

SOP（每日运行流程）

0. 目标

1. 使用单一 MySQL/SQLite 数据库存储 K 线 + 模型输出。
2. 通过三个脚本完成“抓数 → 评分 → 连板信号”，UI 只读展示。
3. 不再依赖 CSV/bingwu/旧 pipeline。

1. 环境准备

- Python 3.10+, 安装依赖：

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

- 配置 `config.ini`（推荐写 `db_url`）；也可使用 CLI `--db-url` / 环境变量 `ASTOCK_DB_URL`。

优先级：`--db-url` > `ASTOCK_DB_URL` > `--db` > `config.ini` > 默认 `data/stock.db`。

2. 初始化（一次）

```
python init.py --config config.ini
```

说明：

- 自动 `CREATE DATABASE IF NOT EXISTS`（如连接为 MySQL）。
- 创建
`stock_info/stock_daily/stock_scores_v3/stock_levels/stock_scores_ywcx/stock_scores_stwg/model_laowang_pool/model_ywcx_pool/model_stwg_pool/model_fhkq`。

3. 抓取 K 线

```
python getData.py --config config.ini --start-date 20250101 --end-date 20260123 --workers 16
```

- `start-date/end-date` 支持 `YYYYMMDD` / `YYYY-MM-DD`。
- 多线程按股票更新，`tqdm` 显示每只股票进度。
- 只负责写入 `stock_info` + `stock_daily`。

4. 计算 LAOWANG

```
python scoring_laowang.py --config config.ini --start-date 2026-01-01 --end-date 2026-01-21 --workers 32 --top 200 --min-score 60
```

- 读取 `stock_daily`，对区间内所有交易日逐股计算评分。
- 结果写入 `stock_scores_v3`、`stock_levels`、`model_laowang_pool`。
- 已内置指标/支撑/评分逻辑，无需其他模块。

5. 计算 YWCX

```
python scoring_ywcx.py --config config.ini --start-date 2026-01-01 --end-date 2026-01-23 --workers 32 --top 120 --min-score 55
```

- 根据 docs/scoring_ywcx.md 的阳痿次新模型，逐股计算评分。
- 结果写入 `stock_scores_ywcx`、`model_ywcx_pool`。

6. 计算 STWG

```
python scoring_stwg.py --config config.ini --start-date 2026-01-01 --end-date 2026-01-22 --workers 32 --top 150 --min-score 55
```

- 根据 docs/scoring_stwg.md 描述的缩头乌龟模型，逐股计算评分。
- 结果写入 `stock_scores_stwg`、`model_stwg_pool`，供 UI 展示。

7. 计算 FHKQ

```
python scoring_fhkq.py --config config.ini --start-date 2026-01-01 --end-date 2026-01-21 --workers 16
```

- 按交易日遍历，寻找连板跌停候选股。
- 候选股按最近 120 日历史并发计算，写入 `model_fhkq`。

8. 查看结果 + 自动任务

```
python ui.py --config config.ini --start-date 20260101 --host 0.0.0.0 --port 80  
# 浏览 http://127.0.0.1:8000
```

- UI 仅从 DB 读取，不会触发计算，但会在后台展示实时数据。
- 默认情况下，UI 会在每日 **15:05** 启动后台线程调用 `everyday.py`，自动完成“抓数 + 模型”。如需关闭可加 `--disable-auto-update`。

- `--start-date` 可限制交易日下拉框只展示近期日期（例如 2024 以后）。

9. 每日例行流程

推荐：

```
python everyday.py --config config.ini --initial-start-date 2026-01-23
```

它会自动：

1. 查询 `stock_daily` 最新日期，增量调用 `getData.py`（若已最新则跳过）。
2. 调用 `scoring_laowang.py`、`scoring_ywcx.py`、`scoring_stwg.py`、`scoring_fhkq.py`，评分区间覆盖“新增交易日或最新一日”。
3. 所有参数（线程数、TopN 等）都可以通过 CLI 覆盖。

UI 已内置相同逻辑：运行时会在每日 15:05 自动调用 `everyday.py`。如需禁用，启动 UI 时加 `--disable-auto-update`。

10. 排错与常见问题

- 确认连接的是哪个 DB：借助 `config.ini + --db-url`，或使用 `SELECT database();`。
- K 线缺失：`SELECT COUNT(*) FROM stock_daily WHERE date='2024-05-31';`
- 评分缺失：`SELECT COUNT(*) FROM stock_scores_v3 WHERE score_date='2024-05-31';`
- UI 没数据：请先运行对应脚本填充 `model_*` 表。

所有旧的多模块脚本已移至 `recycle_bin/`，如需参考历史实现，可自行查阅。