



启动脚本使用指南 (Script Usage Guide)

为了方便在 Linux (Ubuntu/CentOS) 或 Windows 上长期稳定运行机器人，我们提供了位于项目根目录的一键启动脚本。

适用版本: v3.0 (Async Core) 及以上



脚本功能

1. **自动环境识别**: 智能检测 `venv`, `conda` 或系统 Python 环境。
2. **后台静默运行**: (Linux) 使用 `nohup` 让机器人在后台运行，关闭 SSH 窗口也不会断开。
3. **防重复启动**: 自动检测是否已有 `OKXBot_Plus.py` 实例运行，防止资金冲突。
4. **启动自检**: 启动后自动监控前 5 秒状态，确保没有报错退出。



Linux / Mac 快速开始

1. 赋予执行权限 (首次需要)

进入项目根目录:

```
1 | cd OKXBot_Plus_Workspace
2 | chmod +x start_bot.sh
```

2. 启动机器人

直接运行根目录下的脚本:

```
1 | ./start_bot.sh
```

启动成功示例:

```
1 | [✓] 检测到已激活的 Conda 环境: /root/anaconda3/envs/okx_ds
2 | ⚡ 正在启动后台进程...
3 | ⌚ 正在验证进程状态...
4 | [✓] 启动成功! 机器人正在后台运行。
5 | [ID] 进程 PID: 69014
6 | [📄] 日志文件: log/console_output.log
```

3. 查看日志

脚本会将控制台输出重定向到 `log/console_output.log`，而详细的交易日志在 `log/trading_bot_async_xxx.log`。

```
1 | # 查看控制台输出 (启动报错看这里)
2 | tail -f log/console_output.log
3 |
4 | # 查看详细交易逻辑
5 | tail -f log/trading_bot_*.log
```

□ Windows 快速开始

启动方法

1. 进入项目根目录 `OKXBot_Plus_Workspace`。
2. 双击运行 `start_bot.bat`。
3. **注意:** 请保持黑色命令窗口开启。Windows 暂不支持像 Linux 那样的 `nohup` 后台模式，关闭窗口会停止机器人。

故障排查 (Diagnostics)

如果遇到启动失败或网络问题，建议先运行诊断工具：

```
1 # 测试 API 连接和通知功能
2 python test/test_connection.py
```

该工具会分别检测：

1. OKX API 连通性及余额读取。
2. DeepSeek API 连通性。
3. Webhook 通知推送（支持飞书/钉钉）。

高级配置 (自定义虚拟环境)

如果您使用了自定义名称的虚拟环境（例如 `okx_ds`），`start_bot.sh` 默认可能找不到。

方法 A：先激活环境 (推荐)

```
1 conda activate okx_ds
2 ./start_bot.sh
```

方法 B：修改脚本配置

编辑 `start_bot.sh`，修改 `CUSTOM_VENV_NAME` 变量：

```
1 CUSTOM_VENV_NAME="okx_ds"
```

运维管理

停止机器人 (Linux)

```
1 # 查找 PID
2 ps -ef | grep OKXBot_Plus.py
3
4 # 停止进程
5 kill <PID>
```

更新代码

如果您更新了代码，请先停止机器人，拉取代码，然后重新运行启动脚本。

```
1 | git pull
2 | ./start_bot.sh
```