

# 红楹云数据接口C++版使用手册

## V1.2

版权所有 侵权必究

# 目 录

|         |                                  |    |
|---------|----------------------------------|----|
| 1       | 概述.....                          | 5  |
| 2       | 红楹云数据接口 C++版安装与配置.....           | 6  |
| 2.1     | 系统环境配置要求.....                    | 6  |
| 2.2     | 红楹云数据接口 C++版终端安装.....            | 7  |
| 2.3     | 工程配置示例.....                      | 9  |
| 3       | 数据接口函数.....                      | 12 |
| 3.1     | 数据接口功能函数总列表.....                 | 12 |
| 3.2     | 数据编码定义及规则.....                   | 15 |
| 3.2.1   | 交易所编码.....                       | 16 |
| 3.2.2   | PlateID编码规则.....                 | 16 |
| 3.2.3   | SecurityID编码规则.....              | 16 |
| 3.2.4   | SecurityType证券品种类型.....          | 17 |
| 3.2.5   | PlateType板块类型.....               | 17 |
| 3.2.6   | TradeType/TradingType交易品种类型..... | 18 |
| 3.2.7   | ISIN编码.....                      | 18 |
| 3.2.8   | 列式数据集.....                       | 19 |
| 3.2.9   | 无效/最大值.....                      | 19 |
| 3.3     | 数据接口功能函数明细.....                  | 19 |
| 3.3.1   | 公共API登入登出类:.....                 | 20 |
| 3.3.1.1 | QtLogin: 红楹云数据接口账号登录.....        | 20 |
| 3.3.1.2 | QtLogout: 红楹云数据接口账号登出.....       | 20 |
| 3.3.2   | 提取基本信息类:.....                    | 21 |
| 3.3.2.1 | GetExchanges: 取交易所信息.....        | 21 |
| 3.3.2.2 | GetTradeTypes: 取交易品种信息.....      | 21 |
| 3.3.2.3 | GetTradeCalendar: 取交易日历.....     | 23 |
| 3.3.2.4 | GetPlates: 取板块信息.....            | 24 |
| 3.3.2.5 | GetRelatedPlates: 取相关板块信息.....   | 25 |
| 3.3.2.6 | GetPlateSymbols: 取板块成分清单.....    | 26 |
| 3.3.2.7 | GetSecurityInfo: 取证券基本信息.....    | 27 |
| 3.3.2.8 | GetSecurityCurInfo: 取证券盘前信息..... | 29 |
| 3.3.3   | 提取行情数据类:.....                    | 30 |
| 3.3.3.1 | GetDataByTime: 按时间区间取历史数据.....   | 30 |

|          |  |    |
|----------|--|----|
| 3.3.3.2  | GetDataByCount: 按指定数量取历史数据 .....                   | 32 |
| 3.3.3.3  | GetTickByTime: 按时间区间取Tick数据 .....                  | 34 |
| 3.3.3.4  | GetTickByCount: 按指定数量取Tick数据 .....                 | 36 |
| 3.3.3.5  | GetL2TickByTime: 按时间区间取L2Tick数据 .....              | 37 |
| 3.3.3.6  | GetL2TickByCount: 按指定数量取L2Tick数据 .....             | 39 |
| 3.3.3.7  | GetSnapData: 取快照数据 .....                           | 40 |
| 3.3.4    | 提取因子数据类: .....                                     | 42 |
| 3.3.4.1  | GetFinance: 取财务因子数据 .....                          | 42 |
| 3.3.4.2  | GetFactor: 取风控及量化因子数据 .....                        | 44 |
| 3.3.5    | 提取其他数据类: .....                                     | 45 |
| 3.3.5.1  | GetHisMarketInfo: 取证券历史变动信息 .....                  | 45 |
| 3.3.5.2  | GetJointContracts: 取期货主力或连续合约信息 .....              | 47 |
| 3.3.6    | 行情订阅及回调类: .....                                    | 48 |
| 3.3.6.1  | Subscribe: 订阅行情 .....                              | 48 |
| 3.3.6.2  | UnSubscribe: 退订行情 .....                            | 48 |
| 3.3.6.3  | RegSubStsCB: 注册实时分时数据回调方法 .....                    | 48 |
| 3.3.6.4  | RegSubQuoteCB: 注册实时Tick数据回调方法 .....                | 50 |
| 3.3.6.5  | RegSubL2QuoteCB: 注册实时L2 十档行情回调方法 .....             | 52 |
| 3.3.6.6  | RegSubArbitrageCB: 注册L1 套利行情数据回调方法 .....           | 56 |
| 3.3.6.7  | RegSubStaticCB: 注册静态数据回调方法 .....                   | 56 |
| 3.3.6.8  | RegSubBulletinCB: 注册公告及增值信息数据回调方法 .....            | 56 |
| 3.3.6.9  | RegSubL2AuctionCB: 注册L2 集合竞价数据回调方法 .....           | 56 |
| 3.3.6.10 | RegSubL2IndexCB: 注册L2 指数行情数据回调方法 .....             | 56 |
| 3.3.6.11 | RegSubL2OrderCB: 注册L2 逐笔委托数据回调方法 .....             | 57 |
| 3.3.6.12 | RegSubL2OrderQueueCB: 注册L2 委托队列数据回调方法 .....        | 57 |
| 3.3.6.13 | RegSubL2TransactionCB: 注册L2 逐笔成交数据回调方法 .....       | 57 |
| 3.3.6.14 | RegSubL2StatusCB: 注册L2 证券状态数据回调方法 .....            | 57 |
| 3.3.6.15 | RegSubL2ArbitrageCB: 注册L2 套利行情数据回调方法 .....         | 57 |
| 3.3.6.16 | RegSubL2MarchPriceQtyCB: 注册L2 分价成交量数据回调方法 .....    | 58 |
| 3.3.6.17 | RegSubL2OrderStatisticCB: 注册L2 委托统计数据回调方法 .....    | 58 |
| 3.3.6.18 | RegSubL2SettlePriceCB: 注册L2 实时结算数据回调方法 .....       | 58 |
| 3.3.7    | 工具类函数: .....                                       | 58 |
| 3.3.7.1  | GetTypeSize: 取数据类型字节长度 .....                       | 59 |
| 3.3.7.2  | PrintData(DataColumn,index): 屏幕输出数据列中单个数据 .....    | 59 |
| 3.3.7.3  | PrintData(ColumnSeq): 屏幕输出整个数据集 .....              | 59 |
| 3.3.7.4  | PrintData(ColumnSetSeq): 屏幕输出数据集序列 .....           | 59 |
| 3.3.7.5  | WriteData(ColumnSeq): 输出数据集到文件 .....               | 60 |
| 3.3.7.6  | WriteData(ColumnSetSeq): 输出数据集序列到文件 .....          | 60 |
| 3.3.7.7  | GetStrArray: 将字符串数据列转为字符串数组 .....                  | 60 |
| 3.3.7.8  | DateTime2StrArray: 将DateTime数值数据列转为字符串时间数组 .....   | 61 |
| 3.3.7.9  | TimeStamp2StrArray: 将TimeStamp数值数据列转为字符串时间数组 ..... | 61 |
| 3.3.7.10 | GetDataArrayT: 从数据值数据序列中取数值数组模板方法 .....            | 61 |

|  |     |
|--|-----|
| 3.3.7.11 GetDataT: 从数据值数据列中取单个数值模板方法 .....                                   | 62  |
| 3.3.7.12 GetDataByName: 按数据名取数值数据集中数据 .....                                  | 62  |
| 3.3.7.13 GetDataByIndex: 按行列坐标取数值数据集中数据 .....                                | 62  |
| 3.3.7.14 GetStrByName: 按数据名取字符串数据集中数据 .....                                  | 63  |
| 3.3.7.15 GetStrByIndex: 按行列坐标取字符串数据集中数据 .....                                | 63  |
| 3.3.7.16 Null2Val: 将数值数据列中的填充数据转为用户指定值 .....                                 | 63  |
| 3.3.7.17 示例: 以提取快照列式数据集为例 .....  | 64  |
| 4 红楹云数据接口返回的列式结果集汇总 .....  | 66  |
| 4.1 GetExchanges出参返回的列式结果集字段 .....   | 66  |
| 4.2 GetTradeTypes入参fields及出参返回的列式结果集字段 .....                                 | 66  |
| 4.3 GetTradeCalendar入参fields及出参返回的列式结果集字段 .....                              | 70  |
| 4.4 GetPlates出参返回的列式结果集内容 .....  | 71  |
| 4.5 GetRelatedPlates出参返回的列式结果集内容 .....                                       | 71  |
| 4.6 GetPlateSymbols出参返回的列式结果集内容 .....  | 71  |
| 4.7 GetSecurityInfo入参fields及出参返回的列式结果集字段 .....                               | 72  |
| 4.8 GetSecurityCurInfo入参fields及出参返回的列式结果集字段 .....                            | 76  |
| 4.9 GetDataByTime/GetDataByCount入参fields及出参返回的列式结果集字段 .....                  | 78  |
| 4.10 GetTickByTime/GetTickByCount/GetSnapData入参fields及出参返回的列式结果集字段 .....     | 83  |
| 4.11 GetL2TickByTime/GetL2TickByCount/GetSnapData入参fields及出参返回的列式结果集字段 ..... | 91  |
| 4.12 GetFinance入参fields及出参返回的列式结果集字段 .....                                   | 112 |
| 4.13 GetFactor入参fields及出参返回的列式结果集字段 .....                                    | 128 |
| 4.14 GetHisMarketInfo入参fields及出参返回的列式结果集字段 .....                             | 129 |
| 4.15 GetJointContracts 出参返回的列式结果集内容 .....                                    | 130 |
| 5 附录 1 红楹云数据接口数据与行情定义 .....  | 131 |
| 附录 1.1 数据API定义 .....   | 131 |
| 附录 1.2 行情API定义 .....   | 133 |
| 6 附录 2 错误码定义 .....   | 144 |
| 6.1 红楹云数据接口错误码 .....   | 144 |
| 6.2 DSP错误码 .....   | 146 |
| 7 附录 3 高频数据库内容 .....   | 149 |
| 8 FAQ .....  | 151 |

|                              |     |
|------------------------------|-----|
| 8.1 数据提取结果与输出文件的说明 .....     | 151 |
| 8.2 红楹云数据接口C++版本安装路径说明 ..... | 152 |
| 8.3 红楹云数据接口配置账号密码的文件说明 ..... | 152 |
| 8.4 修改数据接口返回超时时限的文件说明 .....  | 152 |
| 8.5 配置数据接口生产环境路径 .....       | 153 |
| 8.6 免费接口权限说明 .....           | 153 |
| 8.7 金融期货与国债期货数据说明 .....      | 153 |
| 8.8 按市场或按板块订阅说明 .....        | 153 |
| 8.9 代码排序说明 .....             | 153 |

# 1 概述

**红楹云数据接口**，以 API 函数调用的形式提供股票、基金、债券、指数、个股期权、股指期货、商品期权、商品期货、股指期货和国债期货金融市场全品种、全频段（近 3 年 L2 TICK（上交所、深交所、大商所、中金所）、近 3 年 L1 TICK（郑商所、上期所）、商品期货、股指期货近 3 年秒频、全品种近 5 年分钟频、日频、周频、月频、年频）、Level1 和 Level2 的历史数据类、历史及实时衍生数据类、证券基本数据类、风控及量化因子与财务因子类的全字段权威数据信息，可支持用户进行数据分析、数据挖掘及策略研究等方面的应用，目前支持 C++和 Python 两种语言。

本说明书为红楹云数据接口 C++版本使用说明，Python 版本的使用说明详见红楹云数据接口 Python 版本使用说明书。

**红楹云数据接口** 支持的品种和频率如下：

| 品种   | L1TICK | L2TICK | 秒频 | 分钟频 | 日频 | 周频 | 月频 | 年频 |
|------|--------|--------|----|-----|----|----|----|----|
| 股票   |        | √      |    | √   | √  | √  | √  | √  |
| 基金   |        | √      |    | √   | √  | √  | √  | √  |
| 债券   |        | √      |    | √   | √  | √  | √  | √  |
| 指数   |        | √      |    | √   | √  | √  | √  | √  |
| 个股期权 | √      |        |    | √   | √  | √  | √  | √  |
| 股指期货 |        | √      |    | √   | √  | √  | √  | √  |
| 商品期权 | √      | √      |    | √   | √  |    |    |    |
| 商品期货 | √      | √      | √  | √   | √  | √  | √  | √  |
| 股指期货 |        | √      | √  | √   | √  | √  | √  | √  |
| 国债期货 |        | √      |    | √   | √  |    |    |    |

红楹云数据接口核心功能介绍如下：

- ◆ **提取历史数据类：**全面涵盖了股票、基金、债券、指数、商品期货、股指期货、国债期货、个股期权、股指期权、商品期权金融市场全品种、全字段、全频段的历史数据内容，支持高速提取数据；
- ◆ **提取证券基本数据类：**包括交易所、交易品种、交易日历、板块信息、证券基本信息、证券盘前信息、证券历史变动信息和期货主力或连续合约信息等；
- ◆ **提取实时衍生数据类：**订阅实时衍生数据，包括 L1 和 L2 两种；其中 L1 数据包括上交所、深交所、上期所个股期权、郑商所商品期权、上交所、大商所、郑商所商品期货的分时数据、分笔（TICK）数据和大商所 L1 套利行情数据；L2 支持沪深股票、债券、基金、指数、大商所商品期货和商品期权、中金所股指期货、国债期货和股指期权；L2 上交所数据类型：集合竞价、指数行情、委托队列、十档行情、逐笔成交、静态数据；L2 深交所数据类型：指数行情、委托队列、十档行情、逐笔成交、静态数据、逐笔委托、证券状态；L2 大商所商品期货数据类型：分笔高频、套利深度、分价成交量、委托统计、实时结算行情数据；L2 大商所商品期权数据类型：分笔高频、分价成交量、委托统计、实时结算行情数据；L2 中金所国债期货数据类型：分笔数据、分时数据；L2 中金所股指期货数据类型：分笔数据、分时数据；L2 股指期权数据类型：分笔数据、分时数据；
- ◆ **提取风控及量化因子和财务数据类：**红楹云数据接口可调用的量化及财经因子包括风控因子、量化因子和财务数据三种类型；其中风控因子包含行业因子和风格因子；量化因子包含宏观因子、行业因子、基本面因子、市场因子、技术因子、行为因子、高频因子、衍生物因子、事件因子和复合因子。财务数据则包含证券的相关财务信息数据，比如利润表、资产负债表、现金流量表等；
- ◆ **工具函数类：**工具类函数主要用于处理输出的数据集格式，主要包括屏幕输出、数据格式和数据内容转换，可实现按自定义需求提数的功能等，工具类函数为 C++ 版本特有函数类；
- ◆ **多语言版本：**红楹云数据接口目前支持 C++ 版本和 Python 版本，数据接口有效封装，用户可直接调用 API 函数进行便捷的数据提取操作。

## 2 红楹云数据接口 C++版安装与配置

### 2.1 系统环境配置要求

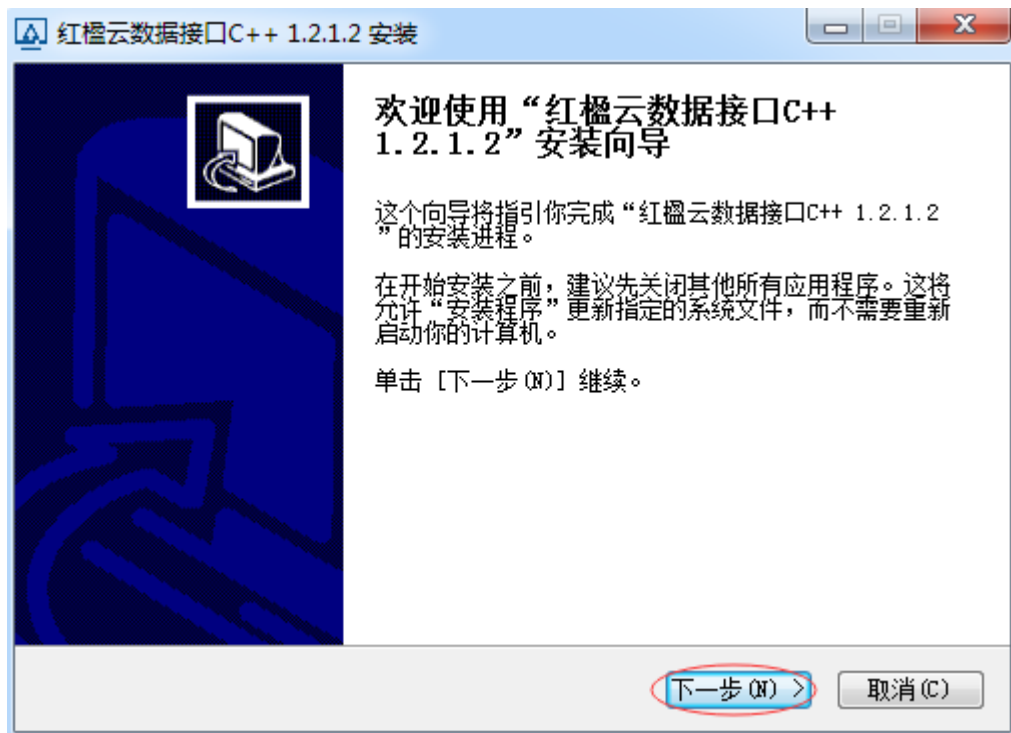
操作系统：Windows 32 位和 64 系统，建议 Windows 7、Windows10（其它 Windows 版本暂未测试）、Server2008、Server2012；

开发工具：VS2010、2013、2015。

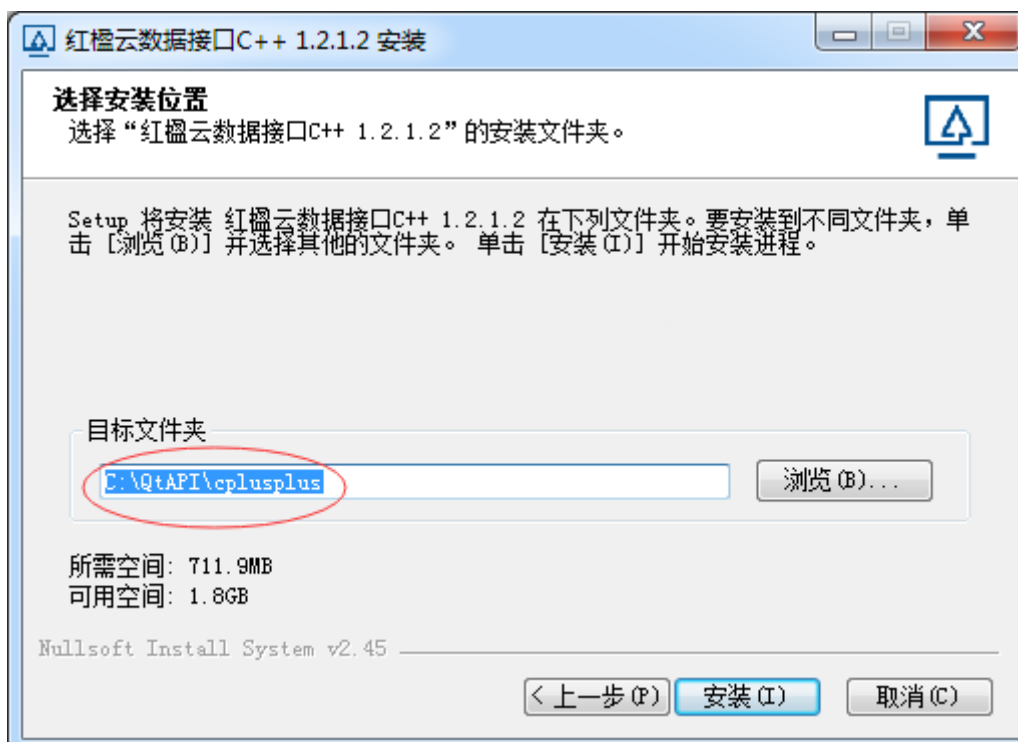
硬件要求：网卡速度  $\geq 100\text{MB}$ ；内存  $\geq 2\text{GB}$ 。

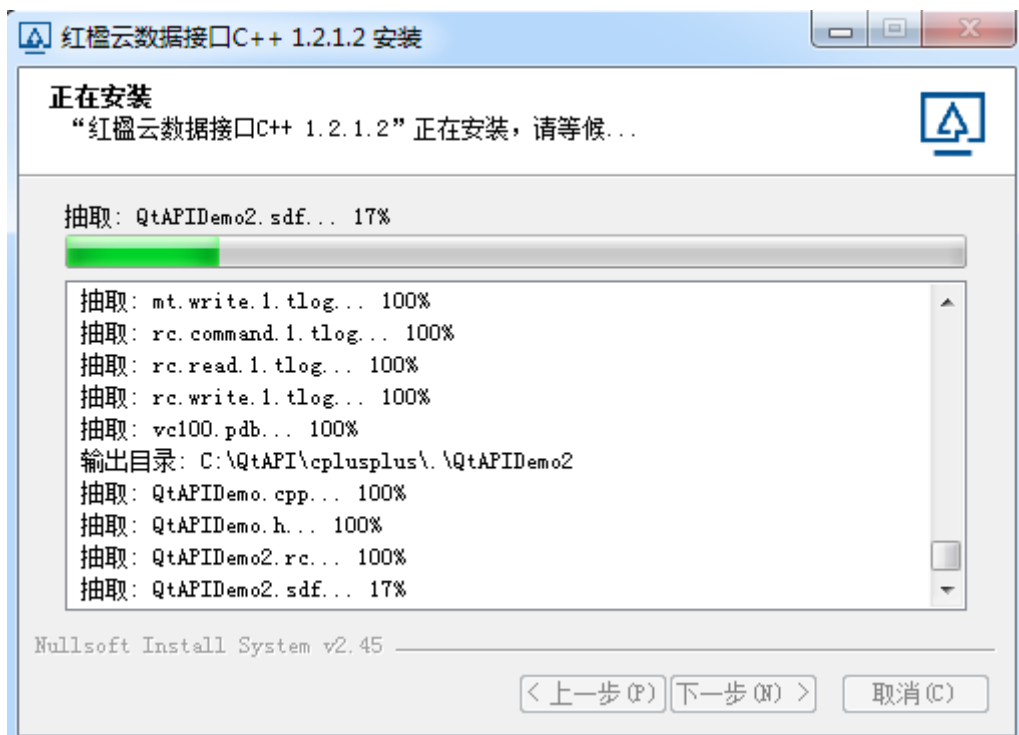
## 2.2 红楹云数据接口 C++版终端安装

双击红楹云数据接口 C++版.exe 文件，进入安装向导，点击下一步，根据向导完成安装。



C++数据接口默认安装在 C 盘 QtAPI 文件 cplusplus 目录下，如需更改安装文件，可点击浏览，然后点击安装。



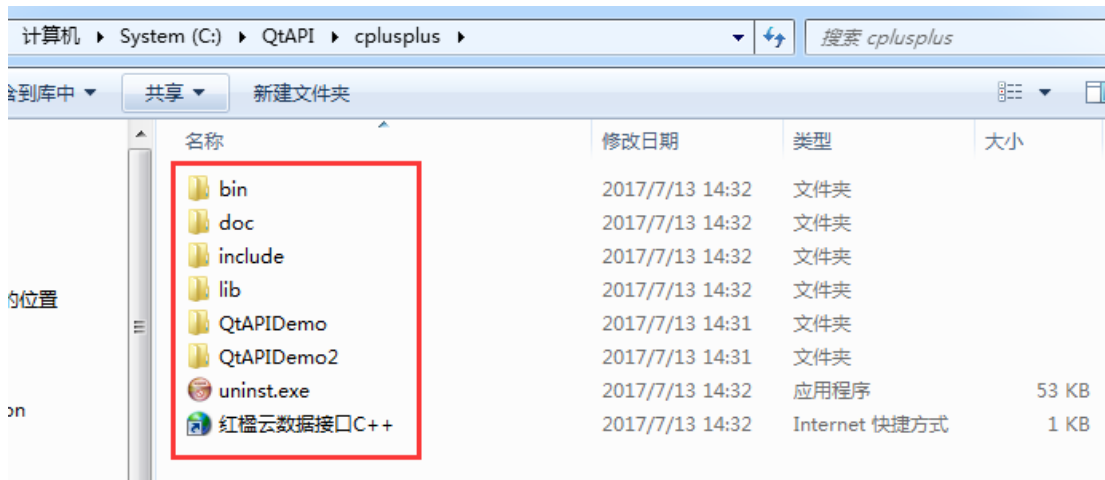


安装成功后，进入安装目录的文件夹，会看到如下 8 个文件，如需卸载双击 **uninst.exe** 应用程序按步骤卸载便可。

**bin** 目录是各种版本的配置好的运行环境，是用户开发的应用程序请放到 **bin** 目录下的对应子目录下予以运行。用户需要自定义运行目录时，请将 **bin** 目录下对应版本子目录下的全部文件复制到用户自定义目录即可。



数据接口 C++示例模板在 QtAPIDemo2 中，其他文件需保持完整才能保证终端的正常运行。

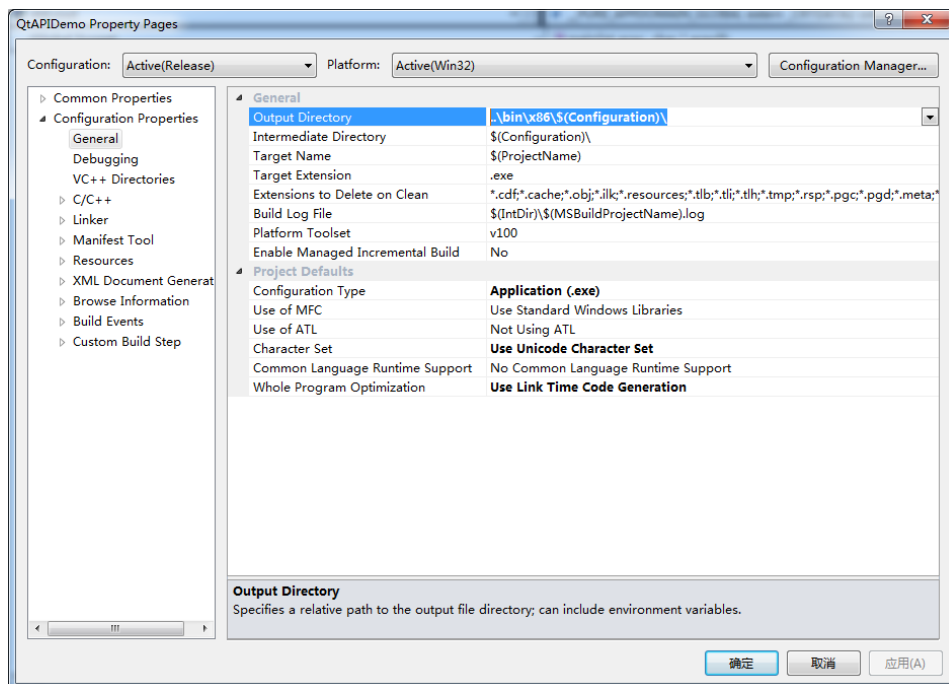


## 2.3 工程配置示例

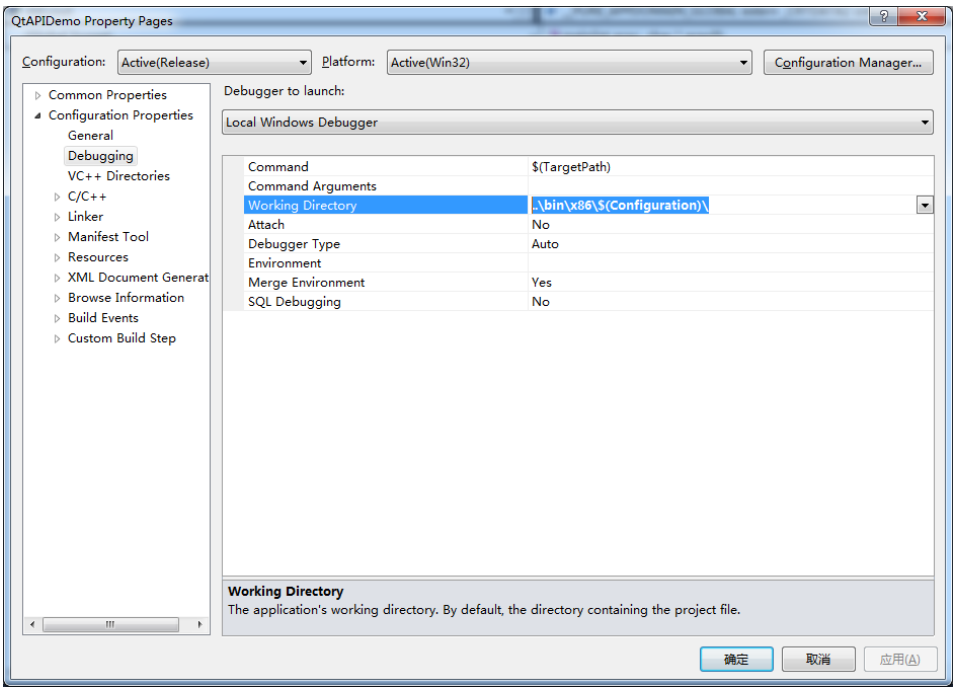
本手册以运行于 Windows7（32 位和 64 位均可）上的 VS2010 为运行和开发环境讲解如何配置使用红楹云数据接口的 C++工程。

为便于用户开发和运行，建议用户自定义的 C++项目目录与 QtAPIDemo 在同一层次。并将生成的执行的程序生成及运行目录设置为 bin 目录下对应版本子目录下。

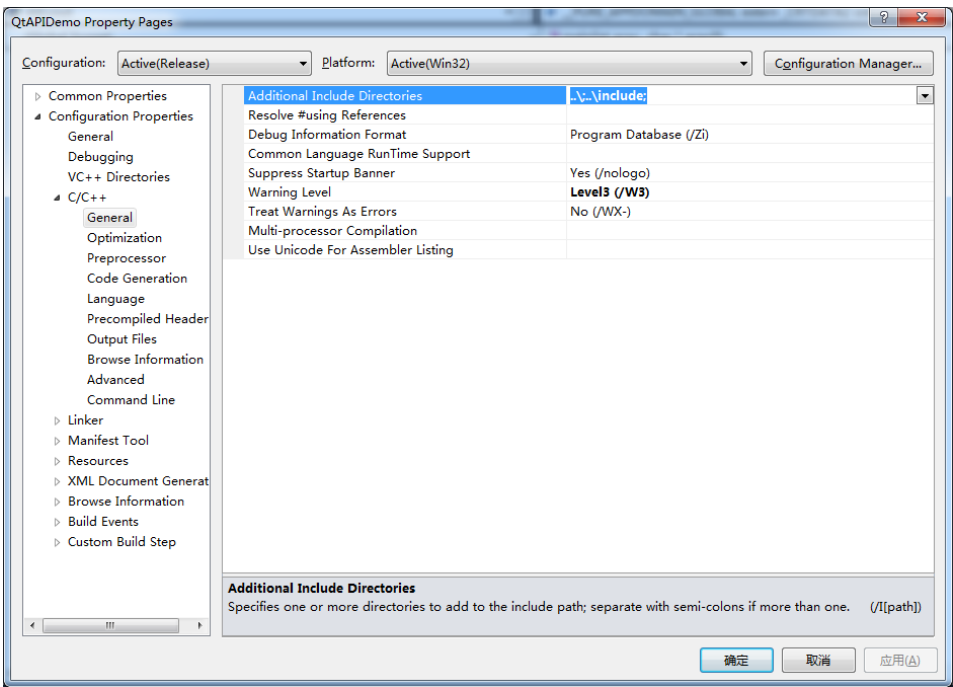
- 使用 VS 新建一个 C++工程（以 QtAPIDemo 为例）；
- 工程属性设置（以 QtAPIDemo 32 位 Debug 为例，其它配置类似）：
  - ◆ 输出目录设置：将 Output Directory 设置为 “..\bin\x86\$(Configuration)”；



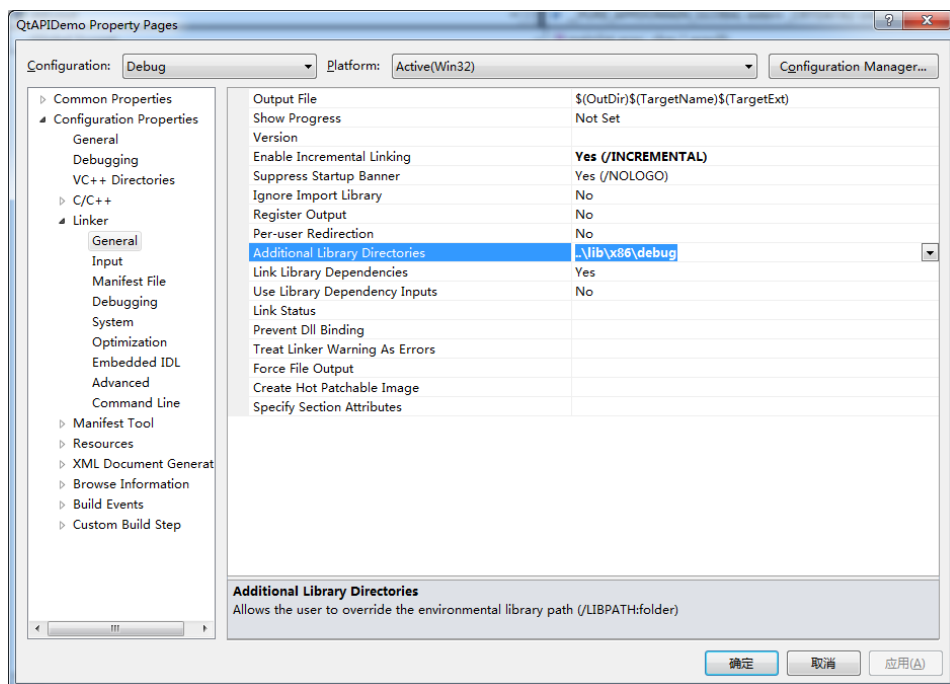
- ◆ 调试目录设置：将 Working Directory 设置为 “..\bin\x86\\$(Configuration)”；



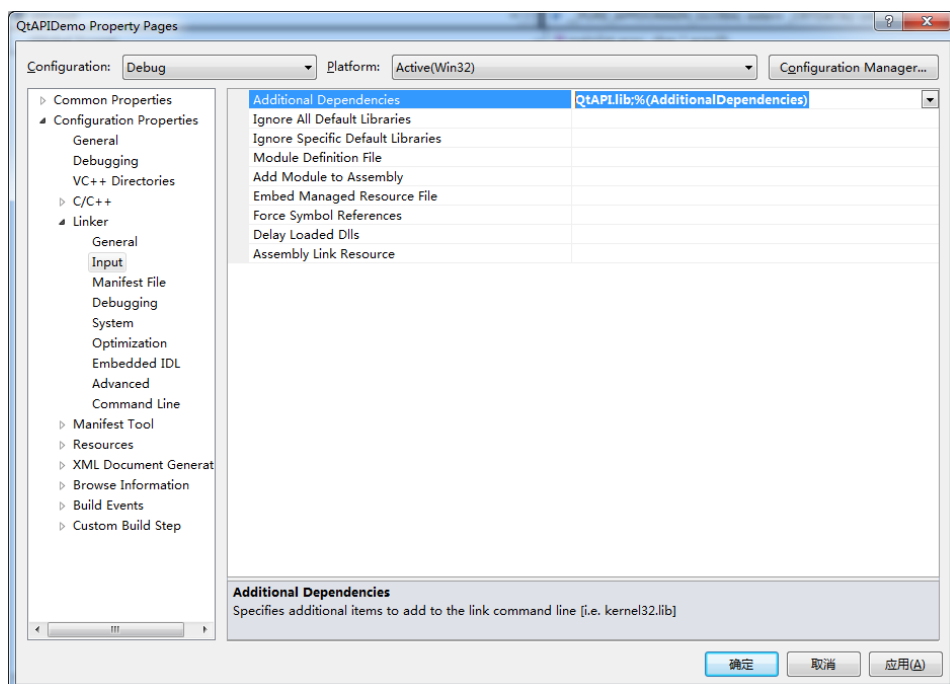
- ◆ 附件包含文件设置：在 Additional Include Directory 中添加 “..\..\include”；



- ◆ 附加静态库目录设置: 在 Additional Library Directory 中添加“..\lib\x86\debug”;



- ◆ 附件静态库文件设置: 在 Additional Dependencies 中添加“QtAPI.lib”;



- ◆ 其它设置: 若使用 QtAPI 的用户程序编译或运行异常, 请参考 QtAPIDemo 工程的工程配置。

## 3 数据接口函数

### 3.1 数据接口功能函数总列表

| 数据接口<br>一览 | 类型              | 函数名                                | 中文名                        |
|------------|-----------------|------------------------------------|----------------------------|
|            | 登入登出类<br>(公共接口) | <a href="#">QtLogin</a>            | 红楹云数据接口账号登录                |
|            |                 | <a href="#">QtLogout</a>           | 红楹云数据接口账号登出                |
|            | 基本信息            | <a href="#">GetExchanges</a>       | 取全部股票、期货等交易所信息列表<br>(免费接口) |
|            |                 | <a href="#">GetTradeTypes</a>      | 取所有交易品种信息(免费接口)            |
|            |                 | <a href="#">GetTradeCalendar</a>   | 取交易日历(免费接口)                |
|            |                 | <a href="#">GetPlates</a>          | 取所有板块信息                    |
|            |                 | <a href="#">GetRelatedPlates</a>   | 取证券所属的所有板块信息               |
|            |                 | <a href="#">GetPlateSymbols</a>    | 取板块包含的证券清单信息               |
|            |                 | <a href="#">GetSecurityInfo</a>    | 提取证券基本信息(静态信息)             |
|            |                 | <a href="#">GetSecurityCurInfo</a> | 提取证券盘前信息                   |
|            | 提取行情<br>数据类     | <a href="#">GetDataByTime</a>      | 按指定时间提取历史分时数据              |
|            |                 | <a href="#">GetDataByCount</a>     | 按指定数量提取历史分时数据              |
|            |                 | <a href="#">GetTickByTime</a>      | 按时间取 TICK 数据               |
|            |                 | <a href="#">GetTickByCount</a>     | 按数量取 TICK 数据               |
|            |                 | <a href="#">GetL2TickByTime</a>    | 按时间取 L2TICK 数据             |

|  |         |                                   |                |
|--|---------|-----------------------------------|----------------|
|  |         | <a href="#">GetL2TickByCount</a>  | 按数量取 L2TICK 数据 |
|  |         | <a href="#">GetSnapData</a>       | 取快照数据          |
|  | 提取因子数据类 | <a href="#">GetFinance</a>        | 取财务因子          |
|  |         | <a href="#">GetFactor</a>         | 取量化及风控因子       |
|  | 提取其他数据类 | <a href="#">GetHisMarketInfo</a>  | 取证券历史变动信息      |
|  |         | <a href="#">GetJointContracts</a> | 取期货主力或连续合约信息   |

|        | 类型       | 函数名                               | 中文名                           |
|--------|----------|-----------------------------------|-------------------------------|
| 数据接口一览 | 行情订阅及回调类 | <a href="#">Subscribe</a>         | 订阅行情数据                        |
|        |          | <a href="#">UnSubscribe</a>       | 退订行情数据                        |
|        |          | <a href="#">RegSubStsCB</a>       | 注册分时数据接收函数                    |
|        |          | <a href="#">RegSubQuoteCB</a>     | 注册行情接收函数                      |
|        |          | <a href="#">RegSubL2QuoteCB</a>   | 注册 L2 行情接收函数                  |
|        |          | <a href="#">RegSubArbitrageCB</a> | 注册套利行情接收函数                    |
|        |          | <a href="#">RegSubStaticCB</a>    | 注册静态行情接收函数                    |
|        |          | <a href="#">RegSubBulletinCB</a>  | 注册公告及增值信息接收函数                 |
|        |          | <a href="#">RegSubL2AuctionCB</a> | 注册 L2 集合竞价行情接收函数<br>( 上交所 )   |
|        |          | <a href="#">RegSubL2IndexCB</a>   | 注册 L2 指数行情接收函数<br>( 上交所、深交所 ) |

|  |       |  |                                 |
|--|-------|--|---------------------------------|
|  |       | <a href="#">RegSubL2OrderCB</a>              | 注册 L2 逐笔委托行情接收函数<br>( 深交所 )     |
|  |       | <a href="#">RegSubL2OrderQueueCB</a>         | 注册 L2 委托队列行情接收函数<br>( 上交所、深交所 ) |
|  |       | <a href="#">RegSubL2TransactionCB</a>        | 注册 L2 逐笔成交行情接收函数<br>( 上交所、深交所 ) |
|  |       | <a href="#">RegSubL2StatusCB</a>             | 注册 L2 证券状态行情接收函数<br>( 深交所 )     |
|  |       | <a href="#">RegSubL2ArbitrageCB</a>          | 注册 L2 套利行情接收函数 ( 大商所 )          |
|  |       | <a href="#">RegSubL2MarchPriceQtyCB</a>      | 注册 L2 分价成交量行情接收函数<br>( 大商所 )    |
|  |       | <a href="#">RegSubL2OrderStatisticCB</a>     | 注册 L2 委托统计行情接收函数<br>( 大商所 )     |
|  |       | <a href="#">RegSubL2SettlePriceCB</a>        | 注册 L2 实时结算价行情接收函数<br>( 大商所 )    |
|  | 工具类函数 | <a href="#">GetTypeSize</a>                  | 取数据类型字节长度                       |
|  |       | <a href="#">PrintData(DataColumn, index)</a> | 屏幕输出数据列中单个数据                    |
|  |       | <a href="#">PrintData(ColumnSeq)</a>         | 屏幕输出整个数据集                       |
|  |       | <a href="#">PrintData(ColumnSetSeq)</a>      | 屏幕输出数据集序列                       |

|  |   |                                |
|--|---|--------------------------------|
|  | <a href="#">WriteData(ColumnSeq)</a>    | 输出数据集到文件                       |
|  | <a href="#">WriteData(ColumnSetSeq)</a> | 输出数据集序列到文件                     |
|  | <a href="#">GetStrArray</a>             | 将字符串数据列转为字符串数组                 |
|  | <a href="#">DateTime2StrArray</a>       | 将 DateTime 数值数据列转为字符串<br>时间数组  |
|  | <a href="#">TimeStamp2StrArray</a>      | 将 TimeStamp 数值数据列转为字符<br>串时间数组 |
|  | <a href="#">GetDataArrayT</a>           | 从数据值数据列中取数值数组模板方<br>法          |
|  | <a href="#">GetDataT</a>                | 从数据值数据列中取单个数值模板方<br>法          |
|  | <a href="#">GetDataByName</a>           | 按数据名取数值数据集中数据                  |
|  | <a href="#">GetDataByIndex</a>          | 按行列坐标取数值数据集中数据                 |
|  | <a href="#">GetStrByName</a>            | 按数据名取字符串数据集中数据                 |
|  | <a href="#">GetStrByIndex</a>           | 按行列坐标取字符串数据集中数据                |
|  | <a href="#">Null2Val</a>                | 将数值数据列中的填充数据转为用户<br>指定值        |

## 3.2 数据编码定义及规则

在红楹云数据接口的输入输出中部分数据按照 GTA 规则进行编码，将常用编码集中在此供参考使用。

### 3.2.1 交易所编码

目前国内交易所的标准代码如下：

SSE： 上海证券交易所；  
SZSE： 深圳证券交易所  
DCE： 大连商品交易所  
CZCE： 郑州商品交易所  
SHFE： 上海期货交易所  
CFFEX： 中国金融期货交易所

其他及香港、国外交易所的标准代码如下：

XXX： 其他  
HKEX： 香港联合交易所  
CFETS： 全国银行间交易市场  
SGX： 新加坡交易所  
NYSE： 纽约证券交易所  
AIM： 伦敦创业板市场  
SQS： 股份报价系统  
AMEX： 美国证券交易所

### 3.2.2 PlateID 编码规则

PlateID 编码规则为级别 0（2 位）+级别 1（2 位）+级别 2（3 位）+级别 3（3 位）…，目前级别 0 的编码：股票为 10，基金为 13，指数为 14，期货为 17，债券为 20，期权为 21。如沪深股票为级别 1，编码 10，后面每一级别编码均是 001,002，依次向后扩展，如 1012003001。

### 3.2.3 SecurityID 编码规则

证券 ID 生成规则：‘前 3 位’+9 位自增长，定长；

|           |     |
|-----------|-----|
| 股票：       | 201 |
| 基金：       | 202 |
| 债券：       | 203 |
| 指数：       | 204 |
| 商品期货（国内）： | 205 |
| 商品期货（国外）： | 206 |
| 股指期货：     | 207 |
| 国债期货（国内）： | 208 |
| 个股期权：     | 209 |
| 股指期权：     | 210 |
| 商品期权：     | 218 |
| 香港权证/窝轮：  | 211 |
| 香港基金：     | 212 |



香港牛熊证： 213  
期货主力连续合约： 299

### 3.2.4 SecurityType 证券品种类型

S0101:股票  
S0102:债券  
S0103:基金  
S0104:指数  
S0105:权证/窝轮  
S0106:资产管理计划  
S0107:商品期货  
S0108:股指期货  
S0109:国债期货  
S0110:币种  
S0111:信托  
S0112:其他  
S0113:理财产品  
S0114:ETF 期权  
S0115:股指期权  
S0116:商品期权  
S0117:牛熊证  
S0118:个股期权  
S0199:期货主力连续合约

### 3.2.5 PlateType 板块类型

注：目前支持股票板块和指数板块。

股票板块：  
P4901：市场类  
P4902：市场特征类  
P4903：证监会行业类  
P4904：申银万国行业类  
P4905：中证行业类  
P4906：证监会行业类（2012 版）  
P4910：地域类(市级)  
P4911：概念类  
P4912：指数成份类  
P4913：机构持股类  
P4914：权证类

指数板块:

P4920: 所属市场

P4921: 指数类别

P4922: 指数类型

### 3.2.6 TradeType/TradingType 交易品种类型

|    |       |    |       |       |         |
|----|-------|----|-------|-------|---------|
| A  | 豆一    | JM | 焦煤    | SM    | 锰硅      |
| AG | 沪白银   | JR | 粳稻谷   | SN    | 沪锡      |
| AL | 沪铝    | L  | 聚乙烯   | SR    | 白糖      |
| AU | 沪黄金   | LR | 晚籼稻   | TA    | 苯甲酸 PTA |
| B  | 豆二    | M  | 豆粕    | TC    | 动力煤     |
| BB | 胶合板   | MA | 甲醇    | V     | 聚氯乙烯    |
| BU | 沪石油沥青 | ME | 动力煤   | WH    | 强麦      |
| C  | 玉米    | NI | 沪镍    | WR    | 沪线材     |
| CF | 棉花    | OI | 菜籽油   | WS    | 强麦      |
| CS | 玉米淀粉  | P  | 棕榈油   | WT    | 硬麦      |
| CU | 沪铜    | PB | 沪铅    | Y     | 豆油      |
| ER | 早籼稻   | PM | 普麦    | ZC    | 动力煤     |
| FB | 纤维板   | PP | 聚丙烯   | ZN    | 沪锌      |
| FG | 玻璃    | RB | 沪螺纹钢  | S0101 | 股票      |
| FU | 沪燃料油  | RI | 早籼稻   | S0102 | 债券      |
| GN | 绿豆    | RM | 菜籽粕   | S0103 | 基金      |
| HC | 沪热轧卷板 | RO | 菜籽油   | S0104 | 指数      |
| I  | 铁矿石   | RS | 油菜籽   | S0108 | 股指期货    |
| J  | 焦炭    | RU | 沪天然橡胶 | S0109 | 国债期货    |
| JD | 鸡蛋    | SF | 硅铁    |       |         |

### 3.2.7 ISIN 编码

国际证券识别编码（ISIN编码）是由 国际标准化组织（ISO）制定的证券编码标准，并在《证券及相关金融工具—国际证券识别编码体系》（ISO6166）中正式发布。ISO6166 主要规定了ISIN编码的具体编码方法及执行维护机构，编码覆盖的范围包括股票、债券、期货、期权、权证等 金融工具 。ISO6166 目前已被许多证券行业国际组织所接受，如证券登记结算 30 人专家小组(G30)、国际清算银行 支付清算系统委员会(CPSS)、国际证监会组织 (IOSCO)、国际证券业行政人员协会 (ISSA) 均将应用ISIN编码作为一条主要建议，用以衡量一个国家或地区金融市场国际化、标准化建设的水平。在组成结构上，ISIN编码由 12 位字母或数字构成，主要包括三部分：前缀是由 2 位字母组成的国家或地区代码(ISO3166)；基本号码为 9 位字母或数字，由各个国家和地区自行编制；最后 1 位是 校验码。国家编码机构协会(ANNA) 是ISIN编码的全球执行维护机构，一个国家或地区的特定机构可在申请成为ANNA会员(NNA) 后负责为该国或地区的证券编制、维护ISIN编码。

### 3.2.8 列式数据集

为适配客户动态组合查询的因子种类和数据数量，并利于优化处理和数据压缩，数据API返回的数据为列式数据集结构（具体定义见 [附录 1.1](#)）。为方便用于处理列式数据集，红楹云数据接口提供相关数据集相关方法，详见工具类函数说明明细。

列式数据集中每列为同一种数据类型，按用户请求的列数返回数据。

对于字符串数据，按字符串截止符进行分割，以优化空间占用。

处理返回的列式数据集时也需要对字符串数据列进行特殊处理，具体由各编程语言具体处理；而其它类型数据列使用相同的下标值可实现按行数据提取。

### 3.2.9 无效/最大值

数据API中，为数据列对齐以方便用于使用需要，对于不存在的数据，以对应数据类型的最大值填充，用户收到数据时可按如下定义自行处理填充数据或使用Null2Val方法将填充数据转换为指定数值。

**填充数据不会计入用户数据流量。**

```
//各数据类型的最大值表示无效值/数据不存在
//#define MaxByte      0xff,
//#define MaxInt8      0x7f,
//#define MaxUInt8     0xff,
//#define MaxInt16     0x7fff,
//#define MaxUInt16    0xffff,
//#define MaxInt32     0x7fffffff,
//#define MaxUInt32    0xffffffff,
//#define MaxInt64     0xffffffffffffffff,
//#define MaxUInt64    0xffffffffffffffff,
//#define MaxFloat     (0x7f7fffff)f,      //3.402823466e+38f
//#define MaxDouble    0x7fefffffffffffff, //1.7976931348623158e+308
```

## 3.3 数据接口功能函数明细

**声明：**

**\_I\_：** 函数的参数前面带有此标识的表示入参；函数参数前面没有任何标识的也为入参

**\_O\_：** 函数的参数前面带有此标识的表示出参

**入参为必填表示：** 该入参为必选参数，如果不输入必填入参，API 将无法正常运行。

**入参为选填表示：** 该入参为可选参数，如果不需要指定的选填入参，可以不输入选填入参，API 也可以正常运行。

**红楹云数据接口使用流程：** 必须先登录红楹云数据接口账号成功后，才可以正常调用后续API；取实时衍生数据需用行情 API 提前订阅相关行情。

**红楹云数据接口账号密码配置：** 账号密码配置文件为 QtAPIDemo.id 文件，打开此文件后输入有效的用户名（usr）与密码（pwd）才能成功登录红楹云数据接口。

### 3.3.1 公共 API 登入登出类:

#### 3.3.1.1 QtLogin: 红楹云数据接口账号登录

| <b>int QtLogin(_O_ string&amp; errorMsg, _I_ string qtUser, _I_ string qtPwd, _I_ string option = "");</b>  |        |        |           |
|---|--------|--------|-----------|
| 输入红楹云数据接口账号和密码，成功登录账号后，才可调用其他数据接口函数   |        |        |           |
| 参数  | 类型     | 入参/出参  | 描述        |
| qtUser  | string | 入参（必填） | 红楹云数据接口账号 |
| qtPwd   | string | 入参（必填） | 红楹云数据接口密码 |
| option  | string | 入参（选填） | 预留参数，默认为空 |
| errorMsg  | string | 出参     | 错误描述      |
| <b>返回值:</b><br>QR_Success 成功<br>QR_QtAPI_Login   QR_Retry_Limit 登录重试超限<br>QR_QtAPI_Login   QR_Invalid_Param 参数无效<br>QR_QtAPI_Login   QR_Network_Fail 网络或UPMS服务错误<br>QR_QtAPI_Login   QR_Invalid_GTAUserPwd 用户或密码错误<br>QR_QtAPI_Login   QR_Fail 其它可能错误 |        |        |           |
| <b>示例:</b><br>登录红楹云数据接口账号，代码如下： <pre>if(QR_Success != QtLogin(errMsg, gtausr, gtapwd))     cout &lt;&lt; "[x] QtLogin Failed: " &lt;&lt; errMsg &lt;&lt; endl; else     cout &lt;&lt; "[i] QtLogin Success!" &lt;&lt; endl;</pre>                     |        |        |           |

#### 3.3.1.2 QtLogout: 红楹云数据接口账号登出

| <b>int QtLogout(_O_ string&amp; errorMsg, _I_ string qtUser);</b>          |        |        |           |
|--|--------|--------|-----------|
| 登出红楹云数据接口  |        |        |           |
| 参数   | 类型     | 入参/出参  | 描述        |
| qtUser   | string | 入参（必填） | 红楹云数据接口账号 |
| errorMsg   | string | 出参     | 错误描述      |
| <b>返回值:</b><br>错误码，0表示登出成功，其他表示登出失败，可对照错误码信息表（ <a href="#">附录2</a> ）查看错误信息 |        |        |           |
| <b>示例:</b><br>登出红楹云数据接口，代码如下： <pre>QtLogout(errMsg, gtausr);</pre>         |        |        |           |

### 3.3.2 提取基本信息类:

#### 3.3.2.1 GetExchanges: 取交易所信息

| <pre>int GetExchanges(_O_ ColumnSeq&amp; cols, _O_ string&amp; errorMsg,                  string countryCode = "", string reserve = "");</pre>  |           |        |           |
|---|-----------|--------|-----------|
| 取全部股票、期货等交易所信息列表  |           |        |           |
| 参数  | 类型        | 入参/出参  | 描述        |
| countrycode   | string    | 入参（选填） | 国家标准代码    |
| reserve   | string    | 入参（选填） | 预留参数，默认为空 |
| cols  | ColumnSeq | 出参     | 列式结果集     |
| errorMsg  | string    | 出参     | 错误描述      |
| <b>返回值:</b><br>列式结果集（见 <a href="#">4.1出参返回的列式结果集字段</a> ）<br>错误码，可对照错误码信息表（ <a href="#">附录2</a> ）查看错误信息  |           |        |           |
| <b>示例:</b><br>取中国证券市场上的全部交易所信息，代码如下：<br>ColumnSeq cols; <pre>if(QR_Success == GetExchanges(cols, errMsg, "CHN")) {     cout &lt;&lt; "[i] GetExchanges Success!" &lt;&lt; endl;     ofile &lt;&lt; "[i] GetExchanges Success:" &lt;&lt; endl;     WriteData(cols, ofile); } else {     cout &lt;&lt; "[x] GetExchanges Failed: " &lt;&lt; errMsg &lt;&lt; endl;     ofile &lt;&lt; "[x] GetExchanges Failed: " &lt;&lt; errMsg &lt;&lt; endl; }</pre> |           |        |           |

#### 3.3.2.2 GetTradeTypes: 取交易品种信息

| <pre>int GetTradeTypes(_O_ ColumnSeq&amp; cols, _O_ string&amp; errorMsg,                   string securityType, StringSeq&amp; fields,                   string date = "", int timezone = 8,                   string reserve = "");</pre> |           |        |                                |
|---|-----------|--------|--------------------------------|
| 取所有交易品种信息（仅支持商品期货(S0107)、金融期货(S0108股指期货和S0119国债期货)、ETF期权(S0114)、股指期货(S0115)、商品期权（S0116）品种的查询）   |           |        |                                |
| 参数  | 类型        | 入参/出参  | 描述                             |
| <a href="#">securityType</a>  | string    | 入参（必填） | 证券品种类型                         |
| fields  | StringSeq | 入参（必填） | 数据字段，一个""表示提取全部字段；如需提取指定字段，需输入 |

|          |           |        |   |
|----------|-----------|--------|---|
|          |           |        | 相应的字段名，支持多字段同时提取，可选择输入的字段名见 <a href="#">4.2入参fields支持的字段</a>      |
| date     | string    | 入参（选填） | 日期，调用时不指定日期，即不对date赋值，则提取当前所有可交易品种的最新信息；若调用时指定日期则提取该日所有可交易品种的品种信息 |
| timezone | int       | 入参（选填） | 时区  |
| reserve  | string    | 入参（选填） | 预留参数，默认为空   |
| cols     | ColumnSeq | 出参     | 列式结果集（本次只支持商品期货、金融期货、ETF期权、股指期货和商品期权品种的查询）                        |
| errorMsg | string    | 出参     | 错误描述  |

**返回信息：**

错误码

列式结果集（见[4.2出参返回的列式结果集字段](#)）错误描述，可对照错误码信息表（[附录2](#)）查看错误信息**示例：**

取商品期货所有交易品种的交易品种、交易所代码、品种ID、交易单位、最小变动价位、交易手续费和交割日期数据，代码如下：

```
ColumnSeq cols;
```

```
StringSeq fields;
```

```
string securityType = "S0107";
```

```
fields.push_back("TradingType");
```

```
fields.push_back("Market");
```

```
fields.push_back("VarietyID");
```

```
fields.push_back("TradingUnit");
```

```
fields.push_back("MinChangeUnit");
```

```
fields.push_back("TradingFee");
```

```
fields.push_back("DeliveryDate");
```

```
if(QR_Success == GetTradeTypes(cols, errMsg, securityType, fields))
```

```
{
```

```
    cout << "[i] GetTradeTypes Success!" << endl;
```

```
    ofile << "[i] GetTradeTypes Success!" << endl;
```

```
    WriteData(cols, ofile);
```

```
}
```

```
else
```

```
{
```

```
    cout << "[x] GetTradeTypes Failed: " << errMsg << endl;
```

```
    ofile << "[x] GetTradeTypes Failed: " << errMsg << endl;
```

```
}
```

### 3.3.2.3 GetTradeCalendar: 取交易日历

| <pre>int GetTradeCalendar(_O_ ColumnSeq&amp; cols, _O_ string&amp; errorMsg,     StringSeq&amp; exchangeCode, string dateBegin,     string dateEnd, StringSeq&amp; fields,     ByteSeq&amp; dayOfWeek = ByteSeq(),     StringSeq&amp; tradeType = StringSeq(),     EQtBool open = k_QtNull, int timezone = 8,     string securityType = "", string reserve = "");</pre> |                                    |        |   |
|---|------------------------------------|--------|---|
| 取交易日历数据   |                                    |        |   |
| 参数  | 类型                                 | 入参/出参  | 描述  |
| <a href="#">exchangeCode</a>  | StringSeq                          | 入参（必填） | 交易所标准代码   |
| dateBegin   | string                             | 入参（必填） | 开始查询时间，YYYY-MM-DD   |
| dateEnd   | string                             | 入参（必填） | 截止查询时间，YYYY-MM-DD   |
| fields  | StringSeq                          | 入参（必填） | 数据字段，一个"*"表示提取全部字段；如需提取指定字段，需输入相应的字段名，支持多字段同时提取，可选择输入的字段名见 <a href="#">4.3入参fields支持的字段</a>   |
| dayOfWeek   | ByteSeq                            | 入参（选填） | 0~6，表示周日~周六，允许多项输入（每字节一项），不配置时认为无该参数限制  |
| tradeType   | StringSeq                          | 入参（选填） | 交易品种代码，不配置表示无该参数限制  |
| open  | EQtBool（参见 <a href="#">附录1.1</a> ） | 入参（选填） | 开市/休市，默认为true，不配置表示无该参数限制   |
| timezone  | int                                | 入参（选填） | 参数中时间参数的时区，默认为东8区   |
| <a href="#">securityType</a>  | string                             | 入参（选填） | 证券品种代码，不配置表示无该参数限制（取交易日历时，对于securityType为S0101、S0102、S0103、S0104、S0108、S0109的tradeType也填同样的值） |
| reserve   | string                             | 入参（选填） | 预留参数，默认为空   |
| cols  | ColumnSeq                          | 出参     | 列式结果集   |
| errorMsg  | string                             | 出参     | 错误描述  |
| <b>返回值：</b><br>列式结果集（见 <a href="#">4.3出参返回的列式结果集字段</a> ）<br>错误码，可对照错误码信息表（ <a href="#">附录2</a> ）查看错误信息  |                                    |        |   |
| <b>示例：</b><br>提取上海证券交易所（股票、债券、基金、指数）2016-09-01到2016-09-18之间的日历日期、市场编码、是否开盘、品种ID以及是否夜盘交易的交易日历数据，代码如下： <pre>ColumnSeq cols; StringSeq fields; StringSeq markets;</pre>  |                                    |        |   |

```

markets.push_back("SSE");

fields.push_back("CalendarDate");
fields.push_back("Market");
fields.push_back("IsOpen");
fields.push_back("TradingType");
fields.push_back("IsNightTrading");

if(QR_Success == GetTradeCalendar(cols, errMsg, markets, "2016-9-1", "2016-9-18", fields))
{
    cout << "[i] GetTradeCalendar Success!" << endl;
    ofile << "[i] GetTradeCalendar Success!" << endl;
    WriteData(cols, ofile);
}
else
{
    cout << "[x] GetTradeCalendar Failed: " << errMsg << endl;
    ofile << "[x] GetTradeCalendar Failed: " << errMsg << endl;
}

```

### 3.3.2.4 GetPlates: 取板块信息

```

int GetPlates(_O_ ColumnSeq& cols, _O_ string& errorMsg,
    bool includeHis = false,
    StringSeq& securityType = StringSeq(),
    StringSeq& plateType = StringSeq(),
    StringSeq& nodeLevel = StringSeq(),
    string plateTreeID = "", string reserve = "");

```

取所有板块（股票板块或指数板块）信息（支持按证券类型、按板块类型、按板块数、按板块级别查询）

| 参数                           | 类型        | 入参/出参  | 描述                         |
|------------------------------|-----------|--------|----------------------------|
| includeHis                   | bool      | 入参（选填） | 是否包含历史板块定义信息，默认为false查当前信息 |
| <a href="#">securityType</a> | StringSeq | 入参（选填） | 证券品种类型，不配置表示全部             |
| <a href="#">plateType</a>    | StringSeq | 入参（选填） | 板块类型（仅支持股票板块或指数板块），不配置表示全部 |
| nodeLevel                    | StringSeq | 入参（选填） | 板块级别                       |
| plateTreeID                  | string    | 入参（选填） | 根板块                        |
| reserve                      | string    | 入参（选填） | 预留参数，默认为空                  |
| cols                         | ColumnSeq | 出参     | 列式结果集                      |
| errorMsg                     | string    | 出参     | 错误描述                       |

**返回值：**

列式结果集（见[4.4出参返回的列式结果集字段](#)）

错误码，可对照错误码信息表（[附录2](#)）查看错误信息



**示例：**

取股票证监会行业类板块的所有板块信息（查当前信息）数据，代码如下：

```
ColumnSeq cols;
StringSeq securityTypes, plateTypes;

securityTypes.push_back("S0101");
plateTypes.push_back("P4906");

if(QR_Success == GetPlates(cols, errMsg, false, securityTypes, plateTypes))
{
    cout << "[i] GetPlates Success!" << endl;
    ofile << "[i] GetPlates Success!" << endl;
    WriteData(cols, ofile);
}
else
{
    cout << "[x] GetPlates Failed: " << errMsg << endl;
    ofile << "[x] GetPlates Failed: " << errMsg << endl;
}
```

**3.3.2.5 GetRelatedPlates：取相关板块信息**

```
int GetRelatedPlates(_O_ ColumnSetSeq& cols, _O_ string& errorMsg,
                    StringSeq& securities,
                    SecurityIDSeq& securityIDs = SecurityIDSeq(),
                    string date = "", int timezone = 8,
                    string reserve = "");
```

取证券（股票或指数）归属的所有板块信息

| 参数                          | 类型            | 入参/出参  | 描述                                    |
|-----------------------------|---------------|--------|---------------------------------------|
| securities                  | StringSeq     | 入参（必填） | 字符型证券标的列表，支持多证券同时查询                   |
| <a href="#">securityIDs</a> | SecurityIDSeq | 入参（选填） | 数字型证券标的列表，支持多证券同时查询，与securities二者只用其一 |
| date                        | string        | 入参（选填） | 日期，若未指定则为当前日期                         |
| timezone                    | int           | 入参（选填） | 参数中时间参数的时区，默认为东8区                     |
| reserve                     | string        | 入参（选填） | 预留参数，默认为空                             |
| cols                        | ColumnSetSeq  | 出参     | 列式结果集                                 |
| errorMsg                    | string        | 出参     | 错误描述                                  |

**返回值：**

列式结果集（见[4.5出参返回的列式结果集字段](#)）

错误码，可对照错误码信息表（[附录2](#)）查看错误信息

**示例：**

取股票平安银行归属的所有板块信息数据，代码如下：

```

ColumnSetSeq colsSets;
StringSeq securities;

securities.push_back("000001.SZSE");

if(QR_Success == GetRelatedPlates(colsSets, errMsg, securities))
{
    cout << "[i] GetRelatedPlates Success!" << endl;
    ofile << "[i] GetRelatedPlates Success!" << endl;
    WriteData(colsSets, ofile);
}
else
{
    cout << "[x] GetPlates Failed: " << errMsg << endl;
    ofile << "[x] GetPlates Failed: " << errMsg << endl;
}

```

### 3.3.2.6 GetPlateSymbols: 取板块成分清单

```

int GetPlateSymbols(_O_ ColumnSeq& cols, _O_ string& errorMsg,
    UInt64Seq& plateIDs, ESetOper setOper,
    SortDefineSeq& sortDefs = SortDefineSeq(),
    string dateBegin = "", string dateEnd = "",
    int timezone = 8, string reserve = "");

```

取板块包含的证券（股票或指数）清单信息（只支持查询叶子节点）

| 参数        | 类型                                  | 入参/出参  | 描述   |
|-----------|-------------------------------------|--------|--|
| plateIDs  | UInt64Seq                           | 入参（必填） | 板块列表，支持多板块（仅支持股票或指数板块）并且plateID需事先通过 GetPlates 或 GetRelatedPlates 函数获取 |
| setOper   | ESetOper（参见 <a href="#">附录1.1</a> ） | 入参（必填） | 结果的集合处理方式，如并集（Union）或交集（Inter）   |
| sortDefs  | SortDefineSeq                       | 入参（选填） | 排序要求，包括不排序（None）、降序（Desc）和升序（Asec）                                     |
| dataBegin | string                              | 入参（选填） | 起始日期，不配置表示当前最新   |
| dateEnd   | string                              | 入参（选填） | 截止日期，不配置表示当前最新   |
| timezone  | int                                 | 入参（选填） | 参数中时间参数的时区，默认为东8区  |
| reserve   | string                              | 入参（选填） | 预留参数，默认为空  |
| cols      | ColumnSeq                           | 出参     | 列式结果集  |
| errorMsg  | string                              | 出参     | 错误描述   |

返回值：

列式结果集（见[4.6出参返回的列式结果集字段](#)）

错误码，可对照错误码信息表（[附录2](#)）查看错误信息

示例：

取1006010001沪深股票板块叶子节点包含的证券清单信息数据，代码如下：

```
ColumnSeq cols;
UInt64Seq plateIDs;
ESetOper setOper = k_SetUnion;

plateIDs.push_back(1006010001);

if(QR_Success == GetPlateSymbols(cols, errMsg, plateIDs, setOper))
{
    cout << "[i] GetPlateSymbol Success!" << endl;
    ofile << "[i] GetPlateSymbol Success!" << endl;
    WriteData(cols, ofile);
}
else
{
    cout << "[x] GetPlateSymbols Failed: " << errMsg << endl;
    ofile << "[x] GetPlateSymbols Failed: " << errMsg << endl;
}
```

### 3.3.2.7 GetSecurityInfo: 取证券基本信息

```
int GetSecurityInfo(_O_ ColumnSeq& cols, _O_ string& errorMsg,
StringSeq& securities, SecurityIDSeq& securityIDs,
StringSeq& fields,
SortDefineSeq& sortDefs = SortDefineSeq(),
string updateTime = "", string reserve = "");
```

提取股票、基金、债券、指数、商品期货、金融期货、个股期权、股指期货、商品期权的基本信息；支持按证券品种、交易所、板块定义提取数据

| 参数                          | 类型            | 入参/出参  | 描述  |
|-----------------------------|---------------|--------|---|
| securities                  | StringSeq     | 入参（必填） | 证券代码输入格式:"代码.交易所"<br>（暂不支持提取期货主力和次主力合约的数据）;交易所输入格式:"*.交易所";板块输入格式:"板块ID.*"                  |
| <a href="#">securityIDs</a> | SecurityIDSeq | 入参（必填） | 数字型证券标的列表，支持多证券同时查询，与securities二者只用其一   |
| fields                      | StringSeq     | 入参（必填） | 数据字段，一个"*"表示提取全部字段；如需提取指定字段，需输入相应的字段名，支持多字段同时提取，可选择输入的字段名见 <a href="#">4.7入参fields支持的字段</a> |
| sortDefs                    | SortDefineSeq | 入参（选填） | 排序要求，包括不排序（None）、降序（Desc）和升序（Asec）  |
| updateTime                  | string        | 入参（选填） | 查询更新记录起始时间（不含该时   |

|          |           |         |            |
|----------|-----------|---------|------------|
|          |           |         | 刻)         |
| reserve  | string    | 入参 (选填) | 预留参数, 默认为空 |
| cols     | ColumnSeq | 出参      | 列式结果集      |
| errorMsg | string    | 出参      | 错误描述       |

**返回值:**

列式结果集 (见[4.7出参返回的列式结果集字段](#))

错误码, 可对照错误码信息表 ([附录2](#)) 查看错误信息

**示例:**

取股票平安银行、万科A、浦发银行和中国石化的证券代码、交易所代码、中文简称、证券上市状态编码、上市日期和发行价格的静态信息数据, 代码如下:

```
ColumnSeq cols;
StringSeq fields;
StringSeq securities;
SecurityIDSeq securityIDS;

securities.push_back("000001.SZSE");
securities.push_back("000002.SZSE");
securities.push_back("600000.SSE");
securities.push_back("600028.SSE");

fields.push_back("Symbol");
fields.push_back("Market");
fields.push_back("ShortName");
fields.push_back("StatusID");
fields.push_back("ListedDate");
fields.push_back("IssuePrice");

if(QR_Success == GetSecurityInfo(cols, errMsg, securities, securityIDS, fields))
{
    cout << "[i] GetSecurityInfo Success!" << endl;
    ofile << "[i] GetSecurityInfo Success!" << endl;
    WriteData(cols, ofile);
}
else
{
    cout << "[x] GetSecurityInfo Failed: " << errMsg << endl;
    ofile << "[x] GetSecurityInfo Failed: " << errMsg << endl;
}
```

### 3.3.2.8 GetSecurityCurInfo: 取证券盘前信息

```
int GetSecurityCurInfo(_O_ ColumnSeq& cols, _O_ string& errorMsg,
    StringSeq& securities, SecurityIDSeq& securityIDs,
    StringSeq& fields,
    SortDefineSeq& sortDefs = SortDefineSeq(),
    string dateBegin = "", string dateEnd = "",
    int timezone = 8, string reserve = "");
```

提取股票、基金、债券、指数、商品期货、股指期货、国债期货、个股期权、股指期权的盘前信息；支持按证券品种、交易所、板块定义提取数据

| 参数                          | 定义            | 入参/出参  | 描述  |
|-----------------------------|---------------|--------|---|
| securities                  | StringSeq     | 入参（必填） | 证券代码输入格式:"代码.交易所"<br>(暂不支持提取期货主力和次主力合约的数据);交易所输入格式:"*.交易所";板块输入格式:"板块ID.*"                  |
| <a href="#">securityIDs</a> | SecurityIDSeq | 入参（必填） | 数字型证券标的列表,支持多证券同时查询,与securities二者只用其一   |
| fields                      | StringSeq     | 入参（必填） | 数据字段,一个"*"表示提取全部字段;如需提取指定字段,需输入相应的字段名,支持多字段同时提取,可选择输入的字段名见 <a href="#">4.8入参fields支持的字段</a> |
| sortDefs                    | SortDefineSeq | 入参（选填） | 排序要求,包括不排序（None）、降序（Desc）和升序（Asec）  |
| dataBegin                   | string        | 入参（选填） | 起始日期,为空则为当前日期   |
| dateEnd                     | string        | 入参（选填） | 截止日期,为空则为当前日期   |
| timezone                    | int           | 入参（选填） | 参数中时间参数的时区,默认为东8区   |
| reserve                     | string        | 入参（选填） | 预留参数,默认为空   |
| cols                        | ColumnSeq     | 出参     | 列式结果集   |
| errorMsg                    | string        | 出参     | 错误描述  |

#### 返回值:

列式结果集（见[4.8出参返回的列式结果集字段](#)）

错误码,可对照错误码信息表（[附录2](#)）查看错误信息

#### 示例:

取股票平安银行、万科A、浦发银行和中国石化的证券代码、交易所代码、中文简称、总股本、前收盘价、涨停价、跌停价、每股净资产、本年每股利润的盘前信息数据,代码如下:

```
ColumnSeq cols;
```

```
StringSeq fields;
```

```
StringSeq securities;
```

```
SecurityIDSeq securityIDs;
```

```
securities.push_back("000001.SZSE");
```

```

securities.push_back("000002.SZSE");
securities.push_back("600000.SSE");
securities.push_back("600028.SSE");

fields.push_back("Symbol");
fields.push_back("Market");
fields.push_back("ShortName");
fields.push_back("TotalShare");
fields.push_back("LCP");
fields.push_back("LimitUp");
fields.push_back("LimitDown");
fields.push_back("NAPS");
fields.push_back("EPS");

if(QR_Success == GetSecurityCurInfo(cols, errMsg, securities, securityIDS, fields))
{
    cout << "[i] GetSecurityCurInfo Success!" << endl;
    ofile << "[i] GetSecurityCurInfo Success!" << endl;
    WriteData(cols, ofile);
}
else
{
    cout << "[x] GetSecurityCurInfo Failed: " << errMsg << endl;
    ofile << "[x] GetSecurityCurInfo Failed: " << errMsg << endl;
}

```

### 3.3.3 提取行情数据类:

#### 3.3.3.1 GetDataByTime: 按时间区间取历史数据

```

int GetDataByTime(_O_ColumnSeq& cols, _O_string& errorMsg,
StringSeq& securities, SecurityIDSeq& securityIDS,
StringSeq& fields, EQuoteType quoteType, uint interval,
TimePeriodSeq& timePeriods,
SortDefineSeq& sortDefs = SortDefineSeq(),
int timezone = 8, EPriceAdjust priceAdj = k_AdjNone,
string reserve = "");

```

按时间取历史分时数据，默认数据先按日期和时间排序，再按证券代码排序；支持按证券品种、交易所、板块定义提取数据

| 参数         | 定义        | 入参/出参  | 描述   |
|------------|-----------|--------|--|
| securities | StringSeq | 入参（必填） | 证券代码输入格式:"代码.交易所"<br>（暂不支持提取期货主力和次主力合约的数据）;交易所输入格式:"*.交易所";板块输入格式:"板块ID.*" |

|  |   |        |   |
|--|---|--------|---|
| <a href="#">securityIDs</a>  | SecurityIDSeq                           | 入参（必填） | 数字型证券标的列表，支持多证券同时查询，与securities二者只用其一   |
| fields   | StringSeq                               | 入参（必填） | 数据字段，一个"*"表示提取全部字段；如需提取指定字段，需输入相应的字段名，支持多字段同时提取，可选择输入的字段名见 <a href="#">4.9入参fields支持的字段</a> |
| quoteType  | EQuoteType（参见 <a href="#">附录1.1</a> ）   | 入参（必填） | 行情采样类型（行情采样频率，秒（商品期货、股指期货1、3、5、10、15、30秒）、全品种分钟频（1、5、10、15、30、60分钟）、日（1天）、周、月、年）            |
| interval   | unit                                    | 入参（必填） | 采样间隔数值，单位按quotePeriod确定   |
| timePeriods  | TimePeriodSeq                           | 入参（必填） | 时间段[begin, end]，支持多时间段合并取值  |
| sortDefs   | SortDefineSeq                           | 入参（选填） | 排序要求，基于数据名指明排序要求  |
| timezone   | int                                     | 入参（选填） | 参数中时间参数的时区，默认为东8区   |
| priceAdj   | EPriceAdjust（参见 <a href="#">附录1.1</a> ） | 入参（选填） | 复权类型（仅低频），行情展示需要，包括不复权(AdjNone)、前复权（AdjForward）和后复权（AdjBackward）                            |
| reserve  | string                                  | 入参（选填） | 预留参数，默认为空   |
| cols   | ColumnSeq                               | 出参     | 列式结果集   |
| errorMsg   | string                                  | 出参     | 错误描述  |
| <b>返回值：</b><br>列式结果集（见 <a href="#">4.9出参返回的列式结果集字段</a> ）<br>错误码，可对照错误码信息表（ <a href="#">附录2</a> ）查看错误信息   |   |        |   |
| <b>示例：</b><br>取2017-02-01到2017-03-01和2017-04-01到2017-05-01两个时间段内股票平安银行、万科A、浦发银行和中国石化5分钟证券代码、交易所代码、交易时间、开盘价、最高价、最低价和收盘价的历史数据，代码如下：<br><pre>ColumnSeq cols; StringSeq fields; StringSeq securities; SecurityIDSeq securityIDs;  TimePeriodSeq timePeriods; TimePeriod time;  time.begin = "2017-02-01 00:00:00";</pre> |   |        |   |

```
time.end = "2017-03-01 00:00:00";
timePeriods.push_back(time);
time.begin = "2017-04-01 00:00:00";
time.end = "2017-05-01 00:00:00";
timePeriods.push_back(time);

securities.push_back("000001.SZSE");
securities.push_back("000002.SZSE");
securities.push_back("600000.SSE");
securities.push_back("600028.SSE");

fields.push_back("Symbol");
fields.push_back("Market");
fields.push_back("TradingTime");
fields.push_back("OP");
fields.push_back("HIP");
fields.push_back("LOP");
fields.push_back("CP");

if(QR_Success == GetDataByTime(cols, errMsg, securities,
    securityIDS, fields, k_Minute, 5, timePeriods))
{
    cout << "[i] GetDataByTime Success!" << endl;
    ofile << "[i] GetDataByTime Success!" << endl;
    WriteData(cols, ofile);
}
else
{
    cout << "[x] GetDataByTime Failed: " << errMsg << endl;
    ofile << "[x] GetDataByTime Failed: " << errMsg << endl;
}
```

3.3.3.2 GetDataByCount: 按指定数量取历史数据

```
int GetDataByCount(_O_ ColumnSeq& cols, _O_ string& errorMsg,
    StringSeq& securities, SecurityIDSeq& securityIDS,
    StringSeq& fields, EQuoteType quoteType, uint interval,
    string datetime, EDirection dir, uint count,
    SortDefineSeq& sortDefs = SortDefineSeq(),
    int timezone = 8, EPriceAdjust priceAdj = k_AdjNone,
    string reserve = "");
```

按数量取历史分时数据，默认数据先按日期和时间排序，再按证券代码排序;支持按证券品种、交易所、板块定义提取数据；由于按数量提取数据，不同品种不同频率无法确定统一规则估算数据期限，故为方便实现规定如下：数据只能跨至多一个月，数据区间在上月1日至下月月末之间

| 参数         | 定义        | 入参/出参  | 描述                |
|------------|-----------|--------|-------------------|
| securities | StringSeq | 入参（必填） | 证券代码输入格式:"代码.交易所" |



|   |  |        |  |
|---|--|--------|--|
|   |  |        | (暂不支持提取期货主力和次主力合约的数据);交易所输入格式:"*.交易所";板块输入格式:"板块ID.*"  |
| <a href="#">securityIDs</a>   | SecurityIDSeq                            | 入参(必填) | 数字型证券标的列表,支持多证券同时查询,与securities二者只用其一  |
| fields  | StringSeq                                | 入参(必填) | 数据字段,一个"*"表示提取全部字段;如需提取指定字段,需输入相应的字段名,支持多字段同时提取,可选择输入的字段名见 <a href="#">4.9 入参fields支持的字段</a> |
| quoteType   | EQuoteType (参见 <a href="#">附录1.1</a> )   | 入参(必填) | 行情采样类型(行情采样频率,秒(商品期货、股指期货1、3、5、10、15、30秒)、全品种分钟频(1、5、10、15、30、60分钟)、日(1天)、周、月、年)             |
| interval  | unit                                     | 入参(必填) | 采样间隔数值,单位按quotePeriod确定  |
| datetime  | string                                   | 入参(必填) | 基准日期及时间 YYYY-MM-DD HH:MM:SS, 不含本时刻数据   |
| dir   | EDirection (参见 <a href="#">附录1.1</a> )   | 入参(必填) | 方向,包括向前(Forward)和向后(Backward);当dir=1(向前取时)最多从上个月的月初到查询的时间点;当dir=2(向后取时)最多查询从开始时间点到下个月的月末     |
| count   | unit                                     | 入参(必填) | 要求的数据数量  |
| sortDefs  | SortDefineSeq                            | 入参(选填) | 排序要求,基于数据名指明排序要求   |
| timezone  | int                                      | 入参(选填) | 参数中时间参数的时区,默认为东8区  |
| priceAdj  | EPriceAdjust (参见 <a href="#">附录1.1</a> ) | 入参(选填) | 复权类型(仅低频),行情展示需要,包括不复权(AdjNone)、前复权(AdjForward)和后复权(AdjBackward)                             |
| reserve   | string                                   | 入参(选填) | 预留参数,默认为空  |
| cols  | ColumnSeq                                | 出参     | 列式结果集  |
| errorMsg  | string                                   | 出参     | 错误描述   |
| <b>返回值:</b><br>列式结果集(见 <a href="#">4.9 出参返回的列式结果集字段</a> )<br>错误码,可对照错误码信息表( <a href="#">附录2</a> )查看错误信息     |  |        |  |
| <b>示例:</b><br>取2017-03-24日凌晨12点前向前5分钟的100条股票平安银行、万科A、浦发银行和中国石化5分钟证券代码、交易所代码、交易时间、开盘价、最高价、最低价和收盘价的历史数据,代码如下: |  |        |  |

```

ColumnSeq cols;
StringSeq fields;
StringSeq securities;
SecurityIDSeq securityIDS;

securities.push_back("000001.SZSE");
securities.push_back("000002.SZSE");
securities.push_back("600000.SSE");
securities.push_back("600028.SSE");

fields.push_back("Symbol");
fields.push_back("Market");
fields.push_back("TradingTime");
fields.push_back("OP");
fields.push_back("HIP");
fields.push_back("LOP");
fields.push_back("CP");

if(QR_Success == GetDataByCount(cols, errMsg, securities, securityIDS,
    fields, k_Minute, 5, "2017-03-24 00:00:00.000", k_Forward, 100))
{
    cout << "[i] GetDataByCount Success!" << endl;
    ofile << "[i] GetDataByCount Success!" << endl;
    WriteData(cols, ofile);
}
else
{
    cout << "[x] GetDataByCount Failed: " << errMsg << endl;
    ofile << "[x] GetDataByCount Failed: " << errMsg << endl;
}

```

### 3.3.3.3 GetTickByTime: 按时间区间取 Tick 数据

```

int GetTickByTime(_O_ ColumnSeq& cols, _O_ string& errorMsg, string security,
    SecurityID securityID, StringSeq& fields,
    TimePeriodSeq& timePeriods, int timezone = 8,
    string reserve = "");

```

按时间取Tick分笔数据，由于tick数据量大，所以tick数据不支持多代码同时提取

| 参数                         | 定义         | 入参/出参  | 描述                               |
|----------------------------|------------|--------|----------------------------------|
| security                   | string     | 入参（必填） | 证券代码，只支持单代码（暂不支持提取期货主力和次主力合约的数据） |
| <a href="#">securityID</a> | SecurityID | 入参（必填） | 数字型证券标的列表，与security二者只用其一，只支持单代码 |
| fields                     | StringSeq  | 入参（必填） | 数据字段，一个""表示提取全部                  |

|             |               |        |  |
|-------------|---------------|--------|--|
|             |               |        | 字段；如需提取指定字段，需输入相应的字段名，支持多字段同时提取，可选择输入的字段名见 <a href="#">4.10入参fields支持的字段</a> |
| timePeriods | TimePeriodSeq | 入参（必填） | 时间段[begin, end]，支持多时间段合并取值   |
| timezone    | int           | 入参（选填） | 参数中时间参数的时区，默认为东8区  |
| reserve     | string        | 入参（选填） | 预留参数，默认为空  |
| cols        | ColumnSeq     | 出参     | 列式结果集  |
| errorMsg    | string        | 出参     | 错误描述   |

**返回值：**

列式结果集（见[4.10出参返回的列式结果集字段](#)）

错误码，可对照错误码信息表（[附录2](#)）查看错误信息

**示例：**

取平安银行2017-03-24到2017-03-25时间段内证券代码、交易所代码、交易时间、开盘价、最高价、最低价和收盘价的TICK数据，代码如下：

```
ColumnSeq cols;
StringSeq fields;
string security = "000001.SZSE";
SecurityID securityID = 0;

TimePeriodSeq timePeriods;
TimePeriod time;

time.begin = "2017-03-24 00:00:00";
time.end = "2017-03-25 00:00:00";
timePeriods.push_back(time);

fields.push_back("Symbol");
fields.push_back("Market");
fields.push_back("TradingTime");
fields.push_back("OP");
fields.push_back("HIP");
fields.push_back("LOP");
fields.push_back("CP");

if(QR_Success == GetTickByTime(cols, errMsg, security,
    securityID, fields, timePeriods))
{
    cout << "[i] GetTickByTime Success!" << endl;
    ofile << "[i] GetTickByTime Success!" << endl;
    WriteData(cols, ofile);
}
```

```

else
{
    cout << "[x] GetTickByTime Failed: " << errMsg << endl;
    ofile << "[x] GetTickByTime Failed: " << errMsg << endl;
}

```

### 3.3.3.4 GetTickByCount: 按指定数量取 Tick 数据

```

int GetTickByCount(_O_ ColumnSeq& cols, _O_ string& errorMsg,
    string security, SecurityID securityID,
    StringSeq& fields, string datetime, EDirection dir,
    uint count, int timezone = 8, string reserve = "");

```

按数量取Tick分笔数据，由于tick数据量大，所以tick数据不支持多代码同时提取；由于按数量提取数据，不同品种不同频率无法确定统一规则估算数据期限，故为方便实现规定如下：数据只能跨至多一个月，数据区间在上月1日至下月月末之间

| 参数  | 定义                                    | 入参/出参  | 描述   |
|---|---------------------------------------|--------|--|
| security  | string                                | 入参（必填） | 证券代码，只支持单代码（暂不支持提取期货主力和次主力合约的数据）   |
| <a href="#">securityID</a>  | SecurityID                            | 入参（必填） | 数字型证券标的列表，与security二者只用其一，只支持单代码   |
| fields  | StringSeq                             | 入参（必填） | 数据字段，一个"*"表示提取全部字段；如需提取指定字段，需输入相应的字段名，支持多字段同时提取，可选择输入的字段名见 <a href="#">4.10</a> <a href="#">参fields支持的字段</a> |
| datetime  | string                                | 入参（必填） | 基准日期及时间 YYYY-MM-DD HH:MM:SS，含本时刻数据   |
| dir   | EDirection（参见 <a href="#">附录1.1</a> ） | 入参（必填） | 方向，包括向前（Forward）和向后（Backward）；当dir=1（向前取时）最多从上个月的月初到查询的时间点；当dir=2（向后取时）最多查询从开始时间点到下个月的月末                     |
| count   | unit                                  | 入参（必填） | 要求的数据数量  |
| timezone  | int                                   | 入参（选填） | 参数中时间参数的时区，默认为东8区  |
| reserve   | string                                | 入参（选填） | 预留参数，默认为空  |
| cols  | ColumnSeq                             | 出参     | 列式结果集  |
| errorMsg  | string                                | 出参     | 错误描述   |
| <b>返回值：</b><br>列式结果集（见 <a href="#">4.10出参返回的列式结果集字段</a> ）<br>错误码，可对照错误码信息表（ <a href="#">附录2</a> ）查看错误信息 |                                       |        |  |
| <b>示例：</b><br>取平安银行2017-03-24凌晨12点向前100条证券代码、交易所代码、交易时间、开盘价、最高价、最低                                      |                                       |        |  |

价和收盘价的TICK数据，代码如下：

```
ColumnSeq cols;
StringSeq fields;
string security = "000001.SZSE";
SecurityID securityID = 0;

fields.push_back("Symbol");
fields.push_back("Market");
fields.push_back("TradingTime");
fields.push_back("OP");
fields.push_back("HIP");
fields.push_back("LOP");
fields.push_back("CP");

if(QR_Success == GetTickByCount(cols, errMsg, security, securityID,
    fields, "2017-03-24 00:00:00.000", k_Forward, 100))
{
    cout << "[i] GetTickByCount Success!" << endl;
    ofile << "[i] GetTickByCount Success!" << endl;
    WriteData(cols, ofile);
}
else
{
    cout << "[x] GetTickByCount Failed: " << errMsg << endl;
    ofile << "[x] GetTickByCount Failed: " << errMsg << endl;
}
```

### 3.3.3.5 GetL2TickByTime: 按时间区间取 L2Tick 数据

```
int GetL2TickByTime(_O_ ColumnSeq& cols, _O_ string& errorMsg,
    string security, SecurityID securityID,
    EQuoteData quoteData, StringSeq& fields,
    TimePeriodSeq& timePeriods, int timezone = 8,
    string reserve = "");
```

L2分笔数据支持深交所所有品种、上交所所有品种、大商所商品期货和商品期权和中金所所有品种；按时间取L2 Tick数据，由于tick数据量大，所以tick数据不支持多代码同时提取；由于tick数据量较大，建议每次提取时间段至多为1个月

| 参数                         | 定义                                    | 入参/出参  | 描述                               |
|----------------------------|---------------------------------------|--------|----------------------------------|
| security                   | string                                | 入参（必填） | 证券代码，只支持单代码                      |
| <a href="#">securityID</a> | SecurityID                            | 入参（必填） | 数字型证券标的列表，与security二者只用其一，只支持单代码 |
| quoteData                  | EQuoteData（参见 <a href="#">附录1.1</a> ） | 入参（必填） | 行情数据类型                           |
| fields                     | StringSeq                             | 入参（必填） | 数据字段，一个""表示提取全部字段；如需提取指定字段，需输入   |

|             |               |        |   |
|-------------|---------------|--------|---|
|             |               |        | 相应的字段名，支持多字段同时提取，可选择输入的字段名见 <a href="#">4.11入参fields支持的字段</a> |
| timePeriods | TimePeriodSeq | 入参（必填） | 时间段[begin, end]，支持多时间段合并取值                                    |
| timezone    | int           | 入参（选填） | 参数中时间参数的时区，默认为东8区   |
| reserve     | string        | 入参（选填） | 预留参数，默认为空   |
| cols        | ColumnSeq     | 出参     | 列式结果集   |
| errorMsg    | string        | 出参     | 错误描述  |

**返回值：**

列式结果集（见[4.11出参返回的列式结果集字段](#)）

错误码，可对照错误码信息表（[附录2](#)）查看错误信息

**示例：**

取深交所股票平安银行 2015-03-24到2017-03-25 时间段内证券代码、交易所代码、交易时间、开盘价、最高价、最低价和收盘价的L2TICK数据，代码如下：

```
ColumnSeq cols;
StringSeq fields;
string security = "000001.SZSE";
SecurityID securityID = 0;

TimePeriodSeq timePeriods;
TimePeriod time;

time.begin = "2017-03-24 00:00:00";
time.end = "2017-03-25 00:00:00";
timePeriods.push_back(time);

fields.push_back("Symbol");
fields.push_back("Market");
fields.push_back("TradingTime");
fields.push_back("OP");
fields.push_back("HIP");
fields.push_back("LOP");
fields.push_back("CP");

if(QR_Success == GetL2TickByTime(cols, errMsg, security,
    securityID, k_Quote, fields, timePeriods))
{
    cout << "[i] GetL2TickByTime Success!" << endl;
    ofile << "[i] GetL2TickByTime Success!" << endl;
    WriteData(cols, ofile);
}
else
```

```
{
    cout << "[x] GetL2TickByTime Failed: " << errMsg << endl;
    ofile << "[x] GetL2TickByTime Failed: " << errMsg << endl;
}
```

### 3.3.3.6 GetL2TickByCount: 按指定数量取 L2Tick 数据

| <pre>int GetL2TickByCount(_O_ ColumnSeq&amp; cols, _O_ string&amp; errorMsg,     string security, SecurityID securityID,     EQuoteData quoteData, StringSeq&amp; fields,     string datetime, EDirection dir, uint count,     int timezone = 8, string reserve = "");</pre> |                                       |        |  |
|--|---------------------------------------|--------|--|
| <p>L2分笔数据支持深交所所有品种、上交所所有品种、大商所商品期货和商品期权、中金所所有品种；按数量取L2 Tick数据，由于tick数据量大，所以tick数据不支持多代码同时提取；由于按数量提取数据，不同品种不同频率无法确定统一规则估算数据期限，故为方便实现规定如下：数据只能跨至多一个月，数据区间在上月1日至下月月末之间</p>  |                                       |        |  |
| 参数   | 定义                                    | 入参/出参  | 描述   |
| security   | string                                | 入参（必填） | 证券代码，只支持单代码  |
| <a href="#">securityID</a>   | SecurityID                            | 入参（必填） | 数字型证券标的列表，与security二者只用其一，只支持单代码   |
| quoteData  | EQuoteData（参见 <a href="#">附录1.1</a> ） | 入参（必填） | 行情数据类型   |
| fields   | StringSeq                             | 入参（必填） | 数据字段，一个"*"表示提取全部字段；如需提取指定字段，需输入相应的字段名，支持多字段同时提取，可选择输入的字段名见 <a href="#">4.11入参fields支持的字段</a> |
| datetime   | string                                | 入参（必填） | 基准日期及时间 YYYY-MM-DD HH:MM:SS，含本时刻数据   |
| dir  | EDirection（参见 <a href="#">附录1.1</a> ） | 入参（必填） | 方向，包括向前（Forward）和向后（Backward）；当dir=1（向前取时）最多从上个月的月初到查询的时间点；当dir=2（向后取时）最多查询从开始时间点到下个月的月末     |
| count  | unit                                  | 入参（必填） | 要求的数据数量  |
| timezone   | int                                   | 入参（选填） | 参数中时间参数的时区，默认为东8区  |
| reserve  | string                                | 入参（选填） | 预留参数，默认为空  |
| cols   | ColumnSeq                             | 出参     | 列式结果集  |
| errorMsg   | string                                | 出参     | 错误描述   |
| <p><b>返回值：</b><br/>           列式结果集（见<a href="#">4.11出参返回的列式结果集字段</a>）<br/>           错误码，可对照错误码信息表（<a href="#">附录2</a>）查看错误信息</p>   |                                       |        |  |
| <p><b>示例：</b><br/>           取深交所股票平安银行 2015-03-24凌晨12点前向前取100条证券代码、交易所代码、交易时间、开盘</p>  |                                       |        |  |

价、最高价、最低价和收盘价的L2TICK数据，代码如下：

```
ColumnSeq cols;
StringSeq fields;
string security = "000001.SZSE";
SecurityID securityID = 0;

fields.push_back("Symbol");
fields.push_back("Market");
fields.push_back("TradingTime");
fields.push_back("OP");
fields.push_back("HIP");
fields.push_back("LOP");
fields.push_back("CP");

if(QR_Success == GetL2TickByCount(cols, errMsg, security,
    securityID, k_Quote, fields, "2017-03-24 00:00:00.000", k_Forward, 100))
{
    cout << "[i] GetL2TickByCount Success!" << endl;
    ofile << "[i] GetL2TickByCount Success!" << endl;
    WriteData(cols, ofile);
}
else
{
    cout << "[x] GetL2TickByCount Failed: " << errMsg << endl;
    ofile << "[x] GetL2TickByCount Failed: " << errMsg << endl;
}
```

### 3.3.3.7 GetSnapData: 取快照数据

```
int GetSnapData(_O_ ColumnSeq& cols, _O_ string& errorMsg,
    StringSeq& securities, SecurityIDSeq& securityIDs,
    EQuoteData quoteData, StringSeq& fields,
    SortDefineSeq& sortDefs = SortDefineSeq(),
    string reserve = "");
```

取快照数据（实为当前最近一笔分笔数据）；支持按证券品种、交易所、板块定义提取数据

| 参数                          | 定义            | 入参/出参  | 描述   |
|-----------------------------|---------------|--------|--|
| securities                  | StringSeq     | 入参（必填） | 证券代码输入格式:"代码.交易所"<br>（暂不支持提取期货主力和次主力合约的数据）;交易所输入格式:"*.交易所";板块输入格式:"板块ID.*" |
| <a href="#">securityIDs</a> | SecurityIDSeq | 入参（必填） | 数字型证券标的列表，支持多证券同时查询，与securities二者只用其一                                      |
| quoteData                   | EQuoteData（参见 | 入参（必填） | 行情数据类型   |



|   |                         |        |  |
|---|-------------------------|--------|--|
|   | <a href="#">附录1.1</a> ) |        |  |
| fields  | StringSeq               | 入参（必填） | 数据字段，一个"*"表示提取全部字段；如需提取指定字段，需输入相应的字段名，支持多字段同时提取，L1数据可选择输入的字段名见 <a href="#">4.10入参fields支持的字段</a> ，L2数据可选择输入的字段名见 <a href="#">4.11入参fields支持的字段</a> |
| sortDefts   | SortDefineSeq           | 入参（选填） | 排序要求，基于数据名指明排序要求   |
| reserve   | string                  | 入参（选填） | 预留，默认为空  |
| cols  | ColumnSetSeq            | 出参     | 列式结果集  |
| errorMsg  | string                  | 出参     | 错误描述   |
| <b>返回值：</b><br>列式结果集（L1数据见 <a href="#">4.10入参fields支持的字段</a> ，L2数据见 <a href="#">4.11入参fields支持的字段</a> ）<br>错误码，可对照错误码信息表（ <a href="#">附录2</a> ）查看错误信息   |                         |        |  |
| <b>示例：</b><br>取平安银行、万科A、浦发银行、中国石化的证券代码、交易所、证券简称、交易发生时间、今开盘价、卖一价、卖一量、买一价和买一量的实时行情快照数据，代码如下： <pre>ColumnSeq cols; StringSeq fields; StringSeq securities; SecurityIDSeq securityIDS;  securities.push_back("000001.SZSE"); securities.push_back("000002.SZSE"); securities.push_back("600000.SSE"); securities.push_back("600028.SSE");  fields.push_back("Symbol"); fields.push_back("Market"); fields.push_back("ShortName"); fields.push_back("TradingTime"); fields.push_back("OP"); fields.push_back("S1"); fields.push_back("SV1"); fields.push_back("B1"); fields.push_back("BV1");  if(QR_Success == GetSnapData(cols, errMsg, securities, securityIDS, fields)) {     cout &lt;&lt; "[i] GetSnapData Success!" &lt;&lt; endl;     ofile &lt;&lt; "[i] GetSnapData Success!" &lt;&lt; endl;     WriteData(cols, ofile); }</pre> |                         |        |  |

```

}
else
{
    cout << "[x] GetSnapData Failed: " << errMsg << endl;
    ofile << "[x] GetSnapData Failed: " << errMsg << endl;
}

```

### 3.3.4 提取因子数据类：

#### 3.3.4.1 GetFinance：取财务因子数据

```

int GetFinance(_O_ ColumnSeq& cols, _O_ string& errorMsg,
               StringSeq& securities, SecurityIDSeq& securityIDs,
               StringSeq& fields, string dateBegin, string dateEnd,
               EReportType rptType, ETrailType trailType,
               ERptDateType dateType,
               SortDefineSeq& sortDefs = SortDefineSeq(),
               int timezone = 8, string reserve = "");

```

取日频财务因子数据（仅适用于股票）；支持按证券品种、交易所、板块定义提取数据

| 参数                          | 定义                                     | 入参/出参  | 描述   |
|-----------------------------|--|--------|--|
| securities                  | StringSeq                              | 入参（必填） | 证券代码输入格式：“代码.交易所”；交易所输入格式：“*.交易所”；板块输入格式：“板块ID.*”  |
| <a href="#">securityIDs</a> | SecurityIDSeq                          | 入参（必填） | 数字型证券标的列表，支持多证券同时查询，与securities二者只用其一  |
| fields                      | StringSeq                              | 入参（必填） | 数据字段，一个"*"表示提取全部字段；如需提取指定字段，需输入相应的字段名，支持多字段同时提取，可选择输入的字段名见 <a href="#">4.12入参fields支持的字段</a> |
| dateBegin                   | string                                 | 入参（必填） | 起始报告日期，YYYY-MM_DD，[begin, end]   |
| dateEnd                     | string                                 | 入参（必填） | 截止报告日期，YYYY-MM_DD  |
| reportType                  | EReportType（参见 <a href="#">附录1.1</a> ） | 入参（必填） | 报表类型，包括合并本期（MergeCur）、合并上期（MergePre）、母公司本期（ParentCur）和母公司上期（ParentPre）                       |
| trailType                   | ETrailType（参见 <a href="#">附录1.1</a> ）  | 入参（必填） | 报表累计类型，包括单季度（TrailSeason）、季度累计（TrailAddup）和TTM12个月滚动累计（Trail12Mon）                           |
| timezone                    | int                                    | 入参（选填） | 参数中时间参数的时区，默认为东8区  |

|          |  |         |  |
|----------|--|---------|--|
| dateType | ERptDateType (参见 <a href="#">附录1.1</a> ) | 入参 (必填) | 报告日期类型, 包括报告发布日期 (ERptDateIssue) 和报告财期截止日期 (ERptDateClose) |
| sortDefs | ortDefineSeq                             | 入参 (选填) | 排序要求, 基于数据名指明排序要求  |
| reserve  | string                                   | 入参 (选填) | 预留, 默认为空   |
| cols     | ColumnSetSeq                             | 出参      | 列式结果集  |
| errorMsg | string                                   | 出参      | 错误描述   |

**返回值:**

列式结果集 (见[4.12出参返回的列式结果集字段](#))

错误码, 可对照错误码信息表 ([附录2](#)) 查看错误信息

**示例:**

取股票平安银行、万科A、浦发银行和中国石化2017-01-01到2017-04-01时间段内BEPS、TOTOR和TOLPRO的财务因子数据 (日频), 代码如下:

```
ColumnSeq cols;
StringSeq fields;
StringSeq securities;
SecurityIDSeq securityIDS;

securities.push_back("000001.SZSE");
securities.push_back("000002.SZSE");
securities.push_back("600000.SSE");
securities.push_back("600028.SSE");

fields.push_back("Symbol");
fields.push_back("Market");
fields.push_back("BEPS");
fields.push_back("TOTOR");
fields.push_back("TOLPRO");

if(QR_Success == GetFinance(cols, errMsg, securities, securityIDS,
    fields, "2017-01-01", "2017-04-01", k_RptMergeCur, k_TrailSeason, k_RptDateIssue))
{
    cout << "[i] GetFinance Success!" << endl;
    ofile << "[i] GetFinance Success!" << endl;
    WriteData(cols, ofile);
}
else
{
    cout << "[x] GetFinance Failed: " << errMsg << endl;
    ofile << "[x] GetFinance Failed: " << errMsg << endl;
}
```

### 3.3.4.2 GetFactor: 取风控及量化因子数据

| <pre>int GetFactor(_O_ ColumnSeq&amp; cols, _O_ string&amp; errorMsg,               StringSeq&amp; securities, SecurityIDSeq&amp; securityIDS,               StringSeq&amp; fields, string dateBegin, string dateEnd,               SortDefineSeq&amp; sortDefs = SortDefineSeq(), int timezone = 8,               string reserve = "");</pre>       |                |        |  |
|--|----------------|--------|--|
| 取日频量化及风控因子数据（仅适用于股票）；支持按证券品种、交易所、板块定义提取数据  |                |        |  |
| 参数   | 定义             | 入参/出参  | 描述   |
| securities   | StringSeq      | 入参（必填） | 证券代码输入格式:"代码.交易所";<br>交易所输入格式:"*.交易所";板块<br>输入格式:"板块ID.*"  |
| <a href="#">securityIDS</a>  | SecurityIDSeq  | 入参（必填） | 数字型证券标的列表, 支持多证券<br>同时查询, 与securities二者只用其<br>一  |
| fields   | StringSeq      | 入参（必填） | 数据字段, 一个"*"表示提取全部<br>字段; 如需提取指定字段, 需输入<br>相应的字段名, 支持多字段同时提<br>取, 可选择输入的字段名见 <a href="#">4.13入<br/>参fields支持的字段</a> |
| dateBegin  | string         | 入参（必填） | 起始报告日期, YYYY-MM_DD ,<br>[begin, end]   |
| dateEnd  | string         | 入参（必填） | 截止报告日期, YYYY-MM_DD   |
| timezone   | int            | 入参（选填） | 参数中时间参数的时区, 默认为东<br>8区   |
| sortDefs   | SortDefineSeq& | 入参（选填） | 排序要求, 基于数据名指明排序要<br>求  |
| reserve  | string         | 入参（选填） | 预留, 默认为空   |
| cols   | ColumnSetSeq   | 出参     | 列式结果集  |
| errorMsg   | string         | 出参     | 错误描述   |
| <b>返回值:</b><br>列式结果集（见 <a href="#">4.13出参返回的列式结果集字段</a> ）<br>错误码, 可对照错误码信息表（ <a href="#">附录2</a> ）查看错误信息   |                |        |  |
| <b>示例:</b><br>取股票平安银行、万科A、浦发银行和中国石化2017-01-01到2017-05-05时间段内的交易日期、每股净<br>资产和市盈率的风控及量化因子数据（日频），代码如下： <pre>ColumnSeq cols; StringSeq fields; StringSeq securities; SecurityIDSeq securityIDS;  securities.push_back("000001.SZSE"); securities.push_back("000002.SZSE"); securities.push_back("600000.SSE"); securities.push_back("600028.SSE");</pre> |                |        |  |

```
fields.push_back("Symbol");
fields.push_back("Market");
fields.push_back("TradingDate");
fields.push_back("QF_NetAssetPS");
fields.push_back("QF_PE");

if(QR_Success == GetFactor(cols, errMsg, securities,
    securityIDS, fields, "2017-01-01", "2017-05-05"))
{
    cout << "[i] GetFactor Success!" << endl;
    ofile << "[i] GetFactor Success!" << endl;
    WriteData(cols, ofile);
}
else
{
    cout << "[x] GetFactor Failed: " << errMsg << endl;
    ofile << "[x] GetFactor Failed: " << errMsg << endl;
}
```

3.3.5 提取其他数据类：

3.3.5.1 GetHisMarketInfo：取证券历史变动信息

| <pre>int GetHisMarketInfo(_O_ ColumnSeq&amp; cols, _O_ string&amp; errorMsg,     StringSeq&amp; securities, SecurityIDSeq&amp; securityIDS,     StringSeq&amp; fields,string dateBegin, string dateEnd,     SortDefineSeq&amp; sortDefs = SortDefineSeq(),     int timezone = 8, string reserve = "");</pre> |               |        |  |
|--|---------------|--------|--|
| 取日频证券历史变动信息（分红派息停复牌等）；支持按证券品种、交易所、板块定义提取数据   |               |        |  |
| 参数   | 定义            | 入参/出参  | 描述   |
| securities   | StringSeq     | 入参（必填） | 证券代码输入格式:"代码.交易所"<br>（暂不支持提取期货主力和次主力合约的数据）;交易所输入格式:"*.交易所";板块输入格式:"板块ID.*"     |
| <a href="#">securityIDs</a>  | SecurityIDSeq | 入参（必填） | 数字型证券标的列表，支持多证券同时查询，与securities二者只用其一  |
| fields   | StringSeq     | 入参（必填） | 数据字段，如需提取指定字段，需输入相应的字段名，支持多字段同时提取，可选择输入的字段名见 <a href="#">4.14入参fields支持的字段</a> |
| dataBegin  | string        | 入参（必填） | 起始日期，YYYY-MM_DD  |

|           |               |        |                       |
|-----------|---------------|--------|-----------------------|
| dateEnd   | string        | 入参（必填） | 截止日期，YYYY-MM_DD       |
| sortDefns | SortDefineSeq | 入参（选填） | 排序要求，基于数据名指明排序要求      |
| timezone  | int           | 入参（选填） | 参数中时间参数的时区，不配置时表示当地时区 |
| reserve   | string        | 入参（选填） | 预留参数，默认为空             |
| cols      | ColumnSeq     | 出参     | 列式结果集                 |
| errorMsg  | string        | 出参     | 错误描述                  |

**返回值：**

列式结果集（见[4.14出参返回的列式结果集字段](#)）

错误码，可对照错误码信息表（[附录2](#)）查看错误信息

**示例：**

取股票平安银行、万科A、浦发银行和中国石化2010-10-03到2017-03-24时间段内的除权除息日、税前每股分红和送股比例的证券历史变动信息数据，代码如下：

```
ColumnSeq cols;
StringSeq fields;
StringSeq securities;
SecurityIDSeq securityIDS;

securities.push_back("000001.SZSE");
securities.push_back("000002.SZSE");
securities.push_back("600000.SSE");
securities.push_back("600028.SSE");

fields.push_back("Symbol");
fields.push_back("Market");
fields.push_back("ExDividendDate");
fields.push_back("DividentBT");
fields.push_back("BonusRatio");

if(QR_Success == GetHisMarketInfo(cols, errMsg, securities,
    securityIDS, fields, "2010-10-03", "2017-03-24"))
{
    cout << "[i] GetHisMarketInfo Success!" << endl;
    ofile << "[i] GetHisMarketInfo Success!" << endl;
    WriteData(cols, ofile);
}
else
{
    cout << "[x] GetHisMarketInfo Failed: " << errMsg << endl;
    ofile << "[x] GetHisMarketInfo Failed: " << errMsg << endl;
}
```

### 3.3.5.2 GetJointContracts: 取期货主力或连续合约信息

```
int GetJointContracts(_O_ ColumnSeq& cols, _O_ string& errorMsg,
                     StringSeq& contracts, string dateBegin,
                     string dateEnd, int timezone = 8, string reserve = "");
```

取期货主力连续或连续合约信息，按连续合约代码和日期排序

| 参数        | 定义        | 入参/出参  | 描述  |
|-----------|-----------|--------|---|
| contracts | StringSeq | 入参（必填） | 主力连续或连续合约代码，按:交易品种 +M+01/02, 交易品种 +CC+01~04 |
| dateBegin | string    | 入参（必填） | 起始日期，YYYY-MM_DD                             |
| dateEnd   | string    | 入参（必填） | 截止日期，YYYY-MM_DD                             |
| timezone  | int       | 入参（选填） | 参数中时间参数的时区，默认为东8区                           |
| reserve   | string    | 入参（选填） | 预留参数，默认为空                                   |
| cols      | ColumnSeq | 出参     | 列式结果集                                       |
| errorMsg  | string    | 出参     | 错误描述  |

**返回值:**

列式结果集（见[4.15出参返回的列式结果集字段](#)）

错误码，可对照错误码信息表（[附录2](#)）查看错误信息

**示例:**

取2017-01-01到2017-04-30时间段内豆一主力连续合约和豆一次主力合约信息数据，代码如下：

```
ColumnSeq cols;
```

```
StringSeq contracts;
```

```
contracts.push_back("ACC01");
```

```
contracts.push_back("AM02");
```

```
if(QR_Success == GetJointContracts(cols, errMsg, contracts, "2017-01-01", "2017-04-30"))
{
    cout << "[i] GetJointContracts Success!" << endl;
    ofile << "[i] GetJointContracts Success!" << endl;
    WriteData(cols, ofile);
}
else
{
    cout << "[x] GetJointContracts Failed: " << errMsg << endl;
    ofile << "[x] GetJointContracts Failed: " << errMsg << endl;
}
```

### 3.3.6 行情订阅及回调类:

#### 3.3.6.1 Subscribe: 订阅行情

| <pre>int Subscribe(_O_string&amp; errorMsg, _I_EQuoteData quoteData, _I_uint freq,              _I_StringSeq&amp; securities, StringSeq&amp; fields = StringSeq(),              bool asPossible = true);</pre> |  |         |                      |
|--|--|---------|----------------------|
| 订阅行情   |  |         |                      |
| 参数   | 定义                                     | 入参/出参   | 描述                   |
| quoteData  | EQuoteData (参见 <a href="#">附录1.1</a> ) | 入参 (必填) | 订阅行情类型               |
| freq   | unit                                   | 入参 (必填) | 订阅行情频率, 单位为秒, 0为实时行情 |
| securities   | StringSeq                              | 入参 (必填) | 订阅标的                 |
| fields   | StringSeq                              | 入参 (选填) | 订阅字段, 暂未使用           |
| asPossible   | bool                                   | 入参 (选填) | 尽最大可能完成订阅            |
| errorMsg   | string                                 | 出参      | 错误描述                 |
| <b>返回值:</b><br>错误码, 可对照错误码信息表 ( <a href="#">附录2</a> ) 查看错误信息   |  |         |                      |

#### 3.3.6.2 UnSubscribe: 退订行情

| <pre>int UnSubscribe(_O_string&amp; errorMsg, _I_EQuoteData quoteData,                _I_uint freq, _I_StringSeq&amp; securities,                StringSeq&amp; fields = StringSeq(), bool asPossible = true);</pre> |  |         |                      |
|--|--|---------|----------------------|
| 退订行情   |  |         |                      |
| 参数   | 定义                                     | 入参/出参   | 描述                   |
| quoteData  | EQuoteData (参见 <a href="#">附录1.1</a> ) | 入参 (必填) | 订阅行情类型               |
| freq   | unit                                   | 入参 (必填) | 订阅行情频率, 单位为秒, 0为实时行情 |
| securities   | StringSeq                              | 入参 (必填) | 订阅标的                 |
| fields   | StringSeq                              | 入参 (选填) | 订阅字段, 暂未使用           |
| asPossible   | bool                                   | 入参 (选填) | 尽最大可能完成订阅            |
| errorMsg   | string                                 | 出参      | 错误描述                 |
| <b>返回值:</b><br>错误码, 可对照错误码信息表 ( <a href="#">附录2</a> ) 查看错误信息   |  |         |                      |

#### 3.3.6.3 RegSubStsCB: 注册实时分时数据回调方法

|   |  |
|---|--|
| <pre>int RegSubStsCB(_I_SubStsCB cbFunction);</pre> |  |
| 注册实时分时数据回调方法  |  |



| 参数         | 定义       | 入参/出参  | 描述     |
|------------|----------|--------|--------|
| cbFunction | SubStsCB | 入参（必填） | 接收数据函数 |

**示例：**

```

void OnSubSts(QSubSts& data)
{
    cout<< "Sts: " << CurrentLocalTime()
        << ": Symbol = " << data.Symbol
        << ", TradingTime = " << data.TradingTime
        << ", TradingDate = " << data.TradingDate
        << ", UNIX = " << data.UNIX
        << ", Market = " << data.Market
        << ", OP = " << data.OP
        << ", HIP = " << data.HIP
        << ", LOP = " << data.LOP
        << ", CP = " << data.CP
        << ", CQ = " << data.CQ
        << ", CM = " << data.CM
        << ", Change = " << data.Change
        << ", ChangeRatio = " << data.ChangeRatio
        << ", TQ = " << data.TQ
        << ", VWAP = " << data.Vwap
        << ", OBPD = " << data.OBPD
        << ", ClvLOP = " << data.ClvLOP
        << ", ClvHIP = " << data.ClvHIP
        << ", ClvVwap = " << data.ClvVwap
        << ", ShortName = " << data.ShortName
        << ", Position = " << data.Position
        << ", ConSign = " << data.ConSign
        << ", Rtn = " << data.Rtn
        << ", ConSignName = " << data.ConSignName
        << ", Varieties = " << data.Varieties
        << ", MarginUnit = " << data.MarginUnit
        << ", MaintainingMargin = " << data.MaintainingMargin;

    cout << ", SellPrice = ";
    for(unsigned int i=0;i<data.SellPrice.size();i++)
    {
        cout << data.SellPrice[i] << ", ";
    }
    cout << ", BuyPrice = ";
    for(unsigned int i=0;i<data.BuyPrice.size();i++)
    {
        cout << data.BuyPrice[i]<< ", " ;
    }
}

```

```
cout << ", BuyVolume = ";
for(unsigned int i=0;i<data.BuyVolume.size();i++)
{
    cout << data.BuyVolume[i] <<",";
}
cout << ", SellVolume = ";
for(unsigned int i=0;i<data.SellVolume.size();i++)
{
    cout << data.SellVolume[i] <<",";
}
cout<<endl;
}
```

3.3.6.4 RegSubQuoteCB: 注册实时 Tick 数据回调方法

```
int RegSubQuoteCB( I SubQuoteCB cbFunction);
```

注册实时Tick数据回调方法

| 参数         | 定义         | 入参/出参  | 描述     |
|------------|------------|--------|--------|
| cbFunction | SubQuoteCB | 入参（必填） | 接收数据函数 |

示例:

```
void OnSubQuote(QtSubQuote& data)
{
    cout<< "L1Quote: " << CurrentLocalTime()
    << ": Symbol = " << data.Symbol
    << ", TradingTime = " << data.TradingTime
    << ", TradingDate = " << data.TradingDate
    << ", UNIX = " << data.UNIX
    << ", Market = " << data.Market
    << ", LCP = " << data.LCP
    << ", OP = " << data.OP
    << ", HIP = " << data.HIP
    << ", LOP = " << data.LOP
    << ", CP = " << data.CP
    << ", CP1 = " << data.CP1
    << ", TQ = " << data.TQ
    << ", TM = " << data.TM;
    cout << ", SellPrice = ";
    for(unsigned int i=0;i<data.SellPrice.size();i++)
    {
        cout << data.SellPrice[i] << ",";
    }
    cout << ", BuyPrice = ";
    for(unsigned int i=0;i<data.BuyPrice.size();i++)
    {
```

```

    cout << data.BuyPrice[i]<< ", " ;
}
cout << ", BuyVolume = ";
for(unsigned int i=0;i<data.BuyVolume.size();i++)
{
    cout << data.BuyVolume[i] << ", ";
}
cout << ", SellVolume = ";
for(unsigned int i=0;i<data.SellVolume.size();i++)
{
    cout << data.SellVolume[i] << ", ";
}
cout << ", BS = "<<data.BS
    << ", TQ = " << data.TQ
    << ", ChangeRatio = " << data.ChangeRatio
    << ", CQ = " << data.CQ
    << ", CM = " << data.CM
    << ", Vwap = " << data.Vwap
    << ", OrderRate = " << data.OrderRate
    << ", OrderDiff = " << data.OrderDiff
    << ", Amplitude = " << data.Amplitude
    << ", VolRate = " << data.VolRate
    << ", SellVOL = " << data.SellVOL
    << ", BuyVOL = " << data.BuyVOL
    << ", ShortName = " << data.ShortName
    << ", LSP = " << data.LSP
    << ", SP = " << data.SP
    << ", PrePosition = " << data.PrePosition
    << ", Position = " << data.Position
    << ", Change = " << data.Change
    << ", OC = " << data.OC
    << ", LocalTimeStamp = " << data.LocalTimeStamp
    << ", PositionChange = " << data.PositionChange
    << ", PrePositionChange = " << data.PrePositionChange
    << ", LimitUp = " << data.LimitUp
    << ", LimitDown = " << data.LimitDown
    << ", PreDelta = " << data.PreDelta
    << ", Delta = " << data.Delta
    << ", ConSign = " << data.ConSign
    << ", Gamma = " << data.Gamma
    << ", Rho = " << data.Rho
    << ", Theta = " << data.Theta
    << ", Vega = " << data.Vega
    << ", SettleGroupID = " << data.SettleGroupID

```

```

    << ", SettleID = " << data.SettleID
    << ", ConSignName = " << data.ConSignName
    << ", LastVol = " << data.LastVol
    << ", LifeLow = " << data.LifeLow
    << ", LifeHigh = " << data.LifeHigh
    << ", avgprice1 = " << data.avgprice1
    << ", BidImPLYQty = " << data.BidImPLYQty
    << ", AskImPLYQty = " << data.AskImPLYQty
    << ", Varieties = " << data.Varieties
    << ", DelSP = " << data.DelSP
    << ", PriceChange01 = " << data.PriceChange01
    << ", Change1 = " << data.Change1
    << ", BuyRatio = " << data.BuyRatio
    << ", SPD = " << data.SPD
    << ", RPD = " << data.RPD
    << ", Depth01 = " << data.Depth01
    << ", Depth02 = " << data.Depth02
    << ", TurnoverRate = " << data.TurnoverRate
    << ", CMV = " << data.CMV
    << ", TMV = " << data.TMV
    << ", PE = " << data.PE
    << ", PB = " << data.PB
    << ", Premium = " << data.Premium
    << ", PremiumRate = " << data.PremiumRate
    << ", StatusID = " << data.StatusID
    << ", AuctionPrice = " << data.AuctionPrice
    << ", AuctionVolume = " << data.AuctionVolume
    << ", BuyLevelNO = " << data.BuyLevelNO
    << ", SellLevelNO = " << data.SellLevelNO
    << ", SymbolSource = " << data.SymbolSource
    << ", NAV = " << data.NAV
    << ", IOPV = " << data.IOPV
    << ", SampleNo = " << data.SampleNo
    << ", TT = " << data.TT
    << ", CT = " << data.CT
    << ", PE1 = " << data.PE1
    << ", PE2 = " << data.PE2
    << endl;
}

```

### 3.3.6.5 RegSubL2QuoteCB: 注册实时 L2 十档行情回调方法

```
int RegSubL2QuoteCB( I_SubL2QuoteCB cbFunction);
```

注册实时L2十档行情数据回调方法（上交所、深交所）

| 参数         | 定义           | 入参/出参  | 描述     |
|------------|--------------|--------|--------|
| cbFunction | SubL2QuoteCB | 入参（必填） | 接收数据函数 |

**示例：**

```

void OnSubL2Quote (QtSubL2Quote& data)
{
    cout<< "L2Quote: " << CurrentLocalTime()
        << ": Symbol = " << data.Symbol
        << ", TradingTime = " << data.TradingTime
        << ", TradingDate = " << data.TradingDate
        << ", UNIX = " << data.UNIX
        << ", Market = " << data.Market
        << ", LCP = " << data.LCP
        << ", OP = " << data.OP
        << ", HIP = " << data.HIP
        << ", LOP = " << data.LOP
        << ", CP = " << data.CP
        << ", CP1 = " << data.CP1
        << ", TT = " << data.TT
        << ", TQ = " << data.TQ
        << ", TM = " << data.TM
        << ", BuyLevelNo = " << data.BuyLevelNo;
    cout << ", BuyPrice = ";
    for(int i=0;i<data.BuyPrice.size();i++)
    {
        cout << data.BuyPrice[i]<< ", " ;
    }
    cout << ", BuyVolume = ";
    for(int i=0;i<data.BuyVolume.size();i++)
    {
        cout << data.BuyVolume[i] << ", " ;
    }
    cout << ", TotalBuyOrderNo = ";
    for(int i=0;i<data.TotalBuyOrderNo.size();i++)
    {
        cout << data.TotalBuyOrderNo[i] << ", " ;
    }

    cout << ", TBOVOL = "<<data.TBOVOL
        << ", SellLevelNo = " << data.SellLevelNo;

    cout << ", SellPrice = ";
    for(int i=0;i<data.SellPrice.size();i++)
    {
        cout << data.SellPrice[i]<< ", " ;
    }
}

```

```

}
cout << ", SellVolume = ";
for(int i=0;i<data.SellVolume.size();i++)
{
    cout << data.SellVolume[i] <<",";
}
cout << ", TotalSellOrderNo = ";
for(int i=0;i<data.TotalSellOrderNo.size();i++)
{
    cout << data.TotalSellOrderNo[i] <<",";
}
cout << ", TSOVOL = "<<data.TSOVOL
    << ", WAvGBP = " << data.WAvGBP
    << ", WAvSP = " << data.WAvSP
    << ", IOPV = " << data.IOPV
    << ", BondWAvBP = " << data.BondWAvBP
    << ", YTM = " << data.YTM
    << ", ETFBNo = " << data.ETFBNo
    << ", ETFBVOL = " << data.ETFBVOL
    << ", ETFBM = " << data.ETFBM
    << ", ETFSNo = " << data.ETFSNo
    << ", ETFSVOL = " << data.ETFSVOL
    << ", ETFSM = " << data.ETFSM
    << ", WithdrawBNo = " << data.WithdrawBNo
    << ", WithdrawBVOL = " << data.WithdrawBVOL
    << ", WithdrawBM = " << data.WithdrawBM
    << ", WithdrawSNo = " << data.WithdrawSNo
    << ", WithdrawSVOL = " << data.WithdrawSVOL
    << ", WithdrawSM = " << data.WithdrawSM
    << ", TotalBNo = " << data.TotalBNo
    << ", TotalSNo = " << data.TotalSNo
    << ", MaxBD = " << data.MaxBD
    << ", MaxSD = " << data.MaxSD
    << ", BOno = " << data.BOno
    << ", SOno = " << data.SOno
    << ", BondWAvBP = " << data.BondWAvBP
    << ", TradeStatus = " << data.TradeStatus
    << ", TradeNo = " << data.TradeNo
    << ", TradeVolume = " << data.TradeVolume
    << ", TradeAmount = " << data.TradeAmount
    << ", Status = " << data.Status
    << ", PE1 = " << data.PE1
    << ", NAV = " << data.NAV
    << ", PE2 = " << data.PE2

```

```

<< ", PremiumRate = " << data.PremiumRate
<< ", LimitUp = " << data.LimitUp
<< ", LimitDown = " << data.LimitDown
<< ", PriceUpdown1 = " << data.PriceUpdown1
<< ", PriceUpdown2 = " << data.PriceUpdown2
<< ", SymbolSource = " << data.SymbolSource
<< ", ShortName = " << data.ShortName
<< ", AvgPrice = " << data.AvgPrice
<< ", SettlePrice = " << data.SettlePrice
<< ", PreSettlePrice = " << data.PreSettlePrice
<< ", LastVolume = " << data.LastVolume
<< ", PreTotalPosition = " << data.PreTotalPosition
<< ", TotalPosition = " << data.TotalPosition
<< ", PositionChange = " << data.PositionChange
<< ", CumulativeHighPrice = " << data.CumulativeHighPrice
<< ", CumulativeLowPrice = " << data.CumulativeLowPrice
<< ", Nolevel = " << data.Nolevel
<< ", Side = " << data.Side;

cout << ", BidImPLYQty = ";
for(int i=0;i<data.BidImPLYQty.size();i++)
{
    cout << data.BidImPLYQty[i]<< ", " ;
}
cout << ", AskImPLYQty = ";
for(int i=0;i<data.AskImPLYQty.size();i++)
{
    cout << data.AskImPLYQty[i] <<" ";
}
cout << ", BestBuyQty = ";
for(int i=0;i<data.BestBuyQty.size();i++)
{
    cout << data.BestBuyQty[i] << ", ";
}
cout << ", BestSellQty = ";
for(int i=0;i<data.BestSellQty.size();i++)
{
    cout << data.BestSellQty[i] << ", ";
}
cout << ", ContinueSign = " << data.ContinueSign
<< ", Theta = " << data.Theta
<< ", Vega = " << data.Vega
<< ", Gamma = " << data.Gamma
<< ", Rho = " << data.Rho

```

```

    << ", Delta = " << data.Delta
    << endl;
}

```

### 3.3.6.6 RegSubArbitrageCB: 注册 L1 套利行情数据回调方法

```
int RegSubArbitrageCB(_I SubArbitrageCB cbFunction);
```

注册 L1 套利行情数据回调方法（大商所）

| 参数         | 定义             | 入参/出参  | 描述     |
|------------|----------------|--------|--------|
| cbFunction | SubArbitrageCB | 入参（必填） | 接收数据函数 |

### 3.3.6.7 RegSubStaticCB: 注册静态数据回调方法

```
int RegSubStaticCB(_I SubStaticCB cbFunction);
```

注册静态数据回调方法（上交所 L2、深交所 L2、上交所个股期权）

| 参数         | 定义          | 入参/出参  | 描述     |
|------------|-------------|--------|--------|
| cbFunction | SubStaticCB | 入参（必填） | 接收数据函数 |

### 3.3.6.8 RegSubBulletinCB: 注册公告及增值信息数据回调方法

```
int RegSubBulletinCB(_I SubBulletinCB cbFunction);
```

注册公告及增值信息数据回调方法（深交所 L1）

| 参数         | 定义            | 入参/出参  | 描述     |
|------------|---------------|--------|--------|
| cbFunction | SubBulletinCB | 入参（必填） | 接收数据函数 |

### 3.3.6.9 RegSubL2AuctionCB: 注册 L2 集合竞价数据回调方法

```
int RegSubL2AuctionCB(_I SubL2AuctionCB cbFunction);
```

注册集合竞价数据回调方法（上交所 L2）

| 参数         | 定义             | 入参/出参  | 描述     |
|------------|----------------|--------|--------|
| cbFunction | SubL2AuctionCB | 入参（必填） | 接收数据函数 |

### 3.3.6.10 RegSubL2IndexCB: 注册 L2 指数行情数据回调方法

```
int RegSubL2IndexCB(_I SubL2IndexCB cbFunction);
```

注册 L2 指数行情数据回调方法（上交所、深交所）

| 参数         | 定义           | 入参/出参  | 描述     |
|------------|--------------|--------|--------|
| cbFunction | SubL2IndexCB | 入参（必填） | 接收数据函数 |



### 3.3.6.11 RegSubL2OrderCB: 注册 L2 逐笔委托数据回调方法

| <b>int RegSubL2OrderCB( I SubL2OrderCB cbFunction);</b> |              |        |        |
|---|--------------|--------|--------|
| 注册L2逐笔委托数据回调方法（深交所）                                     |              |        |        |
| 参数  | 定义           | 入参/出参  | 描述     |
| cbFunction  | SubL2OrderCB | 入参（必填） | 接收数据函数 |

### 3.3.6.12 RegSubL2OrderQueueCB: 注册 L2 委托队列数据回调方法

| <b>int RegSubL2OrderQueueCB( I SubL2OrderQueueCB cbFunction);</b> |                   |        |        |
|---|-------------------|--------|--------|
| 注册L2委托队列数据回调方法（上交所、深交所）   |                   |        |        |
| 参数  | 定义                | 入参/出参  | 描述     |
| cbFunction  | SubL2OrderQueueCB | 入参（必填） | 接收数据函数 |

### 3.3.6.13 RegSubL2TransactionCB: 注册 L2 逐笔成交数据回调方法

| <b>int RegSubL2TransactionCB( I SubL2TransactionCB cbFunction);</b> |                    |        |        |
|---|--------------------|--------|--------|
| 注册L2逐笔成交数据回调方法（上交所、深交所）   |                    |        |        |
| 参数  | 定义                 | 入参/出参  | 描述     |
| cbFunction  | SubL2TransactionCB | 入参（必填） | 接收数据函数 |

### 3.3.6.14 RegSubL2StatusCB: 注册 L2 证券状态数据回调方法

| <b>int RegSubL2StatusCB( I SubL2StatusCB cbFunction);</b> |               |        |        |
|---|---------------|--------|--------|
| 注册L2证券状态数据回调方法（深交所）                                       |               |        |        |
| 参数  | 定义            | 入参/出参  | 描述     |
| cbFunction  | SubL2StatusCB | 入参（必填） | 接收数据函数 |

### 3.2.6.15 RegSubL2ArbitrageCB: 注册 L2 套利行情数据回调方法

| <b>int RegSubL2ArbitrageCB( I SubL2ArbitrageCB cbFunction);</b> |                  |        |        |
|---|------------------|--------|--------|
| 注册L2套利行情数据回调方法（大商所）   |                  |        |        |
| 参数  | 定义               | 入参/出参  | 描述     |
| cbFunction  | SubL2ArbitrageCB | 入参（必填） | 接收数据函数 |

### 3.3.6.16 RegSubL2MarchPriceQtyCB: 注册 L2 分价成交量数据回调方法

| <b>int RegSubL2MarchPriceQtyCB(_I_ SubL2MarchPriceQtyCB cbFunction);</b> |                      |        |        |
|--|----------------------|--------|--------|
| 注册L2分价成交量数据回调方法  |                      |        |        |
| 参数   | 定义                   | 入参/出参  | 描述     |
| cbFunction   | SubL2MarchPriceQtyCB | 入参（必填） | 接收数据函数 |

### 3.3.6.17 RegSubL2OrderStatisticCB: 注册 L2 委托统计数据回调方法

| <b>int RegSubL2OrderStatisticCB(_I_ SubL2OrderStatisticCB cbFunction);</b> |                       |        |        |
|--|-----------------------|--------|--------|
| 注册L2委托统计数据回调方法（大商所）  |                       |        |        |
| 参数   | 定义                    | 入参/出参  | 描述     |
| cbFunction   | SubL2OrderStatisticCB | 入参（必填） | 接收数据函数 |

### 3.3.6.18 RegSubL2SettlePriceCB: 注册 L2 实时结算数据回调方法

| <b>int RegSubL2SettlePriceCB(_I_ SubL2SettlePriceCB cbFunction);</b> |                    |        |        |
|--|--------------------|--------|--------|
| 注册L2实时结算数据回调方法（大商所）  |                    |        |        |
| 参数   | 定义                 | 入参/出参  | 描述     |
| cbFunction   | SubL2SettlePriceCB | 入参（必填） | 接收数据函数 |

## 3.3.7 工具类函数：

为实现数据结构无关性，红楹云数据接口返回的数据集为列式集或列式数据集集合，工具类函数可辅助处理返回的数据集形式。

**备注：QTPAI 列式数据集中的数据按列存放，以下分情况说明工具类函数应用场景：**

- 1、每列为同一种数据类型，在列式结构中有数据类型字段和列数据名等信息，根据每列指定的数据类型进行处理和使用；
- 2、各列的数据个数相同；
- 3、为保证各列数据个数相同，对于不存在的数据以具体数据类型对应的最大值填充，用户必须使用 Null2Val 将填充值转换为指定值；
- 4、对于数值数据列，建议用户按列做为数值数组直接使用，可使用模板方法 GetDataArrayT；
- 5、对于字符串数据列，可使用 GetStrArray 转换为 string 数组使用；
- 6、直接从列式数据集中提取指定横、纵坐标的数值数据，可使用 GetDataByIndex；
- 7、直接从列式数据集中提取指定横、纵坐标的字符串数据，可使用 GetStrByIndex；
- 8、DateTime 数据实质为 uint64 表示的从 19700101 以来的毫秒数，可使用 DateTime2StrArray 将其统一转换为字符串数组；

9、DateTime 数据实质为 uint64 表示的从 19700101 以来的微秒数，可使用 TimeStamp2StrArray 将其统一转换为字符串数组。

### 3.3.7.1 GetTypeSize: 取数据类型字节长度

| uint GetTypeSize(uint type); |      |        |                   |
|------------------------------|------|--------|-------------------|
| 取数据类型的字节长度                   |      |        |                   |
| 参数                           | 定义   | 入参/出参  | 描述                |
| type                         | unit | 入参（必填） | 数据类型（见EDDataType） |
| 返回值：<br>返回数据类型的字节长度          |      |        |                   |

数据列输入输出相关方法：

### 3.3.7.2 PrintData(DataColumn,index): 屏幕输出数据列中单个数据

| void PrintData(_I_ DataColumn& col, _I_ uint colIdx); |            |        |             |
|---|------------|--------|-------------|
| 屏幕输出数据列中单个数据  |            |        |             |
| 参数  | 定义         | 入参/出参  | 描述          |
| col   | DataColumn | 入参（必填） | 列数据         |
| colIdx  | unit       | 入参（必填） | 下标，以0为下标起始值 |
| 返回值：<br>打印输出单个数据                                      |            |        |             |

### 3.3.7.3 PrintData(ColumnSeq): 屏幕输出整个数据集

| void PrintData(_I_ ColumnSeq& cols); |           |        |       |
|--------------------------------------|-----------|--------|-------|
| 屏幕输出整个数据集                            |           |        |       |
| 参数                                   | 定义        | 入参/出参  | 描述    |
| cols                                 | ColumnSeq | 入参（必填） | 列式数据集 |
| 返回值：<br>打印输出整个数据集                    |           |        |       |

### 3.3.7.4 PrintData(ColumnSetSeq): 屏幕输出数据集序列

| void PrintData(_I_ ColumnSetSeq& colsSet); |              |        |         |
|--|--------------|--------|---------|
| 屏幕输出数据集序列                                  |              |        |         |
| 参数   | 定义           | 入参/出参  | 描述      |
| colsSet                                    | ColumnSetSeq | 入参（必填） | 列式数据集序列 |
| 返回值：<br>打印输出多个数据集序列                        |              |        |         |

### 3.3.7.5 WriteData(ColumnSeq): 输出数据集到文件

| <b>void WriteData(_I_ ColumnSeq&amp; cols, _I_ ofstream&amp; ofile, _I_ bool writeHeader = true);</b> |           |        |                     |
|---|-----------|--------|---------------------|
| 输出数据集到文件  |           |        |                     |
| 参数  | 定义        | 入参/出参  | 描述                  |
| cols  | ColumnSeq | 入参（必填） | 列式数据集               |
| ofile   | ofstream  | 入参（必填） | 文件名称                |
| writeHeader   | bool      | 入参（选填） | 打印输出每列的字段名，默认为 true |
| 返回值：<br>输出整个数据集到文件中   |           |        |                     |

### 3.3.7.6 WriteData(ColumnSetSeq): 输出数据集序列到文件

| <b>void WriteData(_I_ ColumnSetSeq&amp; colsSet, _I_ ofstream&amp; ofile);</b> |              |        |         |
|--|--------------|--------|---------|
| 输出数据集序列到文件   |              |        |         |
| 参数   | 定义           | 入参/出参  | 描述      |
| colsSet  | ColumnSetSeq | 入参（必填） | 列式数据集序列 |
| ofile  | ofstream     | 入参（必填） | 文件名     |
| 返回值：<br>打印输出多个数据集序列到文件中  |              |        |         |

解析数据列相关方法：

### 3.3.7.7 GetStrArray: 将字符串数据列转为字符串数组

| <b>bool GetStrArray(_O_ StringSeq&amp; strs, _I_ const DataColumn&amp; col);</b> |                  |        |        |
|--|------------------|--------|--------|
| 将字符串数据列转为字符串数组   |                  |        |        |
| 参数   | 定义               | 入参/出参  | 描述     |
| strs   | StringSeq        | 出参     | 字符串数组  |
| col  | const DataColumn | 入参（必填） | 字符串数据列 |
| 返回值：<br>字符串数组（仅支持字符串数据列）   |                  |        |        |

### 3.3.7.8 DateTime2StrArray: 将 DateTime 数值数据列转为字符串时间数组

| <b>bool DateTime2StrArray(_O_ StringSeq&amp; strs, _I_ const DataColumn&amp; col);</b> |                  |        |         |
|--|------------------|--------|---------|
| DateTime数据列转为字符串时间数组   |                  |        |         |
| 参数   | 定义               | 入参/出参  | 描述      |
| col  | const DataColumn | 入参（必填） | 时间数据列   |
| strs   | StringSeq        | 出参     | 字符串时间数组 |
| 返回值：<br>字符串时间数组  |                  |        |         |

### 3.3.7.9 TimeStamp2StrArray: 将 TimeStamp 数值数据列转为字符串时间数组

| <b>bool TimeStamp2StrArray(_O_ StringSeq&amp; strs, _I_ const DataColumn&amp; col);</b> |                  |        |         |
|---|------------------|--------|---------|
| TimeStamp数据列转为字符串时间数组   |                  |        |         |
| 参数  | 定义               | 入参/出参  | 描述      |
| col   | const DataColumn | 入参（必填） | 时间数据列   |
| strs  | StringSeq        | 出参     | 字符串时间数组 |
| 返回值：<br>字符串时间数组   |                  |        |         |

### 3.3.7.10 GetDataArrayT: 从数据值数据序列中取数值数组模板方法

| <b>T* GetDataArrayT(_I_ const DataColumn&amp; col)</b>  |                  |        |     |
|---|------------------|--------|-----|
| 取数值数组，不支持字符串数据列   |                  |        |     |
| 参数  | 定义               | 入参/出参  | 描述  |
| col   | const DataColumn | 入参（必填） | 列数据 |
| 模板方法应用示例：<br><pre>template&lt; typename T &gt; T* GetDataArrayT(_I_ const DataColumn&amp; col) {     if(k_Char == col.type) //字符串数据需特殊处理         return NULL;      return (T*)&amp;col.data[0]; }</pre> |                  |        |     |

### 3.3.7.11 GetDataT: 从数据值数据列中取单个数值模板方法

| T GetDataT(_I_ const DataColumn& col, _I_ uint colIdx)  |                  |        |                                |
|---|------------------|--------|--------------------------------|
| 按下标取某个数据，不支持字符串数据列  |                  |        |                                |
| 参数  | 定义               | 入参/出参  | 描述                             |
| col   | const DataColumn | 入参（必填） | 列数据                            |
| colIdx  | unit             | 入参（必填） | 下标，指提取数据列中的第几个值，范围为0~数据列中的数据个数 |
| 模板方法应用示例：<br><pre>template&lt; typename T &gt; T GetDataT(_I_ const DataColumn&amp; col, _I_ uint colIdx) {     if(k_Char == col.type) //字符串数据需特殊处理         return T();      return *(((T*)&amp;col.data[0]) + colIdx); }</pre> |                  |        |                                |

### 3.3.7.12 GetDataByName: 按数据名取数值数据集中数据

| bool GetDataByName(_O_ void* data, _I_ ColumnSeq& cols, _I_ uint rowIdx, _I_ string field); |           |        |             |
|---|-----------|--------|-------------|
| 按数据名取指定位置一个数据，data必须为规定的数据类型的地址位置，不支持字符串类型  |           |        |             |
| 参数  | 定义        | 入参/出参  | 描述          |
| cols  | ColumnSeq | 入参（必填） | 列式数据集       |
| rowIdx  | unit      | 入参（必填） | 行号，以0为下标起始值 |
| field   | string    | 入参（必填） | 数据列名        |
| data  | void      | 出参（必填） | 数据地址        |
| 返回值：<br>指定数值数据集中的数据   |           |        |             |

### 3.3.7.13 GetDataByIndex: 按行列坐标取数值数据集中数据

| bool GetDataByIndex(_O_ void* data, _I_ ColumnSeq& cols, _I_ uint rowIdx, _I_ uint colIdx); |           |        |             |
|---|-----------|--------|-------------|
| 按行、列坐标取一个数据，data必须为规定的数据类型的地址位置，不支持字符串类型  |           |        |             |
| 参数  | 定义        | 入参/出参  | 描述          |
| cols  | ColumnSeq | 入参（必填） | 列式数据集       |
| rowIdx  | unit      | 入参（必填） | 行号，以0为下标起始值 |
| colIdx  | unit      | 入参（必填） | 列号，以0为下标起始值 |
| data  | void      | 出参     | 数据地址        |
| 返回值：<br>指定数值数据集中的数据   |           |        |             |

### 3.3.7.14 GetStrByName: 按数据名取字符串数据集中数据

```
bool GetStrByName(_O_string& str, _I_ColumnSeq& cols, _I_uint rowIdx, _I_string field);
```

| 参数     | 定义        | 入参/出参  | 描述          |
|--------|-----------|--------|-------------|
| cols   | ColumnSeq | 入参（必填） | 列式数据集       |
| rowIdx | unit      | 入参（必填） | 行号，以0为下标起始值 |
| field  | string    | 入参（必填） | 数据列名        |
| str    | string    | 出参     | 字符串数据       |

返回值:

指定字符串数据集中的数据

### 3.3.7.15 GetStrByIndex: 按行列坐标取字符串数据集中数据

```
bool GetStrByIndex(_O_string& str, _I_ColumnSeq& cols, _I_uint rowIdx, _I_uint colIdx);
```

按行、列坐标取一个数据，支持字符串类型

| 参数     | 定义        | 入参/出参  | 描述          |
|--------|-----------|--------|-------------|
| cols   | ColumnSeq | 入参（必填） | 列式数据集       |
| rowIdx | unit      | 入参（必填） | 行号，以0为下标起始值 |
| colIdx | Unit      | 入参（必填） | 列号，以0为下标起始值 |
| str    | string    | 出参     | 字符串数据       |

返回值:

指定字符串数据集中的数据

### 3.3.7.16 Null2Val: 将数值数据列中的填充数据转为用户指定值

```
void Null2Val(_I_StringSeq& fields, _I_int val, _IO_ColumnSeq& cols);
```

将指定的字段名数据列中的NULL值转为用户指定值（不适用于字符串数据列）

| 参数     | 定义        | 入参/出参     | 描述                                  |
|--------|-----------|-----------|-------------------------------------|
| fields | StringSeq | 入参（必填）    | 指定的数据列字段名，若不指定，则对列式数据集中所有数值型数据列进行转换 |
| val    | Int       | 入参（必填）    | 用于替换NULL的指定值                        |
| cols   | ColumnSeq | 入参（必填）、出参 | 列式数据集                               |

返回值:

用户指定值

工具类 API 使用示例:

### 3.3.7.17 示例：以提取快照列式数据集为例

示例：

```
string errMsg;
ColumnSeq cols;
StringSeq fields;
StringSeq securities;
SecurityIDSeq securityIDS;

securities.push_back("000001.SZSE");
securities.push_back("000002.SZSE");
securities.push_back("600000.SSE");
securities.push_back("600028.SSE");

fields.push_back("Symbol");
fields.push_back("Market");
fields.push_back("ShortName");
fields.push_back("TradingTime");
fields.push_back("OP");
fields.push_back("S1");
fields.push_back("SV1");
fields.push_back("B1");
fields.push_back("BV1");

if(QR_Success != GetSnapData(cols, errMsg, securities, securityIDS, fields))
    return;
if(0 == cols.size())
    return;

ofstream ofile("ColsDemoOut.txt");

//GetStrArray示例
if(k_Char == cols[0].type)
{
    StringSeq strs;
    GetStrArray(strs, cols[0]);

    ofile << cols[0].name << endl;
    for(uint i = 0; i < strs.size(); ++ i)
        ofile << strs[i] << endl;

    ofile << endl;
}
```



```

//DateTime2StrArray示例, TimeStamp2StrArray的使用类似
if(k_DateTime == cols[3].type)
{
    StringSeq strs;
    DateTime2StrArray(strs, cols[3]);

    ofile << cols[3].name << endl;
    for(uint i = 0; i < strs.size(); ++ i)
        ofile << strs[i] << endl;

    ofile << endl;
}

//GetDataArrayT示例, double类型, 其他数值类型的使用类似
if(k_Double == cols[5].type)
{
    //double类型
    double *S1 = GetDataArrayT< double >(cols[5]);
    ofile << cols[5].name << endl;
    for(uint i = 0; i < cols[5].count; ++ i)
        ofile << S1[i] << endl;

    ofile << endl;

    //模拟填充数据, 将S1全部使用填充值填充
    for(uint i = 0; i < cols[5].count; ++ i)
        S1[i] = MaxDouble;

    //Null2Val示例, 将填充值转换为0
    StringSeq fields;
    fields.push_back(cols[5].name);
    Null2Val(fields, 0, cols);

    ofile << cols[5].name << endl;
    for(uint i = 0; i < cols[5].count; ++ i)
        ofile << S1[i] << endl;

    ofile << endl;
}

//GetDataArrayT示例, uint64类型, 其他数值类型的使用类似
if(k_UInt64 == cols[6].type)
{

```

```

uint64 *SV1 = GetDataArrayT< uint64 >(cols[6]);
ofile << cols[6].name << endl;
for(uint i = 0; i < cols[6].count; ++ i)
    ofile << SV1[i] << endl;

ofile << endl;
}

ofile.close();

```

## 4 红楹云数据接口返回的列式结果集汇总

### 4.1 [GetExchanges](#) 出参返回的列式结果集字段

| 接口支持字段        | 描述                       |
|---------------|--------------------------|
| Market        | 交易所 <a href="#">市场编码</a> |
| MarketName    | 交易所 市场中文名称               |
| ENName        | 交易所 市场英文名称               |
| CountryCode_3 | 国家区域 3 位英文代码             |

### 4.2 [GetTradeTypes](#) 入参 fields 及出参返回的列式结果集字段

| 接口支持字段               | 描述                    | 商品期货 | 金融期货（股指期货、国债期货） |
|----------------------|-----------------------|------|-----------------|
| VarietyID            | 品种 ID                 | ★    | ★               |
| TradingType          | <a href="#">交易品种</a>  | ★    | ★               |
| DeclareDate          | 公告日期                  | ★    | ★               |
| Market               | <a href="#">交易所代码</a> | ★    | ★               |
| DeclareType          | 公告类型                  | ★    | ★               |
| ChangeDate           | 变动实施日期                | ★    | ★               |
| UnderlyingAssets     | 标的品种                  | ★    | ★               |
| SpotID               | 现货 ID                 | ★    |                 |
| UnderlyingSecurityID | 标的证券 ID               |      |                 |
| PYShortName          | 拼音简称                  | ★    | ★               |
| TradingUnit          | 交易单位                  | ★    | ★               |

|                          |              |   |   |
|--------------------------|--------------|---|---|
| QuoteUnit                | 报价单位         | ★ | ★ |
| MinChangeUnit            | 最小变动价位       | ★ | ★ |
| FluctuationLimit         | 每日价格最大波动限制   | ★ | ★ |
| ContractMonth            | 合约月份         | ★ | ★ |
| TradingTime              | 交易时间         | ★ | ★ |
| LastTradingDate          | 最后交易日        | ★ | ★ |
| DeliveryDate             | 交割日期         | ★ | ★ |
| DeliveryGrade            | 交割等级         | ★ |   |
| DeliveryLocation         | 交割地点         | ★ |   |
| TradingMargin            | 交易保证金        | ★ | ★ |
| DeliveryMode             | 交割方式         | ★ | ★ |
| TradingFee               | 交易手续费        | ★ | ★ |
| MaxMarketOrderVolume     | 单笔市价最大委托数量   | ★ |   |
| MaxLimitOrderVolume      | 单笔限价最大委托数量   | ★ |   |
| MinSingleBuyVolume       | 单笔最小买入量      | ★ |   |
| MinSingleSellVolume      | 单笔最小卖出量      | ★ |   |
| DeliveryUnit             | 交割单位         | ★ | ★ |
| TradingFee1              | 交易手续费 1      | ★ |   |
| TradingFee2              | 交易手续费 2      | ★ |   |
| FlatFeeHalved            | 平今手续费减半      | ★ |   |
| DeliveryFee              | 交割手续费        |   | ★ |
| FinalSettlementDate      | 最后结算日        |   | ★ |
| PositionLimit            | 持仓限制         |   | ★ |
| UnderlyingSecuritySymbol | 标的代码         |   | ★ |
| UnderlyingIndex          | 标的指数         |   | ★ |
| ContractMultiple         | 合约单位         |   | ★ |
| LastDayTradingTime       | 最后交易日交易时间    |   | ★ |
| UnderlyingSecurityType   | 标的证券类型       |   |   |
| OptionType               | 期权类型         |   |   |
| ExerciseTradingTime      | 行权日交易时间      |   |   |
| LastTradingDateRule      | 最后交易日规则      |   |   |
| ExpireDateRule           | 到期日规则        |   |   |
| ExerciseDate             | 行权日规则        |   |   |
| DeliveryDateRule         | 交收日规则        |   |   |
| MinOrderVolume           | 限价指令每次最小下单数量 |   |   |
| MaxOrderVolume           | 限价指令每次最大下单数量 |   |   |
| StrikePriceIntervalRule  | 行权价格间距规则     |   |   |
| UnilateralPosLmt         | 单边持仓限额       |   |   |
| MarginAdjustCoefficient  | 保证金调整系数      |   |   |
| MinCoefficient           | 最低保障系数       |   |   |

| 接口支持字段                   | 中文名                   | ETF 期权 | 股指期权 |
|--------------------------|-----------------------|--------|------|
| VarietyID                | 品种 ID                 | ★      | ★    |
| TradingType              | <a href="#">交易品种</a>  | ★      | ★    |
| DeclareDate              | 公告日期                  | ★      | ★    |
| Market                   | <a href="#">交易所代码</a> | ★      | ★    |
| DeclareType              | 公告类型                  | ★      | ★    |
| ChangeDate               | 变动实施日期                | ★      | ★    |
| UnderlyingAssets         | 标的品种                  |        |      |
| SpotID                   | 现货 ID                 |        |      |
| UnderlyingSecurityID     | 标的证券 ID               |        | ★    |
| PYShortName              | 拼音简称                  |        |      |
| TradingUnit              | 交易单位                  |        | ★    |
| QuoteUnit                | 报价单位                  | ★      | ★    |
| MinChangeUnit            | 最小变动价位                | ★      | ★    |
| FluctuationLimit         | 每日价格最大波动限制            | ★      | ★    |
| ContractMonth            | 合约月份                  | ★      | ★    |
| TradingTime              | 交易时间                  | ★      | ★    |
| LastTradingDate          | 最后交易日                 |        |      |
| DeliveryDate             | 交割日期                  |        |      |
| DeliveryGrade            | 交割等级                  |        |      |
| DeliveryLocation         | 交割地点                  |        |      |
| TradingMargin            | 交易保证金                 | ★      |      |
| DeliveryMode             | 交割方式                  | ★      | ★    |
| TradingFee               | 交易手续费                 | ★      |      |
| MaxMarketOrderVolume     | 单笔市价最大委托数量            | ★      |      |
| MaxLimitOrderVolume      | 单笔限价最大委托数量            | ★      |      |
| MinSingleBuyVolume       | 单笔最小买入量               | ★      |      |
| MinSingleSellVolume      | 单笔最小卖出量               | ★      |      |
| DeliveryUnit             | 交割单位                  |        |      |
| TradingFee1              | 交易手续费 1               |        |      |
| TradingFee2              | 交易手续费 2               |        |      |
| FlatFeeHalved            | 平今手续费减半               |        |      |
| DeliveryFee              | 交割手续费                 |        |      |
| FinalSettlementDate      | 最后结算日                 |        |      |
| PositionLimit            | 持仓限制                  |        |      |
| UnderlyingSecuritySymbol | 标的代码                  |        |      |
| UnderlyingIndex          | 标的指数                  |        |      |
| ContractMultiple         | 合约单位                  |        | ★    |
| LastDayTradingTime       | 最后交易日交易时间             |        |      |
| UnderlyingSecurityType   | 标的证券类型                | ★      |      |
| OptionType               | 期权类型                  | ★      | ★    |

|                         |              |   |   |
|-------------------------|--------------|---|---|
| ExerciseTradingTime     | 行权日交易时间      | ★ | ★ |
| LastTradingDateRule     | 最后交易日规则      | ★ | ★ |
| ExpireDateRule          | 到期日规则        | ★ | ★ |
| ExerciseDate            | 行权日规则        | ★ | ★ |
| DeliveryDateRule        | 交收日规则        | ★ |   |
| MinOrderVolume          | 限价指令每次最小下单数量 |   | ★ |
| MaxOrderVolume          | 限价指令每次最大下单数量 |   | ★ |
| StrikePriceIntervalRule | 行权价格间距规则     |   | ★ |
| UnilateralPosLmt        | 单边持仓限额       |   | ★ |
| MarginAdjustCoefficient | 保证金调整系数      |   | ★ |
| MinCoefficient          | 最低保障系数       |   | ★ |

| 接口支持字段               | 中文名                   | 商品期权 |
|----------------------|-----------------------|------|
| VarietyID            | 品种 ID                 |      |
| TradingType          | <a href="#">交易品种</a>  |      |
| DeclareDate          | 公告日期                  | ★    |
| Market               | <a href="#">交易所代码</a> | ★    |
| DeclareType          | 公告类型                  | ★    |
| ChangeDate           | 变动实施日期                | ★    |
| UnderlyingAssets     | 标的品种                  | ★    |
| SpotID               | 现货 ID                 |      |
| UnderlyingSecurityID | 标的证券 ID               |      |
| PYShortName          | 拼音简称                  |      |
| TradingUnit          | 交易单位                  | ★    |
| QuoteUnit            | 报价单位                  | ★    |
| MinChangeUnit        | 最小变动价位                | ★    |
| FluctuationLimit     | 每日价格最大波动限制            | ★    |
| ContractMonth        | 合约月份                  | ★    |
| TradingTime          | 交易时间                  | ★    |
| LastTradingDate      | 最后交易日                 |      |
| DeliveryDate         | 交割日期                  |      |
| DeliveryGrade        | 交割等级                  |      |
| DeliveryLocation     | 交割地点                  |      |
| TradingMargin        | 交易保证金                 |      |
| DeliveryMode         | 交割方式                  |      |
| TradingFee           | 交易手续费                 | ★    |
| MaxMarketOrderVolume | 单笔市价最大委托数量            | ★    |
| MaxLimitOrderVolume  | 单笔限价最大委托数量            | ★    |
| MinSingleBuyVolume   | 单笔最小买入量               | ★    |

|                          |              |   |
|--------------------------|--------------|---|
| MinSingleSellVolume      | 单笔最小卖出量      | ★ |
| DeliveryUnit             | 交割单位         |   |
| TradingFee1              | 交易手续费 1      |   |
| TradingFee2              | 交易手续费 2      |   |
| FlatFeeHalved            | 平今手续费减半      |   |
| DeliveryFee              | 交割手续费        |   |
| FinalSettlementDate      | 最后结算日        |   |
| PositionLimit            | 持仓限制         |   |
| UnderlyingSecuritySymbol | 标的代码         |   |
| UnderlyingIndex          | 标的指数         |   |
| ContractMultiple         | 合约单位         |   |
| LastDayTradingTime       | 最后交易日交易时间    |   |
| UnderlyingSecurityType   | 标的证券类型       |   |
| OptionType               | 期权类型         | ★ |
| ExerciseTradingTime      | 行权日交易时间      |   |
| LastTradingDateRule      | 最后交易日规则      |   |
| ExpireDateRule           | 到期日规则        | ★ |
| ExerciseDate             | 行权日规则        | ★ |
| DeliveryDateRule         | 交收日规则        |   |
| MinOrderVolume           | 限价指令每次最小下单数量 |   |
| MaxOrderVolume           | 限价指令每次最大下单数量 |   |
| StrikePriceIntervalRule  | 行权价格间距规则     |   |
| UnilateralPosLmt         | 单边持仓限额       |   |
| MarginAdjustCoefficient  | 保证金调整系数      |   |
| MinCoefficient           | 最低保障系数       |   |

### 4.3 [GetTradeCalendar](#) 入参 fields 及出参返回的列式结果集字段

| 接口支持字段                 | 描述                   |
|------------------------|----------------------|
| Market                 | <a href="#">市场编码</a> |
| IsOpen                 | 是否开盘                 |
| Weekday                | 星期几                  |
| CalendarDate           | 日历日期                 |
| TypeID                 | 证券类别编码               |
| TradingType            | <a href="#">品种ID</a> |
| IsNightTrading         | 是否夜盘                 |
| NightTradingStartTimeA | 夜盘交易夜晚开始时间           |
| NightTradingEndTimeA   | 夜盘交易夜晚结束时间           |
| NightTradingDateA      | 夜盘夜晚交易所属业务日期         |
| NightTradingStartTimeB | 夜盘交易白天开始时间           |

|                      |              |
|----------------------|--------------|
| NightTradingEndTimeB | 夜盘交易白天结束时间   |
| NightTradingDateB    | 夜盘白天交易所属业务日期 |

#### 4.4 [GetPlates](#) 出参返回的列式结果集内容

| 接口支持字段        | 描述                      |
|---------------|-------------------------|
| PlateID       | <a href="#">板块树ID</a>   |
| TypeID        | 证券分类编码                  |
| PlateTypeCode | <a href="#">板块种类编码</a>  |
| PlateTreeID   | 板块树 PID                 |
| PlateTitle    | 板块名称                    |
| PlateCode     | 板块关联代码                  |
| NodeLevel     | 节点级别                    |
| IsLastNode    | 是否为叶子节点                 |
| UpdateState   | 数据状态（0 为新增，1 为更新，2 为删除） |

#### 4.5 [GetRelatedPlates](#) 出参返回的列式结果集内容

| 接口支持字段        | 描述                     |
|---------------|------------------------|
| PlateID       | 板块树 ID                 |
| TypeID        | 证券分类编码                 |
| PlateTypeCode | <a href="#">板块种类编码</a> |
| PlateTreeID   | 板块树 PID                |
| PlateTitle    | 板块名称                   |
| PlateCode     | 板块关联代码                 |
| NodeLevel     | 节点级别                   |
| IsLastNode    | 是否为叶子节点                |
| UpdateState   | 数据状态                   |

#### 4.6 [GetPlateSymbols](#) 出参返回的列式结果集内容

| 接口支持字段       | 描述                     |
|--------------|------------------------|
| SecurityID   | <a href="#">样本证券ID</a> |
| Symbol       | 样本证券代码                 |
| ExchangeCode | 上市市场编码                 |
| EnterDate    | 纳入板块日期                 |
| OutDate      | 纳出板块日期                 |
| UPDATESTATE  | 更新类型，0 新增，1 更新，2 删除    |

## 4.7 [GetSecurityInfo](#) 入参 fields 及出参返回的列式结果集字段

| 接口支持字段           | 描述                     |
|------------------|------------------------|
| 公用信息             |                        |
| SecurityID       | <a href="#">证券ID</a>   |
| Symbol           | 证券代码                   |
| Market           | <a href="#">市场编码</a>   |
| TypeID           | <a href="#">证券类别编码</a> |
| ShortName        | 中文简称                   |
| PYShortName      | 拼音简称                   |
| FullName         | 中文名称                   |
| EnShortName      | 英文简称                   |
| ENName           | 英文名称                   |
| InstitutionID    | 机构 ID                  |
| ListedDate       | 上市日期                   |
| DelistedDate     | 退市日期                   |
| CurrencyCode     | 交易货币币种编码               |
| ISIN             | <a href="#">ISIN编码</a> |
| StatusID         | 证券上市状态编码               |
| IssuePrice       | 发行价格                   |
| MarketSymbol     | 交易所原始代码                |
| 不同品种可能共有信息       |                        |
| 股票和债券            |                        |
| Currency         | 面值币种                   |
| ParValue         | 面值                     |
| 基金和债券            |                        |
| MaturityDate     | 到期日期                   |
| 期货和期权            |                        |
| ContractMulti    | 合约单位                   |
| LastTradingDate  | 最后交易日                  |
| QuoteUnit        | 报价单位                   |
| TradingUnit      | 交易单位                   |
| MinChangeUnit    | 最小变动单位                 |
| ContractMonth    | 合约月份                   |
| FluctuationLimit | 每日价格最大波动限制             |
| TradingMargin    | 交易保证金                  |
| DeliveryDate     | 交割日期                   |
| DeliveryMode     | 交割方式                   |
| TradingFee       | 交易手续费                  |
| LastDeliveryDate | 最后交割日                  |



|                        |           |
|------------------------|-----------|
| BenchmarkPrice         | 挂牌基准价     |
| MaxLimit               | 最大涨跌幅限制   |
| VarietyID              | 品种 ID     |
| UnderlyingSymbol       | 标的代码      |
| UnderlyingSecurityID   | 标的证券 ID   |
| <b>个股期权和股指期权</b>       |           |
| ContractCode           | 合约编码      |
| DeclareDate            | 公告日期      |
| ChangeDate             | 变动实施日期    |
| UnderlyingExchangeCode | 标的证券交易所代码 |
| UnderlyingShortName    | 标的证券简称    |
| UnderlyingType         | 标的证券类型    |
| ExpireDate             | 到期日       |
| ExerciseDate           | 行权日       |
| CallOrPut              | 认购认沽      |
| StrikePrice            | 行权价       |
| <b>股票专有信息</b>          |           |
| ShareType              | 股票类别      |
| FormerName             | 股票简称曾用名   |
| IpoShares              | 初始上市数量    |
| <b>基金专有信息</b>          |           |
| PurchaseStartDate      | 申购起始日     |
| RedeemStartDate        | 赎回起始日     |
| PurchaseStatus         | 申购状态      |
| AfterTransitionSymbol  | 封转开交叉代码   |
| IsPrimaryMarket        | 是否一级市场交易  |
| IsSecondaryMarket      | 是否二级市场交易  |
| SalesFee               | 销售服务费率    |
| FundID                 | 基金 ID     |
| FrontendCode           | 基金前端代码    |
| BackendCode            | 基金后端代码    |
| Expired                | 基金净值代码    |
| TradingShortName       | 场内简称      |
| RelationCode           | 基金关系编码    |
| Relation               | 基金关系      |
| InceptionDate          | 份额成立日期    |
| Duration               | 封闭年限      |
| Cycle                  | 申购赎回周期    |
| ClosedStartdate        | 封闭期起始日期   |
| ClosedEnddate          | 封闭期截止日期   |
| TransitionStartDate    | 转换开始日期    |
| RedeemStatus           | 赎回状态      |
| MasterFundCode         | 基金主代码     |

|                          |              |
|--------------------------|--------------|
| <b>商品期货专有信息</b>          |              |
| IsNightTrading           | 夜盘标识         |
| UnderlyingAssets         | 标的品种         |
| DeliveryGrade            | 交割等级         |
| DeliveryLocation         | 交割地点         |
| MaxMarketOrderVolume     | 单笔市价最大委托数量   |
| MaxLimitOrderVolume      | 单笔限价最大委托数量   |
| MinSingleBuyVolume       | 单笔最小买入量      |
| MinSingleSellVolume      | 单笔最小卖出量      |
| <b>金融期货专有信息</b>          |              |
| TradingTime              | 交易时间         |
| LastDayTradingTime       | 最后交易日交易时间    |
| DeliveryFee              | 交割手续费        |
| LongTradingMargin        | 多头交易保证金      |
| ShortTradingMargin       | 空头交易保证金      |
| MaxDailyTradingVolume    | 客户日最大交易量     |
| TradingType              | 交易品种         |
| <b>指数专有信息</b>            |              |
| BenchMarkDate            | 指数基期         |
| BenchMarkPoint           | 指数基点         |
| SampleScope              | 样本范围         |
| WeightingMethodID        | 加权方式编码       |
| WeightingMethod          | 加权方式         |
| FormulationInstitutionID | 编制机构 ID      |
| ReleasingInstitutionID   | 发布机构 ID      |
| ReleasingInstitution     | 发布机构         |
| SampleTypeID             | 按样本分类编码      |
| SampleType               | 按样本分类        |
| IndexTypeID              | 指数类型编码       |
| IndexType                | 指数类型         |
| ExchangeID               | 样本所属市场编码     |
| ExchangeName             | 样本所属市场名称     |
| FormulationInstitution   | 编制机构         |
| CalculatingMethod        | 计算方法         |
| AdjustingCycle           | 样本调整周期       |
| CountryCode_3            | 国家区域 3 位英文代码 |
| <b>个股期权专有信息</b>          |              |
| StrikePriceInterval      | 行权价格间距       |
| SpeculatePosLmt          | 个人投机持仓限额     |
| CoveredPosLmt            | 个人备兑开仓持仓限额   |
| CommonInstitutionPosLmt  | 一般机构持仓限额     |
| OwnAccountPosLmt         | 自营账户持仓限额     |
| MarketMakerPosLmt        | 做市商持仓限额      |

|                         |          |
|-------------------------|----------|
| DealersPosLmt           | 自营商持仓限额  |
| BrokersPosLmt           | 经纪商持仓限额  |
| <b>股指期货专有信息</b>         |          |
| MarginAdjustCoefficient | 保证金调整系数  |
| MinCoefficient          | 最低保障系数   |
| UnderlyingIndexTypeID   | 标的指数分类编码 |
| UnderlyingIndexType     | 标的指数类型   |
| <b>债券专有信息</b>           |          |
| BondID                  | 债券主体 ID  |
| MarketNAME              | 交易市场     |
| Type                    | 证券类别     |
| Institution             | 发行机构     |
| PlanDelistedDate        | 法定退市日期   |
| IsExpireID              | 目前状态 ID  |
| IsExpire                | 目前状态     |
| Status                  | 上市状态     |
| InitialIssueShares      | 初始发行量    |
| IssueYear               | 债券年度     |
| BondNatureID            | 债券类型 ID  |
| BondNature              | 债券类型     |
| PeriodTime              | 债券期次     |
| CategoryTypeID          | 品种类别 ID  |
| CategoryType            | 品种类别     |
| IssueShares             | 实际发行量    |
| IssueModeID             | 发行方式 ID  |
| IssueMode               | 发行方式     |
| BondFormID              | 债券形式 ID  |
| BondForm                | 债券形式     |
| Term                    | 期限       |
| IssueDate               | 发行日期     |
| InterestStartDate       | 计息日期     |
| InterestEndDate         | 止息日期     |
| PayTypeID               | 付息方式 ID  |
| PayType                 | 付息方式     |
| CalculationTypeID       | 计息方式 ID  |
| CalculationType         | 计息方式     |
| RateTypeID              | 利率类型 ID  |
| RateType                | 利率类型     |
| YearPayTime             | 年付息次数    |
| PaymentDateDescription  | 付息日期描述   |
| ParValueRate            | 票面利率     |
| Shibor                  | 基准利率     |
| BasicRate               | 基本利差     |

|                        |          |
|------------------------|----------|
| BondCreditLevel        | 债券信用等级   |
| IssuerCreditLevel      | 发行主体信用等级 |
| IsRedeem               | 可赎回性     |
| RedeemClause           | 赎回条款     |
| RedeemDate             | 赎回日期     |
| RedeemPrice            | 赎回价格     |
| IsSaleBack             | 可回售性     |
| SaleBackClause         | 回售条款     |
| SaleBackDate           | 回售日期     |
| SaleBackPrice          | 回售价格     |
| RedemptionDate         | 兑付日      |
| RedemptionInstructions | 兑付说明     |
| IsExchange             | 可调换性     |
| TransformationClause   | 可转换条款    |
| ExchangeBond           | 调换债券     |
| ExchangeRate           | 调换比例     |
| ExchangeDate           | 调换日期     |
| IsGuarantee            | 是否有担保    |
| IsCrossMarket          | 是否跨市场    |
| Explanation            | 跨市场说明    |
| IsSeparation           | 是否可分离    |
| 商品期权专有信息               |          |
| BENCHMARKPRICE         | 挂牌基准价    |
| OPTIONTYPE             | 期权类型     |
| CONTRACTMONTH          | 合约月份     |
| UNDERLYINGVARIETYID    | 标的品种 ID  |

## 4.8 [GetSecurityCurInfo](#) 入参 fields 及出参返回的列式结果集字段

| 接口支持字段              | 描述                   |
|---------------------|----------------------|
| Tradingdate         | 交易日期                 |
| Symbol              | 证券代码                 |
| SecurityID          | <a href="#">证券ID</a> |
| ShortName           | 证券简称                 |
| PYShortName         | 拼音简称                 |
| Market              | 交易所编码                |
| CirShare            | 流通股本                 |
| TotalShare          | 总股本                  |
| IndustryCode        | 所属行业                 |
| PerMinTradingVolume | 前五日分钟均量              |

|                         |            |
|-------------------------|------------|
| LCP                     | 前收盘价       |
| LimitUp                 | 涨停价        |
| LimitDown               | 跌停价        |
| StatusID                | 交易状态编码     |
| MinTickSize             | 价格档位       |
| EPS                     | 本年每股利润     |
| EndDate                 | 截止日期       |
| NAPS                    | 每股净资产      |
| InceptionShares         | 基金成立时份额    |
| NAV                     | 基金份额净值     |
| IssuePrice              | 发行价格       |
| ParValueRate            | 票面利率       |
| InterestStartDate       | 计息日期       |
| InterestDays            | 计息天数       |
| AccruedInterest         | 应计利息       |
| ISLastTradingDate       | 是否最后交易日    |
| DeclareDate             | 业务发生日期     |
| MarketSymbol            | 交易所原始代码    |
| PrePosition             | 昨持仓量       |
| LSP                     | 前结算价       |
| OBPD                    | 开盘基准价      |
| ChangeRatioLimitTypeID  | 涨跌幅限制类型    |
| Clientpositionlimit     | 客户持仓量限制    |
| RemainingDays           | 剩余天数       |
| LastTradingDate         | 最后交易日      |
| ContractCode            | 合约编码       |
| UnderlyingSymbol        | 标的证券交易代码   |
| UnderlyingShortName     | 标的证券简称     |
| UnderlyingSecurityID    | 标的证券证券 ID  |
| Underlyingtype          | 标的证券类型     |
| UnderlyingPreClosePrice | 标的证券前收盘    |
| OptionType              | 欧式美式       |
| CallOrPut               | 认购认沽       |
| ContractMulti           | 合约单位       |
| StrikePrice             | 行权价        |
| ListedDate              | 首日上市日期     |
| ExerciseDate            | 行权日        |
| DeliveryDate            | 交收日        |
| ExpireDate              | 到期日        |
| PreVolume               | 昨成交量       |
| MarginUnit              | 当日单张初始保证金  |
| Interestrates           | 无风险利率      |
| SecurityStatususFlag    | 期权合约状态信息标签 |

|         |        |
|---------|--------|
| TypeID  | 证券分类编码 |
| ISTrade | 是否交易   |

## 4.9 [GetDataByTime/GetDataByCount](#) 入参 fields 及出参返回的列式结果集字段

| 接口支持字段      | 描述                   |
|-------------|----------------------|
| 分时数据字段      |                      |
| SecurityID  | <a href="#">证券ID</a> |
| Symbol      | 证券代码                 |
| TradingTime | 交易时间                 |
| TradingDate | 交易归属日期               |
| ShortName   | 证券名称                 |
| OBPD        | 开盘基准价                |
| OP          | 开盘价                  |
| HIP         | 最高价                  |
| LOP         | 最低价                  |
| CP          | 收盘价                  |
| B1          | 买一价                  |
| B2          | 买二价                  |
| B3          | 买三价                  |
| B4          | 买四价                  |
| B5          | 买五价                  |
| S1          | 卖一价                  |
| S2          | 卖二价                  |
| S3          | 卖三价                  |
| S4          | 卖四价                  |
| S5          | 卖五价                  |
| BV1         | 买一量                  |
| BV2         | 买二量                  |
| BV3         | 买三量                  |
| BV4         | 买四量                  |
| BV5         | 买五量                  |
| SV1         | 卖一量                  |
| SV2         | 卖二量                  |
| SV3         | 卖三量                  |
| SV4         | 卖四量                  |
| SV5         | 卖五量                  |
| Change      | 涨跌                   |
| ChangeRatio | 涨跌幅                  |

|                   |                      |
|-------------------|----------------------|
| CQ                | 成交量                  |
| CM                | 成交额                  |
| Position          | 持仓量                  |
| TQ                | 成交总量                 |
| Vwap              | 加权均价                 |
| MarketID          | 证券市场代码               |
| Unix              | UNIX 时间戳             |
| 以下字段为日频           |                      |
| SecurityID        | <a href="#">证券ID</a> |
| TradingDate       | 交易日期                 |
| Symbol            | 证券代码                 |
| ShortName         | 证券简称                 |
| Filling           | 填充标识                 |
| OBPD              | 开盘基准价                |
| OP                | 开盘价                  |
| HIP               | 最高价                  |
| LOP               | 最低价                  |
| CP                | 收盘价                  |
| CQ                | 成交量                  |
| CM                | 成交额                  |
| Vwap              | 成交均价                 |
| LCP               | 昨收价                  |
| LimitUp           | 涨停价                  |
| LimitDown         | 跌停价                  |
| Change            | 涨跌                   |
| ChangeRatio       | 涨跌幅                  |
| Amplitude         | 振幅                   |
| TurnoverRate      | 换手率                  |
| CMV               | 流通市值                 |
| TMV               | 总市值                  |
| SP                | 结算价                  |
| LSP               | 昨结价                  |
| Position          | 合约持仓量                |
| ContractUnit      | 合约保证金比率              |
| MarginUnit        | 期权初始保证金              |
| MaintainingMargin | 期权维持保证金              |
| AccruedInterest   | 应计利息(债券)             |

## 周频分时数据字段

|             |                      |
|-------------|----------------------|
| SecurityID  | <a href="#">证券ID</a> |
| TradingDate | 交易日期                 |
| TradingWeek | 交易周份                 |
| Symbol      | 证券代码                 |

|                   |                   |
|-------------------|-------------------|
| Filling           | 填充标识              |
| OpenDate          | 周开盘日期             |
| PrvCloseDate      | 上周收盘日             |
| LCP               | 上周收盘价             |
| OP                | 周开盘价              |
| HIP               | 周最高价              |
| LOP               | 周最低价              |
| HighDate          | 周最高价日             |
| LowDate           | 周最低价日             |
| CP                | 周收盘价              |
| CloseDate         | 周收盘日期             |
| HighClosePrice    | 周最高收盘价            |
| LowClosePrice     | 周最低收盘价            |
| Vwap              | 周均价               |
| Change            | 周涨跌               |
| ChangeRatio       | 周涨跌幅              |
| TurnoverRate1     | 周换手率              |
| TurnoverRate      | 周换手率(基准.自由流通股本)   |
| AvgTurnoverRate1  | 周平均换手率            |
| AvgTurnoverRate   | 周平均换手率(基准.自由流通股本) |
| CQ                | 周成交量              |
| CM                | 周成交额              |
| Amplitude         | 周振幅               |
| HighCloseDate     | 周最高收盘价日           |
| LowCloseDate      | 周最低收盘价日           |
| TradingDays       | 周交易天数             |
| SP                | 周结算价              |
| HighSettlePrice   | 周最高结算价            |
| LowSettlePrice    | 周最低结算价            |
| HighSettleDate    | 周最高结算价日           |
| LowSettleDate     | 周最低结算价日           |
| Position          | 周持仓量              |
| HighPosition      | 周最高持仓量            |
| LowPosition       | 周最低持仓量            |
| PrePositionChange | 周持仓量变化            |
| LSP               | 上周结算价             |
| AvgVolume         | 周日均成交量            |
| AvgAmount         | 周日均成交额            |
| AvgPosition       | 周日均持仓量            |
| HighVolume        | 周最高成交量            |
| LowVolume         | 周最低成交量            |
| HighAmount        | 周最高成交金额           |
| LowAmount         | 周最低成交金额           |



| 月频分时数据字段          |                      |
|-------------------|----------------------|
| SecurityID        | <a href="#">证券ID</a> |
| TradingDate       | 交易日期                 |
| TradingMonth      | 交易月份                 |
| Symbol            | 证券代码                 |
| Filling           | 填充标识                 |
| OpenDate          | 月开盘日期                |
| PrvCloseDate      | 上月收盘日                |
| LCP               | 上月收盘价                |
| OP                | 月开盘价                 |
| HIP               | 月最高价                 |
| LOP               | 月最低价                 |
| HighDate          | 月最高价日                |
| LowDate           | 月最低价日                |
| CP                | 月收盘价                 |
| CloseDate         | 月收盘日期                |
| HighClosePrice    | 月最高收盘价               |
| LowClosePrice     | 月最低收盘价               |
| Vwap              | 月均价                  |
| Change            | 月涨跌                  |
| ChangeRatio       | 月涨跌幅                 |
| TurnoverRate1     | 月换手率                 |
| TurnoverRate      | 月换手率(基准.自由流通股本)      |
| AvgTurnoverRate1  | 月平均换手率               |
| AvgTurnoverRate   | 月平均换手率(基准.自由流通股本)    |
| CQ                | 月成交量                 |
| CM                | 月成交额                 |
| Amplitude         | 月振幅                  |
| HighCloseDate     | 月最高收盘价日              |
| LowCloseDate      | 月最低收盘价日              |
| TradingDays       | 月交易天数                |
| SP                | 月结算价                 |
| HighSettlePrice   | 月最高结算价               |
| LowSettlePrice    | 月最低结算价               |
| HighSettleDate    | 月最高结算价日              |
| LowSettleDate     | 月最低结算价日              |
| Position          | 月持仓量                 |
| HighPosition      | 月最高持仓量               |
| LowPosition       | 月最低持仓量               |
| PrePositionChange | 月持仓量变化               |
| LSP               | 上月结算价                |
| AvgVolume         | 月日均成交量               |

|             |         |
|-------------|---------|
| AvgAmount   | 月日均成交额  |
| AvgPosition | 月日均持仓量  |
| HighVolume  | 月最高成交量  |
| LowVolume   | 月最低成交量  |
| HighAmount  | 月最高成交金额 |
| LowAmount   | 月最低成交金额 |

### 年频分时数据字段

|                  |                      |
|------------------|----------------------|
| SecurityID       | <a href="#">证券ID</a> |
| TradingDate      | 交易日期                 |
| TradingYear      | 交易年份                 |
| Symbol           | 证券代码                 |
| Filling          | 填充标识                 |
| OpenDate         | 年开盘日期                |
| PrvCloseDate     | 上年收盘日                |
| LCP              | 上年收盘价                |
| OP               | 年开盘价                 |
| HIP              | 年最高价                 |
| LOP              | 年最低价                 |
| HighDate         | 年最高价日                |
| LowDate          | 年最低价日                |
| CP               | 年收盘价                 |
| CloseDate        | 年收盘日期                |
| HighClosePrice   | 年最高收盘价               |
| LowClosePrice    | 年最低收盘价               |
| Vwap             | 年均价                  |
| Change           | 年涨跌                  |
| ChangeRatio      | 年涨跌幅                 |
| TurnoverRate1    | 年换手率                 |
| TurnoverRate     | 年换手率(基准.自由流通股本)      |
| AvgTurnoverRate1 | 年平均换手率               |
| AvgTurnoverRate  | 年平均换手率(基准.自由流通股本)    |
| CQ               | 年成交量                 |
| CM               | 年成交额                 |
| Amplitude        | 年振幅                  |
| HighCloseDate    | 年最高收盘价日              |
| LowCloseDate     | 年最低收盘价日              |
| TradingDays      | 年交易天数                |
| SP               | 年结算价                 |
| HighSettlePrice  | 年最高结算价               |
| LowSettlePrice   | 年最低结算价               |
| HighSettleDate   | 年最高结算价日              |
| LowSettleDate    | 年最低结算价日              |

|                   |         |
|-------------------|---------|
| Position          | 年持仓量    |
| HighPosition      | 年最高持仓量  |
| LowPosition       | 年最低持仓量  |
| PrePositionChange | 年持仓量变化  |
| LSP               | 上年结算价   |
| AvgVolume         | 年日均成交量  |
| AvgAmount         | 年日均成交额  |
| AvgPosition       | 年日均持仓量  |
| HighVolume        | 年最高成交量  |
| LowVolume         | 年最低成交量  |
| HighAmount        | 年最高成交金额 |
| LowAmount         | 年最低成交金额 |

#### 4.10 [GetTickByTime/GetTickByCount/GetSnapData](#) 入参 fields 及出参返回的列式结果集字段

| 接口支持字段<br>(分笔数据) | 描述     | 个股期权 | 股票、债券、<br>基金、指数 |
|------------------|--------|------|-----------------|
| SecurityID       | 证券统一编码 | ★    | ★               |
| Symbol           | 证券代码   | ★    | ★               |
| ShortName        | 证券名称   |      | ★               |
| TradingDate      | 交易日期   | ★    | ★               |
| TradingTime      | 交易时间   | ★    | ★               |
| UpDateMS         | 更新时间毫秒 |      |                 |
| CP               | 最新价    | ★    | ★               |
| Change           | 涨跌     | ★    | ★               |
| ChangeRatio      | 涨幅     | ★    |                 |
| CQ               | 现量     | ★    | ★               |
| CM               | 现额     | ★    | ★               |
| CT               | 当前成交笔数 |      | ★               |
| OC               | 性质     | ★    |                 |
| S5               | 卖五价    | ★    | ★               |
| S4               | 卖四价    | ★    | ★               |
| S3               | 卖三价    | ★    | ★               |
| S2               | 卖二价    | ★    | ★               |
| S1               | 卖一价    | ★    | ★               |
| B1               | 买一价    | ★    | ★               |
| B2               | 买二价    | ★    | ★               |
| B3               | 买三价    | ★    | ★               |
| B4               | 买四价    | ★    | ★               |

|                   |        |   |   |
|-------------------|--------|---|---|
| B5                | 买五价    | ★ | ★ |
| SV5               | 卖五量    | ★ | ★ |
| SV4               | 卖四量    | ★ | ★ |
| SV3               | 卖三量    | ★ | ★ |
| SV2               | 卖二量    | ★ | ★ |
| SV1               | 卖一量    | ★ | ★ |
| BV1               | 买一量    | ★ | ★ |
| BV2               | 买二量    | ★ | ★ |
| BV3               | 买三量    | ★ | ★ |
| BV4               | 买四量    | ★ | ★ |
| BV5               | 买五量    | ★ | ★ |
| LastVol           | 最新成交量  |   |   |
| BidImpleQty       | 申买推导量  |   |   |
| AskImpleQty       | 申卖推导量  |   |   |
| Change1           | 价格升跌 2 |   | ★ |
| PE1               | 市盈率 1  |   | ★ |
| PE2               | 市盈率 2  |   | ★ |
| BS                | 买卖方向   | ★ | ★ |
| BuyRatio          | 买比     |   | ★ |
| SPD               | 买卖价差   |   | ★ |
| RPD               | 相对价差   |   | ★ |
| LCP               | 昨收价    | ★ | ★ |
| OP                | 开盘价    | ★ | ★ |
| HIP               | 最高价    | ★ | ★ |
| LOP               | 最低价    | ★ | ★ |
| CP1               | 收盘价    | ★ |   |
| Vwap              | 均价     |   |   |
| LimitUp           | 涨停价    |   |   |
| LimitDown         | 跌停价    |   |   |
| TT                | 成交笔数   |   | ★ |
| TQ                | 成交量    | ★ | ★ |
| TM                | 成交额    | ★ | ★ |
| PrePosition       | 昨持仓量   | ★ |   |
| Position          | 持仓量    | ★ |   |
| PrePositionChange | 日增仓    | ★ |   |
| LSP               | 昨结算价   | ★ |   |
| SP                | 结算价    | ★ |   |
| PreDelta          | 昨虚实度   |   |   |
| Delta             | 今虚实度   |   |   |
| SettleGroupID     | 结算组代码  |   |   |
| SettleID          | 结算编号   |   |   |
| ConSign           | 连续合约代码 |   |   |
| ConSignName       | 连续合约名称 |   |   |

|                |                |   |   |
|----------------|----------------|---|---|
| Market         | 市场             | ★ | ★ |
| Varieties      | 品种             |   |   |
| LifeHigh       | 历史最高价          |   |   |
| LifeLow        | 历史最低价          |   |   |
| AuctionPrice   | 波动性中断参考价       | ★ |   |
| AuctionVolume  | 波动性中断集合竞价虚拟匹配量 | ★ |   |
| BuyLevelNO     | 委托买入档位         | ★ |   |
| SellLevelNO    | 委托卖出档位         | ★ |   |
| StatusID       | 交易状态           | ★ |   |
| MessageID      | 消息序号           | ★ |   |
| RecTime        | 接收时间           | ★ |   |
| LocalTimeStamp | 本地时间戳          | ★ |   |
| UNIX           | UNIX 时间戳       | ★ | ★ |
| Theta          | Theta          |   |   |
| Vega           | Vega           |   |   |
| Rho            | Rho            |   |   |
| Gamma          | Gamma          |   |   |

| 接口支持字段<br>(分笔数据) | 描述     | 商品期货 | 商品期权 |
|------------------|--------|------|------|
| SecurityID       | 证券统一编码 | ★    | ★    |
| Symbol           | 证券代码   | ★    | ★    |
| ShortName        | 证券名称   | ★    | ★    |
| TradingDate      | 交易日期   | ★    | ★    |
| TradingTime      | 交易时间   | ★    | ★    |
| UpDateMS         | 更新时间毫秒 |      |      |
| CP               | 最新价    | ★    | ★    |
| Change           | 涨跌     | ★    | ★    |
| ChangeRatio      | 涨幅     | ★    | ★    |
| CQ               | 现量     | ★    | ★    |
| CM               | 现额     |      |      |
| CT               | 当前成交笔数 |      |      |
| OC               | 性质     |      |      |
| S5               | 卖五价    | ★    | ★    |
| S4               | 卖四价    | ★    | ★    |
| S3               | 卖三价    | ★    | ★    |
| S2               | 卖二价    | ★    | ★    |
| S1               | 卖一价    | ★    | ★    |
| B1               | 买一价    | ★    | ★    |
| B2               | 买二价    | ★    | ★    |
| B3               | 买三价    | ★    | ★    |
| B4               | 买四价    | ★    | ★    |

|                   |        |   |   |
|-------------------|--------|---|---|
| B5                | 买五价    | ★ | ★ |
| SV5               | 卖五量    | ★ | ★ |
| SV4               | 卖四量    | ★ | ★ |
| SV3               | 卖三量    | ★ | ★ |
| SV2               | 卖二量    | ★ | ★ |
| SV1               | 卖一量    | ★ | ★ |
| BV1               | 买一量    | ★ | ★ |
| BV2               | 买二量    | ★ | ★ |
| BV3               | 买三量    | ★ | ★ |
| BV4               | 买四量    | ★ | ★ |
| BV5               | 买五量    | ★ | ★ |
| LastVol           | 最新成交量  | ★ | ★ |
| BidImpleQty       | 申买推导量  | ★ | ★ |
| AskImpleQty       | 申卖推导量  | ★ | ★ |
| Change1           | 价格升跌 2 |   |   |
| PE1               | 市盈率 1  |   |   |
| PE2               | 市盈率 2  |   |   |
| BS                | 买卖方向   | ★ | ★ |
| BuyRatio          | 买比     |   |   |
| SPD               | 买卖价差   |   |   |
| RPD               | 相对价差   |   |   |
| LCP               | 昨收价    | ★ | ★ |
| OP                | 开盘价    | ★ | ★ |
| HIP               | 最高价    | ★ | ★ |
| LOP               | 最低价    | ★ | ★ |
| CP1               | 收盘价    | ★ | ★ |
| Vwap              | 均价     | ★ | ★ |
| LimitUp           | 涨停价    | ★ | ★ |
| LimitDown         | 跌停价    | ★ | ★ |
| TT                | 成交笔数   |   |   |
| TQ                | 成交量    | ★ | ★ |
| TM                | 成交额    | ★ | ★ |
| PrePosition       | 昨持仓量   | ★ | ★ |
| Position          | 持仓量    | ★ | ★ |
| PrePositionChange | 日增仓    | ★ | ★ |
| LSP               | 昨结算价   | ★ | ★ |
| SP                | 结算价    | ★ | ★ |
| PreDelta          | 昨虚实度   | ★ | ★ |
| Delta             | 今虚实度   | ★ | ★ |
| SettleGroupID     | 结算组代码  |   |   |
| SettleID          | 结算编号   |   |   |
| ConSign           | 连续合约代码 | ★ | ★ |
| ConSignName       | 连续合约名称 |   |   |

|                |                |   |   |
|----------------|----------------|---|---|
| Market         | 市场             | ★ | ★ |
| Varieties      | 品种             | ★ | ★ |
| LifeHigh       | 历史最高价          | ★ | ★ |
| LifeLow        | 历史最低价          | ★ | ★ |
| AuctionPrice   | 波动性中断参考价       |   |   |
| AuctionVolume  | 波动性中断集合竞价虚拟匹配量 |   |   |
| BuyLevelNO     | 委托买入档位         |   |   |
| SellLevelNO    | 委托卖出档位         |   |   |
| StatusID       | 交易状态           |   |   |
| MessageID      | 消息序号           |   |   |
| RecTime        | 接收时间           |   |   |
| LocalTimeStamp | 本地时间戳          | ★ |   |
| UNIX           | UNIX 时间戳       | ★ | ★ |
| Theta          | Theta          |   | ★ |
| Vega           | Vega           |   | ★ |
| Rho            | Rho            |   | ★ |
| Gamma          | Gamma          |   | ★ |

| 接口支持字段<br>(分笔数据) | 描述     | 股指期货 | 国债期货 |
|------------------|--------|------|------|
| SecurityID       | 证券统一编码 | ★    | ★    |
| Symbol           | 证券代码   | ★    | ★    |
| ShortName        | 证券名称   |      |      |
| TradingDate      | 交易日期   | ★    | ★    |
| TradingTime      | 交易时间   | ★    | ★    |
| UpDateMS         | 更新时间毫秒 | ★    |      |
| CP               | 最新价    | ★    | ★    |
| Change           | 涨跌     | ★    | ★    |
| ChangeRatio      | 涨幅     | ★    | ★    |
| CQ               | 现量     | ★    | ★    |
| CM               | 现额     | ★    | ★    |
| CT               | 当前成交笔数 |      |      |
| OC               | 性质     | ★    | ★    |
| S5               | 卖五价    | ★    | ★    |
| S4               | 卖四价    | ★    | ★    |
| S3               | 卖三价    | ★    | ★    |
| S2               | 卖二价    | ★    | ★    |
| S1               | 卖一价    | ★    | ★    |
| B1               | 买一价    | ★    | ★    |
| B2               | 买二价    | ★    | ★    |
| B3               | 买三价    | ★    | ★    |
| B4               | 买四价    | ★    | ★    |

|                   |        |   |   |
|-------------------|--------|---|---|
| B5                | 买五价    | ★ | ★ |
| SV5               | 卖五量    | ★ | ★ |
| SV4               | 卖四量    | ★ | ★ |
| SV3               | 卖三量    | ★ | ★ |
| SV2               | 卖二量    | ★ | ★ |
| SV1               | 卖一量    | ★ | ★ |
| BV1               | 买一量    | ★ | ★ |
| BV2               | 买二量    | ★ | ★ |
| BV3               | 买三量    | ★ | ★ |
| BV4               | 买四量    | ★ | ★ |
| BV5               | 买五量    | ★ | ★ |
| LastVol           | 最新成交量  |   |   |
| BidImpleQty       | 申买推导量  |   |   |
| AskImpleQty       | 申卖推导量  |   |   |
| Change1           | 价格升跌 2 |   |   |
| PE1               | 市盈率 1  |   |   |
| PE2               | 市盈率 2  |   |   |
| BS                | 买卖方向   | ★ | ★ |
| BuyRatio          | 买比     | ★ |   |
| SPD               | 买卖价差   |   |   |
| RPD               | 相对价差   |   |   |
| LCP               | 昨收价    | ★ | ★ |
| OP                | 开盘价    | ★ | ★ |
| HIP               | 最高价    | ★ | ★ |
| LOP               | 最低价    | ★ | ★ |
| CP1               | 收盘价    | ★ | ★ |
| Vwap              | 均价     |   |   |
| LimitUp           | 涨停价    | ★ | ★ |
| LimitDown         | 跌停价    | ★ | ★ |
| TT                | 成交笔数   |   |   |
| TQ                | 成交量    | ★ | ★ |
| TM                | 成交额    | ★ | ★ |
| PrePosition       | 昨持仓量   | ★ | ★ |
| Position          | 持仓量    | ★ | ★ |
| PrePositionChange | 日增仓    |   | ★ |
| LSP               | 昨结算价   | ★ | ★ |
| SP                | 结算价    | ★ | ★ |
| PreDelta          | 昨虚实度   | ★ | ★ |
| Delta             | 今虚实度   | ★ | ★ |
| SettleGroupID     | 结算组代码  | ★ | ★ |
| SettleID          | 结算编号   | ★ | ★ |
| ConSign           | 连续合约代码 | ★ |   |
| ConSignName       | 连续合约名称 | ★ |   |



|                |                |   |   |
|----------------|----------------|---|---|
| Market         | 市场             | ★ | ★ |
| Varieties      | 品种             |   |   |
| LifeHigh       | 历史最高价          |   |   |
| LifeLow        | 历史最低价          |   |   |
| AuctionPrice   | 波动性中断参考价       |   |   |
| AuctionVolume  | 波动性中断集合竞价虚拟匹配量 |   |   |
| BuyLevelNO     | 委托买入档位         |   |   |
| SellLevelNO    | 委托卖出档位         |   |   |
| StatusID       | 交易状态           |   |   |
| MessageID      | 消息序号           |   | ★ |
| RecTime        | 接收时间           |   | ★ |
| LocalTimeStamp | 本地时间戳          |   | ★ |
| UNIX           | UNIX 时间戳       | ★ | ★ |
| Theta          | Theta          |   |   |
| Vega           | Vega           |   |   |
| Rho            | Rho            |   |   |
| Gamma          | Gamma          |   |   |

| 接口支持字段<br>(分笔数据) | 描述     | 股指期货 |
|------------------|--------|------|
| SecurityID       | 证券统一编码 | ★    |
| Symbol           | 证券代码   | ★    |
| ShortName        | 证券名称   |      |
| TradingDate      | 交易日期   | ★    |
| TradingTime      | 交易时间   | ★    |
| UpDateMS         | 更新时间毫秒 |      |
| CP               | 最新价    | ★    |
| Change           | 涨跌     | ★    |
| ChangeRatio      | 涨幅     | ★    |
| CQ               | 现量     | ★    |
| CM               | 现额     | ★    |
| CT               | 当前成交笔数 |      |
| OC               | 性质     | ★    |
| S5               | 卖五价    | ★    |
| S4               | 卖四价    | ★    |
| S3               | 卖三价    | ★    |
| S2               | 卖二价    | ★    |
| S1               | 卖一价    | ★    |
| B1               | 买一价    | ★    |
| B2               | 买二价    | ★    |
| B3               | 买三价    | ★    |
| B4               | 买四价    | ★    |

|                   |        |   |
|-------------------|--------|---|
| B5                | 买五价    | ★ |
| SV5               | 卖五量    | ★ |
| SV4               | 卖四量    | ★ |
| SV3               | 卖三量    | ★ |
| SV2               | 卖二量    | ★ |
| SV1               | 卖一量    | ★ |
| BV1               | 买一量    | ★ |
| BV2               | 买二量    | ★ |
| BV3               | 买三量    | ★ |
| BV4               | 买四量    | ★ |
| BV5               | 买五量    | ★ |
| LastVol           | 最新成交量  |   |
| BidImPLYQty       | 申买推导量  |   |
| AskImPLYQty       | 申卖推导量  |   |
| Change1           | 价格升跌 2 |   |
| PE1               | 市盈率 1  |   |
| PE2               | 市盈率 2  |   |
| BS                | 买卖方向   | ★ |
| BuyRatio          | 买比     |   |
| SPD               | 买卖价差   |   |
| RPD               | 相对价差   |   |
| LCP               | 昨收价    | ★ |
| OP                | 开盘价    | ★ |
| HIP               | 最高价    | ★ |
| LOP               | 最低价    | ★ |
| CP1               | 收盘价    | ★ |
| Vwap              | 均价     |   |
| LimitUp           | 涨停价    | ★ |
| LimitDown         | 跌停价    | ★ |
| TT                | 成交笔数   |   |
| TQ                | 成交量    | ★ |
| TM                | 成交额    | ★ |
| PrePosition       | 昨持仓量   | ★ |
| Position          | 持仓量    | ★ |
| PrePositionChange | 日增仓    | ★ |
| LSP               | 昨结算价   | ★ |
| SP                | 结算价    | ★ |
| PreDelta          | 昨虚实度   | ★ |
| Delta             | 今虚实度   | ★ |
| SettleGroupID     | 结算组代码  | ★ |
| SettleID          | 结算编号   | ★ |
| ConSign           | 连续合约代码 | ★ |
| ConSignName       | 连续合约名称 |   |

|                |                |   |
|----------------|----------------|---|
| Market         | 市场             | ★ |
| Varieties      | 品种             |   |
| LifeHigh       | 历史最高价          |   |
| LifeLow        | 历史最低价          |   |
| AuctionPrice   | 波动性中断参考价       |   |
| AuctionVolume  | 波动性中断集合竞价虚拟匹配量 |   |
| BuyLevelNO     | 委托买入档位         |   |
| SellLevelNO    | 委托卖出档位         |   |
| StatusID       | 交易状态           |   |
| MessageID      | 消息序号           |   |
| RecTime        | 接收时间           | ★ |
| LocalTimeStamp | 本地时间戳          | ★ |
| UNIX           | UNIX 时间戳       | ★ |
| Theta          | Theta          |   |
| Vega           | Vega           |   |
| Rho            | Rho            |   |
| Gamma          | Gamma          |   |

#### 4.11 [GetL2TickByTime/GetL2TickByCount/GetSnapData](#) 入

参 fields 及出参返回的列式结果集字段

上海证券交易所：

| 接口支持字段名称    | 描述       |
|-------------|----------|
| 集合竞价        |          |
| Symbol      | 证券代码     |
| TradingDate | 交易归属日期   |
| TradingTime | 交易发生时间   |
| AOP         | 虚拟开盘参考价格 |
| AVOL        | 虚拟匹配量    |
| LVOL        | 虚拟未匹配量   |
| BS          | 买卖方向     |
| UNIX        | UNIX 时间戳 |
| Market      | 市场       |
| SecurityId  | 证券统一编码   |
| 指数行情        |          |
| Symbol      | 证券代码     |
| TradingDate | 交易归属日期   |
| TradingTime | 交易发生时间   |

|             |            |
|-------------|------------|
| CP1         | 今收盘价       |
| LCP         | 昨收价        |
| OP          | 开盘价        |
| HIP         | 最高价        |
| LOP         | 最低价        |
| CP          | 现价         |
| TQ          | 成交总量       |
| TM          | 成交总额       |
| UNIX        | UNIX 时间戳   |
| Market      | 市场         |
| SecurityId  | 证券统一编码     |
| 委托队列        |            |
| Symbol      | 证券代码       |
| TradingDate | 交易归属日期     |
| TradingTime | 交易发生时间     |
| BS          | 买卖方向       |
| NPL         | 买卖盘价位数量    |
| Price       | 委托价格       |
| Numorders   | 最优价位上的委托笔数 |
| OV1         | 委托量 01     |
| OV2         | 委托量 02     |
| OV3         | 委托量 03     |
| OV4         | 委托量 04     |
| OV5         | 委托量 05     |
| OV6         | 委托量 06     |
| OV7         | 委托量 07     |
| OV8         | 委托量 08     |
| OV9         | 委托量 09     |
| OV10        | 委托量 10     |
| OV11        | 委托量 11     |
| OV12        | 委托量 12     |
| OV13        | 委托量 13     |
| OV14        | 委托量 14     |
| OV15        | 委托量 15     |
| OV16        | 委托量 16     |
| OV17        | 委托量 17     |
| OV18        | 委托量 18     |
| OV19        | 委托量 19     |
| OV20        | 委托量 20     |
| OV21        | 委托量 21     |
| OV22        | 委托量 22     |
| OV23        | 委托量 23     |
| OV24        | 委托量 24     |

|             |          |
|-------------|----------|
| OV25        | 委托量 25   |
| OV26        | 委托量 26   |
| OV27        | 委托量 27   |
| OV28        | 委托量 28   |
| OV29        | 委托量 29   |
| OV30        | 委托量 30   |
| OV31        | 委托量 31   |
| OV32        | 委托量 32   |
| OV33        | 委托量 33   |
| OV34        | 委托量 34   |
| OV35        | 委托量 35   |
| OV36        | 委托量 36   |
| OV37        | 委托量 37   |
| OV38        | 委托量 38   |
| OV39        | 委托量 39   |
| OV40        | 委托量 40   |
| OV41        | 委托量 41   |
| OV42        | 委托量 42   |
| OV43        | 委托量 43   |
| OV44        | 委托量 44   |
| OV45        | 委托量 45   |
| OV46        | 委托量 46   |
| OV47        | 委托量 47   |
| OV48        | 委托量 48   |
| OV49        | 委托量 49   |
| OV50        | 委托量 50   |
| UNIX        | UNIX 时间戳 |
| Market      | 交易所代码    |
| Noorders    | 发布的委托笔数  |
| SecurityId  | 证券统一编码   |
| VOL         | 委托量      |
| 十档行情        |          |
| Symbol      | 证券代码     |
| TradingDate | 交易归属日期   |
| TradingTime | 交易发生时间   |
| LCP         | 昨收盘价     |
| OP          | 开盘价      |
| HIP         | 最高价      |
| LOP         | 最低价      |
| CP          | 现价       |
| TradeStatus | 当前品种状态   |
| SellLevelNO | 申卖价格档位   |
| S10         | 委卖价 10   |

|            |        |
|------------|--------|
| S9         | 委卖价 9  |
| S8         | 委卖价 8  |
| S7         | 委卖价 7  |
| S6         | 委卖价 6  |
| S5         | 委卖价 5  |
| S4         | 委卖价 4  |
| S3         | 委卖价 3  |
| S2         | 委卖价 2  |
| S1         | 委卖价 1  |
| B1         | 委买价 1  |
| B2         | 委买价 2  |
| B3         | 委买价 3  |
| B4         | 委买价 4  |
| B5         | 委买价 5  |
| B6         | 委买价 6  |
| B7         | 委买价 7  |
| B8         | 委买价 8  |
| B9         | 委买价 9  |
| B10        | 委买价 10 |
| BuyLevelNO | 申买报价档位 |
| SV10       | 委卖量 10 |
| SV9        | 委卖量 9  |
| SV8        | 委卖量 8  |
| SV7        | 委卖量 7  |
| SV6        | 委卖量 6  |
| SV5        | 委卖量 5  |
| SV4        | 委卖量 4  |
| SV3        | 委卖量 3  |
| SV2        | 委卖量 2  |
| SV1        | 委卖量 1  |
| BV1        | 委买量 1  |
| BV2        | 委买量 2  |
| BV3        | 委买量 3  |
| BV4        | 委买量 4  |
| BV5        | 委买量 5  |
| BV6        | 委买量 6  |
| BV7        | 委买量 7  |
| BV8        | 委买量 8  |
| BV9        | 委买量 9  |
| BV10       | 委买量 10 |
| TT         | 成交笔数   |
| TQ         | 成交总量   |
| TM         | 成交总额   |

|              |            |
|--------------|------------|
| TBOVOL       | 委托买入总量     |
| WAvgBP       | 加权平均委买价格   |
| BondWAvgBP   | 债券加权平均委买价格 |
| TSOVOL       | 委托卖出总量     |
| WAvgSP       | 加权平均委卖价格   |
| BondWAvgSP   | 债券加权平均委卖价格 |
| IOPV         | ETF 净值估计   |
| YTM          | 到期收益率      |
| UNIX         | UNIX 时间戳   |
| Market       | 交易所代码      |
| CP1          | 今日收盘价      |
| TBONo1       | 委买笔数 1     |
| TBONo2       | 委买笔数 2     |
| TBONo3       | 委买笔数 3     |
| TBONo4       | 委买笔数 4     |
| TBONo5       | 委买笔数 5     |
| TBONo6       | 委买笔数 6     |
| TBONo7       | 委买笔数 7     |
| TBONo8       | 委买笔数 8     |
| TBONo9       | 委买笔数 9     |
| TBONo10      | 委买笔数 10    |
| TSONo1       | 委卖笔数 1     |
| TSONo2       | 委卖笔数 2     |
| TSONo3       | 委卖笔数 3     |
| TSONo4       | 委卖笔数 4     |
| TSONo5       | 委卖笔数 5     |
| TSONo6       | 委卖笔数 6     |
| TSONo7       | 委卖笔数 7     |
| TSONo8       | 委卖笔数 8     |
| TSONo9       | 委卖笔数 9     |
| TSONo10      | 委卖笔数 10    |
| ETFBNo       | ETF 申购笔数   |
| ETFBVOL      | ETF 申购数量   |
| ETFBM        | ETF 申购金额   |
| ETFSNo       | ETF 赎回笔数   |
| ETFSVOL      | ETF 赎回数量   |
| ETFSM        | ETF 赎回金额   |
| WithdrawBNo  | 买入撤单笔数     |
| WithdrawBVOL | 买入撤单数量     |
| WithdrawBM   | 买入撤单金额     |
| WithdrawSNo  | 卖出撤单笔数     |
| WithdrawSVOL | 卖出撤单数量     |
| WithdrawSM   | 卖出撤单金额     |

|                    |              |
|--------------------|--------------|
| TotalBNo           | 买入总笔数        |
| TotalSNo           | 卖出总笔数        |
| MaxBD              | 买入委托成交最大等待时间 |
| MaxSD              | 卖出委托成交最大等待时间 |
| BONo               | 买方委托价位数      |
| SONo               | 卖方委托价位数      |
| SecurityId         | 证券统一编码       |
| <b>逐笔成交</b>        |              |
| Symbol             | 证券代码         |
| TradingDate        | 交易归属日期       |
| TradingTime        | 交易发生时间       |
| RecID              | 成交序号         |
| TradeChannel       | 成交通道         |
| TradePrice         | 成交价格         |
| TradeVolume        | 成交数量         |
| TradeAmount        | 成交金额         |
| UNIX               | UNIX 时间戳     |
| Market             | 市场           |
| BRecID             | 买方订单号        |
| SRecID             | 卖方订单号        |
| BSFlag             | 内外盘标志        |
| SecurityId         | 证券统一编码       |
| <b>静态数据</b>        |              |
| Symbol             | 证券代码         |
| SecurityName       | 证券名称         |
| TradingDate        | 交易归属日期       |
| TradingTime        | 数据生成时间       |
| SecurityEN         | 证券英文简称       |
| ISIN               | ISIN 代码      |
| SymbolUnderlying   | 基础证券代码       |
| LCP                | 昨收价          |
| MarketType         | 市场种类         |
| CFICode            | 证券类别         |
| SecuritySubType    | 证券子类别        |
| Currency           | 币种           |
| ParValue           | 面值           |
| TradableNo         | 可流通未上市数量     |
| EndDate            | 最后交易日期       |
| ListingDate        | 上市日期         |
| SetNo              | 产品集编号        |
| BuyVolumeUnit      | 买数量单位        |
| SellVolumeUnit     | 卖数量单位        |
| DeclareVolumeFloor | 申报量下限        |



|                      |         |
|----------------------|---------|
| DeclareVolumeCeiling | 申报量上限   |
| TickSize             | 最小报价单位  |
| UpDownLimitType      | 涨跌幅限制类型 |
| LIMITUP              | 涨幅价格    |
| LIMITDOWN            | 跌幅价格    |
| XRRatio              | 除权比例    |
| XDAmount             | 除息金额    |
| CrdbuyUnderlying     | 融资标的标志  |
| SecurityStatus       | 产品状态标识  |
| SampleNo             | 样本数量    |
| SampleAvgPrice       | 样本均价    |
| TradeAmount          | 成交金额    |
| AvgCapital           | 平均股本    |
| TotalMarketValue     | 总市值     |
| MarketValueRatio     | 占比%     |
| StaticPERatio        | 静态市盈率   |
| IndexLevelStatus     | 指数级别标识  |
| SecurityId           | 证券统一编码  |
| Market               | 市场      |

### 深圳证券交易所：

| 接口支持字段名称     | 描述       |
|--------------|----------|
| 指数行情         |          |
| Symbol       | 证券代码     |
| TradingDate  | 交易归属日期   |
| TradingTime  | 交易发生时间   |
| LCP          | 昨收价      |
| OP           | 开盘价      |
| HIP          | 最高价      |
| LOP          | 最低价      |
| CP           | 现价       |
| TQ           | 成交总量     |
| TT           | 成交笔数     |
| TM           | 成交金额     |
| UNIX         | UNIX 时间戳 |
| Market       | 市场       |
| SecurityId   | 证券统一编码   |
| Status       | 当前品种交易状态 |
| SampleNo     | 样本个数     |
| SymbolSource | 证券代码源    |
| 委托队列         |          |
| Symbol       | 证券代码     |

|             |             |
|-------------|-------------|
| TradingDate | 交易归属日期      |
| TradingTime | 交易发生时间      |
| CP          | 现价          |
| BS          | 买卖方向        |
| NPL         | 买卖盘价位数量     |
| Price       | 委托价格        |
| Numorders   | 最优价位上的总委托笔数 |
| OV1         | 委托量 01      |
| OV2         | 委托量 02      |
| OV3         | 委托量 03      |
| OV4         | 委托量 04      |
| OV5         | 委托量 05      |
| OV6         | 委托量 06      |
| OV7         | 委托量 07      |
| OV8         | 委托量 08      |
| OV9         | 委托量 09      |
| OV10        | 委托量 10      |
| OV11        | 委托量 11      |
| OV12        | 委托量 12      |
| OV13        | 委托量 13      |
| OV14        | 委托量 14      |
| OV15        | 委托量 15      |
| OV16        | 委托量 16      |
| OV17        | 委托量 17      |
| OV18        | 委托量 18      |
| OV19        | 委托量 19      |
| OV20        | 委托量 20      |
| OV21        | 委托量 21      |
| OV22        | 委托量 22      |
| OV23        | 委托量 23      |
| OV24        | 委托量 24      |
| OV25        | 委托量 25      |
| OV26        | 委托量 26      |
| OV27        | 委托量 27      |
| OV28        | 委托量 28      |
| OV29        | 委托量 29      |
| OV30        | 委托量 30      |
| OV31        | 委托量 31      |
| OV32        | 委托量 32      |
| OV33        | 委托量 33      |
| OV34        | 委托量 34      |
| OV35        | 委托量 35      |
| OV36        | 委托量 36      |

|             |          |
|-------------|----------|
| OV37        | 委托量 37   |
| OV38        | 委托量 38   |
| OV39        | 委托量 39   |
| OV40        | 委托量 40   |
| OV41        | 委托量 41   |
| OV42        | 委托量 42   |
| OV43        | 委托量 43   |
| OV44        | 委托量 44   |
| OV45        | 委托量 45   |
| OV46        | 委托量 46   |
| OV47        | 委托量 47   |
| OV48        | 委托量 48   |
| OV49        | 委托量 49   |
| OV50        | 委托量 50   |
| UNIX        | UNIX 时间戳 |
| Market      | 市场       |
| Noorders    | 揭示的委托笔数  |
| SecurityId  | 证券统一编码   |
| VOL         | 委托量      |
| 十档行情        |          |
| Symbol      | 证券代码     |
| TradingDate | 交易归属日期   |
| TradingTime | 交易发生时间   |
| LCP         | 昨收价      |
| OP          | 开盘价      |
| HIP         | 最高价      |
| LOP         | 最低价      |
| CP          | 现价       |
| TradeVolume | 分笔期间成交量  |
| TradeAmount | 分笔期间成交金额 |
| PE1         | 市盈率 1    |
| PE2         | 市盈率 2    |
| SellLevelNO | 卖盘价位数    |
| S10         | 申卖价十     |
| S9          | 申卖价九     |
| S8          | 申卖价八     |
| S7          | 申卖价七     |
| S6          | 申卖价六     |
| S5          | 申卖价五     |
| S4          | 申卖价四     |
| S3          | 申卖价三     |
| S2          | 申卖价二     |
| S1          | 申卖价一     |

|              |          |
|--------------|----------|
| B1           | 申买价一     |
| B2           | 申买价二     |
| B3           | 申买价三     |
| B4           | 申买价四     |
| B5           | 申买价五     |
| B6           | 申买价六     |
| B7           | 申买价七     |
| B8           | 申买价八     |
| B9           | 申买价九     |
| B10          | 申买价十     |
| BuyLevelNO   | 买盘价位数    |
| SV10         | 申卖量十     |
| SV9          | 申卖量九     |
| SV8          | 申卖量八     |
| SV7          | 申卖量七     |
| SV6          | 申卖量六     |
| SV5          | 申卖量五     |
| SV4          | 申卖量四     |
| SV3          | 申卖量三     |
| SV2          | 申卖量二     |
| SV1          | 申卖量一     |
| BV1          | 申买量一     |
| BV2          | 申买量二     |
| BV3          | 申买量三     |
| BV4          | 申买量四     |
| BV5          | 申买量五     |
| BV6          | 申买量六     |
| BV7          | 申买量七     |
| BV8          | 申买量八     |
| BV9          | 申买量九     |
| BV10         | 申买量十     |
| TT           | 成交笔数     |
| TQ           | 成交总量     |
| TM           | 成交总额     |
| TBOVOL       | 委托买入总量   |
| WAvgBP       | 加权平均买入价  |
| TSOVOL       | 委托卖出总量   |
| WAvgSP       | 加权平均委卖价  |
| UNIX         | UNIX 时间戳 |
| Market       | 交易所代码    |
| LIMITUP      | 涨停价      |
| LIMITDOWN    | 跌停价      |
| PriceUpdown1 | 升降一      |

|              |            |
|--------------|------------|
| PriceUpdown2 | 涨跌二        |
| TradeNo      | 分笔期间成交笔数   |
| CP1          | 今日收盘价      |
| Status       | 当前品种交易状态   |
| NAV          | 基金 T-1 日净值 |
| IOPV         | 基金实时参考净值   |
| PremiumRate  | 权证溢价率      |
| TBONo1       | 买入总委托笔数    |
| TBONo2       | 买入总委托笔数    |
| TBONo3       | 买入总委托笔数    |
| TBONo4       | 买入总委托笔数    |
| TBONo5       | 买入总委托笔数    |
| TBONo6       | 买入总委托笔数    |
| TBONo7       | 买入总委托笔数    |
| TBONo8       | 买入总委托笔数    |
| TBONo9       | 买入总委托笔数    |
| TBONo10      | 买入总委托笔数    |
| TSONo1       | 卖出总委托笔数    |
| TSONo2       | 卖出总委托笔数    |
| TSONo3       | 卖出总委托笔数    |
| TSONo4       | 卖出总委托笔数    |
| TSONo5       | 卖出总委托笔数    |
| TSONo6       | 卖出总委托笔数    |
| TSONo7       | 卖出总委托笔数    |
| TSONo8       | 卖出总委托笔数    |
| TSONo9       | 卖出总委托笔数    |
| TSONo10      | 卖出总委托笔数    |
| SymbolSource | 证券代码源      |
| SecurityId   | 证券统一编码     |
| 逐笔成交         |            |
| Symbol       | 证券代码       |
| TradingDate  | 交易归属日期     |
| TradingTime  | 成交时间       |
| BOID         | 买方委托索引     |
| SOID         | 卖方委托索引     |
| RecID        | 消息记录号      |
| SetID        | 频道代码       |
| TradePrice   | 成交价格       |
| TradeVolume  | 成交数量       |
| TradeType    | 成交类型       |
| UNIX         | UNIX 时间戳   |
| Market       | 市场         |
| SymbolSource | 证券代码源      |

|                            |              |
|----------------------------|--------------|
| SecurityId                 | 证券统一编码       |
| <b>静态数据</b>                |              |
| Symbol                     | 证券代码         |
| SecurityName               | 证券名称         |
| TradingDate                | 交易归属日期       |
| SymbolSource               | 证券代码源        |
| SecurityEN                 | 证券英文简称       |
| ISIN                       | ISIN 代码      |
| SymbolUnderlying           | 基础证券代码       |
| UnderlyingSecurityIDSource | 基础证券代码源      |
| SecurityType               | 证券类别代码       |
| SecurityStatusTag          | 证券状态标识       |
| LCP                        | 昨收价          |
| Currency                   | 币种           |
| ParValue                   | 每股面值         |
| IssuedVolume               | 总发行量         |
| OutstandingShare           | 流通股数         |
| IndustryType               | 行业种类         |
| PreYearEPS                 | 上年每股利润       |
| YearEPS                    | 本年每股利润       |
| OfferingFlag               | 收购(转股、行权) 标志 |
| NAV                        | 基金 T-1 日累计净值 |
| ListingDate                | 上市日期         |
| CouponRate                 | 票面利率         |
| IssuePrice                 | 贴现发行价        |
| Interest                   | 每百元应计利息      |
| InterestAccrualDate        | 起息日          |
| MaturityDate               | 到期交割日        |
| ConversionPrice            | 行权价格         |
| ConversionRatio            | 行权比例         |
| ConversionBeginDate        | 行权开始日        |
| ConversionEndDate          | 行权结束日        |
| CallOrPut                  | 认购认沽         |
| WarrantClearingType        | 权证结算方式       |
| ClearingPrice              | 结算价格         |
| OptionType                 | 行权类型         |
| EndDate                    | 最后交易日        |
| ExpirationDays             | 购回期限         |
| DayTrading                 | 回转交易标志       |
| GageFlag                   | 保证金证券标志      |
| GageRate                   | 担保品折算率       |
| CrdBuyUnderlying           | 融资标的标志       |
| CrdSellUnderlying          | 融券标的标志       |

|                        |                |
|------------------------|----------------|
| CrdPriceCheckType      | 提价检查方式         |
| PledgeFlag             | 质押入库标志         |
| ContractMultiplier     | 债券折合回购标准券比例    |
| RegularShare           | 对应回购标准券        |
| QualificationFlag      | 投资者适当性管理标志     |
| MarketMakerFlag        | 做市商标志          |
| RoundLot               | 整手数            |
| TickSize               | 最小报价单位         |
| BuyQtyUpperLimit       | 买数量上限          |
| SellQtyUpperLimit      | 卖数量上限          |
| BuyVolumeUnit          | 买数量单位          |
| SellVolumeUnit         | 卖数量单位          |
| LimitUpRateO           | 开盘集合竞价上涨幅度     |
| LimitDownRateO         | 开盘集合竞价下跌幅度     |
| LimitUpAbsoluteO       | 开盘集合竞价上涨限价     |
| LimitDownAbsoluteO     | 开盘集合竞价下跌限价     |
| AuctionUpDownRateO     | 开盘集合竞价有效范围涨跌幅度 |
| AuctionUpDownAbsoluteO | 开盘集合竞价有效范围涨跌价格 |
| LimitUpRateT           | 连续竞价上涨幅度       |
| LimitDownRateT         | 连续竞价下跌幅度       |
| LimitUpAbsoluteT       | 连续竞价上涨限价       |
| LimitDownAbsoluteT     | 连续竞价下跌限价       |
| AuctionUpDownRateT     | 连续竞价有效范围涨跌幅度   |
| AuctionUpDownAbsoluteT | 连续竞价有效范围涨跌价格   |
| LimitUpRateC           | 收盘集合竞价上涨幅度     |
| LimitDownRateC         | 收盘集合竞价下跌幅度     |
| LimitUpAbsoluteC       | 收盘集合竞价上涨限价     |
| LimitDownAbsoluteC     | 收盘集合竞价下跌限价     |
| AuctionUpDownRateC     | 收盘集合竞价有效范围涨跌幅度 |
| AuctionUpDownAbsoluteC | 收盘集合竞价有效范围涨跌价格 |
| SecurityId             | 证券统一编码         |
| Market                 | 市场             |
| 逐笔委托                   |                |
| Symbol                 | 证券代码           |
| TradingDate            | 交易归属日期         |
| TradingTime            | 委托时间           |
| SetID                  | 频道代码           |
| RecID                  | 消息记录号          |
| OrderPrice             | 委托价格           |
| OrderVOL               | 委托数量           |
| OrderType              | 订单类别           |
| UNIX                   | UNIX 时间戳       |
| Market                 | 市场             |

|                     |               |
|---------------------|---------------|
| SymbolSource        | 证券代码源         |
| SecurityId          | 证券统一编码        |
| OrderCode           | 委托买卖方向        |
| <b>证券状态</b>         |               |
| Symbol              | 证券代码          |
| TradingDate         | 交易归属日期        |
| TradingTime         | 交易发生时间        |
| CrdbuyStatus        | 当前融资开放状态      |
| CrdsellStatus       | 当前融券开放状态      |
| SubscribeStatus     | 申购标志          |
| RedemptionStatus    | 赎回标志          |
| UNIX                | UNIX 时间戳      |
| Market              | 市场            |
| SymbolSource        | 证券代码源         |
| FinancialStatus     | 证券状态信息        |
| PurchasingStatus    | 认购标志          |
| StockDiviStatus     | 转股标志          |
| PutableStatus       | 回售标志          |
| ExerciseStatus      | 行权标志          |
| GoldPurchase        | 黄金 ETF 实物申购标志 |
| GoldRedemption      | 黄金 ETF 实物赎回标志 |
| AcceptedStatus      | 预受要约标志        |
| ReleaseStatus       | 解除要约标志        |
| CancStockDiviStatus | 转股撤单标志        |
| CancPutableStatus   | 回售撤单标志        |
| PledgeStatus        | 质押标志          |
| RemovePledge        | 解押标志          |
| VoteStatus          | 表决权标志         |
| StockPledgeRepo     | 股票质押式回购标志     |
| DivideStatus        | 实时分拆标志        |
| MergerStatus        | 实时合并标志        |
| SecurityId          | 证券统一编码        |

### 大连商品期货交易所：

| 接口支持字段名称      | 描述     |
|---------------|--------|
| <b>套利深度行情</b> |        |
| Symbol        | 套利合约号  |
| SecurityId    | 证券统一编码 |
| TradingTime   | 交易发生时间 |
| CP            | 最新成交价  |
| LOP           | 最低价    |
| HIP           | 最高价    |



|               |          |
|---------------|----------|
| LifeLow       | 历史最低价    |
| LifeHigh      | 历史最高价    |
| LimitUp       | 涨停价      |
| LimitDown     | 跌停价      |
| Nolevel       | 深度行情数量   |
| B1            | 买价一      |
| B2            | 买价二      |
| B3            | 买价三      |
| B4            | 买价四      |
| B5            | 买价五      |
| BV1           | 买量一      |
| BV2           | 买量二      |
| BV3           | 买量三      |
| BV4           | 买量四      |
| BV5           | 买量五      |
| BidImPLYQty01 | 申买推导量 1  |
| BidImPLYQty02 | 申买推导量 2  |
| BidImPLYQty03 | 申买推导量 3  |
| BidImPLYQty04 | 申买推导量 4  |
| BidImPLYQty05 | 申买推导量 5  |
| SP1           | 卖价一      |
| SP2           | 卖价二      |
| SP3           | 卖价三      |
| SP4           | 卖价四      |
| SP5           | 卖价五      |
| SV1           | 卖量一      |
| SV2           | 卖量二      |
| SV3           | 卖量三      |
| SV4           | 卖量四      |
| SV5           | 卖量五      |
| AskImPLYQty01 | 申卖推导量 1  |
| AskImPLYQty02 | 申卖推导量 2  |
| AskImPLYQty03 | 申卖推导量 3  |
| AskImPLYQty04 | 申卖推导量 4  |
| AskImPLYQty05 | 申卖推导量 5  |
| BestBuyQty01  | 最优价委买量 1 |
| BestBuyQty02  | 最优价委买量 2 |
| BestBuyQty03  | 最优价委买量 3 |
| BestBuyQty04  | 最优价委买量 4 |
| BestBuyQty05  | 最优价委买量 5 |
| BestBuyQty06  | 最优价委买量 6 |
| BestBuyQty07  | 最优价委买量 7 |
| BestBuyQty08  | 最优价委买量 8 |

|                 |           |
|-----------------|-----------|
| BestBuyQty09    | 最优价委买量 9  |
| BestBuyQty10    | 最优价委买量 10 |
| BestSellQty01   | 最优价委卖量 1  |
| BestSellQty02   | 最优价委卖量 2  |
| BestSellQty03   | 最优价委卖量 3  |
| BestSellQty04   | 最优价委卖量 4  |
| BestSellQty05   | 最优价委卖量 5  |
| BestSellQty06   | 最优价委卖量 6  |
| BestSellQty07   | 最优价委卖量 7  |
| BestSellQty08   | 最优价委卖量 8  |
| BestSellQty09   | 最优价委卖量 9  |
| BestSellQty10   | 最优价委卖量 10 |
| TradingDate     | 交易归属日期    |
| Market          | 市场        |
| UNIX            | UNIX 时间   |
| 分价成交量行情         |           |
| Symbol          | 合约代码      |
| SecurityId      | 证券统一编码    |
| TradingTime     | 交易时间      |
| PriceOne        | 价格 1      |
| PriceOneBOQTY   | 买开数量 1    |
| PriceOneBEQTY   | 买平数量 1    |
| PriceOneSOQTY   | 卖开数量 1    |
| PriceOneSEQTY   | 卖平数量 1    |
| PriceTwo        | 价格 2      |
| PriceTwoBOQTY   | 买开数量 2    |
| PriceTwoBEQTY   | 买平数量 2    |
| PriceTwoSOQTY   | 卖开数量 2    |
| PriceTwoSEQTY   | 卖平数量 2    |
| PriceThree      | 价格 3      |
| PriceThreeBOQTY | 买开数量 3    |
| PriceThreeBEQTY | 买平数量 3    |
| PriceThreeSOQTY | 卖开数量 3    |
| PriceThreeSEQTY | 卖平数量 3    |
| PriceFour       | 价格 4      |
| PriceFourBOQTY  | 买开数量 4    |
| PriceFourBEQTY  | 买平数量 4    |
| PriceFourSOQTY  | 卖开数量 4    |
| PriceFourSEQTY  | 卖平数量 4    |
| PriceFive       | 价格 5      |
| PriceFiveBOQTY  | 买开数量 5    |
| PriceFiveBEQTY  | 买平数量 5    |
| PriceFiveSOQTY  | 卖开数量 5    |

|                  |          |
|------------------|----------|
| PriceFiveSEQTY   | 卖平数量 5   |
| TradingDate      | 交易归属日期   |
| Market           | 市场       |
| UNIX             | UNIX 时间戳 |
| 委托统计行情           |          |
| Symbol           | 证券代码     |
| SecurityId       | 证券统一编码   |
| TradingTime      | 交易时间     |
| TBOVOL           | 委托买入总量   |
| WAvgBP           | 加权平均委买价格 |
| TSOVOL           | 委托卖出总量   |
| WAvgSP           | 加权平均委卖价格 |
| TradingDate      | 交易归属日期   |
| Market           | 市场       |
| UNIX             | UNIX 时间戳 |
| 实时结算行情           |          |
| Symbol           | 证券代码     |
| SecurityId       | 证券统一编码   |
| TradingTime      | 交易时间     |
| RealSP           | 实时结算价    |
| TradingDate      | 交易归属日期   |
| Market           | 市场       |
| UNIX             | UNIX 时间戳 |
| 分笔高频             |          |
| Symbol           | 证券代码     |
| ShortName        | 合约简称     |
| SecurityId       | 证券统一编码   |
| TradingTime      | 交易时间     |
| OP               | 开盘价      |
| CP               | 最新价      |
| HIP              | 最高价      |
| LOP              | 最低价      |
| AvgPrice         | 当日均价     |
| SettlePrice      | 今结算价     |
| PreSettlePrice   | 前结算价     |
| CP1              | 收盘价      |
| LCP              | 昨收盘价     |
| LastVolume       | 当前成交量    |
| TradeVolume      | 分笔期间成交量  |
| TQ               | 成交总量     |
| TradeAmount      | 分笔期间成交金额 |
| TM               | 成交总金额    |
| PreTotalPosition | 昨持仓量     |

|                   |          |
|-------------------|----------|
| TotalPosition     | 持仓量      |
| PrePositionChange | 日增仓      |
| LifeHigh          | 开盘至今最高价  |
| LifeLow           | 开盘至今最低价  |
| LIMITUP           | 涨停价      |
| LIMITDOWN         | 跌停价      |
| Nolevel           | 深度行情数量   |
| BS                | 买卖方向     |
| B1                | 买价一      |
| B2                | 买价二      |
| B3                | 买价三      |
| B4                | 买价四      |
| B5                | 买价五      |
| BV1               | 买量一      |
| BV2               | 买量二      |
| BV3               | 买量三      |
| BV4               | 买量四      |
| BV5               | 买量五      |
| BidImPLYQty01     | 申买推导量 1  |
| BidImPLYQty02     | 申买推导量 2  |
| BidImPLYQty03     | 申买推导量 3  |
| BidImPLYQty04     | 申买推导量 4  |
| BidImPLYQty05     | 申买推导量 5  |
| S1                | 卖价一      |
| S2                | 卖价二      |
| S3                | 卖价三      |
| S4                | 卖价四      |
| S5                | 卖价五      |
| SV1               | 卖量一      |
| SV2               | 卖量二      |
| SV3               | 卖量三      |
| SV4               | 卖量四      |
| SV5               | 卖量五      |
| AskImPLYQty01     | 申卖推导量 1  |
| AskImPLYQty02     | 申卖推导量 2  |
| AskImPLYQty03     | 申卖推导量 3  |
| AskImPLYQty04     | 申卖推导量 4  |
| AskImPLYQty05     | 申卖推导量 5  |
| BestBuyQty01      | 最优价委买量 1 |
| BestBuyQty02      | 最优价委买量 2 |
| BestBuyQty03      | 最优价委买量 3 |
| BestBuyQty04      | 最优价委买量 4 |
| BestBuyQty05      | 最优价委买量 5 |

|               |           |
|---------------|-----------|
| BestBuyQty06  | 最优价委买量 6  |
| BestBuyQty07  | 最优价委买量 7  |
| BestBuyQty08  | 最优价委买量 8  |
| BestBuyQty09  | 最优价委买量 9  |
| BestBuyQty10  | 最优价委买量 10 |
| BestSellQty01 | 最优价委卖量 1  |
| BestSellQty02 | 最优价委卖量 2  |
| BestSellQty03 | 最优价委卖量 3  |
| BestSellQty04 | 最优价委卖量 4  |
| BestSellQty05 | 最优价委卖量 5  |
| BestSellQty06 | 最优价委卖量 6  |
| BestSellQty07 | 最优价委卖量 7  |
| BestSellQty08 | 最优价委卖量 8  |
| BestSellQty09 | 最优价委卖量 9  |
| BestSellQty10 | 最优价委卖量 10 |
| ContinueSign  | 连续合约代码    |
| TradingDate   | 交易归属日期    |
| Market        | 市场        |
| UNIX          | UNIX 时间   |

### 大商所商品期权：

| 接口支持字段          | 描述     |
|-----------------|--------|
| 分价成交量行情         |        |
| Symbol          | 合约代码   |
| SecurityId      | 证券统一编码 |
| TradingTime     | 交易时间   |
| PriceOne        | 价格 1   |
| PriceOneBOQTY   | 买开数量 1 |
| PriceOneBEQTY   | 买平数量 1 |
| PriceOneSOQTY   | 卖开数量 1 |
| PriceOneSEQTY   | 卖平数量 1 |
| PriceTwo        | 价格 2   |
| PriceTwoBOQTY   | 买开数量 2 |
| PriceTwoBEQTY   | 买平数量 2 |
| PriceTwoSOQTY   | 卖开数量 2 |
| PriceTwoSEQTY   | 卖平数量 2 |
| PriceThree      | 价格 3   |
| PriceThreeBOQTY | 买开数量 3 |
| PriceThreeBEQTY | 买平数量 3 |
| PriceThreeSOQTY | 卖开数量 3 |
| PriceThreeSEQTY | 卖平数量 3 |
| PriceFour       | 价格 4   |

|                |          |
|----------------|----------|
| PriceFourBOQTY | 买开数量 4   |
| PriceFourBEQTY | 买平数量 4   |
| PriceFourSOQTY | 卖开数量 4   |
| PriceFourSEQTY | 卖平数量 4   |
| PriceFive      | 价格 5     |
| PriceFiveBOQTY | 买开数量 5   |
| PriceFiveBEQTY | 买平数量 5   |
| PriceFiveSOQTY | 卖开数量 5   |
| PriceFiveSEQTY | 卖平数量 5   |
| TradingDate    | 交易归属日期   |
| Market         | 市场       |
| UNIX           | UNIX 时间戳 |
| <b>委托统计行情</b>  |          |
| Symbol         | 证券代码     |
| SecurityId     | 证券统一编码   |
| TradingTime    | 交易时间     |
| TBOVOL         | 委托买入总量   |
| WAvgBP         | 加权平均委买价格 |
| TSOVOL         | 委托卖出总量   |
| WAvgSP         | 加权平均委卖价格 |
| TradingDate    | 交易归属日期   |
| Market         | 市场       |
| UNIX           | UNIX 时间戳 |
| <b>实时结算行情</b>  |          |
| Symbol         | 证券代码     |
| SecurityId     | 证券统一编码   |
| TradingTime    | 交易时间     |
| RealSP         | 实时结算价    |
| TradingDate    | 交易归属日期   |
| Market         | 市场       |
| UNIX           | UNIX 时间戳 |
| <b>分笔高频</b>    |          |
| Symbol         | 证券代码     |
| ShortName      | 合约简称     |
| SecurityId     | 证券统一编码   |
| TradingTime    | 交易时间     |
| OP             | 开盘价      |
| CP             | 最新价      |
| HIP            | 最高价      |
| LOP            | 最低价      |
| AvgPrice       | 当日均价     |
| SettlePrice    | 今结算价     |
| PreSettlePrice | 前结算价     |

|                   |          |
|-------------------|----------|
| CP1               | 收盘价      |
| LCP               | 昨收盘价     |
| LastVolume        | 当前成交量    |
| TradeVolume       | 分笔期间成交量  |
| TQ                | 成交总量     |
| TradeAmount       | 分笔期间成交金额 |
| TM                | 成交总金额    |
| PreTotalPosition  | 昨持仓量     |
| TotalPosition     | 持仓量      |
| PrePositionChange | 日增仓      |
| LifeHigh          | 开盘至今最高价  |
| LifeLow           | 开盘至今最低价  |
| LIMITUP           | 涨停价      |
| LIMITDOWN         | 跌停价      |
| Nolevel           | 深度行情数量   |
| BS                | 买卖方向     |
| B1                | 买价一      |
| B2                | 买价二      |
| B3                | 买价三      |
| B4                | 买价四      |
| B5                | 买价五      |
| BV1               | 买量一      |
| BV2               | 买量二      |
| BV3               | 买量三      |
| BV4               | 买量四      |
| BV5               | 买量五      |
| BidImPLYQty01     | 申买推导量 1  |
| BidImPLYQty02     | 申买推导量 2  |
| BidImPLYQty03     | 申买推导量 3  |
| BidImPLYQty04     | 申买推导量 4  |
| BidImPLYQty05     | 申买推导量 5  |
| S1                | 卖价一      |
| S2                | 卖价二      |
| S3                | 卖价三      |
| S4                | 卖价四      |
| S5                | 卖价五      |
| SV1               | 卖量一      |
| SV2               | 卖量二      |
| SV3               | 卖量三      |
| SV4               | 卖量四      |
| SV5               | 卖量五      |
| AskImPLYQty01     | 申卖推导量 1  |
| AskImPLYQty02     | 申卖推导量 2  |

|               |           |
|---------------|-----------|
| AskImPLYQty03 | 申卖推导量 3   |
| AskImPLYQty04 | 申卖推导量 4   |
| AskImPLYQty05 | 申卖推导量 5   |
| BestBuyQty01  | 最优价委买量 1  |
| BestBuyQty02  | 最优价委买量 2  |
| BestBuyQty03  | 最优价委买量 3  |
| BestBuyQty04  | 最优价委买量 4  |
| BestBuyQty05  | 最优价委买量 5  |
| BestBuyQty06  | 最优价委买量 6  |
| BestBuyQty07  | 最优价委买量 7  |
| BestBuyQty08  | 最优价委买量 8  |
| BestBuyQty09  | 最优价委买量 9  |
| BestBuyQty10  | 最优价委买量 10 |
| BestSellQty01 | 最优价委卖量 1  |
| BestSellQty02 | 最优价委卖量 2  |
| BestSellQty03 | 最优价委卖量 3  |
| BestSellQty04 | 最优价委卖量 4  |
| BestSellQty05 | 最优价委卖量 5  |
| BestSellQty06 | 最优价委卖量 6  |
| BestSellQty07 | 最优价委卖量 7  |
| BestSellQty08 | 最优价委卖量 8  |
| BestSellQty09 | 最优价委卖量 9  |
| BestSellQty10 | 最优价委卖量 10 |
| ContinueSign  | 连续合约代码    |
| TradingDate   | 交易归属日期    |
| Market        | 市场        |
| UNIX          | UNIX 时间   |
| Delta         | Delta     |
| Gamma         | Gamma     |
| Rho           | Rho       |
| Theta         | Theta     |
| Vega          | Vega      |

## 4.12 [GetFinance](#) 入参 fields 及出参返回的列式结果集字段

公共信息表:

| 接口支持字段        | 描述     |
|---------------|--------|
| Symbol        | 证券代码   |
| StateTypeCode | 报表类型编码 |
| StartDate     | 报告起始时间 |



|            |        |
|------------|--------|
| EndDate    | 报告截止日期 |
| ActRelDate | 实际披露日期 |

**利润表：**

| 接口支持字段     | 描述                 |
|------------|--------------------|
| BEPS       | 基本每股收益             |
| BME        | 业务及管理费             |
| CLMEXP     | 赔付支出               |
| CLMRCVR    | 减：摊回赔付支出           |
| COMPROAMS  | 归属于少数股东的综合收益       |
| COMPROAOPC | 归属于母公司所有者的综合收益     |
| DEPS       | 稀释每股收益             |
| EXPRCRI    | 减：摊回分保费用           |
| EXPRISAC   | 分保费用               |
| FCEXP      | 手续费及佣金支出           |
| FCINC      | 手续费及佣金收入           |
| FOREXG     | 汇兑收益               |
| GADMEXP    | 管理费用               |
| GLFVC      | 公允价值变动收益           |
| IGAJV      | 其中：对联营企业和合营企业的投资收益 |
| IMPLOS     | 资产减值损失             |
| INCTEXP    | 所得税费用              |
| INSRRR     | 减：摊回保险责任准备金        |
| INSUINC    | 保险业务收入             |
| INTEXP     | 利息支出               |
| INTINC     | 利息收入               |
| INVSTG     | 投资收益               |
| MININT     | 少数股东损益             |
| NCLMEXP    | 赔付支出净额             |
| NEPRMM     | 已赚保费               |
| NFCINC     | 手续费及佣金净收入          |
| NIAMC      | 其中：受托客户资产管理业务净收入   |
| NINCSB     | 其中：代理买卖证券业务净收入     |
| NINCSU     | 其中：证券承销业务净收入       |
| NINTINC    | 利息净收入              |
| NLDNCA     | 其中：非流动资产处置净损益      |
| NOPEXP     | 营业外支出              |
| NOPINC     | 营业外收入              |
| NPAOPC     | 归属于母公司所有者的净利润      |
| NPIR       | 提取保险责任准备金净额        |
| OPEXP      | 营业成本               |
| OPPRO      | 营业利润               |

|           |              |
|-----------|--------------|
| OPREV     | 营业收入         |
| OTHCOMPRO | 其他综合收益(损失)   |
| OTHIANP   | 影响净利润的其他项目   |
| OTHOINC   | 其他业务收入       |
| OTHOPEXP  | 其他业务成本       |
| OTHOPPRO  | 其他业务利润       |
| PAYSRD    | 退保金          |
| PLCDIV    | 保单红利支出       |
| PRECR     | 减：分出保费       |
| PROIR     | 提取保险责任准备金    |
| PROUEPR   | 减：提取未到期责任准备金 |
| RINSPINC  | 其中：分保费收入     |
| SELLEXP   | 销售费用         |
| TAFO      | 营业税金及附加      |
| TCOMINCO  | 综合收益总额       |
| TOLPRO    | 利润总额         |
| TOTOEXP   | 营业总成本        |
| TOTOR     | 营业总收入        |
| UNRIL     | 未确认的投资损失     |

### 利润表 TTM:

| 接口支持字段     | 描述                 |
|------------|--------------------|
| BEPS       | 基本每股收益             |
| BME        | 业务及管理费             |
| CLMEXP     | 赔付支出               |
| CLMRCVR    | 减：摊回赔付支出           |
| COMPROAMS  | 归属于少数股东的综合收益       |
| COMPROAOPC | 归属于母公司所有者的综合收益     |
| DEPS       | 稀释每股收益             |
| EXPRCRI    | 减：摊回分保费用           |
| EXPRISAC   | 分保费用               |
| FCEXP      | 手续费及佣金支出           |
| FCINC      | 手续费及佣金收入           |
| FOREXG     | 汇兑收益               |
| GADMEXP    | 管理费用               |
| GLFVC      | 公允价值变动收益           |
| IGAJV      | 其中：对联营企业和合营企业的投资收益 |
| IMPLOS     | 资产减值损失             |
| INCTEXP    | 所得税费用              |
| INSRRR     | 减：摊回保险责任准备金        |
| INSUINC    | 保险业务收入             |
| INTEXP     | 利息支出               |

|           |                  |
|-----------|------------------|
| INTINC    | 利息收入             |
| INVSTG    | 投资收益             |
| MININT    | 少数股东损益           |
| NCLMEXP   | 赔付支出净额           |
| NEPRMM    | 已赚保费             |
| NFCINC    | 手续费及佣金净收入        |
| NIAMC     | 其中：受托客户资产管理业务净收入 |
| NINCSB    | 其中：代理买卖证券业务净收入   |
| NINCSU    | 其中：证券承销业务净收入     |
| NINTINC   | 利息净收入            |
| NLDNCA    | 其中：非流动资产处置净损益    |
| NOEXP     | 营业外支出            |
| NOPINC    | 营业外收入            |
| NPAOPC    | 归属于母公司所有者的净利润    |
| NPIR      | 提取保险责任准备金净额      |
| OPEXP     | 营业成本             |
| OPPRO     | 营业利润             |
| OPREV     | 营业收入             |
| OTHCOMPRO | 其他综合收益(损失)       |
| OTHIANP   | 影响净利润的其他项目       |
| OTHOINC   | 其他业务收入           |
| OTHOPEXP  | 其他业务成本           |
| OTHOPPRO  | 其他业务利润           |
| PAYSRD    | 退保金              |
| PLCDIV    | 保单红利支出           |
| PRECR     | 减：分出保费           |
| PROIR     | 提取保险责任准备金        |
| PROUEPR   | 减：提取未到期责任准备金     |
| RINSPINC  | 其中：分保费收入         |
| SELLEXP   | 销售费用             |
| TAFO      | 营业税金及附加          |
| TCOMINCO  | 综合收益总额           |
| TOLPRO    | 利润总额             |
| TOTOEXP   | 营业总成本            |
| TOTOR     | 营业总收入            |
| UNRIL     | 未确认的投资损失         |

#### 资产负债表：

| 接口支持字段      | 描述         |
|-------------|------------|
| ADVREC      | 预收款项       |
| ACCTPAYABLE | 应付账款       |
| FAPUATS     | 买入返售金融资产净额 |

|                         |                  |
|-------------------------|------------------|
| ASUATR                  | 卖出回购金融资产款        |
| AVFSFINASSET            | 可供出售金融资产净额       |
| BORRFROMCENTRALBANK     | 向中央银行借款          |
| BONDPAYABLE             | 应付债券             |
| CASHASSET               | 货币资金             |
| CASHBALWCENTRALBANK     | 现金及存放中央银行款项      |
| CLAIMPAYABLE            | 应付赔付款            |
| STATUTORYDEPOSITS       | 存出资本保证金          |
| CONSTMATERIAL           | 工程物资             |
| CPAITALRESERVES         | 资本公积             |
| ESTLIAB                 | 预计负债             |
| DUETOBAOFI              | 其中：同业及其他金融机构存放款项 |
| BALWFININS              | 存放同业款项           |
| CBDEPANDDUETOBAOFI      | 吸收存款及同业存放        |
| DEFTAXLIAB              | 递延所得税负债          |
| DEFINCOMETAXASSET       | 递延所得税资产          |
| DIVPAYABLE              | 应付股利             |
| CUSTOMERDEPOSITS        | 其中:客户资金存款        |
| FINASSETFORDERIV        | 衍生金融资产           |
| DEVFINLIAB              | 衍生金融负债           |
| DISPOSALOFFIXEDASSET    | 固定资产清理           |
| CBDEPOSIT               | 其中：吸收存款          |
| DEVEXP                  | 开发支出             |
| STAFFREMUPAYABLE        | 应付职工薪酬           |
| COMMISSIONPAYABLE       | 应付手续费及佣金         |
| REFUNDABLEDEPOSITS      | 存出保证金            |
| FORECURTRANSDIFF        | 外币报表折算差额         |
| GENERALRISKPROVISION    | 一般风险准备           |
| HTMINVT                 | 持有至到期投资净额        |
| INTPAYABLE              | 应付利息             |
| LTDEBT                  | 长期借款             |
| POLIRESFORLTHEALTHINSUR | 其中:长期健康险责任准备金    |
| POLIRESFORLIFEINSUR     | 其中:寿险责任准备金       |
| LTPAYABLE               | 长期应付款            |
| LTPREPAIDEXP            | 长期待摊费用           |
| MINSHINT                | 少数股东权益           |
| PLEDGEBORROWING         | 其中:质押借款          |
| ACCTREC                 | 应收账款净额           |
| PRODBIOASSET            | 生产性生物资产净额        |
| NONCURASSETS DUE ONEYR  | 一年内到期的非流动资产      |
| NCURLIAB DUE ONEYR      | 一年内到期的非流动负债      |
| UNDERCONSTPROJECT       | 在建工程净额           |
| DIVREC                  | 应收股利净额           |

|                      |                     |
|----------------------|---------------------|
| FIXEDASSET           | 固定资产净额              |
| GOODWILL             | 商誉净额                |
| INTANGIBLEASSET      | 无形资产净额              |
| INVTPROPERTY         | 投资性房地产净额            |
| INVENTORY            | 存货净额                |
| INTREC               | 应收利息净额              |
| LTREC                | 长期应收款净额             |
| LTDEBTINVT           | 长期债权投资净额            |
| LTEQUITYINVT         | 长期股权投资净额            |
| LTINVT               | 长期投资净额              |
| LOANANDADVANCES      | 发放贷款及垫款净额           |
| NOTESREC             | 应收票据净额              |
| OILGASASSET          | 油气资产净额              |
| OTHERREC             | 其他应收款净额             |
| NOTESPAYABLE         | 应付票据                |
| NETFUNDSLENDING      | 拆出资金净额              |
| PREPAY               | 预付款项净额              |
| INSPREMIUMREC        | 应收保费净额              |
| REINSREC             | 应收分保赔款净额            |
| REINSCONTRESERVESREC | 应收分保合同准备金净额         |
| SUBROGATIONREC       | 应收代位追偿款净额           |
| STINVT               | 短期投资净额              |
| OTHERASSET           | 其他资产                |
| OTHERCURLIAB         | 其他流动负债              |
| OTHERCURASSETS       | 其他流动资产              |
| OTHERLIAB            | 其他负债                |
| OTHERNCURASSET       | 其他非流动资产             |
| OTHERNCURLIAB        | 其他非流动负债             |
| OTHERPAYABLE         | 其他应付款               |
| SPECIALPAYABLE       | 专项应付款               |
| PLACEMENTFROMBAOFI   | 拆入资金                |
| NIIDFP               | 保户储金及投资款            |
| POLIDIVPAYABLE       | 应付保单红利              |
| POLIPLEDGELOAN       | 保户质押贷款净额            |
| NOBELMETAL           | 贵金属                 |
| SECBROKDEPO          | 代理承销证券款             |
| PREMIUMRECINADV      | 预收保费                |
| CLAIMRESERVES        | 其中:未决赔款准备金          |
| CRRFR                | 其中:应收分保未决赔款准备金净额    |
| LTRFHIRFR            | 其中:应收分保长期健康险责任准备金净额 |
| PRRLIRFR             | 其中:应收分保寿险责任准备金净额    |
| UPRRFR               | 其中:应收分保未到期责任准备金净额   |

|                       |               |
|-----------------------|---------------|
| RETAINPROFIT          | 未分配利润         |
| RECFROMREINSUR        | 应付分保账款        |
| INSURCONTRESERVES     | 保险合同准备金       |
| CUSTOMERBROKDEPO      | 代理买卖证券款       |
| SEPACCOUNTASSET       | 独立账户资产        |
| SEPACCTLIAB           | 独立账户负债        |
| PAIDUPCAPITAL         | 实收资本(或股本)     |
| STBORROWING           | 短期借款          |
| BALWCLEARCO           | 结算备付金         |
| CUSTOMEREXRESERVES    | 其中：客户备付金      |
| SURPLUSRESERVES       | 盈余公积          |
| TOEATP                | 归属于母公司所有者权益合计 |
| TERMDEPOSITS          | 定期存款          |
| TOTALASSET            | 资产总计          |
| TOTALCURASSETS        | 流动资产合计        |
| TOTALCURLIAB          | 流动负债合计        |
| TOTALLIAB             | 负债合计          |
| TOTALTLIAB            | 长期负债合计        |
| LIABANDOWNEREQUITY    | 负债与所有者权益总计    |
| TOTALNCURASSET        | 非流动资产合计       |
| TOTALNCURLIAB         | 非流动负债合计       |
| TOTALSHINT            | 所有者权益合计       |
| TAXPAYABLE            | 应交税费          |
| FINASSETFORTRADING    | 交易性金融资产       |
| HFTFINLIAB            | 交易性金融负债       |
| MEMBERSHIPFEE         | 其中:交易席位费      |
| TREASURYSTOCK         | 其中：库存股        |
| UNEARNPREMIUMRESERVES | 其中:未到期责任准备金   |
| UNCONFIRMINVTLOSS     | 未确认的投资损失      |

### 现金流量表直接法 TTM:

| 接口支持字段   | 描述                      |
|----------|-------------------------|
| BBCCE    | 期初现金及现金等价物余额            |
| CLMPAID  | 支付原保险合同赔付款项的现金          |
| CPACFILA | 购建固定资产、无形资产和其他长期资产支付的现金 |
| CPDDPI   | 分配股利、利润或偿付利息支付的现金       |
| CPEMP    | 支付给职工以及为职工支付的现金         |
| CPGS     | 购买商品、接受劳务支付的现金          |
| CPINVST  | 投资支付的现金                 |
| CPIS     | 吸收权益性投资收到的现金            |
| CPISMSS  | 其中：子公司吸收少数股东投资收到的现      |

|          |                           |
|----------|---------------------------|
|          | 金                         |
| CRAI     | 吸收投资收到的现金                 |
| CRDRI    | 收回投资收到的现金                 |
| CRDTFA   | 处置交易性金融资产净增加额             |
| CREPBOR  | 偿还债务支付的现金                 |
| CRR1     | 取得投资收益收到的现金               |
| CRSGRS   | 销售商品、提供劳务收到的现金            |
| DPDMSS   | 其中：子公司支付给少数股东的股利、利润       |
| EBCCE    | 期末现金及现金等价物余额              |
| EFERCC   | 汇率变动对现金及现金等价物的影响          |
| EOIC     | 其他对现金的影响                  |
| ICRPLCL  | 质押贷款净增加额                  |
| IFCP     | 支付利息、手续费及佣金的现金            |
| IFCR     | 收取利息、手续费及佣金的现金            |
| INCLOAC  | 客户贷款及垫款净增加额               |
| INCREREP | 回购业务资金净增加额                |
| IPBOFI   | 拆入资金净增加额                  |
| IPDI     | 保户储金及投资款净增加额              |
| IPOFI    | 向其他金融机构拆入资金净增加额           |
| NCBCB    | 向中央银行借款净增加额               |
| NCFFA    | 筹资活动产生的现金流量净额             |
| NCFIA    | 投资活动产生的现金流量净额             |
| NCFOP    | 经营活动产生的现金流量净额             |
| NCPASOBU | 取得子公司及其他营业单位支付的现金净额       |
| NCRDFILA | 处置固定资产、无形资产和其他长期资产收回的现金净额 |
| NCRDSOBU | 处置子公司及其他营业单位收到的现金净额       |
| NCRR     | 收到再保险业务现金净额               |
| NICDDBFI | 客户存款和同业存放款项净增加额           |
| NIDCBFI  | 存放中央银行和同业款项净增加额           |
| OCPRFA   | 支付其他与筹资活动有关的现金            |
| OCPRIA   | 支付其他与投资活动有关的现金            |
| OCPROA   | 支付其他与经营活动有关的现金            |
| OCRRIA   | 收到的其他与投资活动有关的现金           |
| OCRROA   | 收到的其他与经营活动有关的现金           |
| OPRFA    | 收到其他与筹资活动有关的现金            |
| PLCDIVP  | 支付保单红利的现金                 |
| PRMMRCV  | 收到原保险合同保费取得的现金            |
| PROBOR   | 取得借款收到的现金                 |
| PROISBD  | 发行债券收到的现金                 |

|          |         |
|----------|---------|
| TAXREF   | 收到的税费返还 |
| VTAXPAID | 支付的各项税费 |

### 现金流量表（直接法）：

| 接口支持字段   | 描述                      |
|----------|-------------------------|
| BBCCE    | 期初现金及现金等价物余额            |
| CLMPAID  | 支付原保险合同赔付款项的现金          |
| CPACFILA | 购建固定资产、无形资产和其他长期资产支付的现金 |
| CPDDPI   | 分配股利、利润或偿付利息支付的现金       |
| CPEMP    | 支付给职工以及为职工支付的现金         |
| CPGS     | 购买商品、接受劳务支付的现金          |
| CPINVST  | 投资支付的现金                 |
| CPIS     | 吸收权益性投资收到的现金            |
| CPISMSS  | 其中：子公司吸收少数股东投资收到的现金     |
| CRAI     | 吸收投资收到的现金               |
| CRDRI    | 收回投资收到的现金               |
| CRDTFA   | 处置交易性金融资产净增加额           |
| CREPBOR  | 偿还债务支付的现金               |
| CRR1     | 取得投资收益收到的现金             |
| CRSGRS   | 销售商品、提供劳务收到的现金          |
| DPDMSS   | 其中：子公司支付给少数股东的股利、利润     |
| EBCCE    | 期末现金及现金等价物余额            |
| EFERCC   | 汇率变动对现金及现金等价物的影响        |
| EOIC     | 其他对现金的影响                |
| ICRPLCL  | 质押贷款净增加额                |
| IFCP     | 支付利息、手续费及佣金的现金          |
| IFCR     | 收取利息、手续费及佣金的现金          |
| INCLOAC  | 客户贷款及垫款净增加额             |
| INCREREP | 回购业务资金净增加额              |
| IPBOFI   | 拆入资金净增加额                |
| IPDI     | 保户储金及投资款净增加额            |
| IPOFI    | 向其他金融机构拆入资金净增加额         |
| NCBCB    | 向中央银行借款净增加额             |
| NCFFA    | 筹资活动产生的现金流量净额           |
| NCFIA    | 投资活动产生的现金流量净额           |
| NCFOP    | 经营活动产生的现金流量净额           |
| NCPASOBU | 取得子公司及其他营业单位支付的现金净额     |
| NCRDFILA | 处置固定资产、无形资产和其他长期资产      |



|          |                     |
|----------|---------------------|
|          | 收回的现金净额             |
| NCRDSOBU | 处置子公司及其他营业单位收到的现金净额 |
| NCRR     | 收到再保险业务现金净额         |
| NICDDBFI | 客户存款和同业存放款项净增加额     |
| NIDCBFI  | 存放中央银行和同业款项净增加额     |
| OCPRFA   | 支付其他与筹资活动有关的现金      |
| OCPRIA   | 支付其他与投资活动有关的现金      |
| OCPROA   | 支付其他与经营活动有关的现金      |
| OCRRIA   | 收到的其他与投资活动有关的现金     |
| OCRROA   | 收到的其他与经营活动有关的现金     |
| OPRFA    | 收到其他与筹资活动有关的现金      |
| PLCDIVP  | 支付保单红利的现金           |
| PRMMRCV  | 收到原保险合同保费取得的现金      |
| PROBOR   | 取得借款收到的现金           |
| PROISBD  | 发行债券收到的现金           |
| TAXREF   | 收到的税费返还             |
| VTAXPAID | 支付的各项税费             |

### 现金流量表间接法 TTM:

| 接口支持字段   | 描述                               |
|----------|----------------------------------|
| AMORINTA | 无形资产摊销                           |
| AMORLPE  | 长期待摊费用摊销                         |
| BEGINBC  | 现金的期初余额                          |
| BEGINBCE | 现金等价物的期初余额                       |
| CBDOY    | 一年内到期的可转换公司债券                    |
| DDITA    | 递延所得税资产减少(增加以“一”号填列)             |
| DECINV   | 存货的减少(增加以“一”号填列)                 |
| DECOPREC | 经营性应收项目的减少(增加以“一”号填列)            |
| DFOGBBA  | 固定资产折旧 油气资产折耗 生产性生物资产折旧          |
| ENDBC    | 现金的期末余额                          |
| ENDBCE   | 现金等价物的期末余额                       |
| FAFL     | 融资租赁固定资产                         |
| GLCFV    | 公允价值变动损失(收益以“一”号填列)              |
| IDITL    | 递延所得税负债增加(减少以“一”号填列)             |
| INCOPPAY | 经营性应付项目的增加(减少以“一”号填列)            |
| LDFIOLA  | 处置固定资产 无形资产和其他长期资产的损失(收益以“一”号填列) |
| LIACVCAP | 债务转为资本                           |

|          |                     |
|----------|---------------------|
| LOSINVST | 投资损失(收益以“—”号填列)     |
| LSFA     | 固定资产报废损失(收益以“—”号填列) |
| NCFOA    | 经营活动产生的现金流量净额       |
| OTHERS   | 其他                  |
| PROIMPA  | 资产减值准备              |
| UINVSTL  | 未确认的投资损失            |

## 现金流量表（间接法）：

| 接口支持字段   | 描述                               |
|----------|----------------------------------|
| AMORINTA | 无形资产摊销                           |
| AMORLPE  | 长期待摊费用摊销                         |
| BEGINBC  | 现金的期初余额                          |
| BEGINBCE | 现金等价物的期初余额                       |
| CBDOY    | 一年内到期的可转换公司债券                    |
| DDITA    | 递延所得税资产减少(增加以“—”号填列)             |
| DECINV   | 存货的减少(增加以“—”号填列)                 |
| DECOPREC | 经营性应收项目的减少(增加以“—”号填列)            |
| DFOGBBA  | 固定资产折旧 油气资产折耗 生产性生物资产折旧          |
| ENDBC    | 现金的期末余额                          |
| ENDBCE   | 现金等价物的期末余额                       |
| FAFL     | 融资租赁固定资产                         |
| GLCFV    | 公允价值变动损失(收益以“—”号填列)              |
| IDITL    | 递延所得税负债增加(减少以“—”号填列)             |
| INCOPPAY | 经营性应付项目的增加(减少以“—”号填列)            |
| LDFIOLA  | 处置固定资产 无形资产和其他长期资产的损失(收益以“—”号填列) |
| LIACVCAP | 债务转为资本                           |
| LOSINVST | 投资损失(收益以“—”号填列)                  |
| LSFA     | 固定资产报废损失(收益以“—”号填列)              |
| NCFOA    | 经营活动产生的现金流量净额                    |
| OTHERS   | 其他                               |
| PROIMPA  | 资产减值准备                           |
| UINVSTL  | 未确认的投资损失                         |

## 其他财务数据：

| 接口支持字段 | 描述       |
|--------|----------|
| CAR    | 现金适合比率   |
| CAR5Y  | 现金满足投资比率 |

|          |                    |
|----------|--------------------|
| CPTTAR   | 全部现金回收率            |
| CRIR     | 现金再投资比率            |
| FCFTE    | 股权自由现金流（原有）        |
| FCFTF    | 企业自由现金流（原有）        |
| OCFTS    | 现金与利润总额比           |
| OPIND    | 营运指数               |
| RCETDA   | 资本支出与折旧摊销比         |
| RCFTOI   | 营业收入现金比净含量         |
| RCTOI    | 营业收入现金比率           |
| SURCC    | 净利润现金净含量           |
| SURCCTTM | 净利润现金净含量 TTM       |
| BUSTR    | 流转税率               |
| EFFTR    | 所得税率               |
| RETETAR  | 留存收益资产比            |
| ROETCI   | 母公司所有者权益与投入资本比     |
| CURAR    | 流动资产比率             |
| CURLR    | 流动负债比率             |
| FAR      | 固定资产比率             |
| LLIAR    | 非流动负债比率            |
| NCURAR   | 非流动资产比率            |
| OELLFALI | 长期资产适合率            |
| SEFAR    | 股东权益对固定资产比率        |
| SHEQR    | 所有者权益比率            |
| TANGAR   | 有形资产比率             |
| CURR     | 流动比率               |
| WCCAR    | 营运资金对流动资产比率        |
| WCNAR    | 营运资金对净资产比率         |
| WCTAR    | 营运资金对资产总额比率        |
| CFICR    | 现金流量利息保障倍数         |
| DCR      | 经营活动产生的现金流量净额/负债合计 |
| RCFTCL   | 经营活动产生的现金流量净额/流动负债 |
| RCTMD    | 现金流到期债务保障倍数        |
| DLTCR    | 长期资本负债率            |
| EQDBR    | 权益对负债比率            |
| EQMTPL   | 权益乘数               |
| EQR      | 产权比率               |
| LMVSER   | 负债与权益市价比率          |
| LTOTAR   | 资产负债率              |
| REBITL   | 息税折旧摊销前利润/负债合计     |
| RLTBTA   | 长期借款与总资产比          |
| RLTLOC   | 长期债务与营运资金比率        |
| ROCB     | 营运资金与借款比           |
| TIEA     | 利息保障倍数 A           |

|          |                   |
|----------|-------------------|
| TIEB     | 利息保障倍数 B          |
| TOTLNTAR | 有形资产负债率           |
| CASHR    | 现金比率              |
| QUKR     | 速动比率              |
| SQUKR    | 保守速动比率            |
| WORKCAP  | 营运资金              |
| ACCIT    | 应计项目              |
| CACCRA   | 资本积累率 A           |
| CACCRB   | 资本积累率 B           |
| CPARA    | 资本保值增值率 A         |
| CPARB    | 资本保值增值率 B         |
| GRAEPSA  | 基本每股收益增长率 A       |
| GRAEPSB  | 基本每股收益增长率 B       |
| GRCFPSOA | 每股经营活动产生的净流量增长率 A |
| GRCFPSOB | 每股经营活动产生的净流量增长率 B |
| GRDEPSA  | 稀释每股收益增长率 A       |
| GRDEPSB  | 稀释每股收益增长率 B       |
| GRFAA    | 固定资产增长率 A         |
| GRFAB    | 固定资产增长率 B         |
| GRNCFFAA | 筹资活动产生的现金流量增长率 A  |
| GRNCFFAB | 筹资活动产生的现金流量增长率 B  |
| GRNCFIAA | 投资活动产生的现金流量增长率 A  |
| GRNCFIAB | 投资活动产生的现金流量增长率 B  |
| GRNCFOAA | 经营活动产生的现金流量增长率 A  |
| GRNCFOAB | 经营活动产生的现金流量增长率 B  |
| GRNPA    | 净利润增长率 A          |
| GRNPB    | 净利润增长率 B          |
| GROPA    | 营业利润增长率 A         |
| GROPB    | 营业利润增长率 B         |
| GRORA    | 营业收入增长率 A         |
| GRORB    | 营业收入增长率 B         |
| GRROEA   | 净资产收益率增长率 A       |
| GRROEB   | 净资产收益率增长率 B       |
| GRTAA    | 总资产增长率 A          |
| GRTAB    | 总资产增长率 B          |
| GRTPA    | 利润总额增长率 A         |
| GRTPB    | 利润总额增长率 B         |
| SUSGR    | 可持续增长率            |
| CDIVC    | 现金股利保障倍数          |
| DIVCOV   | 股利倍数              |
| DIVPAY   | 股利分配率             |
| RETEARR  | 收益留存率             |
| EBIT     | 息税前利润             |

|             |                 |
|-------------|-----------------|
| EBITA       | 息税折旧摊销前收入       |
| EBITATTM    | 息税折旧摊销前收入 TTM   |
| EBITTTM     | 息税前利润 TTM       |
| ERDSP       | 销售期间费用率         |
| ERDSPTTM    | 销售期间费用率 TTM     |
| FEXPR       | 财务费用率           |
| FEXPRTTM    | 财务费用率 TTM       |
| GOPMAR      | 营业毛利率           |
| GOPMARTTM   | 营业毛利率 TTM       |
| MEXPR       | 管理费用率           |
| MEXPRTTM    | 管理费用率 TTM       |
| NOPPMAR     | 营业净利率           |
| NOPPMARTTM  | 营业净利率 TTM       |
| NPMCAA      | 流动资产净利润率 A      |
| NPM CAB     | 流动资产净利润率 B      |
| NPM CAC     | 流动资产净利润率 C      |
| NPM CATTM   | 流动资产净利润率 TTM    |
| NPMFAA      | 固定资产净利润率 A      |
| NPM FAB     | 固定资产净利润率 B      |
| NPM FAC     | 固定资产净利润率 C      |
| NPM FATTM   | 固定资产净利润率 TTM    |
| OPMAR       | 营业利润率           |
| OPMARTTM    | 营业利润率 TTM       |
| PBIAT       | 息前税后利润          |
| REBITOPI    | 息税前营业利润率        |
| REBITOPITTM | 息税前营业利润率 TTM    |
| REBITTA     | 息税前利润与资产总额比     |
| RETAA       | 资产报酬率 A         |
| RETAB       | 资产报酬率 B         |
| RETAC       | 资产报酬率 C         |
| RETATTM     | 资产报酬率 TTM       |
| RNPTP       | 净利润与利润总额比       |
| ROAA        | 总资产净利润率(ROA)A   |
| ROAB        | 总资产净利润率(ROA)B   |
| ROAC        | 总资产净利润率(ROA)C   |
| ROATTM      | 总资产净利润率(ROA)TTM |
| ROEA        | 净资产收益率(ROE)A    |
| ROEB        | 净资产收益率(ROE)B    |
| ROEC        | 净资产收益率(ROE)C    |
| ROETTM      | 净资产收益率(ROE)TTM  |
| ROIC        | 投入资本回报率         |
| ROLC        | 长期资本收益率         |
| RPRC        | 成本费用利润率         |

|            |                 |
|------------|-----------------|
| RPRCTTM    | 成本费用利润率 TTM     |
| RSC        | 营业成本率           |
| RSCTTM     | 营业成本率 TTM       |
| RTPEBIT    | 利润总额与息税前利润相比    |
| SELEXP     | 销售费用率           |
| SELEXPRTTM | 销售费用率 TTM       |
| APTOVA     | 应付账款周转率 A       |
| APTOVB     | 应付账款周转率 B       |
| APTOVC     | 应付账款周转率 C       |
| APTOVTTM   | 应付账款周转率 TTM     |
| ARTOVA     | 应收账款周转率 A       |
| ARTOVB     | 应收账款周转率 B       |
| ARTOVC     | 应收账款周转率 C       |
| ARTOVDA    | 应收账款周转天数 A      |
| ARTOVDB    | 应收账款周转天数 B      |
| ARTOVDC    | 应收账款周转天数 C      |
| ARTOVDTTM  | 应收账款周转天数 TTM    |
| ARTOVTTM   | 应收账款周转率 TTM     |
| BUSCA      | 营业周期 A          |
| BUSCB      | 营业周期 B          |
| BUSCC      | 营业周期 C          |
| BUSCTTM    | 营业周期 TTM        |
| CAPP       | 资本密集度           |
| CCETOVA    | 现金及现金等价物周转率 A   |
| CCETOVB    | 现金及现金等价物周转率 B   |
| CCETOVC    | 现金及现金等价物周转率 C   |
| CURATOVA   | 流动资产周转率 A       |
| CURATOV    | 流动资产周转率 B       |
| CURATOV    | 流动资产周转率 C       |
| CURATOVTTM | 现金及现金等价物周转率 TTM |
| FATOVA     | 固定资产周转率 A       |
| FATOV      | 固定资产周转率 B       |
| FATOV      | 固定资产周转率 C       |
| FATOVTTM   | 固定资产周转率 TTM     |
| INVT       | 存货周转率 A         |
| INVT       | 存货周转率 B         |
| INVT       | 存货周转率 C         |
| INVT       | 存货周转天数 A        |
| INVT       | 存货周转天数 B        |
| INVT       | 存货周转天数 C        |
| INVT       | 存货周转天数 TTM      |
| INVT       | 存货周转率 TTM       |
| LAT        | 非流动资产周转率 A      |

|             |                     |
|-------------|---------------------|
| LATOV B     | 非流动资产周转率 B          |
| LATOV C     | 非流动资产周转率 C          |
| LATOV TTM   | 非流动资产周转率 TTM        |
| RARI        | 应收账款与收入比            |
| RCAI        | 流动资产与收入比            |
| RFAI        | 固定资产与收入比            |
| RINVI       | 存货与收入比              |
| SETOVA      | 股东权益周转率 A           |
| SETOV B     | 股东权益周转率 B           |
| SETOV C     | 股东权益周转率 C           |
| SETOV TTM   | 流动资产周转率 TTM         |
| TOTATOVA    | 总资产周转率 A            |
| TOTATOV B   | 总资产周转率 B            |
| TOTATOV C   | 总资产周转率 C            |
| TOTATOV TTM | 总资产周转率 TTM          |
| WKCPTOVA    | 营运资金(资本)周转率 A       |
| WKCPTOV B   | 营运资金(资本)周转率 B       |
| WKCPTOV C   | 营运资金(资本)周转率 C       |
| WKCPTOV TTM | 营运资金(资本)周转率 TTM     |
| NCFPS       | 每股现金净流量 1           |
| NCFPSFFA    | 每股筹资活动现金净流量         |
| NCFPSFFATTM | 每股筹资活动现金净流量 TTM     |
| NCFPSFIA    | 每股投资活动现金净流量         |
| NCFPSFIATTM | 每股投资活动现金净流量 TTM     |
| NCFPSFOA    | 每股经营活动产生的现金流量净额     |
| NCFPSFOATTM | 每股经营活动产生的现金流量净额 TTM |
| CRPS        | 每股资本公积              |
| EPS         | 每股收益（原文披露）          |
| FCFFPS      | 每股企业自由现金流（原有）       |
| FCFTEPS     | 每股股权自由现金流（原有）       |
| FDEPS       | 归属于母公司每股收益          |
| NAPS        | 每股净资产               |
| OPIPS       | 每股营业收入              |
| OPIPSTTM    | 每股营业收入 TTM          |
| PPSBIT      | 息税前每股收益             |
| PPSBITTTM   | 息税前每股收益 TTM         |
| REPS        | 每股留存收益              |
| SRPS        | 每股盈余公积              |
| UDPPS       | 每股未分配利润             |
| WAEPS       | 每股收益 1              |
| BVRA        | 账面市值比 A             |
| BVRB        | 账面市值比 B             |
| DIVOCSA     | 普通股获利率 A            |

|          |          |
|----------|----------|
| DIVOCBSB | 普通股获利率 B |
| MVA      | 市值 A     |
| MVB      | 市值 B     |
| PB       | 市净率      |
| PCF      | 市现率 2    |
| PE       | 市盈率 2    |
| PETTM    | 市盈率 TTM  |
| PS       | 市销率 2    |
| PSTTM    | 市销率 TTM  |
| RDYOCBS  | 本利比      |
| TOBQA    | 托宾 Q 值 A |
| TOBQB    | 托宾 Q 值 B |
| TOBQC    | 托宾 Q 值 C |
| TOBQD    | 托宾 Q 值 D |
| FINL     | 财务杠杆     |
| OPL      | 经营杠杆     |
| TOTL     | 综合杠杆     |

### 4.13 [GetFactor](#) 入参 fields 及出参返回的列式结果集字段

| 接口支持字段                       | 描述                    |
|------------------------------|-----------------------|
| TradingDate                  | 交易日期                  |
| SecurityID                   | <a href="#">证券ID</a>  |
| Symbol                       | 证券代码                  |
| Market                       | <a href="#">交易所代码</a> |
| QF_AvgPE                     | A 股平均市盈率 （全市场的平均市盈率）  |
| QF_IndustryCode12            | 12 标准行业因子             |
| QF_IndustryCode12_Normal     | 标准化后的 12 标准行业因子       |
| QF_TotalRevenueGrowth        | 营业收入增长率               |
| QF_TotalRevenueGrowth_Normal | 标准化后的营业收入增长率          |
| QF_EPS                       | 每股收益                  |
| QF_EPS_Normal                | 标准化后的每股收益             |
| QF_EPSF                      | 分析师预测每股收益             |
| QF_EPSF_Normal               | 标准化后的分析师预测每股收益        |
| QF_HisVolatility             | 历史波动率因子（252 天）        |
| QF_HisVolatility_Normal      | 标准化后的历史波动率因子          |
| QF_Momentum24M               | 24 个月动量因子             |
| QF_Momentum24M_Normal        | 标准化后的 24 个月动量因子       |
| QF_NetAssetPS                | 每股净资产                 |
| QF_NetAssetPS_Normal         | 标准化后的每股净资产            |
| QF_PE                        | 市盈率                   |
| QF_PE_Normal                 | 标准化后的市盈率              |



|              |          |
|--------------|----------|
| QF_PB        | 市净率      |
| QF_PB_Normal | 标准化后的市净率 |

#### 4.14 [GetHisMarketInfo](#) 入参 fields 及出参返回的列式结果 集字段

| 接口支持字段                     | 描述                       |
|----------------------------|--------------------------|
| SecurityID                 | <a href="#">证券ID</a>     |
| ExDividendDate             | 除权除息日                    |
| Symbol                     | 证券代码                     |
| Market                     | <a href="#">交易所代码</a>    |
| RecordDate                 | 股权登记日                    |
| FinalTradingDate           | 最后交易日（B 股）               |
| AllotmentPrice             | 配股价格                     |
| AllotmentPerShare          | 配股比例                     |
| DividentBT                 | 每股分红(税前)                 |
| DividentAT                 | 每股分红(税后)                 |
| BonusRatio                 | 送股比例                     |
| SplitRatio                 | 拆分比例                     |
| DeclareDate                | 公告日期                     |
| PaymentDate                | 派息日期                     |
| TradingDate                | 交易日期                     |
| BwardFactor1               | 后复权因子                    |
| FwardFactor1               | 前复权因子                    |
| CumulateFwardFactor1       | 前累计复权因子                  |
| ExdividendFwardFactor1     | 不考虑现金分红的前复权因子            |
| ExdividendCumulateFFactor1 | 不考虑现金分红的前累计复权因子          |
| CumulateBwardFactor1       | 后累计复权因子                  |
| ExdividendBwardFactor1     | 不考虑现金分红的后复权因子            |
| ExdividendCumulateBFactor1 | 不考虑现金分红的后累计复权因子          |
| FwardFactor2               | 不考虑拆分、股改的前复权因子           |
| CumulateFwardFactor2       | 不考虑拆分、股改的前累计复权因子         |
| ExdividendFwardFactor2     | 不考虑现金分红、拆分、股改的前复权因子      |
| ExdividendCumulateFFactor2 | 不考虑现金分红、拆分、股改的前累计复权因子    |
| BwardFactor2               | 不考虑拆分、股改的后复权因子           |
| CumulateBwardFactor2       | 不考虑拆分、股改的后累计复权因子         |
| ExdividendBwardFactor2     | 不考虑现金分红、拆分、股改的后复权因子      |
| ExdividendCumulateBFactor2 | 不考虑现金分红、拆分、股改的后累计复权因子    |
| Marketttype                | 市场类型,0 为主板,1 为中小板,2 为创业板 |
| Dretwd                     | 考虑现金红利再投资日个股回报率          |
| Dretnnd                    | 不考虑现金红利日个股回报率            |

|                       |                     |
|-----------------------|---------------------|
| Adjprcwd              | 考虑现金红利再投资日收盘价的可比价格  |
| Adjprcnd              | 不考虑现金红利再投资日收盘价的可比价格 |
| ReturnWeekly1         | 考虑现金红利再投资一周个股回报率    |
| ReturnCurrentWeek1    | 考虑现金红利再投资本周个股回报率    |
| ReturnWeekly2         | 不考虑现金红利再投资一周个股回报率   |
| ReturnCurrentWeek2    | 不考虑现金红利再投资本周个股回报率   |
| ReturnMonthly1        | 考虑现金红利再投资一个月个股回报率   |
| ReturnCurrentMonth1   | 考虑现金红利再投资本月个股回报率    |
| ReturnMonthly2        | 不考虑现金红利再投资一个月个股回报率  |
| ReturnCurrentMonth2   | 不考虑现金红利再投资本月个股回报率   |
| Return3Month1         | 考虑现金红利再投资三个月个股回报率   |
| ReturnCurrentQuarter1 | 考虑现金红利再投资本季度个股回报率   |
| Return3Month2         | 不考虑现金红利再投资三个月个股回报率  |
| ReturnCurrentQuarter2 | 不考虑现金红利再投资本季度个股回报率  |
| Return6Month1         | 考虑现金红利再投资六个月个股回报率   |
| ReturnCurrent6Month1  | 考虑现金红利再投资本半年个股回报率   |
| Return6Month2         | 不考虑现金红利再投资六个月个股回报率  |
| ReturnCurrent6Month2  | 不考虑现金红利再投资本半年个股回报率  |
| ReturnAnnual1         | 考虑现金红利再投资年个股回报率     |
| YearRTD1              | 考虑现金红利再投资本年个股回报率    |
| ReturnAnnual2         | 不考虑现金红利再投资一年个股回报率   |
| YearRTD2              | 不考虑现金红利再投资本年个股回报率   |
| ReturnTotal1          | 考虑现金红利再投资累计个股回报率    |
| ReturnTotal2          | 不考虑现金红利再投资累计个股回报率   |
| ChangeDate            | 变动日期                |
| StateIDChange         | 状态前后编码变动            |
| SuspensionDate        | 停牌开始日期              |
| ResumptionDate        | 复牌日期                |
| SuspensionTypeCode    | 停牌类型编码              |
| SuspensionTime        | 停牌开始时间              |
| ResumptionTime        | 复牌时间                |

#### 4.15 [GetJointContracts](#) 出参返回的列式结果集内容

| 接口支持字段      | 描述                    |
|-------------|-----------------------|
| SecurityID  | <a href="#">证券ID</a>  |
| Symbol      | 交易代码                  |
| ShortName   | 证券简称                  |
| Market      | <a href="#">交易所代码</a> |
| TradingDate | 交易日期                  |
| SignTypeID  | 主力连续分类                |

|                         |           |
|-------------------------|-----------|
| UnderlyingSecurityID    | 标的证券 ID   |
| UnderlyingSymbol        | 标的证券代码    |
| VarietyID               | 品种 ID     |
| UnderlyingAssetsCode    | 标的品种编码    |
| OldUnderlyingAssetsCode | 标的品种编码（旧） |
| ModeID                  | 主力合约规则 ID |

## 5 附录 1 红楹云数据接口数据与行情定义

### 附录 1.1 数据 API 定义

```
typedef vector< unsigned char > bytes;

typedef uint64 SecurityID;
typedef UInt64Seq SecurityIDSeq;          //证券ID序列

////////////////////////////////////

//数据类型，用于指定列式数据的类型及解析，与ProtoBuf协议无关
enum EDataType
{
    k_Char          = 0,    //字符串
    k_Bool          = 1,    //
    k_Byte          = 2,    //
    k_Int8          = 3,    //
    k_UInt8         = 4,
    k_Int16         = 5,
    k_UInt16        = 6,
    k_Int32         = 7,
    k_UInt32        = 8,
    k_Int64         = 9,
    k_UInt64        = 10,
    k_Float         = 11,
    k_Double        = 12,
    k_DateTime      = 13,    //实质为以uint64表示的毫秒数，若库中数据时间信息实质为字符串型，一律转为
    //本类型。分时数据
    k_TimeStamp     = 14,    //实质为以uint64表示的微秒数，若库中数据时间信息实质为字符串型，一律转为
    //本类型。tick信息
};

//QtAPI布尔枚举值
enum EQtBool
{
    k_QtFalse       = 0,
    k_QtTrue        = 1,
    k_QtNull        = 2,
};

//集合操作，用于数据结果集之间的集合操作，暂只考虑并、交两种
enum ESetOper
{
    k_SetUnion       = 1,    //并集
    k_SetInter       = 2,    //交集
    //k_SetComp      = 3,    //补集
};

//排序类型，用于指定数据结果集的排序要求
enum ESortType
{

```

```

    k_SortNone      = 0,    //不排序
    k_SortDesc      = 1,    //降序
    k_SortAsec      = 2,    //升序
};

//用于指明数据提取的方向
enum EDirection
{
    k_Forward       = 1,    //向前, 时间点以前
    k_Backward      = 2,    //向后, 时间点以后
};

//行情采样类型
enum EQuoteType
{
    k_Second        = 1,    //秒, 日内高频
    k_Minute        = 2,    //分, 日内高频
    k_Hour          = 3,    //时, 日内高频
    k_Day           = 4,    //日, 低频
    k_Week          = 5,    //周, 低频
    k_Month         = 6,    //月, 低频
    k_Quarter       = 7,    //季, 低频
    k_Year          = 8,    //年, 低频
};

//行情数据类型
enum EQuoteData
{
    k_QuoteNull     = 0,
    k_Quote         = 1,    //行情类消息, 沪深L1、中金所L2、商品期货L1、上海L1个股期权
    k_Sts           = 2,    //分时消息, 沪深L1、中金所L2、商品期货L1、上海L1个股期权

    k_Arbitrage     = 3,    //L1套利行情(大商所)
    k_Static        = 4,    //静态数据(上海L2、深圳L2、上交所个股期权)
    k_Bulletin      = 5,    //公告及增值信息(深圳L1)

    k_L2Auction     = 6,    //L2集合竞价(上海)
    k_L2Index       = 7,    //L2指数行情(上海、深圳)
    k_L2Order       = 8,    //L2逐笔委托(深圳)
    k_L2Orderqueue  = 9,    //L2委托队列(上海、深圳)
    k_L2Quote       = 10,   //L2十档行情(上海、深圳)
    k_L2Transaction = 11,   //L2逐笔成交(上海、深圳)
    k_L2Status      = 12,   //L2证券状态(深圳)

    k_L2Arbitrage   = 13,   //L2套利行情
    k_L2MarchPriceQty = 14, //L2分价成交量行情
    k_L2OrderStatistic = 15, //L2委托统计行情
    k_L2RealTimePrice = 16, //L2实时结算价
    k_StsPerTick    = 20,   //Tick分时消息, 沪深L1、中金所L2、商品期货L1、上海L1个股期权
};

//报告类型, 主要用于财务因子/报表数据
enum EReportType
{
    k_RptMergeCur  = 1,    //合并本期
    k_RptMergePre   = 2,    //合并上期
    k_RptParentCur = 3,    //母公司本期
    k_RptParentPre  = 4,    //母公司上期
};

//报告日期类型, 主要用于财务因子/报表数据
enum ERptDateType
{
    k_RptDateIssue  = 0,    //报告发布日期
    k_RptDateClose  = 1,    //报告财期截止日期
};

//价格复权调整类型
enum EPriceAdjust

```

```

{
    k_AdjNone      = 0,    //不复权, 依20160902评审建议增加
    k_AdjForward   = 1,    //前复权
    k_AdjBackward  = 2,    //后复权
};

//财务报表累计类型
enum ETrailType
{
    k_TrailSeason   = 1,    //单季度
    k_TrailAddup    = 2,    //季度累计
    k_Trail12Mon    = 3,    //TTM, 12个月滚动累计
};

//时间段[begin, end]
struct TimePeriod
{
    string    begin;        //起始时间, 格式YYYY-MM-DD HH:MM:SS.MMM
    string    end;          //截止时间
};

//排序定义, 基于数据名指明排序要求
struct SortDefine
{
    string    field;        //数据名
    ESortType sort;        //排序类型
};

//数据列, 用于列式数据集
struct DataColumn
{
    string    name;         //数据名
    EDataType type;         //数据类型
    uint      count;        //数据个数
    bytes     data;         //数据, data.size = count * typeSize
    bool      zipped;       //bytes是否压缩
    uint      typeSize;     //数据类型字节长度, 主要用于字符串k_Char, 其它类型根据dataType已经可以确定类型长度
};
typedef vector< DataColumn > ColumnSeq;

//列式数据集
struct ColumnSet
{
    string    identity;     //结果集标识
    ColumnSeq columns;      //结果列式数据集
};
typedef vector< ColumnSet > ColumnSetSeq;

typedef vector< SortDefine > SortDefineSeq;
typedef vector< TimePeriod > TimePeriodSeq;

```

## 附录 1.2 行情 API 定义

```

///< 分时行情推送数据
struct QtSubSts
{
    string Symbol;          ///< 证券代码
    string Market;          ///< 交易所

    int TradingDate;        ///< 交易日期
    string TradingTime;     ///< 交易时间
    int64 UNIX;             ///< UTC时间戳

    double OP;              ///< 分日内开盘价
    double HIP;             ///< 分日内最高价
    double LOP;             ///< 分日内最低价

```

```

double CP;           ///< 分时段收盘价
double CQ;           ///< 分时段成交量
double CM;           ///< 分时段成交金额
double Change;       ///< 涨跌
double ChangeRatio;  ///< 涨跌幅
double TQ;           ///< 成交总量
double Vwap;         ///< 成交量加权平均价
double OBPD;         ///< 开盘基准价
double ClvLOP;       ///< 开盘至今最低价
double ClvHIP;       ///< 开盘至今最高价
double ClvVwap;      ///< 开盘至今交易量加权平均价

//期权、期货, CFFEXFFL2Min, CFFEXIOL2Min, CFFEXTFL2Min, CZCEL1Min, DCEL1Min, SHFEL1Min,
SSEIOL1Min
string ShortName;    ///< 证券简称, SSEIOL1Quote无, SSEIOL1Quote有, SZSEL1Quote有
double Position;     ///< 持仓量
string ConSign;      ///< 连续合约代码, SSEIOL1Quote无
double Rtn;          ///< 简单收益率

//中金所, CFFEXFFL2Min, CFFEXIOL2Min, CFFEXTFL2Min
string ConSignName;  ///< 连续合约名称

//商品期货 CZCEL1Min, DCEL1Min, SHFEL1Min
string Varieties;    ///< 品种

//期权, CFFEXIOL2Min, SSEIOL1Min
double MarginUnit;   ///< 期权初始保证金
double MaintainingMargin; ///< 期权维持保证金
//QTFiveTradeInfo FiveTradeInfo; ///< 五档交易行情

DoubleSeq SellPrice; ///< 买价
DoubleSeq BuyPrice;  ///< 卖价
DoubleSeq BuyVolume;  ///< 买量
DoubleSeq SellVolume; ///< 卖量
};

////! L1行情消息, 适用于沪深L1、中金所L2、商品期货L1、上海L1个股期权
struct QtSubQuote
{
    string Symbol;      ///< 证券代码
    string Market;      ///< 交易所

    int TradingDate;    ///< 交易日期
    string TradingTime; ///< 交易时间
    int64 UNIX;         ///< UTC时间戳

    double LCP;         ///< 昨收盘价
    double OP;          ///< 今开盘价
    double HIP;         ///< 最高价
    double LOP;         ///< 最低价
    double CP;          ///< 最新成交价
    double CP1;         ///< 今收盘价
    double TQ;          ///< 当前总成交量
    double TM;          ///< 当前总成交金额

    //QTFiveTradeInfo FiveTradeInfo; ///< 五档交易行情

    DoubleSeq SellPrice; ///< 买价
    DoubleSeq BuyPrice;  ///< 卖价
    DoubleSeq BuyVolume;  ///< 买量
    DoubleSeq SellVolume; ///< 卖量

    string BS;          ///< 买卖标识
    double ChangeRatio; ///< 涨跌幅
    double CQ;          ///< 分笔期间成交量
    double CM;          ///< 分笔期间成交金额
    double Vwap;        ///< 成交均价
    double OrderRate;   ///< 委比
    double OrderDiff;   ///< 委差

```

```

double Amplitude;        ///< 振幅
double VolRate;          ///< 量比
double SellVOL;          ///< 内盘
double BuyVOL;           ///< 外盘

//期权、期货, CFFEXF2Quote, CFFEXIOL2Quote, CFFEXTFL2Quote, CZCEL1Quote, DCEL1Quote,
SHFEL1Quote, SSEIOL1Quote
string ShortName;        ///< 证券简称, SSEIOL1Quote无, SSEL1Quote有, SZSEL1Quote有
double LSP;              ///< 昨结算价
double SP;               ///< 今结算价
double PrePosition;      ///< 昨持仓量
double Position;         ///< 持仓量
double Change;           ///< 涨跌
string OC;               ///< 开平仓性质
string LocalTimeStamp;   ///< 本地时间戳
double PositionChange;   ///< 仓差
double PrePositionChange;///< 日增仓

double LimitUp;          ///< 涨停价, SSEIOL1Quote无, SZSEL1Quote有
double LimitDown;        ///< 跌停价, SSEIOL1Quote无, SZSEL1Quote有

double PreDelta;         ///< 昨虚实度, SSEIOL1Quote无
double Delta;            ///< 今虚实度, SSEIOL1Quote无
string ConSign;          ///< 连续合约代码, SSEIOL1Quote无

//商品期权
double Gamma;            ///< Gamma
double Rho;              ///< Rho
double Theta;            ///< Theta
double Vega;             ///< Vega

//中金所
string SettleGroupID;    ///< 结算组代码
double SettleID;         ///< 结算编号
string ConSignName;      ///< 连续合约名称

//商品期货(郑商所、大商所、上期所)
double LastVol;          ///< 最新成交量
double LifeLow;          ///< 历史最低价
double LifeHigh;         ///< 历史最高价
double avgprice1;        ///< 成交均价01
double BidImpleQty;      ///< 申买推导量
double AskImpleQty;      ///< 申卖推导量
string Varieties;        ///< 品种
double DelSP;            ///< 当日交割结算价

//沪深L1行情快照
double PriceChange01;    ///< 价格涨跌1(备注:跟上面的涨跌是同个字段名称)
double Change1;          ///< PriceChange02,价格涨跌2
double BuyRatio;         ///< 买比
double SPD;              ///< Spread,买卖价差
double RPD;              ///< RelativeSpread,相对价差
double Depth01;          ///< 报价深度1
double Depth02;          ///< 报价深度2
double TurnoverRate;     ///< 换手率
double CMV;              ///< CirculationMarketValue, 流通市值
double TMV;              ///< TotalMarketVAL, 总市值
double PE;               ///< PE1,市盈率
double PB;               ///< 市净率
double Premium;          ///< 溢价
double PremiumRate;      ///< 溢价率

//上海L1个股期权、深圳L1行情快照
string StatusID;         ///< SecurityPhaseTag,交易状态

//上海L1个股期权
double AuctionPrice;     ///< 波动性中断参考价
double AuctionVolume;    ///< 波动性中断集合竞价虚拟匹配量
int BuyLevelNO;          ///< 委托买入档位

```

```

int SellLevelNO;          ///< 委托卖出档位

//深圳L1
string SymbolSource;      ///< 证券代码源
double NAV;               ///< 基金 T-1 日净值
double IOPV;              ///< 基金实时参考净值
uint SampleNo;            ///< 样本个数
uint64 TT;                ///< TotalNo,当前总成交笔数
uint64 CT;                ///< TradeNo,分笔期间成交笔数
double PE1;               ///< PERatio1,市盈率1
double PE2;               ///< PERatio2,市盈率2
};

////! L2十档行情(上海、深圳)
struct QtSubL2Quote
{
    string Symbol;         ///< 证券代码
    string Market;         ///< 交易所

    int TradingDate;       ///< 交易日期
    string TradingTime;    ///< 交易时间
    int64 UNIX;            ///< UTC时间戳

    double LCP;            ///< 昨收盘价
    double OP;             ///< 开盘价
    double HIP;            ///< 最高价
    double LOP;            ///< 最低价
    double CP;             ///< 现价
    double CP1;            ///< 今日收盘价

    uint64 TT;             ///< TotalNo,成交笔数
    double TQ;             ///< 成交总量
    double TM;             ///< 成交总额

    uint BuyLevelNo;       ///< 申买报价档位
    //QTStkTenTradeInfo tenTradeInfo;
    DoubleSeq BuyPrice;     ///< 委买价
    DoubleSeq BuyVolume;    ///< 委买量
    DoubleSeq TotalBuyOrderNo; ///< 委买笔数
    double TBOVOL;         ///< TotalBuyOrderVolume,委托买入总量

    uint SellLevelNo;      ///< 申卖价格档位
    DoubleSeq SellPrice;    ///< 委卖价
    DoubleSeq SellVolume;   ///< 委卖量
    DoubleSeq TotalSellOrderNo; ///< 委卖笔数
    double TSOVOL;         ///< TotalSellOrderVolume,委托卖出总量

    double WAvGBP;         ///< WtAvgBuyPrice,加权平均委买价格
    double WAvGSP;         ///< WtAvgSellPrice,加权平均委卖价格
    double IOPV;           ///< ETF净值估计

    double BondWtAvGSP;    ///< BondWtAvgSellPrice,债券加权平均委卖价格
    double YTM;            ///< 到期收益率
    double ETFBNo;         ///< ETFBuyNo,ETF申购笔数
    double ETFBVOL;        ///< ETFBuyVolume,ETF申购数量
    double ETFBM;          ///< ETFBuyAmount,ETF申购金额
    double ETFSNo;         ///< ETFSellNo,ETF赎回笔数
    double ETFSVOL;        ///< ETFSellVolume,ETF赎回数量
    double ETFSM;          ///< ETFSellAmount,ETF赎回金额
    double WithdrawBNo;    ///< WithdrawBuyNo,买入撤单笔数
    double WithdrawBVOL;   ///< WithdrawBuyVolume,买入撤单数量
    double WithdrawBM;     ///< WithdrawBuyAmount,买入撤单金额
    double WithdrawSNo;    ///< WithdrawSellNo,卖出撤单笔数
    double WithdrawSVOL;   ///< WithdrawSellVolume,卖出撤单数量
    double WithdrawSM;     ///< WithdrawSellAmount,卖出撤单金额
    double TotalBNo;       ///< TotalBuyNo,买入总笔数
    double TotalSNo;       ///< TotalSellNo,卖出总笔数
    double MaxBD;          ///< MaxBuyDuration,买入委托成交最大等待时间
    double MaxSD;          ///< MaxSellDuration,卖出委托成交最大等待时间
}

```



```

double BONO;          ///< BuyOrderNo, 买方委托价位数
double SONO;          ///< SellOrderNo, 卖方委托价位数
double BondWtAvgBP;   ///< BondWtAvgBuyPrice, 债券加权平均委买价格
string TradeStatus;   ///< TradeStatus, 当前品种状态

int64 TradeNo;        ///< 分笔期间成交笔数
double TradeVolume;   ///< 分笔期间成交量
double TradeAmount;   ///< 分笔期间成交金额
string Status;        ///< SecurityPhaseTag, 当前品种交易状态
double PE1;           ///< PERatio1市盈率 1
double NAV;           ///< 基金T-1日净值
double PE2;           ///< PERatio2, 市盈率 2
double PremiumRate;   ///< 权证溢价率
double LimitUp;        ///< PriceUpLimit, 涨停价
double LimitDown;     ///< PriceDownLimit, 跌停价
double PriceUpdown1;  ///< 涨跌一
double PriceUpdown2;  ///< 涨跌二
string SymbolSource;  ///< 证券代码源

//备注: 以下字段属个股期权字段
string ShortName;     ///< 合约简称
double Vwap;          ///< AvgPrice, 当日均价
double SP;            ///< SettlePrice, 今结算价
double LSP;           ///< PreSettlePrice, 前结算价
double LastVol;       ///< LastVolume, 当前成交量
double PrePosition;   ///< PreTotalPosition, 昨持仓量
double Position;      ///< TotalPosition, 持仓量
double PrePositionChange; ///< PositionChange, 仓差
double LifeHigh;      ///< CumulativeHighPrice, 开盘至今最高价
double LifeLow;       ///< CumulativeLowPrice, 开盘至今最低价

//int Nolevel;        ///< 深度行情数量
string BuyOrSell;     ///< side, 买卖方向
DoubleSeq BidImpleQty; ///< 申买推导量
DoubleSeq AskImpleQty; ///< 申卖推倒量
DoubleSeq BestBuyQty; ///< 最优价委买量
DoubleSeq BestSellQty; ///< 最优价委卖量
string ConSign;       ///< ContinueSign, 连续合约代码
double Theta;         ///< Theta
double Vega;          ///< Vega
double Gamma;         ///< Gamma
double Rho;           ///< Rho
double Delta;         ///< 今虚实度
};

////! L1套利行情(大商所)
struct QtSubArbitrage
{
    string Symbol;      ///< 证券代码
    string Market;      ///< 交易所

    int TradingDate;    ///< 交易日期
    string TradingTime;  ///< 交易时间
    int64 UNIX;         ///< UTC时间戳

    double CP;          ///< 最新价
    double LOP;         ///< 最低价
    double HIP;         ///< 最高价
    double LifeLow;     ///< 历史最低价
    double LifeHigh;    ///< 历史最高价
    double LimitUp;     ///< PriceUpLimit, 涨停板
    double LimitDown;   ///< PriceDownLimit, 跌停板

    //备注, 以下数据参照MFL2_ARBI套利深度行情
    double B1;          ///< BuyPrice01, 最高买
    double BV1;         ///< BuyVolume01, 申买量
    double SP1;         ///< SellPrice01, 最低卖
    double SV1;         ///< SellVolume01, 申卖量
    string RecTime;     ///< 本地时间戳
};

```

```

};

///< 公告及增值信息(深圳L1)
struct QtSubBulletin
{
    string Symbol;           ///< 证券代码
    string Market;          ///< 交易所

    int TradingDate;         ///< 交易日期
    string TradingTime;      ///< 交易时间
    int64 UNIX;              ///< UTC时间戳

    int ChannelNo;           ///< 频道代码
    string NewsID;           ///< 公告标识
    string Headline;         ///< 公告标题
    string RawDataFormat;    ///< 二进制数据格式
    int64 RawDataLength;     ///< 二进制数据长度
    string RawData;          ///< 二进制数据
};

///< 静态数据(上海L2、深圳L2、上交所个股期权)
struct QtSubStatic
{
    string Symbol;           ///< 证券代码
    string Market;          ///< 交易所

    int TradingDate;         ///< 交易日期
    string TradingTime;      ///< 交易时间
    int64 UNIX;              ///< UTC时间戳

    int EndDate;             ///< 最后交易日期
    double LCP;              ///< 昨收价
    double TickSize;         ///< 最小报价单位
    string OptionType;       ///< 期权类型(行权类型)
    string CallOrPut;        ///< 认购认沽
    double RoundLot;         ///< 整手数
    string UpDownLimitType;   ///< 涨跌幅限制类型
    double LimitUp;           ///< PriceUpLimit, 涨幅限制
    double LimitDown;        ///< PriceDownLimit, 跌幅限制
    string SecurityEN;       ///< 证券英文简称
    string ISIN;             ///< ISIN代码
    string SymbolUnderlying;  ///< 基础证券代码
    string Currency;         ///< 币种
    double ParValue;         ///< 面值
    int ListingDate;         ///< 上市日期
    double BuyVolumeUnit;     ///< 买数量单位
    double SellVolumeUnit;   ///< 卖数量单位
    string CrdBuyUnderlying;  ///< 融资标的标志
    string CrdSellUnderlying; ///< 融券标的标志
    string SecurityName;      ///< 证券名称
    string CFICode;          ///< 证券类别
    string SecuritySubType;   ///< 证券子类别
    double TradableNo;       ///< 可流通未上市数量
    int SetNo;               ///< 产品集编号
    int DeclareVolumeFloor;   ///< 申报量下限
    int DeclareVolumeCeiling; ///< 申报量上限

    double XRRatio;          ///< 除权比例
    double XDAmount;         ///< 除息金额
    string SecurityStatus;    ///< 产品状态标识
    int SampleNo;            ///< 样本数量
    double SampleAvgPrice;    ///< 样本均价
    double TradeAmount;       ///< 成交金额
    double AvgCapital;        ///< 平均股本
    double TotalMarketValue;  ///< 总市值
    double MarketValueRatio;  ///< 占比%
    double StaticPERatio;     ///< 静态市盈率
    string IndexLevelStatus;  ///< 指数级别标识
    string MarketType;        ///< 市场种类
};

```

```

string UnderlyingSecurityIDSource;///< 基础证券代码源
int SecurityType;                ///< 证券类别代码
string SecurityStatusTag;        ///< 证券状态标识
double IssuedVolume;             ///< 总发行量
double OutstandingShare;         ///< 流通股数
string IndustryType;             ///< 行业种类
double PreYearEPS;               ///< 上年每股利润
double YearEPS;                  ///< 本年每股利润
string OfferingFlag;             ///< 收购（转股、行权）标志
double NAV;                      ///< 基金T-1日累计净值
double CouponRate;              ///< 票面利率
double IssuePrice;               ///< 贴现发行价
double Interest;                 ///< 每百元应计利息
int InterestAccrualDate;         ///< 起息日
int MaturityDate;                ///< 到期交割日
double ConversionPrice;          ///< 行权价格
double ConversionRatio;          ///< 行权比例
int ConversionBeginDate;         ///< 行权开始日
int ConversionEndDate;           ///< 行权结束日
string WarrantClearingType;      ///< 权证结算方式
double ClearingPrice;            ///< 结算价格

int ExpirationDays;              ///< 购回期限
string DayTrading;               ///< 回转交易标志
string GageFlag;                 ///< 保证金证券标志
double GageRate;                 ///< 担保品折算率

string CrdPriceCheckType;        ///< 提价检查方式
string PledgeFlag;               ///< 质押入库标志
double ContractMultiplier;       ///< 债券折回购标准券比例
string RegularShare;             ///< 对应回购标准券
string QualificationFlag;        ///< 投资者适当性管理标志
string MarketMakerFlag;         ///< 做市商标志

double BuyQtyUpperLimit;         ///< 买数量上限
double SellQtyUpperLimit;        ///< 卖数量上限

double LimitUpRate0;             ///< 开盘集合竞价上涨幅度
double LimitDownRate0;          ///< 开盘集合竞价下跌幅度
double LimitUpAbsolute0;         ///< 开盘集合竞价上涨限价
double LimitDownAbsolute0;       ///< 开盘集合竞价下跌限价
double AuctionUpDownRate0;       ///< 开盘集合竞价有效范围涨跌幅度
double AuctionUpDownAbsolute0;   ///< 开盘集合竞价有效范围涨跌价格
double LimitUpRateT;             ///< 连续竞价上涨幅度
double LimitDownRateT;          ///< 连续竞价下跌幅度
double LimitUpAbsoluteT;         ///< 连续竞价上涨限价
double LimitDownAbsoluteT;       ///< 连续竞价下跌限价
double AuctionUpDownRateT;       ///< 连续竞价有效范围涨跌幅度
double AuctionUpDownAbsoluteT;   ///< 连续竞价有效范围涨跌价格
double LimitUpRateC;             ///< 收盘集合竞价上涨幅度
double LimitDownRateC;          ///< 收盘集合竞价下跌幅度
double LimitUpAbsoluteC;         ///< 收盘集合竞价上涨限价
double LimitDownAbsoluteC;       ///< 收盘集合竞价下跌限价
double AuctionUpDownRateC;       ///< 收盘集合竞价有效范围涨跌幅度
double AuctionUpDownAbsoluteC;   ///< 收盘集合竞价有效范围涨跌价格
string SymbolSource;             ///< 证券代码源

//备注：以下都属于个股期权字段
string ShortName;                ///< 合约简称
string UnderlyingSymbol;         ///< UnderlyingSecuritySymbol,标的证券代码
string UnderlyingShortName;      ///< 标的证券简称
string Underlyingtype;           ///< 标的证券类型
double ContractMulti;            ///< ContractMultiplierUnit,合约单位
double StrikePrice;              ///< ExercisePrice,期权行权价
int StartDate;                   ///< 首个交易日

int ExerciseDate;                ///< 期权行权日
int DeliveryDate;                ///< 行权交割日
int ExpireDate;                  ///< 期权到期日

```

```

//string Version;          ///< 合约版本号

double PrePosition;        ///< 昨持仓量
double LSP;                ///< PreSettlePrice,昨结算价    decimal
double UnderlyingPreClosePrice; ///< 标的证券昨收盘价
double MarginUnit;         ///< 单位保证金
double MarginRatioParam1;  ///< 保证金计算比例参数一
double MarginRatioParam2;  ///< 保证金计算比例参数二

double LmtOrdFloor;        ///< 单笔限价申报下限
double LmtOrdCeiling;      ///< 单笔限价申报上限
double MktOrdFloor;        ///< 单笔市价申报下限
double MktOrdCeiling;      ///< 单笔市价申报上限
string SecurityStatusFlag; ///< 期权合约状态
string ContractCode;       ///< 合约编码
};

////! L2套利行情
struct QtSubL2Arbitrage
{
    string Symbol;          ///< 证券代码
    string Market;          ///< 交易所

    int TradingDate;        ///< 交易日期
    string TradingTime;     ///< 交易时间
    int64 UNIX;             ///< UTC时间戳

    double CP;              ///< 最新价
    double LOP;             ///< 最低价
    double HIP;             ///< 最高价
    double LifeLow;         ///< 历史最低价
    double LifeHigh;        ///< 历史最高价
    double LimitUp;         ///< PriceUpLimit,涨停板
    double LimitDown;       ///< PriceDownLimit,跌停板
    int Nolevel;            ///< 深度行情数量
    DoubleSeq SellPrice;    ///< 买价
    DoubleSeq BuyPrice;     ///< 卖价
    DoubleSeq BuyVolume;    ///< 买量
    DoubleSeq SellVolume;   ///< 卖量

    DoubleSeq BidImpleQty;  ///< 申买推倒量
    DoubleSeq AskImpleQty;  ///< 申卖推倒量
    DoubleSeq BestSellQty;  ///< 最优委卖量
    DoubleSeq BestBuyQty;   ///< 最优委买量

    //QTFiveTradeInfo fiveTradeInfo;
    //QTMFL2ImpleInfo impleInfo;
};

////! L2集合竞价(上海)
struct QtSubL2Auction
{
    string Symbol;          ///< 证券代码
    string Market;          ///< 交易所

    int TradingDate;        ///< 交易日期
    string TradingTime;     ///< 交易时间
    int64 UNIX;             ///< UTC时间戳

    double AOP;             ///< AuctionOpenPrice,虚拟开盘参考价格
    double AVOL;            ///< AuctionVolume,虚拟匹配量
    double LVOL;            ///< LeaveVolume,虚拟未匹配量
    string BS;              ///< Side,买卖方向
};

////! L2指数行情(上海、深圳)
struct QtSubL2Index
{

```

```

string Symbol;          ///< 证券代码
string Market;          ///< 交易所

int TradingDate;        ///< 交易日期
string TradingTime;      ///< 交易时间
int64 UNIX;             ///< UTC时间戳

double LCP;             ///< 昨收价
double OP;              ///< 开盘价
double HIP;             ///< 最高价
double LOP;             ///< 最低价
double CP;              ///< 现价
double TM;              ///< 成交金额
double TQ;              ///< 成交总量
double CP1;             ///< 今收盘价
int64 TT;               ///< TotalNo, 成交笔数
string SymbolSource;     ///< 证券代码源
string Status;          ///< SecurityPhaseTag, 当前品种交易状态
int SampleNo;           ///< 样本个数
};

////! L2分价成交量行情
struct QtSubL2MarchPriceQty
{
    string Symbol;          ///< 证券代码
    string Market;          ///< 交易所

    int TradingDate;        ///< 交易日期
    string TradingTime;      ///< 交易时间
    int64 UNIX;             ///< UTC时间戳

    DoubleSeq Price;        ///< 价格
    DoubleSeq PriceBOQTY;   ///< 买开数量
    DoubleSeq PriceBEQTY;   ///< 买平数量
    DoubleSeq PriceSOQTY;   ///< 卖开数量
    DoubleSeq PriceSEQTY;   ///< 卖平数量
    string RecTime;         ///< 接收时间
};

////! L2逐笔委托(深圳)
struct QtSubL2Order
{
    string Symbol;          ///< 证券代码
    string Market;          ///< 交易所

    int TradingDate;        ///< 交易日期
    string TradingTime;      ///< 交易时间
    int64 UNIX;             ///< UTC时间戳

    int SetID;              ///< 频道代码
    int64 RecID;            ///< 消息记录号
    string SymbolSource;     ///< 证券代码源
    double OrderPrice;       ///< 委托价格
    double OrderVOL;         ///< OrderVolume, 委托数量
    string OrderCode;        ///< 委托买卖方向
    string OrderType;        ///< 订单类别
};

////! L2委托队列(上海、深圳)
struct QtSubL2OrderQueue
{
    string Symbol;          ///< 证券代码
    string Market;          ///< 交易所

    int TradingDate;        ///< 交易日期
    string TradingTime;      ///< 交易时间
    int64 UNIX;             ///< UTC时间戳

    int NPL;                ///< NoPriceLevel, 买卖盘价位数量
};

```

```

double Price;           ///< 委托价格
int64 Numorders;        ///< 最优价位上的委托笔数
uint Noorders;          ///< 发布的委托笔数
double VOL;             ///< 委托量
DoubleSeq OrderVolume;  ///< 委托量01-50
string BS;              ///< Side,买卖方向
double CP;              ///< 现价
};

///< L2委托统计行情
struct QtSubL2OrderStatistic
{
    string Symbol;        ///< 证券代码
    string Market;        ///< 交易所

    int TradingDate;      ///< 交易日期
    string TradingTime;    ///< 交易时间
    int64 UNIX;           ///< UTC时间戳

    double TBOVOL;        ///< TotalBuyOrderVolume,委托买入总量
    double WAvgBP;        ///< WtAvgBuyPrice,加权平均委买价格
    double TSOVOL;        ///< TotalSellOrderVolume,委托卖出总量
    double WAvgSP;        ///< WtAvgSellPrice,加权平均委卖价格
    string RecTime;       ///< 接收时间
};

///< L2实时结算价
struct QtSubL2SettlePrice
{
    string Symbol;        ///< 证券代码
    string Market;        ///< 交易所

    int TradingDate;      ///< 交易日期
    string TradingTime;    ///< 交易时间
    int64 UNIX;           ///< UTC时间戳

    //高频业务通用字段
    double RealSP;        ///< RealTimePrice,实时结算价
    string RecTime;       ///< 接收时间
};

///< L2证券状态(深圳)
struct QtSubL2Status
{
    string Symbol;        ///< 证券代码
    string Market;        ///< 交易所

    int TradingDate;      ///< 交易日期
    string TradingTime;    ///< 交易时间
    int64 UNIX;           ///< UTC时间戳

    string FinancialStatus; ///< 证券状态信息
    string CrdBuyStatus;    ///< 当前融资开放状态
    string CrdSellStatus;   ///< 当前融券开放状态
    string SubscribeStatus; ///< 申购标志
    string RedemptionStatus; ///< 赎回标志
    string PurchasingStatus; ///< 认购标志
    string StockDiviStatus; ///< 转股标志
    string PutableStatus;   ///< 回售标志
    string ExerciseStatus;  ///< 行权标志
    string GoldPurchase;    ///< 黄金ETF实物申购标志
    string GoldRedemption;  ///< 黄金ETF实物赎回标志
    string AcceptedStatus;  ///< 预受要约标志
    string ReleaseStatus;   ///< 解除要约标志
    string CancStockDiviStatus; ///< 转股撤单标志
    string CancPutableStatus; ///< 回售撤单标志
    string PledgeStatus;    ///< 质押标志
    string RemovePledge;    ///< 解押标志
    string VoteStatus;      ///< 表决权标志
};

```

```

    string StockPledgeRepo;    ///< 股票质押式回购标志
    string DivideStatus;      ///< 实时分拆标志
    string MergerStatus;      ///< 实时合并标志
    string SymbolSource;      ///< 证券代码源
};

///< L2逐笔成交(上海、深圳)
struct QtSubL2Transaction
{
    string Symbol;            ///< 证券代码
    string Market;           ///< 交易所

    int TradingDate;         ///< 交易日期
    string TradingTime;      ///< 交易时间
    int64 UNIX;              ///< UTC时间戳

    double TradePrice;       ///< 成交价格
    double TradeVolume;      ///< 成交数量
    uint RecID;              ///< 成交序号(消息记录号)
    int TradeChannel;        ///< 成交通道
    double TradeAmount;      ///< 成交金额
    int BRecID;              ///< 买方订单号
    int SRecID;              ///< 卖方订单号
    string BSFlag;           ///< BuySellFlag, 内外盘标志
    string SymbolSource;     ///< 证券代码源
    int SetID;               ///< 频道代码
    int64 BOID;              ///< BuyOrderID, 买方委托索引
    int64 SOID;              ///< SellOrderID, 卖方委托索引
    string TradeType;        ///< 成交类型
};

//行情回调函数类型(需用户实现)
typedef void (*SubQuoteCB)(QtSubQuote& data);

//L2十档行情(上海、深圳)回调函数类型(需用户实现)
typedef void (*SubL2QuoteCB)(QtSubL2Quote& data);

//L1套利行情(大商所)回调函数类型(需用户实现)
typedef void (*SubArbitrageCB)(QtSubArbitrage& data);

//L1套利行情(大商所)回调函数类型(需用户实现)
typedef void (*SubArbitrageCB)(QtSubArbitrage& data);

//静态数据(上海L2、深圳L2、上交所个股期权)回调函数类型(需用户实现)
typedef void (*SubStaticCB)(QtSubStatic& data);

//公告及增值信息(深圳L1)回调函数类型(需用户实现)
typedef void (*SubBulletinCB)(QtSubBulletin& data);

//L2集合竞价(上海)回调函数类型(需用户实现)
typedef void (*SubL2AuctionCB)(QtSubL2Auction& data);

//L2指数行情(上海、深圳)回调函数类型(需用户实现)
typedef void (*SubL2IndexCB)(QtSubL2Index& data);

//L2逐笔委托(深圳)回调函数类型(需用户实现)
typedef void (*SubL2OrderCB)(QtSubL2Order& data);

//L2委托队列(上海、深圳)回调函数类型(需用户实现)
typedef void (*SubL2OrderQueueCB)(QtSubL2OrderQueue& data);

//L2十档行情(上海、深圳)回调函数类型(需用户实现)
typedef void (*SubL2QuoteCB)(QtSubL2Quote& data);

//L2逐笔成交(上海、深圳)回调函数类型(需用户实现)
typedef void (*SubL2TransactionCB)(QtSubL2Transaction& data);

//L2证券状态(深圳)回调函数类型(需用户实现)
typedef void (*SubL2StatusCB)(QtSubL2Status& data);

```



```

//L2套利行情回调函数类型（需用户实现）
typedef void(*SubL2ArbitrageCB)(QtSubL2Arbitrage& data);

//L2分价成交量行情回调函数类型（需用户实现）
typedef void(*SubL2MarchPriceQtyCB)(QtSubL2MarchPriceQty& data);

//L2委托统计行情回调函数类型（需用户实现）
typedef void(*SubL2OrderStatisticCB)(QtSubL2OrderStatistic& data);

//L2实时结算价回调函数类型（需用户实现）
typedef void(*SubL2SettlePriceCB)(QtSubL2SettlePrice& data);

```

## 6 附录 2 错误码定义

### 6.1 红楹云数据接口错误码

错误码为 32 位无符号整型值（4 个字节），以 16 进制方式定义和使用，以下说明中，以最高字节为第 1 字节，最低字节为第 4 字节。。

- 返回值 0 表示调用成功；
- 首字节为“01”时表示错误视为通信服务平台的错误；
- 首字节为“04”时表示错误是 QtAPI 系统的错误。

首字节为“04”的 QtAPI 错误码按接口分段区分，并组合具体错误类型使用：

- 第 1 字节：04，表示是 QtAPI 本地或远程错误；
- 第 2 字节：用于区分是交易 API 还是数据 API 等；
- 第 3 字节：用于区分接口具体的 API 接口；
- 第 4 字节：用于指明具体的错误原因，用户通常只关注此字节即可。

```

#define QR_Success          0    //0x00000000, 成功
#define QR_Connect         -1   //0xFFFFFFFF, 连接异常
// #define QR_SendTimeout  -2   //0xFFFFFFF, 报文发送超时
#define QR_MsgParseFail    -3   //0xFFFFFFF, 报文解析失败
// #define QR_FileErr      -4   //0xFFFFFFF, 文件访问失败（如文件不存在等）
#define QR_UnknownException -5  //0xFFFFFFF, 未知异常
#define QR_SendFail       -6   //0xFFFFFFF, 报文发送失败

//0.基准定义
#define QR_Quote          0x04010000 //数据API错误码基准编号
#define QR_Trade          0x04020000 //交易API错误码基准编号
#define QR_QtAPI          0x04030000 //QtAPI错误码基准编号(不属于交易API或数据API的接口)

//1.数据API错误码基准
#define QR_Quote_Exchanges 0x04010100 //ReqExchanges错误码基准编号
#define QR_Quote_SecurityTypes 0x04010200 //ReqSecurityTypes错误码基准编号
#define QR_Quote_TradingTypes 0x04010300 //ReqTradingTypes错误码基准编号
#define QR_Quote_TradeCalendar 0x04010400 //ReqTradeCalendar错误码基准编号
#define QR_Quote_Plates 0x04010500 //ReqPlates错误码基准编号
#define QR_Quote_RelatedPlates 0x04010600 //ReqRelatedPlates错误码基准编号
#define QR_Quote_PlateSymbols 0x04010700 //ReqPlateSymbols错误码基准编号
#define QR_Quote_SecurityInfo 0x04010800 //ReqSecurityInfo错误码基准编号
#define QR_Quote_SecurityCurInfo 0x04010900 //ReqSecurityCurInfo错误码基准编号
#define QR_Quote_DataByTime 0x04010A00 //ReqDataByTime错误码基准编号
#define QR_Quote_DataByCount 0x04010B00 //ReqDataByCount错误码基准编号
#define QR_Quote_TickByTime 0x04010C00 //ReqTickByTime错误码基准编号

```



```

#define QR_Quote_TickByCount      0x04010D00    //ReqTickByCount错误码基准编号
#define QR_Quote_SnapData         0x04010E00    //ReqSnapData错误码基准编号
#define QR_Quote_Finance          0x04010F00    //ReqFinance错误码基准编号
#define QR_Quote_Factor           0x04011000    //ReqFactor错误码基准编号
#define QR_Quote_HisMarketInfo    0x04011100    //ReqHisMarketInfo错误码基准编号
#define QR_Quote_JointContracts   0x04011200    //ReqJointContracts错误码基准编号
#define QR_Quote_L2TickByTime     0x04011400    //ReqL2TickByTime错误码基准编号
#define QR_Quote_L2TickByCount    0x04011500    //ReqL2TickByCount错误码基准编号

#define QR_Quote_Subscribe        0x04012000    ///< Subscribe错误码基准编号
#define QR_Quote_UnSubscribe      0x04012100    ///< UnSubscribe错误码基准编号
#define QR_Quote_SubscribeEx      0x04012200    ///< SubscribeEx错误码基准编号
#define QR_Quote_UnSubscribeEx    0x04012300    ///< UnSubscribeEx错误码基准编号
#define QR_Quote_RegSubQuoteCB    0x04012400    ///< RegSubQuoteCB错误码基准编号
#define QR_Quote_RegSubStsCB      0x04012500    ///< RegSubStsCB错误码基准编号
#define QR_Quote_RegSubL2QuoteCB  0x04012600    ///< RegSubL2QuoteCB错误码基准编号
#define QR_Quote_RegSubArbitrageCB 0x04012700    ///< L1套利行情(大商所)
#define QR_Quote_RegSubStaticCB   0x04012800    ///< 静态数据(上海L2、深圳L2、上交所个股期权)
#define QR_Quote_RegSubBulletinCB 0x04012900    ///< 公告及增值信息(深圳L1)
#define QR_Quote_RegSubL2AuctionCB 0x04012A00    ///< L2集合竞价(上海)
#define QR_Quote_RegSubL2IndexCB  0x04012B00    ///< L2指数行情(上海、深圳)
#define QR_Quote_RegSubL2OrderCB  0x04012C00    ///< L2逐笔委托(深圳)
#define QR_Quote_RegSubL2OrderQueueCB 0x04012D00    ///< L2委托队列(上海、深圳)
#define QR_Quote_RegSubL2TransactionCB 0x04012E00    ///< L2逐笔成交(上海、深圳)
#define QR_Quote_RegSubL2StatusCB  0x04012F00    ///< L2证券状态(深圳)
#define QR_Quote_RegSubL2ArbitrageCB 0x04013000    ///< L2套利行情
#define QR_Quote_RegSubL2MarchPriceQtyCB 0x04013100    ///< L2分价成交量行情
#define QR_Quote_RegSubL2OrderStatisticCB 0x04013200    ///< L2委托统计行情
#define QR_Quote_RegSubL2SettlePriceCB 0x04013300    ///< L2实时结算价

//2. QtAPI错误码基准编号(不属于交易API或数据API的接口)
#define QR_QtAPI_Login            0x04020100    //QtAPI登录
#define QR_QtAPI_Logout           0x04020200    //QtAPI登出

//3.交易API错误码基准
#define QR_Trade_AccountLogin      0x04020300    //柜台登入
#define QR_Trade_AccountLogout    0x04020400    //柜台登出
#define QR_Trade_PlaceOrder       0x04020500    //下单
#define QR_Trade_CancelOrder      0x04020600    //撤单
#define QR_Trade_QueryCapital     0x04020700    //查询资金
#define QR_Trade_QueryPosition    0x04020800    //查询持仓
#define QR_Trade_QueryOrder       0x04020900    //查询订单状态
#define QR_Trade_QueryTradePar    0x04020A00    //查询交易参数

//具体错误码
#if 1
//<<
#define QR_Invalid_Param          0x00000001    //参数无效
#define QR_Invalid_Count         0x00000002    //参数数量或维度无效
#define QR_Invalid_Page          0x00000003    //数据分页/段信息无效
#define QR_DatabaseErr           0x00000004    //数据库错误
#define QR_InvalidAuth           0x00000005;    //权限无效或无相关权限
#define QR_RationExhaustion      0x00000006;    //定额用尽
#define QR_Invalid_ExchangeCode  0x00000011    //交易所代码无效
#define QR_Invalid_SecurityType  0x00000012    //证券类型代码无效
#define QR_Invalid_TradeType     0x00000013    //交易品种无效
#define QR_Invalid_Security      0x00000014    //证券代码无效
#define QR_Invalid_Symbol        0x00000015    //证券交易代码无效
#define QR_Invalid_Market        QR_Invalid_ExchangeCode
#define QR_Invalid_Plate         0x00000016    //板块ID无效
#define QR_Invalid_Date          0x00000017    //日期无效
#define QR_Invalid_DateTime      0x00000018    //时间无效
#define QR_Invalid_Field         0x00000019    //数据字段名无效
#define QR_StrLen_Limit          0x00000020    //字符串参数长度超超限
#define QR_OrderType_NotSupport  0x00000021    //柜台不支持订单类型
//add more here...
//>>

//<<

```

```

#define QR_MsgParse_Fail          0x00000080    //消息解析失败
#define QR_Invalid_Broker        0x00000081    //券商无效
#define QR_Invalid_CounterType   0x00000082    //柜台类型无效
#define QR_Invalid_AccountType   0x00000083    //无效账号类型
#define QR_Invalid_Account       0x00000084    //柜台账号或密码无效
#define QR_Invalid_Price         0x00000085    //价格无效
#define QR_Invalid_Volume        0x00000086    //数量无效
#define QR_Invalid_OrderID       0x00000087    //订单号无效
#define QR_Invalid_CltOrderID    0x00000088    //客户订单号无效
#define QR_Account_Offline       0x00000089    //账号未登录或离线
//add more here...
//>>

//<<
#define QR_Invalid_GTAUserPwd    0x000000F1    //用户或密码无效
#define QR_Invalid_GTAToken      0x000000F2    //Token无效
#define QR_Invalid_GTARights     0x000000F3    //权限无效
#define QR_Invalid_GTAAppID      0x000000F4    //AppID无效
#define QR_Network_Fail          0x000000F5    //网络错误
#define QR_Quote_InitFailed      0x000000F6    //数据API初始化失败
#define QR_Trade_InitFailed      0x000000F7    //交易API初始化失败
#define QR_Retry_Limit           0x000000F8    //超出限定重试次数
#define QR_Timeout               0x000000F9    //超时
#define QR_Exception             0x000000FA    //异常
#define QR_CommSendError         0x000000FB    //发送异常
#define QR_CommRecvError         0x000000FC    //接收异常
#define QR_Configuration         0x000000FD    //系统配置或参数设置错误
//add more here...

#define QR_Fail                  0x000000FF    //失败（未定义或一般性错误）
//>>
#endif //1

```

## 6.2 DSP 错误码

错误代码说明：

第 1 个字节：01 表示 DSP 系统；

第 2 个字节：01 表示 SDK，02 表示 MDP，03 表示网关系统，04 表示节点服务，05 表示解析服务 08 表示订阅管理；

第 3、4 个字节：自定义错误代码。

```

const int DSP_RESULT_OK          = 0x00000000;    ///< 正常

///< @brief SDK 错误代码
/*****十进制表示*****/      /**十六进制表示**/      /**异常消息**/
const int DSP_RESULT_INVALID_CONFIG    = 0x01010001;    ///< 无效配置文件
const int DSP_RESULT_INIT_ERROR        = 0x01010002;    ///< 初始化失败
const int DSP_RESULT_START_ERROR       = 0x01010003;    ///< 启动失败
const int DSP_RESULT_CONNECT_ERROR     = 0x01010004;    ///< 连接失败
const int DSP_RESULT_CONNECT_TIMEOUT   = 0x01010005;    ///< 连接超时
const int DSP_RESULT_DISCONNECT        = 0x01010006;    ///< 连接已断开
const int DSP_RESULT_INVALID_PARAM     = 0x01010007;    ///< 非法参数
const int DSP_RESULT_SERIALIZE_ERROR   = 0x01010008;    ///< 序列化数据出错
const int DSP_RESULT_DESERIALIZE_ERROR = 0x01010009;    ///< 反序列化数据出错

```

```

const int DSP_RESULT_SEND_DATA_ERROR          = 0x0101000A;      ///< 发送数据失败
const int DSP_RESULT_RECV_DATA_TIMEOUT        = 0x0101000B;      ///< 接收数据超时
const int DSP_RESULT_SERVER_BUSY              = 0x0101000C;      ///< 服务端繁忙
const int DSP_RESULT_OUT_OF_MEMORY            = 0x0101000D;      ///< 内存不足
const int DSP_RESULT_INVALID_LINK_ID          = 0x0101000E;      ///< 无效连接号
const int DSP_RESULT_UNINIT                   = 0x0101000F;      ///< 未初始化
const int DSP_RESULT_UNSTART                  = 0x01010010;      ///< 未启动
const int DSP_RESULT_SEND_DATA_LARGE_ERROE   = 0x01010011;      ///< 发送数据过大, 大于 100M
const int DSP_RESULT_QUERY_SUBMANAGER_ERROE  = 0x01010012;      ///< 查询订阅管理失败
const int DSP_RESULT_UPDATE_SUBMANAGER_ERROE= 0x01010013;      ///< 更新订阅管理失败

const int DSP_DSPGATEWAY_RESULT_NOMEMORY_RESOURCE = 0x01030000;  //内存不足
const int DSP_DSPGATEWAY_RESULT_NET_PROTOCOL_PARSED = 0x01030001; //网络消息协议解析异常
const int DSP_DSPGATEWAY_RESULT_MQ_PROTOCOL_PARSED = 0x01030002;  //MQ 消息协议解析异常
const int DSP_DSPGATEWAY_RESULT_STATE_LINK        = 0x01030003;  //连接状态异常
const int DSP_DSPGATEWAY_RESULT_LINK_UNEEXIST     = 0x01030004;  //连接不存在
const int DSP_DSPGATEWAY_RESULT_AUTHENTICATION_FAIL = 0x01030005;  //鉴权失败异常
const int DSP_DSPGATEWAY_RESULT_GATEWAY_BUSY      = 0x01030006;  //网关繁忙异常
const int DSP_DSPGATEWAY_RESULT_MQ_SENDMSG_SEND   = 0x01030007;  //MQ 发送时发送异常
const int DSP_DSPGATEWAY_RESULT_NET_SEND          = 0x01030008;  //网络发送异常
const int DSP_DSPGATEWAY_RESULT_MQ_MISSINGQUEUE   = 0x01030009; //发送请求时, 协议的 typeid 找不到对应的 MQ 管道
const int DSP_DSPGATEWAY_RESULT_MQ_MSGRECV        = 0x0103000A;  //接收到的 MQ 消息异常
const int DSP_DSPGATEWAY_RESULT_NET_PROTOCOL_PACKED = 0x0103000B;  //网络消息协议打包异常
const int DSP_DSPGATEWAY_RESULT_MQ_PROTOCOL_ROUTER = 0x0103000C;  //MQ 消息协议 router 异常
const int DSP_DSPGATEWAY_RESULT_MQ_PROTOCOL_PACKED = 0x0103000D;  //MQ 消息协议打包异常
const int DSP_DSPGATEWAY_RESULT_NET_PROTOCOL_LINKID = 0x0103000E;  //网络消息协议 linkid 异常
const int DSP_DSPGATEWAY_RESULT_NET_PROTOCOL_SETLINKID = 0x0103000F;  //网络消息协议设置 linkid 异常
const int DSP_DSPGATEWAY_RESULT_NET_OVER_MAXLINK  = 0x01030010;  //网络连接超过配置, 无法设置连接号
const int DSP_PARSERSERVER_RESULT_INIT_ERROR      = 0x01050001;  ///< 初始化失败
const int DSP_PARSERSERVER_RESULT_INVALID_PARAM   = 0x01050002;  ///< 非法参数
const int DSP_PARSERSERVER_RESULT_GET_MSG_FIELD_ERROR = 0x01050003;  ///< 获取消息包中的字段出错
const int DSP_PARSERSERVER_RESULT_SET_MSG_FIELD_ERROR = 0x01050004;  ///< 设置消息包中的字段出错

```

|   |               |                    |
|---|---------------|--------------------|
| const int DSP_PARSERSERVER_RESULT_CLEAR_MSG_FIELD_ERROR       | = 0x01050005; | ///< 清除消息包中的字段出错   |
| const int DSP_PARSERSERVER_RESULT_SERIALIZE_ERROR             | = 0x01050006; | ///< 序列化数据出错       |
| const int DSP_PARSERSERVER_RESULT_PARSE_ERROR                 | = 0x01050007; | ///< 反序列化数据出错      |
| const int DSP_PARSERSERVER_RESULT_OUT_OF_MEMORY               | = 0x01050008; | ///< 内存不足          |
| const int DSP_PARSERSERVER_RESULT_NO_NEED_SUB_OR_UNSUB        | = 0x01050009; | ///< 无需重新退订或者订阅    |
| const int DSP_PARSERSERVER_RESULT_MSG_TYPE_UNLINK_TIMEOUT_REQ | = 0x0105000A; | ///< 网关超时断线退订的返回码  |
| const int DSP_PARSERSERVER_RESULT_NO_MERG_SUB_OR_UNSUB_REQ    | = 0x0105000B; | ///< 不可拆分的订阅或者退订请求 |
| const int DSP_PARSERSERVER_RESULT_SEND_SUB_TO_MQ_ERROR        | = 0x0105000C; | ///< 发送订阅到 MQ 失败   |
| const int DSP_SUBMANAGER_RESULT_INIT_ERROR                    | = 0x01080001; | ///< 初始化失败         |
| const int DSP_SUBMANAGER_RESULT_INVALID_PARAM                 | = 0x01080002; | ///< 非法参数          |
| const int DSP_SUBMANAGER_RESULT_GET_MSG_FIELD_ERROR           | = 0x01080003; | ///< 获取消息包中的字段出错   |
| const int DSP_SUBMANAGER_RESULT_SET_MSG_FIELD_ERROR           | = 0x01080004; | ///< 设置消息包中的字段出错   |
| const int DSP_SUBMANAGER_RESULT_CLEAR_MSG_FIELD_ERROR         | = 0x01080005; | ///< 清除消息包中的字段出错   |
| const int DSP_SUBMANAGER_RESULT_SERIALIZE_ERROR               | = 0x01080006; | ///< 序列化数据出错       |
| const int DSP_SUBMANAGER_RESULT_PARSE_ERROR                   | = 0x01080007; | ///< 反序列化数据出错      |
| const int DSP_SUBMANAGER_RESULT_OUT_OF_MEMORY                 | = 0x01080008; | ///< 内存不足          |
| const int DSP_SUBMANAGER_RESULT_NO_NEED_SUB_OR_UNSUB          | = 0x01080009; | ///< 无需重新退订或者订阅    |
| const int DSP_SUBMANAGER_RESULT_MSG_TYPE_UNLINK_TIMEOUT_REQ   | = 0x0108000A; | ///< 网关超时断线退订的返回码  |
| const int DSP_SUBMANAGER_RESULT_NO_MERG_SUB_OR_UNSUB_REQ      | = 0x0108000B; | ///< 不可拆分的订阅或者退订请求 |
| /// @brief MDP 错误代码   |               |                    |
| const int DSP_MDP_RESULT_INVALID_CONFIG                       | = 0x01020001; | ///< 无效配置文件        |
| const int DSP_MDP_RESULT_LISTEN_ERROR                         | = 0x01020002; | ///< 启动监听服务失败      |
| const int DSP_MDP_RESULT_START_ERROR                          | = 0x01020003; | ///< 启动 MDP        |

|  |               |                         |
|--|---------------|-------------------------|
| 失败   |               |                         |
| const int DSP_MDP_RESULT_INVALID_PARAM                 | = 0x01020004; | ///< 非法参数               |
| const int DSP_MDP_RESULT_REQ_MSG_DISPATCH_FAILED       | = 0x01020005; | ///< 请求消入队列失败           |
| const int DSP_MDP_RESULT_DATA_INSERT_CACHE_FAILED      | = 0x01020006; | ///< 数据插入缓存失败           |
| const int DSP_MDP_RESULT_DATA_NO_EXIST_FAILED          | = 0x01020007; | ///< 查找的数据不存在           |
| const int DSP_MDP_RESULT_PARSE_ERROR                   | = 0x01020008; | ///< 反序列化数据出错           |
| const int DSP_MDP_RESULT_DEAL_REQMSG_FAILED            | = 0x01020009; | ///< 处理请求消息失败           |
| const int DSP_MDP_RESULT_DEAL_RESPMSG_FAILED           | = 0x0102000A; | ///< 处理响应消息失败           |
| const int DSP_MDP_RESULT_OUT_OF_MEMORY                 | = 0x0105000B; | ///< 内存不足               |
| const int DSP_MDP_RESULT_HANDLE_REQREP_REQUEST_FAILED  | = 0x0102000C; | ///< 处理请求响应的请求失败        |
| const int DSP_MDP_RESULT_HANDLE_REQREP_RESPONSE_FAILED | = 0x0102000D; | ///< 处理请求响应的响应失败        |
| const int DSP_MDP_RESULT_HANDLE_SUBPUB_REQUEST_FAILED  | = 0x0102000E; | ///< 处理订阅请求或退订请求失败      |
| const int DSP_MDP_RESULT_HANDLE_SUBPUB_RESPONSE_FAILED | = 0x01020010; | ///< 处理订阅响应或退订响应失败      |
| const int DSP_MDP_RESULT_NET_OVER_MAXLINK              | = 0x01020011; | ///< 网络连接超过配置, 无法设置会话号  |
| const int DSP_MDP_RESULT_NET_SET_SESSIONID             | = 0x01020012; | ///< 因为连接状态等原因, 无法设置会话号 |
| const int DSP_MDP_RESULT_NET_STOP_SESSIONID            | = 0x01020013; | ///< 因为连接状态等原因, 无法停止会话号 |
| const int DSP_MDP_RESULT_EXCEPTION_SYSMMSG             | = 0x01020014; | ///< 系统消息处理模块返回异常消息     |
| const int DSP_MDP_RESULT_DEAL_SYSMMSG_FAILED           | = 0x01020015; | ///< 处理系统消息失败           |
| const int DSP_MDP_RESULT_SERVER_BUSY                   | = 0x01020016; | ///< MDP 系统繁忙           |

## 7 附录 3 高频数据库内容

| 高频类别                 | 来源                 | 数据内容     | 备注                                    |
|----------------------|--------------------|----------|---------------------------------------|
| 中国证券市场<br>Level_1 高频 | 上海证券交易所<br>深圳证券交易所 | 分笔高频交易数据 | 沪深两家交易所公布的各类证券品种代码等。包含股票、债券、基金、权证、指数等 |

|                      |         |             |                                      |
|----------------------|---------|-------------|--------------------------------------|
|                      |         | 分时高频数据数据    | 包含 1 分时、5 分时、10 分时、15 分时、30 分时、60 分时 |
| 上海证券市场<br>Level_2 高频 | 上海证券交易所 | 指数行情        | 提供上证指数, 中证指数, 沪深 300 指数等指数的行情信息      |
|                      |         | 逐笔成交        | 提供证券的逐笔成交明细数据                        |
|                      |         | 十档行情快照      | 提供 3 秒/次的包含 10 档报价的市场行情快照            |
|                      |         | 集合竞价        | 9:15—9:25 期间的集合竞价数据                  |
|                      |         | 委托队列        | 提供最佳买卖价位上最多 50 个订单队列                 |
| 深圳证券市场<br>Level_2 高频 | 深圳证券交易所 | 证券静态信息      | 提供不同种类证券基本必须的证券静态信息                  |
|                      |         | 证券状态        | 提供证券实时变动的动态状态信息                      |
|                      |         | 十档行情快照      | 提供 3 秒/次的包含 10 档报价的市场行情快照            |
|                      |         | 最优买卖委托队列    | 提供最佳买卖价位上最多 50 个订单队列                 |
|                      |         | 指数快照        | 提供上证指数, 中证指数, 沪深 300 指数等指数的行情信息      |
|                      |         | 逐笔委托        | 提供证券的逐笔委托明细数据                        |
|                      |         | 逐笔成交        | 提供证券的逐笔成交明细数据                        |
|                      |         | 公告及增值信息     | 提供关于市场和证券的公告文件和增值信息文件                |
| 中国股指期货<br>高频         | 中金所     | 股指期货分笔高频交易表 | 基于沪深 300 指数的金融期货合约                   |
|                      |         | 股指期货分时高频交易表 | 包含 1 分时、5 分时、10 分时、15 分时、30 分时、60 分时 |

|          |        |                          |                                      |
|----------|--------|--------------------------|--------------------------------------|
| 中国国债期货高频 | 中金所    | 国债期货分笔高频交易表              | 基于 5 年期国债的金融期货合约                     |
|          |        | 国债期货分时高频交易表 (TFL2_TrMin) | 包含 1 分时、5 分时、10 分时、15 分时、30 分时、60 分时 |
| 中国商品期货高频 | 大商所    | 商品期货分笔高频交易               | 统一了三家商品期货交易所的数据结构                    |
|          | 郑商所    | 商品期货分时高频交易               | 包含 1 分时、5 分时、10 分时、15 分时、30 分时、60 分时 |
|          | 上期所    | 商品期货套利                   | 只有大商所套利数据                            |
| 中国个股期权高频 | 上交所    | 个股期权静态信息                 | 提供不同个股期权基本必须的静态信息                    |
|          |        | 个股期权分笔数据                 | 包含上交所公布的各个期权代码                       |
|          |        | 个股期权分时数据                 | 包含 1 分时、5 分时、10 分时、15 分时、30 分时、60 分时 |
| 中国商品期权高频 | 大商所郑商所 | 商品期权分笔数据                 | 郑商所商品期权                              |
|          |        | 商品期权分时数据                 | 郑商所商品期权                              |
|          |        | 商品期权 L2 分笔数据             | 大商所商品期权                              |
|          |        | 商品期权 L2 分价成交量行情          | 大商所商品期权                              |
|          |        | 商品期权 L2 委托统计行情           | 大商所商品期权                              |
|          |        | 商品期权 L2 实时结算价            | 大商所商品期权                              |

## 8 FAQ

### 8.1 数据提取结果与输出文件的说明

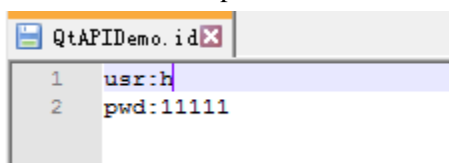
对于数据提取结果以直接调试查看为准，输出文件用 notepad++ 查看为准（以 excel 打开 csv 存在显示异常），对于汉字显示乱码，调整阅读器的编码格式（GBK->UTF-8 即可）。

## 8.2 红楹云数据接口 C++版本安装路径说明

红楹云数据接口 C++版本默认安装路径为 C:\QtAPI\cplusplus，可以修改默认地址。红楹云数据接口 C++版本和 Python 版本的安装包会统一安装到 C:\QtAPI 这个目录下，如果卸载 C++版本，则只会删除 cplusplus 这个文件夹，不会删除 QtAPI 这个总文件夹。修改安装包命名的时候，不能含有中文。

## 8.3 红楹云数据接口配置账号密码的文件说明

红楹云数据接口账号密码配置文件为 QtAPIDemo.id 文件，打开此文件后输入有效的用户名（usr）与密码（pwd）才能成功登录红楹云数据接口。



## 8.4 修改数据接口返回超时时限的文件说明

相应数据接口接收数据超时时限配置文件为

： C:\QtAPI\cplusplus\bin\x86\debug\QtAPI.xml 文件，

<API type\_id="10" msg\_id="1" version="2.0" timeout="10">Subscribe</API> 这里表示：Subscribe 订阅接口超时返回时限为 10 秒，如果需要设置为 20 秒或更长时间，只需改为 timeout = “20”。

```
<DataAPI>
  <API type_id="10" msg_id="1" version="2.0" timeout="10">Subscribe</API>
  <API type_id="10" msg_id="1" version="2.0" timeout="10">UnSubscribe</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="10">GetExchanges</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="10">GetSecurityTypes</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="10">GetTradeTypes</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="10">GetTradeCalendar</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="20">GetPlates</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="20">GetPlateSymbols</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="20">GetRelatedPlates</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="10">GetSecurityInfo</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="10">GetSecurityCurInfo</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="60">GetDataByTime</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="60">GetDataByCount</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="30">GetTickByTime</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="30">GetTickByCount</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="30">GetL2TickByTime</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="30">GetL2TickByCount</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="5">GetSnapData</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="60">GetFinance</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="30">GetFactor</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="30">GetHisMarketInfo</API>
  <API type_id="01" msg_id="0" version="1.2.0.0" timeout="10">GetJointContracts</API>
```



## 8.5 配置数据接口生产环境路径

数据接口的环境配置应该为生产环境地址，生产环境地址如下所示：

```
<?xml version="1.0" encoding="utf-8"?>
<QtAPI>
  <!--UPMS测试环境配置信息-->
  <!--<UPMS protocol="http" ip="10.1.130.23" port="8002" >/UPMS-->
  <!--UPMS生产环境配置信息-->
  UPMS protocol="https" ip="upmsapi.gtarsc.com" port="443" >/UPMS
```

## 8.6 免费接口权限说明

红楹云数据接口免费接口包括：GetExchanges、GetTradeTypes 和 GetTradeCalendar，这三个接口在仅有红楹云数据接口账号的情况下就可以取数成功，不需开特定的权限。

## 8.7 金融期货与国债期货数据说明

金融期货与国债期货的数据源于同一张表，所以用 S0108 或 S0109 取回的数据均包含金融期货和国债期货的所有字段内容。

## 8.8 按市场或按板块订阅说明

按市场或按板块订阅数据时，当前仅支持按单个市场或按单个板块订阅。如果输入了多个市场或板块，数据结果只默认输出第一个市场或板块的数据。

## 8.9 代码排序说明

此版本中 GetDataByTime、GetDataByCount 和 GetSnapData 这三个接口可以按照时间和 Symbol 进行排序。其他接口的其他排序形式暂不支持。