基金管理系统说明书

**前言**

本管理系统基于python语言开发，使用的是开源的Django web开发框架。实现了一个简单的TA——基金管理系统。

django基于web的mvc架构，数据和前端实现分离，模块化高，耦合性低。集中于python语言，并且对sql语言有封装，只需少量html语言就可实现简单功能。

本文档分四个部分介绍：

第一部分是系统的主要功能和操作界面；

第二部分是每个功能详细介绍和使用说明；

第三部分是开发框架的详细介绍，包括每个功能的代码流程和整个逻辑架构，增加可维护性；

第四部分：此部分是独立的爬虫程序。独立于本管理系统，需要单独执行。

目录

[一、功能介绍 4](#_Toc522819836)

[1、主界面： 4](#_Toc522819837)

[2、主要功能： 4](#_Toc522819838)

[二、使用说明 5](#_Toc522819839)

[1、登陆 5](#_Toc522819840)

[2、用户管理 6](#_Toc522819841)

[3、投资账户表 8](#_Toc522819842)

[4、基金净值表 10](#_Toc522819843)

[5、账户变动表 10](#_Toc522819844)

[三、源码框架简介 11](#_Toc522819845)

[1、Django前端设计框架 11](#_Toc522819846)

[2、数据库 13](#_Toc522819847)

[四、爬虫程序 15](#_Toc522819848)

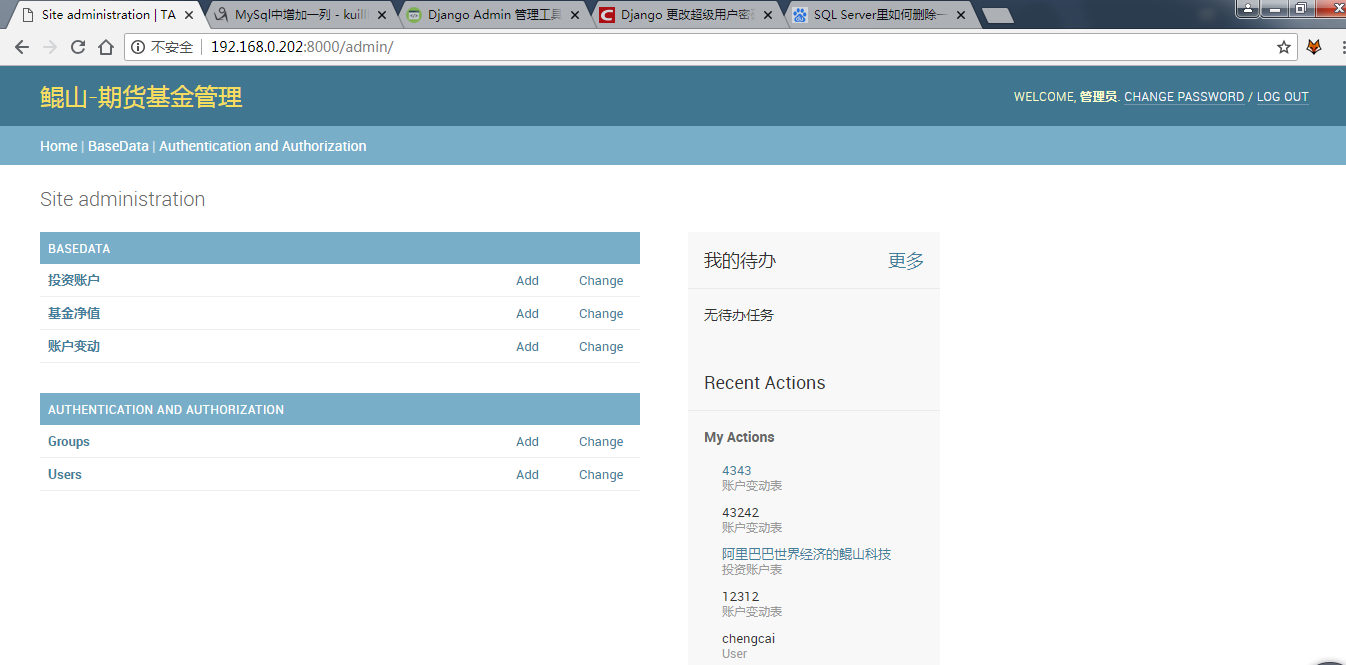
[1、程序介绍 15](#_Toc522819849)

[2、注意问题 15](#_Toc522819850)

[3、代码介绍 15](#_Toc522819851)

## 一、功能介绍

### 1、主界面：



### 2、主要功能：

（1）三个功能界面：投资账户表、基金净值表、账户变动表。读取数据库数据前端显示，并实现通过web前端增删改查写到mysql数据库；

（2）两个管理界面：group设置分组设置不同登陆者的权限，user是登录账号管理界面。实现用户账号的增删，权限设置；

（3）右侧recent action是最近操作的显示，可以查看最近十次的操作记录。确认最近的操作有没有成功。

## 二、使用说明

### 1、登陆

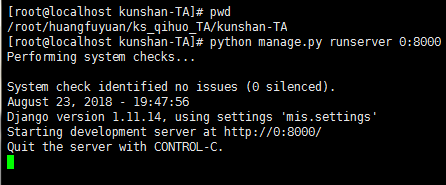
（1）首先在服务器端运行程序，无需编译。在源代码根目录下（manage.py的同级目录）：Python manage.py runserver 0.0.0.0:8000

开启服务器进程。

（2）在浏览器端登录：

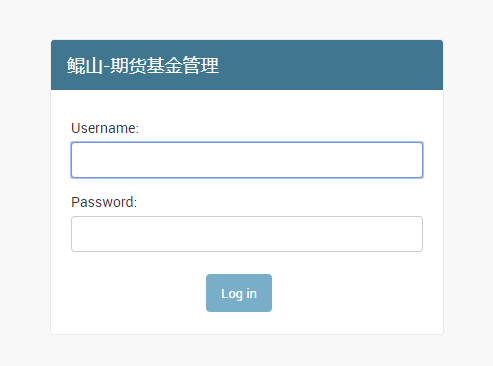
192.168.0.202：8000

即可



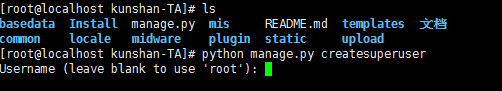
初始化阶段，在命令行界面设置超级用户作为管理员。拥有所有权限，使用管理员登录后可以增删新用户。本程序我已经初始化完成，会出现如下登录界面：

管理员：admin 密码：kunshan19



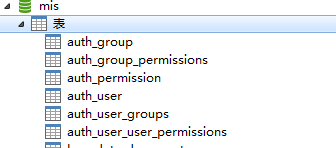
#### （1）Django管理员初始化方法：

在源代码根目录下的manage.py的同级目录下面，运行下图所示代码，并自己设置超级用户名字和密码。



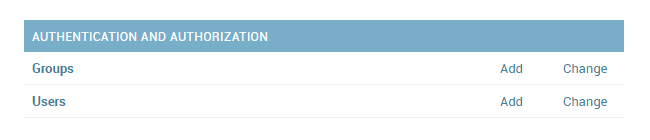
Web界面登录，成功登陆即管理员主界面。

#### （2）登录密码的保存。

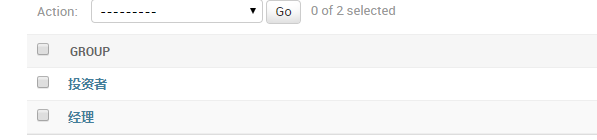


登录密码和账户信息都保存在了这几条库表里面，框架自带加密。我下面会写一个初始化数据库的mis.sql程序，里面包含对这几个库表的更新。若更新了这几个库表需要重新创建管理员账户和所有用户信息。我已初始化完成，无需初始化。

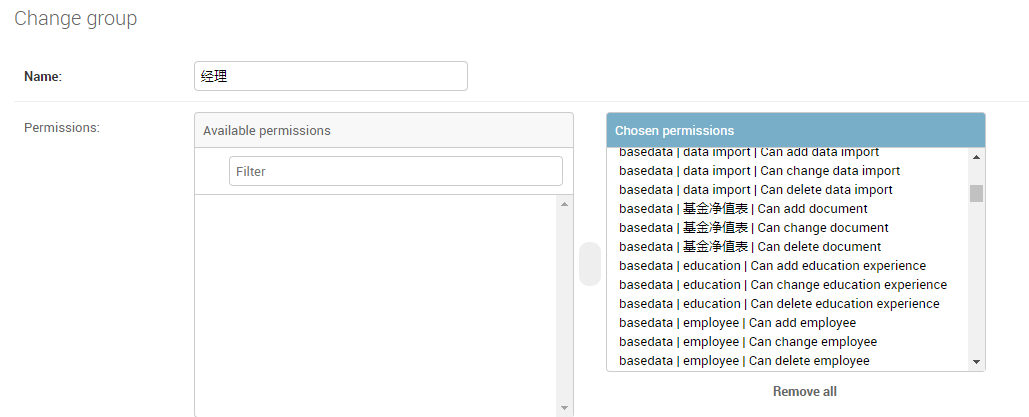
### 2、用户管理



1. group是用户类型管理，可以对不同登录用户设置一定的数据访问权限



经理拥有所有权限，投资者拥有部分权限。具体可以设置

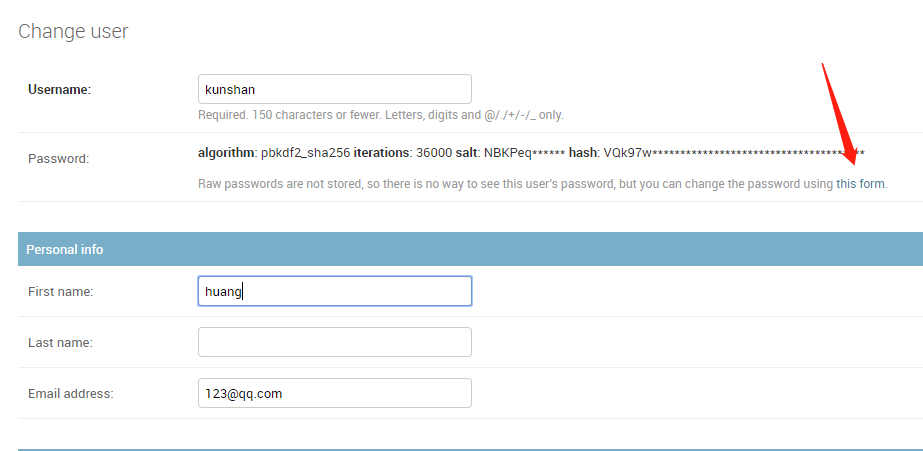


1. user用户管理

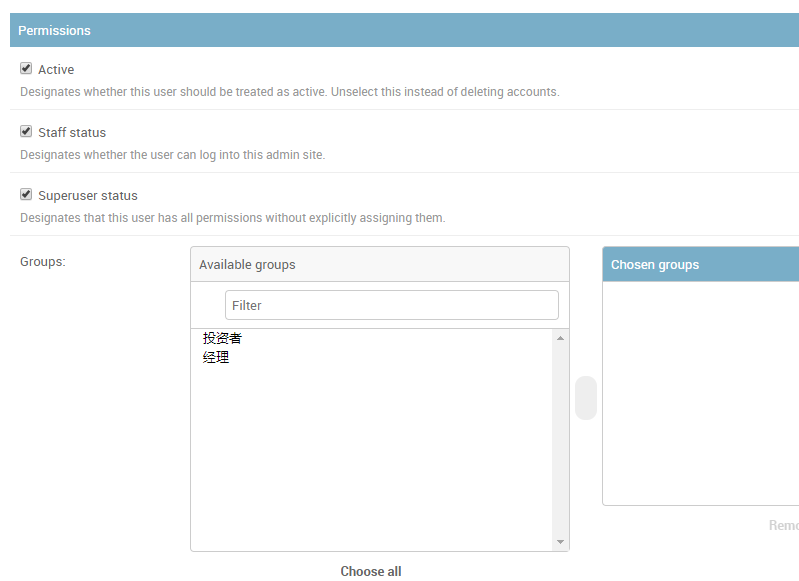
可以右上角增加用户，并设置用户信息和权限：



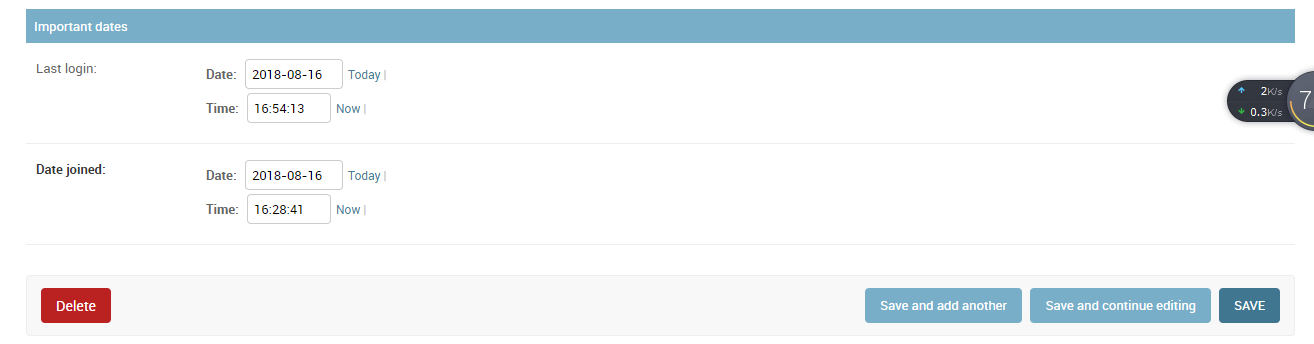
点击箭头所指位置可更改密码，密码是自带加密存储的。



设置用户权限：



可以save 、delete用户。也可以看到用户最近的登录时间：

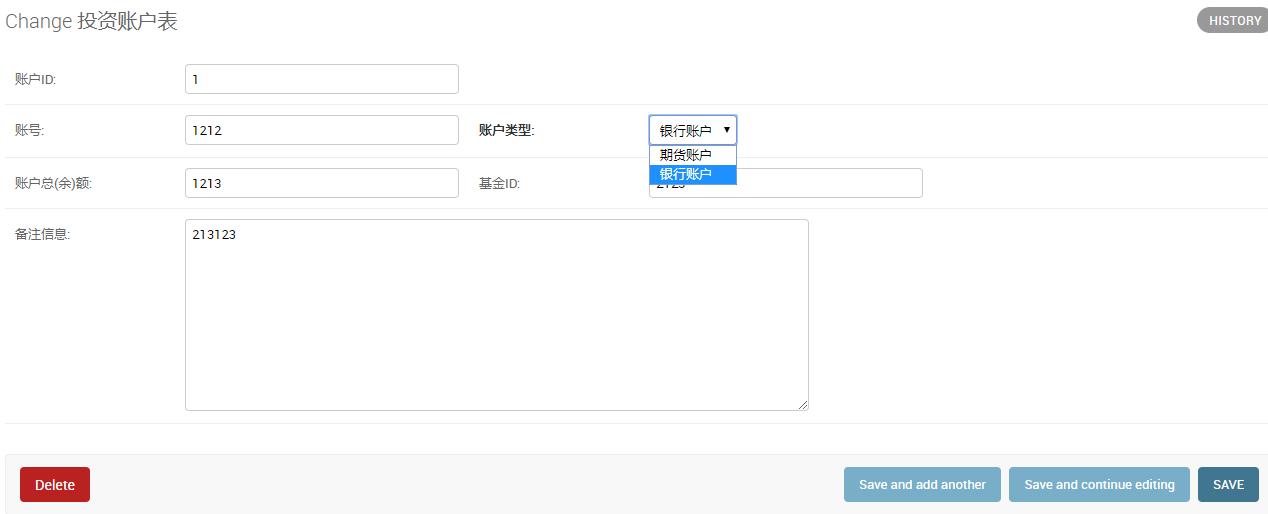


### 3、投资账户表

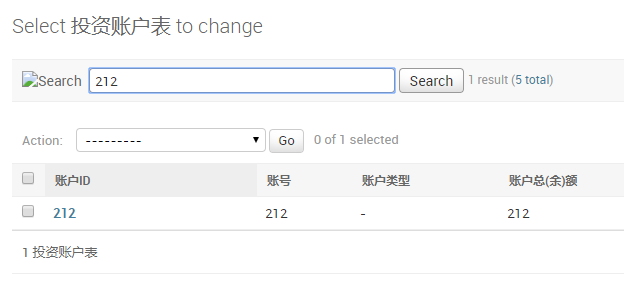
整个表按照“账户ID”的大小 从小到大排列



（1）账户ID：活动窗口，点击对应数据可进入修改界面.。也可删除对应数据，并从数据库删除，底层有封装了sql语言，保持数据同步。

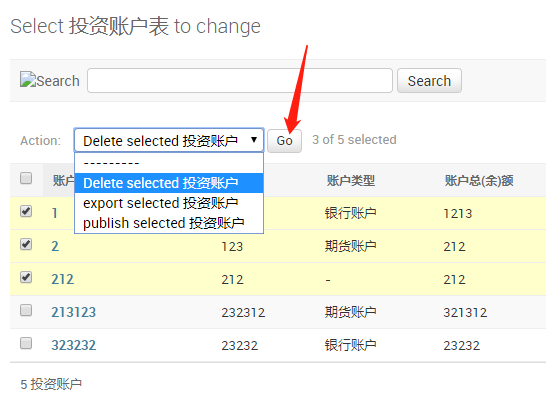


1. modification是最近提交的时间的信息。在数据库初始化阶段设置的字段。账户类型是个选择框，可以从代码里面增加其他选项。
2. search框：使用了“账户ID”字段作为关键字实现筛选，以212为例：



(5)批量删除

选中左侧方框，在action中选择delete然后点击箭头所指，即可删除对应的记录。



（6）将账户信息导出到excel表格

同理，选中方框选择action中的export，点击“go”即可自动下载导出到excel表格。



### 4、基金净值表



此表即时从期货基金平台上爬取的数据存储到数据库之后，web前端的显示。每条记录，没有显示所有爬取的数据，只显示我们关心的净值相关，具体详细数据可连接数据库查看。

若想手动增加调整某条数据也可以，设置了基金id为活动窗口，可单击进入修改。只实现了一张基金净值表，若多个基金可先数据库建立表，然后在basedata文件夹下的model.py和admin.py下增加类似的类和函数，不用设置前端。

### 5、账户变动表

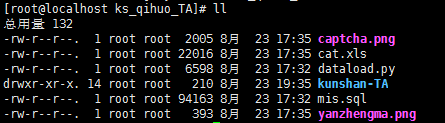


账号ID是活动窗口可点击进入修改，右上角同样也支持增加新的字段，会写入数据库。

同样支持期货账户表的所有功能。

## 源码框架简介

代码地址：/root/huangfuyuan/ks\_qihuo\_TA



Captcha.png和yanzhengma.png是验证码，cat.xls是爬的表格，每次都覆盖保存；

Mis.sql是数据库初始化程序；

Dataload.py是爬虫程序；

Kunshan-TA文件夹是基金管理平台程序。

软件介绍：

django1.11.14;

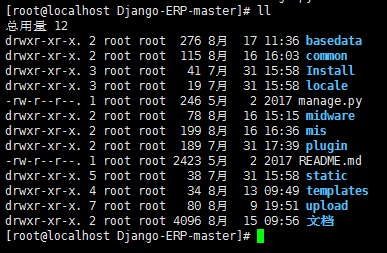
mysql 5.6

python 2.7

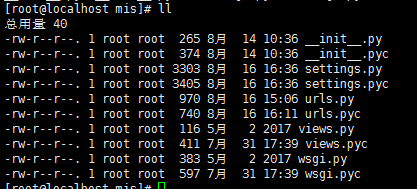
### 1、Django前端设计框架

Django是基于python语言的web框架，集成后台管理，前端显示。使用python和html、CSS就可以开发出一个管理系统，遵循web架构规则MVC，于此基础封装成MVT架构，网页的每个功能由APP组成，并且将SQL操作数据库语言封装成ORM接口。Model.py、view.py、admid.py是每个app的基础文件。在三个文件中分层处理，实现读取数据数据——>后台布局——>html前端界面显示.

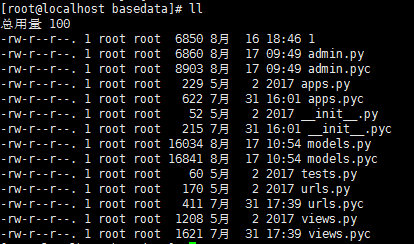
项目代码详解：



（1）mis文件夹：是程序的管理文件。包含整个项目的管理信息，主要是setting.py和urls.py文件，setting.py设置可连接ip、连接数据库、创建的功能APP（项目的各种功能模块）、前端html连接，HTTP中间件。url.py设置网站的网址跳转。\_init\_.py里面是网站的标题设置。



（2）basedata文件夹：一个APP，项目的一个功能模块（本项目只含此一个APP）。此文件夹是本项目的核心文件，



<1>model.py定义了三个主要的类。每个类对应数据库里面的一张表。Model.py的作用就是将数据库中的字段以django的orm接口提取出来。

* + - Expensecount类对应basedata\_expensecount表，此表用来实现账户变动；
    - project类对应basedata\_project表，此表用来实现期货账户；
    - document类对应basedata\_document表，此表用来实现基金净值；

还有其他的类用来是实现网站的其他功能。

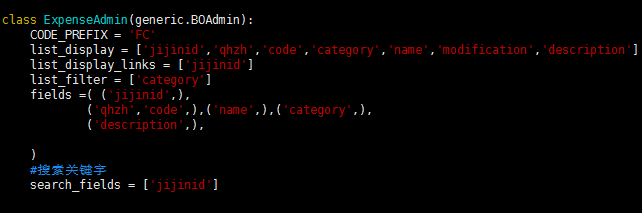
<2>admin.py是APP的后台管理文件。里面定义怎么将model.py里面的数据呈现在前端，每个图标文字的排列方式等等。将文件中的类和model.py里面的绑定，间接连接数据库。

admin.site.register(Document,DocumentAdmin)

admin.site.register(Project,ProjectAdmin)

admin.site.register(ExpenseAccount,ExpenseAdmin)

以expenseadmin举例：



List\_display是前端界面需要展示的字段，名称都是在model.py中定义的；

List\_display\_links是跳转按键，单击可以实现跳转到修改页面；

Fields是跳转到修改，添加页面需要填入的参数和布局，布局会映射到前端html，由于我们采用了统一的前端模板，只需在admin.py里面修改就好了，不用对html语言进行修改；

Search\_fields是搜索关键字，在search对话框里面可以进行搜索，“jijinid”是本表的搜索关键字。

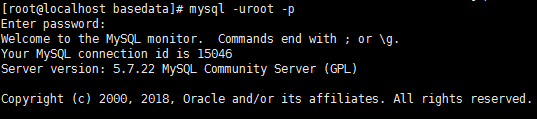
<3>view.py对数据库的操作。在界面定义的各种数据库操作功能，比如保存删除的实现，基于分离原理，这里没有html语言，会连接到html语言所在的文件夹（templates）

* install文件夹：数据库初始化代码，需在mysql环境中运行。对数据库有部分更新，或者添加新的功能app，要迁移到django，运行migrate。http://www.runoob.com/django/django-model.html
* manage.py是一个控制文件，运行django的命令需要带上。
* static、template是前端html和css渲染文件，在里面可以对前端界面设计，将admin.py里面定义的变量呈现在界面上。
* local文件夹是django的“国际化”文件，里面存储了各国语言的翻译包，我们没有使用。
* common、midware是django的框架中间件。进行网站的request请求，让我们免于设计http的一些操作。也会负责网站的一些安全，比如阻挡伪基站请求等，只需加上自带的中间件就可以。当然也可以自己写中间件。其实现原理比较复杂，好在django都封装好了，更多细节：[链接](https://blog.csdn.net/a289237642/article/details/81213067)

### 2、数据库

软件信息：mysql 5.7

登陆方法如下，密码是kunshan19



使用mysql数据库使用mis.sql初始化数据库。django连接数据库，安装pymysql，作为python连接mysql的驱动。

数据库的使用：

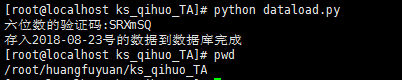
(1)基金净值表用来读取数据库中的内容。基金净值表对应的是mis数据库里面的basedata\_document表，此表的数据是用dataload.py从期货中心的网站上爬取下来的。基金净值表只是用来显示里面的部分数据，比如权益，净值，其中净值是在dataload.py中利用sql语言计算好存入的；

(2)投资账户表是对mis数据库中的basedata\_project表进行增删改查；

(3)账户变动表是对basedata\_expensecount表进行增删改查。

## 爬虫程序

爬虫程序是dataload.py文件，需要手动去执行，使用python2编写。



### 1、程序介绍

程序的主要作用是爬取中国期货中心的个人账户数据，并按照一定格式保存到数据库。

首先，期货中心的网站需要登录，并含有图片验证码。所以，我们使用程序模拟登陆并保持登录时会涉及到验证码和cookie问题。其次，我们登录时跳转到今日的结算数据，对于往日的数据他是一个对话框的形式出现，没有实现爬取往日数据。因为，我们只需每天爬取当日数据就可以了，往日数据可以在第一次使用时手动输入一下，不需要每天爬取往日数据。

### 2、注意问题

#### （1）图片验证码的识别

验证码识别属于OCR字母识别，现在主要采用深度学习方法进行识别。时间能力关系，使用谷歌的tesseract没有训练出高准确率的模型。使用的是腾讯开源免费的ocr接口，我们只需下载验证码到本地然后上传腾讯云，就会返回识别出的验证码字母，准确率约有90%。所以在手动爬取数据的时候，若报错很正常，那就重新运行一下脚本。

#### （2）数据的修改

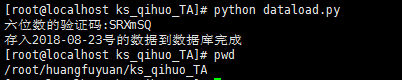
若某日忘记爬取，或者重复爬取，可以在web界面进行修改，会自动写入数据库。方便快捷。

### 3、代码介绍

整个程序由大部分组成。验证码识别部分、登录保持爬取数据部分和数据库计算存储三部分。

#### （1）验证码识别

上述细节。只要出现如下验证码个数识别正确，百分之90就可以识别出来，也存在识别错的。

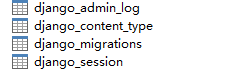


可能会出现的错误有：网站阻塞没有反应过来这是没有办法的事情，在运行一次。

#### （2）登录保持

使用的是django自带的session保持，只要每次登陆携带token就可以实现登陆，不必每次都要输入密码。

登陆信息都会存在数据库中：



#### （3）数值计算及数据库保存

Dataload.py里面的ks\_save()函数实现读取数据，并进行净值计算存入数据库，代码里面有详细介绍。