**API**

API接口由多个模块组成 ，主要模块包括：基础函数、注册/反注册回调、登录/反注册、成员管理、 队列管理、透明通道/呼叫排队、邀请、房间管理、音频管理、视频管理、 屏幕共享、影音共享、本地录制/本地直播、云端录制/互动直播、 电子白板/文档共享、Http文件组件

**接口描述**

**基础函数：**

| **组件** | **方式** | **接口** | **描述** |
| --- | --- | --- | --- |
| CloudroomVideoSDK | 主调 | GetCloudroomVideoSDKVer | 获取sdk的版本信息 |
| init | SDK初始化 |
| uninit | SDK反初始化 |
| serverAddr | 获取服务器地址 |
| setServerAddr | 设置服务器地址 |
| setFileUploadRate | 文件上传的流量控制 |
| setFileDownloadRate | 文件下载的流量控制 |

**注册/反注册回调：**

| **组件** | **方式** | **接口** | **描述** |
| --- | --- | --- | --- |
| CloudroomVideoMgr | 主调 | registerCallBack | 房间管理注册监听回调 |
| unregisterCallBack | 房间管理反注册监听回调 |
| CloudroomVideoMeeting | 主调 | registerCallBack | 注册视频房间核心管理监听回调 |
| unregisterCallBack | 反注册房间管理监听回调 |
| CloudroomQueue | 主调 | registerCallBack | 注册队列管理监听回调 |
| unregisterCallBack | 反注册队列管理监听回调 |
| CloudroomHttpFileMgr | 主调 | registerCallBack | 注册Http文件传输管理监听回调 |
| unregisterCallBack | 反注册Http文件传输管理监听回调 |

**登录/反注册：**

| **组件** | **方式** | **接口** | **描述** |
| --- | --- | --- | --- |
| CloudroomVideoMgr | 主调 | login | 密码登录 |
| loginByToken | Token登录 |
| getUserAuthErrCode | 获取用户认证错误码 |
| getUserAuthErrDesc | 获取用户验证错误描述信息 |
| logout | 反注册本次登陆 |
| updateToken | 更新Token |
| CloudroomVideoMgrCallback | 主调 | loginSuccess | 登录成功响应 |
| loginFail | 登录失败响应 |
| notifyTokenWillExpire | Token即将失效 |
| lineOff | 通知用户掉线 |

**成员管理：**

| **组件** | **方式** | **接口** | **描述** |
| --- | --- | --- | --- |
| CloudroomVideoMeeting | 主调 | getAllMembers | 获取房间所有成员的列表 |
| getMemberInfo | 获取某个用户的信息 |
| getNickName | 获取某个用户的昵称 |
| setNickName | 设置某个用户的昵称 |
| isUserInMeeting | 判断某个用户是否在房间中 |

**队列管理**

| **组件** | **方式** | **接口** | **描述** |
| --- | --- | --- | --- |
| CloudroomQueue | 主调 | initQueue | 初始化用户队列功能数据 |
| refreshAllQueueStatus | 刷新所有队列状态信息 |
| getAllQueueInfo | 获取队列信息 |
| getQueueStatus | 获取指定队列状态 |
| getQueuingInfo | 获取我的排队信息 |
| getServiceQueues | 获取我服务的所有队列 |
| getSessionInfo | 获取我的会话信息 |
| startQueuing | 客户开始排队 |
| stopQueuing | 客户停止排队 |
| startService | 开始服务某个队列(可以多次调用，开启对多个队列的服务) |
| stopService | 停止服务某个队列 |
| reqAssignUser | 请求分配一个客户 |
| acceptAssignUser | 接受系统安排的客户 |
| rejectAssignUser | 拒绝系统安排的客户 |
| CloudroomQueueCallback | 回调 | initQueueDatRslt | 队列初始化操作结果 |
| queueStatusChanged | 队列状态变化通知 |
| queuingInfoChanged | 排队信息变化通知 |
| startQueuingRslt | startQueuing 开始排队操作结果 |
| stopQueuingRslt | stopQueuing 停止排队操作结果 |
| startServiceRslt | startService 开始服务队列操作结果 |
| reqAssignUserRslt | 请求分配客户操作结果 |
| autoAssignUser | 队列系统自动分配客户 |
| cancelAssignUser | 队列系统取消之前自动分配的的客户 |
| responseAssignUserRslt | 接受分配的用户 |

**透明通道/呼叫排队**

| **组件** | **方式** | **接口** | **描述** |
| --- | --- | --- | --- |
| CloudroomVideoMgr | 主调 | setDNDStatus | 设置免打扰状态 |
| getUserStatus | 获取企业下指定用户的在线状态（包括呼叫房间状态、免打扰状态） |
| startUserStatusNotify | 开启用户的状态推送 |
| stopUserStatusNotify | 关闭用户的状态推送 |
| call | 发起呼叫，邀请用户参加视频会话 |
| acceptCall | 接受对方发起的视频请求，开始进入视频会话 |
| rejectCall | 拒绝对方的视频请求 |
| hangupCall | 挂断正在进行的视频呼叫或视频通话 |
| sendCmd | 发送小块数据（50K以内） |
| sendBuffer | 发送大块数据 |
| sendFile | 发送文件 |
| cancelSend | 取消数据发送 |
| setMarkText | 添加打点信息 |
| removeMarkText | 移除打点信息 |
| getAllMarks | 获取所有打点信息 |
| getVideoMarkFile | 获取打点文件的绝对文件路径名 |
| CloudroomVideoMeeting | 主调 | sendMeetingCustomMsg | 房间内发送广播消息 |
| CloudroomVideoMgrCallback | 回调 | setDNDStatusSuccess | 客户端设置免打扰状态操作成功响应 |
| setDNDStatusFail | 客户端设置免打扰状态操作失败响应 |
| getUserStatusRslt | 获取企业内所有用户在线状态响应 |
| startStatusPushRslt | 启动用户状态推送响应 |
| stopStatusPushRslt | 结束用户状态推送响应 |
| notifyUserStatus | 某个用户状态变化通知 |
| callSuccess | 呼叫他人操作成功响应 |
| callFail | 呼叫他人操作失败响应 |
| acceptCallSuccess | 接受他人呼叫操作成功响应 |
| acceptCallFail | 接受他人呼叫操作失败响应 |
| rejectCallSuccess | 拒绝他人的呼叫成功响应 |
| rejectCallFail | 拒绝他人的呼叫失败响应 |
| hangupCallSuccess | 挂断呼叫操作成功响应 |
| hangupCallFail | 拒绝他人呼叫操作失败响应 |
| notifyCallIn | SDK通知自己被呼叫 |
| notifyCallAccepted | SDK通知自己视频呼叫被对方接受 |
| notifyCallRejected | SDK通知自己呼叫被对方拒绝 |
| notifyCallhangup | SDK通知呼叫被挂断 |
| sendCmdRlst | 发送数据时，SDK通知发送结果 |
| sendBufferRlst |
| sendFileRlst |
| sendProgress | 发送数据时，SDK通知发送进度 |
| cancelSendRlst | 取消发送结果 |
| notifyCmdData | SDK通知收到小块数据 |
| notifyBufferData | SDK通知收到大块数据 |
| notifyFileData | SDK通知收到文件数据 |
| notifyCancelSend | SDK通知取消发送文件数据 |
| CloudroomVideoCallback | 回调 | sendMeetingCustomMsgRlst | 发送房间内用户广播结果 |
| sendMeetingCustomMsgRslt | 接收到房间内用户自定义广播 |

**邀请**

| **组件** | **方式** | **接口** | **描述** |
| --- | --- | --- | --- |
| CloudroomVideoMgr | 主调 | invite | 发送邀请 |
| acceptInvite | 接受对方发送的邀请 |
| rejectInvite | 拒绝对方发送的邀请 |
| cancelInvite | 取消自己发送的邀请 |
| CloudroomVideoMgrCallback | 回调 | inviteSuccess | 发送邀请成功 |
| inviteFail | 发送邀请失败 |
| cancelInviteSuccess | 取消自己发送的邀请成功 |
| cancelInviteFail | 取消自己发送的邀请失败 |
| acceptInviteSuccess | 接受对方发送的邀请成功 |
| acceptInviteFail | 接受对方发送的邀请失败 |
| rejectInviteSuccess | 拒绝对方发送的邀请成功 |
| rejectInviteFail | 拒绝对方发送的邀请失败 |
| notifyInviteIn | 通知有人邀请 |
| notifyInviteAccepted | 通知邀请被对方接受 |
| notifyInviteRejected | 通知邀请被对方拒绝 |
| notifyInviteCanceled | 通知有邀请被取消 |

**房间管理**

| **组件** | **方式** | **接口** | **描述** |
| --- | --- | --- | --- |
| CloudroomVideoMgr | 主调 | createMeeting | 创建视频房间 |
| destroyMeeting | 销毁视频房间 |
| CloudroomVideoMeeting | 主调 | enterMeeting | 使用房间ID和密码（可为空）进入指定的视频房间 |
| exitMeeting | 离开房间 |
| destroyMeeting | 结束房间 |
| switchToPage | 功能页切换 |
| getCurrentMainPage | 获取当前主功能区 |
| getCurrentSubPage | 获取当前子页面 |
| CloudroomVideoMgrCallback | 回调 | createMeetingSuccess | 创建房间成功响应 |
| createMeetingFail | 创建房间失败响应 |
| destroyMeetingRslt | 结束房间响应 |
| CloudroomVideoCallback | 回调 | enterMeetingRslt | 进入房间结果 |
| stopMeetingRslt | 通知结束视频房间结果 |
| setNickNameRsp | 设置房间成员昵称的结果 |
| notifyNickNameChanged | 某用户改变了昵称(改昵称的用户自身不会接收到此通知) |
| userEnterMeeting | 某用户进入了房间 |
| userLeftMeeting | 某用户离开了房间 |
| meetingDropped | 通知从房间里掉线了，收到该通知后可以调用enterMeeting尝试重新入会 |
| meetingStoped | 房间已被结束 |
| netStateChanged | 网络变化通知 |
| notifySwitchToPage | 通知主功能页切换 |

**音频管理**

| **组件** | **方式** | **接口** | **描述** |
| --- | --- | --- | --- |
| CloudroomVideoMeeting | 主调 | setAudioCfg | 设置麦克风音频参数 |
| getAudioCfg | 获取音频参数 |
| setMicVolumeScaling | 设置麦克风音量增益 |
| getMicEnergy | 获取用户说话声音大小 |
| openMic | 打开用户的麦克风 |
| closeMic | 关闭用户的麦克风 |
| getAudioStatus | 获取用户的麦状态 |
| getMicVolume | 获取麦克风音量大小 |
| setSpeakerVolume | 设置本地扬声器音量 |
| getSpeakerVolume | 获取本地扬声器音量 |
| setSpeakerMute | 设置播放是否静音 |
| getSpeakerMute | 获取播放是否静音 |
| setSpeakerOut | 设置外放状态 |
| getSpeakerOut | 获取外放状态 |
| setAllAudioClose | 关闭所有用户的麦克风 |
| CloudroomVideoCallback | 主调 | audioDevChanged | 通知本地音频设备有变化 |
| audioStatusChanged | 通知音频状态变化 |
| micEnergyUpdate | 通知用户的说话声音强度更新 |

**视频管理**

| **组件** | **方式** | **接口** | **描述** |
| --- | --- | --- | --- |
| CloudroomVideoMeeting | 主调 | getAllVideoInfo | 获取用户所有的摄像头信息 |
| createCustomVideoDev | 创建自定义摄像头 |
| destroyCustomVideoDev | 消毁自定义摄像头 |
| inputCustomVideoDat | 输入摄像头图像数据 |
| setVideoCfg | 设置摄像头参数 |
| getVideoCfg | 获取设置的摄像头参数 |
| setLocVideoAttributes | 设置某个摄像头私有参数 |
| getWatchableVideos | 获取房间内所有可观看的摄像头 |
| openVideo | 打开用户的摄像头，以便本地、远端显示视频图像 |
| closeVideo | 关闭用户的摄像头 |
| getVideoStatus | 获取用户的摄像头状态 |
| getDefaultVideo | 获取指定用户的默认摄像头 |
| setDefaultVideo | 设置默认的摄像头 |
| getLocalVideoParams | 获取指定本地摄像头参数 |
| setLocalVideoParam | 设置指定本地摄像头参数 |
| getVideoWallMode | 获取视频墙当前分屏模式 |
| setVideoWallMode | 设置视频墙分屏模式 |
| setMainVideo | 设置当前哪个用户为主视频 |
| getMainVideo | 获取当前哪个用户为主视频 |
| CloudroomVideoCallback | 回调 | openVideoRslt | 打开摄像头设备操作结果 |
| videoStatusChanged | 视频状态变化 |
| videoDevChanged | 通知用户的视频设备有变化 |
| notifyVideoData | 通知用户有新的视频数据 |
| defVideoChanged | 通知用户的视频默认设备有变化 |
| notifyVideoWallMode | 通知视频分屏模式切换 |
| notifyMainVideoChanged | 通知主视频更改 |

**屏幕共享**

| **组件** | **方式** | **接口** | **描述** |
| --- | --- | --- | --- |
| CloudroomVideoMeeting | 主调 | isScreenShareStarted | 获取屏幕共享是否已开启 |
|  |  |
| getScreenShareCfg | 获取屏幕共享配置 |
| setScreenShareCfg | 设置屏幕共享配置 |
| startScreenShare | 开始屏幕共享 |
| stopScreenShare | 停止屏幕共享 |
| isEnableOtherMark | 获取是否允许他人标注屏幕 |
| enableOtherMark | 允许他人标注屏幕 |
| CloudroomVideoCallback | 回调 | startScreenShareRslt | 开启屏幕共享的响应事件 |
| stopScreenShareRslt | 停止屏幕共享的响应事件 |
| notifyScreenShareStarted | 通知屏幕共享开始 |
| notifyScreenShareStopped | 通知屏幕共享停止 |
| startScreenMarkRslt | 开始屏幕标注结果 |
| stopScreenMarkRslt | 停止屏幕标注结果 |
| notifyScreenMarkStarted | 通知屏幕标注开始 |
| notifyScreenMarkStopped | 通知屏幕标注停止 |

**影音共享**

| **组件** | **方式** | **接口** | **描述** |
| --- | --- | --- | --- |
| CloudroomVideoMeeting | 主调 | setMediaCfg | 配置远程影音共享时，图像质量参数 |
| getMediaCfg | 获取影音共享图像质量参数 |
| startPlayMedia | 播放影音文件 |
| pausePlayMedia | 暂停播放影音 |
| stopPlayMedia | 停止影音播放 |
| setMediaPlayPos | 设置播放进度 |
| getAllFilesInMediaPath | 取得影音文件夹下的所有可播放文件 |
| getMediaInfo | 取得影音文件信息 |
| setMediaVolume | 设置影音播放的音量 |
| getMediaVolume | 读取影音播放的音量 |
| CloudroomVideoCallback | 回调 | notifyMediaOpened | 通知影音文件打开 |
| notifyMediaStart | 通知影音开始播放 |
| notifyMediaPause | 通知影音暂停播放 |
| notifyMediaStop | 通知影音播放停止 |
| notifyPlayPosSetted | 通知播放进度已设置完成 |
| notifyMemberMediaData | 通知影音帧数据已解码完毕 |
| notifyAudioPCMDat | 通知语音PCM数据 |

**本地录制/本地直播**

| **组件** | **方式** | **接口** | **描述** |
| --- | --- | --- | --- |
| CloudroomVideoMeeting | 主调 | createLocMixer | 创建本地混图器 |
| updateLocMixerContent | 更新本地混图器内容 |
| destroyLocMixer | 消毁本地混图器 |
| getLocMixerState | 获取本地混图器状态 |
| addLocMixerOutput | 开启本地录制、开启直播推流 |
| rmLocMixerOutput | 停止本地录制、直播推流 |
| getAllRecordFiles | 取得所有录制文件信息 |
| addFileToRecordMgr | 添加本地文件到录制文件管理中 |
| removeFromFileMgr | 删除本地的录制文件，上传中的文件会被取消上传 |
| uploadRecordFile | 上传录制文件 |
| uploadRecordFile | 上传录制文件到服务器指定位置 |
| cancelUploadRecordFile | 取消上传中的录制文件 |
| playbackRecordFile | 回放录制文件 |
| CloudroomVideoCallback | 回调 | uploadRecordFileErr | 上传录制文件错误通知 |
| cancelUploadRecordFileErr | 取消上传错误通知 |
| locMixerStateChanged | 本地混图器状态变化通知 |
| locMixerOutputInfo | 本地录制文件、本地直播信息通知 |
| notifyRecordFileStateChanged | 通知录制文件状态更改 |
| uploadRecordFileSuccess | 通知录制文件上传成功 |
| notifyRecordFileUploadProgress | 通知录制文件上传进度 |

**云端录制/互动直播**

| **组件** | **方式** | **接口** | **描述** |
| --- | --- | --- | --- |
| CloudroomVideoMeeting | 主调 | startSvrMixer | 开始云端录制、云端直播 |
| updateSvrMixerContent | 更新云端录制、云端直播内容 |
| stopSvrMixer | 停止云端录制、云端直播 |
| getSvrMixerState | 获取云端录制、云端直播状态 |
| CloudroomVideoCallback | 回调 | svrMixerStateChanged | 云端录制、云端直播状态变化通知 |
| svrMixerCfgChanged | 云端录制、云端直播内容变化通知 |
| svrMixerOutputInfo | 云端录制文件、云端直播信息变化通知 |

**电子白板/文档共享**

| **组件** | **方式** | **接口** | **描述** |
| --- | --- | --- | --- |
| CloudroomVideoMeeting | 主调 | createBoard | 创建电子白板 |
| closeBoard | 关闭电子白板 |
| initBoardPageDat | 初始化白板指定页数据 |
| createElementID | 创建图元标识 |
| addBoardElement | 添加图元信息 |
| modifyBoardElement | 修改图元信息 |
| delBoardElement | 删除图元 |
| setMouseHotSpot | 设置鼠标热点信息 |
| setBoardCurPageNo | 白板翻页 |
| listNetDiskDocFile | 查询服务器指定目录下的内容（子目录名、上传的文档源文件及文档转换状态） |
| uploadDocFileToNetDisk | 上传并转换文档 |
| downloadNetDiskDocFile | 下载源始文档 |
| deleteNetDiskDocFile | 删除文档 |
| getNetDiskDocFilePageInfo | 获取文档的转换信息 |
| downloadNetDiskDocFilePage | 获取文档的转换信息 |
| setPicResource | 将图片资源设置给sdk |
| CloudroomVideoCallback | 回调 | notifyInitBoards | SDK入会后通知房间中已经存在的白板列表 |
| notifyInitBoardPageDat | 初始化白板页数据 |
| notifyCreateBoard | 通知创建白板 |
| notifyCloseBoard | 通知关闭白板 |
| notifyAddBoardElement | 通知添加图元信息 |
| notifyModifyBoardElement | 通知图元信息被修改 |
| notifyDelBoardElement | 通知图元被删除 |
| notifyMouseHotSpot | 通知鼠标热点消息 |
| notifyBoardCurPageNo | 通知白板翻页 |
| listNetDiskDocFileRslt | 通知查询文档列表结果 |
| getNetDiskDocFilePageInfoRslt | 通知查询文档转换结果 |
| deleteNetDiskDocFileRslt | 通知删除文档结果 |
| notifyNetDiskDocFileTrsfProgress | 通知文档传输进度 |

**Http文件组件**

| **组件** | **方式** | **接口** | **描述** |
| --- | --- | --- | --- |
| CloudroomHttpFileMgr | 主调 | startMgr | 启动Http文件文件组件运行 |
| stopMgr | 停止Http文件文件组件运行 |
| getAllTransferInfos | 获取本地上传、下载文件信息 |
| startTransferFile | 开始下载/上传文件 |
| cancelFileTransfer | 取消传输 |
| rmTransferInfo | 删除传输记录及相关文件 |
| CloudroomHttpFileMgrCallback | 回调 | fileStateChanged | 通知用户文件状态更改 |
| fileProgress | 通知用户文件的传输进度 |
| fileFinished | 通知用户文件传输结束 |

**接口函数详情**

云屋SDK的方法/回调接口的相关内容分别实现在基础组件， 房间组件， 视频房间组件 ， 房间内可视化UI显示组件， 队列组件， Http文件传输组件六个组件中

1. 基础组件（CloudroomVideoSDK）的接口函数
2. 房间组件（CloudroomVideoMgr）的接口函数
3. 视频房间组件（CloudroomVideoMeeting）的接口函数
4. 队列组件（CloudroomQueue）的接口函数
5. Http文件传输组件（CloudroomHttpFileMgr）的接口函数

**1. 基础组件（CloudroomVideoSDK）的接口函数**

CloudroomVideoSDK 是基础控件，负责整个SDK的初始化和反初始化操作。整个程序的生命过程中只能有一个实例

**String GetCloudroomVideoSDKVer()**

* **功能**: 获取sdk的版本信息
* **返回值**: 返回云屋SDK版本号

**CRVIDEOSDK\_ERR\_DEF init(Context context, SdkInitDat dat)**

* **功能**: SDK初始化
* **返回值**: CRVIDEOSDK\_NOERR为成功操作，其他详见定义CRVIDEOSDK\_ERR\_DEF

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| context | Context | 程序上下文，不可为空 |
| dat | SdkInitDat | SDK配置、临时文件存放位置，不可为空 |

**void uninit()**

* **功能**: SDK反初始化
* **返回值**: 无

**String serverAddr()**

* **功能**: 获取服务器地址
* **返回值**: 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;）

**void setServerAddr(String serverList)**

* **功能**: 设置服务器地址
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| serverList | String | 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;） |

**void setFileUploadRate(int maxBytePerSec)**

* **功能**: 文件上传的流量控制
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| maxBytePerSec | int | 每秒上传的最大字节数 。-1代表不进行流控 |

**void setFileDownloadRate(int maxBytePerSec)**

* **功能**: 文件下载的流量控制
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| maxBytePerSec | int | 每秒下载的最大字节数。-1代表不进行流控 |

**2. 房间组件（CloudroomVideoMgr）的接口函数**

CloudroomVideoMgr 是房间组件，实现登录、呼叫、房间创建销毁、透明传输等功能。 整个程序的生命过程中只能有一个实例。

**void registerCallBack(CloudroomVideoMgrCallback callBack)**

* **功能**: 注册监听回调
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callBack | CloudroomVideoMgrCallback | 回调对象 |

**void unregisterCallBack(CloudroomVideoMgrCallback callBack)**

* **功能**: 反注册监听回调
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callBack | CloudroomVideoMgrCallback | 反注册的回调对象 |

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callBack | CloudroomVideoMgrCallback | 回调对象 |

**void login(String appID, String appSecret, String nickName, String userID, String userAuthCode, String cookie)**

* **功能**: 登陆服务器
* **返回值**: 无

登录成功则触发回调loginSuccess ， 登录失败则触发回调loginFail

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| appID | String | 项目授权appID |
| appSecret | String | 项目授权appID对应密码 |
| nickName | String | 昵称 |
| userID | String | 用户ID。和云屋系统无关，业务方自由填写，保证唯一性即可（同一appID下，具有唯一性即可） |
| userAuthCode | String | 默认填空。只有开启第三方认证才需要填写。（开启第三方认证时，云屋SDK服务器将连接提前配好的业务方服务器进行实时验证。） |
| cookie | String | 可缺省。详细介绍见关键词 |

**void loginByToken(String token, string nickName, String userID, String userAuthCode, String cookie)**

* **功能**: Token登录
* **返回值**:无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| token | String | Token鉴权码 |
| nickName | String | 昵称 |
| userID | String | 用户ID。和云屋系统无关，业务方自由填写，保证唯一性即可（同一appID下，具有唯一性即可） |
| userAuthCode | String | 默认填空。只有开启第三方认证才需要填写。（开启第三方认证时，云屋SDK服务器将连接提前配好的业务方服务器进行实时验证。） |
| cookie | String | 可缺省。详细介绍见关键词 |

登陆成功则回调loginSuccess,登陆失败则回调loginFail

**void logout()**

* **功能**: 反注册本次登陆
* **返回值**: 无

**int getUserAuthErrCode()**

* **功能**: 获取第三方认证错误码
* **返回值**: 错误码

**String getUserAuthErrDesc()**

* **功能**: 获取第三方认证错误描述
* **返回值**: 错误描述

**void updateToken(String token))**

* **功能**: 更新Token(在Token即将失效的时候更新Token延长有限时间)
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| token | String | Token鉴权码 |

**void setDNDStatus(int DNDStatus, String cookie)**

* **功能**: 设置免打扰状态， 如果使用到了SDK的队列功能，则入会后需调用此接口，设置自己的为免打扰状态，防止系统再次推送自己
* **返回值**: 无

设置免打扰状态成功则触发回调setDNDStatusSuccess ， 设置免打扰状态失败则触发回调setDNDStatusFail

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| DNDStatus | int | 0代表关闭免打扰， 其它值代表开启免打扰，含义自由定义 |
| cookie | String | 自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void getUserStatus(String userID, String cookie)**

* **功能**: 获取企业下指定用户的在线状态
* **返回值**: 无

回调函数getUserStatusRslt

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 具体用户的账户ID |
| cookie | String | 自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void getUserStatus(String cookie)**

* **功能**: 将获取企业下所有用户在线状态（包括呼叫房间状态、免打扰状态）
* **返回值**: 无

回调函数getUserStatusRslt

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| cookie | String | 自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void startUserStatusNotify(String cookie)**

* **功能**: 开启用户的状态推送 。开启后，企业下所有用户状态有变化时(包括呼叫房间状态、免打扰状态)，都会收到通知。开启后，用户量越大消息量越大，所以请按需开启。
* **返回值**: 无

回调函数startStatusPushRslt

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| cookie | String | 自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void stopUserStatusNotify(String cookie)**

* **功能**: 关闭用户的状态推送
* **返回值**: 无

回调函数stopStatusPushRslt

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| cookie | String | 自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void createMeeting(boolean createPswd, String cookie)**

* **功能**: 创建视频房间
* **返回值**: 无

创建房间成功则触发回调createMeetingSuccess，创建房间失败则触发createMeetingFail

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| createPswd | boolean | 是否创建房间密码，密码由系统自动生成 |
| cookie | String | 自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void destroyMeeting(int meetID, String cookie)**

* **功能**: 销毁视频房间
* **返回值**: 无

回调函数destroyMeetingRslt

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| meetID | int | 房间号） |
| cookie | String | 自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**String call(String calledUserID, MeetInfo meetInfo, String userExtDat, String cookie)**

* **功能**: 发起呼叫，邀请用户参加视频会话， 呼叫时，对方迟迟不响应，30秒后系统自动结束呼叫
* **返回值**: 本次呼叫标识码（呼叫ID）

发起呼叫成功则触发回调callSuccess，发起呼叫失败则触发callFail

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| calledUserID | String | 被叫用户的账户ID |
| meetInfo | MeetInfo | 房间信息，空时代表无房间信息，应由被叫创建或提供房间信息 |
| usrExtDat | String | 自定义扩展参数 |
| cookie | String | 自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void acceptCall(String callID, MeetInfo meetInfo, String usrExtDat, String 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

接受对方发起的视频请求成功则触发回调acceptCallSuccess，接受对方发起的视频请求失败则触发acceptCallFail

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callID | String | 呼叫标识码 |
| meetInfo | MeetInfo | 房间信息 |
| usrExtDat | String | 自定义扩展参数 |
| cookie | String | 自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void rejectCall(String callID, String usrExtDat, String cookie)**

* **功能**: 拒绝对方的视频请求
* **返回值**: 无

拒绝对方发起的视频请求成功则触发回调rejectCallSuccess，拒绝对方发起的视频请求失败则触发rejectCallFail

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callID | String | 呼叫标识码 |
| usrExtDat | String | 自定义扩展参数 |
| cookie | String | 自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void hangupCall(String callID, String usrExtDat, String cookie)**

* **功能**: 挂断正在进行的视频呼叫或视频通话
* **返回值**: 无

挂断视频成功则触发回调hangupCallSuccess，挂断视频失败则触发hangupCallFail

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callID | String | 呼叫标识码 |
| usrExtDat | String | 自定义扩展参数 |
| cookie | String | 自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**String sendCmd(String targetUserId, String data)**

* **功能**: 发送小块数据（50K以内），此接口不能被cancelSend, 一次性发送，不会有进度通知
* **返回值**: 分配的任务ID

回调函数sendCmdRlst

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| targetUserId | String | 目标用户ID |
| data | String | 发送的数据，base64编码 |

**String sendBuffer(String targetUserId, String data)**

* **功能**: 发送大块数据 ，分块发送，进度通知事件sendProgress, 调用cancelSend 取消发送
* **返回值**: 分配的任务ID

回调函数sendBufferRlst

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| targetUserId | String | 目标用户ID |
| data | String | 发送的数据，base64编码 |

**String sendFile(String targetUserId, String fileName)**

* **功能**: 发送文件 ，分块发送，进度通知事件sendProgress，调用cancelSend取消发送
* **返回值**: 分配的任务ID

回调函数sendFileRlst

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| targetUserId | String | 目标用户ID |
| fileName | String | 需要发送的文件名 |

**void cancelSend(String taskID)**

* **功能**: 取消数据发送
* **返回值**: 无

回调函数cancelSendRlst

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| taskID | String | 任务ID |

**void setMarkText(String videoFilePathName, int timestamp, String markText)**

* **功能**: 添加视频打点信息
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| videoFilePathName | NSString | 视频文件全路径 |
| timestamp | int | 时间戳 |
| markText | NSString | 打点信息 |

**void removeMarkText(String videoFilePathName, int timestamp)**

* **功能**: 移除视频打点信息
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| videoFilePathName | NSString | 视频文件全路径 |
| timestamp | int | 时间戳 |

**String getAllMarks(String videoFilePathName)**

* **功能**: 获取所有打点信息
* **返回值**: 打点信息

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| videoFilePathName | NSString | 视频文件全路径 |

**String getVideoMarkFile(String videoFilePathName)**

* **功能**: 获取打点文件的绝对文件路径名
* **返回值**: 打点文件的绝对文件路径名

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| videoFilePathName | NSString | 视频文件全路径 |

**通知回调函数**

CloudroomVideoMgrCallback是登录、呼叫、房间创建销毁、透明传输等功能的回调接口

**void loginSuccess(String usrID, String cookie)**

* **功能**: 登录成功响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| usrID | String | 用户账户 |
| cookie | String | 自定义用户数据 |

**void loginFail(CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 登录失败响应
* **返回值**: 无

如果启用了第三方鉴权，并在鉴权失败时（sdkErr为CRVIDEOSDK\_CUSTOMAUTH\_FAILED），可以通过getUserAuthErrCode、getUserAuthErrDesc获取详细原因。

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果代码 |
| cookie | String | 自定义用户数据 |

**void setDNDStatusSuccess(String cookie)**

* **功能**: 客户端设置免打扰状态操作成功响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| cookie | String | 自定义用户数据 |

**void setDNDStatusFail(CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 客户端设置免打扰状态操作失败响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果代码 |
| cookie | String | 自定义用户数据 |

**void getUserStatusRslt(CRVIDEOSDK\_ERR\_DEF sdkErr, ArrayList<UserStatus> userStatus, String cookie)**

* **功能**: 获取企业内所有用户在线状态响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果代码 |
| usersStatus | ArrayList<UserStatus> | 用户在线状态信息列表 |
| cookie | String | 自定义用户数据 |

**void startStatusPushRslt(CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 启动用户状态推送响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果代码 |
| cookie | String | 自定义用户数据 |

**void stopStatusPushRslt(CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 结束用户状态推送响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果代码 |
| cookie | String | 自定义用户数据 |

**void notifyUserStatus(UserStatus userStatus)**

* **功能**: 某个用户状态变化通知
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userStatus | UserStatus | 单个用户在线状态信息 |

**void lineOff(CRVIDEOSDK\_ERR\_DEF sdkErr)**

* **功能**: SDK通知自己掉线
* **返回值**: 无

**void notifyTokenWillExpire()**

* **功能**: 通知Token即将失效,有效时间剩余30s左右(可调用updateToken更新Token)
* **返回值**: 无

**void createMeetingSuccess(MeetInfo meetInfo, String cookie)**

* **功能**: 创建房间成功响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| meetInfo | MeetInfo | 房间信息 |
| cookie | String | 自定义用户数据 |

**void createMeetingFail(CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 创建房间成功响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作失败代码 |
| cookie | String | 自定义用户数据 |

**void destroyMeetingRslt(CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 结束房间响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果代码，成功为CRVIDEOSDK\_NOERR |
| cookie | String | 自定义用户数据 |

**void callSuccess(String callID, String cookie)**

* **功能**: 呼叫他人操作成功响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callID | String | 呼叫全局标识 |
| meetID | int | 房间号 |
| meetPswd | String | 房间密码 |
| cookie | String | 自定义用户数据 |

**void callFail(String callID, CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 呼叫他人操作失败响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callID | String | 呼叫全局标识 |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作失败代码 |
| cookie | String | 自定义用户数据 |

**void acceptCallSuccess(String callID, String cookie)**

* **功能**: 接受他人呼叫操作成功响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callID | String | 呼叫全局标识 |
| cookie | String | 自定义用户数据 |

**void acceptCallFail(String callID, CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 接受他人呼叫操作失败响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callID | String | 呼叫全局标识 |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作失败代码 |
| cookie | String | 自定义用户数据 |

**void rejectCallSuccess(String callID, String cookie)**

* **功能**: 拒绝他人的呼叫成功响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callID | String | 呼叫全局标识 |
| cookie | String | 自定义用户数据 |

**void rejectCallFail(String callID, CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 拒绝他人的呼叫失败响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callID | String | 呼叫全局标识 |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作失败代码 |
| cookie | String | 自定义用户数据 |

**void hangupCallSuccess(String callID, String cookie)**

* **功能**: 挂断呼叫操作成功响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callID | String | 呼叫全局标识 |
| cookie | String | 自定义用户数据 |

**void hangupCallFail(String callID, CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 拒绝他人呼叫操作失败响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callID | String | 呼叫全局标识 |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作失败代码 |
| cookie | String | 自定义用户数据 |

**void notifyCallIn(String callID, MeetInfo meetInfo, String callerID, String usrExtDat)**

* **功能**: 服务端通知被邀请
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callID | String | 呼叫全局标识 |
| meetInfo | MeetInfo | 房间信息，空时代表无房间信息，应由被叫创建或提供房间信息 |
| callerID | String | 呼叫人员的标识ID |
| usrExtDat | String | 自定义扩展参数 |

**void notifyCallAccepted(String callID, MeetInfo meetInfo, String usrExtDat)**

* **功能**: 服务端通知呼叫邀请被接受回调
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callID | String | 呼叫全局标识 |
| meetInfo | MeetInfo | 房间信息，空时代表无房间信息，应由被叫创建或提供房间信息 |
| usrExtDat | String | 自定义扩展参数 |

**void notifyCallRejected(String callID, CRVIDEOSDK\_ERR\_DEF reason, String usrExtDat)**

* **功能**: 服务端通知呼叫邀请被拒绝回调
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callID | String | 呼叫全局标识 |
| reason | CRVIDEOSDK\_ERR\_DEF | 呼叫被对方拒绝的原因代码 |
| usrExtDat | String | 自定义扩展参数 |

**void notifyCallhangup(String callID, String usrExtDat)**

* **功能**: SDK通知呼叫被挂断
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callID | String | 呼叫全局标识 |
| usrExtDat | String | 自定义扩展参数 |

**void sendCmdRlst(String taskID, CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 发送数据时，SDK通知发送结果
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| taskID | String | 发送任务id |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果代码 |
| cookie | String | 自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void sendBufferRlst(String taskID, CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 发送数据时，SDK通知发送结果
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| taskID | String | 发送任务id |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果代码 |
| cookie | String | 自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void sendFileRlst(String taskID, String fileName, CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 发送数据时，SDK通知发送结果
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| taskID | String | 发送任务id |
| fileName | String | 文件名 |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果代码 |
| cookie | String | 自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void sendProgress(String taskID, int sendedLen, int totalLen, String cookie)**

* **功能**: 发送数据时，SDK通知发送进度
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| taskID | String | 发送任务id |
| sendedLen | int | 已发送的数据长度 |
| totalLen | int | 需要发送的总长度 |
| cookie | String | 自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void cancelSendRlst(String taskID, CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 取消发送操作响应
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| taskID | String | 发送任务id |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果代码 |
| cookie | String | 自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void notifyCmdData(String sourceUserId, String data)**

* **功能**: SDK通知收到小块数据
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sourceUserId | String | 数据来源 |
| data | String | 数组数据 |

**void notifyBufferData(String sourceUserId, String data)**

* **功能**: SDK通知收到大块数据
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sourceUserId | String | 数据来源 |
| data | String | 数组数据 |

**void notifyFileData(String sourceUserId, String tmpFile, String orgFileName)**

* **功能**: SDK通知收到文件数据
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sourceUserId | String | 数据来源 |
| tmpFile | String | 临时文件，不需要时，请移除或删除对应文件 |
| orgFileName | String | 源始文件名 |

**void notifyCancelSend(String taskID)**

* **功能**: 通知发来的数据/文件被取消
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| taskID | 发送的任务ID | 数据来源 |

**String invite(String invitedUserID, String usrExtDat, String cookie)**

* **功能**：发送邀请
* **返回值**:返回inviteID(每个邀请对应的唯一ID)
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| inviteeUserID | String | 受邀者用户ID |
| usrExtDat | String | 用户扩展数据 |
| cookie | String | 详细介绍见关键词 |

**void acceptInvite(String inviteID, String usrExtDat, String cookie)**

* **功能**：接受对方的邀请
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| inviteID | String | 邀请ID |
| usrExtDat | String | 用户扩展数据 |
| cookie | String | 详细介绍见关键词 |

**void rejectInvite(String inviteID, String usrExtDat, String cookie)**

* **功能**：拒绝对方的邀请
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| inviteID | String | 邀请ID |
| usrExtDat | String | 用户扩展数据 |
| cookie | String | 详细介绍见关键词 |

**void cancelInvite(String inviteID, String usrExtDat, String cookie)**

* **功能**：取消自己发送的邀请
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| inviteID | String | 邀请ID |
| usrExtDat | String | 用户扩展数据 |
| cookie | String | 详细介绍见关键词 |

**void inviteSuccess(String inviteID, String cookie)**

* **功能**：发送邀请成功
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| inviteID | String | 邀请ID |
| cookie | String | 详细介绍见关键词 |

**void inviteFail(String inviteID, CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**：发送邀请失败
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| inviteID | String | 邀请ID |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作失败代码，定义见ERR\_DEF |
| cookie | String | 详细介绍见关键词 |

**void cancelInviteSuccess(String inviteID, String cookie)**

* **功能**：取消自己发送的邀请成功
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| inviteID | String | 邀请ID |
| cookie | String | 详细介绍见关键词 |

**void cancelInviteFail(String inviteID, CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**：取消自己发送的邀请失败
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| inviteID | String | 邀请ID |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作失败代码，定义见ERR\_DEF |
| cookie | String | 详细介绍见关键词 |

**void acceptInviteSuccess(String inviteID, String cookie)**

* **功能**：接受对方的邀请成功
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| inviteID | String | 邀请ID |
| cookie | String | 详细介绍见关键词 |

**void acceptInviteFail(String inviteID, CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**：接受对方的邀请失败
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| inviteID | String | 邀请ID |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作失败代码，定义见ERR\_DEF |
| cookie | String | 详细介绍见关键词 |

**void rejectInviteSuccess(String inviteID, String cookie)**

* **功能**：拒绝对方的邀请成功
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| inviteID | String | 邀请ID |
| cookie | String | 详细介绍见关键词 |

**void rejectInviteFail(String inviteID, CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**：拒绝对方的邀请失败
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| inviteID | String | 邀请ID |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作失败代码，定义见ERR\_DEF |
| cookie | String | 详细介绍见关键词 |

**void notifyInviteIn(String inviteID, String inviterUsrID, String usrExtDat)**

* **功能**：通知有人邀请
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| inviteID | String | 邀请ID |
| inviterUsrID | String | 邀请者用户ID |
| usrExtDat | String | 用户扩展数据 |

**void notifyInviteAccepted(String inviteID, String usrExtDat)**

* **功能**：通知邀请被接受
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| inviteID | String | 邀请ID |
| usrExtDat | String | 用户扩展数据 |

**void notifyInviteRejected(String inviteID, CRVIDEOSDK\_ERR\_DEF reason, String usrExtDat)**

* **功能**：通知邀请被拒绝
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| inviteID | String | 邀请ID |
| reason | CRVIDEOSDK\_ERR\_DEF | 邀请被拒绝原因，定义见ERR\_DEF |
| usrExtDat | String | 用户扩展数据 |

**void notifyInviteCanceled(String inviteID, CRVIDEOSDK\_ERR\_DEF reason, String usrExtDat)**

* **功能**：通知邀请被取消
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| inviteID | String | 邀请ID |
| reason | CRVIDEOSDK\_ERR\_DEF | 邀请被取消原因，定义见ERR\_DEF |
| usrExtDat | String | 用户扩展数据 |

**3. 视频房间组件（CloudroomVideoMeeting）的接口函数**

CloudroomVideoMeeting 是房间核心控件，实现通话建立、音频采集播入、视频采集编解码、屏幕共享、录制、影音播放等功能，整个程序的生命过程中只能有一个实例

**void registerCallBack(CloudroomVideoCallback callBack)**

* **功能**: 注册监听回调
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callBack | CloudroomVideoCallback | 回调对象 |

**void unregisterCallBack(CloudroomVideoCallback callBack)**

* **功能**: 反注册监听回调
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callBack | CloudroomVideoCallback | 反注册的回调对象 |

**void enterMeeting(int meetID, String pswd, String userID, String nikeName)**

* **功能**: 使用房间ID和密码（可为空）进入指定的视频房间
* **返回值**: 无

回调函数enterMeetingRslt

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| meetID | int | 视频房间ID |
| pswd | String | 本次房间中的密码 |
| userID | String | 用户账号 |
| nickName | String | 用户昵称 |

**void exitMeeting()**

* **功能**: 离开房间 ，调用此接口离开会话时，其他会话用户会收到userLeftMeeting的消息通知
* **返回值**: 无

回调函数userLeftMeeting

**ArrayList<MemberInfo> getAllMembers()**

* **功能**: 获取所有用户的信息
* **返回值**: ArrayList<MemberInfo>-所有用户的信息

**MemberInfo getMemberInfo(String userId)**

* **功能**: 获取某个用户的信息
* **返回值**: MemberInfo - 用户的信息

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userId | String | 用户ID |

**String getNickName(String userId)**

* **功能**: 获取某个用户的昵称
* **返回值**: 用户的昵称

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userId | String | 用户ID |

**void setNickName(String userID, String nickName)**

* **功能**: 设置某个用户的昵称， 调用此接口如果设置成功，其他会话用户会收到 notifyNickNameChanged
* **返回值**: 无

回调函数setNickNameRsp

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userId | String | 用户ID |
| nickname | String | 新的用户昵称 |

**boolean isUserInMeeting(String userID)**

* **功能**: 判断某个用户是否在房间中
* **返回值**: 用户是否在房间中

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userId | String | 用户ID |

**void setAudioCfg(AudioCfg cfg)**

* **功能**: 设置麦克风音频参数
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| cfg | AudioCfg | 音频参数 |

**AudioCfg getAudioCfg()**

* **功能**: 获取音频参数
* **返回值**: AudioCfg-音频参数

**boolean setMicVolumeScaling(int scale)**

* **功能**: 设置麦克风音量增益
* **返回值**: 是否成功

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| scale | int | 麦克风音量增益（范围：1-20） |

**int getMicEnergy(String userID)**

* **功能**: 获取用户说话声音大小
* **返回值**: 音量大小（0~10）

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 用户的ID |

**void openMic(String userID)**

* **功能**: 打开用户的麦克风， 打开麦克风会触发音频状态变化的回调函数audioStatusChanged 新状态参数先会进入到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
* **返回值**: 无

回调函数audioStatusChanged

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 用户的ID |

**void closeMic(String userID)**

* **功能**: 关闭用户的麦克风 ，关麦克风会触发音频状态变化的回调函数audioStatusChanged 新状态参数会变为ACLOSE 关麦操作是立即生效的，会立即停止采集
* **返回值**: 无

回调函数audioStatusChanged

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 用户的ID |

**ASTATUS getAudioStatus(String userID)**

* **功能**: 获取用户的麦状态
* **返回值**: ASTATUS-麦克风设备状态

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 用户的ID |

**int getMicVolume()**

* **功能**: 获取麦克风音量大小 ，音量范围类型 0-255
* **返回值**: 麦克风音量大小

**boolean setSpeakerVolume(int level)**

* **功能**: 设置本地扬声器音量
* **返回值**: 是否成功

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| level | int | 本地扬声器音量 |

**int getSpeakerVolume()**

* **功能**: 获取本地扬声器音量 ，读写属性，音量等级类型 0-255
* **返回值**: 本地扬声器音量

**void setSpeakerMute(boolean bMute)**

* **功能**: 设置播放是否静音
* **返回值**: 是否静音

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| bMute | boolean | 播放是否静音 |

**boolean getSpeakerMute()**

* **功能**: 获取播放是否静音
* **返回值**: 是否静音

**boolean setSpeakerOut(boolean speakerOut)**

* **功能**: 设置外放状态
* **返回值**: 是否成功

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| speakerOut | boolean | 是否外放 |

**boolean getSpeakerOut()**

* **功能**: 获取外放状态
* **返回值**: 是否外放

**void setAllAudioClose()**

* **功能**: 关闭所有用户的麦克风 ， 调用此接口后会话内其他人的麦克风会关闭，同时收到消息audioStatusChanged
* **返回值**: 无

回调函数audioStatusChanged

**ArrayList<UsrVideoInfo> getAllVideoInfo(String userId)**

* **功能**: 获取用户所有的摄像头信息
* **返回值**: ArrayList<UsrVideoInfo>-用户所有的摄像头信息

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 用户ID |

**short createCustomVideoDev(String camName, VIDEO\_FORMAT pixFmt, int width, int height, String extParams)**

* **功能**: 创建自定义摄像头（sdk最大支持5个)， 添加成功后与本地摄像头处理一致；getAllVideoInfo接口可以识别摄像头类型
* **返回值**: 返回值小于0， 则代表创建失败， 返回值大于等于0， 则代表摄像头编号

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| camName | String | 摄像头名称 |
| pixFmt | VIDEO\_FORMAT | 图像格式 |
| width | int | 图像宽度 |
| height | int | 图像高度 |
| extParams | String | 扩展参数，没有为空 |

**void destroyCustomVideoDev(short videoID)**

* **功能**: 消毁自定义摄像头
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| videoID | short | 摄像头id (createCustomVideoDev返回值） |

**void inputCustomVideoDat(short videoID, byte[] data, int timeStamp)**

* **功能**: 输入摄像头图像数据
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| videoID | short | 摄像头id (createCustomVideoDev返回值） |
| data | byte[] | 图像数据（请保证格式、和尺寸与摄像头匹配） |
| timeStamp | int | 保留字段，请填0 |

**void inputCustomVideoDat(short videoID, byte[] data, int timeStamp)**

* **功能**: 输入摄像头图像数据
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| videoID | short | 摄像头id (createCustomVideoDev返回值） |
| data | byte[] | 图像数据（请保证格式、和尺寸与摄像头匹配） |
| timeStamp | int | 保留字段，请填0 |

**void setVideoCfg(VideoCfg cfg)**

* **功能**: 设置摄像头参数
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| cfg | VideoCfg | 摄像头参数 |

**VideoCfg getVideoCfg()**

* **功能**: 获取设置的摄像头参数
* **返回值**: VideoCfg](TypeDefinitions.md#VideoCfg)-摄像头参数

**void setLocVideoAttributes(short videoID, VideoAttributes attributes)**

* **功能**: 设置某个摄像头私有参数
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| videoID | short | 本地摄像头id |
| attributes | VideoAttributes | 摄像头私有参数, |

**VideoAttributes getLocVideoAttributes(short videoID)**

* **功能**: 获取某个摄像头私有参数
* **返回值**: VideoAttributes-摄像头私有参数

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| videoID | short | 本地摄像头id |

**ArrayList<UsrVideoId> getWatchableVideos()**

* **功能**: 获取房间内所有可观看的摄像头
* **返回值**: ArrayList<UsrVideoId>-房间内所有可观看的摄像头

**void openVideo(String userID)**

* **功能**: 打开用户的摄像头，以便本地、远端显示视频图像
* **返回值**: 无

回调函数openVideoRslt, 调用打开和关闭麦克风接口都会触发对应用户的回调videoStatusChanged

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 用户的ID |

**void closeVideo(String userID)**

* **功能**: 关闭用户的摄像头 ，
* **返回值**: 无

调用打开和关闭麦克风接口都会触发对应用户的回调videoStatusChanged

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 用户的ID |

**VSTATUS getVideoStatus(String userID)**

* **功能**: 获取用户的摄像头状态
* **返回值**: VSTATUS-麦克风摄像头状态

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 用户的ID |

**short getDefaultVideo(String userID)**

* **功能**: 获取指定用户的默认摄像头 ，如果用户没有摄像头，返回0
* **返回值**: 摄像头ID

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 用户的ID |

**void setDefaultVideo(String userID, short videoID)**

* **功能**: 设置默认的摄像头 ，videoID 应该从getAllVideoInfo返回值中获取
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 用户的ID |
| videoID | short | 摄像头ID |

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| usrVideoID | UsrVideoId | 用户摄像头ID |

**String getLocalVideoParams(short videoID)**

* **功能**: 获取指定本地摄像头参数
* **返回值**: 指定本地摄像头参数,json格式，如{"摄像头参数名称":[可用参数列表]}

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| videoID | short | 摄像头ID |

**void setLocalVideoParam(short videoID, String param, String value)**

* **功能**: 设置指定本地摄像头参数
* **返回值**: 指定本地摄像头参数,json格式，如{"摄像头参数名称":[可用参数列表]}

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| videoID | short | 摄像头ID |
| param | String | 摄像头参数名称 |
| value | String | 摄像头参数值 |

**boolean isScreenShareStarted()**

* **功能**: 获取屏幕共享是否已开启
* **返回值**: 屏幕共享是否已开启

**string getScreenShareCfg()**

* **功能**: 获取屏幕共享配置
* **返回值**:屏幕共享配置，参见ScreenShareCfg

**void setScreenShareCfg(ScreenShareCfg cfg)**

* **功能**: 设置屏幕共享配置
* **返回值**:无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| string | ScreenShareCfg | 屏幕共享配置 |

**void startScreenShare()**

* **功能**: 开启屏幕共享
* **返回值**:无
* 注意事项:

+ 共享屏幕需要相应授权，请添加下面权限申请界面声明到主配置文件

<activity

android:name="com.cloudroom.screencapture.PermissionActivity"

android:configChanges="orientation|uiMode|screenLayout|screenSize|smallestScreenSize|locale|fontScale|keyboard|keyboardHidden|navigation"

android:launchMode="singleTop"

android:screenOrientation="sensor" >

</activity>

**void stopScreenShare()**

* **功能**: 停止屏幕共享
* **返回值**:无

**void startScreenMark()**

* **功能**: 开始屏幕共享标注
* **返回值**: 无

**void stopScreenMark()**

* **功能**: 停止屏幕共享标注
* **返回值**: 无

**void enableOtherMark(boolean enable)**

* **功能**：允许他人标注屏幕
* **返回值**:无
* **参数**:

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| enable | boolean | 是否允许他人标注 |

**boolean isEnableOtherMark()**

* **功能**：获取是否允许他人标注屏幕
* **返回值**: 是否允许他人标注
* **参数**:

**CRVIDEOSDK\_ERR\_DEF createLocMixer(String mixerID, MixerCfg mixerCfg, ArrayList< MixerContent> MixerContent)**

* **功能**: 创建本地混图器（用于本地录制、本地推流），当需要多个不同内容的录制、或直播时，就要创建多个混图器 。混图器开消比较大，多个同样图像的输出应该有一个混图器加上多个输出实现
* **返回值**: CRVIDEOSDK\_ERR\_DEF-操作结果码

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| mixerID | String | 混图器唯一标识 |
| mixerCfg | MixerCfg | 混图器规格配置 |
| mixerContents | ArrayList<MixerContent> | 混图器内容配置 |

**CRVIDEOSDK\_ERR\_DEF updateLocMixerContent(String mixerID, ArrayList< MixerContent> MixerContents>**

* **功能**: 更新本地混图器内容
* **返回值**: CRVIDEOSDK\_ERR\_DEF-操作结果码

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| mixerID | String | 混图器唯一标识 |
| MixerContents | ArrayList<MixerContent> | 混图器内容配置 |

**void destroyLocMixer(String mixerID)**

* **功能**: 消毁本地混图器 , 消毁本地混图器后， 基于此混图器的录制输出、直播推流输出将自动结束
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| mixerID | String | 混图器唯一标识 |

**MIXER\_STATE getLocMixerState(String mixerID)**

* **功能**: 获取本地混图器状态
* **返回值**: MIXER\_STATE-混图器状态

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| mixerID | String | 混图器唯一标识 |

**CRVIDEOSDK\_ERR\_DEF addLocMixerOutput(String mixerID, ArrayList<MixerOutPutCfg> mixerOutput)**

* **功能**: 开启本地录制、开启直播推流
* **返回值**: CRVIDEOSDK\_ERR\_DEF-操作结果码

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| mixerID | String | 混图器唯一标识 |
| mixerOutput | ArrayList<MixerOutPutCfg> | 本地输出对象 |

* 注意事项:
  + 可以一次添加多个输出，也可以依据业务需要稍后再添加新的输出
  + 多个输出不能同名、或同一url
  + 每个输出，有独立的信息通知，请见事件locMixerOutputInfo
  + 如果输出异常时，将自动停止, 停止本地录制、直播推流

**void rmLocMixerOutput(String mixerID, ArrayList<String> nameOrUrls)**

* **功能**: 停止本地录制、直播推流 , 所有输出停止后并不会消毁混图器，如果混图器不再需要请手工消毁
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| mixerID | String | 混图器唯一标识 |
| nameOrUrls | ArrayList<String> | 要停止的录制文件名或直播url列表 |

**ArrayList<RecordFileShow> getAllRecordFiles()**

* **功能**: 取得所有录制文件信息
* **返回值**: ArrayList<义RecordFileShow>所有录制文件信息

**int addFileToRecordMgr(String fileName, String filePath)**

* **功能**: 添加本地文件到录制文件管理中 , 第三方录制文件调用此接口后可进行本地回放和上传到视频服务器上，和自己录制的文件一样可以正常
* **返回值**: -1：本地文件不存在，0：成功，1：文件已经被添加过

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| filename | String | 文件名，不含路径 |
| filePath | String | 文件路径，不含文件名 |

**void removeFromFileMgr(String fileName)**

* **功能**: 删除本地的录制文件，上传中的文件会被取消上传 , 已上传完成的服务器文件不受影响
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| filename | String | 文件名，不含路径 |

**void uploadRecordFile(String fileName)**

* **功能**: 上传文件
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| filename | String | 文件名，不含路径 |

**void uploadRecordFile(String fileName, String svrPathFileName)**

* **功能**: 上传文件到服务器指定位置
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| filename | String | 文件名，不含路径 |
| svrPathFileName | String | 文件存放在服务器上的相对路径文件名（如/AA/BB/CC/test.mp4） |

**void cancelUploadRecordFile(String filename)**

* **功能**: 取消上传中的录制文件
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| filename | String | 文件名，不含路径 |

**void playbackRecordFile(String filename)**

* **功能**: 回放录制文件 , 可创建影音控件显示录制内容，功能同接口startPlayMedia，如果录制文件被加密，则只能使用playbackRecordFile来回放
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| filename | String | 文件名，不含路径 |

**CRVIDEOSDK\_ERR\_DEF startSvrMixer(Map< String, MixerCfg> mutiMixerCfgs, Map< String, ArrayList< MixerContent>> mutiMixerContents, Map<String, ArrayList<MixerOutPutCfg>> mutiMixerOutputs)**

* **功能**: 开始云端录制、云端直播
* **返回值**: CRVIDEOSDK\_ERR\_DEF-操作结果

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| mutiMixerCfgs | Map<String, MixerCfg> | 服务器混图配置 |
| mutiMixerContents | Map<String, ArrayList<MixerContent>> | 服务器混图内容 |
| mutiMixerOutputs | ArrayList<MixerOutPutCfg>> | 服务器输出 |

* 注意事项:
  + 启动云端功能有一定耗时，请关注混图器的状态变化事件svrMixerStateChanged
  + 每个输出有私有的状态变化事件svrMixerOutputInfo

**CRVIDEOSDK\_ERR\_DEF updateSvrMixerContent(Map< String, ArrayList< MixerContent> > mutiMixerContents)**

* **功能**: 更新云端录制、云端直播内容
* **返回值**: CRVIDEOSDK\_ERR\_DEF-操作结果

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| mutiMixerContents | Map<String, ArrayList<MixerContent> > | 服务器混图内容 |

**void stopSvrMixer()**

* **功能**: 停止云端录制、云端直播
* **返回值**: CRVIDEOSDK\_ERR\_DEF-操作结果
* 注意事项:
  + 停止有一定耗时，请关注混图器的状态变化事件svrMixerStateChanged\*
  + 每个输出有私有的状态变化事件svrMixerOutputInfo\*

**MIXER\_STATE getSvrMixerState()**

* **功能**: 获取云端录制、云端直播状态
* **返回值**: MIXER\_STATE

**void switchToPage(MAIN\_PAGE main, SubPage sub)**

* **功能**: 功能切换
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| main | MAIN\_PAGE | 功能类型 |
| sub | SubPage | 子页面标识（如创建白板时返回的boardID） |

**MAIN\_PAGE getCurrentMainPage()**

* **功能**: 获取当前主功能区
* **返回值**: MAIN\_PAGE-当前主功能区

**SubPage getCurrentSubPage()**

* **功能**: 获取当前子页面
* **返回值**: SubPage-当前子页面

**VIDEO\_WALL\_MODE getVideoWallMode()**

* **功能**: 获取视频墙当前分屏模式
* **返回值**: VIDEO\_WALL\_MODE-分屏模式

**void setVideoWallMode(VIDEO\_WALL\_MODE wallMode)**

* **功能**: 设置视频墙分屏模式
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| videoWallMode | VIDEO\_WALL\_MODE | 分屏模式 |

**void setMainVideo(String userID)**

* **功能**: 设置当前哪个用户为主视频
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 用户ID |

**String getMainVideo()**

* **功能**: 获取当前哪个用户为主视频
* **返回值**: 用户ID

**SubPage createBoard(String title, int width, int height, int pageCount)**

* **功能**: 创建电子白板
* **返回值**: SubPage-白板标识

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| title | String | 白板名称 |
| width | int | 白板宽度 |
| height | int | 白板高度 |
| pageCount | int | 白板内有多个页（一般空白板1页，文档白板为实际页数） |

* 注意事项:
  + 其他参会者会收到notifyCreateBoard事件
  + 后台会记录下白板数据，新入会者会收到notifyInitBoards事件
  + 创建完白板后，一定要及尽快调用initBoardPageDat初始化各页数据
  + 如果需要所有参会者同步切到此白板，请调用switchToPage

**void closeBoard(SubPage boardID)**

* **功能**: 关闭电子白板 , 其他参会者将收到notifyCloseBoard事件；同时后台会移除对应白板的所有信息
* **返回值**: 无

回调函数notifyCloseBoard

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| boardID | SubPage | 白板标识 |

**void initBoardPageDat(SubPage boardID, int boardPageNo, String imgID, String elemetDatas)**

* **功能**: 初始化白板指定页数据
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| boardID | SubPage | 白板标识 |
| boardPageNo | int | 白板第几页（0:代表第一页） |
| imgID | String | 白板的背景图片标识（空代表无背影图） |
| elemetDatas | String | 白板的初始图元（空代表无图元，一般在导入历史文件才用到），白板图元数据Json数组，详见定义BoardElement |

* 注意事项:
  + imgID非空时, 代表背景的图片ID。img来源请参见getNetDiskDocFilePageInfo
  + 其他参会者将收到notifyInitBoardPageDat事件
  + 后台会记录下白板的页数据，在新用户入会时，也会收到notifyInitBoardPageDat事件

**String createElementID()**

* **功能**: 创建图元标识
* **返回值**: elementID图元标识

**void addBoardElement(SubPage boardID, int boardPageNo, String element)**

* **功能**: 添加图元信息 , 其他参会者会收到：notifyAddBoardElement事件同时后台会保存图元，新入会者会在notifyInitBoardPageDat中得到这些图元
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| boardID | SubPage | 白板标识 |
| boardPageNo | int | 白板第几页（0:代表第一页） |
| element | String | 图元信息，Json数据格式详见定义BoardElement |

**void modifyBoardElement(SubPage boardID, int boardPageNo, String element)**

* **功能**: 修改图元信息 , 其他参会者会收到：notifyModifyBoardElement事件，同时后台会覆盖对应图元的数据，新入会者会在notifyInitBoardPageDat中得到这些图元
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| boardID | SubPage | 白板标识 |
| boardPageNo | int | 白板第几页（0:代表第一页） |
| element | String | 图元信息，Json数据格式详见定义BoardElement |

**void delBoardElement(SubPage boardID, int boardPageNo, ArrayList<String> elementIDs)**

* **功能**: 删除图元
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| boardID | SubPage | 白板标识 |
| boardPageNo | int | 白板第几页（0:代表第一页） |
| elementIDs | ArrayList<String> | 图元id列表 |

**void setMouseHotSpot(SubPage boardID, int boardPageNo, int x, int y)**

* **功能**: 设置鼠标热点信息
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| boardID | SubPage | 白板标识 |
| boardPageNo | int | 白板第几页（0:代表第一页） |
| x | int | 屏幕横坐标 |
| y | int | 屏幕纵坐标 |

**void setBoardCurPageNo([SubPage](TypeDefinitions.md#SubPage) boardID, int boardPageNo, int pagePos1, int pagePos2)**

* **功能**: 白板翻页
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| boardID | SubPage | 白板标识 |
| boardPageNo | int | 白板第几页（0:代表第一页） |
| pagePos1 | int | 私有扩展参数1（如可用于页内x方向滚动位置） |
| pagePos2 | int | 私有扩展参数2（如可用于页内y方向滚动位置） |

**void listNetDiskDocFile(String dir)**

* **功能**: 查询服务器指定目录下的内容（子目录名、上传的文档源文件及文档转换状态）
* **返回值**: 无

回调函数listNetDiskDocFileRslt

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| dir | String | 要查询的目录，空或"/"时，代表查询根目录下的内容 |

**void uploadDocFileToNetDisk(String svrPathFileName, String locPathFileName)**

* **功能**: 上传并转换文档
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| svrPathFileName | String | 服务器的路径文件名，包含目录时，服务器自动创建对应目录 |
| locPathFileName | String | 要上传文件的本地的路径文件名 |

* 注意事项:
  + 支持的文件格式有：doc，docx, ppt, pptx, pps, ppsx, xls, xlsx, et, ett, wps, wpt, dps, pdf, txt, log, ini, bmp, png, jpg, jpeg；

**void downloadNetDiskDocFile(String svrPathFileName, String locPathFileName)**

* **功能**: 下载源始文档
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| svrPathFileName | String | 服务器的路径文件名，包含目录时，服务器自动创建对应目录 |
| locPathFileName | String | 要上传文件的本地的路径文件名 |

**void cancelTransforNetDiskDocFile(String svrPathFileName)**

* **功能**: 取消文档传输
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| svrPathFileName | String | 服务器的路径文件名，包含目录时，服务器自动创建对应目录 |

**void deleteNetDiskDocFile(String svrPathFileName)**

* **功能**: 删除文档
* **返回值**: 无

回调函数deleteNetDiskDocFileRslt(删除结果)

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| svrPathFileName | String | 服务器的路径文件名，包含目录时，服务器自动创建对应目录 |

**void getNetDiskDocFilePageInfo(String svrPathFileName)**

* **功能**: 获取文档的转换信息
* **返回值**: 无

回调函数getNetDiskDocFilePageInfoRslt(获取结果)

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| svrPathFileName | String | 服务器的路径文件名，包含目录时，服务器自动创建对应目录 |

* 注意事项:
  + 获取到的文档转换后的信息后， 就可以创建白板createBoard, 然后为白板初始化文档内容

**void downloadNetDiskDocFilePage(String pagePathFileName, String locPathFileName)**

* **功能**: 下载文档转换后的页文件
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| pagePathFileName | String | 页文件在服务器的路径文件名 |
| locPathFileName | String | 文件下载后存放的本地的路径文件名，目录不存在时自动创建 |

* 注意事项:
  + 页文件格式： 6字节保留，4字节图像格式（2:jpg, 4:png），4字节图像内容长度，图像内容

**void setMediaCfg(VideoCfg cfg)**

* **功能**: 配置远程影音共享时，图像质量参数
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| cfg | VideoCfg | 影音共享配置参数 |

**VideoCfg getMediaCfg()**

* **功能**: 获取影音共享配置参数
* **返回值**: VideoCfg-影音共享配置参数

**void startPlayMedia(String filename, boolean bLocPlay)**

* **功能**: 播放影音文件
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| filename | String | 文件名，全路径 |
| bLocPlay | int | 是否仅仅本地播放（true:本地播放，false：房间内播放） |

**void startPlayMedia(String filename, int bLocPlay, boolean bPauseWhenFinished)**

* **功能**: 播放影音文件
* **返回值**: 无

播放视频则触发回调notifyMediaOpened， 停止播放则触发回调notifyMediaStop

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| filename | String | 文件名，全路径 |
| bLocPlay | int | 是否仅仅本地播放（1:本地播放，0：房间内播放） |
| bPauseWhenFinished | boolean | 是否停在最后一帧, 如果播放成功，其他人收到notifyMediaStart，如果播放失败，请关注通知事件notifyMediaStop |

**void pausePlayMedia(boolean bPause)**

* **功能**: 暂停播放影音
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| bPause | boolean | 是否暂停 |

**void stopPlayMedia()**

* **功能**: 停止影音播放
* **返回值**: 无

回调函数notifyMediaStop

**void setMediaPlayPos(int pos)**

* **功能**: 设置播放进度
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| pos | int | 设置播放位置，单位：毫秒 |

**ArrayList<String> getAllFilesInMediaPath()**

* **功能**: 取得影音文件夹下的所有可播放文件
* **返回值**: 文件名列表

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| pos | int | 设置播放位置，单位：毫秒 |

* 注意事项:
  + 影音文件夹位于方法init的第二个参数sdkFilePath，sdk会在此文件中建立media的子文件夹，即为影音文件夹

**MediaInfo getMediaInfo()**

* **功能**: 正在播放的影音信息
* **返回值**: MediaInfo-影音文件信息

**void setMediaVolume(int level)**

* **功能**: 设置影音播放的音量
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| level | int | 影音播放的音量，类型范围（0-255） |

**int getMediaVolume()**

* **功能**: 读取影音播放的音量
* **返回值**: 影音播放的音量，类型范围（0-255）

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 用户id |

**void sendMeetingCustomMsg(String text, String cookie)**

* **功能**:发送房间内自定义广播消息
* **返回值**:无
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| text | String | 用户自定义消息 |
| cookie | String | 可为空字符串。详细介绍见关键词 |

回调函数sendMeetingCustomMsgRlst

**void setPicResource(String resID, Bitmap bitmap)**

* **功能**: 将图片资源设置给sdk
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| resID | String | 资源唯一标识 |
| bitmap | Bitmap | 图片对象，为空代表移除资源 |

**void setPicResource(String resID, String picFile)**

* **功能**: 将图片资源设置给sdk
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| resID | String | 资源唯一标识 |
| picFile | Bitmap | 资源文件路径，为空代表移除资源 |

**boolean setPicResource(String resID, View view)**

* **功能**: 将图片资源设置给sdk
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| resID | String | 资源唯一标识 |
| view | View | 控件对象，为空代表移除资源（注：不能是opengl显示控件，必须是在界面已经显示的控件） |

**boolean setPicResource(String resID, View view, Size size)**

* **功能**: 将图片资源设置给sdk
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| resID | String | 资源唯一标识； |
| view | Bitmap | 控件对象，为空代表移除资源（注：不能是opengl显示控件，必须是在界面已经显示的控件） |
| size | Size | size 图片大小 |

**通知回调函数**

CloudroomVideoCallback是通话建立、音频采集播入、视频采集编解码、屏幕共享、录制、影音播放等功能的回调接口

**void enterMeetingRslt(CRVIDEOSDK\_ERR\_DEF sdkErr)**

* **功能**: 自己进入房间的结果
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果码 |
| cookie | String | 自定义用户数据 |

**void stopMeetingRslt(CRVIDEOSDK\_ERR\_DEF sdkErr)**

* **功能**: 通知结束视频房间结果
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果码 |
| cookie | String | 自定义用户数据 |

**void setNickNameRsp(CRVIDEOSDK\_ERR\_DEF sdkErr, String userid, String newName)**

* **功能**: 设置昵称的结果
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果码 |
| userid | String | 用户id |
| newName | String | 新的昵称 |

**void notifyNickNameChanged(String userID, String oldName, String newName)**

* **功能**: 某用户改变了昵称(改昵称的用户自身不会接收到此通知)
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 改变昵称的某用户id |
| oldName | String | 改变昵称的某用户昵称(改变之前的昵称) |
| newName | String | 改变昵称的某用户昵称(改变之后的昵称) |

**void userEnterMeeting(String userID)**

* **功能**: 某用户进入了房间
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 进入房间的用户id |

**void userLeftMeeting(String userID)**

* **功能**: 某用户离开了房间
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 离开房间的用户id |

**void meetingDropped()**

* **功能**: 通知从房间里掉线了，收到该通知后可以调用enterMeeting尝试重新入会
* **返回值**: 无
* 注意事项:
  + 如果用到了呼叫队列，掉线后不重新入会就必须调用hangupCall释放本次呼叫

**void meetingStoped()**

* **功能**: 房间已被结束
* **返回值**: 无

**void netStateChanged(int level)**

* **功能**: 网络变化通知
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| level | int | 网络状况等级(0~10，10分为最佳网络) |

**void audioDevChanged()**

* **功能**: 通知本地音频设备有变化
* **返回值**: 无

**void audioStatusChanged(String userID, ASTATUS oldStatus, ASTATUS newStatus)**

* **功能**: 通知音频状态变化
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 房间中设备的所有者 |
| oldStatus | ASTATUS | 旧状态 |
| newStatus | ASTATUS | 新状态 |

**void micEnergyUpdate( String userID, int oldLevel, int newLevel)**

* **功能**: 通知用户的说话声音强度更新
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 用户标识ID |
| oldLevel | int | 原来的说话声音强度(0~10) |
| newLevel | int | 现在的说话声音强度(0~10) |

**void openVideoRslt(String devID, boolean bSuccess)**

* **功能**: 打开摄像头设备操作结果
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| devID | String | 摄像头设备ID |
| bSuccess | boolean | 是否成功 |

**void videoStatusChanged(String userID, VSTATUS oldStatus, VSTATUS newStatus)**

* **功能**: 视频状态变化
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 房间中设备的所有者 |
| oldStatus | VSTATUS | 旧状态 |
| newStatus | VSTATUS | 新状态 |

**void videoDevChanged(String userID)**

* **功能**: 通知用户的视频设备有变化
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| String | String | 设备变化的用户ID |

**void notifyVideoData(UsrVideoId usrVideoID, long frameTime)**

* **功能**: 通知用户有新的视频数据
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| usrVideoID | UsrVideoId | 用户摄像头标识ID |
| frmTime | long | 图像的创建时间，可用作时间戳 |

**void defVideoChanged(String userID, short videoID)**

* **功能**: 通知用户的视频默认设备有变化
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 设备变化的用户ID |

**void uploadRecordFileErr(String fileName, int err)**

* **功能**: 上传录制文件错误通知
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| fileName | String | 上传文件名 |
| err | int | 操作失败代码 |

**void cancelUploadRecordFileErr(int sdkErr)**

* **功能**: 取消上传错误通知
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkErr | int | 操作失败代码 |

**void locMixerStateChanged(String mixerID, MIXER\_STATE state)**

* **功能**: 本地混图器状态变化通知
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| mixerID | String | 混图器唯一标识 |
| state | MIXER\_STATE | 状态 |

**void locMixerOutputInfo(String mixerID, String nameOrUrl, MixerOutputInfo outputInfo)**

* **功能**: 本地录制文件、本地直播信息通知
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| mixerID | String | 混图器唯一标识 |
| nameOrUrl | String | 录像名称、或直播url |
| outputInfo | MixerOutputInfo | 通知内容 |

**void svrMixerStateChanged(MIXER\_STATE state, int err，String operatorID)**

* **功能**: 云端录制、云端直播状态变化通知
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| state | MIXER\_STATE | 状态值 |
| err | int | 错误值，请参考CRVIDEOSDK\_ERR\_DEF |
| operatorID | String | 引起变化的用户id |

**void svrMixerCfgChanged()**

* **功能**: 云端录制、云端直播内容变化通知
* **返回值**: 无

**void svrMixerOutputInfo(MixerOutputInfo info)**

* **功能**: 云端录制文件、云端直播信息变化通知
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| outputInfo | MixerOutputInfo | 通知内容 |

**void startScreenShareRslt(CRVIDEOSDK\_ERR\_DEF sdkErr)**

* **功能**: 开启屏幕共享的响应事件
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkEr | CRVIDEOSDK\_ERR\_DEF | 错误码 |

**void stopScreenShareRslt(int sdkErr)**

* **功能**停止屏幕共享的响应事件
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkEr | CRVIDEOSDK\_ERR\_DEF | 错误码 |

**void notifyScreenShareStarted()**

* **功能**: 通知他人开启了屏幕共享
* **返回值**: 无

**void notifyScreenShareStopped()**

* **功能**: 通知他人停止了屏幕共享
* **返回值**: 无

**void startScreenMarkRslt(CRVIDEOSDK\_ERR\_DEF sdkErr)**

* **功能**: 开始屏幕标注结果
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkEr | CRVIDEOSDK\_ERR\_DEF | 错误码 |

**void stopScreenMarkRslt(int sdkErr)**

* **功能**: 停止屏幕标注结果
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkEr | CRVIDEOSDK\_ERR\_DEF | 错误码 |

**void notifyScreenMarkStarted()**

* **功能**: 通知屏幕标注开始
* **返回值**: 无

**void notifyScreenMarkStopped()**

* **功能**: 通知屏幕标注停止
* **返回值**: 无

**void sendMeetingCustomMsgRlst(CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**:通知发送房间内自定义广播消息结果
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | CRVIDEOSDK\_ERR\_DEF（错误码为CRVIDEOSDK\_NOERR表示没有错误） |
| cookie | String | 可为空字符串。详细介绍见关键词 |

**void notifyMeetingCustomMsg(String fromUserID, String text)**

* **功能**:通知收到房间内自定义广播消息
* **参数**：

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| fromUserID | String | 房间内广播消息的发送者的用户id |
| text | String | 广播内容 |

**void notifySwitchToPage(MAIN\_PAGE mainPage, [SubPage](TypeDefinitions.md#SubPage) pageID)**

* **功能**: 通知功能切换
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| mainPage | MAIN\_PAGE | 功能类型 |
| pageID | SubPage | 子页面标识 |

**void notifyVideoWallMode(VIDEO\_WALL\_MODE model)**

* **功能**: 通知视频分屏模式切换
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| model | VIDEO\_LAYOUT\_MODE | 分屏模式 |

**void notifyMainVideoChanged()**

* **功能**: 通知主视频更改
* **返回值**: 无
* **参数**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userID | String | 主视频用户ID |

**void notifyInitBoards(ArrayList<SubPageInfo> boards)**

* **功能**: SDK入会后通知房间中已经存在的白板列表
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| boards | ArrayList<SubPageInfo> | 已经创建好的白板列表 |

**void notifyInitBoardPageDat(SubPage boardID, int boardPageNo, String imgID, String elementDatas,String operatorID)**

* **功能**: 初始化白板页数据
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| boardID | SubPage | 白板标识 |
| boardPageNo | int | 白板页序号 |
| imgID | String | 页背景文件ID（空代表无背景）, 使用downloadNetDiskDocFile进行下载 |
| elementDatas | String | 此页的所有图元,白板图元数据Json数组，详见定义BoardElement |
| operatorID | String | 初始化用户（为空时，代表入会时后台事件） |

**void notifyCreateBoard(SubPageInfo board, String operatorID)**

* **功能**: 通知创建白板
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| board | SubPageInfo | 白板信息 |
| operatorID | String | 创建白板的用户ID |

**void notifyCloseBoard(SubPage boardID, String operatorID)**

* **功能**: 通知关闭白板
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| boardID | SubPage | 白板标识 |
| operatorID | String | 关闭白板的用户ID |

**void notifyAddBoardElement(SubPage boardID, int boardPageNo, String element, String operatorID)**

* **功能**: 通知添加图元信息
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| boardID | SubPage | 白板标识 |
| boardPageNo | int | 白板页序号 |
| element | String | 图元信息, Json数据格式详见定义BoardElement |
| operatorID | String | 添加图元的用户ID |

**void notifyModifyBoardElement(SubPage boardID, int boardPageNo, String element, String operatorID)**

* **功能**: 通知图元信息被修改
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| boardID | SubPage | 白板标识 |
| boardPageNo | int | 白板页序号 |
| element | String | 图元信息, Json数据格式详见定义BoardElement |
| operatorID | String | 添加图元的用户ID |

**void notifyDelBoardElement(SubPage boardID, int boardPageNo, ArrayList<String> elementIDs, String operatorID)**

* **功能**: 通知图元被删除
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| boardID | SubPage | 白板标识 |
| boardPageNo | int | 白板页序号 |
| elementIDs | ArrayList<String> | 图元id列表 |
| operatorID | String | 添加图元的用户ID |

**void notifyMouseHotSpot(SubPage boardID, int boardPageNo, int x, int y, String operatorID)**

* **功能**: 通知鼠标热点消息
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| boardID | SubPage | 白板标识 |
| x | int | 屏幕横坐标 |
| y | int | 屏幕纵坐标 |
| operatorID | String | 添加图元的用户ID |

**void notifyBoardCurPageNo(SubPage boardID, int boardPageNo, int pagePos1, int pagePos2)**

* **功能**: 通知白板翻页
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| boardID | SubPage | 白板标识 |
| boardPageNo | int | 白板的页序号(0为第一页) |
| pagePos1 | int | 私有扩展参数1（如可用于页内x方向滚动位置） |
| pagePos2 | int | 私有扩展参数2（如可用于页内y方向滚动位置） |

**void notifyRecordFileStateChanged(String fileName, int state)**

* **功能**: 通知录制文件状态更改
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| fileName | String | 本地文件路径 |
| state | int | 状态，0未上传，1上传中，2已上传，3上传失败 |

**void uploadRecordFileSuccess(String fileName, String fileUrl)**

* **功能**: 通知录制文件上传成功
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| fileName | String | 本地文件路径 |
| fileUrl | String | 服务端录制文件地址 |

**void notifyRecordFileUploadProgress(String fileName, int percent)**

* **功能**: 通知录制文件上传进度
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| fileName | String | 本地文件路径 |
| percent | int | 进度0-100 |

**void notifyMediaOpened(int totalTime, Size picSZ)**

* **功能**: 通知影音文件打开
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| totalTime | long | 影音时长(毫秒) |
| picSZ | Size | 宽高 |

**void notifyMediaStart(String userid)**

* **功能**: 通知影音开始播放
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userid | String | 操作者的用户id |

**void notifyMediaPause(String userid, boolean bPause)**

* **功能**: 通知设置鼠标热点消息
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userid | String | 操作者的用户id |
| bPause | boolean | 是否暂停 |

**void notifyMediaStop(String userid, int reason)**

* **功能**: 通知影音播放停止
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userid | String | 操作者的用户id |
| reason | int | 播放停止原因，数值参考MEDIA\_STOP\_REASON |

**void notifyPlayPosSetted(int setPTS)**

* **功能**: 通知播放进度已设置完成
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| setPTS | int | 播放进度 |

**void notifyMemberMediaData(String userid, int curPos)**

* **功能**: 通知影音帧数据已解码完毕
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| userid | String | 操作者的用户id |
| curPos | int | 该影音帧的时间，毫秒为单位 |

* 注意事项:
  + 收到此通知消息后，可通过getMediaImg获取图像显示；但如果之前显示的帧时戳更大，说明此通知消息已过时，直接忽略即可 如果使用了影音共享UI显示方式，不再需要自已关注此事件和进行显示处理

**void notifyAudioPCMDat(int aSide, byte[] audioDat)**

* **功能**: 通知语音PCM数据
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| aSide | int | 声道类型 |
| audioDat | byte[] | PCM数据 |

**void listNetDiskDocFileRslt(String dir, CRVIDEOSDK\_ERR\_DEF sdkERR, NetDiskDocDir docDir)**

* **功能**: 通知查询文档列表结果
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| dir | String | 查询的目录 |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果 |
| docDir | NetDiskDocDir | 成功时的查询的结果 |

**void getNetDiskDocFilePageInfoRslt(String svrPathFileName, CRVIDEOSDK\_ERR\_DEF sdkErr, NetDiskDocFilePageInfo pageInfo)**

* **功能**: 通知查询文档转换结果
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| svrPathFileName | String | 查询的文档名 |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果 |
| pageInfo | NetDiskDocFilePageInfo | 成功的查询的结果 |

**void deleteNetDiskDocFileRslt(String svrPathFileName, CRVIDEOSDK\_ERR\_DEF sdkERR)**

* **功能**: 通知删除文档结果
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| svrPathFileName | String | 查询的文档名 |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果 |

**void notifyNetDiskDocFileTrsfProgress(String svrPathFileName, int percent)**

* **功能**: 通知文档传输进度
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| svrPathFileName | String | 查询的文档名 |
| percent | int | 传输进度（等于100代表传输完成） |

**5. 队列组件（CloudroomQueue）的接口函数**

CloudroomQueue 是队列组件，实现队列功能，目的是为了实现用户自动分配。，整个程序的生命过程中只能有一个实例。

**void registerCallBack(CloudroomQueueCallback callBack)**

* **功能**: 注册监听回调
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callBack | CloudroomQueueCallback | 回调对象 |

**void unregisterCallBack(CloudroomQueueCallback callBack)**

* **功能**: 反注册监听回调
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callBack | CloudroomQueueCallback | 指定的反注册回调对象 |

**void initQueue(String cookie)**

* **功能**: 初始化用户队列功能数据
* **返回值**: 无

回调函数initQueueDatRslt

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| cookie | String | 自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

* 注意事项:
  + 在响应回调initQueueDatRslt初始化成功后，才可获取队列队列相关信息

**void refreshAllQueueStatus()**

* **功能**: 刷新所有队列状态信息
* **返回值**: 无

回调函数queueStatusChanged

* 注意事项:
  + 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

**ArrayList<QueueInfo> getAllQueueInfo()**

* **功能**: 获取队列信息
* **返回值**: ArrayList<QueueInfo>-队列信息列表

**QueueStatus getQueueStatus(int queID)**

* **功能**: 获指定取队列状态
* **返回值**: QueueStatus-队列状态

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| queID | int | 队列ID |

**QueuingInfo getQueuingInfo()**

* **功能**: 功能获取我的排队信息
* **返回值**: QueuingInfo-我的排队信息

**ArrayList<Integer> getServiceQueues()**

* **功能**: 获取我服务的所有队列
* **返回值**: 队列ID列表

**VideoSessionInfo getSessionInfo()**

* **功能**: 获取我的会话信息
* **返回值**: VideoSessionInfo-我的会话信息

**void startQueuing(int queID, String usrExtDat, String cookie)**

* **功能**: 客户开始排队
* **返回值**: 无

回调函数startQueuingRslt

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| queID | int | 排队的队列ID |
| usrExtDat | String | 用户排队扩展数据 |
| cookie | String | 用户自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void stopQueuing(String cookie)**

* **功能**: 客户停止排队
* **返回值**: 无

回调函数stopQueuingRslt

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| cookie | String | 用户自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void startService(int queID, String 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

回调函数startServiceRslt

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| queID | int | 排队的队列ID |
| cookie | String | 用户自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void startService(int queID, int priority, String 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

回调函数startServiceRslt

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| queID | int | 排队的队列ID |
| priority | int | 座席优先级（缺省为0, 0为最高优先级, 值越大优先级越低) |
| cookie | String | 用户自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

* 注意事项:
  + 如果没有开启免打扰，那么系统会自动分配客户：将收到事件autoAssignUser
  + 如果开启免打扰，系统就不会分配客户，如需服务客户可调用reqAssignUser
  + 座席优先级描述
    - 座席优先级只在服务的队列内部有效；
    - 客户优先分配给服务此队列优先级最高的，且空闲的座席；
    - 优先级相同时，则分配给最先空闲的座席；
    - 优先级高的座席变空闲时，不抢夺已分配的客户；

**void stopService(int queID, String cookie)**

* **功能**: 停止服务某个队列
* **返回值**: 无

回调函数stopServiceRslt

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| queID | int | 排队的队列ID |
| cookie | String | 用户自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void reqAssignUser(String cookie)**

* **功能**: 请求分配一个客户
* **返回值**: 无

回调函数reqAssignUserRslt

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| cookie | String | 用户自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

* 注意事项:
  + 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

**void acceptAssignUser(int queID, String userID, String cookie)**

* **功能**: 接受系统安排的客户
* **返回值**: 无

回调函数reqAssignUserRslt

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| queID | int | 排队的队列ID |
| userID | String | 队列中的用户ID |
| cookie | String | 用户自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void rejectAssignUser(int queID, String userID, String cookie)**

* **功能**: 拒绝系统安排的客户
* **返回值**: 无

回调函数reqAssignUserRslt

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| queID | int | 排队的队列ID |
| userID | String | 队列中的用户ID |
| cookie | String | 用户自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**通知回调函数**

CloudroomQueueCallback 是队列回调接口

**void initQueueDatRslt(CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 队列初始化操作结果
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果代码，成功为CRVIDEOSDK\_NOERR |
| cookie | String | 用户自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void queueStatusChanged(QueueStatus queStatus)**

* **功能**: 队列状态变化通知
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| queStatus | QueueStatus | 新的队列状态 |

* 注意事项:
  + 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
  + 在调用refreshAllQueueStatus时，查询到的队列数据有变化时，会有通知到来

**void queuingInfoChanged(QueuingInfo queuingInfo)**

* **功能**: 排队信息变化通知
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| queuingInfo | QueuingInfo | 排队信息 |

**void startQueuingRslt(CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 开始排队操作结果
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果代码，CRVIDEOSDK\_NOERR为成功操作 |
| cookie | String | 用户自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void stopQueuingRslt(CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 停止排队操作结果
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果代码，CRVIDEOSDK\_NOERR为成功操作 |
| cookie | String | 用户自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void startServiceRslt(int queID, CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 开始服务队列操作结果
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| queID | int | 排队的队列ID |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果代码，CRVIDEOSDK\_NOERR为成功操作 |
| cookie | String | 用户自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void stopServiceRslt(int queID, CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 停止服务队列操作结果
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| queID | int | 排队的队列ID |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果代码，CRVIDEOSDK\_NOERR为成功操作 |
| cookie | String | 用户自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void reqAssignUserRslt(CRVIDEOSDK\_ERR\_DEF sdkErr, UserInfo usr, String cookie)**

* **功能**: 停止服务队列操作结果
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果代码，CRVIDEOSDK\_NOERR为成功操作 |
| usr | UserInfo | 请求到的队列用户 |
| cookie | String | 用户自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

**void autoAssignUser(UserInfo usr)**

* **功能**: 队列系统自动分配客户
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| usr | UserInfo | 请求到的队列用户 |

* 注意事项:
  + 如果想停止系统的自动分配，请调用CloudroomVideoMgr中的 setDNDStatus 设置免打扰功能。

**void cancelAssignUser(int queID, String userID)**

* **功能**: 队列系统取消之前自动分配的的客户
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| queID | int | 服务的队列 |
| userID | String | 自动分配用户被取消 |

**void responseAssignUserRslt(CRVIDEOSDK\_ERR\_DEF sdkErr, String cookie)**

* **功能**: 拒绝、接受分配的用户
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| sdkErr | CRVIDEOSDK\_ERR\_DEF | 操作结果代码，CRVIDEOSDK\_NOERR为成功操作 |
| cookie | String | 用户自定义数据(在响应消息中回传给调用者)，不需要时传空字符串 |

* 注意事项:
  + 队列系统通过接口autoAssignUser给开始队列服务的人自动推送用户，收到系统分配的用户后， 如果队列服务者还未决定接受acceptAssignUser还是拒绝rejectAssignUser推送的用户，系统可用取消本次推送， 并通过本接口通知队列服务者。

**6. Http文件传输组件（CloudroomHttpFileMgr）的接口函数**

CloudroomHttpFileMgr是Http文件传输组件，实现Http文件上传下载、及文件管理。

**void registerCallBack(CloudroomHttpFileMgrCallback callBack)**

* **功能**: 注册监听回调
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callBack | CloudroomHttpFileMgrCallback | 回调对象 |

**void unregisterCallBack(CloudroomHttpFileMgrCallback callBack)**

* **功能**: 反注册监听回调
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| callBack | CloudroomHttpFileMgrCallback | 指定的反注册回调对象 |

**void startMgr()**

* **功能**: 启动Http文件文件组件运行
* **返回值**: 无
* 注意事项:
  + 在调用CloudroomVideoSDK的 init 初始化SDK后后方可调用

**void stopMgr()**

* **功能**: 停止Http文件文件组件运行
* **返回值**: 无

**ArrayList<FileTransInfo> getAllTransferInfos()**

* **功能**: 获取本地上传、下载文件信息
* **返回值**: ArrayList<FileTransInfo>-文件信息

**void startTransferFile(HttpReqInfo info)**

* **功能**: 开始下载/上传文件
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| info | HttpReqInfo | 上传文件信息 |

* 注意事项:
  + 上传不支持断点续传； 下载支持断点续传；如果文件传输完成，且fileVersion一致，下次再请求时会立即报告完成。

**void cancelFileTransfer(String fileName)**

* **功能**: 取消传输
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| fileName | String | 本地路径文件名 |

* 注意事项:

1. 取消时，只是停止了传输任务，不清理记录及断点文件。

**void rmTransferInfo(String fileName, int bRemoveLocFile)**

* **功能**: 删除传输记录及相关文件
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| fileName | String | 本地路径文件名 |
| bRemoveLocFile | int | 是否移除本地文件 |

* 注意事项:
  + 此接口将文件从管理器中移除（getAllTransferInfos将不再返回相关信息），如果bRemoveLocFile为1时， 那么上传的源始文件、下载的临时文件或结果文件都将被移除。

**通知回调函数**

CloudroomHttpFileMgrCallback是Http上传下载通知回调接口

**void fileStateChanged(String fileName, HTTP\_TRANSFER\_STATE state)**

* **功能**: 通知用户文件状态更改
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| fileName | String | 本地路径文件名 |
| state | HTTP\_TRANSFER\_STATE | 状态 |

**void fileProgress(String fileName, int finisedSize, int totalSize)**

* **功能**: 通知用户文件的传输进度
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| fileName | String | 本地路径文件名 |
| finisedSize | int | 已传输大小 |
| totalSize | int | 文件总大小 |

**void fileFinished(string fileName, int rslt)**

* **功能**: 通知用户文件传输结束
* **返回值**: 无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| fileName | String | 本地路径文件名 |
| rslt | int | 传输结果，详见定义 HTTP\_TRANSFER\_STATE |

**可视化组件**

可视化组件由多个模块组成， 主要模块包括：成员视频UI显示组件， 屏幕共享UI显示组件， 影音共享UI显示组件。 通过集成SDK的相关控件， 让使用者进行更加快速， 高效的开发工作。

**接口描述**

**可视化组件：**

| **组件** | **方式** | **接口** | **描述** |
| --- | --- | --- | --- |
| 成员视频UI显示组件 | 主调 | setUsrVideoId | 设置显示的目标用户视频 |
| getUsrVideoId | 获取当前显示的用户/摄像头编号 |
| savePicToFile | 保存视频截图到图片文件 |
| savePicToBase64 |
| getPicFrameTime | 获取图像时间戳 |
| isPicEmpty | 检查图像是否为空 |
| getPicWidth, getPicHeight | 获取图像宽、高 |
| clear | 清空缓存的图像 |
| setScaleType,getScaleType | 设置图像显示模式 |
| 屏幕共享UI显示组件 | 主调 |  |  |
| 影音共享UI显示组件 | 主调 | savePicToFile | 保存播放影音画面到图片文件 |
| savePicToBase64 |
| getPicFrameTime | 获取图像时间戳 |
| isPicEmpty | 检查图像是否为空 |
| getPicWidth, getPicHeight | 获取图像宽、高 |
| clear | 清空缓存的图像 |
| setScaleType,getScaleType | 设置图像显示模式 |

**接口函数详情**

**1. 成员视频UI显示组件**

com.cloudroom.cloudroomvideosdk.VideoUIView 是视频显示控件，它显示设定的用户的视频， 是由视频房间组件内相关接口的封装而成，如果此方式不满足需求，可以使用视频房间组件内相关的接口自行实现。 多实例方式，在会话中可以为每一个视频用户创建一个用来显示其画面

* 设置视频源
* 获取用户/摄像头编号
* 保存成员画面截图
* 获取图像信息
* 清空成员画面内容
* 设置图像显示模式

**设置视频源**

**void setUsrVideoId(UsrVideoId usrVideoId)**

* **功能**:  设置显示的目标用户视频
* **返回值**:  无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| usrVideoId | UsrVideoId | 目标用户视频ID |

**void setUsrVideoId(UsrVideoId usrVideoId, int qualityLv)**

* **功能**:  设置显示的目标用户视频
* **返回值**:  无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| usrVideoId | UsrVideoId | 目标用户视频ID |
| qualityLv | int | 视频档位(1:默认档位，2:对应quality2，3:对应quality3，如果2,3未开启时将显示1档的视频流） |

* 注意事项:
  + 从房间内可观看摄像头列表中获取userID和videoID

**获取用户/摄像头编号**

**UsrVideoId getUsrVideoId()**

* **功能**:  获取当前显示的用户
* **返回值**:  用户视频ID

**保存成员画面截图**

**int savePicToFile(String pathFileName, CompressFormat format)**

* **功能**:  保存视频截图到图片文件
* **返回值**:  0成功，非0为保存遇到的错误码

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| pathFileName | String | 保存为本地路径文件名 |
| format | CompressFormat | 图片格式 |

**String savePicToBase64(CompressFormat format)**

* **功能**:  保存视频截图为base64数据字符串
* **返回值**:  以base64编码的图片数据，不成功返回空

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| format | CompressFormat | 图片格式 |

**获取图像信息**

* **功能**:  获取图像时间戳
* **返回值**:  当前图像数据对应的时间戳

**boolean isPicEmpty()**

* **功能**:  检查图像是否为空
* **返回值**:  图像是否为空

**int getPicWidth()**

* **功能**:  获取图像宽
* **返回值**:  宽

**int getPicHeight()**

* **功能**:  获取图像高
* **返回值**:  高

**清空成员画面内容**

**void clear()**

* **功能**:  清空缓存的图像
* **返回值**:  无

**设置图像显示模式**

**void setScaleType(int scaleType)**

* **功能**:  设置图像显示模式
* **返回值**:  无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| scaleType | int | 0：等比缩放留空居中显示；  1：等比缩放裁剪铺满显示；  2：不等比缩放铺满显示（可能导致图像拉伸）； |

**int getScaleType()**

* **功能**:  获取视频显示模式
* **返回值**:  int

0：等比缩放留空居中显示；

1：等比缩放裁剪铺满显示；

2：不等比缩放铺满显示（可能导致图像拉伸）；

**2. 屏幕共享UI显示组件**

com.cloudroom.cloudroomvideosdk.ScreenShareUIView 是屏幕共享开启后远端显示控件， 是由视频房间组件内相关接口的封装而成，如果此方式不满足需求，可以使用视频房间组件内相关的接口自行实现。

**3. 影音共享UI显示组件**

com.cloudroom.cloudroomvideosdk.MediaUIView 影音显示控件， 是由视频房间组件内相关接口的封装而成，如果此方式不满足需求，可以使用视频房间组件内相关的接口自行实现功能。

* 保存影音画面截图
* 获取影音图像信息
* 清空影音画面内容
* 设置图像显示模式

**保存影音画面截图**

**int savePicToFile(String pathFileName, CompressFormat format)**

* **功能**:  保存播放影音画面到图片文件
* **返回值**:  0成功，非0为保存遇到的错误码

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| pathFileName | String | 保存为本地路径文件名 |
| format | CompressFormat | 图片格式 |

**String savePicToBase64(CompressFormat format)**

* **功能**:  保存播放截图为base64数据字符串
* **返回值**:  以base64编码的图片数据，不成功返回空

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| format | CompressFormat | 图片格式 |

**获取影音图像信息**

**int getPicFrameTime()**

* **功能**:  获取图像时间戳
* **返回值**:  当前图像数据对应的时间戳

**boolean isPicEmpty()**

* **功能**:  检查图像是否为空
* **返回值**:  图像是否为空

**int getPicWidth()**

* **功能**:  获取图像宽
* **返回值**:  宽

**int getPicHeight()**

* **功能**:  获取图像高
* **返回值**:  高

**清空影音画面内容**

**void clear()**

* **功能**:  清空缓存的图像
* **返回值**:  无

**设置图像显示模式**

**void setScaleType(int scaleType)**

* **功能**:  设置图像显示模式
* **返回值**:  无

| **参数** | **类型** | **含义** |
| --- | --- | --- |
| scaleType | int | 0：等比缩放留空居中显示；  1：等比缩放裁剪铺满显示；  2：不等比缩放铺满显示（可能导致图像拉伸）； |

**int getScaleType()**

* **功能**:  获取视频显示模式
* **返回值**:  int

0：等比缩放留空居中显示；

1：等比缩放裁剪铺满显示；

2：不等比缩放铺满显示（可能导致图像拉伸）；

**常量定义**

相关数值及含义定义

**错误码**

com.cloudroom.cloudroomvideosdk.model.CRVIDEOSDK\_ERR\_DEF

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| 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 | App ID或App Secret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 | 登录用户被踢下线 |
| CRVIDEOSDK\_NOT\_INIT | 11 | sdk未初始化 |
| CRVIDEOSDK\_NOT\_LOGIN | 12 | 还没有登录 |
| CRVIDEOSDK\_CRVIDEOSDK\_BASE64\_COV\_ERR | 13 | base64转换失败 |
| CRVIDEOSDK\_CUSTOMAUTH\_NOINFO | 14 | 启用了第三方鉴权，但没有携带鉴权信息 |
| CRVIDEOSDK\_CUSTOMAUTH\_NOTSUPPORT | 15 | 没有启用第三方鉴权，但携带了鉴权信息 |
| CRVIDEOSDK\_CUSTOMAUTH\_EXCEPTION | 16 | 访问第三方鉴权服务异常 |
| CRVIDEOSDK\_CUSTOMAUTH\_FAILED | 17 | 第三方鉴权不通过 |
| CRVIDEOSDK\_KICKOUT\_INVALID\_TOKEN | 18 | 令牌失效被踢 |
| CRVIDEOSDK\_TOKEN\_AUTHINFOERR | 19 | 鉴权信息错误 |
| CRVIDEOSDK\_TOKEN\_SECRETERR | 20 | 用户秘钥信息错误 |
| CRVIDEOSDK\_TOKEN\_AUTHCHECKERR | 21 | 鉴权校验失败 |
| CRVIDEOSDK\_TOKEN\_AUTHERR | 22 | 鉴权异常 |
| CRVIDEOSDK\_TOKEN\_AUTHINFOTIMEOUT | 23 | 鉴权信息已过期 |
| CRVIDEOSDK\_TOKEN\_APPIDNOTSAME | 24 | 鉴权appid不一致 |
| 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\_VOICEENG\_INITFAILED | 208 | 语音引擎初始化失败 |
| 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 | 房间不存在或已结束 |
| CRVIDEOSDK\_AUTHERROR | 801 | 房间密码不正确 |
| CRVIDEOSDK\_MEMBEROVERFLOWERROR | 802 | 房间终端数量已满（购买的license不够) |
| CRVIDEOSDK\_RESOURCEALLOCATEERROR | 803 | 分配房间资源失败 |
| CRVIDEOSDK\_MEETROOMLOCKED | 804 | 房间已加锁 |
| CRVIDEOSDK\_BALANCELESSERROR | 805 | 余额不足 |
| CRVIDEOSDK\_SEVICE\_NOTENABLED | 806 | 业务权限未开启 |
| CRVIDEOSDK\_ALREADYLOGIN | 807 | 不能再次登录 |
| CRVIDEOSDK\_MIC\_NORIGHT | 808 | 没有mic权限 |
| CRVIDEOSDK\_MIC\_BEING\_USED | 809 | mic已被使用 |
| CRVIDEOSDK\_MIC\_UNKNOWERR | 810 | mic未知错误 |
| CRVIDEOSDK\_SPK\_NORIGHT | 811 | 没有扬声器权限 |
| CRVIDEOSDK\_SPK\_BEING\_USED | 812 | 扬声器已被使用 |
| CRVIDEOSDK\_SPK\_UNKNOWERR | 813 | 扬声器未知错误 |
| CRVIDEOSDK\_PIC\_ISNULL | 814 | 图像为空 |
| 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\_RECORD\_SIZE\_ERR | 903 | 录像的分辨率超出限制 |
| CRVIDEOSDK\_SENDFAIL | 1000 | 发送失败 |
| CRVIDEOSDK\_CONTAIN\_SENSITIVEWORDS | 1001 | 有敏感词语 |
| CRVIDEOSDK\_SENDCMD\_LARGE | 1100 | 发送信令数据过大 |
| CRVIDEOSDK\_SENDBUFFER\_LARGE | 1101 | 发送数据过大 |
| CRVIDEOSDK\_SENDDATA\_TARGETINVALID | 1102 | 目标用户不存在 |
| CRVIDEOSDK\_SENDFILE\_FILEINERROR | 1103 | 文件错误 |
| CRVIDEOSDK\_TRANSID\_INVALID | 1104 | 无效的发送id |
| CRVIDEOSDK\_RECORDFILE\_STATE\_ERR | 1200 | 状态错误不可上传/取消上传 |
| CRVIDEOSDK\_RECORDFILE\_NOT\_EXIST | 1201 | 录制文件不存在 |
| CRVIDEOSDK\_RECORDFILE\_UPLOAD\_FAILED | 1202 | 上传失败 |
| CRVIDEOSDK\_RECORDFILE\_DEL\_FAILED | 1203 | 移除本地文件失败 |
| CRVIDEOSDK\_IPCAM\_URLERR | 1300 | 网络摄像头url不正确 |
| CRVIDEOSDK\_IPCAM\_ALREADYEXIST | 1301 | 网络摄像头已存在 |
| CRVIDEOSDK\_IPCAM\_TOOMUCH | 1302 | 网络摄像头达到最大数量 |
| CRVIDEOSDK\_FILE\_NOT\_EXIST | 1400 | 文件不存在 |
| CRVIDEOSDK\_FILE\_READ\_ERR | 1401 | 文件读失败 |
| CRVIDEOSDK\_FILE\_WRITE\_ERR | 1402 | 文件写失败 |
| CRVIDEOSDK\_FILE\_ALREADY\_EXIST | 1403 | 目标文件已存在 |
| CRVIDEOSDK\_FILE\_OPERATOR\_ERR | 1404 | 文件操作失败 |
| CRVIDEOSDK\_NETDISK\_NOT\_EXIST | 1500 | 网盘不存在 |
| CRVIDEOSDK\_NETDISK\_PERMISSIONDENIED | 1501 | 没有网盘权限 |
| CRVIDEOSDK\_NETDISK\_INVALIDFILENAME | 1502 | 不合法文件名 |
| CRVIDEOSDK\_NETDISK\_FILEALREADYEXISTS | 1503 | 文件已存在 |
| CRVIDEOSDK\_NETDISK\_FILEORDIRECTORYNOTEXISTS | 1504 | 文件或目录不存在 |
| CRVIDEOSDK\_NETDISK\_FILENOTTRANSFORM | 1505 | 文件没有转换 |
| CRVIDEOSDK\_NETDISK\_TRANSFORMFAILED | 1506 | 文件转换失败 |
| CRVIDEOSDK\_NETDISK\_NOSPACE | 1507 | 空间不足 |

**SDK日志等级定义**

com.cloudroom.cloudroomvideosdk.model.SDK\_LOG\_LEVEL\_DEF

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| SDKLEVEL\_DEBUG | 0 | 调试日志 |
| SDKLEVEL\_INFO | 1 | 重要日志 |
| SDKLEVEL\_WARN | 2 | 警告日志 |
| SDKLEVEL\_ERR | 3 | 错误日志 |
| SDKLEVEL\_CRIT | 4 | 崩溃日志 |

**麦克风状态**

com.cloudroom.cloudroomvideosdk.model.ASTATUS

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| AUNKNOWN | 0 | 麦克风状态未知 |
| 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 |
| AACCEPTING | 5 | 向服务器发送帮助他人开麦中 |

**摄像头（视频）状态定义**

com.cloudroom.cloudroomvideosdk.model.VSTATUS

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| VUNKNOWN | 0 | 摄像头状态未知 |
| VNULL | 1 | 没有摄像头设备 |
| VCLOSE | 2 | 摄像头处于关闭状态（软开关） |
| VOPEN | 3 | 摄像头处于打开状态（软开关） |
| VOPENING | 4 | 向服务器发送打开消息中 |

**视频图像格式**

com.cloudroom.cloudroomvideosdk.model.VIDEO\_FORMAT

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| VFMT\_YUV420P | 0 | yuv420p |
| VFMT\_ARGB32 | 1 | 32-bit ARGB format (0xAARRGGBB) |

**混图内容类型**

com.cloudroom.cloudroomvideosdk.model.MIXER\_VCONTENT\_TYPE

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| MIXVTP\_VIDEO | 0 | 摄像头 |
| MIXVTP\_PIC | 1 | 图片 |
| MIXVTP\_SCREEN | 2 | 屏幕 |
| MIXVTP\_MEDIA | 3 | 影音共享 |
| MIXVTP\_TIMESTAMP | 4 | 时间戳 |
| MIXVTP\_REMOTE\_SCREEN | 5 | 远端共享的屏幕 |
| MIXVTP\_WBOARD | 6 | 白板 |
| MIXVTP\_TEXT | 7 | 文本 |

**混图输出类型**

com.cloudroom.cloudroomvideosdk.model.MIXER\_OUTPUT\_TYPE

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| MIXOT\_FILE | 0 | 录像文件 |
| MIXOT\_LIVE | 1 | 直播流 |

**混图器状态**

com.cloudroom.cloudroomvideosdk.model.MIXER\_STATE

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| MIXER\_NULL | 0 | 没有创建 |
| MIXER\_STARTING | 1 | 正在开启 |
| MIXER\_RUNNING | 2 | 正在运行 |
| MIXER\_RUNNING | 4 | 正在结束 |

**混图器输出状态**

com.cloudroom.cloudroomvideosdk.model.MIXER\_OUTPUT\_STATE

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| OUTPUT\_NONE | 0 | 没有创建 |
| OUTPUT\_CREATED | 1 | 输出对象已创建 |
| OUTPUT\_WRITING | 2 | 输出目标信息更新 |
| OUTPUT\_CLOSED | 3 | 输出对象已关闭 |
| OUTPUT\_ERR | 4 | 输出对象异常 |

**录制文件上传状态**

com.cloudroom.cloudroomvideosdk.model.RECORD\_FILE\_STATE

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| RFS\_NoUpload | 0 | 未上传 |
| RFS\_Uploading | 1 | 上传中 |
| RFS\_Uploaded | 2 | 已上传 |
| RFS\_UploadFail | 3 | 上传失败 |

**主功能页类型**

com.cloudroom.cloudroomvideosdk.model.MAIN\_PAGE

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| MAINPAGE\_VIDEOWALL | 0 | 视频墙 |
| MAINPAGE\_SCREENSHARE | 1 | 屏幕共享 |
| MAINPAGE\_WHITEBOARD | 2 | 电子白板 |
| MAINPAGE\_MEDIASHARE | 3 | 影音共享 |

**影音播放状态**

com.cloudroom.cloudroomvideosdk.model.MEDIA\_STATE

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| MEDIA\_START | 0 | 开始 |
| MEDIA\_PAUSE | 1 | 暂停 |
| MEDIA\_STOP | 2 | 停止 |

**影音结束原因**

com.cloudroom.cloudroomvideosdk.model.MEDIA\_STOP\_REASON

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| MEDIA\_CLOSE | 0 | 文件关闭 |
| MEDIA\_FINI | 1 | 播放到文件尾部 |
| MEDIAMEDIA\_FILEOPEN\_ERR\_STOP | 2 | 打开文件失败 |
| MEDIA\_FORMAT\_ERR | 3 | 文件格式错误 |
| MEDIA\_UNSUPPORT | 4 | 影音格式不支持 |
| MEDIA\_EXCEPTION | 5 | 其他异常 |

**视频墙分屏模式**

com.cloudroom.cloudroomvideosdk.model.VIDEO\_WALL\_MODE

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| VLO\_1v1\_M | 0 | 互看 |
| VLO\_WALL1\_M | 1 | 1分屏 |
| VLO\_WALL2 | 2 | 2分屏 |
| VLO\_WALL4 | 3 | 4分屏 |
| VLO\_WALL5\_M | 4 | 5分屏 |
| VLO\_WALL6\_M | 5 | 6分屏 |
| VLO\_WALL9 | 6 | 9分屏 |
| VLO\_WALL13\_M | 7 | 13分屏 |
| VLO\_WALL16 | 8 | 16分屏 |
| VLO\_WALL25 | 9 | 25分屏 |

**Http文件传输状态**

com.cloudroom.cloudroomvideosdk.model.HTTP\_TRANSFER\_STATE

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| HTTPFS\_NULL | 0 | 未开始 |
| HTTPFS\_QUEUE | 1 | 排队中 |
| HTTPFS\_TRANSFERING | 2 | 传输(上传/下载)中 |
| HTTPFS\_FINISHED | 3 | 传输完成 |

**Http文件传输结果**

com.cloudroom.cloudroomvideosdk.model.HTTP\_TRANSFER\_RESULT

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| HTTPR\_Success | 0 | 成功 |
| HTTPR\_InnerErr | 1 | 内部错误 |
| HTTPR\_ParamErr | 2 | 参数错误 |
| HTTPR\_NetworkFail | 3 | 网络不通/地址不对 |
| HTTPR\_NetworkTimeout | 4 | 超时失败 |
| HTTPR\_FileOperationFail | 5 | 文件操作失败 |
| HTTPR\_PathNotSupprot | 6 | 不支持的路径 |
| HTTPR\_FileTransfering | 7 | 文件正在传输 |
| HTTPR\_HTTPERR\_BEGIN | 1000 | http错误码启始 |
| HTTPR\_HTTPERR\_END | 1999 | http错误码结束 |

**用户状态**

com.cloudroom.cloudroomvideosdk.model.CLIENT\_STATUS

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| OFFLINE | 0 | 离线状态 |
| ONLINE | 1 | 在线，且空闲状态 |
| BUSY | 2 | 在线，忙状态 |
| MEETING | 3 | 在线，在房间中状态 |

**呼叫状态**

com.cloudroom.cloudroomvideosdk.model.INVITE\_STATUS

| **代码** | **数值** | **含义** |
| --- | --- | --- |
| IS\_RINGING | 0 | 振铃 |
| IS\_ACCEPTED | 1 | 接通 |
| IS\_DECLINED | 2 | 拒绝 |
| IS\_NORESPONSE | 3 | 未应答 |
| IS\_hangup | 4 | 挂断 |

**对象结构定义**

**Sdk初始化数据实体类**

com.cloudroom.cloudroomvideosdk.model.SdkInitDat

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| sdkDatSavePath | String | SDK内部使用文件位置 |
| noCall | boolean | 是否使用呼叫业务（可减少登录环节、及通信需求） |
| noQueue | boolean | 是否使用sdk的排队功能的业务（可减少登录环节、及通信需求） |
| noMediaDatToSvr | boolean | 是否与服务器实时流媒体数据（可减少复杂度，加快登录速度） |
| timeOut | int | 网络通信超时时间(10000-120000)，单位是毫秒，超出范围时就近取边界值 |
| datEncType | String | 数据加密类型("0":敏感数据加密，"1":全面加密; 缺省:"1") |

**房间对象数据类**

com.cloudroom.cloudroomvideosdk.model.MeetInfo

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| ID | int | 房间号，数值0代表房间信息无效 |
| pswd | String | 房间密码；（空代表房间无密码） |
| subject | String | 房间主题 |
| pubMeetUrl | String | 房间公共链接 |
| hostPswd | String | 主持人密码 |

**用户状态**

com.cloudroom.cloudroomvideosdk.model.UserStatus

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| userID | String | 用户id |
| userStatus | CLIENT\_STATUS | 用户的在线状态 |
| DNDType | boolean | 用户免打扰状态，参见setDNDStatus |

**大小描述**

com.cloudroom.cloudroomvideosdk.model.Size

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| width | int | 宽 |
| height | int | 高 |

**房间成员信息**

com.cloudroom.cloudroomvideosdk.model.MemberInfo

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| userID | String | 用户ID |
| nickname | String | 用户昵称 |
| audioStatus | ASTATUS | 音频状态 |
| videoStatus | VSTATUS | 视频状态 |

**音频配置**

com.cloudroom.cloudroomvideosdk.model.AudioCfg

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| \_micName | String | 麦克风设备名称(空代表系统默认设备) |
| \_speakerName | String | 扬声器名称(空代表系统默认设备) |
| agc | boolean | 是否开启声音增益，默认开启 |
| ans | boolean | 是否开启降噪，默认开启 |
| aec | boolean | 是否开启回声消除，默认开启 |

**视频配置**

com.cloudroom.cloudroomvideosdk.model.VideoCfg

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| size | Size | 视频尺寸 |
| fps | int | 视频帧率(5~30) |
| maxbps | int | 视频码率（1~100*1000*1000）(未配置则使用内部默认值) |
| minQuality | int | 最佳质量(18~51，越小质量越好) (未配置则使用内部默认值25) |
| maxQuality | int | 最差质量(18~51， 越大质量越差) (未配置则使用内部默认值36) |

* 注意事项
  + 我们采用的是vbr编码（由质量+码率，双重控制）
  + qp范围：质量参数，为的是达到目标质量后，无需花费更大码率提高质量
  + maxbps码率控制，是为了确保结果一定不大于“目标码率”（体积受控）, 当要超出码率控制时，自动降低质量；当质量达到目标时，自动减少码率甚至无码率输出 注：使用硬编时第二点是无效的。

**视频设备私有属性**

com.cloudroom.cloudroomvideosdk.model.VideoAttributes

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| disabled | int | 取值0:不禁用此设备(默认值)，1：禁用此设备； |
| quality1Cfg | VideoCfg | 第一档视频配置，未配置时采用全局配置 |
| quality2Cfg | VideoCfg | 第二档视频配置，未配置代表关闭对应档位视频流 |

* 注意事项
  + 一个摄像头出多档质量的视频流，将带来很大的cpu开销；
  + 通过setUsrVideoId可以选择观看的视频流；

**用户视频信息**

com.cloudroom.cloudroomvideosdk.model.UsrVideoInfo

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| videoName | String | 设备名称 |
| isIPCamera | boolean | 是否IP摄像头 |

**用户摄像头**

com.cloudroom.cloudroomvideosdk.model.UsrVideoId

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| userId | String | 用户id |
| videoID | String | 设备id |

**屏幕共享标注信息**

com.cloudroom.cloudroomvideosdk.model.ScreenMarkData

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| userID | String | 标注者用户ID |
| useridSN | short | 标注SN |
| color | int | 标注颜色 |
| mousePosSeq | byte[] | 标注数据 |

**屏幕共享删除标注信息**

com.cloudroom.cloudroomvideosdk.model.RemoveScreenMarkData

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| userID | String | 标注者用户ID |
| useridSN | short | 标注SN |

**混图器规格配置**

com.cloudroom.cloudroomvideosdk.model.MixerCfg

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| dstResolution | Size | 图像大小 |
| frameRate | int | 图像帧率，取值范围:1-30(值越大,cpu要求更高，录像推荐15帧，直播推存25帧) |
| bitRate | int | 录制视频文件的最高码率，当图像变化小时，实际码率会低于此值 |
| defaultQP | int | 录制视频文件的缺省质量，缺省值：26 |
| gop | int | I帧周期(I帧越少码率越小，但直播延时会越大）； 文件录制建议15秒一个I帧取值：fpsx15； 直播建议4秒一个I帧取值: fpsx4; |

* 注意事项
  + bitRate的设置应该按照录制大小：
    - 640 \* 360大小录制对应 500000 -- (500kbps)
    - 1280 \* 720大小录制对应 1000000 -- (1mbps)
    - 1920 \* 1080大小录制对应 2000000 -- (2mbps)
  + defaultQP取值范围：0~51，0表示完全无损, 51表示质量非常差，推荐高质量取值18，中质量26， 低质量34。

**混图器内容配置**

com.cloudroom.cloudroomvideosdk.model.MixerCotent

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| type | MIXER\_VCONTENT\_TYPE | 在录制画面中的区域（相对于录像尺寸） |
| rect | Rect | 在录制画面中的区域（相对于录像尺寸） |
| param | int | 提供混图支持的参数 |

成员方法说明

| **成员方法** | **参数** | **返回值** | **描述** |
| --- | --- | --- | --- |
| createVideoContent(静态) | String userID | MixerCotent | 通过摄像头ID，将摄像头配置到混图器录制内容中 |
| short camID |
| Rect rect |
| createMediaContent(静态) | Rect rect | MixerCotent | 通过摄像头ID，将播放视频配置到混图器录制内容中 |
| createPicContent(静态) | String resID | MixerCotent | 通过摄像头ID，将控件图片配置到混图器录制内容中 |
| Rect rect |

* 注意事项
  + itemRt 在混图画面中的区域（相对于混图器尺寸)
  + type, 类型MIXER\_VCONTENT\_TYPE:
    - 当type=MIXVTP\_VIDEO时，表示混图的是摄像头，param必须包含camid
    - 当type=MIXVTP\_PIC时，表示混图的是指定的图片，param必须包含resourceid。（仅用于本地混图）
    - 当type=MIXVTP\_SCREEN时，表示混图的是本地屏幕，param可以增加附加参数screenid/pid/area
    - 当type=MIXVTP\_TIMESTAMP时，表示混图的是时戳，不用附加任何参数
    - 当type=MIXVTP\_REMOTE\_SCREEN时，表示混图的是远端共享的屏幕，不用附加任何参数
    - 当type=MIXVTP\_WBOARD时，表示混图的是白板，不用附加任何参数。（仅用于云端混图，本地混图应该用MIXVTP\_PIC)
    - 当type=MIXVTP\_TEXT时，表示混图的是文本(支持简单html)，param必须包含resourceid和text
  + param 支持的参数如下：
    - camid 用户id.摄像头id, 如："testuser.1"
    - resourceid 具有唯一属性的字符串id，通过setPicResource方法将图片存储到id内供混图模块使用
    - screenid 屏幕序号，-1表示主屏
    - pid 进程号
    - area 抓屏区域:x,y,w,h, 如："10,20,400,800"，无此参数时代表抓全屏
  + MIXVTP\_TEXT时，width和height可为0，此时元素大小由文本信息自动确定

**混图器输出配置**

com.cloudroom.cloudroomvideosdk.model.MixerOutPutCfg

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| type | MIXER\_OUTPUT\_TYPE | 输出类型MIXER\_OUTPUT\_TYPE |
| filename | String | 录像路径文件名（本地录像名格式如：/sdcard/1.mp4，服务器录像名格式如：/2018-11-21/1.mp4），支持的文件格式为mp4/ts/flv/avi，其中flv和ts两种格式在程序异常结束时产生的录制文件仍可用。 |
| encryptType | int | 录像文件是否加密，0:不加密，1:加密； |
| isUploadOnRecording | boolean | 录像文件是否边录边传，false:不上传，true:边录边传; (此参数仅本地录像有效） |
| serverPathFileName | String | 边录边传时，上传到服务器的路径文件名； (此参数仅本地录像有效） |
| liveUrl | String | 直播推流地址，支持rtmp/rtsp |
| errRetryTimes | int | 直播推流异常时，重试次数 |

**录制文件、直播信息通知**

com.cloudroom.cloudroomvideosdk.model.MixerOutputInfo

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| state | MIXER\_OUTPUT\_STATE | 状态 |
| duration | long | 录像文件时长，单位：毫秒; |
| fileSize | long | 录像文件大小； |
| errCode | CRVIDEOSDK\_ERR\_DEF | 错误码 |

* 注意事项
  + 输出对象已创建;
  + 输出目标信息更新; 可以从结构中取得duration, fileSize参数；
  + 输出对象已关闭; 可以从结构中取得duration, fileSize参数；
  + 输出对象异常；可以从结构中取得err参数；

**录制文件**

com.cloudroom.cloudroomvideosdk.model.RecordFileInfo

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| fileName | String | 文件名，全路径 |
| startTime | long | 开始录制时间 |
| duration | int | 录制时长 |
| fileSize | int | 文件大小 |

成员方法说明

| **成员方法** | **参数** | **返回值** | **描述** |
| --- | --- | --- | --- |
| clear | 无 | 无 | 清空实体类中的参数 |

**录制文件**

com.cloudroom.cloudroomvideosdk.model.RecordFileShow

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| fileName | String | 文件名，全路径 |
| fileSize | int | 文件大小 |
| state | RECORD\_FILE\_STATE | 文件状态 |
| uploadPercent | int | 录制结果中视频尺寸高度上传进度 |

**影音文件信息**

com.cloudroom.cloudroomvideosdk.model.MediaInfo

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| userID | String | 用户id |
| state | MEDIA\_STATE | 播放状态 |
| mediaName | String | 影音文件名 |

**屏幕共享配置对象**

com.cloudroom.cloudroomvideosdk.model.ScreenShareCfg

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| maxFPS | int | 帧率, 缺省为8 （当网络发不动时，帧率会自动下降） |
| maxBps | int | 最大码率，缺省为800kbps |

**白板标识**

com.cloudroom.cloudroomvideosdk.model.SubPage

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| userID | String | 用户id |
| pageID | short | 白板页ID |

**白板信息**

com.cloudroom.cloudroomvideosdk.model.SubPageInfo

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| page | SubPage | 白板标识 |
| title | String | 白板的名字 |
| width | int | 宽 |
| height | int | 高 |
| pageCount | int | 页数 |
| curPage | int | 当前页 |
| pagePos1 | int | 扩展参数1 |
| pagePos2 | int | 扩展参数2 |

**网盘文件**

com.cloudroom.cloudroomvideosdk.model.FileInfo

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| ownerID | short | 文件所有者的ID |
| ownerName | String | 文件所有者的名称 |
| name | String | 在服务器上的文件名 |
| orgFileName | String | 用户的原始文件名 |
| md5 | String | 压缩后的md5(如果不压缩，则原文件md5) |
| ctime | String | 上传时间，字符串，格式"yyyy-MM-dd hh:mm:ss" |
| size | int | 在服务器上的大小(单位B，类型ulong) |
| orgSize | int | 原始文件大小(单位B，类型ulong) |
| status | int | 文件状态 |

**网盘容量信息**

com.cloudroom.cloudroomvideosdk.model.DiskSummary

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| confDiskLimit | int | 网盘总容量（KB） |
| confDiskUsed | int | 网盘已用容量（KB） |

**队列信息**

com.cloudroom.cloudroomvideosdk.model.QueueInfo

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| queID | String | 队列ID |
| name | String | 队列名称 |
| desc | String | 队列描述 |
| prio | int | 优先级，值越小优先级越高 |

**队列状态**

com.cloudroom.cloudroomvideosdk.model.QueueStatus

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| queID | String | 队列ID |
| agent\_num | int | 坐席数量 |
| wait\_num | int | 排队客户数量 |
| srv\_num | int | 正在服务的客户数量 |

**排队信息**

com.cloudroom.cloudroomvideosdk.model.QueuingInfo

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| queID | String | 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) |
| position | int | 我的位置 |
| queuingTime | int | 我排队的时长(单位s) |

**会话信息**

com.cloudroom.cloudroomvideosdk.model.VideoSessionInfo

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| callID | String | 会话中的呼叫ID |
| peerID | String | 会话中的目标用户ID |
| peerName | String | 会话中的目标用户昵称 |
| bCallAccepted | boolean | 呼叫是否被对方接受 0:暂未接受，1:已接受 |
| meetingID | int | 会话中分配的房间ID |
| meetingPswd | String | 房间密码 |
| duration | int | 会话持续的时长(秒) |

**队列用户**

com.cloudroom.cloudroomvideosdk.model.UserInfo

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| queID | String | 队列ID |
| usrID | String | 用户ID |
| name | String | 用户昵称 |
| queuingTime | int | 用户排队的时长(单位s) |
| usrExtDat | String | 用户排队时传入的扩展参数 |

**Http文件传输请求对象**

com.cloudroom.cloudroomvideosdk.model.HttpReqInfo

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| filePathName | String | 本地完整路径文件名 |
| fileVersio | String | 文件版本（可以填版本号，也可以md5，也可以为空） |
| httpUrl | String | 目标URL |
| bUploadType | boolean | 是否上传 |

**Http传输信息对象**

com.cloudroom.cloudroomvideosdk.model.FileTransInfo

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| filePathName | String | 本地完整路径文件名 |
| fileVersio | String | 文件版本（可以填版本号，也可以md5，也可以为空） |
| httpUrl | String | 目标URL |
| bUploadType | boolean | 是否上传 |
| extHeaders | HashMap<String, String> | http头部扩展信息，字典数据 |
| fileSize | int | 文件大小 |
| finishedSize | int | 已传输大小 |
| state | HTTP\_TRANSFER\_STATE | 文件传输状态 |

**文档列表**

com.cloudroom.cloudroomvideosdk.model.NetDiskDocDir

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| name | String | 在网盘里的文件名 |
| files | ArrayList<FileInfo> | 目录下的所有文件 |
| subDirs | ArrayList<NetDiskDocDir> | 目录下的所有目录 |

**文档页信息**

com.cloudroom.cloudroomvideosdk.model.NetDiskDocFilePageInfo

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| width | int | 文档的图像宽度 |
| height | int | 文档的图像高度 |
| fileList | ArrayList<String> | 每页文档在服务器上的路径文件名 |

**白板图元**

BoardElement

| **成员变量** | **类型** | **说明** |
| --- | --- | --- |
| elementID | String | 图元id |
| type | int | 图元的类型，值100以下为云屋保留值，100及以上为自定义值 |
| left | String队列 | 图元在页内距离的左角位置 |
| top | String队列 | 图元在页内距离的上角位置 |

* 注意事项
  + 白板图元是后台服务器返回的json数据，而不是一个具体的实体类
  + elementID必须调用createElementID获取, 即使是曾经调此接口创建的然后存入了磁盘文件，再次读入房间时，所有elementID也需要重新生成
  + 如果要和云屋产品互通，那就需要按云屋的定义取值，可以联系云屋获取相关文档，网盘文件