Solace topic structures用来在app之间传送消息

Outlook给邮件分类

Office 365里面用word

熟悉linux, ip3的操作界面，把需要的东西都准备好

Vim的基本操作？这个不需要了把

Linux基本的system call

First week: 熟悉系统，空闲时间可以看操作系统的理论

花两个小时看一下git，书本和wiki上的信息

<https://bitbucket/projects/BNK/repos/bunker_apps/browse/bunker_apps/load_bbg_closing_prices.py>

<https://bitbucket/projects/BNK/repos/bunker_utils/browse/bunker_utils/bbg_utils.py>

i’m going to have your first project be a system that gets bloomberg data and puts it into bunker

[4:43](https://akunacapital.slack.com/archives/D011FKKNFGF/p1587591831062000)

right now, we do it where we get the data and load it into a database, but we’d like to change it in a number of ways

[4:44](https://akunacapital.slack.com/archives/D011FKKNFGF/p1587591859062400)

i’d like to be able to download the data and then put it onto solace (which is our firm-wide message bus)

[4:44](https://akunacapital.slack.com/archives/D011FKKNFGF/p1587591882063000)

this allows for hubble, our data infrastructure platform, to see the data and store it and people can replay the data and do what they want with it

[4:45](https://akunacapital.slack.com/archives/D011FKKNFGF/p1587591902063500)

then i’d like you to write a listener that listens to the solace data and writes it to the bunker database

[4:45](https://akunacapital.slack.com/archives/D011FKKNFGF/p1587591933064200)

and also, i’d like the system to be scalable so that if people want to get data they can start doing so, with approval because data pulls cost money

you should also note that we have a firewall which might prevent you from running this on your machine so that might be something we have to ask for NOC to poke a hole in the firewall for

原先是直接load到bunker\_db的，现在希望我弄到solace上面，让别人也可以接收到（这个部分我似乎不用管）。然后也还是写一个listener，接收信息，储存到bunker database上面。同时还要scalable。

bunker\_db = akdb.DBInfo(\*\*config["db"][environment]["bunker"])

from send\_bbg\_prices.py

Question to ask:

right now, the existing program gets the data and load it into the bunker database, which is used by the bunker team. From my understanding, now you also want to put the data onto solace. So why do I need to first put the data onto solace and then load it into the bunker database? The existing program already did the loading job.

Main task: (1) put the data onto solace (2) scalable

If so, the main tech difficulty for me is to get familiar with csolace package api.

You said hubble is a data infrastructure platform, but what’s hubble in specific? I didn’t find a wiki page introducing it. It seems I don’t need to worry about it for this project.

What do you mean by scalable?

Order: I should do some tests to make sure I get the right idea before start writing LWP

LWP question (how long I should write it)

Can I test and run the existing code on my machine? Or there is a database for test?

Quite a few tools that I need time to get familiar with, it’s better to have some buddy to ask around. Can I also ask ding jin for tech questions?

Scalable: 相当于分成两步走。其中一步是：

1. Download data from bloomberg

2. Convert bbg data to protocol message

3. push protocol message to solace

4. write data to database（这个是另外一步）

Tech difficulty:

Conda package issues: almost solved

Csolace api

Asyncio

aiob

Convert data to protocol message: almost solved

DB setup: not quite sure how to do it on i3wm. solved

Situate: not sure what this is

Invoke

Styling: there are four

Python memory view

Python logging

Solace里面，request和reply分别指什么？

Should be able to finish some project content part today or tomorrow

Question for Ding:

How to set up datagrip on i3wm? Need to buy a license now?

How many database do I need to use? (like akdb?)

Can I run and play around the code on my server? Does it cost money? What is a test DB?

Where is the test data?

Do I need to make a new git branch? Pr: pull request

What’s situate: automation

Ans: better use my own user account

Run code用-dev flag

Question

How to listen from Solace? How to send to Solace…

(what is a topic)

Listen, get, send

Get, subscribe, send

下面这个是用neosolace做的send，可以参考一下。数据库源不用管

https://bitbucket/projects/BNK/repos/bunker\_apps/browse/bunker\_apps/create\_otc\_grains\_options.py

Is the process like: load the database schema, download the .encode file from bbg, decode it

Meaning of ` Transferring symbol data from devbunkerdb.bunkerdevdb to /mnt/bunker/dev/bloomberg/20191120/200429103609bnkr\_eod.req…`

` Bloomberg was not able to interpret: CSZ9 Comdty`

Ans: .encode是自己写的

Invoke check报错是不是不用管

Akuna asyncio Database Library（aiob）我需要用么，和akdb区别：

Aiob: based on different library asyncio

一般用datagrip就是为了看看数据？

Estimation of time (just for reference)

What is photon

Instrument query database is another one?

Question

How to make one keep running

What’s Kunlun usage?

Project:

(0) I currently have finished the project plan and now I am doing high level designs for the library methods I will use, the database I will use and the protocol message (AIDL) part. I am running examples from the csolace/aiodb library and existing closing price script. In the previous week I had two meeting sessions with Ding to know about the system and I also get familiar with python asynchronous I/O part. So, the next steps I am going to do is what I will do for the high-level design part. As long as I test the example scripts and understand what apis I should use, the project seems straightforward if I am correct. And the running server deployment will be in the later part of the project so I am not going to worry about that now and I will ask Ding for help if needed.

(a) data fetch needs: only closing data or other data. I think it’s better to finish the closing price one first and then extend to other demands. The system we design is scalable so should be easier to extend to other utilities.

(b) run and get the data: -e dev -d 20191120 -r US. This works.

How long do you think is appropriate for the project?

UML写哪儿？应该是在system boundary diagram这边写

然后system boundary diagram不用写的很准确，但是要搞清楚和已经有的东西（比如Solace, db）的互动

Design里面参考M2的Sequence diagram and message

M2: 看Sequence diagram and message

Overall: BBGService在solace的一端，bunker在另一个端。就是两个service，不一样的名字。Duplicated data row?

Main task/difficulty: to write protocol message

如果说user是通过bunker来send request，那么我该怎么写bunker？怎么take input？take什么input？实际上明确了take什么input也就基本明确了我的message protocol要写什么？

应该使用async publisher和async request reply (subscribe and reply)？async request reply就是async RPC，在distributed system里面有

剩下的是异步问题吧：收到request后要处理ack和packet，这两步应该是异步的？

我做的实际上就是类似remote procedure call，中间多加了一层

返回的data一行可以先保存为string，之后再parse成row

我要做的应该是two async RPC (p135)吧。如果不需要最后ack的，那就一个send

What’s your expectation on me?

Where can I do better?

Question:

Does this project work on all BBG data besides closing data?

How do users input their data request to Bunker? By which means do they input their request? Like directly running bunker with some arguments or through solace?

Which Topic should I subscribe to for getting request from users?

Ans:

Receive data request in Akuna format. Hide the bbg terminology (use translater in bbg\_utilities) to transfer. Request takes about 6 minutes from bbg, so do it in batch. (like using map? A list of fields and instruments) 哦是说要查不止一个东西

Not sequence number, maybe some hash value.

Key point I think: simple and fast service crash recovery. Way to do it: stateless

Future price not option price

Get\_request

Get future price. 先做这个

问题：.req和.out有什么区别？

Ding的建议，先从bbg这边，对应trader的要求，看看哪些数据是可以得到的。以及是否可以convert成bbg request文件。

Q:

Does Solace provide request/send mode or subscribe/publish mode or both? Ideally should be both and I need to use both to work because I need to serve one request from user but broadcast to others who are interested in it as well.

Who should subscribe? Only the request user? Then we could use request\_id like a port number. If this port number is in use, reply to the user that choose a different one.

1. Is there a user who first sends to the request to bunker via Solace? Does the bunker need to answer to the user with the fetched data or Hubble will do this part and the bunker only needs to write the data to database? We have closing price database, but traders are asking for data from bbg besides closing price, do we already have database for these data?
2. If the request\_id is given by users, will there be duplicate request\_ids? How do I handle this?
3. There are publish/subscribe example, and also request/reply example. It seems I should use both. I don’t really understand request/reply one (there is also subscribe in it).

Bbg service只是一个bridge，它就是负责把数据中转到bbg

Situate launches bunker

My task: try log into Situate and see what’s happening

Server means bbg service

PJ:  
builder (director) 是什么design pattern: Builder Pattern

用subscribe还是get？difference是啥

Send to Hubble team the last two row\_num, info. And maybe enumerate the error.

看一看builder pattern

跑一下shift strikes

试一下加asyncio.event或者future来等待并且取值：event是ok的

问：为什么要用coroutine…?

Ans：在我的程序里面，我可能会await自己的函数，但最终我await的其实是csolace的方法，而这个东西已经是用non-block的方法写了，就保证了这些message传递是异步非阻塞的。对我使用的人来说，应该可以就当它是没有await的普通函数来看待。参见flags\_asyncio.py以及之后的说明。

似乎await或者说yield from要有事件循环才有意义。

问：那如果我的await是需要返回值的呢？

Ans：

现在在处理的问题是：怎么让我的service在持续subscribe。Solved: use event

什么是S法则，我是不是应该为create task和subRequest另外写一个class？

我是否要确保在我发结果过去之前，对方要先subscribe？（这个暂时可以不用考虑）

Do I need to worry about the case where subscriber is not connected (can solved by using a queue)?

是不是把topic作为一个变量分开来写，这样以后要改的话方便些。Ans:暂时看没需要

一个是写event，还有测试一下await的话会不会消息消失（不会）

要把call\_back里面的东西拿出来

you’re going to want to wait to listen to the result\_info, look at the number of rows and then wait for all of the rows to come in. probably want to use an Event synchronization primitive.

下一步：Typing, git hooks

Conda Package conflicts: solved?

第一件事：怎么改pycharm的conda环境和git branch: solved

第二件事：remote和local的branch不统一，怎么改: solved

第三件事：回到typing

关键是我还没写的怎么给参数占坑

以及是否调整一下考虑整体调用问题 solved

先还是看一下git? solved

顺便可以看下一个新的pj folder怎么创建的 Ans: cookiecutter

Git log要看下，才能知道指针怎么指的 solved

在load\_bbg那边考虑怎么给参数的问题：应该通过类似旧code里面从db获取的方式，只不过我要用aiodb而已

Dependency injection:

<https://stackoverflow.com/questions/130794/what-is-dependency-injection?page=1&tab=votes#tab-top>

找个design pattern的书看一下

正好看一下photon是什么，这是其一。其二，学习一下它的structure，然后我就可以改自己的了。

Partly solved. \_\_main\_\_.py参考shift\_strikes.py。

[用@click.group()](mailto:用@click.group())来装饰即可https://click.palletsprojects.com/en/7.x/quickstart/

RequestMsgListener应该是一个abstract base class，在service和db里面可能都有它具体的实现

第一步：看看handler或者listener是是什么design pattern，还是纯粹就是一个interface

Server继承handler

看起来我这边对应的话是让LoadBBGClosingPrices和SvcUtilApp继承？

先把db.py理清楚再动server

我猜测这个design的好处是让server的使用者自己去定义需要的callback\_func，如果再写一个不同的server（和不一样的callback\_func），就不需要改client code？比如我有server和loadbbg两个。对subscriber来说，它可以变得更generic，至少在它的code里面是和一个interface（msgHandler）打交道而不是和具体的server打交道。不过本来不也就是传一个general的handler么…

Topic builder是不是也要放进去？

但怎么解决有些subscription需要request\_id，有些不需要？注意，对于subscribe而言，它应该只关心两个东西，一个是handler，一个是command，而有没有request\_id应该是对于command/topic而言的，所以这应该是user给我的，所以应该是作为command/topic的constructor传进来的

Topicbuilder必须要拿出来单独写，不然的话typing不清楚，而且topic可能有很多种，我必须要让它单独有一个class，使得之后添加新topic可以变得容易，同时又不需要改变client code（即subscribe那边的参数）

如果我创建的topicbuilder instance太多的话，会不会占用太多内存？

但为什么要set。。。因为我wait的只是subscribe，不是收到消息

要不把topic builder拆成topicbuilder和topic fields两个cls？Bridge design pattern。不过实际上一样是让client code在使用的时候再决定给abstraction配什么implementation作初始化（现在我是直接把可能的几种implementation写进去了）。如果我的topic不会变得太多的话，没有必要弄得这么麻烦。

那我这边db里面实际得到的就是一行数据，所以我要做的事情一个是变换成closing price table的格式，参见transformer。同时还要做一个计算，也在transformer里面，这个计算就要求把前一天的data拿出来。最后做完以后要自己改写一下loader，用aiodb写进去即可。

Bbg\_symbol之类的东西看来都得靠我自己从db拿出来

如果说我要做yesterday的data的计算的话，那也得让db.py来做

这个ClosingPricesTransformer我要拆成两部分。。一部分是给service.py，一部分是给db.py

稍微看一下taxi scheduling

稍微看一下db

关于trader要的那些东西是什么，我是否能得到，我觉得暂时问题不大，我先把closing price的能写出来，然后之后再试一试。

我觉得自己不该投入过多的精力在内部doc/architecture上面，因为太vague，除非我碰到了，用上了。而是掌握general技术。这个的意思是说，我需要别人的方向性指导，但也应该凭借这个提示找到更多可以refer的资源。

Path\_factory是用了factory design pattern？

坑死了。。。akdb的get\_rows是一个generator，aiodb的返回的就是一个DBRows class

Service\_request能接受的是一个bbg\_instrument msg类型的东西，并不接受DBrows类型的。

Aidl里面给如expiry\_dt之类的设置默认值：本来aidl就有默认值，不需要。不过fixedpoint类型是不是应该改成float

还有一个就是run\_in\_executor的问题

Queryextractor改adaptor，还是改dbreader？

Sql query里面product type为什么只有s or f，不能做option么？是不是因为option没有closing price？或者akuna只做stock和future?

注：S是stock，F是future，C是call option，P是put option. WC和WP是对应HK的

能否试一下除了s和f之外的？

UnderliersQuery也要改，要把product和product type传进去

Aiodb里面PostgresClient和connection有啥区别？是否抽象类要initiate？似乎queryextractor目前就够用了。Ans:一个是异步的，一个不是

没理解：str.format(self = self)。就是位置引用。solved

To see: str.format方法，sql join, group by。solved

…我昨天看了半天sql的问题，原来原因只是因为它只显示500行搜索的结果！

是不是画个class diagram？

Error处理似乎没有做（完整），要见class diagram

为什么要create\_task？有些coroutine似乎没有被包在task里面？

两个问题：run\_in\_executor是不是一定要包在task里面？第二个，它是不是也要像管道一样一层层来？

它还不是thread safe的？？？

注意所有的static method，如果确认的话，要把self都删了

1 on 1:

What’s your evaluation on my performance? Where do you think I can improve?

是不是自己试一下ssh add，至少让我不用每次输密码？

看一下实际的key pair是保存在哪里

看起来目前位置我连到公司电脑的ssh并没有用key？Ans:有可能现在用的是纯密码模式。

一定要参考key based的话，大概可以看一下github

要不要试一下和uchicago的机器建立public key auth?同时让我不要输入passphrase. Solved

问他：一个是key放在哪儿，然后怎么能都用它

还有migrate是直接创建新的在Bunker\_apps，旧的要动么

怎么release

我要改的是ABN文件夹下面的ABNJounal里面的task包里面的common文件里面的ABNArchiveProcess，把password删了，把pkey\_path加上去即可。这个是针对load\_abn\_journal\_task

还是先要解决验证的问题

<https://stackoverflow.com/questions/49587932/i-dont-have-ssh-config-file-but-multiple-github-accounts-works-how>

这是答案

把配置文件设置一下，（看一下sftp操作，optional），（git验证方式，optional）. Solved

问题有二：一个是究竟要下载到哪里去，第二个是看一下path factory和process

卧槽，突然意识到这是两个不一样的，注意username是不同的

我就是不知道上传上去的那个locatesin.dat去哪儿了…找不到

原来的staticmethod都要改掉。。。

怎么保证bunker\_apps运行的时候用的是最新的版本？Ans: conda自动安装的是最新版本。那么要用conda update

我猜测的是你上传一个文件到那个位置，它就会返回给你今天的情况

写email。。。

那个发送的邮件，实际上就是把内容在邮件body里面展示一下，然后再把内容用附件发送过去

等会儿可以看下multi…是神马

我对python的module该怎么引用并不十分清楚….

如果pull request被approve了，就可以去merge了。然后就可以invoke release version\_number，这样其它地方就可以用这个package了

先让我测试一下load吧。我要做的事情如下：第一步是自己放一个待下载的文件上去，第二步是创建一个abn的运行环境（conda），然后把代码放过去，然后尝试run一下，记得改一下邮箱名字

再测试一下upload

要解决的几个问题：

问题：假设我做了pull request之后又有更新了，那该怎么弄？是不是说如果别人更新的是非我更新的那块就没事？有点迷茫…之后再看下git

Docker暂时和我应该无关

Linux: export命令

Prod和dev有啥区别？Ans: dev是测试用的，所以跑出来的output也是放在remote里面对应的dev文件夹下（即把prod换成dev）

补充：

第一点：所有重要的数据都是放在共享的网盘上面的，即例如'/mnt/bunker/dev/abn\_files/20200610/'，所以和用哪台机器测试应该关系不大？

'/mnt/bunker/dev/bloomberg/20200611/20200611\_bnkr\_eod.out.gz.encoded'这个是load bbg里面的例子，一样的道理

所以我写的那个load\_abn\_locates其实应该是把dev(test)和prd分开来。。。不过它本来也没分开，都堆在/mnt/ndrive下面

第二点：至少bunker每个entry\_point都有dev和prd两个选项，这就导致了数据放的位置不同？

还有一个问题：我是否可以打tag测试。。不要每次都PR然后release了才能测试

<https://setuptools.readthedocs.io/en/latest/> python打包问题，可以看看

下午把组里的代码稍微浏览一下。也看不出太多名堂…

portfolio\_bridge里面cli.py担任了main.py的任务，就是entry point，里面可以让user传config file。还有一个kill是用linux命令来终止进程。这个service是运行在docker上面的

position\_bridge\_3也差不多

我想把entry也写进去。。不然entry那边太乱了。但其实还好，现在的diagram就是从两个真正开始的地方画起的

\_\_init\_\_ cannot be async…

等下：我写的这个东西，比如self.result里面储藏结果，会不会造成多个process同时在访问这个变量？哦这边应该没事，因为我本来就是想让它们同时访问，因为它是要subscribe多个topic的，而其实这些topic都是同一次请求的结果的分发topic

问题：是该先写test，还是先跑？我觉得一样

Collection[int]和tuple[int]难道不是包含关系？不是应该…

有两件事：第一个是看一下abc，第二个是试一下click

Click简言之就是一样从命令行call函数，只是参数是在用户输入时给定的。

解决问题：为什么from . import 不对

注意：from pktestttt import bb就可以…

记录：config在db里面只有一个地方用，就是得到solaceinfo；在service里面有两个，一个也是得到solaceinfo，另一个是得到bbg file header

Aidl的问题：把expiry\_dt换成string，fixedpoint换成float

我要把logger都加上去…这个还是很有用的

Run\_in\_executor会block？还是我用的不对？

要学习怎么写test

Logging模块就不甩attribute error了？

最后在transform里面把这些数据再还原

现在的版本并不general，应该要fetched\_bbg\_result: List[Tuple[Tuple[str, str, dt.date, float, str, str], Dict[str, str]]]

List问题，extend之后bytes就不对了

核心之一是说，当我的service开始subscribe一个topic并且提供一个callback func后，只要我一直subscribe着，当有request出现在那个topic的时候，我的callback func就会帮我抓取下来。这个我不用关心的api里面用的是机制是while loop？

使用fixture的好处之一，是在写测试时候只需要考虑核心逻辑，而不用考虑准备工作

即Given, when, then. Fixture就是given（前提）的部分

今天想做的事情：把test写一些，然后再改code？还是等到Mike明天一起看完再改？我觉得不能等，还是先不改review，把新功能实现？

附录D看Python包应该长什么样？

我是先改一些code再写test还是怎么样？先写一些test把。比如test\_main肯定可以写

能不能对datagram写一个更详细的type类型？把None或者””之类的东西梳理清楚

我觉得先把test写了肯定没关系，到时候大不了改一改名字之类的，大的test框架不会变

目前核心还是想多一些参数和例子，让test更general一些

Patch的核心就是我封装一些接口，测试的是当下我想测试的逻辑

如果我不mock一个对象，但它的初始化用了mock，那么我call它正常的方法，它会返回什么？

mock\_extract: MagicMock = mocker.patch.object(

closing\_prices\_transformer.extractor, "extract", return\_value=None

)

这玩意太坑爹了，一定是要改那个class的member的method，你改那个member本来的class (Query\_Extractor)的method是没用的

先把load\_bbg\_closing\_price.py测一下，反正是复制过来的，如果有现成testcase. Done

然后可以学photon那边对db里面的aenter和aexit也写个测试

然后把端口都对general cases进行扩展，db这边不用动，就是本来的样子

然后再给service写测试

做一个design pattern把这个转不转成None的东西剥离出来

我觉得test的核心是，对于中间逻辑，测试的就是每个函数是不是被正常的按照计划call了，最后测到底层业务函数时，再去测那个数据是不是准确处理了？

Topic builder有一个违反dependency injection的问题（？）

我其实要把on\_result\_info替换一下，做一个mock

我知道怎么回事了。。因为根本on\_result\_info就没被call，它是在等有东西发送了才会被call，而那边并没有东西发送

所以可以mock之后手动call

花一点时间再读一下第三章，看看是否有收获

在pytest里面，mock似乎兼备了stub和mock的两个功能

我现在是写test还是改代码？

我希望快速把测试框架写出来，然后refactor代码——包括添加功能，support general request

今日目标：既然要改了，是不是看一下pycharm的自动化重构？以及正好看一下那两个代码是干啥的（这个不太重要）

所以目标就主要是重构。当然老板那一手linux查询也很酷炫

Backlog是啥

Changelog

<https://keepachangelog.com/en/1.0.0/>

今天那个无非就是确认一下上传和下载的分别路径、名字即可

Mount下来的就是local的了？

解决方法，给idl\_defs.IdlVariableLengthObject里面的pack method改一改，让它support float,double之类。第二种，你直接用python自带的pack method不可以么？

等下。。我要改的是parser?不是parser，是template

注意的是，我改的其实还是protocol，所以到时候一样build一个测试一下就可以了

目标：我有一个文件夹，但我希望在不同的branch上面做事情

Ans: 很简单，你先创建一个remote branch，然后切换到那个remote branch去即可（同时也创建对应的local branch—git checkout -b <new-branch-name> <remote-branch-name>）

我觉得这个package改完之后，我完全可以本地conda build然后测试？

Ans: protocol应该不可以，还有一个.sh之类的东西运行把，不是直接变成package就行

当我已经conda build了之后，应该就是conda promote，但是它得找得到？

<https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/create-custom-channels.html>

真的要好好看一下conda整个过程

部署总是很麻烦，因为我要改一个地方，然后要别人approve，然后才能做。

所以我希望能够绕开approve也可以测试？（目前network drive我可以测试，那就没啥特别的）然后我可以build自己的tag的package，也可以用pycharm里面local引入来测试

PR到一个branch？PR不是给branch和master合并的么。。。这没问题

Serializable是什么…

如果type是确定的话，特别是长度是确定的话

下一步要造一个msg，然后试一试初始化

比较两个不同type的东西，除了有index索引的东西

变基

压缩commits已经知道了

Conda build

General conda可以看一下

Git可以再看一下

现在看一下orc\_position\_latest是从哪儿来的，

Orc\_position就是一点点加出来的

注意，orc\_position至少在load\_orc\_positions里面是最不重要的一个了。。然后每次都会被重新加载（因为没有先drop，虽然好像也没人care）。为什么要rerun，主要原因是orc\_trades和orc\_positions\_latest可能会被人为改动。

Product table里面有个namespace，和IQS里面的us\_commodity=2是对应的。

注意，这里有个问题。。。必须要import bunker\_apps，然后才能搞到config

我很想画一个datagram让整个数据流程清晰一些

是ayncio压制了错误

为什么一定要有那个跑了才行？把data diagram画出来

Git里面在不同branch的tag有啥区别？

Db不能insert into view?

打算做的事情：

一个是把转换类型的接口分离出来

要把error handling msg加上去

Topic是不是重写一下

注：比如是不是所有的转换都应该放在msgbuilder里面？

从sequence diagram里面再看看哪几个位置是发生了类型格式转换。大不了最简单的就每种转换都写一个toXxx的方法呗

还有一个，设置一些缩写

两个问题可待继续研究：第一个，jenkins上面build和local build有啥区别（一个当然是可以全公司都用了，那么我local build的怎么用呢）？第二个，那就是conda build出来的玩意儿有啥用…

这个神奇的硬盘位置"/net/binlib/build/docs"究竟是啥？从来没找到过。Ans：找到了，就在自己的盘上…

/net和/mnt比较像，也是一个nfs shared drive

Pytest可以指定pdb从哪儿开始么。差不多可以，用—trace。纯pdb可能没有，毕竟还有low level api

我知道了，参数化，如果有两个参数的话，它做的是m\*n次test，其中m和n分别是各自参数的长度。现在让我看看有没有不是cross的？答案就是把输入和期待输出放在一个fixture parameter里面就行了，只需要用一个参数化的fixture即可

注意-v挺好用的

不知道bbg\_header是不是要改

挺难测试的，不知道怎么测，即测试除了closing price之外的东西

究竟怎么Build一个package我可以自己测试？关键似乎是deploy那一步，还是Build那一步？

Ans:首先肯定要build，然后就是要Install，install可以让conda search local-channel，应该可以。其实channal都不用写，直接安装就是了

例子：conda install conda.dist/linux-64/bbg\_service-0.0.0-py36\_0.tar.bz2

令人头疼的python package问题

\_\_main\_\_.db.callback(\*args)可以直接call？这个callback是啥

Collections和tuple哪个是base class

With open和直接open是可能有区别的，因为前者才会call \_\_enter\_\_方法

# <https://bitbucket/projects/BNK/repos/aiodb/browse/aiodb/utils.py#69>

之后可以看下这个文档和asyncio里面的contextlib

<https://docs.python.org/3.7/library/contextlib.html#contextlib.AsyncExitStack>

目前对于哪个underlying sql的做法是先不动它。。。默认它会干正确的事情。如果真的是想查

苍天啊，Mock()弄出来的东西是没有\_\_iter\_\_方法的，很麻烦，要用MagicMock才对

<https://stackoverflow.com/questions/8166633/mocking-file-objects-or-iterables-in-python>

并行的话就可能有上下文切换。所以重点在于sql里面不能写yield？

Aiodb到底aio的是什么？我看它在dbapi.py里面最终调用的是psycopg2库的代码。我猜aio的是和数据库连接，而不是查询，因为查询都在lock里面：

<https://bitbucket/projects/BNK/repos/aiodb/browse/aiodb/postgres/dbapi.py#182>

一个是希望顺便看看怎么提高部署效率，一个是希望看一下finance，一个是希望加测试

Dockerize先？

Path那个，用过往包裹的应该没啥问题吧？Ans:如果不re-build就没问题，re-build的话就有问题

Pycharm里面设置Conda env也有点奇怪

Ans: 在existing env里面设置interpreter即可

<https://www.jetbrains.com/help/pycharm/conda-support-creating-conda-virtual-environment.html>

正解是用existing env，interpretor选python

新建包裹我知道可以用inv build来做，但觉得可以怎么样更快一点。果然不行，并没有覆盖原来的package啊。为什么conda install path/to/package并没有安装最新的？然后我发现conda install dist下面的那个就行，所以dist和conda.dist有啥区别？

给ide加plugin. Solved，好用又便捷

Enter/exit太复杂了，要不要理一理？Solved，就是包装版的\_\_enter\_\_和\_\_exit\_\_

等build了之后把bbg\_service里面依赖的版本号改了，然后make a push。不打算加test了

其实问题也不大，可以先自己tag一个，然后依赖于那个

Bitbucket里面怎么看一段的diff

以及conda install默认安装的是哪个版本？（即我怎么tag才不至于影响到别人）

Sql里面加了个extract出来的缩写就成了？

you can build whatever you want on whatever branch you want but you shouldn’t release unless tag is on master

问题：jenkins这些项目是run在什么机器上的？

还有我assume /net/binlib/build/python\_tools/repo/dev和http那个是同一个东西，只是一个显示在了网上链接里

看起来docker用来测试的核心之一是“共享”了磁盘（volumn？）

我已经build然后push到我们公司的registry了，然后捏？

对于portainer上面的app，它设定的运行时间在哪儿？它的参数是怎么给的？

话说既然有一个UI界面，说实话我觉得自己也可以直接上传？

我算是测过了还是没测过？

Add-on是啥玩意儿

看一下orc service我就知道它是怎么docker部署的，没问题。那其他没有dockerize的service是怎么弄的还得再问问

Docker基本看清楚了，不太清楚的也可以再看看doc，比如env之类的。接下来的问题是，没有dockerize的service是怎么搞的，以及如果我想要dockerize我的service然后部署该怎么搞：后者似乎可以参见orc service。

如果我可以直接docker跑的话，为什么我要那么麻烦再加那些.yml或.env文件？

一个是回答为什么要portainer，我觉得是因为可以在上面跑service，然后trader或者其他人就可以用同样的kunlun env测试？

生产用的是什么kunlun环境？

是不是复制到ch1prd就算release了？

Fast fail会不会让两个process同时访问到？像taxi-sim里面是最简单的情况，Event()没有任何side effect，只是一个dataclass。但如果Event有side effect的呢，event loop实际做的是不是context swtich的事情？显然，如果有一个global variable，然后有两个函数都访问(shared data)的话，哪怕不是asyncio的，肯定也都会更改。比如我们的service的SvcUtilApp实际上就只创建了一个而已，所以它的self变量都是会被重复访问。那关键只要on\_request每次都是像一个“新的”，那就没有问题了。这样的话，就算切换执行，其实也都是切换的不同的东西。关键还是落在callback function到底是怎么让它等候的

以及我的package怎么能像protocol一样放到dev去？

是否要给fast fail写一个strategy design pattern？貌似不合适，因为等于全部重写了

把generator视为task（因为yield可以挂起），用scheduler来做task switching。这就是一个user level的threading，其实没有什么底层玄学在里面。generator就是tasks，而yield就是“操作系统”给它们发送停止/继续运行的信号。

我觉得其实asyncio的核心就是在yield的基础上做了一些调度安排罢了，只是这个调度安排或event loop我们也看不到。（同时generator变成了coroutine）。其余从本质上说和正常运行没啥两样。

是不是说pdle上bunker的项目的Dockerfile和bunker文件夹里面的是一样的？

其实还是挺clear，整个pdle文件夹要做的其实就是两件事，build image以及create container

我就用自己的docker run在situate上跑？

问题：为什么放到pdle上面就可以弄到kunlunch了？是因为build.sh的魔力么，所以我不merge应该也可以搞上去

Image里面不用指定env，可以先搞到kunlunch上面，到创建container的时候再去设置env

为什么一定要docker build接着一个docker push才行？关键是tag貌似

先修改ssh登录方式: solved，实际上就是把key copy过去就行

在我想要jenkins跑之前本地测试package，除了inv build还能怎么做？

Git push origin tag是把tag push上去了，但没有commit？

还想加一个local docker build for test，因为快。现在看上去有两个选择，可以用if condition，或者用multi-stage build

Conda index可以删除么

How to put env in cmd in docker

这个不可行。不过我终于找到了！Env是可以在ch.yml之类的里面设置的

<https://docs.docker.com/compose/environment-variables/>

我是不是应该把entry point的flag什么都放在yml里面？

<https://docs.docker.com/compose/compose-file/#command>

对，应该放在那个里面cmd，这样旧的dockerfile就恢复回来

Db那边也想弄一个exception handling

Sending graceful shutdown heartbeat

关键今天就是弄这个

还有一个dataclass

Asyncio里面raise exception的问题: 即正常退出。这个似乎过于难了对我暂时而言

是不是考虑可以改成try/except/finally？Ans: no need

Conda activte怎么缩写? Ans: <https://askubuntu.com/questions/328162/how-to-create-shortcut-for-a-command-in-terminal>

还要加的话添加到.bash\_aliases即可

Linux cmd line里面也有写

测试通过solace直接发过来的情况，做一些logging

还有python里面能不能写ifdebug，这样我就不用每次删改了

Ans: <https://stackoverflow.com/questions/13352677/python-equivalent-for-ifdef-debug>

那问题在于打包了之后有entry point了，我还怎么写？暂时没有好办法

Ans:打包之后之所以是这样直接call，是因为setup.py里面的setuptools.setup里面的entry point已经写好了。

https://stackoverflow.com/questions/45930920/how-to-start-debugger-with-an-entry-point-script

以及很有趣地发现/sandbox/conda/envs/bbg\_service/bin里面都是些执行python的script，和ch1prd上面是一样的

去看一下上次ding说的conda环境，准备部署

\_\_init\_\_里面要写？目前应该是package引用有很大问题，有cyclic ref

看一下package

<https://stackoverflow.com/questions/16981921/relative-imports-in-python-3>

mmb，最后发现，其实都没问题，只是import的顺序问题？似乎确实是matter的，当我在\_\_init\_\_里面import的时候（cyclic import）

实际上问的问题是，如果我pip install一个editable的package，那么我import的时候import的是它还是directory里面的

这就是为什么需要relative import，看看书上string.py的那个例子

<https://www.pythoncentral.io/how-to-create-a-python-package/>这个告诉我们什么叫在package里面跑，而不是import一个module

万年未解之谜总算知道了！为什么会有这么多奇奇怪怪的问题，是因为把package和module彻底混为一谈了。

Ans:我是在build package，所以实际上内部应该都可以用from . import xxx。但更为关键的问题是，我不应该在内部，比如去运行\_\_main\_\_.py，因为从内部去运行的话，实际上它就不知道.指的是什么东西，或者parent package root，而我之前是通过from bbg\_service import，实际上这个bbg\_service是我Pip install的editable的package。所以正确的测试方式应该是用写的package的entry point去跑。

还有一个goal是熟悉bunker，确实需要

Exception. Solved

Pdb break其它module可以用<https://stackoverflow.com/questions/13589736/how-to-set-breakpoint-in-another-module-dont-set-it-on-function-definition-lin>

输出全名是ok的，import 我又失败了

<https://stackoverflow.com/questions/55695873/python3-7-module-not-found-even-though-it-is-listed-in-sys-path>

是在下学艺不精，结合上下两个看就知道。From bbg\_service import solace，同时solace的路径也要在里面加进去。

<https://docs.conda.io/projects/conda-build/en/latest/resources/variants.html>

exception的问题，似乎是因为我是在callback func里面

是否可以在我定义的exception里面退出？

一个看db view，一个看docker mount（-v mnt/bunker:mnt/bunker），还有一个Log的事情，还有一个delete query（这个不太好弄，delete要删除white\_list\_products，除非再传一个msg，不过问题不大，因为新进来的总是priority高）。证明了写的没有问题。

我猜是load到closing\_prices时没有overwrite的关系

<https://stackoverflow.com/questions/43938648/redirect-log-files-created-in-a-docker-container-to-stdout-stderr>

logging为什么不写到file的答案

<https://stackoverflow.com/questions/20240464/python-logging-file-is-not-working-when-using-logging-basicconfig>

<https://stackoverflow.com/questions/15727420/using-logging-in-multiple-modules>

还是得看一下library doc

<https://docs.python.org/3/howto/logging.html#configuring-logging-for-a-library>

最后我还是强行sudo chmod -R 777 ./加了权限，应该有更好的办法

<https://unix.stackexchange.com/questions/479890/permission-to-write-to-log>

<https://stackoverflow.com/questions/38676437/chmod-changing-permissions-of-my-script-sh-operation-not-permitted>

如果网上不容易找到答案，还是要多看看组里其它的code

Mount drive没有up to date?还是因为是我local的mount drive所以没有update？

Ans:我用mount命令看了下，应该是说在ch1devbuild05上面，实际上并没有mount，只是创建了这么个directory而已。在ch1prd和我自己的box上面，两者就是同步的

Git merge conflict只是照着书来了一遍，自己还是不太明白说实话

下面这个里面，先拉到feature branch，再通过PR的形式应该符合我需求。当然前提是你的local master branch必须是和remote一样最新的。不过这会不会有个问题，导致我的PR里面莫名其妙多了很多其他东西？

<https://community.atlassian.com/t5/Bitbucket-questions/How-can-I-solve-a-conflict-in-a-pull-request/qaq-p/577205>

下面这个是reset merge（取消merge）

https://stackoverflow.com/questions/6006737/git-merge-errors/6007083#6007083

packaging。似乎是inv release的时候，会自动把version update？

<https://packaging.python.org/tutorials/packaging-projects/>

类似的version可以参看Rec\_service，基本就是从version.\_\_version\_\_里面获取当下的

一般情况下直接attach就可以了，但Protocol那个是要build了才行，所以要local build一下再看

我是否可以看看怎么自己build一个package？应该可以conda build

TODO:设置一下akuna登陆的home directory

为什么我在repo里面搜不到自己的bbg\_service？

我想看Pyservice怎么运行，应该用pdb去弄比较好

怎么快速地熟悉一个service

Lockfile in linux: Each copy of the daemon will try to create a file and write its process ID in the file

下一步：service怎么异步或并行的原理？

看一下linux系统权限 ls -l即可。比如我可以log in ch1prd，怎么看我是否在这个名单里？

Windows权限怎么看 property--security

还有sudo到底干了啥，sudo不代表在所有group中？

Violet是干啥

Log out <https://docs.vmware.com/en/VMware-Horizon-Client-for-Windows/5.4/horizon-client-windows-user/GUID-5378232D-008C-47DC-8B61-4F44947C0419.html>

一个是能不能更好地进行搜索，找到我需要的代码

一个是需不需要看看其他service？还是说之后有ticket了再看？

Python Package怎么debug，以及它保存哪儿去了

怎么把现在改变的内容转移到另一个branch上面？

Enum类型用.\_\_dict\_\_来获得key和value?

Egg文件类似包裹

<https://stackoverflow.com/questions/2051192/what-is-a-python-egg>

如果想打开看的话unzip即可

<https://stackoverflow.com/questions/3655352/how-to-access-files-inside-a-python-egg-file>

先找到log文件存在哪儿

/var/log/akunadev/bunker/orc\_service

只要有context manager，创建postgre client时候就会自动连接

db\_client = await exit\_stack.enter\_async\_context(aiodb.create\_postgres\_client(db\_info, APP\_NAME))

<https://bitbucket/projects/BNK/repos/aiodb/browse/aiodb/utils.py#70>

wait\_for\_interruption的关键就是wait\_first，即创建了一个signal的future，当我退出的时候就会触发

看是不是handle exception，通过在我的callback func里面加一个exception即可

所以要让我的db能够handle exception并退出，只要确保有build\_asyncio\_error\_future()？

https://bitbucket/projects/BNK/repos/svcutil/browse/svcutil/direct.py#24

不过这个是对service用的，db还不太能用。要直接low level肉搏了？asyncio\_error\_future

<https://bitbucket/projects/BNK/repos/svcutil/browse/svcutil/aio.py#22>

发现核心问题是，callback里面的error并不直接体现到使用callback的stack上面。所以要么在结束的地方收集所有未处理的async error（但是我又用了wait block住了整个thread）；要么在内部handle

export PATH=/opt/akuna/trade\_bridge/current/bin:$PATH

做的事情是把新的一条path加到之前的path里面去

Source与anaconda/bin/activate

<https://stackoverflow.com/questions/27220145/anaconda-activate>

这个是不是就是conda activate？

<https://askubuntu.com/questions/849470/how-do-i-activate-a-conda-environment-in-my-bashrc>

我觉得看起来是volumn mount出现了问题。Ans：no

核心问题是run daemon，如果是在docker的话，它会直接退出。所以问题就是在于能不能不退出.

<https://stackoverflow.com/questions/25775266/how-to-keep-docker-container-running-after-starting-services>

现在确认两个东西：一个是mounted volumn，一个是scheduled time，还有改了config

可能的问题：先跑一遍全部的，然后再跑单个的

<https://medium.com/@blomsmail/linux-what-does-1-dev-null-2-1-actually-mean-14ebcc07bcdc>

conda真不是很好用…都不能跟着meta.yaml随时更新

看dependency

<https://stackoverflow.com/questions/55912363/how-to-analyze-dependency-tree-for-conda>

conda-tree whoneeds numba

conda-tree whoneeds msgpack-python

conda-tree depends msgpack-python

因为procotol 3.1.79时候没有peg

Spline bridge似乎logging error是写到前台，全部的log是写到log file

之所以spline\_bridge里面splinebridge.py会出现在/sandbox/conda/envs/orcsplinebridge/bin里面，是因为<https://bitbucket/projects/BNK/repos/orcsplinebridge/browse/setup.py#22>，和package install的setup有关

QUES: how is bbg closing price计算出来的？

很奇怪，为什么transform就不能step进去，load就可以？

Solved。因为一个是generator

SPDR就是黄金ETF

Public的ssh可以测试<https://www.sftp.net/public-online-sftp-servers>

接下来要看的：sftp，trading

how to find the overflow db row elegantly?

What does it do with new\_expiry?

Merge master into feature branch

https://stackoverflow.com/questions/16955980/git-merge-master-into-feature-branch

ctypes可以看看

是以idlc为entry program，以idlparser作为parser，idl作为generator(compiler)

自动化测试

Bbg\_service之后想update的地方：

把msg的logging用decorator包起来，这样以后要不要logging某个protocol msg直接在solace的msg函数那边改就可以，不用到处找究竟哪里call了logging

Bbg header可能也要另外单独拿出来，作为config

之后要把run-db分出来，正好趁这个机会把代码重构一下（是否要动setup.py?）

要做的核心有三部分：通过input file里面的信息找到对应的在我们instrumental db里面的东西，然后拉取数据（connect with instrumental部分，似乎有点麻烦，它还是用neosolace和tornado；但不应该麻烦啊，如果只是一个topic的话；还有一个问题是我们的机器上不能connect到prd的solace？）；找到desk mapping；写出到一个csv；转界面（这个容易）。

对我来说，mapping可能最难的，我先看看excel里面的东西。

要直接引用某个库，需要在pycharm里面右键直接run

utf-8-sig的问题

https://stackoverflow.com/questions/57152985/what-is-the-difference-between-utf-8-and-utf-8-sig

yield from以及reader

<https://stackoverflow.com/questions/9708902/in-practice-what-are-the-main-uses-for-the-new-yield-from-syntax-in-python-3>

因为CSVReader其实也是一个generator，一行一行地yield，所以我们直接yield from CSVReader() = for xxx in CSVReader(): yield xxx

现在依然有一个matching的问题。Offset day是啥玩意？

我可以通过exchange+PF code来在abn\_products找underlying,，Underlying可能没有expiry\_dt；那么怎么通过underlying来找那个instrument，进而也就获得expiry date呢？csv里面有contract month

目前流程once i know which underlying in IQS the row refers to, i will use contract month to find the expirations and then get the future. for option i will use strike price and type as well to find the option.

45行找不到；NYM怎么处理(XNYM)。找不到是因为abn\_products里面没有，还是真的不trade；各家的symbol都不太一样？bbg和abn的，有些又都一样，有没有什么mapping除了abn。怎么看能不能booked?还有说reconcil里面看mapping

Gmi code

Neosolace

res = yield ptron\_service.get(pp.named\_desk\_spec)

# p res[1].underlying\_ids

# https://bitbucket/projects/LEG/repos/packages/browse/neosolace/neosolace/idlproxy.py#64

# 不知道哪儿设的getter方法, 很有趣的type，<class 'neosolace.idlproxy.named\_desk\_spec'>，似乎是动态生成的

# https://bitbucket/projects/LEG/repos/packages/browse/neosolace/neosolace/idlproxy.py#139 看这行里面的origclass

# ah 是这个创建了metaclass

<https://bitbucket/projects/LEG/repos/packages/browse/neosolace/neosolace/idlproxy.py#17>

is\_booked还要问一问，orc\_trades里面怎么看？

是不是建一个table

有没有return yield这个写法？

https://stackoverflow.com/questions/44051647/get-params-sent-to-a-subcommand-of-a-click-group

还是packaging的问题，python怎么分包exe包？以及为什么python的exec文件里面都是placeholder，执行不了。Pyinstaller可以么https://pyinstaller.readthedocs.io/en/stable/operating-mode.html

#!/bin/sh

'''exec' /opt/akuna/envs/bunker\_apps/versions/24.14.5/bin/python "$0" "$@"

' '''

# -\*- coding: utf-8 -\*-

import re

import sys

from bunker\_apps.load\_orc\_closing\_prices import entry\_point

if \_\_name\_\_ == '\_\_main\_\_':

sys.argv[0] = re.sub(r'(-script\.pyw|\.exe)?$', '', sys.argv[0])

sys.exit(entry\_point())

#!/bin/sh

'''exec' /sandbox/conda/envs/bunker\_apps/conda-bld/bunker\_apps\_1601053950182/\_h\_env\_placehold\_placehold\_placehold\_placehold\_placehold\_placehold\_placehold\_placehold\_placehold\_placehold\_placehold\_placehold\_placehold\_placehold\_placehold\_placehold\_placehold\_placehold\_/bin/python "$0" "$@"

' '''

# -\*- coding: utf-8 -\*-

import re

import sys

from bunker\_apps.load\_bbg\_closing\_prices import entry\_point

if \_\_name\_\_ == '\_\_main\_\_':

sys.argv[0] = re.sub(r'(-script\.pyw|\.exe)?$', '', sys.argv[0])

sys.exit(entry\_point())

一个是prd里面的，一个是自己生成的package里面的，看到区别了把

pythonw

https://github.com/chriskiehl/Gooey/issues/259

跨平台打包还是有问题

https://stackoverflow.com/questions/2950971/packaging-a-python-script-on-linux-into-a-windows-executable

pyinstaller打包的一些细节技巧

<https://blog.csdn.net/zengxiantao1994/article/details/76578421>

<https://medium.com/@peaceful0907/%E5%B0%87%E4%BD%A0%E7%9A%84python-code-%E6%89%93%E5%8C%85-pyinstaller-6777d0e06f58>

关于python打包的讨论

https://juejin.im/post/6844903936722206728

https://python-packaging.readthedocs.io/en/latest/minimal.html

这是官方教程了吧

https://packaging.python.org/overview/

这也是书的一个章节，还参考了pep，比较抽象宏观，是打包库的作者写的

https://www.aosabook.org/en/packaging.html

这是介绍实践中怎么把一个package上传到pypel

https://realpython.com/pypi-publish-python-package/

这也是一本书

<https://diveintopython3.net/packaging.html>

是不是说pyinstaller是把当前环境下的安装的package都打包进去？

<https://docs.python.org/3/library/decimal.html>

这问题难道实际变成了，如何add site-packages到pyinstaller?

<https://stackoverflow.com/questions/59743234/how-can-i-add-a-python-site-package-folder-thats-not-being-included-to-a-pyin>

还有一个就是cross platform的问题

Exchange名字mapping

<https://support.quotemedia.com/support/solutions/articles/13000020527-exchange-code-group-list>

如果要静态修改.req file的话，一个是要把submit\_req里面write\_reqest\_file注释掉，然后要把path factory里面的request\_id注释掉（不然每次生成的路径都不一样），然后path factory可能得把ts也改成静态的。

qQ19941224!

# 30183565

dash

<https://www.dashregtech.com/>

<https://wiki/display/DEV/Bunker+DASH+process>

明日计划：dash，protocol。如果bbg有回的话那就继续问那个

<https://stackoverflow.com/questions/52494425/running-a-pytest-test-from-cmake-where-the-test-and-sources-are-in-different-fol>

conda.yaml似乎要加pytest在build里面。Ans:老板说不需要

git怎么在merge之前就解决conflict？以及clean PR。看起来核心是squash commits

<https://hackernoon.com/make-clean-pull-requests-version-control-and-collaboration-420ecff2e7b5>

关于git，我的问题是：第一，我知道merge会有conflict，我能不能在PR时候就solve conflict？第二，我从local push到remote的时候，能不能就squash commits（而不是在PR merge的时候）？

正确的做法是不是在local先把最新的master merge到我自己的branch上面？Ans: yes（前提是master上面并没有把你改的地方也改了把）

>>> a=[(1,2,3),(4,5,6)]

>>> b={x:(y,z) for x,y,z in a}

>>> b

{1: (2, 3), 4: (5, 6)}

从来没见过这语法

要做的事情是，替换PeriodicCallback。当然首先得知道这玩意儿究竟是干嘛的

Asyncio教程<https://github.com/jmoldow/jmoldow-python3-async-tutorial>

很糟糕

说generator和coroutine主要区别在于一个是可以决定下一步执行什么，一个不可。为啥说generator是iterator的简化版？kind of，毕竟generator是iterator的subclass。一个是写yield，一个是要explicitly写下一个循环是什么

<https://en.wikipedia.org/wiki/Coroutine#Comparison_with_generators>

希望完成一个功能：能够将一些test版本的设置保留在一个branch上面。所以git checkout -b是把当前branch上最新commits复制过去，还是没有staged也复制过去？Ans:从操作结果来看，没有staged的也复制过去了。

似乎情况是这样，checkout到new branch之后，如果new branch把修改的地方stage和commit了，旧branch上面就没有unstaged的东西了

Bbg service:目前的情况是，对于closing prices or not，我们有截然不同的做法。

不管是closing prices还是其他，我们都可以有comment line（这没问题），并且在code里面标出comment line的格式。对于没有提供expiry date之类的field，我们就不去管它就是了。

这UnderliersQuery是不是应该放到db那边去？只要我确定pass过来的名字是直接可以用的。Ans: yes。bunker\_utils.BBGSymbolConverter（在bbg\_utils.py里面）是之前用的mapping。

Hmm我还是必须要给service留一个mapping/symbol\_convertor的位置，而且这个应该是transparent to user的（意思是不影响protocol msg），所以它放的位置应该是BBGRequestTransformer这边，加一个flag即可，如果user加flag，那就必须在msg里面pass给service？但是我还是觉得应该transparent to service比较好？

现在我知道原本为什么.req里面要写comment了，目的就是为了这个mapping的事情，不需要再逆映射一回。

存一个local的mapping在那边。把这个class还是放在adaptor里面，同时让adapt函数能接受一个cache变量，缓存mapping。

我可以写一个symbol convertor的adaptor，只不过多加一个参数？

我让它先转成protocol msg然后再做converter可以么？而且db这边得把expiry什么都放进去。应该都放在adaptor里面做吧？

Bbg和db应该不用打任何交道。

怎么把比如昨天的所有数据都copy过来，但只是改一下日期

<https://unix.stackexchange.com/questions/153241/how-do-i-delete-a-folder-but-not-the-contents>

消除中间层，但记得用cp xxx/. 而不是mv

Bbg serivce别忘了改yaml里面加bbg\_convert和protocol version

Bunker\_apps也要改

一步步改会比较容易。先把db和service分开来

对于一个一路传进去的变量，有没有比较好的方法(whitelisted\_product，config也有这个问题).Ans:在上层创建

我能不能把config单独拿出来，比如放到一个file里面，然后再单独创建一个class是获取config（类似bunker\_utils里面的get\_config），然后把sftp，header，solace三个config分开来（它们本来就不该在一起）

把两个transformer拿出来另外放一个file

另外第二个tranformer出来不应该用零散的数据，要用type

我觉得bbg\_converter实际上就是一个adaptor

如果我把所有内容，包括msgbuilder都用我自己的type封装起来，这样是不是好？Done

因为你只需要两个converter，所以并没有必要用decorator

在bbg\_converter里面要加一个从db获取数据并存储的script，但是它不见得要打包到Package里面。Ans：看起来不用，因为就是从db导出来一个csv file而已。别忘了dump的话设置一下csv格式，让第一行的header保留着。

Db之前timeout的问题是因为password不对，按照<https://www.jetbrains.com/help/datagrip/reference-ide-settings-password-safe.html>重新设置成每次询问即可

怎么设置路径是个问题.Ans:solved

Mypy检查typing有问题，就用Any搪塞过去就行

如果一个msg里面有一个field要变动，是否应该拆散整个msg type？

要做的事情：mock file，以及改一下meta.yaml(solved，删一个空格即可)？或者应该把data file package进去。原来是因为我没有修改git ignore…data file从来就没上传上去过

What is a good test? Refactor service里面的这些error handling和if statement.

还有一个是在run time initialize的问题

可以用两个mapping，可能想看一下2sum…失败

Header config是只有一个instance，所以我把基本信息放在self里面，每次制作具体的header那就需要做一个hardcopy. Ans: copy.deepcopy

Docker的接口也要重新写一下

Python里面，比如一个类型原本是List[Optional[…]]，在我确认过它非空之后，把它传给一个接受List[…]的函数，怎么在type check里面体现？Ans: typing有个cast函数

excel用计算结果替换公式：比较naïve的方式是复制黏贴…黏贴的时候选择黏贴值

mmb，把fireblocks里面的bunker\_apps comment掉

pdb妙用，在package中断点（可能要先pdb里面import etl）

b etl/task.py:77

为什么会多一行，估计就是每次bank transfer产生的hash不同

原来所谓807条，说的就是全部fireblocks的记录

Crypto service的代码现在我们还看得到么？Ans:已经获得

opus也是一种加密货币。Gemini是一个比特币交易所

python or和and

https://docs.python.org/3/reference/expressions.html#and

Crypto journal:

先从match\_crypto\_transactions.py开始，这个可以选择match一个月（会从月初跑到run script的前一天），或者一天天来。然后默认应该选clear-all flag，至于为啥我还没看懂。

然后这个script里面会先跑一下load\_crypto\_transactions.py，从各种地方下载（可能在db里面，有的是存在mount drive里面的csv，比如fireblocks，但它最后也是load到db。注意fireblocks下载的是全年的。。。虽然代码里会帮忙过滤到其它时间的），自己注释掉不需要的exchange（悉尼那边的不需要）。这个load\_crypto\_transactions.py本质上就是跑存在reconciliation/exch\_tasks下面的各种task，而各种task实质上就是对db先delete（如果force\_download设为true的话，但从match\_crypto\_transactions.py里面的lct看，并没有delete crypto\_transactions的东西？注意fireblocks的task没有delete，所以要人工删除），然后query对应的exchange的table，然后transform一下，然后再load到crypto\_transactions这个db table里面（load是存在reconciliation/load/transactions.py）。这些都跑完了，然后我们又回到了match\_crypto\_transactions.py，真正开始执行它里面的CryptoMatchTask。所以说实际上第一步是load，第二步是match，第0步是谁把数据放到db里面，那里面就是交易transfer数据。Match script的clear-all flag删除的是crypto\_transfer table，并不动crypto\_transactions。而且match script目前是跑一天match一天。

注：现在新加一个bybit的load task。显然bybit\_transfer里面有address，trx\_hash，还有exch\_id。

注：不知道谁写的code，之所以load\_crypto\_transaction.py里面你哪怕不set force\_download，db table也不会因为多次load而重复，是因为exchange task里面的load用的是InsertNoConflictQuery，如果key一样的话就不会复写（在crypto\_transaction table里面key是trx\_id）。不过我觉得最好的还是把force\_download设为True，同时注意，这个deletequery删除的是这个exchange除了comment带有manual insert之外的所有。

注2：想看table schema的话，找到那个table，点击下拉展开箭头就可以了…

然后这个script的核心是CryptoMatchTask。这个task一开始先populate了三个service:alias\_service, account\_service, rate\_service。account\_service在journal/account.py里面，首先把account做了一个分类，比如loan或者bank或者exchange或者pnl什么的，查询了crypto\_accounts table，然后又做了一个mapping，似乎主要这个service功能就是做个账户分类，mapping是说用(exchange—这个exchange是指crypto\_accounts table的column name, currency, acct type)获得(acct number, name, entity)，其中name像description，entity是akda akca之类的东西。注意对于每个account type它有不同的get\_xx\_account的方法。alias\_service查询的是crypto\_party\_aliases table，好像也是做了一个mapping。有四种alias type: 'otc\_wallet', 'bank\_name', 'otc\_counterparty', 'bank\_counterparty'。然后实际上就是alias的一个mapping，从一个名字可以得到另一个名字。第三个是rate\_service，查的是fx\_rates\_view table，这个是每天更新的，从bbg来，就是获得汇率（似乎是以美元为基准货币）。

然后CryptoMatchTask开始做一些pre-process，如果set了clear\_all，db里面会删除dt之后的crypto\_transfer table里面的记录。如果set了clear\_matches，除了是手工修改的（写了manual或disabled），会把dt之后crypto\_transactions table里面所有row的match\_type column都设置为null（但为什么只是设置为null而不全部删除？）。Comment里面说clear\_al会override其它flag，见create\_preprocess\_task()。

然后CryptoMatchTask开始extract，有三个extractor（即最后会返回一个三元tuple），位置都在journal/queries.py下面。要注意的是，虽然我们是按每天跑的，但第一个extractor是会把table里面的所有数据都拿出来的（尽管理论上在前面load的时候也是一天天load的，但如果你每次跑之前不把db crypto\_transactions里面非manual或disabled的东西都删干净的话，就会出错；而且第一个extractor只要求match\_type是null，这不就意味着陈年旧unmatch也显示进去了？）。第一个extractor从crypto\_transactions table里面取出所有match\_type是null的row (这个是主要的路径，我们主要要处理的transactions的位置);第二个用的是reconciliation/parse/bank.py里面的BankTransExtractor，它读取的文件路径是/mnt/bunker/prd/crypto\_transactions/bank\_transfers，但是哪一步把文件放过去的呢（答案是人为load过去的，每个月问finance的人要bank\_tranactions的文件，如果需要的话还可以按日parse一下，代码见parse.py）？这个BankTransExtractor做的事情还是比较清楚的，如果满足bank==counterparty条件的话，表示就是一个银行内部转账，我们不用管，另外如果这个csv file里面的bank或counterparty 不在我们的crypto\_party\_aliases里面(就会raise一个keyError，同时会logging报错，提醒你要更新crypto\_party\_aliases，但一般应该不会)，就返回一条bank\_transactions给后面的transformer，至于从bank\_transfer文件创建一条可以放入crypto\_transaction table的记录(是在哪儿把bank transfer放到db的？)，其中trx\_id什么的都是自己随机生成的（见create\_bank\_trans）；第三个是OTCTransExtractor，和第二个差不多，从/mnt/bunker/prd/crypto\_transactions/trader\_otc那边得到（现在一般是空的，只是每天创建一个空文件）。要注意load\_otc这边有时会load一些奇怪的数据进来，此外otc里面的数据，symbol是真正的货币，而currency是法定货币，是我们用来计算value的。看起来OTC那一条确实symbol是ETH，但值不对，应该是30000美金对应的ETH的值。

注：bank transfer是没有process去删的，这似乎是个问题（目前我在match\_transactions.py的pre\_process里面加了一步）。不过它是在哪儿load到db table的？我猜是最后的TransferLoader干的，因为只有这个和db\_conn有关系。

然后是transform过程，CryptoMatcher。先初始化三个matcher，在reconciliation/match/\*里面，其中如BankMatcher只处理银行间交易，即货币都是法定货币。exch\_matcher是用blockchain hash来match，如果不行再OTC，再不行再bank?

首先我们来看ExchangeMatcher，它扫描所有没有match的数据（就是crypto\_transactions里面的1行），建立一个transaction\_hash（trx\_id冒号后面的部分）的mapping，对每个transaction\_hash，应该有且只有两条交易与之关联，如果确实如此，那就match上了。这部分只需要用exch\_matcher.match\_with\_hash方法，还是比较直观。exch\_matcher.transform方法，先把所有已经match上的交易排除出去，然后把剩下的交易按照qty正负分为withdrawals, deposits两类，这边实际上就是在用qty来做match了，如果qty比较近的，我们就认为match上了。那么crypto\_transactions table里面的数据是从第一步load里面，从各个分散的exchange table里面来的。

而bank\_matcher也是把bank transfer先按qty是正负分成withdrawal和deposit；对从exchange那边获得的trx也做同样的事情，同时会在self.bucket里面把之前已经match上的trx去掉。然后开始match\_bank，对withdrawal和deposit各自做一次，一次是tight match，一次是loose match。核心是match\_bank()函数。

最后是load到两个table, crypto\_transactions和crypto\_transfer，在reconciliation/load/transfers.py里面。Transform返回的new legs会被load到crypto\_transactions table，不过new legs是毛线？

Cypto journal做完之后，一个是load\_crypto\_journal.py，另一个是loan\_and\_pnl\_email.py。

先看看load\_crypto\_journal.py。

这个核心是CryptoMatchTask。它一开始先初始化四个service，分别是account, rates和两个balance（分别是月初和月末的）。

然后是两个extractor，分别是抓去transfer和in transit（这个是从crypto\_transactions里面找match\_type是in transit的东东）。

然后是CryptoJournalTransformer。对transfer，一上来我们先用journal/transaction.py里面的identify\_unmatched\_bank\_transactions跑，目的是parse一下。Parse完之后，就把create\_trx\_entries，create\_pnl\_entries，create\_reval\_entries（pnl和reval似乎是实际上是loan\_and\_pnl\_email.py里面跑？）这些itertools.chain()起来返回，就是把iterable串起来。

我们先来看一下create\_trx\_entries。它只有在特定情况下才会跑fee。它返回的是journal里面的一行JournalEntry。我们要关注一下liquidity和description这些都是怎么产生的。其实liquidity就是amount，只是换成以美元为单位而已。Description就是把src和destination以及用的currency和交易日期放了进去。Name指的是account name，即accounts里面对应的名字。

然后是create\_pnl\_entries（它是在pnl.py里面）。首先也是把balance sheet拿过来。然后是对cypto\_transfer里面的transfer进行循环，然后算的那些liquidity什么的和create\_trx\_entries很像。然后算一个pnl，这个是一些liquidity做加减法得到。最后返回的也是JournalEntry。pnl.py里面对transit是有处理逻辑的，前提是在comment里面要加上Transit这样的字眼。

最后是create\_reval\_entries，见下面loan\_and\_pnl\_email.py。

最后是SplitLoader。这个的目的是把journal entries split到它们各自的entities里面。

Transactions.py里面还有一个generate\_loan\_data()，它是在load\_and\_pnl\_email.py里面会被call。它也是直接从db读取transfer，所以如果上面create\_trx\_entries里面更新了什么，这边也要更新。

注：所以不同的account type实际上也是对应于创建journal中不同的步骤，比如pnl就是对应创建pnl sheet。当然也是对应于不同的交易类型。

注：OTC是不可能有fee，一种做法是create account anyway，另一种做法是把它的fee加到exchange那一侧，这样最后src和dest的liquidity和fee\_liquidify才会balance

注：之前没有credit from exchange的key报错，是因为之前没有碰到这个exchange的这个情况，也就没有在accounts里面建立账户。

注：notional value和market value的区别

https://www.investopedia.com/ask/answers/050615/what-difference-between-notional-value-and-market-value.asp

注：

def chain(\*iterables):

for it in iterables:

for each in it:

yeild (each)

注：我们在不同的exchange以及银行有各种类型（account type）的账户，用来应对不同的需求。

说到loan\_and\_pnl\_email.py，跑的话传的日期应该是每个月月末，看一下crypto\_transactions/balances里面的balance文件是啥时候的就可以。核心是CryptoBreaksTask。一上来先初始化三个service: account\_service, rate\_service和balance\_service（这个根据start和end date创建了两个）。然后还弄了一个start\_dt和prev\_dt，start\_dt是把你传给它的日期里面的日变成1号，prev\_date是算start\_dt的前一天，把周末去掉，end\_dt就是你传进去的那个日子。balance\_service的任务是读取balance file，然后实际上是读两个balance file，即crypto\_transactions/balances里面的。Rate\_service就是为了获得fx\_rate，传给它两个日期，它就会把那段时间里面的所有fx\_rate都从db里面取出来。

然后又是etl，extractor做的是TransferQuery（在journal/queries里面），查的是crypto\_transfers table里面的东西，在起始结束日期之间。

Transformer是CryptoBreaksTransformer，它把前面的account\_service, rate\_service和balance\_service都作为初始化变量，还把prev\_date和end\_dt也传进去，实际上就是说从上上个月月末，跑到上个月月末。里面的transform核心是call了journal/reval里面的create\_reval\_entries。

create\_reval\_entries把crypto\_transfer db table里面取出来的transfer data，account\_service, rate\_service和start\_balance\_service以及开始结束日期作为函数变量。这个function里面report\_date就是前面的end\_dt。首先初始化一个defualtdict loan\_revals，如果key不存在就返回Decimal('0')。这里面主要是是两步。第一步是用rate\_service做一下fx转换，而我们记得rate\_service里面主要就是一个journal/queries.py里面的FXRateQuery，查的是给定时间范围里面的fx\_rates\_view table。然而因为周末的问题（即report\_dt不是在工作日，trans\_dt在工作日），所以我们要人工把report\_dt\_fx更新一下（下次是否可以考虑让finance在folder那边放一个文件，然后我们把coin\_price.py嵌入到现在每次人工复制黏贴的reval.py里面的位置）。然后又创建了一个defualtdict reval\_total。然后又把所有的loan account都拿了出来，即all\_loan\_accounts set。

最后这个create\_reval\_entries返回的也是JournalEntry，用的是reval\_total里面的值，它又是通过new\_loans函数转换loan\_revals里面的值。这个create\_reval\_entries最后是yield两条，一条loan，一条reval。

注：为啥fx\_rate\_view table里面不是所有的price都在每天的Fx email里面发出去？注意这个script是在send\_fina\_fx\_rates.py里面。这个view是由bbg\_fx\_rates，cmc\_coin\_prices，yahoo\_coin\_prices这几个合并而成。为啥QQQ也在fx\_rates\_view里面？而且这个table里面似乎只有工作日的数据，而实际上加密货币每时每刻都在交易。注意在rate.py里面，如果我们没有这个date或currency，我们的做法是找就近日期（向前找）的汇率，即get\_second\_latest\_price这个函数。Script是load\_bbg\_fx\_rates.py。

时间都不对

Crypto journal 常见unmatched及处理方法：

1 对bitstamp来说，它似乎不提供USD的transfer数据，所以如果bitstamp\_akda是在target的话（一般发起方其实就是银行了），我们自己创建反向的transaction。这个新加，所以没啥问题。

2 对itbit来说，首先目前我们并没有获取它的数据的api，所以肯定是match不上的。然后它交易的是BUSD，也比较特殊一些，一般的使用情况是binance -> itbit -> silvergate，所以我们得手工补上这中间的几条记录，或者把api写了。

https://www.paxos.com/itbit/

3 对lmac来说，虽然crypto\_service写了这个exchange，但并没有lmac\_transfer table，所以我们也只能手工加上对应的交易。

4 对otc以及otc interest来说，一般都是没有trx记录的，比如blockfi,Celsius,genesis。所以手工加

5 fireblocks文件里面有notes，有的会写target，如果写了的话我们可以人工修改fireblocks load的trx的target。注意这是修改的。

6 exchange之间不太可能直接转美元，所以如果看到exchange之间转的话，看看是不是逻辑链条弄错了。应该是exchange到bank之间的。做法就是人为修改target，然后在comment里面加Manual。因为我的load的task里是insert\_no\_conflict。

Sql script见聊天记录及db table

注：bakkt是一家管理公司，好像和ICE有关系，所以ice future delivery我们的target是它

注：coinflex的数据db也是missing的

注：似乎原本fireblocks不用insert\_no\_conflict的话会miss掉trx。原本因为没有说do nothing，它就会报错，但又不会退出整个process，所以其实导致那个日期后面的trx没load进去。是因为logging么？

TODO: fireblocks里面，增加一个步骤，在读取trx时，如果发现有重复id的直接就改了（比如加\_1），这样后面还是insert\_no\_conflict。

对于qty比较小的（比如小于1的），manual insert里面增加一个update row task，直接把它们批量都disable掉。

加一个select query是给所有match type =null的加reverse trx。

我们是不是要把manual insert和disabled分开来？对的。

事后人工添加：如果是external确定了，就把target在原始的csv里面更新？或者做法是写real Manual Insertion。在real Manual Insertion之后要先把之前Manul Insertion的东西删了（这个命令请手工跑一下），然后run一遍match，然后再manual insertion，然后再run一遍match（这样我们不会多加manual insertion的东西，因为manual insertion都是在match type = null里面找的）。

现在给load fireblocks的过程加一个采纳note的做法。Fireblocks的文档在

https://docs.fireblocks.com/api/#introduction

看到vault。https://www.geeksforgeeks.org/introduction-of-vault/

项目管理之道

https://book.douban.com/subject/30230063/

Dash:

显示run一个script，这个script会抓取各种数据，并写出到一个文件，符合DASH的格式要求。

似乎是有两个部分，一部分是自动化跑的，一部分是人工跑的？这个在bunker\_apps下面，在situate里面到middle office搜就有了

Dash script部分

核心就是这个DashRiskBasedHaircutTask。

里面首先有pre\_process：CopyOccTheoParamProcess和OccIntradayTheosArchiveProcess。

先看CopyOccTheoParamProcess。就是copy两个xml files，从/mnt/ndrive/ABN Files把RBHtheos和RBHparm两个文件拷贝到/mnt/bunker/dev/dash/import/下面，名字分别是theo.xml和parm.xml。

OccIntradayTheosArchiveProcess是从sftp那边得到两个文件['LPIDPL\_BD.txt', 'LPIDPL\_MM.txt']，放到/mnt/bunker/dev/dash/20201104/13/下面，最后两个数分别是date和hour。所以这个似乎不应该我自己跑，因为现在已经有了（situate已经每天跑了）。然后会把这两个已经存到local的file，再copy到/mnt/bunker/dev/dash/import下面。

然后是extractor：两个db query，分别是StocksAndFuturesQuery和ABNPositionsQuery。

先看StocksAndFuturesQuery。它很简单，就是从products table里面把future和stock类型的product和product拿出来。

ABNPositionsQuery比较长。不过还算直白，就看最后rbh\_pos那个，select出来的就是它里面的全部东西，entity, product, product\_type,expiry\_dt, strike\_price, price, sum\_qty之类的东西。

注：之前人们都是在dev里面跑的，如果我想跑上个月的，等几天dev db也就更新同步了

然后是一个transformer：PositionTransformer。

其中一部分是调整生成的file的格式。前面StocksAndFuturesQuery得到的东西变成{product:product\_type}的mapping即可，主要是辅助作用。PositionTransformer取出来的东西叫position\_rows，它要进行一系列的类型转换，然后yield出去，叫fields，看上去就是把每条trx都记录了。

然后是一个file loader。把前面的数据放到/mnt/bunker/dev/dash/20201104/13/ourpos.txt的位置。第一个post\_process CopyPositionsToDashProcess把这个file copy到/mnt/bunker/dev/dash/import下面

最后还有一个post\_process：CopyPositionsToDashProcess和TriggerDashRBHProcess。

CopyPositionsToDashProcess上面已经讲了。

TriggerDashRBHProcess先在/mnt/bunker/dev/dash/import下面创建一个空的RBHBatch.txt，然后等Dash program自动去运行，如果运行成功的话，RBHBatch.txt应该会被删了。不过事实上我看log，似乎从来就没成功过。如果dash program成功的话，我们就会把/mnt/bunker/dev/dash/print/下面的DASH\_OUTPUT\_FILES复制到/mnt/bunker/dev/dash/20201104/13下面

结论：因为situate每天都跑，所以对我来说，我自己其实都不需要跑，直接就把它跑的在/mnt/bunker/prd/dash/20201102/16里面月初的三个txt放到dash software里面就行了（更确切地说，是放到/mnt/bunker/dev/dash/IMPORT下面）。还有从/mnt/ndrive/ABN Files把RBHtheos和RBHparm两个文件拷贝到/mnt/bunker/dev/dash/import/下面，名字分别是theo.xml和parm.xml。记得用正确的日期的，方法就是看txt和xml的datemodified都是一样的即可，每个月第一个business day。但发邮件的时候说是月末hhh。

https://wiki/display/~cheng.hu/WFH+Set+Up

hmmm怎么umount？似乎只是把/etc/fstab里面的注释掉然后再跑一遍mount -a并没有用，umount我没有权限

dell旋转外接屏幕的问题，通过在mac的setting的display里面设置即可。

关于屏幕全屏的问题，解决办法就是在monitor上直接打开windows remote machine，而不是mac打开之后再拖过去

此外还希望两个屏幕能做到，一个屏幕锁定只显示一个程序，比如windows remote desktop，其它工作在另一个屏幕上完成。

Ssh那个挺直观，一个是ssh-copy-id，还有一个就是在.ssh下面改一下config file就可以创建shortcut了

Csv pivot table还是很好用的

看起来logging的config可以写成”DEBUG”这种

前面logging不显示看来是没有把console和local\_file加到config里面，现在都对了

我应该预留一些时间，如果需要去看reconcile的整个过程（应该不会很复杂）。

我是不是需要把config放到trade support的acc里面？

比如trade bridge，在ch1prdtdb01的/opt/akuna/envs/trade\_bridge/cfg里面。那么它是怎么release的？

docker create --name=${AppName}\_${INSTANCE} ${CommOpt} -v /opt/akuna:/opt/akuna -v /var/log:/var/log -v /tmp:/tmp ${RegServer}/${REPO}/${AppName}:${VERSION} ${AppName} --config-file /opt/akuna/envs/trade\_bridge/cfg/trade\_${INSTANCE}\_config\_docker.xml

从release.sh可以看出，实际上它做的事情，就是在BASE\_DIR/envs/$PKG/versions下面安装已经在网上index里面的package，然后把这个版本放在current version里面。核心问题应该已经封装在conda create里面了。至于用的是什么conda，在里面也有说明。

似乎这个是symlink过去的，ls -l一下就看出来了

如果man一个东西没有的话，可以再试一下help，比如help set

Pycharm有个Source code navigation

<https://www.jetbrains.com/help/pycharm/navigating-through-the-source-code.html>

但看起来不是很好用。。control+click可以跳转

Set -x里面一坨，你按了没啥反应，但是之后再跑其它的就都是反应了。。。幸好这个只是临时设置的，新开一个terminal就没了。

又想到一个问题，你这个config放到外面之后，portainer那边怎么弄？看一看trade bridge是怎么处理config的

Tag dev version on feature branch is good?

Rec\_service里面的akuna\_trade和非akuna\_trade可以看看

然后几个file都用的常量是不是可以放到一个单独的constants文件里？Ans：目前已经这么放了

conda verify为什么会出错？为什么PR不报错，但我inv release就报错？等会儿，因为inv build的时候，它build的确实是只有0.0.0的东西…并没有你那个0.0.0.dev0，所以verify说找不到也是很正常的？但是仔细看

[2020-11-09T20:47:02.993Z] anaconda upload conda.dist/linux-64/bbg\_convert-0.0.0\_dev-py36\_0.tar.bz2

[2020-11-09T20:47:07.143Z] cp conda.dist/linux-64/\*.tar.bz2 dist/linux-64

[2020-11-09T20:47:08.061Z] tar (child): dist/linux-64/bbg\_convert-0.0.0.dev0-\*.tar.bz2: Cannot open: No such file or directory

说明已经弄出来了而且复制过去了啊？

有两个问题。一个是说，既然是在例如cofdevbuild04上build的，那么package最开始是怎么弄过去的？复制过去的么？另一个是说，这个path是在哪儿…

Setup.py里面的version是不是会动态改变的？

注意，ctx.run会把version tag里面的下换线变成点（.）

rc0是成功的。。。

logging也还是有问题

注：BRN是brent oil，https://zh.wikipedia.org/zh-cn/布蘭特原油。是在ICE上交易的https://www.theice.com/products/219/Brent-Crude-Futures

注：利用python进行数据分析的中文版就看这个网页吧：

https://github.com/iamseancheney/python\_for\_data\_analysis\_2nd\_chinese\_version

check breaks:

<https://wiki/display/DEV/Intraday+Transaction+Reconciliation+v1>这个是intraday transaction recon，暂时雨我无瓜

最后send email是在send\_breaks\_email.py:793。这个workflow在situate里面是同名的，每天跑三次。因为file\_name = f'breaks\_{abn\_region}\_{date.strftime("%Y%m%d")}.csv'。似乎应该是看这个。里面一大坨check\_eod\_accounts.py → check\_eod\_deleted\_trades.py → check\_eod\_expirations.py → check\_eod\_positions.py → check\_eod\_transactions.py → send\_breaks\_email.py。

然后这一次我们看的应该是check\_eod\_transactions.py。而这个script实际上运用的内容就是check\_expirations文件夹下expirations.py或queries.py或warrant.py里面的东西。

check\_eod\_transactions.py。核心是EODTransactionCheckTask，其中依然是etl。

Extraction部分有三个，分别是TRXORCTradesQuery，TRXABNTradesQuery以及TRXABNPreviousDayTransactions。

TRXORCTradesQuery的sql是查询orc\_trades table，不知道为啥这个query写的特别长，里面有一个nfx alias的应该不用管了，因为nfx都关闭了。不过这边根据price的不同，qty也是不会合并的，但email里面的file或db table abn\_transaction\_breaks就没有price这个column了？至少在后面transform的第一步clean\_data里面，price column还是保留的。Mismatch\_df里面也是有Price的。得到match\_df和mismatch\_df之后，有一个get\_mismatch\_summary，按照一些column group by了，这里面price莫非就不管了（即不同的price也算同一个group）？Ans:不是，如果price不一样，在concat的时候就不会对应同一个key而是在两行里面。所以20201127里面显示wheat不match，虽然qty是一样的，但其实price不同。

TRXABNTradesQuery是查询abn\_transactions table。

TRXABNPreviousDayTransactions也是查询abn\_transactions table，只不过对trans\_dt这个column要求的是查询日期以前的，而前面一个query是查询日期当日的。

Transformer第一步是clean\_data，第二步是将abn\_df和orc\_df做match。第一步中，把数据导入pandas的dataframe，然后做一些基本清理（比如改变列名，把qty=0的去掉，因为这个不计入sum），但要注意，这边更关键的是设置了两个column qty和trade\_value，同时把其它的原先column都设置为这个df的index（相当于二维数组里面的行号）。现在看第二步get\_match\_and\_mismatch\_df()，进去之后先把abn\_df和orc\_df做pd.concat(axis=1)，这种concat就是把两个df的列名都变成新的列名了（即原本各两列，现在新的就变成4列了），然后把concat之后里面NA的部分用0替换掉，然后加一个match column，注意到这一步结束为止，依然是和sql返回出来的没有什么大差别。第二步后，得到match\_df和mismatch\_df之后，有一个get\_mismatch\_summary()，按照一些column group by了。然后按照get\_mismatch\_summary里面group by的column，加上sum之后的column，加上diy添加的两个column，就得到了最后返回的东西。这一步里面最重要的就是dataframe的groupby函数，这边其实就是和sql里面的做法是一样的，所以核心在于prev\_abn（nooo）？

Transform并不return东西，而是里面一大堆yield。注意yield返回的总是generator object，所以实际上关键点在于load(data)这个函数怎么处理，而它就是把data=list(data)，generator变成list罢了。Transformer yield一堆大杂烩给load，甚至包括header，空行之类的（用来提示是next section）。还有就是如果你在load()函数设置断点，会很明显看到它是在等data yield过来。

注：关键还是在concat那一步，当把abn和orc的数据分别从sql拿出来后，concat就是根据给的key column进行match，我们把match上的就类似扔了？没match上的就是一个里面有而另一个里面没有的数据，放在了unmatched里面，如果恰好只是一方多出来来了，那么会看到另一方在unmatched summary里面是0，否则两方都不是0。

Loader就还是把breaks加载到db table abn\_transaction\_breaks以及保存到文件'/mnt/bunker/dev/reconciliation/US/transaction\_check/202011/TransactionsCheck\_20201127.csv'

里面。

check\_eod\_ expirations.py。首先注意date是bunker\_utils.previous\_business\_day(country\_code="US")。然后得到'/mnt/bunker/prd/reconciliation/US/expiration\_check/202011/ExpirationCheck\_20201109.csv'为file\_path，db table是"abn\_expiration\_breaks"。

check\_eod\_expirations.py整个的流程套路依然是preprocess,etl,postprocess。

Preprocess就是把abn\_expiration\_breaks表里面所有这个date(实际上是前一个工作日)和region的数据先删了。

Extractor是check\_expirations.ExpirationCompositeExtractor，在query.py里面。这个复合db extractor里面由四个sql query组成。

看一下ExpORCTransactionsQuery。它是从orc\_trades和products（这不太重要）和abn\_products和orc\_portfolios四个一起查的，当然orc\_trades其实是最重要的，其它也可以不看。

然后是SingleDayExpirationCheckTransformer。这个实际上就是ExpirationCheckTransformer，其实在expiration.py里面。这个transformer真Tm长。这个transform最开始读取data，这个data有四个部分，里面最重要的其实是abn\_data和orc\_data，也就是前面四个extractor里面后两个抓取出来的东西。还有process\_orc\_row和process\_abn\_row，这两个实际上是作为lambda函数传给self.container，因为它以lambda函数作为value类型。这个transform最后返回的就是report，是一个collections.defaultDict，从str->list。其中key有match和mismatch两种。

最后是ExpirationLoader。实质上就是BreakLoader，在expiration.py里面。拿过来的data说实话就是已经像db里面的row差不多了。Load的file\_path就是前面说的那个。除了file loader还有一个db loader，会把break\_data load进去abn\_expiration\_breaks。所以说email里面的File和db里面的这个table内容是一样的，至少就expirations break来说

A picture containing graphical user interface

Description automatically generated

这次是position breaks。是从check\_eod\_positions.py开始。核心是这个EODPositionCheckTask。最后发现的breaks是存放到PositionsCheck\_{dt}.csv和abn\_position\_breaks table里面去的。

最开始显示extract，一个是查orc\_position query (for orc positions)，是查product\_aliased\_orc\_positions\_latest这个玩意儿，这个玩意儿又是实际上从orc\_positions\_latest这个view里面查出来的（这个view和orc\_positions有点点区别，似乎是去掉了重复）。另一个是abn\_position query，是从abn\_positions里面查出来的(for abn positions)。

注：注意这里有个问题就是，abn\_product的名字都是怪怪的，可能要从abn\_products table里面找到那个名字。比如select \* from abn\_products where product ~ ‘OZC’。而在abn\_position\_breaks里面，product\_group又变成正常的Corn了。应该还有一个表存了mapping。

注2：abn\_positions表里面的数据是从abn/abn\_sod\_load/load\_abn\_postions.py里面导入的。原始文件是在prd/abn\_positions里面

Orc positions的query里面重点是做了两个left (outer) join。关于join参见book，里面讲的很清楚了。

Left and right outer join差别就在于（左右）哪一张是主表。

注：我有一个困惑，为什么orc\_positions里面根本查不到BRN-CME这个product，后面它又是怎么冒出来的？答案就在coalesce以及case when？||竟然是字符串连接符，怎么设计的…coalesce是SQL 特有的函数。该函数会返回可变参数A 中左侧开始第1个不是NULL的值。终于知道BRN-CME从哪儿来的了，是从products里面查product\_gorup得到的，它的product是BZO

查完就开始transform（其实最重要的是开始的那两个表的数据是怎么进去的）。Match的过程和之前上面看过的check\_transactions很像。

Orc\_service

此次考虑的问题是，orc\_trades table的数据是从orc\_service来的，但不是时时更新？首先TradeTableInfo这个class是在orm.py里面，而用到这个class是在handles/ trade.py里面的TradeHandler class，这里面有一个uploader，这个uploader又是定义在base.py里面的base.TableUploader。而从base.TableUploader里面，又可以看出这个东西实际上是继承自async\_db.TableUploader，这个async\_db=akdb.aio。

Orc\_trades里面的ts是utc时间。

Db里面go to ddl可以看这个table的各种type信息

什么叫materiallized view？就是既有view的那些metadata(query)，又有实在的数据（而不是每次查询的时候再生成）。它的更新应该是通过refresh把？

occ\_market\_share\_view是从occ\_market\_volume等一些创建而来。

Occ\_market\_volumn里面有volumn，不过没有Open interest。两者应该还是有区别的吧

https://www.investopedia.com/ask/answers/050615/what-difference-between-open-interest-and-volume.asp

check\_orc\_base\_instruments\_traders\_auto.py：

核心task是CheckClosingPricesTask。

Extractor又是有四个。positions\_data, instruments\_data, desk\_email\_groups, base\_offsets。以instruments\_data为例，它查询的是orc\_instruments table，而这个table又是orc\_service load进来的。

然后是transformer。把四个data拿过来之后，全都转成dataframe。

然后是一个EmailLoader。

Bunker\_apps的meta.yaml里面要加bbg\_convert

新的mac上面保护键要设置一下，不然每次cmd q都是退出整个remote desktop。

之前的test写得不好…不应该test过程吧

为什么git一部分sync了，一部分没有？似乎不是这个问题。而是说，一个在我这个branch根本不存在的文件，却出现在inv test测试的某个文件的全路径搜索里面。

能否选取两个db中的两个table？似乎不需要。让我之后去看一下怎么sync两个table。

Bbg\_symbol和product不是一对一的关系。所以会有一些closing prices没有得到，因为cache吞掉了。原来是之前的convert逻辑出错了

服了。。因为grep没有用\*号。不是，是因为对.gz的file没有用zgrep。。。我服了

Bunker\_apps.egg-info这个咋回事

一直报test\_apps出错，后来我把souces.txt和entry\_points.txt里面的eod\_mark的东西都删了就好了。对git也没影响，因为egg-info里面的东西应该是不上传的。那么egg-info里面的东西最开始从哪儿来的？

原来在async def里面可以定义async def，有趣

把future transfer account原来是ABN发个邮件

绝了，看来exchange的网页是个学习的好地方。<https://www.cmegroup.com/trading/agricultural/livestock/cme-pork-cutout-futures-and-options.html?gclsrc=aw.ds&&gclid=CjwKCAiA17P9BRB2EiwAMvwNyI3V4JnKd9zwtORIcGSRAoa7LGZfPfNFhUjEqF_4ln018c3_xj8_-RoCafsQAvD_BwE&gclsrc=aw.ds>

在这边<https://www.cmegroup.com/markets/agriculture.html>

对期货产品进行学习，在cme这种地方很合适

https://www.cmegroup.com/education/courses/introduction-to-agriculture/learn-about-agricultural-markets.html

有些地方，要做code refactor的时候，因为不知道究竟有没有问题，所以宁愿往上加东西（哪怕是不合理），也不敢轻易改，这就是问题。Refactor那本书里记得有说过怎么加test先测试自己改了不会有影响，然后再改。

果然改了之后test就过不了，所以应该写test，看看mike怎么写的。

Typing里面，要去掉optional，可以用assert，<https://github.com/python/typing/issues/645>

有的test是测试过程，有的是测试实际behavior。

还有一个singleton可以看一下

一件件来。先把这个快捷键的问题搞定。现在看起来mod键都不管用，连2.2换排列布置也不行了。前面3节主要是介绍，讲配置从第4节开始。Mike之前应该帮我配置过文件，在~/.config/i3/config位置。4.3键盘绑定是我最关心的。为什么明明keyboard shortcuts里面也设置了退出键，依然是cmd+q就退出？<https://discussions.apple.com/thread/7634486>这个不行啊，虽然cmd+q是不退出了，但在i3里面也没用啊。对键盘来说还有key symbol和key code两个东西，如果经常换不同地区的键盘的话，应该用key code比较好。还可以man xmodmap。实际上还可以设置鼠标（4.4）。又比如4.6的这个设置，按道理说我可以通过按mod+鼠标来拖动窗口（比如pycharm里面的弹窗），但实际上不行。。。4.15是设置变量。第5部分是设置i3bar，虽然config里面也有，但我觉得应该没怎么用。第6部分是说命令，即我们是把命令bind到keypress的。分号是用来chain cmds，即从左至右一个个执行。

Ans：太傻了，最后发现的结果是，我没有把mac里面的keyboard shortcuts的名字的大小写设置对。。

基本键盘配置应该都在/etc/i3/config这边。<https://i3wm.org/docs/userguide.html>

前面很奇怪的反应似乎是类似大写键被按住了。但实际上又没有，我再按了一次shift就好了。

Scp还可以scp到别人的电脑上？

Pycharm可以checkout到branch，也可以checkout到tag。https://www.jetbrains.com/help/pycharm/use-tags-to-mark-specific-commits.html#jump-to-tag

Logging还是什么level都显示。Ans：下面这两种方式是等价的

# logging\_config = copy.deepcopy(bunker\_utils.DEFAULT\_CONFIG)  
 # logging\_config["root"]["level"] = logging.INFO  
 # bunker\_utils.configure\_logging(environment, APP\_NAME, logging\_config)  
 bunker\_utils.configure\_logging(environment, APP\_NAME)  
 logging.getLogger().setLevel("INFO")

核心是root的level被设置为了什么，即logging.getLogger()返回的是root logger。在bunker\_utils.DEFAULT\_CONFIG里面，root的level是被设置为DEBUG的，所以会有之前的种种问题。

再看一遍logging的package

<https://docs.python.org/3/library/logging.html#logging.basicConfig>

最后我还是用之前版本的办法，把default config复制过来，修改level，再传给config\_logging函数https://bitbucket/projects/BNK/repos/bbg\_service/browse/bbg\_service/\_\_main\_\_.py?at=refs%2Ftags%2Fbbg\_service-0.2.9#38

https://www.cnblogs.com/hf8051/p/11727520.html

logging说可以配置来track where msg come from

说logging主要有四个玩意儿

Text

Description automatically generated

Logger是树状结构的？

要设置logging到什么位置，以及消息格式，是在basicConfig()里面设置。

从logging flow里面可以看出，它是不停地call自己的parent logger来处理消息的。

Logger有三个任务。

Logging.getLogger(name)返回一个logger instance，如果没有specify name的话，返回的就是root logger；如果specify name了，只要name一样，返回的总是同一个logger。

某个logger如果没有设置level的话，就会按照parent的来执行，直到递归到root，root的level是warning（所以这么看的话，我的level应该是某个import的module决定了，不过import module里面的logger和我的logger是在同一个树里面么？）。正因为有这样的传递机制，所以没必要每层logger都设置一下，只需要设置top-level logger（当然你也可以认为把传导机制关了）。

Handlers负责把logging msg分发到合适的地方，每个logger可以有多个handlers。Handlers也有setlevel，不过它的setlevel是决定它处理哪个level的消息。

用户一般有三种（其中两种）config的用法。一种是直接人工用api config logger, handler之类的；另一种是用config file or config dict。

如果没有做configuration，那么就要按照python版本来看啥情况，3.2之后是默认一个输出到sys.stderr的warning level的handler，所以warning以下的信息是看不到的（不输出的）。

最后还说了以下library应该怎么设置logging。认为应该把handlers的配置留给library user来设置。

问题：logging.info()这种用法，用的就是root logger？Ans:对的，lib里面\_\_init\_\_的2038行注释讲了

问题：module里面logger的hierarchy是怎么确定的？Ans:The logger name hierarchy is analogous to the Python package hierarchy, and identical to it if you organize your loggers on a per-module basis using the recommended construction logging.getLogger(\_\_name\_\_). That’s because in a module, \_\_name\_\_ is the module’s name in the Python package namespace。试了一下确实是这样，如果是在module里面，print(\_\_name\_\_)显示的就是由点号分隔的、显示hierarchy的名字。

原来java也有logging，看起来接口api还差不多。

Openfigi可以查这些符号。

Pytest的问题。首先的问题是，我们需要什么样的测试？其次的问题是，我们怎么去达成（做到）这些测试？

Pytest官方文档。<https://docs.pytest.org/en/stable/contents.html#toc>。

它先讲了assert，基本上沿袭了python的assert，常用的还有怎么assert一个exception是否发生。

然后讲test fixture，即测试的基本构件。但是为什么我一定要用@pytest.fixture?直接用正常的python function不好么？Ans:原因是dependency injection，把test\_function（被测试函数）需要的东西给它，而不是让它自己创建。多个地方要用同一个fixture的话，创建一个conftest.py，pytest内部写好了回去搜索这个file来找需要的fixture，有一个搜寻顺序在那边。Fixture里面还可以用fixture，也有一个顺序。Fixture还有参数化。

然后是monkeypatching和mocking，猴子补丁。教程里面只介绍了monkeypatching而没讲mocking，虽然两者干的事情应该是一样的。哦因为Mock（至少我们用的）是unittest.mock.Mock，pytest并没有Mock class。Unittset是python built in lib。https://docs.python.org/3/library/unittest.html。还有sentinel也是。

Pytest还提供tmp\_directory这样的自带临时文件夹。

然后讲了stdout/stderr是怎么handle

Unittest <https://docs.python.org/3/library/unittest.html>。

Fixture的功能应该是setup/cleanup for tests。

TestCase是一个individual unit of testing

Mock。After performing an action, you can make assertions about which methods / attributes were used and arguments they were called with。Mock还提供了sentinel, MagicMock, patch等东西。

写test的问题是，路都不会走就得开始跑，本来这种又有异步又和各种外部东西打交道的test就不好写。

之前那个问题什么加了return value也没用，部分是因为说，我该的是新的mock\_db\_reader，而不见得是之前传给load\_bbg\_closing\_prices初始化用的那个mock\_db\_reader

似乎终极原因是fixture function的问题？我把本来function返回的东西复制黏贴直接传进去，就对了？见db\_reader

比如直接import bunker\_apps.bbg.handler.transform as transform不行（因为没有在\_\_init\_\_里面写），那就可以试试from bunker\_apps.bbg.load\_closing\_prices import LoadBBGClosingPrices

看起来一定要先在conda dev repo里面有了之后才能docker build。。。那还是有点慢（当然你更新dockerfile是不用管的，因为它不是在dev repo中的）。

为什么本地跑的时候没有问题，在docker上就会有no module found: from bbg.infra import tyeps?我确实在infra那个文件夹下面没加\_\_init\_\_，难道本地动态build的package和conda build那个不一样？

python -m pip install –editable .是实时更新的么？还是我更新时候需要重新跑一下这个命令？好像是的，跑完之后conda/envs/bunker\_apps/bin里面的东西就更新了。

但为什么每次我改完code之后，import bbg\_service，它里面的东西却是最新的？

分析思路：我每次import 的package是在conda环境下的，即/sandbox/conda/envs/bunker\_apps里面（例如）。最开始创建conda环境的命令是conda create。可以指定储存环境的路径，默认的path是你安装conda的位置下面（而prd里面，你可以看到它就不是在conda下面，那是因为deploy.sh里面create conda env时故意指定了别的路径）。但deploy.sh里面没有activate环境，这是什么时候做的？因为不activate的话，situate workflow还是不能跑的，而且一个box上面有好几个package，不可能每次都人工activate env吧。补充：在dev situate里面有deploy bunker，.sh和prd差不多。

Conda create之后是inv develop和inv bootstrap，这两个都是安装package。Bootstrap安装的是meta.yaml里面的东西，没有这个包本身。第二个是用pip install包本尊，conda里面也有讲。<https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#using-pip-in-an-environment>

为什么我手工装的conda env里面，在/sandbox/conda/envs/bunker\_apps/bin下面就没有conda？Ans:因为没有安装conda-build包。https://docs.conda.io/projects/conda-build/en/latest/install-conda-build.html

注：可以看一下别人解释conda activate是怎么做的。https://stackoverflow.com/questions/48585040/conda-what-happens-when-you-activate-an-environment

还是merge的问题，正确的做法是不是先merge到feature branch然后再push？我觉得应该是的（如果不能自动merge的话）。

Bbg symbol的话题：

https://www.cmegroup.com/month-codes.html

小救星https://www.cmegroup.com/education/courses/introduction-to-futures/understanding-contract-trading-codes.html

Git为什么checkout到master后，会把另一个branch的修改都带过去？

Dataclass怎么设置defualt value？为什么原本跑就没任何问题…看来原本用的不知道是哪家的DBInstrument

当我大部分数据都只有一个，小部分数据有多个的时候怎么处理？用list如果我需要遍历的话

Bbg\_service别忘了之后acc里面每次也要pr

有空找一下project应该怎么structure。

Prometheus:

<https://prometheus.io/docs/introduction/first_steps/>

metric name就是一个time series的名字。还有label。

有几种类型的Metrics。只能单调递增的counter，可增可减的guage，histogram，summary。

一个endpoint就是一个instance？一群endpoint就叫job？

根据监测需求，可以把services分为以下三类<https://prometheus.io/docs/practices/instrumentation/>，对于每一类service应该用什么样的检测metrics，里面都有介绍。

这个是python client library的介绍：<https://github.com/prometheus/client_python>

Prometheus是怎么获取它要检测的数据的？这句话是核心：Prometheus collects metrics from targets by scraping metrics HTTP endpoints。所以是用http获取target发来（pull or push）的数据？

Prometheus跑的时候有一个config file。等会儿，这个是prometheus server的config file把？不是client side的。

我写的应该是对的，打开的那个网页里面的数据并不是增加新条目，而是直接修改原本数据，所以可以看bbgservice\_solace\_recieve\_message有了变化。

如果result info传过来的数目和self.cache里面的数目不对那就有问题了。但现在我们是把空的数据也会传回来的。

还有就是为什么await delete的时候，似乎它又会回到await on\_result\_row的状态？为啥我没有在bbg这边unsubscribe user request的topic？你不subscribe它当然会持续监听，而send过去的信息并不是说有人接收了它就结束了，而是依旧会在那儿。之前因为说之后哪怕改了self.results，但后面也不再用了，因为update\_results是把result作为变量传过去的。所以应该一get完result就unsubscribe。

Node exporter是监测啥来着？

<https://prometheus.io/docs/guides/node-exporter/>

kafka是用来replace solace的？

Python sort的key原来是take one argument，但可以返回多个值（用来进行多pass的比较）

Crypto\_service是用来做crypto交易的，现在是否还用？如果这样的话，基本可以从它窥探一下trading engine的模样。Orc service也是，不过是古老破旧版本。

Crypto\_service里面还读取一个api key文件，大概是为了安全考虑。

然后就是Context class里面，对各个exchange创建对应的service（不过应该是每个instance跑一个吧？对，见create\_service() method）。

我们以bitmex为例。它是一个交易平台<https://www.bitmex.com/?lang=zh-CN>。

BitmexService这个class先是初始化，用的是有账号密码的key\_config，连接到它的官网，连接用的class是BitmexRESTConnector，这是neosocket包里面的东西。它的parent class就叫RESTConnector。

注：因为用的是交易平台的rest api <https://restfulapi.net>?

一个简单的rest介绍<https://www.ruanyifeng.com/blog/2011/09/restful.html>

有一本书叫 rest实战

不过后面还连了一个websockethandler和bitmexwsconnector，为啥要两个connector？

再看看deribit。Deribit用的是neosocket里面的WSConnector。On\_msg是callback函数，处理从web socket stream里面获得的msg。然后service里面还有各种handler，比如positionhandler。然后这些handler里面都会被pass一个db\_client，这个handler里面会有insert rows到db table的步骤，insert各种东西，position也有，account也有，等等。而我们选择load进去的只是transfer，而没有真实的trade（这一点我们可以通过TradeHandler看到，trades是在deribit\_trades table里面的，而且好像不是实时更新）。

Deribit里面似乎也是采用某种类似solace的topic的获取消息模式，websocket是open source的包裹。

<https://docs.deribit.com/?python#public-get_instruments>

可以看出instrument handler做的事情是Retrieves available trading instruments. This method can be used to see which instruments are available for trading, or which instruments have recently expired.是market data的一部分，而不是下单。

<https://docs.deribit.com/?python#private-get_position>

而position handler做的事情也是获得一个private market data，即用户在某个instrument上面的position。这个数据会被放到deribit\_positions里面。

<https://docs.deribit.com/?python#private-get_account_summary>

account handler里面，get\_accounts也是从deribit那边获得account summary data，然后是每隔一段时间就load一条到deribit\_accounts table里面。至于db table里面的数据分别对应的是deribit传过来的哪一条，见accounts.py里面create\_account\_row函数，它会做一个转换。

<https://docs.deribit.com/?python#private-subscribe>

Trade handler里面有subscribe，这看起来和solace很像。

<https://docs.deribit.com/?python#private-buy>

这个link就是buy的topic。在neosocket里面它也确实implement了，但我没看到具体哪里用了。

需要注意的是，accounts.py里面的account handler里面是load deribit\_transfer table的地方。而具体的是在get\_transfers\_per\_currency()里面，更具体是self.ws\_client.desposites()或self.ws\_client.withdrawl()里面。

<https://docs.deribit.com/?python#private-get_deposits>

然后我们再看一个fireblocks把。

它比较简单，只有一个instrument handler。里面的request是通过rest\_connector来发出的，是直接向fireblocks发送http request。

再看看gdax，现在已经是coinbase pro了。

像gdax没有提供api的话，就是直接用rest的GET之类的东西获得数据，见neosocket/gdax/rest.py里面如account\_history，从这里面获得gdax\_transactions。至于这个http获得的数据怎么转成db table数据，见account.py里面的create\_account\_row()，这是update到gdax\_accounts table的。

再看一下gemini。

Transfer是在transfer\_handler里面处理的，db table是gemini\_transfers。从gemini获取的方式是直接网站上获取(rest api)。至于加载到db gemini\_transfers table里面的row长什么样，见transfer.py里面的create\_transfer\_row()函数。

看一下sync\_db.py。

既然crypto journal关心的是transfer（当然实质上应该也是transaction），那么也得明白一下在crypto coin的意义下，transfer是什么。

Orc\_service:

进入点是在cli.py。orc\_client就是和orc相连的玩意儿。整个orc\_service和db，solace和orc相联系。然后设置为基本配置之后就是service.start了。Start也不是重点，主要是OrcService里面初始化的各种handler开始处于callback状态了。有trade，instrument和两个portfolio的handler，我不知道它们在干啥…每一个handler创建完后，都是让orc\_client来register callback，大概是传递过去？ioloop是tornado的东西，然后把flush这些handler拿到的数据到db里面是靠加入periodicCallback来做，每隔一秒做一次。Upload到db实质上call的是handler的flush\_db方法，而这个方法是base里面的抽象方法，实质上在每个handler里面个性化的是uploader（包含了不同handler如instrument, portfolio, trade）要upload到的目的地db table信息。

然后每个handler都老长老长的，但实质上传给orc\_client的callback只是里面的一个method，比如对instrument来说是on\_instrument\_feed。

注：orc总不会用的也是solace了吧？从asyncOrcClient里面来看，用的是tcp连接，连了之后获得一个self.stream，用这个stream来write。信息传输的数据类型是用class表示的。

然后handler里面还有down\_instrument方法，这个是在service的prepare step被call的，这是干啥用的？也是从orc获取某个东西。

此次考虑的问题是，orc\_trades table的数据是从orc\_service来的，但不是时时更新？首先TradeTableInfo这个class是在orm.py里面，而用到这个class是在handles/ trade.py里面的TradeHandler class，这里面有一个uploader，这个uploader又是定义在base.py里面的base.TableUploader。而从base.TableUploader里面，又可以看出这个东西实际上是继承自async\_db.TableUploader，这个async\_db=akdb.aio。

Orc\_trades里面的ts是utc时间。

<https://www.nasdaq.com/solutions/nasdaq-futures-nfx>。看起来nasdaq future exchange已经没了？

TODO:安装homebrew。然后安装sublime。

Conda env删了之后却还在，可能是因为当初安装的位置就不对。我安装在了~/.conda/envs下面。

https://stackoverflow.com/questions/40180652/condavalueerror-value-error-prefix-already-exists/40435573

wedbush:

situate: middleoffice/wedbush

有趣，遇到这个问题

https://stackoverflow.com/questions/25609153/paramiko-error-reading-ssh-protocol-banner

pip不存在就试试pip3，没有权限就试试pip3 –user。

Ice and akuna volumes是在send\_ice\_volumes.py里面。Situate名称是load\_and\_send\_ice\_volumns。另一个相关script是load\_ice\_volumns.py。

先看一下load\_ice\_volumns.py。核心是ICEVolumnTask。相关的db table是ice\_volumns。

pdb跳出loop可以用until

https://stackoverflow.com/questions/6731119/python-pdb-skip-loop

找ip地址

<https://opensource.com/article/18/5/how-find-ip-address-linux>

似乎第一个是正确的

Position Bridge:

和portfolio bridge似乎是兄弟。

i think you might need to -u 1000:1000 when you docker create to get the logs to be owned by the akuna userhttps://zh.wikipedia.org/wiki/日志轮替

umount的话要用sudo umount dir。

我们的机器用的文件系统的协议似乎是CIFS？<https://cifs.com>。因为在mount里面我们的cmd有cifs

<https://linuxize.com/post/how-to-mount-cifs-windows-share-on-linux/>

<https://linux.die.net/man/8/mount> 搜索cifs就看到了，在-t type里面

https://www.dell.com/community/入门级和中端/分享-CIFS和NFS的区别/td-p/6934849

因为是distributed file system，所以我可以理解为什么account locked了，因为读取文件过程中到处都是authentication。

https://linux.die.net/man/8/mount.cifs

看起来global和nonlocal的区别在于，一个是用在function套function，一个是用在module套function。都是为了告诉一个function去哪儿refer这个变量（module或外层function）

设置port forwarding

https://www.hellotech.com/guide/for/how-to-port-forward

<https://www.whatismyip.com>

hdal client是什么下次可以仔细看看，我猜是包装了一个solace client在里面？是在hdal package里面

load\_ice\_volumes.py

数据似乎是放在/data下面。

核心是ICEVolumeTask。它先是从ice\_products表里面取出查询日的product。

在extractor里面似乎是从多个地方（比如db，h5）拿数据，比如/data/derived/chrysalis/\*/ice\_md\*/{:%Y%m%d}.h5。

Transform挺简单，就是把数据变成db table row的形式，load到ice\_volumes表。

Brew可以安装的好用工具：

<https://orianna-zzo.github.io/AllinOne-html/blog/2018-01/mac%E5%B0%8F%E7%99%BD%E6%8E%A2%E7%B4%A2%E8%AE%B01-%E5%9F%BA%E6%9C%AC%E8%AE%BE%E7%BD%AE%E4%B8%8E%E5%B8%B8%E7%94%A8%E8%BD%AF%E4%BB%B6/>

搜狗好像还是不能安装。

brew install –cask

https://brew.sh/

Svcutil包裹解析。

以build\_signal\_future为例，它用的是python的AsyncExitStack和signal library，以及asyncio.AbstractEventLoop。关键是看看asyncio.AbstractEventLoop.create\_future，以及add\_signal\_handler等一系列方法。这里的signal就是常规的signal.SIGINT, signal.SIGTERM之类的东西。

处理build\_asyncio\_error\_future也是类似的，用的是loop.set\_exception\_handler，这是asyncio封装好的。

不过没看懂，为什么在wait\_for\_interruption里面又重新call了一次build\_signal\_future和build\_asyncio\_error\_future

Metrics需要

Grafana

Consul

Docker

https://github.com/prometheus/prometheus/wiki/Default-port-allocations

为什么需要consul的核心原因是因为prometheus是要export http port的，于是就要管理这个寻址的事情。