BackTestApi期权回测工具使用指南

# 概述

奇点快速期权交易系统（简称奇点交易系统）是一款快速期权交易柜台系统，并提供交易API和行情API（简称奇点Api）接口来编写策略程序。

BackTestApi期权回测工具支持对“基于奇点期权交易系统API编写的策略”进行回测。

BackTestApi采用与“奇点交易系统”的完全一致的API接口，无缝对接实盘。因此，BackTestApi接口编写的策略程序经过回测后，可以直接用“奇点交易系统API”替换掉BackTestApi就可以进行实盘交易。

需要注意的是：BackTestApi只能用来做回测，不能用来连接真实的奇点交易系统。

# 支持哪些证券品种

目前期权BackTestApi支持上交所和深交所的期权竞价交易业务，支持竞价交易报单类型：

|  |
| --- |
| 限价单（支持上交所和深交所） |
| 市价剩余转限价（仅支持上交所） |
| 市价剩余撤销（支持上交所和深交所） |
| 全额即时限价委托（支持上交所和深交所） |
| 全额即时市价委托（支持上交所和深交所） |
| 对手方最优（仅支持深交所） |
| 本方最优（仅支持深交所） |
| 最优五档成交剩余转撤销（仅支持深交所） |

# 支持哪些语言接口

期权BackTestApi采用与“奇点交易系统”的完全一致的API接口，并且接口文件的名称也完全相同。目前，期权BackTestApi支持Linux，X86和X64平台上的C++，Python（Python ***3.6***版本）语言接口。

目前，BackTestAPI支持平台和编程语言接口：

|  |  |  |
| --- | --- | --- |
| 操作系统 | C++ | Python（**只支持*3.6*版本**） |
| 64位Linux | √ | √ |
| 32位Windows | √ | √ |
| 64位Windows | √ | √ |

C++接口文件列表如下：

|  |  |
| --- | --- |
| 文件名 | 详情 |
| TORATstpSPTraderApi.h | 包含交易相关的请求和回调接口 |
| TORATstpSPMdApi.h | 包含获取行情相关的请求和回调接口 |
| TORATstpSPUserApiStruct.h | 包含了所有用到的数据结构。 |
| TORATstpSPUserApiDataType.h | 包含了所有用到的数据类型。 |
| sptraderapi.dll, sptraderapi.lib (windows)  libsptraderapi.so (linux) | 交易部分的动态链接库和静态链接库 |
| spmdapi.dll, spmdapi.lib (windows)  libspmdapi.so (linux) | 行情部分的动态链接库和静态链接库 |
| libtraderapi.dll (windows)  libtraderapi.so (linux) | 基础动态链接库 |

Python接口文件列表如下：

|  |  |
| --- | --- |
| 文件名 | 详情 |
| spmdapi.py | 行情相关Python接口 |
| \_spmdapi.pyd(windows)  \_spmdapi.so(linux) | 行情Python转换C++接口组件 |
| sptraderapi.py | 交易相关Python接口 |
| \_sptraderapi.pyd(windows)  \_sptraderapi.so(linux) | 交易Python转换C++接口组件 |
| sptraderapi.dll(windows)  libsptraderapi.so (linux) | 交易部分的动态链接库 |
| spmdapi.dll(windows)  libspmdapi.so (linux) | 行情部分的动态链接库 |
| libtraderapi.dll (windows)  libtraderapi.so (linux) | 基础动态链接库 |

# 如何快速开始回测（C++）

为了快速开始上手回测，本节将会介绍一个简单例子demo\_option，来演示如何编写和执行一个C++语言编写的回测程序。

## 回测前必须知道的信息

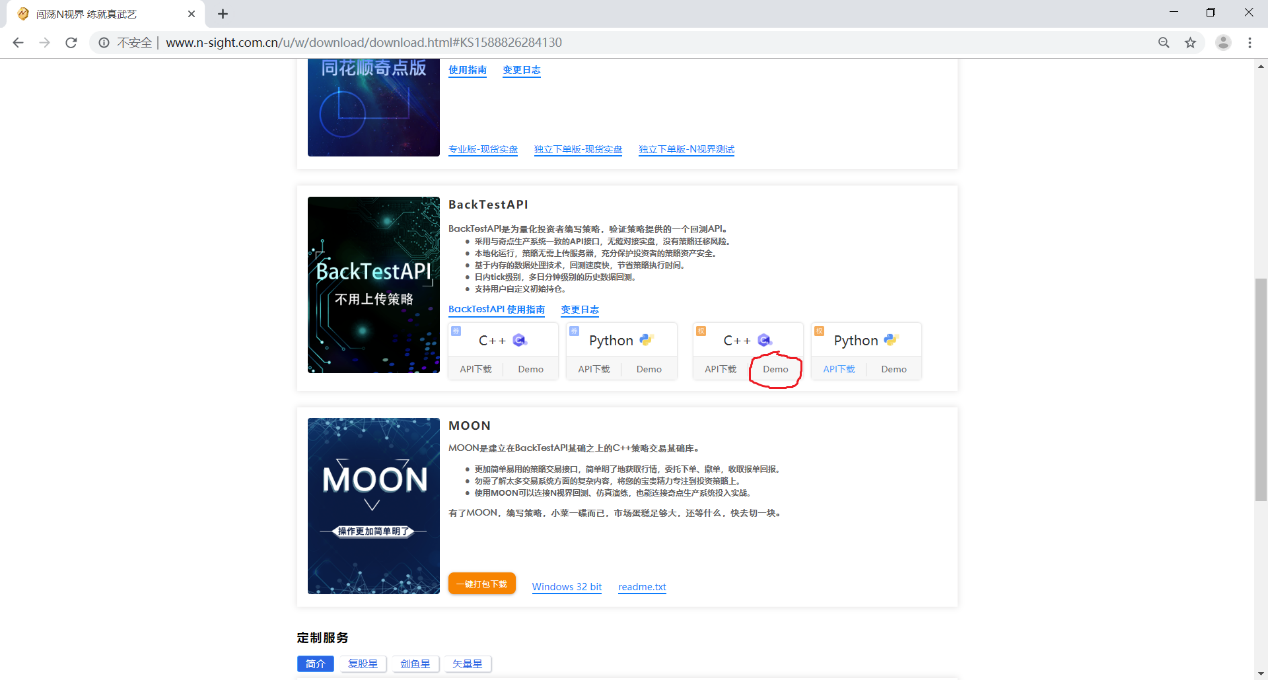
开始回测前，必须要知道以下信息

|  |  |
| --- | --- |
| * BackTestApi只能用来做回测，不能代替生产的API去连接生产交易系统。 * 编写回测程序时，必须同时使用BackTestApi工具中的spmdapi和sptraderapi，不能单独使用其中一个。 * 回测服务器的地址是**tcp://210.14.72.18:26666；**用户必须使用回测sptraderapi的RegisterFront函数来注册此地址去连接上回测服务器。 * 目前期权回测服务允许从**20190301**开始至今的时间段，其他时间段目前暂不支持。因此配置的回测时间段必须在规定的时间段内，且回测开始日期和结束日期都需是有效交易日。 * 如果只进行单个交易日的回测，则收到的行情为秒级行情；如果进行多个交易日的回测，则收到的行情为分钟级行情。 * 在回测过程中，持有的期权合约持仓会在此合约最后交易日日终，会根据当日合约标的收盘价来进行自动行权，产生的盈亏会计入投资者可用资金中。 * 每个用户都有自己特有的回测交易账号和密码，用户必须通过BackTestApi中的sptraderapi的ReqUserLogin接口输入交易账号和密码才能成功登录到回测服务器上。获取自己的交易账号和密码，需要登录N视界网站<http://www.n-sight.com.cn>，点击个人中心按钮去进入个人信息页面，然后在个人信息页面中点击交易密码显示按钮获取账号密码（如下图所示）。  |  | | --- | |  | |

## 开始回测

为了快速开始上手回测，本节将会介绍一个windows平台上的简单示例（BackTestAPI\_demo\_option\_c.zip），来演示如何使用期权BackTestApi去编写和执行一个C++语言编写的回测程序。

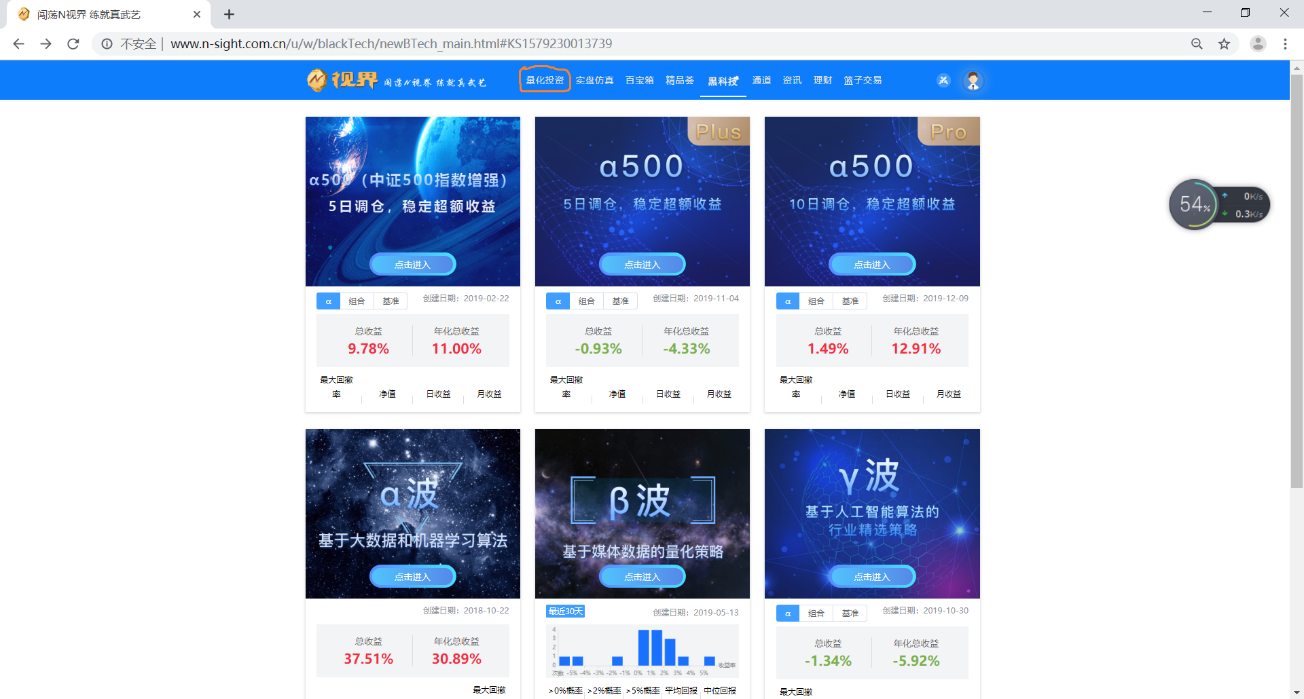
示例下载，可以通过登录N视界网站<http://www.n-sight.com.cn>，并访问百宝箱页面<http://www.n-sight.com.cn/u/w/download/download.html>，然后点击期权BackTestApi的C++ Demo按钮去下载（如下图所示）。



## 查看回测结果

运行完示例回测程序，可以通过登录N视界网站<http://www.n-sight.com.cn>，来查询本次回测结果。如下面五张图，来演示如何查询回测结果。

1. 登录N视界网站成功后，在主页上点击“量化投资“按钮。



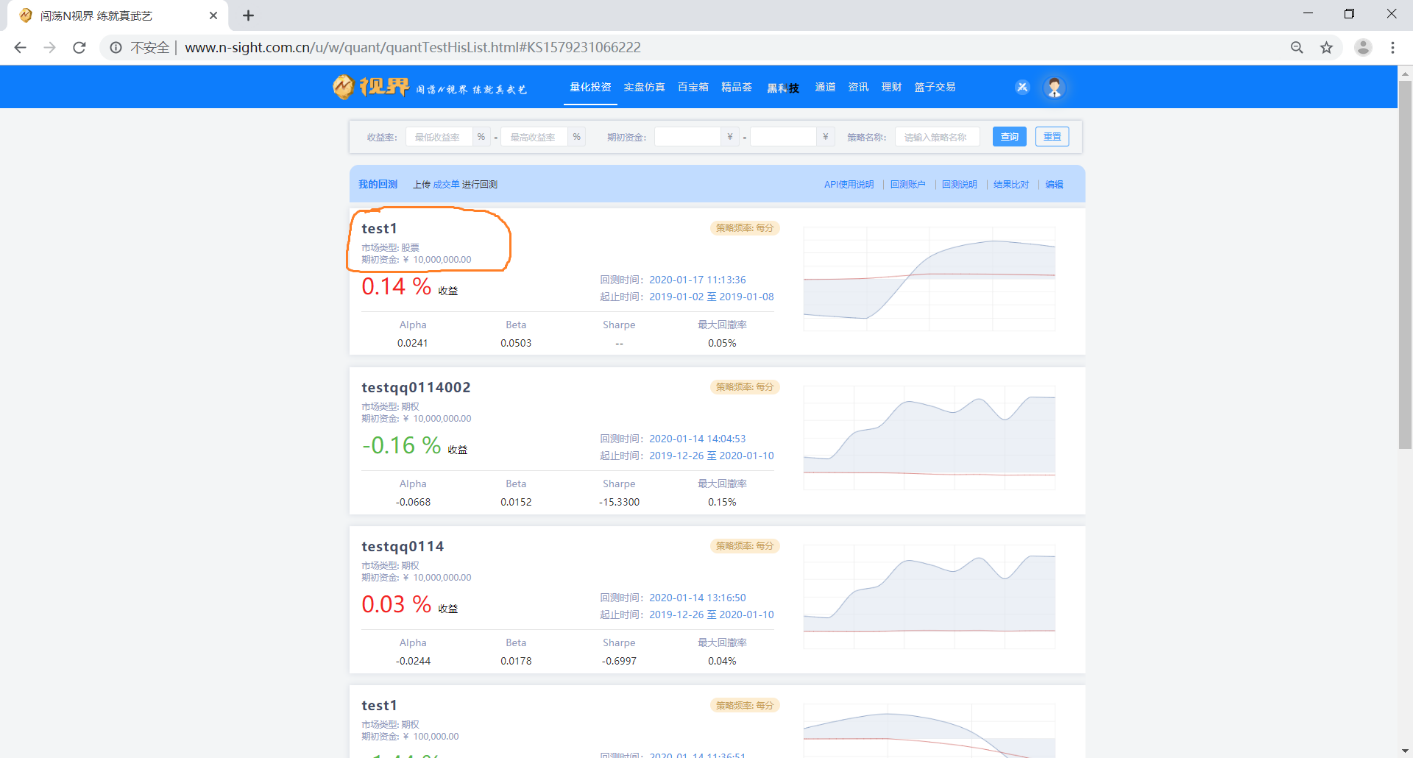
1. 点击“编程式策略“按钮。

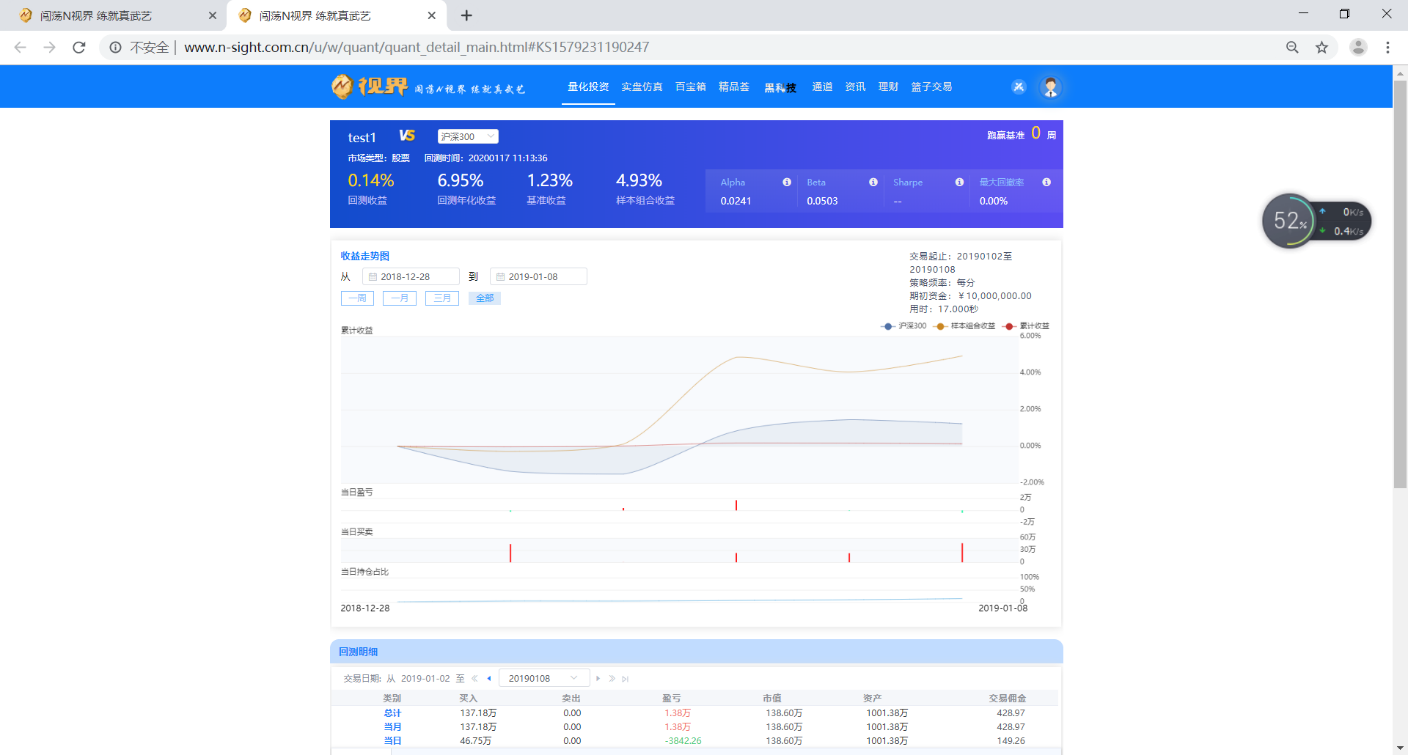


1. 点击“进入我的回测“按钮。



1. 点击某次回测来查看回测结果的详细信息。





## 如何编写回测程序

在本节，将会以demo\_option程序为例(BackTestAPI\_demo\_option\_c.zip附件中有demo\_option.cpp源码)，演示一下如何编写一个简单回测程序。

### 设置回测初始化信息

首先需要在traderapi.ini配置文件中设置回测的初始化信息。traderapi.ini配置说明如下：

在traderapi.ini配置文件中，必须要配置行情起始日期“SubMD\_Begin\_date”和结束日期“SubMD\_End\_date”，这两项配置决定了回测的起止时间段。目前回测服务允许从20190301开始至今的时间段，其他时间段目前暂不支持，因此配置的回测起止时间段必须符合规定的时间段内，并且起始日期和结束日期都需是有效交易日。

|  |
| --- |
| #订阅历史行情的起始日期（必须）  SubMD\_Begin\_date=20190305  #订阅历史行情的结束日期（必须）  SubMD\_End\_date=20190308  #用户设置期权账户的初始资金（可选）。如果不设置则使用默认金额10000000  OptionInitMoney =100000  ##用户期权初始持仓设置：各个持仓之间用分号分隔；定义每个持仓定义用括号来包含，持仓由五个字段来定义(交易所代码,期权代码,仓位数量,持仓多空方向类型,持仓成本)，各个字段之间用逗号分隔。交易所代码1是上证0是深证，持仓多空方向2是多3是空，最后一项持仓成本可省略，如果持仓成本省略，则初始持仓成本=首个交易日的昨结算价\*持仓数量。  InitOptionPosition=(1,10001313,5,2) ; (1,10001417,10,3)  #是否使用缓存行情（可选），如果不设置则默认使用。缓存行情能加速回测过程。  UseMDCache=1 |

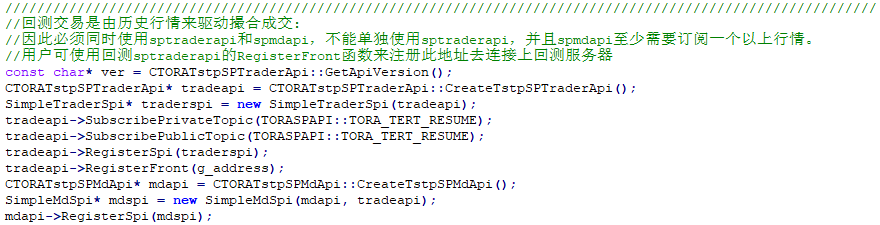
在本demo\_option例子的traderapi.ini文件，可以看到设置回测时间段为20190305到20190308，初始资金设置为10000000，使用缓存行情。

|  |
| --- |
| #订阅历史行情的起始日期  SubMD\_Begin\_date=20190305  #订阅历史行情的结束日期  SubMD\_End\_date=20190308  #用户设置初始资金。如果不设置则使用默认金额10000000  OptionInitMoney=100000  #是否使用缓存行情  UseMDCache=1 |

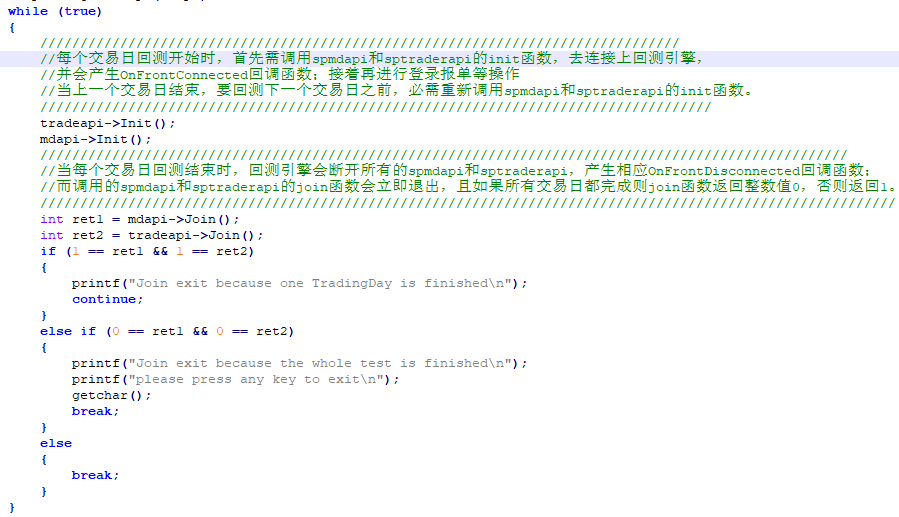
### 编写程序

本节将演示如何编写demo\_option例子程序，具体代码在附件中demo\_option.cpp。

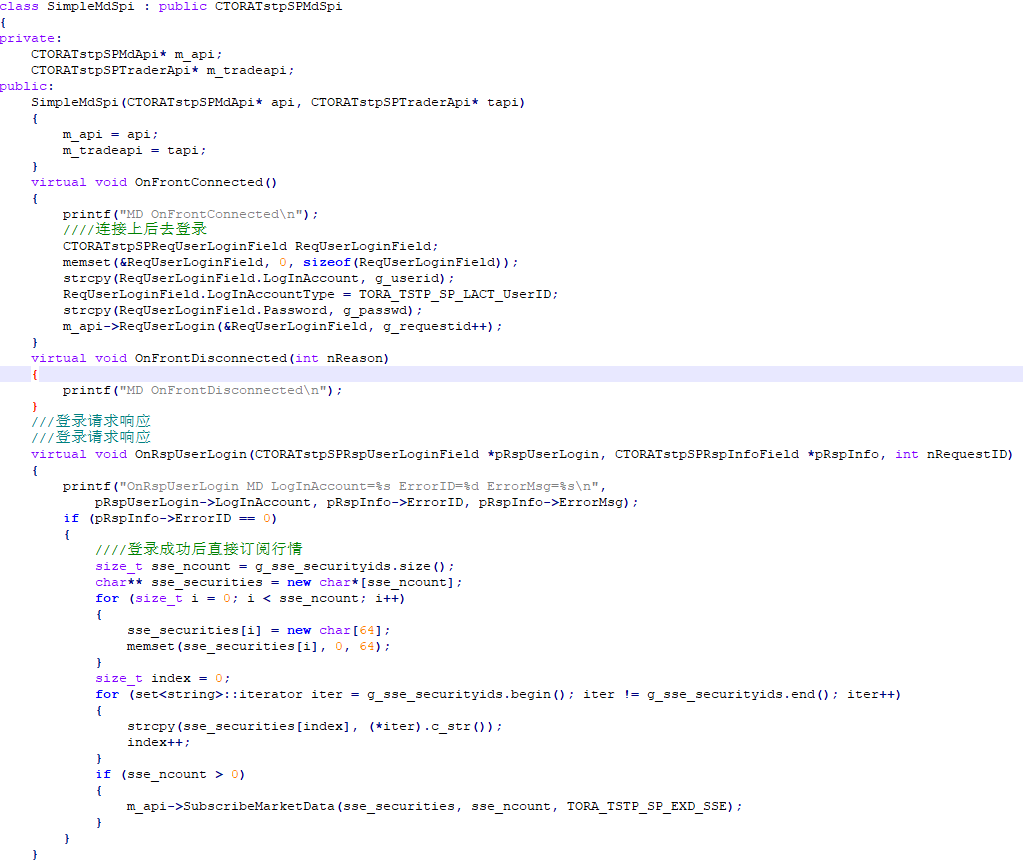
* BackTestApi的spmdapi.dll，sptraderapi.dll，traderapi.dll三个动态库，traderapi.ini，以及回测执行文件demo\_option.exe都必须存放在同一个目录下面。
* 回测程序demo\_option必须同时使用BackTestApi的spmdapi.dll，sptraderapi.dll，不能单独使用其中一个。并且使用sptraderapi的RegisterFront函数来注册回测服务器地址（**tcp://210.14.72.18:26666**）去连接上回测服务器。



* 如果要回测多个交易日，那么在每个交易日开始时，必需重新调用spmdapi和sptraderapi的init函数，去连接上回测引擎，并会产生OnFrontConnected回调函数；而当每个交易日结束时，回测引擎会断开所有的spmdapi和sptraderapi实例，会产生相应OnFrontDisconnected回调函数，而调用的mdapi和traderapi的join函数会立即退出，且如果所有交易日都完成则join函数返回整数值0，否则返回1。



* 回测每个交易日开始时，首先需调用spmdapi和sptraderapi的init函数，去连接上回测引擎，并会产生OnFrontConnected回调函数，接着必须在OnFrontConnected回调函数调用ReqUserLogin函数发送登录请求，并且当调用sptraderapi的ReqUserLogin函数去登录时，可以在登录请求报文中的TerminalInfo字段里填写本次回测的名称。由于回测交易是由行情来驱动的，因此spmdapi在登录成功的OnRspUserLogin回调必须立刻订阅行情。





* 由于回测交易是由行情来驱动的，因此建议务必在CTORATstpSPMdSpi::OnRtnDepthMarketData回调函数中发送报单请求。



录入报单委托使用的函数是 ReqOrderInsert ，其中的核心数据结构是CTORATstpSPInputOrderField，其中的OrderPriceType，TimeCondition和VolumeCondition三个字段决定报单类型:

|  |  |  |  |
| --- | --- | --- | --- |
| **报单类型** | **OrderPriceType** | **TimeCondition** | **VolumeCondition** |
| 限价单（上交所和深交所） | TORA\_TSTP\_SP\_OPT\_LimitPrice | TORA\_TSTP\_SP\_TC\_GFD | TORA\_TSTP\_SP\_VC\_AV |
| 市价剩余转限价（上交所） | TORA\_TSTP\_ SP\_OPT\_AnyPrice | TORA\_TSTP\_ SP\_TC\_IOC | TORA\_TSTP\_SP\_VC\_AV |
| 市价剩余转撤销（上交所和深交所） | TORA\_TSTP\_ SP\_OPT\_AnyPrice | TORA\_TSTP\_SP\_TC\_GFD | TORA\_TSTP\_SP\_VC\_AV |
| 全额即时限价委托（上交所和深交所） | TORA\_TSTP\_SP\_OPT\_LimitPrice | TORA\_TSTP\_ SP\_TC\_IOC | TORA\_TSTP\_SP\_VC\_CV |
| 全额即时市价委托（上交所和深交所） | TORA\_TSTP\_ SP\_OPT\_AnyPrice | TORA\_TSTP\_ SP\_TC\_IOC | TORA\_TSTP\_SP\_VC\_CV |
| 对手方最优（深交所） | TORA\_TSTP\_ SP\_OPT\_BestPrice | TORA\_TSTP\_SP\_TC\_GFD | TORA\_TSTP\_SP\_VC\_AV |
| 本方最优（深交所） | TORA\_TSTP\_ SP\_OPT\_HomeBestPrice | TORA\_TSTP\_SP\_TC\_GFD | TORA\_TSTP\_SP\_VC\_AV |
| 最优五档成交剩余转撤销（深交所） | TORA\_TSTP\_OPT\_FiveLevelPrice | TORA\_TSTP\_ SP\_TC\_IOC | TORA\_TSTP\_SP\_VC\_AV |

### BackTestApi与奇点生产API有哪些不同

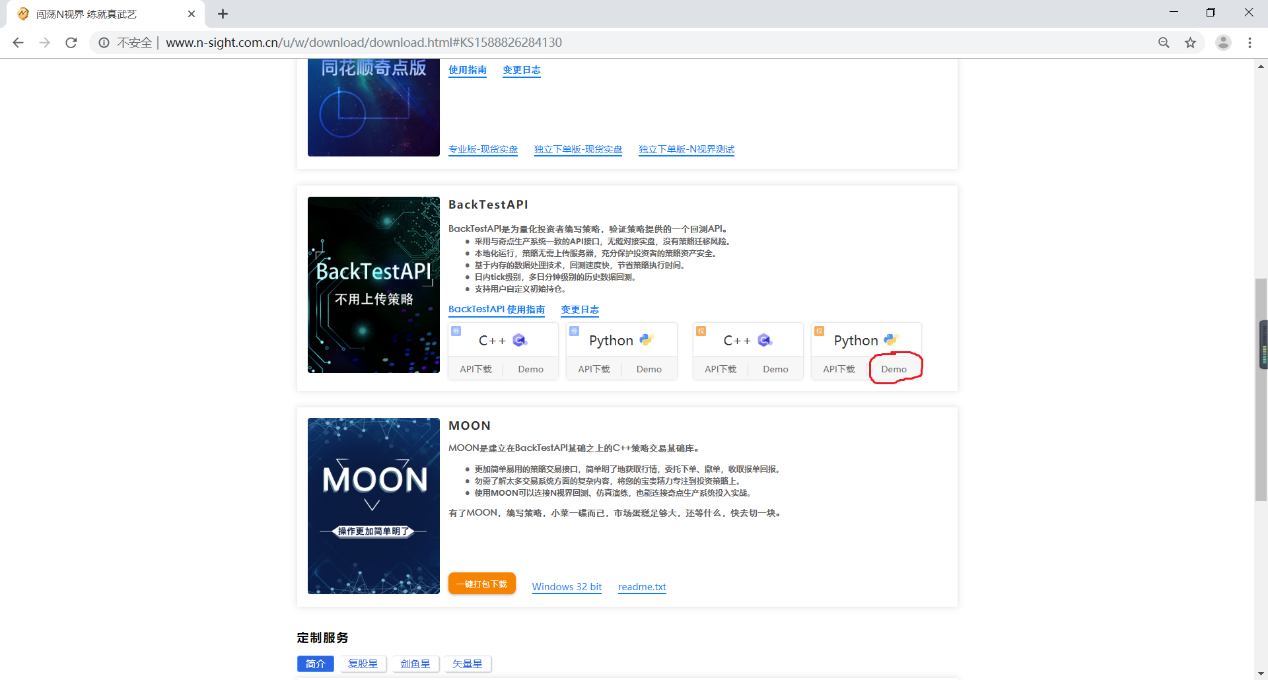
* BackTestApi只能用来做回测，不能代替生产的API去连接生产交易系统。
* 编写回测程序时，必须同时使用BackTestApi工具中的spmdapi和sptraderapi，不能单独使用其中一个；而奇点生产API则可以单独使用其中一个API。
* 需要在traderapi.ini配置文件中设置回测的初始化信息；而奇点生产API不需要配置文件。
* BackTestApi回测多个交易日，那么在每个交易日开始时，必需重新调用spmdapi和sptraderapi的init函数；而当每个交易日结束时，调用的spmdapi和sptraderapi的join函数会立即退出，且如果所有交易日都完成则join函数返回整数值0，否则返回1。
* 调用sptraderapi的ReqUserLogin函数去登录时，可以在登录请求报文中的TerminalInfo字段里填写本次回测的名称。
* 由于回测交易是由行情来驱动的，因此BackTestApi的spmdapi在登录成功的OnRspUserLogin回调必须立刻订阅行情；而奇点生产的spmdapi没有这个要求，只要在登录成功后随时都可以订阅行情。
* 在回测过程中，持有的期权合约持仓会在此合约最后交易日日终，会根据当日合约标的收盘价来进行自动行权，产生的盈亏会计入投资者可用资金中。
* 虽然BackTestApi和奇点生产API的接口相同，但是BackTestApi只支持部分的奇点生产API接口。详情请查看第7章节。

# 如何快速开始回测（Python）

期权BackTestApi支持Python语言接口，需要事先安装***3.6***版本Python。

本节内容将演示用期权BackTestApi的python接口来重新编写上一节的C++ 示例。

示例下载（BackTestAPI\_demo\_option\_python.zip），可以通过登录N视界网站<http://www.n-sight.com.cn>，并访问百宝箱页面<http://www.n-sight.com.cn/u/w/download/download.html>，然后点击期权BackTestApi的Python Demo按钮去下载（如下图所示）。



## 开发环境准备

* 需要安装Python3.6版本
* 安装Python开发IDE，推荐PyCharm

## 如何运行回测程序

traderapi.ini配置文件与C++示例的配置文件相同，查看回测结果方式也与C++示例相同。

使用回测时，traderapi.dll，sptraderapi.dll，spmdapi.dll，traderapi.py，mdapi.py，\_traderapi.pyd，\_mdapi.pyd和traderapi.ini配置文件，都必须和执行程序demo\_option.py放在同一目录下。demo\_python.py是回测程序的源代码。

启动命令如下:

python demo\_option.py tcp://210.14.72.18:26666 user passwd

# 回测如何撮合成交

在回测过程中，报单撮合成交是由行情来驱动。根据行情对手方的五档价格来对价成交，按照一档到五档价格的顺序来依次成交，并且每次成交的数量都会消耗掉该行情档位价格的挂单剩余量。

* 买入时：如果委托价大于等于行情某档卖价格，则成交，成交价为此档位价格，如果委托价小于此档位价格，不能成交，等待更优的行情才能成交；
* 卖出时：如果委托价小于等于行情某档买价格，则成交，成交价为此档位价格，如果委托价大于此档位价格，不能成交，等待更优的行情才能成交。

# 回测BackTestApi支持生产API的哪些接口

虽然BackTestApi和奇点生产API的接口相同，但是BackTestApi只支持部分的奇点生产API接口，其他未支持的接口BackTestApi内部实现为空。

## 支持生产spmdapi哪些接口

BackTestApi的spmdapi.dll和生产行情spmdapi的接口完全相同，但只支持部分生产行情spmdapi的接口。BackTestApi支持的spmdapi的接口如下：

CTORATstpSPMdApi类支持如下接口，其他未支持的接口内部实现为空。

|  |
| --- |
| CreateTstpSPMdApi |
| GetApiVersion |
| RegisterFront |
| Release |
| Init |
| Join |
| RegisterSpi |
| SubscribeMarketData |
| UnSubscribeMarketData |
| ReqUserLogin |
| ReqUserLogout |

CTORATstpSPMdSpi类支持如下回调接口，其他未支持的回调接口不会被调用。

|  |
| --- |
| OnFrontConnected |
| OnFrontDisconnected |
| OnRspUserLogin |
| OnRspUserLogout |
| OnRspSubMarketData |
| OnRspUnSubMarketData |
| OnRtnDepthMarketData |

## 支持sptraderapi那些接口

BackTestApi的sptraderapi.dll和生产交易sptraderapi的接口完全相同，但只支持部分生产sptraderapi的接口。BackTestApi支持的sptraderapi的接口如下：

CTORATstpSPTraderApi类支持如下接口，其他未支持的接口内部实现为空。

|  |
| --- |
| CreateTstpSPTraderApi |
| GetApiVersion |
| Release |
| RegisterFront 用户可使用RegisterFront函数来注册回测服务器地址去连接上该服务器 |
| Init |
| Join |
| RegisterSpi |
| ReqUserLogin 使用用户名和密码通过ReqUserLogin函数去登录上回测服务器 |
| ReqUserLogout |
| ReqOrderInsert |
| ReqOrderAction |
| ReqQryMarketData |
| ReqQrySecurity |
| ReqQryBUProxy |
| ReqQryUser |
| ReqQryInvestor |
| ReqQryShareholderAccount |
| ReqQryTradingAccount |
| ReqQryOrder |
| ReqQryTrade |
| ReqQryPosition |
| ReqQryTradingFee |
| ReqQryInvestorTradingFee |
| ReqQryInvestorMarginFee |
| ReqQryOrderFundDetail |
| ReqQryOrderAction |
| ReqQryInvestorLimitPosition |

CTORATstpSPTraderSpi类支持如下回调接口，其他未支持的回调接口不会被调用。

|  |
| --- |
| OnFrontConnected |
| OnFrontDisconnected |
| OnRspUserLogin |
| OnRspUserLogout |
| OnRspOrderInsert |
| OnRtnOrder |
| OnErrRtnOrderInsert |
| OnRspOrderAction |
| OnErrRtnOrderAction |
| OnRtnTrade |
| OnRspQryMarketData |
| OnRspQrySecurity |
| OnRspQryBUProxy |
| OnRspQryUser |
| OnRspQryInvestor |
| OnRspQryShareholderAccount |
| OnRspQryTradingAccount |
| OnRspQryOrder |
| OnRspQryTrade |
| OnRspQryPosition |
| OnRspQryTradingFee |
| OnRspQryInvestorTradingFee |
| OnRspQryInvestorMarginFee |
| OnRspQryOrderFundDetail |
| OnRspQryOrderAction |
| OnRspQryInvestorLimitPosition |