**国际期货股票API接口开发规范**

**V1.18**

**(2023.09)**

目录

[一、 变更履历 5](#_Toc132104514)

[二、 概览 9](#_Toc132104515)

[1.1. 背景 9](#_Toc132104516)

[1.2. API接口概要及架构 10](#_Toc132104517)

[1.3. 期货接口说明 11](#_Toc132104518)

[1.4. 股票接口说明 12](#_Toc132104519)

[1.5. 全局的错误代码 13](#_Toc132104520)

[三、 股票交易API接口参考 15](#_Toc132104521)

[1. 接口工作过程 16](#_Toc132104522)

[2. 接口模式 16](#_Toc132104523)

[~~3. 请求数据频率~~ 16](#_Toc132104524)

[4. 函数接口说明 17](#_Toc132104525)

[1) CreateStockApi 方法 17](#_Toc132104526)

[2) OnFrontConnected 方法 17](#_Toc132104527)

[3) OnFrontDisconnected方法 17](#_Toc132104528)

[4) OnHeartBeatWarning方法 18](#_Toc132104529)

[5) 登录ReqUserLogin/OnRspUserLogin/OnRspNeedVerify/ OnRspAccount 19](#_Toc132104530)

[6) 登出 ReqUserLogout/OnRspUserLogout方法 21](#_Toc132104531)

[7) 修改密码ReqPasswordUpdate/OnRspPasswordUpdate 22](#_Toc132104532)

[8) 报单ReqOrderInsert/OnRspOrderInsert 23](#_Toc132104533)

[9) 订单推送OnRtnOrder 27](#_Toc132104534)

[10) 成交推送OnRtnTrade 30](#_Toc132104535)

[11) 持仓信息推送OnRtnPosition 32](#_Toc132104536)

[12) 资金信息推送~~OnRtnAccount~~ OnRtnCapital 33](#_Toc132104537)

[12) 改单请求/返回ReqOrderModify/OnRspOrderModify方法 36](#_Toc132104538)

[13) 撤单请求/返回ReqOrderCancel/OnRspOrderCancel 38](#_Toc132104539)

[14) 委托单查询/返回ReqQryOrder/OnRspQryOrder 40](#_Toc132104540)

[15) 成交查询请求/返回ReqQryTrade/OnRspQryTrade 42](#_Toc132104541)

[16) 持仓明细查询请求/返回ReqQryPosition/OnRspQryPosition 43](#_Toc132104542)

[17) 资金查询请求/返回ReqQryCapital/OnRspQryCapital 45](#_Toc132104543)

[18) 经纪商查询请求/返回ReqQryBroker/OnRspQryBroker 48](#_Toc132104544)

[19) 版本号查询请求/返回ReqQryVersion/OnRspQryQryVersion 48](#_Toc132104545)

[20) 币种查询请求/返回ReqQryCurrency/ OnRspQryCurrency 50](#_Toc132104546)

[21) 交易时间查询请求/返回ReqQryTradeTime/ OnRspQryTradeTime 51](#_Toc132104547)

[22) 交易所查询请求/返回ReqQryExchange /OnRspQryExchange 52](#_Toc132104548)

[23) 合约查询请求/返回ReqQryInstrument/OnRspQryInstrument 53](#_Toc132104549)

[24) 跳点查询请求/返回ReqQryTick/ OnRspQryTick 56](#_Toc132104550)

[25) 订单类型查询请求/返回ReqQryOrderType/ OnRspQryOrderType 57](#_Toc132104551)

[26) 设置双重认证密码/返回ReqSetVerifyQA/ OnRspSetVerifyQA 58](#_Toc132104552)

[27) 绑定双重认证需要的MAC地址/返回ReqSafeVerify/ OnRspSafeVerify 59](#_Toc132104553)

[四、 期货交易API接口参考 60](#_Toc132104554)

[1. 接口工作过程 62](#_Toc132104555)

[2. 接口模式 62](#_Toc132104556)

[3. 函数接口说明 63](#_Toc132104557)

[1) CreateFutureApi方法 63](#_Toc132104558)

[2) OnFrontConnected 方法 63](#_Toc132104559)

[3) OnFrontDisconnected方法 63](#_Toc132104560)

[4) OnHeartBeatWarning方法 64](#_Toc132104561)

[5) 登录ReqUserLogin/OnRspUserLogin/OnRspNeedVerify/OnRspAccount 65](#_Toc132104562)

[6) 登出 ReqUserLogout/OnRspUserLogout方法 67](#_Toc132104563)

[7) 修改密码ReqPasswordUpdate/OnRspPasswordUpdate 68](#_Toc132104564)

[8) 报单ReqOrderInsert/OnRspOrderInsert 69](#_Toc132104565)

[9) 订单推送OnRtnOrder 74](#_Toc132104566)

[10) 成交推送OnRtnTrade 76](#_Toc132104567)

[11) 持仓信息推送OnRtnPosition 78](#_Toc132104568)

[12) 资金信息推送~~OnRtnAccount~~ OnRtnCapital 79](#_Toc132104569)

[13) 改单请求/返回ReqOrderModify/OnRspOrderModify方法 82](#_Toc132104570)

[14) 撤单请求/返回ReqOrderCancel/OnRspOrderCancel 84](#_Toc132104571)

[15) 委托单查询/返回ReqQryOrder/OnRspQryOrder 86](#_Toc132104572)

[16) 成交查询请求/返回ReqQryTrade/OnRspQryTrade 89](#_Toc132104573)

[17) 持仓明细查询请求/返回ReqQryPosition/OnRspQryPosition 91](#_Toc132104574)

[18) 资金查询请求/返回ReqQryCapital/OnRspQryCapital 93](#_Toc132104575)

[19) 版本号查询请求/返回ReqQryVersion/OnRspQryVersion 95](#_Toc132104576)

[20) 币种查询请求/返回ReqQryCurrency/ OnRspQryCurrency 96](#_Toc132104577)

[21) 交易时间查询请求/返回ReqQryTradeTime/ OnRspQryTradeTime 97](#_Toc132104578)

[22) 交易所查询请求/返回ReqQryExchange /OnRspQryExchange 98](#_Toc132104579)

[23) 合约查询请求/返回ReqQryInstrument/OnRspQryInstrument 99](#_Toc132104580)

[24) 策略查询请求/返回ReqQryStrategy/ OnRspQryStrategy 102](#_Toc132104581)

[25) 策略明细查询请求/返回ReqQryStrategyDetail/ OnRspQryStrategyDetail 105](#_Toc132104582)

[25) 设置双重认证密码/返回ReqSetVerifyQA/ OnRspSetVerifyQA 107](#_Toc132104583)

[26) 绑定双重认证需要的MAC地址请求/返回ReqSafeVerify/ OnRspSafeVerify 108](#_Toc132104584)

[27) 查询品种列表请求/返回ReqQryCommodity/ OnRspQryCommodity 109](#_Toc132104585)

[28) 持仓总结查询请求/返回ReqQryTotalPosition/ OnRspQryTotalPosition 111](#_Toc132104586)

[29) 交易所冬夏令开盘时间查询请求/返回ReqQryExchangeTime/ OnRspQryExchangeTime 113](#_Toc132104587)

[30) 品种开收盘时间查询请求/返回ReqQryExchangeTime/ OnRspQryCommodityTime 114](#_Toc132104588)

[五、 期货股票通用行情API接口参考 115](#_Toc132104589)

[1. 接口工作过程 115](#_Toc132104590)

[2. 接口模式 116](#_Toc132104591)

[3. 函数接口说明 117](#_Toc132104592)

[1）CreateMarketApi方法 117](#_Toc132104593)

[2）OnFrontConnected 方法 117](#_Toc132104594)

[3）OnFrontDisconnected方法 117](#_Toc132104595)

[4）OnHeartBeatWarning方法 118](#_Toc132104596)

[5）登录ReqUserLogin/OnRspUserLogin 118](#_Toc132104597)

[6）行情推送请求/响应ReqMarketData/OnRspMarketData 119](#_Toc132104598)

[7）经纪商获取请求/响应ReqBrokerData/OnRspBrokerData 124](#_Toc132104599)

[8）交易日推送OnRspTradeDate 125](#_Toc132104600)

[9）设置心跳超时时间SetHeartBeatTimeout 125](#_Toc132104601)

[六、 常见问题-期货交易相关 126](#_Toc132104602)

[七、 常见问题-股票交易相关 128](#_Toc132104603)

[八、 常见问题-行情相关 129](#_Toc132104604)

# 变更履历

**说明：**

版本号格式Vx.x update?仅说明文档有更新，API程序无更新

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **变更内容** | **作者** | **日期** |
| V1.0 | 初版 | Rainer.Jing | 2018/08/24 |
| V1.1 | Chapter2， section1.3, section1.4增加说明  ~~注意：Linux版本的库使用时，编译器最低版本为gcc/g++ 7.2~~  (从V1.7删除) | Rainer.Jing | 2018/09/03 |
| V1.5 | 1. 登录时双重认证    1. Chapter三： 4 - 4), 4 - 26), 4 - 27)    2. Chapter四： 3 – 4), 3 - 25), 4 - 26)    3. 增加一个双重认证说明文档” 国际期货股票API\_双重认证交互说明.xlsx” 2. 删除Chapter三：section3 3. Chapter四：期货接口中增加持仓总结查询 | Rainer.Jing | 2018/09/26 |
| V1.5  update1 | 1. Chapter四：3 – 22)商品类别CommodityType，增加值的说明 2. Chapter三：3 – 23)商品类别CommodityType，增加值的说明 | Rainer.Jing | 2018/10/18 |
| V1.7 | 1. Chapter四：3 – 13) CFutureRspOrderCancelField结构体中，增加OrderNo,同事变量说明进行修正 2. Chapter三：4 – 13) CStockRspOrderCancelField结构体中，增加LocalNo 3. Chapter五：3 – 5) OnRspMarketData结构体中QuoteType字段增加说明 | Rainer.Jing | 2018/10/29 |
| V1.8 | 1. DADataType.h，删除一些无用的常量定义 2. DAFutureStruct.h    * + - 1. Chapter四：3 – 15) CFutureRspTradeField结构体增加ErrorDescription字段          2. 修改一些注释说明 3. DAFutureApi.h   static CFutureApi \*CreateFutureApi(bool bRecordLog, const char\* lpszLogFileName = "Future.log");  修改参数lpszLogFileName的缺省值为"Future.log"   1. DAMarketStruct.h, 修改一些注释说明 2. DAMarketApi.h   static CMarketApi \*CreateMarketApi(bool bRecordLog, const char\* lpszLogFileName = "Market.log");  修改参数lpszLogFileName的缺省值为"Market.log"   1. DAStockStruct.h    * + - 1. Chapter三：3 – 17) CStockRspCapitalField结构体增加ErrorDescription字段          2. 修改一些注释说明          3. 删除没有用的结构体 CStockRspHisMarketDataField、CStockQryHisMarketDataField、CStockRspSetVerifyQAField 2. DAStockApi.h   static CStockApi \*CreateStockApi(bool bRecordLog, const char\* lpszLogFileName = "Stock.log");  修改参数lpszLogFileName的缺省值为"Stock.log"   1. v1.8 API修改两个bug，请参照ReleaseNote.txt 2. Chapter四：3 –22) CFutureQryInstrumentField结构体与程序不一致，更正 | Rainer.Jing | 2018/11/28 |
| V1.9 | 1. Chapter四：3 – 17) CFutureRspCapitalField结构体中，FloatingProfit的说明由” 浮动盈亏”改为”今日平仓盈亏” 2. Chapter四：3 – 7) 增加报单返回消息中几个订单号的说明 3. v1.9 API修改两个bug，请参照ReleaseNote.txt | Rainer.Jing | 2019/01/10 |
| V1.10 | 1. Chapter四：函数列表，新增功能的对应函数说明 2. Chapter四：增加3 –28) 查询品种列表   增加3 –29) 交易所冬夏令开盘时间查询  增加3 –30) 品种开收盘时间查询   1. Chapter五：3-5) 增加行情订阅方式说明，按交易所/品种/合约订阅方式说明 2. Chapter五：3-5) 行情数据中盘中结算价 和 当前交易日前结算价区分 3. Chapter三：3 –7) 回报中AcceptType字段废弃   Chapter三：3 –14) 回报中AcceptType字段废弃   1. Chapter四：3 –7) 回报中AcceptType字段废弃   Chapter四：3 –14) 回报中AcceptType字段废弃 | Rainer.Jing | 2019/03/27 |
| V1.11 | 1. Chapter五：3 – 7) 增加功能：期货品种交易日推送 | Rainer.Jing | 2019/04/04 |
| V1.11  Update1 | 1. Chapter五：3 – 5) 按交易所订阅行情说明中的错误订正，并增加了背景说明 2. Chapter二：1.5 增加全局的错误代码说明 3. Chapter四：3 –17)函数名订正，OnRsqQryCapital 🡪 OnRspQryCapital | Rainer.Jing | 2019/05/14 |
| V1.12 | 1. DAStockApi.h   函数名订正，OnRsqQryCapital 🡪 OnRspQryCapital  virtual void Init() = 0; 改为 virtual bool Init() = 0;   1. DAMarketApi.h   a) virtual void Init() = 0; 改为 virtual bool Init() = 0;   1. DAFutureApi.h   函数名订正，OnRsqQryCapital 🡪 OnRspQryCapital  virtual void Init() = 0; 改为 virtual bool Init() = 0;   1. Chapter三：4 –7)股票下单时一些字段说明的订正 | Rainer.Jing | 2019/08/15 |
| V1.13 | 修改ReleaseNote v1.13中记述的bug | Rainer.Jing | 2019/09/11 |
| V1.14  Update1 | 1. Chapter四：3 – 7) 下单接口中，MinQty的错误描述修正 | Rainer.Jing | 2020/04/20 |
| V1.15 | 1. Chapter三：4 – 11) 函数名错误修正 2. Chapter四：3 – 11) 函数名错误修正 3. Chapter四：3 – 11) 成交数据太多时，可从指定成交号开始分页查询 | Rainer.Jing | 2020/05/12 |
| V1.16 | 接口结构体字段重新命名 | Rainer.Jing | 2020/05/26 |
| V1.17 | 在期货OnRspOrderInsert接口中，更新CFutureRspOrderInsertField结构体，新增OrdSourceType字段和Tag50字段 | Kan.li | 2020/12/25 |
| V1.17.01 | 修改行情SetHeartBeatTimeout函数，新增一个默认为true的参数bIsStrictCheck，一定场景下，可根据要求设置为fasle，避免和前置服务的版本兼容问题。 | Kan.li | 2021/01/12 |
| V1.17.02 | 1.17.01心跳判断逻辑优化 | Kan.li | 2021/11/01 |
| V1.17.03 | 1. 期货下单接口新增IsProgram字段，标记程序化单或人工单。 2. 常见问题中增加撤单必输项的说明 | Kan.li | 2021/11/26 |
| V1.17.03 | IsProgram字段位置调整 | Kan.li | 2022/03/11 |
| V1.17.04 | 下改撤单tag50字段调整 | Kan.li | 2022/05/10 |
| V1.17.05 | Chapter四：3 – 8) 保单回调函数，保持和v1.14字段设置值的行为一致 | Rainer.Jing | 2022/07/04 |
| V1.17.06 | Bug修正：业务消息偶发未回调给上层程序修正 | Rainer.Jing | 2022/08/23 |
| V1.18.0.1 | 1、行情接口，登录调用时，增加行情broker ID  Chapter五→3→ 5) CMarketReqUserLoginField结构体增加字段  2、废除按交易所/品种订阅功能  Chapter五→3→ 6) 订阅说明  3、增加网络底层错误码说明  Chapter二→1.5→ 2 | Rainer.Jing | 2023/03/03 |
| V1.18.1.0 | 期货交易接口下单/改单/撤单结构体增加OrgOrderLocationID字段。  Chapter四→3→ 8)  Chapter四→3→ 13)  Chapter四→3→ 14) | Rainer.Jing | 2023/04/18 |
| V1.18.2.0 | 优化Socket断线时重连机制 | Rainer.Jing | 2023/09/14 |

# 概览

## 背景

本API可用来交易中国境外期货和股票，并且用来获取相应市场的行情数据。

目前支持的交易所/市场有以下(但不限于)这些：

期货类

|  |  |  |
| --- | --- | --- |
| No | 交易所 | 说明 |
| 1 | CME | 包括CME group旗下的CME,COMEX,NYMEX,CBOT  官方网站：  https://www.cmegroup.com/ |
| 2 | HKEX | 香港交易所  官方网站：  http://www.hkex.com.hk |
| 3 | SGX | 新加坡交易所  官方网站：  http://www.sgx.com |
| 4 | ICE | 洲际交易所  官方网站：  https://www.theice.com/index |
| 5 | LME | 伦敦金属交易所  官方网站：  https://www.lme.com/ |

股票类

|  |  |  |
| --- | --- | --- |
| No | 市场 | 说明 |
| 1 | NASD | 美国市场的股票。  需要注意的是，2008年金融危机后，美国通过法案，允许上市公司在美国所有的交易所都可以交易。比如，在NYSE挂牌的股票可以在NASDQ交易所交易，反之亦然。 |
| 2 | HKEX | 香港交易所的股票 |
| 3 | KRX | 韩国的股票 |

## API接口概要及架构

本API是一个基于C++的类库，通过使用和扩展类库提供的接口来实现交易所需的功能。鉴于国内期货界CTP接口的广泛使用，本API接口仿照CTP接口的方式，使得有CTP接口使用经验得用户能够快速上手开发。

同时由于股票和期货的某些不同，股票的交易接口和期货的交易接口被独立开来，但是API库是同一个库，敬请用户留意。下面两个小节按期货和股票分别加以说明。

**期货接口使用方式**

期货交易客户端

(使用DAApi.dll)

期货前置服务器

期货行情客户端

(使用DAApi.dll)

期货行情服务器

**股票接口使用方式**

股票交易客户端

(使用DAApi.dll)

股票前置服务器

股票行情客户端

(使用DAApi.dll)

股票行情服务器

本API中无论任何地方，对于以下的错误码，值为0时表示成功。

// 错误码

TDAIntType ErrorID;

## 期货接口说明

1. 交易接口

交易接口实现的功能，包括期货合约的下单、改单、撤单。成交单查询、持仓查询、合约查询等。

该类库包含以下几个文件：

|  |  |  |
| --- | --- | --- |
| 文件名 | 版本 | 文件描述 |
| DAFutureApi.h |  | 交易接口头文件 |
| DAFutureStruct.h |  | 定义了UserAPI所需的一系列数据类型的头文件 |
| DADataType.h |  | 定义了一系列业务相关的数据结构的头文件 |
| DAApi.dll |  | Windows动态链接库二进制文件 |
| DAApi.lib |  | 导入库文件 |
| libDAApi.so |  | Linux库(only 64bit) |

1. 行情接口

用于获取期货行情，该类库包含以下几个文件：

|  |  |  |
| --- | --- | --- |
| 文件名 | 版本 | 文件描述 |
| DAMarketApi.h |  | 交易接口头文件 |
| DAMarketStruct.h |  | 定义了UserAPI所需的一系列数据类型的头文件 |
| DAApi.dll |  | Windows动态链接库二进制文件 |
| DAApi.lib |  | 导入库文件 |
| libDAApi.so |  | Linux库(only 64bit) |

## 股票接口说明

1. 交易接口

交易接口实现的功能，包括股票的下单、改单、撤单。成交单查询、持仓查询、合约查询等。

该类库包含以下几个文件：

|  |  |  |
| --- | --- | --- |
| 文件名 | 版本 | 文件描述 |
| DAStockApi.h |  | 交易接口头文件 |
| DAStockStruct.h |  | 定义了UserAPI所需的一系列数据类型的头文件 |
| DADataType.h |  | 定义了一系列业务相关的数据结构的头文件 |
| DAApi.dll |  | Windows动态链接库二进制文件 |
| DAApi.lib |  | 导入库文件 |
| libDAApi.so |  | Linux库(only 64bit) |

1. 行情接口

用于获取期货行情，该类库包含以下几个文件：

|  |  |  |
| --- | --- | --- |
| 文件名 | 版本 | 文件描述 |
| DAMarketApi.h |  | 交易接口头文件 |
| DAMarketStruct.h |  | 定义了UserAPI所需的一系列数据类型的头文件 |
| DAApi.dll |  | Windows动态链接库二进制文件 |
| DAApi.lib |  | 导入库文件 |
| libDAApi.so |  | Linux库(only 64bit) |

## 全局的错误代码

1、业务函数中通用错误

//期货接口中的错误信息结构体

struct CFutureRspInfoField

{

TDAIntType ErrorID; // 错误码

TDAStringType ErrorMsg; // 错误描述

};

//股票接口中的错误信息结构体

struct CStockRspInfoField

{

TDAIntType ErrorID; // 错误码

TDAStringType ErrorMsg; // 错误描述

};

|  |  |  |
| --- | --- | --- |
| 错误代码  (ErrorID) | 错误描述 | 说明 |
| 0 | 成功 | 各种函数返回结果成功 |
| 90001 | 解析协议头失败 | 直达系统内部错误 |
| 90002 | 解压缩数据失败 | 直达系统内部错误 |
| 90003 | 未压缩数据体是空的 | 直达系统内部错误；通常指查询的结果为空，服务器端有应答，但是没有业务数据 |
| 90004 | 压缩数据体是空的 | 直达系统内部错误；通常指查询的结果为空，服务器端有应答，但是没有业务数据 |
| 90005 | 解密失败 | 直达系统内部错误 |
| 90006 | 服务器返回数组是空的 | 直达系统内部错误；通常指查询的结果为空，服务器端有应答，但是没有业务数据 |
|  |  |  |
|  |  |  |

2、连接断开通用错误

void OnFrontDisconnected (int nReason)

|  |  |  |
| --- | --- | --- |
| 错误代码  (nReason) | 错误描述 | 说明 |
| 4001 | 设置阻塞失败 | Socket底层错误 |
| 4002 | 设置非阻塞失败 | Socket底层错误 |
| 4003 | 接收数据失败 | Socket底层错误 |
| 4004 | 接收0长度数据，已断开 | Socket底层错误 |
| 4005 | 发送数据失败 | Socket底层错误 |
| 4006 | 发送0长度数据，已断开 | Socket底层错误 |
| 4007 | 选择失败 | Socket底层错误 |
| 4008 | 服务器无响应 | Socket底层错误 |
|  |  |  |
|  |  |  |

# 股票交易API接口参考

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 业务类型 | 业务 | 请求接口 | 说明 | 响应接口 |
| 连接 | 连接 | Init | 连接回报 | OnFrontConnected |
| 断开 |  |  | 连接断开回报 | OnFrontDisconnected |
| 心跳 |  |  | 心跳警告 | OnHeartBeatWarning |
| 登录 | 登录 | ReqUserLogin | 登录回报 | OnRspUserLogin/OnRspNeedVerify |
| 登出 | ReqUserLogout | 登出回报 | OnRspUserLogout |
| 修改密码 | ReqPasswordUpdate | 修改密码回报 | OnRspPasswordUpdate |
| 双重认证step1 | ReqSetVerifyQA | 设置双重认证密码 | OnRspSetVerifyQA |
| 双重认证step2 | ReqSafeVerify | 绑定MAC地址 | OnRspSafeVerify |
| 交易 | 报单 | ReqOrderInsert | 报单回报 | OnRspOrderInsert |
| 改单 | ReqOrderModify | 改单回报 | OnRspOrderModify |
| 撤单 | ReqOrderCancel | 撤单回报 | OnRspOrderCancel |
| 推送 |  |  | 报单状态推送 | OnRtnOrder |
|  |  | 成交信息推送 | OnRtnTrade |
|  |  | *持仓信息推送* | *OnRtnPosition* |
|  |  | 资金信息推送 | OnRtnAccount |
| 查询 | 委托单查询 | ReqQryOrder | 委托单查询回报 | OnRspQryOrder |
| 成交查询 | ReqQryTrade | 成交查询回报 | OnRspQryTrade |
| 持仓明细查询 | ReqQryPosition | 持仓明细查询回报 | OnRspQryPosition |
| 资金查询 | ReqQryCapital | 资金查询回报 | OnRspQryCapital |
| *经纪商查询*  *(港股专有)* | *ReqQryBroker* | *经纪商查询回报* | *OnRspQryBroker* |
| 版本号查询 | ReqQryVersion | 版本号查询回报 | OnRspQryQryVersion |
| 币种查询 | ReqQryCurrency | 币种查询回报 | OnRspQryCurrency |
| 交易时间查询 | ReqQryTradeTime | 交易时间查询回报 | OnRspQryTradeTime |
| 交易所查询 | ReqQryExchange | 交易所查询回报 | OnRspQryExchange |
| 合约查询 | ReqQryInstrument | 合约查询回报 | OnRspQryInstrument |
| 跳点查询 | ReqQryTick | 跳点查询回报 | OnRspQryTick |
| *订单类型查询* | *ReqQryOrderType* | *订单类型查询回报* | *OnRspQryOrderType* |

注：以上斜体字的接口，当前API版本未实现。

## 接口工作过程



## 接口模式

股票API的DAStockApi.h头文件中提供了两个接口，分别为CStockApi和IStockEvent，方便客户端应用程序的开发。客户端应用程序可以通过CStockApi发出操作请求，通过继承IStockEvent并重载回调函数来处理后台服务的响应。

## ~~请求数据频率~~

~~1. 请求资金查询、持仓查询等数据，每秒允许请求1次~~

~~2. 下单、改单、撤单等交易请求，每秒请求次数不受限制。如果服务器来不及处理交易数据，可以通过服务器端发送降级指令，降级处理，限制客户每秒交易数据，查询数据的请求。~~

~~3. 成交数据推送，为实时推送~~

## 函数接口说明

### CreateStockApi 方法

创建实例方法。

**函数原形**：

static CStockApi \*CreateStockApi(bool bRecordLog, const char\* lpszLogFileName = "Stock.log", const char\*lpszLogFileDir = "");

bRecordLog: true表示记录日志，false表示不记录日志。

lpszLogFileName：日志文件。

lpszLogFileDir： 日志文件文件夹路径。

### OnFrontConnected 方法

当客户端与前置建立起通信连接时（还未登录前），该方法被调用。

**函数原形：**

void OnFrontConnected()；

本方法在完成初始化后调用，可以在其中完成用户登录任务。

### OnFrontDisconnected方法

当客户端与前置通信连接断开时，该方法被调用。当发生这个情况

后，API会自动重新连接，客户端可不做处理。自动重连地址，可能是原来注册

的地址，也可能是系统支持的其它可用的通信地址，它由程序自动选择。

**函数原形：**

void OnFrontDisconnected (int nReason)；

**参数：**

nReason：连接断开原因

### OnHeartBeatWarning方法

心跳超时警告。当长时间未收到报文时，该方法被调用。

**函数原形：**

void OnHeartBeatWarning(int nTimeLapse)；

**参数：**

nTimeLapse：距离上次接收报文的时间

### 登录ReqUserLogin/OnRspUserLogin/OnRspNeedVerify/ OnRspAccount

从API v1.5引入双重认证，具体交互过程请参阅**“国际期货股票API\_双重认证交互说明.xlsx”**

函数原型：

|  |
| --- |
| bool ReqUserLogin  (CStockReqUserLoginField \*pReqUserLogin, int iRequestID); |

参数：ReqUserLogin

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 用户密码  TDAStringType UserPwd;  // 用户类型  TDAStringType UserType;  // 软件名称  TDAStringType SoftwareName;  // 软件版本号  TDAStringType SoftwareVersion;  // 授权码  TDAStringType AuthorCode;  //登录机器网卡MAC地址(since API v1.5,双重认证使用)  MacAddress  // 错误信息  TDAStringType ErrorDescription; |

1. 函数原型：

|  |
| --- |
| void OnRspUserLogin  (CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：OnRspUserLogin

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

错误码：

|  |  |
| --- | --- |
| **登录错误码** | **错误内容** |
| 10001 | 用户名不正确 |
| 10002 | 登录密码错误 |
| 10003 | 密码错误次数超限，用户已冻结，请联系客服解冻！ |
| 10004 | 用户已被冻结，请联系客服解冻！ |
| 10013 | 授权码信息不正确，请确认UserType，SoftwareName，SoftwareVersion，AuthorCode的值 |

1. 函数原型：

服务器端在判断到需要做双重认证时，会指示API回调此函数

|  |
| --- |
| void OnRspNeedVerify(bool bFirstLogin, bool bHasSetQA); |

参数：bFirstLogin

|  |
| --- |
| 当设备(根据login消息中的MacAddress字段识别)第一次登录交易前置时， OnRspNeedVerify()函数被回调，且bFirstLogin 的值为true,用户在回调函数中需调用ReqSetVerifyQA()函数设置双重认证信息。  当设备(根据login消息中的MacAddress字段识别)第一次以后登录交易前置时，OnRspNeedVerify()函数不会被回调。 |

1. 函数原型：

在OnRspUserLogin指示成功登录回调后，该函数被回调，CStockRspAccountField结构体中返回每个币种对应的资金账号

|  |
| --- |
| void OnRspAccount(CStockRspAccountField \*pRspAccount, CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：pRspAccount

|  |
| --- |
| // 帐户币种  TDAStringType CurrencyNo;  // 资金账号  TDAStringType AccountNo; |

### 登出 ReqUserLogout/OnRspUserLogout方法

函数原型：

|  |
| --- |
| bool ReqUserLogout  (CStockReqUserLogoutField \*pReqUserLogout, int iRequestID); |

参数：ReqUserLogout

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 资金账号  TDAStringType AccountNo;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspUserLogout  (CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：OnRspUserLogout

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 修改密码ReqPasswordUpdate/OnRspPasswordUpdate

函数原型：

|  |
| --- |
| bool ReqPasswordUpdate(CStockReqPasswordUpdateField \*pReqPasswordUpdate, int iRequestID) = 0; |

参数：CStockReqPasswordUpdateField

|  |
| --- |
| // 用户代码  TDAStringType UserId;  // 用户原密码  TDAStringType OldPassword;  // 用户新密码  TDAStringType NewPassword;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspPasswordUpdate( CStockRspPasswordUpdateField \*pRspPasswordUpdate, CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) |

参数：CStockRspPasswordUpdateField

|  |
| --- |
| // 用户代码  TDAStringType UserId;  // 用户原密码  TDAStringType OldPassword;  // 用户新密码  TDAStringType NewPassword;  // 错误信息  TDAStringType ErrorDescription; |

参数：CStockRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 报单ReqOrderInsert/OnRspOrderInsert

说明：

TDAStringType SystemNo; // 系统编号

TDAStringType LocalNo; // 本地编号

TDAStringType OrderNo; // 定单号

LocalNo是使用API的程序给的订单号,相当于CTP里面的orderref，当日内最好不要重复

SystemNo是直达系统给的订单号；

OrderNo是交易所给的订单号，相当于CTP里面的OrderSysID

函数原型：

|  |
| --- |
| bool ReqOrderInsert  (CStockReqOrderInsertField \*pReqOrderInsert, int iRequestID); |

参数：ReqOrderInsert

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 资金账号(一个用户可以有多个币种的资金，所以会有多个资金账号，下单需指定)  // 请参照登录成功后的回调函数OnRspAccount的说明  TDAStringType AccountNo;  // 本地编号  TDAStringType LocalNo;  // 交易所代码  TDAStringType ExchangeCode;  // 合约代码  TDAStringType ContractCode;  // 买还是卖：1=buy 2=sell  TDAStringType BidAskFlag;  //卖空时使用,1=开仓;2=平仓,同时和订单的买卖方向相反。非卖空时，0或不设置值  TDAStringType OpenCloseFlag;  // 下单数  TDAStringType OrderQty;  // 下单价格  TDAStringType OrderPrice;  // 定单类型：(from v1.12)  // 美股：1=限价单  // 港股：5=竞价限价盘; 6=竞价盘; 7=增强限价;8=特别限价盘  TDAStringType OrderType;  // 触发价格(股票中未使用)  TDAStringType TriggerPrice;  // 有效日期（1=当日有效）  TDAStringType TIF;  // 策略ID(股票中未使用)  TDAStringType StrategyId;  // 显示委托量，必须小于委托量(股票中未使用)  TDAStringType MaxShow;  // 最小成交量(股票中未使用)  TDAStringType MinQty;  // 错误信息  TDAStringType ErrorDescription; |

说明：

API用户下单后，OnRspOrderInsert()函数会被回调两次：

1. 第一次回调返回直达系统给这笔订单指定的系统号SystemNo，表明直达系统收到并确认这笔订单。
2. 第二次回调返回交易所的反馈的订单号OrderNo(如果一切正常)。通常资金不足等情况发生时，不会有订单号，因为订单指令没有向交易所发送。

函数原型：

|  |
| --- |
| void OnRspOrderInsert  (CStockRspOrderInsertField \*pRspOrderInsert,  CStockRspInfoField \*pRspInfo,  int iRequestID,  bool bIsLast); |

参数：CStockRspOrderInsertField

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 资金账号  TDAStringType AccountNo;  // 系统编号  TDAStringType SystemNo;  // 本地编号  TDAStringType LocalNo;  // 定单号  TDAStringType OrderNo;  // 交易所代码  TDAStringType ExchangeCode;  // 合约代码  TDAStringType ContractCode;  // 买还是卖：1=buy 2=sell  TDAStringType BidAskFlag;  // 下单数  TDAStringType OrderQty;  // 下单价格  TDAStringType OrderPrice;  // 定单类型：(from v1.12)  // 美股：1=限价单  // 港股：5=竞价限价盘; 6=竞价盘; 7=增强限价;8=特别限价盘  TDAStringType OrderType;  // 下单日期  TDAStringType OrderDate;  // 下单时间  TDAStringType OrderTime;  // 错误代码(保留，未使用)  TDAStringType ErrorCode;  // 订单状态（2：已排队；3：部分成交；4：完全成交；5：已撤余单；6：已撤单；7：指令失败；8：待送出；9：待更改；A：待撤单）  TDAStringType OrderState;  // 撤单的用户ID  TDAStringType CancelUserId;  // 触发价格  TDAStringType TriggerPrice;  // 有效日期（1=当日有效）  TDAStringType TIF;  //卖空时使用,1=开仓;2=平仓,同时和订单的买卖方向相反。非卖空时，0或不设置值  TDAStringType OpenCloseFlag;  // 策略ID(股票中未使用)  TDAStringType StrategyId;  // 显示委托量，必须小于委托量(股票中未使用)  TDAStringType MaxShow;  // 最小成交量(股票中未使用)  TDAStringType MinQty;  // 交易所返回时间  TDAStringType ExchangeTime; |

参数：CStockRspInfoField

|  |
| --- |
| // 错误码(用来判断报单成功或失败)  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

错误码

|  |  |
| --- | --- |
| **系统号错误码** | **错误内容** |
| 20005 | 系统号生成失败 |

下单错误码

|  |  |
| --- | --- |
| **下单错误码** | **错误内容** |
| 20000 | 下单失败 |
| 20001 | 资金不足 |
| 20002 | 交易服务器未连接 |
| 20003 | 您已被禁止交易，请联系客服解禁 |
| 20004 | 下单被拒绝 |
| 20005 | 系统号生成失败 |
| 20006 | 您的资金账户中没有该合约交易所需的币种 |
| 20007 | 该合约已经到期，不能交易 |
| 20008 | 该交易市场未开市，不能交易 |
| 20009 | 该交易市场已闭市，不能交易 |
| 20010 | 下单价格超出限定范围 |
| 20011 | 下单数量过大 |
| 20012 | 您被禁止交易该合约，请联系客服 |
| 20013 | 您的资金账户风险率过高，已被禁止交易 |
| 20015 | 您的账户交易数据出现异常，暂停交易 |
| 20019 | 下单数量超过客户持仓限量 |
| 20020 | 下单数量超过客户可下单买量 |
| 20021 | 下单数量超过客户可下单卖量 |
| 20022 | 下单数量超过公司持仓限量 |
| 20023 | 下单数量超过公司可下单买量 |
| 20024 | 下单数量超过公司可下单卖量 |
| 20025 | Dealer下单数量超过最大可下单数量 |
| 20026 | 合约临近交割，只可平仓，不能开仓 |
| 20027 | 该合约已过最后交易日时间，禁止电子盘交易 |
| 20028 | 商品未开放，禁止交易 |
| 20029 | 单笔下单数量超过该商品单笔最大可下单数 |
| 20032 | 持仓不足，平仓失败 |
| 20033 | 您不能交易该商品,请联系客服开通 |

### 订单推送OnRtnOrder

函数原型：

|  |
| --- |
| void OnRtnOrder  (CStockRtnOrderField \*pRtnOrder,  CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：CStockRtnOrderField

|  |
| --- |
| // 交易所代码  TDAStringType ExchangeNo;  // 合约代码  TDAStringType TreatyCode;  // 定单号  TDAStringType OrderNo;  // 委托数量  TDAIntType OrderNumber;  // 已成交数量  TDAIntType FilledNumber;  // 成交均价  TDAPriceType FilledAdvPrice;  // 持买数量  TDAIntType BuyHoldNumber;  // 持买开仓均价  TDAPriceType BuyHoldOpenPrice;  // 持买均价  TDAPriceType BuyHoldPrice;  // 持卖数量  TDAIntType SaleHoldNumber;  // 持卖开仓均价  TDAPriceType SaleHoldOpenPrice;  // 持卖均价  TDAPriceType SaleHoldPrice;  // 是否已经撤单（0：没有；1：已撤单）  TDAStringType IsCanceled;  // 成交总的手续费  TDAPriceType FilledTotalFee;  // 顺序号  TDAIntType Status;  // 资金帐号  TDAStringType AccountNo;  // 持仓类型（0：昨仓；1：今仓）  TDAStringType HoldType;  // 持买保证金  TDAPriceType HoldMarginBuy;  // 持卖保证金  TDAPriceType HoldMarginSale;  // 最新价  TDAPriceType CurrPrice;  // 浮动盈亏  TDAPriceType FloatProfit; |

参数：CStockRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 成交推送OnRtnTrade

函数原型：

|  |
| --- |
| void OnRtnTrade  (CStockRtnTradeField \*pRtnTrade,  CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) |

参数：CStockRtnTradeField (typedef CStockRspTradeField CStockRtnTradeField)

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 资金账号  TDAStringType AccountNo;  // 成交编号（要包括7位的订单编号，一共11位）  TDAStringType FilledNo;  // 定单号  TDAStringType OrderNo;  // 系统编号  TDAStringType SystemNo;  // 本地编号  TDAStringType LocalNo;  // 交易所代码  TDAStringType ExchangeCode;  // 合约代码  TDAStringType TreatyCode;  // 买还是卖：1=buy 2=sell  TDAStringType BuySale;  // 成交数  TDAStringType FilledNumber;  // 成交价格  TDAStringType FilledPrice;  // 成交日期(yyyy-MM-dd)  TDAStringType FilledDate;  // 成交时间(hh:mm:ss)  TDAStringType FilledTime;  // 成交手续费  TDAStringType Commsion;  // 委托数量  TDAStringType HTSCode;  // 委托价格  TDAStringType ErrorCode;  // 合约交割日期(yyyyMMdd)  TDAStringType DeliveryDate;  // 成交类别(N：普通下单成交；C、T：调期模拟成交)  TDAStringType FilledType;  // 定单类型（1=限价单, 2=市价单，3=限价止损（stop to limit），4=止损（stop to market）TDAStringType OrderType;  // 有效日期（1=当日有效, 2=永久有效）  TDAStringType ValidDate;  // 开仓还是平仓：1=开仓 2=平仓，3=平今，4=平昨  TDAStringType AddReduce;  // 错误信息  TDAStringType ErrorDescription; |

参数：CStockRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 持仓信息推送OnRtnPosition

函数原型：

|  |
| --- |
|  |

参数：OnRtnPosition

|  |
| --- |
|  |

### 资金信息推送~~OnRtnAccount~~ OnRtnCapital

函数原型：

|  |
| --- |
| virtual void OnRtnCapital(CStockRtnCapitalField \*pRtnCapital, CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) {} |

参数：pRtnAccount

|  |
| --- |
| // 客户号  TDAStringType ClientNo;  // 资金帐号  TDAStringType AccountNo;  //币种  TDAStringType CurrencyNo;  //今可用  TDAPriceType Available;  //昨可用  TDAPriceType YAvailable;  //今可出  TDAPriceType CanCashOut;  //今结存  TDAPriceType Money;  //平仓盈亏  TDAPriceType ExpiredProfit;  //冻结资金  TDAPriceType FrozenDeposit;  //手续费  TDAPriceType Fee;  //保证金  TDAPriceType Deposit;  //维持保证金  TDAPriceType KeepDeposit;  //状态  TDAIntType Status;  //入金  TDAPriceType InMoney;  //出金  TDAPriceType OutMoney;  //未到期平盈  TDAPriceType UnexpiredProfit;  //今权益  TDAPriceType TodayTotal;  //未结平盈  TDAPriceType UnaccountProfit;  //期权权利金  TDAPriceType Royalty;  //交易所代码  TDAStringType ExchangeNo;  //合约代码  TDAStringType TreatyCode;  //定单号  TDAStringType OrderNo;  //委托数量  TDAIntType OrderNumber;  //已成交数量  TDAIntType FilledNumber;  //成交均价  TDAPriceType FilledAdvPrice;  //持买数量  TDAIntType BuyHoldNumber;  //持买开仓均价  TDAPriceType BuyHoldOpenPrice;  //持买均价  TDAPriceType BuyHoldPrice;  //持卖数量  TDAIntType SaleHoldNumber;  //持卖开仓均价  TDAPriceType SaleHoldOpenPrice;  //持卖均价  TDAPriceType SaleHoldPrice;  //是否已经撤单（0：没有；1：已撤单）  TDAStringType IsCanceled;  //成交总的手续费  TDAPriceType FilledTotalFee;  //信任额度  TDAPriceType Credit;  //孖展上限额度 20150727 added for 港股  TDAPriceType MarginLimit;  //借货价值 20150727 added for 港股  TDAPriceType BorrowValue;  //按揭价值 20150727 added for 港股  TDAPriceType FMortgageMoney;  // T1 20160219 added for 港股  TDAPriceType T1;  // T2 20160219 added for 港股  TDAPriceType T2;  //T3 20160219 added for 港股  TDAPriceType T3;  // Tn 20160219 added for 港股  TDAPriceType TN;  //交易限额  TDAPriceType TradeLimit;  //跨市场资金限额  TDAPriceType FCrossMax;  //卖空冻结资金  TDAPriceType SellFreezenMoney;  //卖空利息  TDAPriceType SellInterest;  //需补按金  TDAPriceType SellNeedAddMargin; |

### 改单请求/返回ReqOrderModify/OnRspOrderModify方法

函数原型：

|  |
| --- |
| bool ReqOrderModify  (CStockReqOrderModifyField \*pReqOrderModify,int iRequestID); |

参数：pReqOrderModify

|  |
| --- |
| // 系统编号  TDAStringType SystemNo;  // 用户ID  TDAStringType UserId;  // 本地编号  TDAStringType LocalNo;  // 资金账号  TDAStringType AccountNo;  // 定单号  TDAStringType OrderNo;  // 交易所代码  TDAStringType ExchangeCode;  // 合约代码  TDAStringType ContractCode;  // 买还是卖：1=buy 2=sell  TDAStringType BidAskFlag;  // 下单数  TDAStringType OrderQty;  // 下单价格  TDAStringType OrderPrice;  // 改单数  TDAStringType ModifyQty;  // 改单价格  TDAStringType ModifyPrice;  // 美股：1=限价单  // 港股：5=竞价限价盘; 6=竞价盘; 7=增强限价;8=特别限价盘  TDAStringType OrderType;  // 触发价格(股票中未使用)  TDAStringType TriggerPrice;  // 改单触发价格(股票中未使用)  TDAStringType ModifyTriggerPrice;  // 有效日期（1：当日有效）  TDAStringType TIF;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspOrderModify(CStockRspOrderModifyField \*pRspOrderModify, CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

typedef CStockRspOrderInsertField CStockRspOrderModifyField;

具体改单返回结构体字段，请参照报单返回结构体CStockRspOrderInsertField。

参数：CStockRspInfoField \*pRspInfo

|  |
| --- |
| // 错误码(用来判断改单成功或失败)  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 撤单请求/返回ReqOrderCancel/OnRspOrderCancel

函数原型：

|  |
| --- |
| bool ReqOrderCancel  (CStockReqOrderCancelField \*pReqOrderCancel, int iRequestID); |

参数：ReqOrderCancel

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 本地编号  TDAStringType LocalNo;  // 资金账号  TDAStringType AccountNo;  // 系统编号  TDAStringType SystemNo;  // 定单号  TDAStringType OrderNo;  // 交易所代码  TDAStringType ExchangeCode;  // 合约代码  TDAStringType ContractCode;  // 买还是卖：1=buy 2=sell  TDAStringType BidAskFlag;  // 下单数  TDAStringType OrderQty;  // 下单价格  TDAStringType OrderPrice;  // 订单类型  TDAStringType OrderType;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspOrderCancel  (CStockRspOrderCancelField \*pRspOrderCancel, CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) {} |

参数：OnRspOrderCancel

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 资金账号  TDAStringType AccountNo;  // 系统编号  TDAStringType SystemNo;  // 本地号  TDAStringType LocalNo;  // 定单号  TDAStringType OrderNo;  // 撤单号  TDAStringType CancelNo;  // 交易所代码  TDAStringType ExchangeCode;  // 合约代码  TDAStringType ContractCode;  // 买还是卖：1=buy 2=sell  TDAStringType BidAskFlag;  // 下单数  TDAStringType OrderQty;  // 下单价格  TDAStringType OrderPrice;  // 已成交数  TDAStringType FilledQty;  // 已撤单数  TDAStringType CancelledQty;  // 订单类型  TDAStringType OrderType;  // 撤单日期  TDAStringType CancelledDate;  // 撤单时间  TDAStringType CancelledTime;  // 错误代码(保留，未使用)  TDAStringType ErrorCode |
|  |

参数：CStockRspInfoField \*pRspInfo

|  |
| --- |
| // 错误码(用来判断撤单成功或失败)  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 委托单查询/返回ReqQryOrder/OnRspQryOrder

两个委托单查询之间最小间隔时间为1秒，否则查询失败。(Since API v1.5)

函数原型：

|  |
| --- |
| bool ReqQryOrder  (CStockQryOrderField \*pQryOrder, int iRequestID); |

参数：ReqQryOrder

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 资金账号  TDAStringType AccountNo;  // 是否模拟用户：1：是；0 or other：不是  TDAStringType IsSimulation;  // 取得指定订单号以后的定单  TDAStringType OrderNo;  // 取得指定订单时间以后的定单（格式：yyyy-MM-dd hh:mm:ss）  TDAStringType OrderDateTime;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryOrder  (CStockRspOrderField \*pRspOrder,  CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) {} |

参数：OnRspQryOrder

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 资金账号  TDAStringType AccountNo;  // 系统编号  TDAStringType SystemNo;  // 本地编号  TDAStringType LocalNo;  // 定单号  TDAStringType OrderNo;  // 交易所代码  TDAStringType ExchangeCode;  // 合约代码  TDAStringType ContractCode;  // 买还是卖：1=buy 2=sell  TDAStringType BidAskFlag;  // 下单数  TDAStringType OrderQty;  // 下单价格  TDAStringType OrderPrice;  // 已成交数  TDAStringType FilledQty;  // 成交均价  TDAStringType FilledPrice;  // 定单类型  TDAStringType OrderType;  // 下单日期  TDAStringType OrderDate;  // 下单时间  TDAStringType OrderTime;  // 错误代码  TDAStringType ErrorCode;  // 订单状态（1：已请求；2：已排队；3：部分成交；4：完全成交；5：已撤余单；6：已撤单；7：指令失败；8：待送出；9：待更改；A：待撤单）  TDAStringType OrderState;  // 撤单的用户ID  TDAStringType CancelUserId;  // 触发价格(股票未使用)  TDAStringType TriggerPrice;  // 有效日期（1=当日有效）  TDAStringType TIF;  // 卖空单使用  TDAStringType OpenCloseFlag;  // 策略ID (股票未使用)  TDAStringType StrategyId;  // 显示委托量，必须小于委托量(股票未使用)  TDAStringType MaxShow;  // 最小成交量(股票未使用)  TDAStringType MinQty;  // 交易所返回时间  TDAStringType ExchangeTime;  // 撤单时间  TDAStringType CancelTime; |

### 成交查询请求/返回ReqQryTrade/OnRspQryTrade

两个成交查询之间最小间隔时间为1秒，否则查询失败。(Since API v1.5)

函数原型：

|  |
| --- |
| bool ReqQryTrade  (CStockQryTradeField \*pQryTrade, int iRequestID) = 0; |

参数：ReqQryTrade

|  |
| --- |
| // 用户  TDAStringType UserId;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryTrade(CStockRspTradeField \*pRspTrade, CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：CStockRspTradeField

|  |
| --- |
| 请参照OnRtnTrade数据返回字段 |

### 持仓明细查询请求/返回ReqQryPosition/OnRspQryPosition

两个持仓查询之间最小间隔时间为1秒，否则查询失败。(Since API v1.5)

函数原型：

|  |
| --- |
| bool ReqQryPosition  (CStockQryPositionField \*pQryPosition, int iRequestID) = 0; |

参数：ReqQryPosition

|  |
| --- |
| // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryPosition  (CStockRspPositionField \*pRspPosition,  CStockRspInfoField \*pRspInfo,  int iRequestID, bool bIsLast) {} |

参数：OnRspQryPosition

|  |
| --- |
| // 用户ID  TDAStringType ClientNo;  // 交易所  TDAStringType ExchangeNo;  // 证券代码  TDAStringType ProductCode;  // 持仓方向（1：持买；2：持卖）  TDAStringType LongShortPosFlag;  // 持仓成本价  TDADoubleType PosCostPrice;  // 可卖数量  TDAIntType CanSellShares;  // 今买数量  TDAIntType TodayBuyShares;  // 冻结数量  TDAIntType FrosenShares;  // 持有期内买入金额总和  TDADoubleType TotalBuyMoney;  // 持有期内卖出金额总和  TDADoubleType TotalSellMoney;  // 持有期内买入数量  TDAIntType TotalBuyShares;  // 持有期内卖出数量  TDAIntType TotalSellShares;  // 首次开仓日期(yyyy-MM-dd)  TDAStringType FirstPosDate;  // 平仓盈亏  TDADoubleType ClosePosPL;  // T1日交割股数，正入负出  TDAIntType T1DeliveryShares;  // T2日交割股数，正入负出  TDAIntType T2DeliveryShares;  // T3日交割股数，正入负出  TDAIntType T3DeliveryShares;  // 未交割股数=T1+T2  TDAIntType NotDeliveryShares;  // 已交割股数  TDAIntType DeliveredShares;  // 卖空股数  TDAIntType SellShortShares;  // 卖空冻结资金  TDADoubleType SellShortMoney;  // 卖空仓位的开仓均价(开仓均价)  TDADoubleType SSPosAvgCostPrice; |

### 资金查询请求/返回ReqQryCapital/OnRspQryCapital

两个资金查询之间最小间隔时间为1秒，否则查询失败。(Since API v1.5)

函数原型：

|  |
| --- |
| bool ReqQryCapital  (CStockQryCapitalField \*pQryCapital,  int iRequestID) = 0; |

参数：ReqQryCapital

|  |
| --- |
| // 不使用  TDACharType Unused;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRsqQryCapital(CStockRspCapitalField \*pRspCapital, CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) {} |

参数：CStockRspCapitalField

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 入金  TDAStringType Deposit;  // 出金  TDAStringType Withdraw;  // 今日可用于交易的资金量(今可用)  TDAStringType TodayTradableFund;  // 当日期初权益(今结存)  TDAStringType TodayInitialBalance;  // 当日实时浮动权益(今权益)  TDAStringType TodayRealtimeBalance;  // 冻结资金  TDAStringType FrozenFund;  // 佣金  TDAStringType Commission;  // 初始保证金  TDAStringType InitialMargin;  // 昨日可用于交易的资金量(昨可用)  TDAStringType YdTradableFund;  // 昨日期初权益(昨结存)  TDAStringType YdInitialBalance;  // 昨日期末权益(昨权益)  TDAStringType YdFinalBalance;  // 浮动盈亏  TDAStringType ProfitLoss;  // 币种编号  TDAStringType CurrencyNo;  // 货币与基本的汇率  TDAMoneyType CurrencyRate;  // LME未到期平盈(未到期平盈) (股票中不使用)  TDAMoneyType LMEUnexpiredPL;  // LME未结平盈(未结平盈) (股票中不使用)  TDAMoneyType LMEUnaccountPL;  // 维持保证金  TDAMoneyType MaintenanceMargin;  // 期权权利金  TDAMoneyType Premium;  // 信任额度  TDAMoneyType CreditAmount;  // 初始资金  TDAMoneyType IntialFund;  // 资金帐号 1  TDAStringType FundAccountNo;  // 按揭价值  TDAStringType MortgageInMoney;  // 孖展上限额度  TDAMoneyType MarginLimit;  // 借货价值  TDAMoneyType BorrowInMoney;  // T1日交割的资金，正入负出  TDAMoneyType T1DeliveryMoney;  // T2日交割的资金，正入负出  TDAMoneyType T2DeliveryMoney;  // T3日交割的资金，正入负出  TDAMoneyType T3DeliveryMoney;  // TN日交割的资金，正入负出  TDAMoneyType TNDeliveryMoney;  //交易限额  TDAMoneyType TradeLimit;  // 可取资金==min(T0,T0+T1,T0+T1+T2)-冻结资金(挂单)  TDAMoneyType CanCashOutMoneyAmount;  // 月存款利息  TDAMoneyType DepositInterest;  // 月欠款利息  TDAMoneyType LoanInterest;  // 跨市场资金限额  TDAMoneyType CrossCurrencyMaxMoneyAmt;  // 卖空冻结资金  TDAMoneyType SellShortFrozenMoney;  // 卖空利息  TDAMoneyType SellShortInterest;  // 需补按金(卖空持仓时)  TDAMoneyType ShortPosAddtionalMargin;  // 错误信息  TDAStringType ErrorDescription; |

### 经纪商查询请求/返回ReqQryBroker/OnRspQryBroker

函数原型：

|  |
| --- |
|  |

参数：

|  |
| --- |
|  |

函数原型：

|  |
| --- |
|  |

参数：

|  |
| --- |
|  |

### 版本号查询请求/返回ReqQryVersion/OnRspQryQryVersion

函数原型：

|  |
| --- |
| bool ReqQryVersion  (CStockQryVersionField \*pQryVersion, int iRequestID) = 0; |

参数：ReqQryVersion

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 用户密码  TDAStringType UserPwd;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryVersion(CStockRspVersionField \*pRspVersion, CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) {} |

参数：CStockRspVersionField

|  |
| --- |
| // 版本号  TDAStringType Version;  // 是否必须更新到新版本才能正常使用（0：可以不更新；1：必须更新）  TDAStringType MustUpdate;  // 必须要更到的版本号  TDAStringType MustVersion;  // 版本内容中文  TDAStringType VersionContent\_CN;  // 版本内容英文  TDAStringType VersionContent\_US; |

参数：CStockRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 币种查询请求/返回ReqQryCurrency/ OnRspQryCurrency

函数原型：

|  |
| --- |
| bool ReqQryCurrency  (CStockQryCurrencyField \*pQryCurrency, int iRequestID) = 0; |

参数：ReqQryCurrency

|  |
| --- |
| // 不使用  TDACharType Unused;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryCurrency  (CStockRspCurrencyField \*pRspCurrency,  CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) {} |

参数：OnRspQryCurrency

|  |
| --- |
| // 货币编号  TDAStringType CurrencyNo;  // 基币货币编号  TDAIntType IsBase;  // 与基币的换算汇率  TDADoubleType ChangeRate;  // 货币名称  TDAStringType CurrencyName;  // 货币名称（英文）  TDAStringType CurrencyNameEN; |

### 交易时间查询请求/返回ReqQryTradeTime/ OnRspQryTradeTime

函数原型：

|  |
| --- |
| bool ReqQryTradeTime  (CStockQryTradeTimeField \*pQryTradeTime, int iRequestID) = 0; |

参数：ReqQryTradeTime

|  |
| --- |
| // 不使用  TDACharType Unused;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryTradeTime  (CStockRspTradeTimeField \*pRspTradeTime,  CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) {} |

参数：OnRspQryTradeTime

|  |
| --- |
| // 当前交易日期（yyyyMMdd）  TDAStringType TradeDate;  // 开盘时间（HH:mm:ss）  TDAStringType OpenTime;  // 收盘时间（HH:mm:ss  TDAStringType CloseTime; |

### 交易所查询请求/返回ReqQryExchange /OnRspQryExchange

函数原型：

|  |
| --- |
| bool ReqQryExchange  (CStockQryExchangeField \*pQryExchange, int iRequestID) = 0; |

参数：ReqQryExchange

|  |
| --- |
| TDAStringType ProductGroupID;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryExchange(CStockRspExchangeField \*pRspExchange, CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) |

参数：CStockRspExchangeField

|  |
| --- |
| // 交易所编号  TDAStringType ExchangeNo;  // 交易所名称  TDAStringType ExchangeName;  // 交易所类型（HK：港股；US：美股）  TDAStringType SettleType;  // 交易所名称（英文）  TDAStringType NameEN; |

参数：CStockRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 合约查询请求/返回ReqQryInstrument/OnRspQryInstrument

函数原型：

|  |
| --- |
| bool ReqQryInstrument  (CStockQryInstrumentField \*pQryInstrument, int iRequestID) = 0; |

参数：ReqQryInstrument

|  |
| --- |
| // 不使用  TDACharType Unused;  // 查询的数量  TDAIntType Index;  // 修改时间，增量查询  TDAStringType ModifyDay;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryInstrument  (CStockRspInstrumentField \*pRspInstrument,  CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) {} |

参数：OnRspQryInstrument

|  |
| --- |
| // 交易所编号  TDAStringType ExchangeNo;  // 交易所名称  TDAStringType ExchangeName;  // 合约NO  TDAStringType CommodityNo;  // 商品名  TDAStringType CommodityName;  // 商品类别， 0：股票，1：窝轮及牛熊证，2：认股权，3：ETF，4：债券，5：信托，6：其他，8：指数  TDAStringType CommodityType;  // 货币编号  TDAStringType CurrencyNo;  // 货币名称  TDAStringType CurrencyName;  // 点值（一个最小跳点的价值）  TDADoubleType ProductDot;  // 最小变动单位  TDADoubleType UpperTick;  // 上日结算价  TDADoubleType SettlePrice;  // 交易月 (yyyyMM)/交割日 (yyyyMMdd)  TDAStringType TradeMonth;  // 行情小数点位数  TDAIntType DotNum;  // 进阶单位  TDAIntType LowerTick;  // 调期小数点位数  TDAIntType DotNum\_Carry;  // 调期最小变动单位  TDADoubleType UpperTick\_Carry;  // 首次通知日 (yyyyMMdd)  TDAStringType FirstNoticeDay;  // 冻结保证金百分比  TDADoubleType FreezenPercent;  // 冻结保证金固定值  TDADoubleType FreezenMoney;  // 固定手续费  TDADoubleType FeeMoney;  // 百分比手续费  TDADoubleType FeePercent;  // 现货商品昨结算价  TDADoubleType PriceStrike;  // 现货商品点值  TDADoubleType ProductDotStrike;  // 现货商品最小变动单位  TDADoubleType UpperTickStrike;  // 最后交易日 (yyyyMMdd)(期货专用)  TDAStringType LastTradeDay;  // 最后更新日 (yyyyMMdd)(期货专用)  TDAStringType LastUpdateDay;  // 期权临界价格 (期货专用)  TDADoubleType CriticalPrice;  // 期权临界价格以下的最小跳点 (期货专用)  TDADoubleType CriticalMinChangedPrice;  // 实际交易所(CME细分成3个交易所:CME,CME\_COMEX,CME\_NYMEX)(期货专用)  TDAStringType Exchange2;  // 期权类型(R：看涨；F：看跌)(期货专用)  TDAStringType OptionType;  // 期权年月(yyyyMM)(期货专用)  TDAStringType OptionMonth;  // 期权执行价格(期货专用)  TDAStringType OptionStrikePrice;  // 期权对应期货商品编号(期货专用)  TDAStringType OptionCommodityNo;  // 期权对应期货合约编号(期货专用)  TDAStringType OptionContractNo;  // 按揭价值百分比  TDAStringType MortgagePercent;  // 最小变动单位代码  TDAStringType UpperTickCode;  // 最小批量  TDAStringType LotSize;  // 平仓时间标志(控制T+1：0代表T+0，可以当天卖出平仓；1代表T+1，必须隔天卖出平仓；N代表T+N)  TDAStringType FlatTime;  // 商品名（英文）  TDAStringType CommodityFNameEN;  // 是否支持卖空（0或空：不可以；1：可以）  TDAStringType CanSell;  // 卖空时的按金比例（例：0.5）  TDADoubleType SellRate;  // 卖空持仓上限额  TDADoubleType SellMax;  // 换股比率，表示一份窝轮牛熊証，相当于多少正股  TDADoubleType StrikeRate;  // 行使价  TDADoubleType StrikePrice;  // 收回价  TDADoubleType ReceivePrice;  // 到期日（yyyy-MM-dd）  TDAStringType ExpireDate;  // 卖空时的维持按金比例（例：0.5）  TDADoubleType SellRateKeep;  // 涡轮牛熊证对应正股编号  TDAStringType StrikeCommodityNo;  // 涡轮牛熊证的涨跌标志(C:涨；P:跌)  TDAStringType CallPutFlag;  // 涡轮牛熊证的发行人  TDAStringType Publisher; |

### 跳点查询请求/返回ReqQryTick/ OnRspQryTick

函数原型：

|  |
| --- |
| bool ReqQryTick  (CStockQryTickField \*pQryTick, int iRequestID) = 0; |

参数：ReqQryTick

|  |
| --- |
| // 不使用  TDACharType Unused;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryTick  (CStockRspTickField \*pRspTick,  CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) {} |

参数：OnRspQryTick

|  |
| --- |
| // 最小变动单位代码  TDAStringType UpperTickCode;  // 最小变动单位开始行情价格  TDAStringType PriceFrom;  // 行情最小变动单位  TDAStringType UpperTick;  // 最小变动单位对应的点值  TDAStringType ProductDot;  // 行情小数位数  TDAStringType DotNum;  // 进价单位  TDAStringType LowerTick; |

### 订单类型查询请求/返回ReqQryOrderType/ OnRspQryOrderType

函数原型：

|  |
| --- |
|  |

参数：ReqQryOrderType

|  |
| --- |
|  |

函数原型：

|  |
| --- |
|  |

参数：OnRspQryOrderType

|  |
| --- |
|  |

### 设置双重认证密码/返回ReqSetVerifyQA/ OnRspSetVerifyQA

函数原型：

|  |
| --- |
| bool ReqSetVerifyQA(CFutureReqSetVerifyQAField \*pReqSetVerifyQA, int iRequestID); |

参数：CFutureReqSetVerifyQAField

|  |
| --- |
| // 用户帐号  TDAStringType UserId;  // 交易密码  TDAStringType UserPwd;  // 固定值，“1”  TDAStringType Type;  // 固定值，“1”  TDAStringType Question;  // 双重认证密码，英数字  TDAStringType Answer;  // 固定值，“1”  TDAStringType SaveMac;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspSetVerifyQA(CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：CFutureRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

回调函数错误码指示成功时，调用ReqSafeVerify()函数绑定网卡地址。

### 绑定双重认证需要的MAC地址/返回ReqSafeVerify/ OnRspSafeVerify

函数原型：

|  |
| --- |
| bool ReqSafeVerify(CFutureReqSafeVerifyField \*pReqSafeVerify, int iRequestID); |

参数：CFutureReqSafeVerifyField

|  |
| --- |
| // 用户帐号  TDAStringType UserId;  // 交易密码  TDAStringType UserPwd;  // 固定值，“1”  TDAStringType Type;  // 固定值，“1”  TDAStringType Question;  // 双重认证密码，英数字  TDAStringType Answer;  // 固定值，“1”  TDAStringType SaveMac;  // MAC地址  TDAStringType MacAddress;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspSafeVerify(CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：CFutureRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

# 期货交易API接口参考

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 业务类型 | 业务 | 请求接口 |  | 响应接口 |
| 连接 | 连接 | Init | 连接回报 | OnFrontConnected |
| 断开 |  |  | 连接断开回报 | OnFrontDisconnected |
| 心跳 |  |  | 心跳警告 | OnHeartBeatWarning |
| 登录 | 登录 | ReqUserLogin | 登录回报 | OnRspUserLogin/OnRspNeedVerify |
| 登出 | ReqUserLogout | 登出回报 | OnRspUserLogout |
| 修改密码 | ReqUserPasswordUpdate | 修改密码回报 | OnRspUserPasswordUpdate |
| 双重认证step1 | ReqSetVerifyQA | 设置双重认证密码 | OnRspSetVerifyQA |
| 双重认证step2 | ReqSafeVerify | 绑定MAC地址 | OnRspSafeVerify |
| 交易 | 报单 | ReqOrderInsert | 报单回报 | OnRspOrderInsert |
| 改单 | ReqOrderModify | 改单回报 | OnRspOrderModify |
| 撤单 | ReqOrderCancel | 撤单回报 | OnRspOrderCancel |
| 推送 |  |  | 报单状态推送 | OnRtnOrder |
|  |  | 成交信息推送 | OnRtnTrade |
|  |  | *持仓信息推送* | *OnRtnPosition* |
|  |  | 资金信息推送 | OnRtnAccount |
| 查询 | 委托单查询 | ReqQryOrder | 委托单查询回报 | OnRspQryOrder |
| 成交查询 | ReqQryTrade | 成交查询回报 | OnRspQryTrade |
| 持仓明细查询 | ReqQryPosition | 持仓明细查询回报 | OnRspQryPosition |
| 持仓总结查询 | ReqQryTotalPosition | 持仓总结查询回报 | OnRspQryTotalPosition |
| 资金查询 | ReqQryCapital | 资金查询回报 | OnRspQryCapital |
| 版本号查询 | ReqQryVersion | 版本号查询回报 | OnRspQryQryVersion |
| 币种查询 | ReqQryCurrency | 币种查询回报 | OnRspQryCurrency |
| 交易时间查询 | ReqQryTradeTime | 交易时间查询回报 | OnRspQryTradeTime |
| 策略查询 | ReqQryStrategy | 策略查询回报 | OnRspQryStrategy |
| 交易所查询 | ReqQryExchange | 交易所查询回报 | OnRspQryExchange |
| 合约查询 | ReqQryInstrument | 合约查询回报 | OnRspQryInstrument |
| 策略明细查询 | ReqQryStrategyDetail | 策略明细查询回报 | OnRspQryStrategyDetail |
|  | 交易所冬夏令开盘时间查询(from v1.10) | ReqQryExchangeTime | 查询回报 | OnRspQryExchangeTime |
|  | 交易品种开收盘时间查询(from v1.10) | ReqQryCommodityTime | 查询回报 | OnRspQryCommodityTime |
|  | 查询品种列表(from v1.10) | ReqQryCommodityTime | 查询回报 | OnRspQryCommodity |

注：以上斜体字的接口，当前API版本未实现。

## 接口工作过程



## 接口模式

期货API 的DAFutureApi.h头文件中提供了二个接口，分别为CFutureApi和IFutureEvent，方便客户端应用程序的开发。客户端应用程序可以通过CFutureApi发出操作请求，通过继承IFutureEvent并重载回调函数来处理后台服务的响应。

## 函数接口说明

### CreateFutureApi方法

创建实例方法。

**函数原形**：

static CFutureApi \*CreateFutureApi(bool bRecordLog, const char\* lpszLogFileName = "Future.log",const char\*lpszLogFileDir="");

bRecordLog: true表示记录日志，false表示不记录日志。

lpszLogFileName：日志文件。

lpszLogFileDir： 日志文件文件夹路径。

### OnFrontConnected 方法

当客户端与前置建立起通信连接时（还未登录前），该方法被调用。

**函数原形：**

void OnFrontConnected()；

本方法在完成初始化后调用，可以在其中完成用户登录任务。

### OnFrontDisconnected方法

当客户端与前置通信连接断开时，该方法被调用。当发生这个情况

后，API会自动重新连接，客户端可不做处理。自动重连地址，可能是原来注册

的地址，也可能是系统支持的其它可用的通信地址，它由程序自动选择。

**函数原形：**

void OnFrontDisconnected (int nReason)；

**参数：**

nReason：连接断开原因

### OnHeartBeatWarning方法

心跳超时警告。当长时间未收到报文时，该方法被调用。

**函数原形：**

void OnHeartBeatWarning(int nTimeLapse)；

**参数：**

nTimeLapse：距离上次接收报文的时间

### 登录ReqUserLogin/OnRspUserLogin/OnRspNeedVerify/OnRspAccount

从API v1.5引入双重认证，具体交互过程请参阅**“国际期货股票API\_双重认证交互说明.xlsx”**

函数原型：

|  |
| --- |
| bool ReqUserLogin  (CFutureReqUserLoginField \*pReqUserLogin, int iRequestID); |

参数：CFutureReqUserLoginField

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 用户密码  TDAStringType UserPwd;  // 用户类型  TDAStringType UserType;  // 软件名称  TDAStringType SoftwareName;  // 软件版本号  TDAStringType SoftwareVersion;  // 授权码  TDAStringType AuthorCode;  //登录机器网卡MAC地址(since API v1.5,双重认证使用)  MacAddress  // 错误信息  TDAStringType ErrorDescription; |

1) 函数原型：

|  |
| --- |
| void OnRspUserLogin  (CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：CFutureRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

错误码：

|  |  |
| --- | --- |
| **登录错误码** | **错误内容** |
| 10001 | 用户名不正确 |
| 10002 | 登录密码错误 |
| 10003 | 密码错误次数超限，用户已冻结，请联系客服解冻！ |
| 10004 | 用户已被冻结，请联系客服解冻！ |
| 10005 | 授权码信息不正确，请确认UserType，SoftwareName，SoftwareVersion，AuthorCode的值 |

2) 函数原型：

服务器端在判断到需要做双重认证时，会指示API回调此函数

|  |
| --- |
| void OnRspNeedVerify(bool bFirstLogin, bool bHasSetQA); |

参数：bFirstLogin

|  |
| --- |
| 当设备(根据login消息中的MacAddress字段识别)第一次登录交易前置时， OnRspNeedVerify()函数被回调，且bFirstLogin 的值为true,用户在回调函数中需调用ReqSetVerifyQA()函数设置双重认证信息。  当设备(根据login消息中的MacAddress字段识别)第一次以后登录交易前置时，OnRspNeedVerify()函数不会被回调。 |

1. 函数原型：

在OnRspUserLogin指示成功登录回调后，该函数被回调，CFutureRspAccountField结构体中返回每个币种对应的资金账号

|  |
| --- |
| void OnRspAccount(CFutureRspAccountField \*pRspAccount, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：pRspAccount

|  |
| --- |
| // 帐户币种  TDAStringType CurrencyNo;  // 资金账号  TDAStringType AccountNo; |

### 登出 ReqUserLogout/OnRspUserLogout方法

函数原型：

|  |
| --- |
| bool ReqUserLogout  (CFutureReqUserLogoutField \*pReqUserLogout, int iRequestID); |

参数：CFutureReqUserLogoutField

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 资金账号  TDAStringType AccountNo;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspUserLogout  (CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：CFutureRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 修改密码ReqPasswordUpdate/OnRspPasswordUpdate

函数原型：

|  |
| --- |
| Bool ReqPasswordUpdate (CFutureReqPasswordUpdateField \*pReqPasswordUpdate, int iRequestID); |

参数：CFutureReqPasswordUpdateField

|  |
| --- |
| // 用户代码  TDAStringType UserId;  // 用户原密码  TDAStringType OldPassword;  // 用户新密码  TDAStringType NewPassword;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspPasswordUpdate (CFutureRspPasswordUpdateField \*pRspPasswordUpdate, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) ; |

参数：CFutureRspPasswordUpdateField

|  |
| --- |
| // 用户代码  TDAStringType UserId;  // 用户原密码  TDAStringType OldPassword;  // 用户新密码  TDAStringType NewPassword; |

参数：CFutureRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 报单ReqOrderInsert/OnRspOrderInsert

说明：

TDAStringType SystemNo; // 系统编号

TDAStringType LocalNo; // 本地编号

TDAStringType OrderNo; // 定单号

LocalNo是使用API的程序给的订单号,相当于CTP里面的orderref，当日内最好不要重复

SystemNo是直达系统给的订单号；

OrderNo是交易所给的订单号，相当于CTP里面的OrderSysID

函数原型：

|  |
| --- |
| bool ReqOrderInsert(CFutureReqOrderInsertField \*pReqOrderInsert, int iRequestID) |

参数：CFutureReqOrderInsertField

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 资金账号(一个用户可以有多个币种的资金，所以会有多个资金账号，下单需指定)  // 请参照登录成功后的回调函数OnRspAccount的说明  TDAStringType AccountNo;  // 本地编号  TDAStringType LocalNo;  // 交易所代码  TDAStringType ExchangeCode;  // 合约代码  TDAStringType ContractCode;  // 买还是卖：1=buy 2=sell  TDAStringType BidAskFlag;  // 开仓还是平仓：1=开仓 2=平仓，3=平今  TDAStringType OpenCloseFlag;  // 下单数  TDAStringType OrderQty;  // 下单价格  TDAStringType OrderPrice;  // 定单类型：1=限价单, 2=市价单，3=限价止损（stop to limit），4=止损（stop to market)  TDAStringType OrderType;  // 触发价格  TDAStringType TriggerPrice;  // 有效日期（1=当日有效, 2=永久有效（GTC），4=IOC）  TDAStringType TIF;  // 策略ID  TDAStringType StrategyId;  //冰山单使用：显示委托量 必须小于委托量  TDAStringType MaxShow;  //IOC单使用： 最小成交量  //CME交易所：有效日期=4IOC时, MinQty >=1且小于委托量时是FAK，MinQty=委托量时是FOK  //港交所，新交所：MinQty =1时FAK单，MinQty=委托量时FOK单，不能为其它值  TDAStringType MinQty;  // Tag50(订单受益人/所有者)  TDAStringType Tag50;  // 错误信息  TDAStringType ErrorDescription; |
| // 程序化单(0)或人工单(1)  TDAIntType IsProgram;  // (\*4) ISO(3166-1)标准中定义的国家/地区两个字母代码  TDAStringType OrgOrderLocationID; |

说明：

1：API用户下单后，OnRspOrderInsert()函数会被回调两次：

1. 第一次回调返回直达系统给这笔订单指定的系统号SystemNo，表明直达系统收到并确认这笔订单。
2. 第二次回调返回交易所的反馈的订单号OrderNo(如果一切正常)。通常资金不足等情况发生时，不会有订单号，因为订单指令没有向交易所发送。

2：tag50(订单操作人/所有者)字段说明，下改撤单中tag50类同

1. tag50字段适用于cme旗下交易所,非CME交易所的下单无需设置tag50.(关于tag50下单规则方面有任何疑问,欢迎联系经纪商业务人员咨询.)
2. 如客户长时间交易，例如超过12小时，需设置多轮管理、操作和监控（至少设置2 个班次）。当负责管理、操作和监控的人员发生变化时，新一班次人员需分配不同于前一班次的Tag50，并在订单中传输。
3. Tag50(订单操作人/所有者) 生产环境中，分两种情况：
   1. API不设置或空值的情况下，默认使用客户号作为Tag50（客户号作为Tag50无需向经纪公司申报）。
   2. 上述a)以外，须提前向经纪公司申报Tag50信息，由经纪公司设置Tag50之后，才可在API中使用；API使用任何未经申报的Tag50下单将会被拒绝。客户需确保实际下单传输Tag50与API申请时申报Tag50信息一致。  
      Tag50值的分配逻辑是客户号+‘\_001‘，客户号+’\_002’，依次类推；

3: IsProgram：区分手工单或者程序化单。

API中默认为程序化单，客户需按照实际情况设置0（程序化单）或1（手工单）。

4: OrgOrderLocationID

只有该API账号下还有下层多个不同客户交易时(API程序用作订单路由网关)，才需要填写，分两种填写方式：

国际/地区代码为加拿大时，后面紧跟逗号，再跟省代码，总共5个ASCII字符。

例：加南大魁北克省对应CA,QC

国际/地区地区代码为加拿大以外时，仅填写国际/地区代码，总共2个ASCII字符。

例：Japan对应JP, China对应CN

函数原型：

|  |
| --- |
| Void OnRspOrderInsert(CFutureRspOrderInsertField \*pRspOrderInsert, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：CFutureRspOrderInsertField

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 资金账号  TDAStringType AccountNo;  // 系统编号  TDAStringType SystemNo;  // 本地编号  TDAStringType LocalNo;  // 定单号  TDAStringType OrderNo;  // 交易所代码  TDAStringType ExchangeCode;  // 合约代码  TDAStringType ContractCode;  // 买还是卖：1=buy 2=sell  TDAStringType BidAskFlag;  // 下单数  TDAStringType OrderQty;  // 下单价格  TDAStringType OrderPrice;  // 定单类型：1=限价单, 2=市价单，3=限价止损（stop to limit）  TDAStringType OrderType;  // 下单日期  TDAStringType OrderDate;  // 下单时间  TDAStringType OrderTime;  // 错误代码(保留，未使用)  TDAStringType ErrorCode;  // 订单状态（2：已排队；3：部分成交；4：完全成交；5：已撤余单；6：已撤单；7：指令失败）  TDAStringType OrderState;  // 触发价格  TDAStringType TriggerPrice;  // 有效日期（1=当日有效, 2=永久有效（GTC），4=IOC）  TDAStringType TIF;  // 开仓还是平仓：1=开仓 2=平仓，3=平今，4=平昨  TDAStringType OpenCloseFlag;  // 策略ID  TDAStringType StrategyId;  // 冰山单：显示委托量 必须小于委托量  TDAStringType MaxShow;  // 最小成交量  TDAStringType MinQty;  // 交易所返回时间  TDAStringType ExchangeTime;  // 终端类型(CME交易所特定)  TDAStringType OrdSourceType;  // Tag50(订单受益人/所有者),格式为“机构通ID\_子账户”  TDAStringType Tag50; |

参数：CFutureRspInfoField

|  |
| --- |
| // 错误码(用来判断报单成功或失败)  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

错误码

|  |  |
| --- | --- |
| **系统号错误码** | **错误内容** |
| 20005 | 系统号生成失败 |

下单错误码

|  |  |
| --- | --- |
| **下单错误码** | **错误内容** |
| 20000 | 下单失败 |
| 20001 | 资金不足 |
| 20002 | 交易服务器未连接 |
| 20003 | 您已被禁止交易，请联系客服解禁 |
| 20004 | 下单被拒绝 |
| 20005 | 系统号生成失败 |
| 20006 | 您的资金账户中没有该合约交易所需的币种 |
| 20007 | 该合约已经到期，不能交易 |
| 20008 | 该交易市场未开市，不能交易 |
| 20009 | 该交易市场已闭市，不能交易 |
| 20010 | 下单价格超出限定范围 |
| 20011 | 下单数量过大 |
| 20012 | 您被禁止交易该合约，请联系客服 |
| 20013 | 您的资金账户风险率过高，已被禁止交易 |
| 20015 | 您的账户交易数据出现异常，暂停交易 |
| 20019 | 下单数量超过客户持仓限量 |
| 20020 | 下单数量超过客户可下单买量 |
| 20021 | 下单数量超过客户可下单卖量 |
| 20022 | 下单数量超过公司持仓限量 |
| 20023 | 下单数量超过公司可下单买量 |
| 20024 | 下单数量超过公司可下单卖量 |
| 20025 | Dealer下单数量超过最大可下单数量 |
| 20026 | 合约临近交割，只可平仓，不能开仓 |
| 20027 | 该合约已过最后交易日时间，禁止电子盘交易 |
| 20028 | 商品未开放，禁止交易 |
| 20029 | 单笔下单数量超过该商品单笔最大可下单数 |
| 20032 | 持仓不足，平仓失败 |
| 20033 | 您不能交易该商品,请联系客服开通 |

### 订单推送OnRtnOrder

函数原型：

|  |
| --- |
| void OnRtnOrder(CFutureRtnOrderField \*pRtnOrder, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：CFutureRtnOrderField

|  |
| --- |
| // 交易所代码  TDAStringType ExchangeNo;  // 合约代码  TDAStringType TreatyCode;  // 定单号  TDAStringType OrderNo;  // 委托数量  TDAIntType OrderNumber;  // 已成交数量  TDAIntType FilledNumber;  // 成交均价  TDAPriceType FilledAdvPrice;  // 持买数量  TDAIntType BuyHoldNumber;  // 持买开仓均价  TDAPriceType BuyHoldOpenPrice;  // 持买均价  TDAPriceType BuyHoldPrice;  // 持卖数量  TDAIntType SaleHoldNumber;  // 持卖开仓均价  TDAPriceType SaleHoldOpenPrice;  // 持卖均价  TDAPriceType SaleHoldPrice;  // 是否已经撤单（0：没有；1：已撤单）  TDAStringType IsCanceled;  // 成交总的手续费  TDAPriceType FilledTotalFee;  // 顺序号  TDAIntType Status;  // 资金帐号  TDAStringType AccountNo;  // 持仓类型（0：昨仓；1：今仓）  TDAStringType HoldType;  // 持买保证金  TDAPriceType HoldMarginBuy;  // 持卖保证金  TDAPriceType HoldMarginSale;  // 最新价  TDAPriceType CurrPrice;  // 浮动盈亏  TDAPriceType FloatProfit; |

参数：CFutureRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 成交推送OnRtnTrade

函数原型：

|  |
| --- |
| void OnRtnTrade(CFutureRtnTradeField \*pRtnTrade, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) |

参数：CFutureRtnTradeField (typedef CFutureRspTradeField CFutureRtnTradeField;)

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 资金账号  TDAStringType AccountNo;  // 成交编号（要包括7位的订单编号，一共11位）  TDAStringType FilledNo;  // 定单号  TDAStringType OrderNo;  // 系统编号  TDAStringType SystemNo;  // 本地编号  TDAStringType LocalNo;  // 交易所代码  TDAStringType ExchangeCode;  // 合约代码  TDAStringType TreatyCode;  // 买还是卖：1=buy 2=sell  TDAStringType BuySale;  // 成交数  TDAStringType FilledNumber;  // 成交价格  TDAStringType FilledPrice;  // 成交日期(yyyy-MM-dd)  TDAStringType FilledDate;  // 成交时间(hh:mm:ss)  TDAStringType FilledTime;  // 成交手续费  TDAStringType Commsion;  // 委托数量  TDAStringType OrderNumber;  // 委托价格  TDAStringType OrderPrice;  // 合约交割日期(yyyyMMdd)  TDAStringType DeliveryDate;  // 成交类别(N：普通下单成交；C、T：调期模拟成交)  TDAStringType FilledType;  // 定单类型（1=限价单, 2=市价单，3=限价止损（stop to limit），4=止损（stop to market）  TDAStringType OrderType;  // 有效日期（1=当日有效, 2=永久有效）  TDAStringType ValidDate;  // 开仓还是平仓：1=开仓 2=平仓，3=平今，4=平昨  TDAStringType AddReduce; |

### 持仓信息推送OnRtnPosition

### 资金信息推送~~OnRtnAccount~~ OnRtnCapital

函数原型：

|  |
| --- |
| virtual void OnRtnCapital(CFutureRtnCapitalField \*pRtnCapital, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) |

参数：pRtnCapital

|  |
| --- |
| // 客户号  TDAStringType ClientNo;  // 资金帐号  TDAStringType AccountNo;  //币种  TDAStringType CurrencyNo;  //今可用  TDAPriceType Available;  //昨可用  TDAPriceType YAvailable;  //今可出  TDAPriceType CanCashOut;  //今结存  TDAPriceType Money;  //平仓盈亏  TDAPriceType ExpiredProfit;  //冻结资金  TDAPriceType FrozenDeposit;  //手续费  TDAPriceType Fee;  //保证金  TDAPriceType Deposit;  //维持保证金  TDAPriceType KeepDeposit;  //状态  TDAIntType Status;  //入金  TDAPriceType InMoney;  //出金  TDAPriceType OutMoney;  //未到期平盈  TDAPriceType UnexpiredProfit;  //今权益  TDAPriceType TodayTotal;  //未结平盈  TDAPriceType UnaccountProfit;  //期权权利金  TDAPriceType Royalty;  //交易所代码  TDAStringType ExchangeNo;  //合约代码  TDAStringType TreatyCode;  //定单号  TDAStringType OrderNo;  //委托数量  TDAIntType OrderNumber;  //已成交数量  TDAIntType FilledNumber;  //成交均价  TDAPriceType FilledAdvPrice;  //持买数量  TDAIntType BuyHoldNumber;  //持买开仓均价  TDAPriceType BuyHoldOpenPrice;  //持买均价  TDAPriceType BuyHoldPrice;  //持卖数量  TDAIntType SaleHoldNumber;  //持卖开仓均价  TDAPriceType SaleHoldOpenPrice;  //持卖均价  TDAPriceType SaleHoldPrice;  //是否已经撤单（0：没有；1：已撤单）  TDAStringType IsCanceled;  //成交总的手续费  TDAPriceType FilledTotalFee;  //信任额度  TDAPriceType Credit;  //孖展上限额度 for 港股  TDAPriceType MarginLimit;  //借货价值 for 港股  TDAPriceType BorrowValue;  //按揭价值 for 港股  TDAPriceType FMortgageMoney;  // T1 for 港股  TDAPriceType T1;  // T2 for 港股  TDAPriceType T2;  //T3 for 港股  TDAPriceType T3;  // Tn for 港股  TDAPriceType TN;  //交易限额  TDAPriceType TradeLimit;  //跨市场资金限额  TDAPriceType FCrossMax;  //卖空冻结资金  TDAPriceType SellFreezenMoney;  //卖空利息  TDAPriceType SellInterest;  //需补按金  TDAPriceType SellNeedAddMargin; |

注意：回调数据结构体中有与股票相关的字段，使用时请无视。

### 改单请求/返回ReqOrderModify/OnRspOrderModify方法

函数原型：

|  |
| --- |
| bool ReqOrderModify(CFutureReqOrderModifyField \*pReqOrderModify, int iRequestID); |

参数：CFutureReqOrderModifyField

|  |
| --- |
| // 系统编号  TDAStringType SystemNo;  // 用户ID  TDAStringType UserId;  // 本地编号  TDAStringType LocalNo;  // 资金账号  TDAStringType AccountNo;  // 定单号  TDAStringType OrderNo;  // 交易所代码  TDAStringType ExchangeCode;  // 合约代码  TDAStringType ContractCode;  // 买还是卖：1=buy 2=sell  TDAStringType BidAskFlag;  // 下单数  TDAStringType OrderQty;  // 下单价格  TDAStringType OrderPrice;  // 改单数  TDAStringType ModifyQty;  // 改单价格  TDAStringType ModifyPrice;  // 价格类型：1=限价单, 2=市价单，3=限价止损（stop to limit），4=止损（stop to market）  TDAStringType OrderType;  // 触发价格  TDAStringType TriggerPrice;  // 改单触发价格  TDAStringType ModifyTriggerPrice;  // 有效日期（1：当日有效；2：永久有效）  TDAStringType TIF;  //下单人ID  TDAStringType Tag50;  // 错误信息  TDAStringType ErrorDescription;  // ISO(3166-1)标准中定义的国家/地区两个字母代码  TDAStringType OrgOrderLocationID; |

函数原型：

|  |
| --- |
| void OnRspOrderModify(CFutureRspOrderModifyField \*pRspOrderModify, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

typedef CFutureRspOrderInsertField CFutureRspOrderModifyField;

具体改单返回结构体字段，请参照报单返回结构体CFutureRspOrderInsertField。

参数：CFutureRspInfoField

|  |
| --- |
| // 错误码(用来判断改单成功或失败)  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 撤单请求/返回ReqOrderCancel/OnRspOrderCancel

函数原型：

|  |
| --- |
| bool ReqOrderCancel(CFutureReqOrderCancelField \*pReqOrderCancel, int iRequestID) |

参数：CFutureReqOrderCancelField

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 用户类型：1：一般用户；2：机构通用户；  TDAStringType UserType;  // 本地编号  TDAStringType LocalNo;  // 资金账号  TDAStringType AccountNo;  // 系统编号  TDAStringType SystemNo;  // 定单号  TDAStringType OrderNo;  // 交易所代码  TDAStringType ExchangeCode;  // 合约代码  TDAStringType ContractCode;  // 买还是卖：1=buy 2=sell  TDAStringType BidAskFlag;  // 下单数  TDAStringType OrderQty;  // 下单价格  TDAStringType OrderPrice;  // 价格类型：1=limit order, 2=market order  TDAStringType OrderType;  // 错误信息  TDAStringType ErrorDescription;  // ISO(3166-1)标准中定义的国家/地区两个字母代码  TDAStringType OrgOrderLocationID; |

函数原型：

|  |
| --- |
| Void OnRspOrderCancel(CFutureRspOrderCancelField \*pRspOrderCancel, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) |

参数：CFutureRspOrderCancelField

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 资金账号  TDAStringType AccountNo;  // 系统编号  TDAStringType SystemNo;  // 本地号  TDAStringType LocalNo;  // 定单号  TDAStringType OrderNo;  // 撤单号  TDAStringType CancelNo;  // 交易所代码  TDAStringType ExchangeCode;  // 合约代码  TDAStringType ContractCode;  // 买还是卖：1=buy 2=sell  TDAStringType BidAskFlag;  // 下单数  TDAStringType OrderQty;  // 下单价格  TDAStringType OrderPrice;  // 已成交数  TDAStringType FilledQty;  // 已撤单数  TDAStringType CancelledQty;  // 价格类型：1=limit order, 2=market order  TDAStringType OrderType;  // 撤单日期  TDAStringType CancelledDate;  // 撤单时间  TDAStringType CancelledTime;  // 错误代码(保留，未使用)  TDAStringType ErrorCode; |

参数：CFutureRspInfoField

|  |
| --- |
| // 错误码(用来判断撤单成功或失败)  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 委托单查询/返回ReqQryOrder/OnRspQryOrder

两个委托单查询之间最小间隔时间为1秒，否则查询失败。(Since API v1.5)

函数原型：

|  |
| --- |
| bool ReqQryOrder( CFutureQryOrderField \*pQryOrder, int iRequestID) |

参数：CFutureQryOrderField

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 资金账号  TDAStringType AccountNo;  // 是否模拟用户：1：是；0 or other：不是  TDAStringType IsSimulation;  // 取得指定订单号以后的定单  TDAStringType OrderNo;  // 取得指定订单时间以后的定单（格式：yyyy-MM-dd hh:mm:ss）  TDAStringType OrderDateTime;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryOrder(CFutureRspOrderField \*pRspOrder, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) |

参数：CFutureRspOrderField

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 资金账号  TDAStringType AccountNo;  // 系统编号  TDAStringType SystemNo;  // 本地编号  TDAStringType LocalNo;  // 定单号  TDAStringType OrderNo;  // 交易所代码  TDAStringType ExchangeCode;  // 合约代码  TDAStringType ContractCode;  // 买还是卖：1=buy 2=sell  TDAStringType BidAskFlag;  // 下单数  TDAStringType OrderQty;  // 下单价格  TDAStringType OrderPrice;  // 已成交数  TDAStringType FilledQty;  // 成交均价  TDAStringType FilledPrice;  // 定单类型：1=限价单, 2=市价单，3=限价止损（stop to limit），4=止损（stop to market）  TDAStringType OrderType;  // 下单日期  TDAStringType OrderDate;  // 下单时间  TDAStringType OrderTime;  // 错误代码  TDAStringType ErrorCode;  // 订单状态（1：已请求；2：已排队；3：部分成交；4：完全成交；5：已撤余单；6：已撤单；7：指令失败；）  TDAStringType OrderState;  // 撤单的用户ID  TDAStringType CancelUserId;  // 触发价格  TDAStringType TriggerPrice;  // 有效日期（1=当日有效, 2=永久有效（GTC），4=IOC）  TDAStringType TIF;  // 开仓还是平仓：1=开仓 2=平仓，3=平今，4=平昨  TDAStringType OpenCloseFlag;  // 策略ID  TDAStringType StrategyId;  // 显示委托量 必须小于委托量  TDAStringType MaxShow;  // 最小成交量  TDAStringType MinQty;  // 交易所返回时间  TDAStringType ExchangeTime;  // 撤单时间  TDAStringType CancelTime; |

参数：CFutureRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 成交查询请求/返回ReqQryTrade/OnRspQryTrade

两个成交查询之间最小间隔时间为1秒，否则查询失败。(Since API v1.5)

函数原型：

|  |
| --- |
| bool ReqQryTrade(CFutureQryTradeField \*pQryTrade, int iRequestID) |

参数：CFutureQryTradeField

|  |
| --- |
| // 用户  TDAStringType UserId;  // 错误信息  TDAStringType ErrorDescription; |

由于成交数据可能很多，增加分页查询方式。具体参照demo中的代码。

(Since API v1.15)

函数原型：

|  |
| --- |
| void OnRspQryTrade(CFutureRspTradeField \*pRspTrade, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) |

参数：CFutureRspTradeField

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 资金账号  TDAStringType AccountNo;  // 成交编号（要包括7位的订单编号，一共11位）  TDAStringType FilledNo;  // 定单号  TDAStringType OrderNo;  // 系统编号  TDAStringType SystemNo;  // 本地编号  TDAStringType LocalNo;  // 交易所代码  TDAStringType ExchangeCode;  // 合约代码  TDAStringType TreatyCode;  // 买还是卖：1=buy 2=sell  TDAStringType BuySale;  // 成交数  TDAStringType FilledNumber;  // 成交价格  TDAStringType FilledPrice;  // 成交日期(yyyy-MM-dd)  TDAStringType FilledDate;    // 成交时间(hh:mm:ss)  TDAStringType FilledTime;  // 成交手续费  TDAStringType Commsion;  // 委托数量  TDAStringType OrderNumber;  // 委托价格  TDAStringType OrderPrice;  // 合约交割日期(yyyyMMdd)  TDAStringType DeliveryDate;  // 成交类别(N：普通下单成交；C、T：调期模拟成交)  TDAStringType FilledType;  // 定单类型（1=限价单, 2=市价单，3=限价止损（stop to limit），4=止损（stop to market）  TDAStringType OrderType;  // 有效日期（1=当日有效, 2=永久有效）  TDAStringType ValidDate;  // 开仓还是平仓：1=开仓 2=平仓，3=平今，4=平昨  TDAStringType AddReduce;  // 错误信息  TDAStringType ErrorDescription; |

参数：CFutureRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 持仓明细查询请求/返回ReqQryPosition/OnRspQryPosition

两个持仓明细查询之间最小间隔时间为1秒，否则查询失败。(Since API v1.5)

函数原型：

|  |
| --- |
| bool ReqQryPosition(CFutureQryPositionField \*pQryPosition, int iRequestID) |

参数：CFutureQryPositionField

|  |
| --- |
| // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryPosition(CFutureRspPositionField \*pRspPosition, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) |

参数：CFutureRspPositionField

|  |
| --- |
| // 用户ID  TDAStringType ClientNo;  // 交易所  TDAStringType ExchangeNo;  // 证券代码  TDAStringType CommodityNo;  // 持仓方向（1：持买；2：持卖）  TDAStringType Direct;  // 持仓成本价  TDADoubleType HoldPrice;  // 可卖数量  TDAIntType CanTradeVol;  // 今买数量  TDAIntType TodayBuyVol;  // 冻结数量  TDAIntType FrozenVol;  // 持有期内买入金额总和  TDADoubleType TotalBuyMoney;  // 持有期内卖出金额总和  TDADoubleType TotalSellMoney;  // 持有期内买入数量  TDAIntType TotalBuyVol;  // 持有期内卖出数量  TDAIntType TotalSellVol;  // 首次开仓日期(yyyy-MM-dd)  TDAStringType OpenDate;  // 平仓盈利  TDADoubleType FlatProfit;  // 港股T+1数量  TDAIntType HkexT1;  // 港股T+2数量  TDAIntType HkexT2;  // 美股T+3数量  TDAIntType HkexT3;  // 港股未交收数量=T1+T2  TDAIntType UnsettleVol;  // 港股已交收数量  TDAIntType SettledVol;  // 持仓数量  TDAIntType HoldVol;  // 今卖数量  TDAIntType TodaySaleVol;  // 卖空冻结资金  TDADoubleType SellFrozenMoney;  // 开仓均价  TDADoubleType OpenPrice; |

参数：CFutureRspInfoField

|  |
| --- |
| // 用户ID  TDAStringType AccountNo;  // 错误信息  TDAStringType ErrorDescription; |

### 资金查询请求/返回ReqQryCapital/OnRspQryCapital

两个资金查询之间最小间隔时间为1秒，否则查询失败。(Since API v1.5)

函数原型：

|  |
| --- |
| Bool ReqQryCapital( CFutureQryCapitalField \*pQryCapital, int iRequestID) |

参数：CFutureQryCapitalField

|  |
| --- |
| // 不使用  TDACharType Unused;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryCapital(CFutureRspCapitalField \*pRspCapital, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) |

参数：CFutureRspCapitalField

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 入金  TDAStringType Deposit;  // 出金  TDAStringType Withdraw;  // 今日可用于交易的资金量(今可用)  TDAStringType TodayTradableFund;  // 当日期初权益(今结存)  TDAStringType TodayInitialBalance;  // 当日实时浮动权益(今权益)  TDAStringType TodayRealtimeBalance;  // 冻结资金  TDAStringType FrozenFund;  // 佣金  TDAStringType Commission;  // 初始保证金  TDAStringType InitialMargin;  // 昨日可用于交易的资金量(昨可用)  TDAStringType YdTradableFund;  // 昨日期初权益(昨结存)  TDAStringType YdInitialBalance;  // 昨日期末权益(昨权益)  TDAStringType YdFinalBalance;  // 浮动盈亏  TDAStringType ProfitLoss;  // 币种编号  TDAStringType CurrencyNo;  // 货币与基本的汇率  TDAMoneyType CurrencyRate;  // LME未到期平盈(未到期平盈)  TDAMoneyType LMEUnexpiredPL;  // LME未结平盈(未结平盈)  TDAMoneyType LMEUnaccountPL;  // 维持保证金  TDAMoneyType MaintenanceMargin;  // 期权权利金  TDAMoneyType Premium;  // 信任额度  TDAMoneyType CreditAmount;  // 初始资金  TDAMoneyType IntialFund;  // 资金帐号 1  TDAStringType FundAccountNo;  // 交易限额  TDAMoneyType TradeLimit;  // 可取资金==min(T+0,T+0+T+1,T+0+T+1+T+2)-冻结资金(挂单)  TDAMoneyType CanCashOutMoneyAmount;  // 月存款利息  TDAMoneyType DepositInterest;  // 月欠款利息  TDAMoneyType LoanInterest;  // 错误信息  TDAStringType ErrorDescription; |

参数：CFutureRspInfoField

|  |
| --- |
| // 不使用  TDACharType Unused;  // 错误信息  TDAStringType ErrorDescription; |

### 版本号查询请求/返回ReqQryVersion/OnRspQryVersion

函数原型：

|  |
| --- |
| bool ReqQryVersion(CFutureQryVersionField \*pQryVersion, int iRequestID) |

参数：ReqQryVersion

|  |
| --- |
| // 用户ID  TDAStringType UserId;  // 用户密码  TDAStringType UserPwd;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryVersion(CFutureRspVersionField \*pRspVersion, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) |

参数：CFutureRspVersionField

|  |
| --- |
| // 版本号  TDAStringType Version;  // 是否必须更新到新版本才能正常使用（0：可以不更新；1：必须更新）  TDAStringType MustUpdate;  // 必须要更到的版本号  TDAStringType MustVersion;  // 版本内容中文  TDAStringType VersionContent\_CN;  // 版本内容英文  TDAStringType VersionContent\_US; |

参数：CFutureRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 币种查询请求/返回ReqQryCurrency/ OnRspQryCurrency

函数原型：

|  |
| --- |
| bool ReqQryCurrency(CFutureQryCurrencyField \*pQryCurrency, int iRequestID) |

参数：CFutureQryCurrencyField

|  |
| --- |
| // 不使用  TDACharType Unused;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryCurrency(CFutureRspCurrencyField \*pRspCurrency, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) |

参数：CFutureRspCurrencyField

|  |
| --- |
| // 货币编号  TDAStringType CurrencyNo;  // 基币货币编号  TDAIntType IsBase;  // 与基币的换算汇率  TDADoubleType ChangeRate;  // 货币名称  TDAStringType CurrencyName;  // 货币名称（英文）  TDAStringType CurrencyNameEN; |

参数：CFutureRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 交易时间查询请求/返回ReqQryTradeTime/ OnRspQryTradeTime

函数原型：

|  |
| --- |
| bool ReqQryTradeTime(CFutureQryTradeTimeField \*pQryTradeTime, int iRequestID) |

参数：CFutureQryTradeTimeField

|  |
| --- |
| // 不使用  TDACharType Unused;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryTradeTime(CFutureRspTradeTimeField \*pRspTradeTime, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast){} |

参数：CFutureRspTradeTimeField

|  |
| --- |
| // 当前交易日期（yyyyMMdd）  TDAStringType TradeDate;  // 开盘时间（HH:mm:ss）  TDAStringType OpenTime;  // 收盘时间（HH:mm:ss  TDAStringType CloseTime; |

参数：CFutureRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 交易所查询请求/返回ReqQryExchange /OnRspQryExchange

函数原型：

|  |
| --- |
| bool ReqQryExchange(CFutureQryExchangeField \*pQryExchange, int iRequestID) = 0; |

参数：CFutureQryExchangeField

|  |
| --- |
| TDAStringType ProductGroupID;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryExchange(CFutureRspExchangeField \*pRspExchange, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) {} |

参数：CFutureRspExchangeField

|  |
| --- |
| // 交易所编号  TDAStringType ExchangeNo;  // 交易所名称  TDAStringType ExchangeName;  // 交易所类型（HK：港股；US：美股）  TDAStringType SettleType;  // 交易所名称（英文）  TDAStringType NameEN; |

参数：CFutureRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 合约查询请求/返回ReqQryInstrument/OnRspQryInstrument

函数原型：

|  |
| --- |
| bool ReqQryInstrument(CFutureQryInstrumentField \*pQryInstrument, int iRequestID) = 0; |

参数：CFutureQryInstrumentField

|  |
| --- |
| // 查询多少条,每次返回的条数  TDAIntType PageIndex;  // 交易所代码，如果填写值，查询一个交易所的合约  TDAStringType ExchangeNo;  // 合约代码。查询单个合约  TDAStringType CommodityNo;  // 商品类别  TDAStringType CommodityType;  // 合约NO  TDAStringType ContractNo;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryInstrument(CFutureRspInstrumentField \*pRspInstrument, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) {} |

参数：CFutureRspInstrumentField

|  |
| --- |
| // 交易所编号  TDAStringType ExchangeNo;  // 交易所名称  TDAStringType ExchangeName;  // 合约NO  TDAStringType CommodityNo;  // 商品名  TDAStringType CommodityName;  // 商品类别，O：期权，F：期货，S：同商品跨期套利  TDAStringType CommodityType;  // 货币编号  TDAStringType CurrencyNo;  // 货币名称  TDAStringType CurrencyName;  // 点值（一个最小跳点的价值）  TDADoubleType ProductDot;  // 最小变动单位  TDADoubleType UpperTick;  // 上日结算价  TDADoubleType SettlePrice;  // 交易月 (yyyyMM)/交割日 (yyyyMMdd)  TDAStringType TradeMonth;  // 行情小数点位数  TDAIntType DotNum;  // 进阶单位  TDAIntType LowerTick;  // 调期小数点位数  TDAIntType DotNum\_Carry;  // 调期最小变动单位  TDADoubleType UpperTick\_Carry;  // 首次通知日 (yyyyMMdd)  TDAStringType FirstNoticeDay;  // 冻结保证金百分比  TDADoubleType FreezenPercent;  // 冻结保证金固定值  TDADoubleType FreezenMoney;  // 固定手续费  TDADoubleType FeeMoney;  // 百分比手续费  TDADoubleType FeePercent;  // 现货商品昨结算价  TDADoubleType PriceStrike;  // 现货商品点值  TDADoubleType ProductDotStrike;  // 现货商品最小变动单位  TDADoubleType UpperTickStrike;  // 最后交易日 (yyyyMMdd)(期货专用)  TDAStringType LastTradeDay;  // 最后更新日 (yyyyMMdd)(期货专用)  TDAStringType LastUpdateDay;  // 期权临界价格 (期货专用)  TDADoubleType CriticalPrice;  // 期权临界价格以下的最小跳点 (期货专用)  TDADoubleType CriticalMinChangedPrice;  // 实际交易所(CME细分成3个交易所:CME,CME\_COMEX,CME\_NYMEX)(期货专用)  TDAStringType Exchange2;  // 期权类型(R：看涨；F：看跌)(期货专用)  TDAStringType OptionType;  // 期权年月(yyyyMM)(期货专用)  TDAStringType OptionMonth;  // 期权执行价格(期货专用)  TDAStringType OptionStrikePrice;  // 期权对应期货商品编号(期货专用)  TDAStringType OptionCommodityNo;  // 期权对应期货合约编号(期货专用)  TDAStringType OptionContractNo;  // 按揭价值百分比  TDAStringType MortgagePercent;  // 最小变动单位代码  TDAStringType UpperTickCode;  // 最小批量  TDAStringType LotSize;  // 平仓时间标志(控制T+1：0代表T+0，可以当天卖出平仓；1代表T+1，必须隔天卖出平仓；N代表T+N)  TDAStringType FlatTime;  // 商品名（英文）  TDAStringType CommodityFNameEN;  // 是否支持卖空（0或空：不可以；1：可以）  TDAStringType CanSell;  // 卖空时的按金比例（例：0.5）  TDADoubleType SellRate;  // 卖空持仓上限额  TDADoubleType SellMax;  // 换股比率，表示一份窝轮牛熊証，相当于多少正股  TDADoubleType StrikeRate;  // 行使价  TDADoubleType StrikePrice;  // 收回价  TDADoubleType ReceivePrice;  // 到期日（yyyy-MM-dd）  TDAStringType ExpireDate;  // 卖空时的维持按金比例（例：0.5）  TDADoubleType SellRateKeep;  // 涡轮牛熊证对应正股编号  TDAStringType StrikeCommodityNo;  // 涡轮牛熊证的涨跌标志(C:涨；P:跌)  TDAStringType CallPutFlag;  // 涡轮牛熊证的发行人  TDAStringType Publisher; |

### 策略查询请求/返回ReqQryStrategy/ OnRspQryStrategy

函数原型：

|  |
| --- |
| bool ReqQryStrategy(CFutureQryStrategyField \*pQryStrategy, int iRequestID) = 0; |

参数：CFutureQryStrategyField

|  |
| --- |
| // 合约代码。查询单个合约  TDAStringType InstrumentID;  // 交易所代码，如果填写值，查询一个交易所的合约  TDAStringType ExchangeID;  // 合约在交易所的代码  TDAStringType ExchangeInstID;  // 产品代码 ，如果填写值，查询一个产品的合约  TDAStringType ProductID;  // 开始时间,如果填写，是这个时间以后新增的  TDAStringType InsertTimeStart;  // 查询多少条,每次返回的条数  TDAVolumeType Index;  // 查询合约的类别 直达  TDAStringType ProductType;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryStrategy(CFutureRspStrategyField \*pRspStrategy, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) {} |

参数：CFutureRspStrategyField

|  |
| --- |
| // 策略所属用户ID  TDAStringType UserId;  // 策略Id  TDAStringType FKeyId;  // 策略名称  TDAStringType FName;  // 策略代码  TDAStringType FCode;  // 策略价格类型  TDAIntType PriceType;  // 策略价格详细类型  TDAIntType PriceTypeDetailType;  // 策略价格公式  TDAStringType PriceFormula;  // 触发模式  TDAIntType TriggerMethod;  // 是否启用内盘保护  TDAIntType InnerProtect;  // 是否被动腿下单失败立即平对应主动腿  TDAIntType PassiveFailCloseMainLeg;  // 触发模式为2：成交优先时设置的滑点数  TDAIntType SlipPoint;  // 追加保护策略  TDAIntType RecoverPriceMethod;  // 被动腿挂单多少秒后以市价追单  TDAIntType RecoverPriceSeconds;  // 策略设置类型  TDAStringType SetType;  // 策略行情价格最小变动单位  TDAPriceType MinChangePrice;  // 主动腿单次最大下单量  TDAIntType MaxNum;  // 被动腿合约最小对盘挂单量  TDAIntType SuportQuantity;  // 安全深度  TDAIntType SafeDeep;  // 主动区域  TDAIntType MainRange;  // 单腿手动追被动设置  TDAStringType ManualZhuiDanBeyondNum;  // 单腿手动平主动设置  TDAStringType ManualPingZhuDongBeyondNum;  // 单腿自动定时设置，挂单秒数  TDAIntType AutoGuaDanSeconds;  // 单腿自动定时追被动设置  TDAIntType AutoZhuiBeiDongDots;  // 单腿自动定时平主动设置  TDAIntType AutoPingZhuDongDots;  // 单腿自动止损设置  TDAIntType AutoZhiSunDot;  // 单腿自动止损追被动设置，追价滑点数：  TDAIntType AutoZhiSunZhuiBeiDongDots;  // 单腿自动止损平主动设置，追价滑点数：  TDAIntType AutoZhiSunPingZhuDongDots;  // 策略行情价格小数位数  TDAIntType DotLen;  // 设置交易的时间段  TDAStringType TradeTime; |

参数：CFutureRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 策略明细查询请求/返回ReqQryStrategyDetail/ OnRspQryStrategyDetail

函数原型：

|  |
| --- |
| bool ReqQryStrategyDetail(CFutureQryStrategyDetailField \*pQryStrategyDetail, int iRequestID) = 0; |

参数：CFutureQryStrategyDetailField

|  |
| --- |
| // 用户帐号  TDAStringType UserId;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryStrategyDetail(CFutureRspStrategyDetailField\*pRspStrategyDetail, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) {} |

参数：CFutureRspStrategyDetailField

|  |
| --- |
| // 策略商品编号  TDAStringType StartegyCommodityNo;  // 策略合约组合编号  TDAStringType StartegyContractNo;  // 价格  TDAStringType Price;  // 腿数（2个合约套利为2腿）  TDAIntType LegNum;  // 最后交易日（yyyy-MM-dd）  TDAStringType LastTradeDate;  // 策略中合约顺序号  TDAIntType SeqId;  // 策略中合约的商品编号  TDAStringType CommodityNo;  // 策略中合约的合约编号  TDAStringType ContractNo;  // 买卖方向（1表示买，2表示卖）  TDAStringType Direct;  // 合约下单数量  TDAIntType Quantity;  // 合约下单数量  TDAStringType Model;  // 最后更新日期（yyyy-MM-dd）  TDAStringType ModifyDate;  // 下单类型（限价，市价）  TDAStringType OrderMethod;  // 价格容忍度  TDAPriceType PriceTolerance;  // 价格乘数参数  TDAPriceType PriceParam;  // 合约最小对盘挂单量  TDAIntType SuportQuantity;  // 合约最小移动  TDAIntType MinMove;  // 合约定时追单  TDAIntType TimerOrder;  // 定时追单滑点数  TDAStringType TimerOrderBeyondNum;  // 合约定时追价  TDAIntType TimerOrderPrice;  // 定时追价滑点数  TDAStringType TimerOrderPriceBeyondNum;  // 定时追价触发次数  TDAStringType TimerOrderPriceTriggerNum;  // 止损追单  TDAIntType ZhiSunOrder;  // 止损追单滑点数  TDAStringType ZhiSunOrderBeyondNum;  // 合约定时平仓  TDAIntType TimerPingCang;  // 定时平仓滑点数  TDAStringType TimerPingCangBeyondNum;  // 安全深度  TDAIntType SafeDeep;  // 主动区域  TDAIntType MainRange; |

参数：CFutureRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 设置双重认证密码/返回ReqSetVerifyQA/ OnRspSetVerifyQA

函数原型：

|  |
| --- |
| bool ReqSetVerifyQA(CStockReqSetVerifyQAField \*pReqSetVerifyQA, int iRequestID); |

参数：CStockReqSetVerifyQAField

|  |
| --- |
| // 用户帐号  TDAStringType UserId;  // 交易密码  TDAStringType UserPwd;  // 固定值，“1”  TDAStringType Type;  // 固定值，“1”  TDAStringType Question;  // 双重认证密码，英数字  TDAStringType Answer;  // 固定值，“1”  TDAStringType SaveMac;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspSetVerifyQA(CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：CStockRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

回调函数错误码指示成功时，调用ReqSafeVerify()函数绑定网卡地址。

### 绑定双重认证需要的MAC地址请求/返回ReqSafeVerify/ OnRspSafeVerify

函数原型：

|  |
| --- |
| bool ReqSafeVerify(CStockReqSafeVerifyField \*pReqSafeVerify, int iRequestID); |

参数：CStockReqSafeVerifyField

|  |
| --- |
| // 用户帐号  TDAStringType UserId;  // 交易密码  TDAStringType UserPwd;  // 固定值，“1”  TDAStringType Type;  // 固定值，“1”  TDAStringType Question;  // 双重认证密码，英数字  TDAStringType Answer;  // 固定值，“1”  TDAStringType SaveMac;  // MAC地址  TDAStringType MacAddress;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspSafeVerify(CStockRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：CStockRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 查询品种列表请求/返回ReqQryCommodity/ OnRspQryCommodity

函数原型：

|  |
| --- |
| bool ReqQryCommodity(CFutureQryCommodityField \*pQryCommodity, int iRequestID); |

参数：CFutureQryCommodityField

|  |
| --- |
| //更新日期  TDAStringType UpdateDate;  //交易所编号  TDAStringType ExchangeNo;  // 错误信息  TDAStringType ErrorDescription; |

说明：UpdateDate不填值时，返回所有支持的品种。填入某个日期时，返回那个日期之后变化过的品种。

函数原型：

|  |
| --- |
| void OnRspQryCommodity(CFutureRspCommodityField \*pRspCommodity, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：CFutureRspCommodityField

|  |
| --- |
| //商品编号，如GC，CL  TDAStringType CommodityNo;  // 交易所代码，如CME，HKEX  TDAStringType ExchangeNo;  //商品类型  TDAStringType CommodityType;  //商品名称  TDAStringType Name;  //交易状态  TDAIntType Enabled;  //更新日  TDAIntType RegDate;  //币种编号  TDAPriceType CurrencyNo;  //账面跳点值  TDADoubleType ProductDot;  //进阶单位  TDADoubleType LowerTick;  //最小变动单位  TDADoubleType UpperTick;  //行情小数点位数  TDAIntType DotNum;  //策略合约ID  TDAIntType StrikeCommodityId;  //期权类型  TDAStringType OptionStyle;  //交易所编号2 (在ExchangeNo下的交易所比如CME\_COMEX,CME\_ NYMEX CME\_CBT, CME)  TDAStrngType ExchangeNo2;  //是否是个股期货  TDAPriceType IsSFuture; |

说明：

ExchangeNo2/交易所编号2，这个字段是行情缴费使用，以CME为例，CME可以让投资者选择CME\_COMEX, CME\_ NYMEX,CME\_CBT, CME 其中的一个或多个付费，相应的，当API用户按交易所订阅行情时，应该填写这个交易所代码。

### 持仓总结查询请求/返回ReqQryTotalPosition/ OnRspQryTotalPosition

函数原型：

|  |
| --- |
| bool ReqQryTotalPosition(CFutureQryTotalPositionField \*pQryTotalPosition, int iRequestID); |

参数：CFutureQryTotalPositionField

|  |
| --- |
| //资金账号  TDAStringType AccountNo;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryTotalPosition(CFutureRspTotalPositionField\* pRspTotalPosition, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：CFutureRspTotalPositionField

(typedef CFutureRtnOrderField CFutureRtnPositionField)

|  |
| --- |
| // 本地订单号  TDAStringType LocalOrderNo;  // 交易所代码  TDAStringType ExchangeNo;  // 合约代码  TDAStringType ContractCode;  // 定单号  TDAStringType OrderNo;  // 委托数量  TDAIntType OrderQty;  // 已成交数量  TDAIntType FilledQty;  // 成交均价  TDAPriceType FilledAvgPrice;  // 多仓/持买数量  TDAIntType LongPositionQty;  //多仓/持买开仓均价  TDAPriceType LongPosAveragePrx;  //中国品种(比如INE/sc)盯市用的平均价(持买均价)  TDAPriceType CNLongPosAveragePrx;  // 空仓/持卖数量  TDAIntType ShortPositionQty;  //空仓/持卖开仓均价  TDAPriceType ShortPosAveragePrx;  //中国品种(比如INE/sc)盯市用的平均价（持卖均价）  TDAPriceType CNShortPosAveragePrx;  // 是否已经撤单（0：没有；1：已撤单）  TDAStrngType IsCanceled;  // 成交总的手续费  TDAPriceType FilledTotalFee;  // 顺序号  TDAIntType SequenceNo;  // 资金帐号  TDAStringType AccountNo;  // 持仓类型（0：昨仓；1：今仓）  TDAStringType PositionType;  // 多仓/持买保证金  TDAPriceType LongPosMargin;  // 空仓/持卖保证金  TDAPriceType ShortPosMargin;  // 最新价  TDAPriceType CurrPrice;  // 浮动盈亏  TDAPriceType ProfitLoss; |

### 交易所冬夏令开盘时间查询请求/返回ReqQryExchangeTime/ OnRspQryExchangeTime

函数原型：

|  |
| --- |
| bool ReqQryExchangeTime(CFutureQryExchangeTimeField \*pQryExchangeTime, int iRequestID); |

参数：CFutureQryExchangeTimeField

|  |
| --- |
| //不使用  TDACharType Unused;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryExchangeTime(CFutureRspExchangeTimeField \*pRspExchangeTime, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：CFutureRspExchangeTimeField

|  |
| --- |
| //当前年份  TDAStringType Year;  //夏令时开始时间  TDAStringType SummerBegin;  //冬令时开始时间  TDAStringType WinterBegin;  //交易所代码  TDAStringType ExchangeNo;  //交易所名称  TDAIntType Name; |

### 品种开收盘时间查询请求/返回ReqQryExchangeTime/ OnRspQryCommodityTime

函数原型：

|  |
| --- |
| bool ReqQryCommodityTime(CFutureQryCommodityTimeField \*pQryCommodityTime, int iRequestID); |

参数：CFutureQryCommodityTimeField

|  |
| --- |
| //交易所代码  TDAStringType ExchangeNo;  //品种代码  TDAStringType CommodityNo;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspQryCommodityTime(CFutureRspCommodityTimeField \*pRspCommodityTime, CFutureRspInfoField \*pRspInfo, int iRequestID, bool bIsLast); |

参数：CFutureRspCommodityTimeField

|  |
| --- |
| //是否跨日  TDAStringType CrossTrade;  //交易段  TDAStringType Stage;  //2冬令 1夏令  TDAStringType Summer;  //产品开市时间  TDAStringType Opendate;  //产品收市时间  TDAStringType Closingdate;  //品种代码  TDAStringType CommodityNo;  //品种名称  TDAStringType ComName;  //交易所代码  TDAStringType ExchangeNo;  //交易所名称  TDAStringType ExName; |

# 期货股票通用行情API接口参考

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 业务类型 | 业务 | 请求接口 |  | 响应接口 |
| 连接 | 连接 | Init | 连接回报 | OnFrontConnected |
| 断开 |  |  | 连接断开回报 | OnFrontDisconnected |
| 心跳 |  | SetHeartBeatTimeout | 心跳警告 | OnHeartBeatWarning |
| 登录 | 登录 | ReqUserLogin | 登录回报 | OnRspUserLogin |
| 行情订阅 | 行情请求 | ReqMarketData | 行情推送 | OnRtnMarketData |
|  | 经纪商获取  (港股专有) | ReqBrokerData | 经纪商回报 | OnRspBrokerData |
| 推送数据  (from v1.11) | 交易日推送 |  |  | OnRspTradeDate |

## 接口工作过程



## 接口模式

行情API 的DAMarketApi.h头文件中提供了两个接口，分别为CMarketApi和IMarketEvent，方便客户端应用程序的开发。客户端应用程序可以通过CMarketApi发出操作请求，通过继承IMarketEvent并重载回调函数来处理后台服务的响应。

## 函数接口说明

### 1）CreateMarketApi方法

创建实例方法。

**函数原形**：

static CMarketApi \*CreateMarketApi(bool bRecordLog, const char\* lpszLogFileName = "Market.log");

bRecordLog: true表示记录日志，false表示不记录日志.

lpszLogFileName：日志文件

### 2）OnFrontConnected 方法

当客户端与前置建立起通信连接时（还未登录前），该方法被调用。

**函数原形：**

void OnFrontConnected()；

本方法在完成初始化后调用，可以在其中完成用户登录任务。

### 3）OnFrontDisconnected方法

当客户端与前置通信连接断开时，该方法被调用。当发生这个情况

后，API会自动重新连接，客户端可不做处理。自动重连地址，可能是原来注册

的地址，也可能是系统支持的其它可用的通信地址，它由程序自动选择。

**函数原形：**

void OnFrontDisconnected (int nReason)；

**参数：**

nReason：连接断开原因

### 4）OnHeartBeatWarning方法

心跳超时警告。当长时间未收到报文时，该方法被调用。

**函数原形：**

void OnHeartBeatWarning(int nTimeLapse)；

**参数：**

nTimeLapse：距离上次接收报文的时间

SetHeartBeatTimeout

### 5）登录ReqUserLogin/OnRspUserLogin

函数原型：

|  |
| --- |
| bool ReqUserLogin( CMarketReqUserLoginField \*pReqUserLoginField, int iRequestID) ; |

参数：CMarketReqUserLoginField

|  |
| --- |
| // 用户ID  TDAStringType UserId; （必输）  // 用户密码  TDAStringType UserPwd; （必输）  // 用户类型  TDAStringType UserType;  // 软件名称  TDAStringType SoftwareName; （必输）  // 软件版本号  TDAStringType SoftwareVersion; （必输）  // 授权码  TDAStringType AuthorCode; （必输）  // 错误信息  TDAStringType ErrorDescription;  //行情分发商ID(since v1.18.0.0)  TDAStringType BrokerIDForMarketPrice; |

函数原型：

|  |
| --- |
| void OnRspUserLogin( CMarketRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) {}; |

参数：CMarketRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

错误码：

|  |  |
| --- | --- |
| **登录错误码** | **错误内容** |
| 10001 | 用户名不正确 |
| 10002 | 登录密码错误 |
| 10003 | 密码错误次数超限，用户已冻结，请联系客服解冻！ |
| 10004 | 用户已被冻结，请联系客服解冻！ |

### 6）行情推送请求/响应ReqMarketData/OnRspMarketData

函数原型：

|  |
| --- |
| bool ReqMarketData(CMarketReqMarketDataField\* pReqMarketDataField, int iRequestID) = 0 |

参数：CMarketReqMarketDataField

|  |
| --- |
| // 金融类型  TDACharType MarketType;  // 行情种类  TDACharType SubscMode;  // 单次最大订阅数  TDAIntType MarketCount;  // 单次最大订阅合约  TDAStringType MarketTrcode[MAX\_SUB\_COUNT];  // 错误信息  TDAStringType ErrorDescription; |

说明：

1. 请参照API demo的订阅示例。
2. 每次订阅函数调用，合约个数上限，请查看合约数组的常量定义。
3. 订阅函数调用次数不限，建议每个调用间隔5至10毫秒。
4. 订阅期货合约时，MarketTrcode[x]的字符串使用查询合约返回的结构体中的交易所代码(ExchangeNo)和合约代码(CommodityCode)，用逗号拼接:

CFutureRspInstrumentField.ExchangeNo+”,”+CFutureRspInstrumentField.CommodityCode

1. 订阅股票时，MarketTrcode[x]的字符串使用查询合约返回的结构体中的交易所代码(ExchangeNo)和合约代码(CommodityCode)，用逗号拼接:

CStockRspInstrumentField.ExchangeNo+”,”+CStockRspInstrumentField.CommodityNo

函数原型：

|  |
| --- |
| void OnRspMarketData(CMarketRspMarketDataField \*pRspMarketData, CMarketRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) {} |

参数：CMarketRspMarketDataField

|  |
| --- |
| // 交易所代码  TDAStringType ExchangeCode;  // 合约代码  TDAStringType TreatyCode;  // 买价  TDAStringType BuyPrice;  // 买量  TDAStringType BuyNumber;  // 卖价  TDAStringType SalePrice;  // 卖量  TDAStringType SaleNumber;  // 最新价  TDAStringType CurrPrice;  // 现量  TDAStringType CurrNumber;  // 当天最高价  TDAStringType High;  // 当天最低价  TDAStringType Low;  // 开盘价  TDAStringType Open;  //盘中结算价(股票：收盘价)，CME交易所在交易盘中会推出当前交易日结算价，亚洲的交易所是在收盘 (T session) 后推出结算价(通过最后几分钟竞价得到)  TDAStringType ~~OldClose~~ 🡪 IntradaySettlePrice;  //当前交易日的前结算（股票：昨收盘价）  TDAStringType PreSettlementPrice;  // 废弃字段  TDAStringType Close;  // 行情时间  TDAStringType Time;  // 成交量  TDAStringType FilledNum;  // 持仓量(连接股票的行情前置时，代表成交金额。)  TDAStringType HoldNum;  // 买价2  TDAStringType BuyPrice2;  // 买价3  TDAStringType BuyPrice3;  // 买价4  TDAStringType BuyPrice4;  // 买价5  TDAStringType BuyPrice5;  // 买量2  TDAStringType BuyNumber2;  // 买量3  TDAStringType BuyNumber3;  // 买量4  TDAStringType BuyNumber4;  // 买量5  TDAStringType BuyNumber5;  // 卖价2  TDAStringType SalePrice2;  // 卖价3  TDAStringType SalePrice3;  // 卖价4  TDAStringType SalePrice4;  // 卖价5  TDAStringType SalePrice5;  // 卖量2  TDAStringType SaleNumber2;  // 卖量3  TDAStringType SaleNumber3;  // 卖量4  TDAStringType SaleNumber4;  // 卖量5  TDAStringType SaleNumber5;  // 隐藏买价  TDAStringType HideBuyPrice;  // 隐藏买量  TDAStringType HideBuyNumber;  // 隐藏卖价  TDAStringType HideSalePrice;  // 隐藏卖量  TDAStringType HideSaleNumber;  // 跌停价  TDAStringType LimitDownPrice;  // 涨停价  TDAStringType LimitUpPrice;  // 交易日  TDAStringType TradeDay;  // 买价6  TDAStringType BuyPrice6;  // 买价7  TDAStringType BuyPrice7;  // 买价8  TDAStringType BuyPrice8;  // 买价9  TDAStringType BuyPrice9;  // 买价10  TDAStringType BuyPrice10;  // 买量6  TDAStringType BuyNumber6;  // 买量7  TDAStringType BuyNumber7;  // 买量8  TDAStringType BuyNumber8;  // 买量9  TDAStringType BuyNumber9;  // 买量10  TDAStringType BuyNumber10;  // 卖价6  TDAStringType SalePrice6;  // 卖价7  TDAStringType SalePrice7;  // 卖价8  TDAStringType SalePrice8;  // 卖价9  TDAStringType SalePrice9;  // 卖价10  TDAStringType SalePrice10;  // 卖量6  TDAStringType SaleNumber6;  // 卖量7  TDAStringType SaleNumber7;  // 卖量8  TDAStringType SaleNumber8;  // 卖量9  TDAStringType SaleNumber9;  // 卖量10  TDAStringType SaleNumber10;  // 港交所股票行情：成交类型  TDAStringType TradeFlag;  // 交易所数据时间戳  TDAStringType DataTimestamp;  // 数据来源  TDAStringType DataSourceId;  // 可卖空股数（美股行情用）  TDAStringType CanSellVol;  // 行情区分，分两种情况(意思是Y和2中的成交量可以统计到分钟数据里，Z的不可以)  // 直连交易所的行情 -- Y: 当前回调数据是最新成交数据；Z: 当前回调数据是快照数据  // 非直连交易所的行情 -- 2：当前回调数据包括最新成交数据和盘口变化 Z: 快照数据中成交量// 不能用于统计  TDAStringType QuoteType; |

参数：CMarketRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 7）经纪商获取请求/响应ReqBrokerData/OnRspBrokerData

函数原型：

|  |
| --- |
| bool ReqBrokerData(CMarketReqBrokerDataField\* pReqBrokerDataField, int iRequestID); |

参数：CMarketReqBrokerDataField

|  |
| --- |
| // 合约代码  TDAStringType HeyuCode;  // 错误信息  TDAStringType ErrorDescription; |

函数原型：

|  |
| --- |
| void OnRspBrokerData(CMarketRspBrokerDataField \*pRspBrokerData, CMarketRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) |

参数：CMarketRspBrokerDataField

|  |
| --- |
| // 经济商数据  TDABrokerType BrokerData; |

参数：CMarketRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 8）交易日推送OnRspTradeDate

函数原型：

|  |
| --- |
| void OnRspTradeDate(CMarketRspTradeDateField \*pRspTradeDate, CMarketRspInfoField \*pRspInfo, int iRequestID, bool bIsLast) |

参数：CMarketRspTradeDateField

|  |
| --- |
| // 交易日期(yyyy-MM-dd)  TDAStringType TradeDate;  // 交易品种列表，格式为”GC,SI,HG”  TDAStringType TradeProduct; |

说明:

1. 只有期货有这个推送交易日功能。
2. 交易日推送功能，碰到节假日需要API用户自己调整，比如:

香港交易所2019/4/5日假日，恒指HSI在2019/ 4/4 17:00推送的交易日 是 2019-04-05,实际应该是2019-04-08

1. 每个品种按开盘时间提前15分钟推送。

参数：CMarketRspInfoField

|  |
| --- |
| // 错误码  TDAIntType ErrorID;  // 错误描述  TDAStringType ErrorMsg; |

### 9）设置心跳超时时间SetHeartBeatTimeout

函数原型：

|  |
| --- |
| bool SetHeartBeatTimeout (int iTimeout, bool bIsStrictCheck = true) ; |

参数：iTimeout : 心跳超时时间，单位为毫秒。

bIsStrictCheck ：是否检查心跳超时。默认为true，检查心跳；如果为false不检查心跳。

注意：设置为false不检查心跳，主要为适配行情前置的版本，有些行情前置不应答心跳。

# 常见问题-期货交易相关

1、合约乘数

合约乘数 = (点值/最小变动单位)\*(进阶单位/10的（行情小数点位数）次方)

说明：

如果不做美债

可以只看 点值/最小变动单位

因为大部分品种都是10进制 后面算出来都是1

2.、保证金

保证金 是下单时验证的保证金， 维持保证金 是其他时候计算的保证金

一个大点的点值=合约点值\*进价单位,除以10的小数点位数次方,再除以最小跳点

3、盈亏

持仓浮盈=持仓数\*（开仓价-最新价）\*一个大点的点值

开仓价和最新价都要转成十进制的价格

十进制的价格=价格整数+小数\*10的小数位数次方除以进价单位

合约点值=336 进价单位=532 小数点位数次方=531 最小跳点=337

4、持仓及权益

持仓：通过请求的持仓明细来计算，按时间先后，买卖直接平仓。

盯市盈亏：（昨结算价-行情最新价）\*持仓数量\*合约点值；

今权益：今结存+浮盈+未结平盈+未到期平盈

今权益= Tags.todayAccount +浮盈+Tags. LMEUnaccountPL+Tags. LMEUnexpiredPL;

今可用：今权益-保证金-冻结资金

今可用=今权益-Tags. Deposit-Tags. frozenDeposit;

持过夜仓，开盘价就是昨天的结算价，浮盈使用昨天的结算价计算。

5、期权

期权：

权利金：期权合约开仓价格\*最小跳点价值 ╱最小跳点 \*手数；

期权价值：期权合约最新价\*最小跳点价值 ╱ 最小跳点 \*手数；

期权净盈亏：期权价值-权利金；

6、撤单

需要填系统号 订单号，商品 交易所和下单方向

7、tag50字段说明，下改撤单中tag50雷同

1. 须使用已在经纪公司登记过的操作员识别码（Operator ID，旧称Tag50）下单，否则订单将会被拒绝。申请开通API 的客户可通过《尽职调查暨操作人授权书》登记申报Tag50；
2. 2.如客户长时间交易，例如超过12小时，需设置多轮管理、操作和监控（至少设置2 个班次）。当负责管理、操作和监控的人员发生变化时，新一班次人员需分配不同于前一班次的Tag50，并在订单中传输。

# 常见问题-股票交易相关

1、 get cash balance (USD)

查资金余额

可取资金: CanCashOutMoneyAmount

可用资金:FBalance (DB) 结构体中(TodayTradableFund)

可用资金包括直达借给客户的钱。

可取资金是客户自己的钱。

所以可用资金可能大于可取资金，分成两个字段。

2,、get total asset value (USD)

查总资产，就是可取资金+持仓市值

市值->没有直接的结果，需要客户自己算。

(最新价\*总持仓量(CanSellShares+FrosenShares))

总持仓量由两部分组成：可卖数量(CanSellShares) + 冻结数量(FrosenShares)

冻结数量： 当客户有排队单未成交时，排队单的数量就是冻结数量

3、get positions (should return list of symbols and its holding)

查持仓(结果中要有股票代码和对应股数)

CanSellShares可卖量，需要考虑排队单的冻结数量(FrosenShares)

4、place mkt order (buy and sell)

下市价买单 和 市价卖单

5、撤单

需要填系统号 订单号，商品 交易所和下单方向

# 常见问题-行情相关

1、Question：

有个zd market的问题请教一下。在返回的CMarketRspMarketDataField的数据结构中，

有两个时间field,一个是time,一个是DataTimestamp,这两个有什么区别？有的产品datatimestamp字段有数据，有的没有，这是什么情况？

-->Answer:

time是转换成中国本地时间

DataTimestamp是交易所发出来的数据时间戳，Unix epoch格式

直连交易所的品种，DataTimestamp会有数据，非直连的为空

2、Question：

收到的价格中，bidprice askprice 有时候为空

-->Answer:

请参照行情开发文档中的数据结构体中QuoteType字段的说明

// 行情区分，分两种情况(意思是Y和2中的成交量可以统计到分钟数据里，Z的不可以)

// 直连交易所的行情 -- Y: 当前回调数据是最新成交数据；Z: 当前回调数据是快照数据

// 非直连交易所的行情 -- 2：当前回调数据包括最新成交数据和盘口变化 Z: 快照数据中成交量// 不能用于统计

TDAStringType QuoteType;