入门篇

迅投XtQuant FAQ

XtQuant能提供哪些服务

XtQuant是基于迅投MiniQMT衍生出来的一套完善的Python策略运行框架，对外以Python库的形式提供策略交易所需要的行情和交易相关的API接口。

XtQuant运行依赖环境

XtQuant目前提供的库包括Python3.6、3.7、3.8版本，不同版本的python导入时会自动切换。
在运行使用XtQuant的程序前需要先启动MiniQMT客户端。

版本信息

- v1.0.0
  - 初稿
- v1.0.2 - 2020-10-14
  - 持仓结构添加字段
  - 投资备注相关修正
- v1.0.3 - 2020-10-21
  - 添加信用交易相关委托类型（order_type）枚举
  - 调整XtQuant运行依赖环境说明，更新多版本支持相关说明
- v1.0.4 - 2020-11-13
  - 添加信用交易相关类型定义说明
  - 添加信用交易相关接口说明
  - 添加异步撤单委托反馈结构说明
  - 添加下单失败和撤单失败主推结构说明
  - 添加订阅和反订阅接口
  - 添加创建API实例，注册回调类，准备API环境，创建连接，停止运行，阻塞进程接口说明
  - 调整API接口说明
    - 将接口细分为"系统设置接口"，“操作接口”，“查询接口”，"信用相关查询接口"，“回调类”等五类
    - 接口返回“None”修改为“无”
    - 去掉回调类接口中的示例
    - 添加“备注”项
  - 所有“证券账号”改为“资金账号”
  - 英文“,”调整为中文“，”
  - 示例代码中增加XtQuant API实例对象，修正没有实例，直接调用的错误
  - 添加股票异步撤单接口说明，将原股票撤单修改为股票同步撤单

快速入门

创建策略

```
#coding=utf-8
from xtquant.xttrader import XtQuantTrader, XtQuantTraderCallback
from xtquant.xtquant import StockAccount
from xtquant import xtconstant


class MyXtQuantTraderCallback(XtQuantTraderCallback):
    def on_disconnected(self):
        """
        连接断开
        :return:
        """
        print("connection lost")
    def on_stock_order(self, order):
        """
        委托回报推送
        :param order: XtOrder对象
        :return:
        """
        print("on order callback:")
        print(order.stock_code, order.order_status, order.order_sysid)
    def on_stock_asset(self, asset):
        """
        资金变动推送
        :param asset: XtAsset对象
        :return:
        """
        print("on asset callback")
        print(asset.account_id, asset.cash, asset.total_asset)
    def on_stock_trade(self, trade):
        """
        成交变动推送
        :param trade: XtTrade对象
        :return:
        """
        print("on trade callback")
        print(trade.account_id, trade.stock_code, trade.order_id)
    def on_stock_position(self, position):
        """
        持仓变动推送
        :param position: XtPosition对象
        :return:
        """
        print("on position callback")
        print(position.stock_code, position.volume)
    def on_order_error(self, order_error):
        """
        委托失败推送
        :param order_error:XtOrderError 对象
        :return:
        """
        print("on order_error callback")
```

```
        print(order_error.order_id, order_error.error_id, order_error.error_msg)
    def on_cancel_error(self, cancel_error):
        """
        撤单失败推送
        :param cancel_error: XtCancelError 对象
        :return:
        """
        print("on cancel_error callback")
        print(cancel_error.order_id, cancel_error.error_id,
cancel_error.error_msg)
    def on_order_stock_async_response(self, response):
        """
        异步下单回报推送
        :param response: XtOrderResponse 对象
        :return:
        """
        print("on_order_stock_async_response")
        print(response.account_id, response.order_id, response.seq)


if __name__ == "__main__":
    print("demo test")
    # path为mini qmt客户端安装目录下userdata_mini路径
    path = 'D:\\迅投极速交易终端 睿智融科版\\userdata_mini'
    # session_id为会话编号，策略使用方对于不同的Python策略需要使用不同的会话编号
    session_id = 123456
    xt_trader = XtQuantTrader(path, session_id)
    # 创建资金账号为1000000365的证券账号对象
    acc = StockAccount('1000000365')
    # 创建交易回调类对象，并声明接收回调
    callback = MyXtQuantTraderCallback()
    xt_trader.register_callback(callback)
    # 启动交易线程
    xt_trader.start()
    # 建立交易连接，返回0表示连接成功
    connect_result = xt_trader.connect()
    print(connect_result)
    # 对交易回调进行订阅，订阅后可以收到交易主推，返回0表示订阅成功
    subscribe_result = xt_trader.subscribe(acc)
    print(subscribe_result)
    stock_code = '600000.SH'
    # 使用指定价下单，接口返回订单编号，后续可以用于撤单操作以及查询委托状态
    print("order using the fix price:")
    fix_result_order_id = xt_trader.order_stock(acc, stock_code,
xtconstant.STOCK_BUY, 200, xtconstant.FIX_PRICE, 10.5, 'strategy_name',
'remark')
    print(fix_result_order_id)
    # 使用订单编号撤单
    print("cancel order:")
    cancel_order_result = xt_trader.cancel_order_stock(acc, fix_result_order_id)
    print(cancel_order_result)
    # 使用异步下单接口，接口返回下单请求序号seq，seq可以和on_order_stock_async_response
的委托反馈response对应起来
    print("order using async api:")
    async_seq = xt_trader.order_stock(acc, stock_code, xtconstant.STOCK_BUY,
200, xtconstant.FIX_PRICE, 10.5, 'strategy_name', 'remark')
    print(async_seq)
    # 查询证券资产
```

```
    print("query asset:")
    asset = xt_trader.query_stock_asset(acc)
    if asset:
        print("asset:")
        print("cash {0}".format(asset.cash))
    # 根据订单编号查询委托
    print("query order:")
    order = xt_trader.query_stock_order(acc, fix_result_order_id)
    if order:
        print("order:")
        print("order {0}".format(order.order_id))
    # 查询当日所有的委托
    print("query orders:")
    orders = xt_trader.query_stock_orders(acc)
    print("orders:", len(orders))
    if len(orders) != 0:
        print("last order:")
        print("{0} {1} {2}".format(orders[-1].stock_code,
orders[-1].order_volume, orders[-1].price))
    # 查询当日所有的成交
    print("query trade:")
    trades = xt_trader.query_stock_trades(acc)
    print("trades:", len(trades))
    if len(trades) != 0:
        print("last trade:")
        print("{0} {1} {2}".format(trades[-1].stock_code,
trades[-1].traded_volume, trades[-1].traded_price))
    # 查询当日所有的持仓
    print("query positions:")
    positions = xt_trader.query_stock_positions(acc)
    print("positions:", len(positions))
    if len(positions) != 0:
        print("last position:")
        print("{0} {1} {2}".format(positions[-1].account_id,
positions[-1].stock_code, positions[-1].volume))
    # 根据股票代码查询对应持仓
    print("query position:")
    position = xt_trader.query_stock_position(acc, stock_code)
    if position:
        print("position:")
        print("{0} {1} {2}".format(position.account_id, position.stock_code,
position.volume))
    # 阻塞线程，接收交易推送
    xt_trader.run_forever()
```

进阶篇

XtQuant运行逻辑

XtQuant封装了策略交易所需要的Python API接口，可以和MiniQMT客户端交互进行报单、撤单、查询资产、查询委托、查询成交、查询持仓以及收到资金、委托、成交和持仓等变动的主推消息。

XtQuant数据字典

交易市场(market)

枚举变量名	值	含义
xtconstant.SH_MARKET	0	上海市场
xtconstant.SZ_MARKET	1	深圳市场

账号类型(account_type)

枚举变量名	值	含义
xtconstant.SECURITY_ACCOUNT	2	证券账号
xtconstant.CREDIT_ACCOUNT	3	信用账号

委托类型(order_type)

枚举变量名	值	含义
xtconstant.STOCK_BUY	23	证券买入
xtconstant.STOCK_SELL	24	证券卖出
xtconstant.CREDIT_BUY	23	担保品买入
xtconstant.CREDIT_SELL	24	担保品卖出
xtconstant.CREDIT_FIN_BUY	27	融资买入
xtconstant.CREDIT_SLO_SELL	28	融券卖出
xtconstant.CREDIT_BUY_SECU_REPAY	29	买券还券
xtconstant.CREDIT_DIRECT_SECU_REPAY	30	直接还券
xtconstant.CREDIT_SELL_SECU_REPAY	31	卖券还款
xtconstant.CREDIT_DIRECT_CASH_REPAY	32	直接还款
xtconstant.CREDIT_FIN_BUY_SPECIAL	40	专项融资买入
xtconstant.CREDIT_SLO_SELL_SPECIAL	41	专项融券卖出
xtconstant.CREDIT_BUY_SECU_REPAY_SPECIAL	42	专项买券还券
xtconstant.CREDIT_DIRECT_SECU_REPAY_SPECIAL	43	专项直接还券
xtconstant.CREDIT_SELL_SECU_REPAY_SPECIAL	44	专项卖券还款
xtconstant.CREDIT_DIRECT_CASH_REPAY_SPECIAL	45	专项直接还款

报价类型(price_type)

枚举变量名	值	含义
xtconstant.LATEST_PRICE	5	最新价
xtconstant.FIX_PRICE	11	限价
xtconstant.MARKET_SH_CONVERT_5_CANCEL	42	上海最优五档即时成交剩余撤销
xtconstant.MARKET_SH_CONVERT_5_LIMIT	43	上海最优五档即时成交剩余转限价
xtconstant.MARKET_PEER_PRICE_FIRST	44	深圳对手方最优价格
xtconstant.MARKET_MINE_PRICE_FIRST	45	深圳本方最优价格
xtconstant.MARKET_SZ_INSTBUSI_RESTCANCEL	46	深圳即时成交剩余撤销
xtconstant.MARKET_SZ_CONVERT_5_CANCEL	47	深圳最优五档即时成交剩余撤销
xtconstant.MARKET_SZ_FULL_OR_CANCEL	48	深圳全额成交或撤销

委托状态(order_status)

枚举变量名	值	含义
xtconstant.ORDER_UNREPORTED	48	未报
xtconstant.ORDER_WAIT_REPORTING	49	待报
xtconstant.ORDER_REPORTED	50	已报
xtconstant.ORDER_REPORTED_CANCEL	51	已报待撤
xtconstant.ORDER_PARTSUCC_CANCEL	52	部成待撤
xtconstant.ORDER_PART_CANCEL	53	部撤
xtconstant.ORDER_CANCELED	54	已撤
xtconstant.ORDER_PART_SUCC	55	部成
xtconstant.ORDER_SUCCEEDED	56	已成
xtconstant.ORDER_JUNK	57	废单
xtconstant.ORDER_UNKNOWN	255	未知

XtQuant数据结构说明

资产XtAsset

属性	类型	注释
account_type	int	账号类型，参见数据字典
account_id	str	资金账号
cash	float	可用金额
frozen_cash	float	冻结金额
market_value	float	持仓市值
total_asset	float	总资产

委托XtOrder

属性	类型	注释
account_type	int	账号类型，参见数据字典
account_id	str	资金账号
stock_code	str	证券代码，例如"600000.SH"
order_id	int	订单编号
order_sysid	str	柜台合同编号
order_time	int	报单时间
order_type	int	委托类型，参见数据字典
order_volume	int	委托数量
price_type	int	报价类型，参见数据字典
price	float	委托价格
traded_volume	int	成交数量
traded_price	float	成交均价
order_status	int	委托状态，参见数据字典
status_msg	str	委托状态描述，如废单原因
strategy_name	str	策略名称
order_remark	str	委托备注

成交XtTrade

属性	类型	注释
account_type	int	账号类型，参见数据字典
account_id	str	资金账号
stock_code	str	证券代码
order_type	int	委托类型，参见数据字典
traded_id	str	成交编号
traded_time	int	成交时间
traded_price	float	成交均价
traded_volume	int	成交数量
traded_amount	float	成交金额
order_id	int	订单编号
order_sysid	str	柜台合同编号
strategy_name	str	策略名称
order_remark	str	委托备注

持仓XtPosition

属性	类型	注释
account_type	int	账号类型，参见数据字典
account_id	str	资金账号
stock_code	str	证券代码
volume	int	持仓数量
can_use_volume	int	可用数量
open_price	float	平均建仓成本
market_value	float	市值

异步下单委托反馈XtOrderResponse

属性	类型	注释
account_type	int	账号类型，参见数据字典
account_id	str	资金账号
order_id	int	订单编号
strategy_name	str	策略名称
order_remark	str	委托备注
seq	int	异步下单的请求序号

异步撤单委托反馈XtCancelOrderResponse

属性	类型	注释
account_type	int	账号类型，参见数据字典
account_id	str	资金账号
order_id	int	订单编号
order_sysid	str	柜台委托编号
cancel_result	int	撤单结果
seq	int	异步撤单的请求序号

下单失败错误XtOrderError

属性	类型	注释
account_type	int	账号类型，参见数据字典
account_id	str	资金账号
order_id	int	订单编号
error_id	int	下单失败错误码
error_msg	str	下单失败具体信息
strategy_name	str	策略名称
order_remark	str	委托备注

撤单失败错误XtCancelError

属性	类型	注释
account_type	int	账号类型，参见数据字典
account_id	str	资金账号
order_id	int	订单编号
market	int	交易市场 0:上海 1:深圳
order_sysid	str	柜台委托编号
error_id	int	下单失败错误码
error_msg	str	下单失败具体信息

信用账号资产XtCreditDetail

属性	类型	注释
account_type	int	账号类型，参见数据字典
account_id	str	资金账号
m_nStatus	int	账号状态
m_nUpdateTime	int	更新时间
m_nCalcConfig	int	计算参数
m_dFrozenCash	float	冻结金额
m_dBalance	float	总资产
m_dAvailable	float	可用金额
m_dPositionProfit	float	持仓盈亏
m_dMarketValue	float	总市值
m_dFetchBalance	float	可取金额
m_dStockValue	float	股票市值
m_dFundValue	float	基金市值
m_dTotalDebt	float	总负债
m_dEnableBailBalance	float	可用保证金
m_dPerAssurescaleValue	float	维持担保比例
m_dAssureAsset	float	净资产
m_dFinDebt	float	融资负债
m_dFinDealAvl	float	融资本金
m_dFinFee	float	融资息费
m_dSloDebt	float	融券负债
m_dSloMarketValue	float	融券市值
m_dSloFee	float	融券息费
m_dOtherFare	float	其它费用
m_dFinMaxQuota	float	融资授信额度
m_dFinEnableQuota	float	融资可用额度
m_dFinUsedQuota	float	融资冻结额度
m_dSloMaxQuota	float	融券授信额度
m_dSloEnableQuota	float	融券可用额度
m_dSloUsedQuota	float	融券冻结额度

属性	类型	注释
m_dSloSellBalance	float	融券卖出资金
m_dUsedSloSellBalance	float	已用融券卖出资金
m_dSurplusSloSellBalance	float	剩余融券卖出资金

负债合约StkCompacts

属性	类型	注释
account_type	int	账号类型，参见数据字典
account_id	str	资金账号
compact_type	int	合约类型
cashgroup_prop	int	头寸来源
exchange_id	int	证券市场
open_date	int	开仓日期
business_vol	int	合约证券数量
real_compact_vol	int	未还合约数量
ret_end_date	int	到期日
business_balance	float	合约金额
businessFare	float	合约息费
real_compact_balance	float	未还合约金额
real_compact_fare	float	未还合约息费
repaid_fare	float	已还息费
repaid_balance	float	已还金额
instrument_id	str	证券代码
compact_id	str	合约编号
position_str	str	定位串

融资融券标的CreditSubjects

属性	类型	注释
account_type	int	账号类型，参见数据字典
account_id	str	资金账号
slo_status	int	融券状态
fin_status	int	融资状态
exchange_id	int	证券市场
slo_ratio	float	融券保证金比例
fin_ratio	float	融资保证金比例
instrument_id	str	证券代码

可融券数据CreditSloCode

属性	类型	注释
account_type	int	账号类型，参见数据字典
account_id	str	资金账号
cashgroup_prop	int	头寸来源
exchange_id	int	证券市场
enable_amount	int	融券可融数量
instrument_id	str	证券代码

标的担保品CreditAssure

属性	类型	注释
account_type	int	账号类型，参见数据字典
account_id	str	资金账号
assure_status	int	是否可做担保
exchange_id	int	证券市场
assure_ratio	float	担保品折算比例
instrument_id	str	证券代码

XtQuant API说明

系统设置接口

创建API实例

```
XtQuantTrader(path, session_id)
```

- 释义
  - 创建XtQuant API的实例
- 参数
  - path - str MiniQMT客户端userdata_mini的完整路径
  - session_id - int 与MiniQMT通信的会话ID，不同的会话要保证不重
- 返回
  - XtQuant API实例对象
- 备注
  - 后续对XtQuant API的操作都需要该实例对象
  - 通常情况下只需要创建一个XtQuant API实例
- 示例

```
path = 'D:\\迅投极速交易终端 睿智融科版\\userdata_mini'
# session_id为会话编号，策略使用方对于不同的Python策略需要使用不同的会话编号
session_id = 123456
#后续的所有示例将使用该实例对象
xt_trader = XtQuantTrader(path, session_id)
```

注册回调类

```
register_callback(callback)
```

- 释义
  - 将回调类实例对象注册到API实例中，用以消息回调和主推
- 参数
  - callback - XtQuantTraderCallback 回调类实例对象
- 返回
  - 无
- 备注
  - 无
- 示例

```
# 创建交易回调类对象，并声明接收回调
class MyXtQuantTraderCallback(XtQuantTraderCallback):
    ...
    pass
callback = MyXtQuantTraderCallback()
#xt_trader为XtQuant API实例对象
xt_trader.register_callback(callback)
```

准备API环境

```
start()
```

- 释义
  - 启动交易线程，准备交易所需的环境

- 参数
  - 无
- 返回
  - 无
- 备注
  - 无
- 示例

```
# 启动交易线程
#xt_trader为XtQuant API实例对象
xt_trader.start()
```

创建连接

```
connect()
```

- 释义
  - 连接MiniQMT
- 参数
  - 无
- 返回
  - 连接结果信息，连接成功返回0，失败返回非0
- 备注
  - 该连接为一次性连接，断开连接后不会重连，需要再次主动调用
- 示例

```
# 建立交易连接，返回0表示连接成功
#xt_trader为XtQuant API实例对象
connect_result = xt_trader.connect()
print(connect_result)
```

停止运行

```
stop()
```

- 释义
  - 停止API接口
- 参数
  - 无
- 返回
  - 无
- 备注
  - 无
- 示例

```
#xt_trader为XtQuant API实例对象
xt_trader.stop()
```

阻塞当前线程进入等待状态

```
run_forever()
```

- 释义
  - 阻塞当前线程，进入等待状态，直到stop函数被调用结束阻塞
- 参数
  - 无
- 返回
  - 无
- 备注
  - 无
- 示例

```
#xt_trader为XtQuant API实例对象
xt_trader.run_forever()
```

操作接口

订阅账号信息

```
subscribe(account)
```

- 释义
  - 订阅账号信息，包括资金账号、委托信息、成交信息、持仓信息
- 参数
  - account - StockAccount 资金账号
- 返回
  - 订阅结果信息，订阅成功返回True，订阅失败返回False
- 备注
  - 无
- 示例
  - 订阅资金账号1000000365

```
account = StockAccount('1000000365')
#xt_trader为XtQuant API实例对象
order_id = xt_trader.subscribe(account)
```

反订阅账号信息

```
unsubscribe(account)
```

- 释义
  - 反订阅账号信息
- 参数
  - account - StockAccount 资金账号
- 返回

    - 反订阅结果信息，订阅成功返回True，订阅失败返回False
- 备注
    - 无
- 示例
    - 订阅资金账号1000000365

```
account = StockAccount('1000000365')
#xt_trader为XtQuant API实例对象
order_id = xt_trader.unsubscribe(account)
```

股票同步报单

```
order_stock(account, stock_code, order_type, order_volume, price_type, price,
strategy_name, order_remark)
```

- 释义
    - 对股票进行下单操作
- 参数
    - account - StockAccount 资金账号
    - stock_code - str 证券代码，如'600000.SH'
    - order_type - int 委托类型
    - order_volume - int 委托数量，股票以'股'为单位，债券以'张'为单位
    - price_type - int 报价类型
    - price - float 委托价格
    - strategy_name - str 策略名称
    - order_remark - str 委托备注
- 返回
    - 系统生成的订单编号，成功委托后的订单编号为大于0的正整数，如果为-1表示委托失败
- 备注
    - 无
- 示例
    - 股票资金账号1000000365对浦发银行买入1000股，使用限价价格10.5元, 委托备注为'order_test'

```
account = StockAccount('1000000365')
#xt_trader为XtQuant API实例对象
order_id = xt_trader.order_stock(account, '600000.SH', xtconstant.STOCK_BUY,
1000, xtconstant.FIX_PRICE, 10.5, 'strategy1', 'order_test')
```

股票异步报单

```
order_stock_async(account, stock_code, order_type, order_volume, price_type,
price, strategy_name, order_remark)
```

- 释义
    - 对股票进行异步下单操作，异步下单接口如果正常返回了下单请求序号seq，会收到on_order_stock_async_response的委托反馈
- 参数
    - account - StockAccount 资金账号

    - stock_code - str 证券代码， 如'600000.SH'
    - order_type - int 委托类型
    - order_volume - int 委托数量，股票以'股'为单位，债券以'张'为单位
    - price_type - int 报价类型
    - price - float 委托价格
    - strategy_name - str 策略名称
    - order_remark - str 委托备注
- 返回
    - 返回下单请求序号seq，成功委托后的下单请求序号为大于0的正整数，如果为-1表示委托失败
- 备注
    - 如果失败，则通过下单失败主推接口返回下单失败信息
- 示例
    - 股票资金账号1000000365对浦发银行买入1000股，使用限价价格10.5元，委托备注为'order_test'

```
account = StockAccount('1000000365')
#xt_trader为XtQuant API实例对象
seq = xt_trader.order_stock_async(account, '600000.SH', xtconstant.STOCK_BUY,
1000, xtconstant.FIX_PRICE, 10.5, 'strategy1', 'order_test')
```

股票同步撤单

```
cancel_order_stock(account, order_id)
```

- 释义
    - 根据订单编号对委托进行撤单操作
- 参数
    - account - StockAccount 资金账号
    - order_id - int 同步下单接口返回的订单编号
- 返回
    - 返回是否成功发出撤单指令，0: 成功, -1: 表示撤单失败
- 备注
    - 无
- 示例
    - 股票资金账号1000000365对订单编号为order_id的委托进行撤单

```
account = StockAccount('1000000365')
order_id = 100
#xt_trader为XtQuant API实例对象
cancel_result = xt_trader.cancel_order_stock(account, order_id)
```

股票同步撤单

```
cancel_order_stock_sysid(account, market, order_sysid)
```

- 释义
    - 根据券商柜台返回的合同编号对委托进行撤单操作

- 参数
  - account - StockAccount 资金账号
  - market - int 交易市场
  - order_sysid - str 券商柜台的合同编号
- 返回
  - 返回是否成功发出撤单指令，0: 成功， -1: 表示撤单失败
- 备注
  - 无
- 示例
  - 股票资金账号1000000365对柜台合同编号为order_sysid的上交所委托进行撤单

```
account = StockAccount('1000000365')
market = xtconstant.SH_MARKET
order_sysid = "100"
#xt_trader为XtQuant API实例对象
cancel_result = xt_trader.cancel_order_stock_sysid(account, market, order_sysid)
```

股票异步撤单

```
cancel_order_stock_async(account, order_id)
```

- 释义
  - 根据订单编号对委托进行异步撤单操作
- 参数
  - account - StockAccount 资金账号
  - order_id - int 下单接口返回的订单编号
- 返回
  - 返回撤单请求序号, 成功委托后的撤单请求序号为大于0的正整数, 如果为-1表示委托失败
- 备注
  - 如果失败，则通过撤单失败主推接口返回撤单失败信息
- 示例
  - 股票资金账号1000000365对订单编号为order_id的委托进行异步撤单

```
account = StockAccount('1000000365')
order_id = 100
#xt_trader为XtQuant API实例对象
cancel_result = xt_trader.cancel_order_stock_async(account, order_id)
```

股票异步撤单

```
cancel_order_stock_sysid_async(account, market, order_sysid)
```

- 释义
  - 根据券商柜台返回的合同编号对委托进行异步撤单操作
- 参数
  - account - StockAccount 资金账号
  - market - int 交易市场
  - order_sysid - str 券商柜台的合同编号

- 返回
  - 返回撤单请求序号, 成功委托后的撤单请求序号为大于0的正整数, 如果为-1表示委托失败
- 备注
  - 如果失败，则通过撤单失败主推接口返回撤单失败信息
- 示例
  - 股票资金账号1000000365对柜台合同编号为order_sysid的上交所委托进行异步撤单

```
account = StockAccount('1000000365')
market = xtconstant.SH_MARKET
order_sysid = "100"
#xt_trader为XtQuant API实例对象
cancel_result = xt_trader.cancel_order_stock_sysid_async(account, market,
order_sysid)
```

查询接口

资产查询

```
query_stock_asset(account)
```

- 释义
  - 查询资金账号对应的资产
- 参数
  - account - StockAccount 资金账号
- 返回
  - 该账号对应的资产对象XtAsset或者None
- 备注
  - 返回None表示查询失败
- 示例
  - 查询股票资金账号1000000365对应的资产数据

```
account = StockAccount('1000000365')
#xt_trader为XtQuant API实例对象
asset = xt_trader.query_stock_asset(account)
```

委托查询

```
query_stock_orders(account)
```

- 释义
  - 查询资金账号对应的当日所有委托
- 参数
  - account - StockAccount 资金账号
- 返回
  - 该账号对应的当日所有委托对象XtOrder组成的list或者None
- 备注
  - None表示查询失败或者当日委托列表为空
- 示例

- 查询股票资金账号1000000365对应的当日所有委托

```
account = StockAccount('1000000365')
#xt_trader为XtQuant API实例对象
orders = xt_trader.query_stock_orders(account)
```

成交查询

```
query_stock_trades(account)
```

- 释义
  - 查询资金账号对应的当日所有成交
- 参数
  - account - StockAccount 资金账号
- 返回
  - 该账号对应的当日所有成交对象XtTrade组成的list或者None
- 备注
  - None表示查询失败或者当日成交列表为空
- 示例
  - 查询股票资金账号1000000365对应的当日所有成交

```
account = StockAccount('1000000365')
#xt_trader为XtQuant API实例对象
trades = xt_trader.query_stock_trades(account)
```

持仓查询

```
query_stock_positions(account)
```

- 释义
  - 查询资金账号对应的持仓
- 参数
  - account - StockAccount 资金账号
- 返回
  - 该账号对应的最新持仓对象XtPosition组成的list或者None
- 备注
  - None表示查询失败或者当日持仓列表为空
- 示例
  - 查询股票资金账号1000000365对应的最新持仓

```
account = StockAccount('1000000365')
#xt_trader为XtQuant API实例对象
positions = xt_trader.query_stock_positions(account)
```

信用相关查询接口

信用资产查询

```
query_credit_detail(account)
```

- 释义
  - 查询信用资金账号对应的资产
- 参数
  - account - StockAccount 资金账号
- 返回
  - 该信用账户对应的资产对象XtCreditDetail组成的list或者None
- 备注
  - None表示查询失败
  - 通常情况下一个资金账号只有一个详细信息数据
- 示例
  - 查询信用资金账号1208970161对应的资产信息

```
account = StockAccount('1208970161', 'CREDIT')
#xt_trader为XtQuant API实例对象
datas = xt_trader.query_credit_detail(account)
```

负债合约查询

```
query_stk_compacts(account)
```

- 释义
  - 查询资金账号对应的负债合约
- 参数
  - account - StockAccount 资金账号
- 返回
  - 该账户对应的负债合约对象StkCompacts组成的list或者None
- 备注
  - None表示查询失败或者负债合约列表为空
- 示例
  - 查询信用资金账号1208970161对应的负债合约

```
account = StockAccount('1208970161', 'CREDIT')
#xt_trader为XtQuant API实例对象
datas = xt_trader.query_stk_compacts(account)
```

融资融券标的查询

```
query_credit_subjects(account)
```

- 释义
  - 查询资金账号对应的融资融券标的
- 参数
  - account - StockAccount 资金账号
- 返回
  - 该账户对应的融资融券标的对象CreditSubjects组成的list或者None

- 备注
  - None表示查询失败或者融资融券标的列表为空
- 示例
  - 查询信用资金账号1208970161对应的融资融券标的

```
account = StockAccount('1208970161', 'CREDIT')
#xt_trader为XtQuant API实例对象
datas = xt_trader.query_credit_subjects(account)
```

可融券数据查询

```
query_credit_slo_code(account)
```

- 释义
  - 查询资金账号对应的可融券数据
- 参数
  - account - StockAccount 资金账号
- 返回
  - 该账户对应的可融券数据对象CreditSloCode组成的list或者None
- 备注
  - None表示查询失败或者可融券数据列表为空
- 示例
  - 查询信用资金账号1208970161对应的可融券数据

```
account = StockAccount('1208970161', 'CREDIT')
#xt_trader为XtQuant API实例对象
datas = xt_trader.query_credit_slo_code(account)
```

标的担保品查询

```
query_credit_assure(account)
```

- 释义
  - 查询资金账号对应的标的担保品
- 参数
  - account - StockAccount 资金账号
- 返回
  - 该账户对应的标的担保品对象CreditAssure组成的list或者None
- 备注
  - None表示查询失败或者标的担保品列表为空
- 示例
  - 查询信用资金账号1208970161对应的标的担保品

```
account = StockAccount('1208970161', 'CREDIT')
#xt_trader为XtQuant API实例对象
datas = xt_trader.query_credit_assure(account)
```

回调类

```
class MyXtQuantTraderCallback(XtQuantTraderCallback):
    def on_disconnected(self):
        """
        连接断开
        :return:
        """
        print("connection lost")
    def on_stock_order(self, order):
        """
        委托回报推送
        :param order: XtOrder对象
        :return:
        """
        print("on order callback:")
        print(order.stock_code, order.order_status, order.order_sysid)
    def on_stock_asset(self, asset):
        """
        资金变动推送
        :param asset: XtAsset对象
        :return:
        """
        print("on asset callback")
        print(asset.account_id, asset.cash, asset.total_asset)
    def on_stock_trade(self, trade):
        """
        成交变动推送
        :param trade: XtTrade对象
        :return:
        """
        print("on trade callback")
        print(trade.account_id, trade.stock_code, trade.order_id)
    def on_stock_position(self, position):
        """
        持仓变动推送
        :param position: XtPosition对象
        :return:
        """
        print("on position callback")
        print(position.stock_code, position.volume)
    def on_order_error(self, order_error):
        """
        委托失败推送
        :param order_error:XtOrderError 对象
        :return:
        """
        print("on order_error callback")
        print(order_error.order_id, order_error.error_id, order_error.error_msg)
    def on_cancel_error(self, cancel_error):
        """
        撤单失败推送
        :param cancel_error: XtCancelError 对象
        :return:
        """
        print("on cancel_error callback")
```

```
        print(cancel_error.order_id, cancel_error.error_id,
cancel_error.error_msg)
    def on_order_stock_async_response(self, response):
        """
        异步下单回报推送
        :param response: XtOrderResponse 对象
        :return:
        """
        print("on_order_stock_async_response")
        print(response.account_id, response.order_id, response.seq)
```

连接状态回调

```
on_disconnected()
```

- 释义
  - 失去连接时推送信息
- 参数
  - 无
- 返回
  - 无
- 备注
  - 无

资产变动主推

```
on_stock_asset(asset)
```

- 释义
  - 资产变动信息的主推
- 参数
  - asset - XtAsset 资产对象
- 返回
  - 无
- 备注
  - 无

委托变动主推

```
on_stock_order(order)
```

- 释义
  - 委托变动信息的主推
- 参数
  - order - XtOrder 资产对象
- 返回
  - 无
- 备注
  - 无

成交变动主推

```
on_stock_trade(trade)
```

- 释义
  - 成交变动信息的主推
- 参数
  - trade - XtTrade 资产对象
- 返回
  - 无
- 备注
  - 无

持仓变动主推

```
on_stock_position(position)
```

- 释义
  - 持仓变动信息的主推
- 参数
  - position - XtPosition 资产对象
- 返回
  - 无
- 备注
  - 无

下单失败主推

```
on_order_error(order_error)
```

- 释义
  - 下单失败错误信息的主推
- 参数
  - position - XtOrderError 下单错误对象
- 返回
  - 无
- 备注
  - 无

撤单失败主推

```
on_cancel_error(cancel_error)
```

- 释义
  - 撤单失败错误信息的主推
- 参数
  - position - XtCancelError 撤单错误对象

- 返回
  - 无
- 备注
  - 无