

# 期权零基础入门

【vn.py全实战进阶系列】

## 课时14 – 获取期权行情

主讲人：陈皓伦  
www.vnpy.com

00:00 / 13:43

## 期权零基础入门-课时14-获取期权行情

5 0

倍速设置 1X

2020-02-14 15:18:39

### 本集要点

- 运行ScriptTader的交互式模式，四步前期准备
- 在jupyter notebook里做实操获取实时行情
- Jupyter Notebook的demo下载链接：[https://vnpy-weixin.oss-cn-shanghai.aliyuncs.com/option-start/14\\_demo.ipynb](https://vnpy-weixin.oss-cn-shanghai.aliyuncs.com/option-start/14_demo.ipynb)
- 本课程提问请至vn.py论坛，提问！<<点击

### 完整文字版

欢迎来到vn.py全实战进阶系列——期权零基础入门课程。今天是我们的第14节课，在之前的课程中我们已经学过了ScriptTader的基础使用方式。在这一节课里面，我们将会运行ScriptTader的交互式模式。通过这个模式来获取到期权的实时行情数据。

在开始之前我们要有一些前期的准备，整体上是这么四步：

#### 前期准备

安装VN Studio发行版

准备期货（期权）账户

获取CTP柜台服务器信息

完成穿透式接入认证

第一步，安装VN Studio的发行版。这个发行版其实是一套vn.py运行所需的python环境，一般情况下我们不建议大家自己安装anaconda之类的python环境再去装vn.py，这样的安装操作容易遇到各种各样的小问题。所以一般建议初学者就直接用我打包的VN Studio就好了。

第二步，需要准备一个期货账户。当然期货账户通常也不会包括期货、期权的交易功能。注意这个需要是一个实盘账户，因为现在所有的仿真账户都不提供实盘数据。

第三步，获取CTP柜台的服务器信息。这一步可能麻烦一点，需要打个电话给自己的客户经理，通过他问一下对应的期货公司IT，拿到比如说交易服务器地址、行情服务器地址以及对应的经纪商编号等等。

最后则是需要去完成一下穿透式接入的认证。穿透式接入是怎么回事？在我们vn.py的公众号里面有非常详细的一篇文章，给大家一步一步讲了如何去做。反正如果有不会的就去看看对应的文章就好了。>>关于穿透式认证

### 实操：在jupyter里运行ScriptTader交互模式

那么接下来就是我们直接进入到的正式的内容，就是在jupyter里面运行，这里是我截了个图：

#### 在Jupyter中运行

```
In [3]: # 连接期货服务器
engine = init_cli_trading([CtpGateway])
engine.connect_gateway(setting, "CTP")

2020-02-12 15:18:23.103598 交易账户连接成功
2020-02-12 15:18:23.103598 行情服务器连接成功
2020-02-12 15:18:23.103598 交易数据订阅连接成功
2020-02-12 15:18:23.103598 行情服务器连接成功
2020-02-12 15:18:23.103598 交易数据订阅成功
2020-02-12 15:18:23.287612 行情数据接收成功
2020-02-12 15:18:23.287612 行情数据接收成功

In [4]: # 订阅合约
engine.subscribe(["IO2002-C-4000.CFFEX"])

In [5]: tick = engine.get_tick("IO2002-C-4000.CFFEX")
print(tick)
print("合约代码", tick.symbol)
print("最新价", tick.last_price)
print("买一价", tick.ask_price_1)
print("卖一价", tick.bid_price_1)

TickData(gateway_name='CTP', symbol='IO2002-C-4000', exchange=Exchange.CFFEX, datetime=datetime.datetime(2020, 2, 12, 15, 18, 23, 287612), name='IO2002-C-4000', volume=6, open_price=24.6, high_price=25.6, low_price=24.6, pre_close=24.6, bid_price_1=25.2, bid_price_2=, bid_price_3=, bid_price_4=, bid_price_5=, ask_price_1=24.6, ask_price_2=, ask_price_3=, ask_price_4=, ask_price_5=, bid_volume_1=0, bid_volume_2=, bid_volume_3=, bid_volume_4=, bid_volume_5=, ask_volume_1=0, ask_volume_2=, ask_volume_3=, ask_volume_4=, ask_volume_5=)
最新价: 25.4
买一价: 25.4
卖一价: 24.6
```

它是一个静态图，给大家示意一下，就直接进到我们的jupyter notebook开始实操，跑起来。

第一步，打开jupyter之后，加载模块：

```
from vnpy.app.script_trader import init_cli_trading (加载init_cli_trading函数)
```

```
from vnpy.gateway.ctp import CtpGateway (加载接口类)
```

加载后直接运行：shift + 回车

第二步，创建要连接CTP所用的配置字典setting，其中我们要用到哪些字段？

用户名、密码，这两个字段容易理解，就是你开户之后，期货公司给你的用户名和密码。

然后是经纪商代码、交易服务器和行情服务器，这三个信息就需要你联系期货公司的IT获取。

然后还有附的三个信息：产品名称、授权编码和产品信息，那么产品名称和授权编码是需要完成穿透式认证之后才能拿到。而产品信息，对于大部分期货公司来说都直接留空就好了，可能个别有几家是需要填的，但我相信大家遇到的情况可能都是不需要填的。

那么我这边因为用的是上节课有提到过在浙商期货开的一个实盘账户，所以首先我用户名密码不能写在这了，下面我用一个json文件的方式把它给读进来，大家自己在运行的时候直接写在这就好了，因为你又不录视频，不会有别人看到的。

然后相应的经纪商代码，像这边6060以及对应的下面的地址都是浙商期货的地址，如果大家开浙商户直接用就好，开的别的公司户就问一下IT，然后去把它改一下。产品名称和授权编码这边是同样是vn.py已经在浙商完成的一个认证的编码，那么vn.py目前在上海中期、中信期货、申万宏源等等几家比较大的公司也都做了对应的认证，所以大家如果开在某家期货公司话，可以先问一下，说这家公司就是vn.py，能不能直接给我一个产品，就是穿透式认证的产品名称和授权编码，如果能的话，你就拿过来直接用就好了。如果不能的话，你再来麻烦一点去走穿透式认证的流程。

第三个单元格对大家来说是不需要的，大家删了就好了，因为我刚刚提到用户名密码，我不能直接写在视频里，所以我把它放在一个json文件中，这一段代码是通过读取json的方式来读进来。

到第四行，我们就开始连接服务器，连接到CTP了：

```
engine = init_cli_trading([CtpGateway])
```

首先我们用 init\_cli\_trading这个函数来初始化交易接口的连接。其中是传入了一个参数，这个参数就是一个列表，大家可以看到外面是个方括号，列表里面包含的唯一个元素，就是接口的名称CtpGateway。

Q: 你如果要通过多个接口做交易，怎么办？

比如说你可能想要做股指期货和ETF那边的300ETF期权之间的套利，此时可能股指期货这边走CTP就行了，ETF期权可能你需要走比如说中泰的XTP的接口，那么初始化的时候，再加上XtpGateway就行了。

所以这也是vn.py的优势之一，可以比较方便地同时连接多个交易市场，去做相应的套利交易。

完成初始化之后，返回的engine对象就是脚本引擎的实例，此时我就可以调用：[engine.connect\\_gateway \(setting, "CTP"\)](#)

连接某个交易接口，它的第1个参数就是setting字典，第2个参数是接口的名字。一般情况下接口名字就是上面gateway前面几个英文字母的大写：Ctpgateway >> CTP。

那么当你调用完这个函数之后，后面我们输出了一系列的日志信息。那么这些日志信息像什么交易接口、连接成功等等，一步步都是在告诉你后面的脚本引擎在怎样的一步一步去初始化我们和CTP服务器之间的通讯、准备的过程：

1. 第一步就是要连上去，要确保网能通。
2. 第二步要登录，无论是交易还是行情都要登录上去。
3. 第三步我们要确认结算信息，这个是国内期货特有的功能。
4. 最后我们要等的是“合约信息查询成功”这一句话，看到这一句日志输出完了之后，就说明整个初始化过程都已经好了，此时我们就可以来去运行下面要做的事情了，不管是你去订阅实时的行情，还是再往后我们会做的直接去发委托做实盘的交易等等，都需要等到这句日志出来之后再去。

然后我们要做的这件事就是订阅一个期权的行情：

```
engine.subscribe(["IO2002-C-4000.CFFEX"])
```

传入的参数是一个列表（外面是方括号的列表），这里只放了一个元素，但其实你可以放多个元素，这样就可以同时订阅多个行情。

```
engine.subscribe(["IO2002-C-4000.CFFEX"], ["IO2002-C-4050.CFFEX"])
```

比如说像这样就可以同时订阅两个不同行权价的期权的行情。当然你输入更多就可以同时订阅更多。订阅的时候我们传进去的代码，大家注意到它是一个前面为期权合约本身代码，中间加了一个英文的点后面跟着一个CFFEX，这个是我们中金所的英文名称缩写，一定注意我们这里用到这是这种代码格式：前面是合约代码，后面交易所，中间有个点的。这种代码在vn.py中有个专门的名字叫做vt\_symbol。结合了合约本身的代码以及交易所后缀的部分，那么一定要用这样的代码去订阅行情，包括后面我们交易获取，这个行情数据、查询合约等等，都要用这种vt\_symbol才行。

订阅行情之后，远端的CTP的行情服务器，尽管我们这儿看不到，其实已经把最新的期权合约的行情给我们推过来了。推过来之后会缓存在我们的engine脚本引擎里面。此时就可以调用脚本引擎的一些函数去获取对应的数据了。

比如说我们要查询行情的话，就调用：

```
tick = engine.get_tick("IO2002-C-4000.CFFEX")
```

获取合约最新的行情数据。注意以下两点：

1. get\_tick()的时候，输入的参数就不再是上面的列表了，而是一个单一的vt\_symbol的合约代码。
2. 这边输入的合约代码，一定要是上面已经订阅过的。如果你输入一个没订阅过的，比如说我输入个IO2004什么的，那么此时返回的结果就是一个None，因为你没订阅过这个行情嘛，要查询它肯定是获取不到的。

所以一定注意这两点。

那么get完了之后，我们可以print把它显示出来，然后再输入下面对应的信息。这里我就运行它，大家可以看到，首先直接print(tick)的话是这么一大堆的数据。

当然这个数据是包含了tickdata里面所有的字段信息，但是这个信息可能看上去非常不清楚，密密麻麻的很难受。我们就选择一个更加适应人类视觉习惯的方式来把它一行行打印出来：最新价多少、期权代码多少，最新价多少，卖一多少，买一多少，以及当前行情的时间戳是多少……

```
print("期权代码", tick.symbol)
```

```
print("最新价", tick.last_price)
```

```
.....
```

如果距离刚刚获取又过了一会，再刷新一下的话，大家可以看到行情戳就会发生跳动。

所以可以看到现在在jupyter中通过get\_tick查询到的就是市场上当前最新的期权合约的行情信息是什么。

那么在后面我们的课程里面就会要重点依赖到这一套ScriptTrader里面的环境去获取到实时的行情，然后做对应的分析，无论是搜索交易机会，还是去判断和理解各种期权特征等等，都会依赖这一套环境。

所以我这边的是建议大家自己开一个jupyter notebook，然后一步一步的把这边的代码都敲一遍运行一遍，这样如果有什么遗漏的话可以更容易发现。

如果说确实你掉到某个坑里面解决不了的，那么我们整个notebook代码也会在这节课的正文里面提供给大家下载链接，大家下载用来参考。

下一节里面我们还是围绕jupyter notebook，我们要围绕它去把期权的价值分解开来去看：一个是期权的空间价值，一个是期权的时间价值。

OK，那么这节课的内容就先到这里，

收起描述

0条评论

回到首页

最新

已加载全部

众新新课同步更新

发评论

微信569250313