

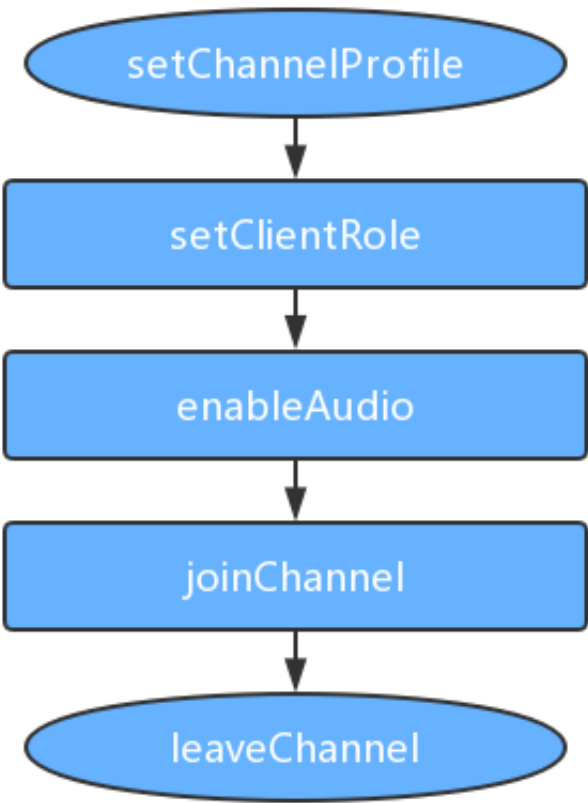
# API参考

Java 接口类	描述
主要方法 (TTTRtcEngine)	TTTRtcEngine接口类包含应用程序调用的主要方法
回调事件 (TTTRtcEngineEventHandler)	TTTRtcEngineEventHandler接口类用于向应用程序发送回调通知

## 主要方法 (TTTRtcEngine)

所有返回值为 int 型的 API，如无特殊说明，返回值 0 为调用成功，返回值小于 0 为调用失败。

语音直播流程图



创建 TTTRtcEngine对象 (create)

---

```
public static TTTRtcEngine create(Context context, String appId, T
TTTRtcEngineEventHandler handler);
```

初始化SDK，在程序生命周期中只需要调用一次，即便执行destroy函数也不需要调用。

名称	描述
context	安卓程序的上下文
appId	SDK初始化需要用到的app id
handler	SDK发送回调消息的接收对象
返回值	返回SDK的实例对象

### 设置频道模式(setChannelProfile)

```
public int setChannelProfile(int profile);
```

设置频道模式。频道模式如下：

- Constants.CHANNEL\_PROFILE\_COMMUNICATION 通信频道模式
- Constants.CHANNEL\_PROFILE\_LIVE\_BROADCASTING 直播频道模式
- Constants.CHANNEL\_PROFILE\_GAME\_FREE\_MODE 游戏频道模式

名称	描述
profile	频道模式
返回值	0代表方法调用成功，其他代表失败。

### 设置用户角色 (setClientRole)

```
public int setClientRole(int role, String permissionKey)
```

设置用户角色。在加入频道前，用户需要通过本方法设置观众(默认)或主播模式。

名称	描述
role	直播场景里的用户角色
permissionKey	连麦鉴权功能暂时没开放，将其设置为空
返回值	0代表方法调用成功，其他代表失败。

加入频道 (joinChannel)

```
public int joinChannel( String channelKey,
                        String channelName,
                        int optionalUid );
```

该方法让用户加入通话频道，在同一个频道内的用户可以互相通话，多个用户加入同一个频道，可以群聊。使用不同 App ID 的应用程序是不能互通的。如果已在通话中，用户必须调用 leaveChannel() 退出当前通话，才能进入下一个频道。

名称	描述
channelKey	频道key
channelName	频道名字
optionalUid	用户ID
返回值	0代表方法调用成功，其他代表失败。

加入频道 (joinChannel)

```
public int joinChannel( String channelKey,
                        String channelName,
                        int optionalUid,
                        boolean enableChat,
                        boolean enableSignal );
```

该方法让用户加入通话频道，在同一个频道内的用户可以互相通话，多个用户加入同一个频道，可以群聊。使用不同 App ID 的应用程序是不能互通的。如果已在通话中，用户必须调用 leaveChannel() 退出当前通话，才能进入下一个频道。

名称	描述
channelKey	频道key
channelName	频道名字
optionalUid	用户ID
enableChat	启用聊天
enableSignal	启用信令
返回值	0代表方法调用成功，其他代表失败。

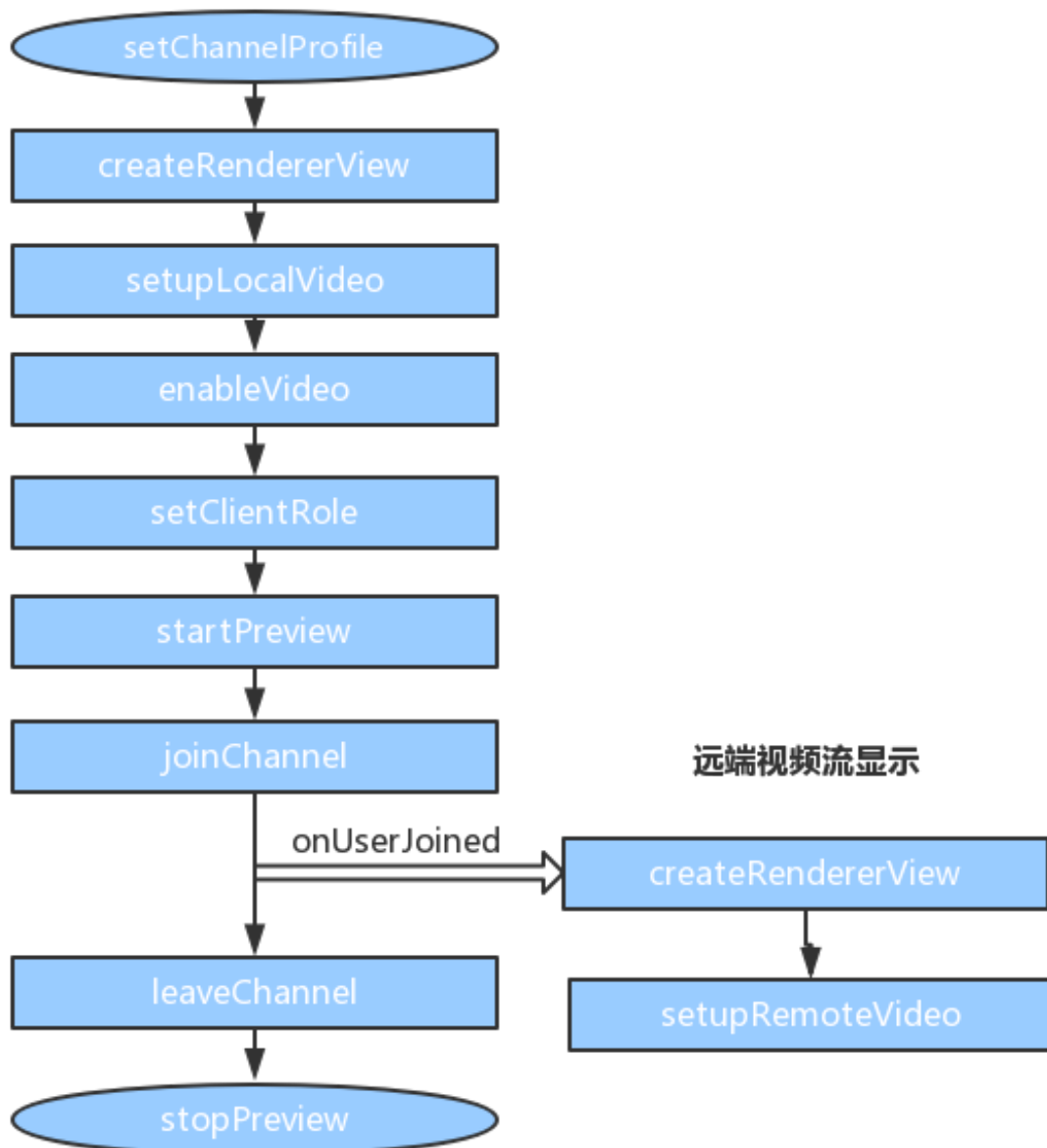
## 离开频道 (leaveChannel)

离开当前频道，当调用 joinChannel 方法后，必须调用 leaveChannel() 结束通话，否则无法开始下一次通话。该方法会把会话相关的所有资源释放掉。在退出频道成功后，SDK 会触发 onLeaveChannel 回调。

```
public int leaveChannel();
```

名称	描述
返回值	0代表方法调用成功，其他代表失败。

## 视频直播流程图



## 创建渲染视图 (createRendererView)

```
public SurfaceView CreateRendererView(Context context)
```

该方法创建视频渲染视图，返回 SurfaceView 的类型。显示视频视图必须调用该方法获取，而不是直接创建SurfaceView。

名称	描述
context	安卓上下文
返回值	SurfaceView 渲染视图

## 打开视频模式 (enableVideo)

该方法用于打开视频模式。可以在加入频道前或者通话中调用，在加入频道前调用，则自动开启视频模式，在通话中调用则由音频模式切换为视频模式。调用 disableVideo() 方法可关闭视频模式。

```
public int enableVideo()
```

名称	描述
返回值	0代表方法调用成功，其他代表失败

## 关闭视频模式 (disableVideo)

该方法用于关闭视频。可以在加入频道前或者通话中调用，在加入频道前调用，则自动开启纯音频模式，在通话中调用则由视频模式切换为纯音频频模式。调用 enableVideo() 方法可开启视频模式。

```
public int disableVideo()
```

名称	描述
返回值	0代表方法调用成功，其他代表失败

## 设置本地视频属性 (setVideoProfile)

该方法设置视频编码属性（Profile）。每个属性对应一套视频参数，如分辨率、帧率、码率等。当设备的摄像头不支持指定的分辨率时，SDK 会自动选择一个合适的摄像头分辨率，但是编码分辨率仍然用 setVideoProfile() 指定的。



```
public int setVideoProfile( int profile, boolean swapWidthAndHeight
)
```

名称	描述
profile	预设的视频质量
swapWidthAndHeight	是否将视频的宽和高互调
返回值	0代表方法调用成功，其他代表失败

视频编码属性表：

视频属性	枚举值	分辨率(宽x高)	帧率(fps)	码率(kbps)
VIDEO_PROFILE_120P	110	160*120	15	65*1000
VIDEO_PROFILE_180P	111	320*180	15	140*1000
VIDEO_PROFILE_240P	112	320*240	15	200*1000
VIDEO_PROFILE_360P	113	640*360	15	400*1000
VIDEO_PROFILE_480P	114	640*480	15	500*1000
VIDEO_PROFILE_720P	115	1280*720	15	1130*1000
VIDEO_PROFILE_1080P	116	1920*1080	15	2080*1000

设置本地视频显示属性 (setupLocalVideo)

```
public int setupLocalVideo(VideoCanvas local, int activityOrientation)
```

该方法设置本地视频显示信息。应用程序通过调用此接口绑定本地视频流的显示视窗 (view)，并设置视频显示模式。在应用程序开发中，通常在初始化后调用该方法进行本地视频设置，然后再加入频道。

Class VideoCanvas:

- mUserID：该视频属性类所属用户ID
- mShowMode：设置视频显示的模式
- mSurface：视频显示视窗

视频显示的模式：

- `RENDER_MODE_FIT`: 如果视频尺寸与显示视窗尺寸不一致，在保持长宽比的前提下，将视频进行缩放后填满视窗。
- `RENDER_MODE_HIDDEN`: 如果视频尺寸与显示视窗尺寸不一致，则视频流会按照显示视窗的比例进行周边裁剪或图像拉伸后填满视窗。
- `RENDER_MODE_ADAPTIVE`: 如果自己和对方都是竖屏，或者如果自己和对方都是横屏，使用`RENDER_MODE_HIDDEN`如果对方和自己一个竖屏一个横屏，则使用`RENDER_MODE_FIT`。

名称	描述
local	视频显示属性对象
activityOrientation	activity的显示方向
返回值	0代表方法调用成功，其他代表失败

## 设置远端视频显示属性 (setupRemoteVideo)

```
public int setupRemoteVideo( VideoCanvas remote )
```

该方法绑定远程用户和显示视图，即设定 uid 指定的用户用哪个视图显示。调用该接口时需要指定远程视频的 uid，一般可以在 APP 收到 onUserJoined 事件时设置。Class VideoCanvas:同上

名称	描述
remote	视频显示属性对象
返回值	0代表方法调用成功，其他代表失败

## 开启视频预览 (startPreview)

```
public int startPreview()
```

该方法用于启动本地视频预览。在开启预览前，必须先调用 `setupLocalVideo` 设置预览窗口及属性，且必须调用 `enableVideo()` 开启视频功能。

名称	描述
返回值	0代表方法调用成功，其他代表失败

## 停止视频预览 (stopPreview)

```
public int stopPreview()
```

该方法用于停止本地视频预览。

名称	描述
返回值	0代表方法调用成功，其他代表失败

### 暂停本地视频流 (muteLocalVideoStream)

```
public int muteLocalVideoStream( boolean muted )
```

该方法暂停发送本地视频流。

名称	描述
muted	True: 不发送本地视频流，False: 发送本地视频流
返回值	0代表方法调用成功，其他代表失败

### 暂停所有远端视频流 (muteAllRemoteVideoStreams)

```
public int muteAllRemoteVideoStreams( boolean muted)
```

该方法用于允许/禁止接收和播放所有人的视频流。

名称	描述
muted	True: 停止播放用户的视频流，False: 允许播放用户的视频流
返回值	0代表方法调用成功，其他代表失败

### 暂停指定用户视频流 (muteRemoteVideoStream)

```
public int muteRemoteVideoStream( int uid, boolean muted )
```

该方法用于允许/禁止播放指定的远端视频流。

名称	描述



uid	用户ID
uid	True: 停止播放指定用户的视频流, False: 允许播放指定用户的视频流
返回值	0代表方法调用成功, 其他代表失败

## 切换前置/后置摄像头 (switchCamera)

```
public int switchCamera()
```

该方法用于在前置/后置摄像头间切换。

名称	描述
返回值	0代表方法调用成功, 其他代表失败

## 修改默认的语音路由 (setDefaultAudioRouteToSpeakerPhone)

```
public int setDefaultAudioRouteToSpeakerPhone( boolean defaultToSpeaker )
```

调用该方法修改默认的语音路由。

名称	描述
defaultToSpeaker	true 默认路由改为外放(扬声器) false 默认路由改为听筒
返回值	0代表方法调用成功, 其他代表失败

## 打开外放 (setEnabledSpeakerphone)

```
public int setEnabledSpeakerphone( boolean enabled )
```

该方法打开外放(扬声器)。调用该方法后, SDK 将返回 onAudioRouteChange 回调提示状态已更改。

名称	描述
enabled	true 打开扬声器(若已插入耳机、蓝牙等设备, 也将切换到扬声器), false 关闭扬声器, 切为听筒

返回值	0代表方法调用成功，其他代表失败
-----	------------------

## 启用说话者音量提示 (enableAudioVolumeIndication)

```
public int enableAudioVolumeIndication(int interval, int smooth)
```

该方法设定外放(扬声器)音量。

名称	描述
interval	指定音量提示的时间间隔。小于0禁用音量提示功能，单位为毫秒
smooth	平滑系数。默认可以设置为3
返回值	0代表方法调用成功，其他代表失败

## 将自己静音 (muteLocalAudioStream)

```
public int muteLocalAudioStream( boolean muted )
```

静音/取消静音。该方法用于允许/禁止往网络发送本地音频流。

名称	描述
muted	True麦克风静音，False取消静音
返回值	0代表方法调用成功，其他代表失败

## 静音所有远端音频 (muteAllRemoteAudioStreams)

```
public int muteAllRemoteAudioStreams( boolean muted )
```

该方法用于允许/禁止接收和播放远端用户的音频流，即对所有远端用户进行静音与否。

名称	描述
muted	True麦克风静音，False取消静音
返回值	0代表方法调用成功，其他代表失败

## 静音指定用户音频 (muteRemoteAudioStream)

```
public int muteRemoteAudioStream(int uid, boolean muted )
```

静音指定远端用户/对指定远端用户取消静音。

名称	描述
uid	指定用户ID
muted	True麦克风静音， False取消静音
返回值	0代表方法调用成功， 其他代表失败

## 开始播放伴奏 (startAudioMixing)

```
public int startAudioMixing(String filePath,  
                             bool loopback,  
                             bool replace,  
                             int cycle)
```

指定本地音频文件来和麦克风采集的音频流进行混音和替换(用音频文件替换麦克风采集的音频流)， 可以通过参数选择是否让对方听到本地播放的音频和指定循环播放的次数。

名称	描述
filePath	指定需要混音的本地音频文件名和文件路径。支持以下音频格式：mp3, aac, m4a, 3gp, wav, flac
loopback	True只有本地可以听到混音或替换后的音频流， False本地和对方都可以听到混音或替换后的音频流
replace	True音频文件内容将会替换本地录音的音频流， False音频文件内容将会和麦克风采集的音频流进行混音
cycle	指定音频文件循环播放的次数。正整数代表循环的次数
返回值	0代表方法调用成功， 其他代表失败

## 停止播放伴奏 (stopAudioMixing)

```
public int stopAudioMixing()
```

---

该方法停止播放伴奏。请在频道内调用该方法。

名称	描述
返回值	0代表方法调用成功，其他代表失败

### 暂停播放伴奏 (pauseAudioMixing)

```
public int pauseAudioMixing()
```

该方法暂停播放伴奏。请在频道内调用该方法。

名称	描述
返回值	0代表方法调用成功，其他代表失败

### 恢复播放伴奏 (resumeAudioMixing)

```
public int resumeAudioMixing()
```

该方法恢复混音，继续播放伴奏。请在频道内调用该方法。

名称	描述
返回值	0代表方法调用成功，其他代表失败

### 调节伴奏音量 (adjustAudioMixingVolume)

```
public int adjustAudioMixingVolume(int volume)
```

该方法调节混音里伴奏的音量大小。请在频道内调用该方法。

名称	描述
volume	伴奏与人声音量比例范围为0~1。0代表无伴奏声音，1代表无人声。
返回值	0代表方法调用成功，其他代表失败

### 获取伴奏时长 (getAudioMixingDuration)

```
public int getAudioMixingDuration()
```

名称	描述
返回值	0代表方法调用成功，其他代表失败

## 获取伴奏播放进度 (getAudioMixingCurrentPosition)

```
public int getAudioMixingCurrentPosition()
```

该方法获取伴奏时长，单位为毫秒。请在频道内调用该方法。

名称	描述
返回值	伴奏时长

## 设置日志文件 (setLogFile)

```
public int setLogFile( String filePath )
```

设置 SDK 的输出 log 文件。SDK 运行时产生的所有 log 将写入该文件。应用程序必须保证指定的目录存在而且可写。

名称	描述
filePath	应用程序必须保证指定的目录存在而且可写
返回值	0代表方法调用成功，其他代表失败

## 设置日志过滤器 (setLogFilter)

```
public int setLogFilter(int filter)
```

设置 SDK 的输出日志过滤器。不同的过滤器可以用或组合。

名称	描述
filter	过滤器
返回值	0代表方法调用成功，其他代表失败

日志过滤等级：

- LOG\_FILTER\_INFO
- LOG\_FILTER\_DEBUG
- LOG\_FILTER\_WARNING
- LOG\_FILTER\_ERROR
- LOG\_FILTER\_OFF

## 销毁引擎实例 (destroy)

```
public void destroy()
```

释放资源。

名称	描述
返回值	0代表方法调用成功，其他代表失败

## 查询 SDK 版本号 (getVersion)

```
public String getVersion()
```

该方法返回 SDK 版本号字符串。

名称	描述
返回值	SDK版本号

# 即时通讯 (TTTRtcEngine)

所有返回值为 int 型的 API，如无特殊说明，返回值 0 为调用成功，返回值小于 0 为调用失败。

## 发送聊天消息(sendChatMessage)

```
public int sendChatMessage(long nDstUserID,  
                           int type,  
                           String sSeqID,  
                           String sData)
```

名称	描述
----	----

nDstUserID	消息发送目标的ID
type	消息类型:文字、图片、语音
sSeqID	唯一标识
sData	消息
返回值	0代表方法调用成功，其他代表失败。

## 发送聊天消息(sendChatMessage)

```
public int sendSignal(long nDstUserID,
                    String sSeqID,
                    String sData)
```

名称	描述
nDstUserID	消息发送目标的ID
sSeqID	唯一标识
sData	消息
返回值	0代表方法调用成功，其他代表失败。

## 开始采集聊天语音(startRecordChatAudio)

```
public int startRecordChatAudio()
```

名称	描述
返回值	0代表方法调用成功，其他代表失败。

## 停止采集并发送语音(stopRecordAndSendChatAudio)

```
public int stopRecordAndSendChatAudio(long nDstUserID, String sSeqID)
```

名称	描述
nDstUserID	语音发送的目标用户ID
sSeqID	消息唯一标识

返回值	0代表方法调用成功，其他代表失败。
-----	-------------------

## 取消语音的录制(cancelRecordChatAudio)

```
public int cancelRecordChatAudio()
```

名称	描述
返回值	0代表方法调用成功，其他代表失败。

## 播放聊天语音(playChatAudio)

```
public int playChatAudio(String audioPath)
```

名称	描述
audioPath	语音文件路径
返回值	0代表方法调用成功，其他代表失败。

## 停止播放聊天语音(stopChatAudio)

```
public int stopChatAudio()
```

名称	描述
返回值	0代表方法调用成功，其他代表失败。

## 是否正在播放(isChatAudioPlaying)

```
public boolean isChatAudioPlaying()
```

名称	描述
返回值	是否正在播放语音

# 回调事件 (TTTRtcEngineEventHandler)



TTTRtcEngineEventHandler 接口类用于SDK向应用程序发送回调事件通知，应用程序通过继承该接口类的方法获取 SDK 的事件通知。

## 加入频道回调 (onJoinChannelSuccess)

```
void onJoinChannelSuccess(String channel, long uid)
```

表示客户端已经成功进入指定房间。

名称	描述
channel	频道名称
uid	登录的用户ID

## 发生错误回调 (onError)

```
void onError(int err)
```

表示 SDK 运行时出现了（网络或媒体相关的）错误。

名称	描述
err	错误码

错误码：

名称	错误码	原因描述
ERROR_KICK_BY_HOST	201	被房主移出房间
ERROR_KICK_BY_PUSHRTMPFAILED	202	RTMP推流失败
ERROR_KICK_BY_SERVEROVERLOAD	203	服务器过载
ERROR_KICK_BY_MASTER_EXIT	204	房主已经离开房间
ERROR_KICK_BY_RELOGIN	205	重复登录
ERROR_KICK_BY_NOAUDIODATA	206	长时间没有上行音频数据
ERROR_KICK_BY_NOVIDEODATA	207	长时间没有上行

		视频数据
ERROR_KICK_BY_NEWCHAIRENTER	208	其他人以主播身份进入房间
ERROR_KICK_BY_CHANNELKEYEXPIRED	209	CHANNEL KEY 失效了

## 离开频道回调 (onLeaveChannel)

```
public void onLeaveChannel( RtcStats stats )
```

应用程序调用 leaveChannel() 方法时，SDK 提示应用程序离开频道成功。在该回调方法中，应用程序可以得到此次通话的总通话时长、SDK 收发数据的流量等信息。

名称	描述
stats	通话相关的统计信息

class RtcStats:

```
/**
 * 通话时长(秒)，累计值
 */
private int mTotalDuration;
/**
 * 发送字节数(bytes)，累计值
 */
private int mTxBytes;
/**
 * 接收字节数(bytes)，累计值
 */
private int mRxBytes;
/**
 * 发送码率(kbps)，瞬时值
 */
private int mTxKBitRate;
/**
 * 接受码率(kbps)，瞬时值
 */
private int mRxKBitRate;
/**
 * 客户端接入网络质量
 */
private int mLastmileQuality;
/**
```

```
    * 当前系统的 CPU 使用率(%)
    */
    private int mCpuTotalUsage;
    /**
    * 当前应用程序的 CPU 使用率(%)
    */
    private int mCpuAppUsage;
```

## 语音路由已变更回调 (onAudioRouteChanged)

```
public void onAudioRouteChanged(int routing)
```

当调用 `setEnabledSpeakerphone` 成功时，SDK 会通过该回调通知 App 语音路由状态已发生变化。该回调返回当前的语音路由已切换至听筒，外放(扬声器)，耳机或蓝牙。

名称	描述
routing	修改成功后的语音路由

routing 定义如下:

```
    /**
    * 默认路由耳机或蓝牙
    */
    public static final int AUDIO_ROUTE_HEADSET = 0;
    /**
    * 默认路由扬声器
    */
    public static final int AUDIO_ROUTE_SPEAKER = 1;
    /**
    * 默认路由手机听筒
    */
    public static final int AUDIO_ROUTE_HEADPHONE = 2;
```

## 说话声音音量提示回调 (onAudioVolumeIndication)

```
void onAudioVolumeIndication(long nUserID,
                             int audioLevel,
                             int audioLevelFullRange)
```

提示谁在说话及其音量。默认禁用。可以通过 `enableAudioVolumeIndication` 方法设置。

名称	描述
----	----

nUserID	说话者的用户ID
audioLevel	说话者的音量，在0-9之间
audioLevelFullRange	说话者的音量，音量范围更大0-32767.5之间

## 其他用户加入当前频道回调 (onUserJoined)

```
public void onUserJoined( int uid, int identity )
```

提示有主播加入了频道。如果该客户端加入频道时已经有人在频道中，SDK 也会向应用程序上报这些已在频道中的用户

名称	描述
uid	加入频道的用户ID
identity	加入频道的用户的身份，副播或观众

## 其他用户离开当前频道回调 (onUserOffline)

```
public void onUserOffline(int uid, int reason)
```

提示有主播离开了频道。

名称	描述
uid	离开频道的用户ID
reason	该用户离开频道的原因

## 统计数据回调 (onRtcStats)

```
public void onRtcStats( RtcStats stats )
```

该回调定期上报 Rtc Engine 的运行时的状态，每两秒触发一次。

名称	描述
stats	通话相关的统计信息

## 连接丢失回调 (onConnectionLost)

```
public void onConnectionLost()
```

该回调方法表示 SDK 和服务端失去了网络连接。

### 本地视频显示回调 (onFirstLocalVideoFrame)

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

提示第一帧本地视频画面已经显示在屏幕上。

名称	描述
width	视频流宽(像素)
height	视频流高(像素)

### 远端视频显示回调 (onFirstRemoteVideoFrame)

```
public void onFirstRemoteVideoFrame(long uid, int width, int height)
```

第一帧远程视频显示在视图上时，触发此调用

名称	描述
uid	加入频道的用户ID
width	视频流宽(像素)
height	视频流高(像素)

### 远端视频接收解码回调 (onFirstRemoteVideoDecoded)

```
public void onFirstRemoteVideoDecoded(long uid, int width, int height)
```

收到第一帧远程视频流并解码成功时，触发此调用。

名称	描述
uid	加入频道的用户ID
width	视频流宽(像素)

height	视频流高(像素)
--------	----------

## 其他用户启用/关闭视频 (onUserEnableVideo)

```
public void onUserEnableVideo(int uid, boolean enabled)
```

提示有其他用户启用/关闭了视频功能。关闭视频功能是指该用户只能进行语音通话，不能显示、发送自己的视频，也不能接收、显示别人的视频。

名称	描述
uid	用户ID
enabled	用户已启用/关闭了视频功能

## 本地视频统计回调 (onLocalVideoStats)

```
public void onLocalVideoStats(LocalVideoStats stats)
```

名称	描述
stats	本地视频相关的统计信息

报告更新本地视频统计信息，该回调函数每两秒触发一次。

## 远端视频统计回调 (onRemoteVideoStats)

```
public void onRemoteVideoStats( RemoteVideoStats stats)
```

报告更新远端视频统计信息，该回调函数每两秒触发一次。

名称	描述
stats	远端视频的统计信息

## 摄像头启用回调 (onCameraReady)

```
public void onCameraReady()
```

提示已成功打开摄像头，可以开始捕获视频。

## 视频功能停止回调 (onVideoStopped)

```
public void onVideoStopped()
```

提示视频功能已停止。

## Channel Key 过期回调 (onRequestChannelKey)

```
public void onRequestChannelKey()
```

在调用 joinChannel() 时如果指定了 Channel Key，由于 Channel Key 具有一定的时效，在通话过程中 SDK 可能由于网络原因和服务端失去连接，重连时可能需要新的 Channel Key。该回调通知 APP 需要生成新的 Channel Key，并需调用 renewChannelKey() 为 SDK 指定新的 Channel Key。

## 聊天消息发送回调 (OnChatMessageSent)

```
public void OnChatMessageSent(ChatInfo chatInfo, int error)
```

名称	描述
chatInfo	消息信息
error	错误信息

消息已经成功发送时，触发此回调。

## 信令发送回调 (OnSignalSent)

```
public void OnSignalSent(String sSeqID, int error)
```

名称	描述
sSeqID	信令ID
error	错误信息

信令已经成功发送时，触发此回调。

## 收到聊天消息回调 (OnChatMessageRecived)

```
public void OnChatMessageRecived(long nSrcUserID, ChatInfo chatInf
```

o)

收到其他用户发来的消息时触发此回调。

名称	描述
nSrcUserID	消息发送方userID
ChatInfo	消息信息

### 信令接收的回调 (OnSignalRecived)

```
public void OnSignalRecivedString filePath(long nSrcUserID,  
                                             int type,  
                                             String strData)
```

收到其他用户发来的消息时触发此回调。

名称	描述
nSrcUserID	消息发送方userID
type	消息类型
strData	消息内容

### 语音消息播放完成回调 (onPlayChatAudioCompletion)

```
public void onPlayChatAudioCompletion(String filePath)
```

接收到的语音消息播放完成触发此回调。

名称	描述
filePath	文件路径