



OST API 说明文档

V2.0.2

东方证券股份有限公司

[HTTPS://WWW.DFZQ.COM.CN/](https://www.dfzq.com.cn/)

文档修订历史

版本	文档发布时间	修订内容
V0.1	201708	创建文档
V0.2	201806	新增期权保证金查询
V0.3	201908	优化 API 性能
V1.0.0	202105	新增融资融券相关操作、ETF 查询等相关接口介绍，新增数据字典，并对文档格式进行调整
V1.0.1	202106	根据 6 月 8 日 API 接口，对文件内容进行更新
V2.0.0	202107	根据最新股票期权 API，完成行权、锁定、撤销行权、查询锁定、查询锁定、查询锁定仓位、查询期权限仓、查询期权限额接口的增加。
V2.0.0	202108	新增期权保证金参数查询请求
V2.0.1	202108	更新 6.2.27 OnRspQryTradingAccount 方法中“可用资金”和“投资者交割保证金”的取值范围
V2.0.2	202201	增加 10 交易所及系统错误码描述

目录

1	介绍.....	9
1.1	接口文件.....	9
1.2	函数命名规则.....	9
2	体系结构.....	9
2.1	通讯模式和数据流.....	9
3	接口模式.....	10
3.1	对话流和查询流编程接口.....	10
3.1.1	请求参数.....	11
3.1.2	响应参数.....	11
3.2	私有流编程接口.....	11
4	运行模式.....	12
4.1	工作线程.....	12
4.2	本地文件.....	12
5	业务与接口对照.....	12
6	开发接口.....	15
6.1	通用规则.....	15
6.1.1	初始化阶段.....	15
6.1.2	功能调用阶段.....	16
6.2	CUTSpi 接口.....	16
6.2.1	OnFrontConnected 方法.....	16
6.2.2	OnRtnOrder 方法.....	16
6.2.3	OnFrontDisconnected 方法.....	18
6.2.4	OnRspError 方法.....	18
6.2.5	OnRtnTrade 方法.....	19
6.2.6	OnErrRtnOrderAction 方法.....	20
6.2.7	OnRspOrderInsert 方法.....	21

6.2.8	OnRspOrderAction 方法.....	23
6.2.9	OnRspLogin 方法.....	24
6.2.10	OnRspUserPasswordUpdate 方法.....	25
6.2.11	OnRspTransferInsert 方法.....	26
6.2.12	OnRtnTransfer 方法.....	27
6.2.13	OnRspFundPaybackInsert 方法.....	28
6.2.14	OnRtnFundPayback 方法.....	28
6.2.15	OnRspStockPaybackInsert 方法.....	29
6.2.16	OnRtnStockPayback 方法.....	30
6.2.17	OnRtnPrivateCreditStock 方法.....	31
6.2.18	OnRtnLock 方法.....	32
6.2.19	OnRspLockInsert 方法.....	33
6.2.20	OnRtnExecOrder 方法.....	34
6.2.21	OnRspExecOrderInsert 方法.....	35
6.2.22	OnRspExecOrderAction 方法.....	36
6.2.23	OnErrRtnExecOrderAction 方法.....	37
6.2.24	OnRspQryInstrument 方法.....	38
6.2.25	OnRspQryDepthMarketData 方法.....	41
6.2.26	OnRspQryInvestorPosition 方法.....	43
6.2.27	OnRspQryTradingAccount 方法.....	45
6.2.28	OnRspQryOptionInstrMarginByVolume 方法.....	47
6.2.29	OnRspQryOptionInstrCommRate 方法.....	48
6.2.30	OnRspQryOrder 方法.....	49
6.2.31	OnRspQryTrade 方法.....	51
6.2.32	OnRspQryInstrumentCommissionRate 方法.....	52
6.2.33	OnRspQryInvestor 方法.....	54
6.2.34	OnRspQryTransfer 方法.....	55
6.2.35	OnRspQryTradingCode 方法.....	56

6.2.36	OnRspQryMaxOrderVolume 方法.....	57
6.2.37	OnRspQryCreditInstrument 方法.....	58
6.2.38	OnRspQryCreditInvestor 方法.....	59
6.2.39	OnRspQryPrivateCreditStock 方法.....	60
6.2.40	OnRspQryCreditConcentration 方法.....	61
6.2.41	OnRspQryCreditFundDetail 方法.....	62
6.2.42	OnRspQryCreditStockDetail 方法.....	63
6.2.43	OnRspQryFundPayback 方法.....	64
6.2.44	OnRspQryStockPayback 方法.....	65
6.2.45	OnRspQryPublicCreditFund 方法.....	67
6.2.46	OnRspQryETFInfo 方法.....	68
6.2.47	OnRspQryETFComponent 方法.....	69
6.2.48	OnRspQryCreditAvailableDetail 方法.....	70
6.2.49	OnRspQryLock 方法.....	71
6.2.50	OnRspQryExecOrder 方法.....	72
6.2.51	OnRspQryLockPosition 方法.....	74
6.2.52	OnRspQryOptPosiLimit 方法.....	75
6.2.53	OnRspQryOptAmountLimit 方法.....	76
6.2.54	OnRspQryOptionMarginAlgo 方法.....	76
6.3	CUTApi 接口.....	77
6.3.1	CreateApi 方法.....	77
6.3.2	GetApiVersion 方法.....	78
6.3.3	Release 方法.....	78
6.3.4	Init 方法.....	78
6.3.5	Join 方法.....	79
6.3.6	RegisterFront 方法.....	79
6.3.7	RegisterSpi 方法.....	79
6.3.8	SubscribePrivateTopic 方法.....	80

6.3.9	SubscribePublicTopic 方法.....	80
6.3.10	ReqOrderInsert 方法.....	80
6.3.11	ReqOrderAction 方法.....	82
6.3.12	ReqLogin 方法.....	83
6.3.13	ReqLogout 方法.....	84
6.3.14	ReqUserPasswordUpdate 方法.....	85
6.3.15	ReqTransferInsert 方法.....	85
6.3.16	ReqFundPaybackInsert 方法.....	86
6.3.17	ReqStockPaybackInsert 方法.....	87
6.3.18	ReqLockInsert 方法.....	88
6.3.19	ReqExecOrderInsert 方法.....	89
6.3.20	ReqExecOrderAction 方法.....	90
6.3.21	ReqQryInstrument 方法.....	90
6.3.22	ReqQryDepthMarketData 方法.....	91
6.3.23	ReqQryInvestorPosition 方法.....	92
6.3.24	ReqQryTradingAccount 方法.....	92
6.3.25	ReqQryOrder 方法.....	93
6.3.26	ReqQryTrade 方法.....	94
6.3.27	ReqQryOptionInstrMarginByVolume 方法.....	95
6.3.28	ReqQryOptionInstrCommRate 方法.....	95
6.3.29	ReqQryInstrumentCommissionRate 方法.....	96
6.3.30	ReqQryInvestor 方法.....	97
6.3.31	ReqQryTransfer 方法.....	97
6.3.32	ReqQryTradingCode 方法.....	98
6.3.33	ReqQryMaxOrderVolume 方法.....	98
6.3.34	ReqQryCreditInstrument 方法.....	99
6.3.35	ReqQryCreditInvestor 方法.....	100
6.3.36	ReqQryPrivateCreditStock 方法.....	100

6.3.37	ReqQryCreditConcentration 方法	101
6.3.38	ReqQryCreditFundDetail 方法	102
6.3.39	ReqQryCreditStockDetail 方法	102
6.3.40	ReqQryFundPayback 方法	103
6.3.41	ReqQryStockPayback 方法	103
6.3.42	ReqQryPublicCreditFund 方法	104
6.3.43	ReqQryETFInfo 方法	105
6.3.44	ReqQryETFComponent 方法	105
6.3.45	ReqQryCreditAvailableDetail 方法	106
6.3.46	ReqQryLock 方法	107
6.3.47	ReqQryExecOrder 方法	107
6.3.48	ReqQryLockPosition 方法	108
6.3.49	ReqQryOptPosiLimit 方法	108
6.3.50	ReqQryOptAmountLimit 方法	109
6.3.51	ReqQryOptionMarginAlgo 方法	110
7	开发示例:	110
8	数据字典:	120
8.1	证件类型	120
8.2	投资者范围	121
8.3	报单操作状态	121
8.4	报单状态	121
8.5	持仓日期类型	121
8.6	产品类型	122
8.7	合约生命周期状态	122
8.8	买卖方向	122
8.9	持仓类型	122
8.10	持仓多空方向	122
8.11	投机套保标志	123

8.12	交易编码类型.....	123
8.13	报单价格条件.....	123
8.14	开平标志.....	124
8.15	报单类型.....	124
8.16	有效期类型.....	124
8.17	成交量类型.....	124
8.18	触发条件.....	125
8.19	操作标志.....	125
8.20	成交类型.....	125
8.21	合约交易状态.....	126
8.22	银证转账类型.....	126
8.23	转账状态.....	126
8.24	大额单边保证金算法.....	127
8.25	期权类型.....	127
8.26	组合类型.....	127
8.27	账户类型.....	127
8.28	交易所代码.....	127
8.29	集中度合约类型.....	128
8.30	替代标志.....	128
8.31	集中度业务类型.....	128
8.32	证券子类别.....	128
8.33	锁定类型.....	130
8.34	执行结果.....	130
9	响应域的错误码描述.....	131
10	交易所及系统错误码描述.....	131
10.1	上海证券交易所错误码.....	131
10.2	深圳证券交易所错误码.....	153
10.3	OST 系统错误码.....	158

1 介绍

本接口说明旨在帮助开发者快速查阅 OST-API 的使用方法、参数说明及注意事项。

1.1 接口文件

OST 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相关交易功能，包括报单录入、报单撤销、报单查询、成交查询、持仓查询等。

该类库包含以下 5 个文件：

文件名	文件描述
UTApi.h	交易接口头文件
UTApiStruct.h	包含了所有用到的数据结构的头文件
UTApiDataType.h	包含了所有用到的数据类型的头文件
utapi.dll	动态链接库二进制文件
utapi.lib	导入库文件

1.2 函数命名规则

文中涉及到的大部分函数命名有一定规则，规则如下：

消息	格式	实例
请求	Req*****	ReqLogin
响应	OnRsp*****	OnRspLogin
查询	ReqQry*****	ReqQryInstrument
查询请求的响应	OnRspQry*****	OnRspQryInstrument
回报	OnRtn*****	OnRtnOrder
错误回报	OnErrRtn*****	OnErrRtnOrderAction

2 体系结构

OST API 使用 TCP 协议与后台进行通讯，后台负责投资者的交易业务处理。

2.1 通讯模式和数据流

交易系统支持三种通讯模式：

- 对话通讯模式，
- 私有通讯模式

- 广播通讯模式

- 1、 对话通讯模式下支持对话数据流和查询数据流：

对话通讯模式是指由 API 主动发起的通讯请求。该请求被后台接收和处理，并给予响应。例如报单、查询等。这种通讯模式与普通的客户/服务器模式相同。

对话数据流是一个双向数据流，API 发送交易请求，后台反馈应答。交易系统不维护对话流的状态。系统故障时，对话数据流会重置，通讯途中的数据可能会丢失。

查询数据流是一个双向数据流，API 发送查询请求，后台反馈应答。交易系统不维护查询流的状态。系统故障时，查询数据流会重置，通讯途中的数据可能会丢失。

- 2、 私有通讯模式下支持私有数据流：

私有通讯模式是指后台主动，向某个特定投资者的发出的信息。例如成交回报等。

私有流是一个单向数据流，由后台发向 API，用于传送投资者私有的通知和回报信息。私有流是一个可靠的数据流，后台维护每个投资者的私有流，在一个交易日内，API 断线后恢复连接时，可以请求后台续传私有流数据。

- 3、 广播通讯模式：

广播通讯模式是指后台主动，向所有投资者都发出相同的信息。例如合约交易状态等。

本接口暂时没有使用广播通讯模式。

3 接口模式

OST API 提供了 2 个接口，分别为 CUTApi 和 CUTSpi。

客户端应用程序可以通过 CUTApi 发出操作请求，通过继承 CUTSpi 并重载回调函数来处理后台服务的响应。

3.1 对话流和查询流编程接口

通过对话流进行通讯的编程接口通常如下：

```
////请求：

int CUTApi::ReqXXX(

    CUTXXXField *pReqXXX,

    int nRequestID)

////响应：

void CUTSpi::OnRspXXX(

    CUTXXXField *pRspXXX,

    CUTRspInfoField *pRspInfo,

    int nRequestID,

    bool bIsLast)
```

3.1.1 请求参数

第一个参数为请求的内容，不能为空。

第二个参数为请求号。请求号由客户端应用程序负责维护，正常情况下每个请求的请求号不要重复。这样在接收后台的响应时，可以得到当时发出请求时填写的请求号，从而可以将响应与请求对应起来。不需要该字段的话也可以全部填 0。

当收到后台服务应答时，CUTSpi 的回调函数会被调用。如果响应数据不止一个，则回调函数会被多次调用。

3.1.2 响应参数

第一个参数为响应的具体数据，如果出错或没有结果有可能为 NULL。

第二个参数为处理结果，表明本次请求的处理结果是成功还是失败。在发生多次回调时，除了第一次回调，其它的回调该参数都可能为 NULL。

第三个参数为请求号，即原来发出请求时填写的请求号。

第四个参数为响应结束标志，表明是否是本次响应的最后一次回调。

3.2 私有流编程接口

私有流中的数据中包含投资者的私有信息，包括报单回报、成交回报等。通过私有流接收回报的编程接口通常如下：

```
void CUTSpi::OnRtnXXX(CUTXXXField *pXXX);  
  
//////或者  
  
void CUTSpi::OnErrRtnXXX(CUTXXXField *pXXX)
```

当收到交易系统通过私有流发布的回报数据时，CUTSpi 的回调函数会被调用。回调函数的参数为回报的具体内容。

4 运行模式

4.1 工作线程

客户端应用程序至少由两个线程组成，一个是应用程序主线程，一个是 API 工作线程。应用程序与后台的通讯是由 API 工作线程驱动的。

CUTApi 提供的接口是线程安全的，可以有多个应用程序线程同时发出请求。

CUTSpi 提供的接口回调是由 API 工作线程驱动，通过实现 SPI 中的接口方法，可以从后台收取所需数据。

如果重载的某个回调函数阻塞，则等于阻塞了 API 工作线程，API 与交易系统的通讯会停止。因此，在 CUTSpi 派生类的回调函数中，通常应迅速返回，可以利用将数据放入缓冲区或通过 Windows 的消息机制来实现。

4.2 本地文件

API 在运行过程中，会将一些数据写入本地文件中。调用 CreateApi 函数，可以传递一个参数，指明存贮本地文件的路径。该路径必须在运行前已创建好。本地文件的扩展名都是“.flow”。

5 业务与接口对照

业务类型	业务	请求接口 响应接口	数据流
登录	登录	CUTApi::ReqLogin	对话流
		CUTSpi::OnRspLogin	
	登出	CUTApi::ReqLogout	对话流
		CUTSpi::OnFrontDisconnected	
	修改用户密码	CUTApi::ReqUserPasswordUpdate	对话流
		CUTSpi::OnRspUserPasswordUpdate	

转账	转账申请	CUTApi:: ReqTransferInsert	对话流
		CUTSpi:: OnRspTransferInsert (正确转账不返回)	
	转账通知	N/A	私有流
		CUTSpi:: OnRtnTransfer	
合约、行情信息查询	转账查询	CUTApi:: ReqQryTransfer	查询流
		CUTSpi:: OnRspQryTransfer	
	合约查询	CUTApi:: ReqQryInstrument	查询流
		CUTSpi:: OnRspQryInstrument	
	信用合约属性查询	CUTApi:: ReqQryCreditInstrument	查询流
		CUTSpi:: OnRspQryCreditInstrument	
ETF	最大下单量查询	CUTApi:: ReqQryMaxOrderVolume	查询流
		CUTSpi:: OnRspQryMaxOrderVolume	
	行情查询	CUTApi:: ReqQryDepthMarketData	查询流
		CUTSpi:: OnRspQryDepthMarketData	
	ETF 信息查询	CUTApi:: ReqQryETFInfo	查询流
		CUTSpi:: OnRspQryETFInfo	
投资者信息查询	ETF 成分股查询	CUTApi:: ReqQryETFComponent	查询流
		CUTSpi:: OnRspQryETFComponent	
	投资者查询	CUTApi:: ReqQryInvestor	查询流
		CUTSpi:: OnRspQryInvestor	
	交易编码查询	CUTApi:: ReqQryTradingCode	查询流
		CUTSpi:: OnRspQryTradingCode	
资金及保证金查询	投资者信用信息查询	CUTApi:: ReqQryCreditInvestor	查询流
		CUTSpi:: OnRspQryCreditInvestor	
	资金查询	CUTApi:: ReqQryTradingAccount	查询流
		CUTSpi:: OnRspQryTradingAccount	
	保证金可用余额明细查询	CUTApi:: ReqQryCreditAvailableDetail	查询流
		CUTSpi:: OnRspQryCreditAvailableDetail	
报单	报单插入	CUTApi:: ReqOrderInsert	对话流
		CUTSpi:: OnRspOrderInsert (正确报单不返回)	
	报单回报	N/A	私有流
		CUTSpi:: OnRtnOrder	
	报单查询	CUTApi:: ReqQryOrder	查询流
		CUTSpi:: OnRspQryOrder	
撤单	报单操作	CUTApi:: ReqOrderAction	对话流
		CUTSpi:: OnRspOrderAction (正确报单操作不返回)	
	报单操作错误回报	N/A	私有流
		CUTSpi:: OnErrRtnOrderAction	
成交	成交回报	N/A	私有流

		CUTSpi::OnRtnTrade	
	成交查询	CUTApi::ReqQryTrade	查询流
		CUTSpi::OnRspQryTrade	
持仓查询	持仓查询	CUTApi::ReqQryInvestorPosition	查询流
		CUTSpi::OnRspQryInvestorPosition	
费率查询	期权合约每手 保证金查询	CUTApi::ReqQryOptionInstrMarginByVolume	查询流
		CUTSpi::OnRspQryOptionInstrMarginByVolume	
	期权手续费率 查询	CUTApi::ReqQryOptionInstrCommRate	查询流
		CUTSpi::OnRspQryOptionInstrCommRate	
	手续费查询	CUTApi::ReqQryInstrumentCommissionRate	查询流
		CUTSpi::OnRspQryInstrumentCommissionRate	
融资	公有融资额度 查询	CUTApi::ReqQryPublicCreditFund	查询流
		CUTSpi::OnRspQryPublicCreditFund	
	投资者融资明 细查询	CUTApi::ReqQryCreditFundDetail	查询流
		CUTSpi::OnRspQryCreditFundDetail	
	集中度参数查 询	CUTApi::ReqQryCreditConcentration	查询流
		CUTSpi::OnRspQryCreditConcentration	
	直接还款	CUTApi::ReqFundPaybackInsert	对话流
		CUTSpi::OnRspFundPaybackInsert (正确直接还款 操作不返回)	
	直接还款回报	N/A	私有流
		CUTSpi::OnRtnFundPayback	
	查询直接还款	CUTApi::ReqQryFundPayback	查询流
		CUTSpi::OnRspQryFundPayback	
融券	投资者可融券 额度查询	CUTApi::ReqQryPrivateCreditStock	查询流
		CUTSpi::OnRtnPrivateCreditStock	
	直接还券	CUTApi::ReqStockPaybackInsert	对话流
		CUTSpi::OnRspStockPaybackInsert	
	直接还券回报	N/A	私有流
		CUTSpi::OnRtnStockPayback	
	查询直接还券	CUTApi::ReqQryStockPayback	查询流
		CUTSpi::OnRspQryStockPayback	
	投资者融券明 细查询	CUTApi::ReqQryCreditStockDetail	查询流
		CUTSpi::OnRspQryCreditStockDetail	
锁定	锁定插入	CUTApi::ReqLockInsert	对话流
		CUTSpi::OnRspLockInsert (正确锁定不返回)	
	锁定回报	N/A	私有流
		CUTSpi::OnRtnLock	
	锁定查询	CUTApi::ReqQryLock	查询流

		CUTSpi:: OnRspQryLock	
	查询锁定仓位	CUTApi:: ReqQryLockPosition	查询流
		CUTSpi:: OnRspQryLockPosition	
行权	行权插入	CUTApi:: ReqExecOrderInsert	对话流
		CUTSpi:: OnRspExecOrderInsert (正确行权不返回)	
	行权回报	N/A	私有流
		CUTSpi:: OnRtnExecOrder	
	撤销行权委托	CUTApi:: ReqExecOrderAction	对话流
		CUTSpi:: OnRspExecOrderAction (正确撤销行权不返回)	
	撤销行权回报	N/A	私有流
		CUTSpi:: OnErrRtnExecOrderAction	
股票期权 限仓限额	查询股票期权 限仓	CUTApi:: ReqQryOptPosiLimit	查询流
		CUTSpi:: OnRspQryOptPosiLimit	
	查询股票期权 限额	CUTApi:: ReqQryOptAmountLimit	查询流
		CUTSpi:: OnRspQryOptAmountLimit	

交易接口和私有流接口会有相互关联，一般只会收到其中一个。如用户报单录入 ReqOrderInsert，如果报单正确，会收到私有流报单回报 OnRtnOrder；如果报单错误，会收到对话流报单响应 OnRspOrderInsert。报单进入交易系统后，如果报单的交易状态发生变化，还会收到相应状态的报单回报 OnRtnOrder。如果报单被撮合(部分)成交，就会收到成交回报 OnRtnTrade。

6 开发接口

6.1 通用规则

客户端和后台的通讯过程分为 2 个阶段：初始化阶段和功能调用阶段。

6.1.1 初始化阶段

在初始化阶段，程序必须完成如下步骤（具体代码请参考开发实例）：

- 1、产生一个 CUTApi 实例
- 2、产生一个事件处理的实例
- 3、注册一个事件处理的实例
- 4、订阅私有流
- 5、订阅公共流

6、 设置后台前置的地址

6.1.2 功能调用阶段

在功能调用阶段，程序可以任意调用交易接口中的请求方法，如 ReqOrderInsert 等。同时按照需要处理回调接口。

其他注意事项：

- 1、 API 请求的输入参数不能为 NULL。
- 2、 API 请求的返回参数，0 表示正确，其他表示错误，详细错误编码请查表。

6.2 CUTSpi 接口

CUTSpi 实现了事件通知接口。用户必须派生。CUTSpi 接口，编写事件处理方法来处理感兴趣的事件。

6.2.1 OnFrontConnected 方法

➤ 功能描述

当客户端与后台建立起通信连接时（还未登录前），该方法被调用。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。

➤ 函数原型

```
void OnFrontConnected() {};
```

6.2.2 OnRtnOrder 方法

➤ 功能描述

报单回报。

当客户端进行报单录入及其它原因（如部分成交）导致报单状态发生变化时，交易系统会主动通知客户端，该方法会被调用。

➤ 函数原型

```
void OnRtnOrder(CUTOrderField *pOrder) {};
```

➤ 函数参数

pOrder：指向报单信息结构的地址。

➤ 函数结构

域名称: CUTOrderField

域描述: 报单

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
OrderRef		报单引用, 关键字
Direction	参见 8.8 买卖方向	买卖方向
OffsetFlag	参见 8.14 开平标志	开平标志
HedgeFlag	参见 8.11 投机套保标志	投机套保标志
OrderPriceType	参见 8.13 报单价格条件	报单价格条件
VolumeTotalOriginal		数量
LimitPrice		价格
TimeCondition	参见 8.16 有效期类型	有效期类型
VolumeCondition	参见 8.17 成交量类型	成交量类型
ContingentCondition	参见 8.18 触发条件	触发条件
GTDDate		GTD 日期
MinVolume		最小成交量
IsAutoSuspend		自动挂起标志
UserForceClose		用户强评标志
StopPrice		止损价
IsSwapOrder		互换单标志
TraderID		交易所交易员代码
OrderLocalID		本地报单编号
ParticipantID		会员代码
ClientID		客户代码
TradingDay	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	交易日
UserProductInfo		用户端产品信息
BranchID		营业部编号
InstallID		报盘安装编号
IPAddressAsInt		IP 地址
MacAddressAsLong		Mac 地址
OrderSysID		报单编号
OrderStatus	参见 8.4 报单状态	报单状态
OrderType	参见 8.15 报单类型	报单类型
VolumeTraded		今成交数量
VolumeTotal		剩余数量

InsertTime	为 int 类型数据，例如 9:02:31:698，则该值为 90231698	委托时间
CancelTime	为 int 类型数据，例如 9:02:31:698，则该值为 90231698	撤销时间
SequenceNo		序号
FrontID		前置编号
SessionID		会话编号
ExchangeErrorID		交易所错误代码

6.2.3 OnFrontDisconnected 方法

➤ 功能描述

当客户端与后台通信连接断开时，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。

➤ 函数原型

```
void OnFrontDisconnected(int nReason) {};
```

➤ 函数参数

nReason: 连接断开原因，连接断开原因，注意该返回值为 10 进制数，所以要转换成 16 进制再对照下列错误号。

错误码	错误码含义
0x1001	网络读失败
0x1002	网络写失败
0x2001	接收心跳超时
0x2002	发送心跳失败
0x2003	收到错误报文

6.2.4 OnRspError 方法

➤ 功能描述

针对用户请求后台不支持的功能时的出错通知。

➤ 函数原型

```
void OnRspError(CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pRspInfo: 指向响应信息的地址。

pRspInfo: 返回用户响应信息的地址。特别注意在有连续的成功响应数据时，中间有可能返回 NULL，但第一次不会，以下同。错误代码为 0 时，表示操作成功，以下同。

nRequestID: 返回用户操作请求的 ID，该 ID 由用户在操作请求时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.5 OnRtnTrade 方法

➤ 功能描述

成交回报。当发生成交时交易系统会通知客户端，该方法会被调用。

➤ 函数原型

```
void OnRtnTrade(CUTTradeField *pTrade) {};
```

➤ 函数参数

pTrade: 指向成交信息结构的地址。

➤ 函数结构

域名称: CUTTradeField

域描述: 成交信息

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
OrderRef		报单引用
TradeID		成交编号，关键字
Direction	参见 8.8 买卖方向	买卖方向，关键字

OrderSysID		报单编号
ClientID		客户代码
OffsetFlag	参见 8.14 开平标志	开平标志
HedgeFlag	参见 8.11 投机套保标志	投机套保标志
Price		价格
Volume		数量
TradeDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	成交日期
TradeTime	为 int 类型数据, 例如 9:02:31:698, 则该值为 90231698	成交时间
TradeType	参见 8.20 成交类型	成交类型
TraderID		交易所交易员代码
OrderLocalID		本地报单编号
SequenceNo		序号
TradingDay	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	交易日
InstallID		报盘安装编号

6.2.6 OnErrRtnOrderAction 方法

➤ 功能描述

报单操作错误回报。报单操作被交易所打回时, 该方法会被调用。

➤ 函数原型

```
void OnErrRtnOrderAction(CUTOrderActionField *pOrderAction) {};
```

➤ 函数参数

pOrderAction: 指向报单操作结构的地址。

➤ 函数结构

域名称: CUTOrderActionField

域描述: 报单操作

字段代码	取值范围	字段描述
OrderActionRef		报单操作引用
FrontID		前置编号
SessionID		会话编号
OrderRef		报单引用
ActionFlag	参见 8.19 操作标志	操作标志

LimitPrice		价格
VolumeChange		数量变化
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
OrderSysID		报单编号
InstrumentID		合约代码
ActionDate	为 int 类型数据，例如 2021 年 6 月 3 日，则该值为 20210603	操作日期
ActionTime	为 int 类型数据，例如 9:02:31:698，则该值为 90231698	操作时间
TraderID		交易所交易员代码
OrderLocalID		本地报单编号
ActionLocalID		操作本地编号
ParticipantID		会员代码
ClientID		客户代码
OrderActionStatus	参见 8.3 报单操作状态	报单操作状态
Direction	参见 8.8 买卖方向	买卖方向，关键字
BranchID		营业部编号
OrderType	参见 8.15 报单类型	报单类型
InstallID		报盘安装编号
ExchangeErrorID		交易所错误信息
ActionFrontID		撤单前置编号
ActionSessionID		撤单会话编号
IPAddressAsInt		IP 地址
MacAddressAsLong		Mac 地址

6.2.7 OnRspOrderInsert 方法

➤ 功能描述

报单错误应答。当客户端发出报单录入（RepOrderInsert）指令后，后台报错时，该方法会被调用。

➤ 函数原型

```
void OnRspOrderInsert(CUTInputOrderField *pInputOrder, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pInputOrder: 指向报单录入结构的地址，包含了提交报单录入时的输入数据。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回报单录入操作请求的 ID, 该 ID 由用户在报单录入时指定。

bIsLast: 指示该次返回是否针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTInputOrderField

域描述: 报单录入

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
OrderRef		报单引用
Direction	参见 8.8 买卖方向	买卖方向, 关键字
OffsetFlag	参见 8.14 开平标志	开平标志
HedgeFlag	参见 8.11 投机套保标志	投机套保标志
OrderPriceType	参见 8.13 报单价格条件	报单价格条件
VolumeTotalOriginal		数量
LimitPrice		价格
TimeCondition	参见 8.16 有效期类型	有效期类型
VolumeCondition	参见 8.17 成交量类型	成交量类型
ContingentCondition	参见 8.18 触发条件	触发条件
GTDDate		GTD 日期
MinVolume		最小成交量
IsAutoSuspend		自动挂起标志
UserForceClose		用户强评标志
StopPrice		止损价
IsSwapOrder		互换单标志
IPAddressAsInt		IP 地址
MacAddressAsLong		Mac 地址

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.8 OnRspOrderAction 方法

➤ 功能描述

报单操作应答。报单操作目前仅包括报单的撤销。当客户端发出报单操作 (ReqOrderAction) 指令后, 后台报错时, 该方法会被调用。

➤ 函数原型

```
void OnRspOrderAction(CUTInputOrderActionField *pInputOrderAction,
  CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pInputOrderAction: 指向报单操作结构的地址, 包含了提交报单操作的输入数据。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户报单操作请求的 ID, 该 ID 由用户在报单操作时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTInputOrderActionField

域描述: 报单操作

字段代码	取值范围	字段描述
OrderActionRef		报单操作引用
FrontID		前置编号
SessionID		会话编号
OrderRef		报单引用
ActionFlag	参见 8.19 操作标志	操作标志
LimitPrice		价格
VolumeChange		数量变化
IPAddressAsInt		IP 地址
MacAddressAsLong		Mac 地址

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.9 OnRspLogin 方法

➤ 功能描述

当客户端发出登录请求之后，后台返回响应时，该方法会被调用，通知客户端登录是否成功。

➤ 函数原型

```
void OnRspLogin(CUTRspLoginField *pRspLogin, CUTRspInfoField *pRspInfo,
int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pRspLogin: 返回用户登录信息的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户登录请求的 ID，该 ID 由用户在登录时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTRspLoginField

域描述: 用户登录信息

字段代码	取值范围	字段描述
TradingDay	为 int 类型数据，例如 2021 年 6 月 3 日，则该值为 20210603	交易日
LoginTime	为 int 类型数据，例如 9:02:31:698，则该值为 90231698	登录成功时间
UserID		用户代码
SystemName		交易系统名称
FrontID		前置编号
SessionID		会话编号
PrivateSeq		私有流序号
PublicSeq		公有流序号
MultiAddress		组播地址

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码

ErrorMsg		错误信息
----------	--	------

6.2.10 OnRspUserPasswordUpdate 方法

➤ 功能描述

用户密码修改应答。当客户端发出用户密码修改指令后，后台返回响应时，该方法会被调用。

➤ 函数原型

```
void OnRspUserPasswordUpdate(CUTUserPasswordUpdateField *pUserPasswordUpdate, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pUserPasswordUpdate: 指向用户密码修改结构的地址，包含了用户密码修改请求的输入数据。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户密码修改请求的 ID，该 ID 由用户在密码修改时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTUserPasswordUpdateField

域描述: 用户密码修改

字段代码	取值范围	字段描述
UserID		用户代码
OldPassword		原来的口令
NewPassword		新的口令

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.11 OnRspTransferInsert 方法

➤ 功能描述

请求转账录入响应。当客户端发出转账请求，后台报错时，该方法会被调用。

➤ 函数原型

```
void OnRspTransferInsert(CUTInputTransferField *pInputTransfer,
  CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pInputTransfer: 指向转账录入的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户转账录入请求的 ID，该 ID 由用户在转账录入时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTInputTransferField

域描述: 转账录入

字段代码	取值范围	字段描述
InvestorID		投资者代码
TransferRef		转账引用
TransferType	参见 8.22 银证转账类型	转账类型
Deposit		金额
CurrencyID	目前仅支持人民币，该值为“CNY”	币种代码
AccType	参见 8.27 账户类型	账户类型
BankID		银行代码
FundPassword		资金密码
BankPassword		银行密码

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.12 OnRtnTransfer 方法

➤ 功能描述

转账通知。转账录入或者转账状态发生变化时，该方法会被调用。

➤ 函数原型

```
void OnRtnTransfer(CUTTransferField *pTransfer) {};
```

➤ 函数参数

pTransfer: 指向转账结构的地址。

➤ 函数结构

域名称: CUTTransferField

域描述: 转账

字段代码	取值范围	字段描述
InvestorID		投资者代码
TransferRef		转账引用
TransferType	参见 8.22 银证转账类型	转账类型
Deposit		金额
CurrencyID	目前仅支持人民币, 该值为 “CNY”	币种代码
AccType	参见 8.27 账户类型	账户类型
BankID		银行代码
FundPassword		资金密码
BankPassword		银行密码
FrontID		前置编号
SessionID		会话编号
TransferLocalID		本地转账号, 关键字
TransferSysID		转账系统号
TransferStatus	参见 8.23 转账状态	转账状态
ExchangeErrorID		交易所错误代码
InsertDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	转账日期
InsertTime	为 int 类型数据, 例如 9:02:31:698, 则该值为 90231698	转账时间

6.2.13 OnRspFundPaybackInsert 方法

➤ 功能描述

直接还款错误回报。当客户端发出直接还款请求, 后台报错时, 该方法会被调用。

➤ 函数原型

```
void OnRspFundPaybackInsert(CUTInputFundPaybackField
*pInputFundPayback, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast)
{};
```

➤ 函数参数

pInputFundPayback: 指向输入直接还款的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户输入直接还款请求的 ID, 该 ID 由用户在输入直接还款时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTInputFundPaybackField

域描述: 输入直接还款

字段代码	取值范围	字段描述
InvestorID		投资者代码
CurrencyID	目前仅支持人民币, 该值为“CNY”	币种代码
FundPaybackRef		直接还款引用
Amount		还款金额
IPAddressAsInt		IP 地址
MacAddressAsLong		Mac 地址

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.14 OnRtnFundPayback 方法

➤ 功能描述

直接还款通知。直接还款录入或者直接还款状态发生变化时, 该方法会被调用。

➤ 函数原型

```
void OnRtnFundPayback(CUTFundPaybackField *pFundPayback) {};
```

➤ 函数参数

pFundPayback: 指向直接还款结构的地址。

➤ 函数结构

域名称: CUTFundPaybackField

域描述: 直接还款

字段代码	取值范围	字段描述
InvestorID		投资者代码
CurrencyID	目前仅支持人民币, 该值为“CNY”	币种代码
FundPaybackRef		直接还款引用
Amount		还款金额
FrontID		前置编号
SessionID		会话编号
FundPaybackLocalID		本地直接还款号, 关键字
InsertDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	转账日期
InsertTime	为 int 类型数据, 例如 9:02:31:698, 则该值为 90231698	转账时间
IPAddressAsInt		IP 地址
MacAddressAsLong		Mac 地址

6.2.15 OnRspStockPaybackInsert 方法

➤ 功能描述

直接还券错误回报。当客户端发出直接还券请求, 后台报错时, 该方法会被调用。

➤ 函数原型

```
void OnRspStockPaybackInsert(CUTInputStockPaybackField *pInputStockPayback, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pInputStockPayback: 指向直接还券结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户直接还券请求的 ID, 该 ID 由用户在直接还券时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTInputStockPaybackField

域描述: 直接还券

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
StockPaybackRef		直接还券引用
Volume		直接还券数量
IPAddressAsInt		IP 地址
MacAddressAsLong		Mac 地址

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.16 OnRtnStockPayback 方法

➤ 功能描述

直接还券通知。直接还券录入或者直接还券状态发生变化时, 该方法会被调用。

➤ 函数原型

```
void OnRtnStockPayback(CUTStockPaybackField *pStockPayback) {};
```

➤ 函数参数

pStockPayback: 指向直接还券结构的地址。

➤ 函数结构

域名称: CUTStockPaybackField

域描述: 直接还券

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码

StockPaybackRef		直接还券引用
Volume		直接还券数量
FrontID		前置编号
SessionID		会话编号
TraderID		交易所交易员代码
StockPaybackLocalID		本地直接还券号
ClientID		客户代码
InstallID		报盘安装编号
InsertDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	直接还券日期
InsertTime	为 int 类型数据, 例如 9:02:31:698, 则该值为 90231698	直接还券时间
StockPaybackSysID		直接还券系统号
StockPaybackStatus	参见 8.3 报单操作状态	直接还券系统号
ExchangeErrorID		交易所错误代码
BranchID		营业部编号
IPAddressAsInt		IP 地址
MacAddressAsLong		Mac 地址

6.2.17 OnRtnPrivateCreditStock 方法

➤ 功能描述

同步组件返回的券源保底额度信息。查询投资者可融券时, 该方法会被调用。

➤ 函数原型

```
void OnRtnPrivateCreditStock(CUTPrivateCreditStockField *pPrivateCreditStock) {};
```

➤ 函数参数

pPrivateCreditStock: 指向私有融券额度结构的地址。

➤ 函数结构

域名称: CUTPrivateCreditStockField

域描述: 私有融券额度

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码

Volume		直接还券数量
BeginDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	直接还券日期
EndDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	直接还券日期
ReserveVolume		可融券数量(含到期)

6.2.18 OnRtnLock 方法

➤ 功能描述

锁定回报。

当客户端进行锁定导致锁定委托状态发生变化时, 交易系统会主动通知客户端, 该方法会被调用。

➤ 函数原型

```
void OnRtnLock(CUTLockField *pLock) {};
```

➤ 函数参数

pLock: 指向锁定结构的地址。

➤ 函数结构

域名称: CUTLockField

域描述: 锁定

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
LockRef		锁定引用, 关键字
Volume		数量
LockType	参见 8.33 锁定类型	锁定类型
TraderID		交易所交易员代码
LockLocalID		本地锁定编号
ClientID		客户代码
TradingDay	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	交易日
BranchID		营业部编号
InstallID		报盘安装编号

IPAddressAsInt		IP 地址
MacAddressAsLong		Mac 地址
LockSysID		锁定编号
LockStatus	参见 8.3 报单操作状态	锁定状态
InsertDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	报单日期
InsertTime	为 int 类型数据, 例如 9:02:31:698, 则该值为 90231698	委托时间
CancelTime	为 int 类型数据, 例如 9:02:31:698, 则该值为 90231698	撤销时间
SequenceNo		序号
FrontID		前置编号
SessionID		会话编号
ExchangeErrorID		交易所错误代码

6.2.19 OnRspLockInsert 方法

➤ 功能描述

锁定错误应答。当客户端发出锁定 (RepLockInsert) 指令后, 后台报错时, 该方法会被调用。

➤ 函数原型

```
void OnRspLockInsert(CUTInputLockField *pInputLock, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pInputLock: 指向输入锁定结构的地址, 包含了提交输入锁定时的输入数据。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回输入锁定操作请求的 ID, 该 ID 由用户在锁定录入时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTInputLockField

域描述: 输入锁定

字段代码	取值范围	字段描述
------	------	------

InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
LockRef		锁定引用
Volume		数量
LockType	参见 8.33 锁定类型	锁定类型
IPAddressAsInt		IP 地址
MacAddressAsLong		Mac 地址

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.20 OnRtnExecOrder 方法

➤ 功能描述

行权回报。

当客户端进行行权操作导致行权委托状态发生变化时,交易系统会主动通知客户端,该方法会被调用。

➤ 函数原型

```
void OnRtnExecOrder(CUTExecOrderField *pExecOrder) {};
```

➤ 函数参数

pExecOrder: 指向行权结构的地址。

➤ 函数结构

域名称: CUTExecOrderField

域描述: 行权

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
ExecOrderRef		报单引用, 关键字
Volume		数量

TraderID		交易所交易员代码
ExecOrderLocalID		本地报单编号
ClientID		客户代码
TradingDay	为 int 类型数据，例如 2021 年 6 月 3 日，则该值为 20210603	交易日
BranchID		营业部编号
InstallID		报盘安装编号
IPAddressAsInt		IP 地址
MacAddressAsLong		Mac 地址
ExecOrderSysID		行权编号
ExecResult	参见 8.34 执行结果	执行结果
InsertDate	为 int 类型数据，例如 2021 年 6 月 3 日，则该值为 20210603	报单日期
InsertTime	为 int 类型数据，例如 9:02:31:698，则该值为 90231698	委托时间
CancelTime	为 int 类型数据，例如 9:02:31:698，则该值为 90231698	撤销时间
SequenceNo		序号
FrontID		前置编号
SessionID		会话编号
ExchangeErrorID		交易所错误代码

6.2.21 OnRspExecOrderInsert 方法

➤ 功能描述

行权错误应答。当客户端发出行权（RepExecOrderInsert）指令后，后台报错时，该方法会被调用。

➤ 函数原型

```
void OnRspExecOrderInsert(CUTInputExecOrderField *pInputExecOrder,
    CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pInputExecOrder: 指向输入行权结构的地址，包含了提交输入行权时的输入数据。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回输入行权操作请求的 ID，该 ID 由用户在行权录入时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTInputExecOrderField

域描述: 输入行权

字段代码	请求时填写	取值范围	字段描述
InvestorID	必填		投资者代码
ExchangeID	必填	参见 8.28 交易所代码	交易所代码
InstrumentID	必填		合约代码
ExecOrderRef	选填		报单引用
Volume	必填		数量
IPAddressAsInt	选填		IP 地址
MacAddressAsLong	选填		Mac 地址

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.22 OnRspExecOrderAction 方法

➤ 功能描述

行权操作应答。行权操作目前仅包括撤销行权委托。当客户端发出报单操作 (ReqExecOrderAction) 指令后, 后台报错时, 该方法会被调用。

➤ 函数原型

```
void OnRspExecOrderAction(CUTInputExecOrderActionField *pInputExecOrderAction, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pInputExecOrderAction: 指向行权操作结构的地址, 包含了提交行权操作的输入数据。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户行权操作请求的 ID, 该 ID 由用户在行权操作时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

结构名称: CUTInputExecOrderActionField

结构描述: 输入行权操作

字段代码	请求时填写	取值范围	字段描述
ExecOrderActionRef	选填		行权操作引用
FrontID	必填		前置编号
SessionID	必填		会话编号
ExecOrderRef	必填		行权引用
IPAddressAsInt	选填		IP 地址
MacAddressAsLong	选填		Mac 地址

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.23 OnErrRtnExecOrderAction 方法

➤ 功能描述

行权操作错误回报。行权操作被交易所打回时, 该方法会被调用。

➤ 函数原型

```
void OnErrRtnExecOrderAction(CUTExecOrderActionField *pExecOrderAction)
{
};
```

➤ 函数参数

pExecOrderAction: 指向行权操作结构的地址。

➤ 函数结构

域名称: CUTExecOrderActionField

域描述: 行权操作

字段代码	取值范围	字段描述
ExecOrderActionRef		行权操作引用
FrontID		前置编号
SessionID		会话编号

ExecOrderRef		行权引用
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
ExecOrderSysID		报单编号
InstrumentID		合约代码
ActionDate	为 int 类型数据，例如 2021 年 6 月 3 日，则该值为 20210603	操作日期
ActionTime	为 int 类型数据，例如 9:02:31:698，则该值为 90231698	操作时间
TraderID		交易所交易员代码
ExecOrderLocalID		本地报单编号
ActionLocalID		操作本地编号
ClientID		客户代码
OrderActionStatus	参见 8.3 报单操作状态	报单操作状态
BranchID		营业部编号
InstallID		报盘安装编号
ExchangeErrorID		交易所错误信息
ActionFrontID		撤单前置编号
ActionSessionID		撤单会话编号
IPAddressAsInt		IP 地址
MacAddressAsLong		Mac 地址

6.2.24 OnRspQryInstrument 方法

➤ 功能描述

请求查询合约响应。当客户端发出请求查询合约指令后，交易系统返回响应时，该方法会被调用。

➤ 函数原型

```
void OnRspQryInstrument(CUTInstrumentField *pInstrument, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pInstrument: 指向合约结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回合约查询请求的 ID，该 ID 由用户在合约查询时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTInstrumentField

域描述: 合约

字段代码	取值范围	字段描述
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
InstrumentName		合约名称
ProductID		产品代码
ProductClass	参见 8.6 产品类型	产品类型
DeliveryYear		交割年份
DeliveryMonth		交割月
MaxMarketOrderVolume		市价单最大下单量
MinMarketOrderVolume		市价单最小下单量
MaxLimitOrderVolume		限价单最大下单量
MinLimitOrderVolume		限价单最小下单量
VolumeMultiple	上交所债券为 10, 其它证券现货都是 1	合约数量乘数
PriceTick		最小变动价位
CreateDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	创建日
OpenDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	上市日
ExpireDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	到期日
StartDelivDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	开始交割日
EndDelivDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	结束交割日
IsTrading		当前是否交易
InstLifePhase		合约生命周期状态
PositionType	参见 8.9 持仓类型	持仓类型
PositionDateType	参见 8.5 持仓日期类型	持仓日期类型
MaxMarginSideAlgorithm	参见 8.24 大额单边保证金算法	是否使用大额单边保证金算法
OptionsType	参见 8.25 期权类型	期权类型
CombinationType	参见 8.26 组合类型	组合类型
UnderlyingInstrID	参见 8.32 证券子类别	证券时表示子类别; 期权时表示标的

StrikePrice		执行价；债券时表示利息
UnderlyingMultiple		合约基础商品乘数
MinBuyVolume		限价单最小买下单单位
MinSellVolume		限价单最小卖下单单位
InstrumentCode		合约标识码；证券时内部使用, 个股期权时表示合约英文名 证券时： 第一位, S 说明是 ST 合约, P 说明退市整理合约； 第二位 0 表示 T+0, 1 表示 T+1； 第三位 Z 表示逆回购 第四位 L 表示债券合格投资者, M 表示债券专业投资者 第五位 1 表示创业板注册制
CurrencyID	目前仅支持人民币, 该值为“CNY”	币种代码
MinMarketBuyVolume		市价单最小买下单单位
MinMarketSellVolume		市价单最小卖下单单位

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.25 OnRspQryDepthMarketData 方法

➤ 功能描述

请求查询深度行情响应。当客户端发出请求查询行情指令后, 交易系统返回响应时, 该方法会被调用。

➤ 函数原型

```
void OnRspQryDepthMarketData(CUTDepthMarketDataField *pDepthMarketData,
    CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pDepthMarketData: 指向深度行情结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回行情深度查询请求的 ID, 该 ID 由用户在查询时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTDepthMarketDataField

域描述: 深度行情

字段代码	取值范围	字段描述
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
TradingDay	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	交易日
PreSettlementPrice		上次结算价; 基金为上一日净值
PreClosePrice		昨收盘价
PreOpenInterest		昨持仓量
UpperLimitPrice		涨停板价
LowerLimitPrice		跌停板价
ActionDay	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	自然日
UpdateTime	为 int 类型数据, 例如 9:02:31:698, 则该值为 90231698	最后修改时间
OpenPrice		今开盘价
ClosePrice		今收盘价
SettlementPrice		本次结算价; 基金为实时参考净值
HighestPrice		最高价
LowestPrice		最低价
InstrumentStatus	参见 8.21 合约交易状态	合约交易状态
LastPrice		最新价
Volume		数量
Turnover		成交金额
OpenInterest		持仓量
BidPrice1		申买价一

AskPrice1		申卖价一
BidVolume1		申买量一
AskVolume1		申卖量一
BidPrice2		申买价二
AskPrice2		申卖价二
BidVolume2		申买量二
AskVolume2		申卖量二
BidPrice3		申买价三
AskPrice3		申卖价三
BidVolume3		申买量三
AskVolume3		申卖量三
BidPrice4		申买价四
AskPrice4		申卖价四
BidVolume4		申买量四
AskVolume4		申卖量四
BidPrice5		申买价五
AskPrice5		申卖价五
BidVolume5		申买量五
AskVolume5		申卖量五

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.26 OnRspQryInvestorPosition 方法

➤ 功能描述

投资者持仓查询应答。当客户端发出投资者持仓查询指令后, 交易系统返回响应时, 该方法会被调用。

➤ 函数原型

```
void OnRspQryInvestorPosition(CUTInvestorPositionField *pInvestorPosition, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pInvestorPosition: 指向投资者持仓应答结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回投资者持仓查询请求的 ID, 该 ID 由用户在查询时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTInvestorPositionField

域描述: 投资者持仓应答

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
HedgeFlag	参见 8.11 投机套保标志	投机套保标志
PosiDirection	参见 8.10 持仓多空方向	持仓多空方向
YdPosition		上日持仓
Position		总持仓
LongFrozen		多头冻结
ShortFrozen		空头冻结
LongFrozenAmount		多头冻结金额
ShortFrozenAmount		空头冻结金额
OpenVolume		开仓量
CloseVolume		平仓量
OpenAmount		开仓金额
CloseAmount		平仓金额
PositionCost	持仓成本=昨开仓成本+开仓成交金额-平仓成交金额	持仓成本
UseMargin		占用的保证金
FrozenMargin		冻结的保证金
FrozenCash	冻结权利金	冻结的资金
FrozenCommission		冻结的手续费
CashIn	权利金	资金差额
Commission		手续费
CloseProfit		平仓盈亏
PositionProfit		持仓盈亏
PreSettlementPrice		上次结算价
SettlementPrice		本次结算价

TradingDay	为 int 类型数据，例如 2021 年 6 月 3 日，则该值为 20210603	交易日
OpenCost	开仓成本=昨持仓成本+开仓成交金额-持仓均价*平仓成交手数*合约乘数	开仓成本
CombPosition		组合成交形成的持仓
CloseProfitByDate		逐日盯市平仓盈亏
CloseProfitByTrade		逐笔对冲平仓盈亏
TodayPosition		今日持仓
MarginByVolume		每手保证金
StrikeFrozen		执行冻结
StrikeFrozenAmount		执行冻结金额
AbandonFrozen		放弃执行冻结
CreditFund		融资金额
RedemptionPosition		ETF 申赎持仓(记入总持仓)
RedemptionFrozen		申赎或直接还券造成的冻结
RedemptionPositionFrozen		买卖造成的 ETF 申赎持仓冻结
CreditFundVolume		融资数量
CreditFundCommission		融资今仓手续费(不可还)

结构名称：CUTRspInfoField

结构描述：响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.27 OnRspQryTradingAccount 方法

➤ 功能描述

请求查询资金账户响应。当客户端发出请求查询资金账户指令后，交易系统返回响应时，该方法会被调用。

➤ 函数原型

```
void OnRspQryTradingAccount(CUTTradingAccountField *pTradingAccount,
CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pTradingAccount: 指向资金账户结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回资金账户查询请求的 ID, 该 ID 由用户在资金账户查询时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTTradingAccountField

域描述: 资金账户

字段代码	取值范围	字段描述
AccountID		投资者账号
CurrencyID	目前仅支持人民币, 该值为“CNY”	币种代码
AccType	参见 8.27 账户类型	账户类型
TradingDay	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	交易日
PreMortgage	0	上次质押金额
PreCredit	0	上次信用额度
PreDeposit		上次存款额
PreBalance	0	上次结算准备金
PreMargin		上次占用的保证金
InterestBase		利息基数
Interest		利息收入
Deposit		入金金额
Withdraw		出金金额
FrozenMargin		冻结的保证金
FrozenCash	冻结权利金+行权冻结资金	冻结的资金
FrozenCommission		冻结的手续费
CurrMargin		当前保证金总额
CashIn	权利金	资金差额
Commission		手续费
CloseProfit		平仓盈亏
PositionProfit		持仓盈亏
Balance	0	期货结算准备金
Available	如为股票期权系统, 该值为静态可用资金, 计算公式=权益-保证金-冻结保证金-冻结手续	可用资金

	费-冻结权利金；实际系统控制的可用为动态可用，计算公式 = 权益-MAX（保证金，实时保证金/0.9）-冻结保证金-冻结手续费-冻结权利金	
WithdrawQuota	可用资金-MAX（权利金，0）	可取资金
Reserve		基本准备金
Credit		信用额度
Mortgage		质押金额
DeliveryMargin	如为股票期权系统，该值为实时保证金	投资者交割保证金
Tax		税费
CreditFund		融资金额
CreditFundFrozen		冻结的融资金额
CreditStock		融券金额
CreditStockFrozen		冻结的融券金额
CreditAvailable		保证金可用余额
CreditAvailableParam		保证金余额变量
CreditStockCloseFrozen		融券平仓冻结(计算保证金可用余额和非高流通开仓可用)
CreditCollBuyFrozenOffset		担保品买入冻结偏移(计算保证金可用余额)
CreditLiquidCost		高流通证券开仓成本
CreditInterest		融资融券利息(计算保证金可用余额和还款)
CreditTotalValue		证券市值
CreditStockValue		融券当前市值
CreditFundPayback		今日已还融资金额(包括利息)
CreditInterestProcess		融资融券未结利息(不可还)
CreditFundCommission		今日融资手续费(不可还)

结构名称：CUTRspInfoField

结构描述：响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码

ErrorMsg		错误信息
----------	--	------

6.2.28 OnRspQryOptionInstrMarginByVolume 方法

➤ 功能描述

请求查询期权合约每手保证金响应。交易系统返回响应时，该方法会被调用。

➤ 函数原型

```
void
OnRspQryOptionInstrMarginByVolume(CUOptionInstrMarginByVolumeField
*pOptionInstrMarginByVolume, CUTRspInfoField *pRspInfo, int nRequestID, bool
bIsLast) {};
```

➤ 函数参数

pOptionInstrMarginByVolume: 指向期权合约每手保证金结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回期权合约每手保证金查询请求 ID, 该 ID 由用户在查询时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUOptionInstrMarginByVolumeField

域描述: 期权合约每手保证金

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
HedgeFlag	参见 8.11 投机套保标志	投机套保标志
MarginByVolume		每手合约保证金

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.29 OnRspQryOptionInstrCommRate 方法

➤ 功能描述

请求查询期权合约手续费率响应。交易系统返回响应时，该方法会被调用。

➤ 函数原型

```
void OnRspQryOptionInstrCommRate(CUOptionInstrCommRateField *pOptionInstrCommRate, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pOptionInstrCommRate: 指向期权合约手续费率结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回期权合约手续费率查询请求 ID, 该 ID 由用户在查询时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUOptionInstrCommRateField

域描述: 期权合约手续费率

字段代码	取值范围	字段描述
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
InvestorRange	参见 8.2 投资者范围	投资者范围
InvestorID		投资者代码
HedgeFlag	参见 8.11 投机套保标志	投机套保标志
PosiDirection	参见 8.10 持仓多空方向	持仓多空方向
OpenRatioByMoney		开仓手续费率
OpenRatioByVolume		开仓手续费
CloseRatioByMoney		平仓手续费率
CloseRatioByVolume		平仓手续费
CloseTodayRatioByMoney		平今手续费率
CloseTodayRatioByVolume		平今手续费
StrikeRatioByMoney		执行手续费率
StrikeRatioByVolume		执行手续费
CommByOrder		佣金按笔数

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.30 OnRspQryOrder 方法

➤ 功能描述

报单查询请求。当客户端发出报单查询指令后, 后台返回响应时, 该方法会被调用。

➤ 函数原型

```
void OnRspQryOrder(CUTOrderField *pOrder, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pOrder: 指向报单信息结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户报单查询请求的 ID, 该 ID 由用户在报单查询时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTOrderField

域描述: 报单信息

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
OrderRef		报单引用, 关键字
Direction	参见 8.8 买卖方向	买卖方向
OffsetFlag	参见 8.14 开平标志	开平标志
HedgeFlag	参见 8.11 投机套保标志	投机套保标志
OrderPriceType	参见 8.13 报单价格条件	报单价格条件
VolumeTotalOriginal		数量
LimitPrice		价格

TimeCondition	参见 8.16 有效期类型	有效期类型
VolumeCondition	参见 8.17 成交量类型	成交量类型
ContingentCondition	参见 8.18 触发条件	触发条件
GTDDate		GTD 日期
MinVolume		最小成交量
IsAutoSuspend		自动挂起标志
UserForceClose		用户强评标志
StopPrice		止损价
IsSwapOrder		互换单标志
TraderID		交易所交易员代码
OrderLocalID		本地报单编号
ParticipantID		会员代码
ClientID		客户代码
TradingDay	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	交易日
UserProductInfo		用户端产品信息
BranchID		营业部编号
InstallID		报盘安装编号
IPAddressAsInt		IP 地址
MacAddressAsLong		Mac 地址
OrderSysID		报单编号
OrderStatus	参见 8.4 报单状态	报单状态
OrderType	参见 8.15 报单类型	报单类型
VolumeTraded		今成交数量
VolumeTotal		剩余数量
InsertTime	为 int 类型数据, 例如 9:02:31:698, 则该值为 90231698	委托时间
CancelTime	为 int 类型数据, 例如 9:02:31:698, 则该值为 90231698	撤销时间
SequenceNo		序号
FrontID		前置编号
SessionID		会话编号
ExchangeErrorID		交易所错误代码

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.31 OnRspQryTrade 方法

➤ 功能描述

成交单查询应答。当客户端发出成交单查询指令后，交易系统返回响应时，该方法会被调用。

➤ 函数原型

```
void OnRspQryTrade(CUTTradeField *pTrade, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pTrade: 指向成交信息结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户成交单请求的 ID, 该 ID 由用户在成交单查询时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTTradeField

域描述: 成交信息

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
OrderRef		报单引用
TradeID		成交编号, 关键字
Direction	参见 8.8 买卖方向	买卖方向, 关键字
OrderSysID		报单编号
ClientID		客户代码
OffsetFlag	参见 8.14 开平标志	开平标志
HedgeFlag	参见 8.11 投机套保标志	投机套保标志
Price		价格
Volume		数量
TradeDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	成交日期
TradeTime	为 int 类型数据, 例如 9:02:31:698, 则该值为 90231698	成交时间

TradeType	参见 8.20 成交类型	成交类型
TraderID		交易所交易员代码
OrderLocalID		本地报单编号
SequenceNo		序号
TradingDay	为 int 类型数据，例如 2021 年 6 月 3 日，则该值为 20210603	交易日
InstallID		报盘安装编号
FrontID		前置编号
SessionID		会话编号

结构名称：CUTRspInfoField

结构描述：响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.32 OnRspQryInstrumentCommissionRate 方法

➤ 功能描述

合约手续费率查询应答。当客户端发出合约手续费率查询指令后，交易系统返回响应时，该方法会被调用。

➤ 函数原型

```
void OnRspQryInstrumentCommissionRate(CUTInstrumentCommissionRateField
*pInstrumentCommissionRate, CUTRspInfoField *pRspInfo, int nRequestID, bool
bIsLast) {};
```

➤ 函数参数

pInstrumentCommissionRate：指向合约手续费率结构的地址。

pRspInfo：指向响应信息结构的地址。

nRequestID：返回合约手续费率查询请求的 ID，该 ID 由用户在查询时指定。

bIsLast：指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称：CUTInstrumentCommissionRateField

域描述：合约手续费率

字段代码	取值范围	字段描述
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
InvestorRange	参见 8.2 投资者范围	投资者范围
InvestorID		投资者代码
HedgeFlag	参见 8.11 投机套保标志	投机套保标志
PosiDirection	参见 8.10 持仓多空方向	持仓多空方向
OpenRatioByMoney		开仓手续费率
OpenRatioByVolume		开仓手续费
CloseRatioByMoney		平仓手续费率
CloseRatioByVolume		平仓手续费
CloseTodayRatioByMoney		平今手续费率
CloseTodayRatioByVolume		平今手续费
OpenTaxRatioByMoney		开仓税率按金额
OpenTaxRatioByVolume		开仓税率按手数
CloseTaxRatioByMoney		平仓税率按金额
CloseTaxRatioByVolume		平仓税率按手数
CommByOrder		手续费按笔数
MinCommByOrder		每笔报单最低手续费(不包含 CommByOrder)

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.33 OnRspQryInvestor 方法

➤ 功能描述

请求查询投资者响应。交易系统返回响应时，该方法会被调用。

➤ 函数原型

```
void OnRspQryInvestor(CUTInvestorField *pInvestor, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pInvestor: 指向投资者结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回投资者查询请求 ID, 该 ID 由用户在查询时指定。

bIsLast: 指示该次返回是否针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTInvestorField

域描述: 投资者

字段代码	取值范围	字段描述
InvestorID		投资者代码
InvestorGroupID		投资者分组代码
InvestorName		投资者名称
IdentifiedCardType	参见 8.1 证件类型	证件类型
IdentifiedCardNo		证件号码
IsActive		是否活跃
Telephone		联系电话
Address		通讯地址
OpenDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	开户日期
Mobile		手机
CommModelID		手续费率模板代码
MarginModelID		保证金率模板代码
DepartmentID		组织架构代码
AuthGroupID		投资者终端认证分组代码
TradingRightGroupID		投资者合约交易权限分组代码
InvestorInstallID		投资者安装编号

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.34 OnRspQryTransfer 方法

➤ 功能描述

请求查询转账响应。当客户端发出请求查询转账指令后, 交易系统返回响应时, 该方法会被调用。

➤ 函数原型

```
void OnRspQryTransfer(CUTTransferField *pTransfer, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast);
```

➤ 函数参数

pTransfer: 指向转账结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回转账查询请求的 ID, 该 ID 由用户在查询转账时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTTransferField

域描述: 转账

字段代码	取值范围	字段描述
InvestorID		投资者代码
TransferRef		转账引用
TransferType	参见 8.22 银证转账类型	转账类型
Deposit		金额
CurrencyID	目前仅支持人民币, 该值为 “CNY”	币种代码
AccType	参见 8.27 账户类型	账户类型
BankID		银行代码
FundPassword		资金密码
BankPassword		银行密码
FrontID		前置编号
SessionID		会话编号
TransferLocalID		本地转账号, 关键字
TransferSysID		转账系统号
TransferStatus	参见 8.23 转账状态	转账状态
ExchangeErrorID		交易所错误代码
InsertDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	转账日期
InsertTime	为 int 类型数据, 例如 9:02:31:698, 则该值为 90231698	转账时间

结构名称: CUTRspInfoField

结构描述：响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.35 OnRspQryTradingCode 方法

➤ 功能描述

请求查询交易编码响应。当客户端发出查询交易编码请求，后台返回响应时，该方法会被调用。

➤ 函数原型

```
void OnRspQryTradingCode(CUTTradingCodeField *pTradingCode,
CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pTradingCode：指向交易编码结构的地址。

pRspInfo：指向响应信息结构的地址。

nRequestID：返回查询交易编码请求的 ID，该 ID 由用户在查询交易编码时指定。

bIsLast：指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称：CUTTradingCodeField

域描述：交易编码

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
ClientID		客户代码
IsActive		是否活跃
ClientIDType	参见 8.12 交易编码类型	交易编码类型
BranchID		营业部编号
PreBranchID		营业部前缀编号
DepartmentBranchID		所属营业部

结构名称：CUTRspInfoField

结构描述：响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.36 OnRspQryMaxOrderVolume 方法

➤ 功能描述

请求最大下单量响应。当客户端发出查询最大下单量请求，后台报错时，该方法会被调用。

➤ 函数原型

```
void OnRspQryMaxOrderVolume(CUTMaxOrderVolumeField *pMaxOrderVolume,
    CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pMaxOrderVolume: 指向查询最大下单量结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回查询最大下单量请求的 ID，该 ID 由用户在查询最大下单量时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTMaxOrderVolumeField

域描述: 查询最大下单量

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
HedgeFlag	参见 8.11 投机套保标志	投机套保标志
Direction	参见 8.8 买卖方向	买卖方向
OffsetFlag	参见 8.14 开平标志	开平标志
Price		报单价格
MaxVolume		最大允许报单数量
MaxVolumeOri		原始最大允许报单数量

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.37 OnRspQryCreditInstrument 方法

➤ 功能描述

请求查询信用合约属性响应。当客户端发出查询信用合约属性请求，后台返回响应时，该方法会被调用。

➤ 函数原型

```
void OnRspQryCreditInstrument(CUTCreditInstrumentField *pCreditInstrument, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pCreditInstrument: 指向信用合约属性结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回查询信用合约请求的 ID，该 ID 由用户在查询信用合约时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTCreditInstrumentField

域描述: 信用合约属性

字段代码	取值范围	字段描述
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
ConvertRate		股票折算率
IsCreditFund		是否可融资
IsCreditStock		是否可融券
IsGuarantee		是否担保品
IsLiquid		是否高流通证券
ConcentrationGroupID		集中度分组代码
AssetPrice		公允价格
AssetPriceCreditStock		融券公允价格

结构名称: CUTRspInfoField

结构描述：响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.38 OnRspQryCreditInvestor 方法

➤ 功能描述

请求查询投资者信用信息响应。当客户端发出查询投资者信用信息请求，后台返回响应时，该方法会被调用。

➤ 函数原型

```
void OnRspQryCreditInvestor(CUTCreditInvestorField *pCreditInvestor,
CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pCreditInvestor：指向投资者信用信息结构的地址。

pRspInfo：指向响应信息结构的地址。

nRequestID：返回查询投资者信用信息请求的 ID，该 ID 由用户在查询投资者信用信息时指定。

bIsLast：指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称：CUTCreditInvestorField

域描述：投资者信用信息

字段代码	取值范围	字段描述
InvestorID		投资者代码
FundRate		融资保证金率
StockRate		融券保证金率
GuaranteeRate		维持担保率
TotalLimit		融资融券总额度
StockLimit		融券额度
FundLimit		融资额度
PrivateCreditFund		私有融资额度

结构名称：CUTRspInfoField

结构描述：响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.39 OnRspQryPrivateCreditStock 方法

➤ 功能描述

请求查询投资者可融券响应。当客户端发出查询投资者可融券请求，后台返回响应时，该方法会被调用。

➤ 函数原型

```
void OnRspQryPrivateCreditStock(CUTPrivateCreditStockField *pPrivateCreditStock, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pPrivateCreditStock：指向私有融券额度结构的地址。

pRspInfo：指向响应信息结构的地址。

nRequestID：返回查询私有融券额度请求的 ID，该 ID 由用户在查询私有融券额度时指定。

bIsLast：指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称：CUTPrivateCreditStockField

域描述：私有融券额度

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
Volume		可融券数量
BeginDate	为 int 类型数据，例如 2021 年 6 月 3 日，则该值为 20210603	开始日期
EndDate	为 int 类型数据，例如 2021 年 6 月 3 日，则该值为 20210603	结束日期
ReserveVolume		可融券数量(含到期)

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.40 OnRspQryCreditConcentration 方法

➤ 功能描述

请求查询集中度参数响应。当客户端发出查询集中度参数请求，后台返回响应时，该方法会被调用。

➤ 函数原型

```
void OnRspQryCreditConcentration(CUTCreditConcentrationField *pCreditConcentration, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pCreditConcentration: 指向集中度参数结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回查询集中度参数请求的 ID，该 ID 由用户在查询集中度参数时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTCreditConcentrationField

域描述: 集中度参数

字段代码	取值范围	字段描述
InvestorID		投资者代码
ConcentrationBizType	参见 8.31 集中度业务类型	集中度业务类型
ConcentrationInstrType	参见 8.29 集中度合约类型	集中度合约类型
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
GuaranteeRateLower		维持担保率下限
GuaranteeRateUpper		维持担保率上限
ConcentrationRate		集中度

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.41 OnRspQryCreditFundDetail 方法

➤ 功能描述

请求查询投资者融资明细响应。当客户端发出查询投资者融资明细请求, 后台返回响应时, 该方法会被调用。

➤ 函数原型

```
void OnRspQryCreditFundDetail (CUTCreditFundDetailField *pCreditFundDetail, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pCreditFundDetail: 指向投资者融资明细结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回查询投资者融资明细请求的 ID, 该 ID 由用户在查询投资者融资明细时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTCreditFundDetailField

域描述: 投资者融资明细

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
OpenDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	融资日期
TradeID		流水号
ExpireDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	到期日

Amount		融资金额
AmountPayback		已归还融资金额
Volume		融资买入数量
VolumePayback		已归还融资数量
YdAmountPayback		昨归还融资金额
Commission		今日手续费(不可还)
CreditAmountPayback		今日卖券还款金额
FundAmountPayback		今日直接还款金额

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.42 OnRspQryCreditStockDetail 方法

➤ 功能描述

请求查询投资者融券明细响应。当客户端发出查询投资者融券明细请求，后台返回响应时，该方法会被调用。

➤ 函数原型

```
void OnRspQryCreditStockDetail(CUTCreditStockDetailField *pCreditStockDetail, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pCreditStockDetail: 指向投资者融券明细结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回查询投资者融券明细请求的 ID, 该 ID 由用户在查询投资者融券明细时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTCreditStockDetailField

域描述: 投资者融券明细

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
OpenDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	融券日期
TradeID		流水号
ExpireDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	到期日
Volume		融券数量
VolumePayback		已归还融券数量
Amount		融券卖出金额
AmountPayback		已归还融券金额

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.43 OnRspQryFundPayback 方法

➤ 功能描述

请求查询投资者直接还款响应。当客户端发出查询投资者直接还款请求, 后台返回响应时, 该方法会被调用。

➤ 函数原型

```
void OnRspQryFundPayback(CUTFundPaybackField *pFundPayback,
    CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pFundPayback: 指向直接还款结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户直接还款请求的 ID, 该 ID 由用户在直接还款时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTInputFundPaybackField

域描述: 直接还款

字段代码	取值范围	字段描述
InvestorID		投资者代码
CurrencyID	目前仅支持人民币, 该值为“CNY”	币种代码
FundPaybackRef		直接还款引用
Amount		还款金额
FrontID		前置编号
SessionID		会话编号
FundPaybackLocalID		本地转账号
InsertDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	直接还款日期
InsertTime	为 int 类型数据, 例如 9:02:31:698, 则该值为 90231698	直接还款时间

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.44 OnRspQryStockPayback 方法

➤ 功能描述

请求查询投资者直接还券响应。当客户端发出查询投资者直接还券请求, 后台返回响应时, 该方法会被调用。

➤ 函数原型

```
void OnRspQryStockPayback(CUTStockPaybackField *pStockPayback,
CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pStockPayback: 指向直接还券结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户直接还券请求的 ID, 该 ID 由用户在直接还券时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTStockPaybackField

域描述: 直接还券

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
StockPaybackRef		直接还券引用
Volume		直接还券数量
FrontID		前置编号
SessionID		会话编号
TraderID		交易所交易员代码
StockPaybackLocalID		本地直接还券号
ClientID		客户代码
InstallID		报盘安装编号
InsertDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	直接还券日期
InsertTime	为 int 类型数据, 例如 9:02:31:698, 则该值为 90231698	直接还券时间
StockPaybackSysID		直接还券系统号
StockPaybackStatus	参见 8.3 报单操作状态	直接还券系统号
ExchangeErrorID		交易所错误代码
BranchID		营业部编号
IPAddressAsInt		IP 地址
MacAddressAsLong		Mac 地址

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.45 OnRspQryPublicCreditFund 方法

➤ 功能描述

请求查询公有融资额度响应。当客户端发出查询公有融资额度请求, 后台返回响应时, 该方法会被调用。

➤ 函数原型

```
void OnRspQryPublicCreditFund(CUTPublicCreditFundField *pPublicCreditFund, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pPublicCreditFund: 指向公有融资额度结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户公有融资额度请求的 ID, 该 ID 由用户在公有融资额度时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTInputFundPaybackField

域描述: 直接还款

字段代码	取值范围	字段描述
TotalAmount		公有融资额度
FrozenAmount		已占用额度

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.46 OnRspQryETFInfo 方法

➤ 功能描述

请求查询 ETF 信息响应。当客户端发出查询 ETF 信息请求, 后台返回响应时, 该方法会被调用。

➤ 函数原型

```
void OnRspQryETFInfo(CUTETFInfoField *pETFInfo, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pETFInfo: ETF 申赎信息结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户查询 ETF 信息请求的 ID, 该 ID 由用户在查询 ETF 信息时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTETFInfoField

域描述: ETF 申赎信息

字段代码	取值范围	字段描述
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
RedemptionID		一级市场申赎代码
CreationRedemptionUnit		每个篮子对应的 ETF 份数, 申赎必须整数倍
MaxCashRatio		最大现金替代比例
PublishIOPV		是否需要发布 IOPV
Creation		是否可申购
Redemption		是否可赎回
RecordNum		成分证券数量
EstimateCashComponent		T 日每个篮子的预估现金差额
NAVperCU		前一日最小申赎单位净值
NAV		前一日基金份额净值
SubstituteAmount		必须现金替代金额
RedemptionStockVolume		赎回成分股股数

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.47 OnRspQryETFComponent 方法

➤ 功能描述

请求查询 ETF 成分股响应。当客户端发出查询 ETF 成分股请求, 后台返回响应时, 该方法会被调用。

➤ 函数原型

```
void OnRspQryETFComponent(CUTETFComponentField *pETFComponent,
    CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pETFComponent: ETF 成分股结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户查询 ETF 成分股请求的 ID, 该 ID 由用户在查询 ETF 成分股时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTETFComponentField

域描述: ETF 成分股

字段代码	取值范围	字段描述
ExchangeID	参见 8.28 交易所代码	交易所代码
ETFID		ETF 代码
InstrumentID		成分股合约代码
ComponentExchangeID		成分股交易所代码
Volume		证券数量
SubstituteFlag	参见 8.30 替代标志	替代标志
CreationPremiumRate		申购溢价比例
RedemptionDiscountRate		赎回折价比例
CreationCashSubstitute		申购替代金额
RedemptionCashSubstitute		赎回替代金额

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.48 OnRspQryCreditAvailableDetail 方法

➤ 功能描述

请求查询保证金可用余额明细响应。当查询保证金可用余额明细请求,后台返回响应时,该方法会被调用。

➤ 函数原型

```
void OnRspQryCreditAvailableDetail(CUTCreditAvailableDetailField *pCreditAvailableDetail, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pCreditAvailableDetail: 保证金可用余额明细结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户查询保证金可用余额明细请求的 ID, 该 ID 由查询用户在保证金可用余额明细时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTCreditAvailableDetailField

域描述: 保证金可用余额明细

字段代码	取值范围	字段描述
AccountID		投资者资金账号
CreditCollateralValue		担保品折算
CreditFundProfit		融资盈亏
CreditStockProfit		融券盈亏

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.49 OnRspQryLock 方法

➤ 功能描述

请求查询锁定响应。当查询锁定请求,后台返回响应时,该方法会被调用。

➤ 函数原型

```
void OnRspQryLock(CUTLockField *pLock, CUTRspInfoField *pRspInfo, int
nRequestID, bool bIsLast) {};
```

➤ 函数参数

pLock: 指向锁定结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户查询锁定请求的 ID, 该 ID 由用户在查询锁定时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTLockField

域描述: 锁定

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
LockRef		锁定引用, 关键字
Volume		数量
LockType	参见 8.33 锁定类型	锁定类型
TraderID		交易所交易员代码
LockLocalID		本地锁定编号
ClientID		客户代码
TradingDay	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	交易日
BranchID		营业部编号
InstallID		报盘安装编号
IPAddressAsInt		IP 地址
MacAddressAsLong		Mac 地址
LockSysID		锁定编号
LockStatus	参见 8.3 报单操作状态	锁定状态
InsertDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	报单日期
InsertTime	为 int 类型数据, 例如 9:02:31:698, 则该值为 90231698	委托时间

CancelTime	为 int 类型数据，例如 9:02:31:698，则该值为 90231698	撤销时间
SequenceNo		序号
FrontID		前置编号
SessionID		会话编号
ExchangeErrorID		交易所错误代码

结构名称：CUTRspInfoField

结构描述：响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.50 OnRspQryExecOrder 方法

➤ 功能描述

请求查询行权响应。当查询行权请求，后台返回响应时，该方法会被调用。

➤ 函数原型

```
void OnRspQryExecOrder(CUTExecOrderField *pExecOrder, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pExecOrder：指向行权结构的地址。

pRspInfo：指向响应信息结构的地址。

nRequestID：返回用户查询行权请求的 ID，该 ID 由用户在查询行权时指定。

bIsLast：指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称：CUTExecOrderField

域描述：行权

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码

ExecOrderRef		报单引用, 关键字
Volume		数量
TraderID		交易所交易员代码
ExecOrderLocalID		本地报单编号
ClientID		客户代码
TradingDay	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	交易日
BranchID		营业部编号
InstallID		报盘安装编号
IPAddressAsInt		IP 地址
MacAddressAsLong		Mac 地址
ExecOrderSysID		行权编号
ExecResult	参见 8.34 执行结果	执行结果
InsertDate	为 int 类型数据, 例如 2021 年 6 月 3 日, 则该值为 20210603	报单日期
InsertTime	为 int 类型数据, 例如 9:02:31:698, 则该值为 90231698	委托时间
CancelTime	为 int 类型数据, 例如 9:02:31:698, 则该值为 90231698	撤销时间
SequenceNo		序号
FrontID		前置编号
SessionID		会话编号
ExchangeErrorID		交易所错误代码

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.51 OnRspQryLockPosition 方法

➤ 功能描述

请求查询锁定仓位响应。当查询锁定仓位请求, 后台返回响应时, 该方法会被调用。

➤ 函数原型

```
void OnRspQryLockPosition(CUTLockPositionField *pLockPosition,  
CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pLockPosition: 指向锁定的证券持仓结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户查询锁定持仓请求的 ID, 该 ID 由用户在查询锁定持仓时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTLockPositionField

域描述: 锁定的证券持仓

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
ExecOrderRef		报单引用, 关键字
Volume		数量
FrozenVolume	解锁未知单也会冻结该数量	冻结数量

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.52 OnRspQryOptPosiLimit 方法

➤ 功能描述

请求查询期权限仓响应。当查询期权限仓请求, 后台返回响应时, 该方法会被调用。

➤ 函数原型

```
void OnRspQryOptPosiLimit(CUOptPosiLimitField *pOptPosiLimit,  
CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pOptPosiLimit: 指向期权限仓冻结结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户查询期权限仓请求的 ID, 该 ID 由用户在查询期权限仓时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTOptPosiLimitField

域描述: 期权限仓冻结

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		合约代码
TotalVolume		总数量限制
LongVolume		多头数量限制
OpenVolume		当日开仓数量限制
LongOpenVolume		当日多头开仓数量限制
TotalVolumeFrozen		总数量冻结
LongVolumeFrozen		多头数量冻结
OpenVolumeFrozen		当日开仓数量冻结
LongOpenVolumeFrozen		当日多头开仓数量冻结

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.53 OnRspQryOptAmountLimit 方法

➤ 功能描述

请求查询期权限额响应。当查询期权限额请求, 后台返回响应时, 该方法会被调用。

➤ 函数原型

```
void OnRspQryOptAmountLimit(CUOptAmountLimitField *pOptAmountLimit,
    CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pOptAmountLimit: 指向期权金额限制冻结结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户查询期权限额请求的 ID, 该 ID 由用户在查询期权限额时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTOptAmountLimitField

域描述: 期权金额限制冻结

字段代码	取值范围	字段描述
InvestorID		投资者代码
ExchangeID	参见 8.28 交易所代码	交易所代码
LongAmount		多头持仓金额限制
LongAmountFrozen		多头持仓金额冻结

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.2.54 OnRspQryOptionMarginAlgo 方法

➤ 功能描述

请求查询期权保证金算法响应。当查询期权保证金算法请求, 后台返回响应时, 该方法会被调用。

➤ 函数原型

```
void OnRspQryOptionMarginAlgo(CUTOptionMarginAlgoField *pOptionMarginAlgo, CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};
```

➤ 函数参数

pOptionMarginAlgo: 指向期权保证金算法结构的地址。

pRspInfo: 指向响应信息结构的地址。

nRequestID: 返回用户查询期权保证金算法请求的 ID, 该 ID 由用户在查询期权保证金算法时指定。

bIsLast: 指示该次返回是否为针对 nRequestID 的最后一次返回。

➤ 函数结构

域名称: CUTOptionMarginAlgoField

域描述: 期权保证金算法

字段代码	取值范围	字段描述
ExchangeID	参见 8.28 交易所代码	交易所代码
InstrumentID		标的代码
OutRatio		虚值额优惠比率, 股票期权为参数 1
GuaranteeRatio		最低保障系数, 股票期权为参数 2

结构名称: CUTRspInfoField

结构描述: 响应信息

字段代码	取值范围	字段描述
ErrorID		错误代码
ErrorMsg		错误信息

6.3 CUTApi 接口

CUTApi 接口提供给用户的功能包括报单录入、报单撤销、报单查询、成交查询、持仓查询、合约查询等功能。

6.3.1 CreateApi 方法

➤ 功能描述

创建一个 CUTApi 的一个实例。如果同一进程内创建多个 api, 参数 pszFlowPath 必须设置为不同的路径。注: 不能通过 new 来产生。

➤ 函数原型

```
static CUTApi *CreateApi(const char *pszFlowPath = "", int nCPUID = 0);
```

➤ 函数参数

pszFlowPath: 常量字符指针, 用于指定一个文件目录来存贮 API 缓存。默认值代表当前目录。

nCPUID: 绑定的 CPU ID, 默认为不绑定。将参数 nCPUID 设置为需要绑定的 CPU, 可开启

极速模式。

➤ 函数返回

返回一个指向 CUTApi 实例的指针。

6.3.2 GetApiVersion 方法

➤ 功能描述

获取当前 API 版本。

➤ 函数原型

```
static const char *GetApiVersion();
```

6.3.3 Release 方法

➤ 功能描述

释放一个 CUTApi 实例。注：不能使用 delete 方法。

➤ 函数原型

```
void Release() = 0;
```

6.3.4 Init 方法

➤ 功能描述

使客户端开始与交易系统建立连接，连接成功后可以进行登录。

➤ 函数原型

```
void Init() = 0;
```

6.3.5 Join 方法

➤ 功能描述

客户端等待一个接口实例线程的结束。

➤ 函数原型

```
int Join() = 0;
```

6.3.6 RegisterFront 方法

➤ 功能描述

设置交易系统的网络通讯地址。

➤ 函数原型

```
void RegisterFront(char *pszFrontAddress) = 0;
```

➤ 函数参数

pszFrontAddress: 指向后台服务器地址的指针。

服务器地址的格式为: protocol://ipaddress:port。如: “tcp://127.0.0.1:8888”, “tcp” 代表传输协议, “127.0.0.1” 代表服务器地址。 “8888” 代表交易端口号。

6.3.7 RegisterSpi 方法

➤ 功能描述

注册一个派生自 CUTSpi 接口类的实例, 该实例将完成事件处理。

➤ 函数原型

```
void RegisterSpi(CUTSpi *pSpi) = 0;
```

➤ 函数参数

pSpi: 实现了 CUTSpi 接口的实例指针。

6.3.8 SubscribePrivateTopic 方法

➤ 功能描述

订阅私有流, 如报单回报, 成交回报。该函数可在任何时候调用, 若不调用则不会收到私有流的数据。

➤ 函数原型

```
void SubscribePrivateTopic(UT_TE_RESUME_TYPE nResumeType) = 0;
```

➤ 函数参数

nResumeType: 私有流重传方式。

枚举值	定义
UT_TERT_RESTART	从本交易日开始重传
UT_TERT_RESUME	从上次收到的续传
UT_TERT_QUICK	只传送登录后私有流的内容

6.3.9 SubscribePublicTopic 方法

➤ 功能描述

订阅公共流。该方法要在登录方法前调用。若不调用则不会收到公共流的数据。

注：该接口尚未启用

➤ 函数原型

```
void SubscribePublicTopic(UT_TE_RESUME_TYPE nResumeType) = 0;
```

➤ 函数参数

nResumeType：私有流重传方式。

枚举值	定义
UT_TERT_RESTART	从本交易日开始重传
UT_TERT_RESUME	从上次收到的续传
UT_TERT_QUICK	只传送登录后公共流的内容

6.3.10 ReqOrderInsert 方法

➤ 功能描述

客户端发出报单录入请求或批量报单录入请求。录入错误时对应响应 OnRspOrderInsert，正确时对应回报 OnRtnOrder、OnRtnTrade。注：如为批量报单录入请求，对应的响应回报会多个。例如，批量报单录入 5 笔报单，前 2 笔录入正确，且第 2 笔成交，后 3 笔均失败。则 OnRtnOrder 将收到 2 个回报，OnRtnTrade 将收到 1 个回报，OnRspOrderInsert 将收到 3 个响应。

可以录入限价单、市价单等交易所支持的指令，撤单时使用 ReqOrderAction。

➤ 函数原型

```

    ///报单录入

    int ReqOrderInsert(CUTInputOrderField *pInputOrder, int nRequestID) = 0;

    ///或者批量报单录入

    int ReqOrderInsert(CUTInputOrderField *pInputOrder, int nCount, int
    nRequestID) = 0;
  
```

➤ 函数参数

pInputOrder: 指向输入报单结构的地址。

nCount: 批量报单录入的数量。该数量由用户指定。

nRequestID: 用户报单请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTInputOrderField

结构描述: 输入报单

字段代码	请求时填写	取值范围	字段描述
InvestorID	必填		投资者代码
ExchangeID	必填	参见 8.28 交易所代码	交易所代码
InstrumentID	必填		合约代码
OrderRef	必填	从 1 开始, 不能重复, 只能单调递增	报单引用
Direction	必填	参见 8.8 买卖方向	买卖方向
OffsetFlag	必填	参见 8.14 开平标志	开平标志
HedgeFlag	必填	参见 8.11 投机套保标志	投机套保标志
OrderPriceType	必填	参见 8.13 报单价格条件	
VolumeTotalOriginal	必填		数量
LimitPrice	必填	市价单可以填 0	价格
TimeCondition	必填	参见 8.16 有效期类型	有效期类型
VolumeCondition	必填		成交量类型
ContingentCondition	选填	参见 8.18 触发条件	触发条件
GTDDate	选填	该字段尚未启用	GTD 日期
MinVolume	选填		最小成交量

IsAutoSuspend	选填		自动挂起标志
UserForceClose	选填		用户强评标志
StopPrice	选填		止损价
IsSwapOrder	选填		互换单标志
IPAddressAsInt	选填		IP 地址
MacAddressAsLong	选填		Mac 地址

上交所证券支持的报单类型：

报单类型 报单字段	普通限价	五档 FAK	五档剩余转限价	本方最优价	对手方最优价 剩余转限价
OrderPriceType	2	G	G	H	3
TimeCondition	3	1	3	3	3
VolumeCondition	1	1	1	1	1

深交所证券支持的报单类型：

报单类型 报单字段	普通限价	本方最优价	对手方最优价 剩余转限价	市价 FAK	市价 FOK	五档 FAK
OrderPriceType	2	H	3	1	1	G
TimeCondition	3	3	3	1	1	1
VolumeCondition	1	1	1	1	3	1

6.3.11 ReqOrderAction 方法

➤ 功能描述

客户端发出报单操作请求或批量报单操作请求，包括报单的撤销。错误响应 OnRspOrderAction, OnErrRtnOrderAction, 正确响应 OnRtnOrder。注：如为批量报单操作请求，对应的响应回报会多个。

➤ 函数原型

```

///报单操作

int ReqOrderAction(CUTInputOrderActionField *pInputOrderAction, int
nRequestID) = 0;

///或者批量报单操作

int ReqOrderAction(CUTInputOrderActionField *pInputOrderAction, int
nCount, int nRequestID) = 0;
  
```

➤ 函数参数

pOrderAction: 指向报单操作结构的地址。

nCount: 批量报单操作的数量。该数量由用户指定。

nRequestID: 用户报单操作请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTOrderActionField

结构描述: 报单操作

字段代码	请求时填写	取值范围	字段描述
OrderActionRef	选填		报单操作引用
FrontID	必填		前置编号
SessionID	必填		会话编号
OrderRef	必填		报单引用
ActionFlag	必填	参见 8.19 操作标志	操作标志
LimitPrice	选填		价格
VolumeChange	选填		数量变化
IPAddressAsInt	选填		IP 地址
MacAddressAsLong	选填		Mac 地址

6.3.12 ReqLogin 方法

➤ 功能描述

用户发出登陆请求。 响应 OnRspLogin。

➤ 函数原型

```
int ReqLogin(CUTReqLoginField *pReqLogin, int nRequestID) = 0;
```

```
int ReqLogin(CUTReqLoginField *pReqLogin, int nRequestID);
```

➤ 函数参数

pReqLogin: 指向用户登录请求结构的地址。

nRequestID: 用户登录请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTReqLoginField

结构描述: 用户登录请求

字段代码	请求时填写	取值范围	字段描述
UserID	必填		用户代码
Password	必填		密码
UserProductInfo	选填	填写客户端的产品信息，如软件开发商，版本号等。例如：UTTraderV1	用户端产品信息
UserProductPassword	选填		用户端认证码
OneTimePassword	选填		动态密码
LoginRemark	选填	备注能够被交易系统的日志查询到	登录备注

6.3.13 ReqLogout 方法

➤ 功能描述

用户发出登出请求。会收到 OnFrontDisconnected 回调。

➤ 函数原型

```
int ReqLogout(CUTReqLogoutField *pReqLogout, int nRequestID) = 0;
```

➤ 函数参数

pReqLogout: 指向用户登出请求结构的地址。

nRequestID: 用户登出请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTReqLogoutField

结构描述: 用户登出请求

字段代码	请求时填写	取值范围	字段描述
UserID	必填		用户代码

6.3.14 ReqUserPasswordUpdate 方法

➤ 功能描述

用户密码修改请求。 响应 OnRspUserPasswordUpdate 。

➤ 函数原型

```
int ReqUserPasswordUpdate(CUTUserPasswordUpdateField  
*pUserPasswordUpdate, int nRequestID) = 0;
```

➤ 函数参数

pUserPasswordUpdate: 指向用户密码修改结构的地址。

nRequestID: 用户密码修改请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTUserPasswordUpdateField

结构描述: 用户密码修改请求

字段代码	请求时填写	取值范围	字段描述
UserID	必填		用户代码
OldPassword	必填		原密码
NewPassword	必填		新密码

6.3.15 ReqTransferInsert 方法

➤ 功能描述

转账请求。 响应 OnRspTransferInsert。

➤ 函数原型

```
int ReqTransferInsert(CUTInputTransferField *pInputTransfer, int  
nRequestID) = 0;
```

➤ 函数参数

pInputTransfer: 指向转账结构的指针。

nRequestID: 用户转账操作请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTInputTransferField

结构描述: 转账

字段代码	请求时填写	取值范围	字段描述
InvestorID	必填		投资者代码
TransferRef	选填		转账引用
TransferType	必填	参见 8.22 银证转账类型	转账类型
Deposit	必填		金额
CurrencyID	必填	目前仅支持人民币, 请输入“CNY”	币种代码
AccType	必填	参见 8.27 账户类型	账户类型
BankID	选填	如 TransferType 为 0、1, 则必填	银行代码
FundPassword	必填		资金密码
BankPassword	选填	如 TransferType 为 0、1, 则必填	银行密码

6.3.16 ReqFundPaybackInsert 方法

➤ 功能描述

直接还款请求。录入错误时对应响应 OnRspFundPaybackInsert, 正确时对应回报 OnRtnFundPayback。

➤ 函数原型

```
int ReqTransferInsert (CUTInputTransferField *pInputTransfer, int nRequestID) = 0;
```

➤ 函数参数

pInputFundPayback: 指向直接还款结构的指针。

nRequestID: 直接还款请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTInputFundPaybackField

结构描述: 直接还款

字段代码	请求时填写	取值范围	字段描述
InvestorID	必填		投资者代码
CurrencyID	必填	目前仅支持人民币, 请输入“CNY”	币种代码
FundPaybackRef	选填		直接还款引用
Amount	必填		还款金额

6.3.17 ReqStockPaybackInsert 方法

➤ 功能描述

直接还券请求。 录入错误时对应响应 OnRspStockPaybackInsert, 正确时对应回报 OnRtnStockPayback。

➤ 函数原型

```
int ReqStockPaybackInsert (CUTInputStockPaybackField
*pInputStockPayback, int nRequestID) = 0;
```

➤ 函数参数

pInputStockPayback: 指向直接还券结构的地址。

nRequestID: 直接还券请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTInputStockPaybackField

结构描述: 直接还券

字段代码	请求时填写	取值范围	字段描述
InvestorID	必填		投资者代码
ExchangeID	必填	参见 8.28 交易所代码	交易所代码
InstrumentID	必填		合约代码
StockPaybackRef	选填		直接还券引用
Volume	必填		直接还券数量
IPAddressAsInt	选填		IP 地址

MacAddressAsLong	选填		Mac 地址
------------------	----	--	--------

6.3.18 ReqLockInsert 方法

➤ 功能描述

锁定插入请求。客户端发出锁定插入请求。录入错误时对应响应 OnRspLockInsert，正确时对应回报 OnRtnLock。

➤ 函数原型

```
int ReqLockInsert(CUTInputLockField *pInputLock, int nRequestID) = 0;
```

➤ 函数参数

pInputLock：指向输入锁定结构的地址。

nRequestID：输入锁定请求的 ID，该 ID 由用户指定，管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称：CUTInputLockField

结构描述：输入锁定

字段代码	请求时填写	取值范围	字段描述
InvestorID	必填		投资者代码
ExchangeID	必填	参见 8.28 交易所代码	交易所代码
InstrumentID	必填		合约代码
LockRef	选填		锁定引用
Volume	必填		锁定数量
LockType	必填	参见 8.33 锁定类型	锁定类型
IPAddressAsInt	选填		IP 地址
MacAddressAsLong	选填		Mac 地址

6.3.19 ReqExecOrderInsert 方法

➤ 功能描述

行权插入请求。客户端发出行权插入请求。录入错误时对应响应 OnRspExecOrderInsert，正确时对应回报 OnRtnExecOrder。

➤ 函数原型

```
int ReqExecOrderInsert (CUTInputExecOrderField *pInputExecOrder, int  
nRequestID) = 0;
```

➤ 函数参数

pInputExecOrder: 指向输入行权结构的地址。

nRequestID: 输入行权请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTInputExecOrderField

结构描述: 输入行权

字段代码	请求时填写	取值范围	字段描述
InvestorID	必填		投资者代码
ExchangeID	必填	参见 8.28 交易所代码	交易所代码
InstrumentID	必填		合约代码
ExecOrderRef	选填		报单引用
Volume	必填		数量
IPAddressAsInt	选填		IP 地址
MacAddressAsLong	选填		Mac 地址

6.3.20 ReqExecOrderAction 方法

➤ 功能描述

行权操作请求。客户端发出行权操作请求。录入错误时对应响应 OnRspExecOrderAction, 正确时对应回报 OnErrRtnExecOrderAction。

➤ 函数原型

```
int ReqExecOrderAction (CUTInputExecOrderActionField  
*pInputExecOrderAction, int nRequestID) = 0;
```

➤ 函数参数

pInputExecOrderAction: 指向输入行权操作结构的地址。

nRequestID: 输入行权操作请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTInputExecOrderActionField

结构描述: 输入行权操作

字段代码	请求时填写	取值范围	字段描述
ExecOrderActionRef	选填		行权操作引用
FrontID	必填		前置编号
SessionID	必填		会话编号
ExecOrderRef	必填		行权引用
IPAddressAsInt	选填		IP 地址
MacAddressAsLong	选填		Mac 地址

6.3.21 ReqQryInstrument 方法

➤ 功能描述

合约查询请求。响应 OnRspQryInstrument。

➤ 函数原型

```
int ReqQryInstrument(CUTQryInstrumentField *pQryInstrument, int nRequestID) = 0;
```

➤ 函数参数

pQryInstrument: 指向查询合约结构的地址。

nRequestID: 合约查询请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryInstrumentField

结构描述: 查询合约

字段代码	请求时填写	取值范围	字段描述
ExchangeID	选填	参见 8.28 交易所代码	交易所代码

InstrumentID	选填		合约代码
--------------	----	--	------

6.3.22 ReqQryDepthMarketData 方法

➤ 功能描述

行情查询请求。响应 OnRspQryDepthMarketData。

➤ 函数原型

```
int ReqQryDepthMarketData(CUTQryDepthMarketDataField *pQryDepthMarketData, int nRequestID) = 0;
```

➤ 函数参数

pQryDepthMarketData: 指向查询行情结构的地址。

nRequestID: 用户查询行情请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryDepthMarketDataField

结构描述: 查询行情

字段代码	请求时填写	取值范围	字段描述
ExchangeID	选填	参见 8.28 交易所代码	交易所代码
InstrumentID	选填		合约代码

6.3.23 ReqQryInvestorPosition 方法

➤ 功能描述

投资者持仓查询请求。响应 OnRspQryInvestorPosition。

➤ 函数原型

```
int ReqQryInvestorPosition(CUTQryInvestorPositionField *pQryInvestorPosition, int nRequestID) = 0;
```

➤ 函数参数

pQryInvestorPosition: 指向持仓查询结构的地址。

nRequestID: 用户持仓查询请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryInvestorPositionField

结构描述: 持仓查询

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码
ExchangeID	选填	参见 8.28 交易所代码	交易所代码
InstrumentID	选填		合约代码

6.3.24 ReqQryTradingAccount 方法

➤ 功能描述

资金账户查询请求。响应 OnRspTradingAccount。

➤ 函数原型

```
int ReqQryTradingAccount(CUTQryTradingAccountField *pQryTradingAccount,  
int nRequestID) = 0;
```

➤ 函数参数

pQryTradingAccount: 指向查询资金账户结构的地址。

nRequestID: 用户资金账户查询请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryTradingAccountField

结构描述: 查询资金账户

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码
CurrencyID	选填	目前仅支持人民币, 请输入“CNY”	币种代码

AccType	选填	参见 8.27 账户类型	合约代码
---------	----	--------------	------

6.3.25 ReqQryOrder 方法

➤ 功能描述

报单查询请求。响应 OnRspQryOrder。

➤ 函数原型

```
int ReqQryOrder(CUTQryOrderField *pQryOrder, int nRequestID) = 0;
```

➤ 函数参数

pQryOrder: 指向报单查询结构的地址。

nRequestID: 用户报单查询请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryOrderField

结构描述: 报单查询

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码
ExchangeID	选填	参见 8.28 交易所代码	交易所代码
InstrumentID	选填		合约代码
SequenceNo	选填		起始编号
Limit	选填		从 SequenceNo 开始, 查询 Limit 条记录
OrderStatus	选填		报单状态

6.3.26 ReqQryTrade 方法

➤ 功能描述

成交单查询请求。响应 OnRspQryTrade。

➤ 函数原型

```
int ReqQryTrade(CUTQryTradeField *pQryTrade, int nRequestID) = 0;
```

➤ 函数参数

pQryTrade: 指向成交查询结构的地址。

nRequestID: 用户成交单查询请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryTradeField

结构描述: 成交查询

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码
ExchangeID	选填	参见 8.28 交易所代码	交易所代码
InstrumentID	选填		合约代码
SequenceNo	选填		起始编号
Limit	选填		从 SequenceNo 开始, 查询 Limit 条记录

6.3.27 ReqQryOptionInstrMarginByVolume 方法

➤ 功能描述

期权合约每手保证金查询请求。响应 OnRspQryOptionInstrMarginByVolume。

➤ 函数原型

```
int
ReqQryOptionInstrMarginByVolume(CUTQryOptionInstrMarginByVolumeField
*pQryOptionInstrMarginByVolume, int nRequestID) = 0;
```

➤ 函数参数

pQryOptionInstrMarginByVolume: 指向期权合约每手保证金查询结构的地址。

nRequestID: 期权合约每手保证金查询请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryOptionInstrMarginByVolumeField

结构描述：期权合约每手保证金查询

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码
ExchangeID	选填	参见 8.28 交易所代码	交易所代码
InstrumentID	选填		合约代码

6.3.28 ReqQryOptionInstrCommRate 方法

➤ 功能描述

期权合约手续费率查询请求。响应 OnRspQryOptionInstrCommRate。

➤ 函数原型

```
int ReqQryOptionInstrCommRate(CUTQryOptionInstrCommRateField *pQryOptionInstrCommRate, int nRequestID) = 0;
```

➤ 函数参数

pQryOptionInstrCommRate: 指向期权合约手续费率查询结构的地址。

nRequestID: 期权合约手续费率查询请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryOptionInstrCommRateField

结构描述：期权合约手续费率查询

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码
ExchangeID	选填	参见 8.28 交易所代码	交易所代码
InstrumentID	选填		合约代码

6.3.29 ReqQryInstrumentCommissionRate 方法

➤ 功能描述

请求查询手续费率。响应 OnRspQryInstrumentCommissionRate。

➤ 函数原型


```
int ReqQryInstrumentCommissionRate(CUTQryInstrumentCommissionRateField
*pQryInstrumentCommissionRate, int nRequestID) = 0;
```

➤ 函数参数

pQryInstrumentCommissionRate: 指向手续费率查询结构的地址。

nRequestID: 手续费率查询请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryInstrumentCommissionRateField

结构描述: 手续费率查询

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码
ExchangeID	选填	参见 8.28 交易所代码	交易所代码
InstrumentID	选填		合约代码

6.3.30 ReqQryInvestor 方法

➤ 功能描述

投资者查询请求。响应 OnRspQryInvestor。

➤ 函数原型

```
int ReqQryInvestor(CUTQryInvestorField *pQryInvestor, int nRequestID) =
0;
```

➤ 函数参数

pQryInvestor: 指向投资者查询结构的地址。

nRequestID: 投资者查询请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryInvestorField

结构描述：投资者查询

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码

6.3.31 ReqQryTransfer 方法

➤ 功能描述

转账查询请求。响应 OnRspQryTransfer。

➤ 函数原型

```
int ReqQryTransfer(CUTQryTransferField *pQryTransfer, int nRequestID) = 0;
```

➤ 函数参数

pQryTransfer：指向查询转账结构的地址。

nRequestID：查询转账请求的 ID，该 ID 由用户指定，管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称：CUTQryTransferField

结构描述：转账查询

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码

6.3.32 ReqQryTradingCode 方法

➤ 功能描述

交易编码查询请求。响应 OnRspQryTradingCode。

➤ 函数原型

```
int ReqQryTradingCode(CUTQryTradingCodeField *pQryTradingCode, int nRequestID) = 0;
```

➤ 函数参数

pQryTradingCode: 指向查询交易编码结构的地址。

nRequestID: 查询交易编码请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryTradingCodeField

结构描述: 查询交易编码

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码

6.3.33 ReqQryMaxOrderVolume 方法

➤ 功能描述

最大下单量查询请求。响应 OnRspQryMaxOrderVolume。

➤ 函数原型

```
int ReqQryMaxOrderVolume(CUTQryMaxOrderVolumeField *pQryMaxOrderVolume,
int nRequestID) = 0;
```

➤ 函数参数

pQryMaxOrderVolume: 指向查询最大下单量结构的地址。

nRequestID: 查询最大下单量请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryMaxOrderVolumeField

结构描述: 查询最大下单量

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码
ExchangeID	选填	参见 8.28 交易所代码	交易所代码
InstrumentID	选填		合约代码
HedgeFlag	选填	参见 8.11 投机套保标志	投机套保标志

Direction	选填	参见 8.8 买卖方向	买卖方向
OffsetFlag	选填	参见 8.14 开平标志	开平标志
Price	选填		报单价格

6.3.34 ReqQryCreditInstrument 方法

➤ 功能描述

信用合约属性查询请求。响应 OnRspQryCreditInstrument。

➤ 函数原型

```
int ReqQryCreditInstrument(CUTQryCreditInstrumentField *pQryCreditInstrument, int nRequestID) = 0;
```

➤ 函数参数

pQryCreditInstrument: 指向查询信用合约属性结构的地址。

nRequestID: 查询信用合约属性请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryCreditInstrumentField

结构描述: 查询信用合约属性

字段代码	请求时填写	取值范围	字段描述
ExchangeID	选填	参见 8.28 交易所代码	交易所代码
InstrumentID	选填		合约代码

6.3.35 ReqQryCreditInvestor 方法

➤ 功能描述

投资者信用信息查询请求。响应 OnRspQryCreditInvestor。

➤ 函数原型

```
int ReqQryCreditInvestor(CUTQryCreditInvestorField *pQryCreditInvestor, int nRequestID) = 0;
```

➤ 函数参数

pQryCreditInvestor: 指向查询投资者信用信息结构的地址。

nRequestID: 查询投资者信用信息请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryCreditInvestorField

结构描述: 查询投资者信用信息

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码

6.3.36 ReqQryPrivateCreditStock 方法

➤ 功能描述

查询投资者可融券额度请求。录入错误时对应响应 OnRspQryPrivateCreditStock, 正确时对应回报 OnRtnPrivateCreditStock。

➤ 函数原型

```
int ReqQryPrivateCreditStock(CUTQryPrivateCreditStockField *pQryPrivateCreditStock, int nRequestID) = 0;
```

➤ 函数参数

pQryPrivateCreditStock: 指向投资者可融券额度查询结构的地址。

nRequestID: 投资者可融券额度查询请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryPrivateCreditStockField

结构描述: 投资者可融券额度查询

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码

6.3.37 ReqQryCreditConcentration 方法

➤ 功能描述

集中度参数查询请求。响应 OnRspQryCreditConcentration。

➤ 函数原型

```
int ReqQryCreditConcentration(CUTQryCreditConcentrationField  
*pQryCreditConcentration, int nRequestID) = 0;
```

➤ 函数参数

pQryCreditConcentration: 指向查询集中度参数结构的地址。

nRequestID: 查询集中度参数请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryCreditConcentrationField

结构描述: 查询集中度参数

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码

6.3.38 ReqQryCreditFundDetail 方法

➤ 功能描述

投资者融资明细查询请求。响应 OnRspQryCreditFundDetail。

➤ 函数原型

```
int ReqQryCreditFundDetail(CUTQryCreditFundDetailField  
*pQryCreditFundDetail, int nRequestID) = 0;
```

➤ 函数参数

pQryCreditFundDetail: 指向查询投资者融资明细的地址。

nRequestID: 查询投资者融资明细请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryCreditFundDetailField

结构描述: 查询投资者融资明细

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码

6.3.39 ReqQryCreditStockDetail 方法

➤ 功能描述

投资者融券明细查询请求。响应 OnRspQryCreditStockDetail。

➤ 函数原型

```
int ReqQryCreditStockDetail(CUTQryCreditStockDetailField *pQryCreditStockDetail, int nRequestID) = 0;
```

➤ 函数参数

pQryCreditFundDetail: 指向查询投资者融券明细的地址。

nRequestID: 查询投资者融券明细请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryCreditStockDetailField

结构描述: 查询投资者融券明细

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码

6.3.40 ReqQryFundPayback 方法

➤ 功能描述

投资者直接还款查询请求。响应 OnRspQryFundPayback。

➤ 函数原型

```
int ReqQryFundPayback(CUTQryFundPaybackField *pQryFundPayback, int nRequestID) = 0;
```

➤ 函数参数

pQryFundPayback: 指向查询投资者直接还款的地址。

nRequestID: 查询投资者直接还款请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryFundPaybackField

结构描述: 查询投资者直接还款

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码

6.3.41 ReqQryStockPayback 方法

➤ 功能描述

投资者直接还券明细查询请求。响应 OnRspQryStockPayback。

➤ 函数原型

```
int ReqQryStockPayback(CUTQryStockPaybackField *pQryStockPayback, int nRequestID) = 0;
```

➤ 函数参数

pQryStockPayback: 指向查询投资者直接还券明细的地址。

nRequestID: 查询投资者直接还券请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryStockPaybackField

结构描述: 查询投资者直接还券

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码

6.3.42 ReqQryPublicCreditFund 方法

➤ 功能描述

公有融资额度查询请求。响应 OnRspQryPublicCreditFund。

➤ 函数原型

```
int ReqQryPublicCreditFund(CUTQryPublicCreditFundField *pQryPublicCreditFund, int nRequestID) = 0;
```

➤ 函数参数

pQryPublicCreditFund: 指向查询公有融资额度的地址。

nRequestID: 查询公有融资额度券请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryPublicCreditFundField

结构描述: 查询公有融资额度

字段代码	请求时填写	取值范围	字段描述
Time	选填	时间精确至毫秒。例如 5:16:17:570, 则该值应为 51617570; 例如 23:59:34:001, 则该值应为 235934001	时间

6.3.43 ReqQryETFInfo 方法

➤ 功能描述

ETF 信息查询请求。响应 OnRspQryETFInfo。

➤ 函数原型

```
int ReqQryETFInfo(CUTQryETFInfoField *pQryETFInfo, int nRequestID) = 0;
```

➤ 函数参数

pQryETFInfo: 指向查询 ETF 申赎信息的地址。

nRequestID: 查询 ETF 申赎信息请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryETFInfoField

结构描述: 查询 ETF 申赎信息

字段代码	请求时填写	取值范围	字段描述
ExchangeID	选填	参见 8.28 交易所代码	交易所代码
InstrumentID	选填		合约代码

6.3.44 ReqQryETFComponent 方法

➤ 功能描述

ETF 成分股查询请求。响应 OnRspQryETFComponent。

➤ 函数原型

```
int ReqQryETFComponent(CUTQryETFComponentField *pQryETFComponent, int nRequestID) = 0;
```

➤ 函数参数

pQryETFComponent: 指向查询 ETF 成分股的地址。

nRequestID: 查询 ETF 成分股请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryETFComponentField

结构描述: ETF 成分股查询

字段代码	请求时填写	取值范围	字段描述
ExchangeID	选填	参见 8.28 交易所代码	交易所代码
ETFID	选填		ETF 代码
InstrumentID	选填		合约代码

6.3.45 ReqQryCreditAvailableDetail 方法

➤ 功能描述

查询保证金可用余额明细。响应 OnRspQryCreditAvailableDetail。

➤ 函数原型

```
int ReqQryCreditAvailableDetail(CUTQryCreditAvailableDetailField *pQryCreditAvailableDetail, int nRequestID) = 0;
```

➤ 函数参数

pQryCreditAvailableDetail: 指向查询保证金可用余额明细的地址。

nRequestID: 查询 ETF 成分股请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryCreditAvailableDetailField

结构描述: 查询保证金可用余额明细

字段代码	请求时填写	取值范围	字段描述
AccountID	选填		投资者资金账号

6.3.46 ReqQryLock 方法

➤ 功能描述

查询锁定。响应 OnRspQryLock。

➤ 函数原型

```
int ReqQryLock(CUTQryLockField *pQryLock, int nRequestID) = 0;
```

➤ 函数参数

pQryLock: 指向锁定查询的地址。

nRequestID: 查询锁定请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryLockField

结构描述：锁定查询

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码

6.3.47 ReqQryExecOrder 方法

➤ 功能描述

查询锁定。响应 OnRspQryExecOrder。

➤ 函数原型

```
int ReqQryExecOrder(CUTQryExecOrderField *pQryExecOrder, int nRequestID)
= 0;
```

➤ 函数参数

pQryExecOrder：指向行权查询的地址。

nRequestID：查询行权请求的 ID，该 ID 由用户指定，管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称：CUTQryExecOrderField

结构描述：行权查询

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码
ExchangeID	选填	参见 8.28 交易所代码	交易所代码
InstrumentID	选填		合约代码

6.3.48 ReqQryLockPosition 方法

➤ 功能描述

查询锁定。响应 OnRspQryLockPosition。

➤ 函数原型

```
int ReqQryLockPosition(CUTQryLockPositionField *pQryLockPosition, int
nRequestID) = 0;
```

➤ 函数参数

pQryLockPosition: 指向持仓冻结查询的地址。

nRequestID: 持仓冻结查询请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryLockPositionField

结构描述: 持仓冻结查询

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码

6.3.49 ReqQryOptPosiLimit 方法

➤ 功能描述

查询期权限仓。响应 OnRspQryOptPosiLimit。

➤ 函数原型

```
int ReqQryOptPosiLimit(CUTQryOptPosiLimitField *pQryOptPosiLimit, int nRequestID) = 0;
```

➤ 函数参数

pQryOptPosiLimit: 指向期权限仓冻结查询的地址。

nRequestID: 期权限仓冻结查询请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryOptPosiLimitField

结构描述: 期权限仓冻结查询

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码

6.3.50 ReqQryOptAmountLimit 方法

➤ 功能描述

查询期权限额。响应 OnRspQryOptAmountLimit。

➤ 函数原型

```
int ReqQryOptAmountLimit(CUTQryOptAmountLimitField *pQryOptAmountLimit,  
int nRequestID) = 0;
```

➤ 函数参数

pQryOptAmountLimit: 指向期权限额冻结查询的地址。

nRequestID: 期权限额冻结查询请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryOptAmountLimitField

结构描述: 期权限额冻结查询

字段代码	请求时填写	取值范围	字段描述
InvestorID	选填		投资者代码

6.3.51 ReqQryOptionMarginAlgo 方法

➤ 功能描述

查询期权保证金算法。响应 OnRspQryOptionMarginAlgo。

➤ 函数原型

```
int ReqQryOptionMarginAlgo(CUTQryOptionMarginAlgoField  
*pQryOptionMarginAlgo, int nRequestID) = 0;
```

➤ 函数参数

pQryOptionMarginAlgo: 指向期权保证金算法查询的地址。

nRequestID: 期权保证金算法查询请求的 ID, 该 ID 由用户指定, 管理。

➤ 函数返回

参见 9 响应域的错误码描述。

➤ 函数结构

结构名称: CUTQryOptionMarginAlgoField

结构描述: 期权保证金算法查询

字段代码	请求时填写	取值范围	字段描述
ExchangeID	选填	参见 8.28 交易所代码	交易所代码
InstrumentID	选填		标的代码

7 开发示例:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#ifdef WINDOWS
#include <Windows.h>
#else
#include <unistd.h>
#endif

#include "UTApi.h"

//以下全局变量需要联系经纪公司进行具体配置
//前置地址
char* g_front_address = "tcp://127.0.0.1:8888";
//用户名
char* g_userid = "00001";
//密码
char* g_password = "111111";
//下单合约
char* g_instrumentid = "510050";
//价格
double g_price = 7.0;

#ifdef WINDOWS
#define SLEEP_MS(ms) Sleep(ms)
#else
#define SLEEP_MS(ms) usleep(ms*1000)
#endif
```

```
int g_reqid = 0;
void gen_order(CUTInputOrderField* pInputOrderField)
{
    memset(pInputOrderField, 0, sizeof(CUTInputOrderField));
    strcpy(pInputOrderField->InvestorID, g_userid);
    strcpy(pInputOrderField->InstrumentID, g_instrumentid);
    strcpy(pInputOrderField->ExchangeID, "SSE");
    //OrderRef必须设置, 同一会话内必须递增, 可以不连续
    pInputOrderField->OrderRef=g_reqid++;
    pInputOrderField->OrderPriceType = UT_OPT_LimitPrice;
    pInputOrderField->Direction = UT_D_Buy;
    pInputOrderField->OffsetFlag = UT_OF_Open;
    pInputOrderField->HedgeFlag = UT_HF_Speculation;
    pInputOrderField->LimitPrice = g_price;
    pInputOrderField->VolumeTotalOriginal = 100;
    pInputOrderField->TimeCondition = UT_TC_GFD;
    pInputOrderField->VolumeCondition = UT_VC_AV;
    pInputOrderField->MinVolume = 0;
    pInputOrderField->ContingentCondition = UT_CC_Immediately;
    pInputOrderField->StopPrice = 0;
    pInputOrderField->IsAutoSuspend = 0;
}

void gen_order_action(CUTInputOrderActionField*
pInputOrderActionField, CUTOrderField* pOrder)
{
    memset(pInputOrderActionField, 0,
sizeof(CUTInputOrderActionField));
    pInputOrderActionField->FrontID=pOrder->FrontID;
    pInputOrderActionField->SessionID = pOrder->SessionID;
    pInputOrderActionField->OrderRef=pOrder->OrderRef;

    pInputOrderActionField->OrderActionRef = ++g_reqid;
    pInputOrderActionField->ActionFlag = UT_AF_Delete;
}

class TradeSpi : public CUTSpi
{
public:
    TradeSpi(CUTApi* api)
```



```
{
    m_api = api;
}

~TradeSpi(void)
{
}

public:
    ///当客户端与交易后台建立起通信连接时（还未登录前），该方法被
    调用。
    virtual void OnFrontConnected()
    {
        printf("OnFrontConnected!\n");

        //登录请求
        CUTReqUserLoginField reqLoginField;
        memset(&reqLoginField, 0, sizeof(reqLoginField));
        strcpy(reqLoginField.UserID, g_userid);
        strcpy(reqLoginField.Password, g_password);
        strcpy(reqLoginField.UserProductInfo, "xxx");
        m_api->ReqUserLogin(&reqLoginField, 0);
    }

    ///当客户端与交易后台通信连接断开时，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。
    ///@param nReason 错误原因
    ///      0x1001 网络读失败
    ///      0x1002 网络写失败
    ///      0x2001 接收心跳超时
    ///      0x2002 发送心跳失败
    ///      0x2003 收到错误报文
    virtual void OnFrontDisconnected(int nReason)
    {
        printf("OnFrontDisconnected:%d\n", nReason);
    }

    ///错误应答
    virtual void OnRspError(CUTRspInfoField *pRspInfo, int
nRequestID, bool bIsLast)
    {
```

```
        if (pRspInfo && pRspInfo->ErrorID)
        {
            printf("OnRspError: Error! [%d] [%s]\n",
pRspInfo->ErrorID, pRspInfo->ErrorMsg);
        }
    }

    ///登录请求响应
    virtual void OnRspLogin(CUTRspUserLoginField *pRspUserLogin,
CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast)
    {
        if (pRspInfo==NULL || pRspInfo->ErrorID == 0)
        {
            printf("OnRspLogin: OK!\n");

            // 合约查询
            CUTQryInstrumentField QryInstrument;
            memset(&QryInstrument, 0,
sizeof(CUTQryInstrumentField));
            //strcpy(QryInstrument.InstrumentID, "510050");
            if(0 != m_api->ReqQryInstrument(&QryInstrument,
++g_reqid))
            {
                printf("ReqQryInstrument: Error!\n");
            }
        }
        else
        {
            printf("OnRspLogin: Error! [%d]
[%s]\n", pRspInfo->ErrorID, pRspInfo->ErrorMsg);
        }
    }

    ///请求查询合约响应
    virtual void OnRspQryInstrument(CUTInstrumentField
*pInstrument, CUTRspInfoField *pRspInfo, int nRequestID, bool
bIsLast)
    {
        if (pInstrument != NULL)
        {
```

```
        //printf("Instrument:[%s] [%d] [%s]\n",
pInstrument->InstrumentID, pInstrument->ExchangeID,
pInstrument->InstrumentName);
    }

    if (bIsLast)
    {
        if (pRspInfo && pRspInfo->ErrorID)
        {
            printf("OnRspQryInstrument: Error! [%d] [%s]\n",
pRspInfo->ErrorID, pRspInfo->ErrorMsg);
        }
        else
        {
            printf("OnRspQryInstrument:OK\n");
        }
        //查询合约结束，查询资金;必须等上一个查询结束后，才可进
行下一个查询
        CUTQryTradingAccountField QryTradingAccount;
        memset(&QryTradingAccount, 0,
sizeof(CUTQryTradingAccountField));
        if (0 !=
m_api->ReqQryTradingAccount(&QryTradingAccount, ++g_reqid))
        {
            printf("ReqQryTradingAccount: Error!\n");
        }
    }
}

///请求查询资金响应
virtual void OnRspQryTradingAccount(CUTTradingAccountField
*pTradingAccount, CUTRspInfoField *pRspInfo, int nRequestID, bool
bIsLast)
{
    if (pTradingAccount != NULL)
    {
        printf("TradingAccount:[%s] [%.2f]\n",
pTradingAccount->AccountID, pTradingAccount->Available);
    }
}
```

```
        if (bIsLast)
        {
            if (pRspInfo && pRspInfo->ErrorID)
            {
                printf("OnRspQryTradingAccount: Error! [%d]
[%s]\n", pRspInfo->ErrorID, pRspInfo->ErrorMsg);
            }
            else
            {
                printf("OnRspQryTradingAccount:OK\n");
            }
            //查询资金结束, 查询持仓
            CUTQryInvestorPositionField QryInvestorPosition;
            memset(&QryInvestorPosition, 0,
sizeof(CUTQryInvestorPositionField));
            if (0 !=
m_api->ReqQryInvestorPosition(&QryInvestorPosition, ++g_reqid))
            {
                printf("ReqQryInvestorPosition: Error!\n");
            }
        }
    }

    ///请求查询持仓响应
    virtual void OnRspQryInvestorPosition(CUTInvestorPositionField
*pInvestorPosition, CUTRspInfoField *pRspInfo, int nRequestID, bool
bIsLast)
    {
        if (pInvestorPosition != NULL)
        {
            printf("InvestorPosition:[%s] [%s] [%d]\n",
pInvestorPosition->InvestorID, pInvestorPosition->InstrumentID,
pInvestorPosition->Position);
        }

        if (bIsLast)
        {
            if (pRspInfo && pRspInfo->ErrorID)
            {
                printf("OnRspQryInvestorPosition: Error! [%d]
[%s]\n", pRspInfo->ErrorID, pRspInfo->ErrorMsg);
            }
        }
    }
}
```

```
    }
    else
    {
        printf("OnRspQryInvestorPosition:OK\n");
    }
    //查询持仓结束，查询报单
    CUTQryOrderField QryOrder;
    memset(&QryOrder, 0, sizeof(CUTQryOrderField));
    if (0 != m_api->ReqQryOrder(&QryOrder, ++g_reqid))
    {
        printf("ReqQryOrder: Error!\n");
    }
}

///请求查询报单响应
virtual void OnRspQryOrder(CUTOrderField *pOrder,
CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast)
{
    if (pOrder != NULL)
    {
        printf("Order:[%s] [%s] [%d]\n", pOrder->InvestorID,
pOrder->InstrumentID, pOrder->VolumeTotalOriginal);
    }

    if (bIsLast)
    {
        if (pRspInfo && pRspInfo->ErrorID)
        {
            printf("OnRspQryOrder: Error! [%d] [%s]\n",
pRspInfo->ErrorID, pRspInfo->ErrorMsg);
        }
        else
        {
            printf("OnRspQryOrder: OK\n");
        }
        //查询报单结束，查询成交
        CUTQryTradeField QryTrade;
        memset(&QryTrade, 0, sizeof(CUTQryTradeField));

        if (0 != m_api->ReqQryTrade(&QryTrade, ++g_reqid))
```

```
        {
            printf("ReqQryTrade: Error!\n");
        }
    }
}

///请求查询成交响应
virtual void OnRspQryTrade(CUTTradeField *pTrade,
    CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast)
{
    if (pTrade != NULL)
    {
        printf("Trade:[%s] [%s] [%d]\n", pTrade->InvestorID,
            pTrade->InstrumentID, pTrade->Volume);
    }

    if (bIsLast)
    {
        if (pRspInfo && pRspInfo->ErrorID)
        {
            printf("OnRspQryTrade: Error! [%d] [%s]\n",
                pRspInfo->ErrorID, pRspInfo->ErrorMsg);
        }
        else
        {
            printf("OnRspQryTrade: OK\n");
        }
        ///查询结束, 开始下单
        CUTInputOrderField inputOrderField;
        for (int i = 0; i < 1; i++)
        {
            gen_order(&inputOrderField);
            m_api->ReqOrderInsert(&inputOrderField, g_reqid);
        }
    }
}

///报单录入请求错误时的响应;正确时不会产生该响应, 而是回调
OnRtnOrder
virtual void OnRspOrderInsert(CUTInputOrderField *pInputOrder,
    CUTRspInfoField *pRspInfo, int nRequestID, bool bIsLast)
```

```
{
    printf("OnRspOrderInsert: Error! [%d] [%d] [%s] [%s] [%s]
[%d]\n", nRequestID, pRspInfo->ErrorID, pRspInfo->ErrorMsg,
pInputOrder->InvestorID, pInputOrder->InstrumentID,
pInputOrder->OrderRef);
}

///报单通知
virtual void OnRtnOrder(CUTOrderField *pOrder)
{
    printf("OnRtnOrder: [%s] [%s] [%d]\n", pOrder->InvestorID,
pOrder->InstrumentID, pOrder->OrderRef);
    //发一笔撤单
    if (pOrder->OrderStatus == UT_OST_Unknown)
    {
        CUTInputOrderActionField inputOrderActionField;
        gen_order_action(&inputOrderActionField, pOrder);
        m_api->ReqOrderAction(&inputOrderActionField, g_reqid);
    }
}

///报单操作错误, 被UT打回时的响应;正确时不会产生该响应, 而是回
调OnRtnOrder
virtual void OnRspOrderAction(CUTInputOrderActionField
*pInputOrderAction, CUTRspInfoField *pRspInfo, int nRequestID, bool
bIsLast)
{
    printf("OnRspOrderAction: Error! [%d] [%d] [%s] [%d]\n",
nRequestID, pRspInfo->ErrorID, pRspInfo->ErrorMsg,
pInputOrderAction->OrderRef);
}

///报单操作错误, 被交易所打回时的回报
virtual void OnErrRtnOrderAction(CUTOrderActionField
*pOrderAction)
{
    printf("OnErrRtnOrderAction: [%s] [%s]\n",
pOrderAction->OrderRef, pOrderAction->StatusMsg);
}

///成交通知
```

```
virtual void OnRtnTrade(CUTTradeField *pTrade)
{
    printf("OnRtnTrade:%s,%s\n", pTrade->InvestorID,
pTrade->InstrumentID);
}

private:
    CUTApi* m_api;
};

int main(int argc, char *argv[])
{
    //创建api;将参数nCPUID设置为需要绑定的CPU,可开启极速模式
    //如果同一进程内创建多个api,参数pszFlowPath必须设置为不同的路径
    CUTApi* api = CUTApi::CreateApi();

    //创建并注册spi
    TradeSpi spi(api);
    api->RegisterSpi(&spi);

    // 注册前置地址
    api->RegisterFront(g_front_address);

    //订阅私有流;这个函数也可以在登录成功后的任何地方调用
    api->SubscribePrivateTopic(UT_TERT_QUICK);
    //暂时没有公有流
    //api->SubscribePublicTopic(UT_TERT_QUICK);

    //启动api开始工作
    api->Init();

    //等待api线程的结束
    api->Join();
    //api释放
    //api->Release();

    return 0;
}
```


8 数据字典:

8.1 证件类型

名称	编码	说明
组织机构代码	0	
中国公民身份证	1	
军官证	2	
警官证	3	
士兵证	4	
户口簿	5	
护照	6	
台胞证	7	
回乡证	8	
营业执照号	9	
税务登记号/当地纳税 ID	A	
港澳居民来往内地通行证	B	
台湾居民来往大陆通行证	C	
驾照	D	
当地社保 ID	F	
当地身份证	G	
商业登记证	H	
港澳永久性居民身份证	I	
人行开户许可证	J	
其他证件	x	

8.2 投资者范围

名称	编码	说明
所有	1	
投资者组	2	
单一投资者	3	

8.3 报单操作状态

名称	编码	说明
已经提交	a	
已经接受	b	

已经被拒绝	c	
-------	---	--

8.4 报单状态

名称	编码	说明
全部成交	0	
部分成交还在队列中	1	
部分成交不在队列中	2	
未成交还在队列中	3	
未成交不在队列中	4	
撤单	5	
未知	a	
尚未触发	b	
已触发	c	

8.5 持仓日期类型

名称	编码	说明
使用历史持仓	1	
不使用历史持仓	2	

8.6 产品类型

名称	编码	说明
期货	1	
期货期权	2	
组合	3	
即期	4	
期转现	5	
现货期权	6	
个股期权	7	
证券	8	

8.7 合约生命周期状态

名称	编码	说明
未上市	0	
上市	1	
停牌	2	

到期	3	
----	---	--

8.8 买卖方向

名称	编码	说明
买	0	
卖	1	

8.9 持仓类型

名称	编码	说明
净持仓	1	
综合持仓	2	

8.10 持仓多空方向

名称	编码	说明
净	1	
多头	2	
空头	3	

8.11 投机套保标志

名称	编码	说明
投机	1	
套利	2	
套保	3	
备兑	4	
做市商	5	
申赎	A	
信用账户融资融券	B	
信用账户担保品买卖	C	

8.12 交易编码类型

名称	编码	说明
投机	1	
套利	2	
套保	3	

做市商	5	
信用账户融资融券	B	

8.13 报单价格条件

名称	编码	说明
任意价	1	
限价	2	
最优价	3	
最新价	4	
最新价浮动上浮 1 个 ticks	5	
最新价浮动上浮 2 个 ticks	6	
最新价浮动上浮 3 个 ticks	7	
卖一价	8	
卖一价浮动上浮 1 个 ticks	9	
卖一价浮动上浮 2 个 ticks	A	
卖一价浮动上浮 3 个 ticks	B	
买一价	C	
买一价浮动上浮 1 个 ticks	D	
买一价浮动上浮 2 个 ticks	E	
买一价浮动上浮 3 个 ticks	F	
五档价	G	
本方最优价	H	

8.14 开平标志

名称	编码	说明
开仓	0	
平仓	1	
强平	2	
平今	3	
平昨	4	
强减	5	
本地强平	6	

8.15 报单类型

名称	编码	说明
正常	0	

报价衍生	1	
组合衍生	2	
组合报单	3	
条件单	4	
互换单	5	
质押式回购	6	
新股申购	7	

8.16 有效期类型

名称	编码	说明
立即完成，否则撤销	1	
本节有效	2	
当日有效	3	
指定日期前有效	4	
撤销前有效	5	
集合竞价有效	6	

8.17 成交量类型

名称	编码	说明
任何数量	1	
最小数量	2	
全部数量	3	

8.18 触发条件

名称	编码	说明
立即	1	
止损	2	
止赢	3	
预埋单	4	
最新价大于条件价	5	
最新价大于等于条件价	6	
最新价小于条件价	7	
最新价小于等于条件价	8	
卖一价大于条件价	9	
卖一价大于等于条件价	A	
卖一价小于条件价	B	

卖一价小于等于条件价	C	
买一价大于条件价	D	
买一价大于等于条件价	E	
买一价小于条件价	F	
买一价小于等于条件价	H	

8.19 操作标志

名称	编码	说明
删除	0	支持
修改	3	不支持

8.20 成交类型

名称	编码	说明
组合持仓拆分为单一持仓	#	初始化不应包含该类型的持仓
普通成交	0	
期权执行	1	
OTC 成交	2	
期转现衍生成交	3	
组合衍生成交	4	
ETF 申赎成分股成交	5	
ETF 申赎资金成交	6	

8.21 合约交易状态

名称	编码	说明
开盘前	0	
集合竞价	1	
集合竞价撮合	2	
连续交易	3	
非交易	4	
收盘集合竞价	5	
收盘	6	
停牌	a	
可恢复熔断	b	
不可恢复熔断	c	
波动性中断	d	
盘后交易	e	

8.22 银证转账类型

名称	编码	说明
银行转证券	0	
证券转银行	1	
主席转本席	2	
本席转主席	3	
本席转它席	4	
它席转本席	5	

8.23 转账状态

名称	编码	说明
已提交	a	
已接受	b	
已拒绝	c	
已冲正	d	

8.24 大额单边保证金算法

名称	编码	说明
不使用大额单边保证金算法	0	
使用大额单边保证金算法	1	

8.25 期权类型

名称	编码	说明
看涨	1	
看跌	2	

8.26 组合类型

名称	编码	说明
期货组合	0	
垂直价差 BUL	1	
垂直价差 BER	2	
跨式组合	3	
宽跨式组合	4	
备兑组合	5	

时间价差组合	6	
--------	---	--

8.27 账户类型

名称	编码	说明
个股期权/期货	1	
证券	2	
信用	3	

8.28 交易所代码

名称	编码	说明
上证	1	
深证	2	
上期所	3	
中金所	4	
大商所	5	
郑商所	6	
能源中心	7	
港交所	8	
其他	z	

8.29 集中度合约类型

名称	编码	说明
单一合约集中度	1	
小类集中度	2	
分组(板块)集中度	3	

8.30 替代标志

名称	编码	说明
禁止现金替代	0	
允许现金替代	1	
必须现金替代	2	
非本市场必须现金替代	3	

8.31 集中度业务类型

名称	编码	说明
----	----	----

所有业务	*	
担保品买入	1	
融资买入	2	
担保品划入	3	
仓单展期	4	
融券卖出	5	

8.32 证券子类别

名称	编码	说明
国债	GBF	上海市场
无息国债	GBZ	上海市场
国债分销（仅用于分销阶段）	DST	上海市场
公司债（地方债）分销	DVP	上海市场
企业债券	CBF	上海市场
可转换企业债券	CCF	上海市场
公司债券（或地方债券）	CPF	上海市场
金融机构发行债券	FBF	上海市场
质押式国债回购	CRP	上海市场
质押式企债回购	BRP	上海市场
买断式债券回购	ORP	上海市场
分离式可转债	CBD	上海市场
其它债券	OBD	上海市场
封闭式基金	CEF	上海市场
开放式基金	OEF	上海市场
交易所交易基金（买卖）	EBS	上海市场
跨市场/跨境资金	FBL	上海市场
其它基金	OFN	上海市场
以人民币交易的股票（主板）	ASH	上海市场
以美元交易的股票	BSH	上海市场
以人民币交易的股票（科创板）	KSH	上海市场
其它股票	OEQ	上海市场
集合资产管理计划	AMP	上海市场
国债预发行	WIT	上海市场
LOF 基金	LOF	上海市场
公开发行优先股	OPS	上海市场
非公开发行优先股	PPS	上海市场
报价回购	QRP	上海市场

控制指令（中登身份认证密码服务产品复用 CMD 证券子类别）	CMD	上海市场
融资分级子基金	LOC	上海市场
多空分级子基金	LOD	上海市场
主板 A 股	001	深圳市场
中小板股票	002	深圳市场
创业板股票	003	深圳市场
主板 B 股	004	深圳市场
国债	005	深圳市场
企业债	006	深圳市场
公司债（含分离债）	007	深圳市场
可转债	008	深圳市场
中小企业私募债	009	深圳市场
中小企业可交换私募债	010	深圳市场
证券公司次级债	011	深圳市场
质押式回购	012	深圳市场
资产支持证券	013	深圳市场
本市场股票 ETF	014	深圳市场
跨市场股票 ETF	015	深圳市场
跨境 ETF	016	深圳市场
本市场实物债券 ETF	017	深圳市场
黄金 ETF	019	深圳市场
货币 ETF	020	深圳市场
标准 LOF	023	深圳市场
分级子基金	024	深圳市场
封闭式基金	025	深圳市场
仅申赎基金	026	深圳市场
权证	028	深圳市场
个股期权	029	深圳市场
ETF 期权	030	深圳市场
优先股	033	深圳市场
证券公司短期债	034	深圳市场
可交换公司债	035	深圳市场
存托凭证	036	深圳市场

8.33 锁定类型

名称	编码	说明
锁定	1	

解锁	2	
----	---	--

8.34 执行结果

名称	编码	说明
没有执行	n	
已经取消	c	
执行成功	0	
期权持仓不够	1	
资金不够	2	
会员不存在	3	
客户不存在	4	
合约不存在	6	
没有执行权限	7	
不合理的数量	8	
没有足够的历史成交	9	
未知	a	

9 响应域的错误码描述

错误码	错误码含义
-3	每秒发送请求数超过许可数
-2	未处理请求超过许可数
-1	网络连接失败
0	成功

10 交易所及系统错误码描述

10.1 上海证券交易所错误码

错误码	错误码含义
0	成功
202	无效操作员代码
203	无效证券代码
204	申报时间为该证券的非交易时间
208	无效买卖标志, 非 'B' 或者 'S'

209	交易品种不对
211	此账号不能进行该证券交易
212	价格申报出错
213	证券已被挂起
214	该股票不参与集合竞价
215	无效帐号
216	该帐号未指定在你席位
217	帐号已被挂起
218	无效申报价格
219	申报价不能为零
220	申报价格不符合最小价格步长要求
222	申报价格超出范围
223	申报买入数量必须是 1000 的整数倍
224	无效的申报数量
225	申报数量必须大于 0
226	新股申报数量至少 1000
227	申报数量不符合最小步长要求
228	申报数量超过单笔最大允许申报的上限
231	账号可卖出的余额不足
234	股票持有量超出限量
236	PBU 没有买入权限
237	PBU 没有卖出权限
243	基金帐号\机构帐号不能买卖此证券
244	'S' 帐号不能买入
245	不能撤销指定（有卖空股票，请先补回卖空股票）
246	不能撤销指定（有申报）
247	不能撤销指定（公司卖空）
248	中国证券登记结算公司不允许撤销指定
250	B 股结算会员代码（数据库接口中的 firmid）错，可从中登公司上海分公司查询。
252	无效操作员代码
253	操作员权限不够
257	交易员申报的 reqNum 序号不连续
263	市价订单不参加集合竞价
264	最优五档即时成交剩余撤销市价订单对手方无未成交订单
265	最优五档即时成交剩余转限价市价订单对手方和本方均无未成交订单
266	该券种不允许做市价订单
267	帐户没有权限进行帐户式质押回购的交易及出入库
268	场内报盘对于出入库申报的统一出错代码

269	申报类型错误
270	申报类型错误
271	该帐号已指定在你席位
272	未做指定不能交易
273	E 帐号不能通过场内申报进行交易
274	信用交易的申报帐户必须是 E 帐户
275	普通交易的申报帐户必须不是 E 帐户
276	该证券不允许担保品买入
277	信用卖出权限不足
278	该证券不允许融资买入
279	卖券还款或融资平仓权限不足
280	买券还券或融券平仓权限不足
281	该证券不允许融券卖出
282	融券卖出申报价格低于最新成交价
283	买券还券、融券卖出、融券平仓申报的信用帐户必须配置证券公司融券专用帐户
401	无效操作员代码
403	无效申报序号
404	撤单对应申报序号检验失败
405	撤单对应的证券帐号检验失败
406	撤单对应的股票代码检验失败
407	撤单对应的买卖方向检验失败
408	对应的申报已经被撤单
409	只能通过买 799998 撤销指定
410	只能通过买 799999 重新指定
411	对应的申报已成交，不能撤单
412	不允许撤单
413	撤单失败, 对应的申报已成交或者已经被撤单
499	无申报可撤
507	后台应用错误
508	后台架构错误
731	基金公司未上传 ETF 申购赎回清单
732	ETF 状态不对
733	ETF 投资者持股不足
734	ETF 现金替代比例超过基金公司规定的比例
735	ETF 基金账号持股余额不足，投资者不能赎回
736	ETF 基金账户不能申购或赎回
752	账户中没有现券
753	账户中现券余额不足

754	账户中没有标准券
755	账户中标准券余额不足
756	账户没有质押券
757	账户的质押券余额不足
758	标准券折算率未初始化
759	标准券代码不存在
1005	登录主机失败
1014	PBU 登录失败
1015	修改口令失败
1021	rec_num 必须连续递增
1022	股票代码错误或者非本市场产品
1023	账号错误
1024	买卖方向错误
1025	价格错误
1026	数量错误
1027	日期错误
1028	申报单或撤消单的标志错误
1029	结算会员代码错误
1030	撤消单的 ordrec 域错误
1031	撤消单的 ordrec 域大于等于撤消单的 rec_num 域
1034	撤消的申报单是失败的数据单
1039	刚读入待处理的单子的 status 是' P' 或' p'
1040	刚读入待处理的单子的 status 不是' R' , 也不是' P'
1042	底层通讯接收时发生错
1043	接收到的数据包内容错误
1050	撤消的申报单不存在或者撤单申报的证券代码错误
1100	申报号太大
1101	记录不匹配
1102	营业部代码格式错误, 格式必须为 1 到 65535 的整数
1103	无法解析订单
2000	与交易所的网络连接出错
2001	被撤单的日期不是今天
2004	被撤单的帐号不是本人
2006	撤单记录中被撤单的股票代码与被撤单不一致
2007	撤单记录中被撤单的股票价格不正确, 与被撤单不一致
2008	只有申报才可被撤单
2101	撤单记录和被撤单记录的股票买卖方向不同
2102	撤单记录和被撤单记录的股票数量不同
3000	连接数据库失败

3001	数据库记录的交易日不正确
3002	truncate 数据库记录失败
4001	获取参考数据失败
4003	请检查旧口令是否正确，或者新口令长度是否正确
4004	连接主机失败，其他错误
4005	IP 绑定失败
4006	连接主机超时
4007	连接主机失败，底层 I/O 未知错误
4008	交易员登录失败
4009	获取交易时间表失败
4010	参考数据中 isix 格式错误
4011	参考数据中 rid 格式错误
4012	产品代码 SecurityID 错误或者业务类型 ReqID 错误
4500	股票代码 securityid 字段格式错误
4501	订单状态 ordstatus 非法
4502	TimeStamp 取值不正确，非当前交易日
4601	没有找到对手方指令
4602	没有对手方指令，指令细节不一致
5000	上行消息超过 4K Message Exceed Max Length
5001	上行消息校验和错误 CheckSum Error
5002	心跳超时 Heartbeat Timeout
5003	平台已有 OMS 登录 Already Login, try again
5004	连接建立后 5 秒内未完成登录 Login Timeout
5005	上行消息 TargetCompId 不正确 CompId Error
5006	TDGW 未登录至交易系统，请稍后重试 Not Ready
5007	内部错误 Internal Error
5008	不能识别的消息类型 Message Type Illegal
5009	平台状态暂不接受申报
5010	SetID 错误
5011	Pbu 错误
5012	首个消息非是 Logon 消息 Login First
5013	BeginReportIndex 取值错误
5014	不支持的接口协议版本 UnsupportedPrctlVersion
5015	消息数据错误 Message Data Error
10000	内部错误，请联系上交所
10002	找不到用户
10004	与原有密码不同
10006	请求已经被处理
10008	繁忙, TXN 被拒绝。请再试一次

10018	无效的市场类型
10022	未填写借贷期限
10028	无效的净资产值（NAV）日期
10030	无效的基础产品
10032	未填写基础产品
10038	无效的价格
10042	无效的现金红利股权登记日期
10044	未填写除息日期
10046	无效的除息日期
10048	未填写红利金额
10052	无效的股票红利股权登记日期
10054	未填写除权日期
10056	无效的除权日期
10058	未填写分红率
10062	无效的配股股权登记日期
10066	无效的除权日期
10070	未填写配股价格
10074	未填写股东大会登记日期
10076	无效的股东大会登记日期
10078	未填写股东大会日期
10080	无效的股东大会日期
10084	无效的股票总数量
10086	无效的每股收益（EPS）
10088	无效的期限：借贷天数
10090	国内标识
10092	每股收益（EPS）日期
10096	无效的权证类型
10098	未填写产品简称
10100	无效的产品子类型
10104	无效的产品群名称
10106	无效的产品类型
10108	该国际证券标识码（ISIN）已存在
10112	无效的产品代码
10114	产品删除无效
10116	无效的缺省交易时间表
10118	未找到产品组
10120	产品和产品组不匹配
10126	无效的外部标识
10128	摘牌日无效

10132	产品数据不一致
10140	无效的产品子类型
10142	错误的时间戳
10144	无效的结算货币
10146	无效的名义货币
10150	未找到记录
10156	无效的订单手数类型
10160	未填写持续时间
10162	未填写启动时间
10172	卖空标识无效
10178	无效的最小披露数量
10182	无效的波动范围
10188	无效的最大剩余数量
10192	无效的整手数
10194	无效的最大订单数量 (A)
10200	无效的收盘价计算方法
10202	无效的参考价格
10204	未填写价格上下限
10206	未填写发行数量
10208	总数量无效
10210	借贷期限无效
10218	未填写抽签日期
10220	无效的抽签日期
10222	无效的价格上下限
10224	无效的转换率
10228	无效的最小协商订单数量
10230	无效的产品组代码
10234	产品群未找到
10236	非法时间戳
10238	访问被拒绝
10240	超过最大订单有效期
10242	订单有效期早于当前日期
10244	错误订单有效期
10246	错误产品类型
10248	无法读取该产品
10250	无效交易事务代码
10252	该交易时段内禁用该功能
10254	未分配该产品
10256	未获代理帐户授权

10258	未获做市商帐户授权
10260	未获自有帐户授权
10262	无效订单类型
10264	无效执行限制类型
10266	无效交易限制类型
10268	未获取做市商资格
10270	无效 PBU 的 PBU 代码
10272	当前用户不是交易所用户
10274	交易被拒绝: PBU 不能在该 MISS 上交易!
10276	无效报价价格或数量
10278	无效交易限制 “SU”
10280	不合理数量
10282	最后修改时间已改变
10286	无效处理日期
10290	PBU 无分配产品
10292	无效的日期输入
10294	无效的日期参数
10296	合约 ISIN 前缀重复
10298	无效的产品关联数据标识或交易员不存在
10302	产品或特许权限不关联到当前子组
10304	资源信息找不到
10306	合约产品缩写前缀重复
10308	该产品已分配特许权限
10310	PBU 未分配特许权限
10318	冰山订单最小数量非整手数量
10320	最小披露数量非整手数量
10322	无效的最小报价数量
10324	无效的挂牌首日
10326	无效最小可交易量
10348	无效 IPO 发行方法
10350	无效是非标识
10352	无效冰山订单最小数量
10354	无效持续时间
10356	无效开始时间
10360	无效价格区间计算方法
10362	无效价格单位
10364	无效交易模型
10366	无法将 RAL 更新至数据库
10368	无效值间关系

10376	无效的市场代号
10378	无效的交易所代号
10380	产品重复(代号, 或简称, 或 ISIN)
10382	分红参数未输入
10390	无效的成交金额的计算类型
10392	市场处于停盘状态
10394	无效的时段转换代码
10396	无法转换到目标状态
10398	产品无法使用
10400	当前状态无法转换到等待批处理
10402	未找到该市场
10404	状态转换请求已被触发
10406	交易时间表已经处于开启状态
10408	交易时间表已经处于关闭状态
10410	已超过交易结束时段的开始时间
10412	无效的产品状态
10414	无效的转换时间表
10416	PBU 已经被删除
10420	无效帐户类型
10424	订单数据已变更
10426	该交易时段内无效执行限制
10428	无效交易限制“OA”
10430	无效交易限制“AU”
10432	无效交易限制“FP”
10434	无效买卖标志
10436	帐户类型不可更改
10438	订单类型无效更改
10440	市价转限价订单不可更改订单价格
10442	无效订单数量及最高揭示数量组合
10444	订单数量小于最低订单数量要求
10446	无效交易员组代码
10448	该交易员不是高级交易员
10450	订单价格超出范围
10454	PBU 已经存在
10456	参与者业务单元没有分配给下一级 PBU
10462	无效的总部 PBU
10464	存在重复记录
10468	PBU 已经存在
10470	对于找到的交易所没有默认的记录

10472	不存在相关的 PBU 关系
10476	无效的 PBU 类型
10478	无效的时间标记
10480	不允许修改
10482	跨交易所会员字段为空
10486	无效 PBU 名称
10488	无效的 PBU ID
10490	无效的 PBU 类别
10492	参与者业务单元已分配了下一级 PBU
10498	交易所身份权限不足
10500	HOSTSEC 初始化失败
10502	权限不足
10504	活动状态权限不足
10506	代表身份权限不足
10508	PBU 状态为停止交易或者 PBU 未激活
10510	不同 PBU 的交易要求被拒绝
10512	PBU 交易标识为未激活（暂停状态，权限不足）
10514	非高级的交易员
10516	非交易员活动状态
10518	未授权状态
10520	该执行限制对接受剩余订单无效
10522	产品群已经存在
10524	PBU 不能输入买方订单
10528	无效的市场代码
10532	无效的产品组代码
10534	无效的 PBU ID
10536	无效参与者业务单元群组
10540	参与者业务单元仍然处于激活状态
10544	指定的产品群中还有相关联的产品
10546	PBU 不能输入卖方订单
10548	无效产品组代码
10550	无效产品组类型
10552	产品组已经存在
10554	已经有产品分配给了将被删除的产品组
10558	产品组已经分配了产品
10560	买方订单不能为零手
10562	卖方订单不能为零手
10564	不存在的产品群
10566	交易时段未定义

10568	订单类型和交易模型不匹配
10570	TP_TICSIZ 数据错误
10572	产品未被配置至市场或产品群
10574	部分产品未得到处理
10576	无效的市场状态
10578	不允许进行产品群的状态转换
10590	订单的数量不是整手数
10594	无效的交易时间表
10596	订单价格未遵循最小价格单位规则
10604	被撤申报不存在
10606	投资者帐户类型对报价无效
10608	错误的止损限价
10610	找不到指定订单
10616	无效的交易模型代码
10618	无效的匹配优先级代码
10620	无效的竞价数量分配方案
10622	无效的竞价匹配算法
10626	错误的营业部号码
10630	不允许改变过去的时段或把将来的时段改到过去
10632	当前改变与下次时段改变之间的时差太短
10636	产品必须处于暂停状态
10640	错误的查询日期
10642	错误的查询限价
10644	查询结束
10646	无查询结果
10648	集合竞价剩余量分配方式未知
10660	PBU 无分配产品组
10662	用户群组，产品组写入错误
10664	币别不存在
10666	币别已经存在
10668	产品没有分配给子群组
10670	PBU 没有被分配特许权限
10672	子群组没有被分配任何做市商特许权限
10674	无效的用户 ID
10676	由于用户是 DEFMBR, 系统拒绝访问
10678	用户不存在
10680	用户已经存在
10682	无效的 Agent 标记, 只能是 Y 或 N
10684	无效的 Proprietary 标记, 只能是 Y 或 N

10686	无效的 Senior 标记, 只能是 Y 或 N
10688	无效的 Market Maker 标记, 只能是 Y 或 N
10690	试图成为 Market Maker 类型, 但是目前尚不是 Proprietary 类型
10694	试图删除 PBU 管理者被拒绝
10696	无效的交易员状态
10698	由于用户是 MBRSPV, 系统拒绝访问
10700	订单数量小于最小协商订单交易数量
10702	投资者帐户与 PBU 无指定交易关系
10704	投资者帐户的指定交易关系未被激活
10706	投资者帐户已被暂停
10708	投资者帐户类型对该产品无效
10712	无效投资者帐户
10716	无效的协商订单信息类型
10722	无效的结算日期
10726	无效的协商订单
10730	无效的场外交易产品
10732	用户不存在或者已经被删除
10734	不能计算收盘价
10738	收市时段已经开始
10740	该订单申报的业务已结束
10744	超出当日最高、最低成交价范围
10746	交易币种不匹配
10748	产品不匹配
10750	订单类型不匹配
10752	订单买卖方向不匹配
10754	系统状态编号不存在
10756	无目标地址
10758	订单数量不匹配
10760	订单价格不匹配
10762	该产品无做市商
10764	对手方 PBU 不匹配
10766	协商订单号不匹配
10768	结算代码不匹配
10770	结算日期不匹配
10772	结算币种不匹配
10774	产品被暂停场外交易
10776	产品被暂停协商交易
10780	日期已存在
10782	日期不多于当前日期两天

10784	日期不早于八天前
10786	对指定的产品没有设置许可证信息
10788	消息长度不正确
10790	托管 PBU 不存在
10792	用户上下文内容太长
10794	投资者账户不存在
10796	不允许创建指定交易关系
10798	无效类型
10800	投资者和当前 PBU 已经存在指定交易关系
10802	无效标志位
10804	该帐号已指定在其它席位
10806	不允许撤销指定交易关系
10808	无效状态
10812	不存在指定交易关系
10820	请求被拒绝:基础数据维护阻塞阶段
10822	无效的基础产品（非债券）
10824	无效的基础产品（不在同一市场）
10826	数据被非法修改
10828	无效更新持仓数量
10836	投资者持仓不足
10838	对手方不存在
10840	买方用户不存在
10842	卖方用户不存在
10844	该做市商未授权输入报价请求
10852	无效的交易状态
10868	重复的记录
10884	存在已分配的被托管 PBU
10886	无效的托管 PBU
10888	无效的产品群（不在同一市场）
10894	订单未被冻结
10896	报价未被冻结
10900	已存在冻结报价
10902	不支持的交易类型
10916	无效的场外交易交易日期
10918	被代理者权限不足
10920	品种特许权限已被分配
10924	品种未被分配给 PBU
10926	品种特许权限已被分配给 Subgroup
10928	品种特许权限未被分配给 Subgroup

10930	不正确的市场状态
10932	产品状态资源访问授权不足
10934	被代理者处于非活动状态
10938	无效品种代码
10940	品种已经存在
10942	品种下仍然存在合约
10944	无效的持仓检查
10946	无效的会员清算代码
10948	外部系统 (EAI) 使用的 PBUID 没有配置
10952	指定的 ISIN 代码无效
10956	PBU 已经被暂停
10958	无效合约到期月份
10960	该月期货合约已经生成
10962	无效消息长度
10964	品种不存在
10966	无效最后合约自动生成日
10968	对同一议案重复投票
10970	更新持仓时所需保证金不足
10972	非法的默认缴费包类型
10978	ETF 请求总量超过当日阈值
10980	无效的非交易类型
10982	无效的分配日期
10986	无效的配号规则表
10988	无效的比例配售比例
10990	无效的交割价格类型
10992	无效的承销商 PBU 代码
10994	无效的承销商投资者账户
11002	无效的最后交易日期（与衍生产品冲突）
11004	无效的最后交易日期（与非交易信息冲突）
11006	无效的首次交易日期（与非交易信息冲突）
11008	未填写转换价格
11012	缺少必填的信息
11018	删除承销商 PBU 被拒绝
11024	EzCS 解密 PBU 密码失败
11026	无效的投资类型
11028	无效的产品代码
11032	该产品不适用
11034	请求超出有效的时期
11036	无效的交易员代码

11038	无效的分红选择
11040	无效的份额来源/份额目的
11042	非交易类型与产品不匹配
11044	无效的持仓解冻数量
11046	无效的非交易订单价格
11048	无效的非交易订单金额
11050	投资者持仓状态不存在
11052	无效的 ETF 订单数量
11054	不允许 ETF 申购
11056	不允许 ETF 赎回
11058	ETF 定义未找到
11060	PBU 对该产品没有买权限。
11062	产品买卖标志无效。
11064	PBU 对该产品没有卖权限。
11068	该 PBU 无权限交易此产品
11070	品种代码或到期月份不能被修改
11072	指定交易类型无效
11076	合约简称前缀重复
11078	合约产品前缀重复
11080	投资者对该产品没有买权限。
11082	投资者对该产品没有卖权限。
11084	订单买/卖方向不能修改。
11090	无效的停牌状态更改
11094	不能删除一个席位连通群组的主参与者单元。
11098	无效的最大订单数量
11100	无效的输入开始日期
11102	无效的输入结束日期
11106	无效的上市交易日
11108	无效的转换价格
11110	无效的非交易订单数量
11116	非法更新已冻结的订单
11118	修改撮合优先权无效
11122	撮合优先权和竞价数量分配方案组合无效
11124	该产品不允许输入单边报价
11126	PBU 无此非交易权限
11128	投资者无此非交易权限
11130	非交易服务不可用
11132	不适用于此子类型产品
11134	投资者账号没有被授权进行权证创设和注销

11138	投资者已经是冻结状态
11140	投资者已经是解冻状态
11148	无效的检验日期
11150	无效的转换基金
11152	指定交易取消请求正在处理中
11154	政府债券担保投资者持仓不足
11156	权证担保帐户不可以被激活
11162	无效的转指定交易目标方
11166	投资者已与其他 PBU 存在指定交易关系
11170	无效的跨交易所代号
11172	无效的产品群代码
11174	指定的产品群中还有相关联的品种
11182	指定的产品组中还有相关联的品种
11194	可引起波动性中断的市价或市价转限价订单
11196	无效的审核标志
11200	内存订单簿耗光
11202	错误的请求 PBU
11204	正在计算闭市价格...
11206	无效的担保品标志
11208	无效的融资标志
11210	无效的融券标志
11212	普通帐户不允许信用交易
11214	信用帐户不允许普通交易
11218	不正确的信用标签类型
11226	没有对应的证券公司融券专用帐户
11228	无效的信用标签
11230	投资者非交易持仓不足
11236	PBU 无融资买入权限
11238	PBU 无融券卖出权限
11240	该时段不允许删除订单
11244	订阅一个广播流的数量限制被超出
11246	交易时间未到
11248	基准价格在上市日期前不允许被更新
11250	基准价格在目前的时段不允许被更新
11252	不允许更新有涨跌幅控制产品的价格范围
11254	券商融券专用帐户不可以激活
11258	转换基金代码不适用
11262	PBU 对该产品子类型没有买权限。
11264	PBU 对该产品子类型没有卖权限。

11266	投资者对该产品子类型没有买权限。
11268	投资者对该产品子类型没有卖权限。
11270	订单重传
11272	老回购不允许操作
11276	后台广播重传不可用
11278	无效的广播重传请求序号区间
11280	后台发现无效的广播记录无法重传
11282	后台广播重传出现未知错误
11284	不能在未开始的集合竞价中插入紧急停牌
11288	投资者有未处理的订单
11290	主 PBU 不存在
11292	主 PBU 不是内部 PBU
11296	主 PBU 的主 PBU 不是它自身
11298	试图将现有主 PBU 的主 PBU 修改成其它 PBU
11300	PBU 的 missid 未被分配无法强制注销
11302	PBU 未登录无法强制注销
12000	报表授权关系已存在
12002	报表授权关系不存在
12004	已达到最大授权 PBU 个数
12006	使用的 PBU 无权限上传对应的 ETF 定义文件
12008	无效的 ETF 类型
12010	无效的 ETF 定义文件日期
12012	PBU 无权限下载指定报表
12014	无效的下载文件大小
12016	后台无法提供服务
12018	ETF 定义文件检查错误
12020	报表 PBU 与授权 PBU 相同
12022	当前时段无法下载报表
12024	ETF 产品代码错误
12026	基金公司与文件不匹配
12028	定义文件版本号小于当前版本
12032	最优五档及时成交剩余撤销市价订单对手方无未成交订单
12034	最优五档及时成交剩余转限价市价订单对手方和本方均无未成交订单
12036	限价即时全部成交类型订单可成交对手方不足
12038	市价即时全部成交类型订单可成交对手方不足
13300	繁忙
13302	连通圈已经存在
13304	产品未到上市日
13306	后台弹回请求

13310	交易时段重迭
13312	CS 弹回请求
13314	ETF 申购赎回失败（基金经理持仓不足）
13316	ETF 申购赎回失败（投资者持仓不足）
13318	ETF 申购赎回失败（现金替代比例超出）
13322	后台未配置任何广播流订阅权限
13324	无任何市场操作权限
13326	无效的 IPO 发行数量
13328	该操作员已经在其他地方登录
13330	无效的基本订单类型
13332	无效的订单类型和限价组合
13334	无效的订单类型和最高揭示数量组合
13336	无效的最高揭示数量
13338	无效的订单类型和产品组合
13342	订单数量小于最高揭示数量
13344	订单数量小于冰山数量的最低要求
13346	最高揭示数量小于最小最高揭示数量的最低要求
13348	无效的订单类型和交易限制组合
13350	无效的订单类型和执行限制组合
13354	无效的执行限制类型
13356	无效的交易限制类型
13358	产品和产品群交易模型不匹配
13360	不能在集合竞价时输入分销类型产品的买单
13362	更新时段太迟, 无法更新
13366	尚在发行周期中的 IPO 不能删除
13368	当天新增的 PBU 不允许登录系统
13370	主席位不能为空
13374	无法登陆，交易员未激活
13378	承销商投资者帐户与 PBU 无指定交易关系
13380	承销商账户已被暂停
13382	EZOES 参考数据为空
13384	不允许参加开盘集合竞价
13386	公共存储区域内存槽耗光
13388	公共存储区域更新请求的时间戳已过时。
13390	公共存储区域内存槽中无数据
13392	关键字查询区含有非法数据
13394	用户 IP 不在 IP 白名单内
13396	OEF 权限不能在线修改
13398	禁止限价订单交易

13400	禁止市价订单交易
13402	禁止市价转限价订单交易
13404	禁止冰山订单交易
13406	禁止协商订单交易
13412	配股除权与送股除权不能在一天进行
13416	停牌时间超出允许范围
13418	复牌时间超出允许范围
13422	开盘集合竞价不允许复牌
13424	超过集合竞价最大订单配置量
13426	被撤订单根本没有输入过
13428	被撤订单输入过，但是是废单
13430	被撤订单输入过，但是已经被完全撮合或撤单
13432	被撤订单本身即是撤单请求
13434	此撤单请求所包含的产品代码与订单本身不相符
13436	该证券不允许担保品买入
13438	该证券不允许融资买入
13440	该证券不允许买券还券或融券平仓
13442	该证券不允许融券卖出
13444	融券卖出申报价格低于最新成交价
13446	前台版本号不匹配
13448	该证券不允许进行担保品卖出/卖券还款/平仓卖出
13450	融券卖出不允许申报市价订单
13452	输入的请求订单掩码非法
13454	不允许撤销非交易订单
13456	ETF 申赎上限超出
13458	交易主机繁忙
13460	指令操作来源无效
13462	指令持仓操作类型无效
13464	冻结/解冻操作类型无效
13466	请求类型无效
13468	过户方 PBU 已经被暂停
13470	无效过户方 PBU 的 PBU 代码
13472	过户方投资者帐户与过户方 PBU 无指定交易关系
13474	过户方投资者帐户已被暂停
13476	无效过户方投资者帐户
13478	无效的过户方投资者类型
13480	请求正在处理中
13482	ETF 定义文件上传时间不合法
13484	ETFPUT 业务请求消息类型不合法

13496	买方 PBU 与登陆 PBU 不为同一会员
13498	卖方 PBU 与登陆 PBU 不为同一会员
13516	消息未找到
13518	无效输入参数
13528	证券代码与原申报订单不一致
13532	无 AddIdx 权限, oper:%x, proc:%x
13534	无 Add 权限, oper:%x, proc:%x
13536	无 update 权限, oper:%x, proc:%x
13538	无 read 权限, oper:%x, proc:%x
13540	无 del data 权限, oper:%x, proc:%x
13542	无 del idx 权限, oper:%x, proc:%x
13556	该产品开盘集合竞价禁止撤单
13560	广播服务数据恢复中
13562	PBU 可用登录节点已用完
19030	使用的 PBU 无权限上传或下载对应的 ZRT 文件
13564	产品简称前缀非法
13594	本方最优市价订单申报无本方未成交订单
13596	对手方最优市价订单申报无对手方未成交订单
13598	超过保护价格
13600	无效的保护价格
13602	交易网关协议转换错误: 无效的申报消息长度
13604	交易网关协议转换错误: 无效的业务编号 (BizID)
13606	交易网关协议转换错误: 无效的会员内部订单编号 (clOrdID)
13608	交易网关协议转换错误: 无效的订单买卖方向 (Side)
13610	交易网关协议转换错误: 无效的价格 (Price)
13612	交易网关协议转换错误: 无效的订单数量 (OrderQty)
13614	交易网关协议转换错误: 无效的订单类型 (OrdType)
13616	交易网关协议转换错误: 无效的订单有效时间类型 (TimeInForce)
13618	交易网关协议转换错误: 无效的信用标签 (CreditTag)
13620	交易网关协议转换错误: 无效的营业部代码 (BranchID)
13622	交易网关协议转换错误: 无效的销售人代码 (Custodian)
13624	交易网关协议转换错误: 无效的分红方式 (DividendSelect)
13626	交易网关协议转换错误: 无效的目标证券代码 (DestSecurity)
13628	交易网关协议转换错误: 无效指定交易注册指令 (DesignationInstruction)
13630	交易网关协议转换错误: 无效的指定交易注册类型 (DesignationTransType)
13632	交易网关协议转换错误: 无效的激活码, 包含非法字符 (validationCode)
17038	非法的有效期
17566	会员结算准备金不足以开新仓
17568	计算权利金价格错误

17570	计算期权保证金错误
17572	资金不足以支付权利金
17574	资金不足以支付保证金
17576	期权开平仓类型错误
17582	无效的投资者类型
17584	开平仓方向不一致
17586	无效的结算会员
17588	无效的期权代码
17590	无效的价格量纲
17592	产品超过最后交易日
17594	非法的行权申请日
17598	无效的备兑标签
17602	未实现的功能请求
17604	无效的账户类型
17606	持仓超过账户标的限制
17608	持仓超过会员标的限制
17610	持仓超过会员全市场限制
17612	备兑持仓不足
17614	买入开仓不允许备兑
17616	卖出平仓不允许备兑
17618	认沽期权不允许备兑
17622	锁定现货不足
17624	FOK 订单触发 VOLA
17626	无效的处理类别
17628	开盘集合竞价不允许申报此类型订单
17630	盘中集合竞价不允许申报此类型订单
17632	无效的资金账户类型
17634	设置或查询产品信息失败
17636	持仓超过账户全市场限制
17638	PBU 对该资金账户没有查询权限
17640	该产品已取消限开仓限制
17642	该产品已设置限开仓限制
17644	当前交易日无效
17646	日期小于当前交易日期
17648	VOLA 最后一分钟不允许删除订单
17650	资金额度过大
17652	PBU 无开仓权限
17654	PBU 无平仓权限
17656	无效的投资者账户子编码

17658	合约无买入开仓权限
17660	合约无非备兑卖出开仓权限
17662	合约无备兑卖出开仓权限
17664	合约结算价异常
17668	收盘集合竞价最后一分钟不允许删除订单
17670	收盘集合竞价不允许申报此类型订单
18200	打包消息中消息头的信息错误
18202	打包消息中的指令长度错误
18204	持仓更新指令掩码中的双边信息与输入参数不一致
18206	持仓更新指令的掩码不存在
18208	持仓更新指令的掩码尚未启用
18210	持仓更新指令的掩码不可用于 ETF 申赎
18600	当前时间不在业务设置时间段内
19000	未指定的格式错误
19002	产品代码格式错误
19006	数量格式错误
19008	日期格式错误
19010	金额格式错误
19022	无效的业务类型
19036	无效订单买卖方向
19042	参数值非法
19044	非法分隔符个数
19046	未定义的模板 key 值
19048	非法账户
19050	价格步长非法
19056	业务 PBU 或 REFF 字段于 STEP 消息中内容不符
19058	业务 PBU 与登陆 PBU 不为同一会员
19060	无效的业务号
19088	非法消息类型
19090	产品已被暂停交易
19092	业务已被暂停交易
19096	订单数量和交易金额小于最低要求
19098	订单的数量不是整手数
19102	订单价格超出范围
19104	PBU 已经被暂停
19106	PBU 对该业务没有权限
19110	产品已停牌
19112	业务类型不可用
19114	产品不可用

19118	子类型不可用
19122	订阅关系不存在
19124	TXNCODE 不存在
19128	PBU 不存在
19132	产品索引号不合法
19136	权限标志不合法
19140	无效的 RID
19142	无效的时段
19162	无效的 INV
19164	账户已经被暂停
19166	账户对该业务没有交易权限
19176	该时段交易无效
19180	不是此账号的指定 PBU
19182	无效的费率
19184	申报数量超过上限
19186	无效的订单状态
19188	该产品已停牌
19190	该产品已复牌
19192	会员不存在
19196	会员对该业务没有交易权限
19198	原始订单正在处理当中
19204	该产品已是参考竞价
19210	无效的数据格式
19212	输入的主键与记录中主键不一致
19214	未定义的文件名
19216	删除数据与文件原始记录不同
19218	位图长度与字段个数不符
19220	位图信息错误
19256	产品已被连续停牌
19258	基础产品已停牌
19268	组合策略构成头寸方向不符合要求
19270	组合策略构成合约标的的不符合要求
19272	组合策略构成合约行权价格不符合要求
19274	组合策略构成合约乘数不一致
19276	组合份数与成分合约比例不符或成分合约比例与策略要求比例不符
19278	组合策略拆分数量超过 SM78 记录值
19280	合约不能进行垂直价差策略的组合
19282	认购期权普通卖开转备兑策略不支持拆分
19284	拆分申报策略与组合时不匹配

19286	拆分申报合约与组合时不匹配
19288	组合时组合编号不为空
19290	拆分申报账户与组合时不同
19292	拆分时，组合编号不能为空
19294	该组合持仓不存在
19296	资金不足以拆分
19298	组合份数必须大于 0
19300	组合策略构成合约数量不符合要求
19302	组合策略构成合约到期日不符合要求
19304	组合策略构成合约期权类型不符合要求
19306	组合策略构成合约备兑标签不符合要求
19310	该产品已经独立停牌
19312	该产品停复牌预设时间有误
19314	禁止做市商自成交

10.2 深圳证券交易所错误码

错误码	错误码含义
0	成功
20001	无效账户
20002	无效账户
20003	无效账户
20005	业务禁止
20006	业务禁止
20007	证券停牌
20008	价格错误
20009	价格错误
20010	数量非法
20011	量额非法
20012	业务禁止
20013	业务禁止
20014	业务禁止
20015	业务禁止
20016	价格错误
20017	无效账户
20018	无效账户
20019	无效单元
20020	无效账户
20021	数量超限

20022	股份卖空
20023	无效单元
20024	无效单元
20025	无效单元
20026	无效单元
20027	无效单元
20028	无效单元
20029	业务禁止
20030	约定号重
20031	配对失败
20032	配对失败
20033	无对手方
20034	金额非法
20035	申购不足
20036	赎回不足
20037	现金超限
20038	无效账户
20039	重复申报
20040	无效单元
20041	解押业务中对应的已质押股份不足
20042	解押业务中对应的标准券不足
20046	市价委托不满足成交条件撤单
20047	规模超限
20048	单个账户基金累计申购或赎回规模超限
20049	基金净申购或净赎回总规模超限
20050	单个账户基金净申购或净赎回规模超限
20051	回购天数超过业务限制
20055	会员无对应业务的交易权限
20056	非业务指定交易的证券账户
20059	当日净买入金额超出设定限额
20060	仓位不足
20061	超过投资者交易量限额
20062	超过结算参与者持仓限额
20063	超过投资者持仓限额
20064	结算参与者可用资金不足
20065	平台已关闭
20067	备兑标签错
20068	结算机构代码错误
20069	委托中的账户为结算受限账户

20070	委托中的账户为不合格账户
20071	强平主动撤单的 OwnerType 必须与原委托的一致
20073	双边报价申报的买价必须低于卖价
20074	强平委托必须是限价委托
20075	报价被更新或撤销
20076	申报方式错误
20077	无做市报价权限
20078	超过结算参与人交易量限额
20079	受限股份减持，可用额度不足
20095	主动撤单请求的 applid 或证券 id 与原委托的不一致
20096	主动撤单请求目标委托不能被撤销
20097	主动撤单未找到原始委托或者没有可以被撤销的报价
20099	重单
20101	应用标识错误
20102	证券不支持所申报的业务
20103	解包失败
20104	平台未开放
20105	网关对交易单元无权限
20106	字段取值错误
20107	不支持的消息类型
20108	平台非法
20110	价格类型错误
20111	成交申报中的多方交易单元不属于同一会员
20112	网关无申报委托权限
20113	资金查询服务暂停
20114	临时停市拒绝
20115	申报方式错误
20116	成交请求发起方错误
20117	证券账户无三方回购业务的正/逆回购权限
20118	篮子非法
20119	质押券非法
20120	质押券非法
20121	质押券非法
20122	质押券非法
20123	成交申报中的对手方交易单元不存在
20124	回购期限错误
20125	合约不存在
20126	业务禁止
20127	业务禁止

20128	金额非法
20129	证券非法
20130	无对手方
20131	业务禁止
20132	业务禁止
20133	成份合约已过组合自动解除日
20134	合约不支持申报的策略
20135	委托成份合约个数不符规则
20136	委托成份合约数量不符规则
20137	委托成份合约权利仓/义务仓不符规则
20138	委托成份合约认购/认沽不符规则
20139	委托成份合约行权价不符规则
20140	申报的组合流水号对应组合策略持仓的与人信息与委托中的不一致
20143	无效交易主体
20144	交易商未报备申报的交易员代码
20145	无效账户
20146	订单已超时
20147	询价交易 ID 重复
20149	无效账户
20150	结算方式错误
20151	结算周期错误
20153	借贷期限错
20154	借贷费用场内支付金额错
20155	申报宽限期错
20156	标的券非法
20157	换出质押券不足
20158	债券借贷在途合约不允许申报解除质押
20159	标的券非法
20160	标的券非法
20161	券面总额非法
20162	券面总额非法
20163	借贷期限错
20164	竞买日期错误
20165	重复申报
20166	无法成交
20167	证券临时停牌
20168	竞买不存在
20169	价格错误
20170	竞买预约已生效

20201	议案代码错
20202	表决意见错
29995	市场互联拒绝
29996	市场互联拒绝
29997	市场互联拒绝
29998	市场互联拒绝
29999	其他错误
1	系统服务不可用
2	无此项业务
3	超出业务办理时间
8	XSD 校验异常
10	D-COM 通信系统已关闭/未完成初始化
11	非工作日异常
12	报文 ID 重复异常
13	不允许与报文接收方通信
14	报文发送方无效
15	报文接收方无效
16	该通信系统不允许发送此类报文
20	AppIdr 不存在
21	密码错误
22	D-COM 网关未登录
23	用户重复登录
24	D-COM 网关用户号错误
25	用户技术系统标识错误
26	收到非法报文, D-COM 网关将在 3 秒 后断开当前连接
999	未分类异常

10. 30ST 系统错误码

错误码	错误码含义
NONE	UT: 正确
1	UT: 不在已同步状态
2	UT: 会话信息不一致
3	UT: 不合法的登录
4	UT: 用户不活跃
5	UT: 重复的登录
6	UT: 还没有登录
7	UT: 还没有初始化
8	UT: 前置不活跃

9	UT:无此权限
10	UT:修改别人的口令
11	UT:找不到该用户
12	UT:找不到该经纪公司
13	UT:找不到投资者
14	UT:原口令不匹配
15	UT:报单字段有误
16	UT:找不到合约
17	UT:合约不能交易
18	UT:经纪公司不是交易所的会员
19	UT:投资者不活跃
20	UT:投资者未在交易所开户
21	UT:该交易席位未连接到交易所
22	UT:报单错误：不允许重复报单
23	UT:错误的报单操作字段
24	UT:撤单已报送，不允许重复撤单
25	UT:撤单找不到相应报单
26	UT:报单已全成交或已撤销，不能再撤
27	UT:不支持的功能
28	UT:没有报单交易权限
29	UT:只能平仓
30	UT:平仓量超过持仓量
31	UT:资金不足
32	UT:主键重复
33	UT:找不到主键
34	UT:设置经纪公司不活跃状态失败
35	UT:经纪公司正在同步
36	UT:经纪公司已同步
37	UT:现货交易不能卖空
38	UT:不合法的结算引用
39	UT:交易所网络连接失败
40	UT:交易所未处理请求超过许可数
41	UT:交易所每秒发送请求数超过许可数
42	UT:结算结果未确认
43	UT:没有对应的入金记录
44	UT:交易所已经进入连续交易状态
45	UT:找不到预埋（撤单）单
46	UT:预埋（撤单）单已经发送
47	UT:预埋（撤单）单已经删除

48	UT:无效的投资者或者密码
49	UT:不合法的登录 IP 地址
50	UT:平今仓位不足
51	UT:平昨仓位不足
52	UT:经纪公司没有足够可用的条件单数量
53	UT:投资者没有足够可用的条件单数量
54	UT:经纪公司不支持条件单
55	UT:重发未知单经纪公司/投资者不匹配
56	UT:同步动态令牌失败
57	UT:动态令牌校验错误
58	UT:找不到动态令牌配置信息
59	UT:不支持的动态令牌类型
60	UT:用户在线会话超出上限
61	UT:该交易所不支持套利/做市商类型报单
62	UT:没有条件单交易权限
63	UT:客户端认证失败
64	UT:客户端未认证
65	UT:该合约不支持互换类型报单
66	UT:该期权合约只支持投机类型报单
67	UT:执行宣告错误, 不允许重复执行
68	UT:重发未知执行宣告经纪公司/投资者不匹配
69	UT:只有期权合约可执行
70	UT:该期权合约不支持执行
71	UT:执行宣告字段有误
72	UT:执行宣告撤单已报送, 不允许重复撤单
73	UT:执行宣告撤单找不到相应执行宣告
74	UT:执行仓位不足
75	UT:连续登录失败次数超限, 登录被禁止
76	UT:非法银期代理关系
77	UT:无此功能
78	UT:发送报单失败
79	UT:发送报单操作失败
80	UT:交易所不支持的价格类型
81	UT:错误的执行类型
82	UT:无效的组合合约
83	UT:该合约不支持询价
84	UT:重发未知报价经纪公司/投资者不匹配
85	UT:该合约不支持报价
86	UT:报价撤单找不到相应报价

87	UT:该期权合约不支持放弃执行
88	UT:该组合期权合约只支持 IOC
89	UT:打开文件失败
90	UTS: 查询未就绪, 请稍后重试
91	UTS: 交易所返回的错误
92	UT:报价衍生单要等待交易所返回才能撤单
93	UT:找不到组合合约映射
94	UT:个股期权只支持投机或者备兑
95	UT:证券只支持投机买开或卖平
96	UT:存在相同的合约编号需指定交易所
97	UT:找到多笔报单
98	UT:锁定仓位不足
99	UT:证券仓位不足
100	UT:该交易所不支持锁定
101	UT:用户在本系统没有报单权限
102	UT:系统缺少灾备标示号
103	UT:该交易所不支持批量撤单
104	UT:数量不正确
105	UT:价格不正确
106	UT:投资者限仓
107	UT:经纪公司限仓
108	UT:证券处置操作字段有误
109	UT:证券处置撤单已报送, 不允许重复撤单
110	UT:证券处置撤单找不到相应证券处置
111	UT:投资者没有该交易指令权限
112	UT:投资者分级权限不足
113	UT:当前时间禁止询价
114	UT:当前价差禁止询价
115	UT:该合约未到行权日
116	UT:下单频率限制
117	UT:拆分数超出限制
118	UT:转备兑只能填组合
119	UT:行权日前一天开始不允许申报价差组合
120	UT:组合数量超出持仓量
121	UT:跨式只支持卖
122	UT:交易所代码错误
123	UT:价格或数量错误
124	UT:找不到行情
125	UT:找不到保证金率

126	UT:找不到手续费率
127	UT:找不到产品资金映射表
128	UT:找不到资金账户
129	UT:找不到产品组
130	UT:找不到终端授权信息
131	UT:密码至少包含数字, 大写字母, 小写字母, 特殊字符中 3 种且不少于 12 位
132	UT:用户名或密码错误
133	UT:该用户未开启短信验证码
134	UT:请求频率限制
135	UT:短信验证码不正确
136	UT:备兑只支持认购卖开或买平
137	UT:需要新版 api 或连接旧端口
138	UT:需要连接新端口或使用旧版 api
139	UT:当前时间禁止行权
140	UT:业务忙, 请再次查询
141	UT:找不到标的行情
142	UT:未订阅行情
143	UT:开仓冻结资金不允许为 0
144	UT:IP 登录失败次数超限, 登录被禁止
145	UT:该交易所当前时间禁止下单
146	UT:修改密码失败。新密码不允许与旧密码相同
147	UT:单次转账金额过大或过小
148	UT:累计转账金额过大或为负值
149	UT:转账类型错误
150	UT:投资者不允许使用该终端
151	UT:该终端不允许从该 IP 登录
152	UT:暂不支持该终端登录
153	UT:找不到合约或行情
154	UT:同一会话报单引用必须递增
155	UT:逆回购只能卖开
156	UT:两融额度不足
157	UT:两融资券不足
158	UT:两融保证金不足
159	UT:该合约不支持融资
160	UT:两融维持保证金低于设定值
161	UT:融资持仓不支持还券
162	UT:还款金额过大或过小
163	UT:找不到信用合约
164	UT:找不到信用投资者

165	UT:该合约不支持担保品买入
166	UT:该合约不支持申赎
167	UT:找不到成分股持仓
168	UT:投保标志不能填申赎
169	UT:成分股持仓不足
170	UT:超出最大替代比例
171	UT:ETF 申赎不支持撤单
172	UT:找不到标的持仓
173	UT:单合约集中度控制
174	UT:小类/组别集中度控制