

***SiE***



**IIOT**

赛意内部资料

Version: V1.0.0

**赛意工业物联网平台**

**「SiE IIOT」 AIGC 智能建模部署文档**

## 目录

|                             |   |
|-----------------------------|---|
| 1. 系统要求.....                | 1 |
| 1.1. 必需条件.....              | 1 |
| 1.2. 可选条件.....              | 1 |
| 2. 快速部署.....                | 1 |
| 2.1. 方法一：一键自动部署（推荐）.....    | 1 |
| 2.2. 方法二：手动部署.....          | 2 |
| 3. 运行服务.....                | 2 |
| 3.1. 方式一：使用启动脚本（推荐）.....    | 2 |
| 3.2. 方式二：手动启动.....          | 3 |
| 3.3. 方式三：使用 Docker（可选）..... | 3 |
| 4. 访问应用.....                | 3 |
| 5. 故障排除.....                | 4 |
| 5.1. 常见问题.....              | 4 |
| 5.2. 环境变量配置.....            | 5 |
| 5.3. 日志查看.....              | 5 |
| 5.4. 服务管理.....              | 5 |
| 5.5. 项目结构.....              | 6 |
| 5.6. 功能特性.....              | 6 |
| 6. 技术支持.....                | 7 |
| 6.1. 获取帮助.....              | 7 |
| 6.2. 常见问题解答.....            | 7 |
| 7. 相关链接.....                | 7 |

# 1. 系统要求

## 1.1 必需条件

- 操作系统: Linux/macOS/Windows (推荐 Linux)
- Python: 3.10 或更高版本
- 内存: 至少 8GB RAM (推荐 16GB+)
- 磁盘空间: 至少 20GB 可用空间
- 网络: 稳定的互联网连接 (首次运行需要下载模型)

## 1.2 可选条件

- GPU: NVIDIA GPU (可选, 用于加速模型推理)
- Docker: 用于容器化部署

# 2. 快速部署

## 2.1 方法一: 一键自动部署 (推荐)

### 1. 下载项目代码

```
```bash
git clone <repository-url>
```

当前仓库地址:[http://192.168.175.55:9888/IIOT/iiot\\_aigc](http://192.168.175.55:9888/IIOT/iiot_aigc)

```
cd IIoT_AIGC
```
```

### 2. 运行一键部署脚本

```
```bash
bash 一键部署脚本.sh
```
```

脚本将自动执行以下操作:

- 检查 Python 版本 (要求  $\geq 3.10$ )
- 创建 Python 虚拟环境 (PythonEnv\_312)
- 安装所有依赖包
- 创建必要的目录结构
- 验证安装结果

### 3. 选择安装源

脚本运行时会提示选择 Python 包安装源：

- 1) 清华镜像源（国内推荐，下载速度快）
- 2) 阿里云镜像源（国内备选）
- 3) 官方 PyPI 源（国外）

## 2.2 方法二：手动部署

如果一键脚本失败，可以手动执行：

### 1. 创建虚拟环境

```
```bash
python3 -m venv PythonEnv_312
source PythonEnv_312/bin/activate # Linux/macOS
# 或在 Windows 下: PythonEnv_312\Scripts\activate
```
```

### 2. 安装依赖

```
```bash
pip install -r requirements.txt
```
```

### 3. 创建必要目录

```
```bash
mkdir -p uploads exports logs chat_logs rag_storage static/graphs static/owl_files
```
```

## 3.快速部署

### 3.1 方式一：使用启动脚本（推荐）

```
```bash
./start_with_local_model.sh
```
```

脚本将自动：

- 设置本地模型环境变量
- 强制使用 CPU 模式（避免 GPU 兼容性问题）
- 激活 Python 虚拟环境

- 检查并释放 5002 端口
- 显示环境信息
- 启动 Web 服务

## 3.2 方式二：手动启动

### 1. 激活虚拟环境

```
```bash
source PythonEnv_312/bin/activate
```
```

### 2. 设置模型环境

```
```bash
source GraphRAG_Tool/set_model_env.sh
```
```

### 3. 启动服务

```
```bash
python run.py
```
```

## 3.3 方式三：使用 Docker（可选）

如果您有 Docker 环境：

```
```bash
# 构建镜像
docker build -t iiot-aigc .

# 运行容器
docker run -p 5002:5002 -v $(pwd)/data:/app/data iiot-aigc
```
```

# 4. 访问应用

部署成功后，访问以下地址：

- 主应用：<http://localhost:5002>
- API 文档：<http://localhost:5002/docs>（如果支持）

## 默认配置

- 端口: 5002
- 主机: 127.0.0.1 (本地访问)
- 模型: 本地模型模式

# 5.故障排除

## 5.1 常见问题

### 1. Python 版本不兼容

...

错误: Python 版本 X.X 不满足要求, 需要  $\geq$  3.10

...

解决: 升级 Python 版本或使用 conda 创建新环境

### 2. 依赖安装失败

...

错误: pip install 失败

...

解决:

- 检查网络连接
- 尝试更换安装源
- 手动安装失败的包

### 3. 端口被占用

...

错误: 5002 端口被占用

...

解决:

- 脚本会自动释放端口
- 或手动杀死占用进程: ``lsof -ti:5002 | xargs kill -9``

### 4. 内存不足

...

错误: 内存不足

...

解决:

- 关闭其他程序

- 增加系统内存
- 使用更小的模型

## 5. 模型下载失败

```

错误：无法下载模型

```

解决：

- 检查网络连接
- 设置代理（如果在国内网络环境）
- 使用离线模型包

## 5.2 环境变量配置

如果遇到模型路径问题，可以手动设置：

```
```bash
export HF_HOME="./GraphRAG_Tool/LLM_Debedding"
export TRANSFORMERS_CACHE="${HF_HOME}/transformers"
export HF_HUB_OFFLINE=0
```
```

## 5.3 日志查看

查看应用日志：

```
```bash
tail -f logs/app.log
```
```

## 5.4 服务管理

停止服务：

- 在终端按 `Ctrl+C`
- 或查找进程：`ps aux | grep python`

重启服务：

```
```bash
```

```
./start_with_local_model.sh
...
```

## 5.5 项目结构

```
...
IIoT_AIGC/
├── PythonEnv_312/          # Python 虚拟环境
├── 一键部署脚本.sh        # 自动部署脚本
├── start_with_local_model.sh # 服务启动脚本
├── requirements.txt        # Python 依赖
├── run.py                  # 主应用入口
├── app.py                  # Flask 应用
├── modules/               # 核心模块
│   ├── llm/               # LLM 集成
│   ├── rag/               # RAG 系统
│   ├── modeling/          # 建模功能
│   └── prompt/            # 提示词管理
├── GraphRAG_Tool/         # GraphRAG 工具
├── static/                 # 静态文件
├── templates/             # 模板文件
├── docs/                   # 文档
└── logs/                   # 日志文件
...
```

## 5.6 功能特性

- **智能对话**: 支持多种大语言模型
- **文档处理**: 支持多种文档格式
- **知识图谱**: GraphRAG 增强检索
- **模型微调**: 支持本地模型训练
- **API 接口**: RESTful API 设计
- **多租户**: 支持多用户隔离

## 6.技术支持

### 6.1 获取帮助

1. 查看日志: ``tail -f logs/app.log``
2. 检查端口: ``netstat -tlnp | grep 5002``
3. 验证环境: ``python -c "import flask; print('OK')"``

### 6.2 常见问题解答

Q: 为什么启动很慢?

A: 首次运行需要下载模型文件, 请耐心等待。

Q: 如何更换模型?

A: 修改 ``GraphRAG_Tool/set_model_env.sh`` 中的环境变量。

Q: 如何备份数据?

A: 备份 ``rag_storage/`` 和 ``exports/`` 目录。

Q: 如何升级版本?

A: 重新运行部署脚本, 或手动更新依赖。

## 7.相关链接

- [详细文档](./docs/)
- [API 文档](./docs/api/)
- [部署指南](./docs/一键部署指南.md)
- [故障排除](./docs/troubleshooting.md)

最后更新时间: 2025-01-07

版本: v1.0.0