



TDF JAVA Remote API Quick Start

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

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

文档管理信息表

主题	TDF C++ Remote API Quick Start
版本	V1.2
内容	简述 TDF JAVA Remote API 接口
关键字	TDF, JAVA, Remote
参考文档	-
创建时间	2016 年 03 月 23 日
创建人	唐龙涛
最新发布日期	2017年08月08日

文档变更记录表

修改人	修改时间	修改内容
唐龙涛	2016 年 03 月 23 日	创建, v1.1
唐龙涛	2016 年 08 月 04 日	新增证券类型细分定义
唐龙涛	2017 年 02 月 09 日	代码表及行情结构调整, 增加 ETF 获取, V1.2
唐龙涛	2017 年 04 月 06 日	新增单个市场代码表通知, 修改获取期权接口
唐龙涛	2017 年 08 月 08 日	BBQ 增加 DoneID 字段, windcode 说明

Wind Confidential

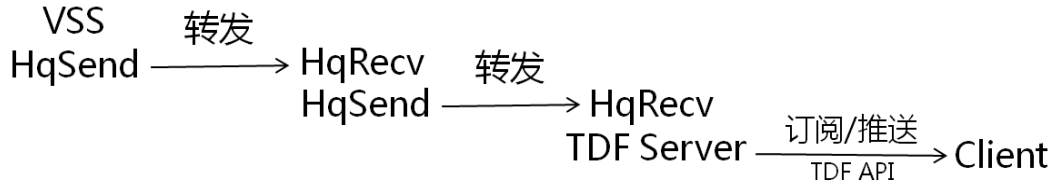
目 录

1	简介.....	1
1.1	TD 系统简介.....	1
1.2	TDF LIFE-CYCLE	1
1.3	TDF 特性	2
1.4	REMOTE API 相对 TDFAPI2.5 变化	2
1.4.1	Remote API 优点.....	2
1.4.2	Remote API 接口变化.....	2
2	JAVA 文件包.....	3
3	JAVA 使用过程.....	3
4	TDFCLIENT 类(API)	3
4.1	SETEnv	3
4.2	OPEN	4
4.3	OPEN_EXT	4
4.4	OPENPROXY	4
4.5	CLOSE	4
4.6	GETCODETABLE	4
4.7	SETSUBSCRIPTION	5
4.8	GETOPTIONCODEINFO	5
5	API 通知消息详解.....	5
5.1	系统消息回调	5
5.2	行情数据回调	6
5.3	消息结构体(TDF_MSG)	6
5.4	期权处理注意事项	7
6	数据结构详解.....	7
6.1	OPEN/OPEN_EXT/OPENPROXY 相关结构	7
6.1.1	TDF_OPEN_SETTING.....	7
6.1.2	TDF_OPEN_SETTING_EXT.....	7
6.1.3	TDF_SERVER_INFO.....	8
6.1.4	TDF_PROXY_TYPE.....	8
6.1.5	TDF_ERR.....	8
6.1.6	TDF_PROXY_SETTING.....	9
6.2	系统消息结构体	9
6.2.1	TDF_CONNECT_RESULT	9
6.2.2	TDF_LOGIN_RESULT	9
6.2.3	TDF_CODE_RESULT.....	9
6.2.4	TDF_QUOTATIONDATE_CHANGE.....	10
6.3	数据结构体	10
6.3.1	证券代码表 - 通用 (TDF_CODE)	10

6.3.2	证券代码表 - 期权 (TDF_OPTION_CODE)	10
6.3.3	行情快照 - 股票(TDF_MARKET_DATA).....	10
6.3.4	行情快照 - 期货/期权(TDF_FUTURE_DATA).....	11
6.3.5	行情快照 - 指数(TDF_INDEX_DATA)	12
6.3.6	逐笔成交(TDF_TRANSACTION).....	12
6.3.7	逐笔委托(TDF_ORDER).....	13
6.3.8	委托队列(TDF_ORDER_QUEUE)	13
6.3.9	BBQ 现券成交数据(TDF_BBQTRANSACTION).....	13
6.3.10	BBQ 现券报价数据(TDF_BBQBID_DATA)	14
6.3.11	港股经纪商队列(TDF_BROKER_QUEUE)	14
6.3.12	代码信息 (TdfCodeInfo)	15
6.3.12.1	基本代码信息.....	15
6.3.12.2	扩展代码信息 - 期权(TdOptionCodeInfo)	15
6.3.13	交易所编号.....	15
7	示例.....	16
8	附录.....	16
8.1	状态含义	16
8.2	上证指数代码转换对应表	17
8.3	委托类别和委托代码含义	17
8.4	成交类别和成交代码含义	18
8.5	行情数据 (MARKET DATA) 状态与交易所状态转换表	19
8.6	证券类型基础定义	20
8.7	证券类型细分定义	20
9	FAQ (常见问题)	21

1 简介

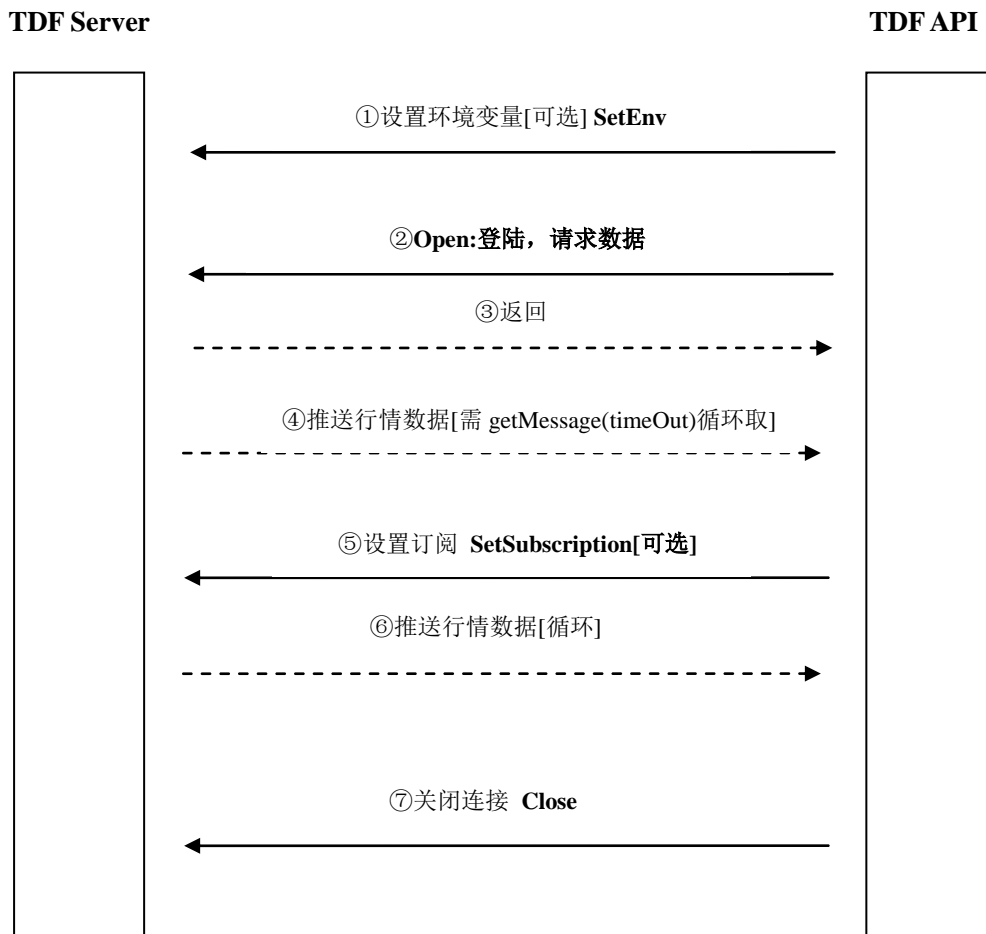
1.1 TD 系统简介



TD 流式行情系统，包含转码机程序 VSS、行情转发程序 HqSend、行情接收程序 HqRecv 和实时行情服务程序 TDF Server。

转码机程序 VSS 接入交易所，将数据由交易所协议转化为 TD 内部协议（高度压缩），通过行情转发程序 HqSend 发送至下游服务器。下游服务器开启 HqRecv 接收，与上游的 HqSend 配对。可使用 HqRecv-HqSend 完成多次级联转发。TDF Server 是实时行情服务程序，在接收行情时根据用户请求类型，发送对应数据到客户端，客户端使用 TDF Remote API 接受数据并解压。本文简述 TDF JAVA Remote API 接口。

1.2 TDF Life-Cycle



1.3 TDF 特性

- (1) 使用 TCP 协议;
- (2) 支持 Windows 操作系统, 32 位和 64 位平台
- (3) 支持 Linux 操作系统, 64 位平台
- (4) 支持的接口语言: C/C++, C#, Java;
- (6) 不对交易所行情数据内容作修改, 仅对数据格式作转换;
- (6) 目前 TDF 接口支持以下数据内容

交易所	数据类型
上交所	五档行情快照 (Level-1)
	十档行情快照 (Level-2)
	逐笔成交 (Level-2)
	委托队列 (Level-2)
深交所	五档行情快照 (Level-1)
	十档行情快照 (Level-2)
	逐笔成交 (Level-2)
	逐笔委托 (Level-2)
	委托队列 (Level-2)
中金所	一档行情快照 (Level-1)
	五档行情快照 (Level-2)
上期所	一档行情快照 (Level-1)
郑商所	一档行情快照 (Level-1)
大商所	一档行情快照 (Level-1)
港交所 (HK)	十档行情快照 (Level-2)
	经纪商队列 (Level-2)
新三板市场 (PT)	五档行情快照 (Level-1)
	逐笔委托 (Level-1)

(6) 采用订阅-回调的模式推送数据。回调分为两类, 一类为系统消息回调, 一类为行情数据回调。

1.4 Remote API 相对 TDF API 2.5 变化

1.4.1 Remote API 优点

- 1) 流量减少到原来的 1/4 - 1/6
- 2) 因为流量减少, 会减少网络对延迟的影响
- 3) 解压放到客户端, 减少服务器压力, 服务器更稳定, 而且不会随客户数量增加性能下降
- 4) 双活双源: 1) 增加稳定性, 2) 更快, 取 2 路中最快的各路行情 3) 双路热切换, 没有切换时间

1.4.2 Remote API 接口变化

- 1) 每个数据消息中都加入一个新字段: 指向代码表的指针:
`TdfCodeInfo codeInfo; //代码信息`
- 2) 数据类型订阅行为改变: 默认只发送行情数据, 并且只支持 3 种数据类型订阅,

如下:

```
DATA_TYPE_FLAG {
    public final static int DATA_TYPE_NONE = 0; //只要行情，其他都不订阅
    public final static int DATA_TYPE_TRANSACTION = 0x2;
    public final static int DATA_TYPE_ORDER = 0x4;
    public final static int DATA_TYPE_ORDERQUEUE = 0x8;
    //订阅多个类型可用DATA_TYPE_TRANSACTION|DATA_TYPE_ORDER|DATA_TYPE_ORDERQUEUE
};
```

- 3) 不支持取历史数据，不能回放当天数据
- 4) TDF_OPEN_SETTING 结构体去除不必要的字段：nReconnectCount；nReconnectGap；nDate；nProtocol
- 5) 支持双活 API，调用函数为 open_ext
- 6) 市场订阅格式改为市场简称-数据 level-Source；比如 SH-2-0;SH-1-1;
- 7) 市场和代码必须订阅其中一个，两个都不订阅，open 会失败

2 JAVA 文件包

包括 TDFAPI30.dll(C++)、tdfapi_jni.dll(JNI)、tdfapi.jar(API)和 demo.jar(Demo)，另外包括使用的几个 jar 包(opencsv-2.3.jar,protobuf-java-2.5.0.jar)，使用时需导入 jar 包并将对应的 dll 放到同一路径下。

3 JAVA 使用过程

- 1、创建 TDFClient 类：所有的交互都基于这个类；
- 2、setEnv：设置环境，在 open 前调用，可选；
- 3、open(TDF_OPEN_SETTING setting)：连接 TDFServer 服务器，需配置参数 setting；
- 4、循环调用 getMessage(int timeout)：收取的数据放在队列中，调用此方法读取数据，有数据立即返回，没数据，等待 timeout 毫秒后返回。调用该方法的线程仅做存储动作，以免造成数据积压和丢失；
- 5、close:使用完成需要关闭连接；

4 TDFClient 类(API)

4.1 SetEnv

说明	设置TDF环境变量值，在调用Open之前设置	
参数	int nEnv	需要被设置的环境变量，TDF_ENVIRON_SETTING值
	int value	为环境变量设置的值
返回值	Int	TDF_ERR类型对应值

```
class TDF_ENVIRON_SETTING
```

```
{
```

```
TDF_ENVIRON_HEART_BEAT_INTERVAL = 0, //Heart Beat间隔（秒数），若值为0则表示默认值10秒钟
TDF_ENVIRON_MISSED_BEART_COUNT = 1, //如果没有收到心跳次数超过这个值，且没收到其他任何数据，则判断为掉线，若值0为默认次数2次
```

```
TDF_ENVIRON_OPEN_TIME_OUT = 2, //在调TDF_Open期间，接收每一个数据包的超时时间（秒数，不是TDF_Open调用总的最大等待时间），若值为0则默认30秒
```

TDF_ENVIRON_USE_PACK_OVER = 3//一个完整网络包解压完后发一次该消息(未实现)

TDF_ENVIRON_HEART_BEAT_FLAG = 4//心跳数据发送方式(0: 取消发送心跳包 1: 没有数据发送的时候发送心跳数据, 2: 有规律发送心跳数据)

TDF_ENVIRON_SOURCE_MODE= 3//双活数据源模式, 值参考TDF_SOURCE_SETTING类

TDF_ENVIRON_SOURCE_MODE_VALUE= 3//双活数据源模式下参数的值

TDF_ENVIRON_OUT_LOG= 3//1、当前目录下创建log, 否则在当前路径有log文件夹时创建到log文件夹中

};

4.2 open

说明	打开到TDFServer的连接, 如果成功, 则返回句柄, 否则返回NULL。 在open期间发生了网络断开, 将不会自动重连, 外部需要自己做重连逻辑 在调用期间, 如果没有错误, 系统通知函数将依次收到以下消息: MSG_SYS_CONNECT_RESULT MSG_SYS_LOGIN_RESULT MSG_SYS_CODETABLE_RESULT 如果上述三条系统通知消息都收到后, 可以认为Open成功。此后, 如果网络断开, 则会收到MSG_SYS_DISCONNECT_NETWORK, pErr中存放错误代码, 并且API会自动重连 参数说明, 请见 TDF_OPEN_SETTING	
参数	TDF_OPEN_SETTING setting	连接设置
返回值	int	TDF_ERR 类型对应值

4.3 open_ext

说明	open的扩展版本, 行为特征一致, 但: , 可以同时连接多个服务器 对于行情快照数据, 多路来的行情只要比上一笔新, 就提交, 否则丢弃 对于其他类型数据, 根据编号取最新数据, 如果断掉, 自动切换到另一路	
参数	TDF_OPEN_SETTING_EXT setting	连接设置
返回值	int	TDF_ERR 类型

4.4 openProxy

说明	通过代理打开到TDFServer的连接, 如果成功, 则返回句柄, 否则返回NULL。 在open期间发生了网络断开, 将不会自动重连。 在调用期间, 如果没有错误, 系统通知函数将收到以下消息: MSG_SYS_CONNECT_RESULT MSG_SYS_LOGIN_RESULT MSG_SYS_CODETABLE_RESULT 如果前面的消息都是成功的话。如果网络断开, 则会收到MSG_SYS_DISCONNECT_NETWORK, 并且API会自动重连, pErr中存放错误代码,	
参数	TDF_OPEN_SETTING settings	连接设置
	TDF_PROXY_SETTING proxySetting	代理连接设置
返回值	int	TDF_ERR 类型

4.5 close

说明	同步函数, 关闭连接	
参数	-	-
返回值	int	TDF_ERR 类型

4.6 getCodeTable

说明	获取指定市场的代码表, 如果为空, 则获取所有的代码表, 在已经收到
----	------------------------------------

	MSG_SYS_CODETABLE_RESULT 消息之后, 可以获得代码表。	
参数	-	-
	String Market	市场代号, 一次取一个市场, 例如"SH-2-0"
返回值	TDF_CODE []	代码表结果

4.7 setSubscription

说明	异步函数, 完成连接后设置订阅。退出该函数之后, 回调中收到的数据将做相应的改变。该订阅与open时设置的市场一起对数据过滤, 两者取并集。	
参数	-	-
	String szSubscriptions	订阅列表。例如:"600000.sh;ag.shf;000001.sz", 需要订阅的股票(单个股票格式为原始Code+.+市场, 如999999.SH), 以";"分割, 为空则订阅全市场
	int nSubStyle	订阅模式: 全市场、设置、增加和删除 class SUBSCRIPTION_STYLE { SUBSCRIPTION_FULL = 3, //全市场订阅 SUBSCRIPTION_SET=0, //设置为订阅列表中股票, SUBSCRIPTION_ADD=1, //增加订阅列表中股票 SUBSCRIPTION_DEL=2, //删除订阅列表中的订阅 }
返回值	int	TDF_ERR 类型

4.8 getOptionCodeInfo

说明	根据代码获取详细的期权代码信息; 对于非期权代码, 也可以获得基本信息, 但期权相关数据为空; pCodeInfo由调用者提供。	
参数	-	-
	String Code	订阅代码, 只接受单个Code, 格式为原始code + . + 市场(如ag.SHF)
	String Market	市场代码(格式: SH-1-1)
返回值	TDF_OPTION_CODE	返回的数据结构

5 API 通知消息详解

获取数据方法定义:

public [TDF MSG](#) getMessage(int timeout) 需要循环调用该方法。参数含义: 等待timeout毫秒后还没有可取数据就返回, 如果有数据, 立刻返回。

5.1 系统消息回调

消息类型:

消息	说明
MSG_SYS_DISCONNECT_NETWORK	网络断开事件
MSG_SYS_CONNECT_RESULT	主动发起连接的结果
MSG_SYS_LOGIN_RESULT	登陆应答
MSG_SYS_CODETABLE_RESULT	获取代码表结果
MSG_SYS_QUOTATIONDATE_CHANGE	行情日期变更通知

MSG_SYS_HEART_BEAT	服务器心跳消息
MSG_SYS_SINGLE_CODETABLE_RESULT	单个市场代码表结果，收到此消息后可获取对应的单个市场代码表

5.2 行情数据回调

消息类型:

消息	说明
MSG_DATA_MARKET	行情快照数据
MSG_DATA_INDEX	指数行情快照数据
MSG_DATA_FUTURE	期货和期权行情快照数据
MSG_DATA_TRANSACTION	逐笔成交
MSG_DATA_ORDERQUEUE	委托队列
MSG_DATA_ORDER	逐笔委托
MSG_DATA_BBQTRANSACTION	BBQ现券成交
MSG_DATA_BBQBID	BBQ现券报价
MSG_DATA_BROKERQUEUE	经纪商队列(港股)

5.3 消息结构体(TDF_MSG)

```
class TDF_MSG
{
    public int  getDataType(); //数据类型(参见5.1, 5.2)
    public int  getServerTime(); //32位TDF服务器时间戳(精确到毫秒HHMMSSmmm), 对于系统消息为0, 该值为JAVA收到数据时的时间
    public TDF_APP_HEAD  getAppHead(); //应用头
};

class TDF_APP_HEAD
{
    public int  getItemCount(); //记录条数
};
```

获取到对应消息后, 使用 getMessageData(TDF_MSG, int), 获取 TDF_MSG_DATA 类型的数据, 并获取对应的类。

```
class TDF_MSG_DATA
{
    //系统消息
    public TDF_LOGIN_RESULT  getLoginResult(); //登录结果
    public TDF_CONNECT_RESULT  getConnectResult(); //连接结果
    public TDF_CODE_RESULT  getCodeTableResult(); //代码表结果
    public TDF_QUOTATIONDATE_CHANGE  getDateChange(); //行情日期变更(已废)
    public TDF_MARKET_CLOSE  getMarketClose(); //闭市消息(已废)
    //数据消息
    public TDF_MARKET_DATA  getMarketData(); //行情
    public TDF_INDEX_DATA  getIndexData(); //指数
    public TDF_FUTURE_DATA  getFutureData(); // 期权、期货
    public TDF_ORDER_QUEUE  getOrderQueue(); //委托队列
```

```

public TDF_TRANSACTION getTransaction();           //逐笔成交
public TDF_ORDER getOrder();                       //逐笔委托
public TDF_BBQTRANSACTION getBbqtransaction();    //BBQ 现券成交数据
public TDF_BBQBID_DATA getBbqbid();               //BBQ 现券报价数据
public TDF_BROKER_QUEUE getBrokerQueue ();        //港股经纪商队列
}

```

5.4 期权处理注意事项

期权数据结构和期货是一样的，数据类型为：MSG_DATA_FUTURE。数据结构为 TDF_FUTURE_DATA。区分期权和期货的方法是 TDF_FUTURE_DATA.getCodeInfo().getType() 类型信息。如果 (TDF_FUTURE_DATA.getCodeInfo().getType() & 0xf0) == 0x90 为期权数据。

6 数据结构详解

6.1 open/open_ext/openProxy 相关结构

6.1.1 TDF_OPEN_SETTING

方法名	备注
void setIP(String value)	连接TDFServer主机IP
void setPort(String value)	连接TDFServer主机的端口
void setUser(String value)	登陆用户名
void setPwd(String value)	登陆密码
void setMarkets(String value)	需要订阅的市场列表，以“;”分割，例如：“SZ-2-0;SH-1-1;SHF-1-0”
void setSubscriptions(String value)	需要订阅的股票，以“;”分割，例如：“600000.SH;ag.SHF;000001.SZ”，单个股票格式为原始Code+.市场，如999999.SH)，为空则订阅全市场
void setTypeFlags(long value)	取值为DATA_TYPE_FLAG中多种类别： class DATA_TYPE_FLAG { DATA_TYPE_NONE = 0, //只订阅行情快照 DATA_TYPE_TRANSACTION = 0x2, // 逐 笔 成 交 DATA_TYPE_ORDER = 0x4, // 逐 笔 委 托 DATA_TYPE_ORDERQUEUE=0x8, //委托队列 }; 任何情况下，都会传送行情数据，例如： DATA_TYPE_TRANSACTION 会推送行情数据+指数+逐笔成交，多种类别使用：DATA_TYPE_TRANSACTION DATA_TYPE_ORDER
void setTime(long value)	需为0表示请求实时行情
void setConnectionID(long value)	用户可以使用此字段设置连接ID，在连接成功后的系统消息MSG_SYS_CONNECT_RESULT中会返回此字段

6.1.2 TDF_OPEN_SETTING_EXT

方法名	备注
void setServerInfo(TDF_SERVER_INFO[] value)	连接 TDFServer 主机 IP，最多可连接 4 个
void setMarkets(String value)	需要订阅的市场列表，以“;”分割，例如：“SZ-2-0;SH-1-1;SHF-1-0”

<code>void setSubscriptions(String value)</code>	需要订阅的股票，以“;”分割，例如： “600000.SH;ag.SHF;000001.SZ”，单个股票格式为原始Code+.市场，如999999.SH)，为空则订阅全市场
<code>void setTypeFlags(long value)</code>	取值为DATA_TYPE_FLAG中多种类别： class DATA_TYPE_FLAG { DATA_TYPE_NONE = 0, //只订阅行情快照 DATA_TYPE_TRANSACTION = 0x2, //逐笔成交 DATA_TYPE_ORDER = 0x4, //逐笔委托 DATA_TYPE_ORDERQUEUE=0x8, //委托队列 }; 任何情况下，都会传送行情数据，例如： DATA_TYPE_TRANSACTION 会推送行情数据+指数+逐笔成交，多种类别使用： DATA_TYPE_TRANSACTION DATA_TYPE_ORDER
<code>void setTime(long value)</code>	需为0表示请求实时行情
<code>void setConnectionID(long value)</code>	用户可以使用此字段设置连接ID，在连接成功后的系统消息MSG_SYS_CONNECT_RESULT中会返回此字段

6.1.3 TDF_SERVER_INFO

方法名	备注
<code>void setIP(String value)</code>	连接TDFServer主机IP
<code>void setPort(String value)</code>	连接TDFServer主机的端口
<code>void setUser(String value)</code>	登陆用户名
<code>void setPwd(String value)</code>	登陆密码

6.1.4 TDF_PROXY_TYPE

代理连接协议类型

```
public final class TDF_PROXY_TYPE
{
    public final static int TDF_PROXY SOCK4    = 0;
    public final static int TDF_PROXY SOCK4A  = 1;
    public final static int TDF_PROXY SOCK5    = 2;
    public final static int TDF_PROXY HTTP11   = 3;
}
```

6.1.5 TDF_ERR

名	值	备注
TDF_ERR_UNKOWN	-200	未知错误
TDF_ERR_INITIALIZE_FAILURE	-100	初始化 socket 环境失败
TDF_ERR_NETWORK_ERROR	-99	网络连接出现问题
TDF_ERR_INVALID_PARAMS	-98	输入参数无效
TDF_ERR_VERIFY_FAILURE	-97	登陆验证失败：原因为用户名或者密码错误；超出登陆数量

TDF_ERR_NO_AUTHORIZED_MARKET	-96	所有请求的市场都没有授权
TDF_ERR_NO_CODE_TABLE	-95	所有请求的市场该天都没有代码表
TDF_ERR_SUCCESS	0	成功

6.1.6 TDF_PROXY_SETTING

方法名	备注
void setProxyType(int value)	代理类型, TDF_PROXY_TYPE
void setProxyHostIP(String value)	代理主机IP
void setProxyPort(String value)	代理主机的端口
void setProxyUser(String value)	代理登陆用户名
void setProxyPwd(String value)	代理登陆密码

6.2 系统消息结构体

系统消息对应的数据结构。

6.2.1 TDF_CONNECT_RESULT

系统消息: MSG_SYS_CONNECT_RESULT 对应的结构

方法名	备注
String getIP()	连接TDFServer主机IP
String getPort()	连接TDFServer主机的端口
String getUser()	登陆用户名
String getPwd()	登陆密码
int getConnResult()	为0则表示连接失败, 非0则表示连接成功
int getConnectionID()	连接ID

6.2.2 TDF_LOGIN_RESULT

系统消息: MSG_SYS_LOGIN_RESULT 对应的结构

方法名	备注
long getLoginResult()	为0则表示登陆验证失败, 非0则表示验证成功
String getInfo()	登陆结果文本
int getMarkets()	市场个数
String[] getMarket()	市场代码 SZ-2-0, SH-1-1, SHF-1-0
Int[] getDynDate()	动态数据日期

6.2.3 TDF_CODE_RESULT

系统消息: MSG_SYS_CODETABLE_RESULT 对应的结构

方法名	备注
String getInfo()	代码表结果文本
int getMarkets()	市场个数
String[] getMarket()	市场代码 SZ-2-0, SH-1-1, SHF-1-0
int[] getCodeCount	代码表项数
int[] getCodeDate()	代码表日期

当回调函数接收到该消息后, 客户端需要调用[getCodeTable](#)接口来获得代码表, 并将代码表信息保存下来做相关处理。

请注意, [getCodeTable](#)只能获得调用该函数为止的最新代码表, 在运行过程中, 当在回调函数中接收到新代码时(全市场订阅情况下), 客户端可以再次调用[getCodeTable](#)获得该新代码信息, 更新代码表。

6.2.4 TDF_QUOTATIONDATE_CHANGE

6.3 数据结构体

6.3.1 证券代码表 – 通用 (TDF_CODE)

代码元数据，即 [getCodeTable](#) 函数的返回数组元素。

方法名	备注
String getWindCode()	WindCode 代码，如：AG1302.SHF
String getMarket()	交易所名称
String getCode()	原始代码，如：ag1302
String getENName()	代码英文名称
String getCNName()	代码中文名称，如：沪银1302
int getType()	可参考 8.6, 8.7
String getWindType()	windType : FUR(期货), 测试
int getLotSize()	//交易量纲, 港股有效

注：Wind(万得)代码为万得代码，不区分大小写；原始代码为宏汇代码，区分大小写，可直接用于交易，其中上交所指数有做过代码转换，具体转换规则参见 [8.2](#)。

6.3.2 证券代码表 – 期权 (TDF_OPTION_CODE)

方法名	备注
TDF_CODE getBasicCode()	基本代码数据
以下数据为期权数据专有 (0x90 <= nType <= 0x95)。对于非期权数据，以下内容均为空。	
String getContractID()	期权合约代码
String getUnderlyingSecurityID()	标的证券代码
char getCallOrPut()	认购认沽C1(认购，则本字段为“C”；若为认沽，则本字段为“P”)
int getExerciseDate()	期权行权日，YYYYMMDD
char getUnderlyingType()	标的证券类型C3:0-A股 1-ETF (EBS - ETF, ASH - A股)
char getOptionType()	欧式美式C1(若为欧式期权，则本字段为“E”；若为美式期权，则本字段为“A”)
char getPriceLimitType()	涨跌幅限制类型C1(‘N’表示有涨跌幅限制类型，‘R’表示无涨跌幅限制类型)
int getContractMultiplierUnit()	合约单位：经过除权除息调整后的合约单位，一定是整数
long getExercisePrice()	期权行权价：经过除权除息调整后的期权行权价，右对齐，精确到厘
int getStartDate()	期权首个交易日，YYYYMMDD
int getEndDate()	期权最后交易日/行权日，YYYYMMDD
int getExpireDate()	期权到期日，YYYYMMDD

6.3.3 行情快照 - 股票(TDF_MARKET_DATA)

数据消息：MSG_DATA_MARKET 对应的结构

方法名	备注
String getWindCode()	Wind 代码
String getCode()	原始代码
int getActionDay()	业务发生日
int getTradingDay()	交易日

int getTime()	时间 (HHMMSSmmm)
int getStatus()	状态
long getPreClose()	前收盘价
long getOpen()	开盘价
long getHigh()	最高价
long getLow()	最低价
long getMatch()	最新价
long[] getAskPrice()	申卖价
long[] getAskVo()	申卖量
long[] getBidPrice()	申买价
long[] getBidVol()	申买量
long getNumTrades()	成交笔数
long getVolume()	成交总量
long getTurnover()	成交总金额
long getTotalBidVol()	委托买入总量
long getTotalAskVol()	委托卖出总量
long getWeightedAvgBidPrice()	加权平均委买价格
long getWeightedAvgAskPrice()	加权平均委卖价格
int getIOPV()	IOPV 净值估值
int getYieldToMaturity()	到期收益率
long getHighLimited()	涨停价
long getLowLimited()	跌停价
String getPrefix()	证券信息前缀
int getSyl1()	市盈率 1
int getSyl2()	市盈率 2
int getSD2()	涨跌 2 (对比上一笔)
TdfCodeInfo getCodeInfo()	代码信息 (连接关闭或清盘或重连后, 此指针无效)

6.3.4 行情快照 - 期货/期权(TDF_FUTURE_DATA)

数据消息: MSG_DATA_FUTURE 对应的结构

方法名	备注
String getWindCode()	Wind 代码
String getCode()	原始代码
int getActionDay()	业务发生日
int getTradingDay()	交易日
int getTime()	时间 (HHMMSSmmm)
int getStatus()	状态
long getPreOpenInterest()	昨持仓
long getPreClose()	前收盘价
long getPreSettlePrice()	昨结算
long getOpen()	开盘价
long getHigh()	最高价
long getLow()	最低价
long getMatch()	最新价
long getVolume()	成交总量
long getTurnover()	成交总金额
long getOpenInterest()	持仓总量
long getClose()	今收盘
long getSettlePrice()	今结算
long getHighLimited()	涨停价
long getLowLimited()	跌停价

int getPreDelta()	昨虚实度
int getCurrDelta()	今虚实度
long[] getAskPrice()	申卖价
long[] getAskVol()	申卖量
long[] getBidPrice()	申买价
long[] getBidVol()	申买量
long getAuctionPrice()	波动性中断参考价
int getAuctionQty()	波动性中断集合竞价虚拟匹配量
int getAvgPrice()	郑商所期货均价
TdfCodeInfo getCodeInfo()	代码信息 (连接关闭或清盘或重连后, 此指针无效)

6.3.5 行情快照 - 指数(TDF_INDEX_DATA)

数据消息: MSG_DATA_INDEX 对应的结构

方法名	备注
String getWindCode()	Wind 代码
String getCode()	原始代码
int getActionDay()	业务发生日
int getTradingDay()	交易日
int getTime()	时间 (HHMMSSmmm)
int getStatus()	状态
long getOpenIndex()	今开盘指数
long getHighIndex()	最高指数
long getLowIndex()	最低指数
long getLastIndex()	最新指数
long getTotalVolume()	参与计算相应指数的交易数量
long getTurnover()	参与计算相应指数的成交金额
long getPreCloseIndex()	前盘指数
TdfCodeInfo getCodeInfo()	代码信息 (连接关闭或清盘或重连后, 此指针无效)

6.3.6 逐笔成交(TDF_TRANSACTION)

数据消息: MSG_DATA_TRANSACTION 对应的结构

方法名	备注
String getWindCode()	Wind 代码
String getCode()	原始代码
int getActionDay()	业务发生日
int getTime()	时间 (HHMMSSmmm)
int getIndex()	成交编号
long getPrice()	成交价格
int getVolume()	成交数量
long getTurnover()	成交金额
int getBSFlag()	买卖方向 (买: 'B', 卖: 'S', 不明: '')
char getOrderKind()	成交类别
char getFunctionCode()	成交代码
int getAskOrder()	叫卖方委托序号
Int getBidOrder()	叫买方委托序号
TdfCodeInfo getCodeInfo()	代码信息 (连接关闭或清盘或重连后, 此指针无效)

6.3.7 逐笔委托(TDF_ORDER)

数据消息: MSG_DATA_ORDER对应的结构

方法名	备注
String getWindCode()	Wind 代码
String getCode()	原始代码
int getActionDay()	业务发生日
int getTime()	委托时间 (HHMMSSmmm)
int getOrder()	委托号
long getPrice()	委托价格
int getVolume()	委托数量
char getOrderKind()	<u>委托类别</u>
int getBroker()	经纪商编码 (PT市场有效)
char getStatus()	委托状态 (PT市场有效)
char getFlag()	标志 (PT市场有效)
char getFunctionCode()	委托代码 ('B', 'S', 'C')
TdfCodeInfo getCodeInfo()	<u>代码信息 (连接关闭或清盘或重连后, 此指针无效)</u>

6.3.8 委托队列(TDF_ORDER_QUEUE)

数据消息: MSG_DATA_ORDERQUEUE对应的结构

方法名	备注
String getWindCode()	Wind 代码
String getCode()	原始代码
int getActionDay()	业务发生日
int getTime()	时间 (HHMMSSmmm)
int getSide()	买卖方向 ('B':Bid 'A':Ask)
long getPrice()	委托价格
int getOrders()	订单数量
int getABItems()	明细个数
int[] getABVolume()	订单明细
TdfCodeInfo getCodeInfo()	<u>代码信息 (连接关闭或清盘或重连后, 此指针无效)</u>

6.3.9 BBQ 现券成交数据(TDF_BBQTRANSACTION)

数据消息: MSG_DATA_BBQTRANSACTION 对应的结构

方法名	备注
String getWindCode()	Wind 代码
int getActionDay()	业务发生日
int getTime()	时间 (HHMMSSmmm)
int getDoneID()	成交报价编号
long getPrice()	成交收益率(%)或价格 *10000
char getPriceStatus()	收益率或价格标识 1: 收益率 2: 价格
char getStatus()	收益率或价格标识 1: 收益率 2: 价格
char getDirection()	成交方向 1: done 2: gvn 3: tkn 4: 其他
char getSource()	报价机构

	1: 国际 2: 平安利顺 3: 中诚 4: 天津信唐 5: 国利
char getSpecialFlag()	标识报价是否是行权或到期 0无标记 1备注中包含到期 2备注中包含行权

6.3.10 BBQ 现券报价数据(TDF_BBQBID_DATA)

数据消息: MSG_DATA_BBQBID 对应的结构

方法名	备注
String getWindCode()	Wind 代码
int getActionDay()	业务发生日
int getTime()	时间 (HHMMSSmmm)
char getSource()	报价机构
long getBidPrice()	买入收益率或价格 x10000
long getBidVolume()	买入量 x10000
char getBidPriceStatus()	收益率或价格标识 1: 收益率 2: 价格
char getIsBid()	是否bid 0: 不是bid (普通报价) 1: 是bid (意向报价)
char getBidSpecialFlag()	标识报价是否是行权或到期 0无标记 1备注中包含到期 2备注中包含行权
char getBidStatus()	买入最优报价状态 0: 正常报价 1: 撤销报价
long getOfrPrice()	卖出收益率或价格 x10000
long getOfrVolume()	卖出量 所有量的加总, 如2000+1000, 就合并为3000; 如字符串中含- 则视为0. x10000
char getOfrPriceStatus()	收益率或价格标识 1: 收益率 2: 价格
char getIsOfr()	是否ofr 0: 不是ofr (普通报价) 1: 是ofr (意向报价)
char getOfrSpecialFlag()	标识报价是否是行权或到期 0无标记 1备注中包含到期 2备注中包含行权
char getOfrStatus()	卖出最优报价状态 0: 正常报价 1: 撤销报价
int getDoneID	成交报价编号

6.3.11 港股经纪商队列(TDF_BROKER_QUEUE)

数据消息: MSG_DATA_BROKERQUEUE 对应的结构

方法名	备注
String getWindCode()	Wind 代码
String getCode()	原始代码
int getActionDay()	业务发生日
int getAskTime()	叫卖时间 (HHMMSSmmm)
int getBidTime()	叫买时间 (HHMMSSmmm)
int getAskBrokers()	叫卖经纪个数
int getBidBrokers()	叫买经纪个数

int[] getAskBroker()	叫卖前40经纪
int[] getBidBroker()	叫买前40经纪
TdfCodeInfo getCodeInfo()	代码信息（连接关闭或清盘或重连后，此指针无效）

6.3.12 代码信息（TdfCodeInfo）

6.3.12.1 基本代码信息

方法名	备注
int getType()	类型
String getCode()	原始代码
String getName()	代码名称
TdOptionCodeInfo getExCodInfo()	扩展代码信息

6.3.12.2 扩展代码信息 - 期权(TdOptionCodeInfo)

字段名	备注
String getContractID()	期权合约代码C19
String getUnderlyingSecurityID()	标的证券代码
int getUnderlyingType()	标的证券类型C3: 0-A股 1-ETF (EBS-ETF, ASH-A股)
int getOptionType()	欧式美式C1: 若为欧式期权, 则本字段为 'E' 若为美式期权, 则本字段为 'A'
int getCallOrPut()	认购认沽C1: 认购, 则本字段为 'C' 认沽, 则本字段为 'P'
int getPriceLimitType()	涨跌幅限制类型C1: 'N' 表示有涨跌幅限制类型, 'R' 表示无涨跌幅限制类型
int getContractMultiplierUnit()	合约单位: 经过除权除息调整后的合约单位, 一定是整数
long getExercisePrice()	期权行权价: 经过除权除息调整后的期权行权价, 右对齐, 精确到厘
int getStartDate()	期权首个交易日, YYYYMMDD
int getEndDate()	期权最后交易日/行权日, YYYYMMDD
int getExerciseDate()	期权行权日, YYYYMMDD
int getExpireDate()	期权到期日, YYYYMMDD

6.3.13 交易所编号

宏定义	值	中文名
ID_MARKET_SZ	0	深圳
ID_MARKET_SH	1	上海
ID_MARKET_PT	2	新三板
ID_MARKET_HK	6	港股
ID_MARKET_CF	10	中国金融指数
ID_MARKET_SHF	11	上海金属商品交易所
ID_MARKET_CZC	12	郑州商品
ID_MARKET_DCE	13	大连商品

ID_MARKET_HKF	23	香港交易所期货/期权
---------------	----	------------

7 示例

Demo 是一份 TDF JAVA API 的示例。

8 附录

8.1 状态含义

状态	含义
null	正常交易
0	首日上市
1	增发新股
2	上网定价发行
3	上网竞价发行
'A'	交易节休市
'B'	整天停牌
'C'	全天收市
'D'	暂停交易
'E' (START)	启动交易盘
'F' (PRETR)	盘前处理
'G' (DEL)	不可恢复交易的熔断阶段（上交所的N）
'H' (HOLIDAY)	放假
'I' (OCALL)	开市集合竞价
'J' (ICALL)	盘中集合竞价
'K' (OPOBB)	开市订单簿平衡前期
'L' (IPOBB)	盘中订单簿平衡前期
'M' (OOBB)	开市订单簿平衡
'N' (IOBB)	盘中订单簿平衡
'O' (TRADE)	连续撮合
'P' (BREAK)	休市
'Q' (VOLA)	波动性中断
'R' (BETW)	交易间
'S' (NOTRD)	非交易服务支持
'T' (FCALL)	固定价格集合竞价

'U' (POSTR)	盘后处理
'V' (ENDTR)	结束交易
'W' (HALT)	暂停
'X' (SUSP)	停牌
'Y' (ADD)	新增产品
'Z' (DEL)	可删除的产品
'd'	集合竞价阶段结束到连续竞价阶段开始之前的时段（如有）
'q'	可恢复交易的熔断时段(上交所的M)

8.2 上证指数代码转换对应表

系统中对上证指数代码做了转换，可参加下表：

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
商业指数	000005	999995
地产指数	000006	999994
公用指数	000007	999993
综合事业	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，其对应得上证所代码是 000999。

8.3 委托类别和委托代码含义

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

8.4 成交类别和成交代码含义

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
1	-	市价（深交所V5）
0	-	限价（深交所V5）
U	-	本方最优（深交所V5）

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

8.5 行情数据(Market Data)状态与交易所状态转换表

深交所状态名称	说明	宏汇TDF行情数据对应状态
0	开市前	数字0
1	开盘集合竞价	I
2	连续竞价阶段	字母0
3	盘中临时停市	W
4	收盘集合竞价	J
5	集中竞价闭市	A
6	协议转让结束	B
7	闭市	C

上交所状态名称	说明	宏汇TDF行情数据对应状态
. ADD	新产品	Y
. BETW	交易间, 禁止任何交易活动	R
. BREAK	休市, 例如: 午餐休市。 无撮合和市场内部信息披露	P
. CLOSE	闭市, 自动计算闭市价格	C
. DEL	产品待删除	Z
. ENDTR	交易结束	V
. FCALL	固定价格集合竞价	T
. HALT	暂停, 除了自有订单和交易的查询之外, 任何交易活动都被禁止	W
. SUSP	停牌 (SUSP 和HALT 的区别在于 SUSP 时可以撤单)	X
. ICALL	盘中集合竞价	J
. IOBB	盘中集合竞价订单簿平衡 (OBB)	N
. IPOBB	盘中集合竞价 PreOBB	L
. OCALL	开市集合竞价	I
. OOB	开市集合竞价 OBB	M
. OPOBB	开市集合竞价订单簿平衡 (OBB) 前期时段	K
. NOTRD	非交易支持非交易服务	S
. POSTR	盘后处理	U
. PRETR	盘前处理	F
. START	启动	E

. TRADE	连续自动撮合	字母0
. VOLA	连续交易和集合竞价交易的波动性中断 (VOLA)	Q

注：宏汇 TDF 系统默认将初始化状态和未识别的状态置为数字 0。

8.6 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_BT_EX_HASWARRANT	0x80000000
已经删除	ID_BT_EX_DELETE	0x40000000
退市整理板块	ID_BT_EX_DELISTING	0x08000000
风险警示板块	ID_BT_EX_RISKWARNING	0x04000000
其他	ID_BASECLASS_OTHER	0xf0

8.7 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Level12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
政府债券	ID_BT_BOND_NA	0x30
企业债券	ID_BT_BOND_CORP	0x31

类型名称	宏定义	类型值
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
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
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(Interal)	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

9 FAQ (常见问题)

请参考另一份文档《TDFAPI 使用常见问题》。