

# TDB 系统 原生 C++ API 接口说明书

---

上海万得信息技术股份有限公司

**Shanghai Wind Information Co., Ltd.**

地 址: 上海市浦东新区福山路 33 号建工大厦 9 楼

邮编 Zip: 200120

电话 Tel: (8621) 6888 2280

传真 Fax: (8621) 6888 2281

Email: [sales@wind.com.cn](mailto:sales@wind.com.cn)

<http://www.wind.com.cn>

文档管理信息表

主题	TDB 系统 原生 C++ API 接口说明书 V3.0
版本	3.0
内容	TDB 系统 C++ API 接口说明书
关键字	TDB, API, C++, 接口说明书
参考文档	TDBAPI C++接口说明书 v1.1.docx
创建时间	2015-11-9
创建人	唐龙涛
最新发布日期	2016-08-04

文档变更记录表

A – 增加 M – 修订 D – 删除

版本号	修改时间	修改人	变更类型	修改内容
v1.0	2015-11-09	唐龙涛	M	原生压缩版本接口调整
v3.0	2016-03-24	唐龙涛	M	–
v3.0	2016-08-04	唐龙涛	M	新增证券类型细分定义

## 目录

1	概述	5
2	系统特性	5
3	数据说明	6
4	数据类型	6
4.1	证券代码表	6
4.1.1	上证指数代码转换对应表	6
4.1.2	证券类型基础定义	7
4.1.3	证券类型细分定义	8
4.2	K 线	10
4.2.1	请求 K 线数据结构	10
4.2.2	K 线数据结构	11
4.3	Tick	12
4.3.1	请求 Tick 数据结构	12
4.3.2	Tick 数据结构	12
4.4	逐笔成交 (Transaction)	13
4.4.1	请求逐笔成交数据结构	13
4.4.2	逐笔成交数据结构	13
4.5	逐笔委托 (Order)	14
4.5.1	请求逐笔委托数据结构	14
4.5.2	逐笔委托数据结构	15
4.6	委托队列 (OrderQueue)	16
4.6.1	请求委托队列数据结构	16
4.6.2	委托队列数据结构	16
4.7	请求返回值	17
5	调用 API 方法	17
5.1	登录	17
5.2	代理登录	18
5.3	获取 K 线数据	19
5.4	获取 Tick 数据	19
5.4.1	获取 Tick 数据	19
5.5	获取逐笔成交数据	19
5.6	获取逐笔委托数据	19
5.7	获取委托队列数据	20
5.8	计算指标公式	20
5.8.1	上载指标公式	20
5.8.2	获取指标公式	20
5.8.3	删除指标公式	21
5.8.4	计算指标公式	21
5.8.5	服务器指标编写实例	22
5.9	代码表	22
5.9.1	获取代码表	22
5.9.2	获取代码信息	23
5.10	关闭	23

6	TDB 系统结构.....	23
7	TDB 常见问题.....	25

## 1 概述

本文档介绍了 TDB 系统现有的数据类型以及数据结构，介绍了 TDBAPI 使用方式。并且规定了相关协议。

## 2 系统特性

### ◆ 专业性

公司有着十余年证券行情数据处理经验，拥有业界先进的行情压缩、数据处理及数据存储技术的 TSDB 数据库，数据访问速度更快、延时更小，占用网络带宽更少，应用更加灵活。系统使用 TCP/IP 通讯协议，用户可以跨平台使用，并提供 C/C++ 实例程序。

### ◆ 高效性

采用适合证券数据的 TSDB 数据储存方式(宏汇证券数据库)，并非基于 SQLServer、Oracle 等数据库，不仅大大提高了数据的访问速度，且占用的存储空间更少，具有更强大的证券数据处理能力。

宏汇证券数据库（TSDB）特点：

1. 采取专业数据压缩技术，占用空间更小
2. 提供了完整的历史数据
3. 更易于维护

### ◆ 完整性

本系统可提供沪深市场及国内期货交易所等各类历史行情数据，涵盖内容全面。

#### 1. K 线数据

- 1) 日、周、月、季、年线数据
- 2) 分钟线数据
- 3) 秒、Tick 数据
- 4) 上述 K 线各个周期数据，如 2 秒 K 线等
2. Tick 数据，可根据需要提取是否包含买卖盘
3. 逐笔成交数据(深沪 Level-2)
4. 逐笔委托数据(深 Level-2)
5. 前 50 笔委托队列(深沪 Level-2)

说明：

访问历史数据时，可指定访问的时间段、是否除权等，较为灵活；

包含当日实时数据；

数据内容可以根据客户需求扩展；

支持任意时间段变频；

#### ◆ 安全性

采用宏汇独有专业数据加密技术，防止数据被非法篡改，护航业务系统。

#### ◆ 易扩展

模块化架构易于系统扩展。可选择不同的市场数据、不同的数据周期，如上海 Level-2 行情、深圳 Level-2 行情、深沪 Level-1 行情以及股指期货等行情，当增加某市场行情时，只需增加对应程序即可，无需修改接口以及第三方应用终端。

#### ◆ 易部署

系统构架方便，部署和安装较为简单，客户可结合自身的特点灵活选择部署方式。

## 3 数据说明

所有价格都用 INT 表示，其值为(实际价格+0.00005)\*10000

例如：成交价为 16.80 表示为 168000

## 4 数据类型

### 4.1 证券代码表

字段	类型	描述
<b>TDBDefine_Code</b>		
chWindCode	char[32]	万得代码，如：000001.SZ
chCode	char[32]	交易所代码，如：000001
chMarket	char[24]	市场，如：SZ-2-0(上海-L2-Source0)
chCNName	char[32]	证券中文名称
chENName	char[32]	证券英文名称
nType	int	证券类型

证券代码表是由数据商每日产生，记录每一个证券代码的交易代码、所属市场、证券名称、证券类型等信息。

#### 4.1.1 上证指数代码转换对应表

指数名称	原上证指数代码	转换后代码
上证指数	000001	999999
上证 A 股	000002	999998
上证 B 股	000003	999997
工业指数	000004	999996

指数名称	原上证指数代码	转换后代码
商业指数	000005	999995
地产指数	000006	999994
公用指数	000007	999993
综合指数	000008	999992
上证 180	000010	999991
基金指数	000011	999990
国债指数	000012	999989
企债指数	000013	999988
上证 50	000016	999987
红利指数	000015	999986
沪深 300	000300	000300

注：其他指数是把指数前两位改为 00，即对应上证指数代码；例如“两岸三地”的代码为 990999，其对应的上证代码是 000999。

#### 4.1.2 证券类型基础定义

类型名称	宏定义	类型值
指数	ID_BASECLASS_INDEX	0x00
股票	ID_BASECLASS_SHARES	0x10
基金	ID_BASECLASS_FUND	0x20
债券 & 可转债	ID_BASECLASS_BOND	0x30
回购	ID_BASECLASS_REPO	0x40
权证	ID_BASECLASS_QZ	0x60
期货	ID_BASECLASS_FUTURES	0x70
外汇	ID_BASECLASS_FOREX	0x80
期权	ID_BASECLASS_OPTION	0x90
银行利率	ID_BASECLASS_BANKRATE	0xd0
贵金属(noble metal)	ID_BASECLASS_NMETAL	0xe0
正股(有权证的股票)	ID_BTEx_HASWARRANT	0x80000000
已经删除	ID_BTEx_DELETE	0x40000000
退市整理板块	ID_BTEx_DELISTING	0x08000000
风险警示板块	ID_BTEx_RISKWARNING	0x04000000
其他	ID_BASECLASS_OTHER	0xf0

## 4.1.3 证券类型细分定义

类型名称	宏定义	类型值
交易所指数	ID_BT_INDEX	0x01
亚洲指数	ID_BT_INDEX_ASIA	0x03
国际指数	ID_BT_INDEX_FOREIGN	0x04
系统分类指数	ID_BT_INDEX_HH	0x05
用户分类指数	ID_BT_INDEX_USER	0x06
期货指数	ID_BT_INDEX_FUTURES	0x07
指数现货	ID_BT_INDEX_MD	0x08
A 股	ID_BT_SHARES_A	0x10
中小板股	ID_BT_SHARES_S	0x11
创业板股	ID_BT_SHARES_G	0x12
B 股	ID_BT_SHARES_B	0x16
H 股(在香港上市的 A 股, 2008.12.4 为上海 Level2 行情 H 股增加)	ID_BT_SHARES_H	0x17
US	ID_BT_SHARES_US	0x1a
US ADR	ID_BT_SHARES_USADR	0x1b
扩展板块股票(港)	ID_BT_SHARES_E	0x1e
基金	ID_BT_FUND	0x20
未上市开放基金	ID_BT_FUND_OPEN	0x21
上市开放基金	ID_BT_FUND_LOF	0x22
交易型开放式指数基金	ID_BT_FUND ETF	0x23
扩展板块基金(港)	ID_BT_FUND_ETS	0x25
分级子基金	ID_BT_FUND_GRD	0x26
仅申赎基金	ID_BT_FUND_OAR	0x27
企业债券	ID_BT_BOND_CORP	0x31
金融债券	ID_BT_BOND_FIN	0x32
可转债	ID_BT_BOND_CON	0x33
债券预发行	ID_BT_BOND_WI	0x34



类型名称	宏定义	类型值
国债回购	ID_BT_REPO_NA	0x40
企债回购	ID_BT_REPO_CORP	0x41
权证	ID_BT_QZ	0x60
认购权证	ID_BT_QZ_C	0x61
认沽权证	ID_BT_QZ_P	0x62
认购权证(B 股)	ID_BT_QZ_C_B	0x64
认沽权证(B 股)	ID_BT_QZ_P_B	0x65
牛证 (moo-cow)	ID_BT_QZ_M	0x66
熊证 (bear)	ID_BT_QZ_B	0x67
指数期货	ID_BT_FUTURES_IDX	0x70
商品期货	ID_BT_FUTURES	0x71
股票期货	ID_BT_FUTURES_SHA	0x72
同业拆借利率期货	ID_BT_FUTURES_IBOR	0x73
Exchange Fund Note Futures	ID_BT_FUTURES_EFN	0x74
Exchange For Physicals	ID_BT_FUTURES_EFP	0x75
Exchange of Futures For Swaps	ID_BT_FUTURES_EFS	0x76
指数期货连线 CX	ID_BT_FUTURES_CX_IDX	0x78
指数期货连线 CC	ID_BT_FUTURES_CC_IDX	0x79
商品期货连线 CX	ID_BT_FUTURES_CX	0x7a
商品期货连线 CC	ID_BT_FUTURES_CC	0x7b
股票期货连线 CX	ID_BT_FUTURES_CX_SHA	0x7c
股票期货连线 CC	ID_BT_FUTURES_CC_SHA	0x7d
期现差价线	ID_BT_FUTURES_QX	0x7e
跨期差价线	ID_BT_FUTURES_KQ	0x7f
基本汇率	ID_BT_FOREX_BASE	0x80
交叉汇率	ID_BT_FOREX_CROSE	0x81

类型名称	宏定义	类型值
反向汇率	ID_BT_FOREX_REVERSE	0x82
认购期权	ID_BT_OPTION_C	0x90
认沽期权	ID_BT_OPTION_P	0x91
银行利率	ID_BT_BANKRATE	0xd0
银行利率 (HK)	ID_BT_BANKRATE_HK	0xd1
银行利率 (Interl)	ID_BT_BANKRATE_WORLD	0xd2
贵金属 (noble metal)	ID_BT_NMETAL	0xe0
其他	ID_BT_OTHERS	0xf0
A 股新股申购	ID_BT_OTHER_SHARES_SG_A	0xf1
A 股增发	ID_BT_OTHER_SHARES_ZF_A	0xf2

## 4.2 K 线

### 4.2.1 请求 K 线数据结构

请求 K 线数据结构:

字段	类型	描述
<b>TDBDefine_ReqKLine</b>		
chCode	char[32]	万得代码, 如: 000001.SZ
chMarketkey	char[24]	市场, 如: SZ-2-0(上海-L2-Source0)
nCQFlag	REFILLFLAG	除权标志
nCQDate	int	复权日期
nQJFlag	int	全价标志(债券)(0:净价, 1:全价)
nCycType	CYCTYPE	数据周期
nCycDef	int	周期数量
nAutoComplete	int	自动补齐(0:不补齐, 非 0:补齐)
nBeginDate	int	开始日期(0:从今天开始)
nEndDate	int	结束日期(0:从今天开始)
nBeginTime	int	开始时间
nEndTime	int	结束时间

除权标志宏定义:

宏定义	值	描述
-----	---	----

REFILLFLAG		
REFILL_NONE	0	不复权
REFILL_BACKWARD	1	全程向前复权(从现在向过去)
REFILL_FORWARD	2	全程向后复权(从过去向现在)

向前复权，就是保持现有价位不变，将以前的价格缩减，将除权前的 K 线向下平移，使图形吻合，保持股价走势的连续性。

向后复权，就是保存先前的价格不变，而将以后的价格增加。

数据周期宏定义：

宏定义	值	描述
CYCTYPE		
CYC_SECOND	0	秒线
CYC_MINUTE	1	分钟线
CYC_DAY	2	日线
CYC_WEEK	3	周线
CYC_MONTH	4	月线
CYC_SEASON	5	季线
CYC_HAFLYEAR	6	半年线
CYC_YEAR	7	年线
CYC_TICKBAR	8	TickBar 线

注 1：自动补齐功能，只支持秒 K 线、分钟 K 线数据补齐。K 线补齐是以前一笔 K 线数据生成不存在的后一笔，高、开、低都与前一笔一致，但是成交量、成交额都为 0。

注 2：分钟 K 线每次最多可提取 1 个自然月（31）天的数据如：2015. 1. 1-2015. 1. 31

错误设置：（2015. 1. 1-2015. 2. 1）超出 31 天则获取不到任何数据

注 3：秒线和 TickBar 线，每次最多可提取一周+1 天数据，计算时以每月 30 天计算

#### 4.2.2 K 线数据结构

字段	类型	描述
TDBDefine_KLine		
chWindCode	char[32]	万得代码，如：000001.SZ
chCode	char[32]	交易所代码，如：000001
nDate	int	日期(自然日 YYYYMMDD)
nTime	int	时间(HHMMSSmmm)
nOpen	int	开盘价
nHigh	int	最高价
nLow	int	最低价
nClose	int	收盘价
iVolume	long long	成交量
iTurover	long long	成交额(元)
nMatchItems	int	成交笔数
nInterest	int	持仓量(期货)、IOPV(基金)、

利息(债券)

## 4.3 Tick

### 4.3.1 请求 Tick 数据结构

字段	类型	描述
<b>TDBDefine_ReqTick</b>		
字段	类型	描述
chCode	char[32]	万得代码, 如: 000001.SZ
chMarketkey	char[24]	市场, 如: SZ-2-0(上海-L2-Source0)
nDate	int	请求日期(<0:取上市日期,0:取今天数据)
nBeginTime	int	开始时间
nEndTime	int	结束时间

### 4.3.2 Tick 数据结构

字段	类型	描述
<b>TDBDefine_Tick</b>		
chWindCode	char[32]	万得代码, 如: 000001.SZ
chCode	char[32]	交易所代码, 如: 000001
nDate	int	日期(自然日 YYYYMMDD)
nTime	int	时间(HHMMSSmmm)
nPrice	int	成交价
iVolume	long long	成交量
iTurover	long long	成交额
nMatchItems	int	成交笔数
nInterest	int	IOPV(基金)、利息(债券)
chTradeFlag	char	成交标志
chBSFlag	char	BS 标志
iAccVolume	long long	当日累计成交量
iAccTurover	long long	当日成交额(元)
nHigh	int	最高价
nLow	int	最低价
nOpen	int	开盘价
nPreClose	int	前收盘
下面字段期货使用		
nSettle	int	结算价
nPosition	int	持仓量
nCurDelta	int	虚实度
nPreSettle	int	昨结算

字段	类型	描述
nPrePosition	int	昨持仓
下面字段包含为买卖盘字段		
nAskPrice	int[10]	十档叫卖价
nAskVolume	unsigned int[10]	十档叫卖量
nBidPrice	int[10]	十档叫买价
nBidVolume	unsigned int[10]	十档叫买量
nAskAvPrice	int	加权平均叫卖价(上海 L2)
nBidAvPrice	int	加权平均叫买价(上海 L2)
iTotalAskVolume	long long	叫卖总量(上海 L2)
iTotalBidVolume	long long	叫买总量(上海 L2)
下面字段指数使用		
nIndex	int	不加权指数
nStocks	int	品种总数
nUps	int	上涨品种数
nDowns	int	下跌品种数
nHoldLines	int	持平品种数
下面字段无意义，为保留字段		
nResv1	int	保留字段 1
nResv2	int	保留字段 2
nResv3	int	保留字段 3

注 1：客户端可以根据自己的需求从该结构体中获取所需字段

注 2：Tick 数据，一次只能提取一天的数据

## 4.4 逐笔成交(Transaction)

### 4.4.1 请求逐笔成交数据结构

字段	类型	描述
<b>TDBDefine_ReqTransaction</b>		
chCode	char[32]	万得代码，如：000001.SZ
chMarketkey	char[24]	市场，如：SZ-2-0(上海-L2-Source0)
nDate	int	请求日期(<0:取上市日期,0:取今天数据)
nBeginTime	int	开始时间
nEndTime	int	结束时间

### 4.4.2 逐笔成交数据结构

字段	类型	描述
<b>TDBDefine_Transaction</b>		
chWindCode	char[32]	万得代码，如：000001.SZ
chCode	char[32]	交易所代码，如：000001

nDate	int	日期(自然日 YYYYMMDD)
nTime	int	时间(HHMMSSmmm)
nIndex	int	成交编号
chFunctionCode	char	成交代码
chOrderKind	char	委托代码
chBSFlag	char	BS 标志
nTradePrice	int	成交价格
nTradeVolume	int	成交数量
nAskOrder	int	叫卖序号
nBidOrder	int	叫买序号

其中 chOrderKind 域和 chFunctionCode 域的 2 个字符组合可能的取值即对应《深圳证券交易所数据接口规范》中的回报库 SJSHB.dbf 一节 HBYWLB（业务类别）字段取值范围，请自行参考。

8930 域 OrderKind	8931 域 FunctionCode	业务含义
0	0	交易业务成交记录
0	C	交易业务撤单回报记录
2	C	“即时成交剩余撤销委托”未能成交部分或其他原因的自动撤单回报记录
K	0	ETF 基金申购/赎回成功回报记录或 ETF 基金赎回/赎回成功证券给付明细回报记录
K	C	ETF 基金申购/赎回撤单回报记录
V	C	“最优五档即时成交剩余撤销委托”未能成交部分的自动撤单或其他原因的自动撤单回报记录
W	C	“全额成交或撤销委托”未能成交时的自动撤单或其他原因自动撤单回报记录
X	C	本方最优价格委托的撤单回报记录
Y	C	对手方最优价格委托的撤单回报记录
Z	0	ETF 基金申购/赎回成功允许/必须现金替代明细回报记录

## 4.5 逐笔委托(Order)

### 4.5.1 请求逐笔委托数据结构

字段	类型	描述
<b>TDBDefine_ReqOrder</b>		
chCode	char[32]	万得代码，如：000001.SZ
chMarketkey	char[24]	市场，如：SZ-2-0(上海-L2-Source0)
nDate	int	请求日期(0:取今天数据)
nBeginTime	int	开始时间

nEndTime	int	结束时间
----------	-----	------

#### 4.5.2 逐笔委托数据结构

字段	类型	描述
<b>TDBDefine_Order</b>		
chWindCode	char[32]	万得代码, 如: 000001.SZ
chCode	char[32]	交易所代码, 如: 000001
nDate	int	日期(自然日 YYYYMMDD)
nTime	int	时间(HHMMSSmmm)
nIndex	int	委托编号(从 1 开始递增)
nOrder	int	交易所委托号
chOrderKind	char	委托类型
chFunctionCode	char	委托代码, B、S、C
nOrderPrice	int	委托价格
nOrderVolume	int	委托数量

其中 chOrderKind 域和 chFunctionCode 域的 2 个字符组合可能的取值来源于《深圳证券交易所数据接口规范》中的委托库 SJSWT.dbf 一节 WTYWLB (业务类别) 字段取值范围。下表为参考, 以《深圳证券交易所数据接口规范》为准。

8930 域 OrderKind	8931 域 FunctionCode	业务含义
0	B	正常交易业务的买委托记录
0	S	正常交易业务的卖委托记录
0	C	正常交易业务的撤单记录
2	B	即时成交剩余撤销买委托记录
2	S	即时成交剩余撤销卖委托记录
K	B	ETF 基金申购申报记录
K	S	ETF 基金赎回申报记录
V	B	最优五档即时成交剩余撤销买委托记录
V	S	最优五档即时成交剩余撤销卖委托记录
W	B	全额成交或撤销买委托记录
W	S	全额成交或撤销卖委托记录
X	B	本方最优价格买委托记录
X	S	本方最优价格卖委托记录
X	C	本方最优价格撤单委托记录
Y	B	对手方最优价格买委托记录
Y	S	对手方最优价格卖委托记录
Y	C	对手方最优价格撤单委托记录

chOrderKind 域和 chFunctionCode 域适用于深交所 V5 系统

逐笔委托			
FunctionCode (交易所)	FunctionCode (TD)	OrderKind (交易所)	OrderKind (TD)
1 (买)	B	1 (市价)	1
2 (卖)	S	2 (限价)	0
G (借入)	与交易所保持一致	U (本方最优)	U
F (借出)	与交易所保持一致	其他	与交易所保持一致
逐笔成交			
4 (撤销)	C	0	0
F (成交)	0	0	0

## 4.6 委托队列(OrderQueue)

### 4.6.1 请求委托队列数据结构

字段	类型	描述
<b>TDBDefine_ReqOrderQueue</b>		
chCode	char[32]	万得代码, 如: 000001.SZ
chMarketkey	char[24]	市场, 如: SZ-2-0(上海-L2-Source0)
nDate	int	请求日期(<0:取上市日期,0:取今天数据)
nBeginTime	int	开始时间
nEndTime	int	结束时间

### 4.6.2 委托队列数据结构

字段	类型	描述
<b>TDBDefine_OrderQueue</b>		
chWindCode	char[32]	万得代码, 如: 000001.SZ
chCode	char[32]	交易所代码, 如: 000001
nDate	int	日期(自然日 YYYYMMDD)
nTime	int	时间(HHMMSSmmm)
nSide	int	买卖方向('B':Bid,'A':Ask)
nPrice	int	成交价格
nOrderItems	int	订单数量
nABItems	int	明细个数
nABVolume	int[50]	订单明细



## 4.7 请求返回值

宏定义	值	描述
<b>TDB_ERROR</b>		
TDB_SUCCESS	0	成功
TDB_NETWORK_ERROR	-1	网络错误
TDB_NETWORK_TIMEOUT	-2	网络超时
TDB_NO_DATA	-3	没有数据
TDB_OUT_OF_MEMORY	-4	内存耗尽
TDB_LOGIN_FAILED	-5	登陆失败
TDB_INVALID_PARAMS	-11	无效的参数
TDB_INVALID_CODE_TYPE	10	无效的代码类型, 比如向 TDB_GetFuture 传入非期货类型代码, 返回之
TDB_WRONG_FORMULA	-50	指标公式错误

## 5 调用 API 方法

### 5.1 登录

```

THANDLE TDB_Open(
    const OPEN_SETTING* pSetting,    登录设置
    TDBDefine_ResLogin* loginRes    登录结果
);

```

在获取数据之前, API 必须与 TDB 服务器建立连接; 首先是配置好登录设置, 调用 TDB\_Open 函数请求与服务器建立连接并认证账号; 该函数为同步模式, 并返回登录结果和连接句柄。

字段	类型	描述
<b>OPEN_SETTING</b>		
szIP	char[24]	服务器 IP 地址
szPort	char[8]	服务器端口
szUser	char[32]	用户名
szPassword	char[32]	密码
nTimeOutVal	unsigned int	获取数据时的网络超时; 若超时, 内部会关闭连接
nRetryCount	unsigned int	保留
nRetryGrap	unsigned int	保留

字段	类型	描述
<b>TDBDefine_ResLogin</b>		
szInfo	char[64]	信息
nMarkets	int	支持市场个数
szMarket	char[256][24]	市场标志(SZ-2-0,SH-1-0...)

nDynDate	int[256]	动态数据日期
----------	----------	--------

## 5.2 代理登录

```

THANDLE TDB_OpenProxy(
    const OPEN_SETTING* pSetting,      登录设置
    TDB_PROXY_SETTING* pProxySetting  代理设置
    TDBDefine_ResLogin* loginRes      登录结果
);

```

在获取数据之前，API 必须与 TDB 服务器建立连接；首先是配置好登录设置，调用 TDB\_Open 函数请求与服务器建立连接并认证账号；该函数为同步模式，并返回登录结果和连接句柄。

字段	类型	描述
<b>OPEN_SETTING</b>		
szIP	char[24]	服务器 IP 地址
szPort	char[8]	服务器端口
szUser	char[32]	用户名
szPassword	char[32]	密码
nTimeOutVal	unsigned int	获取数据时的网络超时；若超时，内部会关闭连接
nRetryCount	unsigned int	保留
nRetryGrap	unsigned int	保留

字段	类型	描述
<b>TDB_PROXY_SETTING</b>		
nProxyType	TDB_PROXY_TYPE	代理类型
szPeoxyHostIp	char[24]	代理服务器 IP 地址
szPeoxyPort	char[8]	代理服务器端口
szPeoxyUser	char[32]	代理用户名
szPeoxyPwd	char[32]	代理密码

宏定义	类型	描述
<b>TDB_PROXY_TYPE</b>		
TDB_PROXY SOCK4	0	SOCK 4
TDB_PROXY SOCK4A	1	SOCK 4A
TDB_PROXY SOCK5	2	SOCK 5
TDB_PROXY HTTP11	3	HTTP11

字段	类型	描述
<b>TDBDefine_ResLogin</b>		
szInfo	char[64]	信息
nMarkets	int	支持市场个数

szMarket	char[256][24]	市场标志(SZ-2-0,SH-1-0...)
nDynDate	int[256]	动态数据日期

### 5.3 获取 K 线数据

```
int TDB_GetKLine(
    THANDLE hTdb,                [in]句柄
    const TDBDefine_ReqKLine* pReq, [in]K 线请求包
    TDBDefine_KLine** pData,      [out]K 线数据包
    int* nCount                   [out]K 线数据包个数
);
```

配置 K 线数据请求包，调用 TDB\_GetKLine 获取 K 线数据，**K 线请求包结构**，参照[请求 K 线数据结构](#)

### 5.4 获取 Tick 数据

#### 5.4.1 获取 Tick 数据

```
int TDB_GetTick(
    THANDLE hTdb,                [in]TDB 句柄
    const TDBDefine_ReqTick* pReq, [in]Tick 请求包
    TDBDefine_Tick** pData,       [out]Tick 数据包
    int* pCount                  [out]Tick 数据包个数
);
```

配置 Tick 数据请求包，调用该方法获取 Tick 数据，**Tick 请求包结构**，参照[请求 Tick 数据结构](#)

### 5.5 获取逐笔成交数据

```
int TDB_GetTransaction(
    THANDLE hTdb,                [in]句柄
    const TDBDefine_ReqTransaction* pReq, [in]请求包
    TDBDefine_Transaction** pData,       [out]返回数据包
    int* pCount                  [out]数据包个数
);
```

配置逐笔成交数据请求包，调用该方法获取逐笔成交数据。**请求包结构**参照[请求逐笔成交数据结构](#)

### 5.6 获取逐笔委托数据

```
int TDB_GetOrder(
    THANDLE hTdb,                [in]句柄
    const TDBDefine_ReqOrder* pReq, [in]请求包
    TDBDefine_Order** pData,       [out]返回数据包
    int* pCount                  [out]数据包个数
);
```

配置逐笔委托数据请求包，调用该方法获取逐笔委托数据。**请求包结构**参照[请求逐笔](#)

## 5.7 获取委托队列数据

```
int TDB_GetOrderQueue(
    THANDLE hTdb,                [in]句柄
    const TDBDefine_ReqOrderQueue* pReq, [in]请求包
    TDBDefine_OrderQueue** pData,    [out]返回数据包
    int* pCount                   [out]数据包个数
);
```

配置委托队列数据请求包，调用该方法获取委托队列数据。[请求包结构参照请求委托队列数据结构](#)

## 5.8 计算指标公式

### 5.8.1 上载指标公式

```
int TDB_AddFormula(
    THANDLE hTdb,                [in]TDB 句柄
    const char* szName,          [in]指标名称
    const char* szContent,       [in]指标公式内容
    TDBDefine_AddFormulaRes* pRes [out]上载结果
);
```

调用 TDB\_AddFormula 方法向服务器上传指标公式。

字段	类型	描述
<b>TDBDefine_AddFormulaRes</b>		
nErrline	int	错误行
chText	char[132]	错误行指标公式正文
chInfo	char[68]	错误信息

### 5.8.2 获取指标公式

```
int TDB_GetFormula(
    THANDLE hTdb,                [in]TDB 句柄
    const char* szFormulaName,    [in]指标名称
    TDBDefine_FormulaItem** pFormula, [out]返回指标数据包
    int *pCount                  [out]指标数据包个数
);
```

调用 TDB\_GetFormula 方法获取服务器上存在的指标公式，szFormulaName 为空表示请求服务器上所有的指标公式，反之，则请求对应的指标公式。

字段	类型	描述
----	----	----

TDBDefine_FormulaItem		
chFormulaName	char[28]	指标公式名称
chParam	char[132]	参数

### 5.8.3 删除指标公式

```
int TDB_DeleteFormula(
    THANDLE hTdb,                [in]TDB 句柄
    const char* szFormulaName,    [in]指标名称
    TDBDefine_DelFormulaRes* pDelRes [out]删除结果
);
```

调用 TDB\_DeleteFormula 方法删除服务器上存在的指标公式。

字段	类型	描述
TDBDefine_DeleteFormula		
chFormulaName	char[28]	删除指标公式名称
chInfo	char[68]	删除信息

### 5.8.4 计算指标公式

```
int TDB_CalcFormula(
    THANDLE hTdb,                [in]TDB 句柄
    const TDBDefine_ReqCalcFormula* pReqCalc, [in]请求计算
    TDBDefine_CalcFormulaRes* pResult        [out]计算结果
);
```

调用 TDB\_CalcFormula 方法利用指定指标公式计算某个品种的数据。

字段	类型	描述
TDBDefine_ReqCalcFormula		
chFormulaName	char[28]	指标公式名称
chParam	char[32]	参数
chCode	char[32]	万得代码
chMarketkey	char[24]	市场，如：SZ-2-0(上海-L2-Source0)
nCycType	CYCTYPE	数据周期，类型参考 <a href="#">请求 K 线数据结构</a>
nCycDef	int	数据周期定义
nCQFlag	REFILLFLAG	除权标志，类型参考 <a href="#">请求 K 线数据结构</a>
nBondDataType	int	债券全价标志
nCQDate	int	除权日期
nCalcMaxItems	int	计算的最大数据量

nResultMaxItems	int	传送的结果最大数据量
-----------------	-----	------------

字段	类型	描述
<b>TDBDefine_CalcFormulaRes</b>		
chWindCode	char[32]	万得代码
chCode	char[32]	交易所代码
nRecordCount	int	记录数量
nFieldCount	int	字段数据, 最多 50 个
chFileName	char[50][28]	字段名称
dataFiled	*int[50]	最多 50 个字段

```
void TDB_ReleaseCalcFormula(
    TDBDefine_CalcFormulaRes* pResult    [in]删除计算指标结果
);
```

在每次计算完指标数据之后, 需要对内存进行释放, TDB\_ReleaseCalcFormula 释放内存。

### 5.8.5 服务器指标编写实例

指标名称	代码
KDJ	INPUT:N(9), M1(3,1,100,2), M2(3); //KDJ参数定义 RSV:=(CLOSE-LLV(LOW,N))/(HHV(HIGH,N)-LLV(LOW,N))*100; K:SMA(RSV,M1,1); D:SMA(K,M2,1); J:3*K-2*D;
MACD	INPUT:SHORT(12,2,200,2), LONG(26,2,200,2), MID(9,2,200,2); DIF:EMA(CLOSE,SHORT)-EMA(CLOSE,LONG); DEA:EMA(DIF,MID); MACD:(DIF-DEA)*2,COLORSTICK;
MA	INPUT:M1(5,2,999), M2(10,2,999), M2(30,2,999), M2(60,2,999); MA1:MA(CLOSE,M1); MA2:MA(CLOSE,M2); MA3:MA(CLOSE,M3); MA4:MA(CLOSE,M4);

## 5.9 代码表

### 5.9.1 获取代码表

```
int TDB_GetCodeTable(
    THANDLE hTdb,                [in]句柄
    const char* szMarket,        [in]请求市场, 格式为 (Market-Level-Source)
```

```
TDBDefine_Code** pData,           [out]返回数据包
int* pCount                        [out]数据包个数
);
调用 TDB_GetCodeTable 函数获取对应市场的代码表，数据包参考证券代码表
```

### 5.9.2 获取代码信息

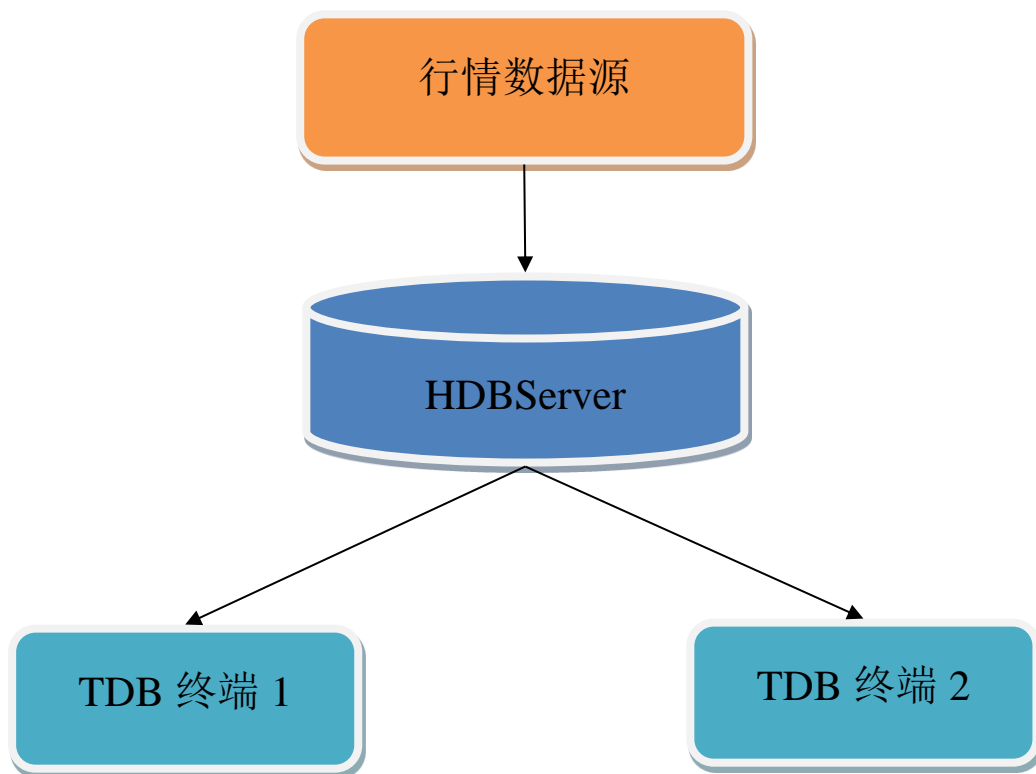
```
const TDBDefine_Code * TDB_GetCodeInfo(
    THANDLE hTdb,                 [in]句柄
    const char* szWindCode,        [in]请求代码的万得代码
    const char* szMarket,          [in]请求市场，格式为（Market-Level-Source）
);
调用 TDB_GetCodeInfo 函数获取对应代码信息，返回数据结构参考证券代码表
```

### 5.10 关闭

```
int TDB_Close(
    THANDLE hTdb    [in]TDB 连接句柄
);
调用 TDB_Close 函数主动与 TDB 服务器断开连接。
```

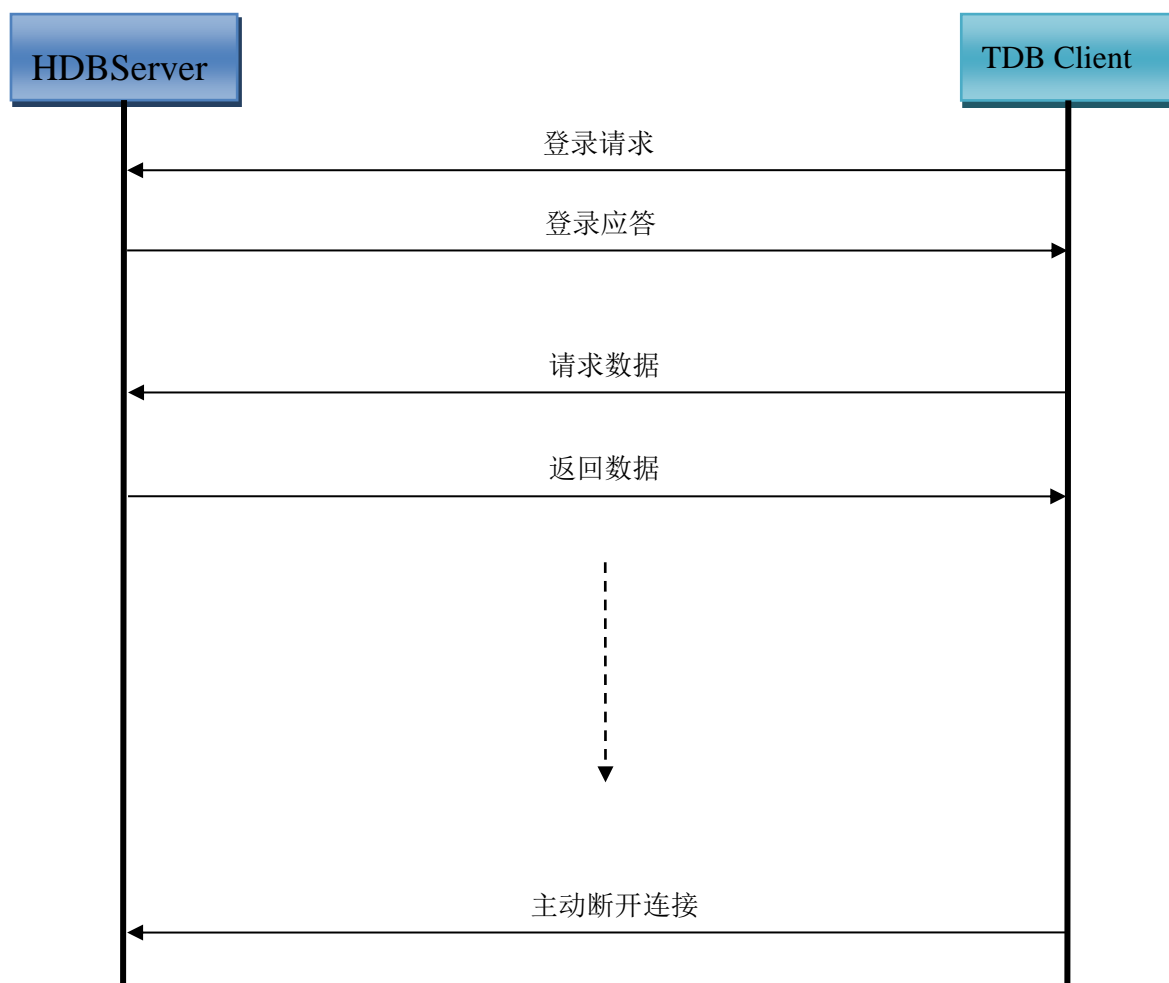
## 6 TDB 系统结构

宏汇 TDB 系统示意图：



系统流程图:





## 7 TDB 常见问题

见文档《TDB 原生 API 使用常见问题》

此文档含有具体的技术和业务相关的专有信息，文中涉及的所有资料必须严格保密。除因需要而得到授权的客户和本公司有关人员外，一律不得向他人泄露。