

3.3 智能浏览器自动化

- Browser-use集成: 基于Playwright的LLM驱动浏览器
- 视觉能力: 启用vision支持
- 智能提示工程: 扩展系统提示处理机器人检测、付费墙、PDF下载等场景
- 结构化输出: 支持markdown/json/text多种格式

3.4 推理增强

- 专用推理模型: 使用KIMI-K2等推理模型处理复杂问题
- 问题分类: 支持数学、代码竞赛、逻辑谜题等场景
- 推理风格: 提供详细/简洁/逐步三种推理模式

3.5 任务编排与控制

- 配置管理: AgentConfig 统一LLM配置
- 会话管理: 通过 session_id 管理会话和工作空间
- 步数限制: max_steps=20 控制任务执行
- 自动评估: 集成GAIA数据集的自动答案评分

4. 环境搭建

4.1 详细安装步骤

步骤1: 获取项目

⚠️ 使用我提供的项目代码文件，官方项目代码有Bug，不要使用。

步骤2: 创建Conda环境

```
# 创建并激活环境
conda env create -f examples/gaia/aworld-gaia.yml
conda activate aworld-gaia
```

步骤3: 安装AWorld框架

```
# 进入根目录
cd AWorld/

# 构建Web UI
sh -c "cd aworld/cmd/web/webui && npm install && npm run build"

# 安装AWorld
pip install -e .
```

步骤4: 安装系统依赖

macOS:

```
# 安装Homebrew (如果未安装)
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

# 安装系统依赖
brew install libmagic ffmpeg
brew install --cask libreoffice

# 安装Playwright浏览器
pip install playwright
playwright install chromium --with-deps --no-shell
```

Linux (Ubuntu/Debian):

```
# 安装系统依赖
sudo apt-get update
sudo apt-get install -y --no-install-recommends \
    libmagic1 \
    libreoffice \
    ffmpeg

# 安装Playwright浏览器
pip install playwright
playwright install chromium --with-deps
```

步骤5: 下载GAIA数据集

```
# 需要先配置Hugging Face SSH密钥
# 访问: https://huggingface.co/settings/keys
pip install huggingface-hub

# 登录(需要输入密钥tokens)
huggingface-cli login

# 下载数据集
hf download gaia-benchmark/GAIA --repo-type dataset --local-dir
/Users/tanguoliang/Desktop/GAIA
```

步骤6: 配置环境变量

```
# 编辑.env文件, 填写必要的配置
vim gaia/.env
```

4.2 环境变量配置详解

```
# LLM 配置 - 硅基流动
LLM_PROVIDER=openai
LLM_MODEL_NAME=MiniMaxAI/MiniMax-M2
LLM_API_KEY=sk-hkjqqjegn1jrvvnxmkskgvuhqppjokjazlcypuxqxihhzhwgyar
LLM_BASE_URL=https://api.siliconflow.cn/v1
LLM_TEMPERATURE=0.0

# 可选配置
BROWSER_HEADLESS=false
BROWSER_WIDTH=1280
BROWSER_HEIGHT=720

# Browser Use 配置 - 用于浏览器自动化任务
# 如果使用 MCP 服务器, 需要单独配置
BROWSERUSE_LLM_BASE_URL=https://api.siliconflow.cn/v1
BROWSERUSE_LLM_MODEL_NAME=MiniMaxAI/MiniMax-M2
BROWSERUSE_LLM_API_KEY=sk-hkjqqjegn1jrvvnxmkskgvuhqppjokjazlcypuxqxihhzhwgyar

# =====Path Configurations=====
GAIA_DATASET_PATH="./data/GAIA/2023"
AWORLD_WORKSPACE="/tmp"

# =====MCP Server Configurations=====
# 访问 https://tavily.com 获取 Tavily API Key
TAVILY_API_KEY=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

# [E2B Server](https://e2b.dev/docs/quickstart) 获取 API Key
E2B_API_KEY=e2b_574a97a88989ad5f82f444dc21a1f86d2c802da1
```

```
# Image MCP Server 配置
IMAGE_LLM_MODEL_NAME="Qwen/Qwen3-VL-235B-A22B-Thinking"
IMAGE_LLM_API_KEY="sk-hkjqqjegn1jrvnxmskgvuhqpjokjazlcypuxqxihhzhwgyar"
IMAGE_LLM_BASE_URL="https://api.siliconflow.cn/v1"

# Reasoning MCP Server 配置
THINK_LLM_MODEL_NAME="deepseek-ai/DeepSeek-V3.2-Exp"
THINK_LLM_API_KEY="sk-hkjqqjegn1jrvnxmskgvuhqpjokjazlcypuxqxihhzhwgyar"
THINK_LLM_BASE_URL="https://api.siliconflow.cn/v1"
```

5. 运行方式详解

5.1 Web UI模式

Web UI提供可视化的交互界面，适合调试和演示。

5.1.1 启动步骤

```
# 1. 进入命令目录
cd examples/gaia/cmd

# 2. 启动Web UI
aworld web
```

5.1.2 访问界面

打开浏览器访问: `http://localhost:8000`



Ask or input / use skills



GAIA Agent



5.1.3 使用方式

方式1：运行GAIA任务

输入JSON格式的task_id:

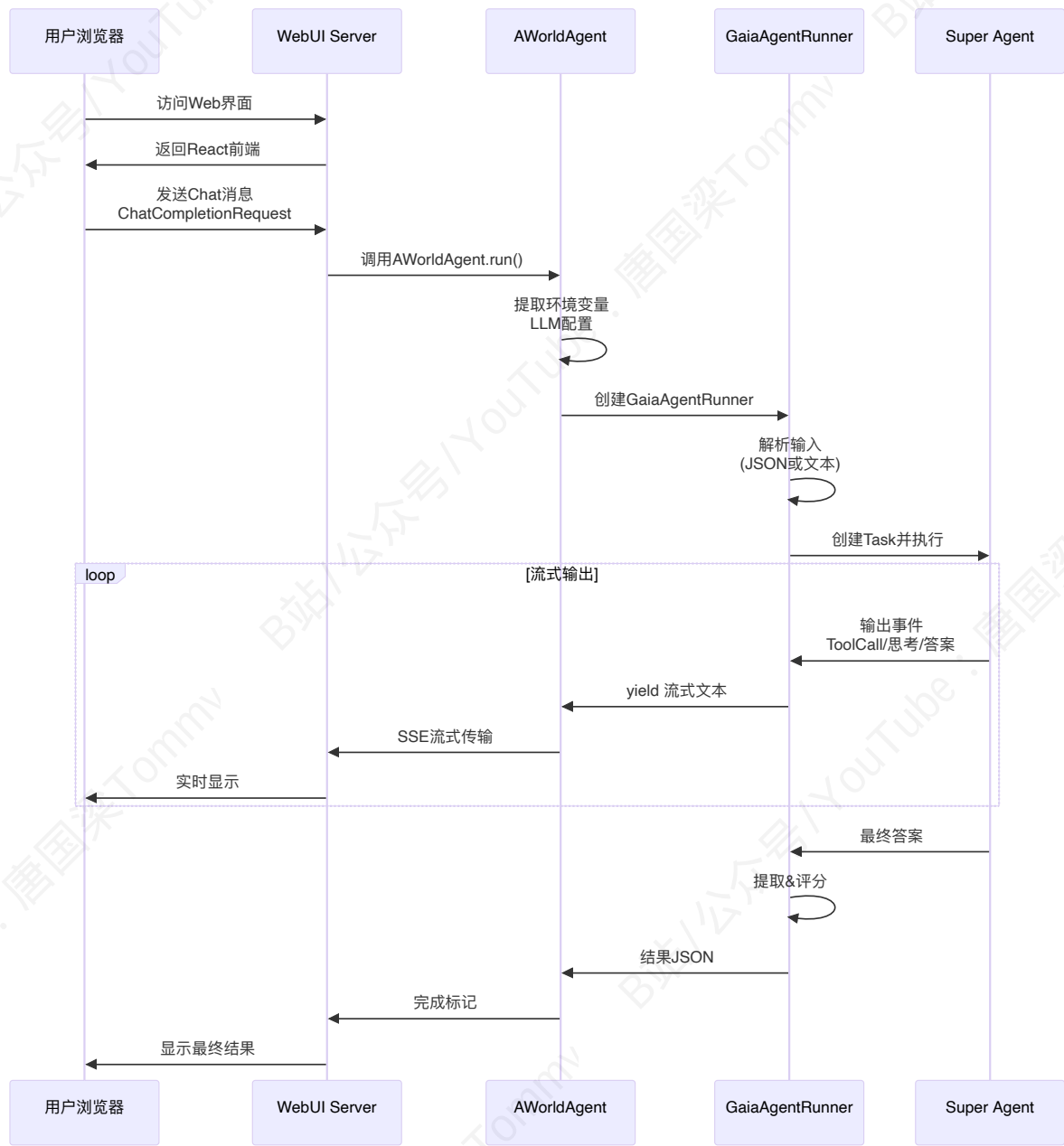
```
{ "task_id": "e1fc63a2-da7a-432f-be78-7c4a95598703" }
```

方式2：自由对话

直接输入问题:

```
帮我查询NVIDIA的最新股价
```

5.1.4 Agent workflow



5.2 命令行模式

命令行模式适合批量处理和自动化。

5.2.1 基础用法

```
# 运行单个问题
cd gaia/

python run.py \
    --split validation \
    --q e1fc63a2-da7a-432f-be78-7c4a95598703
```

```
# 运行一批问题 (索引0-19)
python examples/gaia/run.py \
    --split validation \
    --start 0 \
    --end 2
```

5.2.2 高级用法

1. 跳过已完成的问题

```
python examples/gaia/run.py \
    --start 0 \
    --end 100 \
    --skip
```

2. 使用黑名单

创建黑名单文件 `blacklist.txt`:

```
e1fc63a2-da7a-432f-be78-7c4a95598703
another-task-id
```

运行时指定:

```
python examples/gaia/run.py \
    --start 0 \
    --end 100 \
    --blacklist_file_path blacklist.txt
```

3. 运行测试集

```
python examples/gaia/run.py \  
  --split test \  
  --start 0 \  
  --end 100
```

5.2.3 Agent workflow

