

69gg fix: address PR #312 review issues	5b1ca05 · last week
.github/workflows	fix(ci): verify mac dmg payload before upload 2 months ago
agentserver	feat: improve travel workflows and hub inst... 2 months ago
apiserver	fix: address PR #312 review issues last week
characters	feat: refine prompt pipeline and agent runti... 2 months ago
docs	feat: improve travel workflows and hub inst... 2 months ago
frontend	fix: address PR #312 review issues last week
guide_engine	feat: TTS 全链路修复 + 端口清理增强 + 文... 3 months ago
hooks	OpenClaw回调改进与工具调用修复 (#187) 3 months ago
mcpserver	fix: openclaw 2 months ago
scripts	feat: expand agent orchestration and packa... 2 months ago
skills	feat: expand agent orchestration and packa... 2 months ago
summer_memory	Enhance quintuple extractor for Anthropic A... 3 weeks ago
system	Refactor config paths and server port retrieval 3 weeks ago
vendor/openclaw	Update package-lock.json to manage depe... 2 weeks ago
voice	Update ASR health check URL with config ... 3 weeks ago
.codex	fix: address PR #312 review issues last week
.gitattributes	Update project code and documentation, o... 7 months ago
.gitignore	fix(gitignore): scope broad root-only patterns 2 months ago
LICENSE	Add LICENSE file 3 months ago
LICENSE-CLOSED-SOURCE	chore: CLOSED-SOURCE.LICENSE -> LI... 3 months ago
README.md	Revise README with updates and installat... 3 weeks ago
README_en.md	Revise README with updates and installat... 3 weeks ago
README_ja.md	Revise README with updates and installat... 3 weeks ago
SOUL.md	feat: Heartbeat v3 事件驱动心跳 + 消息队... 3 months ago
build.md	嵌入式 OpenClaw 打包 (#204) 3 months ago
build.py	fix(build): preserve windows node runtime ... 2 months ago
clear.py	添加占位文件逻辑 5 months ago
config.json.example	Update API configuration and add new par... 3 weeks ago
kill_auth.py	fix: address PR #312 review issues last week
	Refactor config handling and sync source c... 3 weeks ago

main.py		
naga-backend.spec	fix(packaging): slim runtime and re-sign ma...	2 months ago
naga_checkin.py	fix: address PR #312 review issues	last week
openclaw.LICENSE	嵌入式 OpenClaw 打包 (#204)	3 months ago
package-lock.json	开发分支合并：新版前端与功能大更新 (#180)	3 months ago
proactive_vision_config.json	refactor: 恢复论坛/旅行界面 + 优化启动流...	3 months ago
pyproject.toml	Add anthropic dependency version 0.42.0	3 weeks ago
requirements.txt	feat: 积分好感度系统 + 悬浮球阴影/拖拽修...	3 months ago
setup_venv.bat	chore: 结婚横MCP/NagaPortal键稿另添 g ...	3 months ago
start.bat	未完成的打包实现与一键启动脚本 (#200)	3 months ago
uv.lock	Update uv.lock for revision and package ch...	3 weeks ago

README AGPL-3.0 license License

NagaAgent

你的二次元超能AI助手

流式工具调用 · 知识图谱记忆 · Live2D 虚拟形象 · 语音交互 · 娜迦网络社区

《娜迦协议》包括对话、记忆、MCP、skills、openclaw集成，以打造丰富且易用的ai工具为核心，成为用户端侧的二次元超能ai助手。

软件包含如下功能：①一键登陆自动配置所有apikey，与具备Live2d形象的娜迦进行鼠标互动和亲切对话；②通过内置openclaw快速探索自己需要的知识领域，或是列出任务方向让其自动探索执行；③根据对话历史自动构建三维记忆云海，并将记忆注入后续对话；④通过对话之余留下的线索碎片，探索娜迦网络中的神秘娜迦世界；⑤内置游戏攻略功能，可通过交互或自动识别游戏画面，了解到游戏正在进行的细节信息，并给出建议。可通过MAA等游戏操作插件来自动打游戏；⑥还具有自我配置、浏览器操纵、音乐盒等丰富组件。

娜迦的一切未来，等你探索。

[简体中文](#) | [English](#) | [日本語](#)

NAGAAGENT	5.1.0	PLATFORM	WINDOWS MACOS LINUX	LICENSE	AGPL 3.0 PROPRIETARY	PYTHON	3.11
-----------	-------	----------	-------------------------	---------	------------------------	--------	------

Stars 1.5k
 Forks 161
 Issues 4 open
 Nerq Trust Score

[QQ 机器人联动 : Undefined QQbot](#)

双许可证 · 开源采用 [AGPL-3.0](#)，闭源采用 [专属许可](#)（需书面授权）。商业合作：contact@nagaagent.com / bilibili【柏斯阔落】

更新日志

日期	版本	内容
2026-04-15	—	配置同步重构：source config 与 runtime config 双向合并；ASR 健康检查 URL 改用配置值；服务端口获取与配置路径重构
2026-04-14	—	新增 Anthropic API 格式支持（ <code>api_format</code> 字段）；五元组提取器兼容 Anthropic SDK；Live2D 空文本调用处理
2026-04-12	—	修复 py2neo <code>Graph()</code> timeout 参数不兼容与 Neo4j 连接状态误报

日期	版本	内容
 2026-04-05	—	论坛功能增强：Feed 模式与解锁进度特性
 2026-03-21	—	新增日文 README (README_ja.md)
 2026-03-20	—	改进旅行模块工作流与 Hub 安装流程
 2026-03-17	—	QQ 通知：专用验证端点、服务端 mention 处理、错误透传
 2026-03-16	—	论坛加载与打包视图恢复修复；macOS DMG 签名与文件系统修复；OpenClaw 打包从源码编译与运行时修复
 2026-03-15	—	OpenClaw 打包版运行时加固；遥测通知与运行时集成优化；Windows 构建脚本控制台编码修复
 2026-03-14	—	干员通讯录升级为设置弹窗（名称 / 人设 / 引擎 / 灵魂文档 / 专有 MCP-Skill）；技能工坊接入 NagaHub 与通用 MCP 预热；探索链路新增 QQ / 飞书完成通知；云端记忆优先时不再回退本地 Neo4j
 2026-03-13	—	OpenClaw 编排链路 with 打包集成继续扩展；后端 spec 修复 Windows 控制台 Unicode 输出报错
 2026-03-11	—	OpenClaw Skill 自动执行；干员独立 workspace；模型选择器与定价显示接入 Default / Deepseek-V3.2 / Kimi-K2.5
 2026-03-09	—	深度集成 OpenClaw vendor 源码编译与统一配置；论坛未登录 401 风暴修复；Windows 托盘图标修复
 2026-03-08	—	naga-backend.spec 持续修正；应用扫描器支持环境变量与 macOS；工具结果默认折叠；枢机集市移除记忆云迁 / MCP 工具 / 智能体技能三板块
 2026-03-07	—	Node.js 与 uv 运行时随包分发；MCP 命令解析器统一；OpenClaw Gateway 启动诊断增强；打包版音乐 / 唤醒语音 / 记忆云海等问题修复
 2026-03-06	—	GitHub Actions 构建发布流水线；Electron 自动更新替换旧补丁系统；CI 构建资源与 charset_normalizer 打包修复；论坛与积分轮询细节优化
 2026-03-05	—	TTS 开关、消息队列与串行发送重构；RAG 记忆召回增强；Gemini / 自动 Function Calling 支持；充值界面与远程记忆稳定性修复
 2026-03-04	—	迁移到原生 Function Calling；DogTag 接管 heartbeat / proactive vision；前后端热补丁系统与 4 层安全防护；流式文本与渐进式 TTS 优化
 2026-03-03	—	web_search 直接接入 NagaBusiness 搜索代理；论坛连接、TTS 流式播放与 OpenClaw 轮询降级修复；Live2D 与 Electron 稳定性修复
 2026-03-02	—	Heartbeat v3 事件驱动重构；naga_control 自编排工具；打包时嵌入 agent-browser；启动诊断、健康检查与 OpenClaw 配置路径修复
 2026-03-01	—	已登录走 NagaBusiness、未登录走 Brave / OpenClaw 的搜索代理成型；OpenClaw 工具直调与自动启动完善；游戏攻略与角色音色配置更新
 2026-02-28	—	持久化存储统一迁移到 ~/.naga 用户目录；ForumQuotaView 网络探索控制中心；旅行模块与语音交互全面升级
 2026-02-27	—	接入 ASR 语音识别 (MediaRecorder + NagaBusiness 代理)；会话风格、Electron 背景、MCP vision 与服务器配置整理
 2026-02-26	5.1.0	娜迦网络社区论坛上线；设置三合一重构；旅行模式；积分配额页；枢机集市与主面板更新
 2026-02-25	5.1.0	TTS 全链路修复 (CORS / asyncio)；build.py 跨平台构建；上下文压缩持久化；角色系统更新；提示词注入架构重构
 2026-02-24	—	Neo4j 连接超时修复；统一 BGM 播放器；音律坊歌单编辑；MCP 管理 UI；悬浮球透明窗口 + 悬停亮度

日期	版本	内容
🔧 2026-02-23	—	跨平台构建完善；版本号统一 pyproject.toml 管理；提示词/截图/视觉优化；角色文件打包迁移
💖 2026-02-22	—	积分好感度系统（签到 / 好感度 / 积分）；悬浮球阴影与拖拽修复；登录自动恢复；OpenClaw hooks 修复
🎵 2026-02-21	—	音律坊图标更新；MCP Agents 更新；悬浮球小按钮
🗨️ 2026-02-20	—	上下文压缩三级重构（<compress> 标签 / 跨会话继承）；MCP 管理 UI；悬浮球透明窗口；音律坊功能修正
🔄 2026-02-19	—	SSE 去除 base64 直接 JSON 传输；移除冗余后台意图分析器；config_manager 自动检测编码
🔧 2026-02-17	—	悬浮球序列帧路径改为相对路径，修复打包后头像不显示
🚀 2026-02-16	5.0.0	NagaModel 网关统一接入；DeepSeek 推理链实时展示；记忆云海 UI 自适应修复
🧠 2026-02-15	—	统一附加知识块 + 消除历史污染；LLM 流式重试；七天自动登录；开机自启动
🌊 2026-02-14	—	NagaMemory 云端远程记忆；意识海 3D 重写；启动粒子动画；版本更新检查弹窗；用户使用协议
🌟 2026-02-13	—	悬浮球 4 状态模式；截屏多模态视觉切换；技能工坊重构；Live2D 表情通道独立
🍷 2026-02-12	—	NagaCAS 认证；Live2D 4 通道正交动画架构；Agentic Tool Loop；明日方舟风格启动界面
📦 2026-02-11	—	嵌入式 OpenClaw 打包；启动自动从模板生成配置文件
🔧 2026-02-10	—	后端打包优化；技能工坊 MCP 状态修复；去除冗余 Agent/MCP 仅保留 OpenClaw
🌱 2026-02-09	—	前端重构；Live2D 禁用眼睛追踪；OpenClaw 更名为 AgentServer

目录

1. [快速开始](#)
2. [功能导览 \(主面板\)](#)
3. [对话](#)
4. [记忆云海](#)
5. [技能工坊](#)
6. [娜迦网络](#)
7. [枢机集市](#)
8. [终端设置](#)
9. [音律坊](#)
10. [悬浮球](#)
11. [全局功能](#)
12. [后端架构](#)
13. [可选配置](#)
14. [端口一览](#)
15. [故障排除](#)

快速开始

环境要求

- Python 3.11 (`>=3.11, <3.12`)
- 可选 : [uv](#) — 加速依赖安装
- 可选 : Neo4j — 本地知识图谱记忆

安装

```
git clone https://github.com/Xxiii8322766509/NagaAgent.git
cd NagaAgent

#前端安装
cd frontend
npm install
cd..

#后端安装
# 方式一 : uv (推荐)
uv sync

# 方式二 : 手动
python -m venv .venv
source .venv/bin/activate # Windows: .\.venv\Scripts\activate
pip install -r requirements.txt
```

最小配置

复制 `config.json.example` 为 `config.json` , 填入 LLM API 信息 :

```
{
  "api": {
    "api_key": "your-api-key",
    "base_url": "https://api.deepseek.com",
    "model": "deepseek-v3.2",
    "api_format": "openai"
  }
}
```

支持所有 OpenAI 兼容 API (DeepSeek、通义千问、OpenAI、Ollama 等) , 也支持 Anthropic 原生格式 (将 `api_format` 设为 `"anthropic"`) 。

启动

```
cd frontend && npm run dev (配置了一键启动)
```

功能导览 (主面板)

启动后进入主面板 (PanelView) , 采用 3D 视差效果 (鼠标移动触发透视旋转) 。 所有功能从主面板的八个入口按钮展开 :

#	入口	路由	功能概要
1	对话	<code>/chat</code>	AI 对话、流式工具调用、上下文压缩
2	记忆云海	<code>/mind</code>	知识图谱 3D 可视化与 GRAG 记忆管理
3	技能工坊	<code>/skill</code>	MCP 工具管理与社区 Skill 安装
4	娜迦网络	<code>/forum / /forum/quota</code>	社区论坛、积分好感度

#	入口	路由	功能概要
5	枢机集市	/market	背景、音乐、角色、记忆迁移、充值
6	终端设置	/config	模型连接、记忆连接、音画配置 (三合一)
7	音律坊	/music	BGM 播放器与歌单管理
8	悬浮球	—	进入轻量悬浮球窗口模式

1. 对话 · MessageView

流式工具调用

对话引擎通过 SSE 流式输出，同时实时送达前端显示与 TTS 分句播放。工具调用不依赖 OpenAI Function Calling API，LLM 在文本中以 ````tool```` 代码块嵌入 JSON，任何 OpenAI 兼容提供商均可使用。同时支持通过 `api_format: "anthropic"` 使用 Anthropic 原生 API 格式。

单轮工具调用流程：



- 文本解析：json5 容错解析，全角字符自动标准化
- SSE 格式：data: {"type":"content"|"reasoning","text":"..."}\n\n (直接 JSON，不含 base64)
- 循环上限：max_loop_stream = 5 (可配置)

源码：[apiserver/agent_tool_loop.py](#)

上下文压缩

会话 token 超过 100k 时自动触发压缩，避免上下文溢出：

阶段	触发时机	行为
启动压缩	会话加载时	历史超阈值则立即压缩前段消息
运行时压缩	每轮对话后	超限则压缩并注入 <compress> 标签
跨会话继承	新会话启动	读取上次摘要，滚动累积上下文

摘要结构 (6 分区)：关键事实 / 用户偏好 / 重要决定 / 待办事项 / 背景信息 / 最近状态。 <compress> 标签持久化到会话文件，不计入 LLM token 统计。

DeepSeek 推理链展示

使用 DeepSeek 时，reasoning 字段通过 SSE 实时推送，前端以独立样式展示思考过程。

2. 记忆云海 · MindView

GRAG 知识图谱记忆

GRAG (Graph-RAG) 从对话中自动提取五元组并存入 Neo4j, 对话时自动检索作为 LLM 上下文。

五元组结构: (主体, 主体类型, 谓词, 客体, 客体类型)

提取流程:

- 结构化提取 (优先): `beta.chat.completions.parse()` + Pydantic `QuintupleResponse`, `temperature=0.3`, 重试 3 次
- JSON 兜底: 解析失败时提取首个 [到末尾] 的内容
- 过滤规则: 只保留事实 (行为、关系、状态、偏好), 过滤隐喻、假设、纯情感

实体类型: `person` / `location` / `organization` / `item` / `concept` / `time` / `event` / `activity`

任务管理器:

- 3 个 `asyncio worker` 消费 `asyncio.Queue(maxsize=100)`
- SHA-256 去重: 相同文本的重复任务自动跳过
- 每小时清理超过 24h 的已完成任务

双重存储:

- 本地: `logs/knowledge_graph/quintuples.json`
- 云端: Neo4j 图数据库, `graph.merge()` `upsert`

RAG 检索: 关键词提取 → Cypher 查询 → 格式化为 `主体(类型) -[谓词]- 客体(类型)` 注入上下文

远程记忆: 登录用户优先使用 NagaMemory 云端; 未登录时可使用本地 GRAG。为避免性能损耗, 云端链路不再自动回退本地 Neo4j。

源码: [summer_memory/](#)

意识海 3D 可视化

Canvas 2D + 手写 3D 投影 (非 WebGL), 球面坐标相机, 透视除法 `700 / depth`。

7 层渲染顺序: 背景渐变 → 地面网格 → 水面平面 → 体积光 (3 束光柱) → 粒子系统 (3 层 125 颗) → 生物荧光浮游生物 (10 个带拖尾) → 知识图谱节点与边 (深度排序)

图谱映射: `subject/object` → 节点, `predicate` → 有向边, 度中心性 → 节点高度权重, 上限 100 节点

交互: 拖拽旋转、中键平移、滚轮缩放、节点点选/拖拽、关键词搜索过滤

3. 技能工坊 · SkillView

内置 MCP Agent

基于 [Model Context Protocol](#) 的可插拔工具架构, 每个工具以独立 Agent 运行:

Agent	功能
<code>weather_time</code>	天气查询 / 预报、系统时间、自动城市 / IP 检测
<code>open_launcher</code>	扫描系统已安装应用, 自然语言启动程序
<code>game_guide</code>	游戏策略问答、伤害计算、配队推荐、自动截图注入
<code>online_search</code>	基于 SearXNG 的网络搜索
<code>crawl4ai</code>	基于 Crawl4AI 的网页内容提取
<code>playwright_master</code>	基于 Playwright 的浏览器自动化
<code>vision</code>	截图分析与视觉问答

Agent	功能
<code>mqtt_tool</code>	MQTT 协议 IoT 设备控制
<code>office_doc</code>	docx / xlsx 内容提取

注册与发现：`mcp_registry.py` glob 扫描 `**/agent-manifest.json`，`importlib.import_module` 动态实例化。

MCP 管理 UI

前端 `McpAddDialog.vue` 提供图形化 MCP 工具管理界面，支持在线添加 / 删除工具（无需重启）。

社区 Skill 安装

技能工坊支持一键安装社区发布的 Skill（Agent Browser、Brainstorming、Context7、Firecrawl Search 等）。后端接口：`GET /openclaw/market/items`、`POST /openclaw/market/items/{id}/install`

源码：[mcpserver/](#)

4. 娜迦网络 · 论坛社区

社区论坛

从主面板“娜迦网络”区块进入，内嵌完整社区功能：

视图	路由	功能
<code>ForumListView</code>	<code>/forum</code>	帖子列表、分类筛选
<code>ForumPostView</code>	<code>/forum/post/:id</code>	帖子详情浏览（当前版本为只读，不提供前端评论与“想要认识”操作）
<code>ForumMessagesView</code>	<code>/forum/messages</code>	私信消息
<code>ForumMyPostsView</code>	<code>/forum/my-posts</code>	我的发帖
<code>ForumMyRepliesView</code>	<code>/forum/my-replies</code>	我的回复
<code>ForumQuotaView</code>	<code>/forum/quota</code>	积分配额与探索入口

源码：[frontend/src/forum/](#)

积分好感度系统

登录用户专属的游戏化互动体系：

维度	说明
积分 (Credits)	签到 / 连签奖励积累，用于兑换模型额度
好感度 (Affinity)	每次签到增长，反映与娜迦的关系深度
每日签到	用户菜单一键签到，连续签到触发额外奖励

相关 API（通过 API Server 代理至 Naga 门户）：`/api/checkin`、`/api/affinity`、`/api/credits`

5. 枢机集市 · MarketView

枢机集市整合了所有资源获取与管理入口，分为七个 Tab：

Tab	说明
界面背景	切换应用主题背景
音之巷	购买 / 解锁音乐专辑（当前：沙之书）

Tab	说明
角色注册	绑定 / 切换 AI 角色 (需登录)
记忆云迁	云端记忆数据迁移与管理
MCP 工具	MCP 工具图形化管理
智能体技能	社区 Skill 一键安装
模型充值	Naga 门户积分充值

6. 终端设置 · ConfigView

设置页三合一重构，原分散配置集中在单一页面的三个 Tab：

Tab	内容
模型连接	LLM API Key、Base URL、模型选择
记忆连接	Neo4j 连接参数、NagaMemory 云端配置
音画配置	角色档案、Live2D 模型与 SSAA、TTS 声音、聊天字号

角色卡系统

`characters/` 目录管理可切换的 AI 角色，每个角色以 JSON 配置文件描述：

```
{
  "ai_name": "娜杰日达",
  "user_name": "用户",
  "live2d_model": "NagaTest2/NagaTest2.model3.json",
  "prompt_file": "conversation_style_prompt.txt",
  "portrait": "Naga.png",
  "bio": "由开发者柏斯阔落亲手创造的AI助手，简称娜迦。"
}
```

- 每个角色目录包含独立的对话风格提示词、Live2D 模型资源、立绘图像
- 激活角色后，AI 名称与 Live2D 模型由角色 JSON 统一管理，不可在界面手动覆盖
- 默认角色：**娜杰日达**

源码：[characters/](#)

7. 音律坊 · MusicView

独立音乐播放器，与主界面 BGM **共享同一播放实例**（统一 BGM 架构）：

- **歌单编辑** (`MusicEditView`)：管理曲目列表，保存后实时同步至全局播放器
- **播放状态同步**：播放 / 暂停图标与音频事件实时联动
- **列表循环**：当前曲目结束后自动播放下一首
- **Live2D 口型同步**：TTS 播放期间，`AdvancedLipSyncEngineV2` 以 60FPS 驱动 Live2D 嘴形

8. 悬浮球 · FloatingView

点击主面板“悬浮”按钮进入轻量悬浮球窗口模式，四种状态循环切换：

```
ball (100×100 圆球) - compact (420×100 折叠条) - full (420×N 展开) - classic (正常窗口)
```

外观与动效：

- 序列帧眨眼动画：5 帧（睁眼 → 半闭 → 闭眼 → 半闭 → 睁眼），70ms / 帧，随机间隔触发
- 生成回复时：球体光晕脉冲特效
- 悬停时：亮度提升效果
- 透明无边框窗口，支持自由拖拽定位

功能：

- 悬浮状态下可直接输入对话，历史消息在 compact / full 状态下可回溯
- 截图捕获面板：选取屏幕窗口作为图像附件
- 文件上传支持
- 右键菜单通过 Electron 原生菜单实现（避免小窗口裁剪）

全局功能

语音交互

TTS (语音合成)

- 引擎：Edge-TTS，OpenAI 兼容接口 `/v1/audio/speech`
- 架构：3 线程流水线 — 分句队列 → TTS 调用 (Semaphore(2) 并发) → pygame 播放
- Live2D 口型同步：60FPS 提取 5 参数 (mouth_open / mouth_form / mouth_smile / eye_brow_up / eye_wide)
- 端口清理：启动时自动检测并释放占用端口

ASR (语音识别)

- 本地引擎：FunASR，支持 VAD 端点检测与 WebSocket 实时流
- 三模式自动切换：LOCAL (FunASR) → END_TO_END (Qwen Omni) → HYBRID

实时语音对话 (需 DashScope API Key)

- 基于 Qwen Omni 的全双工 WebSocket 语音交互
- 回声抑制、VAD 检测、音频分块 (200ms)、会话冷却控制

```
{
  "voice_realtime": {
    "enabled": true,
    "provider": "qwen",
    "api_key": "your-dashscope-key",
    "model": "qwen3-omni-flash-realtime"
  }
}
```

源码：[voice/](#)

Live2D 虚拟形象

使用 `pixi-live2d-display` + `PixiJS WebGL` 渲染 Cubism Live2D 模型。SSAA 超采样抗锯齿：Canvas 按 `width × ssaa` 渲染，CSS `transform: scale(1/ssaa)` 缩放。

4 通道正交动画系统 (live2dController.ts)：

通道	控制内容	特性
体态 (State)	idle / thinking / talking 循环	hermite 平滑插值，从 <code>naga-actions.json</code> 加载
动作 (Action)	点头 / 摇头等头部动作	FIFO 队列，单一执行
表情 (Emotion)	<code>.exp3.json</code> 表情文件	Add / Multiply / Overwrite 三种混合模式，指数衰减过渡
追踪 (Tracking)	鼠标视线跟随	可配延迟启动， <code>tracking_hold_delay_ms</code>

合并顺序：体态 → 嘴形 → 动作 → 手动覆盖 → 表情混合 → 追踪混合

OpenClaw 电脑控制

对接 OpenClaw Gateway (端口 20789)，通过自然语言调度 AI 编程助手执行本地任务。

- 三级回退启动：打包内嵌 → 全局 `openclaw` 命令 → 自动 `npm install -g openclaw`
- 支持 sessionKey hooks (2026.2.17+)，可配置自定义 hooks 路径
- `POST /openclaw/send` 发送指令，最长等待 120 秒

任务调度器 (TaskScheduler)：

- 任务步骤记录 (目的 / 内容 / 输出 / 分析 / 成功与否)
- 自动提取"关键发现"标记
- 内存压缩：步骤数超阈值时调用 LLM 生成 `CompressedMemory` (`key_findings / failed_attempts / current_status / next_steps`)
- `schedule_parallel_execution()` 通过 `asyncio.gather()` 并行执行任务列表

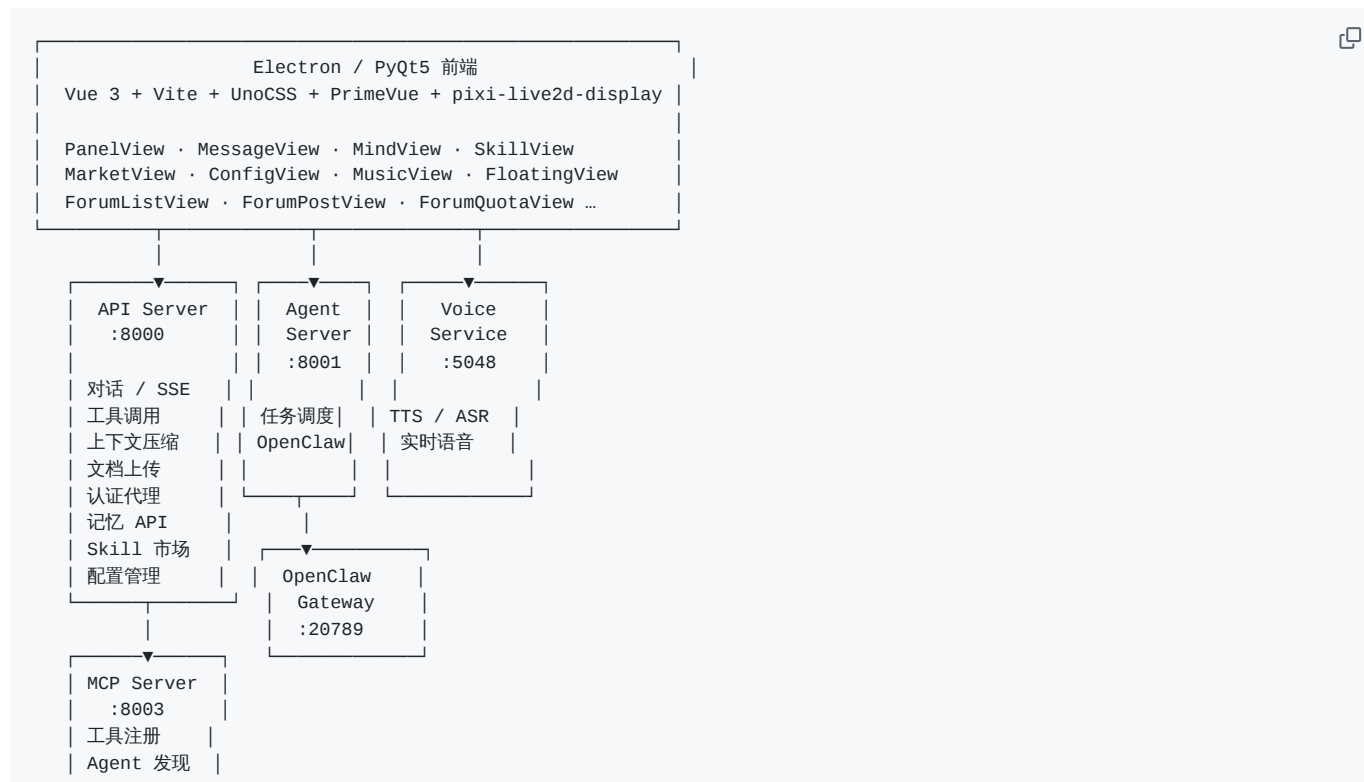
源码：[agentserver/](#)

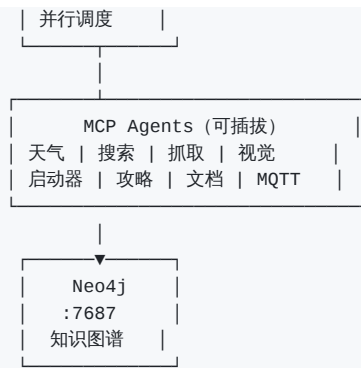
启动动画

阶段	内容
标题阶段	黑色遮罩 + 40 颗金色上升粒子 + 标题图片 2.4s CSS keyframe；标题出现时自动播放唤醒语音
进度阶段	Neural Network 粒子背景 + Live2D 透出框 + 金色进度条 (<code>requestAnimationFrame</code> 插值, 最低速度 0.5 兜底)
停滞检测	3 秒无进度变化显示重启提示；进度 25% 后每秒轮询后端 <code>/health</code>
唤醒	进度 100% 后显示"点击唤醒"脉冲提示

后端架构

NagaAgent 由五个独立微服务组成，`main.py` 统一编排启动：





目录结构

```

NagaAgent/
├── main.py           # 统一入口，编排所有服务
├── build.py         # 跨平台构建脚本
├── config.json      # 运行时配置 (从 config.json.example 复制)
├── pyproject.toml   # 版本 5.1.0，项目元数据与依赖
├── apiserver/      # API Server (:8000)
│   ├── api_server.py   # FastAPI 主应用
│   ├── agentic_tool_loop.py # 多轮工具调用循环
│   ├── llm_service.py  # LiteLLM 统一 LLM 调用
│   └── streaming_tool_extractor.py # 流式分句 + TTS 分发
├── agentserver/    # Agent Server (:8001)
│   ├── agent_server.py
│   └── task_scheduler.py # 任务编排 + 压缩记忆
├── mcpserver/      # MCP Server (:8003)
│   ├── mcp_server.py
│   ├── mcp_registry.py # manifest 扫描 + 动态注册
│   ├── mcp_manager.py  # unified_call() 路由
│   ├── agent_weather_time/
│   ├── agent_open_launcher/
│   ├── agent_game_guide/
│   ├── agent_online_search/
│   ├── agent_crawl4ai/
│   ├── agent_playwright_master/
│   ├── agent_vision/
│   ├── agent_mqtt_tool/
│   └── agent_office_doc/
├── summer_memory/  # GRAG 知识图谱记忆
│   ├── quintuple_extractor.py # 支持 OpenAI / Anthropic 格式
│   ├── quintuple_graph.py
│   ├── quintuple_rag_query.py
│   ├── task_manager.py
│   ├── memory_manager.py
│   └── memory_client.py # NagaMemory 远程客户端
├── voice/          # 语音服务 (:5048 / :5060)
│   ├── output/     # TTS + 口型同步
│   └── input/       # ASR + 实时语音
├── characters/     # 角色配置目录
│   └── 娜杰日达/   # prompt / Live2D 模型 / 立绘
├── frontend/      # Electron + Vue 3 前端
│   ├── electron/  # 主进程
│   │   └── modules/ # backend / hotkeys / menu / tray / updater / window
│   └── src/
│       ├── views/ # 所有页面视图
│       ├── forum/ # 论坛模块
│       ├── components/ # 通用组件
│       ├── composables/ # useAuth / useBackground / useAudio ...
│       └── utils/ # live2dController / session / config
  
```

```

├── system/           # 配置加载、环境检测、系统提示词
├── guide_engine/    # 游戏攻略引擎
└── logs/            # 运行日志、知识图谱文件

```

可选配置

- ▶ 知识图谱记忆 (Neo4j)
- ▶ 语音交互 (TTS / ASR)
- ▶ Live2D 虚拟形象 (自定义模型)
- ▶ MQTT 物联网控制

端口一览

服务	端口	说明
API Server	8000	主接口：对话、配置、认证、Skill 市场
Agent Server	8001	任务调度、OpenClaw
MCP Server	8003	MCP 工具注册与调度
Memory Server	8004	记忆服务
Voice Service	5048	TTS / ASR
Neo4j	7687	知识图谱 (可选)
OpenClaw Gateway	20789	AI 电脑控制 (可选)

故障排除

问题	解决方案
Python 版本报错	必须使用 Python 3.11；推荐用 uv 自动管理版本
端口被占用	检查 8000、8001、8003、5048 是否可用
Neo4j 连接超时 / 挂起	已在 2.24 修复；确认 Neo4j 服务已启动
Neo4j 连接状态误报	已在 4.12 修复 py2neo Graph() timeout 参数兼容性
TTS 无声音 / CORS 报错	已在 2.25 修复；确认 voice_enabled: true
启动卡在进度条	检查 API Key 是否正确；等待 3 秒后出现重启提示
悬浮球头像不显示	已在 2.17 修复序列帧路径；确认使用最新打包版本
config.json 乱码	已在 2.19 修复：config_manager 自动检测文件编码
OpenClaw 启动失败	已在 2.24 修复全局模式缺少配置文件的问题

```

python main.py --check-env --force-check # 完整环境诊断
python main.py --quick-check           # 快速检查

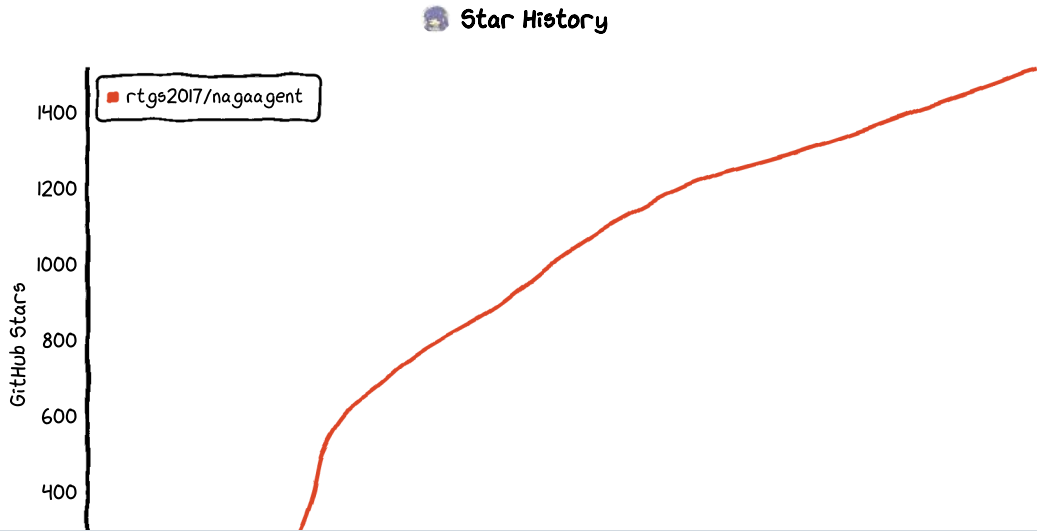
```



贡献

欢迎提交 Issue 和 Pull Request。如有问题，可加入 QQ 频道 [nagaagent1](#)。

Star History



Releases 1

NagaAgent 5.1.0 Latest
on Mar 6

Packages

No packages published

Contributors 26



[+ 12 contributors](#)

Languages

