## Redis的用处

1. 存储cookie
2. Pub/sub数据，用于不同服务进程间的数据同步。
3. 键过期通知。用于实现用户过期自动下线的日志记录。

## 值得说的内容

### Redis

在pub、sub的时候，需要对cluster的每个节点都创建一个listener线程来监听，因为它在自动删除key的时候是其中一个server发出的，别的server是不会发出的，并且只会发送一次。如果不全部监听就会漏掉。

### Postgresql

时区的坑。由于数据库时间区域的不同，在执行timestamp计算的时候导致了8个小时的偏差。最好的办法就是前后端用long传递，数据库用有时区的时间戳来存储。

### Ws

居然还需要设置cacerts

需要打开网页，然后利用浏览器获取该网页的证书文件.cer文件，然后把该文件的内容导入到.cacerts中，确认密码：changeit，然后重启电脑就可以了。

## SSO

从系统A点击按钮进入系统B的界面，随后，点击系统B的登录按钮，系统B会来调用我们的接口，我们的接口此时要做判断，如果用户没有登录，就跳转到登录界面，用户登陆成功后，会利用username,email,timestamp(10位),secret（这个是系统B给A的一个秘钥，只有双方知道）加密出一个hashcode，然后，调用B的登录接口并把username，email，timestamp以及hashcode一起发送过去。B受到数据后会提取B中记录的A的secret并与传来的uername，email,timestamp一起加密，得出一个hashcode，再与传来的hashcode做对比，如果正确，则允许登录。

### 实现方案

**下面以已经存在子系统A，B并尽量减少修改两个系统代码的情况下，考虑给两个系统增加SSO机制，由于两者已经有了各自的登录逻辑和认证逻辑，因此比较优的考虑就是增加认证服务器C。下面说明方法：**

### 方案1

系统A和B在各自的登录成功逻辑最后，增加将认证数据交给C，C将认证数据保存。随后，用户从A链接到B的时候，

有A，B，C三个（子）系统，实现SSO的方案：

情景1 ：虽然有三个系统，但是B和C需要从A出发，由A进入到B和C的界面，而B和C拥有用户在未登录下也可以看的公共页面，也有只有登录后才能看的页面，所以用户可以点击BC中的“登录”的按钮，但是如果用户已经登录了，就可以免用户名密码登录。

实现逻辑1：只有A具备认证逻辑，而B和C则是依赖A，B和A各自存储一个secret，C和A各自存储一个secret。由此，客户端对B和C的认证请求就会被B和C重定向至A的一个sso接口，A依据自己的判断逻辑来判断用户是否是处于登录状态（从cookie或者header或者请求参数中提取ticket），并依据重定向来源来判断是从哪个子系统重定向过来的，从而获取对应的secret。如果用户没有登录，则进入登录界面实现登录，如果已经登录成功或登录成功后，则用secret和用户名以及别的某些参数作为加密元素，做加密运算，生成一个运算结果，再重定向回来源者B或C的sso，同时将加密结果和加密元素（secret除外）一同发回，接下来B或C再利用存储的secret与传来的加密元素做与A相同的加密运算，将计算出的加密结果跟传来的加密结果做对比，如果正常，则认证成功，否则失败。

Secret的目的是

## 买卖逻辑

### 限价



1. 限价就是按照我指定的价格和数量去成交，会出现多个部分成交来完成一整个数量成交的情况。
2. 百分比的意思是指从我的钱包中直接利用百分比来提取数量，不用我直接自己手写了。

在买的时候是计算总价，卖的时候是计算数量。

1. 有效时间：



IOC的意思就是下单后，能成交的部分立马成交，不能成交的部分立马退回。

### 市价



百分比就是从钱包中按百分比来填充该币种的数量。



止损的情况下：

1. 买的时候的触发价格是指，我在下单时候指定了我要花的总钱数（也就是总价），然后会进入市场进行交易，交易一部分后，如果在此期间市场的价格是一路飙升的，那么在达到我的触发价格后，为了及时止损，就会用这个价格把我剩下的钱全部花掉。
2. 卖的百分比则就是按钱包中的数量乘以百分比把数额自动填写在这里。

在止盈中则就好理解了：

1. 买的时候，如果市场价格在一路下降，达到这个触发价格的时候，我就可以把剩下的总价全部花掉。

## 指数



1. 百分比，以买为例，这个百分比指的是比市场的价格高或者低一定百分比价格的价格，比如我写了10%，意思就是比市场价高出百分之10的价格我可以接受，可以去成交。那么就会出现高于10%后，价格实在是太高了，成交的话我会很亏的情况，所以又有了最高价格。
2. 隐形，所指定的百分比的意思是，每次抛出在市场上的数量或者价格。比如，在买的时候，这个隐形设置为10%，而我要买10个，那么点击下单后，系统会抛出以我指定的价格去买一个的订单，当成交后，再抛出一个，直至10次全部完毕。