Vibe Coding 介绍 · 装上你的第一个Agent

Claude Code　Codex　Google Antigravity　DeepSeek Harness

以 Windows 原生环境为主，附 macOS 与 Linux 安装入口。第三方供应商部分分别演示 Magpie 和 CC Switch，使用 DeepSeek API 作为填写示例。

资料核对日期：2026 年 10 月 17 日。Magpie 与 CC Switch 部分沿用 v0.1.1279、v4.0.14 界面。CLI 操作另与当前文档核对，社区教程入口见附录。

先认识 CLI TUI GUI

这三个缩写描述软件的操作方式。CLI 通过命令使用程序；TUI 把菜单、输入框和状态信息放进终端；GUI 使用窗口、按钮和图标。

名称	含义	操作时看到什么
CLI Command Line Interface	命令行界面	输入一条命令，程序执行并输出文字。 例如：`claude --version`。
TUI Text User Interface	文本用户界面，也称终端用户界面	终端里有持续更新的对话区、输入框、菜单和状态栏，可用方向键、回车及快捷键操作。
GUI Graphical User Interface	图形用户界面	窗口里有侧栏、按钮和对话框。桌面应用及浏览器里的 Web UI 都属于这一类。

CLI 版工具可以带有 TUI。 例如，在 PowerShell 中输入 `codex` 是用命令启动程序；启动后出现的交互聊天界面就是 TUI。终端界面也能输入中文自然语言。

CLI 看起来是什么样

下面是命令行示意。第一行在提示符后输入命令，第二行是结果，最后一行表示已经可以输入下一条命令。

PowerShell 命令行示意

```
PS C:\work\harness-demo> Get-ChildItem -Name
README.txt
PS C:\work\harness-demo>
```

`PS ...>` 是终端显示的提示符，不需要复制。只输入 `Get-ChildItem -Name`，就会列出当前目录里的文件名。

先打开哪一个窗口

在 Windows 开始菜单搜索 **PowerShell** 并打开，或在 **Windows Terminal** 的新标签页选择 PowerShell。Windows Terminal 是承载终端会话的窗口；PowerShell 是解释命令的 Shell。

本教程标有 **Windows PowerShell** 的代码粘贴到 PowerShell；`/model`、`/resume` 这类斜杠命令，则输入在已经启动的工具对话框内。

概念与示例 Microsoft Windows Terminal · Codex CLI 与 TUI

终端界面与图形界面对照

TUI 在终端内对话

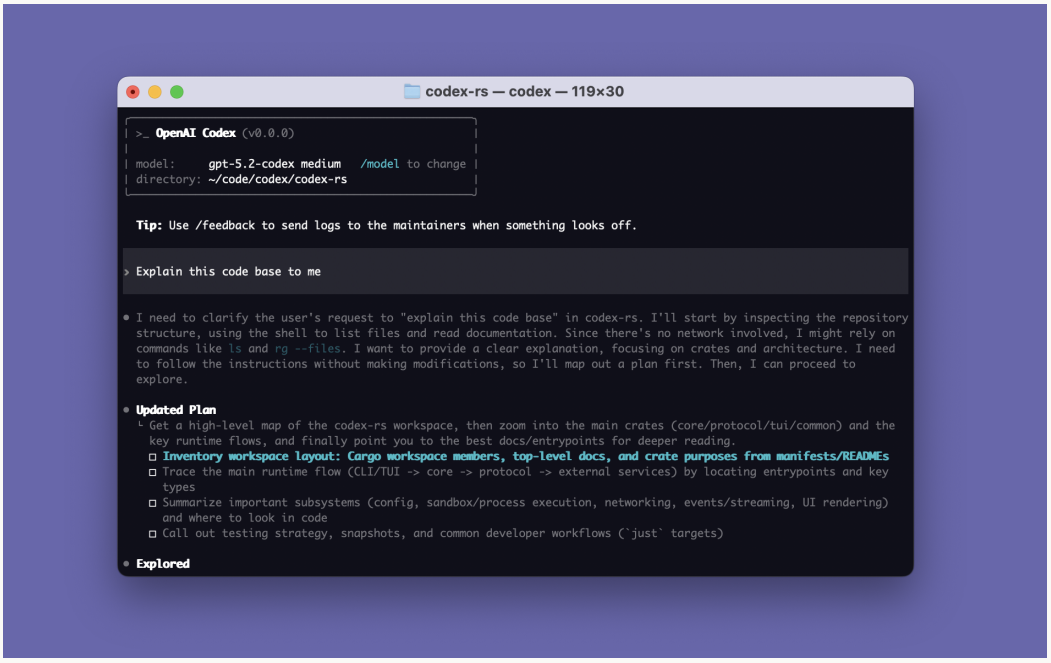


图 1 Codex CLI 的终端界面。模型与目录显示在上方，任务和工具执行过程排列在对话区。

图片来源 OpenAI Codex 仓库界面图

GUI 用窗口和按钮操作



图 2 DeepSeek Harness 的桌面界面。通过工作区侧栏、模型菜单和底部输入框完成操作。

图片来源 DeepSeek 官方下载页

配图保留来源页面发布时的界面。图中版本号、模型和演示目录用于说明外观；安装与模型选择按正文步骤进行。

安装前准备

工具	推荐安装方式	必需条件与账号
Claude Code	官方原生安装脚本	Windows 10 1809+；4 GB 以上内存；可用的 Claude 订阅、Console API 或兼容供应商凭据。
Codex	官方 CLI 安装脚本	Windows 原生 PowerShell；可用的 ChatGPT 登录、OpenAI API 或自定义供应商凭据。
Google Antigravity	官方桌面安装包	64 位 Windows 10+；符合年龄与地区条件的 Google 账号，或组织分配的企业访问权限。
DeepSeek Harness	官方桌面版或 npm Web UI	桌面版：Windows 10+ x64，或 Apple 芯片 Mac、macOS 13+。npm 方式另需 Node.js 22.19+ 的 22.x，或 24+，以及模型凭据。

官方桌面安装包和 Claude Code、Codex 原生安装脚本均无需预装 Node.js。采用 DeepSeek Harness 的 npm 方式时，需要先安装 Node.js；本教程的社区 TUI 插件还需要 pnpm 10+。从源码构建另需 Git 和 pnpm。

教程连接的是云端模型 API，本机负责读写项目、运行命令和显示界面。无需下载模型权重；运行你的项目所需的 Python、Node.js、编译器等，仍要按项目要求安装。

准备一个练习目录

在文件资源管理器中新建 `Documents\harness-demo`，放入一个写有 `harness-demo` 的 `README.txt`，供读取验证使用。已有同名目录时，换一个新目录。启动工具前，用 PowerShell 进入它：

Windows PowerShell

```
Set-Location "$env:USERPROFILE\Documents\harness-demo"
```

下文中标有 PowerShell 的命令粘贴到 Windows 终端；标有 Bash 的命令用于 macOS、Linux 或 WSL。Windows 与 WSL 的安装目录、环境变量和配置文件分别管理。

后面会用到的读写练习

完成连接配置后，在工具对话框发送下面这段请求，再用文本编辑器打开生成的文件核对内容。这就是后文的 HELLO.txt 读写验证。

在工具对话框中输入

```
读取 README.txt，新建 HELLO.txt 并写入 hello harness，
再读取 HELLO.txt，报告实际读到的内容。
```

内容索引

1 Claude Code 安装与终端界面 2 Codex 安装与终端界面
3 Antigravity 桌面与 CLI 4 DSH 桌面 Web CLI 与社区 TUI
5 第三方接入前准备 6 Magpie 7 CC Switch
8 故障排查与配置备份 附录 社区教程与图文来源

官方资料 Claude Code 系统要求 · Codex CLI · Antigravity 下载 · DSH 开发环境

1 Claude Code

Claude Code 在终端中读取项目、修改代码和执行命令。先进入项目文件夹，再运行 `claude`；启动时所在的目录决定了这次会话从哪个项目开始。后面的 Magpie、CC Switch 配置以这个 CLI 为基础。

前置要求

Windows 10 1809+ 或 Windows Server 2019+，x64 或 ARM64，至少 4 GB 内存。官方登录需可用的 Claude Pro、Max、Team、Enterprise 订阅，或有 API 额度的 Claude Console 账号；服务地区需符合 Anthropic 的支持范围。

Git for Windows 是推荐组件，可提供 Git 和 Bash。当前 Claude Code 在未安装 Git for Windows 时使用 PowerShell 执行命令；原生安装不要求 Node.js，也不要求 WSL。

准备入口 Git for Windows · Anthropic 支持地区

Windows 安装与首次运行

1 运行官方安装命令

打开普通 PowerShell 窗口，执行：

```
irm https://claude.ai/install.ps1 | iex
```

2 重新打开终端检查版本

```
claude --version
claude doctor
```

第一条显示版本号；第二条检查安装状态。若提示找不到命令，先关闭并重新打开终端，再检查安装器提示的 PATH。

3 进入练习目录启动

```
Set-Location "$env:USERPROFILE\Documents\harness-demo"
claude
```

使用官方账号时，跟随终端提示在浏览器登录并返回终端。使用第三方 API 时，先完成第 6 节或第 7 节的供应商配置，再启动新会话；API Key 填在供应商配置中。

4 确认模型与工作目录

在会话中输入 `/status` 查看当前配置，再发送：

在 Claude Code 对话框中输入

```
请报告当前工作目录，并列出其中的文件。不要修改文件。
```

能返回当前目录中的实际文件，说明模型请求与本地工具调用都已开始工作。需要切换官方账号时，在会话中使用 `/login`。

macOS 与 Linux

macOS 需 13.0+；Linux 官方列出 Ubuntu 20.04+、Debian 10+、Alpine 3.19+。在对应系统终端执行：

Bash

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

原生安装会自动更新，也可执行 `claude update`。若使用 WinGet 安装，则用 `winget upgrade Anthropic.ClaudeCode` 更新。

官方资料 安装与系统要求 · 首次使用 · 网关配置

Claude Code 的终端界面

安装和登录完成后，在项目目录运行 `claude`。出现下面这样的界面，就可以在底部输入请求；中间会显示回复、读取文件和执行命令的过程。

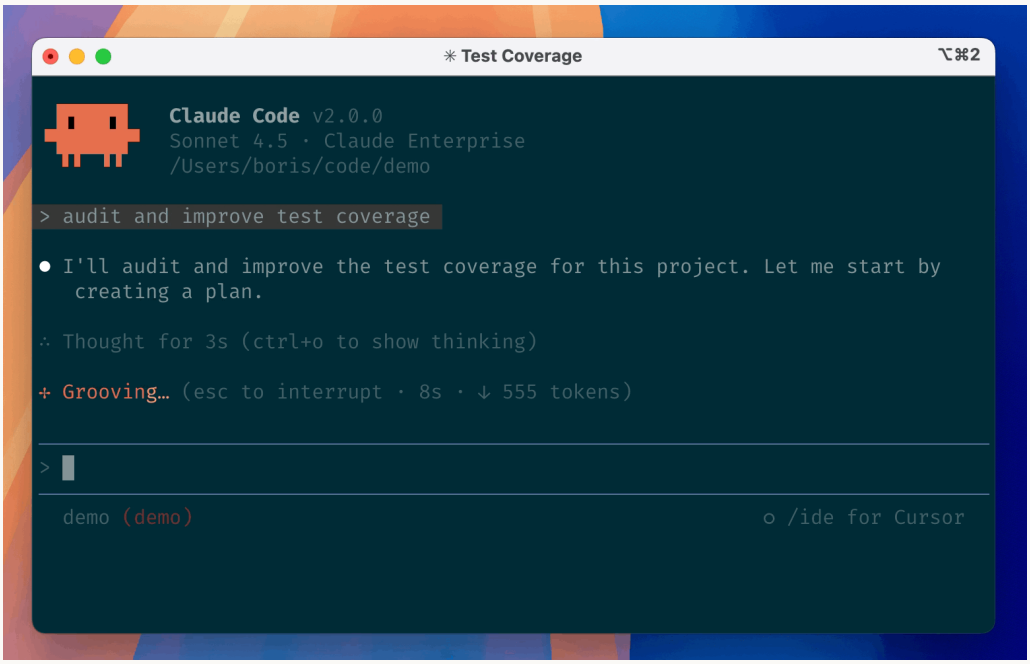


图 3 Claude Code 官方演示中的终端界面。顶部显示模型和工作目录，底部的 > 后面是输入位置。

图片来源 Anthropic Claude Code 官方演示

第一次使用时常用的操作

输入位置	命令或按键	用途
工具对话框	`/help`	查看本机版本的命令说明。
工具对话框	`/model` · `/status`	选择模型；查看当前会话配置。
工具对话框	`@README.txt` 后接问题	把文件路径加入提示。例如：解释这个文件。
工具对话框	`/resume`	从历史会话列表继续已有任务。
工具对话框	`/exit`	退出普通前台会话，回到 PowerShell。
任务运行时	`Ctrl+C`	中断当前操作。空闲时会先清空输入，再按一次退出。

关掉终端后怎样继续

重新打开 PowerShell，进入原来的项目目录，然后选择下面一条执行。`--continue` 接着该目录最近的对话，`--resume` 打开历史列表。

```
claude --continue
claude --resume
```

官方资料 Claude Code 交互操作 · 斜杠命令 · 启动与恢复参数

2 Codex

Codex CLI 在本机终端中工作，可以检查代码、编辑文件并运行项目命令。安装后，在项目目录运行 `codex`，按首次启动提示登录。Magpie、CC Switch 均可为这个客户端配置第三方供应商。

前置要求

Windows 可直接使用 PowerShell，WSL 不是必需项。原生安装脚本不要求 Node.js。准备一个可使用 Codex 的 ChatGPT 账号，或 API 凭据；ChatGPT 订阅与 API 账单分别计算。

安装 Git 可使用版本比较、提交和工作树功能。首次设置 Windows 沙箱时按界面提示完成；若所选沙箱方案要求管理员授权，由系统弹出相应提示。

Windows 安装与首次运行

1 安装 CLI

在新开的 PowerShell 窗口中执行官方命令：

```
powershell -ExecutionPolicy ByPass -c "irm https://chatgpt.com/codex/install.ps1 | iex"
```

2 重新打开终端检查版本

```
codex --version
codex doctor
```

3 启动并选择认证方式

```
Set-Location "$env:USERPROFILE\Documents\harness-demo"
codex
```

使用官方服务时，选择 **Sign in with ChatGPT** 并在浏览器完成登录，或选择可用的 API Key 登录方式。使用第三方服务时，先完成 Magpie 或 CC Switch 配置，再重启 Codex。

4 检查当前连接

在会话里输入 `/status`，检查模型和供应商，再发送第 1 节中的文件列表请求。第三方连接应同时在供应商控制台或本机网关中看到对应请求。

macOS 与 Linux

Bash

```
curl -fsSL https://chatgpt.com/codex/install.sh | sh
codex --version
```

原生安装的更新方式是重新运行同一安装命令。如果已经通过 npm 安装，可继续用 `npm install -g @openai/codex` 更新；无需为了本教程改用第二种安装方式。

桌面入口

官方当前文档将 Windows 桌面入口列为 ChatGPT desktop app，可从下方官方页面下载安装。CLI 和桌面入口使用同一用户配置目录时，读取同一份 `config.toml`；供应商或模型列表变更后，桌面应用应彻底退出再打开。

常见配置位置：`%USERPROFILE%\.codex\config.toml`。设置了 `CODEX_HOME` 时，以该目录为准。

官方资料 CLI 安装 · 登录方式 · Windows 桌面入口 · Windows 沙箱 · 自定义供应商

Codex 的终端界面

在项目目录运行 codex 后，会进入持续对话的 TUI。输入自然语言任务即可开始；工具调用、文件修改和结果会显示在同一窗口中。

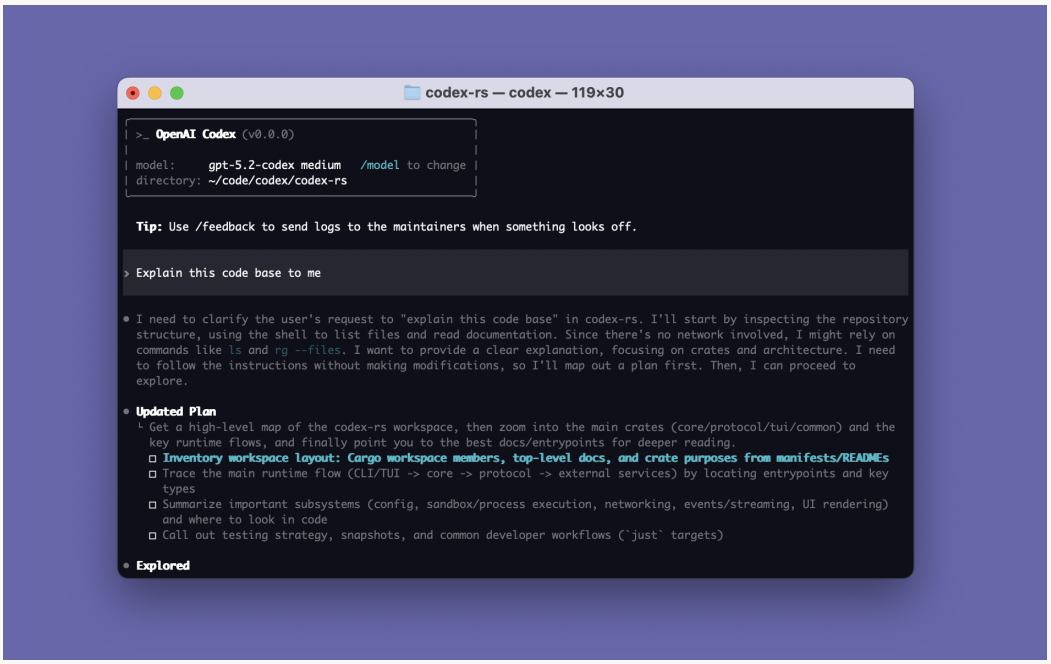


图 4 Codex 官方仓库中的终端示例。model 是当前模型，directory 是工作目录；下方展示任务和执行进度。

图片来源 OpenAI Codex 仓库界面图

第一次使用时常用的操作

输入位置	命令或按键	用途
工具对话框	/model · /status	选择模型；核对当前模型、权限和工作目录。
工具对话框	/permissions	查看和调整这次会话允许的操作。
工具对话框	@ 后接文件名	搜索项目文件并加入提示。
工具对话框	/diff	在 Git 项目中查看改动；普通练习目录可直接打开生成的文件。
工具对话框	/resume · /new	继续历史会话；另开一个新会话。
工具对话框	/quit 或 /exit	退出 CLI，回到 PowerShell。

重新进入历史会话

```
codex resume
codex resume --last
```

在原项目目录执行：第一条显示会话选择器；第二条直接继续该目录最近的会话。单独输入 codex 会开始新的对话。

官方资料 Codex 斜杠命令 · CLI 启动与恢复

3 Google Antigravity

Google 提供 Antigravity App、CLI、编辑器扩展和独立 IDE。App 使用图形窗口；CLI 的命令是 `agy`，运行后进入终端交互界面。两种入口分别安装。

前置要求

Windows 需 64 位 Windows 10+，macOS 需 12+，Linux 需 glibc 2.28+、glibcxx 3.4.25+。选择符合设备架构的安装包，原生安装无需 Node.js、Python 或 VS Code。

个人登录使用符合年龄与地区要求的 Google 账号，个人使用需年满 18 岁；企业登录使用组织提供的 Gemini Enterprise 访问权限。

安装桌面版并添加项目

1 下载安装包

打开官方下载页，在 **Antigravity App** 区域选择系统与架构。Windows 运行下载的 `.exe` 并完成安装。

下载 Antigravity 官方下载页

2 登录

启动 App，点击 **Sign in**，在浏览器完成 Google 登录后返回应用。企业用户按组织提供的 Google Cloud 项目与许可完成登录。

3 加入练习目录

点击 **New Project** → **Add Folder**，选择 `harness-demo`，点击 **Create**。同一项目可加入多个相关仓库。

4 选择工作位置并发送请求

首次练习选择 **Local**，直接使用当前目录。**New Worktree** 为 Git 项目创建独立工作树。输入“请列出当前工作目录中的文件，不要修改文件”，核对返回的列表。

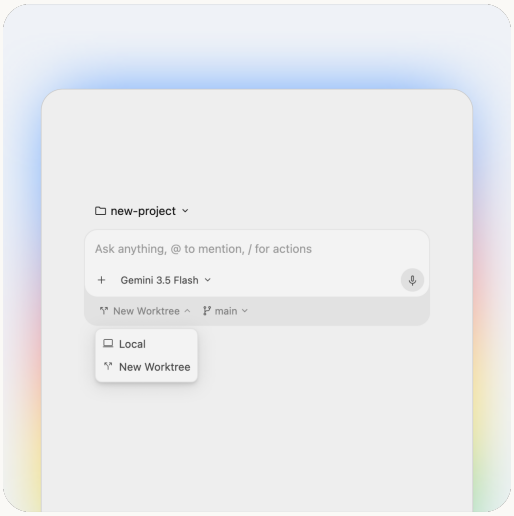


图 5 Antigravity App 的项目、模型与运行模式入口。Local 表示在当前目录中工作。

图片来源 Google Antigravity 官网

官方资料 首次项目设置 · 认证说明 · 年龄与地区要求

Antigravity CLI 的安装与终端界面

1 在 PowerShell 安装

运行官方安装脚本，完成后重新打开终端：

```
irm https://antigravity.google/cli/install.ps1 | iex
```

2 检查并启动

```
agy --version
Set-Location "$env:USERPROFILE\Documents\harness-demo"
agy
```

首次启动选择配色与显示模式，确认当前项目目录的信任提示，再按提示登录。**Alt-Screen** 使用全屏终端缓冲区，**Inline** 将内容保留在终端滚动历史中。

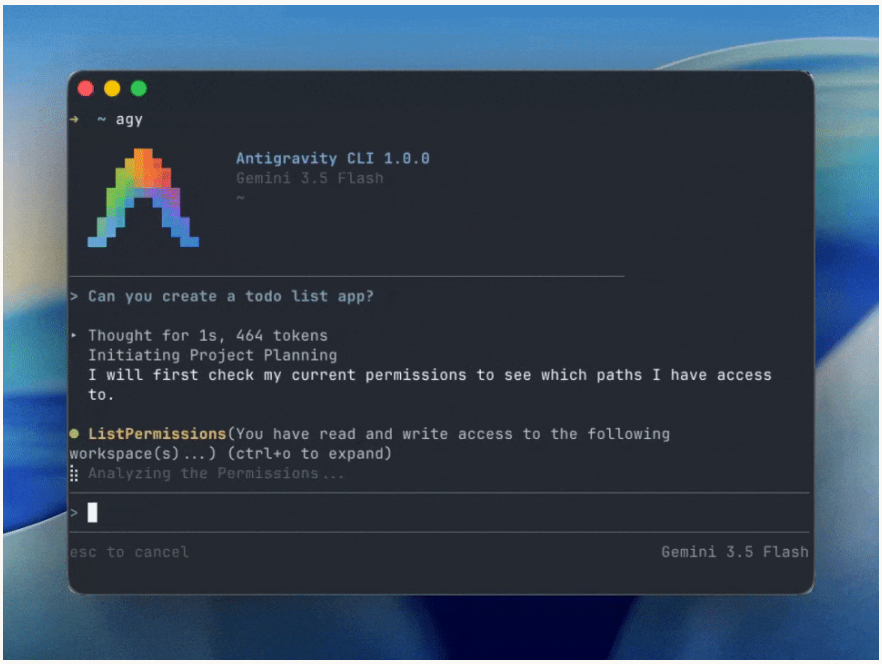


图 6 Antigravity CLI 官方演示帧。中间显示任务与工具结果，底部 > 后面可继续输入。

图片来源 Google Antigravity CLI 官方演示

在工具对话框输入	作用
`/help` · `/model`	查看帮助；选择可用模型。
`@README.txt` 后接问题	引用当前项目里的文件。
`/resume` · `/diff`	继续历史会话；检查代码改动。
`/exit` 或 `/quit`	退出 TUI 并返回 PowerShell。

macOS、Linux 安装命令如下，完成后在项目目录运行 `agy`：

Bash

```
curl -fsSL https://antigravity.google/cli/install.sh | bash
```

官方资料 CLI 首次设置 · CLI 命令与快捷键

4 DeepSeek Harness

DeepSeek Harness，简称 DSH，提供官方桌面版，也可以通过 npm 在本机启动 Web UI。桌面版已打包运行时；npm 方式则由你在终端启动服务。两种方式都能围绕工作区读写文件和运行任务。

直接安装官方桌面版

Windows 版要求 Windows 10 或更新版本、x64 设备。macOS 版面向 Apple 芯片，要求 macOS 13 或更新版本。无需为桌面版预先安装 Node.js、pnpm。

1 下载并安装

打开 DeepSeek 官方下载页，在 **DeepSeek Harness** 区域选择 Windows 版，下载并运行安装程序。macOS 用户选择 Apple 芯片版本。

下载 DeepSeek 官方下载页 · Harness 产品页

2 打开练习工作区

启动应用，完成首次启动页面中的账号或模型配置；添加并选中 `harness-demo` 工作区，再发送文件列表请求。若使用自己的 API Key，可在 **设置** → **模型** 中填写。

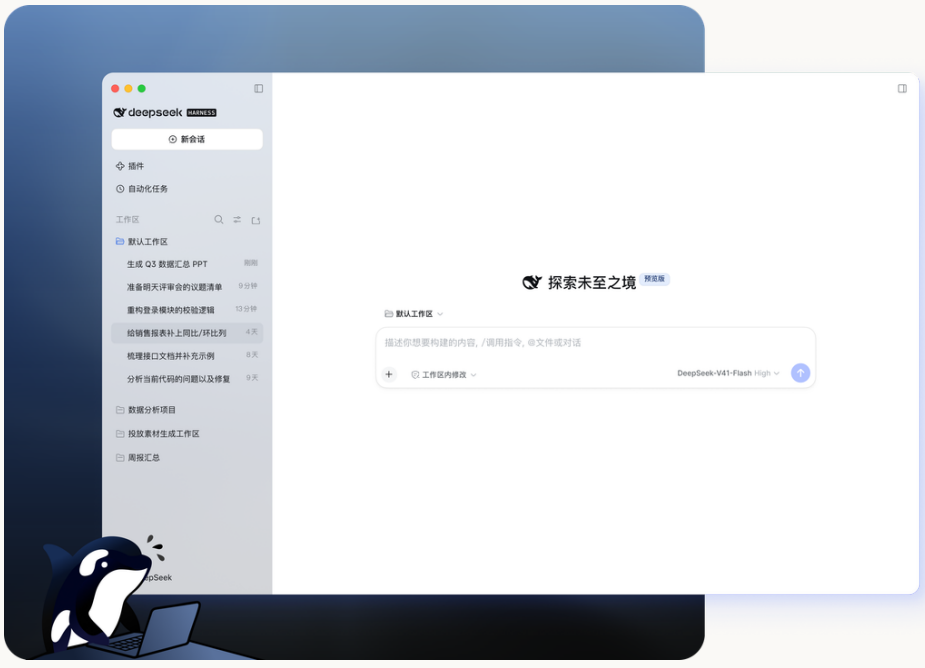


图 7 DeepSeek 官方下载页展示的 Harness 桌面界面，左侧是工作区和会话，中间是任务输入区。

图片来源 DeepSeek 官方下载页

Windows 关闭主窗口后，应用可继续在托盘运行，点击托盘图标可恢复窗口。需要停止后台任务时，从托盘菜单选择 **退出 DeepSeek Harness**。

官方资料 桌面运行时与退出行为

通过 npm 启动 Web UI

这种方式适合在终端中启动和停止服务，也适用于 Linux。浏览器只是界面，运行任务的 DSH 进程仍在你的电脑上。项目处于开发者预览阶段，升级前保留配置与项目副本。

安装 Node.js 24 LTS，安装器同时提供 npm 与 npx。项目支持 Node.js 22.19+ 的 22.x 或 24+。本页的 Web 启动无需 pnpm；后面的社区 TUI 插件安装需要 pnpm 10+。准备一个有可用额度的 DeepSeek API Key，也可在模型设置中接入其他供应商。

准备入口 Node.js 官方下载 · DeepSeek API 控制台 · DSH 官方仓库

Windows 启动 Web UI

1 安装并检查 Node.js

在 Node.js 下载页选择 24 LTS、Windows 和设备架构，下载安装器。安装后重新打开 PowerShell，执行：

```
node --version
npm.cmd --version
```

node 应显示符合要求的版本。Windows 下使用 npm.cmd、npx.cmd 可避免调用被执行策略拦截的 .ps1 启动文件。

2 在练习目录中启动

```
Set-Location "$env:USERPROFILE\Documents\harness-demo"
npx.cmd @deepseek-ai/dsh@latest web
```

第一次运行时，npx 会下载所需包；出现安装确认时确认继续。服务默认在 http://127.0.0.1:3080 启动，并打开浏览器。若未自动打开，使用终端打印的完整访问链接，其中包含启动认证信息。

浏览器中的 Web UI → 本机 DSH 进程 → 模型供应商 API

3 填写模型凭据

在 Web UI 打开 **设置** → **模型**，找到 DeepSeek 卡片，填入 API Key 并保存。模型配置会在下一次请求时生效。

4 选择工作区和模型

点击 **选择工作区**，添加并选中刚才的 harness-demo。在模型选择器中选择已配置供应商的模型，然后启动会话。未选中工作区时，输入框不可用。

5 检查任务执行

发送文件列表请求。确认界面显示真实文件读取结果。关闭浏览器不会关闭终端服务；停止服务时回到运行它的终端按 Ctrl+C，下次使用时重新运行同一命令。

macOS 与 Linux 启动

macOS、Linux 安装满足版本要求的 Node.js 后，在项目目录执行：

Bash

```
node --version
npx @deepseek-ai/dsh@latest web
```

官方资料 启动命令 · Web UI 操作 · 模型配置

DSH CLI 的安装与运行方式

`dsh` 是官方命令行启动器。它通过 **Profile** 选择运行方式：`web` 启动浏览器界面，`headless` 执行一次任务后退出。持续在终端里聊天时，使用下一页的社区 TUI 插件。

把命令安装到本机

先按前一页安装 Node.js。在 PowerShell 执行以下命令；已全局安装 DSH 时，只需检查版本。

```
npm.cmd install -g @deepseek-ai/dsh@latest
dsh.cmd --version
dsh.cmd --help
```

全局安装后可直接运行 `dsh`，不必每次输入完整 npm 包名。Windows 示例使用 `.cmd` 启动文件，避免 PowerShell 的 `.ps1` 执行策略拦截；macOS、Linux 使用 `npm` 和 `dsh`。

运行方式	命令	运行后看到什么
Web	`dsh web`	终端打印服务地址，浏览器打开图形界面。
Web 换端口	`dsh web --port 8080`	把本次 Web 服务改为 8080 端口。
Headless	`dsh --profile headless "任务"`	在终端输出结果，任务结束后进程退出。
社区 TUI	`dsh-tui`	显示持续对话的终端界面；需先安装下一页的插件。

试一次 Headless 任务

Headless 适合一次性提问或脚本调用。当前官方默认配置从启动环境或 `.env` 读取凭据，**不会自动使用 GUI 中保存的 Key 或登录状态**。以下先在当前窗口设置 DeepSeek Key：

```
$env:DEEPSEEK_API_KEY = "替换为你的_API_Key"
Set-Location "$env:USERPROFILE\Documents\harness-demo"
dsh.cmd --profile headless "读取 README.txt 并概括内容，不修改文件"
```

把占位文字替换为自己的密钥。这个环境变量只在当前 PowerShell 窗口及其启动的程序中生效；关闭窗口后，下次需要重新设置。

这次调用会输出结果并结束。要继续多轮交谈，改用 Web UI 或 TUI；不要等待 Headless 出现聊天输入框。

更新和停止

更新官方 npm 安装时重新运行 `npm.cmd install -g @deepseek-ai/dsh@latest`。Web 服务运行期间保留启动它的终端，回到该终端按 `Ctrl+C` 停止服务。

官方资料 官方 CLI 入口与参数 · 官方 Headless 凭据说明

DSH 的社区 TUI 插件

这里使用 `ccch1mneyyy/dsh-TUI`。它是独立社区项目，为官方 DSH 增加终端聊天界面。仅安装官方 DSH 后，不能直接把任意 `tui` Profile 当作已安装入口。

1 安装插件及包管理器

需要 Node.js 22.19+ 的 22.x 或 24+、官方 DSH 和 pnpm 10+。已安装符合要求的 pnpm 时，跳过第一行。

```
npm.cmd install -g pnpm@10
npm.cmd install -g @deepseek-harness-tui/dsh-tui@latest
```

2 设置凭据并启动

使用前一页的 DeepSeek API Key。在新开的 PowerShell 中重新设置环境变量，然后进入练习目录：

```
$env:DEEPSEEK_API_KEY = "替换为你的_API_Key"
Set-Location "$env:USERPROFILE\Documents\harness-demo"
dsh-tui.cmd
```

首次启动会初始化 `dsh-tui` Profile 并安装所需依赖。下次进入同一项目目录运行 `dsh-tui.cmd` 即可，无需重复安装。



图 8 dsh-TUI 作者发布的终端聊天示例，使用了自定义终端背景。顶部显示模型和目录，底部 > 后面输入请求。

图片来源 dsh-TUI 项目界面图

3 核对模型并对话

在界面中用 `/model` 选择可用模型，发送文件列表请求。`/help` 查看命令，`/resume` 选择历史会话；清空输入后按 `Ctrl+D` 退出。

排错可在 PowerShell 运行 `dsh-tui.cmd doctor`，在 TUI 中运行 `/doctor`。pnpm 低于 10 时先升级；若安装器报告 DSH 与插件版本不兼容，按报告的版本范围安装匹配版本。

macOS、Linux 的安装命令去掉 `.cmd`，设置 Key 使用 `export DEEPSEEK_API_KEY="替换为你的_API_Key"`，最后运行 `dsh-tui`。

社区资料 社区插件中文安装教程 · 交互操作与命令

5 第三方接入前准备

Magpie 和 CC Switch 都能为 Claude Code、Codex 配置模型供应商。选其中一种管理同一个客户端即可；同时切换两边会反复改写同一份客户端配置。

先准备供应商提供的信息

字段	填写内容	示例
API Key	在供应商控制台创建的 API 密钥。	在输入框内粘贴实际密钥；本文不放真实密钥。
Base URL	供应商明确提供的 API 基础地址。	DeepSeek Anthropic：`https://api.deepseek.com/anthropic`
接口协议	Anthropic Messages、OpenAI Responses 或 Chat Completions。	Codex 直接连接时使用 Responses。
模型 ID	供应商请求参数中使用的准确名称。	当前 DeepSeek 示例：`deepseek-flash`。
额度与权限	该 Key 所属账号或套餐能调用所选模型。	网页聊天订阅与 API 套餐分别确认。

Base URL 不是网页聊天地址，也不是随意拼接的完整请求路径。保留供应商文档给出的路径前缀；不要重复追加 `/v1`、`/messages` 或 `/responses`。

按接口决定连接方式

供应商接口	Claude Code	Codex
Anthropic Messages	可直接连接。	需要管理工具转换为 Responses。
OpenAI Responses	需要管理工具转换为 Messages。	可直接连接。
OpenAI Chat Completions	需要管理工具转换协议。	需要管理工具转换协议。

此表针对 Claude Code 的 Anthropic 兼容接入和 Codex 的自定义供应商接入。Magpie 使用本机网关转换；CC Switch 在协议不一致时启用“路由”。模型还需支持工具调用，才能完成读文件、修改代码等任务。

两种工具的请求路径

方式	请求路径	是否需要常驻
Magpie	客户端 → Magpie 本机网关 → 供应商	需要保持 Magpie 网关运行。
CC Switch 直连	客户端 → 供应商 CC Switch 只写入连接配置。	配置生效后，API 请求不经过 CC Switch。
CC Switch 路由	客户端 → CC Switch 本机路由 → 供应商	需要保持 CC Switch 路由运行。

使用其他厂商模型时，协议兼容由供应商与管理工具提供。Anthropic 的官方支持范围不包含通过网关运行非 Claude 模型。

官方资料 DeepSeek Anthropic 地址 · DeepSeek Responses · Claude 网关支持范围 · Codex 网关配置

6 用 Magpie 配置第三方供应商

这里的 Magpie 指 `yetone/magpie`，官网是 `usemagpie.ai`。**Providers** 保存供应商地址和密钥，**Agents** 决定每个客户端使用哪个模型。一家供应商添加一次，就可以分别分配给 Claude Code、Codex。先按前文安装要使用的客户端。

安装 Magpie

1 下载桌面程序

在官网或官方发布页选择 Windows 版本。Intel、AMD 电脑下载 `magpie-windows-amd64.exe`；ARM 设备下载 `magpie-windows-arm64.exe`。带 `magpie-cli-` 前缀的是终端版本，本节使用不带该前缀的桌面版。

下载 Magpie 官网 · 官方发布页

2 启动程序

将 `.exe` 放在固定目录后打开。预编译程序无需安装 Go 或 Node.js。Windows 窗口使用 WebView2；如果启动时提示缺少运行时，安装 Microsoft WebView2 Runtime。

运行时 Microsoft WebView2 Runtime

macOS 可下载对应架构的 DMG。macOS、Linux 也可运行 `curl -fsSL https://usemagpie.ai/install.sh | sh`；Linux 桌面界面需要 WebKitGTK 4.1，没有图形环境时使用 CLI、TUI 或 Web UI。

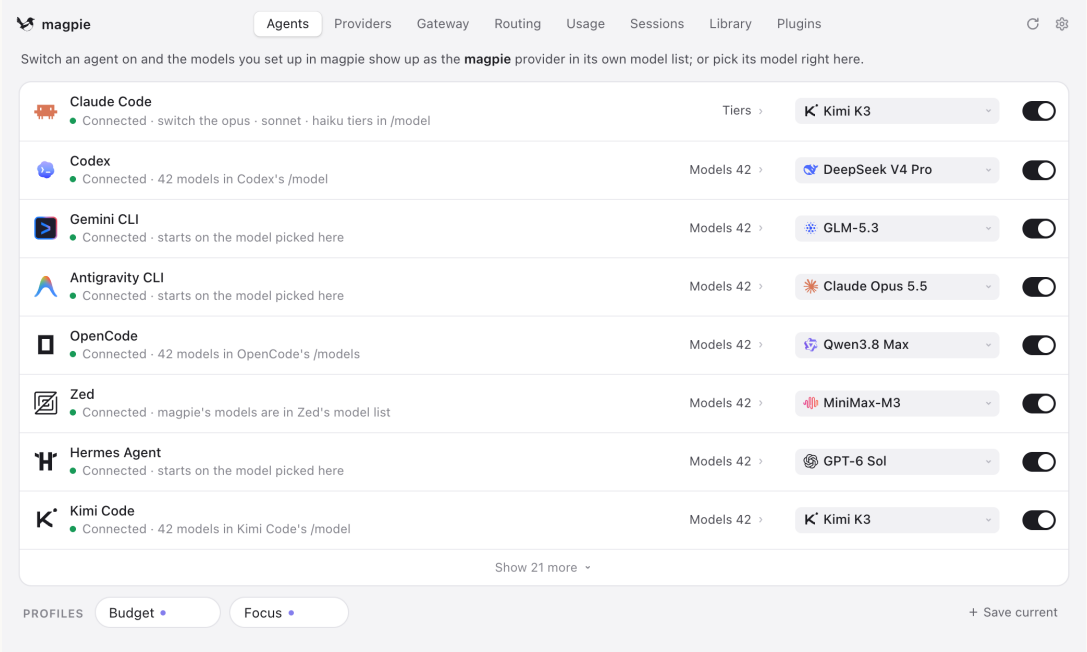


图 9 Magpie 的 Agents 页面。Claude Code 与 Codex 各占一行，可以分别选择供应商和模型；行末开关显示连接状态。

图片来源 Magpie 作者的快速上手指南

添加 DeepSeek 供应商

1 创建 API Key

登录 DeepSeek API 控制台，创建 API Key，并确认该账号有可用 API 额度。

2 选择供应商预设

进入 **Providers** → **Add provider**，搜索并选择 **DeepSeek**。在 API Key 输入框粘贴密钥并保存。预设自动提供对应接口信息。

3 确认模型列表

展开已保存的供应商，查看获取到的模型。选择你要使用的实际模型，例如 `deepseek-flash`；用 **Test** 检查该供应商。若自动列表为空，从供应商文档获取准确模型 ID。

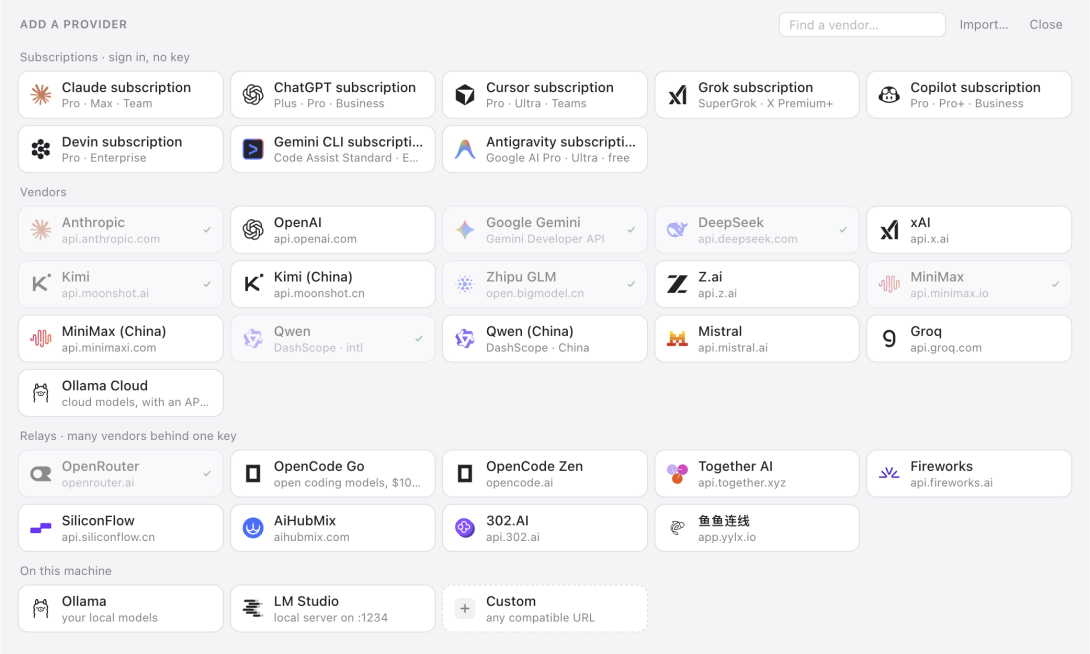


图 10 添加供应商面板。使用 DeepSeek API Key 时，选择 Vendors 中的 DeepSeek；没有预设的接口选择底部 Custom。

图片来源 Magpie 作者的快速上手指南

连接 Claude Code

1 选中客户端

进入 **Agents**，找到 **Claude Code**。若处于未连接状态，打开行末开关，再点击模型名称。

2 指定第三方模型

在 DeepSeek 分组下选择模型。Claude Code 的主模型与 Opus、Sonnet、Haiku 档位可分别指定；首次配置可让各档位跟随主模型。

3 启动新会话

保持 Magpie 运行，在项目目录重新启动 `claude`。Magpie 会写入 Claude Code 的地址、凭据和模型设置。

客户端连接的是 Magpie 的本机地址，再由 Magpie 使用你保存的供应商 API Key 发出请求。不要把客户端中的本机网关凭据误认为供应商的真实 API Key。

连接 Codex

1 选择 Codex 的模型

在 **Agents** → **Codex** 的模型选择器中，选同一个 DeepSeek 供应商下的模型。推理强度使用该模型支持的选项。

2 重新启动 Codex

退出已有 `codex` 进程再启动。使用桌面入口时，彻底退出应用后重新打开，以加载新的模型目录。

3 确认实际调用

在新会话输入 `/status`，确认供应商为 Magpie、模型为刚才选择的模型。发送一次短请求，并核对 Magpie 的用量或请求记录。

Claude Code 或 Codex → Magpie 127.0.0.1:3425 → DeepSeek API

添加没有预设的供应商

在 **Providers** → **Add provider** 选择 **Custom**，填写名称、API Key 和接口地址。按对方实际支持的协议填写，提供多个接口时可分别填写。

接口字段	基础地址示例	填写要点
OpenAI compatible	`https://gateway.example.com/v1`	使用供应商的 OpenAI 兼容基础地址；示例域名不能直接使用。
Anthropic compatible	`https://gateway.example.com`	填客户端使用的基础地址。供应商要求 `/anthropic` 前缀时保留它。
模型	供应商给出的模型 `ID`	从模型列表选择；手动输入时严格使用供应商的请求名称。

保存后分别到 Claude Code、Codex 的模型选择器中选择此供应商的模型。Magpie 会处理客户端协议与上游协议之间的转换。

验证与恢复

在练习目录中发送：“新建 HELLO.txt，内容仅为 hello harness；保存后重新读取并返回内容。”核对文件确实存在、内容正确，且 Magpie 记录的供应商与模型符合选择。

恢复官方模型时，在对应客户端的模型选择器中选回原生模型，或使用该客户端的 **Disconnect** 操作，再启动新会话。不要先退出网关却让客户端仍指向本机网关地址。

默认客户端配置：Claude Code 为 `%USERPROFILE%\.claude\settings.json`；Codex 为 `%USERPROFILE%\.codex\config.toml`。设置了自定义配置目录时，以客户端实际目录为准。

官方资料 中文快速上手 · 安装和配置参考 · DeepSeek 模型接入

7 用 CC Switch 配置第三方供应商

这里的 CC Switch 指 `farion1231/cc-switch`，官网是 `ccswitch.io`。它为每个客户端保存多套供应商配置：直连时把选中的配置写入客户端文件，需要转换接口协议时通过本地路由转发。以下使用 4.x 的左侧栏界面。

安装与首次打开

1 下载安装包

Windows x64 下载 `CC-Switch-v版本号-Windows.msi`；ARM 设备下载带 `Windows-arm64` 的 MSI。也可选择 Portable ZIP，完整解压后运行 `CC-Switch.exe`。

下载 CC Switch 官方下载页 · GitHub 发布页

2 确认前置条件

CC Switch 本身不要求 Node.js。Windows 需 10+；macOS 需 12+；Linux 需 glibc 2.35+ 和 WebKitGTK 4.1。先按前文安装要管理的 Claude Code、Codex。

macOS 可下载通用 DMG，或执行 `brew install --cask cc-switch`。Linux 根据发行版下载 `.deb`、`.rpm` 或 AppImage。

3 检查客户端识别

打开 CC Switch，在侧栏选择 **Claude Code** 或 **Codex**。若侧栏没有该工具，进入 **应用** 页，检查安装状态并启用侧栏显示。

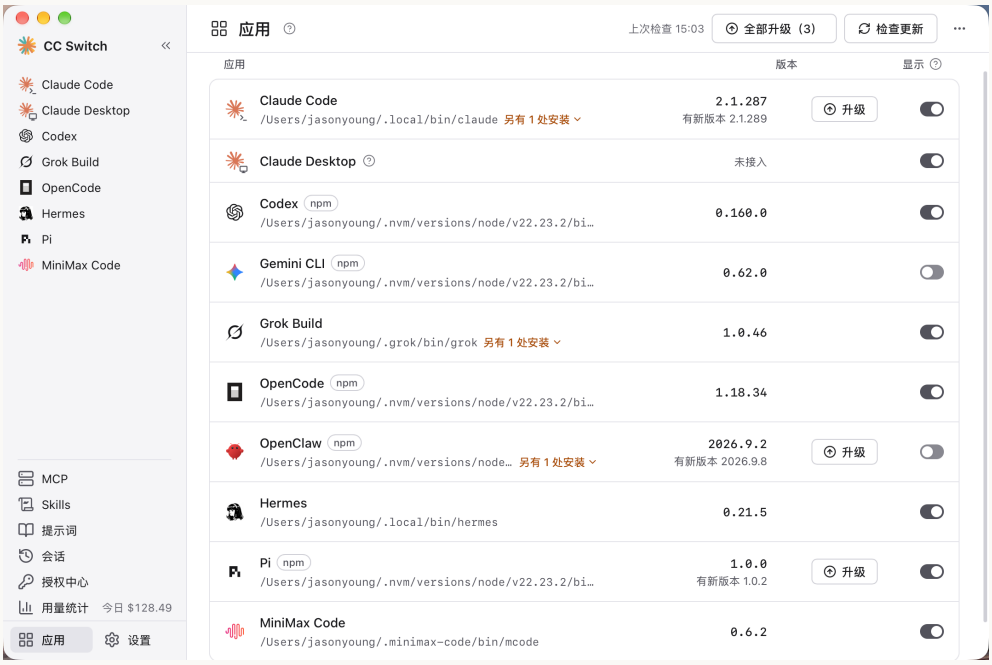


图 11 CC Switch 的应用管理页，可检查程序路径、已安装版本和侧栏显示开关。图中的版本号与路径为项目作者的示例。

图片来源 CC Switch 项目仓库 4.0 界面图

给 Claude Code 添加 DeepSeek

1 选择预设

侧栏选择 **Claude Code**，点击右上角 **添加供应商**。搜索 **DeepSeek**，选择对应的官方 API 预设。

2 填写凭据和模型

粘贴 DeepSeek API Key，核对预设的请求地址。Anthropic 兼容基础地址应对应 `https://api.deepseek.com/anthropic`，上游格式为 **Anthropic Messages**。选择或填入实际模型 ID。

如果表单提供主模型和 Sonnet、Opus、Haiku 映射，初次验证可统一使用 `deepseek-flash`；也可保留预设给出的有效映射。档位名称代表客户端选择，不代表上游仍是 Claude 模型。

3 保存并切换

点击页脚 **添加**。回到 **直连** 页，在这张供应商卡片上点击 **切换**，看到 **使用中** 状态。仅添加卡片不会替代这一步。

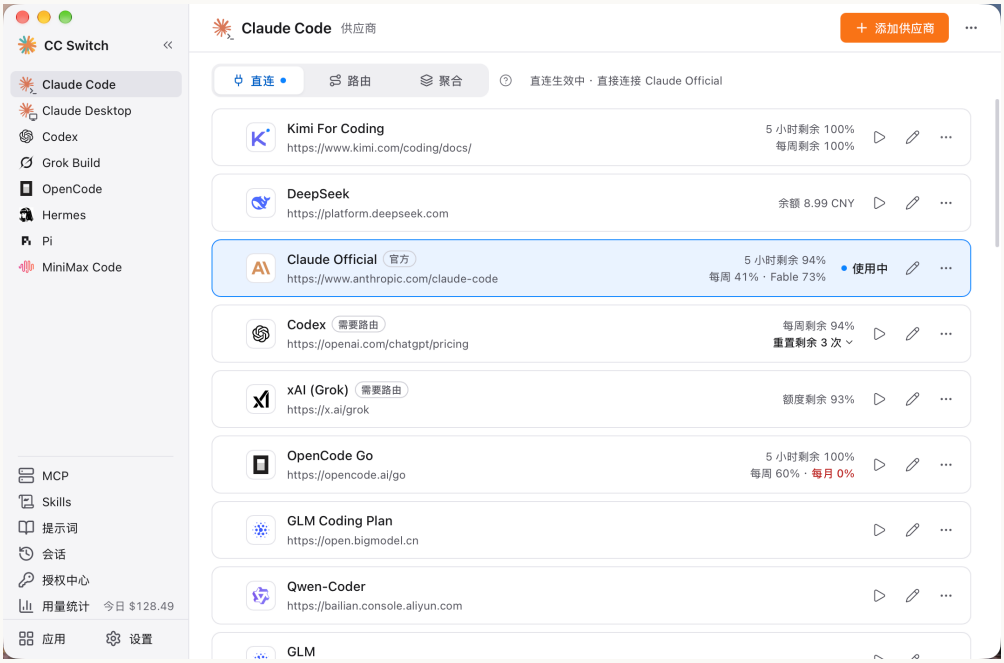


图 12 CC Switch 的供应商页面。作者示例当前启用 Claude Official；按本文切换后，DeepSeek 卡片应显示“使用中”，顶部状态也应随之更新。

图片来源 CC Switch 项目仓库 4.0 界面图

4 验证

启动 Claude Code 新会话，检查当前模型并发送 HELLO.txt 读写请求。Claude Code 支持配置热重载，首次配置仍用新会话确认实际连接。

官方资料 安装说明 · 添加供应商 · 切换方式

给 Codex 添加 DeepSeek

1 在 Codex 页单独添加

侧栏切到 **Codex** → **添加供应商**，搜索 **DeepSeek** 并选择预设。Claude Code 中已添加的卡片不会自动成为 Codex 的当前供应商。

2 核对协议和凭据

填写同一平台的 API Key，保留预设提供的 Responses 端点，确认 **上游格式** 为 **Responses 原生**。DeepSeek 当前已原生支持 Responses，因此这条接入路径可直连。

3 检查模型映射

在 **模型映射** 中保留或添加可用模型。下表展示同一个模型的填写方法：

字段	填写示例
菜单显示名	DeepSeek Flash
实际请求模型	`deepseek-flash`
默认模型	选择该映射；留空时使用映射表第一行。
上下文窗口与思考等级	使用预设或供应商明确给出的值，不用猜测数值。

4 切换并重启

点击 **添加**，再在 **直连** 页点击此卡片的 **切换**。退出已有 Codex CLI 再启动；桌面入口需彻底退出再打开。模型映射生成的列表也在重启时加载。

5 核对调用

输入 `/status` 检查模型和供应商，再做 HELLO.txt 读写验证。最后查看供应商控制台请求记录，确认实际使用了所选 API。

没有预设时使用自定义配置

在目标应用的 **添加供应商** 页选择 **自定义配置**，填写名称、请求地址和 API Key。接口协议由供应商文档决定，然后按下面对应字段配置。

客户端	关键字段	对应含义
Claude Code	`ANTHROPIC_BASE_URL`	Anthropic 兼容基础地址。
Claude Code	`ANTHROPIC_API_KEY` 或 `ANTHROPIC_AUTH_TOKEN`	按供应商要求选择密钥或 Bearer Token 认证，不混用两份不同凭据。
Claude Code	`ANTHROPIC_MODEL` 与各档位模型	使用上游真实模型 ID，或管理工具配置好的映射。
Codex	API Key 与 `base_url`	密钥填 API Key 区域；基础地址填供应商给出的 Codex 接口地址。
Codex	`model_provider` 与 `model_providers`	两处供应商标识一致；保留表单生成的结构。
Codex	`wire_api = "responses"`	Codex 客户端一侧保持 Responses；其他上游协议交给路由转换。

Codex 添加页的认证编辑区仍称为 `auth.json`；当前版本将第三方 Key 写入 `config.toml` 中供应商的 `experimental_bearer_token`。在表单中保存密钥即可。

上游协议不一致时开启路由

例如：给 Codex 接入只提供 Chat Completions 的网关，或给 Claude Code 接入只提供 Responses 的网关。此时需要 CC Switch 转换协议。

1 设置上游格式

编辑该供应商，在高级选项中将 **上游格式** 设为它实际支持的协议。不要把上游的 Chat Completions 填成 Responses。

2 真正开启路由

进入该应用页的 **路由** 标签，点击 **开始路由** 并确认；再在目标卡片上点击 **路由到此**。标有“需要路由”的卡片也可通过提示中的 **开始路由并使用它** 启用。

3 确认运行状态

在 **设置** → **本地路由** → **服务** 中确认服务正在运行。默认监听 `127.0.0.1:15721`。应用页的“生效中”与卡片的“路由中”显示当前实际状态。

Codex 的 Responses 请求 → CC Switch → 上游 Chat Completions

点击“直连 / 路由 / 聚合”标签只是在查看页面；**开始路由** 或 **回到直连** 才改变实际连接。路由运行期间保持 CC Switch 开启。

验证与回到官方服务

供应商卡片的 **更多** → **检测连通** 只检查地址可达性，401、403、404 也算收到了响应。最终仍需发送真实模型请求，并完成一次文件读取或写入。

表单中的 **获取模型列表** 需要供应商提供相应模型列表接口。返回 404 时，改为按供应商文档手动填写模型 ID，再测试实际对话。

恢复官方服务时，回到对应应用的 **直连** 页，使用 **回到直连** 或卡片上的 **回到直连并使用**，再切到官方供应商。Codex 重启后按需重新登录；Claude Code 可用 `/login` 重新认证。

若要改用 Magpie，先在 CC Switch 中退出路由或聚合，恢复客户端连接，再让 Magpie 设置该客户端。这样不会把两个本机网关串在一起。

官方资料 本地路由 · 模型与协议字段 · 连通检测的含义 · 配置文件位置

8 故障排查与配置备份

按现象检查

现象	下一步检查
找不到 claude、codex 或 agy	重新打开终端；运行 `Get-Command claude` 等对应命令，检查程序路径。安装器提示加入 PATH 的目录应存在。
npm.ps1 或 npx.ps1 被执行策略阻止	Windows 下改用 `npm.cmd`、`npx.cmd`。无需为这两个启动文件永久放开整个系统的脚本执行策略。
401 或 403	先看供应商返回的错误文字，核对 Key、认证方式、模型权限和账号状态。DSH 页面 403 另需检查启动链接与访问主机。
404 或接口不存在	核对协议、基础地址和路径，检查是否重复了 `/v1`。模型列表接口的 404 与聊天请求的 404 分开判断。
model not found	复制供应商文档中的精确模型 ID；刷新模型列表。Codex 修改模型映射后重启。
429 或额度不足	查看供应商返回的限流或余额信息；按提示等待重试，或调整模型、套餐与调用频率。
连接 127.0.0.1 失败	检查 Magpie 网关或 CC Switch 路由是否正在运行。Windows 客户端先连接 Windows 本机网关，避免混用 WSL 地址。
已切换但仍使用旧模型	确认卡片为“使用中”或“路由中”；检查环境变量冲突、配置目录与启动参数，再开启新会话。
能聊天但不能完成文件操作	确认所选模型支持工具调用，查看工具调用返回的具体错误；检查协议转换与项目运行环境。
DSH 页面能开但不能输入	先添加并选中工作区，再选择有凭据的模型。

备份哪些文件

首次让管理工具接管前，将已有配置复制到个人备份目录。不存在的文件无需创建。下面是 Windows 默认位置；使用自定义目录时备份对应位置。

工具	文件或目录
Claude Code	`%USERPROFILE%\.claude\settings.json`
Codex	`%USERPROFILE%\.codex\config.toml` 以及该文件引用的自定义模型目录 JSON。
Magpie	`%USERPROFILE%\.config\magpie`
CC Switch	`%USERPROFILE%\.cc-switch` 也可使用“设置 → 数据 → 备份与恢复”。
DSH	`%USERPROFILE%\.dsh` 设置了 `DSH_HOME` 时备份该目录。

配置与备份中包含 API Key 或登录信息时，保存在个人可控的位置；向他人求助时只分享删去密钥后的错误信息与配置片段。

官方资料 Claude 安装排错 · Codex 配置 · CC Switch 环境变量冲突 · Magpie 配置文件

附录 社区教程与练习

下面的资料可配合对应章节阅读，包含作者的操作记录、完整演示和练习项目。安装命令、界面字段与快捷键已经在正文按当前项目文档核对。

Claude Code 的首次使用与日常操作

中文教程 想风 超详细 Claude Code 终端教程

涵盖启动登录、第三方接口、文件引用、会话恢复、权限与快捷键。本教程补入了对应的输入位置说明、常用命令表和终端演示图。

Codex 的完整上手路线

中文教程 Vink567 等 Codex 橙皮书

可阅读其中的“Codex CLI 安装与上手”以及登录、安装排错章节，再按兴趣查看完整项目案例。原稿有 npm 安装路径；本教程使用原生安装脚本时无需预装 Node.js。

Antigravity CLI 的命令速查与练习

英文教程 Kenneth Kousen 培训课程 · 命令速查表 · 动手练习

内容包括交互模式、一次性提问、文件引用和项目练习。课程仓库提到的 Node.js 用于运行幻灯片；安装 Antigravity 原生 CLI 无需先安装课程环境。

DSH 的不同运行方式

社区指南 DeepSeekDSH 命令与 Profile 指南 · DSH 中文社区 快速上手

适合对照理解 Web 服务、Headless 任务和 Profile。正文补入了全局 CLI 安装、Web 换端口、Headless 凭据来源，以及服务的退出方式。

DSH 的终端聊天界面

插件作者教程 ccch1mneyyy dsh-TUI 中文入门

包含 npm 安装、pnpm 前置条件、历史会话和诊断命令。其他教程中的 `dsh --profile tui` 只有在对应 Profile 已安装时才能运行；本文采用作者当前使用的 `dsh-tui` 包与入口。

遇到不同写法时

先确认教程的操作系统、安装方式与发布日期。例如 `npm install` 和原生安装脚本的前置条件不同，`$env:KEY=...` 属于 PowerShell，`export KEY=...` 属于 Bash。命令能否执行，要看本机安装的版本与它的帮助输出。

附录 图文来源

下列页面提供正文的安装依据与原始界面图。Claude Code、Antigravity 的终端图取自官方演示动图的一帧；图片下方均保留原始来源。社区教程另见前一附录。

1 Anthropic Claude Code Quickstart

用于 Claude Code 的安装和首次登录说明。终端截图取自 anthropics/claude-code 仓库的 demo.gif，交互与恢复命令另链接当前官方参考。

2 OpenAI Codex CLI

用于 Codex CLI 的安装入口与首次启动说明。终端配图来自 openai/codex 仓库的 codex-cli-splash.png；命令说明对应当前官方 CLI 参考。

3 Google Antigravity Getting started · 官网界面图

用于项目、目录和 Local／New Worktree 模式说明。桌面配图来自 Google 官网，终端配图来自 google-antigravity/antigravity-cli 的 agy-cli-demo.gif。

4 DeepSeek Harness 桌面下载 · 桌面说明

用于桌面安装、系统要求和退出说明；桌面配图来自 DeepSeek 下载页。CLI 与 Headless 的命令、凭据来源另链接官方源码文档。

5 yetone Magpie 中文快速上手 · 项目许可证

用于 Providers、Agents、模型选择和重启说明；两张配图均为作者快速上手指南中的原始界面图。

6 CC Switch 项目说明 · 添加供应商教程 · 项目许可证

用于客户端识别、供应商管理和连接模式说明；两张配图来自仓库的 4.0 版本界面图。

7 dsh-TUI 项目说明 · 项目许可证

用于社区 TUI 的安装与操作说明；配图为仓库 screenshots/social-preview.png。

开源素材许可

Magpie、CC Switch 与 dsh-TUI 的上述项目素材采用 MIT License，以下保留版权与许可声明。其他产品演示图的权利归各原权利人，出处见正文图下链接。

MIT License
Copyright (c) 2026 yetone
Copyright (c) 2025 Jason Young
Copyright (c) 2026, chimney (ccch1mneyyy)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.