



TalkCloud Room SDK for Android

参考手册

1.0 版

support@talk-cloud.com

© 2018 talk_cloud.com 2018 年 6 月

TalkCloud Room SDK for Android 简介	5
TalkCloud Room SDK for Android - 接口 参考	12
1. TKRoomManager 接口类	12
全局初始化 (init)	12
获取 TKRoomManager 单例 (getInstance)	12
设置日志打印级别 (setLoggerInfo)	12
设置回调 (registerRoomObserver)	13
进入房间 (joinRoom)	13
离开房间 (leaveRoom)	14
获得房间类型 (getRoomType)	14
获得房间名字 (getRoomName)	14
获得房间的详细属性 (getRoomProperties)	14
获得本地用户 (getMySelf)	14
获得用户 (getUser)	15

获得当前房间内的所有用户（getUser）	15
播放某个用户的视频（playVideo）	15
停止播放某个用户的视频（unPlayVideo）	15
修改某个用户的一个属性（changeUserProperty）	16
修改某个用户的音视频发布状态（changeUserPublish）	16
发送聊天消息（sendMessage）	16
发布自定义消息（pubMsg）	17
删除自定义消息（delMsg）	17
将一个用户踢出房间（evictUser）	18
切换本地摄像头（selectCameraPosition）	18
当前本地麦克风是否被启用（isLocalAudioEnabled）	18
设置启用/禁用麦克风（enableLocalAudio）	19
设置是否使用扬声器播放声音（useLoudSpeaker）	19
暂停本地摄像头使用（pauseLocalCamera）	19
恢复本地摄像头使用（resumeLocalCamera）	19
开始录制(startRecordStream)	19
结束录制(stopRecordStream)	19
设置房间视频分辨率(setVideoProfile)	20
播放音频(playAudio)	20
取消播放音频(unPlayAudio)	20
播放网络共享媒体(playMedia)	20
停止播放网络媒体(unPlaymedia)	20
播放屏幕共享（playScreen）	21
停止播放屏幕共享(unPlayScreen)	21
播放文件共享媒体(playFile)	21
停止播放文件共享媒体(unPlayFile)	21
发布自己的视频(publishVideo)	21
取消发布自己的视频(unPublishVideo)	21

发布自己的音频(publishAudio)	21
取消发布自己的音频(unPublishAudio)	22
设置某一路视频音量(setRemoteAudioVolume)	22
开始共享网络媒体(startShareMedia)	22
停止共享网络媒体(stopShareMedia)	22
设置网络媒体播放进度(seekMedia)	22
设置扩展属性(setAttributes)	22
切换纯音频教室 (switchOnlyAudioRoom)	23
获取大并发教室人数(getRoomUserNum)	23
获取大并发教室成员列表(getRoomUser)	23
开始录制件接口 (startServerRecord)	23
结束录制件接口 (stopServerRecord)	24
2. TKRoomManagerObserver 接口类	24
成功进入房间的回调 (onRoomJoined)	24
离开房间的回调 (onRoomLeaved)	24
错误回调 (onError)	24
警告回调 (onWarning)	24
其他用户进入房间的回调 (onUserJoined)	25
其他用户离开房间 (onUserLeft)	25
用户属性改变 (onUserPropertyChanged)	25
用户音视频状态改变 (onUserPublishState)	25
自己被请出房间的回调 (onKickedout)	25
收到文本消息 (onMessageReceived)	26
收到信令消息回调 (onRemotePubMsg)	26
收到信令消息删除回调 (onRemoteDelMsg)	26
网络媒体播放进度, 状态回调 (onUpdateAttributeStream)	27
有网络媒体文件共享的回调 (onShareMediaState)	27

有屏幕共享的回调（onShareScreenState）	27
有文件媒体共享的回调（onShareFileState）	28
音量回调（onAudioVolume）	28
视频监控数据（onVideoStatsReport）	28
音频监控数据（onAudioStatsReport）	28
切换纯音频教室回调（onAudioRoomSwitch）	29
获取大并发用户列表回调（onGetRoomUsersBack）	29
获取大并发教室中人数（onGetRoomUserNumBack）	29
纯音频教室切换的回调（onAudioRoomSwitch）	
视频数据首帧回调（onFirstVideoFrame）	
音频数据首帧回调（onFirstAudioFrame）	
3. TKMediaFrameObserver 接口.....	30
音频本地原始数据（onCaptureAudioFrame）	31
音频远端原始数据（onRenderAudioFrame）	31
视频本地原始数据（onCaptureVideoFrame）	31
视频远端原始数据（onRenderVideoFrame）	31
4. RoomUser 类.....	32
错误码参考.....	33
FAQ.....	38

TalkCloud Room SDK for Android 简介

为客户提供基于互联网的可靠的音视频技术服务在线教育产业。基于新一代通信协议标准打造的拓课云课堂，真正做到全平台互通，轻松实现国际间实时互动教学直播，丰富的

API 接口及 SDK, 可帮助机构更快速的接入原有业务系统, 并且可以根据业务场景, 定制开发相应的功能, 确保满意的用户体验质量。TalkCloud 致力于解决移动设备的用户体验问题、3G/4G/Wi-Fi 网络性能各异、全球网络瓶颈等一系列问题, 为用户带来优质的互动体验。

TalkCloud PaaS 包含用于移动设备的 Android SDK(安卓智能手机研发) 和 iOS SDK (iOS 智能手机研发), 其通信基于 TalkCloud Global Network, 并针对不同移动设备平台进行优化。TalkCloud Room SDK 也有 Windows 和 Mac 版本。

TalkCloud Room SDK 在程序生成时直接与应用程序建立连接。在下文的使用入门一节中将为您阐述如何使用 TalkCloud Room SDK 组件为 Android 平台搭建工程环境。

除上述产品外, 另有 TalkCloud Room Web SDK 针对网页通信开发, 为任何支持标准 WebRTC (网络实时通信) 的网页浏览器提供 TalkCloud Global Network 通信支持, 而无需额外下载或安装插件。详见 TalkCloud Room Web SDK 参考手册。

TalkCloud Room SDK for Android 实现以下功能:

- **建立通话:** 加入或离开会话 (会话以频道为标识), 既可实现全球用户多方会话, 也可是一对一的聊天对话。频道应为唯一的, 由 TalkCloud WebApi 创建及维护, 通常由用户、教室和时间信息组成。
- **音视频控制:** 启用和禁用音视频、静音设置、并可设置多种音视频参数以达到通话最佳效果。TalkCloud SDK 的大部分操作均设置有默认值, 因此即使未给参数赋值也可运行。
- **设备控制:** 访问麦克风或话筒、音量设置、选择摄像头、设置视频显示等。
- **通话管理:** 当其他用户加入或离开通话 (频道) 时可接收事件通知, 可获知通话对方是谁, 静音的用户是谁以及哪个用户在观看视频通话等等。这些事件通知有助于应用程序决定是否对音视频流做调整, 或修改用户及状态信息等。
- **质量管理:** 获取网络质量统计数据, 以及启用各级错误记录等。

获取 TalkCloud Room SDK for Android

您可登录 <http://www.talk-cloud.com> 网站下载 TalkCloud Room SDK for Android, 或联系 sales@talk-cloud.com 获取最新版本。

获取 authkey

您所获取的 authkey 是唯一的，它将在 TalkCloud Room SDK for Android 被应用程序调用时使用，详见示例程序和本手册 API 参考文档一节。在示例程序中启用音频或视频通话时，需要在启动窗口中输入您的 authkey。在开发程序时，需要将 authkey 嵌入您的代码中以便调用 SDK。

如何获取 authkey:

通过 TalkCloud 管理人员注册并开通企业账户后，该 authkey 也就自动有了。

搭建工程环境要求

开发者可使用 Android Studio（推荐），这是 Android 官方集成开发环境。如果使用其他开发环境需要额外支持，请联系 support@talk-cloud.com。

所需库

TalkCloud Room SDK for Android 开发者可使用其接口进行开发。详见本手册接口参考一节。

Android SDK 中有 libs 文件夹

- libs 包含 sdk 的相关文件 classroomsdk-release.aar

搭建工程

重要概念

1.1 房间

使用本 SDK 之前，您需要使用 webapi 或访问管理系统页面来创建房间。使用获得的房间号即可与同房间用户进行音视频和其他数据的交换了。

1.2 用户

使用网页、客户端或手机 app，皆可进入房间，每一个接入房间的客户端实例，我们称为一个用户。

1.3 音视频

每个具有可用的音视频采集设备的用户都可以单独发布自己的音频数据，也可以同时发布音视频数据。其他用户会自动接收房间内已经发布的音频，可选择接收已发布的视频。

1.4 自定义信令

利用自定义信令接口，您可以完成您设计的各种业务流程，创造各种激动人心的用户体验。我们把各种信令都抽象成了消息，和用户类似，消息有自己的 id (msgId) 和昵称 (msgName)，昵称可以和其他消息重名，但 id 必须在房间内唯一，id 相同的消息，后来的会替换之前的。除了这两个属性，消息还有这些属性：

toId: 谁会接收到该消息的通知。

data: 数据（可存放各种数据结构，包括数字、字符串、Json）

save: 是否保存。当为真时，服务器会在房间内保存该条消息，新进入的用户会收到该条消息。如果连续发两条 id 相同且需要保存的消息，服务器只会保存后一条。；当为假时，服务器在转发完这条消息后，即将此消息抛弃

具体流程如下：

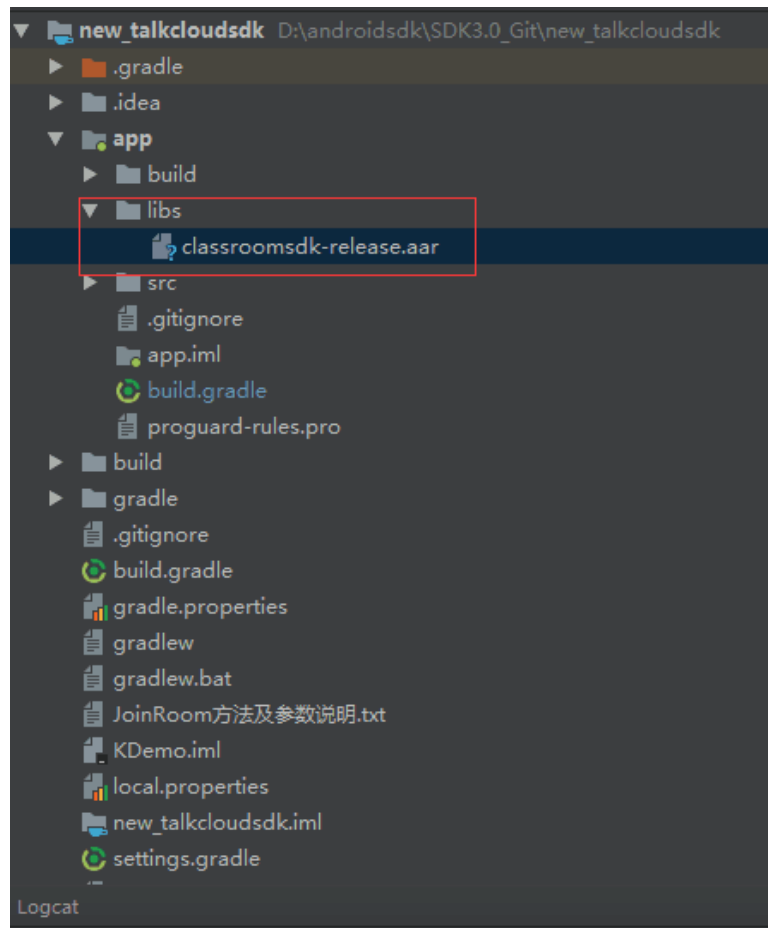
调用 pubMsg 接口发布一条消息，则所有指定用户将收到事件 room-pubmsg（pubMsg 消息事件）。

如果这条消息是指定保存的，则之后新进入房间的用户会收到事件 room-connected(房间连接成功事件)的数据中带有当前服务器的所有保存的且未删除的消息数组。在发布消息之后，可以再次调用 pubMsg 接口，发布一条相同 id 的消息来覆盖它，则指定用户会再次收到事件 room-pubmsg (pubMsg 消息发布事件)。也可以调用 delMsg 接口来删除此消息，此时指定用户会收到事件 room-delmsg (delMsg 消息删除事件)。

用 Android studio 打开工程，如下所示：

假设您下载我们的 TalkCloud Room SDK for Android 并解压到 F:\talk_sdk_all\TKRoomSDK_Android_2.2.1\demo\new_talkcloudsdk 目录下

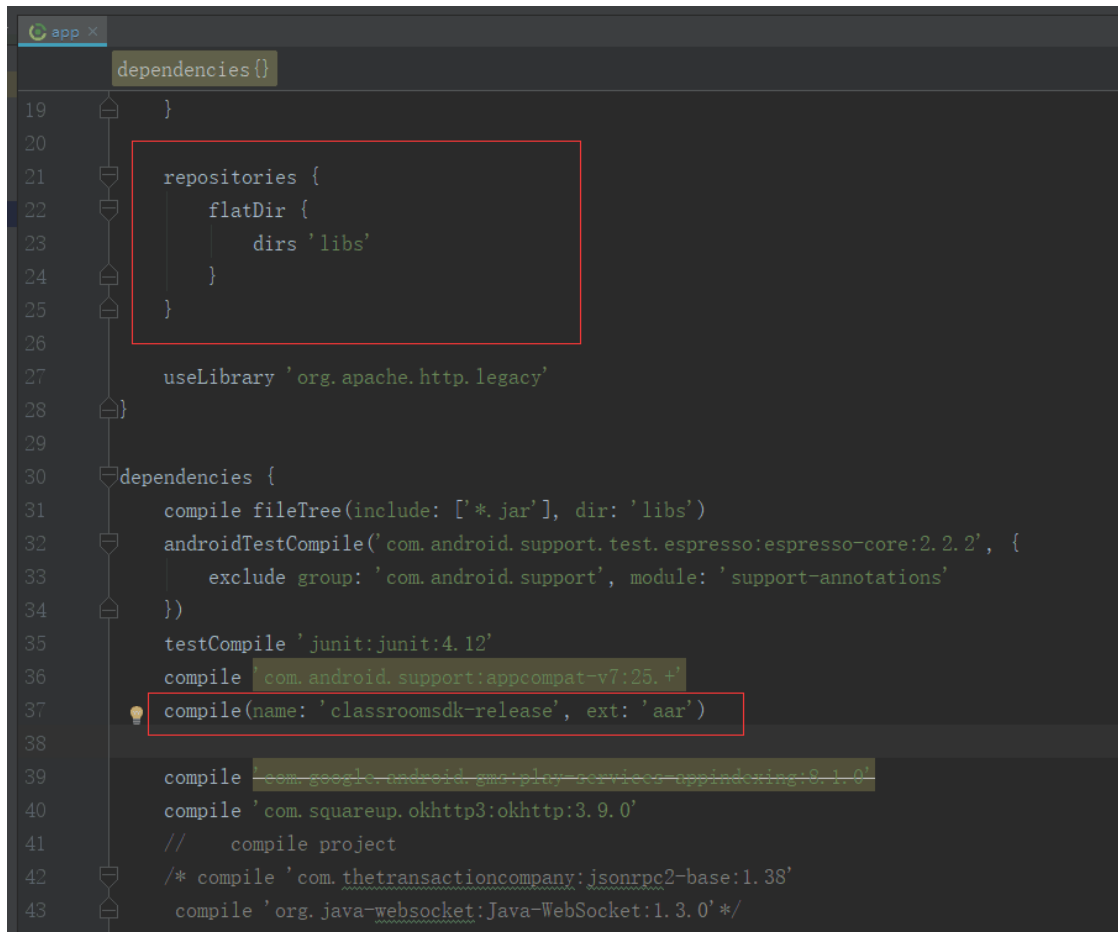
将项目 classroomsdk-release.aar 导入你的项目中：如下图



其中 app 是我们的主要 module, libs 下存放外部的 library; build.gradle 是配置 APP module 的，接下来复制 **classroomsdk-release.aar** 包到 libs 目录下。

添加repositories flatDir { dirs 'libs' }}这个是添加一个本地仓库,并把libs目录作为仓库的地址

修改dependencies:添加一行 compile(name:'classroomsdk-release', ext:'aar') 其中 name 就是 libs 目录下 classroomsdk-release.aar 文件名称, ext, 就是 classroomsdk-release.aar 的扩展名, 如下图

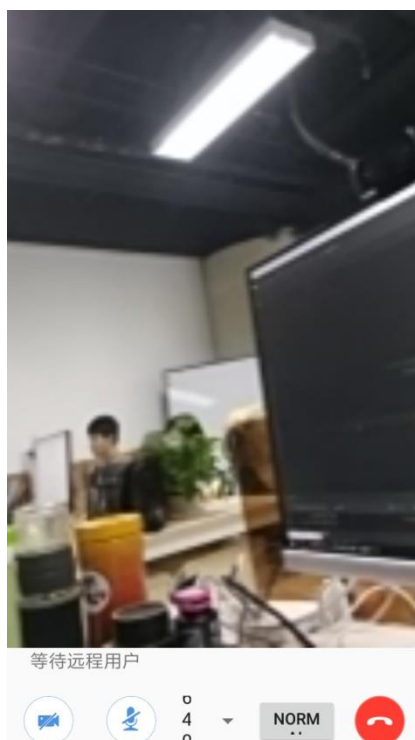


配置 Android Manifest.xml, 添加下面权限

```
<uses-library android:name="android.test.runner"/>  
<uses-feature android:name="android.hardware.camera" />  
<uses-feature android:name="android.hardware.camera.autofocus" />  
<uses-feature android:glEsVersion="0x00020000"  
    android:required="true" />  
<uses-permission  
    android:name="android.permission.ACCESS_NETWORK_STATE" />  
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"  
/>
```

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission
android:name="android.permission.CAPTURE_VIDEO_OUTPUT" />
<uses-permission
android:name="android.permission.CHANGE_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WAKE_LOCK" />
```

Demo 运行效果



TalkCloud Room SDK for Android - 接口 参考

1. TKRoomManager 接口类

TKRoomManager 类是使用 TalkCloud Room SDK forAndroid 的主要接口类

全局初始化（init）

// 应用启动后，只需要调用一次

```
public static void init(Context applicationContext,String  
                        appId,HashMap<String,Object> params);
```

名称	描述
applicationContext	应用上下文。请传入一个使用 getApplicationContext方法获得的context
appId	为企业authkey 企业注册时获得的 authkey
Params（可以为null）	配置参数 例： HashMap<String, Object> params = new HashMap<>(); params.put(TKRoomManager.autoSubAV, true) ; TKRoomManager.getInstance().init(getAppl icationContext(),"test", params);

获取 TKRoomManager 单例（getInstance）

```
public static TKRoomManager getInstance();
```

设置日志打印级别（setLogFilter）

```
public static void setLogFilter(int level);
```

名称	描述
level	日志信息级别

设置日志打印地址（[setLogFile](#)）

```
public static int setLogFile(String filepath);
```

名称	描述
filepath	日志的本地打印地址，注：地址必须存在且可写

设置回调（[registerRoomObserver](#)）

```
public void registerRoomObserver(TKRoomManagerObserver callBack);
```

名称	描述
<i>callback</i>	实现了TKRoomManager回调接口的对象，详见本文下一节

进入房间（[joinRoom](#)）

// 您需要提前预约好房间，详见会议接口文档

// 返回值：请见错误码定义

```
public int joinRoom(final String host, final int port, String nickname, Map<String, Object> params, Map<String, Object> properties, boolean useOpenGL);
```

名称	描述
host	服务器地址，通常是global.talk-cloud.net
port	服务器http端口，通常是80
nickname	本地用户的昵称
params	map格式，内含进入房间所需的基本参数 例如： <i>HashMap<String, Object> params = new HashMap<String, Object>(); params.put("domain", "www");</i> 企业域名（用您的

	企业账号登录后台系统可以查到) params.put("serial", "12345678"); 房间号 params.put("password", "1111"); 预约房间时 指定的密码, 注意不同身份对应不同密码
properties	map格式, 可为null 用户自定义属性
useOpenGL	使用 OpenGL 进行视频渲染。通常是 true, 除非您 知道 false 意味着什么, 并且打算这么做

离开房间 (leaveRoom)

// 返回值: 请见错误码定义

```
public int leaveRoom() ;
```

```
public int leaveRoom(boolean force)
```

名称	描述
force	布尔型, true -- 强制用户离开

获得房间类型 (getRoomType)

// 请在进入房间成功后调用

// 返回值: 房间类型。0: 一对一; 1: 小班

```
public int getRoomType();
```

获得房间名字 (getRoomName)

// 请在进入房间成功后调用

// 返回值: 房间类型。0: 一对一; 1: 小班

```
public final String getRoomName();
```

获得房间的详细属性 (getRoomProperties)

// 请在进入房间成功后调用

// 返回值: 房间属性 json 格式

```
public final JSONObject getRoomProperties ();
```

获得本地用户 (getMySelf)

// 请在进入房间成功后调用

// 返回值：本地用户对象

```
public final RoomUser getMySelf();
```

获得用户（getUser）

// 请在进入房间成功后调用

// 返回值：本地用户对象

```
public final RoomUser getUser(String peerID);
```

名称	描述
peerID	用户 ID

获得当前房间内的所有用户（getUsers）

// 请在进入房间成功后调用

// 返回值：所有用户的 Map，Key 是 peerID，value 是用户

```
public final Map<String, RoomUser>getUsers();
```

播放某个用户的视频（playVideo）

// 请在该用户发布视频后再调用

// 对同一个用户，可以调用多次此函数。当传入的 view 和上次传入的一致时，函数不执行任何操作，直接返回成功；当传入的 view 和上次传入的不一致时，换用新的 view 播放该用户的视频

// 返回值：请见错误码定义

```
public int playVideo(String peerId, Object view, RendererCommon.ScalingType mode);
```

名称	描述
peerID	用户 ID
view	用于播放视频的surfaceView
mode	视频缩放类型

停止播放某个用户的视频（unPlayVideo）

// 返回值：请见错误码定义

```
public int unPlayVideo(String peerID);
```

名称	描述
----	----

peerID	用户ID
--------	------

修改某个用户的一个属性（changeUserProperty）

// 返回值：请见错误码定义

```
public int changeUserProperty(String peerID, String tellWhom, HashMap<String, Object>
property);
```

名称	描述
peerID	要修改的用户ID
tellWhom	要将此修改通知给谁。 “__All”：所有人； “__allExceptSender”：除自己以外的所有人；“__allExceptAuditor”：除旁听用户以外的所有人； “__None”：不通知任何人； 某用户的 peerID：只发给该用户
property	Map string：修改的用户属性名字，可以是您自定义的名字 Object：要修改的用户属性，可以是 Number、String、Map 或 List

修改某个用户的音视频发布状态（changeUserPublish）

// 返回值：请见错误码定义

```
public int changeUserPublish(String peerID, int publish);
```

名称	描述
peerID	该用户的peerID，可以是自己的，也可以是其他人的
publish	0：不发布；1：只发布音频；2：只发布视频；3：发布音视频

发送聊天消息（sendMessage）

// 返回值：请见错误码定义

```
public int sendMessage(String message, int type);
```

名称	描述
message	聊天内容
type	消息类型

发布自定义消息（pubMsg）

// 返回值：请见错误码定义

```
public int pubMsg(String msgName, String msgId, String toId, Object data, boolean save,
String associatedMsgID, String associatedUserID);
```

名称	描述
msgName	消息名字
msgId	消息id
toId	要通知给哪些用户。 “__all”：所有人； “__allExceptSender”：除自己以外的所有人； “__allExceptAuditor”：除旁听用户以外的所有人； “__None”：不通知任何人； 某用户的peerID：只发给该用户
data	消息数据，可以是Number、String、Map或List
save	是否保存，详见：自定义信令
associatedMsgID	绑定的父级信令id
associatedUserID	绑定的用户id

删除自定义消息（delMsg）

// 返回值：请见错误码定义

```
public int delMsg(String msgName, String msgId, String toId, Object data);
```

名称	描述
msgName	消息名字
msgId	消息id
toId	要通知给哪些用户。 “__all”：所有人； “__allExceptSender”：除自己以外的所有人； “__allExceptAuditor”：除旁听用户以外的所有人； “__None”：不通知任何人； 某用户的peerID：只发给该用户
data	消息数据，可以是Number、String、Map或List

将一个用户踢出房间（evictUser）

// 返回值：请见错误码定义

```
public int evictUser(String peerID, int reason);
```

名称	描述
peerID	该用户的id
reason	踢人的原因

切换本地摄像头（selectCameraPosition）

// 返回值：请见错误码定义

```
public int selectCameraPosition(boolean front);
```

名称	描述
front	true：使用前置摄像头；false：使用后置摄像头

当前本地麦克风是否被启用（isLocalAudioEnabled）

// 返回值：true：麦克风可用；false：麦克风被静音

```
public boolean isLocalAudioEnabled();
```

设置启用/禁用麦克风（enableLocalAudio）

// 返回值：请见错误码定义

```
public int enableLocalAudio(boolean enable);
```

名称	描述
enable	true：启用麦克风；false：禁用麦克风

设置是否使用扬声器播放声音（useLoudSpeaker）

// 返回值：请见错误码定义

```
public int useLoudSpeaker(boolean use) ;
```

名称	描述
use	true：使用扬声器；false：使用电话听筒

暂停本地摄像头使用（pauseLocalCamera）

```
public int pauseLocalCamera();
```

恢复本地摄像头使用（resumeLocalCamera）

```
public int resumeLocalCamera();
```

开始录制（startRecordStream）

```
public void startRecordStream (String peerid, int convert, final StreamRecordCallBack  
callBack);
```

名称	描述
peerid	用户 ID
conver	是否转换格式 0 - 不转换, 1 - webm, 2 - mp4
callBack	i :响应码 0 为成功 非 0 失败 s: 录制的地址 成功有值 失败没有

结束录制（stopRecordStream）

```
public void stopRecordStream(String peerid, final StreamRecordCallBack callBack);
```

名称	描述
peerid	用户 ID

callback	i :响应码 0 为成功 非 0 失败 s: null
----------	------------------------------------

设置房间视频分辨率 (setVideoProfile)

`public void setVideoProfile(int width, int height)`

名称	描述
width	视频宽
height	视频高

播放音频 (playAudio)

`public void playAudio(String peerid)`

名称	描述
peerId	用户 id

取消播放音频 (unPlayAudio)

`public void unPlayAudio(String peerId)`

名称	描述
peerId	用户 id

播放网络共享媒体 (playMedia)

`public void playMedia(String peerId, Object view)`

名称	描述
peerId	用户 id
view	用于播放视频的 surfaceView

停止播放网络媒体 (unPlaymedia)

`public void unPlaymedia(String peerId)`

名称	描述
peerId	用户 id

播放屏幕共享 (playScreen)

```
public void playScreen(String peerId, Object view)
```

名称	描述
peerId	用户 id
view	用于播放视频的 surfaceView

停止播放屏幕共享 (unPlayScreen)

```
public void unPlayScreen(String peerId)
```

名称	描述
peerId	用户 id

播放文件共享媒体 (playFile)

```
public void playFile(String peerId, Object view)
```

名称	描述
peerId	用户 id
view	用于播放视频的 surfaceView

停止播放文件共享媒体 (unPlayFile)

```
public void unPlayFile(String peerId)
```

名称	描述
peerId	用户 id

发布自己的视频 (publishVideo)

```
public void publishVideo()
```

取消发布自己的视频 (unPublishVideo)

```
public void unPublishVideo()
```

发布自己的音频 (publishAudio)

```
public void publishAudio()
```

取消发布自己的音频(unPublishAudio)

```
public void unPublishAudio()
```

设置某一路视频音量(setRemoteAudioVolume)

```
public void setRemoteAudioVolume(double volume, String peerId, int type)
```

名称	描述
volume	音量
peerId	用户 id
type	流类型 1 video, 2 MP3/MP4 media, 3 screen, 4 file media

开始共享网络媒体(startShareMedia)

```
public void startShareMedia(String url, boolean isVideo, String toWho, Map<String, Object> attsMap)
```

名称	描述
url	网络文件地址
isVideo	是否是视频
toWho	共享给谁
attsMap	扩展参数

停止共享网络媒体(stopShareMedia)

```
public void stopShareMedia()
```

设置网络媒体播放进度(seekMedia)

```
public void seekMedia(long pos)
```

名称	描述
pos	进度时间戳

设置扩展属性(setAttributes)

```
public void setAttributes(Map<String, Object> attributes)
```

名称	描述
attributes	扩展属性字典

切换纯音频教室（switchOnlyAudioRoom）

```
public void switchOnlyAudioRoom(boolean isSwitch);
```

名称	描述
isSwitch	true-切换， false-不切换

获取大并发教室人数（getRoomUserNum）

```
public void getRoomUserNum(int[] role)
```

名称	描述
role	身份数组

获取大并发教室成员列表（getRoomUser）

```
public void getRoomUser(int[] role, int start, int max)
```

名称	描述
role	身份数组
start	起始
max	每次获取用户的最大数量

开始录制件接口（startServerRecord）

```
public int startServerRecord(int recordtype, int convert, int layout, long expiresabs, long expires)
```

名称	描述
recordtype	缺省： 0； 0:生成录制件 1 生成视频列表（调用 getuserrocordlist，获取视频） 2： 生成 mp3（只会生成 mp3） 3： 混屏（会同时生成 mp3 和 mp4）
convert	只有在 recordtype =0 与 1 的情况下起作用 缺省： 0； 0 不转换（mkv） 1 webm（recordtype 其他值时，固定状态） 2 mp4

layout	只有在 recordtype =3 的情况下起作用 缺省： 0 ; 0 横屏 1 纵屏
expiresabs	
expires	

结束录制件接口（stopServerRecord）

```
public int stopServerRecord()
```

2. TKRoomManagerObserver 接口类

是 TKRoomManager 接口类的回调接口

```
public interface TKRoomManagerObserver{
```

成功进入房间的回调（onRoomJoined）

```
void onRoomJoined();
```

离开房间的回调（onRoomLeaved）

```
void onRoomLeaved();
```

错误回调（onError）

```
void onError(int errorCode, String errMsg);
```

名称	描述
errorCode	错误码
errMsg	错误信息

警告回调（onWarning）

```
void onWarning(int warning);
```

名称	描述
warning	10001 摄像头打开 10002 摄像头关闭

其他用户进入房间的回调（onUserJoined）

```
void onUserJoined(RoomUser user, boolean inList);
```

名称	描述
user	进入的用户
inList	是否在我之前进入房间，true—之前，false—之后

其他用户离开房间（onUserLeft）

```
void onUserLeft(RoomUser user);
```

名称	描述
user	离开的用户

用户属性改变（onUserPropertyChanged）

```
void onUserPropertyChanged(RoomUser user, Map<String, Object> changedProperties, String fromId);
```

名称	描述
user	改变属性的用户
changedProperties	改变的属性集合
fromId	改变该用户属性的用户的用户 ID

用户音视频状态改变（onUserPublishState）

```
void onUserPublishState(String userId, int state);
```

名称	描述
userId	用户 id
state	音视频状态

自己被请出房间的回调（onKickedout）

```
void onKickedout(int reason);
```

名称	描述
reason	原因

收到文本消息（onMessageReceived）

```
void onMessageReceived(RoomUser user, JSONObject message, long ts);
```

名称	描述
user	发送文本消息的用户
message	文本消息，json 格式
ts	消息发送时间戳

收到信令消息回调（onRemotePubMsg）

```
void onRemotePubMsg(String id, String name, long ts, Object data, boolean inList, String fromID, String associatedMsgID, String associatedUserID);
```

名称	描述
id	消息 id
name	消息名字
ts	消息发送时间戳
data	消息携带数据
inList	消息是否在消息列表中
fromID	发送者 id
associatedMsgID	消息关联消息的 id（该消息删除时会跟随删除）
associatedUserID	消息关联用户的 id（该用户退出时会跟随删除）

收到信令消息删除回调（onRemoteDelMsg）

```
void onRemoteDelMsg(String id, String name, long ts, Object data, boolean inList, String fromID, String associatedMsgID, String associatedUserID);
```

名称	描述
id	消息 id
name	消息名字
ts	消息发送时间戳
data	消息携带数据
inList	消息是否在消息列表中
fromID	发送者 id
associatedMsgID	消息关联消息的 id（该消息删除时会跟随删除
associatedUserID	消息关联用户的 id（该用户退出时会跟随删除）

网络媒体播放进度，状态回调（onUpdateAttributeStream）

```
void onUpdateAttributeStream(Stream stream, long pos, boolean isplay, HashMap<String, Object> attrMap);
```

名称	描述
stream	具体媒体文件流
pos	播放进度
isplay	是否在播放
attrMap	流自定义扩展属性

有网络媒体文件共享的回调（onShareMediaState）

```
void onShareMediaState(String peerId, int state, Map<String, Object> attrs);
```

名称	描述
peerId	共享者用户 id
state	媒体共享状态 0 停止 1 开始
attrs	自定义数据

有屏幕共享的回调（onShareScreenState）

```
void onShareScreenState(String peerId, int state);
```

名称	描述
peerId	共享者用户 id
state	媒体共享状态 0 停止 1 开始

有文件媒体共享的回调（onShareFileState）

```
void onShareFileState(String peerId, int state);
```

名称	描述
peerId	共享者用户 id
state	媒体共享状态 0 停止 1 开始

音量回调（onAudioVolume）

```
void onAudioVolume(String peerId, int volume);
```

名称	描述
peerId	用户 id
volume	音量

视频监控数据（onVideoStatsReport）

```
void onVideoStatsReport(String peerId, TkVideoStatsReport statsReport);
```

名称	描述
peerId	用户 id
statsReport	网络状态信息 TK_NET_QUALITY_EXCELLENT = 1; --非常好 TK_NET_QUALITY_GOOD = 2; --好 TK_NET_QUALITY_ACCEPTED = 3; --可接受 TK_NET_QUALITY_BAD = 4; -- 坏的 TK_NET_QUALITY_VERYBAD=5; --非常坏

音频监控数据（onAudioStatsReport）

```
void onAudioStatsReport(String peerId, TkAudioStatsReport statsReport);
```

名称	描述
peerId	用户 id

statsReport	网络状态信息 TK_NET_QUALITY_EXCELLENT = 1; --非常好 TK_NET_QUALITY_GOOD = 2; --好 TK_NET_QUALITY_ACCEPTED = 3; --可接受 TK_NET_QUALITY_BAD = 4; -- 坏的 TK_NET_QUALITY_VERYBAD=5; --非常坏
-------------	---

切换纯音频教室回调（onAudioRoomSwitch）

```
void onAudioRoomSwitch (String fromId, boolean isSwitch);
```

名称	描述
fromId	执行该命令的用户 ID
isSwitch	true-切换， false-不切换

获取大并发用户列表回调（onGetRoomUsersBack）

```
void onGetRoomUsersBack(int errorCode, ArrayList<RoomUser> arrayList);
```

名称	描述
errorCode	获取是否成功 0-成功
arrayList	用户列表（成功）

获取大并发教室中人数（onGetRoomUserNumBack）

```
void onGetRoomUserNumBack(int errorCode, int num);
```

名称	描述
errorCode	获取是否成功 0-成功
num	人数

视频数据首帧回调（onFirstVideoFrame）

```
void onFirstVideoFrame(String peerID, int mediaType, int width, int height);
```

名称	描述
peerID	用户 id

mediaType	媒体类型 TK_MEDIA_UNKNOW = -1; TK_MEDIA_CAMERA = 0; TK_MEDIA_MIC = 11; TK_MEDIA_SPEAKER = 12; TK_MEDIA_FILE = 101; TK_MEDIA_SCREEN = 102; TK_MEDIA_MEDIA = 103;
width	视频宽
height	视频高

音频数据首帧回调（onFirstAudioFrame）

`void onFirstAudioFrame(String peerID, int mediaType);`

名称	描述
peerID	用户 id
mediaType	媒体类型 TK_MEDIA_UNKNOW = -1; TK_MEDIA_CAMERA = 0; TK_MEDIA_MIC = 11; TK_MEDIA_SPEAKER = 12; TK_MEDIA_FILE = 101; TK_MEDIA_SCREEN = 102; TK_MEDIA_MEDIA = 103;

}

3. TKMediaFrameObserver 接口

```
public interface TKRoomManagerObserver{
```

音频本地原始数据（onCaptureAudioFrame）

```
boolean onCaptureAudioFrame(TKAudioFrame audioFrame, String uid, int type);
```

名称	描述
audioFrame	音频本地原始数据
uid	用户 id
type	类型

音频远端原始数据（onRenderAudioFrame）

```
boolean onRenderAudioFrame(TKAudioFrame audioFrame, String uid, int type);
```

名称	描述
audioFrame	音频远端原始数据
uid	用户 id
type	类型

视频本地原始数据（onCaptureVideoFrame）

```
boolean onCaptureVideoFrame(TKVideoFrame videoFrame, String id);
```

名称	描述
videoFrame	视频本地原始数据
id	用户 id
type	类型

视频远端原始数据（onRenderVideoFrame）

```
boolean onRenderVideoFrame(TKVideoFrame videoFrame, String uid, int type);
```

名称	描述
videoFrame	视频远端原始数据
uid	用户 id
type	类型

```
}
```

4. RoomUser 类

```
public class RoomUser {

    public static int PUBLISH_STATE_NONE = 0;
    public static int PUBLISH_STATE_AUDIOONLY = 1;
    public static int PUBLISH_STATE_VIDEOONLY = 2;
    public static int PUBLISH_STATE_BOTH = 3;

    public String nickName = ""; // 用户昵称
    public String peerId = ""; // 用户 Id
    public int role = -1; // 用户身份, 0: 老师; 1: 助教; 2: 学生; 3: 旁听; 4: 隐身用户

    public boolean hasAudio = true; // 该用户是否有麦克风
    public boolean hasVideo = true; // 该用户是否有摄像头
    public boolean canDraw = true; // 该用户是否有权在白板和文档上进行绘制
    public int publishState = PUBLISH_STATE_NONE; // 发布状态, 0: 未发布, 1: 发布音频; 2: 发布视频; 3: 发布音视频
    public JSONObject properties; // 用户属性。Sdk 调用者设置的自定义属性会在这里被找到

}
```

错误码参考

错误码	表示
-10	上传文件大于 100（文件页数超限）
-8	该房间不是本企业
-6	非公开文件，但未指定房间
-5	未指定房间号或 key
-4	非法文件类型
-3	文件移动失败
-2	文件上传失败
-1	内部异常
0	表示成功
1	尚未初始化

2	当前状态不适合此操作
3	web 接口请求失败
4	方法参数错误
5	无此用户
6	此用户尚未发布视频
7	尚未播放此用户的视频
3001	服务器过期
3002	公司已经冻结
3003	房间已删除或过期
4001	该公司不存在
4002	用户名或密码错误
4003	房间名称不允许为空
4004	时间格式错误
4005	时间设置有误
4006	老师密码格式错误
4007	房间不存在、房间无效已被删除、房间过期
4008	房间密码错误
4009	学生用户密码格式错误
4010	密码格式错误
4011	房间类型必填
4101	需要验证企业 ID
4102	验证企业 ID 错误
4103	房间数超过最大限制
4105	没有符合的数据
4106	房间文件数量超过数量限制
4107	数据库操作错误
4108	没有权限

4109	Auth 不正确
4110	该房间需要密码，请输入密码
4111	非注册用户或密码不正确，该房间要求必须注册用户才能参会
4200	组 id 必填
4201	组名必填
4202	文件 id 必填
5000	用户名密码必填
5001	房间是否有密码必填
5002	主席密码必填
5003	房间普通密码必填
5004	开始时间，结束时间必填
5005	如果允许旁听用户则应有密码
5006	房间学员密码格式错误
5007	旁听用户密码错误
5008	最大用户数格式错误
5009	最大用户数超过限制
5010	最大旁听用户数错误
5011	最大旁听用户数超过限制
5012	最大视频数格式错误
5013	最大视频数错误
5014	最大音频数格式错误
5015	最大音频数错误
5016	视频分辨率参数格式错误
5017	布局参数格式错误
5018	助教密码必填
5019	巡课密码必填
5020	视频分辨率错误或视频分辨率超过企业限制

5100	需要房间编号
5101	需要用户账号
5102	房间密码不可重复
5200	主席密码错误
5201	旁听密码错误
5203	学员密码
5204	没有密码
5205	昵称或名字必填
5206	数据写入失败
5207	用户类型 0: 主讲(老师) 1: 助教 2: 学员 3: 直播用户 4:巡检员
5208	用户已登陆
5300	登录 40 位 key 必填
5301	40 位 key 错误
6000	serverid 不正确
6001	ip 地址不正确
6002	普通用户数超出范围
6003	旁听用户数超出范围
7001	userid 必填
7002	accout 必填
7003	密码必填
7004	部门 ID 必填
7005	公司 ID 必填，缺省为 1
7006	Accout 已经存在
7007	用户账号不存在
8001	部门 id 必填
8002	部门名称必填

8003	公司 ID 必填，缺省为 1
8004	父部门 ID 必填
8005	部门级别必填
9001	公司 ID 必填
9002	公司名必填
9003	二级域名必填
9004	公司管理员 ID
9006	该公司已经存在
9007	企业 ID 必填
10001	摄像头丢失
10002	socket 5 次链接失败，应退出教室
10004	udp 链接异常

FAQ

Q:SDK 支持哪些 CPU 版本?

A:安卓 SDK 目前支持 x86、x86_64、armv7、arm64;