VideoCall SDK(V3.2.0)

**开发手册**

深圳市云屋科技有限公司

Cloudroom Technology Co., Ltd.

[http://www.cloudroom.com](http://www.cloudroom.com/)

目录

目录

[1 系统概述 7](#_Toc32666)

[1.1 系统介绍 7](#_Toc19496)

[1.2 系统特性 7](#_Toc10634)

[1.2.1 视频技术 7](#_Toc29427)

[1.2.2 音频技术 8](#_Toc12796)

[1.2.3 网络适应技术 8](#_Toc9034)

[1.3 关于云屋科技 8](#_Toc916)

[1.4 技术支持 8](#_Toc18376)

[1.5 版权申明 9](#_Toc1764)

[2 SDK使用指南 9](#_Toc13714)

[3 SDK常量及数据结构定义 11](#_Toc30707)

[3.1 常量定义 12](#_Toc20033)

[3.1.1 错误码定义（CRVIDEOSDK\_ERR\_DEF） 12](#_Toc6388)

[3.1.2 麦克风状态（ASTATUS） 13](#_Toc7730)

[3.1.3 视频尺寸定义（VIDEO\_SHOW\_SIZE） 13](#_Toc13284)

[3.1.4 视频状态定义（VSTATUS） 13](#_Toc16955)

[3.1.5 视频图像格式定义（VIDEO\_FORMAT） 13](#_Toc10724)

[3.1.6 影音播放状态定义（MEDIA\_STATE） 14](#_Toc30736)

[3.1.7 影音停止播放原因定义（MEDIA\_STOP\_REASON） 14](#_Toc1669)

[3.2 结构体定义 14](#_Toc5864)

[3.2.1 Sdk初始化数据（SdkInitDat） 14](#_Toc32361)

[3.2.2 Sdk登录数据（LoginDat） 14](#_Toc23065)

[3.2.3 音频配置（AudioCfg） 15](#_Toc15325)

[3.2.4 视频配置（VideoCfg） 15](#_Toc30150)

[3.2.5 视频设备信息 15](#_Toc12636)

[3.2.6 一帧图像数据（VideoFrame） 15](#_Toc28120)

[3.2.7 屏幕共享图像数据（ScreenShareImg） 16](#_Toc10525)

[3.2.8 影音播放图像数据（MediaDataFrame） 16](#_Toc10706)

[3.2.9 队列信息（QueueInfo） 16](#_Toc16920)

[3.2.10 队列状态（QueueStatus） 17](#_Toc21575)

[3.2.11 排队等待信息（QueuingInfo） 17](#_Toc20005)

[3.2.12 视频会话信息信息（VideoSessionInfo） 17](#_Toc19193)

[3.2.13 客户信息（UserInfo） 17](#_Toc25124)

[3.2.14 会议成员信息（MemberInfo） 18](#_Toc19685)

[3.2.15 媒体信息（MediaInfo） 18](#_Toc12362)

[3.3 CloudroomVideoSDK类接口 18](#_Toc15823)

[3.3.1 GetCloudroomVideoSDKVer 18](#_Toc17938)

[3.3.2 init: 19](#_Toc21256)

[3.3.3 isInitSuccess 19](#_Toc21079)

[3.3.4 Uninit 19](#_Toc23427)

[3.3.5 setLogOpen: 19](#_Toc32166)

[3.3.6 setServerAddr: 20](#_Toc9589)

[3.3.7 serverAddr 20](#_Toc28113)

[3.4 CloudroomQueue类接口 20](#_Toc22035)

[3.4.1 setQueueCallback: 20](#_Toc5947)

[3.4.2 initQueueDat:cookie: 21](#_Toc27694)

[3.4.3 refreshAllQueueStatus 21](#_Toc20864)

[3.4.4 getQueueInfo 21](#_Toc11452)

[3.4.5 getQueueStatus: 22](#_Toc4707)

[3.4.6 getQueuingInfo 22](#_Toc28811)

[3.4.7 getServiceQueues 22](#_Toc2346)

[3.4.8 getSessionInfo 22](#_Toc17589)

[3.4.9 startQueuing:cookie: 23](#_Toc26062)

[3.4.10 stopQueuing: 23](#_Toc7052)

[3.4.11 startService:cookie: 23](#_Toc12962)

[3.4.12 stopService:cookie: 24](#_Toc31260)

[3.4.13 acceptAssignUser:userID:cookie: 24](#_Toc7277)

[3.4.14 rejectAssignUser:userID:cookie: 24](#_Toc11886)

[3.4.15 reqAssignUser: 25](#_Toc9218)

[3.5 CloudroomQueueCallBack协议接口 25](#_Toc15665)

[3.5.1 initQueueDatRslt:cookie: 25](#_Toc19926)

[3.5.2 queueStatusChanged: 25](#_Toc27004)

[3.5.3 queuingInfoChanged: 26](#_Toc18228)

[3.5.4 startQueuingRslt:cookie: 26](#_Toc4034)

[3.5.5 stopQueuingRslt:cookie: 26](#_Toc30187)

[3.5.6 startServiceRslt:errCode:cookie: 27](#_Toc5416)

[3.5.7 stopServiceRslt:errCode:cookie: 27](#_Toc20151)

[3.5.8 responseAssignUserRslt:cookie: 27](#_Toc30438)

[3.5.9 autoAssignUser: 27](#_Toc28598)

[3.5.10 reqAssignUserRslt:userInfo:cookie: 28](#_Toc8538)

[3.5.11 cancelAssignUser:usrID: 28](#_Toc10143)

[3.6 CloudroomVideoMgr类接口 28](#_Toc13550)

[3.6.1 setMgrCallback: 28](#_Toc6040)

[3.6.2 login:/login:cookie: 29](#_Toc20232)

[3.6.3 logout 29](#_Toc19678)

[3.6.4 setDNDStatus:/setDNDStatus:cookie: 30](#_Toc24473)

[3.6.5 createMeeting:createPswd:cookie: 30](#_Toc22691)

[3.6.6 call:meetInfo:param:cookie: 31](#_Toc6151)

[3.6.7 acceptCall: meetInfo:/acceptCall: meetInfo:cookie: 31](#_Toc24515)

[3.6.8 rejectCall:/rejectCall:cookie: 32](#_Toc9863)

[3.6.9 hungupCall:/hungupCall:cookie: 33](#_Toc10564)

[3.7 CloudroomVideoMgrCallBack协议接口 33](#_Toc18721)

[3.7.1 loginSuccess:cookie: 33](#_Toc10688)

[3.7.2 loginFail:cookie: 33](#_Toc24048)

[3.7.3 lineOff: 34](#_Toc15141)

[3.7.4 setDNDStatusSuccess: 34](#_Toc615)

[3.7.5 setDNDStatusFail:cookie: 34](#_Toc27504)

[3.7.6 createMeetingSuccess:password:cookie: 34](#_Toc8598)

[3.7.7 createMeetingFail:cookie: 35](#_Toc25169)

[3.7.8 callSuccess: 35](#_Toc4552)

[3.7.9 callFail: 35](#_Toc7384)

[3.7.10 acceptCallSuccess:cookie: 35](#_Toc9208)

[3.7.11 acceptCallFail:errCode: 36](#_Toc30618)

[3.7.12 rejectCallSuccess: 36](#_Toc23523)

[3.7.13 rejectCallFail:errCode: 36](#_Toc750)

[3.7.14 hangupCallSuccess:cookie: 37](#_Toc32201)

[3.7.15 hangupCallFail:errCode: 37](#_Toc14904)

[3.7.16 notifyCallIn: 37](#_Toc18202)

[3.7.17 notifyCallAccepted:meetInfo:meetPswd: 37](#_Toc15959)

[3.7.18 notifyCallRejected:reason: 38](#_Toc4512)

[3.7.19 notifyCallHungup: 38](#_Toc29874)

[3.8 CloudroomVideoMeeting类接口 38](#_Toc23784)

[3.8.1 setMeetingCallBack: 38](#_Toc14197)

[3.8.2 enterMeeting:pswd:userID:nikeName: 38](#_Toc22513)

[3.8.3 isUserInMeeting: 39](#_Toc26882)

[3.8.4 stopMeeting 39](#_Toc5492)

[3.8.5 exitMeeting 39](#_Toc2977)

[3.8.6 getNickName: 40](#_Toc10044)

[3.8.7 getAllMembers 40](#_Toc18420)

[3.8.8 getMemberInfo: 40](#_Toc358)

[3.8.9 setAllAudioClose 41](#_Toc24501)

[3.8.10 setMicVolume: 41](#_Toc19588)

[3.8.11 getMicVolume 41](#_Toc14287)

[3.8.12 setSpeakerVolume: 42](#_Toc17913)

[3.8.13 getSpeakerVolume 42](#_Toc15310)

[3.8.14 getMicEnergy: 42](#_Toc2051)

[3.8.15 openMic: 42](#_Toc11831)

[3.8.16 closeMic: 43](#_Toc8017)

[3.8.17 getAudioStatus: 43](#_Toc21584)

[3.8.18 getSpeakerOut 43](#_Toc16924)

[3.8.19 setSpeakerOut: 44](#_Toc8530)

[3.8.20 setSpeakerMute: 44](#_Toc23824)

[3.8.21 getSpeakerMute 44](#_Toc25642)

[3.8.22 setVideoCfg: 44](#_Toc3217)

[3.8.23 getVideoCfg 45](#_Toc21546)

[3.8.24 getVideoStatus: 45](#_Toc3364)

[3.8.25 openVideo: 45](#_Toc31835)

[3.8.26 closeVideo: 46](#_Toc17398)

[3.8.27 watchVideos: 46](#_Toc16978)

[3.8.28 setVideoFormat: 46](#_Toc18658)

[3.8.29 getVideoImg: 46](#_Toc27829)

[3.8.30 getAllVideoInfo: 47](#_Toc25958)

[3.8.31 setDefaultVideo:videoID: 47](#_Toc28365)

[3.8.32 getDefaultVideo: 47](#_Toc22617)

[3.8.33 getWatchableVideos: 48](#_Toc10720)

[3.8.34 isScreenShareStarted 48](#_Toc27764)

[3.8.35 getScreenSharer 48](#_Toc6699)

[3.8.36 getShareScreenDecodeImg 49](#_Toc8160)

[3.9 CloudroomMeetingSDKCallBack(协议)接口 49](#_Toc21667)

[3.9.1 enterMeetingRslt: 49](#_Toc19812)

[3.9.2 userEnterMeeting: 49](#_Toc9205)

[3.9.3 userLeftMeeting: 50](#_Toc3058)

[3.9.4 endMeetingRslt: 50](#_Toc18015)

[3.9.5 meetingStopped 50](#_Toc24476)

[3.9.6 meetingDropped 50](#_Toc6987)

[3.9.7 netStateChanged: 50](#_Toc28476)

[3.9.8 micEnergyUpdate:oldLevel:newLevel: 51](#_Toc18140)

[3.9.9 audioDevChanged 51](#_Toc19541)

[3.9.10 audioStatusChanged:oldStatus:newStatus: 51](#_Toc17100)

[3.9.11 videoStatusChanged:oldStatus:newStatus: 51](#_Toc1779)

[3.9.12 openVideoRslt: 52](#_Toc20296)

[3.9.13 notifyVideoData:frameTime: 52](#_Toc21910)

[3.9.14 videoDevChanged: 52](#_Toc2998)

[3.9.15 defVideoChanged:videoID: 52](#_Toc6507)

[3.9.16 notifyScreenShareStarted 53](#_Toc32381)

[3.9.17 notifyScreenShareStopped 53](#_Toc2403)

[3.9.18 notifyScreenShareData:changedRect: 53](#_Toc19534)

[3.9.19 notifyMediaStart: 53](#_Toc25318)

[3.9.20 notifyMediaStop: reason: 54](#_Toc8835)

[3.9.21 notifyMediaPause: bPause: 54](#_Toc13618)

[3.9.22 notifyMemberMediaData: 54](#_Toc13699)

# 系统概述

首先非常感谢您选择深圳市云屋科技视频呼叫组件（以下简称VideoCall SDK）产品， 我们将热诚为您提供最好的服务。

本手册可能存在排版或描述不准确的地方，我们将会即时更新到官方网站。我们的产品在更新升级时，接口可能会有所变化，所以请使用产品对应版本的手册。

## 系统介绍

VideoCall SDK是基于云屋科技的网络音视频会议为有点对点音视频需求的客户量身订制的一套二次开发接口。它集成了云屋科技多年积累、创新而形成的先进的技术：高保真的宽带音频、出色的视频编解码技术，先进的网络丢包补偿、智能流控处理，分布式超大容量的云架构技术，让您能在互联网上体验流原汁原味的声音、流畅逼真的视频效果。

同时，VideoCall更是扩展了呼叫相关接口、实现了一套呼叫排队接口（您也可以根据实际需求不使用这套排队接口而自已实现）,大大减化了您开发应用的复杂度。

VideoCall分为客户端SDK、服务端配置WEB、服务端SDK三部份，其中客户端 SDK

用于实现音视频、呼叫、排队以及其它客户端相关的功能；服务端配置WEB用于配置排队数据（自已实现队列时，可以不用配置）； 而服务端 SDK 主要实现业务层逻辑控制，以及与第三方平台的互联。

以下内容如无特别指明，均指客户端SDK。

VideoCall SDK支持的平台有：Windows、Android、iOS、Mac

## 系统特性

VideoCall SDK是一套跨平台的时实通讯解决方案，支持当前主流的操作系统：Windows、Android、iOS、Mac。

视频编解码我们使用了先进的 H.264 High Profile，再加上云屋智能的流控、防马赛克处理和动态码率算法，使视频具有低带宽占用高清晰度的特点。

语音我们采用了AMR-WB高保真宽频语音编码，辨识度很高，最大程度保留了发言者的音色，再加上云屋智能丢包补偿算法，在丢包40%的情况下仍然可以保障语音的正常沟通；

整个系统采用负荷分担的云架构，使其在可靠性、可扩展性上具有了先天的优势，也为用户在自建系统时提供了更多更灵活的选择。

### 视频技术

视频制式：PAL；

支持设备：USB摄像头、视频采集卡、虚拟摄像头；

分辨率：144\*80~1920\*1080；

帧率：5~30；

视频编码算法：H.264

视频码率：10kbps~2mbps (VBR)

### 音频技术

支持声卡采样率：16000 Hz、22050Hz、44100Hz、48000Hz

采样量化：16 bit

音频编码器：AMR\_WB

音频流码率：6kbps~23.85kbps

音效处理：回音消除（AEC） 、噪音抑制（NS） 、自动增益控制（AGC） 、静音检测（VAD）

### 网络适应技术

使用网络抖动缓冲技术

使用丢包补偿技术

在网适波动时，动态帧率动态码率，平衡图像的质量和流量性；

## 关于云屋科技

深圳市云屋科技有限公司是一家专业从事远程多媒体交互和互联网多方通信技术服务的高科技企业公司专注于全球互联网商务会议，协同商务办公、在线培训、在线视讯课堂等领域，目前的产品主要有电话会议、网络视频会议、云视真（TV版）视频会议终端、万店通多媒体交互门店管理系统等。

公司总部位于有“中国硅谷”之称的深圳南山科技园，办公面积1000平米。公司拥有和引进了多项世界尖端通讯技术，致力于为企业客户持续降低沟通成本，提高效率，提升企业竞争力。公司领先的经营理念、广泛的合作策略、出色的研发技术等赢得了众多企业和机构的信赖。

## 技术支持

在您使用SDK 的过程中，遇到任何困难，请与我们联系，我们将热忱为您提供帮助。

您可以通过如下方式与我们取得联系：

1. 公司官网：http://www.cloudroom.com
2. 电子邮件：[support@cloudroom.com](mailto:support@cloudroom.com)
3. 服务热线：400-863-0755
4. 传真号码：0755-21677530

## 版权申明

VideoCall SDK是深圳市云屋科技开发，拥有自主知识产权的产品(登记号：2014SR049731)（深圳市云屋科技增值电信业务经营许可证：粤B2-20130567 粤ICP备12041260号）

本产品禁止反编译分析， 在未经深圳市云屋科技事先书面授权下，不得以任何形式或借助任何手段复制本产品的任何部分。

随同SDK包一同发布的Demo程序源代码版权归深圳市云屋科技公司所有。

使用到的第三方开源组件有：FFmpeg（LGPL）、WebRTC（BSD）、Qt(LGPL) 。

# SDK使用指南

VideoCall SDK是由众多的.a静态库组合而成的，其中对外接口都集中在libCloudroomVideoSDK\_IOS.a、libMeetingMgrSDK.a、libMeetingSDK.a上。

VideoCall SDK中队列相关接口是可选项，您可以使用它也可以自已另外实现，这并不影响视频呼叫、音视频通话功能。

使用VideoCall SDK有一系列的交互过程，主要包括：

1. 初始化SDK；
2. 登录；
3. 队列获取，客户排队/座席服务；
4. 呼叫、应答
5. 通话中视频图像呈现；（语音将由SDK内部完成采集播放， 视频将由SDK采集编码解码，图像将以RGB/YUV格式回调给用户）

以下是各阶段涉及函数接口：

|  |  |  |
| --- | --- | --- |
| 阶段 | 函数名 | 功能 |
| 初始化 | init: | 初始化 |
| uninit | 反初始化 |
| 登录 | setServerAddr:: | 设置服务器地址 |
| serverAddr | 获取设置的服务器地址 |
| login:cookie: | 登录 |
| logout | 登出 |
| setDNDStatus:cookie: | 设置免打扰 |
| 呼叫 | call:meetID:passwordparam:cookie | 呼叫 |
| acceptCall:cookie: | 接收呼叫 |
| rejectCall:cookie: | 拒绝呼中 |
| hungupCall:cookie: | 挂断 |
| 队列 | setQueueCallback: | 设置回调 |
| initQueueDat:cookie: | 初始化队列数据 |
| refreshAllQueueStatus | 刷新所有队列状态信息 |
| getQueueInfo | 获取队列信息 |
| getQueueStatus: | 获取队列状态信息 |
| getQueuingInfo | 获取我的排队信息(客户专用接口） |
| getServiceQueues | 获取我服务的所有队列（座席专用接口） |
| getSessionInfo | 获取我的会话信息 |
| startQueuing:cookie: | 开始排队(客户专用接口） |
| stopQueuing: | 停止排队(客户专用接口) |
| startService:cookie: | 开始服务某个队列（座席专用接口） |
| stopService:cookie: | 停止服务某个队列（座席专用接口） |
| acceptAssignUser:userID:cookie: | 接受系统自动安排的客户（座席专用接口） |
| rejectAssignUser:userID:cookie: | 拒绝系统自动安排的客户（座席专用接口） |
| reqAssignUser: | 请求分配一个客户（座席专用接口，免打扰模式下手工接口） |
| 通话 | setMeetingCallBack: | 设置回调 |
| enterMeeting:pswd: userID: nikeName: | 进入会议 |
| isUserInMeeting: | 该用户是否在会话中 |
| stopMeeting | 结束会议 |
| exitMeeting | 离开会议 |
| getNickName: | 获取昵称 |
| getAllMembers | 获取所有成员信息列表 |
| getMemberInfo: | 获取指定成员信息 |
| setAllAudioClose | 关闭所有音频设备 |
| setMicVolume: | 设置本地麦音量 |
| getMicVolume | 获取本地麦音量 |
| getSpeakerVolume | 获取本地喇叭音量 |
| setSpeakerVolume: | 设置本地喇叭音量 |
| getMicEnergy: | 获取自已或对端的说话声音大小 |
| openMic: | 打开麦克风 |
| closeMic: | 关闭麦克风 |
| getAudioStatus: | 获取我或对方的麦克风状态 |
| getSpeakerOut | 获取外放状态 |
| setSpeakerOut: | 设置外放状态 |
| getSpeakerMute | 查询扬声器是否静音 |
| setSpeakerMute: | 设置扬声器是否静音 |
| setVideoCfg: | 设置视频信息 |
| getVideoCfg | 获取视频信息 |
| getVideoStatus: | 得到我或对方视频的开关状态 |
| openVideo: | 打开摄像头 |
| closeVideo: | 关闭摄像头 |
| watchVideos: | 设置需要观看视频的所有者 |
| setVideoFormat: | 设置输出的视频图像格式 |
| getVideoImg: | 获取自己或他人的图像数据 |
| getAllVideoInfo: | 获取摄像头设备信息 |
| setDefaultVideo: videoID:: | 设置当前摄像头设备信息 |
| getDefaultVideo: | 获取当前摄像头设备信息 |
| getWatchableVideos | 获取可观看的摄像头 |
| 屏幕共享 | isScreenShareStarted | 是否已开始屏幕共享 |
| getScreenSharer | 获取屏幕共享者 |
| getShareScreenDecodeImg | 此共享Image数据应该每次获取，外界不能缓存使用 |
| 影音播放 | getMediaInfo | 获取影音文件信息 |
| GetMediaImg: | 获取影音图像信息 |

# SDK常量及数据结构定义

## 常量定义

### 错误码定义（CRVIDEOSDK\_ERR\_DEF）

|  |  |  |
| --- | --- | --- |
| CRVIDEOSDK\_NOERR | 0 | 没有错误 |
| CRVIDEOSDK\_UNKNOWERR | 1 | 未知错误 |
| CRVIDEOSDK\_OUTOF\_MEM | 2 | 内存不足 |
| CRVIDEOSDK\_INNER\_ERR | 3 | sdk内部错误 |
| CRVIDEOSDK\_MISMATCHCLIENTVER | 4 | 不支持的sdk版本 |
| CRVIDEOSDK\_MEETPARAM\_ERR | 5 | 参数错误 |
| CRVIDEOSDK\_ERR\_DATA | 6 | 无效数据 |
| CRVIDEOSDK\_ANCTPSWD\_ERR | 7 | 帐号密码不正确 |
| CRVIDEOSDK\_SERVER\_EXCEPTION | 8 | 服务异常 |
| CRVIDEOSDK\_LOGINSTATE\_ERROR | 9 | 登录状态错误 |
| CRVIDEOSDK\_USER\_BEEN\_KICKOUT | 10 | 被挤下线（帐号在别处登录） |
| CRVIDEOSDK\_NETWORK\_INITFAILED | 200 | 网络初始化失败 |
| CRVIDEOSDK\_NO\_SERVERINFO | 201 | 没有服务器信息 |
| CRVIDEOSDK\_NOSERVER\_RSP | 202 | 服务器没有响应 |
| CRVIDEOSDK\_CREATE\_CONN\_FAILED | 203 | 创建连接失败 |
| CRVIDEOSDK\_SOCKETEXCEPTION | 204 | socket异常 |
| CRVIDEOSDK\_SOCKETTIMEOUT | 205 | 网络超时 |
| CRVIDEOSDK\_FORCEDCLOSECONNECTION | 206 | 连接被关闭 |
| CRVIDEOSDK\_CONNECTIONLOST | 207 | 连接丢失 |
| CRVIDEOSDK\_QUE\_ID\_INVALID | 400 | 队列ID错误 |
| CRVIDEOSDK\_QUE\_NOUSER | 401 | 没有用户在排队 |
| CRVIDEOSDK\_QUE\_USER\_CANCELLED | 402 | 排队用户已取消 |
| CRVIDEOSDK\_QUE\_SERVICE\_NOT\_START | 403 | 队列服务还未开启 |
| CRVIDEOSDK\_ALREADY\_OTHERQUE | 404 | 已在其它队列排队(客户只能在一个队列排队) |
| CRVIDEOSDK\_INVALID\_CALLID | 600 | 无效的呼叫ID |
| CRVIDEOSDK\_ERR\_CALL\_EXIST | 601 | 已在呼叫中 |
| CRVIDEOSDK\_ERR\_BUSY | 602 | 对方忙 |
| CRVIDEOSDK\_ERR\_OFFLINE | 603 | 对方不在线 |
| CRVIDEOSDK\_ERR\_NOANSWER | 604 | 对方无应答 |
| CRVIDEOSDK\_ERR\_USER\_NOT\_FOUND | 605 | 用户不存在 |
| CRVIDEOSDK\_ERR\_REFUSE | 606 | 对方拒接 |
| CRVIDEOSDK\_MEETNOTEXIST | 800 | 会议不存在或已结束 |
| CRVIDEOSDK\_AUTHERROR | 801 | 会议密码不正确 |
| CRVIDEOSDK\_MEMBEROVERFLOWERROR | 802 | 会议终端数量已满（购买的license不够) |
| CRVIDEOSDK\_RESOURCEALLOCATEERROR | 803 | 分配会议资源失败 |
| CRVIDEOSDK\_CATCH\_SCREEN\_ERR | 900 | 抓屏失败 |
| CRVIDEOSDK\_RECORD\_MAX | 901 | 单次录制达到最大时长(8h) |
| CRVIDEOSDK\_RECORD\_NO\_DISK | 902 | 磁盘空间不够 |
| CRVIDEOSDK\_SENDFAIL | 1000 | 发送失败 |
| CRVIDEOSDK\_CONTAIN\_SENSITIVEWORDS | 1001 | 有敏感词语 |

### 麦克风状态（ASTATUS）

|  |  |  |
| --- | --- | --- |
| AUNKNOWN | 0 | 音频状态未知 |
| ANULL | 1 | 没有麦克风设备 |
| ACLOSE | 2 | 麦克风处于关闭状态（软开关） |
| AOPEN | 3 | 麦克风处于打开状态（软开关） |
| AOPENING | 4 | 向服务器发送打开消息中 |

### 视频尺寸定义（VIDEO\_SHOW\_SIZE）

|  |  |  |
| --- | --- | --- |
| VIDEO\_SZ\_80 | 0 | 144\*80, 默认码率：56kbps |
| VSIZE\_SZ\_128 | 1 | 224\*128, 默认码率：72kbps |
| VSIZE\_SZ\_160 | 2 | 288\*160, 默认码率：100kbps |
| VSIZE\_SZ\_192 | 3 | 336\*192, 默认码率：150kbps |
| VSIZE\_SZ\_256 | 4 | 448\*256, 默认码率：200kbps |
| VSIZE\_SZ\_288 | 5 | 512\*288, 默认码率：250kbps |
| VSIZE\_SZ\_320 | 6 | 576\*320, 缺省：300kbps |
| VSIZE\_SZ\_360 | 7 | 640\*360, 缺省：350kbps |
| VSIZE\_SZ\_400 | 8 | 720\*400, 缺省：420kbps |
| VSIZE\_SZ\_480 | 9 | 848\*480, 缺省：500kbps |
| VSIZE\_SZ\_576 | 10 | 1024\*576, 缺省：650kbps |
| VSIZE\_SZ\_720 | 11 | 1280\*720, 缺省：1mbps |
| VSIZE\_SZ\_1080 | 12 | 1920\*1080, 缺省：2mbps |

### 视频状态定义（VSTATUS）

|  |  |  |
| --- | --- | --- |
| VUNKNOWN | 0 | 视频状态未知 |
| VNULL | 1 | 没有视频设备 |
| VCLOSE | 2 | 视频处于关闭状态（软开关） |
| VOPEN | 3 | 视频处于打开状态（软开关） |
| VOPENING | 4 | 向服务器发送打开消息中 |

### 视频图像格式定义（VIDEO\_FORMAT）

|  |  |  |
| --- | --- | --- |
| VFMT\_YUV420P | 0 | planar YUV 4:2:0, 12bpp, (1 Cr & Cb sample per 2x2 Y samples) |
| VFMT\_ARGB32 | 1 | 32-bit ARGB format (0xAARRGGBB) |

### 影音播放状态定义（MEDIA\_STATE）

|  |  |  |
| --- | --- | --- |
| MEDIA\_START | 0 | 开始播放 |
| MEDIA\_PAUSE | 1 | 暂停播放 |
| MEDIA\_STOP | 2 | 停止播放 |

### 影音停止播放原因定义（MEDIA\_STOP\_REASON）

|  |  |  |
| --- | --- | --- |
| MEDIA\_CLOSE | 0 | 关闭 |
| MEDIA\_FINI | 1 | 播放完成 |
| MEDIA\_FILEOPEN\_ERR | 2 | 文件打开错误 |
| MEDIA\_FORMAT\_ERR | 3 | 格式错误 |
| MEDIA\_UNSUPPORT | 4 | 不支持 |
| MEDIA\_EXCEPTION | 5 | 异常 |

## 结构体定义

### Sdk初始化数据（SdkInitDat）

@interface SdkInitDat : NSObject

@property (nonatomic, strong) NSString \*oemID; //向云屋申请的开发商ID

@property (nonatomic, strong) NSString \*cfgPathFileName; //sdk内部使用的配置文件位置

@property (nonatomic, strong) NSString \*loggerPathFileName; //sdk内部使用的日志文件位置

@property (nonatomic, assign) BOOL showSDKLogConsole; //显示日志到控制台

@end

### Sdk登录数据（LoginDat）

@interface LoginDat : NSObject

//云屋帐号

@property (nonatomic, strong) NSString \*authAcnt; // 云屋鉴权帐号

@property (nonatomic, strong) NSString \*authPswd; // 云屋鉴权密码 MD5(密码)

@property (nonatomic, strong) NSString \*nickName; // 昵称

// 自定义帐号, 云屋服务器将会去配置的服务器进行认证；

//（当不使用自定义帐号时，acnt, pswd应为NULL）

@property (nonatomic, strong) NSString \*privAcnt; // 自定义帐号

@property (nonatomic, strong) NSString \*privAuthCode; // 自定义验证码(有复杂要求

@property (nonatomic, strong) NSString \*param; // 用户扩展信息

@end

### 音频配置（AudioCfg）

@interface AudioCfg : NSObject

@property (nonatomicz, strong) NSString \*micName; // 麦克风设备(空代表默认设备)

@property (nonatomic, strong) NSString \*speakerName; // 喇叭设备(空代表默认设备)

@property (nonatomic, assign) BOOL privEC; // 是否启用私有回声消除

@property (nonatomic, assign) BOOL privAgc; // 是否启用私有声音增益

@property (nonatomic, assign) BOOL HILevelEC; // 超强回声消息

@end

### 视频配置（VideoCfg）

@interface VideoCfg : NSObject

@property (nonatomic, assign) VIDEO\_SIZE\_TYPE sizeType; // 视频尺寸

@property (nonatomic, assign) int fps; // 视频帧率(5~30)

@property (nonatomic, assign) int maxbps; // 视频码率

@property (nonatomic, assign) int minQuality; // 质量范围\_最小

@property (nonatomic, assign) int maxQuality; // 质量范围\_最大(质量最差)

@end

### 视频设备信息

@interface UsrVideoId: NSObject

@property (nonatomic, copy) NSString \*userId;

@property (nonatomic, assign) short videoID;

@end

@interface UsrVideoInfo: UsrVideoId

@property (nonatomic, copy) NSString \*videoName;

@property (nonatomic, copy) NSString \*videoDevPath;

@end

@interface MediaInfo : NSObject

@property (nonatomic, copy) NSString \*userID;

@property (nonatomic, assign) MEDIA\_STATE state;

@property (nonatomic, copy) NSString \*mediaName;

@end

### 一帧图像数据（VideoFrame）

@interface VideoFrame : NSObject

{

@public void \*dat; // 图像数据

}

@property (nonatomic, assign) int camShowNo; //摄像头显示序号(0开始编号)

@property (nonatomic, assign) VIDEO\_FORMAT fmt; //图像格式

@property (nonatomic, assign) int datLength; // 图像大小

@property (nonatomic, assign) int frameWidth; // 图像大小

@property (nonatomic, assign) int frameHeight;

@property (nonatomic,assign) long frmTime;

@end

### 屏幕共享图像数据（ScreenShareImg）

@interface ScreenShareImg: NSObject

{

@public void \*rgbDat; // 图像数据

}

@property (nonatomic, assign) int datLength; // 图像大小

@property (nonatomic, assign) int rgbWidth; // 图像大小

@property (nonatomic, assign) int rgbHeight;

@end

### 影音播放图像数据（MediaDataFrame）

@interface MediaDataFrame : NSObject

{

@public void \*buf; // 图像数据

}

@property (nonatomic, assign) int datLength; // 图像大小

@property (nonatomic, assign) int fmt;

@property (nonatomic, assign) int w;

@property (nonatomic, assign) int h;

@property (nonatomic, assign) int64\_t pts;

@end

### 队列信息（QueueInfo）

@interface QueueInfo : NSObject

@property (nonatomic ,assign) int queID; // 队列ID

@property (nonatomic, strong) NSString \*name; // 队列名称

@property (nonatomic, strong) NSString \*desc; // 队列描述信息

@property (nonatomic, assign) int prio; // 队列优先级(约定：值越小优先级越高)

@end

### 队列状态（QueueStatus）

@interface QueueStatus : NSObject

@property (nonatomic, assign) int queID; // 队列ID

@property (nonatomic, assign) int agent\_num; // 坐席数

@property (nonatomic, assign) int wait\_num; // 等待人数

@property (nonatomic, assign) int srv\_num; // 正在服务人数

@end

### 排队等待信息（QueuingInfo）

@interface QueuingInfo : NSObject

@property (nonatomic, assign) int queID; // 我排的队列(-1:代表我没有排队；-2:代表我正在会话中,通过GetSessionInfo可获取相关信息)

@property (nonatomic, assign) int position; // 我的位置

@property (nonatomic, assign) int queuingTime; // 我排队的时长(单位s)

@end

### 视频会话信息信息（VideoSessionInfo）

@interface VideoSessionInfo : NSObject

@property (nonatomic, strong) NSString \*callID; // 我所在的会话(空字符串代表还未建立会话，以下数据都无效)

@property (nonatomic, copy) NSString \*peerID; //对方的id和昵称

@property (nonatomic, copy) NSString \*peerName;

@property (nonatomic, assign) BOOL bCallAccepted; // 被叫是否已接受呼叫(被叫掉线重登后，如果曾接受呼叫直接入会，如果未接受应该弹屏提示(也可做自动应答处理))

@property (nonatomic, assign) int meetingID; // 为会话分配的会议号

@property (nonatomic, strong) NSString \*meetingPswd; // 进入会议的密码

@property (nonatomic, assign) int duration; // 会话时长

@end

### 客户信息（UserInfo）

@interface UserInfo : NSObject

@property (nonatomic, assign) int queID; // 用户所在队列

@property (nonatomic, strong) NSString \*usrID; // 用户ID

@property (nonatomic, strong) NSString \*name; // 用户的昵称

@property (nonatomic, strong) NSString \*param; // 用户的扩展数据

@end

### 会议成员信息（MemberInfo）

@interface MemberInfo : NSObject

@property (nonatomic,copy) NSString\* userId; // 用户ID

@property (nonatomic,copy) NSString\* nickName; // 昵称

@property (nonatomic,assign) int audioStatus; // 音频状态

@property (nonatomic,assign) int videoStatus; // 视频状态

@end

### 媒体信息（MediaInfo）

@interface MemberInfo : NSObject

@property (nonatomic, copy) NSString\* userId; // 用户ID

@property (nonatomic, copy) NSString\* nickName; // 昵称

@property (nonatomic, assign) int audioStatus; // 音频状态

@property (nonatomic, assign) int videoStatus; // 视频状态

@end

函数详细说明

## CloudroomVideoSDK类接口

### GetCloudroomVideoSDKVer

|  |  |
| --- | --- |
| + (NSString \*)GetCloudroomVideoSDKVer | |
| 功能 | 获取SDK版本号 |
| 返回值 | SDK版本号 |
| 参数 | 无 |
| 说明 |  |

### init:

|  |  |
| --- | --- |
| - (CRVIDEOSDK\_ERR\_DEF)init:(SdkInitDat \*)dat | |
| 功能 | SDK初始化 |
| 返回值 | VCALLSDK\_NOERR为成功，否则是失败代码 |
| 参数 | dat 初始化数据，定义见类:SdkInitDat |
| 说明 |  |

### isInitSuccess

|  |  |
| --- | --- |
| * (BOOL)isInitSuccess | |
| 功能 | 初始化是否成功 |
| 返回值 | YES为成功，否则失败 |
| 参数 | 无 |
| 说明 |  |

### Uninit

|  |  |
| --- | --- |
| - (void)uninit | |
| 功能 | SDK反初始化 |
| 返回值 | 无 |
| 参数 | 无 |
| 说明 | 每次init后，都应对一次uninit |

### setLogOpen:

|  |  |
| --- | --- |
| * (void)setLogOpen:(BOOL)isOpen | |
| 功能 | 设置是否输出日志到日志文件 |
| 返回值 | 无 |
| 参数 | isOpen 是否打开 |
| 说明 |  |

### setServerAddr:

|  |  |
| --- | --- |
| * (void)setServerAddr:(NSString \*)serverList | |
| 功能 | 设置服务器地址 |
| 返回值 | 无 |
| 参数 | serverList 服务器地址 |
| 说明 | 支持单个服务器地址（如：www.cloudroom.com）或多个服务器地址串（如：www.cloudroom.com:8080;183.60.47.52:8080;） |

### serverAddr

|  |  |
| --- | --- |
| * (NSString \*)serverAddr | |
| 功能 | 获取设置的服务器地址 |
| 返回值 | 服务器地址（串） |
| 参数 | 无 |
| 说明 |  |

## CloudroomQueue类接口

### setQueueCallback:

|  |  |
| --- | --- |
| * (void)setQueueCallback:(id<CloudroomQueueCallBack>)callBack | |
| 功能 | 设置代理回调 |
| 返回值 | 无 |
| 参数 | 一切遵循协议: CloudroomQueueCallBack的对象 |
| 说明 |  |

### initQueueDat:cookie:

|  |  |
| --- | --- |
| - (void)initQueueDat:(NSString \*)cookie) | |
| 功能 | 初始化用户队列功能数据 |
| 返回值 | 无 |
| 参数 | cookie 用户自己数据 |
| 说明 | 操作回调initQueueDatRslt:cookie:  初始化成功后，才可获取队列队列相关信息； |

### refreshAllQueueStatus

|  |  |
| --- | --- |
| - (void)refreshAllQueueStatus | |
| 功能 | 刷新所有队列状态信息 |
| 返回值 | 无 |
| 参数 | 无 |
| 说明 | 操作回调queueStatusChanged:  （当前排队的队列，或服务的队列，sdk自动有状态变化回调； 其它队列则需要此函数来查询） |

### getQueueInfo

|  |  |
| --- | --- |
| - (NSMutableArray<QueueInfo \*> \*) getQueueInfo | |
| 功能 | 获取队列信息 |
| 返回值 | 无 |
| 参数 | queInfos 队列信息，类定义见 QueueInfo |
| 说明 | 函数参数为传出参数 |

### getQueueStatus:

|  |  |
| --- | --- |
| - (QueueStatus \*)getQueueStatus:(int)queID | |
| 功能 | 获取队列状态 |
| 返回值 | 无 |
| 参数 | queID 队列  queInfos 队列状态信息，类定义见 QueueStatus |
| 说明 | 函数参数为传出参数 |

### getQueuingInfo

|  |  |
| --- | --- |
| - (QueuingInfo \*)getQueuingInfo | |
| 功能 | 获取我的排队信息 |
| 返回值 | 无 |
| 参数 | queuingInfo 我的信息，类定义见 QueuingInfo |
| 说明 | 函数参数为传出参数 |

### getServiceQueues

|  |  |
| --- | --- |
| - (NSMutableArray<NSNumber \*> \*) getServiceQueues | |
| 功能 | 获取我服务的所有队列 |
| 返回值 | 无 |
| 参数 | queueList 我服务的队列列表 |
| 说明 | 函数参数为传出参数 |

### getSessionInfo

|  |  |
| --- | --- |
| - (VideoSessionInfo \*)getSessionInfo | |
| 功能 | 获取我的会话信息 |
| 返回值 | 无 |
| 参数 | sessionInfo 会话信息，类定义见 VideoSessionInfo |
| 说明 | 函数参数为传出参数 |

### startQueuing:cookie:

|  |  |
| --- | --- |
| - (void)startQueuing:(int)queID cookie:(NSString \*)cookie | |
| 功能 | 客户开始排队 |
| 返回值 | 无 |
| 参数 | queID 排队的队列  cookie 用户自定义数据(在回调时，回传给调用者) |
| 说明 | 操作回调startQueuingRslt:cookie: |

### stopQueuing:

|  |  |
| --- | --- |
| - (void)stopQueuing:(NSObject \*)cookie | |
| 功能 | 客户停止排队 |
| 返回值 | 无 |
| 参数 | cookie 用户自定义数据(在回调时，回传给调用者) |
| 说明 | 操作回调stopQueuingRslt:cookie: |

### startService:cookie:

|  |  |
| --- | --- |
| - (void)startService:(int)queID cookie:(NSString \*)cookie | |
| 功能 | 开始服务某个队列(可以多次调用，开启对多个队列的服务) |
| 返回值 | 无 |
| 参数 | queID 排队的队列  cookie 用户自定义数据(在回调时，回传给调用者) |
| 说明 | 操作回调startServiceRslt:errCode:cookie  开启成功后：  a. 如果没有开启免打挽，那么系统会自动分配客户:autoAssignUser:；  b. 如果开启免打挽，系统就不会分配客户，如需服务客户可调用：reqAssignUser: |

### stopService:cookie:

|  |  |
| --- | --- |
| - (void)stopService:(int)queID cookie:(NSString \*)cookie | |
| 功能 | 停止服务某个队列 |
| 返回值 | 无 |
| 参数 | queID 排队的队列  cookie 用户自定义数据(在回调时，回传给调用者) |
| 说明 | 操作回调stopServiceRslt:errCode:cookie: |

### acceptAssignUser:userID:cookie:

|  |  |
| --- | --- |
| - (void)acceptAssignUser:(int)queID userID:(NSString \*)userID cookie:(NSString \*)cookie | |
| 功能 | 接受系统安排的客户 |
| 返回值 | 无 |
| 参数 | queID 排队的队列  userID 队列中的用户  cookie 用户自定义数据(在回调时，回传给调用者) |
| 说明 |  |

### rejectAssignUser:userID:cookie:

|  |  |
| --- | --- |
| - (void)rejectAssignUser:(int)queID userID:(NSString \*)userID cookie: | |
| 功能 | 拒绝系统安排的客户 |
| 返回值 | 无 |
| 参数 | queID 排队的队列  userID 队列中的用户  cookie 用户自定义数据(在回调时，回传给调用者) |
| 说明 | 被拒绝的客户将重新回到队列的最前端 |

### reqAssignUser:

|  |  |
| --- | --- |
| - (void)reqAssignUser:(NSString \*)cookie | |
| 功能 | 请求分配一个客户 |
| 返回值 | 无 |
| 参数 | cookie 用户自定义数据(在回调时，回传给调用者) |
| 说明 | 操作回调reqAssignUserRslt:userInfo:cookie:  当关闭免打扰时，系统将自动分配客户，无需调用此函数；  当开启免打扰时，系统不再自动分配客户，座席如需服务客户可使用此函数分配； |

## CloudroomQueueCallBack协议接口

### initQueueDatRslt:cookie:

|  |  |
| --- | --- |
| - (void)initQueueDatRslt:(CRVIDEOSDK\_ERR\_DEF)errCode cookie:(NSString \*)cookie | |
| 作用 | 队列初始化操作结果 |
| 参数 | errCode 操作结果代码，VCALLSDK\_NOERR为成功操作  cookie 自定义用户数据 |

### queueStatusChanged:

|  |  |
| --- | --- |
| - (void)queueStatusChanged:(QueueStatus \*)queStatus | |
| 作用 | 队列状态变化通知 |
| 参数 | queStatus 新的队列状态，类定义见 QueueStatus |
| 说明 | 在排队的队列、或服务的队列发生变化时，将有队列状态变化通知到来；  或者在调用RefreshAllQueueStatus时，查询到的队列数据有变化时，会有通知到来； |

### queuingInfoChanged:

|  |  |
| --- | --- |
| - (void)queuingInfoChanged:(QueuingInfo \*)queuingInfo | |
| 作用 | 排队信息变化通知 |
| 参数 | queuingInfo 新的队列信息，类定义见 QueuingInfo |

### startQueuingRslt:cookie:

|  |  |
| --- | --- |
| - (void)startQueuingRslt:(CRVIDEOSDK\_ERR\_DEF)errCode cookie:(NSString \*)cookie | |
| 作用 | 开始排队操作结果 |
| 参数 | errCode 操作结果代码，VCALLSDK\_NOERR为成功操作  cookie 自定义用户数据 |

### stopQueuingRslt:cookie:

|  |  |
| --- | --- |
| - (void)stopQueuingRslt:(CRVIDEOSDK\_ERR\_DEF)errCode cookie:(NSString \*)cookie | |
| 作用 | 停止排队操作结果 |
| 参数 | errCode 操作结果代码，VCALLSDK\_NOERR为成功操作  cookie 自定义用户数据 |

### startServiceRslt:errCode:cookie:

|  |  |
| --- | --- |
| - (void)startServiceRslt:(int)queID errCode:(CRVIDEOSDK\_ERR\_DEF)errCode cookie:(NSString \*)cookie | |
| 作用 | 开始服务队列操作结果 |
| 参数 | queID 服务的队列  errCode 操作结果代码，VCALLSDK\_NOERR为成功操作  cookie 自定义用户数据 |

### stopServiceRslt:errCode:cookie:

|  |  |
| --- | --- |
| - (void)stopServiceRslt:(int)queID errCode:(CRVIDEOSDK\_ERR\_DEF)errCode cookie:(NSString \*)cookie | |
| 作用 | 停止服务队列操作结果 |
| 参数 | queID 服务的队列  errCode 操作结果代码，VCALLSDK\_NOERR为成功操作  cookie 自定义用户数据 |

### responseAssignUserRslt:cookie:

|  |  |
| --- | --- |
| - (void)responseAssignUserRslt:(CRVIDEOSDK\_ERR\_DEF)errCode cookie:(NSString \*)cookie | |
| 作用 | 拒绝、接受分配的用户 |
| 参数 | errCode 操作结果代码，VCALLSDK\_NOERR为成功操作  cookie 自定义用户数据 |

### autoAssignUser:

|  |  |
| --- | --- |
| - (void)autoAssignUser:(UserInfo \*)usr | |
| 作用 | 系统自动安排客户 |
| 参数 | usr 安排的用户，类定义见 UserInfo |
| 说明 | 如果想停止系统的自动分配，请通setDNDStatus:cookie:设置免打扰功能； |

### reqAssignUserRslt:userInfo:cookie:

|  |  |
| --- | --- |
| - (void)reqAssignUserRslt:(CRVIDEOSDK\_ERR\_DEF)errCode userInfo:(UserInfo \*)usr cookie:(NSString \*)cookie | |
| 作用 | 请求分配客户操作结果 |
| 参数 | errCode 操作结果代码，VCALLSDK\_NOERR为成功操作  usr 分到的用户  cookie 自定义用户数据 |

### cancelAssignUser:usrID:

|  |  |
| --- | --- |
| -(void) cancelAssignUser:(int)queID usrID:(NSString \*)usrID | |
| 作用 | 系统取消自动安排客户 |
| 参数 | queID 服务的队列  usrID 用户id |
| 说明 |  |

## CloudroomVideoMgr类接口

### setMgrCallback:

|  |  |
| --- | --- |
| * (void)setMgrCallback:(id<CloudroomVideoMgrCallBack>)callBack | |
| 功能 | 设置代理回调 |
| 返回值 | 无 |
| 参数 | 一切遵循协议: CloudroomVideoMgrCallBack的对象 |
| 说明 |  |

### login:/login:cookie:

|  |  |
| --- | --- |
| - (void)login:(LoginDat \*)loginDat | |
| 功能 | 登陆服务器 |
| 返回值 | 无 |
| 参数 | LoginDat 登陆类，定义见 LoginDat |
| 说明 | 成功回调loginSuccess:cookie:  失败回调loginFail:cookie: |

|  |  |
| --- | --- |
| - (void)login:(LoginDat \*)loginDat cookie:(NSString \*)cookie | |
| 功能 | 登陆服务器 |
| 返回值 | 无 |
| 参数 | LoginDat 登陆类，定义见 LoginDat  cookie 自定义数据 (在回调时，回传给调用者) |
| 说明 | 成功回调loginSuccess:cookie:  失败回调loginFail:cookie: |

### logout

|  |  |
| --- | --- |
| - (void)logout | |
| 功能 | 注销本次登陆 |
| 返回值 | 无 |
| 参数 | 无 |
| 说明 | 没有回调 |

### setDNDStatus:/setDNDStatus:cookie:

|  |  |
| --- | --- |
| - (void)setDNDStatus:(int)DNDStatus | |
| 功能 | 设置免打扰状态 |
| 返回值 | 无 |
| 参数 | DNDStatus 0代表关闭免打扰，其它值代表开启免打扰，含义可自由定义 |
| 说明 | 成功回调setDNDStatusSuccess:  失败回调setDNDStatusFail:cookie: |
| - (void)setDNDStatus:(int)DNDStatus cookie:(NSString \*)cookie | |
| 功能 | 设置免打扰状态 |
| 返回值 | 无 |
| 参数 | DNDStatus 0代表关闭免打扰， 其它值代表开启免打扰，含义可自由定义  cookie 自定义数据(在回调时，回传给调用者) |
| 说明 | 成功回调setDNDStatusSuccess:  失败回调setDNDStatusFail:cookie: |

### createMeeting:createPswd:cookie:

|  |  |
| --- | --- |
| * (void)createMeeting:(NSString \*)meetSubject createPswd:(BOOL)createPswd | |
| 功能 | 创建会议 |
| 返回值 | 无 |
| 参数 | meetSubject:会议名称  createPswd:是否创建密码 |
| 说明 |  |
| * (void)createMeeting:(NSString \*)meetSubject createPswd:(BOOL)createPswd cookie:(NSString \*)cookie | |
| 功能 | 创建会议 |
| 返回值 | 无 |
| 参数 | meetSubject 会议名称  createPswd:是否创建密码  cookie 自定义数据(在回调时，回传给调用者) |
| 说明 |  |

### call:meetInfo:param:cookie:

|  |  |
| --- | --- |
| -(NSString \*)call:(NSString \*)calledUserID meetInfo:(MeetInfo \*)meetInfo param:(NSString \*)param cookie:(NSString \*)cookie | |
| 功能 | 发起呼叫，邀请用户参加视频会话 |
| 返回值 | 本次呼叫标识码（呼叫ID） |
| 参数 | calledUserID 被叫用户的账户ID  meetInfo会议信息  param 呼叫参数，将传给被叫用户  cookie 自定义数据(在回调时，回传给调用者) |
| 说明 | 成功回调callSuccess:cookie:  失败回调callFail:errCode:ookie:  呼叫时，对方迟迟不响应，30秒后系统自动结束呼叫 |

### acceptCall: meetInfo:/acceptCall: meetInfo:cookie:

|  |  |
| --- | --- |
| - (void)acceptCall:(NSString \*)callID meetInfo:(MeetInfo\*)meetInfo | |
| 功能 | 接受对方发起的视频请求，开始进入视频会话 |
| 返回值 | 无 |
| 参数 | callID 呼叫标识码  meetInfo 会议信息 |
| 说明 | 成功回调acceptCallSuccess:cookie:  失败回调acceptCallFail:errCode: |
| - (void)acceptCall:(NSString \*)callID meetInfo:(MeetInfo\*)meetInfo cookie:(NSString \*)cookie | |
| 功能 | 接受对方发起的视频请求，开始进入视频会话 |
| 返回值 | 无 |
| 参数 | callID 呼叫标识码  meetInfo 会议信息  cookie 自定义数据(在回调时，回传给调用者) |
| 说明 | 成功回调acceptCallSuccess:cookie:  失败回调acceptCallFail:errCode: |

### rejectCall:/rejectCall:cookie:

|  |  |
| --- | --- |
| - (void)rejectCall:(NSString \*)callID | |
| 功能 | 拒绝对方的视频请求 |
| 返回值 | 无 |
| 参数 | callID 呼叫标识码 |
| 说明 | 成功回调rejectCallSuccess:cookie:  失败回调rejectCallFail:errCode: |
| - (void)rejectCall:(NSString \*)callID cookie:(NSString \*)cookie | |
| 功能 | 拒绝对方的视频请求 |
| 返回值 | 无 |
| 参数 | callID 呼叫标识码  cookie 自定义数据(在回调时，回传给调用者) |
| 说明 | 成功回调rejectCallSuccess:cookie:  失败回调rejectCallFail:errCode: |

### hungupCall:/hungupCall:cookie:

|  |  |
| --- | --- |
| - (void)hungupCall:(NSString \*)callID | |
| 功能 | 挂断正在进行的视频呼叫或视频通话 |
| 返回值 | 无 |
| 参数 | callID 呼叫标识码 |
| 说明 | 操作成功回调hangupCallSuccess:cookie:  失败回调hangupCallFail:errCode: |
| - (void)hungupCall:(NSString \*)callID cookie:(NSString \*)cookie | |
| 功能 | 挂断正在进行的视频呼叫或视频通话 |
| 返回值 | 无 |
| 参数 | callID 呼叫标识码  cookie 自定义数据(在回调时，回传给调用者) |
| 说明 | 操作成功回调hangupCallSuccess:cookie:  失败回调hangupCallFail:errCode: |

## CloudroomVideoMgrCallBack协议接口

### loginSuccess:cookie:

|  |  |
| --- | --- |
| - (void)loginSuccess:(NSString \*)usrID cookie:(NSString \*)cookie | |
| 功能 | 登录成功响应 |
| 参数 | usrID 用户ID  cookie 自定义用户数据 |

### loginFail:cookie:

|  |  |
| --- | --- |
| - (void)loginFail:(CRVIDEOSDK\_ERR\_DEF)sdkErr cookie:(NSString \*)cookie | |
| 功能 | 登录失败响应 |
| 参数 | sdkErr 操作失败代码，定义见 CRVIDEOSDK\_ERR\_DEF  cookie 自定义用户数据 |

### lineOff:

|  |  |
| --- | --- |
| - (void)lineOff:(CRVIDEOSDK\_ERR\_DEF)sdkErr | |
| 功能 | SDK通知自己掉线 |
| 参数 | sdkErr 掉线的错误代码，VCALLSDK\_ERR\_OFFLINE：掉线, VCALLSDK\_USER\_BEEN\_KICKOUT:被踢 |

### setDNDStatusSuccess:

|  |  |
| --- | --- |
| - (void)setDNDStatusSuccess:(NSObject \*)cookie | |
| 功能 | 客户端设置免打扰状态操作成功响应 |
| 参数 | cookie 自定义用户数据 |

### setDNDStatusFail:cookie:

|  |  |
| --- | --- |
| - (void)setDNDStatusFail:(CRVIDEOSDK\_ERR\_DEF)sdkErr cookie:(NSString \*)cookie | |
| 功能 | 客户端设置免打扰状态操作失败响应 |
| 参数 | sdkErr 操作失败代码  cookie 自定义用户数据 |

### createMeetingSuccess:password:cookie:

|  |  |
| --- | --- |
| * (void)createMeetingSuccess:(MeetInfo \*) meetInfo cookie:(NSString \*)cookie | |
| 功能 | 创建会议成功 |
| 参数 | meetInfo 会议信息  cookie 自定义用户数据 |

### createMeetingFail:cookie:

|  |  |
| --- | --- |
| - (void)createMeetingFail:(CRVIDEOSDK\_ERR\_DEF)sdkErr  cookie:(NSString \*)cookie | |
| 功能 | 创建会议失败 |
| 参数 | sdkErr错误码，定义见 CRVIDEOSDK\_ERR\_DEF  cookie 自定义用户数据 |

### callSuccess:

|  |  |
| --- | --- |
| - (void)callSuccess:(NSString \*)callID cookie:(NSString \*)cookie | |
| 功能 | 呼叫他人操作成功响应 |
| 参数 | callID 呼叫全局标识  cookie 自定义用户数据 |

### callFail:

|  |  |
| --- | --- |
| - (void)callFail:(NSString \*)callID errCode:(CRVIDEOSDK\_ERR\_DEF)sdkErr cookie:(NSString \*)cookie | |
| 功能 | 呼叫他人操作失败响应 |
| 参数 | callID 呼叫全局标识  sdkErr 操作失败代码  cookie 自定义用户数据 |

### acceptCallSuccess:cookie:

|  |  |
| --- | --- |
| - (void)acceptCallSuccess:(NSString \*)callID cookie:(NSString \*)cookie | |
| 功能 | 接受他人呼叫操作成功响应 |
| 参数 | callID 呼叫全局标识  cookie 自定义用户数据 |

### acceptCallFail:errCode:

|  |  |
| --- | --- |
| - (void)acceptCallFail:(NSString \*)callID errCode:(CRVIDEOSDK\_ERR\_DEF)sdkErr cookie:(NSString \*)cookie | |
| 功能 | 接受他人呼叫操作失败响应 |
| 参数 | callID 呼叫全局标识  sdkErr 操作失败代码  cookie 自定义用户数据 |

### rejectCallSuccess:

|  |  |
| --- | --- |
| - (void)rejectCallSuccess:(NSString \*)callID cookie:(NSString \*)cookie | |
| 功能 | 拒绝他人的呼叫成功响应 |
| 参数 | callID 呼叫全局标识  cookie 自定义用户数据 |

### rejectCallFail:errCode:

|  |  |
| --- | --- |
| - (void)rejectCallFail:(NSString \*)callID errCode:(CRVIDEOSDK\_ERR\_DEF)sdkErr cookie:(NSString \*)cookie | |
| 功能 | 拒绝他人的呼叫失败响应 |
| 参数 | callID 呼叫全局标识  sdkErr 操作失败代码  cookie 自定义用户数据 |

### hangupCallSuccess:cookie:

|  |  |
| --- | --- |
| - (void)hangupCallSuccess:(NSString \*)callID cookie:(NSString \*)cookie | |
| 功能 | 挂断呼叫操作成功响应 |
| 参数 | callID 呼叫全局标识  cookie 自定义用户数据 |

### hangupCallFail:errCode:

|  |  |
| --- | --- |
| - (void)hangupCallFail:(NSString \*)callID errCode:(CRVIDEOSDK\_ERR\_DEF)sdkErr cookie:(NSString \*)cookie | |
| 功能 | 拒绝他人呼叫操作失败响应 |
| 参数 | callID 呼叫全局标识  sdkErr 操作失败代码  cookie 自定义用户数据 |

### notifyCallIn:

|  |  |
| --- | --- |
| - (void)notifyCallIn:(NSString \*)callID meetInfo:(MeetInfo \*)meetInfo callerID:(NSString \*)callerID param:(NSString \*)param | |
| 功能 | SDK通知自己被呼叫 |
| 参数 | callID 呼叫全局标识  meetInfo 会议信息  callerID 呼叫人员的标识ID  param 自定义扩展参数 |

### notifyCallAccepted:meetInfo:meetPswd:

|  |  |
| --- | --- |
| - (void)notifyCallAccepted:(NSString \*)callID meetInfo:(MeetInfo \*)meetInfo | |
| 功能 | SDK通知自己视频呼叫被对方接受 |
| 参数 | callID 呼叫全局标识  meetInfo 会议信息 |

### notifyCallRejected:reason:

|  |  |
| --- | --- |
| - (void)notifyCallRejected:(NSString \*)callID reason:(CRVIDEOSDK\_ERR\_DEF)reason | |
| 功能 | SDK通知自己呼叫被对方拒绝 |
| 参数 | callID 呼叫全局标识  sdkErr 呼叫被对方拒绝的原因代码，定义见 CRVIDEOSDK\_ERR\_DEF |

### notifyCallHungup:

|  |  |
| --- | --- |
| - (void)notifyCallHungup:(NSString \*)callID | |
| 功能 | SDK通知自己呼叫被挂断 |
| 参数 | callID 呼叫全局标识 |

## CloudroomVideoMeeting类接口

### setMeetingCallBack:

|  |  |
| --- | --- |
| * (void)setMeetingCallBack:(id<CloudroomVideoMeetingCallBack>)callBack | |
| 功能 | 设置代理回调 |
| 返回值 | 无 |
| 参数 | 一切遵循协议:CloudroomVideoMeetingCallBack的对象 |
| 说明 |  |

### enterMeeting:pswd:userID:nikeName:

|  |  |
| --- | --- |
| * (void)enterMeeting:(int)meetID pswd:(NSString \*)pswd userID:(NSString \*)userID nikeName:(NSString \*)nikeName | |
| 功能 | 进入会议（只有入会成功，才能进行后续操作） |
| 返回值 | 无 |
| 参数 | meetID 视频会话ID  pswd 本次会议中的密码（系统自动生成，在呼叫回调中取得）  userID用户ID  nikeName 用户在本次会议中使用的昵称 |
| 说明 | 操作完成回调enterMeetingRslt:  失败则释放本次呼叫。 |

### isUserInMeeting:

|  |  |
| --- | --- |
| - (BOOL)isUserInMeeting:(NSString\*)userID | |
| 功能 | 判断某个用户是否在会话中 |
| 返回值 | true：在；false：不在 |
| 参数 | userID 用户ID |
| 说明 |  |

### stopMeeting

|  |  |
| --- | --- |
| - (void)stopMeeting | |
| 功能 | 结束会议 |
| 返回值 | 无 |
| 参数 | 无 |
| 说明 | 回调endMeetingRslt: |

### exitMeeting

|  |  |
| --- | --- |
| - (void)exitMeeting | |
| 功能 | 离开会议 |
| 返回值 | 无 |
| 参数 | 无 |
| 说明 |  |

### getNickName:

|  |  |
| --- | --- |
| - (NSString \*)getNickName:(NSString \*)userID | |
| 功能 | 通过用户ID获取昵称 |
| 返回值 | 无 |
| 参数 | userID 用户ID |
| 说明 |  |

### getAllMembers

|  |  |
| --- | --- |
| - (NSMutableArray <MemberInfo \*> \*)getAllMembers | |
| 功能 | 获取所有成员列表 |
| 返回值 | 所有成员列表 |
| 参数 | 无 |
| 说明 |  |

### getMemberInfo:

|  |  |
| --- | --- |
| - (MemberInfo \*)getMemberInfo:(NSString \*)userId | |
| 功能 | 通过用户ID获取会议成员信息 |
| 返回值 | 会议成员信息 |
| 参数 | userID:用户ID |
| 说明 |  |

### setAllAudioClose

|  |  |
| --- | --- |
| - (void)setAllAudioClose | |
| 功能 | 关闭麦克风 |
| 返回值 | 无 |
| 参数 | 无 |
| 说明 |  |

### setMicVolume:

|  |  |
| --- | --- |
| - (bool)setMicVolume:(int)level | |
| 功能 | 设置麦克风音量大小 |
| 返回值 | 设置成功返回true，否则返回false |
| 参数 | level音量等级（ 取值范围：0~100） |
| 说明 |  |

### getMicVolume

|  |  |
| --- | --- |
| - (int)getMicVolume | |
| 功能 | 获取麦克风音量 |
| 返回值 | 麦克风音量（0-100） |
| 参数 | 无 |
| 说明 |  |

### setSpeakerVolume:

|  |  |
| --- | --- |
| - (bool)setSpeakerVolume:(int)level | |
| 功能 | 设置本地扬声器音量 |
| 返回值 | 成功返回true，失败返回false |
| 参数 | level 音量等级（0-255） |
| 说明 |  |

### getSpeakerVolume

|  |  |
| --- | --- |
| - (int)getSpeakerVolume | |
| 功能 | 获取本地扬声器音量 |
| 返回值 | 扬声器音量(0~255) |
| 参数 | 无 |
| 说明 |  |

### getMicEnergy:

|  |  |
| --- | --- |
| - (int)getMicEnergy:(NSString \*)userID | |
| 功能 | 获取用户说话声音大小 |
| 返回值 | 音量大小（0~10） |
| 参数 | userID 用户ID |
| 说明 |  |

### openMic:

|  |  |
| --- | --- |
| - (void)openMic:(NSString \*)userId | |
| 功能 | 打开自己的麦克风 |
| 返回值 | 无 |
| 参数 | userID 用户ID |
| 说明 | 打开自已的麦克风时，先会进入到AOPENING状态，等服务器处理后才会进入AOPEN状态，此时说话才能被采集到； |

### closeMic:

|  |  |
| --- | --- |
| - (void)closeMic:(NSString \*)userId | |
| 功能 | 关闭自己的麦克风 |
| 返回值 | 无 |
| 参数 | userID 用户ID |
| 说明 | 关麦操作是立即生效的，本地会立即停止采集； |

### getAudioStatus:

|  |  |
| --- | --- |
| - (ASTATUS)getAudioStatus:(NSString \*)userID | |
| 功能 | 获取用户的麦状态 |
| 返回值 | 麦克风设备状态，请参见ASTATUS定义 |
| 参数 | userID 用户ID |
| 说明 |  |

### getSpeakerOut

|  |  |
| --- | --- |
| * (BOOL)getSpeakerOut | |
| 功能 | 获取外放状态 |
| 返回值 | YES:外放开启,否则外放关闭 |
| 参数 |  |
| 说明 |  |

### setSpeakerOut:

|  |  |
| --- | --- |
| * (BOOL)setSpeakerOut:(BOOL)speakerOut | |
| 功能 | 设置外放状态 |
| 返回值 | YES:设置成功,否则设置失败 |
| 参数 | YES:设置外放开启,否则设置外放关闭 |
| 说明 |  |

### setSpeakerMute:

|  |  |
| --- | --- |
| * (void)setSpeakerMute:(BOOL)bMute | |
| 功能 | 设置扬声器是否静音 |
| 返回值 | 无 |
| 参数 | YES:设置扬声器静音,否则不静音 |
| 说明 |  |

### getSpeakerMute

|  |  |
| --- | --- |
| * (BOOL)getSpeakerMute | |
| 功能 | 查询扬声器是否静音 |
| 返回值 | YES:扬声器静音,否则不静音 |
| 参数 |  |
| 说明 |  |

### setVideoCfg:

|  |  |
| --- | --- |
| * (BOOL)setVideoCfg:(VideoCfg \*)cfg | |
| 功能 | 设置视频配置信息 |
| 返回值 | YES:设置成功，否则失败 |
| 参数 | cfg 视频配置，数据结构见 VideoCfg |
| 说明 |  |

### getVideoCfg

|  |  |
| --- | --- |
| * (VideoCfg \*)getVideoCfg | |
| 功能 | 获取视频配置信息 |
| 返回值 | 视频配置信息 |
| 参数 | 无 |
| 说明 |  |

### getVideoStatus:

|  |  |
| --- | --- |
| - (VSTATUS)getVideoStatus:(NSString \*)userID | |
| 功能 | 获取用户的视频状态 |
| 返回值 | 设备状态，请参见VSTATUS定义 |
| 参数 | userID 用户ID |
| 说明 |  |

### openVideo:

|  |  |
| --- | --- |
| - (void)openVideo:(NSString \*)userID | |
| 功能 | 打开本地摄像头，以便本地、远端显示视频图像 |
| 返回值 | 无 |
| 参数 | userID 用户ID |
| 说明 |  |

### closeVideo:

|  |  |
| --- | --- |
| - (void)closeVideo:(NSString \*)userID | |
| 功能 | 关闭本地视频 |
| 返回值 | 无 |
| 参数 | userID 用户ID |
| 说明 |  |

### watchVideos:

|  |  |
| --- | --- |
| - (void)watchVideos:(NSMutableArray <UsrVideoId \*> \*)videoIds | |
| 功能 | 设置需要观看视频的所有者 |
| 返回值 | 无 |
| 参数 | videoIds:观看视频的所有者列表 |
| 说明 |  |

### setVideoFormat:

|  |  |
| --- | --- |
| - (void)setVideoFormat:(VIDEO\_FORMAT)fmt | |
| 功能 | 设置视频输出的格式 |
| 返回值 | 无 |
| 参数 | fmt 视频格式，参见VIDEO\_FORMAT定义 |
| 说明 | 推荐使用：VFMT\_YUV420P（sdk内部的开销更小） |

### getVideoImg:

|  |  |
| --- | --- |
| * (VideoFrame \*)getVideoImg:(UsrVideoId \*)userID | |
| 功能 | 获取用户视频数据 |
| 返回值 | 视频帧 |
| 参数 | userID 详见数据结构:UsrVideoId |
| 说明 |  |

### getAllVideoInfo:

|  |  |
| --- | --- |
| - (NSMutableArray <UsrVideoInfo \*> \*)getAllVideoInfo:(NSString \*)userId | |
| 功能 | 获取多摄像头信息（自己videoIdList可以多个，远端只能一个） |
| 返回值 | 摄像头信息列表 |
| 参数 | userID 用户ID |
| 说明 |  |

### setDefaultVideo:videoID:

|  |  |
| --- | --- |
| - (void)setDefaultVideo:(NSString \*)userId videoID:(short)videoID | |
| 功能 | 设置默认摄像头 |
| 返回值 | 无 |
| 参数 | userID 用户ID  videoID摄像头ID |
| 说明 |  |

### getDefaultVideo:

|  |  |
| --- | --- |
| - (short)getDefaultVideo:(NSString \*)userId | |
| 功能 | 获取默认摄像头ID |
| 返回值 | 默认摄像头ID |
| 参数 | userID 用户ID |
| 说明 |  |

### getWatchableVideos:

|  |  |
| --- | --- |
| - (NSMutableArray <UsrVideoId \*> \*)getWatchableVideos | |
| 功能 | 获取可观看的摄像头 |
| 返回值 | 可观看的摄像头列表 |
| 参数 | 无 |
| 说明 |  |

### isScreenShareStarted

|  |  |
| --- | --- |
| -(BOOL)isScreenShareStarted | |
| 功能 | 是否已开始屏幕共享 |
| 返回值 | true为开始屏幕共享，false为未开始屏幕共享 |
| 参数 | 无 |
| 说明 |  |

### getScreenSharer

|  |  |
| --- | --- |
| -(NSString \*)getScreenSharer | |
| 功能 | 获取屏幕共享者 |
| 返回值 | 屏幕共享者 |
| 参数 | 无 |
| 说明 |  |

### getShareScreenDecodeImg

|  |  |
| --- | --- |
| - (ScreenShareImg \*)getShareScreenDecodeImg | |
| 功能 | 获取屏幕共享数据 |
| 返回值 | 屏幕共享数据 |
| 参数 | 无 |
| 说明 | 此共享Image数据应该每次获取，外界不能缓存使用 |

## CloudroomMeetingSDKCallBack(协议)接口

### enterMeetingRslt:

|  |  |
| --- | --- |
| - (void)enterMeetingRslt: (CRVIDEOSDK\_ERR\_DEF)code | |
| 功能 | 进入视频会话完成结果 |
| 参数 | 无 |

### userEnterMeeting:

|  |  |
| --- | --- |
| - (void)userEnterMeeting:(NSString \*)userID | |
| 功能 | 某用户进入了会议 |
| 参数 | userID 用户ID |

### userLeftMeeting:

|  |  |
| --- | --- |
| - (void)userLeftMeeting:(NSString \*)userID | |
| 功能 | 某用户离开了会议 |
| 参数 | userID 用户ID |

### endMeetingRslt:

|  |  |
| --- | --- |
| - (void)endMeetingRslt:(CRVIDEOSDK\_ERR\_DEF)code | |
| 功能 | 结束会议的结果 |
| 参数 | code 错误码，定义见 CRVIDEOSDK\_ERR\_DEF |

### meetingStopped

|  |  |
| --- | --- |
| - (void)meetingStopped | |
| 功能 | 会议被结束 |
| 参数 | 无 |

### meetingDropped

|  |  |
| --- | --- |
| - (void)meetingDropped | |
| 功能 | 会议掉线 |
| 参数 | 无 |

### netStateChanged:

|  |  |
| --- | --- |
| - (void)netStateChanged:(int)level | |
| 作用 | SDK通知我的网络变化 |
| 参数 | level 网络状况等级(0~10，10分为最佳网络) |

### micEnergyUpdate:oldLevel:newLevel:

|  |  |
| --- | --- |
| - (void)micEnergyUpdate:(NSString \*)userID oldLevel:(int)oldLevel newLevel:(int)newLevel | |
| 作用 | SDK通知用户的说话声音强度更新 |
| 参数 | userID 用户ID  oldLevel 原来的说话声音强度  newLevel 现在的说话声音强度 |

### audioDevChanged

|  |  |
| --- | --- |
| - (void)audioDevChanged | |
| 作用 | SDK通知本地音频设备有变化 |
| 参数 | 无 |

### audioStatusChanged:oldStatus:newStatus:

|  |  |
| --- | --- |
| - (void)audioStatusChanged:(NSString \*)userID oldStatus:(ASTATUS)oldStatus newStatus:(ASTATUS)newStatus | |
| 作用 | SDK通知打开本地音频状态变化 |
| 参数 | userID 会话中设备的所有者  oldStatus 旧状态  newStatus 新状态 |

### videoStatusChanged:oldStatus:newStatus:

|  |  |
| --- | --- |
| -(void)videoStatusChanged:(NSString\*)userID oldStatus:(VSTATUS)oldStatus newStatus:(VSTATUS)newStatus | |
| 作用 | SDK通知打开本地音频状态变化 |
| 参数 | userID 会话中设备的所有者  oldStatus 旧状态  newStatus 新状态 |

### openVideoRslt:

|  |  |
| --- | --- |
| * (void)openVideoRslt:(BOOL)bSuccess | |
| 作用 | 打开视频结果 |
| 参数 | userID 会话中设备的所有者  oldStatus 旧状态  newStatus 新状态 |

### notifyVideoData:frameTime:

|  |  |
| --- | --- |
| * (void)notifyVideoData:(UsrVideoId \*)userID frameTime:(long)frameTime | |
| 作用 | 通知用户有新的视频数据 |
| 参数 | userID 详见数据结构: UsrVideoId  frameTime当前帧创建时间 |

### videoDevChanged:

|  |  |
| --- | --- |
| - (void)videoDevChanged:(NSString \*)userID | |
| 作用 | SDK通知本地视频设备有变化 |
| 参数 | userID 用户ID |

### defVideoChanged:videoID:

|  |  |
| --- | --- |
| - (void)defVideoChanged:(NSString \*)userID videoID:(short)videoID | |
| 作用 | 默认摄像头变更 |
| 参数 | userID 用户ID  videoID 摄像头ID |

### notifyScreenShareStarted

|  |  |
| --- | --- |
| - (void)notifyScreenShareStarted | |
| 作用 | 开始屏幕共享 |
| 参数 | 无 |

### notifyScreenShareStopped

|  |  |
| --- | --- |
| - (void)notifyScreenShareStopped | |
| 作用 | 结束屏幕共享 |
| 参数 | 无 |

### notifyScreenShareData:changedRect:

|  |  |
| --- | --- |
| - (void)notifyScreenShareData:(NSString \*)userID changedRect:(CGRect)changedRect | |
| 作用 | 屏幕共享数据更新，用户收到该回调消息后应该调用getShareScreenDecodeImg获取最新的共享数据 |
| 参数 | userID用户ID  changedRect 改变区域 |

### notifyMediaStart:

|  |  |
| --- | --- |
| * (void)notifyMediaStart:(NSString \*)userid | |
| 作用 | 影音开始通知 |
| 参数 | userID用户ID |

### notifyMediaStop: reason:

|  |  |
| --- | --- |
| * (void)notifyMediaStop:(NSString \*)userid reason:(MEDIA\_STOP\_REASON)reason | |
| 作用 | 影音停止播放的通知 |
| 参数 | userID用户ID  reason原因 |

### notifyMediaPause: bPause:

|  |  |
| --- | --- |
| * (void)notifyMediaPause:(NSString \*)userid bPause:(BOOL)bPause | |
| 作用 | 影音暂停播放的通知 |
| 参数 | userID用户ID  bPause是否暂停 |

### notifyMemberMediaData:

|  |  |
| --- | --- |
| * (void)notifyMemberMediaData:(NSString \*)userid | |
| 作用 | 视频帧数据已解好通知 |
| 参数 | userID用户ID |