OpenClaw安装与使用教程

1. OpenClaw 是什么

OpenClaw（原 Clawdbot/Moltbot）是一款开源 AI 智能体，核心是**用自然语言指挥电脑执行真实操作**，实现从 “理解意图→规划任务→落地执行→反馈结果” 的完整闭环。它的应用场景覆盖**个人效率、办公协作、开发运维、垂直行业**四大类：

命名寓意：“Open”代表开源社区驱动；“Claw”象征灵活强大的执行能力。

核心优势：高可塑性、全模型适配、跨平台协同、主动执行、持久记忆，核心架构由 Gateway、Agent、Skills、Memory 四部分组成。

OpenClaw的应用场景：

一、个人效率与生活自动化（最易上手）

- **办公自动化**
  - 邮件管理：自动分类、提取信息、生成回复、定时发送。
  - 文档处理：生成报告 / 简历 / 会议纪要，批量格式转换、提取要点。
  - 日程管理：创建日程、设置提醒，自动同步会议邀请。
  - 信息整理：从网页 / PDF / 图片提取文本，整理为表格 / 清单。
  - 周报 / 日报：自动汇总工作内容，生成结构化周报。
- **日常事务处理**
  - 文件管理：批量重命名、分类整理、自动备份。
  - 信息查询：自动查天气、航班、快递，汇总行业资讯。
  - 定时任务：自动关机、备份、下载订阅内容。

- 报销自动化：OCR 识别收据，自动分类并更新报销表。
- 个人知识管理：整理碎片信息，构建个人知识图谱。

二、企业办公与团队协作（全行业通用）

- **团队协作**
  - 任务管理：同步任务清单、跟踪进度、发送催办提醒。
  - 协作文档：自动汇总成员文档，统一格式并共享。
  - 会议支持：生成议程、记录纪要、自动分发、同步日程。
  - 工作简报：定时汇总进展，生成团队简报并推送。
- **行政与 HR**
  - 简历筛选：自动解析简历，匹配岗位要求。
  - 新员工入职：提供入职指引、资料收集、流程自动化。
  - 差旅 / 报销：自动处理差旅申请、费用核算、报销流程。

三、软件开发与 IT 运维（开发者专属）

- **开发辅助**
  - 代码审查：自动审查 PR，检查规范、漏洞、优化点。
  - 代码重构：辅助遗留系统跨语言重构、代码优化。
  - API 文档：自动同步 API 文档，保持代码与文档一致。
  - 数据库迁移：辅助 Schema 迁移、数据校验、回滚预案。
  - 代码搜索：全局语义搜索，快速定位代码片段。
- **运维自动化**
  - 基础设施：生成 Terraform 脚本、K8s 配置、Nginx 规则。
  - 日志分析：监控 K8s 异常日志，自动告警与定位问题。
  - CI/CD 优化：自动调优流水线，提升构建与部署效率。
  - 云成本优化：巡检云资源，识别闲置与浪费，给出优化建议。
  - 异常检测：实时监控业务指标，异常自动告警与初步排查。

四、垂直行业定制化应用（深度落地）

- **内容创作 / 新媒体**
  - 热点监控：自动扫描全网热点，生成社交媒体内容。
  - 内容分发：长视频自动切片、适配平台、添加标签、排期发布。

- 品牌监控：24 小时监控品牌提及，情感分析，及时处理差评。
- 多平台运营：自动管理微信、微博、抖音等矩阵账号。

- **电商零售**
  - 竞品监控：自动抓取竞品价格、库存、活动信息。
  - 订单处理：自动同步订单、发货、售后、评价管理。
  - 数据报表：自动生成销售、库存、用户分析报表。
- **金融 / 投资**
  - 市场监控：7×24 小时监控行情、新闻、公告，自动预警。
  - 数据整理：自动汇总财报、研报，生成投资分析报告。
  - 交易辅助：执行预设交易策略，自动下单、止盈止损。
- **法律 / 财税**
  - 合同处理：自动生成、审查、比对合同，提取关键条款。
  - 票据处理：OCR 识别发票、单据，自动分类、记账、报税。
  - 法规检索：自动检索相关法规、判例，生成法律意见书。
- **教育 / 培训**
  - 助教助手：自动批改作业、答疑、生成学习报告。
  - 资源管理：自动整理课件、题库、试卷，智能分发。
  - 学习规划：根据学生情况，自动生成个性化学习计划。
- **工业 / 物联网**
  - 设备巡检：驱动巡检机器人 7×24 小时作业，自动上报异常。
  - 数据采集：自动采集工业传感器数据，实时分析与预警。
  - 流程控制：优化生产流程，降低能耗，提升效率。

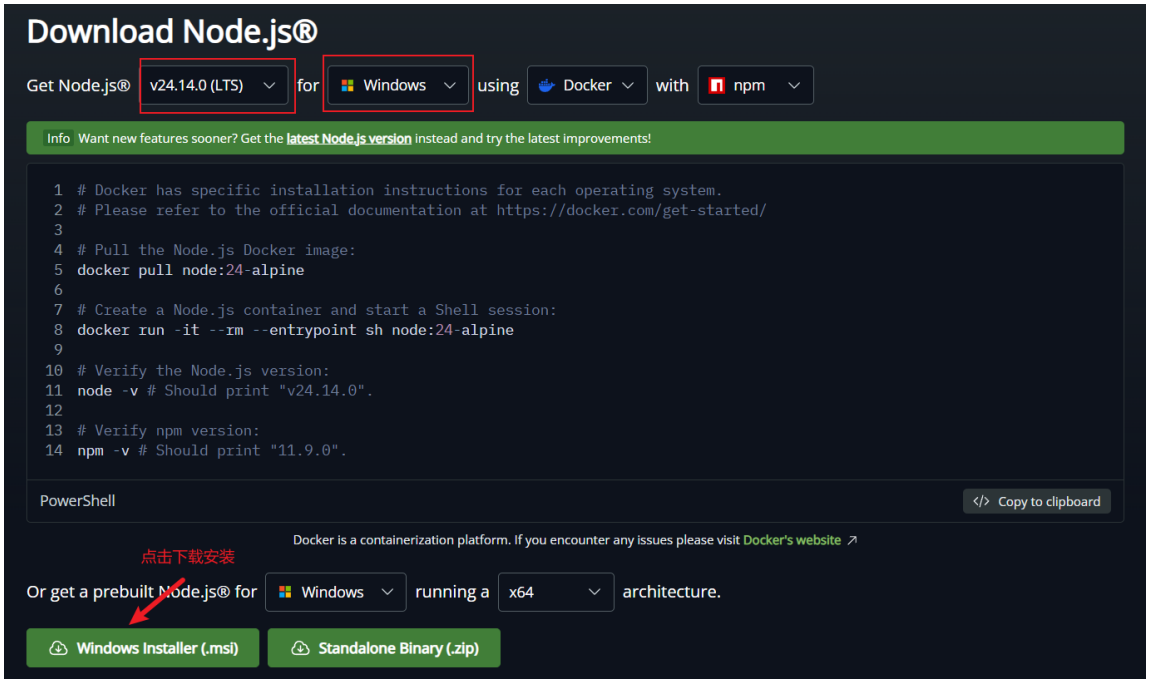
2. OpenClaw 安装

支持 macOS、Linux、Windows、树莓派，新手优先选 npm 安装或一键脚本，开发者可选源码安装，安装前完成简单前置准备即可。

2.1 前置准备（必做）

一、安装Nodejs

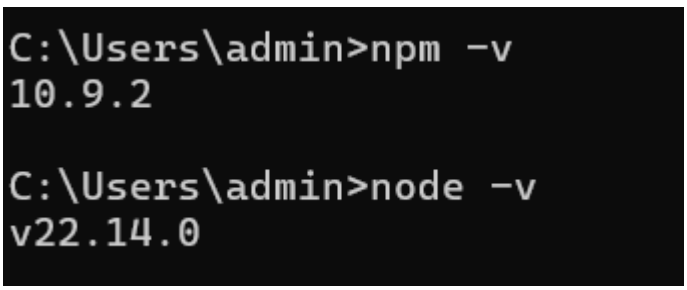
1. 下载地址：https://nodejs.org/en/download



2. 下载的软件安装包：

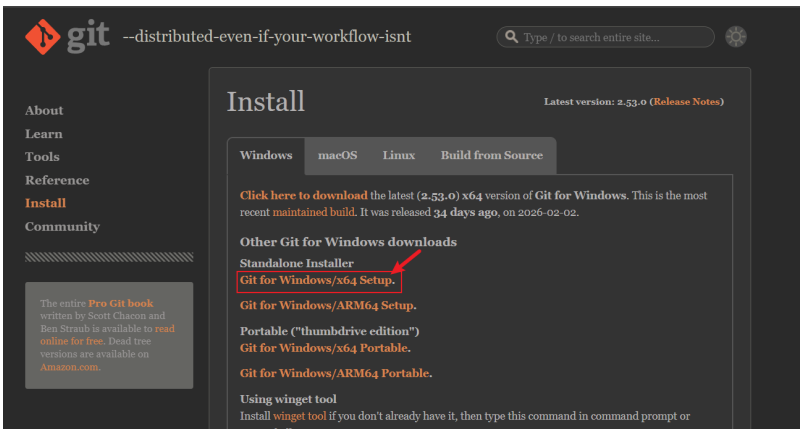
3. 双击安装，一直下一步就可以

4. 验证安装结果



二、安装Git

1. 下载地址：https://git-scm.com/install/windows



2. 下载的软件安装包

3. 双击安装，一直下一步就可以

4. 验证安装

```
C:\Users\admin>git --version
git version 2.35.1.windows.2

C:\Users\admin>
```

三、windows权限设置

代码块

```
1   Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

```
PS C:\WINDOWS\system32> Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser

执行策略更改
执行策略可帮助你防止执行不信任的脚本。更改执行策略可能会产生安全风险，如 https:/go.microsoft.com/fwlink/?LinkID=135170
中的 about_Execution_Policies 帮助主题所述。是否要更改执行策略?
[Y] 是(Y)  [A] 全是(A)  [N] 否(N)  [L] 全否(L)  [S] 暂停(S)  [?] 帮助 (默认值为“N”): y
PS C:\WINDOWS\system32>
```

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
这句代码的作用

Qwen3.5-Plus

这条 PowerShell 命令的作用是：**允许当前用户运行本地创建的脚本，以及由受信任发布者签名的远程脚本。**

2.2 安装方式

openclaw官网：https://docs.openclaw.ai/zh-CN

中文社区版：https://clawd.org.cn/

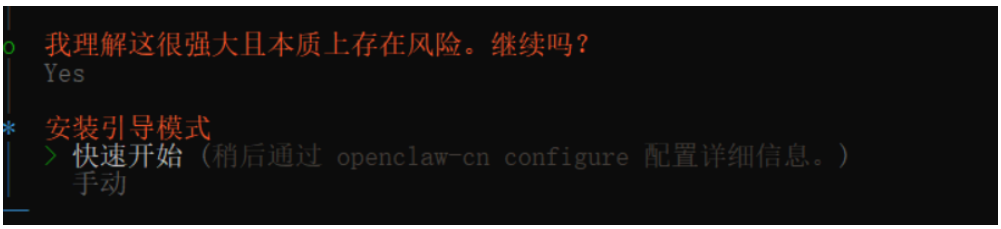
代码块

```
1   # 全局安装（使用国内镜像）
2   npm install -g openclaw-cn@latest --registry https://registry.npmmirror.com
3
```

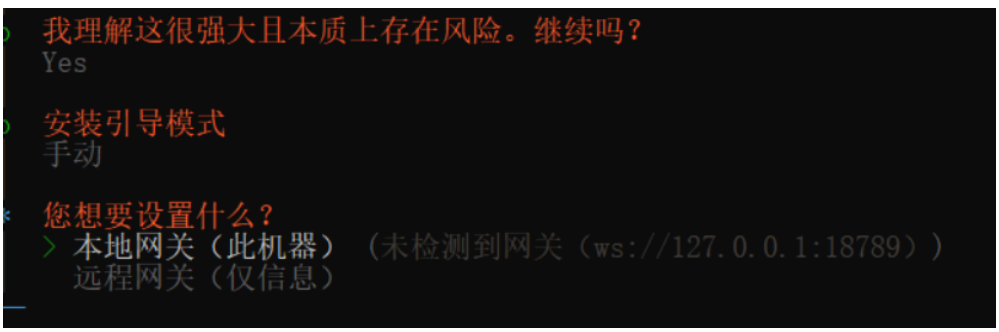
```
#检查安装的脚本
openclaw-cn --version

# 运行向导
openclaw-cn onboard --install-daemon
```

点击键盘的左右箭头，选择yes后敲回车

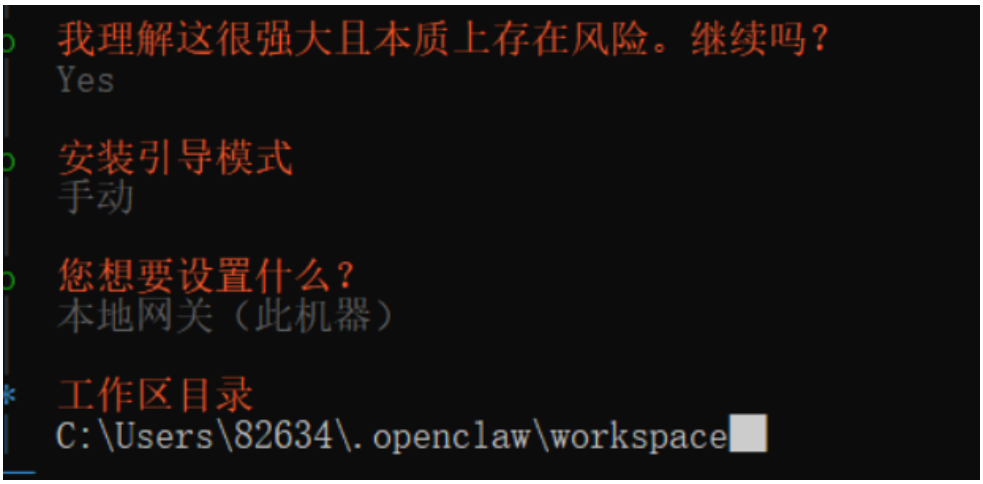
敲上下箭头，选择“手动”后敲回车



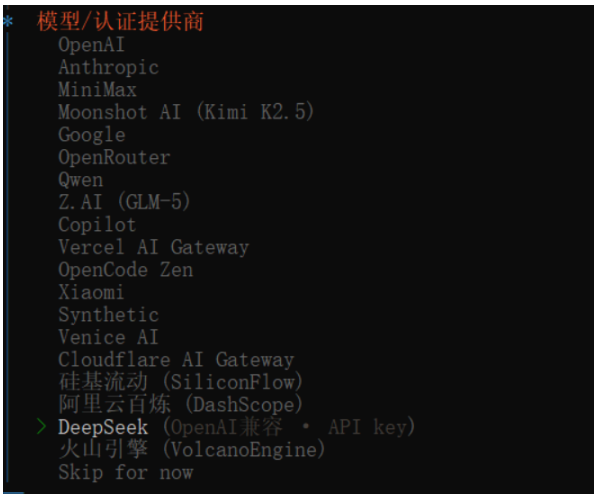
选“本地网关”



默认工作区目录，直接敲回车



选择deepseek模型，敲回车

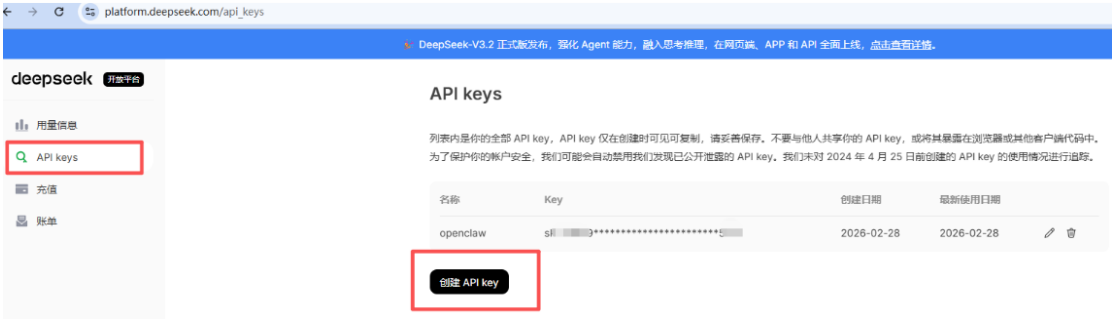


敲回车后如图

选择“DeepSeek API key”后回车

此时，要求输入deepseek官网的API key，所以我们需要登录deepseek找到自己的APIkey，打开deepseek官网

创建自己的APIkey

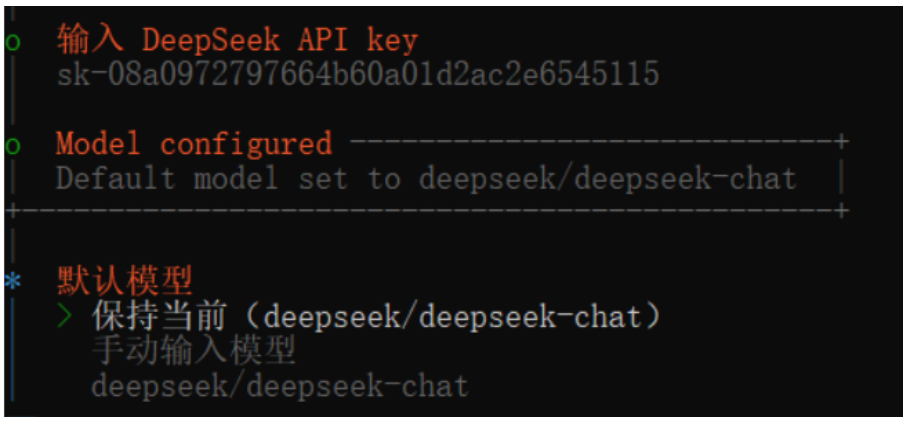


注意：

1. 创建后一定记得复制出来保存一下，因为只能复制一次，以后就不能复制了。
2. 记得充值，不充值的话，后边是无法调用大模型的

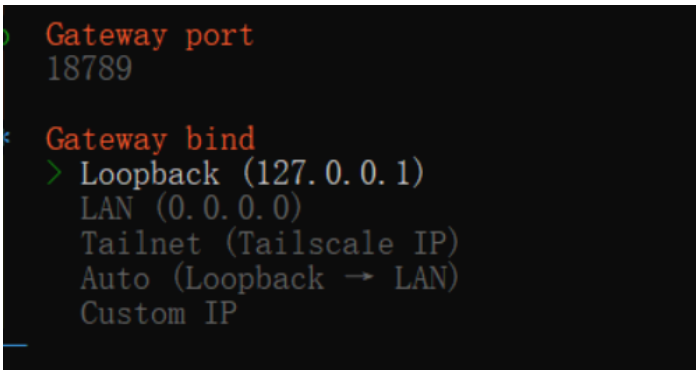
将我们复制好的APIkey，粘贴到cmd窗口里

敲回车后如图，选择“保持当前（deepseek/deepseek-chat）”

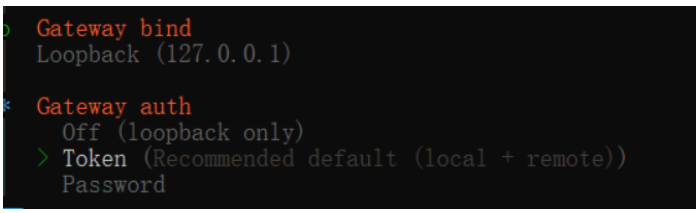


端口不需要改，直接回车

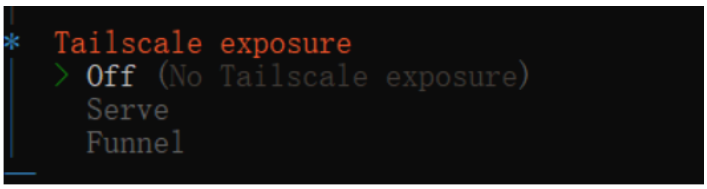
选择“Loopback (127.0.0.1)”，回车



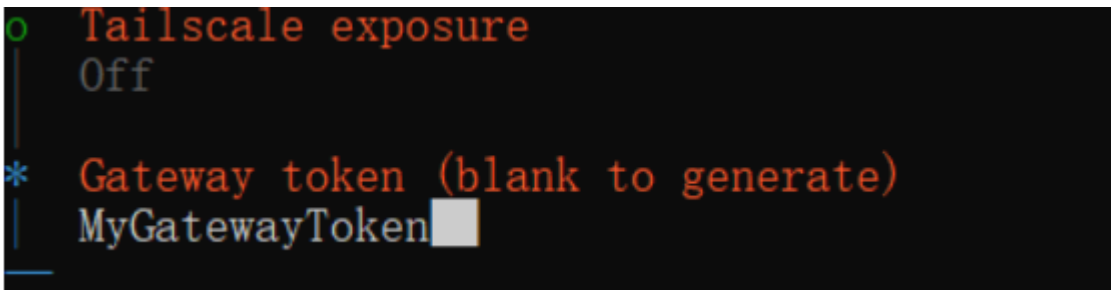
选择“Token”，回车



选择“Off”，回车

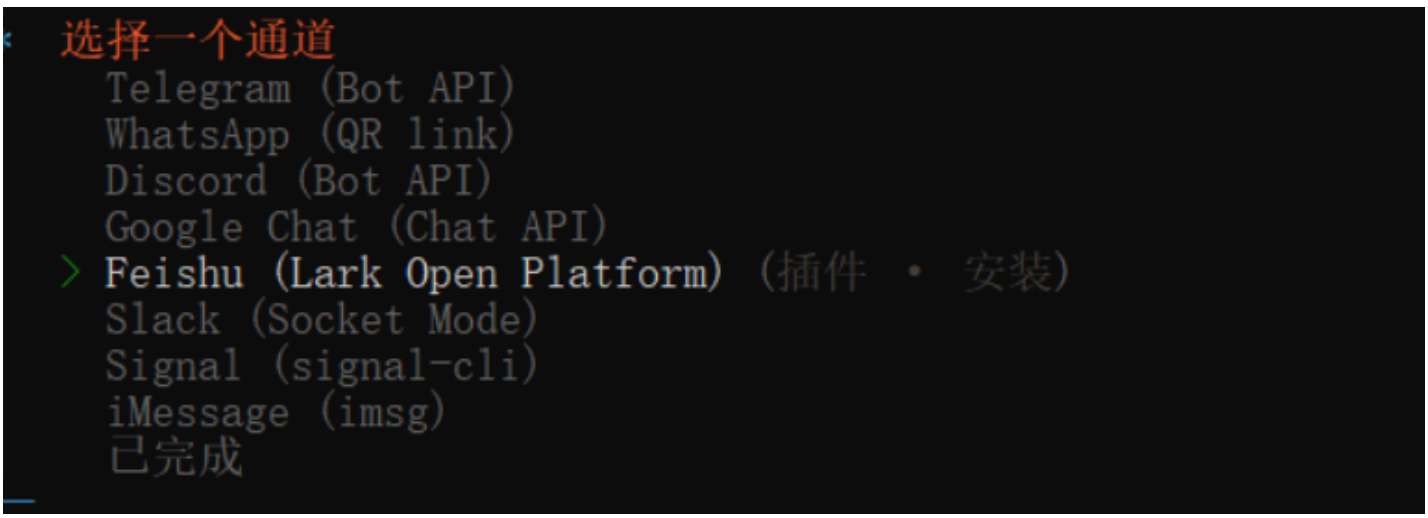


这里需写一个网关名称，可以随意写，我写的是 MyGatewayToken，如图



回车后，提示开始配置聊天通道，选择“yes”后回车

这里我们选择飞书



选择使用本地插件路径

选择“飞书国内版”

此时，提示要输入飞书AppID

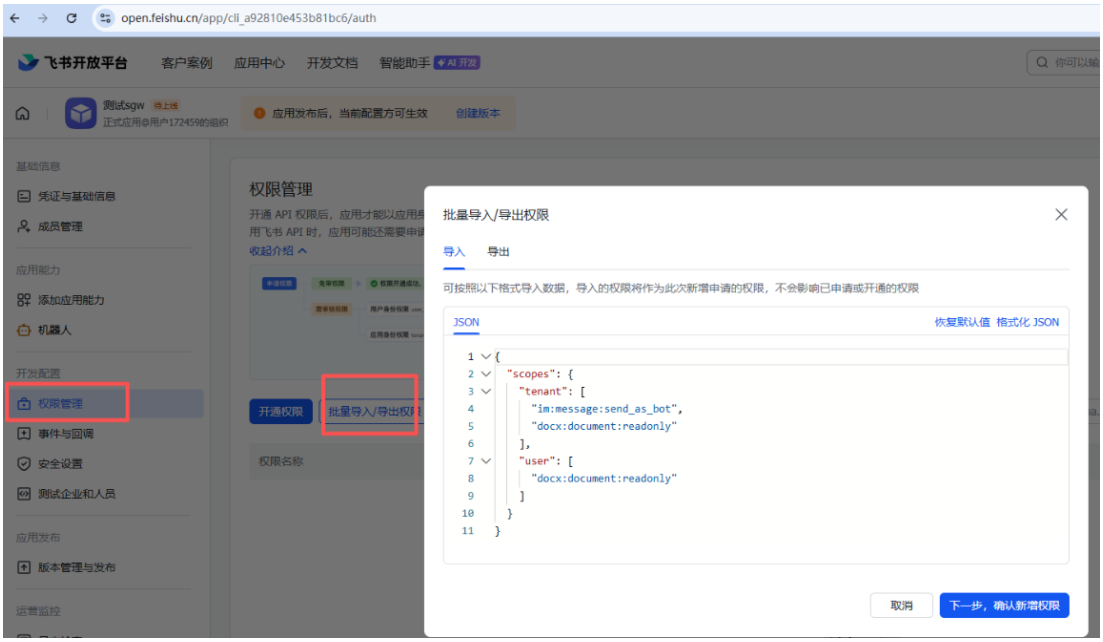
我们需要打开飞书开放平台，登录自己的账号

https://open.feishu.cn/app

这里信息随便填

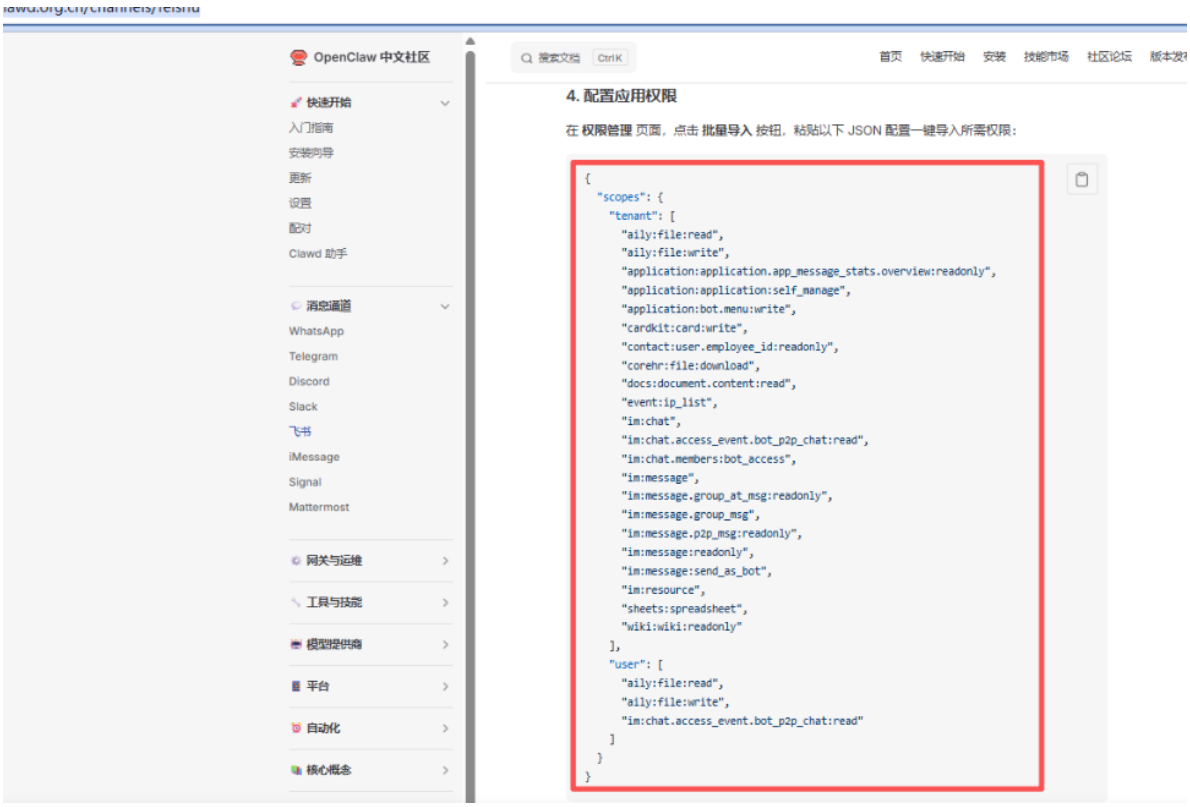
添加机器人

权限管理→批量导入权限

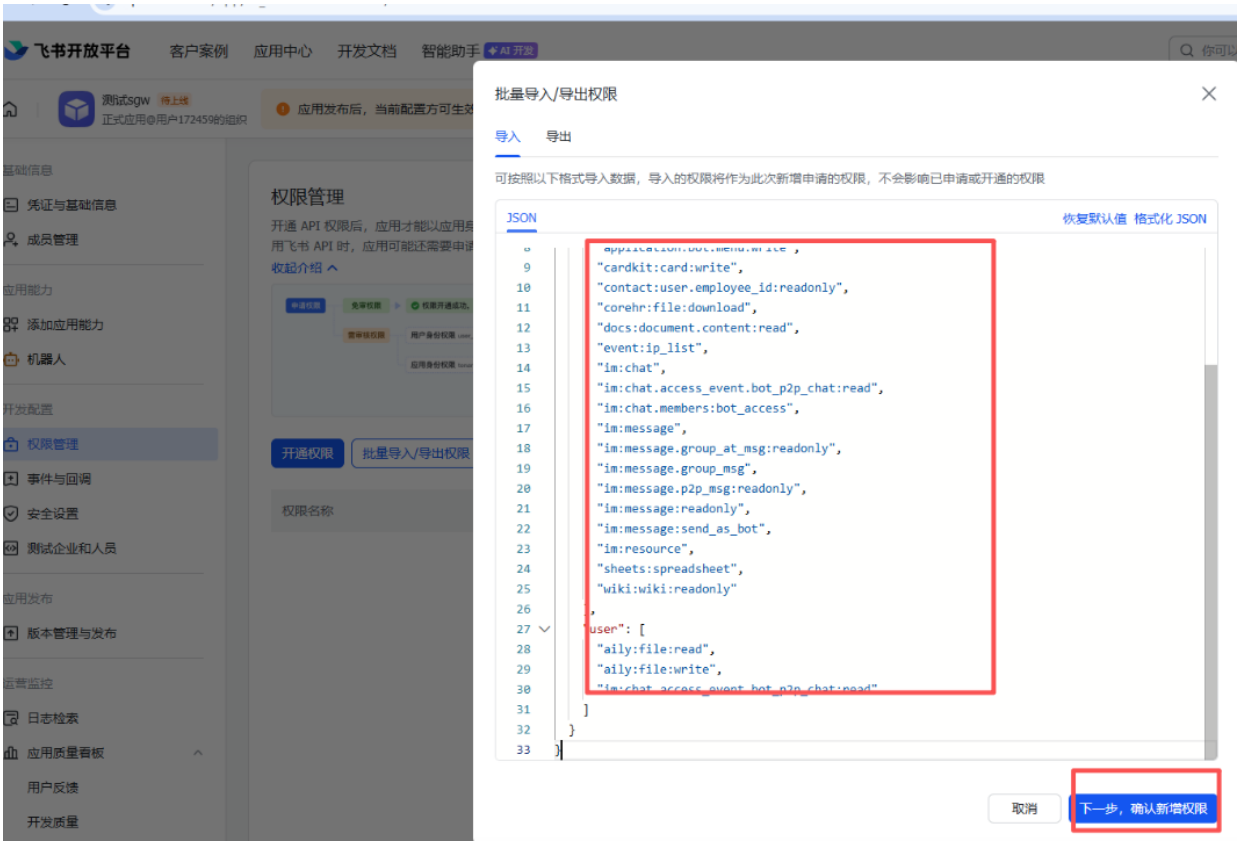


打开飞书官网，寻找权限的json格式数据，权限官网，打开后，往下划动页面，复制如下内容

https://clawd.org.cn/channels/feishu



粘贴进来（注意，要先删除之前的配置，然后再粘贴进来）


飞书开放平台
客户案例
应用中心
开发文档
智能助手
批量导入/导出权限
导入
导出
可按照以下格式导入数据，导入的权限将作为此次新增申请的权限，不会影响已申请或开通的权限
JSON
恢复默认值 格式化 JSON
"cardkit:card:write",
"contact:user.employee_id:readonly",
"corehr:file:download",
"docs:document.content:read",
"event:ip_list",
"im:chat",
"im:chat.access_event.bot_p2p_chat:read",
"im:chat.members:bot_access",
"im:message",
"im:message.group_at_msg:readonly",
"im:message.group_msg",
"im:message.p2p_msg:readonly",
"im:message:readonly",
"im:message:send_as_bot",
"im:resource",
"sheets:spreadsheet",
"wiki:wiki:readonly"
"user": [
"aily:file:read",
"aily:file:write",
取消
下一步，确认新增权限

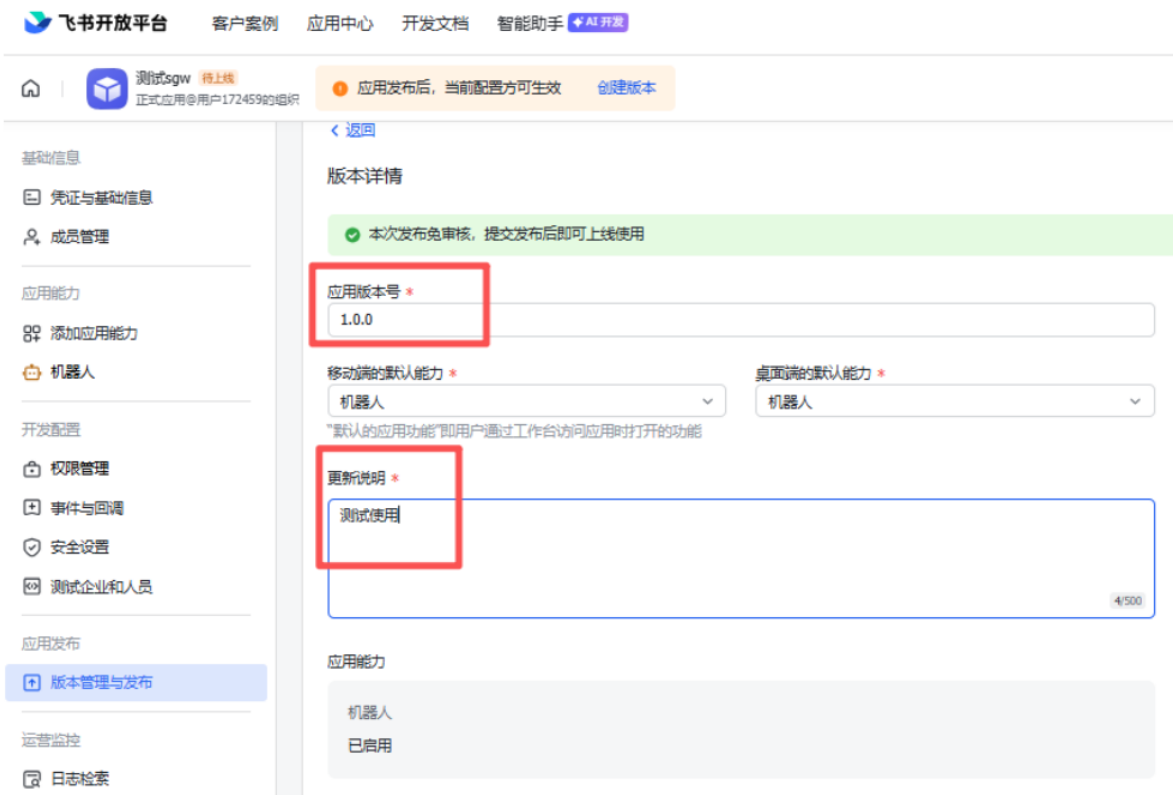
申请开通


权限管理
开通 API 权限后，应用才能以应用身
用飞书 API 时，应用可能还需要申请
收起介绍
开通权限
批量导入/导出权限
权限名称
确认导入权限
应用身份权限
用户身份权限
本次新增权限（22）
获取文件
上传文件
管理应用自身资源
获取机器人发送消息的概览数据
创建、更新、删除机器人菜单
创建与更新卡片
获取用户 user ID
下载文件
查看云文档内容
获取事件的出口 IP
获取与更新群组信息
订阅访问机器人会话相关事件
订阅机器人进、出群事件
获取与发送单聊、群组消息
获取单聊、群组消息
以应用的身份发消息
接收群聊中@机器人消息事件
获取群组中所有消息（敏感权限）
读取用户发给机器人的单聊消息
获取与上传图片或文件资源
查看、评论、编辑和管理电子表格
查看知识库
取消
上一步，导入权限
申请开通

点击“创建版本”

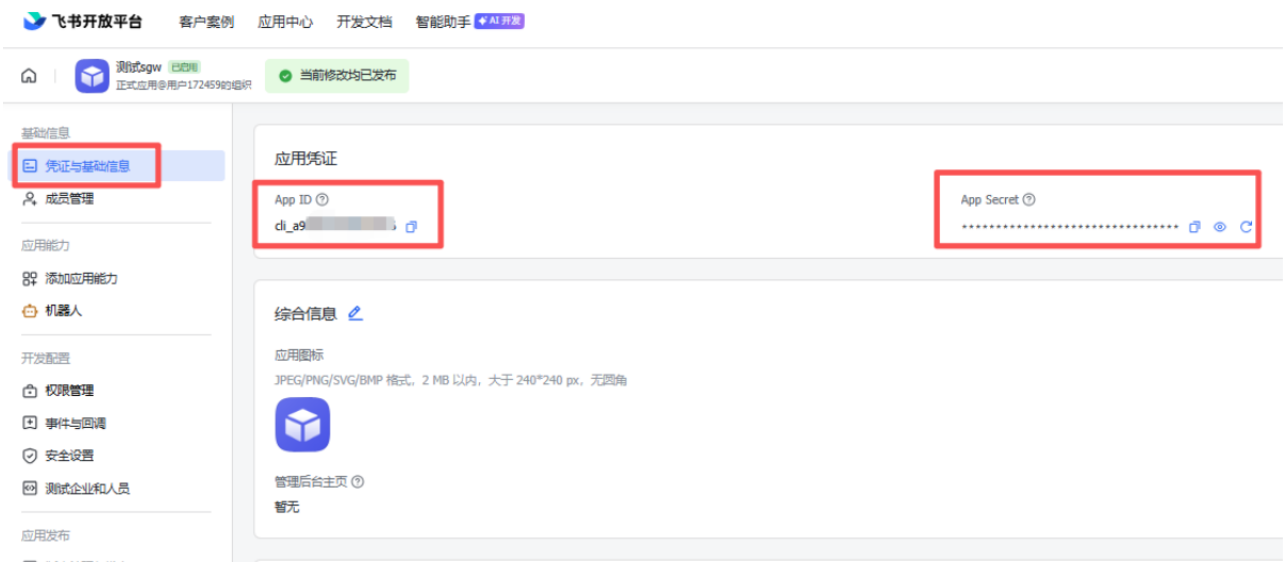

飞书开放平台
客户案例
应用中心
开发文档
智能助手
AI 开发
测试sgw
待上线
正式应用@用户172459的组织
应用发布后，当前配置方可生效
创建版本
基础信息
凭证与基础信息
成员管理
应用能力
添加应用能力
机器人
开发配置
权限管理
事件与回调
安全设置
测试企业和人员
应用发布
版本管理与发布
运营监控
权限管理
开通 API 权限后，应用才能以应用身份（tenant_access_token）或用户身份（user_access_token）调用飞书 API；以应用身份调用飞书 API 时，应用可能还需要申请对应的数据权限。了解更多
收起介绍
开通权限
批量导入/导出权限
例如：获取群组信息
权限名称
权限类型
权限状态
可访问的数据范围 配置
获取文件
aily:file:read
应用身份
已开通
获取文件
aily:file:read
用户身份
已开通
与用户权限范围一致
上传文件
应用身份
已开通

填写信息后，保存


飞书开放平台
客户案例
应用中心
开发文档
智能助手
AI 开发
测试sgw
待上线
正式应用@用户172459的组织
应用发布后，当前配置方可生效
创建版本
返回
版本详情
本次发布免审核，提交发布后即可上线使用
应用版本号 *
1.0.0
移动端的默认能力 *
机器人
桌面端的默认能力 *
机器人
"默认的应用功能"即用户通过工作台访问应用时打开的功能
更新说明 *
测试使用
4/500
应用能力
机器人
已启用
基础信息
凭证与基础信息
成员管理
应用能力
添加应用能力
机器人
开发配置
权限管理
事件与回调
安全设置
测试企业和人员
应用发布
版本管理与发布
运营监控
日志检索

发布

点击“凭证与基础信息”，得到AppID和AppSecret的值，将这俩值复制出来备用

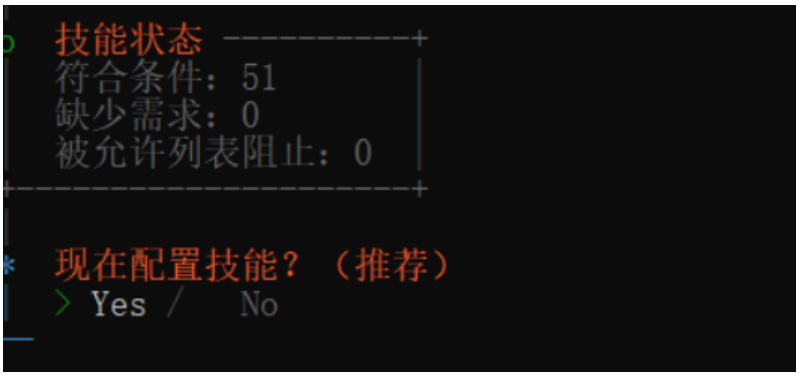


回到CMD窗口，在这里粘贴上AppID的值后回车

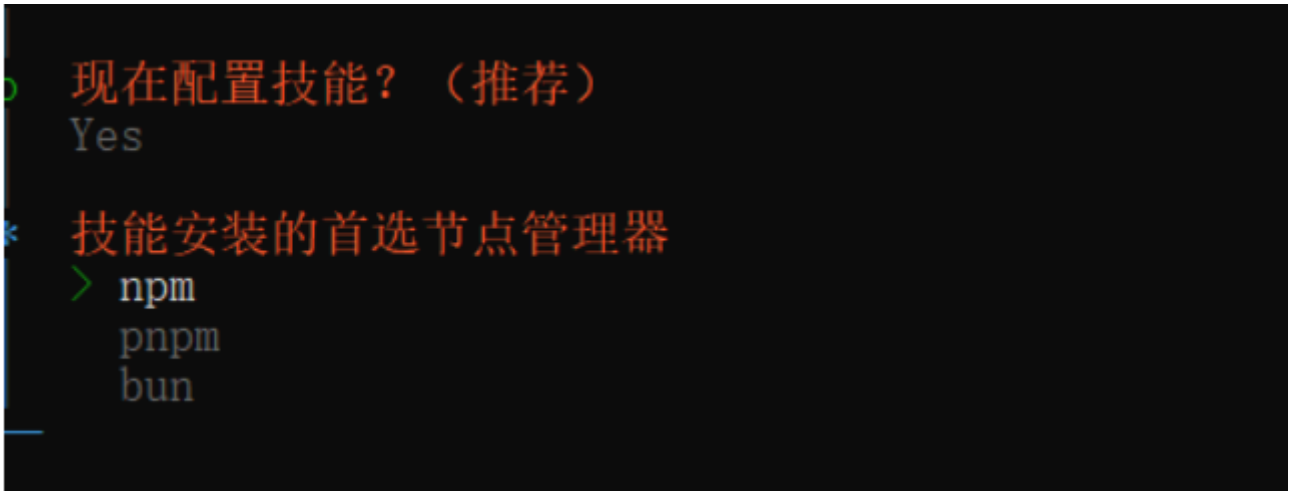
再粘贴上AppSecret的值后回车

选择“已完成”回车

选择yes回车



选择npm回车

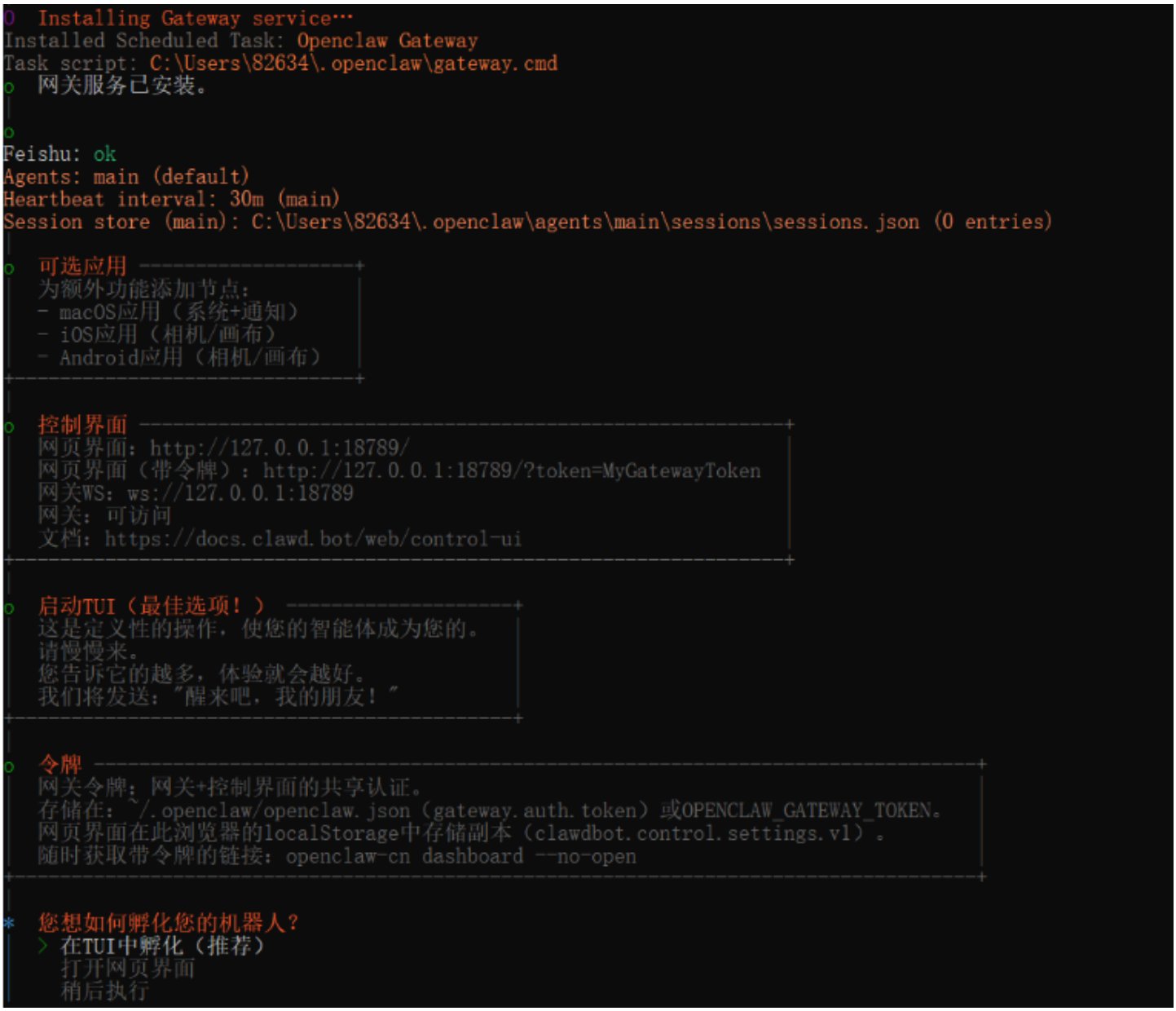


回车后看到如下图

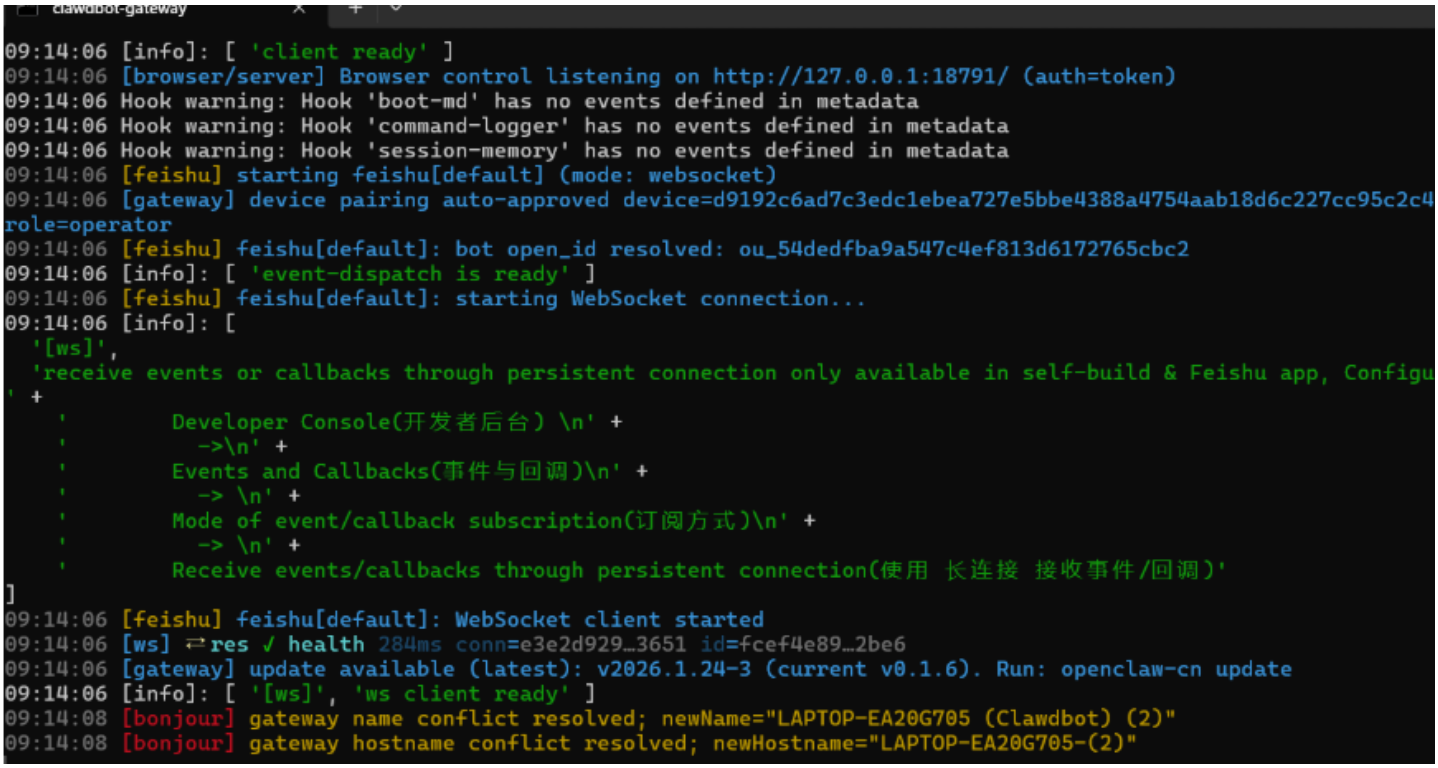
需要注意，这里是多选，不能直接回车，直接回车会有一堆问题。我们使用上下箭头，然后敲击space空格键选择boot-md、command-logger、session-memory

如上图，变为+后代表选中了，然后敲回车结果如下。

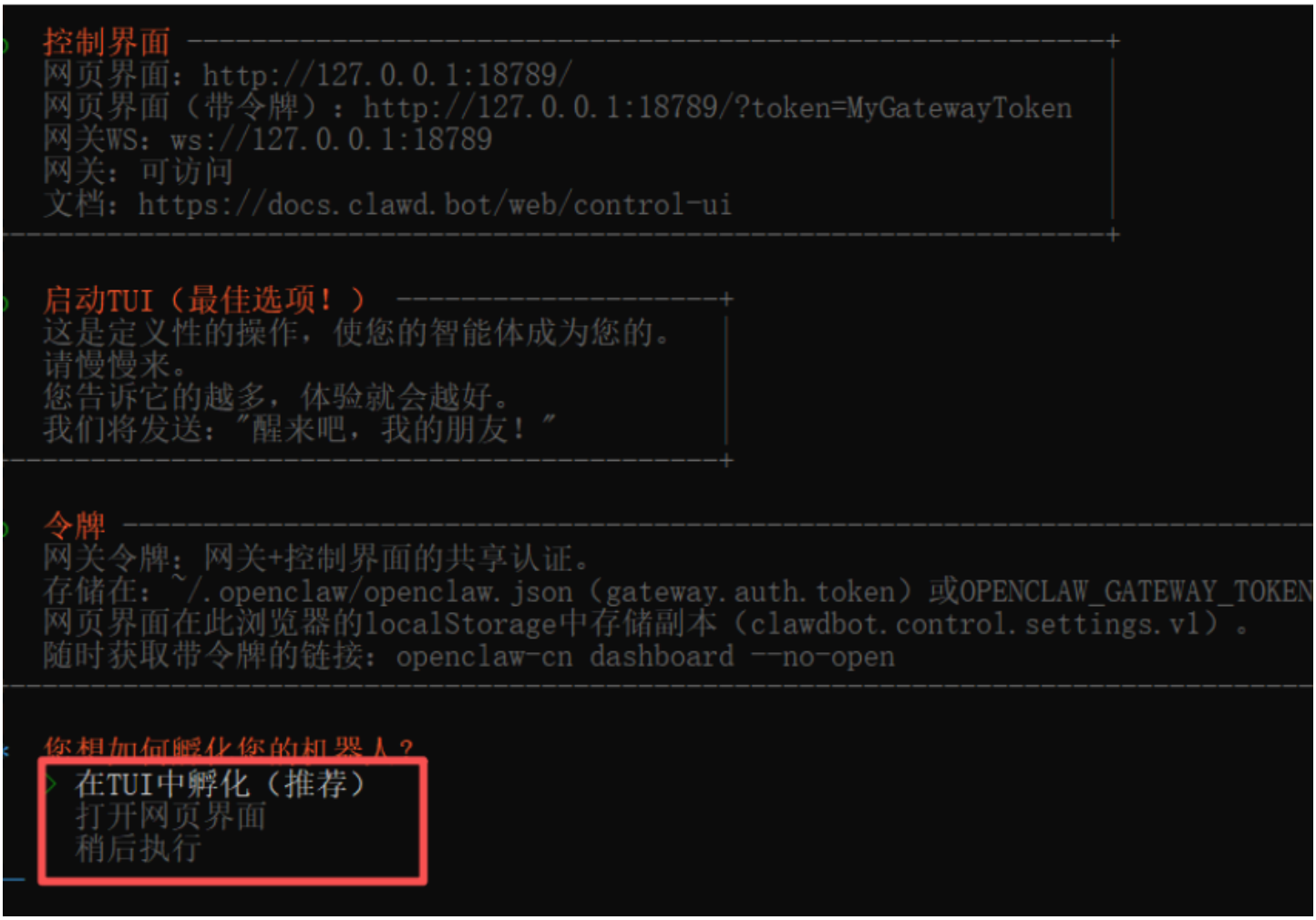
继续敲回车


Installing Gateway service…
Installed Scheduled Task: Openclaw Gateway
Task script: C:\Users\82634\.openclaw\gateway.cmd
网关服务已安装。
Feishu: ok
Agents: main (default)
Heartbeat interval: 30m (main)
Session store (main): C:\Users\82634\.openclaw\agents\main\sessions\sessions.json (0 entries)
可选应用
为额外功能添加节点：
- macOS应用（系统+通知）
- iOS应用（相机/画布）
- Android应用（相机/画布）
控制界面
网页界面：http://127.0.0.1:18789/
网页界面（带令牌）：http://127.0.0.1:18789/?token=MyGatewayToken
网关WS：ws://127.0.0.1:18789
网关：可访问
文档：https://docs.clawd.bot/web/control-ui
启动TUI（最佳选项！）
这是定义性的操作，使您的智能体成为您的。
请慢慢来。
您告诉它的越多，体验就会越好。
我们将发送："醒来吧，我的朋友！"
令牌
网关令牌：网关+控制界面的共享认证。
存储在：~/.openclaw/openclaw.json（gateway.auth.token）或OPENCLAW_GATEWAY_TOKEN。
网页界面在此浏览器的localStorage中存储副本（clawdbot.control.settings.v1）。
随时获取带令牌的链接：openclaw-cn dashboard --no-open
您想如何孵化您的机器人？
在TUI中孵化（推荐）
打开网页界面
稍后执行

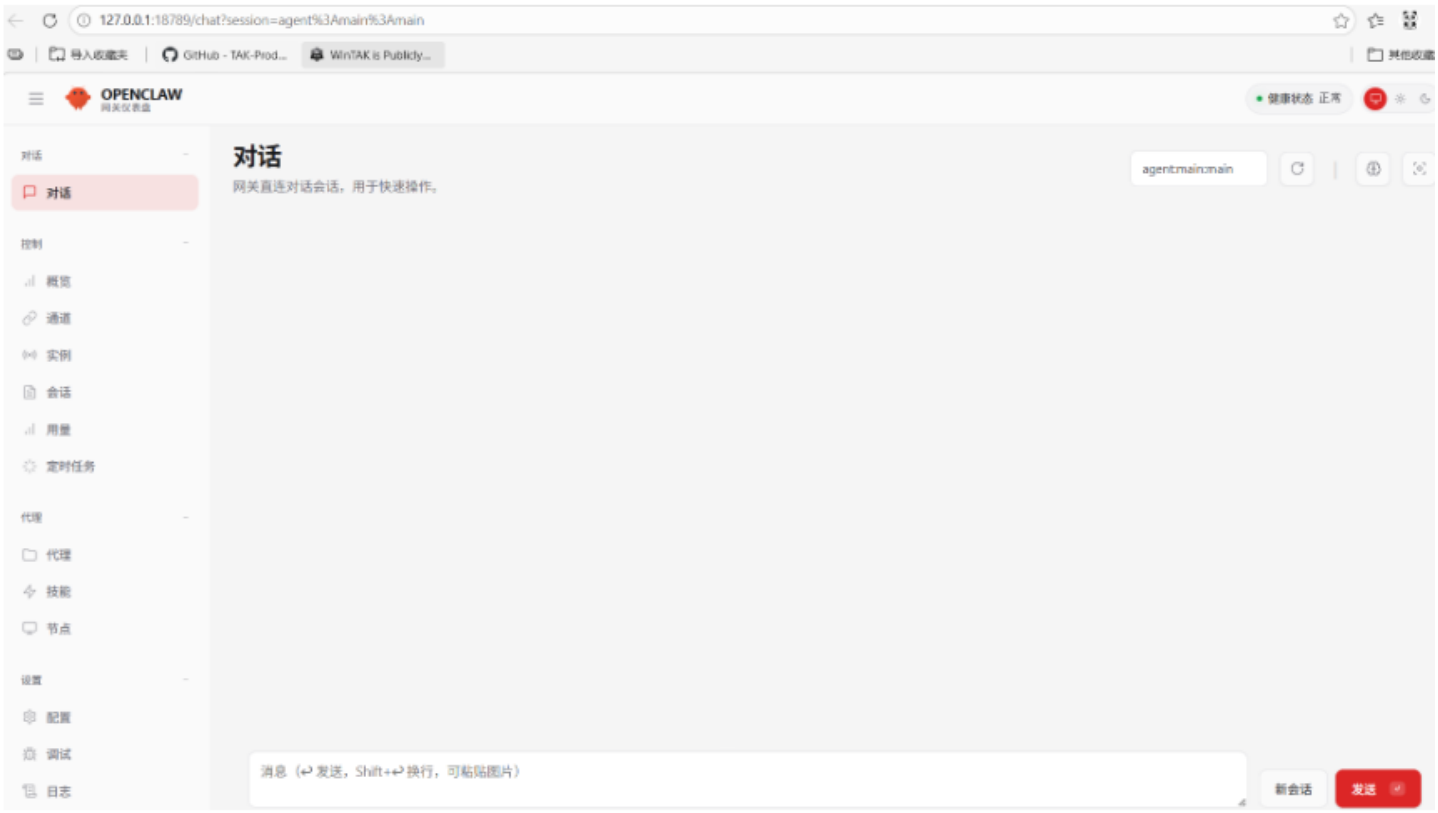
会新弹出一个窗口



此时已经安装成功了。新窗口我们先不管，回到之前的cmd窗口，有如下三个选项



其中“在TUI中孵化（推荐）”代表接下来在命令行与大模型进行交互，“打开网页界面”代表接下来在浏览器进行交互。我们选择“打开网页界面”，敲回车，会自动打开浏览器。如下

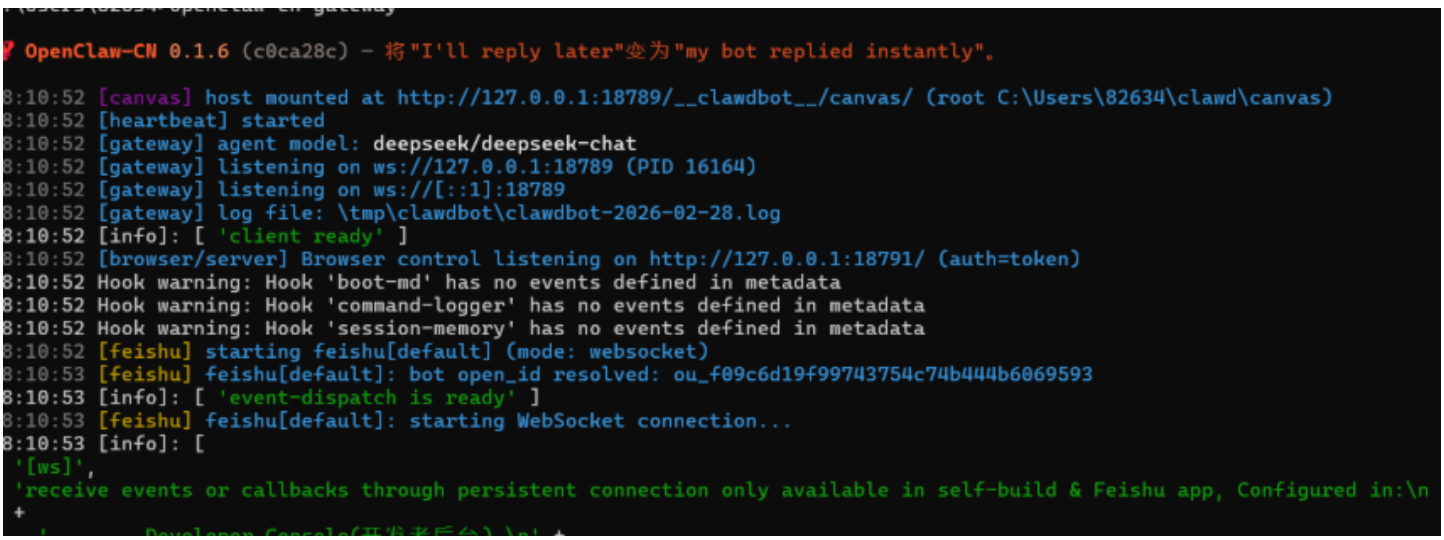


备注：此时本地部署正式完成。想要关闭服务的话，直接关闭cmd窗口即可。

以后启动openclaw

代码块

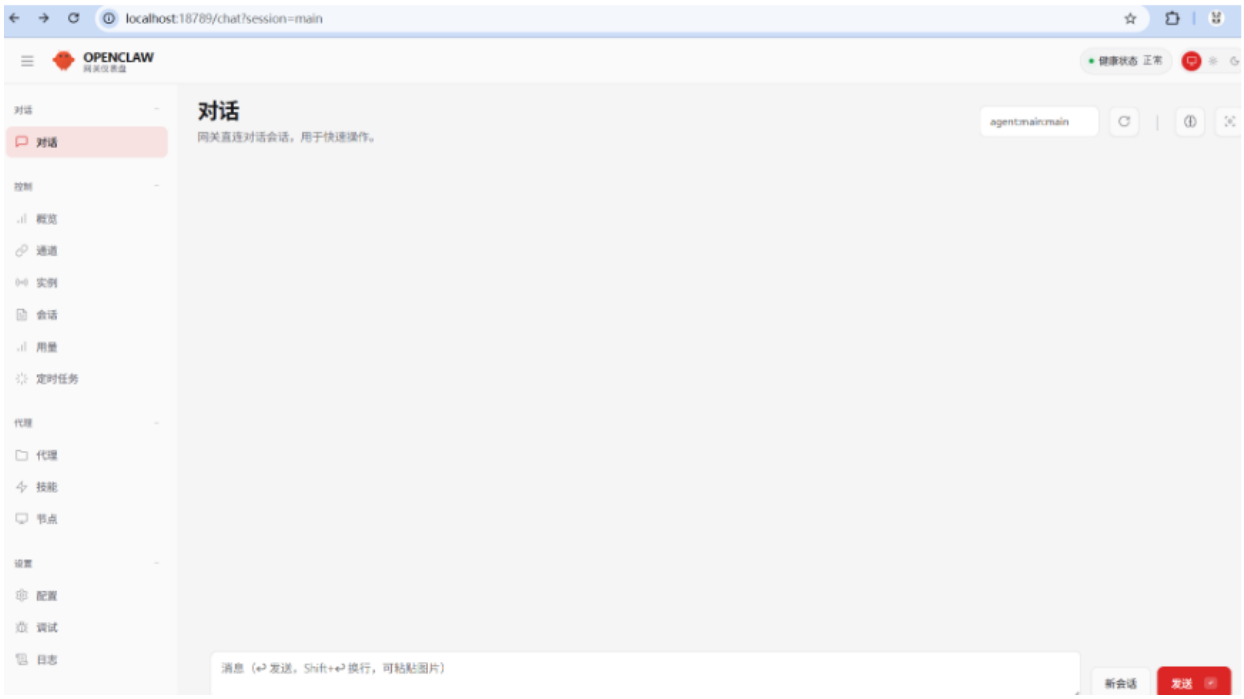
```
openclaw-cn  gateway run
```



浏览器访问：http://localhost:18789/chat?session=main&token=MyGatewayToken

注意：参数里的MyGatewayToken是你前边设置过的，不是随便写的

浏览器访问结果如下：

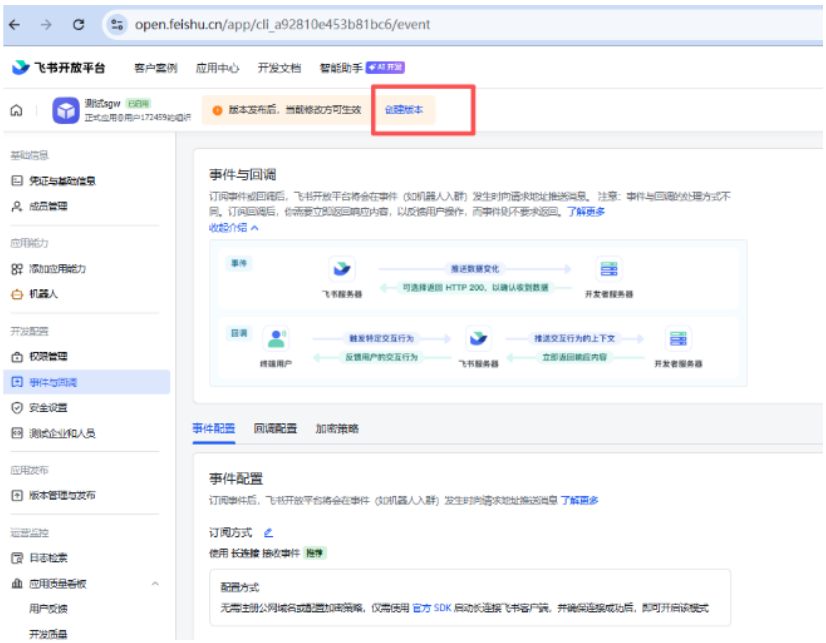


此时，我们只是在本地部署完毕了，但是飞书机器人还不能实时与我们本地openclaw交互，需要在飞书继续做一些配置

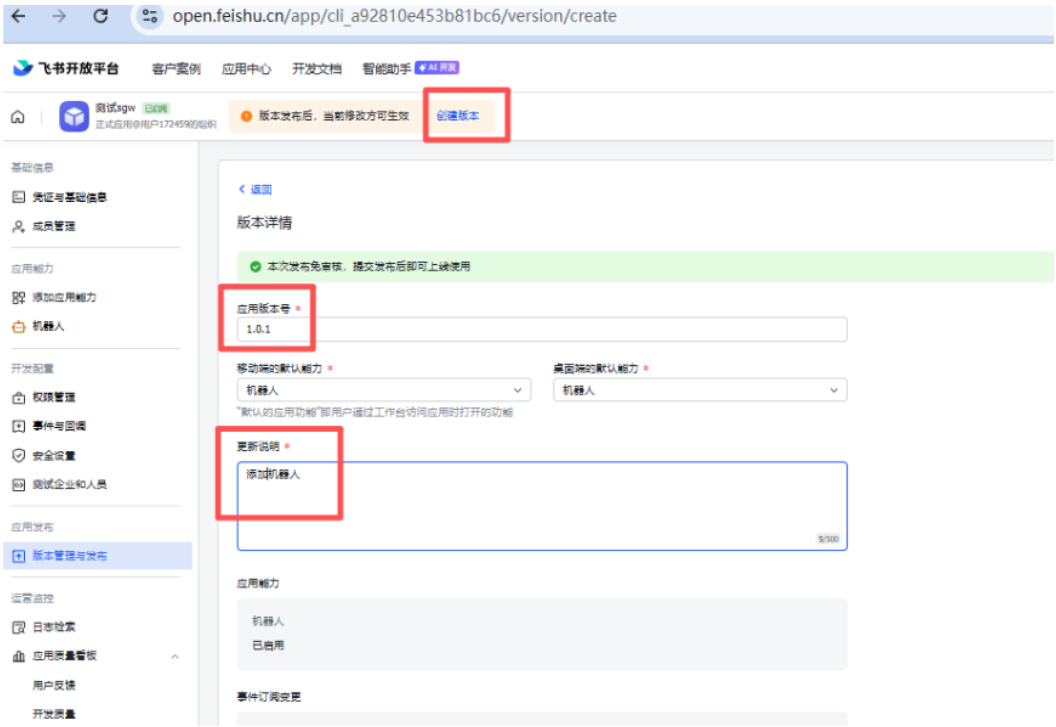
我们继续添加事件，填入官方设置好的事件名

搜索“机器人”，点击添加

保存好之后，我们和之前一样，需要重新发布一个版本



填写内容 (记得保存)



如果以后想让openclaw操作chrome浏览器，那么需要安装浏览器插件

代码块

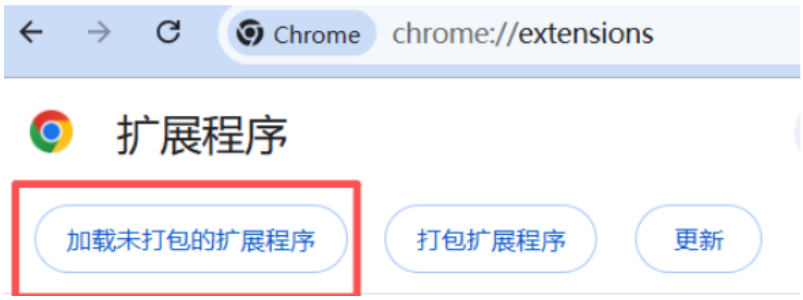
```
# 安装插件
openclaw-cn browser extension install
# 查看插件位置
openclaw-cn browser extension path
```

```
C:\Users\82634>openclaw browser extension path
Config warnings:\n- plugins.entries.feishu: plugin feishu: duplicate plugin id detected; later plugin may be overr
rs\82634\AppData\Roaming\npm\node_modules\openclaw\extensions\feishu\index.ts)

OpenClaw 2026.2.26 (bc50708) — It's not "failing," it's "discovering new ways to configure the same thing wrong

|
o  Config warnings ---------------------------------------------------------------------------+
|                                                                                             |
|  - plugins.entries.feishu: plugin feishu: duplicate plugin id detected; later plugin may    |
|    be overridden                                                                            |
|    (C:\Users\82634\AppData\Roaming\npm\node_modules\openclaw\extensions\feishu\index.ts)    |
|                                                                                             |
+---------------------------------------------------------------------------------------------+

~\.openclaw\browser\chrome-extension
Copied to clipboard.

C:\Users\82634>
```

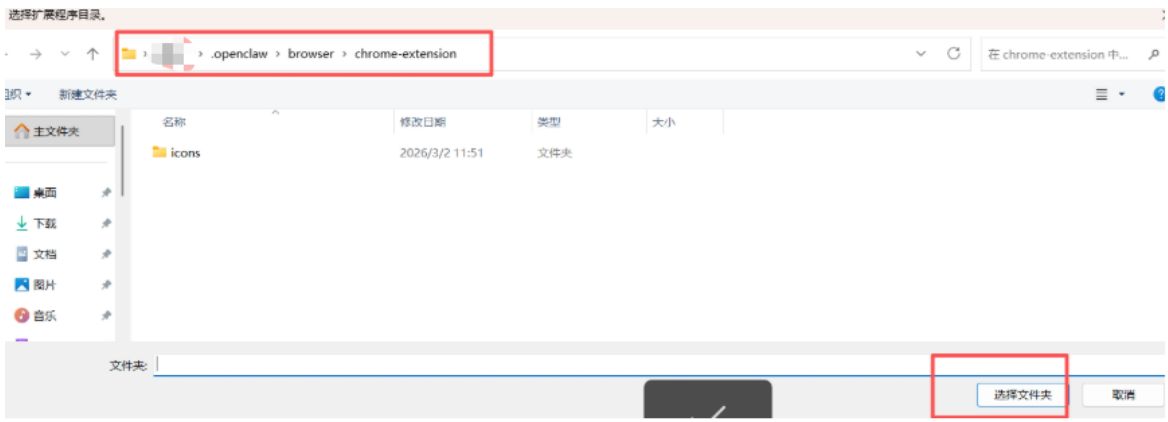
打开Chrome浏览器，在浏览器里输入：

代码块

```
1   chrome://extensions
```

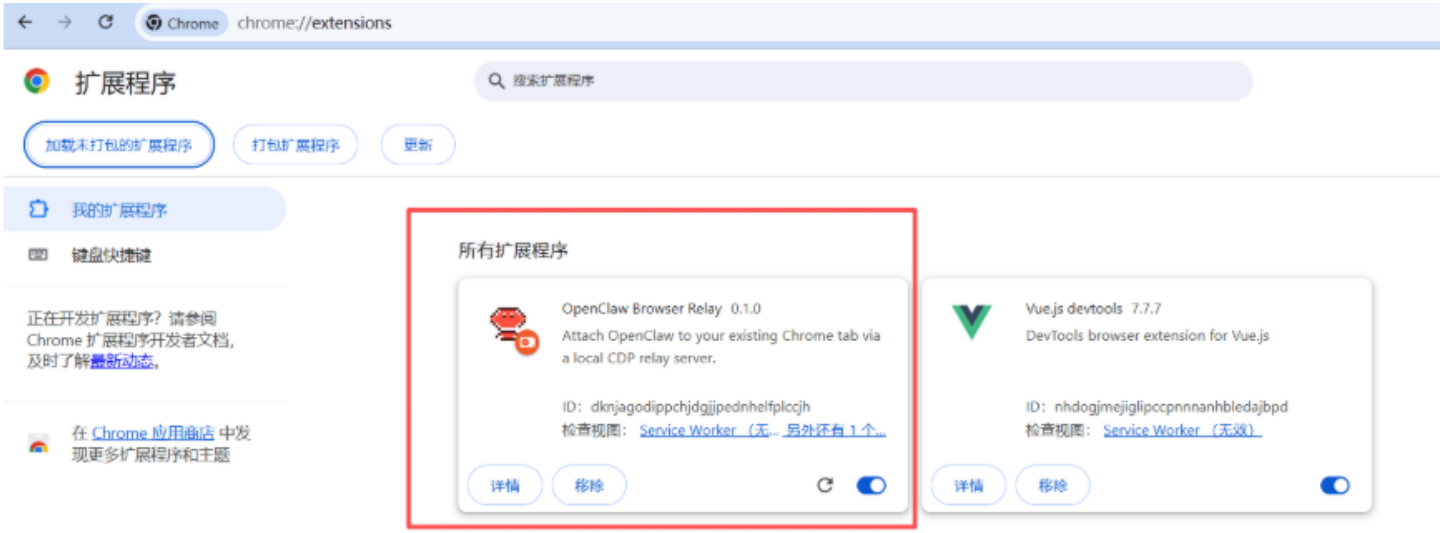


选择插件文件夹



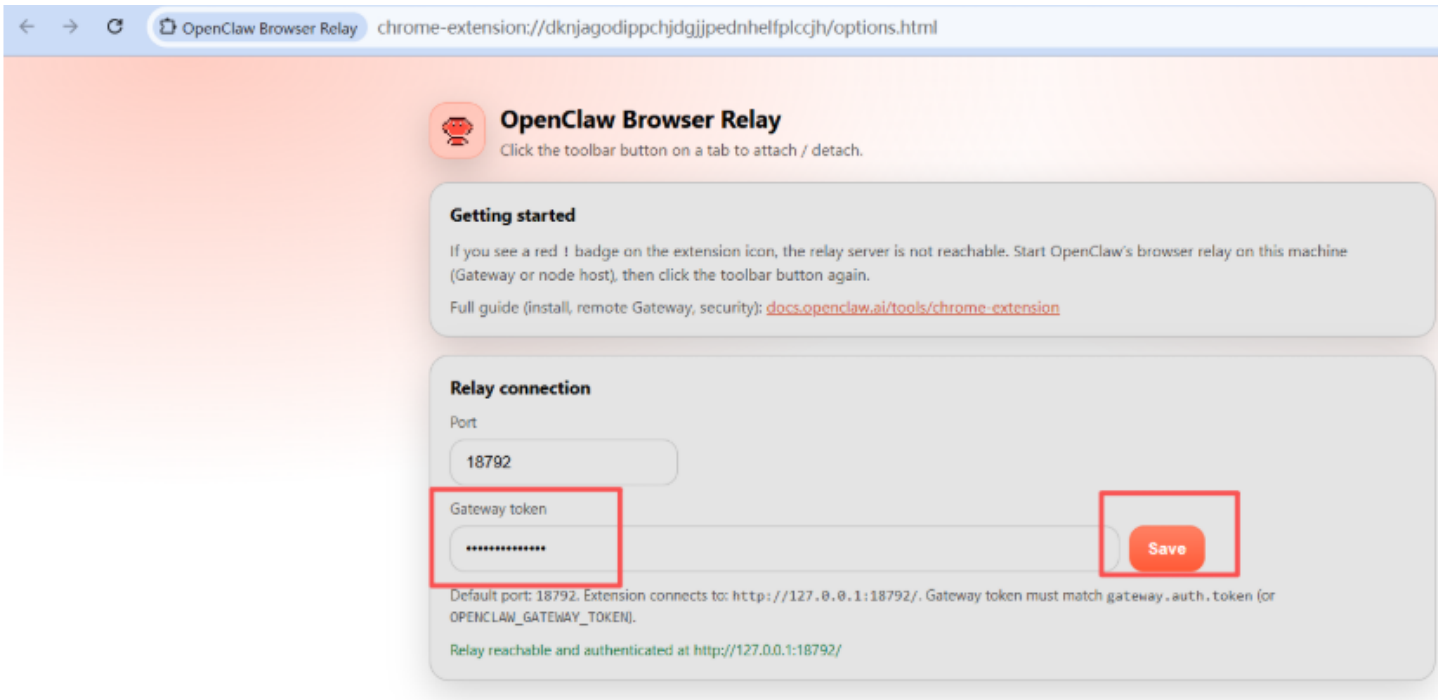
查看结果

查看结果

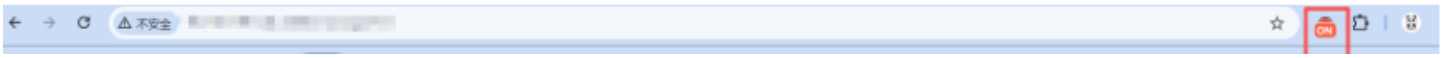


点击刷新按钮

写上之前建好的token，我的是MyGatewayToken



注意，上边这个页面不能关闭，否则openclaw无法控制浏览器。在你想让openclaw控制的页面，打开这个插件（处于on状态），如下



此时，openclaw就能控制你的浏览器当前这个页面了。

停止openclaw的开机自启

以管理员打开cmd，执行如下命令

```
代码块
1   schtasks /Delete /F /TN "OpenClaw Gateway"
```

2.3 常见问题

- Node.js 版本低：升级至 22.0.0 以上。
- 权限不足：加 sudo 或用管理员身份运行终端。
- 模型连接失败：检查 API Key 正确性。

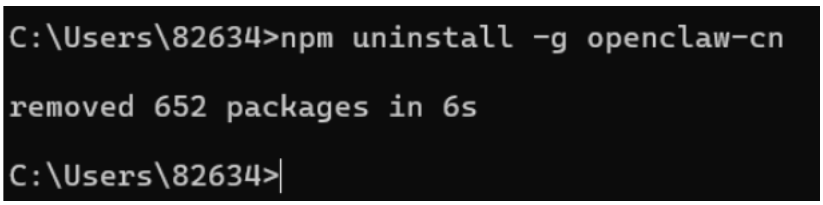
3. 卸载openclaw

如果之前开启了开机启动openclaw，则需要先停服务

```
代码块
1   openclaw-cn gateway stop
2   # 或者尝试
3   openclaw-cn service stop
```

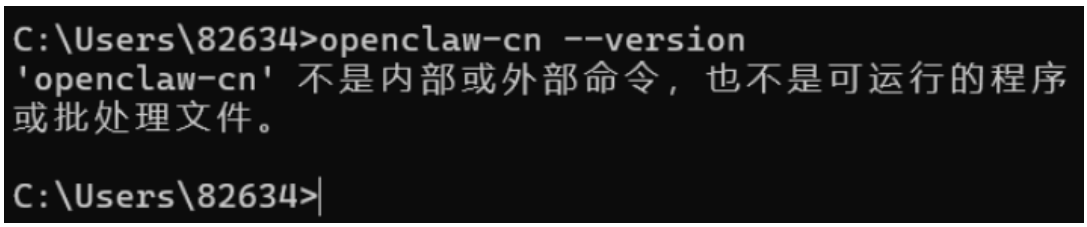
如果没有配置过开机启动的话，直接把cmd网关窗口关闭即可停止服务，然后新开一个cmd窗口执行卸载命令

```
代码块
1   # 卸载openclaw
2   npm uninstall -g openclaw-cn
```

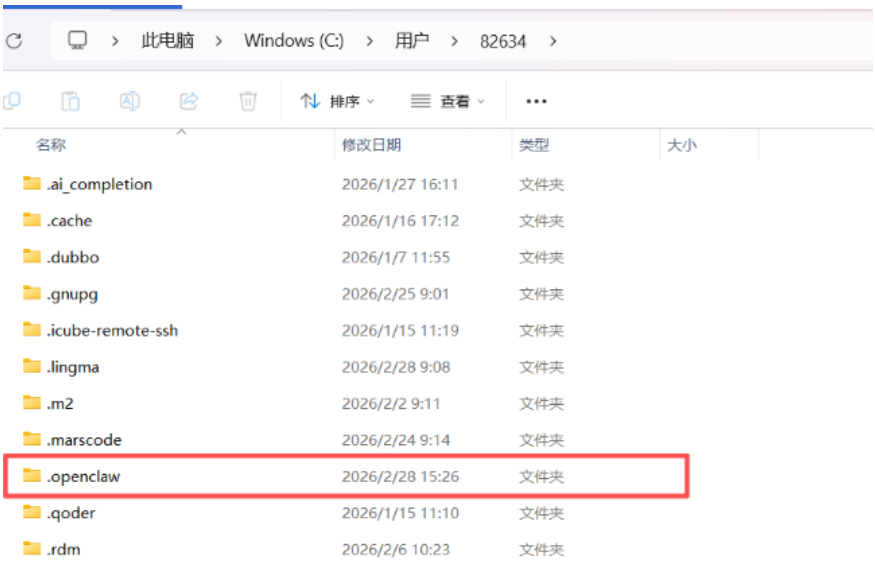


验证卸载是否成功：

```
代码块
1   openclaw-cn --version
```



删除系统磁盘里的openclaw文件夹



这将删除所有的对话历史、已配置的 API Key 和钩子设置。如果你以后还想保留这些数据，请不要删除此文件夹，或者将其备份到别处。