# 统一SDK Agent对外提供接口设计

## 接口定义

### 1.1 启停sdk接口

/\*

### \* 启动SDK agent

\* 参数： callback\_func 回调接口

\* imei 串号

\* npsIp Nps服务器地址

\* 返回值： 0 成功 非0 失败

\*/

MEETINGSDKAGENT\_API int Start(PFunc\_CallBack callback\_func,char \*imei,char \*npsIp);

/\*

\* 启SDK agent 鉴权传appKey uid 和 视讯号

\*/

MEETINGSDKAGENT\_API int StartEx(PFunc\_CallBack callback\_func,char \*appKey,char\* uid,char\* phoneId,char \*npsIp);

/\*

### \* 停止SDK agent (同步)

\*/

MEETINGSDKAGENT\_API int Stop();

### 1.2 会议终端属性相关接口

/\*

\* 修改终端显示名称

\* 参数： newName 新的显示名称

\* ctx 上下文参数

\*/

MEETINGSDKAGENT\_API int ModifyDisplayName(char \*newName,long long ctx);

/\*

\* 根据视讯号查询显示名称

\* 参数： phoneId 视讯号

\*/

MEETINGSDKAGENT\_API int QueryDisplayNameByPhoneId(char \*phoneId);

/\*

### \* 取得设备串号

\* 参数：devSerialNo 存储串号的指针，默认长度24字节

\*/

MEETINGSDKAGENT\_API int GetSerialNo(char \*devSerialNo);

### 1.3 创建会议接口

/\*

### \* 创建会议

\* 参数： pUsers 被邀请用户列表

\* count 被邀请用户个数

\*/

MEETINGSDKAGENT\_API int CreateMeeting(ContactInfo \*pUsers,int count);

/\*

\* 创建预约会议

\* 参数： 会议开始时间，年月日时分秒

\*/

MEETINGSDKAGENT\_API int CreateDateMeeting(int year,int month,int day,int hour,int min,int second);

### 1.4 会议控制接口

/\*

### \* 加入会议

\* 参数： meetingId 参加的会议ID

\*/

MEETINGSDKAGENT\_API int JoinMeeting(int meetingId);

/\*

### \* 申请发言

\*/

MEETINGSDKAGENT\_API int ApplyToSpeak();

/\*

### \* 停止发言

\*/

MEETINGSDKAGENT\_API int StopSpeak();

/\*

### \* 邀请参会

\* 参数： meetingId 邀请别人参加的会议ID

\* pUsers 被邀请用户列表

\* count 被邀请用户个数

\*/

MEETINGSDKAGENT\_API int InviteParticipants(int meetingId,ContactInfo \*pUsers,int count);

/\*

### \* 向会议中的其他用户发送消息 （指令、数据透传接口，支持一对一，一对其他所有人）

\* 参数：messageId 消息Id

\* pData 消息附带参数

\* dataLen 消息附带参数的长度（最大不超过160）

\* toPhoneId 消息目的参会者的视讯号，为NULL时，广播给其他所有参会者

\*/

MEETINGSDKAGENT\_API int SendUIMessage(int messageId,char \*pData,int dataLen,char \*toPhoneId);

/\*

### \* 让指定用户开始发言

\*/

MEETINGSDKAGENT\_API int RequireUserSpeak(char\* phoneId);

/\*

### \* 让指定用户停止发言

\*/

MEETINGSDKAGENT\_API int RequireUserStopSpeak(char\* phoneId);

/\*

### \* 退出当前会议

\*/

MEETINGSDKAGENT\_API int ExitMeeting();

/\*

### \* 开始直播 （单直播url）

\*/

MEETINGSDKAGENT\_API int StartBroadcast(char\* url);

/\*

### \* 开始直播 （双直播url）

\*/

MEETINGSDKAGENT\_API int StartBroadcastEx(char\* url1,char\* url2);

/\*

### \* 停止直播

\*/

MEETINGSDKAGENT\_API int StopBroadcast();

/\*

### \* 窗口布局调整结束（直播功能使用，用于通知底层调整直播视频布局）

\*/

MEETINGSDKAGENT\_API void AdjustVideoLayoutEnd();

/\*

### \* 开始本地录制

\*/

MEETINGSDKAGENT\_API int StartLocalRecord();

/\*

### \* 停止本地录制

\*/

MEETINGSDKAGENT\_API int StopLocalRecord();

/\*

### \* 打开摄像头

\* 参数：videoId 1：主摄像头 2：辅摄像头

\*/

MEETINGSDKAGENT\_API int OpenVideo(int videoId);

/\*

### \* 关闭摄像头

\*/

MEETINGSDKAGENT\_API int CloseVideo();

/\*

### \* 设置麦克风静音状态

\* 参数：muteState 0：取消静音 1：静音

\*/

MEETINGSDKAGENT\_API int SetMicMuteState(int muteState);

/\*

### \* 开始桌面共享 （本机屏幕分享）

\*/

MEETINGSDKAGENT\_API int StartScreenSharing();

/\*

### \* 停止桌面共享 （本机屏幕分享）

\*/

MEETINGSDKAGENT\_API int StopScreenSharing();

/\*

### \* 设置视频区域窗口句柄，用于计算推流时视频窗口位置

\*/

MEETINGSDKAGENT\_API int SetViewReginHwnd(void\* hwnd);

/\*

### \* 开始渲染用户的音视频流

\* 参数：phoneId 用户视讯号 hwndCam 摄像头视频显示窗口句柄 hwndDoc 文档视频显示窗口句柄 streamType 流类型

\*/

MEETINGSDKAGENT\_API int StartRenderUserStreams(char\* phoneId,void\* hwndCam,void\* hwndDoc,int streamType);

### 1.5 查询会议接口

/\*

### \* 获取可参加的会议列表

\*/

MEETINGSDKAGENT\_API int GetMeetingList();

/\*

### \* 判断会议是否存在

\* 参数：meetingId 会议ID

\*/

MEETINGSDKAGENT\_API int QueryMeetingExist(int meetingId);

/\*

### \* 取得当前会议中参会人个数

\* 参数 返回值：参会人个数

\*/

MEETINGSDKAGENT\_API int GetParticipantsCount(int \*count);

/\*

### \* 取得当前会议中参会人列表

\* 参数： pUsers 保存参会人信息指针

\* pGetNum 实际取到的个数

\* maxCount 最多可以取多少个

\*/

MEETINGSDKAGENT\_API int GetParticipants(ContactInfo \*pUsers,int \*pGetNum,int maxCount);

### 1.6 参数设置相关接口

~~/\*~~

~~\* 设置服务器地址 (同步) （废弃接口，所有的产品都不用这个接口了）~~

~~\* 参数： pIp 服务器ip地址~~

~~\*/~~

~~MEETINGSDKAGENT\_API int SetServerAddr(char \*pIp);~~

/\*

### \* 取得设备列表

 \*    参数：devType    设备类型  1：摄像头采集源  2：文档视频采集源

 \*                 3：音频采集源  4：音频播放设备

 \*          pDevList   存放设备列表的指针

 \*          maxCount   设备列表最多接收多少个设备信息

 \*          pGetCount  实际取到的设备数量

 \*/

MEETINGSDKAGENT\_API int GetDeviceList(int devType,DeviceInfo \*pDevList,int maxCount,int \*pGetCount);

/\*

### \* 设置默认的音视频采集渲染设备

\* 参数：devType 设备类型 1：摄像头 2：文档采集源 3：音频采集源 4：音频播放设备 5：文档音频采集设备 6：主摄像头 7：辅摄像头

\* devName 选择的设备名称

\*/

MEETINGSDKAGENT\_API int SetDefaultDevice(int devType,char \*devName);

/\*

### \* 设置视频采集分辨率

\* 参数：devType 设备类型 1：摄像头 2：文档采集源

\*/

MEETINGSDKAGENT\_API int SetVideoCapResolution(int devType,int width,int height);

/\*

### \* 设置视频采集编码码率

\* 参数：devType 设备类型 1：摄像头 2：文档采集源

\*/

MEETINGSDKAGENT\_API int SetVideoCapBitRate(int devType,int bitrate);

/\*

### \* 设置音频采样率

\*/

MEETINGSDKAGENT\_API int SetAudioCapSampleRate(int samplerate);

/\*

### \* 设置音频采集码率

\*/

MEETINGSDKAGENT\_API int SetAudioCapBitRate(int bitrate);

/\*

### \* 设置音频参数

\*/

MEETINGSDKAGENT\_API int SetAudioParam(char \*micDevName,char \*renderDevName,int sampleRate);

/\*

### \* 设设置推流参数

\*/

MEETINGSDKAGENT\_API int SetBroadcastParam(int videoWidth,int videoHeight,int videoBitrate);

/\*

### \* 设置会议视频采集参数

\*/

MEETINGSDKAGENT\_API int SetVideoCapParam(int videoWidth,int videoHeight,int videoBitrate);

/\*

### \* 设置主摄像头和辅摄像头

\*/

MEETINGSDKAGENT\_API int SetDefVideoDevices(char\* cam1Name,char\* cam2Name);

/\*

### \* 设置本地录制文件存放路径

\*/

MEETINGSDKAGENT\_API int SetRecordDirectory(char\* dir);

### 1.7 调试功能相关接口

/\*

### \* 显示QOS

\*/

MEETINGSDKAGENT\_API int ShowQosView();

/\*

### \* 关闭QOS

\*/

MEETINGSDKAGENT\_API int CloseQosView();

/\*

### \* 显示摄像头属性窗口

\*/

MEETINGSDKAGENT\_API int ShowCameraProtityPage(char\* camName);

### 1.8 同步课堂版本sdk使用的接口，活动直播版本不再使用

/\*

\* 修改本地显示的名称

\* 参数：name 显示名称 hwnd 窗口句柄

\*/

MEETINGSDKAGENT\_API int SetViewDisplayName(void \*hwnd,char \*name);

/\*

### \* 打开课件 (同步课堂版使用的采集卡课件)

\*/

MEETINGSDKAGENT\_API int StartShareDoc();

/\*

### \* 关闭课件 (同步课堂版使用的采集卡课件)

\*/

MEETINGSDKAGENT\_API int StopShareDoc();

/\*

### \* 课件是否打开 (同步)

\* 返回值：0 没有打开 非0 已经打开

\*/

MEETINGSDKAGENT\_API int IsShareDocOpened();

/\*

### \* 取得视频窗口的显示位置 (同步)

\*/

MEETINGSDKAGENT\_API int GetViewPosition(void\* hwnd,ViewRect \*viewPos);

/\*

### \* 设置视频窗口的显示位置 (同步)

\*/

MEETINGSDKAGENT\_API int SetViewPosition(void\* hwnd,ViewRect viewPos);

/\*

### \* 设置视频窗口的可见性 (同步)

\* 参数：hwnd 窗口句柄

\* visible 0 隐藏窗口 非0 显示窗口

\*/

MEETINGSDKAGENT\_API int GetViewVisile(void\* hwnd,int visible);

/\*

### \* 判断视频窗口是否可见 (同步)

\* 参数：hwnd 窗口句柄

\* 返回值： 0 窗口隐藏 非0 窗口可见

\*/

MEETINGSDKAGENT\_API int IsViewVisile(void\* hwnd);

/\*

\* 将视频窗口提到最前面 （上交所 同步课堂抢焦点版本使用）

\*/

MEETINGSDKAGENT\_API int BringViewPanelFront(void\* hwnd);

## 回调接口定义

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*回调类型定义\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

enum ENUM\_CALLBACK\_CMD\_TYPE

{

/\*异步调用结果\*/

/\*异步调用结果\*/

enum\_startup\_result = 100, /\*启动结果 回调参数 StartResult \*/

enum\_createMeeting\_result, /\*创建会议结果 回调参数 CreateMeetingResult\*/

enum\_invite\_result, /\*邀请参会结果 回调参数 AsynCallResult \*/

enum\_joinmeeting\_result, /\*进入会议结果 回调参数 JoinMeetingResult \*/

enum\_applyspeak\_result, /\*申请发言结果 回调参数 AsynCallResult \*/

enum\_exitmeeting\_result, /\*退出会议结果 回调参数 AsynCallResult \*/

enum\_getmeetinglist\_result, /\*取得会议列表结果回调参数 GetMeetingListResult\*/

enum\_queryname\_result, /\*查询名称结果 回调参数 QueryNameResult \*/

enum\_modify\_name\_result, /\*修改显示名称结果回调参数 AsynCallResult \*/

enum\_senduimessage\_result, /\*发送消息结果 回调参数 AsynCallResult \*/

enum\_verifymeetexist\_result, /\*查询会议是否存在结果 回调参数 AsynCallResult \*/

enum\_open\_doc\_result, /\*打开课件结果 回调参数 AsynCallResult \*/

enum\_close\_doc\_result, /\*关闭课件结果 回调参数 AsynCallResult \*/

enum\_start\_broadcast\_result, /\*开始推流结果 回调参数 AsynCallResult \*/

enum\_stop\_broadcast\_result, /\*停止推流结果 回调参数 AsynCallResult \*/

enum\_create\_datemeeting\_result, /\*创建预约会议结果回调参数 CreateMeetingResult\*/

enum\_req\_user\_speak\_result, /\*请求用户发言结果 回调参数 AsynCallResult \*/

enum\_req\_user\_stopspeak\_result, /\*请用户停止发言结果 回调参数 AsynCallResult \*/

enum\_start\_record\_result, /\*开始录制结果 回调参数 AsynCallResult \*/

enum\_stop\_record\_result, /\*停止录制结果 回调参数 AsynCallResult \*/

enum\_open\_video\_result, /\*打开摄像头结果 回调参数 AsynCallResult \*/

enum\_close\_video\_result, /\*关闭摄像头结果 回调参数 AsynCallResult \*/

enum\_set\_mute\_state\_result, /\*设置麦克风静音结果 回调参数 AsynCallResult \*/

enum\_start\_screensharing\_result,/\*开始屏幕分享结果 回调参数 AsynCallResult \*/

enum\_stop\_screensharing\_result, /\*停止桌面分享结果 回调参数 AsynCallResult \*/

/\*通知消息\*/

enum\_recive\_invitation = 200, /\*收到参会邀请通知 回调参数 MeetingInvitation \*/

enum\_view\_created, /\*新视频窗口创建通知 回调参数 SpeakerView \*/

enum\_view\_closed, /\*视频窗口关闭通知 回调参数 被关闭窗口句柄 \*/

enum\_startspeak, /\*开始发言通知 无回调参数 \*/

enum\_stopspeak, /\*停止发言通知 无回调参数 \*/

enum\_other\_startspeak, /\*其他人开始发言通知 回调参数 ContactInfo（发言人信息）\*/

enum\_other\_stopspeak, /\*其他人停止发言通知 回调参数 ContactInfo（发言人信息）\*/

enum\_other\_joinmeeting,/\*其他人进入会议通知 回调参数 ContactInfo（进入者信息）\*/

enum\_other\_exitmeeting,/\*其他人退出会议通知 回调参数 ContactInfo（退出者信息）\*/

enum\_other\_message, /\*收到来自其他参会者的消息 回调参数 UIMessage \*/

enum\_user\_closestream, /\*其户关闭音视频流 回调参数 StreamInfo \*/

enum\_user\_openstream, /\*其户打开音视频流 回调参数 StreamInfo \*/

/\*通用的底层回调的错误提示信息\*/

enum\_error\_message = 300, /\*底层错误提示信息 回调参数 AsynCallResult \*/

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*回调函数定义\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

typedef void(\*PFunc\_CallBack)(int cmdId,void\* pData,int dataLen,long long ctx);

## 结构体定义

### 联系人信息

typedef struct \_tagContactInfo

{

char m\_szPhoneId[16]; /\*视讯号\*/

char m\_szDisplayName[32]; /\*名称\*/

}ContactInfo;

### 窗口位置信息

typedef struct \_tagRect

{

int m\_left;

int m\_right;

int m\_top;

int m\_bottom;

}ViewRect;

### 设备信息

typedef struct \_tagDeviceInfo

{

int m\_isDefault; /\*是否是默认设备\*/

char m\_szDevName[128]; /\*设备名称\*/

}DeviceInfo;

### 参会者信息结构体定义

typedef struct \_tagParticipantInfo

{

ContactInfo m\_contactInfo; /\*参会者视讯号 和 名称\*/

int m\_bIsSpeaking; /\*是否正在发言 0 否 非0 是\*/

}ParticipantInfo;

### 当前会议信息结构体定义

typedef struct \_tagCurMeetingInfo

{

char m\_szTeacherPhoneId[16]; /\*老师的视讯号\*/

ParticipantInfo\* m\_pParticipantsInfo; /\*参会人列表指针\*/

int m\_iParticipantsCount; /\*参会人列表中成员的个数\*/

}CurMeetingInfo;

### 异步调用结果

typedef struct \_tagAsynCallResult

{

int m\_rc; /\*错误码 0：成功 非0：失败\*/

char m\_message[128]; /\*错误描述信息\*/

}AsynCallResult;

### 启动结果

typedef struct \_tagStartResult

{

AsynCallResult m\_result; /\*异步调用结果\*/

ContactInfo m\_selfInfo; /\*自己的视讯号和名称，异步调用成功时有效\*/

}StartResult;

### 创建会议结果

typedef struct \_tagCreateMeetingResult

{

AsynCallResult m\_result; /\*异步调用结果\*/

int m\_meetingId; /\*创建的会议Id\*/

}CreateMeetingResult;

### 加入会议结果

typedef struct \_tagJoinMeetingResult

{

AsynCallResult m\_result; /\*异步调用结果\*/

CurMeetingInfo m\_meetingInfo; /\*会议信息，调用成功时有效\*/

}JoinMeetingResult;

### 邀请参会通知

typedef struct \_tagMeetingInvitation

{

ContactInfo m\_invitor; /\*邀请者信息\*/

int m\_meetingId; /\*邀请参加的会议Id\*/

}MeetingInvitation;

### 发言者视频窗口信息

typedef struct \_tagSpeakerView

{

ContactInfo m\_speaker; /\*发言者信息\*/

ViewRect m\_viewPosition; /\*窗口默认位置\*/

void \* m\_viewHwnd; /\*窗口句柄\*/

int m\_viewType; /\*视图类型 1：摄像头 2：课件 \*/

}SpeakerView;

### 参会者间的透传消息

typedef struct \_tagUIMessage

{

ContactInfo m\_sender; /\*消息发送者\*/

int m\_messageId; /\*消息Id\*/

char m\_szData[160]; /\*消息携带数据\*/

}UIMessage;

### 当前可参加的会议列表

typedef struct \_tagMeetingInfo

{

int m\_meetingId; /\*会议Id\*/

int m\_meetingType; /\*会议类型 1即时会议 2 预约会议\*/

char m\_szCreatorId[16]; /\*创建者视讯号\*/

char m\_szCreatorName[32]; /\*创建者名称\*/

char m\_szStartTime[24]; /\*会议开始时间\*/

}MeetingInfo;

typedef struct \_tagMeetingList

{

MeetingInfo \* m\_pMeetings; /\*会议信息指针\*/

int m\_count; /\*会议数\*/

}MeetingList;

/\*用户打开关闭流通知\*/

typedef struct \_tagStreamInfo

{

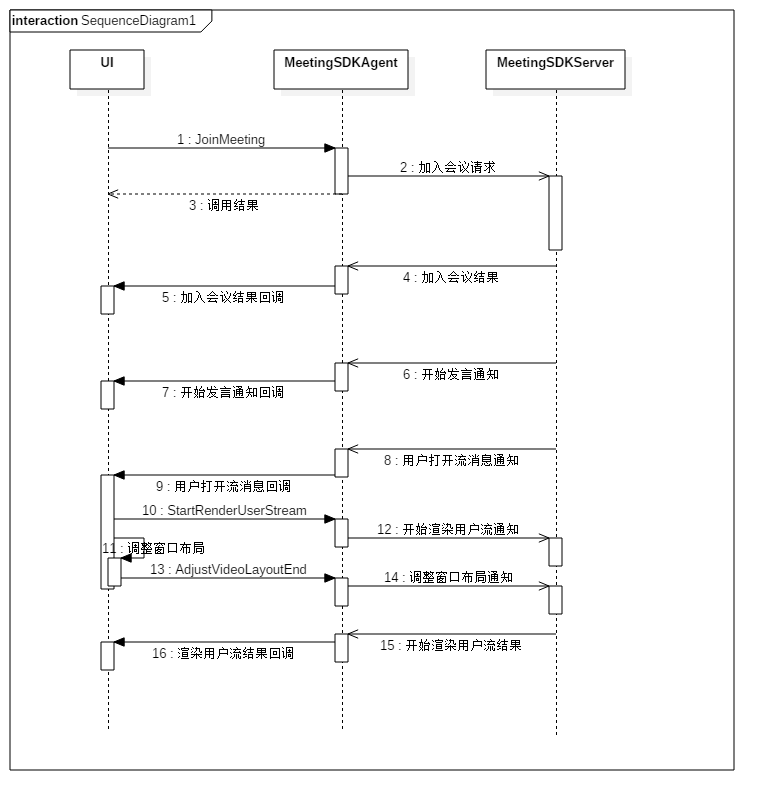
char m\_szPhoneId[16]; /\*视讯号\*/

int m\_type; /\*流类型 第0位标识摄像头 第1位标识麦克风 第2位标识文档视频 第4位标识文档音频（暂时没有文档音频需求） \*/

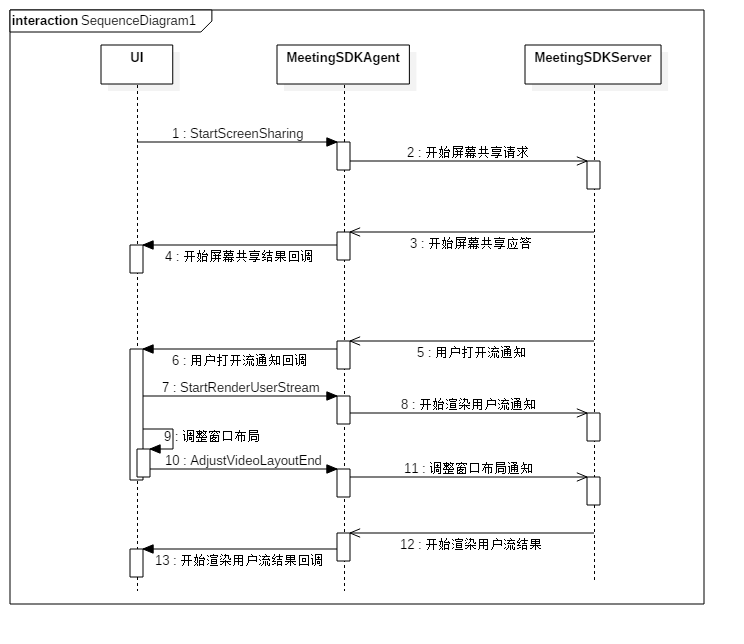
}StreamInfo;

## 相关功能时序

### 加入会议时序



### 开始屏幕分享时序



### 停止屏幕分享时序

