



迅投Python版本文档说明v1.2.3

修改记录

| 文档版本 | 修订日期 | 修订作者 | 修订内容 |
|------|------------|------|---|
| 1.0 | 2024/02/21 | 代鹏 | 新建api文档 |
| 1.1 | 2024/2/28 | 代鹏 | 新增两融综合信息同步接口 |
| 1.2 | 2024/3/6 | 代鹏 | CAccountDetail新增账号名称，经济公司名称，经纪公司编号，COrderInfo新增成交均价，成交金额 |
| 1.3 | 2024/4/2 | 代鹏 | 新增查询ETF申赎清单，ETF成分股，期货持仓统计接口。 修改委托明细和成交明细结构新增组合持仓编号字段。 |
| 1.4 | 2024/5/9 | 代鹏 | 重要更新：修复中文乱码导致的崩溃问题。 功能更新：新增委托入参指令跨日开关，合约查询新增只查可交易合约参数。 说明更新：m_nLimitedPriceType字段用法说明，新增常见错误说明 |
| 1.5 | 2024/5/17 | 代鹏 | 参数更新：智能算法和主动算法新增时间类型参数m_nTimeType和执行时间参数m_nTimeValue。 |
| 1.6 | 2024/5/21 | 代鹏 | 数据结构更新：CCreditAccountDetail后续不再维护，改为CCreditDetail。 |
| 1.7 | 2024/6/7 | 代鹏 | 数据结构更新：CAccountDetail新增字段m_dAssetBalance, m_dRiskDegree, m_dInitCloseMoney.CInstrumentInfo新增字段m_strOptUndlName, m_nOptUnit。 函数更新：新增接口 reqExchangeStatusSync，废弃reqPriceDataByMarket，后续合并到 |

| 文档版本 | 修订日期 | 修订作者 | 修订内容 |
|------|----------------|------|---|
| | | | reqInstrumentInfoByMarket, 新增同步函数 reqInstrumentInfoByMarketSync 新增主推函数 onRtnMarketAndTradeDay。 |
| 1.8 | 2024/6/ 20 | 代鹏 | 函数更新: reqAccountDetailSync和 reqPriceDataSync函数重构, 返回值变更。 |
| 1.9 | 2024/6/ 26 | 代鹏 | 数据结构更新: COrderInfo新增字段 m_nLimitedPriceType m_strOtherParam,m_nCmdSource. CDealDetail新增字段m_dOccupiedMargin。 CAlgorithmOrder新增字段 m_nLimitedPriceType。 CIntelligentAlgorithmOrder和 CEXternAlgorithmOrder新增字段 m_strRemark1。 |
| 1.10 | 2024/ 8/7 | 代鹏 | 函数更新: 新增刷新负债接口 refreshStkClosedCompacts和 refreshStkUnCloseCompacts。 数据结构更新: CPositionStatics新增字段 m_dReferenceRate. CDealDetail新增字段 m_dReferenceRate。 |
| 1.11 | 2024/8/ 12 | 代鹏 | 数据结构更新: CPositionStatics新增字段 m_strExpireDate和m_nPortfolioPositionType。 CDealDetail新增字段m_strEntryDate, m_strExpireDate和m_nPortfolioDealType。新 增枚举定义PortfolioDealType和 PortfolioPositionType |
| 1.12 | 2024/9/ 19 | 代鹏 | 新增枚举定义: EBrokerPriceType新增 BROKER_PRICE_PROP_MARKET, 数据结构更 新: CDealDetail结构m_strCombID类型修复, CIntelligentAlgorithmOrder新增 m_nOpenTrade字段 |
| 1.13 | 2024/ 10/12 | 代鹏 | 新增产品信息结构CProductData结构定义, 产品 净值信息CNetValue结构定义 |

| 文档版本 | 修订日期 | 修订作者 | 修订内容 |
|-------|----------------|------|--|
| 1.2.0 | 2024/ 10/16 | 代鹏 | 重构python文档 |
| 1.2.1 | 2024/ 10/23 | 代鹏 | 新增投组类型EPortfolioType结构定义, CAlgorithmOrder新增算法波动区间上下限 |
| 1.2.2 | 2025/ 05/22 | 代鹏 | CCreditDetail结构新增字段 CPositionStatics新增字段 CNetValue新增字段 CIntelligentAlgorithmOrder新增字段 |
| 1.2.3 | 2025/ 06/13 | 代鹏 | COrdinaryOrder CAlgorithmOrder COrdinaryGroupOrder结构新增字段 新增枚举类型 OrderPriceType MetelD 新增退订主推函数unSubscribeMetes以及对应的 回调函数onUnSubscribeMetes |

目录

迅投Python版本文档说明v1.2.3

- 修改记录

- 目录

1. 概述

1.1. 迅投python版本XtTraderApi

- 1.1.1. XtTraderApi运行逻辑
- 1.1.2. XtTraderApi运行依赖环境
- 1.1.3. XtTraderApi各个组件说明

2. 字典和数据结构

2.1. XtTraderApi数据类型和数据结构字典

- 2.1.1. 交易市场
- 2.1.2. 账号类型EXTBrokerType
- 2.1.3. 委托类型EOperationType
- 2.1.4. 报价类型EPriceType
- 2.1.5. 委托状态EEntrustStatus
- 2.1.6. 普通算法单笔基准量类型EVolumeType
- 2.1.7. 智能算法类型m_strOrderType

- 2.1.8. 主动算法类型m_strOrderType
- 2.1.9. 登录状态EBrokerLoginStatus
- 2.1.10. 买卖方向EEntrustBS
- 2.1.11. 指令状态EOrderCommandStatus
- 2.1.12. 委托类型EEntrustTypes
- 2.1.13. 期货交易类型EFutureTradeType
- 2.1.14. 委托状态EEntrustStatus
- 2.1.15. 期货委托发送状态EEntrustSubmitStatus
- 2.1.16. 两融标的状态EXTSubjectsStatus
- 2.1.17. 两融负债状态EXTCompactStatus
- 2.1.18. 下单超过流控时处理方式EXtOverFreqOrderMode
- 2.1.19. 停牌标志EXtSuspendedType
- 2.1.20. 备兑标志ECoveredFlag
- 2.1.21. 期权持仓类型ESideFlag
- 2.1.22. 订阅行情平台类型EXTOfferStatus
- 2.1.23. 期货时间条件单类型ETimeCondition
- 2.1.24. 期货数量条件单类型EVolumeCondition
- 2.1.25. 股票期权合约类型EOptionType
- 2.1.26. 币种EMoneyType
- 2.1.27. 任务操作ETaskFlowOperation
- 2.1.28. 涨跌停控制EStopTradeForOwnHiLow
- 2.1.29. 两融负债头寸来源EXTCompactBrushSource
- 2.1.30. 触价类型EOpTriggerType
- 2.1.31. 市场状态EXTExchangeStatus
- 2.1.32. 是否使用大额单边保证金算法EXtMaxMarginSideAlgorithmType
- 2.1.33. 普通算法订单类型EAlgoPriceType
- 2.1.34. 银证转账方向ETransDirection
- 2.1.35. 算法下单方式EOrderStrategyType
- 2.1.36. 双中心状态 EDualStatus
- 2.1.37. 股份划拨信用划拨类别 ETransTypeCreditFlag
- 2.1.38. 资金股份划拨划拨方向 ESecuFundTransDirection
- 2.1.39. 委托明细价格类型 EBrokerPriceType
- 2.1.40. 开平标志 EOffsetFlagType
- 2.1.41. 投保标志 EHedgeFlagType
- 2.1.42. 股票状态 EStockStatus
- 2.1.43. 两融负债类型 EXTCompactType
- 2.1.44. 两融负债状态 EXTCompactStatus
- 2.1.45. 备兑标志 ECoveredFlag

- 2.1.46. 股票期权合约类型 EOptionType
- 2.1.47. 期权持仓类型 ESideFlag
- 2.1.48. 涨跌停控制 EStopTradeForOwnHiLow
- 2.1.49. 两融负债头寸来源 EXTCompactBrushSource
- 2.1.50. 触价类型 EOpTriggerType
- 2.1.51. 市场状态 EXTExchangeStatus
- 2.1.52. 银证转账方向 ETransDirection
- 2.1.53. 算法下单方式 EOrderStrategyType
- 2.1.54. 股东号主副标记 EMainFlag
- 2.1.55. 证券类别 EStockType
- 2.1.56. EReplaceFlag 现金替代标识
- 2.1.57. EXTCommandDateLimit 指令跨日标志
- 2.1.58. PortfolioDealType 投资组合成交类型
- 2.1.59. PortfolioPositionType 投资组合持仓类型
- 2.1.60. EPortfolioType 投组类型
- 2.1.61. OrderPriceType 期货限价委托类型
- 2.1.62. MetelD 退订枚举类型
- 2.2. XtTraderApi数据结构说明
 - 2.2.1. 股票、期货、股票期权的资产结构 CAccountDetail
 - 2.2.2. 信用综合资产结构 CCreditAccountDetail 不再维护，改用 CCreditDetail
 - 2.2.3. 指令状态 COrderInfo
 - 2.2.4. 委托明细 COrderDetail
 - 2.2.5. 成交明细 CDealDetail
 - 2.2.6. 持仓明细 CPositionDetail
 - 2.2.7. 持仓统计 CPositionStatics
 - 2.2.8. 普通委托 COrdinaryOrder
 - 2.2.9. 算法委托 CAlgorithmOrder
 - 2.2.10. 智能算法委托 CIntelligentAlgorithmOrder
 - 2.2.11. 主动算法委托 CExternAlgorithmOrder
 - 2.2.12. 错误消息 XtError
 - 2.2.13. 委托错误信息 COrderError(委托请求被迅投风控或者柜台打回，会推送该消息)
 - 2.2.14. 撤销委托错误信息 CCancelError (撤销委托请求被迅投风控或者柜台打回，会推送该消息)
 - 2.2.15. 两融标的信息 CStkSubjects
 - 2.2.16. 两融标的状态枚举类型 EXTSubjectsStatus
 - 2.2.17. 个股期权备兑持仓信息 CCoveredStockPosition

- 2.2.18. 股票期权组合策略持仓信息 CStockOptionCombPositionDetail
- 2.2.19. 订阅行情请求 CSubscribData
- 2.2.20. 汇率信息 CReferenceRate
- 2.2.21. 行情数据 CPriceData
- 2.2.22. 股票（合约）信息 CInstrumentDetail
- 2.2.23. 组合单下单参数 CGroupOrder
- 2.2.24. 算法组合单下单参数 CAlgGroupOrder
- 2.2.25. 外部算法组合单下单参数 CExternAlgGroupOrder
- 2.2.26. 普通组合单下单参数 COrdinaryGroupOrder
- 2.2.27. 账号主键 CAccountKey
- 2.2.28. 普通柜台资金信息 CStockComFund
- 2.2.29. 普通柜台持仓信息 CStockComPosition
- 2.2.30. ETF申赎清单 CETFPCFDdetail
- 2.2.31. ETF成分股信息 CETFComponentStockInfo
- 2.2.32. 期货持仓统计 CFuturePositionStatics
- 2.2.33. 用户配置信息 CUserConfig
- 2.2.34. 信用账号信息 CCreditDetail
- 2.2.35. 合约信息 CInstrumentInfo
- 2.2.36. 产品净值信息 CNetValue
- 2.2.37. 产品信息 CProductData

3. XtTraderApi

▪3.0 XtTraderApi API说明

3.1. 会话接口

- 3.1.1. 获取XtTraderApi实例
- 3.1.2. 初始化API实例并建立连接
- 3.1.3. 启动XtTraderApi单实例线程
- 3.1.4. 异步启动XtTraderApi单实例线程
- 3.1.5. 启动XtTraderApi多实例线程
- 3.1.6. 异步启动XtTraderApi单实例线程
- 3.1.7. 析构XtTraderApi实例
- 3.1.8. 销毁XtTraderApi多实例线程
- 3.1.9. 设置数据回调对象

3.2. 交易接口

- 3.2.1. 普通单下单 同步接口
- 3.2.2. 普通单下单
- 3.2.3. 直接下单
- 3.2.4. 组合算法单下单

- 3.2.5. 组合智能算法单下单 同步接口
- 3.2.6. 组合智能算法单下单
- 3.2.7. 组合外部算法单下单 同步接口
- 3.2.8. 组合外部算法单下单
- 3.2.9. 算法单下单 同步接口
- 3.2.10. 算法单下单
- 3.2.11. 随机量交易下单
- 3.2.12. 智能算法下单 同步接口
- 3.2.13. 智能算法下单
- 3.2.14. 主动算法下单 同步接口
- 3.2.15. 主动算法下单
- 3.2.16. 普通组合下单
- 3.2.17. 撤单 同步接口
- 3.2.18. 撤单
- 3.2.19. 按委托号撤单 同步接口
- 3.2.20. 按委托号撤单
- 3.2.21. 按指令号暂停指令并暂停任务 同步接口
- 3.2.22. 按指令号恢复指令并恢复任务 同步接口
- 3.2.23. 指令暂停恢复
- 3.2.24. 智能算法指令改单
- 3.2.25. 普通算法指令改单

3.3. 查询接口

- 3.3.1. 获取账号key对应产品Id
- 3.3.2. 获取迅投交易用户名
- 3.3.3. 获取Api版本
- 3.3.4. 用户登陆 同步接口
- 3.3.5. 用户登陆 异步接口
- 3.3.6. 资产查询 同步接口
- 3.3.7. 资产查询 异步接口
- 3.3.8. 请求账号委托明细信息
- 3.3.9. 请求账号委托明细信息 同步接口
- 3.3.10. 根据指令号请求账号委托明细信息
- 3.3.11. 同步根据指令号请求账号委托明细信息
- 3.3.12. 同步请求账号成交明细信息
- 3.3.13. 同步请求账号成交统计信息
- 3.3.14. 同步根据指令号请求账号成交明细信息
- 3.3.15. 请求账号成交明细信息
- 3.3.16. 根据指令号请求账号成交明细信息

- 3.3.17. 请求账号持仓明细信息
- 3.3.18. 同步请求账号持仓明细信息
- 3.3.19. 请求账号持仓统计信息
- 3.3.20. 同步请求账号持仓统计信息
- 3.3.21. 请求账号日初持仓统计信息
- 3.3.22. 同步请求账号日初持仓统计信息
- 3.3.23. 请求信用账号标的信息
- 3.3.24. 请求期权账号备兑持仓信息
- 3.3.25. 请求期权账号组合持仓信息
- 3.3.26. 请求账号汇率信息
- 3.3.27. 请求两融账号综合资金信息
- 3.3.28. 上传终端ctp采集信息，用于传入ctp柜台需要留痕的终端信息
- 3.3.29. 设置用户自己的定时器
- 3.3.30. 停止用户自己的定时器
- 3.3.31. 设置用户下单指令冻结选项
- 3.3.32. 请求账号新股额度信息
- 3.3.33. 请求信用账号未了结负债信息
- 3.3.34. 请求信用账号未了结负债信息 同步接口
- 3.3.35. 请求信用账号已了结负债信息
- 3.3.36. 请求信用账号已了结负债信息 同步接口
- 3.3.37. 获取用户下所有账号key 同步接口
- 3.3.38. 获取用户下所有账号key
- 3.3.39. 根据委托号请求账号成交明细信息
- 3.3.40. 请求账号结算单信息
- 3.3.41. 请求账号合约行情信息
- 3.3.42. 请求账号可下单量
- 3.3.43. 请求账号可下单量
- 3.3.44. 请求账号可下单量 同步接口
- 3.3.45. 请求两融账号融券可融数量信息
- 3.3.46. 请求两融账号融资融券标的信息
- 3.3.47. 请求两融账号担保标的信息
- 3.3.48. 请求账号银证转账银行信息
- 3.3.49. 请求账号银证转账银行流水
- 3.3.50. 请求账号银证转账银行余额
- 3.3.51. 银证转账
- 3.3.52. 请求账号可撤单委托明细信息
- 3.3.53. 请求账户所有指令信息
- 3.3.54. 请求账户所有指令信息 同步接口

- 3.3.55. 资金划拨
- 3.3.56. 股份划拨
- 3.3.57. 请求账号普通（集中）柜台资金
- 3.3.58. 请求账号普通（集中）柜台持仓
- 3.3.59. 获取当前交易日
- 3.3.60. 请求账号历史委托明细
- 3.3.61. 请求账号历史委托明细 同步接口
- 3.3.62. 请求账号历史成交明细
- 3.3.63. 请求账号历史成交明细 同步接口
- 3.3.64. 请求账号历史持仓统计
- 3.3.65. 请求账号历史持仓统计 同步接口
- 3.3.66. 请求账号历史资金信息 同步接口
- 3.3.67. 请求期货账号的保证金率
- 3.3.68. 请求期货账号的手续费率
- 3.3.69. 获取用户下所有的产品Id
- 3.3.70. 创建新投资组合
- 3.3.71. 查询产品Id下所有的投资组合
- 3.3.72. 请求投资组合委托信息
- 3.3.73. 请求投资组合一段时间内的委托信息
- 3.3.74. 请求投资组合成交信息
- 3.3.75. 请求投资组合一段时间内的委托信息
- 3.3.76. 请求投资组合持仓信息
- 3.3.77. 请求收益互换账号框架号
- 3.3.78. 请求股东号
- 3.3.79. 请求资金账号状态
- 3.3.80. 请求资金账号状态 同步接口
- 3.3.81. 查询投资组合资金信息
- 3.3.82. 请求市场状态信息
- 3.3.83. 查询枚举名称
- 3.3.84. 请求用户产品信息
- 3.3.85. 请求用户产品信息 同步接口
- 3.3.86. 请求两融账号综合资金信息
- 3.3.87. 查询ETF申赎清单
- 3.3.88. 查询ETF申赎清单 同步接口
- 3.3.89. 查询ETF成分股信息
- 3.3.90. 查询ETF成分股信息 同步接口
- 3.3.91. 请求期货账号持仓统计信息 同步接口
- 3.3.92. 请求期货账号持仓统计信息

- 3.3.93. 查询用户设置
- 3.3.94. 查询用户设置 同步接口
- 3.3.95. 请求市场状态信息
- 3.3.96. 按市场请求合约信息 同步接口
- 3.3.97. 刷新信用账号未了结负债信息
- 3.3.98. 刷新信用账号已了结负债信息

3.4. 行情接口

- 3.4.1. 请求行情数据信息
- 3.4.2. 请求行情数据信息 同步接口
- 3.4.3. 按市场请求行情数据信息 已废弃
- 3.4.4. 请求账号期权行情信息
- 3.4.5. 订阅行情数据
- 3.4.6. 批量订阅行情数据
- 3.4.7. 退订行情数据
- 3.4.8. 批量退订行情数据
- 3.4.9. 按市场请求合约信息

4. XtTraderApiCallback

▪4.0 XtTraderApiCallback 类说明

4.1. 连接类回调

- 4.1.1. 连接服务器的回调函数
- 4.1.2. 用户登录的回调函数

4.2. 交易接口回调

- 4.2.1. 下单的回调函数
- 4.2.2. 直接下单的回调函数
- 4.2.3. 撤指令的回调函数
- 4.2.4. 撤指令的回调函数
- 4.2.5. 撤委托的回调函数
- 4.2.6. 风险试算的回调函数
- 4.2.7. 暂停恢复任务回调
- 4.2.8. 指令改参的回调函数

4.3. 非交易接口回调

- 4.3.1. 请求账号资金的回调函数
- 4.3.2. 请求账号资金的回调函数
- 4.3.3. 请求账号资金的回调函数
- 4.3.4. 请求账号资金的回调函数
- 4.3.5. 请求账号委托明细的回调函数
- 4.3.6. 请求账号委托明细的回调函数

- 4.3.7. 请求账号委托明细的回调函数
- 4.3.8. 请求账号委托明细的回调函数
- 4.3.9. 请求账号成交明细的回调函数
- 4.3.10. 请求账号成交明细的回调函数
- 4.3.11. 请求账号成交明细的回调函数
- 4.3.12. 请求账号成交明细的回调函数
- 4.3.13. 请求账号持仓明细的回调函数
- 4.3.14. 请求账号持仓明细的回调函数
- 4.3.15. 请求账号持仓明细的回调函数
- 4.3.16. 请求账号持仓明细的回调函数
- 4.3.17. 请求账号持仓统计的回调函数
- 4.3.18. 请求账号持仓统计的回调函数
- 4.3.19. 请求账号持仓统计的回调函数
- 4.3.20. 请求账号持仓统计的回调函数
- 4.3.21. 请求账号日初持仓统计的回调函数
- 4.3.26. 请求两融账号标的的回调函数
- 4.3.27. 请求两融账号标的的回调函数
- 4.3.28. 请求两融账号标的的回调函数
- 4.3.29. 请求两融账号标的的回调函数
- 4.3.30. 请求两融账号负债的回调函数
- 4.3.31. 请求两融账号负债的回调函数
- 4.3.32. 请求两融账号负债的回调函数
- 4.3.33. 请求两融账号负债的回调函数
- 4.3.34. 请求期权账号备兑持仓的回调函数
- 4.3.35. 请求期权账号备兑持仓的回调函数
- 4.3.36. 请求期权账号备兑持仓的回调函数
- 4.3.37. 请求两融账号负债的回调函数
- 4.3.38. 请求产品信息的回调函数
- 4.3.39. 请求产品信息的回调函数
- 4.3.40. 请求合约数据的回调函数
- 4.3.41. 请求合约数据的回调函数
- 4.3.42. 请求合约数据的回调函数
- 4.3.43. 请求合约数据的回调函数
- 4.3.44. 请求行情数据的回调函数
- 4.3.45. 请求期权账号组合持仓数据的回调函数
- 4.3.46. 请求期权账号组合持仓数据的回调函数
- 4.3.47. 请求期权账号组合持仓数据的回调函数
- 4.3.48. 请求期权账号组合持仓数据的回调函数

- 4.3.49. 行情订阅的回调函数
- 4.3.50. 行情退订的回调函数
- 4.3.51. 请求港股账号汇率数据的回调函数
- 4.3.52. 请求港股账号汇率数据的回调函数
- 4.3.53. 请求港股账号汇率数据的回调函数
- 4.3.54. 请求港股账号汇率数据的回调函数
- 4.3.55. 请求两融账号两融综合资金数据的回调函数
- 4.3.56. 请求两融账号两融综合资金数据的回调函数
- 4.3.57. 请求两融账号两融综合资金数据的回调函数
- 4.3.58. 请求两融账号两融综合资金数据的回调函数
- 4.3.59. 用户自己的定时器回调 已废弃
- 4.3.60. 用户自己的定时器回调
- 4.3.61. 请求新股额度数据的回调函数
- 4.3.62. 请求新股额度数据的回调函数
- 4.3.63. 请求新股额度数据的回调函数
- 4.3.64. 请求新股额度数据的回调函数
- 4.3.65. 请求未了结负债信息的回调函数
- 4.3.66. 请求未了结负债信息的回调函数
- 4.3.67. 请求未了结负债信息的回调函数
- 4.3.68. 请求未了结负债信息的回调函数
- 4.3.69. 请求已了结负债信息的回调函数
- 4.3.70. 请求已了结负债信息的回调函数
- 4.3.71. 请求已了结负债信息的回调函数
- 4.3.72. 请求已了结负债信息的回调函数
- 4.3.73. 获取用户下所有账号key的回调函数
- 4.3.74. 获取用户下所有账号key的回调函数
- 4.3.75. 根据委托号请求账号成交明细信息的回调函数
- 4.3.76. 根据委托号请求账号成交明细信息的回调函数
- 4.3.77. 根据委托号请求账号成交明细信息的回调函数
- 4.3.78. 根据委托号请求账号成交明细信息的回调函数
- 4.3.79. 请求账号结算单信息的回调函数
- 4.3.80. 请求账号结算单信息的回调函数
- 4.3.81. 请求账号结算单信息的回调函数
- 4.3.82. 请求账号结算单信息的回调函数
- 4.3.83. 请求账号合约行情信息的回调函数
- 4.3.84. 请求账号可下单量的回调函数
- 4.3.85. 请求账号可下单量的回调函数
- 4.3.86. 请求账号可下单量的回调函数

- 4.3.87. 请求两融账号融券可融数量的回调函数
- 4.3.88. 请求两融账号融券可融数量的回调函数
- 4.3.89. 请求两融账号融券可融数量的回调函数
- 4.3.90. 请求两融账号融券可融数量的回调函数
- 4.3.91. 请求两融账号融资融券标的的回调函数
- 4.3.92. 请求两融账号融资融券标的的回调函数
- 4.3.93. 请求两融账号融资融券标的的回调函数
- 4.3.94. 请求两融账号融资融券标的的回调函数
- 4.3.95. 请求两融账号担保标的的回调函数
- 4.3.96. 请求两融账号担保标的的回调函数
- 4.3.97. 请求两融账号担保标的的回调函数
- 4.3.98. 请求两融账号担保标的的回调函数
- 4.3.99. 请求账号银证转账银行信息的回调函数
- 4.3.100. 请求账号银证转账银行信息的回调函数
- 4.3.101. 请求账号银证转账银行信息的回调函数
- 4.3.102. 请求账号银证转账银行信息的回调函数
- 4.3.103. 请求账号银证转账银行流水的回调函数
- 4.3.104. 请求账号银证转账银行流水的回调函数
- 4.3.105. 请求账号银证转账银行流水的回调函数
- 4.3.106. 请求账号银证转账银行流水的回调函数
- 4.3.107. 请求账号银证转账银行余额的回调函数
- 4.3.108. 请求账号银证转账银行余额的回调函数
- 4.3.109. 银证转账的回调函数
- 4.3.110. 按市场请求合约信息的回调函数
- 4.3.111. 按市场请求合约信息的回调函数
- 4.3.112. 按市场请求合约信息的回调函数
- 4.3.113. 按市场请求合约信息的回调函数
- 4.3.114. 请求账号可撤单委托明细的回调函数
- 4.3.115. 请求账号可撤单委托明细的回调函数
- 4.3.116. 请求账号可撤单委托明细的回调函数
- 4.3.117. 请求账号可撤单委托明细的回调函数
- 4.3.118. 请求所有下单信息的回调函数
- 4.3.119. 请求所有下单信息的回调函数
- 4.3.120. 资金划拨的回调函数
- 4.3.121. 股份划拨的回调函数
- 4.3.122. 请求账号账号普通柜台资金的回调函数
- 4.3.123. 请求账号账号普通柜台资金的回调函数
- 4.3.124. 请求账号账号普通柜台资金的回调函数

- 4.3.125. 请求账号普通柜台资金的回调函数
- 4.3.126. 请求账号普通柜台持仓的回调函数
- 4.3.127. 请求账号普通柜台持仓的回调函数
- 4.3.128. 请求账号普通柜台持仓的回调函数
- 4.3.129. 请求账号普通柜台持仓的回调函数
- 4.3.130. 请求当前交易日的回调函数
- 4.3.131. 请求账号历史委托明细的回调函数
- 4.3.132. 请求账号历史委托明细的回调函数
- 4.3.133. 请求账号历史委托明细的回调函数
- 4.3.134. 请求账号历史委托明细的回调函数
- 4.3.135. 请求账号历史成交明细的回调函数
- 4.3.136. 请求账号历史成交明细的回调函数
- 4.3.137. 请求账号历史成交明细的回调函数
- 4.3.138. 请求账号历史成交明细的回调函数
- 4.3.139. 请求账号历史持仓统计的回调函数
- 4.3.140. 请求账号历史持仓统计的回调函数
- 4.3.141. 请求账号历史持仓统计的回调函数
- 4.3.142. 请求账号历史持仓统计的回调函数
- 4.3.143. 请求期货账号保证金率的回调函数
- 4.3.144. 请求期货账号手续费率的回调函数
- 4.3.145. 获取用户下所有的产品Id的回调函数
- 4.3.146. 创建新投资组合的回调函数
- 4.3.147. 查询产品Id下所有的投资组合的回调函数
- 4.3.148. 查询产品Id下所有的投资组合的回调函数
- 4.3.149. 请求投资组合委托信息的回调函数
- 4.3.150. 请求投资组合委托信息的回调函数
- 4.3.151. 请求投资组合一段时间内的委托信息的回调函数
- 4.3.152. 请求投资组合一段时间内的委托信息的回调函数
- 4.3.153. 请求账号成交明细的回调函数
- 4.3.154. 请求账号成交明细的回调函数
- 4.3.155. 请求投资组合一段时间内的委托信息的回调函数
- 4.3.156. 请求投资组合一段时间内的委托信息的回调函数
- 4.3.157. 请求投资组合一段时间内的委托信息的回调函数
- 4.3.158. 请求投资组合一段时间内的委托信息的回调函数
- 4.3.159. 请求收益互换账号框架号的回调函数
- 4.3.160. 请求收益互换账号框架号的回调函数
- 4.3.161. 请求股东号的回调函数
- 4.3.162. 请求股东号的回调函数

- 4.3.163. 请求资金账号状态的回调函数
- 4.3.164. 根据投资组合编号请求投资组合资金信息回调接口
- 4.3.165. 市场状态信息回调
- 4.3.166. 请求ETF申赎清单的回调函数
- 4.3.167. 请求ETF成分股信息的回调函数
- 4.3.168. 请求账号持仓统计的回调函数
- 4.3.169. 请求用户设置数据的回调函数

4.4. 主推接口

- 4.4.1. 主推的用户登录状态
- 4.4.2. 主推的用户登录状态
- 4.4.3. 主推的用户登录状态
- 4.4.4. 主推的报单状态（指令）
- 4.4.5. 主推的指令统计信息（指令）
- 4.4.6. 主推的委托明细
- 4.4.7. 主推的成交明细
- 4.4.8. 主推的委托错误信息
- 4.4.9. 主推的撤单失败信息
- 4.4.10. 主推的资金账号信息
- 4.4.11. 主推的两融资金账号信息
- 4.4.12. 主推的产品净值信息
- 4.4.13. 主推行情数据
- 4.4.14. 主推市场状态信息
- 4.4.15. 主推两融综合资金
- 4.4.16. 主推的算法母单错误信息
- 4.4.17. 主推风控信息
- 4.4.18. 主推市场和交易日
- 4.4.19. 退订主推函数
- 4.4.20. 退订主推函数的回调

Q&A 常见问题与解答

- 1. onUserLogin回调里的error包含错误信息: End of file
- 2. onUserLogin回调里的error包含错误信息: md5不匹配禁止登录
- 3. onUserLogin回调里的error包含错误信息: 客户端mac不匹配
- 4. 周末要测试，但是账号休市
- 5. 周一到周五盘后要测试，但是账号休市
- 6. 近场交易说明
- 7. 请求持仓明细和持仓统计说明
- 8. 关于委托成交返回的买卖方向的说明
- 9. 关于期货下单类型今昨仓的的处理

- 10. 特殊字段 m_nLimitedPriceType 说明
- 11. 常见报错说明

1. 概述

1.1. 迅投python版本XtTraderApi

1.1.1. XtTraderApi运行逻辑

XtTraderApi封装了策略交易所需要的Python API接口, 可以和迅投统一交易服务交互, 进行报单、撤单、查询资产、查询委托、 查询成交、查询持仓以及接收委托和成交的实时主推数据。

1.1.2. XtTraderApi运行依赖环境

XtTraderApi需要连接统一交易服务器上的XtApiService服务, 在运行XtTraderApi的程序前需要先保证迅投统一交易服务正常。

1.1.3. XtTraderApi各个组件说明

XtTraderApi.h

- XtTraderApi程序API类, 负责处理用户的请求函数功能, 创建连接, 绑定回调, 初始化, 报单撤单请求, 查询各种数据的请求, 都通过XtTraderApi类传入给服务
- XtTraderApiCallback程序回调函数触发类, 返回所有请求的回调函数, 报单撤单的消息推送通知, 委托明细, 成交明细, 资金明细等数据的通知。

MarketType.h

- 市场代码定义文件

XtDataType.h

- 枚举定义文件

XtStructs.h

- 数据结构定义文件

XtError.h

- XtError类，错误信息，包含错误号和错误描述
- bool isSuccess(), 调用isSuccess判断功能是否报错。
- int errorID(), 获取报错的错误号
- const char* errorMsg(), 获取报错的错误信息

2. 字典和数据结构

2.1. XtTraderApi数据类型和数据结构字典

2.1.1. 交易市场

| 枚举值 | 数据类型 | 含义 |
|---------|--------|----------|
| "SH" | string | 上交所 |
| "SZ" | string | 深交所 |
| "CFFEX" | string | 股指期货 |
| "SHFE" | string | 上海商品期货 |
| "DCE" | string | 大连商品期货 |
| "CZCE" | string | 郑州商品期货 |
| "SHO" | string | 上海期权 |
| "SZO" | string | 深圳期权 |
| "HGT" | string | 沪港通 |
| "SGT" | string | 深港通 |
| "INE" | string | 国际能源中心期货 |
| "GFEX" | string | 广州商品期货 |
| "BJ" | string | 北京证券交易所 |

2.1.2. 账号类型EXTBrokerType

| 枚举变量名 | 值 | 含义 |
|-------------------|----|----------|
| AT_FUTURE | 1 | 期货账号 |
| AT_STOCK | 2 | 股票账号 |
| AT_CREDIT | 3 | 信用账号 |
| AT_GOLD | 4 | 贵金属账号 |
| AT_FUTURE_OPTION | 5 | 期货期权账号 |
| AT_STOCK_OPTION | 6 | 股票期权账号 |
| AT_HUGANGTONG | 7 | 沪港通账号 |
| AT_INCOME_SWAP | 8 | 美股收益互换账号 |
| AT_NEW3BOARD | 10 | 全国股转账号 |
| AT_SHENGANGTONG | 11 | 深港通账号 |
| AT_FICC_COMMODITY | 14 | 期货电子盘账号 |
| AT_FICC_INTEREST | 15 | 利率电子盘账号 |
| AT_ABROAD_FUTURE | 16 | 外盘期货账号 |

2.1.3. 委托类型EOperationType

| 变量名 | 值 | 含义 |
|-------------------------|----|-----|
| OPT_INVALID | -1 | 无效值 |
| 期货业务 | | |
| OPT_OPEN_LONG | 0 | 开多 |
| OPT_CLOSE_LONG_HISTORY | 1 | 平昨多 |
| OPT_CLOSE_LONG_TODAY | 2 | 平今多 |
| OPT_OPEN_SHORT | 3 | 开空 |
| OPT_CLOSE_SHORT_HISTORY | 4 | 平昨空 |

| 变量名 | 值 | 含义 |
|--|----|---------------|
| OPT_CLOSE_SHORT_TODAY | 5 | 平今空 |
| OPT_CLOSE_LONG_TODAY_FIRST | 6 | 平多, 优先 平今 |
| OPT_CLOSE_LONG_HISTORY_FIRST | 7 | 平多, 优先 平昨 |
| OPT_CLOSE_SHORT_TODAY_FIRST | 8 | 平空, 优先 平今 |
| OPT_CLOSE_SHORT_HISTORY_FIRST | 9 | 平空, 优先 平昨 |
| 期货两键 | | |
| OPT_CLOSE_LONG_TODAY_HISTORY_THEN_OPEN_SHORT | 10 | 开空, 优先 平今多 |
| OPT_CLOSE_LONG_HISTORY_TODAY_THEN_OPEN_SHORT | 11 | 开空, 优先 平昨多 |
| OPT_CLOSE_SHORT_TODAY_HISTORY_THEN_OPEN_LONG | 12 | 开多, 优先 平今空 |
| OPT_CLOSE_SHORT_HISTORY_TODAY_THEN_OPEN_LONG | 13 | 开多, 优先 平昨空 |
| OPT_SELL_PRIORITY_OPEN | 14 | 卖出, 优先 开仓 |
| OPT_BUY_PRIORITY_OPEN | 15 | 买入, 优先 开仓 |
| OPT_SELL_OPTIMAL_COMSSION | 16 | 卖出, 最优 手续费 |
| OPT_BUY_OPTIMAL_COMSSION | 17 | 买入, 最优 手续费 |
| 股票业务 | | |
| OPT_BUY | 18 | 买入, 针对 |

| 变量名 | 值 | 含义 |
|-------------------------|----|---------|
| | | 股票 |
| OPT_SELL | 19 | 卖出，针对股票 |
| 信用交易 | | |
| OPT_FIN_BUY | 20 | 融资买入 |
| OPT_SLO_SELL | 21 | 融券卖出 |
| OPT_BUY_SECU_REPAY | 22 | 买券还券 |
| OPT_DIRECT_SECU_REPAY | 23 | 直接还券 |
| OPT_SELL_CASH_REPAY | 24 | 卖券还款 |
| OPT_DIRECT_CASH_REPAY | 25 | 直接还款 |
| OPT_FUND_SUBSCRIBE | 26 | 基金申购 |
| OPT_FUND_REDEMPTION | 27 | 基金赎回 |
| 分级基金 | | |
| OPT_FUND_MERGE | 28 | 基金合并 |
| OPT_FUND_SPLIT | 29 | 基金分拆 |
| 正回购 | | |
| OPT_PLEDGE_IN | 30 | 质押入库 |
| OPT_PLEDGE_OUT | 31 | 质押出库 |
| 期权业务 | | |
| OPT_OPTION_BUY_OPEN | 32 | 买入开仓 |
| OPT_OPTION_SELL_CLOSE | 33 | 卖出平仓 |
| OPT_OPTION_SELL_OPEN | 34 | 卖出开仓 |
| OPT_OPTION_BUY_CLOSE | 35 | 买入平仓 |
| OPT_OPTION_COVERED_OPEN | 36 | 备兑开仓 |

| 变量名 | 值 | 含义 |
|--------------------------------|------|--------|
| OPT_OPTION_COVERED_CLOSE | 37 | 备兑平仓 |
| OPT_OPTION_CALL_EXERCISE | 38 | 认购行权 |
| OPT_OPTION_PUT_EXERCISE | 39 | 认沽行权 |
| OPT_OPTION_SECU_LOCK | 40 | 证券锁定 |
| OPT_OPTION_SECU_UNLOCK | 41 | 证券解锁 |
| 期货期权 | | |
| OPT_FUTURE_OPTION_EXERCISE | 50 | 期货期权行权 |
| OPT_CONVERT_BONDS | 51 | 可转债转股 |
| OPT_SELL_BACK_BONDS | 52 | 可转债回售 |
| 担保品划转 | | |
| OPT_COLLATERAL_TRANSFER_IN | 55 | 担保品划入 |
| OPT_COLLATERAL_TRANSFER_OUT | 56 | 担保品划出 |
| ETF业务 | | |
| OPT ETF_PURCHASE | 1004 | ETF申购 |
| OPT ETF_REDEMPTION | 1005 | ETF赎回 |
| 科创板和创业板盘后定价买卖 | | |
| OPT_AFTER_FIX_BUY | 1043 | 盘后定价买入 |
| OPT_AFTER_FIX_SELL | 1044 | 盘后定价卖出 |
| 期权组合 | | |
| OPT_OPTION_COMB_EXERCISE | 1089 | 组合行权 |
| OPT_OPTION_BUILD_COMB_STRATEGY | 1090 | 构建组合策略 |

| 变量名 | 值 | 含义 |
|----------------------------------|------|-------------|
| OPT_OPTION_RELEASE_COMB_STRATEGY | 1091 | 解除组合策略 |
| 信用专项交易 | | |
| OPT_SLO_SELL_SPECIAL | 1010 | 专项融券卖出 |
| OPT_BUY_SECU_REPAY_SPECIAL | 1011 | 专项买券还券 |
| OPT_DIRECT_SECU_REPAY_SPECIAL | 1012 | 专项直接还券 |
| OPT_FIN_BUY_SPECIAL | 1022 | 专项融资买入 |
| OPT_SELL_CASH_REPAY_SPECIAL | 1023 | 专项卖券还款 |
| OPT_DIRECT_CASH_REPAY_SPECIAL | 1024 | 专项直接还款 |
| 股票期权 | | |
| OPT_OPTION_BUY_CLOSE_THEN_OPEN | 1154 | 股票期权 买入优先平仓 |
| OPT_OPTION_SELL_CLOSE_THEN_OPEN | 1155 | 股票期权 卖出优先平仓 |
| 全国股转 | | |
| OPT_N3B_PRICE_BUY | 42 | 协议转让-定价买入 |
| OPT_N3B_PRICE_SELL | 43 | 协议转让-定价卖出 |
| OPT_N3B_CONFIRM_BUY | 44 | 协议转让-成交确认买入 |

| 变量名 | 值 | 含义 |
|-------------------------------|------|-----------------------|
| OPT_N3B_CONFIRM_SELL | 45 | 协议转让-成交确认卖出 |
| OPT_N3B_REPORT_CONFIRM_BUY | 46 | 协议转让-互报成交确认买入 |
| OPT_N3B_REPORT_CONFIRM_SELL | 47 | 协议转让-互报成交确认卖出 |
| OPT_N3B_LIMIT_PRICE_BUY | 48 | 全国股转-挂牌公司交易-做市转让-限价买入 |
| OPT_N3B_LIMIT_PRICE_SELL | 49 | 全国股转-挂牌公司交易-做市转让-限价卖出 |
| OPT_NEEQ_O3B_LIMIT_PRICE_BUY | 1013 | 全国股转-两网及退市交易-限价买入 |
| OPT_NEEQ_O3B_LIMIT_PRICE_SELL | 1014 | 全国股转-两网及退市交易-限价卖出 |
| OPT_N3B_CALL_AUCTION_BUY | 1027 | 协议转让-集合竞价买入 |
| OPT_N3B_CALL_AUCTION_SELL | 1028 | 协议转让-集合竞价卖 |

| 变量名 | 值 | 含义 |
|--------------------------------------|------|----------------------|
| | | 出 |
| OPT_N3B_AFTER_HOURS_BUY | 1029 | 全国股转- 盘后协议买入 |
| OPT_N3B_AFTER_HOURS_SELL | 1030 | 全国股转- 盘后协议卖出 |
| OPT_NEEQ_O3B_CONTINUOUS_AUCTION_BUY | 1099 | 全国股转- 北交所买入 |
| OPT_NEEQ_O3B_CONTINUOUS_AUCTION_SELL | 1100 | 全国股转- 北交所卖出 |
| OPT_NEEQ_O3B_ASK_PRICE | 1101 | 全国股转- 申购-询价 申报 |
| OPT_NEEQ_O3B_PRICE_CONFIRM | 1102 | 全国股转- 申购-申购 申报 |
| OPT_NEEQ_O3B_BLOCKTRADING_BUY | 1103 | 全国股转- 大宗交易买入 |
| OPT_NEEQ_O3B_BLOCKTRADING_SELL | 1104 | 全国股转- 大宗交易卖出 |

2.1.4. 报价类型EPriceType

| 变量名 | 值 | 含义 |
|--------------|----|-----|
| PRTP_INVALID | -1 | 无效值 |
| PRTP_SALE5 | 0 | 卖5 |
| PRTP_SALE4 | 1 | 卖4 |

| 变量名 | 值 | 含义 |
|--------------------------------|----|-----------------------|
| PRTP_SALE3 | 2 | 卖3 |
| PRTP_SALE2 | 3 | 卖2 |
| PRTP_SALE1 | 4 | 卖1 |
| PRTP_LATEST | 5 | 最新价 |
| PRTP_BUY1 | 6 | 买1 |
| PRTP_BUY2 | 7 | 买2 |
| PRTP_BUY3 | 8 | 买3 |
| PRTP_BUY4 | 9 | 买4 |
| PRTP_BUY5 | 10 | 买5 |
| PRTP_FIX | 11 | 指定价 |
| PRTP_MARKET | 12 | 市价 |
| PRTP_HANG | 13 | 挂单价 跟盘价 |
| PRTP_COMPETE | 14 | 对手价 |
| PRTP_MARKET_BEST | 18 | 期货市价_最优价 郑商所 |
| PRTP_MARKET_CANCEL | 19 | 期货市价_即成剩撤 大商所 |
| PRTP_MARKET_CANCEL_ALL | 20 | 期货市价_全额成交或撤 大商所 |
| PRTP_MARKET_CANCEL_1 | 21 | 期货市价_最优1档即成剩撤 中金所 |
| PRTP_MARKET_CANCEL_5 | 22 | 期货市价_最优5档即成剩撤 中金所 上期所 |
| PRTP_MARKET_CONVERT_1 | 23 | 期货市价_最优1档即成剩转 中金所 |
| PRTP_MARKET_CONVERT_5 | 24 | 期货市价_最优5档即成剩转 中金所 上期所 |
| PRTP_STK_OPTION_FIX_CANCEL_ALL | 26 | 限价即时全部成交否则撤单 - |

| 变量名 | 值 | 含义 |
|------------------------------------|----|---------------------------------------|
| | | 上海/深圳股票期权 |
| PRTP_STK_OPTION_MARKET_CACEL_LEFT | 27 | 市价_即时成交剩余撤单 上海股票期权市价 |
| PRTP_STK_OPTION_MARKET_CANCEL_ALL | 28 | 市价_即时全部成交否则撤单 上海股票期权市价 |
| PRTP_STK_OPTION_MARKET_CONVERT_FIX | 29 | 市价_剩余转限价 上海股票期权市价 |
| PRTP_MARKET_SH_CONVERT_5_CANCEL | 42 | 市价_最优5档即时成交剩余撤销 上海股票市价 |
| PRTP_MARKET_SH_CONVERT_5_LIMIT | 43 | 市价_最优5档即时成交剩转限价 上海股票市价 |
| PRTP_MARKET_PEER_PRICE_FIRST | 44 | 市价_对手方最优价格委托 深圳股票期权和深圳股票市价，可用于上海科创板市价 |
| PRTP_MARKET_MINE_PRICE_FIRST | 45 | 市价_本方最优价格委托 深圳股票期权和深圳股票市价，可用于上海科创板市价 |
| PRTP_MARKET_SZ_INSTBUSI_RESTCANCEL | 46 | 市价_即时成交剩余撤销委托 深圳股票期权和深圳股票市价 |
| PRTP_MARKET_SZ_CONVERT_5_CANCEL | 47 | 市价_最优5档即时成交剩余撤销委托 深圳股票期权和深圳股票市价 |
| PRTP_MARKET_SZ_FULL_REAL_CANCEL | 48 | 市价_全额成交或撤销委托 深圳股票期权和深圳股票市价 |
| PRTP_AFTER_FIX_PRICE | 49 | 盘后定价申报 |
| PRTP_OPTION_COMB_STRATEGY_CNSJC | 50 | 股票期权组合保证金策略-认购牛市价差策略 |
| PRTP_OPTION_COMB_STRATEGY_PXSJC | 51 | 股票期权组合保证金策略-认沽熊市价差策略 |

| 变量名 | 值 | 含义 |
|---------------------------------|----|------------------------|
| PRTP_OPTION_COMB_STRATEGY_PNSJC | 52 | 股票期权组合保证金策略-认沽牛市价差策略 |
| PRTP_OPTION_COMB_STRATEGY_CXSJC | 53 | 股票期权组合保证金策略-认购熊市价差策略 |
| PRTP_OPTION_COMB_STRATEGY_KS | 54 | 股票期权组合保证金策略-跨式空头 |
| PRTP_OPTION_COMB_STRATEGY_KKS | 55 | 股票期权组合保证金策略-宽跨式空头 |
| PRTP_OPTION_COMB_STRATEGY_ZBD | 56 | 股票期权组合保证金策略-保证金开仓转备兑开仓 |
| PRTP_OPTION_COMB_STRATEGY_ZXJ | 57 | 股票期权组合保证金策略-备兑开仓转保证金开仓 |

2.1.5. 委托状态EEntrustStatus

| 变量名 | 值 | 含义 |
|--------------------------------|----|--|
| ENTRUST_STATUS_WAIT_END | 0 | 委托状态已经在 ENTRUST_STATUS_CANCELED 或以上，但是成交数额还不够， 等成交回报来 |
| ENTRUST_STATUS_UNREPORTED | 48 | 未报 |
| ENTRUST_STATUS_WAIT_REPORTING | 49 | 待报 |
| ENTRUST_STATUS_REPORTED | 50 | 已报 |
| ENTRUST_STATUS_REPORTED_CANCEL | 51 | 已报待撤 |
| ENTRUST_STATUS_PARTSUCC_CANCEL | 52 | 部成待撤 |
| ENTRUST_STATUS_PART_CANCEL | 53 | 部撤 |
| ENTRUST_STATUS_CANCELED | 54 | 已撤 |
| ENTRUST_STATUS_PART_SUCC | 55 | 部成 |

| 变量名 | 值 | 含义 |
|---------------------------------|-----|----------------|
| ENTRUST_STATUS_SUCCEEDED | 56 | 已成 |
| ENTRUST_STATUS_JUNK | 57 | 废单 |
| ENTRUST_STATUS_ACCEPT | 58 | 已受理 |
| ENTRUST_STATUS_CONFIRMED | 59 | 已确认 担保品划转已确认状态 |
| ENTRUST_STATUS_DETERMINED | 86 | 已确认 协议回购已确认状态 |
| ENTRUST_STATUS_PREPARE_ORDER | 87 | 预埋 |
| ENTRUST_STATUS_PREPARE_CANCELED | 88 | 预埋已撤 |
| ENTRUST_STATUS_UNKNOWN | 255 | 未知 |

2.1.6. 普通算法单笔基准量类型EVolumeType

| 枚举变量名 | 值 | 含义 |
|------------------------------|----|----------|
| EVolumeType.VOLUME_SALE12345 | 0 | 卖1到卖5量总和 |
| EVolumeType.VOLUME_SALE1234 | 1 | 卖1到卖4量总和 |
| EVolumeType.VOLUME_SALE123 | 2 | 卖1到卖3量总和 |
| EVolumeType.VOLUME_SALE12 | 3 | 卖1到卖2量总和 |
| EVolumeType.VOLUME_SALE1 | 4 | 卖1量 |
| EVolumeType.VOLUME_BUY1 | 5 | 买1量 |
| EVolumeType.VOLUME_BUY12 | 6 | 买1到买2量总和 |
| EVolumeType.VOLUME_BUY123 | 7 | 买1到买3量总和 |
| EVolumeType.VOLUME_BUY1234 | 8 | 买1到买4量总和 |
| EVolumeType.VOLUME_BUY12345 | 9 | 买1到买5量总和 |
| EVolumeType.VOLUME_FIX | 10 | 目标量 |
| EVolumeType.VOLUME_LEFT | 11 | 目标剩余量 |
| EVolumeType.VOLUME_POSITION | 12 | 持仓数量 |

2.1.7. 智能算法类型m_strOrderType

| 算法名称 | 数据类型 | 含义 |
|-----------|--------|--------------|
| "VWAP" | string | VWAP |
| "TWAP" | string | TWAP |
| "VP" | string | 跟量 |
| "PINLINE" | string | 跟价 |
| "FLOAT" | string | 盘口 |
| "ICEBERG" | string | 冰山 |
| "DMA" | string | 快捷 |
| "MOC" | string | 尾盘 |
| "MOO" | string | 开盘 |
| "SWITCH" | string | 调仓 |
| "STWAP" | string | 分时加权平均价格 |
| "VP+" | string | 量加权平均价格（带限价） |
| "XTFAST" | string | 市价单（融资融券） |
| "SLOS" | string | 止损限价单 |
| "SLOH" | string | 止损市价单 |
| "SNIPER" | string | 狙击 |
| "IS" | string | 隔夜单 |
| "VWAP+" | string | 量加权平均价格（带限价） |
| "MOOPLUS" | string | 零点单 |

2.1.8. 主动算法类型m_strOrderType

- 算法具体逻辑可以咨询客户经理

| 算法名称 | 类型 | 含义 |
|------------|--------|------------|
| VWAPA1 | string | 主动VWAP |
| VWAPPLUS | string | 主动VWAPPLUS |
| VWAP_ALPHA | string | 阿尔法VWAP |

2.1.9. 登录状态EBrokerLoginStatus

| 枚举变量名 | 值 | 含义 |
|--|---|------------|
| EBrokerLoginStatus.BROKER_LOGIN_STATUS_OK | 0 | 登录成功，初始化完成 |
| EBrokerLoginStatus.BROKER_LOGIN_STATUS_WAITING_LOGIN | 1 | 连接中 |
| EBrokerLoginStatus.BROKER_LOGIN_STATUS_LOGINING | 2 | 登录中 |
| EBrokerLoginStatus.BROKER_LOGIN_STATUS_FAIL | 3 | 失败 |
| EBrokerLoginStatus.BROKER_LOGIN_STATUS_INITING | 4 | 在初始化中 |
| EBrokerLoginStatus.BROKER_LOGIN_STATUS_CORRECTING | 5 | 数据刷新校正中 |
| EBrokerLoginStatus.BROKER_LOGIN_STATUS_CLOSED | 6 | 收盘后 |

2.1.10. 买卖方向EEntrustBS

| 变量名 | 值 | 含义 |
|--------------------|----|------|
| ENTRUST_BUY | 48 | 买入 |
| ENTRUST_SELL | 49 | 卖出 |
| ENTRUST_COVERED | 50 | 备兑 |
| ENTRUST_PLEDGE_IN | 81 | 质押入库 |
| ENTRUST_PLEDGE_OUT | 66 | 质押出库 |

2.1.11. 指令状态EOrderCommandStatus

| 变量名 | 值 | 含义 |
|-------------------------|----|-----------|
| OCS_CHECKING | -1 | 风控检查中 |
| OCS_APPROVING | 0 | 审批中 |
| OCS_REJECTED | 1 | 已驳回 |
| OCS_RUNNING | 2 | 运行中 |
| OCS_CANCELING | 3 | 撤销中 |
| OCS_FINISHED | 4 | 已完成 |
| OCS_STOPPED | 5 | 已撤销 |
| OCS_FROCE_COMPLETED | 6 | 强制撤销 |
| OCS_CHECKFAILED | 7 | 风控驳回 |
| OCS_CANCELING_APPROVING | 8 | 撤销审批中 |
| OCS_CANCELING_REJECTED | 9 | 撤销驳回 |
| OCS_PAUSE_PAUSEORDER | 14 | 暂停指令并暂停任务 |
| OCS_PAUSE_CANCELORDER | 15 | 暂停指令并撤销任务 |

2.1.12. 委托类型EEntrustTypes

| 变量名 | 值 | 含义 |
|------------------|----|------|
| ENTRUST_BUY_SELL | 48 | 买卖 |
| ENTRUST_QUERY | 49 | 查询 |
| ENTRUST_CANCEL | 50 | 撤单 |
| ENTRUST_APPEND | 51 | 补单 |
| ENTRUST_COMFIRM | 52 | 确认 |
| ENTRUST_BIG | 53 | 大宗 |
| ENTRUST_FIN | 54 | 融资委托 |

| 变量名 | 值 | 含义 |
|---|----|----------|
| ENTRUST_SLO | 55 | 融券委托 |
| ENTRUST_CLOSE | 56 | 信用平仓 |
| ENTRUST_CREDIT_NORMAL | 57 | 信用普通委托 |
| ENTRUST_CANCEL_OPEN | 58 | 撤单补单 |
| ENTRUST_TYPE_OPTION_EXERCISE | 59 | 行权 |
| ENTRUST_TYPE_OPTION_SECU_LOCK | 60 | 锁定 |
| ENTRUST_TYPE_OPTION_SECU_UNLOCK | 61 | 解锁 |
| ENTRUST_TYPE_OPTION_COMB_EXERCISE | 65 | 组合行权 |
| ENTRUST_TYPE_OPTION_BUILD_COMB_STRATEGY | 66 | 构建组合策略持仓 |
| ENTRUST_TYPE_OPTION_RELEASE_COMB_STRATEGY | 67 | 解除组合策略持仓 |

2.1.13. 期货交易类型EFutureTradeType

| 变量名 | 值 | 含义 |
|---------------------------------------|----|-------------------------|
| FUTRUE_TRADE_TYPE_COMMON | 48 | 普通成交 |
| FUTURE_TRADE_TYPE_OPTIONSEXECUTION | 49 | 期权成交 ptionsExecution |
| FUTURE_TRADE_TYPE_OTC | 50 | OTC成交 |
| FUTURE_TRADE_TYPE_EFPDIRVED | 51 | 期转现衍生成交 |
| FUTURE_TRADE_TYPE_COMBINATION_DERIVED | 52 | 组合衍生成交 |

2.1.14. 委托状态EEntrustStatus

| 变量名 | 值 | 含义 |
|-------------------------|---|--|
| ENTRUST_STATUS_WAIT_END | 0 | 委托状态已经在 ENTRUST_STATUS_CANCELED 或以上，但是成交数额还不够， 等成交回报来 |

| 变量名 | 值 | 含义 |
|---------------------------------|-----|----------------|
| ENTRUST_STATUS_UNREPORTED | 48 | 未报 |
| ENTRUST_STATUS_WAIT_REPORTING | 49 | 待报 |
| ENTRUST_STATUS_REPORTED | 50 | 已报 |
| ENTRUST_STATUS_REPORTED_CANCEL | 51 | 已报待撤 |
| ENTRUST_STATUS_PARTSUCC_CANCEL | 52 | 部成待撤 |
| ENTRUST_STATUS_PART_CANCEL | 53 | 部撤 |
| ENTRUST_STATUS_CANCELED | 54 | 已撤 |
| ENTRUST_STATUS_PART_SUCC | 55 | 部成 |
| ENTRUST_STATUS_SUCCEEDED | 56 | 已成 |
| ENTRUST_STATUS_JUNK | 57 | 废单 |
| ENTRUST_STATUS_ACCEPT | 58 | 已受理 |
| ENTRUST_STATUS_CONFIRMED | 59 | 已确认 担保品划转已确认状态 |
| ENTRUST_STATUS_DETERMINED | 86 | 已确认 协议回购已确认状态 |
| ENTRUST_STATUS_PREPARE_ORDER | 87 | 预埋 |
| ENTRUST_STATUS_PREPARE_CANCELED | 88 | 预埋已撤 |
| ENTRUST_STATUS_UNKNOWN | 255 | 未知 |

2.1.15. 期货委托发送状态EEntrustSubmitStatus

| 变量名 | 值 | 含义 |
|---------------------------------------|----|---------|
| ENTRUST_SUBMIT_STATUS_InsertSubmitted | 48 | 已经提交 |
| ENTRUST_SUBMIT_STATUS_CancelSubmitted | 49 | 撤单已经提交 |
| ENTRUST_SUBMIT_STATUS_ModifySubmitted | 50 | 修改已经提交 |
| ENTRUST_SUBMIT_STATUS_OSS_Accepted | 51 | 已经接受 |
| ENTRUST_SUBMIT_STATUS_InsertRejected | 52 | 报单已经被拒绝 |

| 变量名 | 值 | 含义 |
|--------------------------------------|----|---------|
| ENTRUST_SUBMIT_STATUS_CancelRejected | 53 | 撤单已经被拒绝 |
| ENTRUST_SUBMIT_STATUS_ModifyRejected | 54 | 改单已经被拒绝 |

2.1.16. 两融标的状态EXTSubjectsStatus

| 变量名 | 值 | 含义 |
|------------------------|----|-----|
| SUBJECTS_STATUS_NORMAL | 48 | 正常 |
| SUBJECTS_STATUS_PAUSE | 49 | 暂停 |
| SUBJECTS_STATUS_NOT | 50 | 非标的 |

2.1.17. 两融负债状态EXTCompactStatus

| 变量名 | 值 | 含义 |
|---------------------------------|----|-------|
| COMPACT_STATUS_ALL | 32 | 不限制 |
| COMPACT_STATUS_UNDONE | 48 | 未归还 |
| COMPACT_STATUS_PART_DONE | 49 | 部分归还 |
| COMPACT_STATUS_DONE | 50 | 已归还 |
| COMPACT_STATUS_DONE_BY_SELF | 51 | 自行了结 |
| COMPACT_STATUS_DONE_BY_HAND | 52 | 手工了结 |
| COMPACT_STATUS_NOT_DEBT | 53 | 未形成负债 |
| COMPACT_STATUS_EXPIRY_NOT_CLOSE | 54 | 到期未平仓 |

2.1.18. 下单超过流控时处理方式EXtOverFreqOrderMode

| 变量名 | 值 | 含义 |
|------------|---|----|
| OFQ_FORBID | 0 | 禁止 |
| OFQ_QUEUE | 1 | 队列 |

2.1.19. 停牌标志EXtSuspendedType

| 变量名 | 值 | 含义 |
|-----------------|---|-----|
| XT_NO_SUSPENDED | 0 | 不停牌 |
| XT_SUSPENDED | 1 | 停牌 |

2.1.20. 备兑标志ECoveredFlag

| 变量名 | 值 | 含义 |
|--------------------|-----|-----|
| COVERED_FLAG_FALSE | '0' | 非备兑 |
| COVERED_FLAG_TRUE | '1' | 备兑 |

2.1.21. 期权持仓类型ESideFlag

| 变量名 | 值 | 含义 |
|-------------------|-----|----|
| SIDE_FLAG_RIGHT | '0' | 权利 |
| SIDE_FLAG_DUTY | '1' | 义务 |
| SIDE_FLAG_COVERED | '2' | 备兑 |

2.1.22. 订阅行情平台类型EXTOfferStatus

| 变量名 | 值 | 含义 |
|--------------------|----|-----|
| XT_OFFER_STATUS_SP | 48 | 实盘 |
| XT_OFFER_STATUS_MN | 49 | 模拟盘 |

2.1.23. 期货时间条件单类型ETimeCondition

| 变量名 | 值 | 含义 |
|--------------------|-----|------------|
| TIME_CONDITION_IOC | '1' | 立即完成, 否则撤销 |
| TIME_CONDITION_GFS | '2' | 本节有效 |
| TIME_CONDITION_GFD | '3' | 当日有效 |

| 变量名 | 值 | 含义 |
|--------------------|-----|---------|
| TIME_CONDITION_GTD | '4' | 指定日期前有效 |
| TIME_CONDITION_GTC | '5' | 撤销前有效 |
| TIME_CONDITION_GFA | '6' | 集合竞价有效 |

2.1.24. 期货数量条件单类型EVolumeCondition

| 变量名 | 值 | 含义 |
|------------------------|-----|------|
| VOLUME_CONDITION_ANY | '1' | 任何数量 |
| VOLUME_CONDITION_MIN | '2' | 最小数量 |
| VOLUME_CONDITION_TOTAL | '3' | 全部数量 |

2.1.25. 股票期权合约类型EOptionType

| 变量名 | 值 | 含义 |
|------------------|----|------|
| OPTION_TYPE_CALL | 48 | 认购合约 |
| OPTION_TYPE_PUT | 49 | 认沽合约 |

2.1.26. 币种EMoneyType

| 变量名 | 值 | 含义 |
|----------------|----|-----|
| MONEY_TYPE_RMB | 48 | 人民币 |
| MONEY_TYPE_USD | 49 | 美元 |
| MONEY_TYPE_HK | 50 | 港币 |

2.1.27. 任务操作ETaskFlowOperation

| 变量名 | 值 | 含义 |
|------------|---|----|
| TFO_PAUSE | 5 | 暂停 |
| TFO_RESUME | 6 | 恢复 |

2.1.28. 涨跌停控制EStopTradeForOwnHiLow

| 变量名 | 值 | 含义 |
|-----------------------|---|----------|
| STOPTRADE_NONE | 0 | 无 |
| STOPTRADE_NO_BUY_SELL | 1 | 涨停不卖跌停不买 |
| STOPTRADE_SCHEDULED | 2 | 原定时长执行 |
| STOPTRADE_NOT_CHASE | 3 | 涨停不买跌停不卖 |

2.1.29. 两融负债头寸来源EXTCompactBrushSource

| 变量名 | 值 | 含义 |
|------------------------------|----|------|
| COMPACT_BRUSH_SOURCE_ALL | 32 | 不限制 |
| COMPACT_BRUSH_SOURCE_NORMAL | 48 | 普通头寸 |
| COMPACT_BRUSH_SOURCE_SPECIAL | 49 | 专项头寸 |

2.1.30. 触价类型EOpTriggerType

| 变量名 | 值 | 含义 |
|----------|---|-------|
| OTT_NONE | 0 | 不使用触价 |
| OTT_UP | 1 | 向上触价 |
| OTT_DOWN | 2 | 向下触价 |

2.1.31. 市场状态EXTExchangeStatus

| 变量名 | 值 | 含义 |
|-----------------------------------|----|--------|
| EXCHANGE_STATUS_INVALID | 0 | 无效状态 |
| EXCHANGE_STATUS_IN_BEFORE_TRADING | 48 | 开盘前 |
| EXCHANGE_STATUS_NOTRADING | 49 | 非交易 |
| EXCHANGE_STATUS_IN_CONTINUOUS | 50 | 连续交易 |
| EXCHANGE_STATUS_AUCTION_ORDERING | 51 | 集合竞价报单 |

| 变量名 | 值 | 含义 |
|---|----|------------|
| EXCHANGE_STATUS_AUCTION_BALANCE | 52 | 集合竞价价格平衡 |
| EXCHANGE_STATUS_AUCTION_MATCH | 53 | 集合竞价撮合 |
| EXCHANGE_STATUS_IN_CLOSED | 54 | 收盘 |
| EXCHANGE_STATUS_ONLY_CANCEL | 55 | 仅允许撤单 |
| EXCHANGE_STATUS_DAZONG_ORDERING | 56 | 大宗交易申报 |
| EXCHANGE_STATUS_DAZONG_INTENTION_ORDERING | 57 | 大宗交易意向申报 |
| EXCHANGE_STATUS_DAZONG_CONFIRM_ORDERING | 58 | 大宗交易成交申报 |
| EXCHANGE_STATUS_DAZONG_CLOSE_PRICE_ORDERING | 59 | 大宗交易盘后定价申报 |
| EXCHANGE_STATUS_ONLY_GOLD_DELIVERY | 60 | 交割申报 |
| EXCHANGE_STATUS_ONLY_GOLD_MIDDLE | 61 | 中立仓申报 |
| EXCHANGE_STATUS_AFTER_HOURS_SALE | 62 | 盘后协议买卖 |
| EXCHANGE_STATUS_CLOSING_AUCTION_MATCH | 63 | 收盘集合竞价 |
| EXCHANGE_STATUS_CLOSE_PRICE_ORDERING | 64 | 盘后定价申报 |

2.1.32. 是否使用大额单边保证金算法 EXtMaxMarginSideAlgorithmType

| 变量名 | 值 | 含义 |
|------------------|----|--------------|
| XT_FTDC_MMSA_NO | 48 | 不使用大额单边保证金算法 |
| XT_FTDC_MMSA_YES | 49 | 使用大额单边保证金算法 |

2.1.33. 普通算法订单类型EAlgoPriceType

| 变量名 | 值 | 含义 |
|------------------|---|----|
| EALGO_PRT_MARKET | 0 | 市价 |
| EALGO_PRT_FIX | 1 | 限价 |

2.1.34. 银证转账方向ETransDirection

| 变量名 | 值 | 含义 |
|------------------------------------|----|-------|
| TRANS_DIRECTION_BANK_TO_SECURITIES | 49 | 银行转证券 |
| TRANS_DIRECTION_SECURITIES_TO_BANK | 50 | 证券转银行 |
| TRANS_DIRECTION_QUICK_TO_CENTER | 51 | 快速转集中 |
| TRANS_DIRECTION_CENTER_TO_QUICK | 52 | 集中转快速 |
| TRANS_DIRECTION_QUERY | 53 | 查询 |

2.1.35. 算法下单方式EOrderStrategyType

| 变量名 | 值 | 含义 |
|--------------------------------|----|-------------|
| E_ORDER_STRATEGY_TYPE_NORMAL | -1 | 普通 |
| E_ORDER_STRATEGY_TYPE_APPROACH | 0 | 近场交易，需要服务支持 |

2.1.36. 双中心状态 EDualStatus

| 变量名 | 值 | 含义 |
|--------------------|----|-----------|
| E_DUAL_STATUS_NONE | -1 | 无双中心 |
| E_DUAL_STATUS_SH | 0 | 仅有上海 |
| E_DUAL_STATUS_SZ | 1 | 仅有深圳 |
| E_DUAL_STATUS_ALL | 2 | 上海深圳双中心均有 |

2.1.37. 股份划拨信用划拨类别 ETransTypeCreditFlag

| 变量名 | 值 | 含义 |
|----------------------------------|---|----------------|
| TRANS_TRANSFER_SHARE | 0 | 操作客户股份划拨, 默认送0 |
| TRANS_TRANSFER_SPECIAL_POSITIONS | 1 | 操作客户专向头寸调拨 |
| TRANS_TRANSFER_CREDIT_SHARE | 2 | 融资融券客户股份调拨 |

2.1.38. 资金股份划拨划拨方向 ESecuFundTransDirection

| 变量名 | 值 | 含义 |
|---------------------------------------|---|--------------|
| SECUFUNDTRANS_TRANSFER_NORMAL_TO_FAST | 0 | 从普通柜台拨入到极速柜台 |
| SECUFUNDTRANS_TRANSFER_FAST_TO_NORMAL | 1 | 从极速柜台拨出到普通柜台 |
| SECUFUNDTRANS_TRANSFER_SH_TO_SZ | 2 | 上海划到深圳(节点划拨) |
| SECUFUNDTRANS_TRANSFER_SZ_TO_SH | 3 | 深圳划到上海(节点划拨) |

2.1.39. 委托明细价格类型 EBrokerPriceType

| 枚举变量名 | 值 | 含义 |
|-----------------------------|----|-----|
| BROKER_PRICE_ANY | 49 | 市价 |
| BROKER_PRICE_LIMIT | 50 | 限价 |
| BROKER_PRICE_BEST | 51 | 最优价 |
| BROKER_PRICE_PROP_ALLOTMENT | 52 | 配股 |
| BROKER_PRICE_PROP_REFER | 53 | 转托 |
| BROKER_PRICE_PROP_SUBSCRIBE | 54 | 申购 |
| BROKER_PRICE_PROP_BUYBACK | 55 | 回购 |
| BROKER_PRICE_PROP_PLACING | 56 | 配售 |
| BROKER_PRICE_PROP_DECIDE | 57 | 指定 |

| 枚举变量名 | 值 | 含义 |
|------------------------------------|----|----------------|
| BROKER_PRICE_PROP_EQUITY | 58 | 转股 |
| BROKER_PRICE_PROP_SELLBACK | 59 | 回售 |
| BROKER_PRICE_PROP_DIVIDEND | 60 | 股息 |
| BROKER_PRICE_PROP_SHENZHEN_PLACING | 68 | 深圳 配售 确认 |
| BROKER_PRICE_PROP_CANCEL_PLACING | 69 | 配售 放弃 |
| BROKER_PRICE_PROP_WDZY | 70 | 无冻 质押 |
| BROKER_PRICE_PROP_DJZY | 71 | 冻结 质押 |
| BROKER_PRICE_PROP_WDJY | 72 | 无冻 解押 |
| BROKER_PRICE_PROP_JDJY | 73 | 解冻 解押 |
| BROKER_PRICE_PROP_VOTE | 75 | 投票 |
| BROKER_PRICE_PROP YYSYGYS | 76 | 要约 收购 预售 |
| BROKER_PRICE_PROP_YSYYJC | 77 | 预售 要约 解除 |
| BROKER_PRICE_PROP_FUND_DEVIDEND | 78 | 基金 设红 |
| BROKER_PRICE_PROP_FUND_ENTRUST | 79 | 基金 申赎 |
| BROKER_PRICE_PROP_CROSS_MARKET | 80 | 跨市 转托 |

| 枚举变量名 | 值 | 含义 |
|---------------------------------------|----|---------------------------------------|
| BROKER_PRICE_PROP ETF | 81 | ETF 申购 |
| BROKER_PRICE_PROP EXERCIS | 83 | 权证 行权 |
| BROKER_PRICE_PROP_PEER_PRICE_FIRST | 91 | 对手 方最 优价 格 |
| BROKER_PRICE_PROP_L5_FIRST_LIMITPX | 92 | 最优 五档 即时 成交 剩余 转限 价 |
| BROKER_PRICE_PROP_MIME_PRICE_FIRST | 93 | 本方 最优 价格 |
| BROKER_PRICE_PROP_INSTBUSI_RESTCANCEL | 94 | 即时 成交 剩余 撤销 |
| BROKER_PRICE_PROP_L5_FIRST_CANCEL | 95 | 最优 五档 即时 成交 剩余 撤销 |
| BROKER_PRICE_PROP_FULL_REAL_CANCEL | 96 | 全额 成交 并撤 单 |

| 枚举变量名 | 值 | 含义 |
|-------------------------------------|-----|---------------------------------|
| BROKER_PRICE_PROP_FUND_CHAIHE | 90 | 基金 拆合 |
| BROKER_PRICE_PROP_MARKET | 130 | 市价 _涨 跌停 价 |
| BROKER_PRICE_PROP_MARKET_BEST | 131 | 市价 _最 优价 |
| BROKER_PRICE_PROP_MARKET_CANCEL | 132 | 市价 _即 成剩 撤 |
| BROKER_PRICE_PROP_MARKET_CANCEL_ALL | 133 | 市价 _全 额成 交或 撤 |
| BROKER_PRICE_PROP_MARKET_CANCEL_1 | 134 | 市价 _最 优1 档即 成剩 撤 |
| BROKER_PRICE_PROP_MARKET_CANCEL_5 | 135 | 市价 _最 优5 档即 成剩 撤 |
| BROKER_PRICE_PROP_MARKET_CONVERT_1 | 136 | 市价 _最 |

| 枚举变量名 | 值 | 含义 |
|--|-----|---|
| | | 优1 档即 成剩 转 |
| BROKER_PRICE_PROP_MARKET_CONVERT_5 | 137 | 市价 _最 优5 档即 成剩 转 |
| BROKER_PRICE_PROP_STK_OPTION_ASK | 138 | 股票 期权 -询 价 |
| BROKER_PRICE_PROP_STK_OPTION_FIX_CANCEL_ALL | 139 | 股票 期权 -限 价即 时全 部成 交否 则撤 单 |
| BROKER_PRICE_PROP_STK_OPTION_MARKET_CACEL_LEFT | 140 | 股票 期权 -市 价即 时成 交剩 余撤 单 |
| BROKER_PRICE_PROP_STK_OPTION_MARKET_CANCEL_ALL | 141 | 股票 期权 |

| 枚举变量名 | 值 | 含义 |
|--|-----|---|
| | | -市 价即 时全 部成 交否 则撤 单 |
| BROKER_PRICE_PROP_STK_OPTION_MARKET_CONVERT_FIX | 142 | 股票 期权 -市 价剩 余转 限价 |
| BROKER_PRICE_PROP_OPTION_COMB_EXERCISE | 164 | 股票 期权 -组 合行 权 |
| BROKER_PRICE_PROP_OPTION_BUILD_COMB_STRATEGY_CNSJC | 165 | 股票 期权 -构 建认 购牛 市价 差策 略 |
| BROKER_PRICE_PROP_OPTION_BUILD_COMB_STRATEGY_PXSJC | 166 | 股票 期权 -构 建认 沽熊 市价 差策 略 |

| 枚举变量名 | 值 | 含义 |
|--|-----|-----------------|
| BROKER_PRICE_PROP_OPTION_BUILD_COMB_STRATEGY_PNSJC | 167 | 股票期权-构建认沽牛市价差策略 |
| BROKER_PRICE_PROP_OPTION_BUILD_COMB_STRATEGY_CXSJC | 168 | 股票期权-构建认购熊市价差策略 |
| BROKER_PRICE_PROP_OPTION_BUILD_COMB_STRATEGY_KS | 169 | 股票期权-构建跨式空头策略 |
| BROKER_PRICE_PROP_OPTION_BUILD_COMB_STRATEGY_KKS | 170 | 股票期权-构建宽跨式空头策略 |
| BROKER_PRICE_PROP_OPTION_BUILD_COMB_STRATEGY_ZBD | 171 | 股票期权-普通转 |

| 枚举变量名 | 值 | 含义 |
|--|-----|-----------------|
| | | 备兑 |
| BROKER_PRICE_PROP_OPTION_BUILD_COMB_STRATEGY_ZXJ | 172 | 股票期权-备兑转普通 |
| BROKER_PRICE_PROP_OPTION_RELEASE_COMB_STRATEGY_CNSJC | 173 | 股票期权-解除认购牛市价差策略 |
| BROKER_PRICE_PROP_OPTION_RELEASE_COMB_STRATEGY_PXSJC | 174 | 股票期权-解除认购熊市价差策略 |
| BROKER_PRICE_PROP_OPTION_RELEASE_COMB_STRATEGY_PNSJC | 175 | 股票期权-解除认购牛市价差策略 |
| BROKER_PRICE_PROP_OPTION_RELEASE_COMB_STRATEGY_CXSJC | 176 | 股票期权-解 |

| 枚举变量名 | 值 | 含义 |
|--|-----|----------------|
| | | 除认购熊市价差策略 |
| BROKER_PRICE_PROP_OPTION_RELEASE_COMB_STRATEGY_KS | 177 | 股票期权-解除跨式空头策略 |
| BROKER_PRICE_PROP_OPTION_RELEASE_COMB_STRATEGY_KKS | 178 | 股票期权-解除宽跨式空头策略 |

2.1.40. 开平标志 EOffsetFlagType

| 枚举变量名 | 值 | 含义 |
|------------------------------------|----|------|
| EOFF_THOST_FTDC_OF_INVALID | -1 | 未知 |
| EOFF_THOST_FTDC_OF_Open | 48 | 开仓 |
| EOFF_THOST_FTDC_OF_Close | 49 | 平仓 |
| EOFF_THOST_FTDC_OF_ForceClose | 50 | 强平 |
| EOFF_THOST_FTDC_OF_CloseToday | 51 | 平今 |
| EOFF_THOST_FTDC_OF_CloseYesterday | 52 | 平昨 |
| EOFF_THOST_FTDC_OF_ForceOff | 53 | 强减 |
| EOFF_THOST_FTDC_OF_LocalForceClose | 54 | 本地强平 |

| 枚举变量名 | 值 | 含义 |
|------------------------------------|----|------|
| EOFF_THOST_FTDC_OF_PLEDGE_IN | 81 | 质押入库 |
| EOFF_THOST_FTDC_OF_PLEDGE_OUT | 66 | 质押出库 |
| EOFF_THOST_FTDC_OF_ALLOTMENT | 67 | 股票配股 |
| EOFF_THOST_FTDC_OF_OPTION_EXERCISE | 68 | 行权 |
| EOFF_THOST_FTDC_OF_OPTION_LOCK | 69 | 锁定 |
| EOFF_THOST_FTDC_OF_OPTION_UNLOCK | 70 | 解锁 |

2.1.41. 投保标志 EHedgeFlagType

| 枚举变量名 | 值 | 含义 |
|------------------------|----|----|
| HEDGE_FLAG_SPECULATION | 49 | 投机 |
| HEDGE_FLAG_ARBITRAGE | 50 | 套利 |
| HEDGE_FLAG_HEDGE | 51 | 套保 |

2.1.42. 股票状态 EStockStatus

| 枚举变量名 | 值 | 含义 |
|------------------------------------|---|--------|
| STOCK_STATUS_DEFAULT | 0 | 默认状态 |
| STOCK_STATUS_BEFORE_TRADING | 1 | 开盘前 |
| STOCK_STATUS_AUCTION | 2 | 集合竞价 |
| STOCK_STATUS_CONTINUOUS | 3 | 连续交易 |
| STOCK_STATUS_BREAK | 4 | 休市 |
| STOCK_STATUS_CLOSED | 5 | 闭市 |
| STOCK_STATUS_VOLATILITY_DISRUPTION | 6 | 波动性中断 |
| STOCK_STATUS_SIDEAR | 7 | 临时停牌 |
| STOCK_STATUS_CLOSING_BIDDING | 8 | 收盘集合竞价 |
| STOCK_STATUS_CALL_AUCTION | 9 | 盘中集合竞价 |

| 枚举变量名 | 值 | 含义 |
|-------------------------------|----|------------|
| STOCK_STATUS_HALT_TRADING | 10 | 暂停交易至闭市 |
| STOCK_STATUS_ERROR | 11 | 获取状态异常 |
| STOCK_STATUS_POST_TRADE | 12 | 盘后固定价格交易 |
| STOCK_STATUS_POST_TRADE_CLOSE | 13 | 盘后固定价格行情完毕 |
| STOCK_STATUS_CLOSE_REMOVE | 14 | 盘后固定价格恢复闭市 |
| STOCK_STATUS_FAKED | 15 | 伪造的状态 |

2.1.43. 两融负债类型 EXTCompactType

| 枚举变量名 | 值 | 含义 |
|------------------|----|-----|
| COMPACT_TYPE_ALL | 32 | 不限制 |
| COMPACT_TYPE_FIN | 48 | 融资 |
| COMPACT_TYPE_SLO | 49 | 融券 |

2.1.44. 两融负债状态 EXTCompactStatus

| 枚举变量名 | 值 | 含义 |
|---------------------------------|----|-------|
| COMPACT_STATUS_ALL | 32 | 不限制 |
| COMPACT_STATUS_UNDONE | 48 | 未归还 |
| COMPACT_STATUS_PART_DONE | 49 | 部分归还 |
| COMPACT_STATUS_DONE | 50 | 已归还 |
| COMPACT_STATUS_DONE_BY_SELF | 51 | 自行了结 |
| COMPACT_STATUS_DONE_BY_HAND | 52 | 手工了结 |
| COMPACT_STATUS_NOT_DEBT | 53 | 未形成负债 |
| COMPACT_STATUS_EXPIRY_NOT_CLOSE | 54 | 到期未平仓 |

2.1.45. 备兑标志 ECoveredFlag

| 枚举变量名 | 值 | 含义 |
|--------------------|-----|-----|
| COVERED_FLAG_FALSE | '0' | 非备兑 |
| COVERED_FLAG_TRUE | '1' | 备兑 |

2.1.46. 股票期权合约类型 EOptionType

| 枚举变量名 | 值 | 含义 |
|------------------|----|------|
| OPTION_TYPE_CALL | 48 | 认购合约 |
| OPTION_TYPE_PUT | 49 | 认沽合约 |

2.1.47. 期权持仓类型 ESideFlag

| 变量名 | 值 | 含义 |
|-------------------|-----|----|
| SIDE_FLAG_RIGHT | '0' | 权利 |
| SIDE_FLAG_DUTY | '1' | 义务 |
| SIDE_FLAG_COVERED | '2' | 备兑 |

2.1.48. 涨跌停控制 EStopTradeForOwnHiLow

| 变量名 | 值 | 含义 |
|-----------------------|---|---------------------|
| STOPTRADE_NONE | 0 | 主动算法是无，智能算法是最优价尽快执行 |
| STOPTRADE_NO_BUY_SELL | 1 | 涨停不卖跌停不买 |
| STOPTRADE_SCHEDULED | 2 | 原定时长执行 |
| STOPTRADE_NOT_CHASE | 3 | 涨停不买跌停不卖 |

2.1.49. 两融负债头寸来源 EXTCompactBrushSource

| 枚举变量名 | 值 | 含义 |
|--------------------------|----|-----|
| COMPACT_BRUSH_SOURCE_ALL | 32 | 不限制 |

| 枚举变量名 | 值 | 含义 |
|------------------------------|----|------|
| COMPACT_BRUSH_SOURCE_NORMAL | 48 | 普通头寸 |
| COMPACT_BRUSH_SOURCE_SPECIAL | 49 | 专项头寸 |

2.1.50. 触价类型 EOpTriggerType

| 变量名 | 值 | 含义 |
|----------|---|-------|
| OTT_NONE | 0 | 不使用触价 |
| OTT_UP | 1 | 向上触价 |
| OTT_DOWN | 2 | 向下触价 |

2.1.51. 市场状态 EXTExchangeStatus

| 变量名 | 值 | 含义 |
|---|----|----------|
| EXCHANGE_STATUS_INVALID | 0 | 不使用触价 |
| EXCHANGE_STATUS_IN_BEFORE_TRADING | 48 | 开盘前 |
| EXCHANGE_STATUS_NOTRADING | 49 | 非交易 |
| EXCHANGE_STATUS_IN_CONTINUOUS | 50 | 连续交易 |
| EXCHANGE_STATUS_AUCTION_ORDERING | 51 | 集合竞价报单 |
| EXCHANGE_STATUS_AUCTION_BALANCE | 52 | 集合竞价价格平衡 |
| EXCHANGE_STATUS_AUCTION_MATCH | 53 | 集合竞价撮合 |
| EXCHANGE_STATUS_IN_CLOSED | 54 | 收盘 |
| EXCHANGE_STATUS_ONLY_CANCEL | 55 | 仅允许撤单 |
| EXCHANGE_STATUS_DAZONG_ORDERING | 56 | 大宗交易申报 |
| EXCHANGE_STATUS_DAZONG_INTENTION_ORDERING | 57 | 大宗交易意向申报 |
| EXCHANGE_STATUS_DAZONG_CONFIRM_ORDERING | 58 | 大宗交易成交申报 |

| 变量名 | 值 | 含义 |
|---|----|------------|
| EXCHANGE_STATUS_DAZONG_CLOSE_PRICE_ORDERING | 59 | 大宗交易盘后定价申报 |
| EXCHANGE_STATUS_ONLY_GOLD_DELIVERY | 60 | 交割申报 |
| EXCHANGE_STATUS_ONLY_GOLD_MIDDLE | 61 | 中立仓申报 |
| EXCHANGE_STATUS_AFTER_HOURS_SALE | 62 | 盘后协议买卖 |
| EXCHANGE_STATUS_CLOSING_AUCTION_MATCH | 63 | 收盘集合竞价 |
| EXCHANGE_STATUS_CLOSE_PRICE_ORDERING | 64 | 盘后定价申报 |

2.1.52. 银证转账方向 ETransDirection

| 变量名 | 值 | 含义 |
|------------------------------------|----|-------|
| TRANS_DIRECTION_BANK_TO_SECURITIES | 49 | 银行转证券 |
| TRANS_DIRECTION_SECURITIES_TO_BANK | 51 | 证券转银行 |
| TRANS_DIRECTION_QUICK_TO_CENTER | 51 | 快速转集中 |
| TRANS_DIRECTION_CENTER_TO_QUICK | 52 | 集中转快速 |
| TRANS_DIRECTION_QUERY | 53 | 查询 |

2.1.53. 算法下单方式 EOrderStrategyType

| 变量名 | 值 | 含义 |
|--------------------------------|----|-------------|
| E_ORDER_STRATEGY_TYPE_NORMAL | -1 | 普通 |
| E_ORDER_STRATEGY_TYPE_APPROACH | 0 | 近场交易，需要服务支持 |

2.1.54. 股东号主副标记 EMainFlag

| 变量名 | 值 | 含义 |
|----------------|----|-----|
| MAIN_FLAG_VICE | 48 | 副账户 |
| MAIN_FLAG_MAIN | 49 | 主账户 |

2.1.55. 证券类别 EStockType

| 枚举变量名 | 值 | 含义 |
|----------------------------|----|-------------------------------|
| 大类 | | |
| MINOR_KIND_UNDEFINED | 0 | 默认 |
| MAJOR_KIND_INDEX | 1 | 指数 |
| MAJOR_KIND_STOCK | 2 | 股票 |
| MAJOR_KIND_FUND | 3 | 基金 |
| MAJOR_KIND_BOND | 4 | 债券 |
| MAJOR_KIND_WARRANT | 5 | 权证 |
| MAJOR_KIND_REPO | 6 | 质押式 回购交易 |
| MAJOR_KIND_OPTION | 7 | 期权 |
| MAJOR_KIND_PREFERRED_STOCK | 8 | 优先股 |
| MAJOR_KIND_ABS | 9 | 资产支持 证券 |
| MAJOR_KIND_FJY | 10 | 非交易 |
| MAJOR_KIND_REIT | 11 | 房地产 信托投 资基金 (REITs) |
| 上海 | | 可交易 类 |
| SH_MINOR_KIND_STOCK_ASH | 12 | 以人民 币交易 的股票 (主 板) |

| 枚举变量名 | 值 | 含义 |
|-------------------------|----|---------------------|
| SH_MINOR_KIND_STOCK_BSH | 13 | 以美元交易的股票 |
| SH_MINOR_KIND_STOCK_KSH | 14 | 以人民币交易的股票 (科创板) |
| SH_MINOR_KIND_STOCK_OEQ | 15 | 其它股票 |
| SH_MINOR_KIND_FUND_CEF | 16 | 封闭式基金 |
| SH_MINOR_KIND_FUND_OEF | 17 | 开放式基金 |
| SH_MINOR_KIND_FUND_EBS | 18 | ETF基金 |
| SH_MINOR_KIND_FUND_FBL | 19 | 跨市场 / 跨境基金 |
| SH_MINOR_KIND_FUND_OFN | 20 | 其它基金 |
| SH_MINOR_KIND_FUND_LOF | 21 | LOF基金 |
| SH_MINOR_KIND_BOND_GBF | 22 | 国债、地方债、政府支持债、政策性金融债 |
| SH_MINOR_KIND_BOND_GBZ | 23 | 无息国 |

| 枚举变量名 | 值 | 含义 |
|------------------------|----|--------------------|
| | | 债 (弃用) |
| SH_MINOR_KIND_BOND_DST | 24 | 国债分销 (仅用于分销阶段) |
| SH_MINOR_KIND_BOND_DVP | 25 | 公司债 (地方债) 分销 |
| SH_MINOR_KIND_BOND_CBF | 26 | 企业债券 |
| SH_MINOR_KIND_BOND_CCF | 27 | 可转换企业债券 |
| SH_MINOR_KIND_BOND_CPF | 28 | 公司债、企业债、可交换债、政府支持债 |
| SH_MINOR_KIND_BOND_FBF | 29 | 金融机构发行债券 (弃用) |
| SH_MINOR_KIND_BOND_CRP | 30 | 通用质押式回购 |
| SH_MINOR_KIND_BOND_BRP | 31 | 质押式企债回 |

| 枚举变量名 | 值 | 含义 |
|------------------------|----|---|
| | | 购 (弃用) |
| SH_MINOR_KIND_BOND_ORP | 32 | 买断式 债券回 购 (弃 用) |
| SH_MINOR_KIND_BOND_CBD | 33 | 分离式 可转债 (弃用) |
| SH_MINOR_KIND_BOND_OBD | 34 | 其它债 券 |
| SH_MINOR_KIND_BOND_WIT | 35 | 国债预 发行 |
| SH_MINOR_KIND_AMP | 36 | 集合资 产管理 计划 |
| SH_MINOR_KIND_OPS | 37 | 公开发 行优先 股 |
| SH_MINOR_KIND_PPS | 38 | 非公开 发行优 先股 |
| SH_MINOR_KIND_QRP | 39 | 报价回 购 |
| SH_MINOR_KIND_CMD | 40 | 控制指 令 (中 登身份 认证密 码服务 产品复 用) |

| 枚举变量名 | 值 | 含义 |
|----------------------|----|---------------------------------|
| | | CMD 证券子 类别) |
| 上海 | | 非交易 类 |
| SH_FJY_MINOR_KIND_IN | 41 | 上网证 券发行 申购 |
| SH_FJY_MINOR_KIND_IS | 42 | 老股东 增发证 券发行 申购 |
| SH_FJY_MINOR_KIND_PH | 43 | IPO配 号 |
| SH_FJY_MINOR_KIND_KK | 44 | IPO扣 款 |
| SH_FJY_MINOR_KIND_HK | 45 | IPO还 款 |
| SH_FJY_MINOR_KIND_CV | 46 | 可转债 转股 / 可交换 公司债 换股 |
| SH_FJY_MINOR_KIND_CR | 47 | 可转债 回售 |
| SH_FJY_MINOR_KIND_R1 | 48 | 股票配 股行权 行权 |
| SH_FJY_MINOR_KIND_R2 | 49 | 股票转 配股配 股行权 |

| 枚举变量名 | 值 | 含义 |
|----------------------|----|------------------------|
| SH_FJY_MINOR_KIND_R3 | 50 | 职工股 转配股 配股行 权 |
| SH_FJY_MINOR_KIND_R4 | 51 | 股票配 转债行 权 |
| SH_FJY_MINOR_KIND_OS | 52 | 开放式 基金认 购 |
| SH_FJY_MINOR_KIND_OC | 53 | 开放式 基金申 购 |
| SH_FJY_MINOR_KIND_OR | 54 | 开放式 基金赎 回 |
| SH_FJY_MINOR_KIND_OD | 55 | 开放式 基金分 红选择 |
| SH_FJY_MINOR_KIND_OT | 56 | 开放式 基金份 额转出 |
| SH_FJY_MINOR_KIND_OV | 57 | 开放式 基金转 换 |
| SH_FJY_MINOR_KIND_EC | 58 | ETF申 购 |
| SH_FJY_MINOR_KIND_ER | 59 | ETF赎 回 |
| SH_FJY_MINOR_KIND_EZ | 60 | 申赎资 金代码 |

| 枚举变量名 | 值 | 含义 |
|----------------------|----|----------|
| SH_FJY_MINOR_KIND_BD | 61 | 回购入库 |
| SH_FJY_MINOR_KIND_BW | 62 | 回购出库 |
| SH_FJY_MINOR_KIND_FS | 63 | 要约预受 |
| SH_FJY_MINOR_KIND_FC | 64 | 要约撤销 |
| SH_FJY_MINOR_KIND_ST | 65 | 余券划转 |
| SH_FJY_MINOR_KIND_SR | 66 | 还券划转 |
| SH_FJY_MINOR_KIND_CI | 67 | 担保品划入 |
| SH_FJY_MINOR_KIND_CO | 68 | 担保品划出 |
| SH_FJY_MINOR_KIND_SI | 69 | 券源划入 |
| SH_FJY_MINOR_KIND_SO | 70 | 券源划出 |
| SH_FJY_MINOR_KIND_PA | 71 | 密码激活(注销) |
| SH_FJY_MINOR_KIND_DT | 72 | 指定登记 |
| SH_FJY_MINOR_KIND_DC | 73 | 指定撤销 |
| SH_FJY_MINOR_KIND_QT | 74 | 其它 |
| 上海 | | 新增可 |

| 枚举变量名 | 值 | 含义 |
|-------------------------------------|----|------------------|
| | | 交易类 |
| SH_MINOR_KIND_FUND_KES | 75 | 科创板 ETF基金 |
| SH_MINOR_KIND_FUND_KOF | 76 | 科创板 LOF基金 |
| SH_MINOR_KIND_BOND_TCB | 77 | 定向可 转债 |
| SH_MINOR_KIND_BOND_KCCF | 78 | 科创板 可转债 |
| SH_MINOR_KIND_REITS_RET | 79 | REITs |
| 深圳 | | 可交易 类 |
| SZ_MINOR_KIND_STOCK_ZHU_BAN_A | 80 | 主板 A 股 |
| SZ_MINOR_KIND_STOCK_ZHONG_XIAO_BAN | 81 | 中小板 股票 |
| SZ_MINOR_KIND_STOCK_CHUANG_YE_BAN | 82 | 创业板 股票 |
| SZ_MINOR_KIND_STOCK_ZHU_BAN_B | 83 | 主板 B 股 |
| SZ_MINOR_KIND_BOND_TREASURY_BOND | 84 | 国债 (含地 方债) |
| SZ_MINOR_KIND_BOND_ENTERPRISE_BOND | 85 | 企业债 |
| SZ_MINOR_KIND_BOND_CORPORATE_BOND | 86 | 公司债 |
| SZ_MINOR_KIND_BOND_CONVERTIBLE_BOND | 87 | 可转债 |

| 枚举变量名 | 值 | 含义 |
|---|-----|---------------------|
| SZ_MINOR_KIND_BOND_PRIVATELY_RAISED_COMPANY_BONDS | 88 | 私募债 |
| SZ_MINOR_KIND_BOND_EXCHANGEABLE_PB | 89 | 可交换 私募债 |
| SZ_MINOR_KIND_BOND_SECURITY_COMPANY_SUB_DEBT | 90 | 证券公 司次级 债 |
| SZ_MINOR_KIND_REPO_PLEDGE_STYLE_REPO | 91 | 质押式 回购 |
| SZ_MINOR_KIND_ASSET_BACKED_SECURITIES | 92 | 资产支 持证券 |
| SZ_MINOR_KIND_FUND_STOCK_ETF | 93 | 本市场 股票 ETF |
| SZ_MINOR_KIND_FUND_INTER_MARKET_STOCK_ETF | 94 | 跨市场 股票 ETF |
| SZ_MINOR_KIND_FUND_CROSS_BORDER_ETF | 95 | 跨境 ETF |
| SZ_MINOR_KIND_FUND_BEARER_BOND_ETF | 96 | 本市场 实物债 券 ETF |
| SZ_MINOR_KIND_FUND_CASH_BOND_ETF | 97 | 现金债 券 ETF |
| SZ_MINOR_KIND_FUND_GOLD_ETF | 98 | 黄金 ETF |
| SZ_MINOR_KIND_FUND_CURRENCY_ETF | 99 | 货币 ETF |
| SZ_MINOR_KIND_FUND_LEVER_ETF | 100 | 杠杆 ETF |

| 枚举变量名 | 值 | 含义 |
|---|-----|------------|
| SZ_MINOR_KIND_FUND_COMMODITY_FUTURES ETF | 101 | 商品期货ETF |
| SZ_MINOR_KIND_FUND_STANDARD_LOF | 102 | 标准LOF |
| SZ_MINOR_KIND_FUND_GRADED_SUB_FUNDS | 103 | 分级子基金 |
| SZ_MINOR_KIND_FUND_CLOSED_END_FUNDS | 104 | 封闭式基金 |
| SZ_MINOR_KIND_FUND_REDEMPTION_FUND | 105 | 仅申赎基金 |
| SZ_MINOR_KIND_WARRANT | 106 | 权证 |
| SZ_MINOR_KIND_OPTION_STOCK_OPTION | 107 | 个股期权 |
| SZ_MINOR_KIND_OPTION ETF_OPTION | 108 | ETF期权 |
| SZ_MINOR_KIND_PREFERRED_STOCK | 109 | 优先股 |
| SZ_MINOR_KIND_BOND_SECURITY_COMPANY_SHORT_TERM_BOND | 110 | 证券公司短期债 |
| SZ_MINOR_KIND_BOND_EXCHANGEABLE_CORPORATE_BOND | 111 | 可交换公司债 |
| SZ_MINOR_KIND_STOCK_CDR | 112 | 主板、中小板存托凭证 |
| SZ_MINOR_KIND_STOCK_CHUANG_YE_CDR | 113 | 创业板存托凭证 |
| SZ_MINOR_KIND_FUND_INFRASTRUCTURE_FUND | 114 | 基础设施 |

| 枚举变量名 | 值 | 含义 |
|--|-----|-------------|
| | | 施基金 |
| SZ_MINOR_KIND_BOND_ORIENT_CONVERTIBLE_BOND | 115 | 定向可转债 |
| 深圳 | | 非交易类 |
| SZ_FJY_MINOR_KIND_ISSUE | 116 | 网上认购 |
| SZ_FJY_MINOR_KIND_BOND_DISTRIBUTION | 117 | 债券分销 |
| SZ_FJY_MINOR_KIND_RIGHTS_ISSUE | 118 | 配股 |
| SZ_FJY_MINOR_KIND_DERIVATIVEAUCTION | 119 | 衍生品交易 |
| SZ_FJY_MINOR_KIND_NEGOTIATION | 120 | 协议交易 |
| SZ_FJY_MINOR_KIND_ISSUE_ADDITIONNAL | 121 | 增发 |
| SZ_FJY_MINOR_KIND_BOND_RIGHTS_ISSUE | 122 | 配债 |
| 新三板 | | 可交易类 |
| NEEQ_MINOR_KIND_STOCK_DELIST_COMPANY_A | 123 | 两网公司及退市公司A股 |
| NEEQ_MINOR_KIND_STOCK_DELIST_COMPANY_B | 124 | 两网公司及退市公司B股 |
| NEEQ_MINOR_KIND_STOCK_LISTED_COMPANY | 125 | 挂牌公司股票 |
| NEEQ_MINOR_KIND_PREFERRED_STOCK | 126 | 优先股 |

| 枚举变量名 | 值 | 含义 |
|---|-----|----------|
| NEEQ_MINOR_KIND_TENDER_OFFER | 127 | 要约收购 |
| NEEQ_MINOR_KIND_OFFER_TO_REPURCHASE | 128 | 要约回购 |
| NEEQ_MINOR_KIND_EQUITY_INCENTIVE_OPTION | 129 | 股权激励期权 |
| NEEQ_MINOR_KIND_INDEX | 130 | 指数 |
| 新三板 | | 非交易类 |
| NEEQ_MINOR_KIND_ISSUE_TRANSACTION | 131 | 发行业务 |
| | | 新增可交易类 |
| NEEQ_MINOR_KIND_CONVERTIBLE_BOND | 132 | 可转债 |
| NEEQ_MINOR_KIND_ORIENT_CONVERTIBLE_BOND | 133 | 定向可转债 |
| NEEQ_MINOR_KIND_DELIST_CONVERTIBLE_BOND | 134 | 退市可转债 |
| SHFI_MINOR_KIND_TREASURY_BOND | 135 | 国债 |
| SHFI_MINOR_KIND_CORPORATE_BOND | 136 | 公司债 |
| SHFI_MINOR_KIND_PRIVATE_PLACEMENT_BOND | 137 | 私募债券 |
| SHFI_MINOR_KIND_ABS | 138 | 信贷资产支持证券 |
| SHFI_MINOR_KIND_SPECIAL_BOND | 139 | 特定债券 |
| SHFI_MINOR_KIND_REITS | 140 | 公募 |

| 枚举变量名 | 值 | 含义 |
|----------------------------------|-----|-------|
| | | REITs |
| SHFI_MINOR_KIND_CONVERTIBLE_BOND | 141 | 公募可转债 |

2.1.56. EReplaceFlag 现金替代标识

| 变量名 | 值 | 含义 |
|------------------------------------|-----|--------------------------|
| XT_REPLACE_FLAG_CASH_FORBID | '0' | 禁止现金替代（必须有股票） |
| XT_REPLACE_FLAG_CASH_ENABLE | '1' | 允许现金替代（先用股票，股票不足的话用现金替代） |
| XT_REPLACE_FLAG_CASH_REQUIRED | '2' | 必须现金替代 |
| XT_REPLACE_FLAG_NO_SH_CASH_FILL | '3' | 非沪市（股票）退补现金替代 |
| XT_REPLACE_FLAG_NO_SH_CASH_REQ | '4' | 非沪市（股票）必须现金替代 |
| XT_REPLACE_FLAG_NO_SH_SZ_CASH_FILL | '5' | 非沪深退补现金替代 |
| XT_REPLACE_FLAG_NO_SH_SZ_CASH_REQ | '6' | 非沪深必须现金替代 |
| XT_REPLACE_FLAG_RETURN_SH_SZ_HK | '7' | 港市退补现金替代（仅适用于跨沪深ETF产品） |
| XT_REPLACE_FLAG_MUST_SH_SZ_HK | '8' | 港市必须现金替代（仅适用于跨沪深港ETF产品） |

2.1.57. EXTCommandDateLimit 指令跨日标志

| 变量名 | 值 | 含义 |
|----------------------------------|---|---------|
| XT_COMMAND_LIMIT_NO_OVER_DAY | 0 | 不跨日 |
| XT_COMMAND_LIMIT_CROSS_OVER_DAY | 1 | 将跨日指令 |
| XT_COMMAND_LIMIT_ALREADY_CROSSED | 2 | 已经跨日的指令 |

2.1.58. PortfolioDealType 投资组合成交类型

| 变量名 | 值 | 含义 |
|--|----|-------|
| XT_PORTFOLIO_ADJUST_COMMON | 0 | 普通调整 |
| XT_PORTFOLIO_ADJUST_RESTRICTED_STOCK | 1 | 限售股 |
| XT_PORTFOLIO_ADJUST_SUB_FUND | 2 | 子基金 |
| XT_PORTFOLIO_ADJUST_INCOME | 3 | 收入 |
| XT_PORTFOLIO_ADJUST_COST | 4 | 费用 |
| XT_PORTFOLIO_ADJUST_BONDS_CONVERT | 5 | 可转债转股 |
| XT_PORTFOLIO_ADJUST_BONDS_SELLBACK | 6 | 可转债回售 |
| XT_PORTFOLIO_ADJUST_STOCK_ALLOTMENT | 7 | 股票配股 |
| XT_PORTFOLIO_ADJUST_STOCK_SUBSCRIPTION | 8 | 新股申购 |
| XT_PORTFOLIO_ADJUST_DIVIDEND_STOCK | 9 | 除权送股 |
| XT_PORTFOLIO_ADJUST_DIVIDEND_CASH | 10 | 除权分红 |

2.1.59. PortfolioPositionType 投资组合持仓类型

| 变量名 | 值 | 含义 |
|--|------|-------|
| XT_PORTFOLIO_POSITION_COMMON | 0 | 默认值 |
| XT_PORTFOLIO_POSITION_RESTRICTED_STOCK | 1000 | 限售持仓 |
| XT_PORTFOLIO_POSITION_SUB_FUND | 1001 | 子基金持仓 |
| XT_PORTFOLIO_POSITION_FIN_STOCK | 1002 | 融资持仓 |
| XT_PORTFOLIO_POSITION_SLO_STOCK | 1003 | 融券持仓 |

2.1.60. EPortfolioType 投组类型

| 变量名 | 值 | 含义 |
|----------------|---|------|
| PF_TYPE_NORMAL | 0 | 普通 |
| PF_TYPE_USER | 1 | 用户默认 |

| 变量名 | 值 | 含义 |
|--------------------------------|----|---------------------------|
| PF_TYPE_ACCOUNT | 2 | 账号默认 |
| PF_TYPE_DAILY_PRODUCT_DEFAULT | 3 | 按照产品每日自动创建的用于归集没有投组编号的数据 |
| PF_TYPE_EMULATION_ACCOUNT | 4 | 仿真账号 |
| PF_TYPE_PRODUCT_DEFAULT | 5 | 按照产品归集没有投组编号的数据 |
| PF_TYPE_NON_SELF_DEFAULT | 6 | 用于存储用户所在产品非用户产生的所有交易的默认投组 |
| PF_TYPE_COMPETITION_ACCOUNT | 7 | 炒股大赛仿真账号，不能调用修改持仓类的API |
| PF_TYPE_ETF_ARBITRAGE | 8 | ETF套利数据 |
| PF_TYPE_ACCOUNT_ETF_REDEMPTION | 9 | 用于存储账号所有ETF申赎数据 |
| PF_TYPE_ASSETUNIT_STAT | 10 | 用于资产单元比对数据 |

2.1.61. OrderPriceType 期货限价委托类型

| 变量名 | 值 | 含义 |
|------------|---|--------------|
| XT_OPF_GFD | 0 | 当日有效 |
| XT_OPF_FAK | 1 | 限价即时成交剩余撤销 |
| XT_OPF_FOK | 2 | 限价即时全部成交否则撤销 |

2.1.62. MetelD 退订枚举类型

| 变量名 | 值 | 含义 |
|---------------------|---|----------|
| XT_ACCOUNT_DETAIL | 0 | 资金信息推送 |
| XT_ORDER_DETAIL | 1 | 委托信息推送 |
| XT_DEAL_DETAIL | 2 | 成交信息推送 |
| XT_POSITION_DETAIL | 3 | 持仓明细信息推送 |
| XT_POSITION_STATICS | 4 | 持仓统计信息推送 |

| 变量名 | 值 | 含义 |
|---------------|---|--------|
| XT_ORDER_INFO | 5 | 指令信息推送 |

2.2. XtTraderApi数据结构说明

2.2.1. 股票、期货、股票期权的资产结构 CAccountDetail

| 成员变量 | 类型 | 描述 |
|---------------------|--------|------------------|
| m_strAccountID | string | 资金账号 |
| m_nAccountType | int | 账号类型 |
| m_strStatus | string | 状态 |
| m_strTradingDate | string | 交易日 |
| m_dFrozenMargin | float | 冻结保证金 股票不需要 |
| m_dFrozenCash | float | 内外源冻结保证金和手续费四个的和 |
| m_dFrozenCommission | float | 冻结手续费 |
| m_dRisk | float | 风险度 |
| m_dNav | float | 单位净值 |
| m_dPreBalance | float | 期初权益 |
| m_dBalance | float | 动态权益, 总资产 |
| m_dAvailable | float | 可用资金 |
| m_dCommission | float | 已用手续费 |
| m_dPositionProfit | float | 持仓盈亏 |
| m_dCloseProfit | float | 平仓盈亏 |
| m_dCashIn | float | 出入金净值 |
| m_dCurrMargin | float | 当前占用保证金 |
| m_dInstrumentValue | float | 合约价值 |
| m_dDeposit | float | 入金 |

| 成员变量 | 类型 | 描述 |
|----------------------|--------|--------------------------|
| m_dWithdraw | float | 出金 |
| m_dCredit | float | 信用额度 |
| m_dMortgage | float | 质押 |
| m_dStockValue | float | 股票总市值, 期货没有 |
| m_dLoanValue | float | 债券总市值, 期货没有 |
| m_dFundValue | float | 基金总市值, 包括ETF和封闭式基金, 期货没有 |
| m_dRepurchaseValue | float | 回购总市值, 所有回购, 期货没有 |
| m_dLongValue | float | 多单总市值, 现货没有 |
| m_dShortValue | float | 空单总市值, 现货没有 |
| m_dNetValue | float | m_dNetValue |
| m_dAssureAsset | float | 净资产 |
| m_dTotalDebit | float | 总负债 |
| m_dPremiumNetExpense | float | 权利金净支出 用于股票期权 |
| m_dEnableMargin | float | 可用保证金 用于股票期权 |
| m_dFetchBalance | float | 可取金额 |
| m_eDualStatus | float | 双中心状态 |
| m_dAvailableSH | float | 双中心上海可用 |
| m_dAvailableSZ | float | 双中心深圳可用 |
| m_strAccountKey | string | 账号key |
| m_nProductId | int | 账号 |
| m_dUsedMargin | float | 占用保证金 用于股票期权 |
| m_dRoyalty | float | 权利金支出 用于股票期权 |
| m_strProductName | string | 迅投产品名称 |
| m_dDaysProfit | float | 当日盈亏 |

| 成员变量 | 类型 | 描述 |
|-------------------|-------|---------|
| m_strAccountName | str | 账号名称 |
| m_strBrokerID | str | 经纪公司编号 |
| m_strBrokerName | str | 经纪公司名称 |
| m_dAssetBalance | float | 动态权益含期权 |
| m_dRiskDegree | float | 风险度含期权 |
| m_dInitCloseMoney | float | 初始平仓盈亏 |

2.2.2. 信用综合资产结构 CCreditAccountDetail 不再维护，改用 CCreditDetail

| 成员变量 | 类型 | 描述 |
|------------------------|-------|----------|
| m_dPerAssurescaleValue | float | 个人维持担保比例 |
| m_dEnableBailBalance | float | 可用保证金 |
| m_dUsedBailBalance | float | 已用保证金 |
| m_dAssureEnbuyBalance | float | 可买担保品资金 |
| m_dFinEnbuyBalance | float | 可买标的券资金 |
| m_dSloEnrepaidBalance | float | 可还券资金 |
| m_dFinEnrepaidBalance | float | 可还款资金 |
| m_dFinMaxQuota | float | 融资授信额度 |
| m_dFinEnableQuota | float | 融资可用额度 |
| m_dFinUsedQuota | float | 融资已用额度 |
| m_dFinUsedBail | float | 融资已用保证金额 |
| m_dFinCompactBalance | float | 融资合约金额 |
| m_dFinCompactFare | float | 融资合约费用 |
| m_dFinCompactInterest | float | 融资合约利息 |
| m_dFinMarketValue | float | 融资市值 |

| 成员变量 | 类型 | 描述 |
|---------------------------------|-------|-----------|
| m_dFinIncome | float | 融资合约盈亏 |
| m_dSloMaxQuota | float | 融券授信额度 |
| m_dSloEnableQuota | float | 融券可用额度 |
| m_dSloUsedQuota | float | 融券已用额度 |
| m_dSloUsedBail | float | 融券已用保证金额 |
| m_dSloCompactBalance | float | 融券合约金额 |
| m_dSloCompactFare | float | 融券合约费用 |
| m_dSloCompactInterest | float | 融券合约利息 |
| m_dSloMarketValue | float | 融券市值 |
| m_dSloIncome | float | 融券合约盈亏 |
| m_dOtherFare | float | 其它费用 |
| m_dUnderlyMarketValue | float | 标的证券市值 |
| m_dFinEnableBalance | float | 可融资金额 |
| m_dDiffEnableBailBalance | float | 可用保证金调整值 |
| m_dBuySecuRepayFrozenMargin | float | 买券还券冻结资金 |
| m_dBuySecuRepayFrozenCommission | float | 买券还券冻结手续费 |

2.2.3. 指令状态 COrderInfo

| 成员变量 | 类型 | 描述 |
|----------------|--------|------|
| m_strAccountID | string | 资金账号 |
| m_eBrokerType | int | 账号类型 |
| m_nOrderID | int | 指令ID |
| m_startTime | int | 下达时间 |
| m_endTime | int | 结束时间 |

| 成员变量 | 类型 | 描述 |
|----------------------|----------------|----------------------|
| m_eStatus | int | 指令状态 |
| m_dTradedVolume | float | 成交量 |
| m_dTradedPrice | float | 成交均价 |
| m_dTradedAmount | float | 成交金额 |
| m_strMsg | string | 指令执行信息 |
| m_canceler | string | 撤销者 |
| m_eBrokerType | int | 账号类型 |
| m_strRemark | string | 投资备注 |
| m_strMarket | string | 市场 仅三方算法指令 有值 |
| m_strInstrument | string | 合约 仅三方算法指令 有值 |
| m_strOrderType | string | 算法名称 仅三方算法 指令有值 |
| m_dPrice | float | 委托基准价 仅三方算 法指令有值 |
| m_nVolume | int | 下单总量 仅三方算法 指令有值 |
| m_nValidTimeStart | int | 有效开始时间 仅三方 算法指令有值 |
| m_nValidTimeEnd | int | 有效结束时间 仅三方 算法指令有值 |
| m_dMaxPartRate | float | 量比比例 仅三方算法 指令有值 |
| m_dMinAmountPerOrder | float | 委托最小金额 仅三方 算法指令有值 |
| m_eOperationType | EOperationType | 下单操作：买入卖出 |

| 成员变量 | 类型 | 描述 |
|-------------------------|-----------------------|---------------------|
| | | 仅三方算法指令有值 |
| m_nStopTradeForOwnHiLow | EStopTradeForOwnHiLow | 涨跌停控制 仅三方算法指令有值 |
| m_strAccountKey | string | 账号key |
| m_nLimitedPriceType | int | 条件单价格限制类型 |
| m_strOtherParam | str | 条件单其他参数 |
| m_nCmdSource | int | 指令来源，比如普通客户端类型、API等 |

2.2.4. 委托明细 COrderDetail

| 成员变量 | 类型 | 描述 |
|---------------------|--------|----------------|
| m_strAccountID | string | 资金账号 |
| m_nAccountType | int | 账号类型，参见数据字典 |
| m_strExchangeID | string | 交易所代码，参见数据字典 |
| m_strProductID | string | 合约品种 |
| m_strInstrumentID | string | 合约代码 |
| m_dLimitPrice | float | 委托价格 |
| m_strOrderSysID | string | 委托号 |
| m_nTradedVolume | int | 已成交量 |
| m_nTotalVolume | int | 当前总委托量 |
| m_dFrozenMargin | float | 冻结保证金 |
| m_dFrozenCommission | float | 冻结手续费 |
| m_dAveragePrice | float | 成交均价 |
| m_dTradeAmount | float | 成交额 期货=均价量合约乘数 |
| m_nErrorID | int | 交易所废单情况下的错误编号 |

| 成员变量 | 类型 | 描述 |
|----------------------|----------------|---|
| m_strErrorMsg | string | 废单原因 |
| m_strInsertDate | string | 日期 |
| m_strInsertTime | string | 时间 |
| m_nOrderID | int | 指令ID |
| m_nOrderPriceType | int | 委托明细价格类型，例如市价单类型，例如市价单 限价单 |
| m_nDirection | EEntrustBS | 期货多空,该字段与m_eOffsetFlag一起判断期货的报单类型，股票无用。 |
| m_eOffsetFlag | int | 期货开平—股票单，用该字段判断方向 |
| m_eHedgeFlag | EHedgeFlagType | 投机 套利 套保 |
| m_eOrderSubmitStatus | int | 期货提交状态，股票里面不需要报单状态 |
| m_eOrderStatus | int | 委托状态 |
| m_eEntrustType | int | 委托类别 |
| m_eCoveredFlag | int | 备兑标记 '0' - 非备兑, '1' - 备兑 |
| m_strRemark | string | 投资备注 |
| m_strCancelInfo | string | 废单信息 |
| m_strInstrumentName | string | 合约名称 |
| m_strAccountKey | string | 账号key |
| m_strStrategyID | string | 收益互换策略ID |
| m_strSecuAccount | string | 股东号 |
| m_strCombID | str | 组合持仓编号，用于解除组合策略 |

2.2.5. 成交明细 CDealDetail

| 成员变量 | 类型 | 描述 |
|-------------------|------------|--|
| m_strAccountID | string | 资金账号 |
| m_nAccountType | int | 账号类型, 参见数据字典 |
| m_strExchangeID | string | 交易所代码 |
| m_strProductID | string | 合约品种 |
| m_strInstrumentID | string | 合约代码 |
| m_strTradeID | string | 成交编号 |
| m_strOrderSysID | string | 委托号 |
| m_dAveragePrice | float | 成交均价 |
| m_nVolume | int | 成交量 (期货单位手 股票做到股) |
| m_strTradeDate | string | 成交日期 |
| m_strTradeTime | string | 成交时间 |
| m_dComssion | float | 手续费 |
| m_dAmount | float | 成交额 (期货=均价*量*合约乘数) |
| m_nOrderID | int | 指令ID |
| m_nDirection | EEntrustBS | 期货多空 该字段与m_eOffsetFlag一起判断期货的报单类型。股票无用 |
| m_nOffsetFlag | int | 期货开平 股票买卖 |
| m_nHedgeFlag | int | 投机 套利 套保 |
| m_nOrderPriceType | int | 委托明细价格类型, 例如市价单类型, 例如市价单 限价单 |
| m_eEntrustType | int | 委托类别 |
| m_eCoveredFlag | int | 备兑标记 '0' - 非备兑, '1' - 备兑 |
| m_strRemark | string | 投资备注 |

| 成员变量 | 类型 | 描述 |
|----------------------|-------------------|-----------------|
| m_strInstrumentName | string | 合约名称 |
| m_strAccountKey | string | 账号key |
| m_strStrategyID | string | 收益互换策略ID |
| m_strSecuAccount | string | 股东号 |
| m_nProductId | int | 迅投产品ID |
| m_strProductName | string | 迅投产品名称 |
| m_strCombID | str | 组合持仓编号，用于解除组合策略 |
| m_dOccupiedMargin | double | 占用保证金 |
| m_dReferenceRate | double | 汇率 |
| m_strEntryDate | string | 投资组合-录入日期 |
| m_strExpireDate | string | 投资组合-解禁日期 |
| m_nPortfolioDealType | PortfolioDealType | 投资组合成交类型 |

2.2.6. 持仓明细 CPositionDetail

| 成员变量 | 类型 | 描述 |
|---------------------|--------|------------|
| m_strAccountID | string | 资金账号 |
| m_strExchangeID | string | 市场代码 |
| m_strProductID | string | 合约品种 |
| m_strInstrumentID | string | 合约代码 |
| m_strInstrumentName | string | 合约名称 |
| m_strOpenDate | string | 开仓日期 |
| m_strTradeID | string | 最初开仓位的成交编号 |
| m_nVolume | int | 持仓量 当前持股 |
| m_dOpenPrice | float | 开仓价 |

| 成员变量 | 类型 | 描述 |
|------------------------|--------|---|
| m_strTradingDay | string | 交易日 |
| m_dMargin | float | 使用的保证金 (历史的直接用ctp的, 新的自己用成本价存量系数算股票不需要) |
| m_dOpenCost | float | 开仓成本 (等于股票的成本价*第一次建仓的量, 后续减持不影响, 不算手续费 股票不需要) |
| m_dSettlementPrice | string | 结算价 对于股票的当前价 |
| m_nCloseVolume | int | 平仓量 等于股票已经卖掉的 股票不需要 |
| m_dCloseAmount | float | 平仓额 等于股票每次卖出的量卖出价合约乘数 (股票为1) 的累加 股票不需要 |
| m_dFloatProfit | float | 浮动盈亏 浮动盈亏 当前量* (当前价-开仓价) *合约乘数 (股票为1) |
| m_dCloseProfit | float | 平仓盈亏 平仓额 - 开仓价平仓量合约乘数 (股票为1) 股票不需要 |
| m_dMarketValue | float | 市值 |
| m_dPositionCost | float | 持仓成本 股票不需要 |
| m_dPositionProfit | float | 持仓盈亏 股票不需要 |
| m_dLastSettlementPrice | float | 最新结算价 股票不需要 |
| m_dInstrumentValue | float | 合约价值 股票不需要 |
| m_bIsToday | int | 是否今仓 |
| m_nOrderID | int | 指令ID |
| m_nFrozenVolume | int | 冻结数量 期货不用这个字段 |
| m_nCanUseVolume | int | 可用数量 期货不用这个字段 |
| m_nOnRoadVolume | int | 在途数量 期货不用这个字段 |

| 成员变量 | 类型 | 描述 |
|--------------------|----------------|-----------------------------------|
| m_nYesterdayVolume | int | 昨日拥股 期货不用这个字段 |
| m_dLastPrice | float | 最新价 |
| m_nHedgeFlag | EHedgeFlagType | 投机 套利 套保 |
| m_nDirection | EEntrustBS | 期货多空，该字段与m_eOffsetFlag一起判断期货的报单类型 |
| m_nCoveredAmount | int | 备兑数量 |
| m_nAccountType | int | 账号类型 |
| m_dProfitRate | float | 持仓盈亏比例 |
| m_strAccountKey | string | 账号key |
| m_strStrategyID | string | 收益互换策略ID |
| m_strSecuAccount | string | 股东号 |

2.2.7. 持仓统计 CPositionStatics

| 成员变量 | 类型 | 描述 |
|---------------------|--------|------|
| m_strAccountID | string | 资金账号 |
| m_strExchangeID | string | 市场代码 |
| m_strProductID | string | 合约品种 |
| m_strInstrumentID | string | 合约代码 |
| m_strInstrumentName | string | 合约名称 |
| m_bIsToday | int | 是否今仓 |
| m_nPosition | int | 持仓量 |
| m_dOpenCost | float | 开仓成本 |
| m_dPositionCost | float | 持仓成本 |
| m_dAveragePrice | float | 开仓均价 |
| m_dPositionProfit | float | 持仓盈亏 |

| 成员变量 | 类型 | 描述 |
|---------------------|------------|---|
| m_dFloatProfit | float | 浮动盈亏 |
| m_dOpenPrice | float | 期货开仓均价 |
| m_nCanCloseVolume | int | 可平数量 |
| m_dUsedMargin | float | 已使用保证金 |
| m_dUsedCommission | float | 已使用的手续费 |
| m_dFrozenMargin | float | 冻结保证金 |
| m_dFrozenCommission | float | 冻结手续费 |
| m_dInstrumentValue | float | 合约价值 |
| m_nOpenTimes | int | 开仓次数 |
| m_nOpenVolume | int | 总开仓量 |
| m_nCancelTimes | int | 撤单次数 |
| m_nFrozenVolume | int | 冻结数量 |
| m_nCanUseVolume | int | 可用数量 |
| m_nOnRoadVolume | int | 在途数量 |
| m_nYesterdayVolume | int | 昨日拥股 |
| m_dSettlementPrice | float | 前收盘价 |
| m_dProfitRate | float | 持仓盈亏比例 |
| m_nDirection | EEntrustBS | 期货多空，该字段与 m_eOffsetFlag一起判断期 货的报单类型 |
| m_nHedgeFlag | int | 投机 套利 套保 |
| m_nCoveredAmount | int | 备兑数量 |
| m_nAccountType | int | 账号类型 |
| m_dLastPrice | float | 最新价 |

| 成员变量 | 类型 | 描述 |
|--------------------------|-----------------------|-------------|
| m_strStrategyID | string | 收益互换策略ID |
| m_strAccountKey | string | 账号key |
| m_strSecuAccount | string | 股东号 |
| m_eSideFlag | ESideFlag | 期权合约持仓类型 |
| m_dReferenceRate | double | 汇率 |
| m_strExpireDate | string | 到期日/投资组合解禁日 |
| m_nPortfolioPositionType | PortfolioPositionType | 投资组合持仓类型 |
| m_dSingleCost | double | 单股成本 |

2.2.8. 普通委托 COrdinaryOrder

| 成员变量 | 类型 | 描述 |
|--------------------|----------------|------------------|
| m_strAccountID | string | 资金账号 |
| m_dPrice | float | 委托价格 |
| m_dSuperPriceRate | float | 单笔超价百分比, 小数 |
| m_nVolume | int | 委托量, 直接还券的数量 |
| m_strMarket | string | 市场代码 |
| m_strInstrument | string | 合约代码 |
| m_ePriceType | EPriceType | 报价方式 |
| m_eOperationType | EOperationType | 下单类型 |
| m_eHedgeFlag | EHedgeFlagType | 投机、套利、套保 |
| m_dOccurBalance | float | 直接还款的金额 //仅直接还款用 |
| m_eTimeCondition | int | 期货条件单时间条件 |
| m_eVolumeCondition | int | 期货条件单数量条件 |

| 成员变量 | 类型 | 描述 |
|----------------------------|---------------------|---|
| m_strSecondInstrument | string | 期权组合委托合约 |
| m_dSecondPrice | float | 期权组合委托价 |
| m_eFirstSideFlag | int | 第一腿合约持仓类型 |
| m_eSecondSideFlag | int | 第二腿合约持仓类型 |
| m_strCombID | string | 组合持仓编号，用于解除组合策略 |
| m_strRemark | string | 投资备注 |
| m_eTriggerType | EOpTriggerType | 触价类型 |
| m_dTriggerPrice | float | 触价价格 |
| m_dSuperPrice | float | 单笔超价,和 m_dSuperPriceRate只 用设置一个，优先使用 m_dSuperPriceRate |
| m_nPortfolioID | int | 投组类型编号 |
| m_strPortfolioStrategyName | int | 投组策略名称，优先使用 m_nPortfolioID |
| m_eAbroadDurationType | EAbroadDurationType | 外盘期货报价类型 |
| m_strStrategyID | string | 收益互换策略ID |
| m_strSecuAccount | string | 股东号 |
| m_eAdjustOrderNum | EAdjustOrderNum | 下单根据可用调整可下 单量，仅api直连pb时 有效 |
| m_nLimitedPriceType | int | 条件单价格限制类型 |
| m_strOtherParam | str | 条件单其他参数 |
| m_eCmdDateLimit | EXTCommandDateLimit | 指令跨日开关 |
| m_eLimitOrderPriceType | OrderPriceType | 限价单价格限制类型 GFD FAK FOK |

2.2.9. 算法委托 CAlgorithmOrder

| 成员变量 | 类型 | 描述 |
|--------------------------|----------------|--------------|
| m_strAccountID | string | 资金账号 |
| m_strMarket | string | 市场代码 |
| m_strInstrument | string | 合约代码 |
| m_dPrice | float | 委托价格 |
| m_dSuperPriceRate | float | 单笔超价百分比, 小数 |
| m_nSuperPriceStart | int | 超价起始笔数 |
| m_nVolume | int | 委托量 |
| m_dSingleVolumeRate | float | 单比下单比率 |
| m_nSingleNumMin | int | 单比下单量最小值 |
| m_nSingleNumMax | int | 单比下单量最大值 |
| m_nLastVolumeMin | int | 尾单最小量 |
| m_dPlaceOrderInterval | float | 下单间隔 |
| m_dWithdrawOrderInterval | float | 撤单间隔 |
| m_nMaxOrderCount | int | 最大下单次数 |
| m_nValidTimeStart | int | 有效开始时间 |
| m_nValidTimeEnd | int | 有效结束时间 |
| m_eOperationType | EOperationType | 下单类型 |
| m_eSingleVolumeType | int | 单笔基准量 |
| m_ePriceType | EPriceType | 报价方式 |
| m_eHedgeFlag | EHedgeFlagType | 投机、套利、套保 |
| m_eOverFreqOrderMode | int | 委托频率过快时的处理方式 |
| m_eTimeCondition | int | 期货条件单时间条件 |

| 成员变量 | 类型 | 描述 |
|------------------------|---------------------|--|
| m_eVolumeCondition | int | 期货条件单数量条件 |
| m_strRemark | string | 投资备注 |
| m_eTriggerType | EOpTriggerType | 触价类型 |
| m_dTriggerPrice | float | 触价价格 |
| m_dSuperPrice | float | 单笔超价,和 m_dSuperPriceRate只 用设置一个, 优先使用 m_dSuperPriceRate |
| m_eAlgoPriceType | EAlgoPriceType | 订单类型 |
| m_nExtraLimitType | int | 扩展限价类型 |
| m_dExtraLimitValue | float | 扩展限价 |
| m_strStrategyID | string | 收益互换策略ID |
| m_eCmdDateLimit | EXTCommandDateLimit | 指令跨日开关 |
| m_nLimitedPriceType | int | 价格限制类型 |
| m_eLimitOrderPriceType | OrderPriceType | 限价单价格限制类型 GFD FAK FOK |

2.2.10. 智能算法委托 CIntelligentAlgorithmOrder

| 成员变量 | 类型 | 描述 |
|-----------------|--------|--------------------------------|
| m_strAccountID | string | 资金账号, 如果是 多账号下单, 该字 段无意义 |
| m_strMarket | string | 市场代码 |
| m_strInstrument | string | 合约代码 |
| m_strOrderType | string | 智能算法名称, 可 以参考数据字典 |
| m_dPrice | float | 委托价格 |

| 成员变量 | 类型 | 描述 |
|-----------------------------|-----------------------|---------------------------------|
| m_nVolume | int | 委托量 |
| m_nValidTimeStart | int | 有效开始时间 |
| m_nValidTimeEnd | int | 有效结束时间 |
| m_dMaxPartRate | float | 量比比例 |
| m_dMinAmountPerOrder | float | 委托最小金额 |
| m_eOperationType | EOperationType | 下单类型 |
| m_ePriceType | EPriceType | 报价方式 |
| m_strRemark | string | 投资备注 |
| m_dOrderRateInOpenAcution | float | 开盘集合竞价参与比例(取值0-1) 仅开盘+算法有用 |
| m_dPriceOffsetBpsForAuction | int | 开盘集合竞价价格偏移量(取值0-10000) 仅开盘+算法有用 |
| m_nStopTradeForOwnHiLow | EStopTradeForOwnHiLow | 涨跌停控制 |
| m_bOnlySellAmountUsed | bool | 仅用卖出金额 0,1 换仓特有 |
| m_dBuySellAmountDeltaPct | float | 买卖偏差上限 0.03-1 换仓特有 |
| m_nMaxTradeDurationAfterET | int | 收盘后是否继续执行, 0不继续, 非0继续 |
| m_eOrderStrategyType | EOrderStrategyType | 算法下单方式 |
| m_strStrategyID | string | 收益互换策略ID |
| m_nLimitedPriceType | int | 条件单价格限制类型 |

| 成员变量 | 类型 | 描述 |
|------------------------|---------------------|-----------------------------|
| m_strOtherParam | str | 条件单其他参数 |
| m_eCmdDateLimit | EXTCommandDateLimit | 指令跨日开关 |
| m_nTimeType | int | 时间类型, 0, 按区间, 1, 按执行时间, 默认0 |
| m_nTimeValue | int | 执行时间, 单位秒 |
| m_strRemark1 | str | 投资备注1 |
| m_nOpenTrade | int | 开盘集合竞价, 0, 不参与, 1, 参与, 默认0 |
| m_dCancelRateThreshold | double | 撤单率(取值0-1) |

2.2.11. 主动算法委托 CExternAlgorithmOrder

| 成员变量 | 类型 | 描述 |
|----------------------|--------|--|
| m_strAccountID | string | 资金账号 |
| m_strMarket | string | 市场代码 |
| m_strInstrument | string | 合约代码 |
| m_strOrderType | string | 主动算法名称, FTAIWAP, ALGOINTERFACE, ZEUS.... |
| m_dPrice | float | 委托价格 |
| m_nVolume | int | 委托量 |
| m_nValidTimeStart | int | 有效开始时间 |
| m_nValidTimeEnd | int | 有效结束时间 |
| m_dMaxPartRate | float | 量比比例 |
| m_dMinAmountPerOrder | float | 委托最小金额 |

| 成员变量 | 类型 | 描述 |
|----------------------------|-----------------------|------------------------|
| m_eOperationType | EOperationType | 下单类型 |
| m_strRemark | string | 投资备注 |
| m_nStopTradeForOwnHiLow | EStopTradeForOwnHiLow | 涨跌停控制 |
| m_eOrderStrategyType | EOrderStrategyType | 算法下单方式 |
| m_nMaxTradeDurationAfterET | float | 收盘后是否继续执行，0不继续，非0继续 |
| m_strStrategyID | int | 收益互换策略ID |
| m_nLimitedPriceType | int | 条件单价格限制类型 |
| m_strOtherParam | str | 条件单其他参数 |
| m_eCmdDateLimit | EXTCommandDateLimit | 指令跨日开关 |
| m_nTimeType | int | 时间类型，0，按区间，1，按执行时间，默认0 |
| m_nTimeValue | int | 执行时间，单位秒 |
| m_strRemark1 | str | 投资备注1 |

2.2.12. 错误消息 XtError

| 成员变量 | 类型 | 描述 |
|---------------|--------|------|
| m_nErrorID | int | 错误编号 |
| m_strErrorMsg | string | 错误原因 |

2.2.13. 委托错误信息 COrderError(委托请求被迅投风控或者柜台打回，会推送该消息)

| 成员变量 | 类型 | 描述 |
|----------------|--------|------|
| m_strAccountID | string | 资金账号 |

| 成员变量 | 类型 | 描述 |
|-----------------|--------|-------|
| m_nErrorID | int | 错误编号 |
| m_strErrorMsg | string | 错误原因 |
| m_nOrderID | string | 指令编号 |
| m_nRequestID | int | 请求编号 |
| m_strRemark | string | 投资备注 |
| m_strAccountKey | string | 账号key |

2.2.14. 撤销委托错误信息 CCancelError (撤销委托请求被迅投风控或者柜台打回，会推送该消息)

| 成员变量 | 类型 | 描述 |
|-----------------|--------|-------|
| m_strAccountID | string | 资金账号 |
| m_nErrorID | int | 错误编号 |
| m_strErrorMsg | string | 错误原因 |
| m_nOrderID | string | 指令编号 |
| m_nRequestID | int | 请求编号 |
| m_strRemark | string | 投资备注 |
| m_strAccountKey | string | 账号key |

2.2.15. 两融标的信息 CStkSubjects

| 成员变量 | 类型 | 描述 |
|-------------------|-------------------|-----------|
| m_strAccountID | string | 账号 |
| m_strExchangeID | string | 市场 |
| m_strInstrumentID | string | 合约 |
| m_dSloRatio | float | 融资融券保证金比例 |
| m_eSloStatus | EXTSubjectsStatus | 融券状态 |

| 成员变量 | 类型 | 描述 |
|-------------------|-------------------|---------|
| m_nEnableAmount | int | 融券可融数量 |
| m_dFinRatio | float | 融资保证金比例 |
| m_eFinStatus | EXTSubjectsStatus | 融资状态 |
| m_dAssureRatio | float | 担保品折算比例 |
| m_eAssureStatus | EXTSubjectsStatus | 是否可做担保 |
| m_dFinRate | float | 融资日利率 |
| m_dSloRate | float | 融券日利率 |
| m_dFinPenaltyRate | float | 融资日罚息利率 |
| m_dSloPenaltyRate | float | 融券日罚息利率 |

2.2.16. 两融标的状态枚举类型 EXTSubjectsStatus

| 变量名 | 值 | 含义 |
|------------------------|----|-----|
| SUBJECTS_STATUS_NORMAL | 48 | 正常 |
| SUBJECTS_STATUS_PAUSE | 49 | 暂停 |
| SUBJECTS_STATUS_NOT | 50 | 非标的 |

2.2.17. 个股期权备兑持仓信息 CCoveredStockPosition

| 成员变量 | 类型 | 描述 |
|---------------------|--------|------|
| m_strAccountID | string | 账号 |
| m_strExchangeID | string | 交易类别 |
| m_strExchangeName | string | 市场名字 |
| m_strInstrumentID | string | 合约代码 |
| m_strInstrumentName | string | 合约名称 |
| m_nTotalAmount | int | 总持仓量 |
| m_nLockAmount | int | 锁定量 |

| 成员变量 | 类型 | 描述 |
|------------------|-----|------|
| m_nUnlockAmount | int | 未锁定量 |
| m_nCoveredAmount | int | 备兑数量 |

2.2.18. 股票期权组合策略持仓信息 CStockOptionCombPositionDetail

| 成员变量 | 类型 | 描述 |
|----------------------|-------------|---------|
| m_strAccountID | string | 账号 |
| m_strExchangeID | string | 交易类别 |
| m_strExchangeName | string | 市场名字 |
| m_strContractAccount | string | 合约账号 |
| m_strCombID | string | 组合编号 |
| m_strCombCode | string | 组合策略编码 |
| m_strCombCodeName | string | 组合策略名称 |
| m_nVolume | int | 总持仓量 |
| m_nFrozenVolume | int | 冻结数量 |
| m_nCanUseVolume | int | 可用数量 |
| m_strFirstCode | string | 合约一 |
| m_eFirstCodeType | EOptionType | 合约一类型 |
| m_strFirstCodeName | string | 合约一名称 |
| m_eFirstCodePosType | ESideFlag | 合约一持仓类型 |
| m_nFirstCodeAmt | int | 合约一数量 |
| m_strSecondCode | string | 合约二 |
| m_eSecondCodeType | EOptionType | 合约二类型 |
| m_strSecondCodeName | string | 合约二名称 |
| m_eSecondCodePosType | ESideFlag | 合约二持仓类型 |

| 成员变量 | 类型 | 描述 |
|--------------------|-------|-------|
| m_nSecondCodeAmt | int | 合约二数量 |
| m_dCombBailBalance | float | 占用保证金 |

2.2.19. 订阅行情请求 CSubscribData

| 成员变量 | 类型 | 描述 |
|-------------------|----------------|---------------------------|
| m_nPlatformID | int | 平台ID |
| m_strExchangeID | string | 市场代码 |
| m_strInstrumentID | string | 合约代码,当其值为allCode时, 订阅整个市场 |
| m_eOfferStatus | EXTOfferStatus | 报盘状态 |

2.2.20. 汇率信息 CReferenceRate

| 成员变量 | 类型 | 描述 |
|----------------------|------------|--------------|
| m_strAccountID | string | 账号 |
| m_strExchangeID | string | 市场代码 |
| m_strExchangeName | string | 市场名字 |
| m_eMoneyType | EMoneyType | 币种 |
| m_bBidReferenceRate | float | 买入参考汇率 |
| m_bAskReferenceRate | float | 卖出参考汇率 |
| m_bBidSettlementRate | float | 买入结算汇率 |
| m_bAskSettlementRate | float | 卖出结算汇率 |
| m_dDayBuyRiseRate | float | 日间买入参考汇率浮动比例 |
| m_dNightBuyRiseRate | float | 夜市买入参考汇率浮动比例 |
| m_dDaySaleRiseRate | float | 日间卖出参考汇率浮动比例 |
| m_dNightSaleRiseRate | float | 日间卖出参考汇率浮动比例 |

| 成员变量 | 类型 | 描述 |
|---------------------|-------|---------|
| m_bMidReferenceRate | float | 参考汇率中间价 |

2.2.21. 行情数据 CPriceData

| 成员变量 | 类型 | 描述 |
|-----------------------|--------|-----------|
| m_strTradingDay | string | 交易日 |
| m_strExchangeID | string | 交易所代码 |
| m_strInstrumentID | string | 合约代码 |
| m_strInstrumentName | string | 合约名称 |
| m_strExchangeInstID | string | 合约在交易所的代码 |
| m_dLastPrice | float | 最新价 |
| m_dUpDown | float | 涨跌 |
| m_dUpDownRate | float | 涨跌幅 |
| m_dAveragePrice | float | 当日均价 |
| m_nVolume | int | 数量 |
| m_dTurnover | float | 成交金额 |
| m_dPreClosePrice | float | 昨收盘 |
| m_dPreSettlementPrice | float | 上次结算价 |
| m_dPreOpenInterest | float | 昨持仓量 |
| m_dOpenInterest | float | 持仓量 |
| m_dSettlementPrice | float | 本次结算价 |
| m_dOpenPrice | float | 今开盘 |
| m_dHighestPrice | float | 最高价 |
| m_dLowestPrice | float | 最低价 |
| m_dClosePrice | float | 今收盘 |

| 成员变量 | 类型 | 描述 |
|--------------------|--------|--------|
| m_dUpperLimitPrice | float | 涨停板价 |
| m_dLowerLimitPrice | float | 跌停板价 |
| m_dPreDelta | float | 昨虚实度 |
| m_dCurrDelta | float | 今虚实度 |
| m_strUpdateTime | string | 最后修改时间 |
| m_nUpdateMillisec | int | 最后修改毫秒 |
| m_dBidPrice1 | float | 申买价一 |
| m_nBidVolume1 | int | 申买量一 |
| m_dAskPrice1 | float | 申卖价一 |
| m_nAskVolume1 | int | 申卖量一 |
| m_dBidPrice2 | float | 申买价二 |
| m_nBidVolume2 | int | 申买量二 |
| m_dAskPrice2 | float | 申卖价二 |
| m_nAskVolume2 | int | 申卖量二 |
| m_dBidPrice3 | float | 申买价三 |
| m_nBidVolume3 | int | 申买量三 |
| m_dAskPrice3 | float | 申卖价三 |
| m_nAskVolume3 | int | 申卖量三 |
| m_dBidPrice4 | float | 申买价四 |
| m_nBidVolume4 | int | 申买量四 |
| m_dAskPrice4 | float | 申卖价四 |
| m_nAskVolume4 | int | 申卖量四 |
| m_dBidPrice5 | float | 申买价五 |
| m_nBidVolume5 | int | 申买量五 |

| 成员变量 | 类型 | 描述 |
|----------------|-------|--------|
| m_dAskPrice5 | float | 申卖价五 |
| m_nAskVolume5 | int | 申卖量五 |
| m_dPrePrice | float | 前一次的价格 |
| m_nStockStatus | int | 股票状态 |

2.2.22. 股票（合约）信息 CInstrumentDetail

| 成员变量 | 类型 | 描述 |
|---------------------|------------------|--------------------|
| m_strInstrumentID | string | 合约代码 |
| m_strInstrumentName | string | 合约名称 |
| m_strProductID | string | 品种代码 |
| m_strExchangeID | string | 交易所代码 |
| m_strOptUndlCode | string | 期权标的代码 |
| m_strOptUndlName | string | 期权标的证券名称 |
| m_strEndDelivDate | string | 最后日期 |
| m_dPriceTick | float | 价格波动 |
| m_nOptUnit | int | 期权合约单位 |
| m_dMarginUnit | float | 保证金单位 |
| m_dUpStopPrice | float | 涨停价 |
| m_dDownStopPrice | float | 跌停价 |
| m_dSettlementPrice | float | 前结算 |
| m_dOptExercisePrice | float | 期权行权价格 |
| m_nVolumeMultiple | int | 合约乘数 |
| m_nSuspendedType | EXtSuspendedType | 停牌状态 |
| m_nCallOrPut | int | 合约种类(个股期权：0认购，1认沽) |

2.2.23. 组合单下单参数 CGroupOrder

- 注意 list 容量限制1000

| 成员变量 | 类型 | 描述 |
|------------------|----------------------|-----------|
| m_orderParam | CAlgorithmOrder | 下单配置 |
| m_strMarket | List[string] | 市场列表 |
| m_strInstrument | List[string] | 证券代码 |
| m_nVolume | List[int] | 每只股票的下单量 |
| m_eOperationType | List[EOperationType] | 每只股票的下单类型 |
| m_nOrderNum | int | 股票只数 |
| m_strRemark | string | 投资备注 |

2.2.24. 算法组合单下单参数 CAlgGroupOrder

- 注意 list 容量限制1000
- 注意: 组合单兼容多资金账号组合下单功能, 如果是单账号组合单, m_strAccountKey可不填, 如果是多账号下单, m_strAccountKey必填, 否则会导致下单异常
- 使用多资金账户下单, 必须保证每个合约的买卖方向一致性。
- 每个账户对应的每个合约必须保证唯一性。
- 单账号多合约可以选择买入卖出双方向。
- 单账号多合约是组合单, 多账号多合约也是组合单, 多账号单合约只是批量单, 在对应业务界面展示。

| 成员变量 | 类型 | 描述 |
|------------------|----------------------------|----------------|
| m_orderParam | CIntelligentAlgorithmOrder | 下单配置 |
| m_strMarket | List[string] | 市场列表 |
| m_strInstrument | List[string] | 证券代码 |
| m_nVolume | List[int] | 每只股票的下单量 |
| m_eOperationType | List[EOperationType] | 每只股票的下单类型 |
| m_strAccountKey | List[str] | 每个合约的下单资金账号Key |

| 成员变量 | 类型 | 描述 |
|----------------|--------------|---|
| m_dMaxPartRate | List[double] | 量比比例, 用户设定, 当 MaxPartRate==100%, 表示没有限制, 如果量比比例全都相同可以只填写 CIntelligentAlgorithmOrder 里面的m_dMaxPartRate |
| m_nOrderNum | int | 股票只数 |
| m_strRemark | string | 投资备注 |

2.2.25. 外部算法组合单下单参数 CExternAlgGroupOrder

- 注意 list 容量限制1000

| 成员变量 | 类型 | 描述 |
|------------------|-----------------------|-----------|
| m_orderParam | CExternAlgorithmOrder | 下单配置 |
| m_strMarket | List[string] | 市场列表 |
| m_strInstrument | List[string] | 证券代码 |
| m_nVolume | List[int] | 每只股票的下单量 |
| m_eOperationType | List[EOperationType] | 每只股票的下单类型 |
| m_nOrderNum | int | 股票只数 |
| m_strRemark | string | 投资备注 |

2.2.26. 普通组合单下单参数 COrdinaryGroupOrder

- 注意: 组合单兼容多资金账号组合下单功能, 如果是单账号组合单, m_strAccountKey可不填, 如果是多账号下单, m_strAccountKey必填, 否则会导致下单异常
- 使用多资金账户下单, 必须保证每个合约的买卖方向一致性。
- 每个账户对应的每个合约必须保证唯一性。
- 单账号多合约可以选择买入卖出双方向。
- 单账号多合约是组合单, 多账号多合约也是组合单, 多账号单合约只是批量单, 在对应业务界面展示。

| 成员变量 | 类型 | 描述 |
|----------------------|----------------------|--|
| m_strAccountID | string | 资金账户ID, 如果为子账户, 则为子账户ID |
| m_dSuperPriceRate | float | 单笔超价百分比 |
| m_ePriceType | EPriceType | 报价方式: 指定价, 最新价 对手价..... |
| m_eHedgeFlag | EHedgeFlagType | 套利标志 |
| m_eOverFreqOrderMode | EXtOverFreqOrderMode | 委托频率过快时的处理方式 |
| m_eTimeCondition | ETimeCondition | 期货条件单时间条件 |
| m_eVolumeCondition | EVolumeCondition | 期货条件单数量条件 |
| m_strMarket | list[string] | 市场列表 |
| m_strInstrument | list[string] | 证券代码 |
| m_nVolume | list[int] | 每只股票的下单量 |
| m_dPrice | list[float] | 每只股票的下单价格 |
| m_eOperationType | list[EOperationType] | 每只股票的下单类型 |
| m_strAccountKey | list[str] | 每个合约的下单资金账号Key |
| m_nOrderNum | int | 股票只数 |
| m_strRemark | string | 投资备注 |
| m_dSuperPrice | float | 单笔超价,和 m_dSuperPriceRate只用 设置一个, 优先使用 m_dSuperPriceRate |
| m_strStrategyID | int | 收益互换策略ID |
| m_nLimitedPriceType | int | 条件单价格限制类型 |
| m_strOtherParam | str | 条件单其他参数 |

| 成员变量 | 类型 | 描述 |
|------------------------|----------------|--------------------------|
| m_eLimitOrderPriceType | OrderPriceType | 限价单价格限制类型 GFD FAK FOK |

2.2.27. 账号主键 CAccountKey

| 成员变量 | 类型 | 描述 |
|------------------|---------------|------------|
| m_nPlatformID | int | 经纪公司编号 |
| m_eBrokerType | EXTBrokerType | 经纪公司类别 |
| m_nAccountType | int | 账号类型编号 |
| m_strAccountID | string | 资金账号 |
| m_strSubAccount | string | 账号类型 |
| m_strAccountKey | string | 资金账号对应唯一主键 |
| m_strAccountName | string | 账号名称 |
| m_strBrokerName | string | 经纪公司名称 |

2.2.28. 普通柜台资金信息 CStockComFund

| 成员变量 | 类型 | 描述 |
|-----------------|--------|-------------------------|
| m_strAccountID | string | 资金账户ID, 如果为子账户, 则为子账户ID |
| m_strAccountKey | string | 账号key |
| m_dAvailable | float | 可用资金 |

2.2.29. 普通柜台持仓信息 CStockComPosition

| 成员变量 | 类型 | 描述 |
|-----------------|--------|-------------------------|
| m_strAccountID | string | 资金账户ID, 如果为子账户, 则为子账户ID |
| m_strAccountKey | string | 账号key |
| m_strExchangeID | string | 市场代码 |

| 成员变量 | 类型 | 描述 |
|---------------------|--------|-------------|
| m_strInstrumentID | string | 合约代码 |
| m_strInstrumentName | string | 合约名称 |
| m_nVolume | int | 持仓量 |
| m_nCanUseVolume | int | 可用数量 |
| m_dLastPrice | float | 最新价 |
| m_dCostPrice | float | 成本价 |
| m_dIncome | float | 盈亏 |
| m_dIncomeRate | float | 盈亏比例 |
| m_strHandFlag | string | 股手标记 |
| m_strStockAccount | string | 证券账号 |
| m_dInstrumentValue | float | 市值 |
| m_dCostBalance | float | 成本总额 |
| m_nOnRoadVolume | int | 在途量 |
| m_nPREnableVolume | int | 申赎可用量 |
| m_bFast | bool | 是否是从极速接口查到的 |

2.2.30. ETF申赎清单 CETFPCFDetail

| 成员变量 | 类型 | 描述 |
|----------------|--------|--------|
| m_nType | int | 基金类型 |
| m_strEtfExchID | string | ETF市场 |
| m_strEtfCode | string | ETF代码 |
| m_strPrCode | string | 基金申赎代码 |
| m_strName | string | 基金名称 |
| m_dCashBalance | double | 现金差额 |

| 成员变量 | 类型 | 描述 |
|---------------------|--------|----------------------|
| m_dMaxCashRatio | double | 现金替代比例上限 |
| m_nReportUnit | int | 最小申购,赎回单位 单位:份 |
| m_dNavPerCU | double | 最小申购,赎回单位净值 单位:元 |
| m_dNav | double | 基金份额净值 单位:元 |
| m_dEcc | double | 预估现金差额 单位:元 |
| m_nNeedPublish | int | 是否需要公布IOPV 1,是,0,否 |
| m_nEnableCreation | int | 是否允许申购 1,是,0,否 |
| m_nEnableRedemption | int | 是否允许赎回 1,是,0,否 |
| m_nCreationLimit | int | 申购上限 1,是,0,否 |
| m_nRedemptionLimit | int | 赎回上限 1,是,0,否 |
| m_strTradingDay | string | ETF交易日期 格式 YYYYMMDD |
| m_strPreTradingDay | string | ETF前交易日期 格式 YYYYMMDD |

2.2.31. ETF成分股信息 CETFComponentStockInfo

| 成员变量 | 类型 | 描述 |
|----------------------|--------------|---------|
| m_strExchangeID | string | ETF市场 |
| m_strEtfCode | string | ETF代码 |
| m_strEtfName | string | ETF基金名称 |
| m_strComponentCode | string | 成份股代码 |
| m_strComponentName | string | 成份股名称 |
| m_strComponentExchID | string | 成份股市场 |
| m_nComponentVolume | int | 成份股数量 |
| m_eReplaceFlag | EReplaceFlag | 替代标志 |
| m_dReplaceRatio | double | 溢价比率 |

| 成员变量 | 类型 | 描述 |
|-------------------------|--------|--------|
| m_dReplaceBalance | double | 替代金额 |
| m_dDiscountRatio | double | 赎回折价比率 |
| m_dRedeemReplaceBalance | double | 赎回替代金额 |

2.2.32. 期货持仓统计 CFuturePositionStatics

| 成员变量 | 类型 | 描述 |
|---------------------|--------|------------------------------------|
| m_strAccountID | string | 资金账号 |
| m_strExchangeID | string | 交易所 |
| m_strExchangeName | string | 市场名称 |
| m_strProductID | string | 品种代码 |
| m_strInstrumentID | string | 合约代码 |
| m_strInstrumentName | string | 合约名称 |
| m_nDirection | Enum | 期货多空，该字段与 m_eOffsetFlag一起判断期货的报单类型 |
| m_nHedgeFlag | Enum | 投机 套利 套保 |
| m_nPosition | int | 总持仓 |
| m_nYestodayPosition | int | 昨仓 |
| m_nTodayPosition | int | 今仓 |
| m_nCanCloseVol | int | 可平量 |
| m_dPositionCost | float | 持仓成本 detail的汇总 |
| m_dAvgPrice | float | 持仓均价 |
| m_dPositionProfit | float | 持仓盈亏 detail的汇总 |
| m_dFloatProfit | float | 浮动盈亏 detail的汇总 |
| m_dOpenPrice | float | 开仓均价 |

| 成员变量 | 类型 | 描述 |
|----------------------------------|--------|------------|
| m_dUsedMargin | float | 已使用保证金 |
| m_dUsedCommission | float | 已使用的手续费 |
| m_dFrozenMargin | float | 冻结保证金 |
| m_dFrozenCommission | float | 冻结手续费 |
| m_dInstrumentValue | float | 合约价值 |
| m_nOpenTimes | int | 开仓次数 |
| m_nOpenVolume | int | 总开仓量 |
| m_nCancelTimes | int | 撤单次数 |
| m_dLastPrice | float | 最新价 |
| m_dRiseRatio | float | 当日涨幅 |
| m_strProductName | string | 产品名称 |
| m_dRoyalty | float | 权利金市值 |
| m_strExpireDate | string | 到期日 |
| m_dAssestWeight | float | 资产占比 |
| m_dIncreaseBySettlement | float | 当日涨幅（结） |
| m_dMarginRatio | float | 保证金占比 |
| m_dFloatProfitDivideByUsedMargin | float | 浮盈比例（保证金） |
| m_dFloatProfitDivideByBalance | float | 浮盈比例（动态权益） |
| m_dTodayProfitLoss | float | 当日盈亏（结） |
| m_nYestodayInitPosition | int | 昨日持仓 |
| m_dFrozenRoyalty | float | 冻结权利金 |
| m_dTodayCloseProfitLoss | float | 当日盈亏（收） |
| m_dCloseProfit | float | 平仓盈亏 |
| m_strFtProductName | string | 品种名称 |

| 成员变量 | 类型 | 描述 |
|-----------------------------|-------|----------|
| m_dOpenCost | float | 开仓成本 |
| m_dEstimateSettleProfitLoss | float | 保证金（预结） |
| m_dEstimateSettleMargin | float | 当日盈亏（预结） |
| m_dSettlementPrice | float | 结算价 |

2.2.33. 用户配置信息 CUserConfig

| 成员变量 | 类型 | 描述 |
|------------------|--------|--------|
| m_nId | int | |
| m_nMetaId | int | 数据ID |
| m_nPermission | int | 私人还是全局 |
| m_strUser | string | 用户 |
| m_strUpdateTime | string | 更新时间 |
| m_strContent | string | 内容 |
| m_strDescription | string | 描述 |
| m_bsonContent | string | 内容串 |

2.2.34. 信用账号信息 CCreditDetail

| 成员变量 | 类型 | 描述 |
|------------------------|--------|--------|
| m_strAccountID | string | 账号 |
| m_dPerAssurescaleValue | double | 维持担保比例 |
| m_dBalance | double | 动态权益 |
| m_dTotalDebt | double | 总负债 |
| m_dAssureAsset | double | 净资产 |
| m_dMarketValue | double | 合约价值 |
| m_dEnableBailBalance | double | 可用保证金 |

| 成员变量 | 类型 | 描述 |
|--------------------------|--------|------------------------|
| m_dAvailable | double | 可用资金 |
| m_dFinDebt | double | 融资负债 |
| m_dFinDealAvl | double | 融资本金 |
| m_dFinFee | double | 融资息费 |
| m_dSloDebt | double | 融券负债 |
| m_dSloMarketValue | double | 融券市值 |
| m_dSloFee | double | 融券息费 |
| m_dOtherFare | double | 其它费用 |
| m_dFinMaxQuota | double | 融资授信额度 |
| m_dFinEnableQuota | double | 融资可用额度 |
| m_dFinUsedQuota | double | 融资冻结额度 |
| m_dSloMaxQuota | double | 融券授信额度 |
| m_dSloEnableQuota | double | 融券可用额度 |
| m_dSloUsedQuota | double | 融券冻结额度 |
| m_dSloSellBalance | double | 融券卖出资金 |
| m_dUsedSloSellBalance | double | 已用融券卖出资金 |
| m_dSurplusSloSellBalance | double | 剩余融券卖出资金 |
| m_dStockValue | double | 股票总市值，期货没有 |
| m_dFundValue | double | 基金总市值，包括ETF和封闭式基金，期货没有 |
| m_dMaxQuota | double | 总授信额度 |
| m_dAssureEnbuyBalance | double | 可买担保品资金 |
| m_dFetchBalance | double | 可取资金 |
| m_dEquityCompensation | double | 权益补偿资金 |

| 成员变量 | 类型 | 描述 |
|--------------------|--------|--------------|
| m_dSpecialNetRatio | double | D与E类证券净持仓集中度 |
| m_dStibGemNetRatio | double | 双创业板净持仓集中度 |
| m_dEnableQuota | double | 可用授信额度 |

2.2.35. 合约信息 CInstrumentInfo

| 成员变量 | 类型 | 描述 |
|---------------------|---------------|-----------|
| m_strInstrumentID | string | 合约代码 |
| m_strInstrumentName | string | 合约名称 |
| m_strExchangeID | string | 交易所代码 |
| m_strExchangeInstID | string | 合约在交易所的代码 |
| m_strProductID | string | 品种代码 |
| m_strCreateDate | string | 创建日 |
| m_strOpenDate | string | 上市日 |
| m_strExpireDate | string | 到期日 |
| m_strStartDelivDate | string | 开始交割日 |
| m_strEndDelivDate | string | 结束交割日 |
| m_strOptUndlCode | string | 期权标的代码 |
| m_strOptUndlMarket | string | 期权标的市场 |
| m_eProductClass | EProductClass | 合约类型, 仅期 |

| 成员变量 | 类型 | 描述 |
|---------------------------|-------------------------------|-------------------|
| | | 货和期货 期权合约 用 |
| m_nDeliveryYear | int | 交割年份 |
| m_nDeliveryMonth | int | 交割月 |
| m_nMaxMarketOrderVolume | int | 市价单最 大下单量 |
| m_nMinMarketOrderVolume | int | 市价单最 小下单量 |
| m_nMaxLimitOrderVolume | int | 限价单最 大下单量 |
| m_nMinLimitOrderVolume | int | 限价单最 小下单量 |
| m_nVolumeMultiple | int | 合约数量 乘数 |
| m_nIsTrading | int | 当前是否 交易 |
| m_dPriceTick | double | 最小变动 价位 |
| m_dLongMarginRatio | double | 多头保证 金率 |
| m_dShortMarginRatio | double | 空头保证 金率 |
| m_dUpStopPrice | double | 涨停价 |
| m_dDownStopPrice | double | 跌停价 |
| m_dSettlementPrice | double | 前结算 |
| m_eMaxMarginSideAlgorithm | EXtMaxMarginSideAlgorithmType | 是否使用 大额单边 |

| 成员变量 | 类型 | 描述 |
|---------------------|------------------|--------------------------|
| | | 保证金算法 |
| m_eSuspendedType | EXtSuspendedType | 停牌状态 |
| m_eExDivdendType | EXtExDivdendType | 除权除息标志 |
| m_dOptExercisePrice | double | 期权行权价 |
| m_nCallOrPut | int | 合约种类 (个股期权: 0认购, 1认沽) |
| m_dMarginUnit | double | 保证金单位 |
| m_eStockType | EStockType | 证券类别 |
| m_strOptUndlName | string | 期权标的 证券名称 |
| m_nOptUnit | int | 期权合约 单位 |

2.2.36. 产品净值信息 CNetValue

| 成员变量 | 类型 | 描述 |
|------------------|--------|----------------------|
| m_nProductId | int | 迅投产品ID |
| m_nTypes | int | 产品类型: 1-普通基金, 2-分级基金 |
| m_dTotalNetValue | double | 产品净资产 (产品净值) |
| m_dNetValue | double | 母基金单位净值 |
| m_dBNetValue | double | B级基金单位净值 |
| m_nUpdateTime | int | 更新时间 |

| 成员变量 | 类型 | 描述 |
|----------------------------|--------|--------|
| m_dShare | double | 产品份额 |
| m_dTotalIncome | double | 当日产品盈亏 |
| m_dAccumulateIncome | double | 产品累计盈亏 |
| m_dCloseTotalIncome | double | 产品收盘盈亏 |
| m_dPrevTotalNetValue | double | 前日产品净值 |
| m_dPrevNetValue | double | 前日单位净值 |
| m_dAccumulateNetAssetValue | double | 累计单位净值 |
| m_dAccumulateNetValue | double | 产品累计净值 |

2.2.37. 产品信息 CProductData

| 成员变量 | 类型 | 描述 |
|--------------------|----------|------------------------|
| m_nProductId | int | 迅投产品ID |
| m_strProductName | char[64] | 迅投产品名称 |
| m_strProductCode | char[64] | 迅投产品代码 |
| m_strCreateDate | char[64] | 迅投产品创建日期 |
| m_dTotalNetValue | double | 产品净资产, 产品净值 |
| m_dBalance | double | 当前资金余额（期货的动态权益和证券的可用） |
| m_dPreBalance | double | 期初资金余额（期货静态权益和证券的资金余额） |
| m_dAvaliableFuture | double | 期货帐号的可用资金之和 |
| m_dCurrMargin | double | 期货账号占用保证金 |
| m_dBalancefuture | double | 期货动态权益之和 |
| m_dStockValue | double | 股票总市值 |
| m_dLoanValue | double | 债券总市值，期货没有 |

| 成员变量 | 类型 | 描述 |
|----------------------|--------|------------------------|
| m_dFundValue | double | 基金总市值，包括ETF和封闭式基金，期货没有 |
| m_dRepurchaseValue | double | 回购总市值，所有回购，期货没有 |
| m_dTotalDebt | double | 总负债 |
| m_dGGTStockValue | double | 港股市值（人民币） |
| m_dShare | double | 产品份额 |
| m_dTotalIncome | double | 当日产品盈亏 |
| m_dAccumulateIncome | double | 产品累计盈亏 |
| m_dCloseTotalIncome | double | 产品收盘盈亏 |
| m_dPrevTotalNetValue | double | 前日产品净值 |
| m_dPrevNetValue | double | 前日单位净值 |

3. XtTraderApi

3.0 XtTraderApi API说明

XtTraderApi是迅投GT交易api应用程序标准接口，API的接口包含同步和异步两种模式，采用异步方式需要继承XtTraderApiCallback类，并实现回调函数，接受异步响应。api还提供高性能的主推函数，实现快速响应委托，成交等数据的快速推送响应。

3.1. 会话接口

3.1.1. 获取XtTraderApi实例

```
def createXtTraderApi(address: str) -> XtTraderApi:
```

- 释义：获取XtTraderApi实例
- 参数

| 参数 | 参数释义 | 参数类型 |
|---------|------------------|--------|
| address | XtApiService监听端口 | string |

3.1.2. 初始化API实例并建立连接

```
def init(self, configFilePath: str = '../config') -> bool:
```

- 释义：初始化API实例并建立连接
- 参数

| 参数 | 参数释义 | 参数类型 |
|----------------|--|--------|
| configFilePath | 配置文件夹目录，默认是"../config"，运行目录上一层的config下 | string |

3.1.3. 启动XtTraderApi单实例线程

```
def join(self) -> None:
```

- 释义：启动XtTraderApi单实例线程, 多实例情况必须调用joinAll()函数或者joinAll_async()函数。
- 注意：需要调用该函数阻塞线程，否则没有请求XtTraderApi实例会退出。使用同步函数必须调用joinAll()或者joinAll_async()。

3.1.4. 异步启动XtTraderApi单实例线程

```
def joinAll_async(self) -> None:
```

- 释义：异步启动启动XtTraderApi单实例线程, 多实例情况必须调用joinAll()函数或者joinAll_async()函数。
- 注意：需要调用该函数阻塞线程，否则没有请求XtTraderApi实例会退出。使用同步函数必须调用joinAll()或者joinAll_async()。

3.1.5. 启动XtTraderApi多实例线程

```
def joinAll() -> None:
```

- 释义：启动XtTraderApi多实例线程, 多实例情况下必须使用该函数, 同时单实例

也可调用该函数。

- 注意：需要调用该函数阻塞线程，否则没有请求XtTraderApi实例会退出。使用同步函数必须调用joinAll()或者joinAll_async()。

3.1.6. 异步启动XtTraderApi单实例线程

```
def joinAll_async(self) -> None:
```

- 释义：异步启动启动XtTraderApi多实例线程, 多实例情况必须调用joinAll()函数或者joinAll_async()函数。
- 注意：需要调用该函数阻塞线程，否则没有请求XtTraderApi实例会退出。使用同步函数必须调用joinAll()或者joinAll_async()。

3.1.7. 析构XtTraderApi实例

```
def destroy(self) -> None:
```

- 释义：析构XtTraderApi实例

3.1.8. 销毁XtTraderApi多实例线程

```
def destroyAll() -> None:
```

- 释义：销毁XtTraderApi多实例线程, 所有实例停止工作, joinAll()函数解除等待状态

3.1.9. 设置数据回调对象

```
def setCallback(self, pCallback: ...) -> None:
```

- 释义：设置数据回调对象
- 参数

| 参数 | 参数释义 | 参数类型 |
|-----------|------------------------|---------------------|
| pCallback | XtTraderApiCallback类实例 | XtTraderApiCallback |

3.2. 交易接口

3.2.1. 普通单下单 同步接口

```
def orderSync(self, orderInfo: COrdinaryOrder, error: XtError,  
accountKey: str = '') -> int:
```

- 释义：普通单下单，支持股票，期货，个股期权，期货期权，沪港通，深港通，返回指令号orderId，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数名 | 参数释义 | 参数类型 |
|------------|--------------------------|----------------|
| orderInfo | 普通单请求参数 | COrdinaryOrder |
| error | 错误信息 | XtError |
| accountKey | 账号Key，用于区分不同类型的账号ID相同的账号 | string |

3.2.2. 普通单下单

```
def order(self, orderInfo: COrdinaryOrder, nRequestId: int,  
accountKey: str = '') -> None:
```

- 释义：普通单下单，支持股票，期货，个股期权，期货期权，沪港通，深港通，回调 onOrder
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-----------------|----------------|
| orderInfo | 普通单的下单结构，参见数据字典 | COrdinaryOrder |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 唯一标识一个账号的key | string |

3.2.3. 直接下单

```
def directOrder(self, orderInfo: COrdinaryOrder, nRequestId: int,  
accountKey: str = '') -> None:
```


- 释义：直接下单，支持股票，期货，个股期权，期货期权，沪港通，深港通，只支持指定价普通下单
- note 期货六键普通委托，OPT_OPEN_LONG - OPT_CLOSE_SHORT_TODAY
- note 股票港股普通委托，OPT_BUY - OPT_SELL
- note 两融和专项委托，OPT_FIN_BUY - OPT_DIRECT_CASH_REPAY和OPT_SLO_SELL_SPECIAL - OPT_DIRECT_CASH_REPAY_SPECIAL
- note 期权普通委托，OPT_OPTION_BUY_OPEN - OPT_OPTION_COVERED_CLOSE和OPT_OPTION_SECU_LOCK - OPT_OPTION_SECU_UNLOCK
- note 普通单，回调 onDirectOrder
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-----------------|----------------|
| orderInfo | 普通单的下单结构，参见数据字典 | COrdinaryOrder |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 唯一标识一个账号的key | string |

3.2.4. 组合算法单下单

```
def order(self, orderInfo: CGroupOrder, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：组合算法单下单，只支持股票，回调 onOrder
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|-------------|
| orderInfo | 组合算法单请求参数 | CGroupOrder |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 唯一标识一个账号的key | string |

3.2.5. 组合智能算法单下单 同步接口

```
def orderSync(self, orderInfo: CAlgGroupOrder, error: XtError,
accountKey: str = '') -> int:
```

- 释义：组合智能算法单下单同步接口，支持股票，期货，个股期权，期货期权，沪港通，深港通，返回指令号orderId，处理返回数据之前需要先判断error.isSuccess()
- 注意：组合单兼容多资金账号组合下单功能，如果是单账号组合单，m_strAccountKey可不填，如果是多账号下单，m_strAccountKey必填，否则会导致下单异常（m_strAccountKey指CAIlgGroupOrder里面的入参，非函数入参accountKey，m_strAccountKey填写之后，函数入参accountKey填写无意义）。
- 参数

| 参数名 | 参数释义 | 参数类型 |
|------------|--------------------------|-----------------|
| orderInfo | 组合智能算法单请求参数 | CAIlgGroupOrder |
| error | 错误信息 | XtError |
| accountKey | 账号Key，用于区分不同类型的账号ID相同的账号 | string |

3.2.6. 组合智能算法单下单

```
def order(self, orderInfo: CAIlgGroupOrder, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：组合智能算法单下单，支持股票，期货，个股期权，期货期权，沪港通，深港通，回调 onOrder
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|-----------------|
| orderInfo | 组合智能算法单请求参数 | CAIlgGroupOrder |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 唯一标识一个账号的key | string |

3.2.7. 组合外部算法单下单 同步接口

```
def orderSync(self, orderInfo: CExternAlIlgGroupOrder, error: XtError,
accountKey: str = '') -> int:
```

- 释义：组合外部算法单下单同步接口，支持股票，期货，个股期权，期货期权，

沪港通，深港通，返回指令号orderId，处理返回数据之前需要先判断error.isSuccess()

- 参数

| 参数名 | 参数释义 | 参数类型 |
|------------|--------------------------|----------------------|
| orderInfo | 组合外部算法单请求参数 | CExternAlgGroupOrder |
| error | 错误信息 | XtError |
| accountKey | 账号Key，用于区分不同类型的账号ID相同的账号 | string |

3.2.8. 组合外部算法单下单

```
def order(self, orderInfo: CExternAlgGroupOrder, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：组合外部算法单下单，支持股票，期货，个股期权，期货期权，沪港通，深港通，回调 onOrder
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|----------------------|
| orderInfo | 组合外部算法单请求参数 | CExternAlgGroupOrder |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 唯一标识一个账号的key | string |

3.2.9. 算法单下单 同步接口

```
def orderSync(self, orderInfo: CAlgorithmOrder, error: XtError,
accountKey: str = '') -> int:
```

- 释义：算法单下单，支持股票，期货，个股期权，期货期权，沪港通，深港通，返回指令号orderId，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数名 | 参数释义 | 参数类型 |
|------------|--------------------------|-----------------|
| orderInfo | 算法单请求参数 | CAlgorithmOrder |
| error | 错误信息 | XtError |
| accountKey | 账号Key，用于区分不同类型的账号ID相同的账号 | string |

3.2.10. 算法单下单

```
def order(self, orderInfo: CAlgorithmOrder, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：算法单下单，支持股票，期货，个股期权，期货期权，沪港通，深港通，回调 onOrder
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|-----------------|
| orderInfo | 算法单请求参数 | CAlgorithmOrder |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 唯一标识一个账号的key | string |

3.2.11. 随机量交易下单

```
def order(self, orderInfo: CRandomOrder, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：随机量交易下单，支持股票，期货，个股期权，期货期权，沪港通，深港通，回调 onOrder
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------------|
| orderInfo | 随机量交易请求参数 | CRandomOrder |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 唯一标识一个账号的key | string |

3.2.12. 智能算法下单 同步接口

```
def orderSync(self, orderInfo: CIntelligentAlgorithmOrder,  
error: XtError, accountKey: str = '') -> int:
```

- 释义：智能算法下单同步接口，支持股票，期货，个股期权，期货期权，沪港通，深港通，返回指令号orderId，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数名 | 参数释义 | 参数类型 |
|------------|--------------------------|----------------------------|
| orderInfo | 智能算法交易请求参数 | CIntelligentAlgorithmOrder |
| error | 错误信息 | XtError |
| accountKey | 账号Key，用于区分不同类型的账号ID相同的账号 | string |

3.2.13. 智能算法下单

```
def order(self, orderInfo: CIntelligentAlgorithmOrder,  
nRequestId: int, accountKey: str = '') -> None:
```

- 释义：智能算法下单，支持股票，期货，个股期权，期货期权，沪港通，深港通，回调 onOrder
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|----------------------------|
| orderInfo | 智能算法交易请求参数 | CIntelligentAlgorithmOrder |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 唯一标识一个账号的key | string |

3.2.14. 主动算法下单 同步接口

```
def orderSync(self, orderInfo: CExternAlgorithmOrder,  
error: XtError, accountKey: str = '') -> int:
```

- 释义：主动算法下单同步接口，支持股票，期货，个股期权，期货期权，沪港通，深港通，返回指令号orderId，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数名 | 参数释义 | 参数类型 |
|------------|--------------------------|-----------------------|
| orderInfo | 主动算法交易请求参数 | CExternAlgorithmOrder |
| error | 错误信息 | XtError |
| accountKey | 账号Key，用于区分不同类型的账号ID相同的账号 | string |

3.2.15. 主动算法下单

```
def order(self, orderInfo: CExternAlgorithmOrder,
nRequestId: int, accountKey: str = '') -> None:
```

- 释义：主动算法下单，支持股票，期货，个股期权，期货期权，沪港通，深港通，回调 onOrder
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|-----------------------|
| orderInfo | 主动算法交易请求参数 | CExternAlgorithmOrder |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 唯一标识一个账号的key | string |

3.2.16. 普通组合下单

```
def order(self, orderInfo: COrdinaryGroupOrder,
nRequestId: int, accountKey: str = '') -> None:
```

- 释义：普通组合下单，支持股票，期货，个股期权，期货期权，沪港通，深港通，回调 onOrder
- 注意：组合单兼容多资金账号组合下单功能，如果是单账号组合单，m_strAccountKey可不填，如果是多账号下单，m_strAccountKey必填，否则会导致下单异常（m_strAccountKey指COrdinaryGroupOrder里面的入参，非

函数入参accountKey, m_strAccountKey填写之后, 函数入参accountKey填写无意义)。

- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-----------------|---------------------|
| orderInfo | 普通组合交易请求参数 | COrdinaryGroupOrder |
| nRequestId | 请求ID, 每个请求的ID自增 | int |
| accountKey | 唯一标识一个账号的key | string |

3.2.17. 撤单 同步接口

```
def cancelSync(self, orderID: int, error: XtError,
accountKey: str = '') -> None:
```

- 释义: 同步撤单, 按指令号撤单, 撤销指令, 终止某个单子的运行
- 参数

| 参数名 | 参数释义 | 参数类型 |
|------------|---------------------------|---------|
| orderID | 指令号 | int |
| error | 错误信息 | XtError |
| accountKey | 账号Key, 用于区分不同类型的账号ID相同的账号 | string |

3.2.18. 撤单

```
def cancel(self, orderID: int, nRequestId: int) -> None:
```

- 释义: 根据指令编号orderID对指令进行撤单操作, orderID对应onOrder里的orderID以及onRtnOrder里的data->m_nOrderID
- 参数

| 参数名称 | 参数释义 | 参数类型 |
|------------|-----------------|------|
| orderID | 指令编号 | int |
| nRequestId | 请求ID, 每个请求的ID自增 | int |

3.2.19. 按委托号撤单 同步接口

```
def cancelOrderSync(self, accountID: str, orderSyedId: str,
                    exchangeId: str, instrumentId: str, error: XtError,
                    accountKey: str = '') -> None:
```

- 释义：根据券商柜台返回的合同编号对委托进行撤单操作，error, isSuccess() 判断是否撤单成功
- 参数

| 参数名 | 参数释义 | 参数类型 |
|--------------|-------|---------|
| accountID | 证券账号 | string |
| orderSyedId | 合同编号 | string |
| exchangeId | 市场代码 | string |
| instrumentId | 合约代码 | string |
| error | 错误信息 | XtError |
| accountKey | 账号key | string |

3.2.20. 按委托号撤单

```
def cancelOrder(self, accountID: str, orderSyedId: str, exchangeId: str,
                instrumentId: str, nRequestId: int, accountKey: str = '') -> None:
```

- 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

| 参数名 | 参数释义 | 参数类型 |
|--------------|----------------|--------|
| accountID | 证券账号 | string |
| orderSyedId | 合同编号 | string |
| exchangeId | 市场代码 | string |
| instrumentId | 合约代码 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |

| 参数名 | 参数释义 | 参数类型 |
|------------|-------|--------|
| accountKey | 账号key | string |

3.2.21. 按指令号暂停指令并暂停任务 同步接口

```
def pauseSync(self, orderId: int, error: XtError) -> None:
```

- 释义：按指令号暂停指令并暂停任务
- 参数

| 参数 | 参数释义 | 参数类型 |
|---------|--------------------------------|---------|
| orderId | 下单的指令号，可通过onRtnOrder和onOrder获得 | int |
| error | 错误信息 | XtError |

3.2.22. 按指令号恢复指令并恢复任务 同步接口

```
def resumeSync(self, orderId: int, error: XtError) -> None:
```

- 释义：按指令号恢复指令并恢复任务
- 参数

| 参数 | 参数释义 | 参数类型 |
|---------|--------------------------------|---------|
| orderId | 下单的指令号，可通过onRtnOrder和onOrder获得 | int |
| error | 错误信息 | XtError |

3.2.23. 指令暂停恢复

```
def operateTask(self, op: CTaskOpRecord, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：指令暂停恢复
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------|---------------|
| op | 指令暂停恢复请求 | CTaskOpRecord |
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key，用于区分不同类型的账号ID相同的账号 | string |

3.2.24. 智能算法指令改单

```
def modifyAlgoCommands(self, orderInfo: CIntelligentAlgorithmOrder,
nOrderID: int, nRequestId: int, accountKey: str = '') -> None:
```

- 释义：智能算法指令改单
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------|----------------------------|
| orderInfo | 智能算法指令改单参数 | CIntelligentAlgorithmOrder |
| nOrderID | 需要改参的指令号 | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key，用于区分不同类型的账号ID相同的账号 | string |

3.2.25. 普通算法指令改单

```
def modifyAlgoCommands(self, orderInfo: CAlgorithmOrder, nOrderID: int,
nRequestId: int, accountKey: str = '') -> None:
```

- 释义：普通算法指令改单
- 参数

| 参数 | 参数释义 | 参数类型 |
|-----------|--|-----------------|
| orderInfo | 普通算法指令改单参数，只能改m_dPrice, m_nVolume, m_ePriceType, m_strRemark | CAlgorithmOrder |

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------|--------|
| nOrderID | 需要改参的指令号 | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key，用于区分不同类型的账号ID相同的账号 | string |

3.3. 查询接口

3.3.1. 获取账号key对应productId

```
def reqProductIdByAccountKey(self, accountKey: str) -> int:
```

- 释义：获取账号key对应productId
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------|--------|
| accountKey | 唯一标识一个账号的key | string |

3.3.2. 获取迅投交易用户名

```
def getUserName(self) -> str:
```

- 释义：获取迅投交易用户名

3.3.3. 获取Api版本

```
def getVersion(self) -> str:
```

- 释义：获取Api版本

3.3.4. 用户登陆 同步接口

```
def userLoginSync(self, userName: str, password: str, machineInfo: str
= None, ppid: str = None, authcode: str = None) -> XtError:
```

- 释义：用户登陆 同步接口

- 注意：调用此函数后，返回 XtError结构, 判断 isSuccess
- 参数

| 参数 | 参数释义 | 参数类型 |
|-------------|--|--------|
| userName | 用户名 | string |
| password | 用户密码 | string |
| machineInfo | 如果需要下单设置站点信息，则传入，否则可以传空 | string |
| appid | 如果是期货中继模式，需传入监管处申请的appid，则传入，否则可以传空 | string |
| authcode | 如果是期货中继模式，需传入监管处申请的authcode，则传入，否则可以传空 | string |

3.3.5. 用户登陆 异步接口

```
def userLogin(self, userName: str, password: str, nRequestId: int,
machineInfo: str = None, appid: str = None, authcode: str = None) -> None:
```

- 释义：用户登陆 异步接口
- note：调用此函数后，回调 onUserLogin
- 参数

| 参数 | 参数释义 | 参数类型 |
|-------------|--|--------|
| userName | 用户名 | string |
| password | 用户密码 | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| machineInfo | 如果需要下单设置站点信息，则传入，否则可以传空 | string |
| appid | 如果是期货中继模式，需传入监管处申请的appid，则传入，否则可以传空 | string |
| authcode | 如果是期货中继模式，需传入监管处申请的authcode，则传入，否则可以传空 | string |

3.3.6. 资产查询 同步接口

```
def reqAccountDetailSync(self, accountID: str, error: XtError,
    accountKey: str = '') -> CAccountDetail:
```

- 释义：查询资金账号对应的资产，返回CAccountDetail结构，处理返回数据之前需要先判断error.isSuccess() note:信用账号的资产数据需要调用reqCreditDetail接口
- 参数

| 参数名 | 参数释义 | 参数类型 |
|------------|-------|---------|
| accountID | 资金账号 | string |
| error | 错误信息 | XtError |
| accountKey | 账号key | string |

3.3.7. 资产查询 异步接口

```
def reqAccountDetail(self, accountID: str, nRequestId: int, accountKey:
    str = '') -> None:
```

- 释义：查询资金账号对应的资产，信用账号的资产数据需要调用reqCreditDetail接口 note: 调用此函数后，回调 onReqAccountDetail，两融账号还会回调onReqCreditAccountDetail
- 参数

| 参数名 | 参数释义 | 参数类型 |
|------------|-------|---------|
| accountID | 资金账号 | string |
| error | 错误信息 | XtError |
| accountKey | 账号key | string |

3.3.8. 请求账号委托明细信息

```
def reqOrderDetail(self, accountID: str, nRequestId: int,
    accountKey: str = '') -> None:
```

- 释义：查询资金账号对应的当日所有委托
- note：调用此函数后，回调 onReqOrderDetail
- 参数

| 参数名称 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.9. 请求账号委托明细信息 同步接口

```
def reqOrderDetailSync(self, accountID: str, error: XtError,
accountKey: str = '') -> list[COrderDetail]:
```

- 释义：查询资金账号对应的当日所有委托，返回 vector 对应 COrderDetail 结构，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数名称 | 参数释义 | 参数类型 |
|------------|-------|---------|
| accountID | 资金账号 | string |
| error | 错误信息 | XtError |
| accountKey | 账号key | string |

3.3.10. 根据指令号请求账号委托明细信息

```
def reqOrderDetail(self, accountID: str, nRequestId: int, nOrderID: int,
accountKey: str = '') -> None:
```

- 释义：根据指令号请求账号委托明细信息
- note：调用此函数后，回调 onReqOrderDetail
- 参数

| 参数名称 | 参数释义 | 参数类型 |
|-----------|------|--------|
| accountID | 资金账号 | string |

| 参数名称 | 参数释义 | 参数类型 |
|------------|--------------------------------|--------|
| nRequestId | 请求ID，每个请求的ID自增 | int |
| nOrderID | 下单的指令号，可通过onRtnOrder和onOrder获得 | int |
| accountKey | 账号key | string |

3.3.11. 同步根据指令号请求账号委托明细信息

```
def reqOrderDetailSyncByOrderID(self, accountID: str, error: XtError,
nOrderID: int, accountKey: str = '') -> list[COrderDetail]:
```

- 释义：根据指令号请求账号委托明细信息，返回 vector 对应 COrderDetail 结构，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数名称 | 参数释义 | 参数类型 |
|------------|------|---------|
| accountID | 账号ID | string |
| error | 错误信息 | XtError |
| orderID | 指令号 | int |
| accountKey | 错误信息 | string |

3.3.12. 同步请求账号成交明细信息

```
def reqDealDetailSync(self, accountID: str, error: XtError,
accountKey: str = '') -> list[CDealDetail]:
```

- 释义：同步请求账号成交明细信息，返回 deallist 对应 CDealDetail 结构，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数 | 参数释义 | 参数类型 |
|-----------|------|---------|
| accountID | 资金账号 | string |
| error | 错误信息 | XtError |

| 参数 | 参数释义 | 参数类型 |
|------------|-------|--------|
| accountKey | 账号key | string |

3.3.13. 同步请求账号成交统计信息

```
def reqDealStaticsSync(self, accountID: str, error: XtError,
accountKey: str = '') -> list[CDealStatics]:
```

- 释义：请求账号成交统计信息，返回 deallist 对应 CDealDetail 结构，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-------|---------|
| accountID | 资金账号 | string |
| error | 错误信息 | XtError |
| accountKey | 账号key | string |

3.3.14. 同步根据指令号请求账号成交明细信息

```
def reqDealDetailSyncByOrderID(self, accountID: str, error: XtError,
nOrderID: int, accountKey: str = '') -> list[CDealDetail]:
```

- 释义：同步根据指令号请求账号成交明细信息，返回 vector 对应 CDealDetail 结构，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数名称 | 参数释义 | 参数类型 |
|------------|------|---------|
| accountID | 账号ID | string |
| error | 错误信息 | XtError |
| orderID | 指令号 | int |
| accountKey | 错误信息 | string |

3.3.15. 请求账号成交明细信息

```
def reqDealDetail(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：请求账号成交明细信息，调用此函数后，回调 onReqDealDetail
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.16. 根据指令号请求账号成交明细信息

```
def reqDealDetail(self, accountID: str, nRequestId: int, nOrderID: int,
accountKey: str = '') -> None:
```

- 释义：请求账号成交明细信息，调用此函数后，回调 onReqDealDetail
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| nOrderID | 下单的指令号，可通过onRtnOrder和onOrder获得 | int |
| accountKey | 账号key | string |

3.3.17. 请求账号持仓明细信息

```
def reqPositionDetail(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：查询资金账号对应的持仓，调用此函数后，回调 onReqPositionDetail
- 参数

| 参数名 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.18. 同步请求账号持仓明细信息

```
def reqPositionDetailSync(self, accountID: str, error: XtError,
accountKey: str = '') -> list[CPositionDetail]:
```

- 释义：查询资金账号对应的持仓，返回 vector 对应 CPositionDetail 结构，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数名 | 参数释义 | 参数类型 |
|------------|-------|---------|
| accountID | 资金账号 | string |
| error | 错误信息 | XtError |
| accountKey | 账号key | string |

3.3.19. 请求账号持仓统计信息

```
def reqPositionStatics(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：请求账号持仓统计信息，调用此函数后，回调 onReqPositionStatics
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.20. 同步请求账号持仓统计信息

```
def reqPositionStaticsSync(self, accountID: str, error: XtError,
accountKey: str = '') -> list[CPositionStatics]:
```

- 释义：同步请求账号持仓统计信息，返回 vector 对应 CPositionStatics 结构，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-------|---------|
| accountID | 资金账号 | string |
| error | 错误信息 | XtError |
| accountKey | 账号key | string |

3.3.21. 请求账号日初持仓统计信息

```
def reqInitialPositionStatics(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：请求账号日初持仓统计信息，调用此函数后，回调onReqInitialPositionStatics
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.22. 同步请求账号日初持仓统计信息

```
def reqInitialPositionStaticsSync(self, accountID: str, error: XtError,
accountKey: str = '') -> list[CPositionStatics]:
```

- 释义：同步请求账号日初持仓统计信息，返回 vector 对应 CPositionStatics 结构，处理返回数据之前需要先判断error.isSuccess()

- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-------|---------|
| accountID | 资金账号 | string |
| error | 错误信息 | XtError |
| accountKey | 账号key | string |

3.3.23. 请求信用账号标的信息

```
def reqStksubjects(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：请求信用账号标的信息，调用此函数后，回调 onReqStksubjects
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.24. 请求期权账号备兑持仓信息

```
def reqCoveredStockPosition(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：请求期权账号备兑持仓信息，调用此函数后，回调 onReqCoveredStockPosition
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.25. 请求期权账号组合持仓信息

```
def reqStkOptCombPositionDetail(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：请求期权账号组合持仓信息，调用此函数后，回调 onReqStkOptCombPositionDetail
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.26. 请求账号汇率信息

```
def reqGGTReferenceRate(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：请求账号汇率信息，调用此函数后，回调 onReqReferenceRate
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.27. 请求两融账号综合资金信息

```
def reqCreditDetail(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：请求两融账号综合资金信息，调用此函数后，回调 onReqCreditDetail
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID, 每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.28. 上传终端ctp采集信息，用于传入ctp柜台需要留痕的终端信息

```
def registerUserSystemInfo(self, accountID: str, IpPortAddr: str,
len: int, CTPSystemInfo: str, nRequestId: int, accountKey: str = '')
-> None:
```

- 释义：上传终端ctp采集信息，用于传入ctp柜台需要留痕的终端信息
- 参数

| 参数 | 参数释义 | 参数类型 |
|---------------|---|--------|
| accountID | 资金账号 | string |
| IpPortAddr | 上传终端信息机器的ip和端口信息(格式为ip:port, 例如127.0.0.1:58000) | string |
| len | ctp采集信息内容的长度len, 即CTP_GetSystemInfo入参nLen | int |
| CTPSystemInfo | ctp采集到的内容 | string |
| nRequestId | 请求ID, 每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.29. 设置用户自己的定时器

```
def startNamedTimer(self, timerName: str, intervalInMilliSecond: int,
time: int, nextDay: bool) -> None:
```

- 释义：设置用户自己的定时器，调用此函数后，回调 onNamedTimer
- 参数

| 参数 | 参数释义 | 参数类型 |
|-----------------------|--|--------|
| timerName | 定时器名称 | string |
| intervalInMilliSecond | 定时器间隔，单位毫秒 | int |
| time | time 定时器第一次触发的时间，单位毫秒，如14点25分36秒789毫秒就是142536789，输入负数表示从当前时间开始计时 | int |
| nextDay | 触发的时间是否在下一日 | bool |

3.3.30. 停止用户自己的定时器

```
def stopNamedTimer(self, timerName: str) -> None:
```

- 释义：停止用户自己的定时器
- 参数

| 参数 | 参数释义 | 参数类型 |
|-----------|-------|--------|
| timerName | 定时器名称 | string |

3.3.31. 设置用户下单指令冻结选项

```
def setCmdFrzCheckOption(self, nCmdFrzCheckOption: int) -> None:
```

- 释义：设置用户下单指令冻结选项
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------------|----------------------|------|
| nCmdFrzCheckOption | 用户下单指令冻结选项，1 禁止 2 警告 | int |

3.3.32. 请求账号新股额度信息

```
def reqSubscribeInfo(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：请求账号新股额度信息，调用此函数后，回调 onReqSubscribeInfo

- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.33. 请求信用账号未了结负债信息

```
def reqStkUnCloseCompacts(self, accountID: str, nRequestId: int,
    accountKey: str = '') -> None:
```

- 释义：请求信用账号未了结负债信息，调用此函数后，回调 onReqStkUnCloseCompact
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.34. 请求信用账号未了结负债信息 同步接口

```
def reqStkUnCloseCompactsSync(self, accountID: str, error: XtError,
    accountKey: str = '') -> list[CStkUnClosedCompacts]:
```

- 释义：请求信用账号未了结负债信息，返回 vector 对应 CStkUnClosedCompacts 结构，处理返回数据之前需要先判断 error.isSuccess()
- 参数

| 参数 | 参数释义 | 参数类型 |
|-----------|---------------------|---------|
| accountID | 资金账号 | string |
| error | 错误信息，用于接收同步请求时的错误信息 | XtError |

| 参数 | 参数释义 | 参数类型 |
|------------|-------|--------|
| accountKey | 账号key | string |

3.3.35. 请求信用账号已了结负债信息

```
def reqStkClosedCompacts(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：请求信用账号已了结负债信息，调用此函数后，回调 onReqStkClosedCompact
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.36. 请求信用账号已了结负债信息 同步接口

```
def reqStkClosedCompactsSync(self, accountID: str, error: XtError,
accountKey: str = '') -> list[CStkClosedCompacts]:
```

- 释义：请求信用账号已了结负债信息，返回 vector 对应 CStkClosedCompacts 结构，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------------|---------|
| accountID | 资金账号 | string |
| error | 错误信息，用于接收同步请求时的错误信息 | XtError |
| accountKey | 账号key | string |

3.3.37. 获取用户下所有账号key 同步接口

```
def reqAccountKeysSync(self, error: XtError) -> list[CAccountKey]:
```

- 释义：获取用户下所有账号key，返回 vector 对应 CAccountKey 结构，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数 | 参数释义 | 参数类型 |
|-------|---------------------|---------|
| error | 错误信息，用于接收同步请求时的错误信息 | XtError |

3.3.38. 获取用户下所有账号key

```
def reqAccountKeys(self, nRequestId: int) -> None:
```

- 释义：获取用户下所有账号key，调用此函数后，回调 onReqAccountKey
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|------|
| nRequestId | 请求ID，每个请求的ID自增 | int |

3.3.39. 根据委托号请求账号成交明细信息

```
def reqDealDetailBySysID(self, accountID: str, nRequestId: int,
orderSysId: str, exchangeId: str, accountKey: str = '') -> None:
```

- 释义：根据委托号请求账号成交明细信息，调用此函数后，回调 onReqDealDetailBySysID
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| orderSysId | 下单的委托号 | string |
| exchangeId | 委托所属市场 | string |
| accountKey | 账号key | string |

3.3.40. 请求账号结算单信息

```
def reqDeliveryDetail(self, accountID: str, startDate: str, endDate: str,
nRequestId: int, accountKey: str = '') -> None:
```

- 释义：请求账号结算单信息，调用此函数后，回调 onReqDeliveryDetail
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| startDate | 查询开始时间 | string |
| endDate | 查询结束时间 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.41. 请求账号合约行情信息

```
def reqSingleInstrumentInfo(self, exchangeId: str, instrumentId: str,
nRequestId: int) -> None:
```

- 释义：请求账号合约行情信息，调用此函数后，回调 onReqSingleInstrumentInfo
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|----------------|--------|
| exchangeId | 市场代码 | string |
| instrumentId | 合约代码 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |

3.3.42. 请求账号可下单量

```
def reqOpVolume(self, opVolumeReq: COpVolumeReq, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：请求账号可下单量，调用此函数后，回调 onReqOpVolume
- 参数

| 参数 | 参数释义 | 参数类型 |
|-------------|--------------------------|--------|
| opVolumeReq | 可下单量请求参数 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号Key，用于区分不同类型的账号ID相同的账号 | string |

3.3.43. 请求账号可下单量

```
def reqCanOrderVolume(self, opVolumeReq: COpVolumeReq, nRequestId: int,
    accountKey: str = '') -> None:
```

- 释义：请求账号可下单量 (根据迅投系统当前持仓和资金计算，可能跟实际情况有误差)，调用此函数后，回调 onReqCanOrderVolume
- 参数

| 参数 | 参数释义 | 参数类型 |
|-------------|--------------------------|--------|
| opVolumeReq | 可下单量请求参数 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号Key，用于区分不同类型的账号ID相同的账号 | string |

3.3.44. 请求账号可下单量 同步接口

```
def reqCanOrderVolumeSync(self, opVolumeReq: COpVolumeReq,
    error: XtError, accountKey: str = '') -> int:
```

- 释义：请求账号可下单量 (根据迅投系统当前持仓和资金计算，可能跟实际情况有误差)，返回可下单量，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数 | 参数释义 | 参数类型 |
|-------------|---------------------|---------|
| opVolumeReq | 可下单量请求参数 | string |
| error | 错误信息，用于接收同步请求时的错误信息 | XtError |

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------|--------|
| accountKey | 账号Key，用于区分不同类型的账号ID相同的账号 | string |

3.3.45. 请求两融账号融券可融数量信息

```
def reqCreditSloCode(self, accountID: str, nRequestId: int, accountKey:
    str = '') -> None:
```

- 释义：请求两融账号融券可融数量信息，调用此函数后，回调 onReqCreditSloCode
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.46. 请求两融账号融资融券标的的信息

```
def reqCreditSubjects(self, accountID: str, nRequestId: int,
    accountKey: str = '') -> None:
```

- 释义：请求两融账号融资融券标的的信息，调用此函数后，回调 onReqCreditSubjects
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.47. 请求两融账号担保标的信息

```
def reqCreditAssure(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：请求两融账号担保标的信息，调用此函数后，回调 onReqCreditAssure
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.48. 请求账号银证转账银行信息

```
def reqTransferBank(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：请求账号银证转账银行信息，调用此函数后，回调 onReqTransferBank
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.49. 请求账号银证转账银行流水

```
def reqTransferSerial(self, accountID: str, startDate: str, endDate: str,
nRequestId: int, accountKey: str = '') -> None:
```

- 释义：请求账号银证转账银行流水，调用此函数后，回调 onReqTransferSerial
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-----------------|--------|
| accountID | 资金账号 | string |
| startDate | 查询开始时间 | string |
| endDate | 查询结束时间 | string |
| nRequestId | 请求ID, 每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.50. 请求账号银证转账银行余额

```
def reqBankAmount(self, bankInfo: CQueryBankInfo, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义: 请求账号银证转账银行余额, 调用此函数后, 回调 onReqBankAmount
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-----------------|----------------|
| bankInfo | 银证转账银行余额请求参数 | CQueryBankInfo |
| nRequestId | 请求ID, 每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.51. 银证转账

```
def transfer(self, transferReq: CTransferReq, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义: 银证转账, 调用此函数后, 回调 onTransfer
- 参数

| 参数 | 参数释义 | 参数类型 |
|-------------|-----------------|--------------|
| transferReq | 银证转账请求参数 | CTransferReq |
| nRequestId | 请求ID, 每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.52. 请求账号可撤单委托明细信息

```
def reqCanCancelOrderDetail(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：请求账号可撤单委托明细信息，调用此函数后，回调 onReqCanCancelOrderDetail
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.53. 请求账户所有指令信息

```
def reqCommandsInfo(self, nRequestId: int) -> None:
```

- 释义：请求账户所有指令信息，调用此函数后，回调 onReqCommandsInfo
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|------|
| nRequestId | 请求ID，每个请求的ID自增 | int |

3.3.54. 请求账户所有指令信息 同步接口

```
def reqCommandsInfoSync(self, error: XtError) -> list[COrderInfo]:
```

- 释义：请求账户所有指令信息，返回vector是COrderInfo，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数 | 参数释义 | 参数类型 |
|-------|---------------------|---------|
| error | 错误信息，用于接收同步请求时的错误信息 | XtError |

3.3.55. 资金划拨

```
def fundTransfer(self, fundTransferReq: CSecuFundTransferReq, nRequestId: int,
    accountKey: str = '') -> None:
```

- 释义：资金划拨，调用此函数后，回调 onFundTransfer
- 参数

| 参数 | 参数释义 | 参数类型 |
|-----------------|----------------|----------------------|
| fundTransferReq | 资金划拨请求参数 | CSecuFundTransferReq |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号Key | string |

3.3.56. 股份划拨

```
def secuTransfer(self, secuTransferReq: CSecuFundTransferReq, nRequestId: int,
    accountKey: str = '') -> None:
```

- 释义：股份划拨，调用此函数后，回调 onSecuTransfer
- 参数

| 参数 | 参数释义 | 参数类型 |
|-----------------|----------------|----------------------|
| secuTransferReq | 股份划拨请求参数 | CSecuFundTransferReq |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号Key | string |

3.3.57. 请求账号普通（集中）柜台资金

```
def reqComFund(self, accountID: str, nRequestId: int, accountKey: str = '') -> None:
```

- 释义：请求账号普通（集中）柜台资金，调用此函数后，回调 onReqComFund
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID, 每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.58. 请求账号普通（集中）柜台持仓

```
def reqComPosition(self, accountID: str, nRequestId: int,
    accountKey: str = '') -> None:
```

- 释义：请求账号普通（集中）柜台持仓，调用此函数后，回调 onReqComPosition
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID, 每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.59. 获取当前交易日

```
def reqTradeDay(self, nRequestId: int) -> None:
```

- 释义：获取当前交易日，调用此函数后，回调 onReqTradeDay
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-----------------|------|
| nRequestId | 请求ID, 每个请求的ID自增 | int |

3.3.60. 请求账号历史委托明细

```
def reqHistoryOrderDetail(self, accountID: str, startDate: str,
    endDate: str, nRequestId: int, accountKey: str = '') -> None:
```

- 释义：请求账号历史委托明细，调用此函数后，回调 onReqHistoryOrderDetail
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| startDate | 查询开始时间 | string |
| endDate | 查询结束时间 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.61. 请求账号历史委托明细 同步接口

```
def reqHistoryOrderDetailSync(self, accountID: str, startDate: str,
endDate: str, error: XtError, accountKey: str = '') -> list[COrderDetail]:
```

- 释义：请求账号历史委托明细 同步接口，返回vector是COrderDetail，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------------|---------|
| accountID | 资金账号 | string |
| startDate | 查询开始时间 | string |
| endDate | 查询结束时间 | string |
| error | 错误信息，用于接收同步请求时的错误信息 | XtError |
| accountKey | 账号key | string |

3.3.62. 请求账号历史成交明细

```
def reqHistoryDealDetail(self, accountID: str, startDate: str,
endDate: str, nRequestId: int, accountKey: str = '') -> None:
```

- 释义：请求账号历史成交明细，调用此函数后，回调 onReqHistoryDealDetail
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| startDate | 查询开始时间 | string |
| endDate | 查询结束时间 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.63. 请求账号历史成交明细 同步接口

```
def reqHistoryDealDetailSync(self, accountID: str, startDate: str,
endDate: str, error: XtError, accountKey: str = '') -> list[CDealDetail]:
```

- 释义：请求账号历史成交明细同步接口，返回vector是CDealDetail，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------------|---------|
| accountID | 资金账号 | string |
| startDate | 查询开始时间 | string |
| endDate | 查询结束时间 | string |
| error | 错误信息，用于接收同步请求时的错误信息 | XtError |
| accountKey | 账号key | string |

3.3.64. 请求账号历史持仓统计

```
def reqHistoryPositionStatics(self, accountID: str, startDate: str,
endDate: str, nRequestId: int, accountKey: str = '') -> None:
```

- 释义：请求账号历史持仓统计，调用此函数后，回调 onReqHistoryPositionStatics

- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-----------------|--------|
| accountID | 资金账号 | string |
| startDate | 查询开始时间 | string |
| endDate | 查询结束时间 | string |
| nRequestId | 请求ID, 每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.65. 请求账号历史持仓统计 同步接口

```
def reqHistoryPositionStaticsSync(self, accountID: str, startDate: str,
endDate: str, error: XtError, accountKey: str = '') -> list[CPositionStatics]:
```

- 释义: 请求账号历史持仓统计同步接口, 返回vector是CPositionStatics, 处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------------|---------|
| accountID | 资金账号 | string |
| startDate | 查询开始时间 | string |
| endDate | 查询结束时间 | string |
| error | 错误信息, 用于接收同步请求时的错误信息 | XtError |
| accountKey | 账号key | string |

3.3.66. 请求账号历史资金信息 同步接口

```
def reqHistoryAccountDetailSync(self, accountID: str, startDate: str,
endDate: str, error: XtError, accountKey: str = '') -> list[CAccountDetail]:
```

- 释义: 请求账号历史资金信息同步接口, 返回vector是CAccountDetail, 处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------------|---------|
| accountID | 资金账号 | string |
| startDate | 查询开始时间 | string |
| endDate | 查询结束时间 | string |
| error | 错误信息，用于接收同步请求时的错误信息 | XtError |
| accountKey | 账号key | string |

3.3.67. 请求期货账号的保证金率

```
def reqFtAccCommissionRateDetail(self, accountID: str, exchangeId: str,
instrumentId: str, nRequestId: int, accountKey: str = '') -> None:
```

- 释义：请求期货账号的保证金率，调用此函数后，回调 onReqFtAccCommissionRateDetail
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|----------------|--------|
| accountID | 资金账号 | string |
| exchangeId | 市场代码 | string |
| instrumentId | 合约代码 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.68. 请求期货账号的手续费率

```
def reqFtAccMarginRateDetail(self, accountID: str, exchangeId: str,
instrumentId: str, nRequestId: int, accountKey: str = '') -> None:
```

- 释义：请求期货账号的手续费率，调用此函数后，回调 onReqFtAccMarginRateDetail
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|----------------|--------|
| accountID | 资金账号 | string |
| exchangeId | 市场代码 | string |
| instrumentId | 合约代码 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.3.69. 获取用户下所有的产品Id

```
def reqProductIds(self, nRequestId: int) -> None:
```

- 释义：获取用户下所有的产品Id，调用此函数后，回调 onReqProductIds
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|------|
| nRequestId | 请求ID，每个请求的ID自增 | int |

3.3.70. 创建新投资组合

```
def createPortfolio(self, newPortfolioReq: CNewPortfolioReq,
nRequestId: int) -> None:
```

- 释义：创建新投资组合，调用此函数后，回调 onCreatePortfolio
- 参数

| 参数 | 参数释义 | 参数类型 |
|-----------------|----------------|------------------|
| newPortfolioReq | 创建新投资组合请求参数 | CNewPortfolioReq |
| nRequestId | 请求ID，每个请求的ID自增 | int |

3.3.71. 查询产品Id下所有的投资组合

```
def reqProductPortfolio(self, nProductID: int, nRequestId: int)
-> None:
```

- 释义：查询产品ID下所有的投资组合，调用此函数后，回调 onReqProductPortfolio
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|------|
| nProductID | 产品ID | int |
| nRequestId | 请求ID，每个请求的ID自增 | int |

3.3.72. 请求投资组合委托信息

```
def reqPortfolioOrder(self, nPortfolioID: int, nDate: int,
nRequestId: int) -> None:
```

- 释义：请求投资组合委托信息，调用此函数后，回调 onReqPortfolioOrder
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|----------------|------|
| nPortfolioID | 投资组合ID | int |
| nDate | 查询日期 | int |
| nRequestId | 请求ID，每个请求的ID自增 | int |

3.3.73. 请求投资组合一段时间内的委托信息

```
def reqPortfolioMultiOrder(self, nPortfolioID: int, nFromDate: int,
nToDate: int, nRequestId: int) -> None:
```

- 释义：请求投资组合一段时间内的委托信息，调用此函数后，回调 onReqPortfolioMultiOrder
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|--------|------|
| nPortfolioID | 投资组合ID | int |
| startDate | 查询开始日期 | int |
| endDate | 查询结束日期 | int |

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|------|
| nRequestId | 请求ID，每个请求的ID自增 | int |

3.3.74. 请求投资组合成交信息

```
def reqPortfolioDeal(self, nPortfolioID: int, nDate: int,
nRequestId: int) -> None:
```

- 释义：请求投资组合成交信息，调用此函数后，回调 onReqPortfolioDeal
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|----------------|------|
| nPortfolioID | 投资组合ID | int |
| nDate | 查询日期 | int |
| nRequestId | 请求ID，每个请求的ID自增 | int |

3.3.75. 请求投资组合一段时间内的委托信息

```
def reqPortfolioMultiOrder(self, nPortfolioID: int, nFromDate: int,
nToDate: int, nRequestId: int) -> None:
```

- 释义：请求投资组合一段时间内的成交信息，调用此函数后，回调 onReqPortfolioMultiDeal
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|----------------|------|
| nPortfolioID | 投资组合ID | int |
| startDate | 查询开始日期 | int |
| endDate | 查询结束日期 | int |
| nRequestId | 请求ID，每个请求的ID自增 | int |

3.3.76. 请求投资组合持仓信息

```
def reqPortfolioPosition(self, nPortfolioID: int, nDate: int,  
    nRequestId: int) -> None:
```

- 释义：请求投资组合持仓信息，调用此函数后，回调 onReqPortfolioPosition
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|----------------|------|
| nPortfolioID | 投资组合ID | int |
| nDate | 查询日期 | int |
| nRequestId | 请求ID，每个请求的ID自增 | int |

3.3.77. 请求收益互换账号框架号

```
def reqStrategyInfo(self, accountID: str, nRequestId: int,  
    accountKey: str = '') -> None:
```

- 释义：请求收益互换账号框架号，调用此函数后，回调 onReqStrategyInfo
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 账号ID | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号Key | string |

3.3.78. 请求股东号

```
def reqSecuAccount(self, accountID: str, nRequestId: int,  
    accountKey: str = '') -> None:
```

- 释义：请求股东号，用于多股东时指定股东号，调用此函数后，回调 onReqSecuAccount
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-----------------|--------|
| accountID | 账号ID | string |
| nRequestId | 请求ID, 每个请求的ID自增 | int |
| accountKey | 账号Key | string |

3.3.79. 请求资金账号状态

```
def reqAccountStatus(self, nRequestId: int, accountKey: str = '') -> None:
```

- 释义：请求资金账号状态，调用此函数后，回调 onReqAccountStatus
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-----------------|--------|
| nRequestId | 请求ID, 每个请求的ID自增 | int |
| accountKey | 账号Key | string |

3.3.80. 请求资金账号状态 同步接口

```
def reqAccountStatusSync(self, accountKey: str = '') -> bool:
```

- 释义：请求资金账号状态，调用此函数后，返回状态
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-------|--------|
| accountKey | 账号Key | string |

3.3.81. 查询投资组合资金信息

```
def reqPortfolioStats(self, nPortfolioIds: list[int], nTradeDate: int, nRequestId: int) -> None:
```

- 释义：查询投资组合资金信息，调用此函数后，回调 onReqPortfolioStat
- 参数

| 参数 | 参数释义 | 参数类型 |
|---------------|----------------------------|--------------|
| nPortfolioIds | 投资组合id | vector< int> |
| nTradeDate | 查询日期，如果需要查当天，nTradeDate送-1 | int |
| nRequestId | 请求ID，每个请求的ID自增 | int |

3.3.82. 请求市场状态信息

```
def reqExchangeStatus(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：请求市场状态信息，调用此函数后，回调 onReqExchangeStatus
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 账号ID | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号Key | string |

3.3.83. 查询枚举名称

```
def reqEnumItemName(self, enumName: str, enumItemValue: int) -> str:
```

- 释义：查询枚举名称，调用此函数后，返回枚举中文名称
- 参数

| 参数 | 参数释义 | 参数类型 |
|---------------|--------|--------|
| enumName | 枚举类型名称 | string |
| enumItemValue | 枚举值 | int |

3.3.84. 请求用户产品信息

```
def reqProductData(self, nRequestId: int) -> None:
```

- 释义：请求用户产品信息，调用此函数后，回调 onReqProductData
- 参数

| 参数名 | 参数释义 | 参数类型 |
|------------|----------------|------|
| nRequestId | 请求ID，每个请求的ID自增 | int |

3.3.85. 请求用户产品信息 同步接口

```
def reqProductDataSync(self, error: XtError) -> list[CProductData]:
```

- 释义：请求用户产品信息，调用此函数后，返回的vector的结构是 CProductData。
- 参数

| 参数名 | 参数释义 | 参数类型 |
|-------|------|---------|
| error | 错误信息 | XtError |

3.3.86. 请求两融账号综合资金信息

```
def reqCreditDetailSync(self, accountID: str, error: XtError,
accountKey: str = '') -> CCreditDetail:
```

- 释义：请求两融账号综合资金信息，调用此函数后，返回 CCreditDetail
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-------|---------|
| accountID | 资金账号 | string |
| error | 错误信息 | XtError |
| accountKey | 账号key | string |

3.3.87. 查询ETF申赎清单

```
def reqEtfDetails(self, nRequestId: int) -> None:
```

- 释义：查询ETF申赎清单，调用此函数后，返回 onReqEtfDetails

- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|------|------|
| nRequestId | 请求编号 | int |

3.3.88. 查询ETF申赎清单 同步接口

```
def reqEtfDetailsSync(self, error: XtError) -> list[CETFPCFDetail]:
```

- 释义：查询ETF申赎清单，调用此函数后，返回 std::vector<xti::CETFPCFDetail>
- 参数

| 参数 | 参数释义 | 参数类型 |
|-------|------|---------|
| error | 错误信息 | XtError |

3.3.89. 查询ETF成分股信息

```
def reqETFComponentStockInfo(self, exchangeId: str, instrumentId: str,
nRequestId: int) -> None:
```

- 释义：查询ETF成分股信息，调用此函数后，回调 onReqETFComponentStockInfo
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|------|--------|
| exchangeId | 市场代码 | string |
| instrumentId | 合约代码 | string |
| nRequestId | 请求编号 | int |

3.3.90. 查询ETF成分股信息 同步接口

```
def reqETFComponentStockInfoSync(self, exchangeId: str, instrumentId: str,
error: XtError) -> list[CETFComponentStockInfo]:
```

- 释义：查询ETF成分股信息，调用此函数后，返回

std::vector<xt::CETFComponentStockInfo>

- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|------|---------|
| exchangeId | 市场代码 | string |
| instrumentId | 合约代码 | string |
| error | 错误信息 | XtError |

3.3.91. 请求期货账号持仓统计信息 同步接口

```
def reqFuturePositionStaticsSync(self, accountID: str, nRequestId:
    XtError, accountKey: str) -> list[CFuturePositionStatics]:
```

- 释义：同步请求期货账号持仓统计信息，调用此函数后，返回std::vector<xt::CFuturePositionStatics>
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-------|---------|
| accountID | 账号ID | string |
| error | 错误信息 | XtError |
| accountKey | 账号Key | string |

3.3.92. 请求期货账号持仓统计信息

```
def reqFuturePositionStatics(self, accountID: str, nRequestId: int,
    accountKey: str) -> None:
```

- 释义：请求期货账号持仓统计信息，调用此函数后，回调onReqFuturePositionStatics
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|------|--------|
| accountID | 账号ID | string |
| nRequestId | 请求编号 | int |

| 参数 | 参数释义 | 参数类型 |
|------------|-------|--------|
| accountKey | 账号Key | string |

3.3.93. 查询用户设置

```
def reqUserConfig(self, metaID: int, startDate: str, endDate: str, strTag: str,
    nRequestId: int) -> None:
```

- 释义：查询用户设置，调用此函数后，回调 onReqUserConfig
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|------------------|--------|
| metaID | 数据ID, 查询用户自定义送 3 | int |
| startDate | 起始日期 | string |
| endDate | 终止日期 | string |
| strTag | 附加信息 | string |
| nRequestId | 请求编号 | int |

3.3.94. 查询用户设置 同步接口

```
def reqUserConfigSync(self, metaID: int, startDate: str, endDate: str, strTag: str,
    error: XtError) -> list[CUserConfig]:
```

- 释义：查询用户设置，调用此函数后，回调CUserConfig组成的vector
- 参数

| 参数 | 参数释义 | 参数类型 |
|-----------|------------------|--------|
| metaID | 数据ID, 查询用户自定义送 3 | int |
| startDate | 起始日期 | string |
| endDate | 终止日期 | string |
| strTag | 附加信息 | string |

| 参数 | 参数释义 | 参数类型 |
|-------|------|---------|
| error | 错误信息 | XtError |

3.3.95. 请求市场状态信息

```
def reqExchangeStatus(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：请求市场状态信息，调用此函数后，回调onReqExchangeStatus
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------|--------|
| accountID | 账号ID | string |
| nRequestId | 请求编号 | int |
| accountKey | 账号Key，用于区分不同类型的账号ID相同的账号 | string |

3.3.96. 按市场请求合约信息 同步接口

```
def reqInstrumentInfoByMarket(self, exchangeId: str, nRequestId: int,
tradable: int = 0) -> None:
```

- 释义：按市场请求合约信息，调用此函数后，回调onReqInstrumentInfoByMarket
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-----------------------|--------|
| exchangeId | 市场，取值参考MarketType.h | string |
| nRequestId | 请求编号 | int |
| tradable | 是否只查询可交易合约，默认0,否， 1,是 | int |

3.3.97. 刷新信用账号未了结负债信息

```
def refreshStkUnCloseCompacts(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：刷新信用账号未了结负债信息，调用此函数后，主动向柜台同步未了结负债信息
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------|--------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key，用于区分不同类型的账号ID相同的账号 | string |

3.3.98. 刷新信用账号已了结负债信息

```
def refreshStkClosedCompacts(self, accountID: str, nRequestId: int,
    accountKey: str = '') -> None:
```

- 释义：刷新信用账号已了结负债信息，调用此函数后，主动向柜台同步未了结负债信息
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------|--------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key，用于区分不同类型的账号ID相同的账号 | string |

3.4. 行情接口

3.4.1. 请求行情数据信息

```
def reqPriceData(self, exchangeId: str, instrumentId: str,
    nRequestId: int) -> None:
```

- 释义：请求行情数据信息，调用此函数后，回调 onReqPriceData
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|---------------------|--------|
| exchangeId | 市场，取值参考MarketType.h | string |
| instrumentId | 合约代码 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |

3.4.2. 请求行情数据信息 同步接口

```
def reqPriceDataSync(self, exchangeId: str, instrumentId: str,
error: XtError) -> CPriceData:
```

- 释义：请求行情数据信息 同步接口，返回 vector 对应 CPriceData 结构，处理返回数据之前需要先判断error.isSuccess()
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|---------------------------|---------|
| exchangeId | 市场，取值参考MarketType.h | string |
| instrumentId | 合约代码 | string |
| error | error 错误信息，用于接收同步请求时的错误信息 | XtError |

3.4.3. 按市场请求行情数据信息 已废弃

```
def reqPriceDataByMarket(self, exchangeId: str, nRequestId: int,
tradable: int = 0) -> None:
```

- 已废弃，后续不在维护，改用reqInstrumentInfoByMarket
- 释义：按市场请求行情数据信息，调用此函数后，回调onReqCInstrumentDetail
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|------------------------|--------|
| exchangeId | 市场，取值参考MarketType.h | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| tradable | 是否只查询可交易合约，默认0,否， 1， 是 | int |

3.4.4. 请求账号期权行情信息

```
def reqInstrumentDetail(self, accountID: str, nRequestId: int,
accountKey: str = '') -> None:
```

- 释义：请求账号期权行情信息，调用此函数后，回调 onReqCInstrumentDetail
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|--------|
| accountID | 资金账号 | string |
| nRequestId | 请求ID，每个请求的ID自增 | int |
| accountKey | 账号key | string |

3.4.5. 订阅行情数据

```
def subscribQuote(self, data: CSubscribData, nRequestId: int) -> None:
```

- 释义：订阅行情数据，调用此函数后，回调 onSubscribQuote
- note: code为"allCode"时，订阅整个市场
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|---------------|
| data | 订阅行情参数 | CSubscribData |
| nRequestId | 请求ID，每个请求的ID自增 | int |

3.4.6. 批量订阅行情数据

```
def batchSubscribQuote(self, data: list[CSubscribData],
nRequestId: int) -> None:
```

- 释义：批量订阅行情数据，调用此函数后，回调 onSubscribQuote
- note: code为"allCode"时，订阅整个市场
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-----------------|------------------------|
| data | 订阅行情参数 | vector< CSubscribData> |
| nRequestId | 请求ID, 每个请求的ID自增 | int |

3.4.7. 退订行情数据

```
def unSubscribQuote(self, data: CSubscribData, nRequestId: int) -> None:
```

- 释义: 退订行情数据, 调用此函数后, 回调 onUnSubscribQuote
- note: code为"allCode"时, 订阅整个市场
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-----------------|---------------|
| data | 订阅行情参数 | CSubscribData |
| nRequestId | 请求ID, 每个请求的ID自增 | int |

3.4.8. 批量退订行情数据

```
def batchUnSubscribQuote(self, data: list[CSubscribData],  
nRequestId: int) -> None:
```

- 释义: 批量退订行情数据, 调用此函数后, 回调 onUnSubscribQuote
- note: code为"allCode"时, 订阅整个市场
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-----------------|------------------------|
| data | 订阅行情参数 | vector< CSubscribData> |
| nRequestId | 请求ID, 每个请求的ID自增 | int |

3.4.9. 按市场请求合约信息

```
def reqInstrumentInfoByMarket(self, exchangeId: str,  
nRequestId: int, tradable: int = 0) -> None:
```

- 释义: 按市场请求合约信息, 调用此函数后, 回调

onReqInstrumentInfoByMarket

- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-----------------|--------|
| exchangeId | 市场代码 | string |
| nRequestId | 请求ID, 每个请求的ID自增 | int |

4. XtTraderApiCallback

4.0 XtTraderApiCallback 类说明

XtTraderApiCallback是api回调接口类, 所有的异步请求回调和数据推送都是通过该类返回, 回调函数中的nRequestId 和请求函数中 nRequestId 相对应, error反应请求函数的返回是否成功, 如果失败有错误信息。

4.1. 连接类回调

4.1.1. 连接服务器的回调函数

```
def onConnected(self, success: bool, errorMsg: str) -> None:
```

- 释义: 连接服务器的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|----------|-------------------|--------|
| success | 服务器连接是否成 | bool |
| errorMsg | 如果服务器连接失败, 存储错误信息 | string |

4.1.2. 用户登录的回调函数

```
def onUserLogin(self, userName: str, password: str, nRequestId: int,  
error: XtError) -> None:
```

- 释义: 用户登录的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|----------|------------------|--------|
| success | 服务器连接是否成 | bool |
| errorMsg | 如果服务器连接失败，存储错误信息 | string |

4.2. 交易接口回调

4.2.1. 下单的回调函数

```
def onOrder(self, nRequestId: int, orderID: int, strRemark: str,
error: XtError) -> None:
```

- 释义：下单的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|---------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| orderID | 指令号 | int |
| strRemark | 下单时填写的投资备注 | string |
| error | 错误信息 | XtError |

4.2.2. 直接下单的回调函数

```
def onDirectOrder(self, nRequestId: int, strOrderSysID: str,
strRemark: str, error: XtError) -> None:
```

- 释义：直接下单的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|---------------|---------------|--------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| strOrderSysID | 委托号 | string |
| strRemark | 下单时填写的投资备注 | string |

| 参数 | 参数释义 | 参数类型 |
|-------|------|---------|
| error | 错误信息 | XtError |

4.2.3. 撤指令的回调函数

```
def onCancel(self, nRequestId: int, error: XtError) -> None:
```

- 释义：撤指令的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|---------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| error | 错误信息 | XtError |

4.2.4. 撤指令的回调函数

```
def onCancelWithRemark(self, nRequestId: int, strRemark: str,
error: XtError) -> None:
```

- 释义：撤指令的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------------------|---------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| strRemark | 下单时填写的投资备注，如果撤销的指令号不存在返回空 | string |
| error | 错误信息 | XtError |

4.2.5. 撤委托的回调函数

```
def onCancelOrder(self, nRequestId: int, error: XtError) -> None:
```

- 释义：撤委托的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|---------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| error | 错误信息 | XtError |

4.2.6. 风险试算的回调函数

```
def onCheck(self, nRequestId: int, data: CCheckData,
error: XtError) -> None:
```

- 释义：风险试算的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 风险试算结果 | CCheckData |
| error | 错误信息 | XtError |

4.2.7. 暂停恢复任务回调

```
def onOperateTask(self, accountID: str, nRequestId: int,
accountKey: str, error: XtError) -> None:
```

- 释义：暂停恢复任务回调
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|---------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号 key | string |
| error | 错误信息 | XtError |

4.2.8. 指令改参的回调函数

```
def onModifyAlgoCommands(self, nRequestId: int, orderID: int, strRemark: str,
error: XtError) -> None:
```

- 释义： 指令改参的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|---------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| orderID | 指令号 | int |
| strRemark | 下单时填写的投资备注 | string |
| error | 错误信息 | XtError |

4.3. 非交易接口回调

4.3.1. 请求账号资金的回调函数

```
def onReqAccountDetail(self, accountID: str, nRequestId: int, data: CAccountDetail,
isLast: bool, error: XtError) -> None:
```

- 释义： 请求账号资金的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号资金数据 | CAccountDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.2. 请求账号资金的回调函数

```
def onBatchReqAccountDetail(self, accountID: str, nRequestId: int,  
data: list[CAccountDetail], isLast: bool, error: XtError) -> None:
```

- 释义：请求账号资金的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号资金数据 | vector< CAccountDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.3. 请求账号资金的回调函数

```
def onReqAccountDetailWithAccKey(self, accountID: str, nRequestId: int,  
accountKey: str, data: CAccountDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号资金的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号资金数据 | CAccountDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |

| 参数 | 参数释义 | 参数类型 |
|-------|------|---------|
| error | 错误信息 | XtError |

4.3.4. 请求账号资金的回调函数

```
def onBatchReqAccountDetailWithAccKey(self, accountID: str, nRequestId:
    int, accountKey: str, data: list[CAccountDetail],
    isLast: bool, error: XtError) -> None:
```

- 释义：请求账号资金的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号资金数据 | vector<CAccountDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.5. 请求账号委托明细的回调函数

```
def onReqOrderDetail(self, accountID: str, nRequestId: int,
    data: COrderDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号委托明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|--------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |

| 参数 | 参数释义 | 参数类型 |
|--------|--------------------------------------|--------------|
| data | 账号委托明细数据 | COrderDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.6. 请求账号委托明细的回调函数

```
def onBatchReqOrderDetail(self, accountID: str, nRequestId: int,
    data: list[COrderDetail], isLast: bool, error: XtError) -> None:
```

- 释义：请求账号委托明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号委托明细数据 | vector<COrderDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.7. 请求账号委托明细的回调函数

```
def onReqOrderDetailWithAccKey(self, accountID: str, nRequestId: int,
    accountKey: str, ata: COrderDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号委托明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|-----------|------|--------|
| accountID | 账号ID | string |

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号委托明细数据 | COrderDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.8. 请求账号委托明细的回调函数

```
def onBatchReqOrderDetailWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: list[COrderDetail], isLast: bool, error: XtError) -> None:
```

- 释义：请求账号委托明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号委托明细数据 | vector<COrderDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.9. 请求账号成交明细的回调函数

```
def onReqDealDetail(self, accountID: str, nRequestId: int, data:
CDealDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号成交明细的回调函数

- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号委托明细数据 | CDealDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.10. 请求账号成交明细的回调函数

```
def onBatchReqDealDetail(self, accountID: str, nRequestId: int,
    data: list[CDealDetail], isLast: bool, error: XtError) -> None:
```

- 释义：请求账号成交明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|---------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号委托明细数据 | vector<CDealDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.11. 请求账号成交明细的回调函数

```
def onReqDealDetailWithAccKey(self, accountID: str, nRequestId: int,
    accountKey: str, data: CDealDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号成交明细的回调函数

- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号委托明细数据 | CDealDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.12. 请求账号成交明细的回调函数

```
def onBatchReqDealDetailWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: list[CDealDetail], isLast: bool, error: XtError) -> None:
```

- 释义：请求账号成交明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|---------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号委托明细数据 | vector<CDealDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.13. 请求账号持仓明细的回调函数

```
def onReqPositionDetail(self, accountID: str, nRequestId: int,  
data: CPositionDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号持仓明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号持仓明细数据 | CPositionDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.14. 请求账号持仓明细的回调函数

```
def onBatchReqPositionDetail(self, accountID: str, nRequestId: int,  
data: list[CPositionDetail], isLast: bool, error: XtError) -> None:
```

- 释义：请求账号持仓明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号持仓明细数据 | vector< CPositionDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.15. 请求账号持仓明细的回调函数

```
def onReqPositionDetailWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: CPositionDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号持仓明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号持仓明细数据 | CPositionDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.16. 请求账号持仓明细的回调函数

```
def onBatchReqPositionDetailWithAccKey(self, accountID: str, nRequestId:
int, accountKey: str, data: list[CPositionDetail], isLast: bool,
error: XtError) -> None:
```

- 释义：请求账号持仓明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|-------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号持仓明细数据 | vector<CPositionDetail> |

| 参数 | 参数释义 | 参数类型 |
|--------|--------------------------------------|---------|
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.17. 请求账号持仓统计的回调函数

```
def onReqPositionStatics(self, accountID: str, nRequestId: int, data: CPositionStatics,
    isLast: bool, error: XtError) -> None:
```

- 释义：请求账号持仓统计的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号持仓统计数据 | CPositionStatics |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.18. 请求账号持仓统计的回调函数

```
def onBatchReqPositionStatics(self, accountID: str, nRequestId: int,
    data: list[CPositionStatics], isLast: bool, error: XtError) -> None:
```

- 释义：请求账号持仓统计的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|--------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |

| 参数 | 参数释义 | 参数类型 |
|--------|--------------------------------------|--------------------------|
| data | 账号持仓统计数据 | vector<CPositionStatics> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.19. 请求账号持仓统计的回调函数

```
def onReqPositionStaticsWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: CPositionStatics, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号持仓统计的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号持仓统计数据 | CPositionStatics |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.20. 请求账号持仓统计的回调函数

```
def onBatchReqPositionStaticsWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: list[CPositionStatics], isLast: bool, error: XtError) -> None:
```

- 释义：请求账号持仓统计的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|------------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号持仓统计数据 | vector< CPositionStatics> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.21. 请求账号日初持仓统计的回调函数

```
def onReqInitialPositionStatics(self, accountID: str, nRequestId: int,
    accountKey: str, data: list[CPositionStatics], isLast: bool, error:
    XtError) -> None:
```

- 释义：请求账号日初持仓统计的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|------------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号日初持仓统计数据 | vector< CPositionStatics> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.26. 请求两融账号标的的回调函数

```
def onReqStksubjects(self, accountID: str, nRequestId: int,
data: CStkSubjects, isLast: bool, error: XtError) -> None:
```

- 释义：请求两融账号标的的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 两融账号标的的数据 | CStkSubjects |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.27. 请求两融账号标的的回调函数

```
def onBatchReqStksubjects(self, accountID: str, nRequestId: int,
data: list[CStkSubjects], isLast: bool, error: XtError) -> None:
```

- 释义：请求两融账号标的的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 两融账号标的的数据 | vector<CStkSubjects> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.28. 请求两融账号标的的回调函数

```
def onReqStksubjectsWithAccKey(self, accountID: str, nRequestId: int,  
    accountKey: str, data: CStkSubjects, isLast: bool, error: XtError) -> None:
```

- 释义：请求两融账号标的的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 两融账号标的的数据 | CStkSubjects |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.29. 请求两融账号标的的回调函数

```
def onBatchReqStksubjectsWithAccKey(self, accountID: str, nRequestId: int,  
    accountKey: str, data: list[CStkSubjects], isLast: bool, error: XtError) -> None:
```

- 释义：请求两融账号标的的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-------------------------|--------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 两融账号标的的数据 | vector< CStkSubjects> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是 | bool |

| 参数 | 参数释义 | 参数类型 |
|-------|---------------|---------|
| | 否是一次请求的最后一次回调 | |
| error | 错误信息 | XtError |

4.3.30. 请求两融账号负债的回调函数

```
def onReqStkcompacts(self, accountID: str, nRequestId: int, data:
    CStkCompacts, isLast: bool, error: XtError) -> None:
```

- 释义：请求两融账号负债的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 两融账号标的的数据 | CStkSubjects |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.31. 请求两融账号负债的回调函数

```
def onBatchReqStkcompacts(self, accountID: str, nRequestId: int,
    data: list[CStkCompacts], isLast: bool, error: XtError) -> None:
```

- 释义：请求两融账号负债的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|---------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 两融账号标的的数据 | vector< |

| 参数 | 参数释义 | 参数类型 |
|--------|--------------------------------------|---------------|
| | | CStkSubjects> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.32. 请求两融账号负债的回调函数

```
def onReqStkcompactsWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: CStkCompacts, isLast: bool, error: XtError) -> None:
```

- 释义：请求两融账号负债的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 两融账号标的的数据 | CStkSubjects |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.33. 请求两融账号负债的回调函数

```
def onBatchReqStkcompactsWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: list[CStkCompacts], isLast: bool, error: XtError) -> None:
```

- 释义：请求两融账号负债的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 两融账号标的的数据 | vector<CStkSubjects> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.34. 请求期权账号备兑持仓的回调函数

```
def onReqCoveredStockPosition(self, accountID: str, nRequestId: int,
data: CCoveredStockPosition, isLast: bool, error: XtError) -> None:
```

- 释义：请求期权账号备兑持仓的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 期权账号备兑持仓数据 | CCoveredStockPosition |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.35. 请求期权账号备兑持仓的回调函数

```
def onBatchReqCoveredStockPosition(self, accountID: str, nRequestId: int,
data: list[CCoveredStockPosition], isLast: bool, error: XtError) -> None:
```

- 释义：请求期权账号备兑持仓的回调函数

- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 期权账号备兑持仓数据 | vector< CCoveredStockPosition> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.36. 请求期权账号备兑持仓的回调函数

```
def onReqCoveredStockPositionWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: CCoveredStockPosition, isLast: bool, error: XtError) -> None:
```

- 释义：请求期权账号备兑持仓的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 期权账号备兑持仓数据 | CCoveredStockPosition |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.37. 请求两融账号负债的回调函数

```
def onBatchReqCoveredStockPositionWithAccKey(self, accountID: str,
nRequestId: int, accountKey: str, data: list[CCoveredStockPosition],
isLast: bool, error: XtError) -> None:
```

- 释义：请求期权账号备兑持仓的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 期权账号备兑持仓数据 | vector< CCoveredStockPosition> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.38. 请求产品信息的回调函数

```
def onReqProductData(self, nRequestId: int, data: CProductData, isLast:
bool, error: XtError) -> None:
```

- 释义：请求产品信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 产品信息数据 | CProductData |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.39. 请求产品信息的回调函数

```
def onBatchReqProductData(self, nRequestId: int, data: list[CProductData],
isLast: bool, error: XtError) -> None:
```

- 释义：请求产品信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 产品信息数据 | vector<CProductData> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.40. 请求合约数据的回调函数

```
def onReqCInstrumentDetail(self, accountID: str, nRequestId: int,
data: CInstrumentDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求合约数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 合约数据 | CInstrumentDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.41. 请求合约数据的回调函数

```
def onBatchReqCInstrumentDetail(self, accountID: str, nRequestId: int,
data: list[CInstrumentDetail], isLast: bool, error: XtError) -> None:
```

- 释义：请求合约数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--|-------------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 合约数据 | vector< CInstrumentDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数， 标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.42. 请求合约数据的回调函数

```
def onReqCInstrumentDetailWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: CInstrumentDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求合约数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--|-------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 合约数据 | CInstrumentDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标 记是否是一次请求的最后一次回调 | bool |

| 参数 | 参数释义 | 参数类型 |
|-------|------|---------|
| error | 错误信息 | XtError |

4.3.43. 请求合约数据的回调函数

```
def onBatchReqCInstrumentDetailWithAccKey(self, accountID: str,
nRequestId: int, accountKey: str, data: list[CInstrumentDetail],
isLast: bool, error: XtError) -> None:
```

- 释义：请求合约数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--|-------------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 合约数据 | vector< CInstrumentDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数， 标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.44. 请求行情数据的回调函数

```
def onReqPriceData(self, nRequestId: int, data: CPriceData,
error: XtError) -> None:
```

- 释义：请求行情数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 行情数据 | CPriceData |

| 参数 | 参数释义 | 参数类型 |
|-------|------|---------|
| error | 错误信息 | XtError |

4.3.45. 请求期权账号组合持仓数据的回调函数

```
def onReqStkOptCombPositionDetail(self, accountID: str, nRequestId: int,
data: CStockOptionCombPositionDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求期权账号组合持仓数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 期权账号组合持仓数据 | CStockOptionCombPositionDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.46. 请求期权账号组合持仓数据的回调函数

```
def onBatchReqStkOptCombPositionDetail(self, accountID: str, nRequestId: int,
data: list[CStockOptionCombPositionDetail], isLast: bool, error: XtError) -> None:
```

- 释义：请求期权账号组合持仓数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|--|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 期权账号组合持仓数据 | vector< CStockOptionCombPositionDetail> |

| 参数 | 参数释义 | 参数类型 |
|--------|--------------------------------------|---------|
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.47. 请求期权账号组合持仓数据的回调函数

```
def onReqStkOptCombPositionDetailWithAccKey(self, accountID: str,
nRequestId: int, accountKey: str, data: CStockOptionCombPositionDetail,
isLast: bool, error: XtError) -> None:
```

- 释义：请求期权账号组合持仓数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 期权账号组合持仓数据 | CStockOptionCombPositionDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.48. 请求期权账号组合持仓数据的回调函数

```
def onBatchReqStkOptCombPositionDetailWithAccKey(self, accountID: str,
nRequestId: int, accountKey: str, data: list[CStockOptionCombPositionDetail],
isLast: bool, error: XtError) -> None:
```

- 释义：请求期权账号组合持仓数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 期权账号组合持仓数据 | vector< CStockOptionCombPositionDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.49. 行情订阅的回调函数

```
def onSubscribeQuote(self, nRequestId: int, data: CSubscribeData,
error: XtError) -> None:
```

- 释义：行情订阅的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|----------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 行情订阅数据 | CSubscribeData |
| error | 错误信息 | XtError |

4.3.50. 行情退订的回调函数

```
def onUnSubscribeQuote(self, nRequestId: int, data: CSubscribeData,
error: XtError) -> None:
```

- 释义：行情退订的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|---------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 行情订阅数据 | CSubscribData |
| error | 错误信息 | XtError |

4.3.51. 请求港股账号汇率数据的回调函数

```
def onReqReferenceRate(self, accountID: str, nRequestId: int,
data: CReferenceRate, isLast: bool, error: XtError) -> None:
```

- 释义：请求港股账号汇率数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 港股账号汇率数据 | CReferenceRate |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.52. 请求港股账号汇率数据的回调函数

```
def onBatchReqReferenceRate(self, accountID: str, nRequestId: int,
data: list[CReferenceRate], isLast: bool, error: XtError) -> None:
```

- 释义：请求港股账号汇率数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|--------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |

| 参数 | 参数释义 | 参数类型 |
|--------|--------------------------------------|------------------------|
| data | 港股账号汇率数据 | vector<CReferenceRate> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.53. 请求港股账号汇率数据的回调函数

```
def onReqReferenceRateWithAccKey(self, accountID: str, nRequestId: int,
    accountKey: str, data: CReferenceRate, isLast: bool, error: XtError) -> None:
```

- 释义：请求港股账号汇率数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 港股账号汇率数据 | CReferenceRate |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.54. 请求港股账号汇率数据的回调函数

```
def onBatchReqReferenceRateWithAccKey(self, accountID: str, nRequestId: int,
    accountKey: str, data: list[CReferenceRate], isLast: bool, error: XtError) -> None:
```

- 释义：请求港股账号汇率数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 港股账号汇率数据 | vector<CReferenceRate> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.55. 请求两融账号两融综合资金数据的回调函数

```
def onReqCreditDetail(self, accountID: str, nRequestId: int, data:
    CCreditDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求两融账号两融综合资金数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|---------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 两融账号两融综合资金数据 | CCreditDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.56. 请求两融账号两融综合资金数据的回调函数

```
def onBatchReqCreditDetail(self, accountID: str, nRequestId: int,
    data: list[CCreditDetail], isLast: bool, error: XtError) -> None:
```

- 释义：请求两融账号两融综合资金数据的回调函数

- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 两融账号两融综合资金数据 | vector<CCreditDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.57. 请求两融账号两融综合资金数据的回调函数

```
def onReqCreditDetailWithAccKey(self, accountID: str, nRequestId: int,
    accountKey: str, data: CCreditDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求两融账号两融综合资金数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|---------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 两融账号两融综合资金数据 | CCreditDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.58. 请求两融账号两融综合资金数据的回调函数

```
def onBatchReqCreditDetailWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: list[CCreditDetail], isLast: bool, error: XtError) -> None:
```

- 释义：请求两融账号两融综合资金数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|---------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 两融账号两融综合资金数据 | vector< CCreditDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.59. 用户自己的定时器回调 已废弃

```
def onCustomTimer(self) -> None:
```

- 释义：用户自己的定时器回调
- note：基本废弃，建议用下面onNamedTimer
- 参数

| 参数 | 参数释义 | 参数类型 |
|----|------|------|
|----|------|------|

4.3.60. 用户自己的定时器回调

```
def onNamedTimer(self, timerName: str) -> None:
```

- 释义：用户自己的定时器回调

- 参数

| 参数 | 参数释义 | 参数类型 |
|-----------|-------|--------|
| timerName | 定时器名称 | string |

4.3.61. 请求新股额度数据的回调函数

```
def onReqSubscribeInfo(self, accountID: str, nRequestId: int,
data: CSubscribeInfo, isLast: bool, error: XtError) -> None:
```

- 释义：请求新股额度数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 新股信息数据 | CSubscribeInfo |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.62. 请求新股额度数据的回调函数

```
def onBatchReqSubscribeInfo(self, accountID: str, nRequestId: int,
data: list[CSubscribeInfo], isLast: bool, error: XtError) -> None:
```

- 释义：请求新股额度数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|---------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 新股信息数据 | vector< |

| 参数 | 参数释义 | 参数类型 |
|--------|--------------------------------------|-----------------|
| | | CSubscribeInfo> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.63. 请求新股额度数据的回调函数

```
def onReqSubscribeInfoWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: CSubscribeInfo, isLast: bool, error: XtError) -> None:
```

- 释义：请求新股额度数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 新股信息数据 | CSubscribeInfo |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.64. 请求新股额度数据的回调函数

```
def onBatchReqSubscribeInfoWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: list[CSubscribeInfo], isLast: bool, error: XtError) -> None:
```

- 释义：请求新股额度数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 新股信息数据 | vector<CSubscribeInfo> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.65. 请求未了结负债信息的回调函数

```
def onReqStkUnCloseCompact(self, accountID: str, nRequestId: int,
data: CStkUnClosedCompacts, isLast: bool, error: XtError) -> None:
```

- 释义：请求未了结负债信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 未了结负债信息数据 | CStkUnClosedCompacts |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.66. 请求未了结负债信息的回调函数

```
def onBatchReqStkUnCloseCompact(self, accountID: str, nRequestId: int,
data: list[CStkUnClosedCompacts], isLast: bool, error: XtError) -> None:
```

- 释义：请求未了结负债信息的回调函数

- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 未了结负债信息数据 | vector< CStkUnClosedCompacts> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.67. 请求未了结负债信息的回调函数

```
def onReqStkUnCloseCompactWithAccKey(self, accountID: str, nRequestId:
int, accountKey: str, data: CStkUnClosedCompacts, isLast: bool, error: XtError)
-> None:
```

- 释义： 请求未了结负债信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 未了结负债信息数据 | CStkUnClosedCompacts |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.68. 请求未了结负债信息的回调函数

```
def onBatchReqStkUnCloseCompactWithAccKey(self, accountID: str,
nRequestId: int, accountKey: str, data: list[CStkUnClosedCompacts],
isLast: bool, error: XtError) -> None:
```

- 释义：请求未了结负债信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 未了结负债信息数据 | vector< CStkUnClosedCompacts> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.69. 请求已了结负债信息的回调函数

```
def onReqStkClosedCompact(self, accountID: str, nRequestId: int,
data: CStkClosedCompacts, isLast: bool, error: XtError) -> None:
```

- 释义：请求已了结负债信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------|--------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 已了结负债信息数据 | CStkClosedCompacts |
| isLast | 请求数据可能有多条，需要多次回调该函 | bool |

| 参数 | 参数释义 | 参数类型 |
|-------|--------------------|---------|
| | 数，标记是否是一次请求的最后一次回调 | |
| error | 错误信息 | XtError |

4.3.70. 请求已了结负债信息的回调函数

```
def onBatchReqStkClosedCompact(self, accountID: str, nRequestId: int,
data: list[CStkClosedCompacts], isLast: bool, error: XtError) -> None:
```

- 释义：请求已了结负债信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 已了结负债信息数据 | vector< CStkClosedCompacts> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.71. 请求已了结负债信息的回调函数

```
def onReqStkClosedCompactWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: CStkClosedCompacts, isLast: bool, error: XtError) -> None:
```

- 释义：请求已了结负债信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|--------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------------|
| accountKey | 账号Key | string |
| data | 已了结负债信息数据 | CStkClosedCompacts |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.72. 请求已了结负债信息的回调函数

```
def onBatchReqStkClosedCompactWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: list[CStkClosedCompacts], isLast: bool, error: XtError) -> None:
```

- 释义：请求已了结负债信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 已了结负债信息数据 | vector< CStkClosedCompacts> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.73. 获取用户下所有账号key的回调函数

```
def onReqAccountKey(self, nRequestId: int, data: CAccountKey, isLast:
bool, error: XtError) -> None:
```

- 释义：获取用户下所有账号key的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | accountKey数据 | CAccountKey |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.74. 获取用户下所有账号key的回调函数

```
def onBatchReqAccountKey(self, nRequestId: int, data: list[CAccountKey],
    isLast: bool, error: XtError) -> None:
```

- 释义：获取用户下所有账号key的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | accountKey数据 | vector< CAccountKey> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.75. 根据委托号请求账号成交明细信息的回调函数

```
def onReqDealDetailBySysID(self, accountID: str, nRequestId: int,
    orderSysId: str, exchangeId: str, data: CDealDetail, isLast: bool,
    error: XtError) -> None:
```

- 释义：根据委托号请求账号成交明细信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| orderSyeld | 委托号 | string |
| exchangeId | 委托所属市场 | string |
| data | 账号成交明细数据 | CDealDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.76. 根据委托号请求账号成交明细信息的回调函数

```
def onBatchReqDealDetailBySysID(self, accountID: str, nRequestId: int,
    orderSyeld: str, exchangeId: str, data: list[CDealDetail], isLast: bool,
    error: XtError) -> None:
```

- 释义：根据委托号请求账号成交明细信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|---------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| orderSyeld | 委托号 | string |
| exchangeId | 委托所属市场 | string |
| data | 账号成交明细数据 | vector<CDealDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.77. 根据委托号请求账号成交明细信息的回调函数

```
def onReqDealDetailBySysIDWithAccKey(self, accountID: str, nRequestId:
    int, accountKey: str, orderSyeId: str, exchangeId: str, data:
    CDealDetail, isLast: bool, error: XtError) -> None:
```

- 释义： 根据委托号请求账号成交明细信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| orderSyeld | 委托号 | string |
| exchangeld | 委托所属市场 | string |
| data | 账号成交明细数据 | CDealDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.78. 根据委托号请求账号成交明细信息的回调函数

```
def onBatchReqDealDetailBySysIDWithAccKey(self, accountID: str,
nRequestId: int, accountKey: str, orderSyeId: str, exchangeId: str,
    data: list[CDealDetail], isLast: bool, error: XtError) -> None:
```

- 释义： 根据委托号请求账号成交明细信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|--------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|---------------------|
| accountKey | 账号Key | string |
| orderSyeld | 委托号 | string |
| exchangeId | 委托所属市场 | string |
| data | 账号成交明细数据 | vector<CDealDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.79. 请求账号结算单信息的回调函数

```
def onReqDeliveryDetail(self, accountID: str, nRequestId: int,
data: CDeliveryDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号结算单信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号结算单数据 | CDeliveryDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.80. 请求账号结算单信息的回调函数

```
def onBatchReqDeliveryDetail(self, accountID: str, nRequestId: int,
data: list[CDeliveryDetail], isLast: bool, error: XtError) -> None:
```

- 释义：请求账号结算单信息的回调函数

- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号结算单数据 | vector<CDeliveryDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.81. 请求账号结算单信息的回调函数

```
def onReqDeliveryDetailWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: CDeliveryDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号结算单信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号结算单数据 | CDeliveryDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.82. 请求账号结算单信息的回调函数

```
def onBatchReqDeliveryDetailWithAccKey(self, accountID: str, nRequestId:
    int, accountKey: str, data: list[CDeliveryDetail], isLast: bool,
    error: XtError) -> None:
```

- 释义：请求账号结算单信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号结算单数据 | vector<CDeliveryDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.83. 请求账号合约行情信息的回调函数

```
def onReqSingleInstrumentInfo(self, nRequestId: int, data:
    CInstrumentInfo, error: XtError) -> None:
```

- 释义：请求账号合约行情信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|-----------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 合约数据 | CInstrumentInfo |
| error | 错误信息 | XtError |

4.3.84. 请求账号可下单量的回调函数

```
def onReqOpVolume(self, accountID: str, nRequestId: int, dataKey: str,  
nVolume: int, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号可下单量的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|---------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| dataKey | 数据标识，市场加合约 | string |
| nVolume | 可下单量 | int |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.85. 请求账号可下单量的回调函数

```
def onReqOpVolumeWithAccKey(self, accountID: str, nRequestId: int,  
accountKey: str, dataKey: str, nVolume: int, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号可下单量的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|--------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| dataKey | 数据标识，市场加合约 | string |
| nVolume | 可下单量 | int |

| 参数 | 参数释义 | 参数类型 |
|--------|--------------------------------------|---------|
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.86. 请求账号可下单量的回调函数

```
def onReqCanOrderVolume(self, accountID: str, nRequestId: int, accountKey: str, market: str, instrument: str, nVolume: int, error: XtError) -> None:
```

- 释义：请求账号可下单量的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|---------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| market | 市场代码 | string |
| instrument | 合约代码 | string |
| nVolume | 可下单量 | int |
| error | 错误信息 | XtError |

4.3.87. 请求两融账号融券可融数量的回调函数

```
def onReqCreditSloCode(self, accountID: str, nRequestId: int, data: CCreditSloCode, isLast: bool, error: XtError) -> None:
```

- 释义：请求两融账号融券可融数量的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|-----------|------|--------|
| accountID | 账号ID | string |

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 融券可融数量数据 | CCreditSloCode |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.88. 请求两融账号融券可融数量的回调函数

```
def onBatchReqCreditSloCode(self, accountID: str, nRequestId: int,
data: list[CCreditSloCode], isLast: bool, error: XtError) -> None:
```

- 释义：请求两融账号融券可融数量的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 融券可融数量数据 | vector<CCreditSloCode> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.89. 请求两融账号融券可融数量的回调函数

```
def onBatchReqCreditSloCodeWithAccKey(self, accountID: str, nRequestId:
int, accountKey: str, data: list[CCreditSloCode], isLast: bool, error: XtError)
-> None:
```

- 释义：请求两融账号融券可融数量的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 融券可融数量数据 | CCreditSloCode |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.90. 请求两融账号融券可融数量的回调函数

```
def onBatchReqSubscribeInfoWithAccKey(self, accountID: str, nRequestId: int,
    accountKey: str, data: list[CSubscribeInfo], isLast: bool, error: XtError) -> None:
```

- 释义：请求两融账号融券可融数量的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 融券可融数量数据 | vector< CCreditSloCode> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.91. 请求两融账号融资融券标的的回调函数

```
def onReqCreditSubjects(self, accountID: str, nRequestId: int,
    data: CCreditSubjects, isLast: bool, error: XtError) -> None:
```


- 释义： 请求两融账号融资融券标的的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 融资融券标的的数据 | CCreditSubjects |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.92. 请求两融账号融资融券标的的回调函数

```
def onBatchReqCreditSubjects(self, accountID: str, nRequestId: int,
    data: list[CCreditSubjects], isLast: bool, error: XtError) -> None:
```

- 释义： 请求两融账号融资融券标的的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 融资融券标的的数据 | vector< CCreditSubjects> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.93. 请求两融账号融资融券标的的回调函数

```
def onReqCreditSubjectsWithAccKey(self, accountID: str, nRequestId: int,
    accountKey: str, data: CCreditSubjects, isLast: bool, error: XtError) -> None:
```

- 释义： 请求两融账号融资融券标的的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 融资融券标的的数据 | CCreditSubjects |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.94. 请求两融账号融资融券标的的回调函数

```
def onBatchReqCreditSubjectsWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: list[CCreditSubjects], isLast: bool, error: XtError) -> None:
```

- 释义： 请求两融账号融资融券标的的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 融资融券标的的数据 | vector<CCreditSubjects> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.95. 请求两融账号担保标的的回调函数

```
def onReqCreditAssure(self, accountID: str, nRequestId: int,  
data: CCreditAssure, isLast: bool, error: XtError) -> None:
```

- 释义： 请求两融账号担保标的的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|---------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 担保标的的数据 | CCreditAssure |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.96. 请求两融账号担保标的的回调函数

```
def onBatchReqCreditAssure(self, accountID: str, nRequestId: int,  
data: list[CCreditAssure], isLast: bool, error: XtError) -> None:
```

- 释义： 请求两融账号担保标的的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|---------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 担保标的的数据 | vector< CCreditAssure> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.97. 请求两融账号担保标的的回调函数

```
def onReqCreditAssureWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: CCreditAssure, isLast: bool, error: XtError) -> None:
```

- 释义： 请求两融账号担保标的的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|---------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 担保标的的数据 | CCreditAssure |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.98. 请求两融账号担保标的的回调函数

```
def onBatchReqCreditAssureWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: list[CCreditAssure], isLast: bool, error: XtError) -> None:
```

- 释义： 请求两融账号担保标的的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|------------------------|---------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 担保标的的数据 | vector< CCreditAssure> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记 | bool |

| 参数 | 参数释义 | 参数类型 |
|-------|----------------|---------|
| | 是否是一次请求的最后一次回调 | |
| error | 错误信息 | XtError |

4.3.99. 请求账号银证转账银行信息的回调函数

```
def onReqTransferBank(self, accountID: str, nRequestId: int,
    data: CQueryBankInfo, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号银证转账银行信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号银证转账银行信息数据 | CQueryBankInfo |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.100. 请求账号银证转账银行信息的回调函数

```
def onBatchReqTransferBank(self, accountID: str, nRequestId: int,
    data: list[CQueryBankInfo], isLast: bool, error: XtError) -> None:
```

- 释义：请求账号银证转账银行信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|--------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |

| 参数 | 参数释义 | 参数类型 |
|--------|--------------------------------------|------------------------|
| data | 账号银证转账银行信息数据 | vector<CQueryBankInfo> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.101. 请求账号银证转账银行信息的回调函数

```
def onReqTransferBankWithAccKey(self, accountID: str, nRequestId: int,
    accountKey: str, data: CQueryBankInfo, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号银证转账银行信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号银证转账银行信息数据 | CQueryBankInfo |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.102. 请求账号银证转账银行信息的回调函数

```
def onBatchReqTransferBankWithAccKey(self, accountID: str, nRequestId:
    int, accountKey: str, data: list[CQueryBankInfo], isLast: bool, error: XtError)
    -> None:
```

- 释义：请求账号银证转账银行信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号银证转账银行信息数据 | vector< CQueryBankInfo> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.103. 请求账号银证转账银行流水的回调函数

```
def onReqTransferSerial(self, accountID: str, nRequestId: int,
data: CTransferSerial, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号银证转账银行流水的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号银证转账银行流水数据 | CTransferSerial |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.104. 请求账号银证转账银行流水的回调函数

```
def onBatchReqTransferSerial(self, accountID: str, nRequestId: int,
data: list[CTransferSerial], isLast: bool, error: XtError) -> None:
```

- 释义：请求账号银证转账银行流水的回调函数

- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号银证转账银行流水数据 | vector<CTransferSerial> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.105. 请求账号银证转账银行流水的回调函数

```
def onReqTransferSerialWithAccKey(self, accountID: str, nRequestId: int,
    accountKey: str, data: CTransferSerial, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号银证转账银行流水的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号银证转账银行流水数据 | CTransferSerial |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.106. 请求账号银证转账银行流水的回调函数

```
def onBatchReqTransferSerialWithAccKey(self, accountID: str, nRequestId:
    int, accountKey: str, data: list[CTransferSerial], isLast: bool, error: XtError)
    -> None:
```

- 释义：请求账号银证转账银行流水的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号银证转账银行流水数据 | vector<CTransferSerial> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.107. 请求账号银证转账银行余额的回调函数

```
def onReqBankAmount(self, accountID: str, nRequestId: int,
    data: CQueryBankAmount, error: XtError) -> None:
```

- 释义：请求账号银证转账银行余额的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号银证转账银行流水数据 | CQueryBankAmount |
| error | 错误信息 | XtError |

4.3.108. 请求账号银证转账银行余额的回调函数

```
def onReqBankAmountWithAccKey(self, accountID: str, nRequestId: int,
    accountKey: str, data: CQueryBankAmount, error: XtError) -> None:
```

- 释义： 请求账号银证转账银行余额的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号key | string |
| data | 账号银证转账银行流水数据 | CQueryBankAmount |
| error | 错误信息 | XtError |

4.3.109. 银证转账的回调函数

```
def onTransfer(self, nRequestId: int, error: XtError) -> None:
```

- 释义： 银证转账的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|---------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| error | 错误信息 | XtError |

4.3.110. 按市场请求合约信息的回调函数

```
def onReqInstrumentInfoByMarket(self, nRequestId: int,
    data: CInstrumentInfo, isLast: bool, error: XtError) -> None:
```

- 释义： 按市场请求合约信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 合约信息数据 | CInstrumentInfo |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.111. 按市场请求合约信息的回调函数

```
def onBatchReqInstrumentInfoByMarket(self, nRequestId: int,
data: list[CInstrumentInfo], isLast: bool, error: XtError) -> None:
```

- 释义：按市场请求合约信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 合约信息数据 | vector< CInstrumentInfo> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.112. 按市场请求合约信息的回调函数

```
def onReqInstrumentInfoByMarketWithMkt(self, nRequestId: int, exchangeId: str,
data: CInstrumentInfo, isLast: bool, error: XtError) -> None:
```

- 释义：按市场请求合约信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|------|
| nRequestId | 客户自己维护的请求顺序ID | int |

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------|
| exchangeId | 市场代码 | string |
| data | 合约信息数据 | CInstrumentInfo |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.113. 按市场请求合约信息的回调函数

```
def onBatchReqInstrumentInfoByMarketWithMkt(self, nRequestId: int,
exchangeId: str, data: list[CInstrumentInfo], isLast: bool, error: XtError) -> None:
```

- 释义：按市场请求合约信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| exchangeId | 市场代码 | string |
| data | 合约信息数据 | vector< CInstrumentInfo> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.114. 请求账号可撤单委托明细的回调函数

```
def onReqCanCancelOrderDetail(self, accountID: str, nRequestId: int,
data: COrderDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号可撤单委托明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号委托明细数据 | COrderDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.115. 请求账号可撤单委托明细的回调函数

```
def onBatchReqCanCancelOrderDetail(self, accountID: str, nRequestId: int,
data: list[COrderDetail], isLast: bool, error: XtError) -> None:
```

- 释义：请求账号可撤单委托明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号委托明细数据 | vector<COrderDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.116. 请求账号可撤单委托明细的回调函数

```
def onReqCanCancelOrderDetailWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: COrderDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号可撤单委托明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号委托明细数据 | COrderDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.117. 请求账号可撤单委托明细的回调函数

```
def onBatchReqCanCancelOrderDetailWithAccKey(self, accountID: str,
nRequestId: int, accountKey: str, data: list[COrderDetail], isLast: bool,
error: XtError) -> None:
```

- 释义：请求账号可撤单委托明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号委托明细数据 | vector< COrderDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.118. 请求所有下单信息的回调函数

```
def onReqCommandsInfo(self, nRequestId: int, data: COrderInfo, isLast: bool,
    error: XtError) -> None:
```

- 释义：请求所有下单信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 下单信息数据 | COrderInfo |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.119. 请求所有下单信息的回调函数

```
def onBatchReqCommandsInfo(self, nRequestId: int, data: list[COrderInfo],
    isLast: bool, error: XtError) -> None:
```

- 释义：请求所有下单信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 下单信息数据 | vector<COrderInfo> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.120. 资金划拨的回调函数

```
def onFundTransfer(self, nRequestId: int, error: XtError) -> None:
```

- 释义： 资金划拨的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|---------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| error | 错误信息 | XtError |

4.3.121. 股份划拨的回调函数

```
def onSecuTransfer(self, nRequestId: int, error: XtError) -> None:
```

- 释义： 股份划拨的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|---------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| error | 错误信息 | XtError |

4.3.122. 请求账号账号普通柜台资金的回调函数

```
def onReqComFund(self, accountID: str, nRequestId: int, data: CStockComFund, isLast: bool, error: XtError) -> None:
```

- 释义： 请求账号账号普通柜台资金的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------|---------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号资金划拨普通柜台资金数据 | CStockComFund |

| 参数 | 参数释义 | 参数类型 |
|--------|--------------------------------------|---------|
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.123. 请求账号账号普通柜台资金的回调函数

```
def onBatchReqComFund(self, accountID: str, nRequestId: int,
data: list[CStockComFund], isLast: bool, error: XtError) -> None:
```

- 释义：请求账号账号普通柜台资金的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|---------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号资金划拨普通柜台资金数据 | vector< CStockComFund> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.124. 请求账号账号普通柜台资金的回调函数

```
def onReqComFundWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: CStockComFund, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号账号普通柜台资金的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|--------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|---------------|
| accountKey | 账号Key | string |
| data | 账号资金划拨普通柜台资金数据 | CStockComFund |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.125. 请求账号账号普通柜台资金的回调函数

```
def onBatchReqComFundWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: list[CStockComFund], isLast: bool, error: XtError) -> None:
```

- 释义：请求账号账号普通柜台资金的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|---------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号资金划拨普通柜台资金数据 | vector< CStockComFund> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.126. 请求账号普通柜台持仓的回调函数

```
def onReqComPosition(self, accountID: str, nRequestId: int,
data: CStockComPosition, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号普通柜台持仓的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号股份划拨普通柜台持仓数据 | CStockComPosition |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.127. 请求账号普通柜台持仓的回调函数

```
def onBatchReqComPosition(self, accountID: str, nRequestId: int,
data: list[CStockComPosition], isLast: bool, error: XtError) -> None:
```

- 释义：请求账号普通柜台持仓的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|---------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号股份划拨普通柜台持仓数据 | vector<CStockComPosition> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.128. 请求账号普通柜台持仓的回调函数

```
def onReqComPositionWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: CStockComPosition, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号普通柜台持仓的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号股份划拨普通柜台持仓数据 | CStockComPosition |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.129. 请求账号普通柜台持仓的回调函数

```
def onBatchReqComPositionWithAccKey(self, accountID: str, nRequestId: int,
    accountKey: str, data: list[CStockComPosition], isLast: bool, error: XtError)
    -> None:
```

- 释义：请求账号普通柜台持仓的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号股份划拨普通柜台持仓数据 | vector< CStockComPosition> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.130. 请求当前交易日的回调函数

```
def onReqTradeDay(self, tradeDay: str, nRequestId: int, error: XtError) -> None:
```

- 释义： 请求当前交易日的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|---------|
| tradeDay | 当前交易日 | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| error | 错误信息 | XtError |

4.3.131. 请求账号历史委托明细的回调函数

```
def onReqHistoryOrderDetail(self, accountID: str, nRequestId: int,
data: COrderDetail, isLast: bool, error: XtError) -> None:
```

- 释义： 请求账号历史委托明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号委托明细数据 | COrderDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.132. 请求账号历史委托明细的回调函数

```
def onReqHistoryOrderDetailWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: COrderDetail, isLast: bool, error: XtError) -> None:
```

- 释义： 请求账号历史委托明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号委托明细数据 | vector<COrderDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.133. 请求账号历史委托明细的回调函数

```
def onBatchReqHistoryOrderDetailWithAccKey(self, accountID: str,
nRequestId: int, accountKey: str, data: list[COrderDetail],
isLast: bool, error: XtError) -> None:
```

- 释义：请求账号历史委托明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号委托明细数据 | COrderDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.134. 请求账号历史委托明细的回调函数

```
def onBatchReqSubscribeInfoWithAccKey(self, accountID: str, nRequestId: int,
accountKey: str, data: list[CSubscribeInfo], isLast: bool, error: XtError) -> None:
```

- 释义： 请求账号历史委托明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号委托明细数据 | vector<COrderDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.135. 请求账号历史成交明细的回调函数

```
def onReqHistoryDealDetail(self, accountID: str, nRequestId: int,
data: CDealDetail, isLast: bool, error: XtError) -> None:
```

- 释义： 请求账号历史成交明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号历史成交明细数据 | CDealDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.136. 请求账号历史成交明细的回调函数

```
def onBatchReqHistoryDealDetail(self, accountID: str, nRequestId: int,  
    data: list[CDealDetail], isLast: bool, error: XtError) -> None:
```

- 释义：请求账号历史成交明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号历史成交明细数据 | vector< CDealDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.137. 请求账号历史成交明细的回调函数

```
def onReqHistoryDealDetailWithAccKey(self, accountID: str, nRequestId: int,  
    accountKey: str, data: CDealDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号历史成交明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号历史成交明细数据 | CDealDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |

| 参数 | 参数释义 | 参数类型 |
|-------|------|---------|
| error | 错误信息 | XtError |

4.3.138. 请求账号历史成交明细的回调函数

```
def onBatchReqHistoryDealDetailWithAccKey(self, accountID: str,
nRequestId: int, accountKey: str, data: list[CDealDetail],
isLast: bool, error: XtError) -> None:
```

- 释义：请求账号历史成交明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号历史成交明细数据 | vector< CDealDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.139. 请求账号历史持仓统计的回调函数

```
def onReqHistoryPositionStatics(self, accountID: str, nRequestId: int,
data: CPositionStatics, isLast: bool, error: XtError) -> None:
```

- 释义：请求账号历史成交明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|--------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |

| 参数 | 参数释义 | 参数类型 |
|--------|--------------------------------------|------------------|
| data | 账号持仓统计数据 | CPositionStatics |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.140. 请求账号历史持仓统计的回调函数

```
def onBatchReqHistoryPositionStatics(self, accountID: str, nRequestId: int,
data: list[CPositionStatics], isLast: bool, error: XtError) -> None:
```

- 释义：请求账号历史成交明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号持仓统计数据 | vector<CPositionStatics> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.141. 请求账号历史持仓统计的回调函数

```
def onReqHistoryPositionStaticsWithAccKey(self, accountID: str,
nRequestId: int, accountKey: str, data: CPositionStatics,
isLast: bool, error: XtError) -> None:
```

- 释义：请求账号历史持仓统计的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号持仓统计数据 | CPositionStatics |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.142. 请求账号历史持仓统计的回调函数

```
def onBatchReqHistoryPositionStaticsWithAccKey(self, accountID: str,
nRequestId: int, accountKey: str, data: list[CPositionStatics],
isLast: bool, error: XtError) -> None:
```

- 释义：请求账号历史持仓统计的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|------------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 账号持仓统计数据 | vector< CPositionStatics> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.143. 请求期货账号保证金率的回调函数

```
def onReqFtAccCommissionRateDetail(self, accountID: str, nRequestId: int,
    accountKey: str, data: CCommissionRateDetail, error: XtError) -> None:
```

- 释义：请求期货账号保证金率的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|-----------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 保证金率 | CCommissionRateDetail |
| error | 错误信息 | XtError |

4.3.144. 请求期货账号手续费率的回调函数

```
def onReqFtAccMarginRateDetail(self, accountID: str, nRequestId: int,
    accountKey: str, data: CMarginRateDetail, isLast: bool, error: XtError) -> None:
```

- 释义：请求期货账号手续费率的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|-------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 手续费率 | CMarginRateDetail |
| error | 错误信息 | XtError |

4.3.145. 获取用户下所有的产品Id的回调函数

```
def onReqProductIds(self, nRequestId: int, nProductID: int, accountKey: str, isLast: bool) -> None:
```

- 释义： 获取用户下所有的产品Id的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| nProductID | 产品ID | int |
| accountKey | 账号Key | string |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |

4.3.146. 创建新投资组合的回调函数

```
def onCreatePortfolio(self, nRequestId: int, nPortfolioID: int, strRemark: str, error: XtError) -> None:
```

- 释义： 创建新投资组合的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|---------------|---------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| nPortfolioID | 投资组合编号 | int |
| strRemark | 下单时填写的投资备注 | string |
| error | 错误信息 | XtError |

4.3.147. 查询产品Id下所有的投资组合的回调函数

```
def onReqProductPortfolio(self, nProductID: int, nRequestId: int, data: CPortfolioInfo, isLast: bool, error: XtError) -> None:
```

- 释义： 查询产品Id下所有的投资组合的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------|
| nProductID | 产品ID | int |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 投资组合数据 | CPortfolioInfo |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.148. 查询产品Id下所有的投资组合的回调函数

```
def onBatchReqProductPortfolio(self, nProductID: int, nRequestId: int,
data: list[CPortfolioInfo], isLast: bool, error: XtError) -> None:
```

- 释义： 查询产品Id下所有的投资组合的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|----------------------------|
| nProductID | 产品ID | int |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 投资组合数据 | vector< CPortfolioInfo> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.149. 请求投资组合委托信息的回调函数

```
def onReqPortfolioOrder(self, nPortfolioID: int, nRequestId: int,
data: COrderDetail, isLast: bool, error: XtError) -> None:
```

- 释义： 请求投资组合委托信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|--------------------------------------|--------------|
| nPortfolioID | 投资组合ID | int |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号委托明细数据 | COrderDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.150. 请求投资组合委托信息的回调函数

```
def onBatchReqPortfolioOrder(self, nPortfolioID: int, nRequestId: int,
data: list[COrderDetail], isLast: bool, error: XtError) -> None:
```

- 释义： 请求投资组合委托信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|--------------------------------------|--------------------------|
| nPortfolioID | 投资组合ID | int |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号委托明细数据 | vector< COrderDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.151. 请求投资组合一段时间内的委托信息的回调函数

```
def onReqPortfolioMultiOrder(self, nPortfolioID: int, nRequestId: int,
data: COrderDetail, isLast: bool, error: XtError) -> None:
```

- 释义： 请求投资组合一段时间内的委托信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|--------------------------------------|--------------|
| nPortfolioID | 投资组合ID | int |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号委托明细数据 | COrderDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.152. 请求投资组合一段时间内的委托信息的回调函数

```
def onBatchReqPortfolioMultiOrder(self, nPortfolioID: int, nRequestId: int,
data: list[COrderDetail], isLast: bool, error: XtError) -> None:
```

- 释义： 请求投资组合一段时间内的委托信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|--------------------------------------|----------------------|
| nPortfolioID | 投资组合ID | int |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号委托明细数据 | vector<COrderDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.153. 请求账号成交明细的回调函数

```
def onReqPortfolioDeal(self, nPortfolioID: int, nRequestId: int,
data: CDealDetail, isLast: bool, error: XtError) -> None:
```


- 释义： 请求账号成交明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|--------------------------------------|-------------|
| nPortfolioID | 投资组合ID | int |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号成交明细数据 | CDealDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.154. 请求账号成交明细的回调函数

```
def onBatchReqPortfolioDeal(self, nPortfolioID: int, nRequestId: int,
data: list[CDealDetail], isLast: bool, error: XtError) -> None:
```

- 释义： 请求账号成交明细的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|--------------------------------------|----------------------|
| nPortfolioID | 投资组合ID | int |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号成交明细数据 | vector< CDealDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.155. 请求投资组合一段时间内的委托信息的回调函数

```
def onReqPortfolioMultiDeal(self, nPortfolioID: int, nRequestId: int,
data: CDealDetail, isLast: bool, error: XtError) -> None:
```

- 释义： 请求投资组合一段时间内的委托信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|--------------------------------------|-------------|
| nPortfolioID | 投资组合ID | int |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号成交明细数据 | CDealDetail |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.156. 请求投资组合一段时间内的委托信息的回调函数

```
def onBatchReqPortfolioMultiDeal(self, nPortfolioID: int, nRequestId: int,
data: list[CDealDetail], isLast: bool, error: XtError) -> None:
```

- 释义： 请求投资组合一段时间内的委托信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|--------------------------------------|----------------------|
| nPortfolioID | 投资组合ID | int |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号成交明细数据 | vector< CDealDetail> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.157. 请求投资组合一段时间内的委托信息的回调函数

```
def onReqPortfolioPosition(self, nPortfolioID: int, nRequestId: int,
data: CPositionStatics, isLast: bool, error: XtError) -> None:
```

- 释义： 请求投资组合一段时间内的委托信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|--------------------------------------|------------------|
| nPortfolioID | 投资组合ID | int |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号持仓统计数据 | CPositionStatics |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.158. 请求投资组合一段时间内的委托信息的回调函数

```
def onBatchReqPortfolioPosition(self, nPortfolioID: int, nRequestId: int,
data: list[CPositionStatics], isLast: bool, error: XtError) -> None:
```

- 释义： 请求投资组合一段时间内的委托信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|--------------------------------------|------------------------------|
| nPortfolioID | 投资组合ID | int |
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 账号持仓统计数据 | vector< CPositionStatics> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.159. 请求收益互换账号框架号的回调函数

```
def onReqStrategyInfo(self, accountID: str, nRequestId: int, accountKey: str,
data: CStrategyInfo, isLast: bool, error: XtError) -> None:
```

- 释义： 请求收益互换账号框架号的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|---------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 收益互换账号框架号数据 | CStrategyInfo |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.160. 请求收益互换账号框架号的回调函数

```
def onBatchReqStrategyInfo(self, accountID: str, nRequestId: int,
    accountKey: str, data: list[CStrategyInfo], isLast: bool,
    error: XtError) -> None:
```

- 释义： 请求收益互换账号框架号的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|-----------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 收益互换账号框架号数据 | vector<CStrategyInfo> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.161. 请求股东号的回调函数

```
def onReqSecuAccount(self, accountID: str, nRequestId: int, accountKey: str,
data: CSecuAccount, isLast: bool, error: XtError) -> None:
```

- 释义：请求股东号的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|--------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 股东号数据 | CSecuAccount |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.162. 请求股东号的回调函数

```
def onBatchReqSecuAccount(self, accountID: str, nRequestId: int,
accountKey: str, data: list[CSecuAccount], isLast: bool,
error: XtError) -> None:
```

- 释义：请求股东号的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|--------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 股东号数据 | vector< CSecuAccount> |

| 参数 | 参数释义 | 参数类型 |
|--------|--------------------------------------|---------|
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.163. 请求资金账号状态的回调函数

```
def onReqAccountStatus(self, nRequestId: int, accountKey: str,
status: bool) -> None:
```

- 释义：请求资金账号状态的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|--------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| status | 账号状态 | bool |

4.3.164. 根据投资组合编号请求投资组合资金信息回调接口

```
def onReqPortfolioStat(self, nRequestId: int, nPortfolioId: int,
data: CPortfolioStat, isLast: bool, error: XtError) -> None:
```

- 释义：请求新股额度数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|--------------|--|----------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| nPortfolioId | 投资组合编号 与reqPortfolioStats接口入参的nPortfolioId对应 | int |
| data | 投资组合资金信息 | CPortfolioStat |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |

| 参数 | 参数释义 | 参数类型 |
|-------|------|---------|
| error | 错误信息 | XtError |

4.3.165. 市场状态信息回调

```
def onReqExchangeStatus(self, accountID: str, nRequestId: int, accountKey: str,
data: list[CExchangeStatus], error: XtError) -> None:
```

- 释义：市场状态信息回调
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---|--------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 市场状态信息，具体参见XtStruct.h中CExchangeStatus结构定义 | vector< CExchangeStatus> |
| error | 错误信息 | XtError |

4.3.166. 请求ETF申赎清单的回调函数

```
def onReqEtfDetails(self, nRequestId: int, data: list[CETFPCFDetail],
error: XtError) -> None:
```

- 释义：请求ETF申赎清单的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|------------------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | ETF申赎清单数据 | vector< CETFPCFDetail> |
| error | 错误信息 | XtError |

4.3.167. 请求ETF成分股信息的回调函数

```
def onReqETFComponentStockInfo(self, nRequestId: int,  
data: list[CETFComponentStockInfo], error: XtError) -> None:
```

- 释义： 请求ETF成分股信息的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|---------------------------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | ETF成分股数据 | vector< CETFComponentStockInfo> |
| error | 错误信息 | XtError |

4.3.168. 请求账号持仓统计的回调函数

```
def onReqFuturePositionStatics(self, accountID: str, nRequestId: int,  
accountKey: str, data: list[CFuturePositionStatics],  
isLast: bool, error: XtError) -> None:
```

- 释义： 请求期货账号持仓统计的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------------------------------------|------------------------------------|
| accountID | 账号ID | string |
| nRequestId | 客户自己维护的请求顺序ID | int |
| accountKey | 账号Key | string |
| data | 期货账号持仓统计数据 | vector< CFuturePositionStatics> |
| isLast | 请求数据可能有多条，需要多次回调该函数，标记是否是一次请求的最后一次回调 | bool |
| error | 错误信息 | XtError |

4.3.169. 请求用户设置数据的回调函数

```
def onReqUserConfig(self, nRequestId: int, data: list[CUserConfig],  
error: XtError) -> None:
```

- 释义： 请求用户设置数据的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|---------------|----------------------|
| nRequestId | 客户自己维护的请求顺序ID | int |
| data | 用户设置数据 | vector< CUserConfig> |
| error | 错误信息 | XtError |

4.4. 主推接口

4.4.1. 主推的用户登录状态

```
def onRtnLoginStatus(self, accountID: str, status: EBrokerLoginStatus,  
brokerType: int, errorMsg: str) -> None:
```

- 释义： 主推的用户登录状态
- note： brokerType取值1:期货账号, 2:股票账号, 3:信用账号, 4:贵金属账号, 5:期货期权账号, 6:股票期权账号, 7:沪港通账号, 10:全国股转账号
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-------------|--------------------|
| accountID | 账号ID | string |
| status | 主推资金账号的登录状态 | EBrokerLoginStatus |
| brokerType | 主推资金账号的类型 | int |
| errorMsg | 错误信息 | string |

4.4.2. 主推的用户登录状态

```
def onRtnLoginStatusWithActKey(self, accountID: str,
status: EBrokerLoginStatus, brokerType: int, accountKey: str,
errorMsg: str) -> None:
```

- 释义： 主推的用户登录状态
- note： brokerType取值1:期货账号, 2:股票账号, 3:信用账号, 4:贵金属账号, 5:期货期权账号, 6:股票期权账号, 7:沪港通账号, 10:全国股转账号
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|----------------------------|--------------------|
| accountID | 账号ID | string |
| status | 主推资金账号的登录状态 | EBrokerLoginStatus |
| brokerType | 主推资金账号的类型 | int |
| accountKey | 账号 key, 用于区分不同类型的相同账号ID的账号 | string |
| errorMsg | 错误信息 | string |

4.4.3. 主推的用户登录状态

```
def onRtnLoginStatusCustom(self, accountID: str,
status: EBrokerLoginStatus, brokerType: int, accountKey: str,
userName: str, errorMsg: str) -> None:
```

- 释义： 主推的用户登录状态
- note： brokerType取值1:期货账号, 2:股票账号, 3:信用账号, 4:贵金属账号, 5:期货期权账号, 6:股票期权账号, 7:沪港通账号, 10:全国股转账号
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|-------------|--------------------|
| accountID | 账号ID | string |
| status | 主推资金账号的登录状态 | EBrokerLoginStatus |
| brokerType | 主推资金账号的类型 | int |

| 参数 | 参数释义 | 参数类型 |
|------------|---------------------------|--------|
| accountKey | 账号 key，用于区分不同类型的相同账号ID的账号 | string |
| userName | 用户名 | string |
| errorMsg | 错误信息 | string |

4.4.4. 主推的报单状态（指令）

```
def onRtnOrder(self, data: COrderInfo) -> None:
```

- 释义：主推的报单状态（指令）
- 参数

| 参数 | 参数释义 | 参数类型 |
|------|--------|------------|
| data | 下单指令信息 | COrderInfo |

4.4.5. 主推的指令统计信息（指令）

```
def onRtnOrderStat(self, data: COrderStat) -> None:
```

- 释义：主推的指令统计信息（指令）
- 参数

| 参数 | 参数释义 | 参数类型 |
|------|--------|------------|
| data | 指令统计信息 | COrderStat |

4.4.6. 主推的委托明细

```
def onRtnOrderDetail(self, data: COrderDetail) -> None:
```

- 释义：主推的委托明细
- 参数

| 参数 | 参数释义 | 参数类型 |
|------|--------|--------------|
| data | 委托明细信息 | COrderDetail |

4.4.7. 主推的成交明细

```
def onRtnDealDetail(self, data: CDealDetail) -> None:
```

- 释义： 主推的成交明细
- 参数

| 参数 | 参数释义 | 参数类型 |
|------|--------|-------------|
| data | 成交明细信息 | CDealDetail |

4.4.8. 主推的委托错误信息

```
def onRtnOrderError(self, data: COrderError) -> None:
```

- 释义： 主推的委托错误信息
- 参数

| 参数 | 参数释义 | 参数类型 |
|------|-----------|-------------|
| data | 下单失败的错误信息 | COrderError |

4.4.9. 主推的撤单失败信息

```
def onRtnCancelError(self, data: CCancelError) -> None:
```

- 释义： 主推的撤单失败信息
- 参数

| 参数 | 参数释义 | 参数类型 |
|------|-----------|--------------|
| data | 撤单失败的错误信息 | CCancelError |

4.4.10. 主推的资金账号信息

```
def onRtnAccountDetail(self, accountID: str, data: CAccountDetail) -> None:
```

- 释义： 主推的资金账号信息
- 参数

| 参数 | 参数释义 | 参数类型 |
|-----------|--------|----------------|
| accountID | 账号ID | string |
| data | 账号资金信息 | CAccountDetail |

4.4.11. 主推的两融资金账号信息

```
def onRtnCreditAccountDetail(self, accountID: str,  
data: CCreditAccountDetail) -> None:
```

- 释义： 主推的两融资金账号信息
- 参数

| 参数 | 参数释义 | 参数类型 |
|-----------|----------|----------------------|
| accountID | 账号ID | string |
| data | 账号两融资金信息 | CCreditAccountDetail |

4.4.12. 主推的产品净值信息

```
def onRtnNetValue(self, data: CNetValue) -> None:
```

- 释义： 主推的产品净值信息
- 参数

| 参数 | 参数释义 | 参数类型 |
|------|--------|-----------|
| data | 产品净值信息 | CNetValue |

4.4.13. 主推行情数据

```
def onRtnPriceData(self, data: CPriceData) -> None:
```

- 释义： 主推行情数据
- 参数

| 参数 | 参数释义 | 参数类型 |
|------|------|------------|
| data | 行情数据 | CPriceData |

4.4.14. 主推市场状态信息

```
def onRtnExchangeStatus(self, data: CExchangeStatus) -> None:
```

- 释义： 主推市场状态信息
- 参数

| 参数 | 参数释义 | 参数类型 |
|------|--------|-----------------|
| data | 市场状态信息 | CExchangeStatus |

4.4.15. 主推两融综合资金

```
def onRtnCreditDetail(self, data: CCreditDetail) -> None:
```

- 释义： 主推两融综合资金
- 参数

| 参数 | 参数释义 | 参数类型 |
|------|--------|---------------|
| data | 两融综合资金 | CCreditDetail |

4.4.16. 主推的算法母单错误信息

```
def onRtnAlgoError(self, nOrderID: int, strRemark: str,  
error: XtError) -> None:
```

- 释义： 主推的算法母单错误信息

- 参数

| 参数 | 参数释义 | 参数类型 |
|-----------|------------|---------|
| nOrderID | 指令ID | int |
| strRemark | 下单时填写的投资备注 | string |
| error | 错误信息 | XtError |

4.4.17. 主推风控信息

```
def onRtnRCMsg(self, accountID: str, accountKey: str,
strMsg: str) -> None:
```

- 释义： 主推风控信息
- 参数

| 参数 | 参数释义 | 参数类型 |
|-----------|-------|--------|
| accountID | 账号ID | string |
| strRemark | 账号Key | string |
| strMsg | 风控信息 | string |

4.4.18. 主推市场和交易日

```
def onRtnMarketAndTradeDay(self, market: str, tradeDay: str) -> None:
```

- 释义： 主推市场和交易日
- 参数

| 参数 | 参数释义 | 参数类型 |
|----------|------|--------|
| market | 交易市场 | string |
| tradeDay | 交易日 | string |

4.4.19. 退订主推函数

```
virtual void unsubscribeMetes(const std::vector<int>* data, int nRequestId) {}
```

- 释义：退订主推函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|--------|--------|
| data | 退订枚举参数 | vector |
| nRequestId | 请求ID | intint |

4.4.20. 退订主推函数的回调

```
virtual void onUnsubscribeMetes(int nRequestId, const XtError& error) {}
```

- 释义：退订主推函数的回调函数
- 参数

| 参数 | 参数释义 | 参数类型 |
|------------|------|---------|
| nRequestId | 请求ID | intint |
| error | 错误信息 | XtError |

Q&A 常见问题与解答

1. onUserLogin回调里的error包含错误信息: End of file

原因：是由于API本地的配置文件traderApi.ini里的isUseSSL和一体化服务器上/home/rzrk/server/config/xtapiservice.lua里的isUseSSL的值不匹配导致的。

解决办法：把两者都修改为0，然后重启API程序和一体化服务器上的XtApiService服务即可。把两者都修改为0，然后重启API程序和一体化服务器上的XtApiService服务即可。

2. onUserLogin回调里的error包含错误信息: md5不匹配禁止登录

原因：是由于一体化服务器上开启了API的md5校验导致的。

解决办法：修改一体化服务器上/home/rzrk/server/config/xtapiservice.lua里的

isUseMD5Check的值修改为0，然后重启一体化服务器上的XtApiService服务即可。

3. onUserLogin回调里的error包含错误信息: 客户端mac不匹配

原因：是由于一体化服务器上开启了API的mac地址校验导致的。

解决办法：修改一体化服务器上/home/rzrk/server/config/xtapiservice.lua里的isUseMacCheck的值修改为0，然后重启一体化服务器上的XtApiService服务即可。

4. 周末要测试，但是账号休市

原因：周末因为股市休市，所以测试环境对于账号也是默认休市处理。

解决办法：在一体化管理端左侧菜单'系统监控->交易日设置'里，设置当天日期为交易日。如果调整之后，管理端上账号状态不是登录成功，可以在ET软件上对该账号关联的交易端执行'停用监控'操作后再'启用监控'。

5. 周一到周五盘后要测试，但是账号休市

原因：目前系统默认16:00账号休市，所以通常16:00以后账号会进入休市状态。

解决办法：在一体化管理端左侧菜单'基础管理->经纪公司'里，对要测试的经纪公司，点击设置来修改日盘时间即可。如果调整之后，管理端上账号状态不是登录成功，可以在ET软件上对该账号关联的交易端执行'停用监控'操作后再'启用监控'。

6. 近场交易说明

智能算法交易CIntelligentAlgorithmOrder和三方主动算法交易CExternAlgorithmOrder，采用近场交易，需要在m_eOrderStrategyType入参填写E_ORDER_STRATEGY_TYPE_APPROACH

7. 请求持仓明细和持仓统计说明

查询持仓明细返回的CPositionDetail一般用于期货持仓展示明细，对股票等业务而言意义不大，股票等业务一般查询使用reqPositionStatics查询CPositionStatics结构

8. 关于委托成交返回的买卖方向的说明

返回的结构中一般存在m_nDirection和m_eOffsetFlag字段，如果是期货，买卖方向依靠m_nDirection判断，如果是股票，m_nDirection无意义，只通过m_eOffsetFlag判断，EOFF_THOST_FTDC_OF_Open 就是买入，EOFF_THOST_FTDC_OF_Close就是卖出。

9. 关于期货下单类型今昨仓的处理

对于区分今昨仓的交易所，比如上期所，平仓操作EOperationType可以选择：

- OPT_CLOSE_SHORT_HISTORY, ///< 平昨空 4
- OPT_CLOSE_SHORT_TODAY, ///< 平今空 5
- OPT_CLOSE_LONG_TODAY_FIRST, ///< 平多，优先平今 6
- OPT_CLOSE_LONG_HISTORY_FIRST, ///< 平多，优先平昨 7
- OPT_CLOSE_SHORT_TODAY_FIRST, ///< 平空，优先平今 8
- OPT_CLOSE_SHORT_HISTORY_FIRST, ///< 平空，优先平昨 9

对于不区分今昨仓的交易所，比如大商所，平仓操作EOperationType可以选择：

- OPT_CLOSE_LONG_TODAY_FIRST, ///< 平多，优先平今 6
- OPT_CLOSE_LONG_HISTORY_FIRST, ///< 平多，优先平昨 7
- OPT_CLOSE_SHORT_TODAY_FIRST, ///< 平空，优先平今 8
- OPT_CLOSE_SHORT_HISTORY_FIRST, ///< 平空，优先平昨 9

不区分今昨仓的交易所，传入平昨空和平今空在风控层可能会报错

10. 特殊字段 m_nLimitedPriceType 说明

- 1, 限价
- 2, 允许改参
- 4, 算法交易用限价计算波动区间
- 8, 不允许指令自动执行
- 16, 价格输入0的不限价可以改参指令
- 96, 指令模式mask
- 128, 条件指令
- 256, 撤销任务撤指令
- 512, 循环交易
- 1024, 手工交易
- 6144, 近场指令GT端指令
- 8192, 有线指令
- 根据用户需求将上面数字相加传入，比如用户需要允许改参和条件指令，就传入 130。

11. 常见报错说明

ImportError: DLL load failed while importing XtTraderPyApi: 找不到指定的模块。原因是缺少Windows相关VC库，可尝试安装微软常用运行库合集，或者联系迅投工程师解决。