

目录

| | |
|-------------------------------------|----|
| Mysite 网页安装说明..... | 2 |
| 前言 | 2 |
| 一、Python 环境的安装配置 | 2 |
| 1. Ananconda 的安装 | 2 |
| 2. 环境变量测试 | 3 |
| 二、部署 Web 项目 Mysite | 3 |
| 1. 选择部署根目录(绝对路径) | 3 |
| 三、Django 框架及附属包的安装 | 4 |
| 四、SQLiteExpertPersonal 的安装与关联 | 4 |
| 1. 安装 SQLiteExpert | 4 |
| 2. 测试安装是否成功 | 6 |
| 3. 关联数据库 | 7 |
| 五、Django 模型的数据库迁移 | 9 |
| 六、阿里云服务器安全组配置 | 11 |
| 七、PNL Sender 介绍与启动 | 11 |
| 7.1 使用方法 | 11 |
| 7.2 Sender 的重要附属文件: | 12 |
| 7.3 PNL 计算流程: | 12 |
| 7.4 Sender 使用注意事项: | 12 |
| 八、文件依赖关系介绍 | 13 |
| 8.1 json 数据文件路径介绍 | 13 |
| 8.2 策略管理层级 | 13 |
| 九、测试启动 Web | 14 |
| 十、其他 | 15 |

Mysite 网页安装说明

前言

本项目基于 Django 框架开发，需对接 AntHill 程序化交易系统，可以实现实时分账户分策略对程序化交易进行实时监控，历史净值清算与记录，以及策略管理的可视化。

在服务器上部署该网页平台，主要有以下几个步骤：

1. Python 环境的安装配置
2. 选择部署 mysite 的路径
3. Django 框架及附属包的安装
4. 数据库 SQLiteExpertPersonal 的安装
5. 项目模型迁移至数据库
6. 阿里云安全组设置
7. 项目文件依赖关系介绍
8. 测试启动 Web

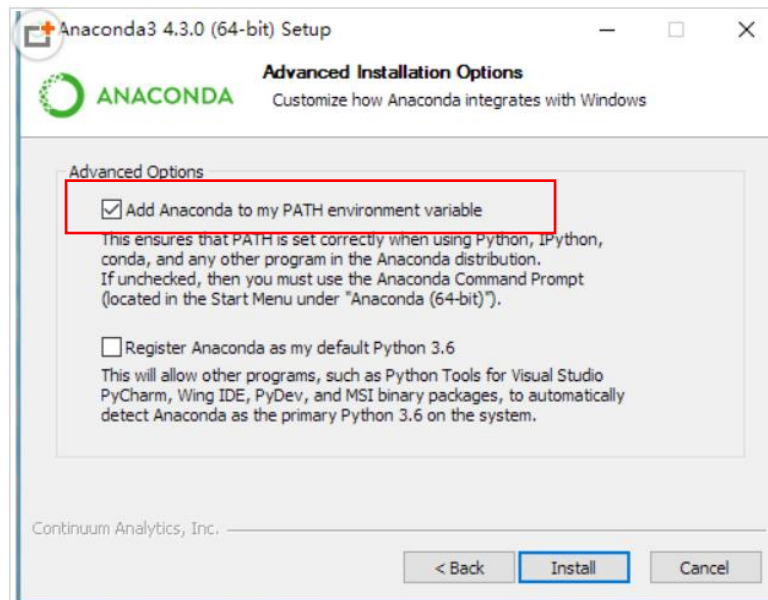
一、Python 环境的安装配置

1. Ananconda 的安装

可通过清华镜像下载 Anaconda 安装包，下载版本号 3.5 或以上的版本，链接：

<https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/>

双击安装包，按步骤依次进行到最后一步，安装在 C 盘即可，勾选 Add Anaconda to my PATH...，点击 Install 完成安装。



2. 环境变量测试

Windows 搜索框输入 cmd, 回车, 输入 python, 不报错且可进行简单计算, 即说明环境变量配置成功。

```
cmd 命令提示符 - python
Python 3.7.0 (default, Jun 28 2018, 08:04:48) [MSC v.1912 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 1+1
2
>>> _
```

若显示“python 不是内部或外部命令, 也不是可运行的程序”, 则需要手动添加 anaconda python, 详情请参考百度经验

<https://jingyan.baidu.com/article/6dad5075b9d9b8e122e36e69.html>

注: Anaconda 本身带有丰富的 Python 工具包, 建议使用 Anaconda 作为 python 的解器。

二、部署 Web 项目 Mysite

1. 选择部署根目录(绝对路径)

在 C 盘根目录下新建文件夹 DjangoWeb, 将压缩包 mysite 以“解压到当前文件夹”的方式解压到 DjangoWeb 中, 双击打开

“C:/DjangoWeb/mysite/interface/datapath.py”

三、Django 框架及附属包的安装

此项目需要导入 django 的包有以下几个，请依次进行安装

```
pip install django,
```

```
pip install django_apscheduler
```

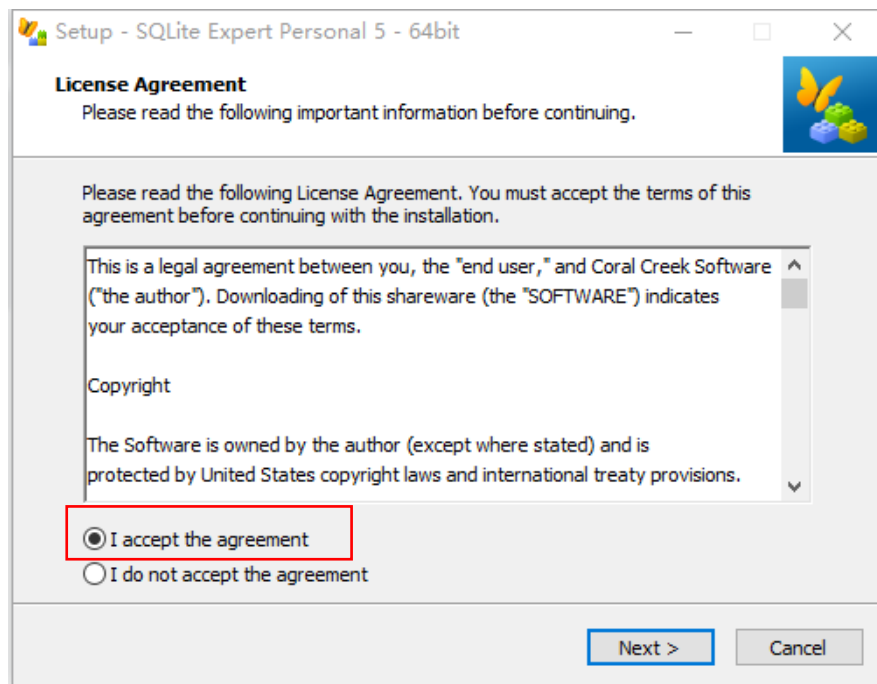
导入显示成功即可，也可以在 Ananconda Prompt 中通过 conda 安装。

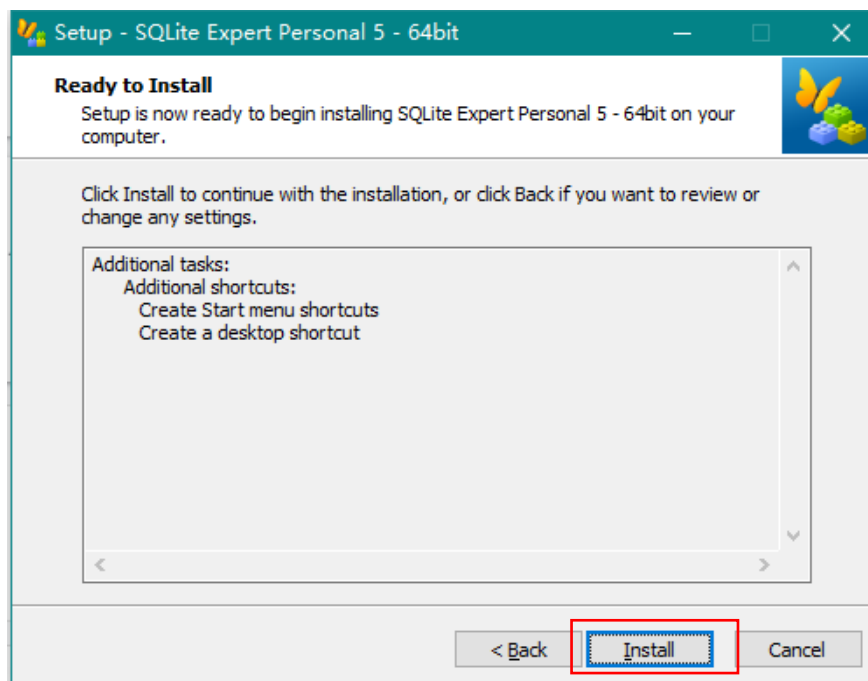
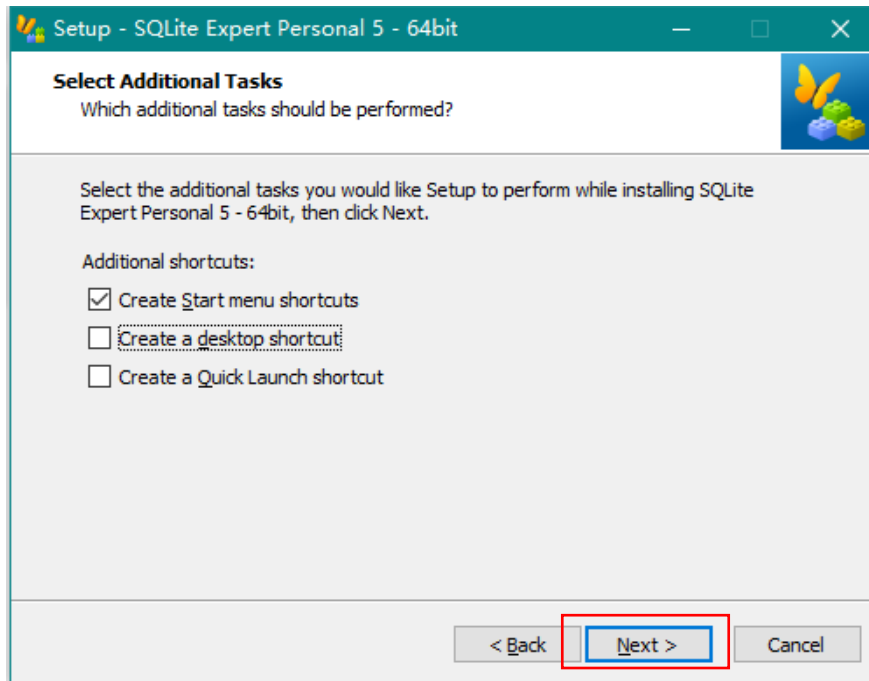
注：其他诸如 Pandalas, datetime 等常见包不做阐述，如未安装，请自行安装。

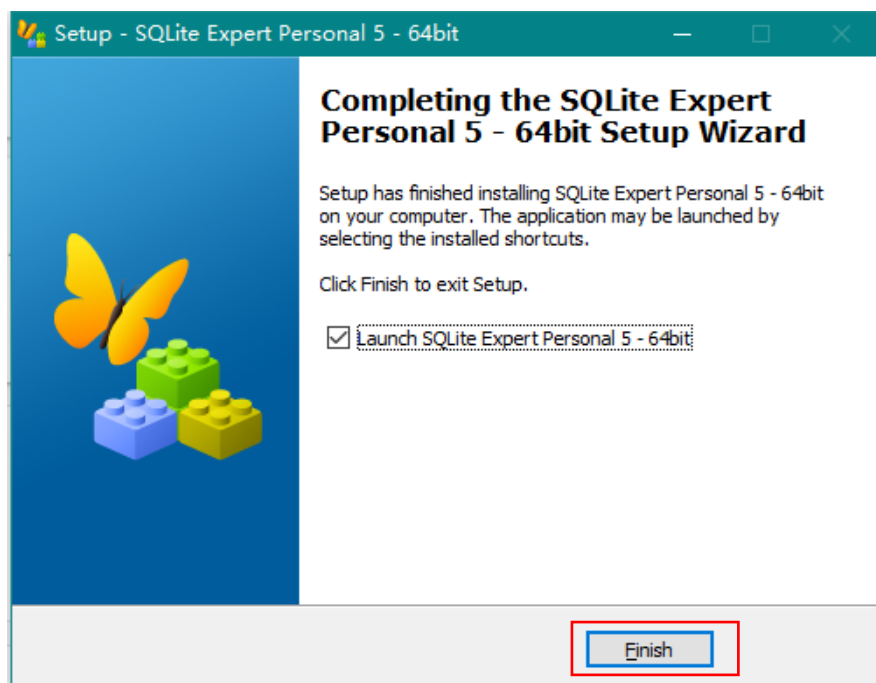
四、SQLiteExpertPersonal 的安装与关联

1. 安装 SQLiteExpert

双击 SQLiteExpertPersSetup64.exe 逐步点击安装即可。（个人版不收费）







说明: *SQLiteExpert* 是一个轻量型的数据库软件, 我们的网页展示不需要进行大数据管理, 并且考虑现实交易情况, 往往需要对交易流水或数据进行修改, 重复上传等操作, 使用数据库管理运行成本较高, 我们选取文件管理的方法, 以减少数据库运维成本, 在此, 选用 *Django* 默认的轻量型数据库, 仅保存 *Django* 的账号密码信息 *user*, 定时任务信息 *task (django-apscheduler)*, 及浏览器缓存信息 *session*。

2. 测试安装是否成功

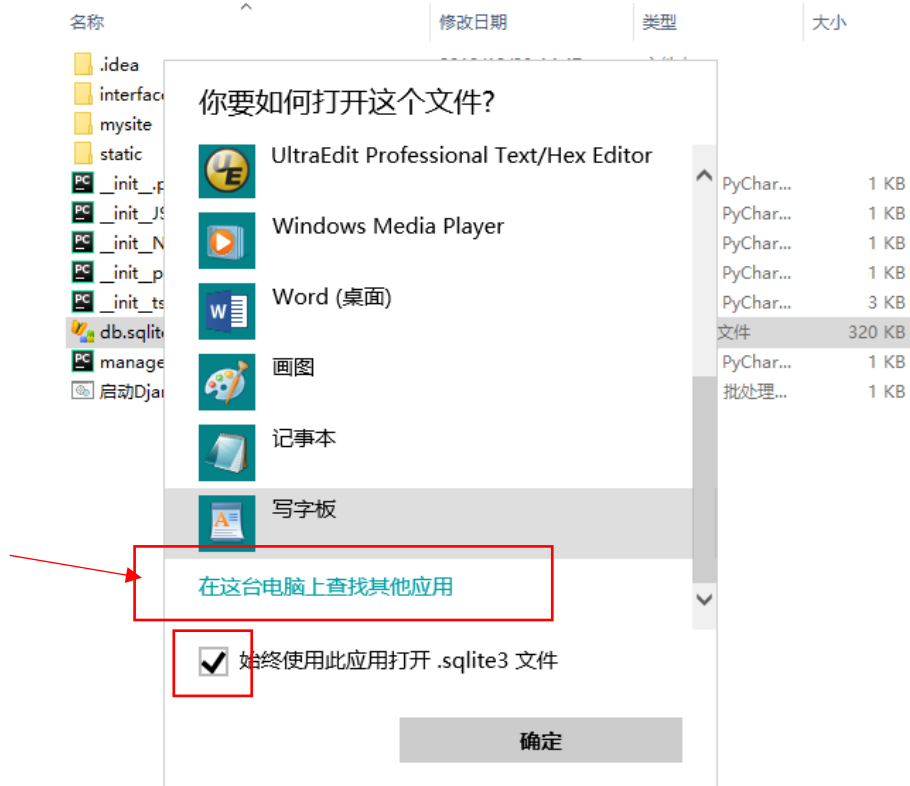
安装成功后, 在搜索栏搜索 *sqlite*, 应出现开始启动项

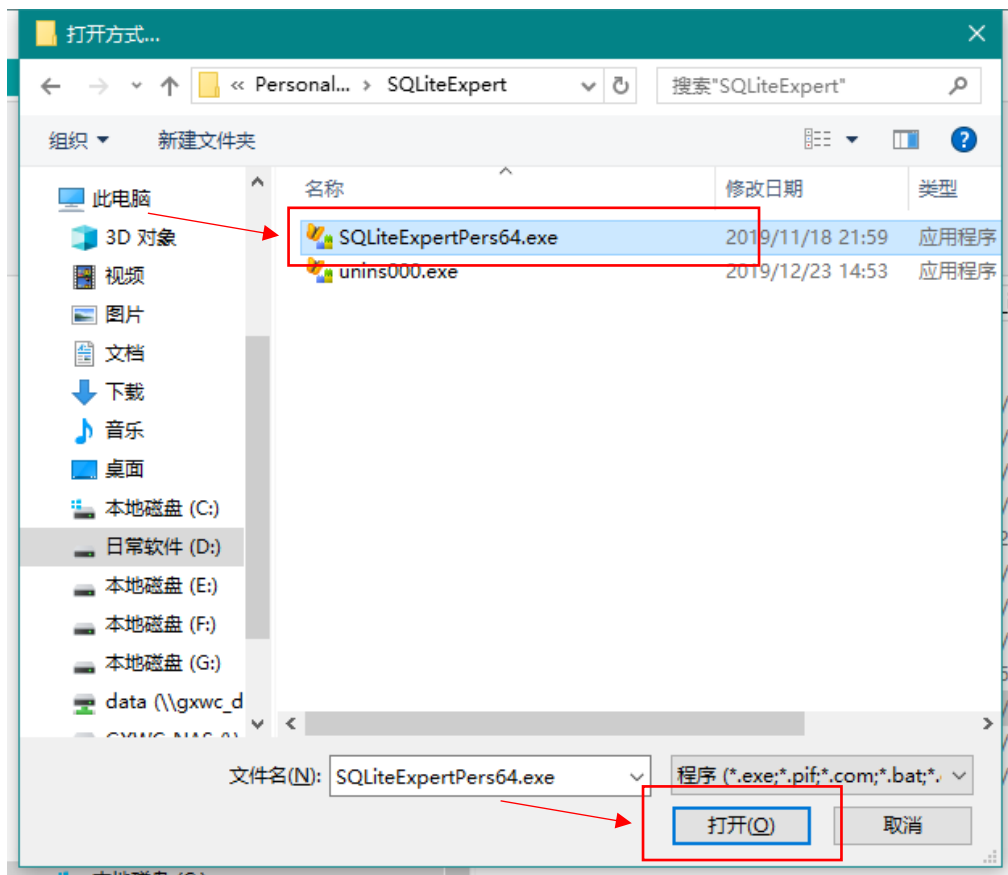


3. 关联数据库

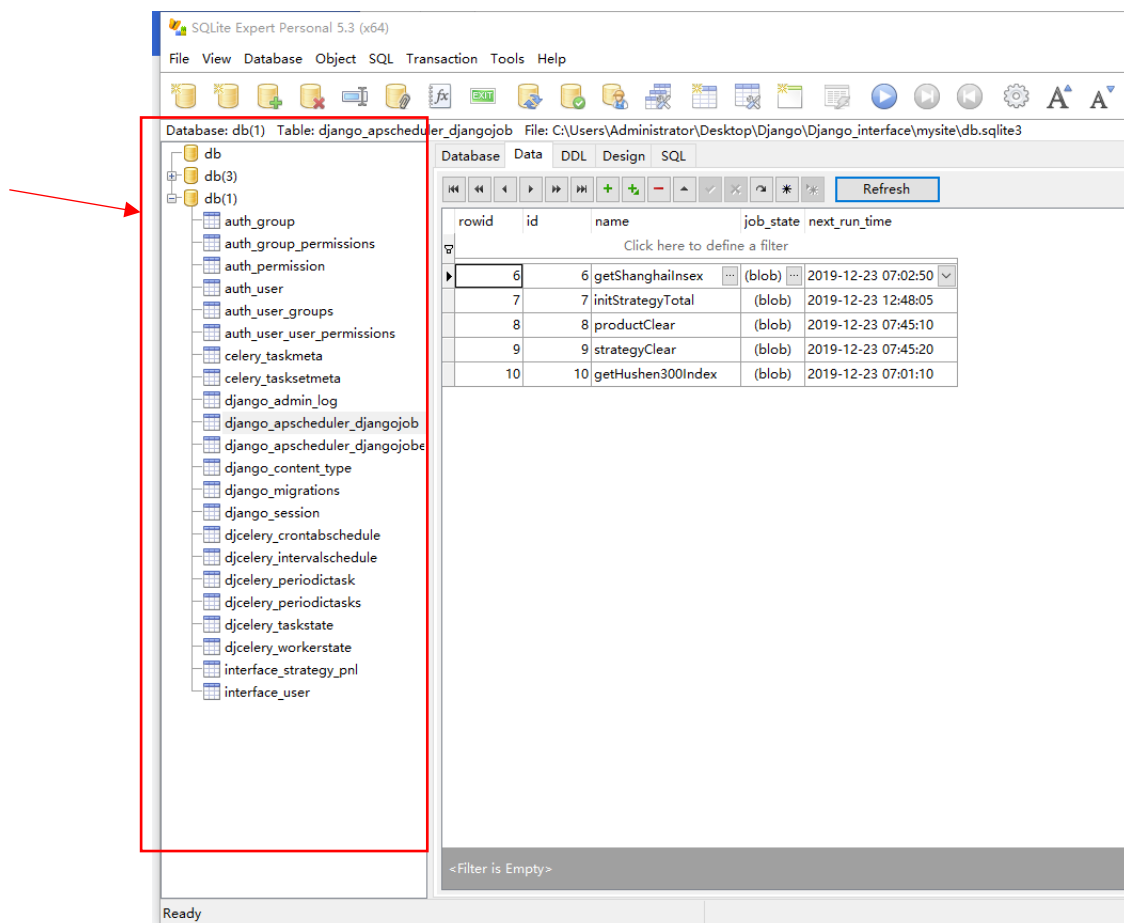
Mysite 中已有数据库表文件，需要关联 SQLite 进行打开或修改操作，需要把这个表文件的“打开方式”设置为 SQLite 即可，即为如下步骤：

打开“C:/DjangoWeb/mysite/”路径，双击或右击 db.sqlite3，选择打开方式（在刚才安装 Sqlite 的路径下找到 QLiteExpertPers64.exe





关联打开方式成功后，打开 db.sqlite3 可以看到如下界面



除了 Django 自动生成的数据表，其他数据表简单介绍：

django_apscheduler_djangojob

是应用 django_apscheduler 应用注册的 scheduler model 类，主要包括每天 20:58 分对策略进行初始化，收盘前从新浪财经拿到上证指数，沪深 300 与科创板指数等定时任务。定时任务详情参见 task.py。

django_apscheduler_djangojobexecution

记录所有的定时任务的执行情况，若为正常执行，会将错误信息保存在该表中。

interface_user

是应用 interface 应用注册的用户 model 类，所有网页用户的用户名与密码保存在该表。

五、Django 模型的数据库迁移

说明：由于 Django 的 ORM 模式（即将数据库中的表映射成类，才能在 View 视图函数中进行使用），所以项目中其建立的所有 Model 都要进行 migrate 操作对 Model 进行类的映射。

首先，确保“C:\DjangoWeb\mysite\mysite\settings.py”中如下配置正确（默认情况下，请勿更改 settings.py 的设置）

数据库引擎设置

```
81
82 DATABASES = {
83     'default': {
84         'ENGINE': 'django.db.backends.sqlite3',
85         'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
86     }
87 }
88
```

INSTALLED_APPS 设置

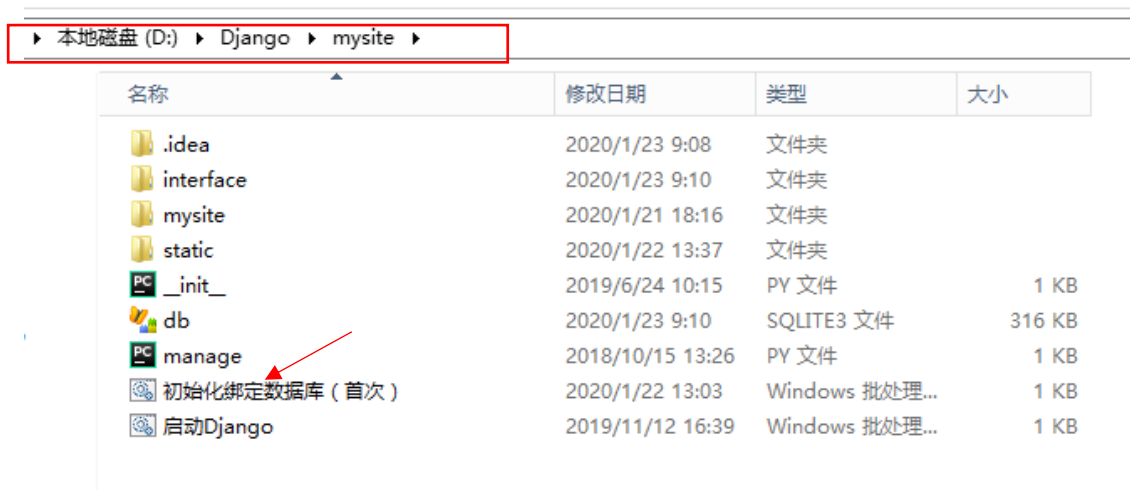
```

35 INSTALLED_APPS = [
36     'django.contrib.admin',
37     'django.contrib.auth',
38     'django.contrib.contenttypes',
39     'django.contrib.sessions',
40     'django.contrib.messages',
41     'django.contrib.staticfiles',
42     'django_apscheduler', #定时执行任务
43     'interface',
44     'mysite'
45 ]
46 APSCHEDULER_DATETIME_FORMAT = "%N j, Y, f:s a" # Default
47 MIDDLEWARE = [
48     'django.middleware.security.SecurityMiddleware',
49     'django.contrib.sessions.middleware.SessionMiddleware',
50     'django.middleware.common.CommonMiddleware',
51     # 'django.middleware.csrf.CsrfViewMiddleware',
52     'django.contrib.auth.middleware.AuthenticationMiddleware',
53     'django.contrib.messages.middleware.MessageMiddleware',
54     'django.middleware.clickjacking.XFrameOptionsMiddleware',
55 ]
56

```

确认以上设置无误后，执行数据库迁移，操作方法如下：

在“C:/Django/mysite/mysite/manage.py”所在路径，在红框中输入cmd，回车



执行如下语句（或只点击“初始化绑定数据库（首次）.bat”）：

```

python manage.py makemigrations
python manage.py migrate

```

注：执行数据库迁移的前提是不缺包不漏包，且已 SQLite 已经成功关联，若执行一次不成功，可再执行一次。

```
管理员: C:\Windows\System32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

D:\Django安装\mysite>python manage.py makemigrations
Job with id initStrategyTotal already in jobstore. I'll refresh it
Job with id getShanghaiInsex already in jobstore. I'll refresh it
Job with id productClear already in jobstore. I'll refresh it
Job with id strategyClear already in jobstore. I'll refresh it
Job with id getHushen300Index already in jobstore. I'll refresh it
No changes detected

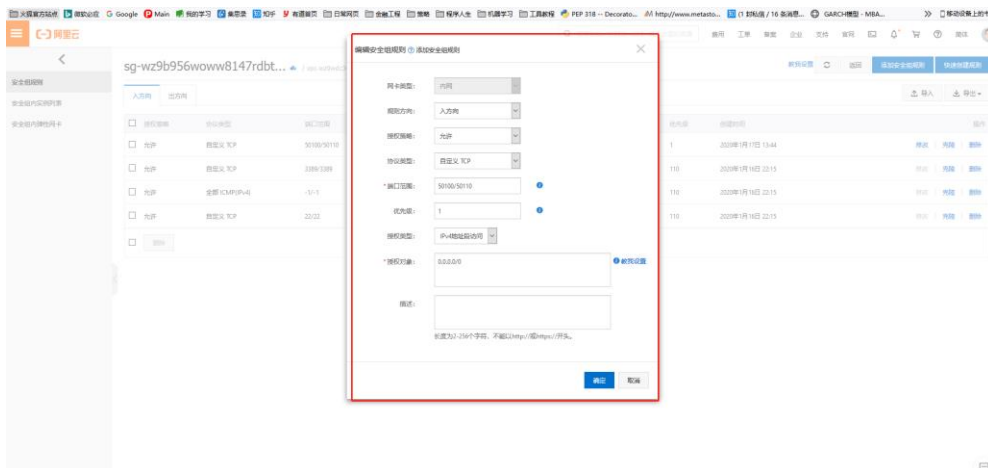
D:\Django安装\mysite>python manage.py migrate
Job with id initStrategyTotal already in jobstore. I'll refresh it
Job with id getShanghaiInsex already in jobstore. I'll refresh it
Job with id productClear already in jobstore. I'll refresh it
Job with id strategyClear already in jobstore. I'll refresh it
Job with id getHushen300Index already in jobstore. I'll refresh it
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, django_apscheduler, interface, sessions
Running migrations:
  No migrations to apply.

D:\Django安装\mysite>
```

六、阿里云服务器安全组配置

端口配置文件

打开 50100 到 50110 的端口为 TCP 和 UDP



七、PNL Sender 介绍与启动

7.1 使用方法

双击“盘中实时数据传输.bat”

7.2 Sender 的重要附属文件:

数据样板: 至少应包括

last_price.csv(最新价文件) ,
yesterday_position.csv(昨持仓文件),
xxxx_trade.csv(交易流水),
xxxx_order.csv(订单流水),

其中 xxxx 为当前交易日期。具体数据格式请参见文件夹数据样板

1. config_path.csv

配置文件, 包含策略名 strategy_name 及对应的数据路径 strategy_path.

2. config.ini

服务器等属性配置文件, 包括服务器公网地址及 Django 启动的端口号

3. instrument_fee.csv

配置手续费参数文件, 直接读进内存。

4. stock_names.csv

Web 显示证券名称, 直接读进内存。格式: 000001.SZ, 平安银行, 需定期更新, 添加新上市证券代码及名称。

5. 盘中实时数据传输

对应脚本: 盘中实时数据传输.bat

新服务器: 120.77.179.170:50100

7.3 PNL 计算流程:

--> 确定今天交易日期(期货夜盘算明天)

--> 通过 init_CurrentPosition 读取昨持仓文件, 生成今日开盘持仓详情 current_position(字段与 Web 网页显示一致)

--> 删除 Django 端的持仓与订单

--> 实时监控交易流水文件与订单流水文件, 确定当前时刻已经计算到第几条

--> 如有新交易流水发生, 计算交易证券标的的当前持仓, 刷新持仓(计算较复杂)

-->> 手续费对于成本价的影响是: 对于多头仓位, 不管是开仓还是平仓, 都会拉高成本价。但是对于空头仓位。不管是开仓还是平仓, 都会降低成本价

--> 待当前策略所有持仓的新增交易流水都刷新之后, 再根据其最新价文件, 更新持仓的最新价, 重新计算当前的持仓盈亏等指标

--> 遍历字典 current_position, 分批发送所有策略的持仓及 pnl 盈亏

--> 新增订单直接发送

7.4 Sender 使用注意事项:

1. 若发现网页持仓显示异常, 90%的概率是由于昨持仓或交易流水的数据格式与数据样板中的数据有出入, 或者是 lastprice 没有某只股票的价格等等。

2. Sender 中添加了很多 try:except, 以规避诸如: 网络不稳定(可通过 timeout 参数设置更大的延时重连), 无数据文件(自动生成一个带表头的空文件)等问题

八、文件依赖关系介绍

8.1 json 数据文件路径介绍

| 本地磁盘 (D:) ▸ Django ▸ mysite ▸ static ▸ tableData ▸ | | | |
|--|------------------|---------|--|
| 名称 | 修改日期 | 类型 | |
| CommodityPriceData | 2019/12/30 15:37 | 文件夹 | |
| HistoryNetValue | 2020/1/22 12:29 | 文件夹 | |
| Manage | 2020/1/22 12:48 | 文件夹 | |
| TempData | 2020/1/22 16:03 | 文件夹 | |
| all_strategy_pnl | 2020/1/22 17:12 | JSON 文件 | |

BackUpData: 备份每天所有的 json 数据文件

CommodityPriceData: 现货价格数据, 对应 Web 现货投研模块

HistoryNetValue: 历史净值路径, 保存所有策略及产品的历史净值等数据

Manage: 保存策略管理的三张表

产品配置表 product_manage.json

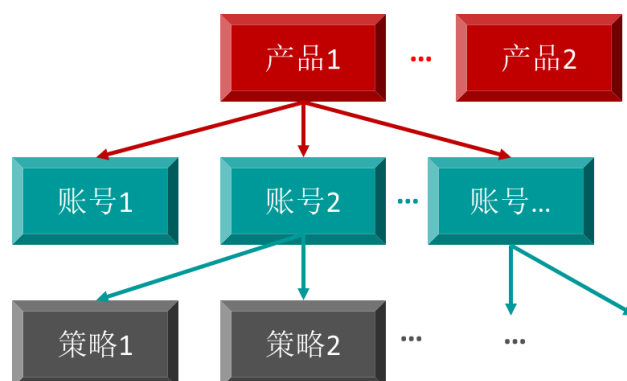
产品-账号对应表 product_account_manage.json

策略配置表 strategy_manage.json

TempData: 动态临时文件, 其中包括各策略的持仓及订单详情

all_strategy_pnl.json 文件记录所有 sender 端的 pnl 数据, 会在 Sender 启动的时候不断地被读写, 若该文件不存在或出现问题导致无法读写, 则会从 Manage 中读取复制生成一份。

8.2 策略管理层级



简言之, 就是一个产品可以存在多个账号交易, 一个账号可以有多个策略在运行。
PNL 逐层叠加结算。

九、测试启动 Web

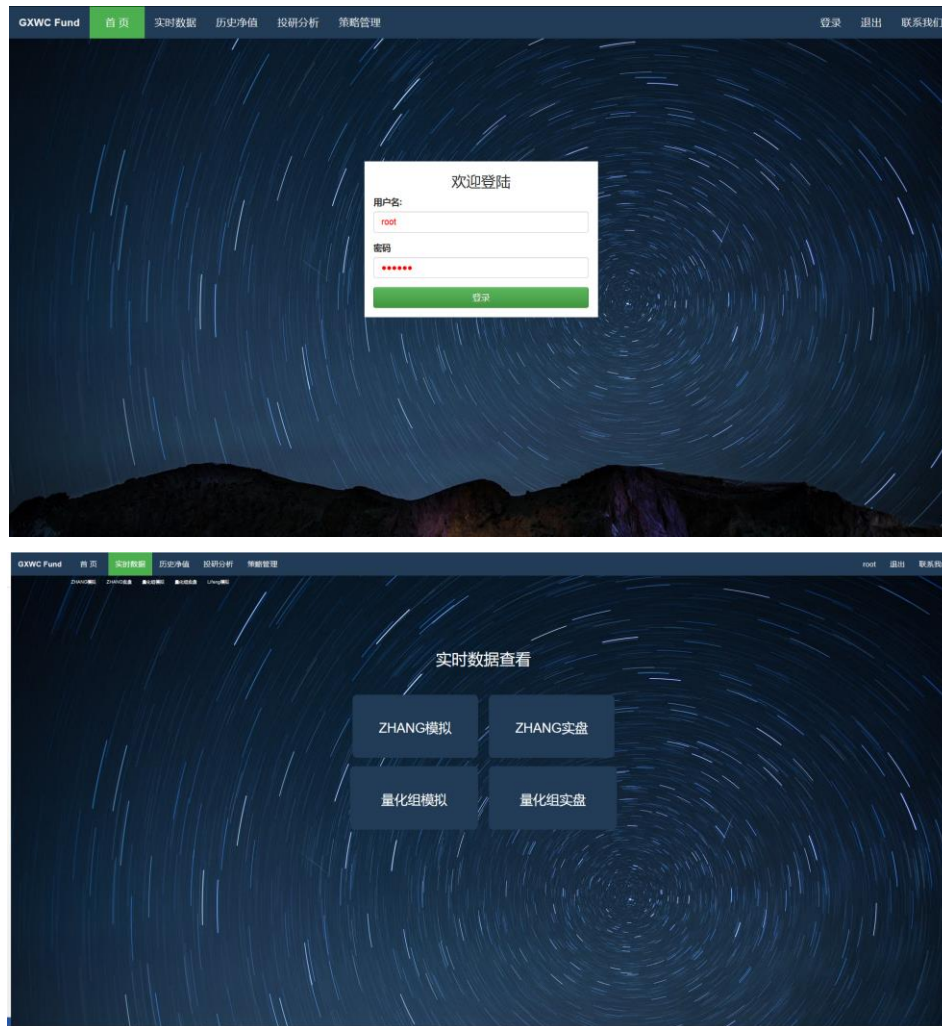
同样地，在“C:/Django/mysite/manage.py” 所在路径打开 cmd 命令行窗口，执行语句：**python manage.py runserver**

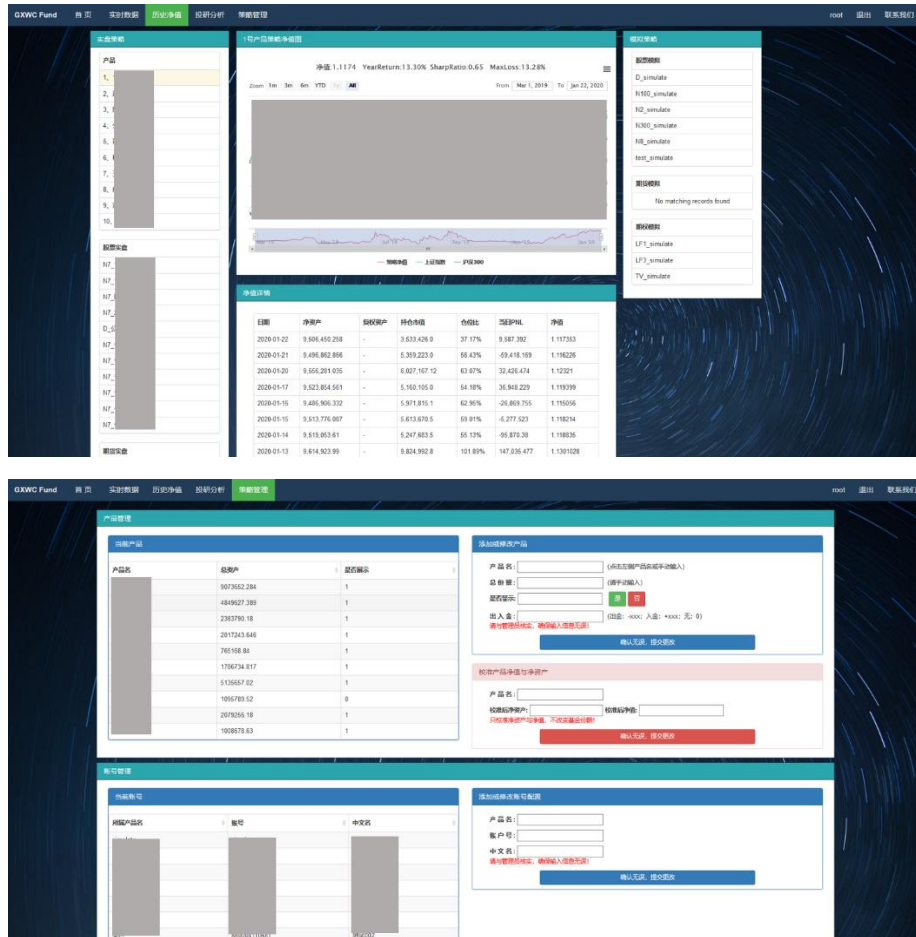
执行以上语句，默认服务启动在 localhost 127.0.0.1:8000/

在浏览器中输入地址 127.0.0.1:8000/login/，回车，即可看到如下画面。

如需在云服务器上部署 Django, 请与管理员核实要启动的端口号，需修改执行 **python manage.py runserver 0.0.0.0:50100**

注：50100 为服务的端口号。





十、其他

可能在新服务器上第一次部署 Django 的时候，需要给网页添加新用户，需在确认 Django 启动的前提下，点击 http://120.77.179.170:50100/add_user/（以 120.77.179.177 服务器为例），输入新用户名与密码即可，Django 会自动将新增的用户添加到数据库 user 表中。

此外，如果想查看任何一个网页 url 对应的前后端源码均可以在 `urls.py` 中找到对应的 `views` 视图响应函数，查看响应函数的 HTTP 响应，并在响应 HTML 模板中查看引入的 js 脚本。

网页管理密码：gxwc@admin

完！