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

- 2020-09-01
  - 初稿
- 2020-10-14
  - 持仓结构添加字段
  - 投资备注相关修正
- 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
- 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
- 2020-11-19
  - 添加账号状态主推接口

    - 添加账号状态数据结构说明
    - 添加账号状态枚举值
    - 回调类接口说明调整
        - 将回调函数定义及函数说明标题调整一致
        - 补充异步下单回报推送、异步撤单回报推送接口说明
- 2021-07-20
    - 修改回调/主推函数实现机制，提升报撤单回报的速度，降低穿透延时波动
    - `XtQuantTrader.run_forever()` 修改实现，支持 `Ctrl+C` 跳出
- 2022-06-27
    - 委托查询支持仅查询可撤委托
    - 添加新股申购相关接口
        - `query_new_purchase_limit` 查询新股申购额度
        - `query_ipo_data` 查询新股信息
    - 添加账号信息查询接口
        - `query_account_infos`
- 2022-07-15
    - 添加券源费率信息查询接口 `query_smt_secu_rate`
    - 添加约券异步报单接口 `smt_appointment_async`
    - 添加约券合约信息查询接口 `query_appointment_info`
- 2022-09-29
    - 添加券源券单信息查询接口 `query_smt_secu_info`
    - 修改券源费率信息查询接口 `query_smt_secu_rate`
- 2022-11-15
    - 修复 `XtQuantTrader.unsubscribe` 的实现
- 2022-11-17
    - 交易数据字典格式调整
- 2022-11-28
    - 为主动请求接口的返回增加专用线程以及相关控制，以支持在on_stock_order等推送接口中调用同步请求
        - `XtQuantTrader.set_relaxed_response_order_enabled`

快速入门

创建策略

```
#coding=utf-8
from xtquant.xttrader import XtQuantTrader, XtQuantTraderCallback
from xtquant.xttype 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)
```

```
    def on_account_status(self, status):
        """
        :param response: XtAccountStatus 对象
        :return:
        """
        print("on_account_status")
        print(status.account_id, status.account_type, status.status)

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')
    # StockAccount可以用第二个参数指定账号类型，如沪港通传'HUGANGTONG'，深港通
传'SHENGANGTONG'
    # acc = StockAccount('1000000365','STOCK')
    # 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)

- 上交所 - `xtconstant.SH_MARKET`
- 深交所 - `xtconstant.SZ_MARKET`

账号类型(account_type)

- 期货 - `xtconstant.FUTURE_ACCOUNT`
- 股票 - `xtconstant.SECURITY_ACCOUNT`

- 信用 - `xtconstant.CREDIT_ACCOUNT`
- 期货期权 - `xtconstant.FUTURE_OPTION_ACCOUNT`
- 股票期权 - `xtconstant.STOCK_OPTION_ACCOUNT`
- 沪港通 - `xtconstant.HUGANGTONG_ACCOUNT`
- 深港通 - `xtconstant.SHENGANGTONG_ACCOUNT`

委托类型(order_type)

- 股票
  - 买入 - `xtconstant.STOCK_BUY`
  - 卖出 - `xtconstant.STOCK_SELL`
- 信用
  - 担保品买入 - `xtconstant.CREDIT_BUY`
  - 担保品卖出 - `xtconstant.CREDIT_SELL`
  - 融资买入 - `xtconstant.CREDIT_FIN_BUY`
  - 融券卖出 - `xtconstant.CREDIT_SLO_SELL`
  - 买券还券 - `xtconstant.CREDIT_BUY_SECU_REPAY`
  - 直接还券 - `xtconstant.CREDIT_DIRECT_SECU_REPAY`
  - 卖券还款 - `xtconstant.CREDIT_SELL_SECU_REPAY`
  - 直接还款 - `xtconstant.CREDIT_DIRECT_CASH_REPAY`
  - 专项融资买入 - `xtconstant.CREDIT_FIN_BUY_SPECIAL`
  - 专项融券卖出 - `xtconstant.CREDIT_SLO_SELL_SPECIAL`
  - 专项买券还券 - `xtconstant.CREDIT_BUY_SECU_REPAY_SPECIAL`
  - 专项直接还券 - `xtconstant.CREDIT_DIRECT_SECU_REPAY_SPECIAL`
  - 专项卖券还款 - `xtconstant.CREDIT_SELL_SECU_REPAY_SPECIAL`
  - 专项直接还款 - `xtconstant.CREDIT_DIRECT_CASH_REPAY_SPECIAL`
- 期货六键风格
  - 开多 - `xtconstant.FUTURE_OPEN_LONG`
  - 平昨多 - `xtconstant.FUTURE_CLOSE_LONG_HISTORY`
  - 平今多 - `xtconstant.FUTURE_CLOSE_LONG_TODAY`
  - 开空 - `xtconstant.FUTURE_OPEN_SHORT`
  - 平昨空 - `xtconstant.FUTURE_CLOSE_SHORT_HISTORY`
  - 平今空 - `xtconstant.FUTURE_CLOSE_SHORT_TODAY`
- 期货四键风格
  - 平多，优先平今 - `xtconstant.FUTURE_CLOSE_LONG_TODAY_FIRST`
  - 平多，优先平昨 - `xtconstant.FUTURE_CLOSE_LONG_HISTORY_FIRST`
  - 平空，优先平今 - `xtconstant.FUTURE_CLOSE_SHORT_TODAY_FIRST`
  - 平空，优先平昨 - `xtconstant.FUTURE_CLOSE_SHORT_HISTORY_FIRST`
- 期货两键风格
  - 卖出，如有多仓，优先平仓，优先平今，如有余量，再开空 - `xtconstant.FUTURE_CLOSE_LONG_TODAY_HISTORY_THEN_OPEN_SHORT`
  - 卖出，如有多仓，优先平仓，优先平昨，如有余量，再开空 - `xtconstant.FUTURE_CLOSE_LONG_HISTORY_TODAY_THEN_OPEN_SHORT`
  - 买入，如有空仓，优先平仓，优先平今，如有余量，再开多 - `xtconstant.FUTURE_CLOSE_SHORT_TODAY_HISTORY_THEN_OPEN_LONG`
  - 买入，如有空仓，优先平仓，优先平昨，如有余量，再开多 - `xtconstant.FUTURE_CLOSE_SHORT_HISTORY_TODAY_THEN_OPEN_LONG`
  - 买入，不优先平仓 - `xtconstant.FUTURE_OPEN`
  - 卖出，不优先平仓 - `xtconstant.FUTURE_CLOSE`

- 期货 - 跨商品套利
  - 开仓 - `xtconstant.FUTURE_ARBITRAGE_OPEN`
  - 平, 优先平昨 - `xtconstant.FUTURE_ARBITRAGE_CLOSE_HISTORY_FIRST`
  - 平, 优先平今 - `xtconstant.FUTURE_ARBITRAGE_CLOSE_TODAY_FIRST`
- 期货展期
  - 看多, 优先平昨 - `xtconstant.FUTURE_RENEW_LONG_CLOSE_HISTORY_FIRST`
  - 看多，优先平今 - `xtconstant.FUTURE_RENEW_LONG_CLOSE_TODAY_FIRST`
  - 看空，优先平昨 - `xtconstant.FUTURE_RENEW_SHORT_CLOSE_HISTORY_FIRST`
  - 看空，优先平今 - `xtconstant.FUTURE_RENEW_SHORT_CLOSE_TODAY_FIRST`
- 股票期权
  - 买入开仓，以下用于个股期权交易业务 - `xtconstant.STOCK_OPTION_BUY_OPEN`
  - 卖出平仓 - `xtconstant.STOCK_OPTION_SELL_CLOSE`
  - 卖出开仓 - `xtconstant.STOCK_OPTION_SELL_OPEN`
  - 买入平仓 - `xtconstant.STOCK_OPTION_BUY_CLOSE`
  - 备兑开仓 - `xtconstant.STOCK_OPTION_COVERED_OPEN`
  - 备兑平仓 - `xtconstant.STOCK_OPTION_COVERED_CLOSE`
  - 认购行权 - `xtconstant.STOCK_OPTION_CALL_EXERCISE`
  - 认沽行权 - `xtconstant.STOCK_OPTION_PUT_EXERCISE`
  - 证券锁定 - `xtconstant.STOCK_OPTION_SECU_LOCK`
  - 证券解锁 - `xtconstant.STOCK_OPTION_SECU_UNLOCK`
- 期货期权
  - 期货期权行权 - `xtconstant.OPTION_FUTURE_OPTION_EXERCISE`

报价类型(price_type)

- 最新价 - `xtconstant.LATEST_PRICE`
- 指定价 - `xtconstant.FIX_PRICE`
- 上交所 股票
  - 最优五档即时成交剩余撤销 - `xtconstant.MARKET_SH_CONVERT_5_CANCEL`
  - 最优五档即时成交剩转限价 - `xtconstant.MARKET_SH_CONVERT_5_LIMIT`
  - 对手方最优价格委托 - `xtconstant.MARKET_PEER_PRICE_FIRST`
  - 本方最优价格委托 - `xtconstant.MARKET_MINE_PRICE_FIRST`
- 深交所 股票 期权
  - 对手方最优价格委托 - `xtconstant.MARKET_PEER_PRICE_FIRST`
  - 本方最优价格委托 - `xtconstant.MARKET_MINE_PRICE_FIRST`
  - 即时成交剩余撤销委托 - `xtconstant.MARKET_SZ_INSTBUSI_RESTCANCEL`
  - 最优五档即时成交剩余撤销 - `xtconstant.MARKET_SZ_CONVERT_5_CANCEL`
  - 全额成交或撤销委托 - `xtconstant.MARKET_SZ_FULL_OR_CANCEL`

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

账号状态(account_status)

枚举变量名	值	含义
xtconstant.ACCOUNT_STATUS_INVALID	-1	无效
xtconstant.ACCOUNT_STATUS_OK	0	正常
xtconstant.ACCOUNT_STATUS_WAITING_LOGIN	1	连接中
xtconstant.ACCOUNT_STATUSING	2	登陆中
xtconstant.ACCOUNT_STATUS_FAIL	3	失败
xtconstant.ACCOUNT_STATUS_INITING	4	初始化中
xtconstant.ACCOUNT_STATUS_CORRECTING	5	数据刷新校正中
xtconstant.ACCOUNT_STATUS_CLOSED	6	收盘后
xtconstant.ACCOUNT_STATUS_ASSIS_FAIL	7	穿透副链接断开
xtconstant.ACCOUNT_STATUS_DISABLEBYSYS	8	系统停用（总线使用-密码错误超限）
xtconstant.ACCOUNT_STATUS_DISABLEBYUSER	9	用户停用（总线使用）

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

异步约券委托反馈XtAppointmentResponse

属性	类型	注释
account_id	str	资金账号
order_sysid	str	柜台合同编号，失败为-1
error_id	int	错误编号
error_msg	str	错误信息
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

账号状态XtAccountStatus

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
status	int	账号状态，参见数据字典

账号信息XtAccountInfo

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
broker_type	int	同 account_type
platform_id	int	平台号
account_classification	int	账号分类
login_status	int	账号状态，参见数据字典

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()
```

开启主动请求接口的专用线程

```
set_relaxed_response_order_enabled(enabled)
```

- 释义

    - 控制主动请求接口的返回是否从额外的专用线程返回，以获得宽松的数据时序
- 参数
    - enabled - bool 是否开启，默认为False关闭
- 返回
    - 无
- 备注
    - 如果开启，在on_stock_order等推送回调中调用同步请求不会卡住，但查询和推送的数据在时序上会变得不确定
    - ```
      timeline    t1      t2      t3      t4
      callback    push1   push2   push3   resp4
      do          query4 ------------------^
      ```
    - 例如：分别在t1 t2 t3时刻到达三条委托数据，在on_push1中调用同步委托查询接口query_orders()
    - 未开启宽松时序时，查询返回resp4会在t4时刻排队到push3完成之后处理，这使得同步等待结果的查询不能返回而卡住执行
    - 开启宽松时序时，查询返回的resp4由专用线程返回，程序正常执行，但此时查到的resp4是push3之后的状态，也就是说resp4中的委托要比push2 push3这两个前一时刻推送的数据新，但在更早的t1时刻就进入了处理
    - 使用中请根据策略实际情况来开启，通常情况下，推荐在on_stock_order等推送回调中使用查询接口的异步版本，如`query_stock_orders_async`

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0，订阅失败返回-1
- 备注
    - 无
- 示例
    - 订阅资金账号1000000365

```
account = StockAccount('1000000365')
#xt_trader为XtQuant API实例对象
subscribe_result = 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0，订阅失败返回-1
- 备注
  - 无
- 示例
  - 订阅资金账号1000000365

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

股票查询接口

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, cancelable_only = False)
```

- 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

    - cancelable_only - bool 仅查询可撤委托
- 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, False)
```

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)
```

其他查询接口

新股申购额度查询

```
query_new_purchase_limit(account)
```

- 释义
    - 查询新股申购额度
- 参数
    - account - StockAccount 资金账号
- 返回
    - dict 新股申购额度数据集
        - { type1: number1, type2: number2, ... }
            - type - str 品种类型
                - `KCB` - 科创板，`SH` - 上海，`SZ` - 深圳
            - number - int 可申购股数
- 备注
    - 数据仅代表股票申购额度，债券的申购额度固定10000张

当日新股信息查询

```
query_ipo_data()
```

- 释义
    - 查询当日新股新债信息
- 参数
    - 无
- 返回
    - dict 新股新债信息数据集
        - { stock1: info1, stock2: info2, ... }
            - stock - str 品种代码，例如 '301208.SZ'
            - info - dict 新股信息
                - name - str 品种名称
                - type - str 品种类型
                    - `STOCK` - 股票，`BOND` - 债券
                - minPurchaseNum / maxPurchaseNum - int 最小 / 最大申购额度
                    - 单位为股（股票）/ 张（债券）

        - purchaseDate - str 申购日期
        - issuePrice - float 发行价
    - 返回值示例

```
{'754810.SH': {'name': '丰山发债', 'type': 'BOND', 'maxPurchaseNum': 10000, 'minPurchaseNum': 10, 'purchaseDate': '20220627', 'issuePrice': 100.0}, '301208.SZ': {'name': '中亦科技', 'type': 'STOCK', 'maxPurchaseNum': 16500, 'minPurchaseNum': 500, 'purchaseDate': '20220627', 'issuePrice': 46.06}}
```

- 备注
    - 无

账号信息查询

```
query_account_infos()
```

- 释义
    - 查询所有资金账号
- 参数
    - 无
- 返回
    - list 账号信息列表
        - [XtAccountInfo]
- 备注
    - 无

约券相关接口

券源券单信息查询

```
query_smt_secu_info(account)
```

- 释义
    - 查询券源券单信息
- 参数
    - account - StockAccount 资金账号
- 返回
    - dict 券源券单数据集
        - { stock1: info1, stock2: info2, ... }
            - stock - str 证券代码，例如 '600000.SH'
            - info - dict 券源券单信息
                - success - bool
                - error - str
                - stockName - str 证券名称
                - creditType - str 资券类型
                - tradeType - str 业务类型

        - compactTerm - int 约定期限
        - maxTerm - int 最大约定期限
        - lendAmount - int 出借数量
        - remark - str 备注
        - fareWay - str 折扣标志
        - fareRateNew - float 费用比例
    - 返回值示例

```
{'000157.SZ': {'success': True, 'error': '', 'stockName': '中联重科',
'creditType': '1', 'tradeType': '0', 'compactTerm': 14, 'maxTerm':
182, 'lendAmount': 1280, 'remark': ' ', 'fareWay': '3',
'fareRateNew': 0.01}, '000006.SZ': {'success': True, 'error': '',
'stockName': '深振业A', 'creditType': '1', 'tradeType': '0',
'compactTerm': 7, 'maxTerm': 7, 'lendAmount': 99040, 'remark': ' ',
'fareWay': '3', 'fareRateNew': 0.01}, '000166.SZ': {'success':
True, 'error': '', 'stockName': '申万宏源', 'creditType': '1',
'tradeType': '0', 'compactTerm': 2, 'maxTerm': 182, 'lendAmount':
5350, 'remark': ' ', 'fareWay': '3', 'fareRateNew': 0.01},
'000046.SZ': {'success': True, 'error': '', 'stockName': '泛海控股',
'creditType': '1', 'tradeType': '0', 'compactTerm': 6, 'maxTerm':
6, 'lendAmount': 49440, 'remark': ' ', 'fareWay': '3',
'fareRateNew': 0.01}, '300748.SZ': {'success': True, 'error': '',
'stockName': '金力永磁', 'creditType': '1', 'tradeType': '0',
'compactTerm': 14, 'maxTerm': 14, 'lendAmount': 28700, 'remark': '
', 'fareWay': '3', 'fareRateNew': 0.01}, '000021.SZ': {'success':
True, 'error': '', 'stockName': '深科技', 'creditType': '1',
'tradeType': '0', 'compactTerm': 7, 'maxTerm': 7, 'lendAmount':
189200, 'remark': ' ', 'fareWay': '3', 'fareRateNew': 0.01}}
```

- 备注
    - 无

券源费率信息查询

```
query_smt_secu_rate(account, stock_code, max_term, fare_way, credit_type,
trade_type)
```

- 释义
    - 利用券源券单查询接口返回的券单信息，查询券源费率信息
- 参数
    - account - StockAccount 资金账号
    - stock_code - str 证券代码， 如'600000.SH'
    - max_term - int 最大约定期限
    - fare_way - str 折扣标志
    - credit_type - str 资券类型
    - trade_type -str 业务类型
- 返回
    - dict 券源费率信息
        - success - bool
        - error - str
        - fareRatio - float 合约利率

        - subRareRatio - float 提前终止利率
        - fineRatio - float 罚息利率
        - 返回值示例

```
{'success': True, 'error': '', 'fareRatio': 0.1, 'subRareRatio':
0.1, 'fineRatio': 0.1}
```

- 备注
    - 无

约券异步报单

```
smt_appointment_async(account, stock_code, apt_days, apt_volume, fare_ratio,
sub_rare_ratio, fine_ratio, begin_date)
```

- 释义
    - 对约券进行异步下单操作，异步下单接口如果正常返回了下单请求序号seq，会收到on_smt_appointment_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'
    - apt_days - int 约定期限
    - apt_volume - int 约定数量
    - fare_ratio - float 约券费率
    - sub_rare_ratio - float 提前归还费率
    - fine_ratio - float 违约金率
    - begin_date - str 约定日期
- 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，也会通过on_smt_appointment_async_response委托反馈收到错误信息
- 示例
    - 信用资金账号8009248627对000001.SZ约券

```
account = StockAccount('8009248627')
account.account_type = 3
#xt_trader为XtQuant API实例对象
seq =
xt_trader.smt_appointment_async(account,'000001.SZ',10,100,0.5,0.3,0.1,
'20220715')
```

约券合约信息查询

```
query_appointment_info
```

- 释义
    - 查询约券合约信息
- 参数

    - account - StockAccount 资金账号
- 返回
    - dict 约券合约信息数据集
        - { compactId1: info1, compactId2: info2, ... }
            - compactId - str 合约编号，例如 '20220429009980000000018'
            - info - dict 约券合约信息
                - success - bool
                - error - str
                - fundAccount - str 资金账号
                - origCompactId - str 原合约编号
                - exchangeType - str 交易市场类别
                - stockCode - str 证券代码
                - stockName - str 股票名称
                - contractEndDate - int 合约到期日
                - feeRatio - float 费用利率
                - compactTerm - int 合约期限
                - compactAmount - int 合约数量
                - compactRepayDate - int 合约还券日
                - compactStatus - str 合约状态
                - positionStr - str 定位串
        - 返回值示例

```
{'20220429009980000000018': {'success': True, 'error': '',
'fundAccount': '8009248627', 'origCompactId':
'20220223009980000004481', 'exchangeType': 'SZ', 'stockCode':
'002407', 'stockName': '', 'contractEndDate': 20220306, 'feeRatio':
0.08, 'compactTerm': 12, 'compactAmount': 100, 'compactRepayDate':
0, 'compactStatus': '0', 'positionStr': ''},
'20220628009980000000031': {'success': True, 'error': '',
'fundAccount': '8009248627', 'origCompactId':
'20220223009980000004534', 'exchangeType': 'SZ', 'stockCode':
'002407', 'stockName': '', 'contractEndDate': 20220306, 'feeRatio':
0.08, 'compactTerm': 12, 'compactAmount': 100, 'compactRepayDate':
0, 'compactStatus': '0', 'positionStr': ''}}
```

- 备注
    - 无

回调类

```
class MyXtQuantTraderCallback(XtQuantTraderCallback):
    def on_disconnected(self):
        """
        连接状态回调
        :return:
        """
        print("connection lost")
    def on_account_status(self, status):
        """
        账号状态信息推送
        :param response: XtAccountStatus 对象
        :return:
```

```
        """
        print("on_account_status")
        print(status.account_id, status.account_type, status.status)
    def on_stock_asset(self, asset):
        """
        资金信息推送
        :param asset: XtAsset对象
        :return:
        """
        print("on asset callback")
        print(asset.account_id, asset.cash, asset.total_asset)
    def on_stock_order(self, order):
        """
        委托信息推送
        :param order: XtOrder对象
        :return:
        """
        print("on order callback:")
        print(order.stock_code, order.order_status, order.order_sysid)
    def on_stock_trade(self, trade):
        """
        成交信息推送
        :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):
        """
        持仓信息推送
        :param position: XtPosition对象
        :return:
        """
        print("on position callback")
        print(position.stock_code, position.volume)
    def on_order_error(self, order_error):
        """
        下单失败信息推送
        :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):
        """
        撤单失败信息推送
        :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)
    def on_smt_appointment_async_response(self, response):
        """
        :param response: XtAppointmentResponse 对象
        :return:
        """
        print("on_smt_appointment_async_response")
        print(response.account_id, response.order_sysid, response.error_id,
response.error_msg, 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

账号状态信息推送

```
on_account_status(data)
```

- 释义
  - 账号状态信息变动推送
- 参数
  - data - XtAccountStatus 账号状态信息
- 返回
  - 无
- 备注
  - 无

资产信息推送

```
on_stock_asset(data)
```

- 释义
  - 资产信息变动推送
- 参数
  - data - XtAsset 资产信息
- 返回
  - 无
- 备注
  - 无

委托信息推送

```
on_stock_order(data)
```

- 释义
  - 委托信息变动推送
- 参数
  - data - XtOrder 委托信息
- 返回
  - 无
- 备注
  - 无

成交信息推送

```
on_stock_trade(data)
```

- 释义
  - 成交信息变动推送
- 参数
  - data - XtTrade 成交信息
- 返回
  - 无
- 备注
  - 无

持仓信息推送

```
on_stock_position(data)
```

- 释义
  - 持仓信息变动推送
- 参数
  - data - XtPosition 持仓信息
- 返回
  - 无
- 备注
  - 无

下单失败信息推送

```
on_order_error(data)
```

- 释义
  - 下单失败信息推送
- 参数
  - data - XtOrderError 下单失败信息

- 返回
  - 无
- 备注
  - 无

撤单失败信息推送

```
on_cancel_error(data)
```

- 释义
  - 撤单失败信息的推送
- 参数
  - data - XtCancelError 撤单失败信息
- 返回
  - 无
- 备注
  - 无

异步下单回报推送

```
on_order_stock_async_response(data)
```

- 释义
  - 异步下单回报推送
- 参数
  - data - XtOrderResponse 异步下单委托反馈
- 返回
  - 无
- 备注
  - 无

异步约券回报推送

```
on_smt_appointment_async_response(data)
```

- 释义
  - 异步下单回报推送
- 参数
  - data - XtAppointmentResponse 异步约券委托反馈
- 返回
  - 无
- 备注
  - 无