 与 iPad 版 Vibe42 提供基于 Metal 硬件加速的原生终端、Agent 状态推送、文件浏览与远程桌面，切入后台时不保留常驻连接、不额外耗电。本文说明 iOS 与 iPadOS 的系统要求、扫码配对、iPad 多窗口与外接键盘特性及平台限制。

系统要求与获取方式

- **操作系统**：iOS 17 或更高版本 / iPadOS 17 或更高版本。
- **获取方式**：从 App Store 下载安装：apps.apple.com/us/app/vibe42/id6792229966。
- **设备角色**：iPhone 与 iPad 仅作为操控端的**设备**（Device），用于连接和操控已配对的 Mac 与 Linux 电脑。

配对接入与系统权限

iOS 版 Vibe42 不要求你在手机键盘上输入账号密码：

1. 在任意已登录的设备（桌面 App 或浏览器打开 `https://app.vibe42.ai`）中进入 **电脑与设备 → 下载与安装 → 配对手机/平板**。
2. 在 iPhone 或 iPad 上点击 **扫码配对** 扫描二维码，或手动输入 8 位配对码。
3. 回到原设备核对并批准请求。

如果你手头只有手机，也可以先用手机浏览器登录 Web 应用，再用它给手机 App 配对，详见配对设备并开始第一个会话。

系统权限按需申请：

- **相机权限**：仅在点击扫码配对或在终端内拍照上传时请求。
- **通知权限**：在配对完成后请求，用于接收 Agent 状态与电脑事件通知。

iOS 与 iPadOS 特有的行为

零后台驻留与 APNs 系统推送

Vibe42 在 iOS 与 iPadOS 上采用**零后台驻留**设计。一旦你锁屏或切换到其他应用，Vibe42 会立即断开网络连接，不会在后台保活耗电。你的 shell 与 Agent 会继续在远端电脑上运行；当 Agent 状态变为 **已完成**、**需要处理** 或 **出错** 时，通知经由 Apple APNs 推送到锁屏与通知中心，点击通知即可重连并直达对应电脑的窗口。详见会话持久化与通知设置。

GPU 终端渲染与触屏输入

终端使用 `libghostty-vt` 与 Metal GPU 渲染，自动匹配屏幕刷新率。针对触屏操作提供了 **发送到终端** 命令栏、可拖动的方向键（D-pad）、单击生效或双击锁定的修饰键、双指捏合调节字号，以及跨设备同步的自定义快捷按钮。详见移动端输入。

iPad 多窗口与外接物理键盘

在 iPadOS 上，Vibe42 适配了系统多任务与外接键盘：

- **系统级多窗口**：支持通过分屏浏览（Split View）或台前调度（Stage Manager）并排打开多个 Vibe42 窗口，例如左边看 Claude Code 终端、右边开远程桌面窗口（在新窗口打开 仅在 iPad 提供，iPhone 上不提供）。
- **外接键盘与快捷键速查**：连接妙控键盘（Magic Keyboard）或蓝牙键盘后支持完整物理按键与 Kitty 键盘协议；长按外接键盘上的 `⌘` 键即可呼出当前可用的快捷键速查面板（见快捷键）。

更新方式

通过 App Store 自动或手动更新。当过旧的客户端版本低于服务端允许的最低版本要求时，App 会提示你前往 App Store 更新。如需解绑旧手机并释放设备名额，可在 Console 或任意其他已登录设备中将其移除，详见更新与卸载。

限制与注意事项

- **App 内不提供价格与购买界面**：iPhone 与 iPad App 内不展示套餐价格、结账购买或订阅管理入口。开启试用、订阅变更、名额管理与查看 Credits 请在浏览器中访问 Console（见套餐与名额）。
- **iPhone 不支持多窗口**：在新窗口打开 属于 iPadOS 多任务特性，在 iPhone 上不显示。

日志与崩溃报告

常规网络诊断与排障流程见日志与诊断。如需导出系统级闪退报告发给支持团队，可在 iPhone 或 iPad 的系统 **设置 → 隐私与安全性 → 分析与改进 → 分析数据** 中查找对应的崩溃日志。

Android

在 Android 11+ 上配对 Vibe42，零后台驻留经 FCM 接收推送，适配折叠屏与平板布局，并用 adb 收集崩溃日志。

vibe42.ai/zh-HANS/docs/platforms/android

Android 版 Vibe42 把你的手机、折叠屏或平板变成操控 Mac 与 Linux 电脑的设备，提供 GPU 渲染的终端、带推送的 Agent 状态、文件浏览与远程桌面。本文说明 Android 的系统要求、配对与权限、平台特有行为、更新方式与日志收集方法。

系统要求与获取方式

- **操作系统**：Android 11 或更高版本。
- **获取方式**：从 Google Play 安装：play.google.com/store/apps/details?id=ai.vibecafe.x；下载页 vibe42.ai/#downloads 也会指向这里。
- **设备角色**：Android 设备只作为操控端的**设备**（Device），不能运行 Vibe42 后台服务，也不能作为被控电脑。

配对与权限

Android App 不会要求你输入账号密码，统一从已登录的设备上配对：

1. 在桌面 App 或 `https://app.vibe42.ai` 中打开 **电脑与设备 → 下载与安装 → 配对手机/平板**，界面会显示二维码与 8 位配对码。
2. 在手机上点击 **扫码配对** 扫码，或手动输入配对码。
3. 回到原设备，核对请求信息并批准。

配对码 10 分钟内有效、只能用一次、每次配对一台设备。如果手边只有手机，可以先在手机浏览器登录 `https://app.vibe42.ai`，再用这个网页会话给 App 配对。详见配对设备并开始第一个会话。

系统权限按需申请：相机权限只在扫描配对二维码或在终端里拍照上传时请求；通知权限在配对完成后请求，用于接收推送。

Android 特有的行为

零后台驻留与 FCM 推送

App 不保留后台连接。切到其他应用或锁屏后，App 会主动断开连接；你的 shell 与 Agent 继续在电脑上运行。Agent 状态变为 **已完成**、**需要处理** 或 **出错** 时，通知经 Firebase Cloud Messaging（FCM）推送到手机，点击通知即可重连并直达对应电脑的窗口；切回前台后几秒内自动恢复连接。详见通知与会话、窗口与 Recents。

渲染与输入

终端使用 `libghostty-vt` 在 `TextureView` 上渲染，自动匹配屏幕刷新率。触控输入与 iOS 版一致：命令栏（**发送到终端**）、可拖动的方向键（D-pad）、单击生效或双击锁定的修饰键、长按选词、双指捏合调字号、三种软键盘避让模式，以及跨设备同步的自定义快捷按钮。详见在手机和平板上输入。

拍照、相册或文件选择器选中的文件会上传到电脑并把路径插入终端；超过 4 MiB 的图片与 HEIC 照片会自动转为 JPEG。

折叠屏、平板与桌面模式

布局自适应折叠屏内屏与平板宽度。**在新窗口打开** 只在 Android 桌面模式下提供，普通手机布局不提供。外接键盘时，适用键盘快捷键中 **other** 一列的键位。

连接与 Credits

网络允许时，Android App 会与电脑建立点对点直连，否则经最近的中转节点转发。直连不计费，经中转的流量计入 Team 的 Credits。当前连接路径可在 **设置 → 网络诊断** 查看。详见连接是怎么建立的与 Credits 与用量。

更新方式

更新通过 Google Play 分发。当客户端低于服务端允许的最低版本时，App 会提示 **本版本不受支持，请前往 vibe42.ai 下载最新版本。**，更新后才能继续连接。要把一部手机从 Team 中移除并释放设备名额，请在 Console 或其他已登录设备上将它登出。详见更新与卸载。

限制

- **App 内不提供价格与购买**：Android App 不展示套餐价格、购买与订阅管理，请在浏览器中使用 Console（见套餐与名额）。
- **只能作为操控端**：Android 不能运行终端，也不能被其他设备连接。
- **多窗口**：独立应用窗口仅限桌面模式。

日志与崩溃报告

App 日志保存在应用私有存储中，无法从 App 内导出。联系支持时请记录发生时间、设置页显示的 App 版本并附上截图。如果 App 崩溃且手机可以连接电脑，开启 USB 调试后运行以下命令过滤崩溃日志：

```
adb logcat -s VibexPanic:*
```

完整清单（包括发送前需遮盖的敏感信息）见收集日志。

Web 应用与 PWA

在任意现代浏览器打开 app.vibe42.ai 使用 Vibe42，安装为 PWA，并了解中转连接、浏览器快捷键与 WebCodecs 要求。

vibe42.ai/zh-HANS/docs/platforms/web

Web 应用 `https://app.vibe42.ai` 让你在任何装有现代浏览器的电脑上使用终端、查看 Agent 状态、管理文件与连接远程桌面，包括 Windows 电脑和临时借用的笔记本，无需安装客户端。本文说明如何在浏览器中使用 Vibe42、如何把它安装为 PWA（可以安装到桌面或主屏幕的网页应用），以及它与原生 App 的区别。

要求与访问方式

- **浏览器**：任意现代浏览器。终端使用 WebGL 渲染，WebGL 不可用时自动回退到 Canvas 2D；远程桌面需要浏览器支持 **WebCodecs**。
- **登录**：打开 `https://app.vibe42.ai`，使用 `vibe42.ai` 账号登录（邮箱加密码，或 **使用 GitHub 继续** / **使用 Google 继续**）。未验证邮箱的账号无法登录。详见账号与登录。
- **设备名额**：每个登录的浏览器占 Team 的 1 个设备名额；同一浏览器重新登录不会重复占用；登录 Console 不占名额。详见电脑、设备与 Team。

Web 应用仅作为操控端，你的 Mac 或 Linux 电脑仍需安装 Vibe42，见安装到电脑。

安装为 PWA

使用浏览器自带的安装选项（例如 **Install app** 或“添加到主屏幕”）即可把 Web 应用安装成带独立窗口和图标的 PWA。PWA 的行为与浏览器标签页完全一致：共用登录状态、共用同一个设备名额，同样走中转连接。

手机上还没装原生 App 时，也可以先用手机浏览器登录 `https://app.vibe42.ai`，再在该页面打开 **电脑与设备 → 下载与安装 → 配对手机/平板** 给原生 App 配对。详见配对设备并开始第一个会话。

浏览器特有的行为

永远走中转

浏览器无法建立点对点直连，因此每个 Web 会话都经最近的中转节点转发。经中转传输的终端、文件与远程桌面流量计入 Team 的 Credits；原生 App 建立直连后则不消耗 Credits。如果要传输大量数据或长时间使用远程桌面，建议优先使用原生 App。详见连接是怎么建立的与 Credits 与用量。

终端引擎

Web 终端运行与原生 App 相同的 Ghostty 内核，编译为 WebAssembly（`ghostty-vt`）并用 WebGL 渲染。终端内图片采用 Kitty 图形协议，浏览器中每个分屏（pane）最多显示 5 张。输入法组字状态会先在终端内显示，确认后再提交。详见终端引擎。

键盘快捷键

浏览器占用了部分系统组合键，因此 Web 应用改用替代按键：例如 **新窗口** 在 Apple 平台是 `⌥⌘N`、其他平台是 `Alt+Shift+N`；**新建终端** 是 `⌥⌘T` 或 `Alt+Shift+T`。Firefox 占用了 `Ctrl+Shift+P`，因此 Firefox 中 **全局搜索** 没有快捷键。完整键位表见键盘快捷键。

通知

标签页或 PWA 窗口打开期间，Web 应用在页面内弹出提示；关闭窗口后不会收到通知。离开电脑时如需接收提醒，请配对手机或在 **通知 → 消息通道** 绑定 Telegram，见通知。

远程桌面

浏览器中的远程桌面需要 WebCodecs 支持。画质档位、显示器切换与纯文本剪贴板同步均与原生 App 一致。详见远程桌面：开始使用。

更新方式

刷新页面即可加载最新版本。如果已打开的标签页版本过旧，界面会提示 **当前版本已过期，请刷新页面获取新版本。**，刷新页面即可继续使用。

限制

- **没有直连**：所有会话均走中转并计入 Credits。
- **没有端口转发**：`v42 forward` 是 CLI 功能，Web 应用不支持端口转发。见预览产物与端口转发。
- **受浏览器剪贴板权限约束**：远端程序通过 OSC 52（终端程序写入剪贴板的控制序列）发起的复制会写入当前浏览器的剪贴板，受浏览器剪贴板权限限制。
- **音视频预览整文件加载**：文件中的音视频预览在浏览器里按整文件读取，不像原生 App 那样按区间分段读取。

日志与诊断

打开浏览器开发者工具，在 **Console** 面板查看报错，在 **Network** 面板查看失败的请求或 WebSocket 连接，并配合 App 内的 **设置 → 网络诊断** 排查。详见收集日志。

Windows

Windows PC 可以通过 WSL 安装 Vibe42 Server 版成为被控电脑，原生版本近期上线；同时仍可用浏览器通过 Web 应用操控其他电脑。

`vibe42.ai/zh-HANS/docs/platforms/windows`

Windows PC 可以通过在 WSL（Windows Subsystem for Linux）里安装 Vibe42 Server 版成为被控电脑，Windows 原生版本将在近期上线；同时你仍可以在 Windows 的浏览器里通过 Web 应用操控其他电脑。

通过 WSL 把 Windows PC 变成被控电脑

在 WSL 2 里安装 Vibe42 后台服务的做法与 Linux 服务器完全一样：

1. **安装 WSL 2 与 Linux 发行版**（例如 Ubuntu）：在 Windows 侧的 PowerShell 或命令提示符中运行 `wsl --install`。
2. **打开发行版终端**，确认发行版里已装有 `curl`、`tar` 和 `tmux` 3.0+。
3. **运行一行安装命令**：

   ```
   curl -fsSL https://get.vibe42.app/install.sh | sh
   ```

   如果发行版没有启用 systemd 用户会话，安装器会自动切换到 supervisor 模式。
4. **批准配对码**：在浏览器中打开终端打印出的审批链接，或在任意已登录设备的 **电脑与设备 → 连接新的电脑/服务器** 输入配对码。
5. **在 App 里连接使用**：审批通过后即可在 App 里看到这台电脑并开始第一个会话。完整安装与配对细节见安装到电脑和一条命令接入 Linux 服务器。

能做什么、还不能做什么

通过 WSL 能得到什么：

- **终端、coding agent 与文件**：访问 WSL 环境里的持久终端、coding agent 和文件，与其他 Linux 被控电脑一致。Windows 的磁盘在 WSL 里位于 `/mnt/c` 这类路径下，可以在终端和文件面板里访问。
- **端口转发与 CLI**：使用 `v42 forward` 转发 WSL 里的服务（见预览与端口转发）；`v42` CLI 随 Server 版装在 WSL 里（不是 Windows 原生程序，见 CLI 概览）。
- **Agent 状态与推送通知**：Agent 状态跟踪与推送通知与其他 Linux 被控电脑一致。
- **在线状态**：这台电脑只在 WSL 运行期间在线。Windows 关机、休眠，或 WSL 被停止（例如 `wsl --shutdown`）时，它在 Vibe42 里显示离线；重新打开 WSL 终端后后台服务会自动恢复（systemd 用户会话或 supervisor 自启动）。

Windows 上目前还没有的：

- **不支持远程桌面控制 Windows 屏幕**：不能用远程桌面查看或操作 Windows 的屏幕（远程桌面的被控端只支持 macOS 与 Linux 桌面）。
- **尚无 Windows 原生 App**：从 Windows 这台设备去连别的电脑仍然用浏览器里的 Web 应用（始终走中转）。

在 Windows 上能做什么

要从 Windows 操控其他电脑，在任意现代浏览器打开 `https://app.vibe42.ai` 并登录，即可使用完整的 Web 应用：操作 Mac 与 Linux 电脑上的终端和 Agent、查看 Agent 状态、浏览文件、使用远程桌面（远程桌面需要浏览器支持 WebCodecs）。你还可以把 Web 应用安装为 PWA（带独立窗口的网页应用）。Web 应用与 PWA 的所有说明均适用，主要包括：

- 浏览器会话始终经中转节点转发，流量计入 Team 的 Credits（见 Credits 与用量）。
- 快捷键采用键盘快捷键中 **other** 一列的键位，并使用 Web 替代按键，例如 **新窗口** 为 `Alt+Shift+N`。
- 每个登录的浏览器占 1 个设备名额。

原生 Windows 版本

Windows 原生版本（含桌面 App）将在近期上线。完整的平台支持情况见 Vibe42 是什么。

更新与卸载

各平台 Vibe42 的更新方式、更新时的签名校验与健康检查、版本不受支持提示的含义，以及如何干净地卸载。

vibe42.ai/zh-HANS/docs/platforms/updates-and-uninstall

当你想了解 Vibe42 的 App 或 Vibe42 后台服务如何更新、客户端提示版本不受支持时怎么处理，或者想从机器上卸载 Vibe42 并释放名额时，请阅读本页。

各平台的更新方式

平台	更新方式
macOS App	自动检查更新，新版本就绪后提示 **重启以更新到 …**。
Linux 桌面 App	自动检查更新；托盘菜单也提供 **检查更新** 与 **重启以更新到 …**。校验通过的新版本先暂存，下次启动时替换，所管理的后台服务同步更新。
Server 版（一行命令安装）	每天本地时间 **07:00** 前后自动检查。手动更新可执行 `v42-companion update`，或重跑安装命令（原地升级或修复，保留配对）。
iPhone 与 iPad	App Store。
Android	Google Play。
Web 应用	刷新页面。

```
~/.v42/bin/v42-companion update --check
~/.v42/bin/v42-companion update
~/.v42/bin/v42-companion update --json
```

`v42` CLI 与随它一起安装的后台服务共用同一份安装目录与版本号，更新 App 或后台服务时 CLI 会一并更新。

更新时会校验什么

桌面 App 的整包更新只通过 HTTPS 下载，安装前校验 SHA-256 摘要与 Ed25519 签名；替换完成后，内置的后台服务必须通过健康检查。

Server 版的更新按单个事务执行：

1. 下载
2. 验签
3. 预检：可用磁盘空间、终端运行环境，以及是否有终端或文件正在传输
4. 等待后台服务空闲

5. 原子切换到新程序文件
6. 健康检查

有终端正在连接或有文件正在传输时，后台服务会等待空闲，不会直接重启。更新不提供自动回滚：如果更新后后台服务异常，请重跑安装命令修复，或参考连接问题。

更新不会影响正在运行的终端会话及其内容；修复或升级后的安装会保留原有身份、配对与名额。

“This version is no longer supported”

旧版本客户端被拒绝连接是服务端版本策略，不是网络故障。提示文案因客户端而异：

位置	提示	处理
桌面与手机 App	**本版本不受支持，请前往 vibe42.ai 下载最新版本。**	从 vibe42.ai/#downloads、App Store 或 Google Play 更新。
Web 应用	**当前版本已过期，请刷新页面获取新版本。**	刷新页面。
CLI 与后台服务	**请升级 Vibe42 CLI 后再继续。**	运行 `v42-companion update`，或重跑一行安装命令。

卸载

卸载包含两件独立的事：从机器上移除软件，以及在 Console 中把这台电脑或设备从 Team 移除以释放名额。停用一台机器时请完成这两步。

macOS App

1. 选择 **完全退出** 停止后台服务（仅关闭窗口或按 `⌘Q` 不会停止后台服务）。
2. 从 `Applications` 删除 `Vibe42.app`。
3. 如需一并删除后台服务的身份、配对与 `~/.v42`（旧安装为 `~/.vibex`）下的本地数据，运行下方命令并按提示确认。
4. 在 Console 的 **电脑** 中删除这台电脑，并在 **已授权设备** 中登出该 App。

```
~/.v42/bin/v42-companion uninstall --purge
```

Linux 桌面 App

1. 退出 App。
2. 删除 AppImage 以及它在 `~/.local/share` 下注册的桌面文件（`.desktop` 入口、图标与 `vibex://` 处理器）。
3. 执行 `v42-companion uninstall` 移除后台服务；加上 `--purge` 会一并删除身份、配对与本地数据。
4. 在 Console 删除这台电脑并登出设备。

```
~/.v42/bin/v42-companion uninstall --purge
```

Server 版

以安装时的系统用户身份运行卸载命令。不加 `--purge` 只移除服务与程序文件，保留账号关联与本地数据，之后重装会恢复为同一台电脑；加 `--purge` 则删除身份、配对与全部本地数据。`--purge` 会要求确认，在脚本中执行可加 `--yes`。

```
~/.v42/bin/v42-companion uninstall
~/.v42/bin/v42-companion uninstall --purge
~/.v42/bin/v42-companion uninstall --purge --yes
```

supervisor 模式下，卸载会一并清理写入 shell rc 文件的自启动块；systemd 模式下会移除 `v42-companion.service` 与 `v42-watchdog.timer`。

iPhone、iPad 与 Android

在手机或平板上直接删除 App，然后在 Console 的 **已授权设备**（或其他已登录设备上）登出这台设备，释放设备名额。

Web 应用

在 Web 应用内登出，或在 Console 中登出该浏览器，即可释放设备名额。已安装的 PWA（网页应用）按浏览器的常规方式卸载。

保留什么，删除什么

- **本地终端会话**：卸载后台服务不会关闭正在运行的终端会话及其中的进程，需要清理请自行关闭。
- **不带 `--purge` 的 `uninstall`**：保留账号关联与本地数据，重装后仍是同一台电脑。
- **`uninstall --purge`**：删除身份、配对与全部本地 Vibe42 数据，重装后需要作为新电脑重新配对。
- **在 Console 删除电脑**：从 Team 中移除这台电脑及其云端 Workspace 记录与备忘，机器上的文件与目录不受影响。装有桌面 App 的 Mac 删除电脑只释放电脑名额，要释放设备名额还需在 **已授权设备** 页登出该 App。详见电脑、设备与 Team。
- **登出设备**：该设备需要重新登录或重新配对才能再次使用。

09 # v42 CLI

v42 CLI 概览

了解 v42 CLI 的用途、安装位置、浏览器登录方式、退出码与 JSON 输出格式，以及脚本和 Agent 常用的全局选项。

vibe42.ai/zh-HANS/docs/cli/overview

`v42` 是 Vibe42 为脚本、CI 任务和 coding agent 提供的命令行工具。它连接的电脑、Workspace 和终端与 App 完全一致，默认输出 JSON 并返回固定的退出码（exit code）。在编写自动化脚本之前，或者想了解 `v42` 安装在哪里、怎么登录时，可以先看本页。

它负责什么

CLI 涵盖不需要图形界面的核心操作：新建和连接终端、启动 Agent、复制和同步文件、转发端口、查看套餐名额与网络路径，以及在隔离副本中工作。它不替代 App，远程桌面、文件预览和推送通知仍在 App 内使用。

和 App 一样，`v42` 先连接最近的接入点（AP），网络允许时会自动切换到与电脑的点对点（P2P）直连。直连不消耗 Credits，经过中转的流量才消耗 Credits（见 Credits 与用量）。如果需要强制走中转，可以加上 `--force-relay`。

从哪里获得

大多数情况下，你的机器上已经带了 `v42`：

你所在的环境	`v42` 来自哪里	身份与设备名额
Mac 或 Linux 桌面 App	随桌面 App 一起安装	共用桌面 App 的安装与登录状态，不用重新登录，也不额外占用设备名额
一行命令安装的 Server 版	与 `v42-companion` 一起装在 `~/.v42/bin`	共用 Vibe42 后台服务的安装
其他任何机器	运行 `v42 login --standalone`	注册为独立的 CLI 安装，占用 1 个设备名额

CLI 的版本号始终与随附的后台服务一致，更新方式也和桌面 App 或后台服务相同（见更新与卸载）。

登录

CLI 通过浏览器授权登录，不使用粘贴 token 或 API key 的方式：

```
v42 login
```

运行后会输出 `verify_url`、`user_code` 和 `expires_at`，然后等待你在浏览器里确认。CLI **不会**自动打开浏览器，请自己打开 `verify_url`，核对标题为 **授权设备登录** 的页面上显示的验证码一致，再点击批准。如果同一台电脑上的桌面 App 已经登录，`v42` 默认直接复用桌面 App 的登录状态，无需再运行 `v42 login`。

可用选项：

- `--standalone`：不复用桌面 App 的登录状态，把当前 CLI 注册为独立安装，占用 Team 的 1 个设备名额。
- `--no-wait`：打印授权链接和验证码后立即退出，适合由脚本把信息发给其他人去网页上批准。
- `--poll`：继续等待之前用 `--no-wait` 发起的授权完成。

运行 `v42 logout` 退出登录；运行 `v42 whoami` 查看当前登录的账号与所在的 Team。

> **说明**
>
> Vibe42 不提供 token 或 API key 登录。如果自动化任务需要独立身份，请在它运行的机器上手动执行一次 `v42 login --standalone` 。

输出约定

`v42` 默认输出 JSON，脚本直接读取字段即可，不需要解析普通文本：

- 成功时向 **stdout** 输出 `{"ok":true,"data":…}` 。
- 失败时向 **stderr** 输出 `{"ok":false,"error":{"code":…,"message":…,"details":…}}` 。
- 加上 `--format text` 会改为输出适合阅读的文本（可通过 `--locale` 或 `V42_LOCALE` 指定语言）。
- 流式命令（`attach` 、`ws enter` 、`term attach` 和 `term new --attach` ）的 stdout 只输出远端 PTY（终端与程序之间的底层输入输出通道）原始字节，不附加其他内容。

```
{"ok":true,"data":{"user_id":"…","install_id":"…","team_id":"…"}}
```

上面的字段值只是示例，顶层 JSON 结构在所有命令中保持一致。

Exit code

代码	含义
`0`	成功
`2`	参数错误：未知命令、参数不合法或未知的 `--agent` ID
`3`	未登录，或登录授权已过期
`4`	无权限：HTTP 403、当前 Team 不可用，或网络拒绝请求
`5`	本地存储或读写错误
`6`	其他客户端错误（例如已在登录状态下再次运行 `login` ）
`7`	服务端 5xx 错误或网络故障
`130` / `143`	被 `SIGINT` / `SIGTERM` 信号中断

流式命令会直接返回远端程序的退出码。

全局选项与环境变量

每个全局选项都有对应的环境变量，方便在 CI 任务和 Agent 环境中配置：

选项	环境变量	作用
`--format json|text`	—	输出格式（默认 `json`）
`--locale <tag>`	`V42_LOCALE`	`--format text` 使用的语言
`--profile <name>`	`V42_PROFILE`	使用指定的本地配置档案（保存独立的登录状态）
`--team <uuid>`	`V42_TEAM`	对指定 Team 执行操作，而不是当前默认 Team
`--agent-name <s>`	`V42_AGENT_NAME`	将自动化名称写入 User-Agent 和审计日志（最多 128 字节）
`--force-relay`	—	本次命令强制走 AP 中转
`--data-dir <path>`	`V42_DATA_DIR`	CLI 保存本地状态的目录
`--companion-root <path>`	`V42_COMPANION_ROOT`	后台服务安装目录的位置

只要由 Agent 调用 `v42`，都建议设置 `V42_AGENT_NAME`，这样管理审计记录里就能看清是哪个自动化执行了操作（见让 Agent 使用 Vibe42）。

最小示例

```
v42 whoami
v42 resources
```

`whoami` 用来确认登录状态和当前 Team；`resources` 会返回一份完整的 JSON，列出 Team 下的电脑、你的已登录安装、Workspace，以及每台在线电脑上打开的终端。拿到电脑名称后，就可以对照命令参考执行操作，或直接使用自动化示例里的现成脚本。

命令参考

按分组列出 v42 的全部命令——会话、Team、Workspace、终端、文件、名额与网络、端口转发、隔离副本——含语法、参数、返回字段与示例。

vibe42.ai/zh-HANS/docs/cli/commands

本页按分组列出 `v42` 的所有命令、参数及 JSON 返回字段。登录方式、JSON 输出结构、退出码（exit code）与全局选项请先看 v42 CLI 概览。下文默认你已经完成登录。

约定

- `<computer>` 是 `v42 resources` 列出的电脑名称。
- 远端文件路径格式为 `<Computer>:/绝对路径`；不带电脑名称前缀的路径代表当前本机路径。
- 所有命令都支持在子命令前传入全局选项（如 `--format`、`--team`、`--agent-name`、`--force-relay` 等）。
- 下文说明的返回字段都位于 `{"ok":true,"data":…}` 响应的 `data` 对象内；示例中的值仅供演示。

会话

```
v42 login [--standalone] [--poll] [--no-wait]
v42 logout
v42 whoami
```

- `login` 会打印 `verify_url`、`user_code` 与 `expires_at`，并等待你在浏览器里批准（CLI 不会自动打开浏览器）。`--standalone` 将当前 CLI 注册为独立安装（占用 1 个设备名额）；`--no-wait` 打印信息后立即退出；`--poll` 继续等待之前未完成的授权。如果同一台机器上的桌面 App 已经登录，无需运行 `login`，`v42` 会直接复用桌面 App 的登录状态。
- `logout` 退出当前 CLI 的登录状态。
- `whoami` 查看当前登录的账号与生效中的 Team。

Team

```
v42 team list
v42 team use <team_id>
```

`team list` 返回当前账号所属的全部 Team。`team use` 将其中一个设为后续命令默认使用的 Team；如果只想在单次命令中临时切换 Team 而不修改默认设置，可以使用 `--team <uuid>`（或环境变量 `V42_TEAM`）。

Workspace

```
v42 ws list
v42 ws enter <workspace> [--stdio]
```

`ws list` 返回 Team 下的所有 Workspace 及其对应的电脑和目录。`ws enter` 会在该 Workspace 所在的电脑和目录下新建一个终端，并把你的当前终端连接过去。`ws enter` 是流式命令，stdout 只输出 PTY（终端与程序之间的底层输入输出通道）字节流，退出码直接来自远端程序。如果该 Workspace 开启了终端隔离，终端会在一份全新的隔离副本中启动。

`--stdio` 会跳过交互式终端设置（不把本机终端设备 TTY 设为 raw 模式），直接通过 stdin 和 stdout 收发原始字节，适合由其他程序驱动终端会话。

终端

```
v42 term new <computer> [--workspace <ws>] [--cwd <dir>] [--name <n>] [--agent <id>] [--command "
<cmd>"] [--attach [--stdio]]
v42 term attach <computer> <target> [--stdio]
v42 attach <computer> <target> [--stdio]
```

`term new` 在电脑上新建一个终端窗口（与在 App 里新建终端完全一致），并返回：

```
{
  "computer_id": "…",
  "window_id": "@12",
```

```
  "session": "…",
  "target": "…:@12",
  "workspace_id": null,
  "terminal_isolation": false,
  "agent": "claude-code",
  "command": "claude"
}
```

- `--workspace <ws>` 在指定 Workspace 中打开终端（默认进入 Workspace 目录；如果开启了终端隔离，则在新的隔离副本中启动，此时返回的 `terminal_isolation` 为 `true`）。未指定 Workspace 时，可用 `--cwd <dir>` 设置起始目录。`--name <n>` 用于设置窗口名称。
- `--agent <id>` 启动内置目录里的一个 Agent，并由 Vibe42 后台服务自动跟踪它的状态与通知。参数请传目录中的 ID，而不是显示名称：`claude-code`、`codex`、`gemini`、`opencode`、`cursor`、`copilot`、`pi`、`omp`、`devin`、`antigravity`、`cline`、`kimi`、`kiro`、`droid`、`amp`、`grok-build`、`aider`、`goose`、`crush` 等。如果传入未知 ID，命令会以退出码 `2` 失败并列出所有可用 ID。如果目标电脑上的 Vibe42 版本过旧、还不支持启动 Agent，命令会返回错误码 `launch_unsupported`。
- `--command "<cmd>"` 不启动内置 Agent，而是直接运行单行命令 `<program> <args>`，规则与 **Agent** 设置中的自定义命令一致（见启动 Agent）。
- `--attach` 在新建窗口后立即连接过去；再加 `--stdio` 则以纯字节流方式通信。

`term attach`（或简写为 `attach`）将当前终端连接到一个已有的窗口。`<target>` 传入 `term new` 返回的 `target` 值（格式为会话名后可选跟上 `:@<window>`；只传会话名则连接到整个会话）。这两条都是流式命令：stdout 输出远端 PTY 数据，退出码直接取自远端程序。

文件

```
v42 cp <src> <dst> [-r] [--restart]
v42 sync <src> <dst> [--delete] [--restart]
```

- `cp` 在本机路径和 `<Computer>:/绝对路径` 之间双向复制文件；加上 `-r` 可递归复制整个目录。
- `sync` 单向将源目录的内容同步到目标目录。目标目录中多出的文件默认**保留**，只有加上 `--delete` 才会一并删除。
- `--restart` 在传输中断后从头重新传输，而不是断点续传。

返回的 JSON 会汇总传输结果，包含 `files_copied`、`files_skipped`、`bytes` 和 `entries_deleted`。文件传输使用独立的 `file` 通道，不会拖慢已打开的终端（见连接是怎么建立的）；经过 AP 中转的流量会消耗 Credits。

名额、资源与设备

```
v42 quota
v42 resources
v42 device revoke <install_id>
```

- `quota` 返回当前 Team 的套餐额度（各项名额与 Credits），方便脚本在配对或发起任务前先检查余量（见套餐与名额）。

- `resources` 一次性返回当前 Team 的电脑列表、你的已登录安装、Workspace，以及每台在线电脑上已打开的终端（离线电脑的 `status` 为 `"offline"`，终端列表为空）。
- `device revoke <install_id>` 撤销指定安装对当前 Team 的访问权限并释放其占用的设备名额。`install_id` 可从 `resources` 的输出中获取，执行成功后返回 `{"team_id":…,"install_id":…,"revoked":true}`。

网络

```
v42 network [<computer>]
```

不传 `<computer>` 时，`network` 报告 CLI 到最近 AP 的连接状态：`ap_path`、`ap_round_trip_ms`、`ap_lanes` 和 `force_relay`。传入 `<computer>` 时，它还会尝试连接该电脑，短暂等待点对点直连建立，并在返回结果中附带 `target` 对象：

```
{
  "ap_path": "relay",
  "ap_round_trip_ms": 23.4,
  "ap_lanes": ["…", "…", "…"],
  "force_relay": false,
  "target": {
    "computer_id": "…",
    "p2p_allowed": true,
    "p2p_connected": true,
    "lanes": ["…", "…", "…"]
  }
}
```

只要三条通道（`control`、`terminal`、`file`）中有一条建立了直连，`p2p_connected` 就为 `true`；`lanes` 按上述顺序列出三条通道各自的路径。这与 App 内 **设置 → 网络诊断** 展示的信息完全一致。加上 `--force-relay` 会把 `force_relay` 设为 `true`，且本次调用不尝试建立直连。

端口转发

```
v42 forward <computer>:<remote-port> [<local-addr>:]<local-port>
```

在本机监听一个 TCP 端口，并通过与终端相同的加密通道，将连接转发到远端电脑的 `127.0.0.1:<remote-port>`：

- 不填 `<local-addr>` 时，默认只监听本机 `127.0.0.1`；如果绑定到其他地址，命令会打印警告，提醒局域网内的其他设备也能访问该端口。
- `<local-port>` 设为 `0` 时，由操作系统自动分配空闲端口，并在就绪后打印实际监听地址。
- 最多同时转发 **64** 条连接。
- 默认只能转发到远端电脑本机的回环地址（loopback，即 `127.0.0.1`，该权限 `forward:loopback` 随终端访问权默认开启）。如果需要通过该电脑转发到它所在内网的其他机器，必须先在远端电脑上显式放行。
- 连接经过 AP 中转时，转发流量会计入 Credits；审计日志不会记录转发的具体内容。

命令启动后会持续运行，按 `Ctrl+C` 即可停止。转发开发服务器的常见用法见预览产物与端口转发。

隔离副本

```
v42 copy status [--root <dir>]
v42 copy save [-m <message>]
v42 copy versions
v42 copy changes [--from <id>] [--to <id>]
v42 copy diff <path> [--from <id>] [--to <id>]
v42 copy apply <paths…> | --all
v42 copy push --branch <branch>
v42 copy rebuild-index [--root <dir>] --yes
v42 copy skill
```

`copy` 系列命令直接在远端电脑本机、由终端隔离创建的隔离副本中运行，无需登录。在副本目录内直接运行，或用环境变量 `V42_COPY` 指向副本目录即可。

- `status` 查看当前副本的状态；加上 `--root <dir>` 则列出指定 Workspace 目录下的所有副本。
- `save` 将当前文件保存为一个新版本（`-m` 添加备注说明）；`versions` 列出已保存的版本。
- `changes` 列出相对于初始基线（或两个版本 ID 之间）发生改动的文件路径及应用状态；`diff <path>` 输出单个文件的差异对比（unified diff）。
- `apply <paths…>` 或 `apply --all` 将改动写回原 Workspace 目录；`push --branch <branch>` 将副本的 Git `HEAD` 作为新分支推送到 Workspace 仓库。
- `rebuild-index --yes` 清空并重建某个 Workspace 目录的版本索引。这会清除该目录下所有副本的历史版本记录，因此必须显式加上 `--yes`。
- `skill` 输出一份 `SKILL.md` 说明文档，用来指导 coding agent 在隔离副本中工作。

自动化示例

可直接套用的 v42 脚本片段：启动 Agent 终端并连接、把构建产物同步回本机、用 jq 判断是否直连、临时转发开发服务器、在隔离副本里保存并推送分支。

vibe42.ai/zh-HANS/docs/cli/automation

本页提供几段可直接套用的脚本，展示如何把 `v42` 接入脚本、CI 任务与 coding agent。每段示例都标明了运行位置和前提条件。完整的命令语法与返回字段见命令参考，JSON 输出结构与退出码（exit code）见 v42 CLI 概览。

开始之前

以下所有脚本默认满足这些前提：

- 运行脚本的机器已安装并登录 `v42`（随桌面 App 附带、由一行命令安装器安装，或通过 `v42 login --standalone` 登录）。

- `my-mac` 是 `v42 resources` 列出的某台在线电脑名称，使用时请替换为你自己的电脑名。
- 机器上已安装 `jq` 用于解析 JSON。由于返回结果都包裹在 `{"ok":true,"data":…}` 中，所有 `jq` 路径都从 `.data` 开始。
- 由 Agent 调用这些命令时已设置环境变量 `V42_AGENT_NAME`，以便在审计记录中标明是哪个自动化执行的。

在脚本中判断命令是否成功时，请直接检查退出码，不要解析输出文本：

```
out=$(v42 whoami)
status=$?
if [ "$status" -ne 0 ]; then
  echo "v42 failed with exit code $status" >&2
  exit 1
fi
```

1. 启动一个 Agent 终端并附着

在你的笔记本或任何装有 `v42` 的机器上运行。下面的命令会在 `my-mac` 的 `api` Workspace 中启动 Claude Code，并立即把你的当前终端连接过去：

```
V42_AGENT_NAME=nightly-runner v42 term new my-mac --workspace api --agent claude-code --attach
```

`--attach` 是流式命令：stdout 直接传输远端 PTY（终端的底层输入输出通道）数据，退出码也来自远端程序。如果想先创建终端窗口，稍后（或在另一个 shell 里）再连接过去，可以从返回的 JSON 中读取 `target`：

```
target=$(v42 term new my-mac --workspace api --agent claude-code | jq -r '.data.target')
v42 term attach my-mac "$target"
```

无论有没有终端连接上去，这个窗口都会在远端电脑上持续运行，并且像其他终端一样带着实时 Agent 状态显示在 App 里。如果要运行你自己的脚本而非内置目录中的 Agent，把 `--agent` 换成 `--command "<program> <args>"` 即可。

2. 把构建产物同步回本机

在接收文件的机器上运行。下面的命令把 `my-mac` 上的 `dist` 目录单向同步到本机的 `./dist`，同时保留本机多出来的文件：

```
v42 sync my-mac:/Users/me/app/dist ./dist
```

加上 `--delete` 可以删掉本机多余文件，让 `./dist` 与远端目录完全一致；如果传输中断后想从头开始传，可以加上 `--restart`：

```
v42 sync my-mac:/Users/me/app/dist ./dist --delete
```

返回的 JSON 会列出本次传输的详情，例如 `.data.files_copied` 和 `.data.bytes`。如果只传单个文件或需要反向传输，请使用 `cp`：

```
v42 cp ./config.toml my-mac:/Users/me/app/config.toml
```

3. 传大文件前先确认走的是直连

在任意装有 `v42` 的机器上运行。经过接入点（AP）中转同步大量文件会消耗 Credits，因此可以先让脚本确认当前已建立点对点（P2P）直连：

```
if [ "$(v42 network my-mac | jq -r '.data.target.p2p_connected')" = "true" ]; then
  v42 sync my-mac:/Users/me/app/dist ./dist
else
  echo "relayed path only; skipping the large sync" >&2
fi
```

只要三条通道中有一条建立直连，`p2p_connected` 就为 `true`；如果需要精确检查某条通道，可以读取 `.data.target.lanes`（按 `control`、`terminal`、`file` 的顺序列出各通道的路径）。浏览器始终走中转，而 CLI 只要两侧网络条件允许就会自动直连（见连接是怎么建立的与 Credits 与用量）。

4. 把开发服务器转发到手边的机器

在你要打开浏览器的本机上运行。下面的命令把 `my-mac` 上的 `3000` 端口映射到本地的 `127.0.0.1:3000`：

```
v42 forward my-mac:3000 3000
```

随后在浏览器中打开 `http://127.0.0.1:3000` 即可访问。命令会一直运行，直到你按下 `Ctrl+C`。在自动化脚本里，可以把它放到后台运行，测试完成后再结束进程：

```
v42 forward my-mac:3000 0 &
forward_pid=$!
# … 对监听器打印出来的端口执行检查 …
kill "$forward_pid"
```

把本地端口设为 `0` 时，操作系统会自动挑一个空闲端口，并在就绪时打印出实际地址。默认情况下，监听器只绑定本机的 `127.0.0.1`，只转发到远端电脑本机的回环地址（loopback，即 `127.0.0.1`），且最多同时转发 64 条连接。

5. 在隔离副本里保存并推送工作成果

以下命令**直接在远端电脑本机**、由终端隔离创建的隔离副本终端里运行，不需要登录。Agent 完成任务时的典型收尾步骤如下：

```
v42 copy save -m "tests green"
v42 copy changes
v42 copy push --branch agent/fix-login
```

`save` 保存一个新版本快照，`changes` 列出相对于初始基线的改动及是否已应用，`push --branch` 则把当前副本的 Git `HEAD` 作为新分支推送到 Workspace 仓库供你审阅。如果想把改动的文件直接写回原 Workspace 目录，可以运行 `v42 copy apply --all`。如果不在副本目录内执行，请把环境变量 `V42_COPY` 设为副本目录的路径。

> **提示**
>
> 运行 `v42 copy skill` 会输出一份介绍这套流程的 `SKILL.md` 文档。请手动把它提供给你的 Agent——Vibe42 不会自动改动任何 Agent 的配置文件（见不注入、不锁定）。

10 故障排查

连接问题

排查电脑显示离线、配对码过期、持续重连、无法直连以及后台服务与授权报错。

vibe42.ai/zh-HANS/docs/troubleshooting/connection

当电脑显示离线、配对超时、App 反复重连，或弹出授权、服务与版本报错时，可以按本页逐项排查。关于 AP（接入点）选路、点对点直连与断线恢复机制的说明，请参阅连接原理。

先看网络诊断

打开设置中的 **网络诊断**，查看当前选中的最近 AP（区域路由节点）、RTT（往返延迟），以及每台电脑的 `control`、`terminal`、`file` 三条通道走的是直连（`direct`）还是中转（`relayed`）。你也可以在 CLI 中检查当前路径：

```
v42 network
```

> **提示**
> 在安装 Server 版（只有 Vibe42 后台服务、没有桌面 App 的版本）的电脑上运行 `~/.v42/bin/v42-companion status`（旧安装目录可能为 `~/.vibex`），可以确认后台服务是否正在运行及是否已完成配对。

配对与登录

配对码过期或丢失

8 位配对码（`XXXX-XXXX`）有效期为 10 分钟，每次只能配对一台电脑或设备。

请生成新配对码并尽快批准。Server 版电脑运行以下命令查看配对码（加 `--json` 可输出 `pairing.code` 与 `pairing.approve_url`），再到 `https://vibe42.ai/console/pair` 或已登录 App 的 **电脑与设备 → 连接新的电脑/服务器** 中批准：

```
~/.v42/bin/v42-companion status --json
```

配对手机或平板时，在已登录设备上打开 **电脑与设备 → 下载与安装 → 配对手机/平板** 生成新码，再在手机上点 **扫码配对**。

"Login or pairing did not finish in time (…)"（登录或配对超时）

一行命令安装脚本打印配对码后最多等待 10 分钟，超时未批准便会退出。

运行 `~/.v42/bin/v42-companion status` 获取新的配对码与链接，打开链接并在 10 分钟内批准。

"登录尚未完成，请在系统浏览器中完成登录。"

桌面 App 已在系统默认浏览器中打开授权页，但你还没有点击 **批准**，或提前关掉了标签页。

切回系统浏览器并点击 **批准**。如果标签页已关闭，回到 Vibe42 点击 **登录** 重新发起登录。

"This device's authorization was revoked."（设备授权已被撤销）

这台设备或电脑已在 Console 或其他设备上被登出或移除，现有连接会在 60 秒内断开并停止自动重连。

桌面与网页端请重新点 **登录** 登录；手机请从另一台已登录设备打开 **电脑与设备 → 下载与安装 → 配对手机/平板** 重新扫码配对；Server 版电脑请运行 `~/.v42/bin/v42-companion status` 获取新配对码。

电脑离线与网络路径

电脑显示离线或 "暂时无法连接云端服务，请检查网络后重试。"

电脑处于休眠或断网状态、后台服务（`v42-companion`）未运行，或防火墙拦截了访问 Vibe42 AP 的出站 HTTPS 与 QUIC（基于 UDP 的传输协议）流量。

唤醒该电脑并确认出站网络畅通，随后在电脑上查看后台服务状态与实时日志（若问题仍在，见收集日志）：

```
~/.v42/bin/v42-companion status
~/.v42/bin/v42-companion logs -f
```

App 一直在重连

Wi-Fi 不稳、切换网络或手动锁定了距离较远的 AP 会导致连接中断。断线 30 秒内 App 会尝试在原 AP 快速恢复，超时后按 1–60 秒间隔重新连接。

在 **网络诊断** 中取消手动锁定的 AP，让 Vibe42 自动测速选路。如果路由器导致 UDP 直连频繁断开，可在电脑卡片或设置中开启 **强制中继** 强制走中转。

总是中转，从不直连

你正在使用浏览器（网页端 `https://app.vibe42.ai` 始终走 AP 中转）、开启了 **强制中继**，或两侧路由器的 NAT 与 UDP 防火墙限制阻止了点对点（P2P）直连。

改用桌面 App、手机 App 或 CLI，并在设备与电脑两侧确认 **强制中继** 已关闭，再用 `v42 network` 检查通道是否切换为 `direct`。

公司网络连不上

企业防火墙或网关通常会拦截用于点对点直连的 UDP/QUIC 流量，或封锁部分区域节点。

在设置或电脑卡片中开启 **强制中继** 走 AP 中转，确认网络放行出站 HTTPS，必要时在 **网络诊断** 中锁定可达的 AP 区域。

"设备数量已达套餐上限"、"电脑名额已用完" 或 "This month's Credits are used up"

遇到名额或额度限制时，App 会立即停止自动重试。"设备数量已达套餐上限"、"电脑数量已达套餐上限"、"电脑名额已用完"、"成员名额已满" 与 "暂时无法接入这个团队" 表示 Team 名额已满；"This month's Credits are used up" 会暂停 AP 中转连接，P2P 直连不受影响。

在 Console 移除闲置电脑或设备，或联系 Team 的 Owner 升级套餐（见套餐与名额和电脑、设备与 Team）。Credits 用完时可使用原生客户端走 P2P 直连，或由 Owner 在 Console 续费升级（见Credits 与用量）。

后台服务、安装器与版本报错

"Vibe42 is installed but the companion service is not running."（后台服务未运行）

后台服务已停止，或 macOS 后台运行权限未开启（提示 `"Allow Vibe42 to run in the background in System Settings → General → Login Items & Extensions, then retry."`）。如果看到 `"This computer's Vibe42 companion is managed by the Vibe42 App…"`，说明后台服务由桌面 App 托管，无需运行命令行安装器。

macOS 请前往 **System Settings → General → Login Items & Extensions** 允许 Vibe42 后台运行后重试；Server 版可手动启动后台服务（仅在已卸载桌面 App 并准备改用 Server 版时，才运行 `uninstall --purge` 后重装）：

```
~/.v42/bin/v42-companion start
```

"systemd user session bus is unavailable; log in normally or run loginctl enable-linger $USER"（systemd 用户总线不可用）

常见于仅通过 SSH 登录的 Linux 服务器，用户级 `systemd --user` 未启动且未开启 linger。

开启 linger 后退出并**新开一个 SSH 会话**重跑安装命令，或强制使用无需 systemd 的 supervisor 模式：

```
sudo loginctl enable-linger "$USER"
# 开启新的 SSH 会话后重跑安装脚本，或直接强制使用 supervisor 模式：
curl -fsSL https://get.vibe42.app/install.sh | VIBEX_SERVICE_MODE=supervisor sh
```

"没有找到可安全使用的 tmux，或 tmux 版本不兼容。" 或 "Cannot mount AppImage, please check your FUSE setup."

Server 版要求安装 `tmux` 3.0 及以上版本（`"tmux 3.0+ is required"`）；Linux 桌面 AppImage 挂载运行依赖 FUSE（Linux 挂载文件系统的组件）。

用系统包管理器安装或升级 `tmux`（3.0+）后重跑安装脚本。Linux AppImage 请安装 `fuse3`（见 Linux 桌面端），或添加 `--appimage-extract-and-run` 参数直接运行。

"本版本不受支持，请前往 vibe42.ai 下载最新版本。"

客户端或后台服务版本过旧（网页端显示 "当前版本已过期，请刷新页面获取新版本。"，CLI 显示 "请升级 Vibe42 CLI 后再继续。"）。

桌面 App 请前往 `https://vibe42.ai/#downloads` 下载新版，手机端在 App Store 或 Google Play 更新，网页端刷新页面，Server 版运行以下命令原地升级（见更新与卸载）：

```
~/.v42/bin/v42-companion update
```

收不到通知

排查手机收不到推送、单台电脑无通知、Agent 完成未提醒、通知过多以及桌面端或 Telegram 投递问题。

vibe42.ai/zh-HANS/docs/troubleshooting/notifications

当手机收不到推送、Agent 跑完没有提醒、桌面端或 Telegram 没有消息，或者点击通知没有进入对应窗口时，可以对照本页排查。关于通知规则与渠道的配置说明，请参阅通知。

各投递渠道的工作方式

不同渠道的权限要求与离线表现不同：

投递渠道	何时可收通知	应用关闭时的表现
手机推送	无需后台常驻，随时经系统推送服务（APNs / FCM）接收	关掉 App 照常送达
桌面 App（macOS / Linux）	仅在桌面 App 运行期间弹出系统通知	退出期间的通知不会在下次打开时补发
网页端（`app.vibe42.ai`）	仅在标签页打开期间显示页内弹窗	关闭页面后无通知
Telegram	绑定账号后随时接收	由 Telegram 正常保存并推送

常见症状与处理

手机完全收不到推送

Vibe42 手机 App 在首次配对完成前不会请求系统通知权限。如果你跳过了系统弹窗、在系统设置里关了通知、开启了勿扰或专注模式，或者设备授权已被撤销，系统推送就无法显示。

前往 iOS 或 Android 系统设置开启 Vibe42 的通知权限，并确认专注模式允许 Vibe42 通知。随后打开 Vibe42 设置 → **通知**，确认目标电脑的总开关已打开，且 **送达到** 下已勾选 **手机推送**。

只有某台电脑收不到通知

每台电脑的通知是独立设置的。那台电脑的总开关、投递目标或具体事件类型可能被关掉了，或者电脑上的 Vibe42 后台服务处于离线状态。

进入设置 → **通知** 并切换到这台电脑，确认通知总开关已打开、**送达到** 勾选了当前渠道、**事件类型** 中的对应事件已开启，同时确认该电脑在线（见连接问题）。

Agent 完成了但没收到推送

如果你直接在普通 shell 里敲命令启动 Agent，除非该工具原生支持发送 OSC 7501（程序向 Vibe42 报告状态用的一串控制序列，Claude Code 2.1.295+ 与 Pi 1.1.0+ 已内置支持），否则后台服务不会代报状态。此外，推送只在状态**变化**为 已完成、需要处理 或 出错 时触发，同一分屏（终端窗口切分出的一个区域）在 3 秒内会去重并有 15 秒摘要等待窗口；也可能是对应事件开关没打开。

请在新建终端的第三步选择 Agent，或通过 CLI 启动以启用状态跟踪（见启动 Agent与 Agent 状态）：

```
v42 term new <computer> --agent claude-code
```

如果你习惯在现有 shell 中直接运行 Claude Code 或 Pi，请升级到 Claude Code 2.1.295+ 或 Pi 1.1.0+。最后打开设置 → **通知 → 事件类型**，确认 **Agent 已完成**、**Agent 需要处理** 与 **Agent 出错** 都已勾选。

收到太多通知

繁忙的电脑上同时开启了终端响铃、窗口或分屏关闭、电脑上下线等高频事件。通知频率超过约 20 条/分钟时会自动合并为一条摘要，但低于这个频率时仍会逐条提醒。

打开设置 → **通知** 选中该电脑，在 **事件类型** 下关闭不需要的高频事件（例如只保留 **Agent 需要处理**、**Agent 已完成** 和 **Agent 出错**，关闭终端响铃与分屏关闭事件），或在 **送达到** 中取消不想被打扰的渠道。

桌面 App 或网页端没有通知

桌面 App 只在运行期间弹出系统通知，退出后不会驻留，重新打开也不会补发期间错过的通知；网页端（`app.vibe42.ai`）只在页面打开时显示页内弹窗。系统通知权限或 **送达到** 里的网页与桌面 App 选项也可能未开启。

打开设置 → **通知 → 送达到**，确认已勾选网页与桌面 App 投递，并在 macOS 或 Linux 系统通知设置中放行 Vibe42。如果需要在合上电脑或关掉浏览器后继续收提醒，请开启 **手机推送** 或绑定 Telegram。

Telegram 没收到消息

账号还未绑定 Telegram、你在 Telegram 里停用或静音了机器人对话，或者该电脑的 **送达到** 未勾选 Telegram。

打开设置 → **通知** 检查或重新绑定 Telegram，确认 Telegram 里的机器人对话处于开启状态，然后为每台需要收提醒的电脑勾选 **送达到** 下的 Telegram。

点通知没跳到对应的窗口

点击通知时，Vibe42 会检查目标事件、电脑、窗口和分屏是否还在。如果命令执行完后窗口或分屏已经关闭，或者电脑已离线、已解绑，就会回到首页。

在首页确认目标电脑是否在线。如果脚本跑完后自动退出了窗口，请在启动命令后保留交互式 shell，让点击通知时目标分屏依然存在（见会话管理）。

终端与输入问题

解决快捷键冲突、中日韩输入法不出字、贴图与终端图片显示、OSC 52 复制、Linux 文件预览以及软键盘遮挡问题。

vibe42.ai/zh-HANS/docs/troubleshooting/terminal-and-input

当快捷键不生效、中日韩输入法打不出字、贴图或终端内图片不显示、剪贴板复制没反应，或者 Linux 文件预览与远程桌面画面异常时，可以按本页逐项排查。

键盘、输入法与剪贴板

快捷键被终端吃掉，或在浏览器里不起作用

当光标焦点在终端内时，Vibe42 会优先把按键传给远端 shell 和正在运行的程序（例如 `Ctrl+B`），只拦截应用明确声明的快捷键。在网页端，`Ctrl+Shift+N` 等浏览器保留快捷键无法被网页捕获，且 Firefox 不支持使用 `Ctrl+Shift+P`。

在浏览器中使用专门的替代快捷键（例如 `Alt+Shift+N`），或改用桌面 App 获得完整快捷键支持。完整键位表见快捷键；iPad 连接外接键盘时可长按 `⌘` 查看可用快捷键。

中文、日文或韩文输入法不出字

在 Linux X11 桌面会话下，Vibe42 AppImage 依赖 `fcitx5` 或 `ibus` 的 GTK4 输入法模块才能弹出候选框并输入文字；Wayland 会话与 macOS 原生支持输入法，不需要额外安装模块。

为你的输入法框架安装对应的 GTK4 模块（以 Ubuntu/Debian 为例，安装 `fcitx5-frontend-gtk4` 或 `ibus-gtk4`，包名因发行版而异）后重启桌面 App，或切换到 Wayland 会话登录。详见 Linux 桌面端。

复制到剪贴板没反应

远端程序（如 Neovim）通过 OSC 52（用于向本地剪贴板写入内容的终端控制序列）复制文本时，内容会写入**你当前正在查看的这台设备**的剪贴板；在桌面 App 中，收到 OSC 52 序列时 Vibe42 窗口必须处于焦点状态。

复制时保持 Vibe42 窗口处于当前焦点，并确认终端环境或编辑器已开启 OSC 52 透传（见终端引擎）。在手机或平板上，也可以直接长按终端文字选词复制。

手机软键盘挡住输入行

手机弹出软键盘后可视区域变矮，全屏 TUI（在终端里画界面的程序，比如 Claude Code）与普通命令行对高度变化的反应不同。

在移动端终端设置里切换三种软键盘避让模式——**页面平移**（整体上移页面）、**终端缩放**（按剩余空间调整终端尺寸）或 **光标对齐**（自动跟随光标），选一个能让当前输入行保持可见的模式。详见移动端输入。

渲染、图片与字体

贴图失败或提示图片太大

直接粘贴剪贴板内容时，纯文本上限为 1 MiB，单张图片上限为 4 MiB。超过上限的图片与 HEIC 照片会自动转成 JPEG；其他超过上限的内容无法通过剪贴板直接粘贴。

改为把文件拖放或粘贴上传到终端（手机端可选择拍照、相册或选取文件），Vibe42 会把文件上传到电脑并在光标处自动填入路径；大文件也可以通过文件面板上传。如果文件本来就在那台电脑上，用附件菜单里的 **插入远端路径** 直接插入它的路径，不需要上传。

终端内图片不显示

终端内直接显示图片使用 Kitty 图形协议（一种在终端里渲染图像的标准协议），单张图片上限为 16 MiB；在网页端，每个分屏（终端窗口切分出的一个区域）最多同时显示 5 张图片。

确认命令行看图工具输出的是 Kitty 图形协议，且单张图片不超过 16 MiB（见终端引擎）。如果在浏览器里因超过 5 张而被替换，可以在文件面板中直接预览或下载原图，或改用桌面与手机 App 查看。

字体或宽字符错位

当开启连字改变字符宽度，或字体本身字符宽度不一致时，中日韩（CJK）双宽字符或终端表格边框可能出现错位。

打开主题与字体设置，切换为内置等宽字体（**Geist Mono**、**JetBrains Mono**、**Fira Code**、**Blex Mono** 或 **Noto Sans Mono**，已针对 CJK 双宽对齐做过优化），关闭连字并按需调整行高。详见主题与字体。

锁屏或切 App 回来后终端短暂空白

手机 App 不在后台常驻，锁屏或切到后台期间会断开实时数据流。切回前台时 App 会自动重连并重新绘制终端画面；电脑上的终端会话一直在后台运行，不会中断。

稍等片刻让重连与画面刷新完成即可（见会话管理）。如果长时间停留在空白画面，请按连接问题检查当前网络。

文件预览与远程桌面

Linux 桌面端文件预览变成纯文本

Linux 桌面 App 依赖 `bubblewrap`（`bwrap`）与 `xdg-dbus-proxy` 在沙盒内渲染代码、Markdown、HTML 与 PDF 预览；缺少这两个软件包时，文本文件会退回纯文本显示，PDF 则只能下载。

用发行版包管理器安装这两个软件包后重启 Vibe42 AppImage（见 Linux 桌面端）：

```
sudo apt install bubblewrap xdg-dbus-proxy
```

远程桌面黑屏或提示缺少解码器

在浏览器里使用远程桌面需要浏览器支持 WebCodecs（网页硬件视频解码接口）；Linux 桌面 App 若没有 VA-API 或 NVIDIA 硬件解码，需要按需下载 Cisco OpenH264 解码器；被控 Mac 如果没开启屏幕录制权限也会显示黑屏。

网页端请使用支持 WebCodecs 的现代浏览器；Linux 桌面端按提示下载 Cisco OpenH264，或配置 VA-API / NVIDIA 硬件解码驱动；被控 Mac 请在 **设置 → 远程桌面** 中开启 **屏幕录制** 与 **辅助功能** 权限。详见远程桌面入门。

收集日志

按平台查找并导出后台服务、桌面端、移动端与浏览器日志，检查敏感信息并向技术支持提交完整排查报告。

vibe42.ai/zh-HANS/docs/troubleshooting/logs

当按连接问题、收不到通知或终端与输入问题操作后仍未解决问题时，可以按本页收集各平台日志与状态信息，自行排查原因或发送至 `support@vibe42.ai`。

收集前的准备

先记下**问题发生的准确时间与时区**（例如 `2026-11-02 14:20 UTC+8`）。桌面端日志单个文件满 4 MiB 会自动轮转，刚复现问题后马上收集最容易保留完整记录。

在运行 Vibe42 后台服务的电脑上，同时导出一份 JSON 格式的状态信息一并附上（旧版安装目录可能为 `~/.vibex` 而非 `~/.v42`）：

```
~/.v42/bin/v42-companion status --json > v42-status.json
```

> **说明**
>
> 如果电脑当前处于待配对状态，`status --json` 会包含 `pairing.code` 与 `pairing.approve_url`。发送前请删掉这两个字段，或等 10 分钟配对码自动过期。

各平台日志位置速查

平台	主要日志路径	说明
macOS 桌面 App	`~/Library/Application Support/ai.vibecafe.x/ai.vibecafe.x/logs/native.log`	满 4 MiB 轮转为 `native.log.1`
后台服务（macOS / Linux / Server）	`~/.v42/logs/companion.log`、`companion-install.log`	supervisor 模式另有 `supervisor.log`；旧目录为 `~/.vibex/logs/`
iOS / Android	应用私有目录（用户无法直接导出）	请提供截图、发生时间、App 版本或系统崩溃日志
网页端	浏览器开发者工具 **Console** 与 **Network** 面板	截图或导出报错信息与失败请求

各平台收集命令

macOS 桌面 App

复制 `native.log`（若刚发生轮转请一并带上 `native.log.1`）、后台服务日志、最近 1 小时的系统日志，以及相关的崩溃报告文件名：

```
ls -lh ~/Library/Application\ Support/ai.vibecafe.x/ai.vibecafe.x/logs/
ls -lh ~/.v42/logs/

# 导出最近 1 小时的 macOS 系统日志
log show --predicate 'subsystem == "ai.vibecafe.x.native"' --last 1h --info > v42-macos-
system.log

# 查找最近的 Vibe42 崩溃报告
ls -t ~/Library/Logs/DiagnosticReports/ | grep -i -E 'v42|vibe' | head
```

Linux 与一行命令安装（Server 版）

适用于 Linux 服务器、容器、macOS 命令行安装，以及由 Linux 桌面 AppImage 管理的后台服务：

```
~/.v42/bin/v42-companion status --json
~/.v42/bin/v42-companion logs -f
```

日志文件位于 `~/.v42/logs/companion.log` 与 `~/.v42/logs/companion-install.log`（supervisor 模式下另有 `~/.v42/logs/supervisor.log`）。在 systemd 主机上，`journalctl` 只记录服务的启动与停止事件，连接细节仍在 `companion.log` 中：

```
journalctl --user -u v42-companion.service -n 100
```

iPhone、iPad 与 Android

iOS、iPadOS 与 Android 的运行日志保存在系统私有沙盒内，无法直接导出。遇到问题请记录发生时间、设置页显示的 App 版本号，并附上界面截图。

如果遇到应用闪退崩溃：

- **iPhone / iPad**：打开系统 **设置 → 隐私与安全性 → 分析与改进 → 分析数据**，找到与 Vibe42 对应的崩溃报告并导出。
- **Android**：在系统开发者选项中开启 USB 调试，连接电脑后通过 `adb logcat` 过滤崩溃标签 `VibexPanic`：

```
adb logcat -s VibexPanic:*
```

网页端

在 `https://app.vibe42.ai` 中打开浏览器开发者工具，查看 **Console** 面板中的报错信息，以及 **Network** 面板中失败的 HTTP 或 WebSocket 请求。

发送前检查敏感信息

Vibe42 日志不包含访问令牌、配对码、终端输入输出或文件内容（见安全架构）。即便如此，发送前仍请检查日志片段与 `status --json`，按需删去你不希望公开的信息——例如**主机名、本地目录路径、邮箱地址与 IP 地址**。

> 注意
>
> 切勿在邮件中向技术支持或任何人发送密码、邮箱验证链接、银行卡号、TOTP（身份验证器）密钥或动态码、两步验证恢复码。

发送给技术支持的邮件清单

请发送邮件至 `support@vibe42.ai`，并附上以下内容：

1. 问题发生时你正在做的操作。
2. 界面显示的完整报错文字或截图。
3. 问题发生的时间与所在时区。
4. 电脑与设备各自的操作系统和 Vibe42 版本号，以及 `~/.v42/bin/v42-companion status --json` 输出。
5. 问题发生时间点前后的日志片段或崩溃报告。

常见问题

快速解答平台支持、Team 名额、coding agent、Credits 计费、端到端安全、邮箱验证以及试用与付款常见报错。

`vibe42.ai/zh-HANS/docs/troubleshooting/faq`

本页汇总了关于平台安装、Team 与名额、Agent 状态、Credits 计费、安全隐私、邮箱验证以及试用付款的高频问题。

安装与平台

Windows 能用 Vibe42 吗?

可以。你可以在 WSL 里安装 Server 版把 Windows PC 作为被控电脑（支持终端、Agent 和文件；不支持远程桌面），从 Windows 控制其他电脑则在现代浏览器打开 `https://app.vibe42.ai`（可安装为 PWA，即能装到桌面或任务栏独立运行的网页应用）。Windows 原生版本将在近期上线。见 Windows。

我的 Mac 还是 macOS 13 怎么办?

macOS 桌面 App 需要 macOS 14+。macOS 13 可运行一行命令安装 Server 版（只有后台服务、没有桌面 App 的版本）作为被控电脑接入，再从其他设备或浏览器连接使用。见 macOS 与安装。

需要公网 IP 或配置路由器端口转发吗?

不需要。后台服务和设备都主动向外连接最近的 AP（区域路由节点），客户端之间会自动尝试建立点对点（P2P）直连，无法直连时自动走 AP 加密中转。见连接原理。

需要自己购买一台 VPS 吗?

不需要。直接把 Vibe42 装在你日常使用的 Mac 或 Linux 电脑上即可；如果你已有云服务器或容器，也可以用一行命令把它接入。见 Linux 服务器。

Vibe42 会修改我的 shell 配置文件吗?

桌面 App 不会改动任何 shell 配置文件。一行命令安装 Server 版只会在你的 shell 配置文件中追加一行 `~/.v42/bin` 的 `PATH` 配置，不注入任何 hook 或插件。见零注入。

账号与 Team

受邀加入 Team 的成员需要自己买套餐吗?

不需要。套餐、电脑名额和设备名额都归属于 Team，由 Owner 统一订阅付款；受邀成员共享 Team 的名额，不用单独购买。见 Team 管理。

换了新电脑或新手机怎么办?

在新电脑或新手机上安装 Vibe42 并登录或配对，随后在 Console 或已登录设备的 **电脑与设备** 中移除旧电脑或旧设备，即可释放名额。见套餐与名额和电脑、设备与 Team。

删除电脑会删掉我机器上的本地文件吗?

不会。移除电脑只会删除云端保存的 Workspace 记录与备忘，你硬盘上的代码、目录和本地文件都不会受影响。见 Workspace。

Agent

Vibe42 支持哪些 coding agent?

任何在终端里运行的命令行 Agent 都能使用。内置目录收录约 30 款（默认展示 Claude Code、OpenAI Codex CLI、OMP、Pi Coding Agent、OpenCode、Grok Build、Devin CLI、Antigravity CLI、Cursor Agent），也可以在设置 **Agent** 中添加自定义命令。见支持的 Agent。

我的模型 API key 会经过 Vibe42 吗?

不会。Agent 直接在你的电脑上运行并直连模型服务商，Vibe42 不改写 Agent 配置，也不中转 API key。见零注入与安全架构。

我自己在终端里敲命令启动的 Agent 为什么没有状态?

除非 Agent 原生支持发送 OSC 7501（程序向 Vibe42 报告状态用的一串控制序列，Claude Code 2.1.295+ 与 Pi 1.1.0+ 已内置支持），否则后台服务只会跟踪从新建终端流程或 `v42 term new --agent` 启动的 Agent。见启动 Agent与 Agent 状态。

电脑重启后，终端和 Agent 会怎样?

电脑重启后，后台服务再次启动时会重建窗口、分屏和所在文件夹，并在顶部回放之前的输出。用新建终端流程或 `v42 term new --agent` 启动的 Agent 会重新打开上一次的对话，但重启时中断的那一轮需要你再发一条消息让它接着做；自己在 shell 里敲命令启动的 Agent 和其他程序不会自动重新运行。见会话与窗口管理。

费用与 Credits

什么流量会消耗 Credits?

只有经 AP 中转的数据流量（终端、远程桌面、文件传输、`v42 forward`）按 64 MiB = 1 Credit 计费，中转连接进出两侧各计一次（传输 64 MiB 共消耗 2 Credits）。连接建立、控制通道与 Console 接口均不计费，当月剩余 Credits 不结转。见 Credits 与用量。

点对点直连免费吗?

免费。桌面 App、手机 App、CLI 与后台服务建立 P2P 直连后传输数据不消耗任何 Credits（浏览器始终走中转）。可以运行 `v42 network` 查看当前路径是 `direct` 还是 `relayed`。见连接原理。

当月 Credits 用完会怎样?

出现 `"This month's Credits are used up"` 后，AP 中转数据通道会暂停，但基础控制连接与 P2P 直连不受影响；续费或升级套餐即可立即恢复中转。见 Credits 与用量。

安全

Vibe42 能看到我的终端内容吗?

不能。每次安装都会在本地生成私钥永不离开本机的 Ed25519 根密钥，每条连接都经过 Noise 握手进行端到端加密。AP 只转发密文，我们也绝不会用你的数据训练模型。见安全架构。

数据存在哪里，服务器会记录什么?

终端会话与文件都留在你自己的电脑上。服务器只保存账号与密码哈希（Argon2id）、Team 元数据，以及登录、配对、撤销、所用 AP 区域与字节数等路由审计记录，从不记录终端文字或文件内容。通知的标题与正文是例外：它们要经过并保存在服务器上才能推送到你的手机，可按电脑关闭（见通知）。详见安全架构与隐私政策。

怎么撤销一台丢失或不再使用的设备?

在 Console 或任意已登录设备的 **电脑与设备** 中移除该设备。现有连接会在 60 秒内断开，并显示 `"This device's authorization was revoked."`。见账号与登录。

邮件验证

为什么提示 "请先验证邮箱再登录。"?

未验证邮箱的账号（也会显示 "还没有验证成功。点开邮件里的链接后再试一次。"）无法在任何设备上登录。请先打开发送到你邮箱的验证链接完成激活。见账号与登录。

提示 "验证链接无效或已过期。" 怎么办?

验证链接 24 小时后过期。在注册页点击重发，或用同一邮箱重新提交一次注册表单，即可收到新的验证链接，原密码不会被修改。见账号与登录。

可以用 GitHub 或 Google 登录来完成邮箱验证吗?

可以。用同一邮箱点击 **使用 GitHub 继续** 或 **使用 Google 继续** 可直接完成验证。这会清除未验证账号之前设置的密码；后续如果仍需密码登录，可在 `vibe42.ai` 的 `/password-reset` 页面重设密码。见账号与登录。

试用与付款

为什么提示 "此团队无法开始免费试用，请直接订阅。"?

3 天免费试用仅向符合条件的新 Team Owner 提供一次。请直接前往 `https://vibe42.ai/console/checkout?resource=access` 选择套餐订阅。见试用、兑换与退款。

为什么提示 "This card has already been used for a free trial"?

每张银行卡只能开通一次免费试用。请更换其他银行卡，或直接订阅套餐。见试用、兑换与退款。

"Card setup expired" 或 "Card setup canceled" 会扣款吗?

不会。这表示绑卡验证超时或中途关闭，试用尚未开始且没有产生任何扣款，点击 **重新开始试用** 重试即可。见试用、兑换与退款。

看到 "正在确认付款结果。你可以稍后回来查看，请先不要发起另一笔付款。" 怎么办?

支付渠道正在确认扣款结果。请稍等片刻后刷新 Console 查看状态，不要重复提交付款。见试用、兑换与退款。

为什么提示 "这项操作需要 Team 所有者完成，请联系对方处理套餐或付款。"?

只有 Team 的 Owner 可以选择套餐、管理账单或使用兑换码（且手机 App 内不显示价格与订阅管理）。请由 Owner 在浏览器打开 Console 操作。见 Team 管理与试用、兑换与退款。