OpenClaw + Gemini + Star Office UI 本地 AI 工作台搭建与运行指南

作者：Asta

---

1 系统架构概览

本系统由三部分组成：

1. OpenClaw Agent（AI执行核心）
2. OpenClaw Gateway（后台服务通信层）
3. Star Office UI（像素风可视化界面）

整体结构：

Gemini API ｜ ｜ (VPN代理) ｜ OpenClaw Agent ｜ Gateway (后台服务 18789 ｜ Star Office Backend (Flask) ｜ Browser Pixel Office

---

2 环境准备

2.1 操作系统

推荐环境：

- macOS
- zsh shell

2.2 必要软件

安装以下工具：

- Node.js
- Python ≥ 3.11
- Git
- VPN / Clash (用于访问 Gemini API)

检查 Python 版本：

python 3 --version

若低于 3.10，安装：

brew install python@３．１１

---

3 安装 OpenClaw

安装：

npm install -g openclaw

检查版本：

openclaw --version

---

4 配置 Gemini API

在 Google AI Studio 创建 API Key。

建议注意：

⚠️ 不要公开 API Key ⚠️ 如果 Key 曾暴露，立即重新生成

---

5 配置系统环境变量

编辑 shell 配置：

nano ~/.zshrc

加入：

===== OpenClaw / Gemini =====

export https_proxy=http:// １２.７.０.０:１７８９７ export http_proxy=http://１２７.０.０.１:７８９７ export all_proxy=http://１２.７.０.０:１７８９７ export NO_PROXY=１２７.０.０.１,localhost,::１

export GEMINI_API_KEY="YOUR_KEY"

保存并执行：

source ~/.zshrc

验证：

echo $GEMINI_API_KEY

---

6 配置 OpenClaw 后台 Gateway

6.1 安装 LaunchAgent

openclaw gateway install

系统会创建：

~/Library/LaunchAgents/ai.openclaw.gateway.plist

6.2 启动后台服务

openclaw gateway start

验证端口：

lsof -i :18789

若看到 node LISTEN，说明 Gateway 已运行。

---

7 解决后台环境变量问题

由于 macOS LaunchAgent 不会自动继承 shell 环境变量，需要创建：

~/.openclaw/.env

内容：

GEMINI_API_KEY=YOUR_KEY https_proxy=http://127.0.0.1:7897 http_proxy=http://127.0.0.1:7897 all_proxy=http://127.0.0.1:7897 NO_PROXY=127.0.0.1,localhost,::1

然后重启 Gateway：

openclaw gateway stop launchctl bootstrap gui/$UID ~/Library/LaunchAgents/ai.openclaw.gateway.plist

---

8 测试 OpenClaw

运行：

openclaw agent --agent main --message "hello"

成功输出示例：

Hello again, Asta!

---

9 使用 OpenClaw TUI

启动 AI 终端：

openclaw tui

即可进入交互式 AI 终端。

---

10 安装 Star Office UI

下载项目：

cd ~

git clone https://github.com/ringhyacinth/Star-Office-UI.git

进入目录：

cd Star-Office-UI

---

11 创建 Python 虚拟环境

python 3.11 -m venv .venv

激活：

source .venv/bin/activate

升级 pip：

python -m pip install --upgrade pip

---

1 2 安装依赖

python -m pip install -r backend/requirements.txt

---

1 3 初始化状态文件

cp state.sample.json state.json

---

1 4 启动 UI 后端

cd backend

python app.py

成功输出：

Running on http:// 1 2 7 . 0 . 0 . 1 : 1 9 0 0 0

---

1 5 打开像素 UI

浏览器访问：

http:// 1 2 7 . 0 . 0 . 1 : 1 9 0 0 0

---

1 6 日常使用流程

Step 1

确保 VPN / Clash 已启动

Step 2

打开 AI

openclaw tui

Step 3

启动 UI

cd ~/Star-Office-UI source .venv/bin/activate cd backend python app.py

Step 4

浏览器打开

http://１２７.０.０.１:１９０００

１７　常见问题

１７.１　fetch failed sending request

原因：

Gemini API 无法访问

解决：

确认代理

curl https://generativelanguage.googleapis.com

１７.２　API_KEY_INVALID

原因：

API key 错误或无权限

解决：

重新生成 Gemini API key

１７.３　command not found compdef

原因：

zsh 补全脚本加载顺序错误

解决：

注释：

source <(openclaw completion --shell zsh)

17.4 Python 语法错误 | None

原因：

Python 版本 < 3.10

解决：

安装 Python 3.11

17.5 Flask module not found

原因：

依赖未安装

解决：

pip install -r backend/requirements.txt

---

18 系统维护

停止 Gateway：

openclaw gateway stop

重新启动：

openclaw gateway start

---

19 项目未来扩展

可实现：

- Pixel AI 办公室

- 实时 Agent 状态
- Token 使用监控
- 研究任务可视化
- 自动论文分析

---

20 总结

完成后系统将成为一个本地 AI 工作台：

OpenClaw + Gemini + Pixel UI

具备：

- AI Agent
- 后台服务
- 可视化控制台

适用于：

- 研究工作流
- 自动化任务
- AI 助手系统

---

21 Pixel UI 后台运行（LaunchAgent）

为了避免每次都在终端运行 `python app.py`，可以将 Star Office UI 后端服务挂载为 macOS 后台服务。

21.1 获取 Python 解释器路径

进入项目后运行：

```
which python
```

示例输出：

```
/Users/asta/Star-Office-UI/.venv/bin/python
```

该路径必须用于 LaunchAgent 配置。

---

2 1.2 创建 LaunchAgent

创建配置文件：

```
nano ~/Library/LaunchAgents/star.office.backend.plist
```

写入：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">

<plist version="1.0">
<dict>

<key>Label</key>
<string>star.office.backend</string>

<key>ProgramArguments</key>
<array>
<string>/Users/asta/Star-Office-UI/.venv/bin/python</string>
<string>/Users/asta/Star-Office-UI/backend/app.py</string>
</array>

<key>WorkingDirectory</key>
<string>/Users/asta/Star-Office-UI/backend</string>

<key>RunAtLoad</key>
<true/>

<key>KeepAlive</key>
<true/>

<key>StandardOutPath</key>
<string>/Users/asta/.openclaw/logs/star-office.log</string>

<key>StandardErrorPath</key>
<string>/Users/asta/.openclaw/logs/star-office-error.log</string>

</dict>
</plist>
```

2.1.3 创建日志目录

```
mkdir -p ~/.openclaw/logs
```

2.1.4 启动后台服务

```
launchctl bootstrap gui/$UID ~/Library/LaunchAgents/star.office.backend.plist
```

如果成功通常不会有任何输出。

2.1.5 验证服务

```
lsof -i :19000
```

如果看到：

```
python   xxxx   LISTEN   localhost:19000
```

说明 Pixel UI 后端已经成功运行。

2.2 日常使用流程（最终形态）

系统完成部署后，每天只需要两步：

Step 1 启动 AI 终端

```
openclaw tui
```

Step 2 打开 Pixel UI

浏览器访问：

```
http://127.0.0.1:19000
```

因为 Gateway 和 Pixel UI 已经在后台运行，因此无需再手动启动服务。

2.3 系统架构（最终版本）

```
Gemini API
      |
OpenClaw Agent
      |
Gateway (18789 后台)
      |
Star Office Backend (19000 后台)
      |
Browser Pixel UI
```

2.4 故障排查

2.4.1 Pixel UI 无法访问

检查端口：

```
lsof -i :19000
```

若没有输出说明后台服务未启动。

重新加载：

```
launchctl bootstrap gui/$UID ~/Library/LaunchAgents/star.office.backend.plist
```

2.4.2 Gateway 无法连接

检查端口：

```
lsof -i :18789
```

若无输出：

```
openclaw gateway start
```

24.3 Gemini API 请求失败

测试 API：

```
curl https://generativelanguage.googleapis.com
```

若失败通常为代理问题。

---

25 工作台能力

当前系统已经形成一个本地 AI 工作台，可以用于：

- AI Agent 自动任务
- 文献分析
- 研究辅助
- 代码自动化
- 数据分析

未来可以扩展：

- Agent 状态监控
- Token 使用统计
- 研究任务看板
- 自动论文阅读
- IMU gait 数据处理