Agora Rtc Engine SDK for Android API 参考文档

support@agora

1.1.0 版本　2015 年 11 月

目录

所需库

将 **Agora Rtc Engine SDK** 中 **libs** 文件夹下的以下库复制到项目的 **libs** 文件夹下：

- armeabi-v7a
- agora-rtc-sdk.jar

SDK 目前只提供 32 位库版本，但可以以 32 位兼容模式运行在 64 位 Android 设备上。

安装在 64 位设备上的 APK 应确保 arm64_v8a 下没有 so，否则 Android 系统将以 64 位进程模式启动程序。**Agora Rtc Engine SDK** 不支持这种模式。

所需权限

在 AndroidManisfest.xml 中，需要加入以下许可：

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.RECORD_AUDIO" />

<uses-permission android:name="android.permission.CAMERA" />

<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />

<uses-permission android:name="*android.permission.ACCESS_NETWORK_STATE*" />

Agora Rtc Engine SDK 接口方法

1. RtcEngine 接口类

套件: io.agora.rtc

`RtcEngine` 类是使用 **Agora Rtc Engine SDK** 的主要接口类，对 SDK 功能的使用都通过这个类进行。调用 `RtcEngine` 的接口最好在同一个线程进行，不建议在不同的线程同时调用。在之前的版本中，该类名为 `AgoraAudio`，从 1.0 版本更改为 `RtcEngine`。

创建 RtcEngine 对象 (create)

```
Public static RtcEngine create(Context context, String vendorKey,
IRtcEngineEventHandler handler)
```

创建 `RtcEngine` 对象。目前 **Agora Rtc Engine SDK** 只支持一个 `RtcEngine` 实例，应用程序应该只创建一个 `RtcEngine` 对象。`RtcEngine` 类的所有接口函数，如无特殊说明，都是异步调用，对接口的调用建议在同一个线程进行。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。

名称	描述
context	安卓活动 (Android Activity) 的上下文
vendorKey	Agora 为应用程序开发者签发的厂商 Key。如果没有，请向 Agora 申请。

handler	IRtcEngineEventHandler 是一个提供了缺省实现的抽象类，SDK 通过该抽象类向应用程序报告 SDK 运行时的各种事件。
Return Value	RtcEngine 对象。

开启视频模式 (enableVideo)

public int enableVideo()

该方法用于开启视频模式。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。

名称	描述
Return Value	0：方法调用成功 <0：方法调用失败

开启纯音频模式 (disableVideo)

public int disableVideo()

该方法用于关闭视频，开启纯音频模式。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。

名称	描述
Return Value	0：方法调用成功 <0：方法调用失败

开启视频预览(startPreview)

int startPreview()

该方法用于启动本地视频预览关闭视频。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
Return Value	0：方法调用成功 <0：方法调用失败

停止视频预览(stopPreview)

int stopPreview()

该方法用于停止本地视频预览关闭视频。

名称	描述
Return Value	0：方法调用成功 <0：方法调用失败

加入频道 (joinChannel)

public int joinChannel(String key, String channelName, String optionalInfo, 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 vendor key 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
key	此为程序动态生成的 Token； 当用户使用静态 Key 也即只使用 vendor key 时, 该参数是可选的，传 NULL 即可； 当用户使用动态 Key 时，Agora 为应用程序开发者额外签发一个签名秘钥 sign key，开发者通过 Agora 提供的算法和秘钥生成此用户 Token，用于服务器端用户验证。 一般来说使用静态 Key 即可，对于安全有极高要求的使用者请联系 Agora 客服或销售人员获得动态 Key 使用支持。
channelName	频道名称。可以是任何描述性的名称，如“游戏 1”或“通话 2”。Channel 的最大长度为 128 个字符。
optionalInfo	可选。一般可设置为空字符串。
optionalUid	可选。用户 ID：32 位无符号整数。如果不指定（即设成 0），SDK 会自动分配一个，并在 onJoinChannelSuccess 方法中返回；如果指定，必须保证不同用户分配到的 uid 是唯一的。 **注：**UID 在 SDK 内部用 32 位无符号整数表示，由于 Java 不支持无符号整数，uid 被当成 32 位有符号整数处理，对于过大的整数，Java 会表示为负数，如有需要可以用(uid&0xffffffffL)转换成 64 位整数。
Return Value	0：方法调用成功 <0：方法调用失败 ERR_INVALID_ARGUMENT (-2)：传递的参数无效 ERR_NOT_READY (-3)：没有成功初始化 ERR_REFUSED (-5)：SDK 不能发起通话，可能是因为处于另一个通话中，或者创建频道失败。

离开频道(leaveChannel)

public int leaveChannel();

离开频道，即挂断或退出通话。joinChannel 后，必须调用 leaveChannel 以结束通话，否则不能进行下一次通话。不管当前是否在通话中，都可以调用 leaveChannel，没有副作用。leaveChannel 会把会话相关的所有资源释放掉。

名称	描述
Return Value	0：方法调用成功 <0：方法调用失败

更新频道动态 key (renewChannelDynamicKey)

int renewChannelDynamicKey(const char* key)

该方法用于更新 token。如果启用了 token 机制，过一段时间后 token 会失效。当 onError 回调报告 ERR_DYNAMIC_KEY_TIMEOUT(109)时，APP 应重新获取 token，然后调用该 API 更新 token，否则 SDK 无法和服务器建立连接。

名称	描述
key	指定要更新的 token。
Return Value	0：方法调用成功 <0：方法调用失败

设置频道通话 Profile (setChannelProfile)

int setChannelProfile (int profile)

该方法用于设置频道 Profile。为了更好的优化，Agora RtcEngine 需要知道 APP 的使用场景（比如群聊模式还是主播模式），从而使用不同的优化手段。目前支持四种 Profile：自由、主播、听众、WEB SDK 兼容。

自由模式用于常见的一对一或者群聊，频道中的任何用户都可以自由说话，这是默认 Profile。

主播和听众 Profile 需要配合使用，主要用于主播模式。主播 Profile 的用户可以发送和接收音视频；听众 Profile 只能接收音视频，不能发送。

WEB SDK 兼容模式用于需要和 Agora Web SDK 互通的场景。默认的 FREE 模式是不能互通的。

注意：在同一个频道中，自由 Profile 不能和主播、听众 Profile 混用。

注意：该方法必选在进入频道前设置，在频道中设置无效。

名称	描述
Profile	指定频道 Profile，目前支持三种： • CHANNEL_PROFILE_FREE（默认） • CHANNEL_PROFILE_BROADCASTER • CHANNEL_PROFILE_AUDIENCE
Return Value	0：方法调用成功 <0：方法调用失败

启动语音通话测试(startEchoTest)

public int startEchoTest();

该方法启动回声测试，目的是测试系统的音频设备（耳麦、扬声器等）和网络连接是否正常。在测试过程中，用户先说一段话，在 10 秒后，声音会回放出来。如果 10 秒后用户能正常听到自己刚才说的话，就表示系统音频设备和网络连接都是正常的。

注：调用 startEchoTest 后必须调用 stopEchoTest 以结束测试，否则不能进行下一次回声测试，或者调用 joinChannel 进行通话。

名称	描述
Return Value	0：方法调用成功 <0：方法调用失败 ERR_REFUSED (-5)：不能启动测试，可能没有成功初始化。

终止回声测试 (stopEchoTest)

public int stopEchoTest();

该方法停止回声测试。

名称	描述
Return Value	0：方法调用成功 <0：方法调用失败 ERR_REFUSED(-5)：停止 EchoTest 失败，可能是因为不在进行 EchoTest。

启用网络测试 (enableNetworkTest)

public int enableNetworkTest();

该方法启用网络连接质量测试，用于检测用户网络接入质量。启用后，SDK 通过 onNetworkQuality 回调方法（不定期）通知应用程序当前的网络质量。

注：启用后，在没有通话时也会消耗一些网络流量。建议的使用方法一般如下：

应用程序在前台时，调用 enableNetworkTest 启用网络接入检测；应用程序被切换到后台后，调用 disableNetworkTest 禁用检测以节省流量。网络测试是默认关闭的。

名称	
Return Value	0：方法调用成功 <0：方法调用失败

禁用网络测试 (disableNetworkTest)

public int disableNetworkTest();

该方法禁用网络测试。

名称	描述
Return Value	0：方法调用成功 <0：方法调用失败

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。

注：该方法不影响录音状态，并没有禁用麦克风。

名称	描述
On	True：麦克风静音 False：取消静音
Return Value	0：方法调用成功

	<0：方法调用失败

静音所有远端音频 (muteAllRemoteAudioStreams)

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

静音所有远端用户/对所有远端用户取消静音。本方法用于允许/禁止播放远端用户的音频流。

注：该方法不影响音频数据流的接收，只是不播放音频流。

名称	描述
On	True：麦克风静音 False：取消静音
Return Value	0：方法调用成功 <0：方法调用失败

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。本方法用于允许/禁止播放远端用户的音频流。

注：该方法不影响音频数据流的接收，只是不播放音频流。

名称	描述
uid	指定用户
muted	True：麦克风静音 False：取消静音
Return Value	0：方法调用成功 <0：方法调用失败

扬声器通话 (setEnableSpeakerphone)

```
public int setEnableSpeakerphone(boolean enabled);
```

切换音频输出方式：麦克风或听筒。

名称	描述
enabled	True：音频输出至扬声器 False：音频输出至听筒
Return Value	0：方法调用成功 <0：方法调用失败

开始客户端录音(startAudioRecording)

```
public int startAudioRecording(String filePath);
```

开始录音。SDK 支持在通话中进行录音，录音文件的格式为 wav。应用程序必须保证指定的目录存在而且可写。该接口需要在 `joinChannel` 之后调用；在 `leaveChannel` 时如果还在录音，会自动停止。

名称	描述
filePath	存储录音文件的文件路径。

Return Value	0：方法调用成功 <0：方法调用失败

停止客户端录音(stopAudioRecording)

public int stopAudioRecording();

停止录音。该接口需要在 leaveChannel 之前调用，如果没有调用，在 leaveChannel 时会自动停止。

名称	描述
Return Value	0：方法调用成功 <0：方法调用失败

获取通话 ID (getCallId)

public String getCallId();

获取当前的通话 ID，客户端在每次 joinChannel 后会生成一个对应的 CallId，标识该客户端的此次通话。有些方法如 rate, complain 需要在通话结束后调用，向 SDK 提交反馈，这些方法必须指定 CallId 参数。使用这些反馈方法，需要在通话过程中调用 getCallId 方法获取 CallId，在通话结束后在反馈方法中作为参数传入。

名称	描述
Return Value	通话 ID

给通话评分(rate)

public int rate(String callId, int rating, String description);

该方法能够让用户为通话评分，一般在通话结束后调用。

名称	描述
callId	通过 getCallId 函数获取的通话 ID
rating	给通话的评分，最低 1 分，最高 10 分
description	给通话的描述，可选，长度应小于 800 字节
Return Value	0：方法调用成功 <0：方法调用失败 ERR_INVALID_ARGUMENT (-2)：传入的参数无效，比如 callId 无效。 ERR_NOT_READY (-3)：SDK 不在正确的状态，可能是因为没有成功初始化。

投诉通话质量 (complain)

public int complain(String callId, String description);

该方法让用户就通话质量进行投诉。一般在通话结束后调用。

名称	描述
callId	通过 getCallId 函数获取的通话 ID。
description	给通话的描述，可选，长度应小于 800 字节。

Return Value	0：方法调用成功 <0：方法调用失败 ERR_INVALID_ARGUMENT (-2)：传入的参数无效，比如 callId 无效。 ERR_NOT_READY (-3)：SDK 不在正确的状态，可能是因为没有成功初始化

监测耳机插拔事件 (monitorHeadsetEvent)

```
public void monitorHeadsetEvent(boolean monitor);
```

监听耳机插拔事件，在加入通话前调用。默认监听。

名称	描述
monitor	True：监听耳机插拔事件（默认） False：不监听耳机插拔事件

监测蓝牙耳机事件 (monitorBluetoothHeadsetEvent)

```
public void monitorBluetoothHeadsetEvent(boolean monitor);
```

监听蓝牙耳机事件，在加入通话前调用。默认监听。

名称	描述
monitor	True：监听蓝牙耳机事件（默认） False：不监听蓝牙耳机事件

监测网络连接状态 (monitorConnectionEvent)

```
public void monitorConnectionEvent(boolean monitor);
```

监听网络连接状态事件，在加入通话前调用。默认监听。

名称	描述
monitor	True：监测网络连接状态（默认） False：不监测网络连接状态

是否是扬声器状态 (isSpeakerphoneEnabled)

```
public boolean isSpeakerphoneEnabled();
```

名称	描述
Return Value	True：表明输出到扬声器 False：表明输出到非扬声器（如听筒、耳机等）

设定扬声器音量 (setSpeakerphoneVolume)

```
public int setSpeakerphoneVolume(int volume);
```

该方法设定扬声器音量。

名称	描述
Volume	设定音量，最小为 0，最大为 255

Return Value	0：方法调用成功 <0：方法调用失败

启用说话者音量提示 (enableAudioVolumeIndication)

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

该方法允许 SDK 定期向应用程序反馈当前谁在说话以及说话者的音量。

名称	描述
interval	指定音量提示的时间间隔。 <=0：禁用音量提示功能 >0 ：提示间隔，单位为毫秒。建议设置到大于 200 毫秒。
smooth	平滑系数。默认可以设置为 3。
Return Value	0：方法调用成功 <0：方法调用失败

设置本地视频 Profile (setVideoProfile)

```
public int setVideoProfile(int profile);
```

该方法设置视频 Profile。每个 Profile 对应一套视频参数，如分辨率、帧率、码率等。

注意：应该在调用 joinChannel 进入频道前设置视频 Profile。

名称	描述
profile	视频 Profile。细节见下面的定义。Profile 取值在 io.agora.rtc. IRtcEngineEventHandler.VideoProfile 中定义。
Return Value	0：方法调用成功 <0：方法调用失败

视频 Profile 定义

视频 Profile	枚举值	分辨率(宽 x 高)	帧率(fps)	码率(kbps)
120P	0	160x120	15	80
120P_2	1	120x160	15	80
120P_3	2	120x120	15	60
180P	10	320x180	15	160
180P_2	11	180x320	15	160
180P_3	12	180x180	15	120
240P	20	320x240	15	200
240P_2	21	240x320	15	200

240P_3	22	240x240	15	160
360P	30	640x360	15	400
360P_2	31	360x640	15	400
360P_3	32	360x360	15	300
360P_4	33	640x360	30	800
360P_5	34	360x640	30	800
360P_6	35	360x360	30	600
480P	40	640x480	15	500
480P_2	41	480x640	15	500
480P_3	42	480x480	15	400
480P_4	43	640x480	30	1000
480P_5	44	480x640	30	1000
480P_6	45	480x480	30	800
720P	50	1280x720	15	1000
720P_2	51	720x1080	15	1000
720P_3	52	1280x720	30	2000
720P_4	53	720x1280	30	2000
1080P	60	1920x1080	15	1500
1080P_2	61	1080x1920	15	1500
1080P_3	62	1920x1080	30	3000
1080P_4	63	1080x1920	30	3000
1080P_5	64	1920x1080	60	6000
1080P_6	65	1080x1920	60	6000
4K	70	3840x2160	30	8000
4K_2	71	2160x3840	30	8000
4K_3	72	3840x2160	60	16000
4K_4	73	2160x3840	60	16000

设置本地视频显示属性 (setupLocalVideo)

public int setupLocalVideo(VideoCanvas local);

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。

名称	描述
local	设置视频属性。Class VideoCanvas: • view: 视频显示视窗。 • renderMode: 视频显示模式。 RENDER_MODE_HIDDEN (1): 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_FIT(2): 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_ADAPTIVE(3)：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。 • uid: 本地用户 ID，设置为 0。
Return Value	0：方法调用成功 <0：方法调用失败

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，一般可以在进频道前提前设置好，如果应用程序不能事先知道对方的 uid，可以在 APP 收到 onUserJoined 事件时设置。解除某个用户的绑定视图可以把 view 设置为空。

名称	描述
remote	设置视频属性。Class VideoCanvas: • view: 视频显示视窗。 • renderMode: 视频显示模式。 RENDER_MODE_HIDDEN (1): 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_FIT(2): 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_ADAPTIVE(3)：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。

	• uid: 用户 ID，指定远端视频来自哪个用户。
Return Value	0：方法调用成功 <0：方法调用失败

设置本地视频显示模式 (setLocalRenderMode)

```
public int setLocalRenderMode(int mode);
```

该方法设置本地视频显示模式。应用程序可以多次调用此方法更改显示模式。

名称	描述
mode	设置视频显示模式。 RENDER_MODE_HIDDEN (1): 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_FIT(2): 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_ADAPTIVE(3)：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。
Return Value	0：方法调用成功 <0：方法调用失败

设置远端视频显示模式 (setRemoteRenderMode)

```
public int setRemoteRenderMode(int uid, int mode);
```

该方法设置远端视频显示模式。应用程序可以多次调用此方法更改显示模式。

名称	描述
uid	用户 ID
mode	设置视频显示模式。 RENDER_MODE_HIDDEN (1): 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_FIT(2): 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_ADAPTIVE(3)：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。
Return Value	0：方法调用成功 <0：方法调用失败

切换视频流的显示视窗 (switchView)

```
public void switchView(int uid1, int uid2);
```

该方法用于切换不同用户视频流的显示窗口。

名称	描述
uid1	用户 ID，指定需要切换的远程视频来自哪个用户。
uid2	用户 ID，指定需要切换的另一个视频流。

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
Return Value	0：方法调用成功 <0：方法调用失败

暂停发送本地视频流 (muteLocalVideoStream)

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

暂停/恢复发送本地视频流。该方法用于允许/禁止往网络发送本地视频流。

注：该方法不影响本地视频流获取，没有禁用摄像头。

名称	描述
mute	True: 不发送本地视频流 False: 发送本地视频流
Return Value	0：方法调用成功 <0：方法调用失败

暂停所有远端视频流 (muteAllRemoteVideoStreams)

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

暂停/恢复所有人视频流。本方法用于允许/禁止播放所有人的视频流。

注：该方法不影响视频数据流的接收，只是不播放视频流。

名称	描述
mute	True: 停止播放接收到的所有视频流 False: 允许播放接收到的所有视频流
Return Value	0：方法调用成功 <0：方法调用失败

暂停指定远端视频流 (muteRemoteVideoStream)

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

暂停/恢复指定远端视频流。本方法用于允许/禁止播放指定远端视频流。

注：该方法不影响视频数据流的接收，只是不播放视频流。

名称	描述
uid	指定用户
mute	True: 停止播放接收到的视频流 False: 允许播放接收到的视频流
Return Value	0：方法调用成功 <0：方法调用失败

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	log 文件的全路径名
Return Value	0：方法调用成功 <0：方法调用失败

设置日志过滤器(setLogFilter)

```
int setLogFilter(unsigned 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： • 1：INFO • 2：WARNING • 4：ERROR • 8：FATAL • 0x800：DEBUG
Return Value	0：方法调用成功 <0：方法调用失败

报告通话质量 (makeQualityReportUrl)

```
public abstract String makeQualityReportUrl(String channel, int
listenerUid, int speakerUid, int format);
```

该方法生成报告通话质量的 url。通过返回的 url，应用程序可以获取详细的通话质量报告数据。使用时需要知道频道名和双方用户 ID。比如一个频道中有 A 和 B，可以分别获取 A 说话、B 收听的报告，以及 B 说话、A 收听的报告。

名称	描述
Channel	在 joinChannel 方法中确定的频道号码。
listenerUid	用户 ID，可能是使用 joinChannel 函数确定的。如果该项没有设定，SDK 会分配一个，可以使用

	onJoinChannelSuccess 回调将其取回（见下文）。
speakerUid	发言者 ID，用户可以使用 onUserJoined 回调将其取回（见下文）。如果没有设定该项，则会接收聊天室中所有发言者的报告。
Format	QualityReportJson(0)：json 格式 QualityReportHtml(1)：html 格式
Return Value	报告 url。应用程序可以用 HTTP 请求从此 url 获取通话质量报告。

2. IRtcEngineEventHandler 接口类

套件: io.agora.rtc

IRtcEngineEventHandler 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。接口类的所有方法都有缺省（空）实现，应用程序可以根据需要只继承关心的事件。在回调方法中，应用程序不应该做耗时或者调用可能会引起阻塞的 API（如 SendMessage），否则可能影响 SDK 的运行。

加入频道回调 (onJoinChannelSuccess)

```
public void onJoinChannelSuccess(String channel, int uid, int elapsed);
```

表示客户端已经登入服务器，且分配了频道 ID 和用户 ID。频道 ID 的分配是根据 join() API 中指定的频道名称。如果调用 join() 时并未指定用户 ID，服务器就会分配一个。

名称	描述
channel	频道名
uid	用户 ID。如果 joinChannel 中指定了 uid，则此处返回该 ID；否则使用 Agora 服务器自动分配的 ID。
elapsed	从 joinChannel 开始到该事件产生的延迟（毫秒）

重新加入频道回调 (onRejoinChannelSuccess)

```
public void onRejoinChannelSuccess(String channel, int uid, int
elapsed);
```

有时候由于网络原因，客户端可能会和服务器失去连接，SDK 会进行自动重连，自动重连成功后触发此回调方法。

Name	描述
channel	频道名 ID
uid	用户 ID
elapsed	从 joinChannel 开始到该事件产生的延迟（毫秒）

发生警告回调(onWarning)

```
virtual void onWarning(int warn, const char* msg)
```

该回调方法表示 SDK 运行时出现了（网络或媒体相关的）警告。通常情况下，SDK 上报的警告信息 APP 可以忽略，SDK 会自动恢复。比如和服务器失去连接时，SDK 可能会上报 ERR_OPEN_CHANNEL_TIMEOUT 警告，同时自动尝试重连。

名称	描述
warn	警告代码
msg	警告描述

发生错误回调 (OnError)

```
public void onError(int err);
```

表示 SDK 运行时出现了（ 网络或媒体相关的）错误。通常情况下，SDK 上报的错误意味着 SDK 无法自动恢复，需要 APP 干预或提示用户。比如启动通话失败时，SDK 会上报 ERR_START_CALL 错误。APP 可以提示用户启动通话失败，并调用 leaveChannel 退出频道。

Name	描述
err	错误代码： ERR_INVALID_VENDOR_KEY(101):无效的厂商 Key。 ERR_INVALID_CHANNEL_NAME(102):无效的频道名。 ERR_LOOKUP_CHANNEL_REJECTED(105):查找频道失败，意味着服务器主动拒绝了请求。 ERR_OPEN_CHANNEL_REJECTED(107):加入频道失败，意味着媒体服务器主动拒绝了请求。 ERR_LOAD_MEDIA_ENGINE(1001):加载媒体引擎失败。 ERR_START_CALL（1002）:打开本地音视频设备、启动通话失败。 ERR_START_CAMERA(1003):打开本地摄像头失败。

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（会话数据）	通话相关的统计信息。 struct RtcStat { int totalDuration; int txBytes; int rxBytes; int txKBitRate; int rxKBitRate; int lastmileQuality; int cpuTotalUsage; int cpuAppUsage;

	}; • totalDuration: 通话时长（秒），累计值 • txBytes: 发送字节数（bytes），累计值 • rxBytes：接收字节数（bytes），累计值 • txKBitRate：发送码率（kbps），瞬时值 • rxKBitRate：接收码率（kbps），瞬时值 • lastmileQuality：客户端接入网络质量 • cpuTotalQuality：当前系统的 CPU 使用率（%） • cpuAppQuality：当前应用程序的 CPU 使用率（%）

声音质量回调 (onAudioQuality)

public void onAudioQuality(int uid, int quality, short delay, short lost);

在通话中，该回调方法每两秒触发一次，报告当前通话的（嘴到耳）音频质量。默认启用。

名称	描述
uid	说话方的用户 ID。
quality	语音质量评分： • QUALITY_EXCELLENT(= 1) • QUALITY_GOOD(= 2) • QUALITY_POOR(= 3) • QUALITY_BAD(= 4) • QUALITY_VBAD(= 5) • QUALITY_DOWN(= 6)
delay	延迟（毫秒）
lost	丢包率（百分比）

说话声音音量提示回调 (onAudioVolumeIndication)

public void onAudioVolumeIndication(AudioVolumeInfo[] speakers, int totalVolume)

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
speakers	说话者（数组）。每个 speaker()： • uid: 说话者的用户 ID • volume：说话者的音量（0~255）
totalVolume	（混音后的）总音量（0~255）

其他用户加入当前频道回调 (onUserJoined)

public void onUserJoined(int uid, int elapsed);

提示有用户加入了频道。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。

注意：该回调只在 Channel Profile 为自由 Profile 时有效。

名称	描述
uid	用户 ID
elapsed	joinChannel 开始到该回调触发的延迟（毫秒）

其他用户离开当前频道回调 (onUserOffline)

```
public void onUserOffline(int uid);
```

提示用户离线（主动离开或掉线）。
注意：SDK 判断用户离开频道（或掉线）的依据是超时：在一定时间内（15 秒）没有收到对方的任何数据包，判定为对方掉线。在网络较差的情况下，可能会有误报。建议可靠的掉线检测应该由信令来做。
注意：该回调只在 Channel Profile 为自由 Profile 时有效。

名称	描述
uid	用户 ID
reason	离线原因： • Constants.USER_OFFLINE_QUIT：用户主动离开 • Constants.USER_OFFLINE_DROPPED：因过长时间收不到对方数据包，超时掉线。注意：由于 SDK 使用的是不可靠通道，也有可能对方主动离开本方没收到对方离开消息而误判为超时掉线。

用户静音回调 (onUserMuteAudio)

```
public void onUserMuteAudio(int uid, boolean muted);
```

提示有其他用户将他的音频流静音/取消静音。
注意：该回调只在 Channel Profile 为自由 Profile 时有效。

名称	描述
uid	用户 ID
muted	True - 该用户将音频静音 False - 该用户取消了音频静音

Rtc Engine 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	见上文的 RtcStats 类定义。

网络质量报告回调 (onNetworkQuality)

```
public void onNetworkQuality(int quality);
```

报告网络质量，该回调函数每 2 秒触发一次。

名称	描述
quality	• QUALITY_EXCELLENT(1) • QUALITY_GOOD(2) • QUALITY_POOR(3) • QUALITY_BAD(4) • QUALITY_VBAD(5) • QUALITY_DOWN(6)

连接丢失回调 (onConnectionLost)

public void onConnectionLost()

该回调方法表示 SDK 和服务器失去了网络连接。

本地视频显示回调 (onFirstLocalVideoFrame)

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

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)
elapsed	加入频道开始到该回调触发的延迟（毫秒）

远端视频显示回调 (onFirstRemoteVideoFrame)

public void onFirstRemoteVideoFrame(int uid, int width, int height, int elapsed)

第一帧远程视频显示在视图上时，触发此调用。应用程序可在此调用中获知出图时间（elapsed）。

名称	描述
uid	用户 ID，指定是哪个用户的视频流
width	视频流宽(像素)
height	视频流高(像素)
elapsed	加入频道开始到该回调触发的延迟（毫秒）

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

public void onFirstRemoteVideoDecoded(int uid, int width, int height, int elapsed)

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	用户 ID，指定是哪个用户的视频流
width	视频流宽(像素)
height	视频流高(像素)
elapsed	加入频道开始到该回调触发的延迟（毫秒）

用户停止/重启视频回调 (onUserMuteVideo)

public void onUserMuteVideo(int uid, boolean muted)

提示有其他用户暂停/恢复了视频流的发送。

注意：该回调只在 Channel Profile 为自由 Profile 时有效。

名称	描述
uid	用户 ID
muted	True - 该用户暂停了视频发送 False - 该用户恢复了视频发送

本地视频统计回调 (onLocalVideoStat)

public void onLocalVideoStat(int sentBitrate, int sentFrameRate)

报告更新本地视频统计信息，该回调函数每两秒触发一次。

名称	描述
sentBitrate	（上次统计后）发送的码率(kbps)
sentFrameRate	（上次统计后）发送的帧率(fps)

远端视频统计回调 (onRemoteVideoStat)

public void onRemoteVideoStat(int uid, int delay, int receivedBitrate, int receivedFrameRate)

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
uid	用户 ID，指定是哪个用户的视频流
delay	延时(毫秒)
receivedBitrate	接收码率(kbps)
receivedFrameRate	接收帧率(fps)

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。如果摄像头打开失败，可在 `onError()` 中处理相应错误。

视频功能停止回调(onVideoStopped)

virtual void onVideoStopped()

提示视频功能已停止。APP 如需在停止视频后对 view 做其他处理（比如显示其他画面），可以在这个回调中进行。

Agora为Agora Lab的注册商标。本文档中提及的其他产品或公司名称均为其各自公司的注册商标。

© 2015 Agora Lab. 版权所有。