

易盛行情 API 使用说明

一. 接口提供方式

以动态链接库(dll)方式提供, 包括 EsunnyQuot.h,EsunnyQuot.lib,EsunnyQuot.dll,易盛行情接口说明.doc 四个文件

二. API 函数说明

1.创建接口指针(使用前调用)

`IESunnyQuotClient * __cdecl CreateEsunnyQuotClient(IESunnyQuotNotify * notify);`

Notify 为实现行情通知接口的实例指针

返回值为返回的行情操作实例指针, 失败返回 NULL

2.释放接口指针(使用后调用)

`void __cdecl DelEsunnyQuotClient(IESunnyQuotClient * client);`

client 为 CreateEsunnyQuotClient 创建的行情操作实例指针

无返回值

3.连接行情数据服务器

`virtual int __cdecl Connect(const char *ip,int port)=0;`

参数 ip 为服务器的 IP

参数 port 为服务器的端口

返回值 0 表示连接成功, 否则失败

4.断开当前连接

`virtual void __cdecl Disconnect()=0;`

5. 登录行情数据服务器

`virtual int __cdecl Login(const char *user,const char *password)=0;`

参数 user 登录用户名

参数 password 登录密码

返回值 0 表示登录指令发送成功, 否则发送失败, 是否登录成功通过回调 OnRspLogin 获得

6.请求/取消订阅品种即时行情

`virtual int __cdecl RequestQuot(const char *market,const char *stk,int need)=0;`

参数 market 市场中文名

参数 stk 合约代码

参数 need 1 表示订阅, 0 表示取消订阅

返回值 0 表示订阅请求成功, 否则失败, 成功之后将会在 OnStkQuot 中收到品种的即时行情

7.请求品种历史行情

`virtual int __cdecl RequestHistory(const char *market,const char *stk,int period)=0;`

参数 market 市场中文名

参数 stk 合约代码

参数 period 周期 1: 1 分钟, 2: 5 分钟, 3: 60 分钟, 4: 日线

返回值 0 表示请求成功, 否则失败, 历史数据结果在 OnRspHistoryQuot 返回

8. 请求品种明细数据

```
virtual int __cdecl RequestTrace(const char *market,const char *stk,const char *date)=0;
```

参数 market 市场中文名

参数 stk 合约代码

参数 date 要请求明细的日期, 形如 20090901

返回值 0 表示请求成功, 否则失败, 历史数据结果在 OnRspTraceData 返回

9. 登录反馈回调函数

```
virtual int __cdecl OnRspLogin(int err,const char *errtext)=0;
```

参数 err 错误号 0 表示登录成功, 否则失败

参数 errtext 错误信息

返回值 0 表示成功, 否则失败

10. 物理连接出错回调函数

```
virtual int __cdecl OnChannelLost(int err,const char *errtext)=0;
```

参数 err 错误号

参数 errtext 错误信息

返回 0 表示成功, 否则失败

11. 即时行情回调函数

```
virtual int __cdecl OnStkQuot(struct STKDATA *pData)=0;
```

参数 pData 行情结构指针

返回值 0 表示成功, 否则失败

12. 历史行情反馈回调函数

```
virtual int __cdecl OnRspHistoryQuot(struct STKHISDATA *pHisData)=0;
```

参数 pHisData 历史行情数据结构指针

返回值 0 表示成功, 否则失败

13. 明细数据反馈回调函数

```
virtual int __cdecl OnRspTraceData(struct STKTRACEDATA *pTraceData)=0;
```

参数 pTraceData 明细数据结构指针

返回值 0 表示成功, 否则失败

14. 市场品种信息反馈回调函数

```
virtual int __cdecl OnRspMarketInfo(struct MarketInfo *pMarketInfo,int bLast)=0;
```

参数 pMarketInfo 市场合约数据结构指针

参数 bLast 为 1 表示是最后一个, 0 表示后面还有市场数据

返回值 0 表示成功, 否则失败

三. 数据结构

1. 即时数据

struct STKDATA //即时数据结构

```
{
    char Market[32];        //市场中文名
    char Code[16];          //合约代码

    float YClose;           //昨收盘
    float YSettle;          //昨结算
    float Open;             //开盘价
    float High;             //最高价
    float Low;              //最低价
    float New;              //最新价
    float NetChg;           //涨跌
    float Markup;           //涨跌幅
    float Swing;            //振幅
    float Settle;           //结算价
    float Volume;           //成交量
    float Amount;           //持仓量

    float Ask[5];           //申卖价
    float AskVol[5];        //申卖量
    float Bid[5];           //申买价
    float BidVol[5];        //申买量

    float AvgPrice;         //平均价

    float LimitUp;          //涨停板
    float LimitDown;        //跌停板
    float HistoryHigh;      //合约最高
    float HistoryLow;       //合约最低

    long YOPI;              //昨持仓
    float ZXSD;              //昨虚实度
    float JXSD;              //今虚实度
    float CJJE;              //成交金额
};
```

2. 品种历史数据

struct HISTORYDATA //历史数据

```
{
    char time[20]; //时间,形如: 2009-06-01 08:00:00
    float fOpen;    //开盘
    float fHigh;    //最高
    float fLow;     //最低
```

```

        float fClose;        //收盘
        float fVolume;       //成交量(手)
        float fAmount;       //成交额(元)
    };

    struct STKHISDATA //品种历史数据
    {
        char        Market[32]; //市场中文名
        char        Code[16];   //品种代码
        short        nPeriod;    //周期
        short        nCount;     //历史数据数目
        struct HISTORYDATA HisData[1];
    };

```

3. 品种成交明细数据

```

    struct STOCKTRACEDATA //成交明细
    {
        char time[20]; //时间,形如: 2009-06-01 08:00:00
        float m_NewPrice; //最新价
        float m_Volume;   //总量
        float m_Amount;   //持仓
        float m_BuyPrice; //委买价
        float m_SellPrice; //委卖价
        float m_BuyVol;   //申买量
        float m_SellVol;  //申卖量
    };

```

```

    struct STKTRACEDATA //品种明细
    {
        char        Market[32]; //市场中文名
        char        Code[16];   //品种代码
        unsigned short nCount;   //记录数目
        struct STOCKTRACEDATA TraceData[1];
    };

```

```

    struct StockInfo//合约信息
    {
        char szCode[16]; //合约代码
    };

```

```

    struct MarketInfo//单个市场的合约信息
    {
        char        Market[32]; //市场中文名
        unsigned short stocknum; //合约数目
        struct StockInfo stockdata[1]; //合约信息数据
    };

```

四. 使用例子

```
//EsunnyQuotNotify.h
#include "EsunnyQuot.h"
class CESunnyQuotNotify
{
public:
    CESunnyQuotNotify(void);
    ~CESunnyQuotNotify(void);
public:
    virtual int __cdecl OnRspLogin(int err,const char *errtext);
    virtual int __cdecl OnChannelLost(int err,const char *errtext);
    virtual int __cdecl OnStkQuot(struct STKDATA *pData);
    virtual int __cdecl OnRspHistoryQuot(struct STKHISDATA *pHisData);
    virtual int __cdecl OnRspTraceData(struct STKTRACEDATA *pTraceData);
    virtual int __cdecl OnRspMarketInfo(struct MarketInfo *pMarketInfo,int bLast);
};

//EsunnyQuotNotify.cpp
#include "esunnyquotnotify.h"

CESunnyQuotNotify::CESunnyQuotNotify(void)
{
}

CESunnyQuotNotify::~CESunnyQuotNotify(void)
{
}

int CESunnyQuotNotify::OnRspLogin(int err,const char *errtext)
{
    if(err==0)
    {
        //登录成功
    }
    return 0;
}

int CESunnyQuotNotify::OnChannelLost(int err,const char *errtext)
{
    //连接断开
    return 0;
}
```

```

int CESunnyQuotNotify::OnStkQuot(struct STKDATA *pData)
{
    //收到行情即时数据
    return 0;
}

int CESunnyQuotNotify::OnRspHistoryQuot(struct STKHISDATA *pHisData)
{
    //收到行情历史数据
    return 0;
}

int CESunnyQuotNotify::OnRspTraceData(struct STKTRACEDATA *pTraceData)
{
    //收到行情明细数据
    return 0;
}

int CESunnyQuotNotify:: OnRspMarketInfo(struct MarketInfo *pMarketInfo,int bLast)
{
    //收到市场品种数据
    return 0;
}

```

//main.cpp(部分代码)

```
#include "EsunnyQuotNotify.h"
```

```
CESunnyQuotNotify m_QuotNotify;;
```

```
IESunnyQuotClient *m_pQuotClient=CreateEsunnyQuotClient((IESunnyQuotNotify
*)&m_QuotNotify);
```

```
m_pQuotClient->Connect("123.15.58.21",6062);
```

```
m_pQuotClient->Login("guest","guest");
```

```
m_pQuotClient->RequestQuot("郑州商品","SR201",TRUE);
```

```
m_pQuotClient->Disconnect();
```

```
DelEsunnyQuotClient(m_pQuotClient);
```