



**SUNGARD** 金仕达

# 个股期权系统 OTP\_API 用户手册

文档标识

文档名称	金仕达个股期权系统 OTP_API 用户手册
版本号	<V1.0>
状况	<input type="radio"/> 草案 <input type="radio"/> 评审过的 <input type="radio"/> 更新过的 <input checked="" type="radio"/> 定为基线的

文档修订历史

版本	日期	描述	文档所有者
V1.0	<2014-01-08>	创建文档	沈明明

此版本文档的正式核准

姓名	签字	日期

分发控制

副本	接受人	机构

# 目 录

1.	第一章 介绍.....	3
1.1	概述.....	3
1.2	API 文件.....	3
2.	第二章 体系结构.....	4
2.1	通讯模式.....	4
2.2	数据流.....	4
2.3	通讯流程.....	5
2.4	报单指令（特别说明）.....	6
2.4.1	指令正常.....	6
2.4.2	指令异常.....	6
2.5	非交易报单指令（特别说明）.....	8
3.	第三章 接口模式.....	8
3.1	对话模式的编程接口.....	8
3.2	私有模式的编程接口.....	9
3.3	广播模式的编程接口.....	9
4.	第四章 开发接口.....	9
4.1	工作线程.....	9
4.2	通用规则.....	10
4.3	CKSOTPTraderSpi 接口.....	10
4.4	CKSOTPTraderApi 接口.....	11
5.	第五章 示例代码.....	11
6.	第六章 问题反馈.....	11

## 1. 第一章 介绍

### 1.1 概述

本手册为 Kingstar OTP API 与 Kingstar 服务器之间的通讯数据包接口规范。通过 API，投资者可以接收来自交易所的行情数据，并发送交易指令，接收相应的反馈和交易状态等信息。

### 1.2 API 文件

Kingstar 平台上使用的 OTP API 是一个基于 C++ 的类库，通过使用和扩展类库提供的接口来实现客户端和 Kingstar 服务器之间的数据传输。通过 API，客户端可以发出或撤销普通单、非交易报单录入查询及操作、查询委托或交易状态、查询账户实时信息和交易头寸。Windows 平台下 API 程序库所包含文件如下：

库文件名	库文件描述
KSOTPTTraderApi.h	交易接口头文件
KSOTPTUserApiDataType.h	定义了 API 所需的一系列数据类型的头文件
KSOTPTUserApiStruct.h	定义了一系列业务相关的数据结构的头文件
ksotptraderapi.lib	API 导入库文件
ksotptraderapi.dll	API 动态链接库
lkcdll.dll	API 运行需附加的动态链接库
ksPortalAPI.dll	门户集成功能接口库
SSPXEncode.dll	Spx 协议编码接口库
KSInterB2C.lkc	客户端授权文件

注：编程请使用头文件命名空间 namespace KingstarAPI。

## 2. 第二章 体系结构

### 2.1 通讯模式

金仕达的 API 和 Kingstar 服务器之间使用的通讯模式包括以下三种模式：

对话模式，客户端给 Kingstar 服务器发送请求，该请求被交易所端接收和处理，并给予响应。

例如报单、查询等。这种通讯模式与普通的客户/服务器模式相同。

私有模式，Kingstar 服务器把特定的私人信息发送给对应的客户端，包括持仓信息、交易确认信息等。

广播模式，Kingstar 服务器将把公告等信息发送给所有的注册用户。

通讯模式和网络的连接不一定存在简单的一对一的关系。也就是说，一个网络连接中可能传送多种不同通讯模式的报文，一种通讯模式的报文也可以在多个不同的连接中传送。

### 2.2 数据流

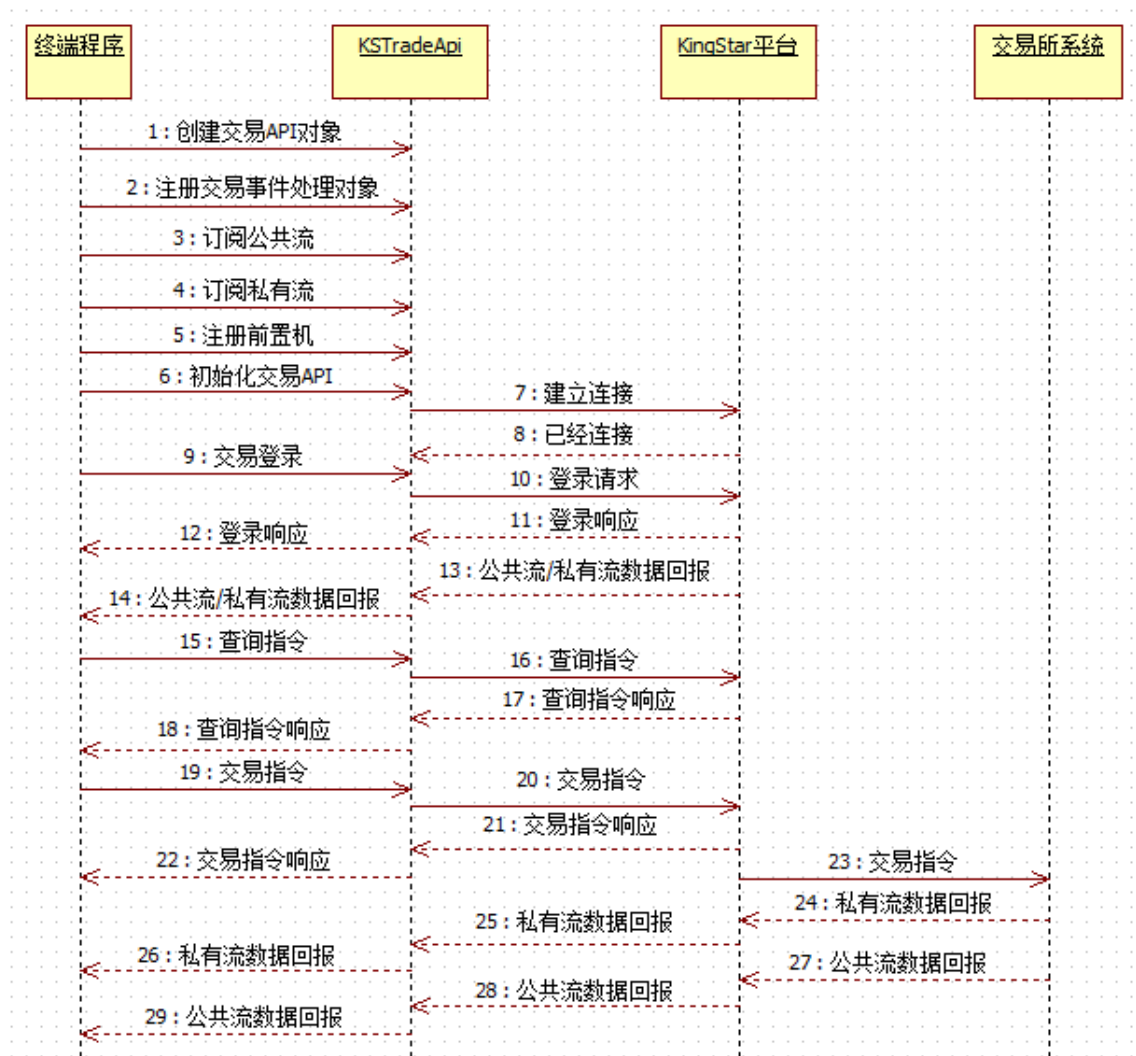
Kingstar 服务器提供了对话、私有、广播等三种通讯模式。

在对话模式中，传输的是对话数据流和查询数据流。对话和查询数据流是双向数据流，客户端发出请求，Kingstar 服务器返回结果。Kingstar 服务器并不保存对话和查询数据流。当故障发生时，比如连接中断后又重新连接，对话和查询数据流将会重置，通讯途中的数据可能会丢失。

在私人通讯模式中，传输的是私人数据流。私人数据流是单向数据流，Kingstar 服务器就是利用它来把相应的私人信息发给提出申请的客户端。私人信息包括，风险提示、指令状态、指令确认、交易确认等。私人数据流是可靠的。当客户端和 Kingstar 服务器失去连接后，在同一交易日的任何时间，客户端都可重新连上 Kingstar 服务器，获取一系列指定的私人信息而不用担心这些数据会丢失。

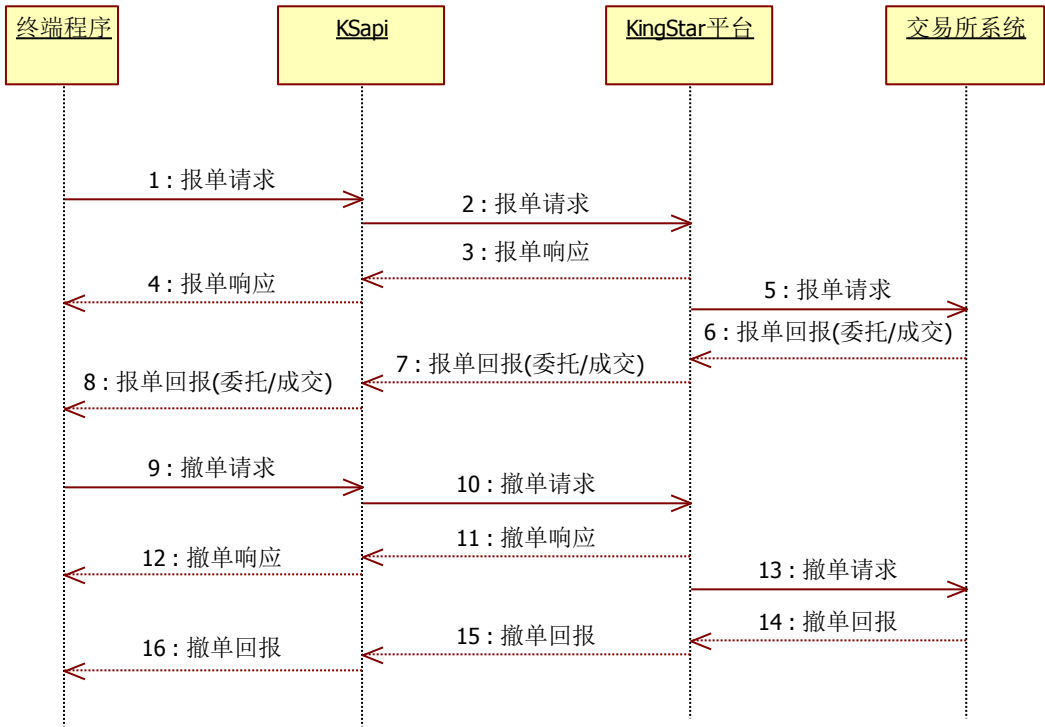
在广播通讯模式中，传输的是公共数据流。和私人数据流一样，它是单向数据流，而且可靠。区别在于，广播通讯数据会发送到所有连接的客户端上。它的主要用途就是发布公共设备的状态信息或重要的公共信息。

## 2.3 通讯流程



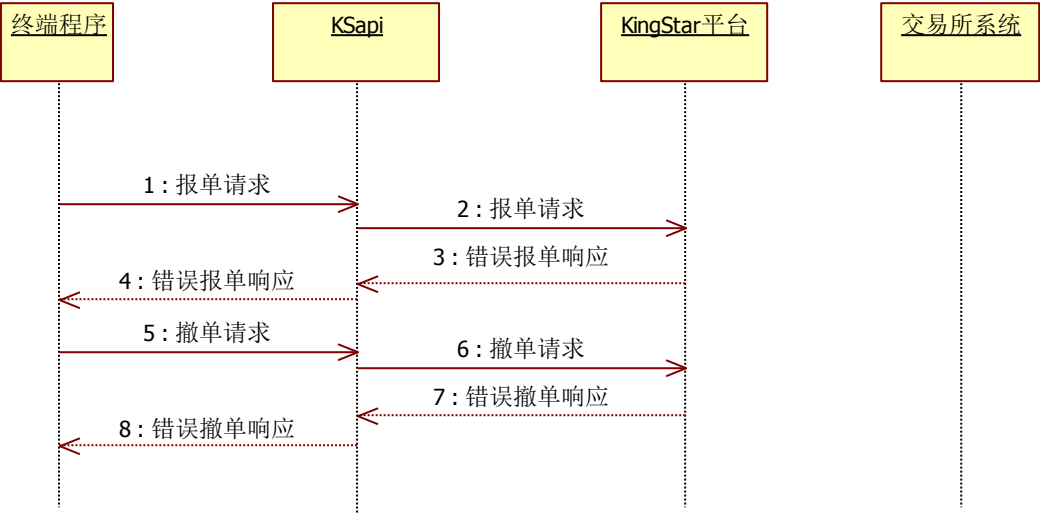
2.4 报单指令（特别说明）

2.4.1 指令正常

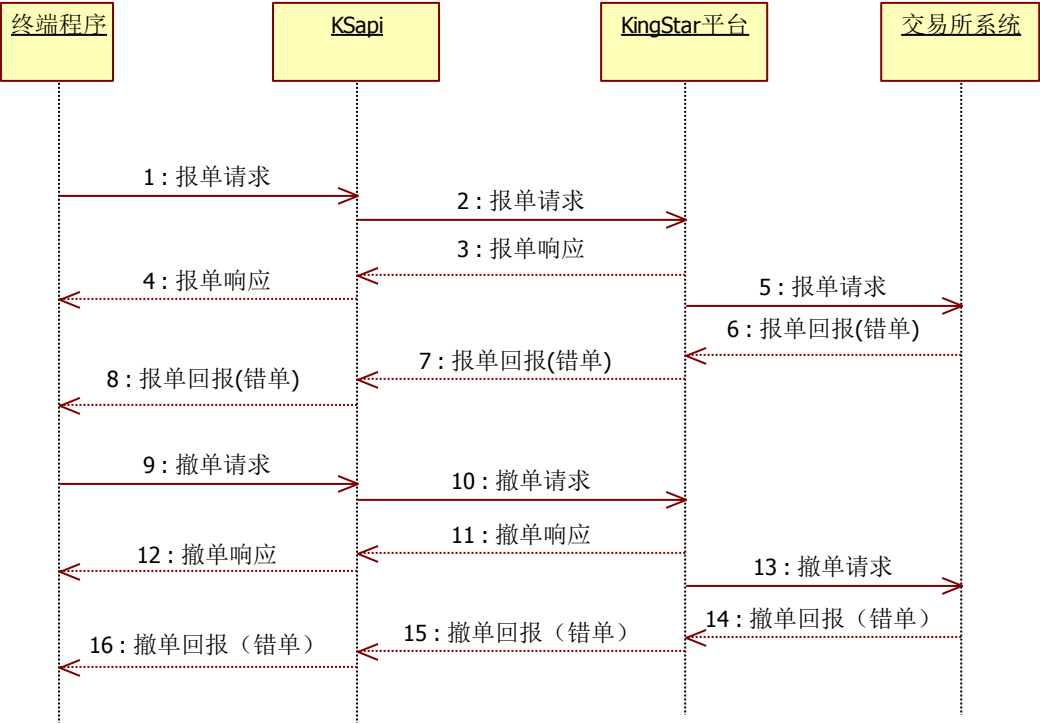


2.4.2 指令异常

当终端程序发出报单、撤单指令错误时，分两种情况：图（一）显示指令填写错误，Kingstar 平台会在送交交易所之前进行检测给出错误提示；图（二）显示交易所系统对所发出的指令进行处理，当不符合交易条件时，交易所给出错误提示。



(一)



(二)



## 2.5 非交易报单指令（特别说明）

非交易报单录入即指行权单录入，流程与普通单类似。撤销非交易报单执行时使用非交易报单操作函数。

## 3. 第三章 接口模式

Kingstar API 提供了两种编程接口：CKSOTPTraderApi 与 CKSOTPTraderSpi。这两个接口对 FTD 协议进行了封装，方便客户端应用程序的开发。客户端可使用 CKSOTPTraderApi 来发出操作请求，并通过继承 CKSOTPTraderSpi 并重载回调函数来处理 Kingstar 服务器的响应。

### 3.1 对话模式的编程接口

对话模式的通讯函数常常这样定义：

```
请求:  int CKSOTPTraderApi::ReqXXX(
        CKSOTPXXXField *pReqXXX,
        int nRequestID)

响应:  void CKSOTPTraderSpi::OnRspXXX(
        CKSOTPXXXField *pRspXXX,
        CKSOTPRspInfoField *pRspInfo,
        int nRequestID,
        bool bIsLast)
```

其中请求接口第一个参数为请求的内容，不能为空。

第二个参数为请求号。请求号由客户端应用程序负责维护，正常情况下每个请求的请求号不要重复。在接收 Kingstar 服务器的响应时，可以得到当时发出请求时填写的请求号，从而可以将响应与请求对应起来。

当客户端收到 Kingstar 服务器发出的反馈时，回调函数 CKSOTPTraderSpi 被激活。如果反馈多于一条，回调函数 CKSOTPTraderSpi 会反复被激活，直到数据全部接收完毕。

回调函数的第一个参数为响应的具体数据，如果出错或没有结果有可能为 NULL。

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

第三个参数为请求号，即原来发出请求时填写的请求号。

最后一个参数是结束标识，其值为“true”时表示进行的是该请求的最后一个反馈。

### 3.2 私有模式的编程接口

下面的例子为私有模式的常用接口：

```
void CKSOTPTraderSpi::OnRtnXXX(CKSOTPXXXField *pXXX)
void CKSOTPTraderSpi::OnErrRtnXXX(CKSOTPXXXField *pXXX,
                                   CKSOTPRspInfoField *pRspInfo)
```

在私有模式里并没有连接 API 与 Kingstar 服务器的行情函数。当 Kingstar 服务器发出私有数据流时，CKSOTPTraderSpi 的回调函数将被激活。所有回调函数的第一个参数都是 Kingstar 服务器返回的具体内容。OnErrRtnXXX 函数的第二个参数是报错时详细的错误信息。

### 3.3 广播模式的编程接口

使用广播模式时，客户端可以用以下两种方式与 Kingstar 服务器进行通讯：

```
void CKSOTPTraderSpi::OnRtnInstrumentStatus(
    CKSOTPIstrumentStatusField *pInstrumentStatus)
void CKSOTPTraderSpi::OnRtnDepthMarketData(
    CKSOTPDepthMarketDataField *pDepthMarketData)
```

回调函数 *OnRtnInstrumentStatus* 用于通知客户端合约状态的变化。

回调函数 *OnRtnDepthMarketData* 用于公布最新的交易所行情数据。

## 4. 第四章 开发接口

### 4.1 工作线程

客户端进程需要两种线程，一是应用程序主线程，另一种是 API 工作线程。API 工作线程连接了客户端和 Kingstar 服务器。

API 线程是安全线程，客户端的应用程序可以同时使用两种或多种的工作线程，而无须担心线程冲突。客户端的应用程序要能尽快的处理回调信息，这样才能避免未处理的回调信息堵塞工作线程。要避免

免通讯堵塞，客户端的应用程序需要使用缓冲层来储存从 Kingstar 服务器接收的数据，当然缓冲层也可以用来保护客户端自有数据，以便使之与 API 的数据区分开。

## 4.2 通用规则

客户端应用程序需要经过两步才能连接到 Kingstar 服务器：初始化与功能启用。

使用交易 API，客户端交易应用程序需要编写：

- (1) 创建一个“CKSOTPTraderApi”实例。
- (2) 创建一个处理来自“CKSOTPTraderSpi”接口的事件处理器，然后使用“CKSOTPTraderApi”的“RegisterSpi”函数记录下这些事件。
- (3) 使用“CKSOTPTraderApi”的“SubscribePrivateTopic”函数处理私有数据流。
- (4) 使用“CKSOTPTraderApi”的“SubscribePublicTopic”函数处理公共数据流。
- (5) 使用“CKSOTPTraderApi”的“RegisterFront”函数记录 Kingstar 服务器的前端地址。客户端多运行几次这种函数，以便与服务器建立更可靠的联系。强烈建议。
- (6) 使用“CKSOTPTraderApi”的“Init”函数来连接 Kingstar 服务器。
- (7) Kingstar 服务器连上之后，“CKSOTPTraderSpi”接口的回调函数“OnFrontConnected”将被激活。函数运行过程中，客户端的应用程序需要使用“CKSOTPTraderApi”的“ReqUserLogin”函数来提交“login”请求。
- (8) 当 Kingstar 服务器确认登陆成功后，“CKSOTPTraderSpi”接口回调函数“OnRspUserLogin”将被激活。
- (9) 这样，客户端与 Kingstar 服务器的通讯就建立起来了。客户端交易应用程序可以使用其他金仕达 API 业务来与 Kingstar 服务器进行通讯。

其他注意事项：

- (1) API 请求的输入参数不能为 NULL；
- (2) API 请求的返回参数，0 表示正确，其他表示返回错误。

## 4.3 CKSOTPTraderSpi 接口

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

参见《金仕达个股期权系统 OTP\_API Reference.chm》

## 4.4 CKSOTPTraderApi 接口

CKSOTPTraderApi 接口提供给用户的功能包括，报单与报价的录入、报单与报价的撤销、报单与报价的查询、成交单查询、持仓查询、合约查询、合约交易状态查询、交易所公告查询、非交易报单查询、行权单录入与撤销、非交易报单锁定与解锁等功能。

参见《金仕达个股期权系统 OTP\_API Reference.chm》

## 5. 第五章 示例代码

参见开发包内的示例文件夹或《金仕达个股期权系统 OTP\_API Reference.chm》。

## 6. 第六章 问题反馈

若接口在使用过程有任何问题，请将详细信息反馈至电子邮箱：

[Mingming.shen@sungard.com](mailto:Mingming.shen@sungard.com) 或 [Yanyun.liu@sungard.com](mailto:Yanyun.liu@sungard.com)。

谢谢。