KSFT\_API4Spy设计

# 修订历史

|  |  |  |
| --- | --- | --- |
| 修改日期 | 修改备注 | 修订人 |
| 2013-09-16 | 创建KSFT\_API4Spy文档 | 刘艳云 |

# 接口设计

// 初始化该API，必须先调用，失败返回false，允许重复调用；

bool SpyInit();

// 销毁释放该API，可以最后调用，允许重复调用；

void SpyFinalize();

// 服务启动，调用时，触发Spy的外部web服务，默认情况下Spy服务不启动，启动失败返回false；

bool SpyStartWebServer(char \*ip="127.0.0.1", int port = 8080);

// 服务关闭，调用时，关闭Spy的web服务，关闭失败返回false；

bool SpyStopWebServer();

// 获取当前Spy操作失败时的错误代码和错误信息，如果没有则返回false，每条错误信息获取后即不再保留，且当前只保留最新的一条信息；

bool SpyGetErrorInfo(int \*err\_code, char \*info, int size);

// 获取Spy数据值，根据key，如果没有对应key，则返回false

bool SpyGet(const char \*key, char \*value);

// 设置Spy数据值，根据key，如果设置失败，则返回false

bool SpyPut(const char \*key, char \*value);

// 删除Spy数据，根据key，如果没有对应key，则返回false, recursive为真时，同时清除key下属的成员key数据；

bool SpyDelete(const char \*key, const bool recursive=false);

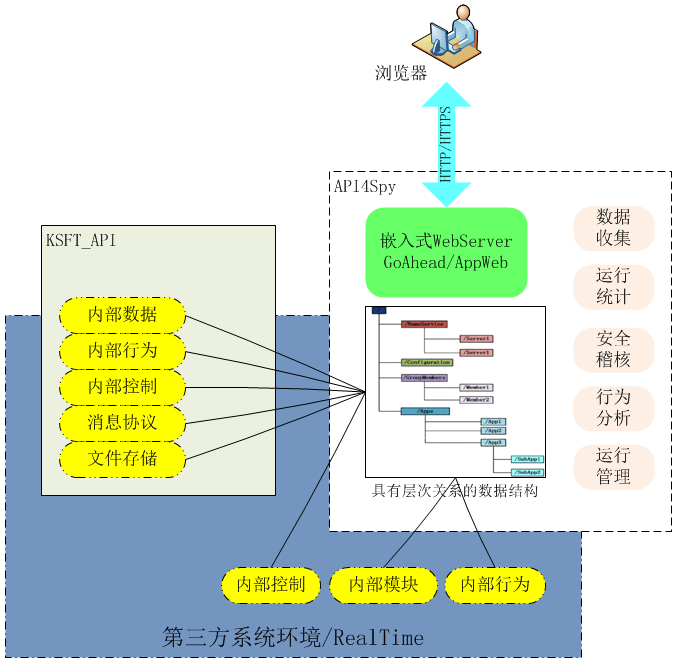
// 清除所有Spy数据，并不触发或处理其他关联功能，如果清除失败，则返回false

bool SpyClearAll();

// 导出所有Spy数据值，格式为(key,value)集合，如果导出失败，则返回false

bool SpyDumpAll(const char \*file\_name="ksft\_api4spy\_dump.txt" );

# 内部结构



typedef struct Node{

struct Node\* parent;//当前结点的父结点

string Key;//当前结点的key

string Value;//当前结点的value值

map<string, Node \*> childNodes;//子结点的<key,value>

string timestep;//时间戳，用作显示更新

}Node;

//根结点

typedef struct NodeTree{

string root\_value;//key为’/’的结点value值

map<string, Node \*> childNodes;

}NodeTree;

# 数据区定义

采用<key,value>格式，如下。

## key

1实例：/s\_nn,值为实例号

如交易实例/s\_TradeApi\_n；

行情实例/s\_MdApi\_n

2 资源：/s\_nn/thread；/s\_nn/queue；/s\_nn/obj

如交易线程/s\_ TradeApi \_n/threadTrade；

查询请求队列/s\_ TradeApi \_n/queueQueryReq

3 对象信息：队列大小/s\_nn/queue/size；队列数据/s\_nn/queue/data；

线程状态/s\_nn/thread/state

## value(使用JSONCPP)

叶子结点value的一般形式

交易线程的状态：

[＂open＂]

非叶子结点value的一般形式

交易线程:

{ "state":"open" }

查询请求队列：

{"size":\*\* , "data":"\*\*\*\*" }

# 数据分区

|  |  |  |
| --- | --- | --- |
| Key | Value | Timestep |
| /KSFT\_GLOBAL  /交易实例\_n  /线程  /查询线程  /交易线程  /推送线程  /链接线程  /队列  /查询请求队列  /交易请求队列  /查询应答队列  /交易应答队列  /推送回报队列  /行情实例\_n  /线程  /会话线程  /推送线程  /链接线程  /队列  /会话请求队列  /会话应答队列  /推送回报队列 | CPU资源占用率、交易实例数、行情实例数      状态（开启、繁忙、空闲、停止）  状态  状态  状态    队列大小、队列数据  队列大小、队列数据  队列大小、队列数据  队列大小、队列数据  队列大小、队列数据      状态  状态  状态  队列大小、队列数据  队列大小、队列数据  队列大小、队列数据 | 时间戳  HH:MM:SS:SSS |