



Agora Rtc Engine SDK for iOS

API 参考文档

support@agora

目录

所需库.....	5
Agora Rtc Engine SDK 接口方法.....	5
1. AgoraRtcEngineKit 接口类.....	5
初始化 (initWithVendorKey).....	5
开启视频模式 (enableVideo).....	5
开启纯音频模式 (disableVideo)	6
开启视频预览(startPreview).....	6
停止视频预览(stopPreview)	6
加入频道 (joinChannelByKey)	6
离开频道 (leaveChannel).....	7
更新频道动态 key (renewChannelDynamicKey).....	7
设置频道通话 Profile(setChannelProfile)	8
获取通话 ID (getCallId)	8
给通话评分 (rate)	9
投诉通话质量 (complain)	9
启动语音通话测试 (startEchoTest)	9
终止应答测试 (stopEchoTest).....	10
启用网络测试 (enableNetworkTest).....	10
禁用网络测试 (disableNetworkTest)	10
报告通话质量 URL (makeQualityReportUrl)	10
静音 (muteLocalAudioStream).....	11
静音所有远端音频流 (muteAllRemoteAudioStreams).....	11
静音指定远端音频流 (muteRemoteAudioStream)	11
扬声器通话 (setEnabledSpeakerphone).....	12
是否是扬声器状态 (isSpeakerphoneEnabled)	12
设定扬声器音量 (setSpeakerphoneVolume).....	12
启用说话者音量提示 (enableAudioVolumeIndication)	12
设置视频 Profile (setVideoProfile)	13
设置本地视频视图 (setupLocalVideo).....	14
设置远端视频显示视图 (setupRemoteVideo).....	15
设置本地视频显示模式 (setLocalRenderMode)	15
设置远端视频显示模式 (setRemoteRenderMode)	16
切换视频流的显示视窗 (switchView)	16
切换前置/后置摄像头 (switchCamera)	17

暂停发送本地视频流 (muteLocalVideoStream)	17
暂停所有远端视频流 (muteAllRemoteVideoStreams).....	17
暂停指定远端视频流 (muteRemoteVideoStream)	17
开始客户端录音 (startAudioRecording)	18
停止客户端录音 (stopAudioRecording).....	18
设置日志文件 (setLogFile)	18
设置日志文件 (setLogFilter).....	18
2. 代理方法 (AgoraRtcEngineDelegate)	19
发生警告回调 (didOccurWarning).....	19
发生错误回调 (didOccurError)	19
音量提示回调 (reportAudioVolumeIndicationOfSpeakers)	20
本地首帧视频显示回调 (firstLocalVideoFrameWithSize)	20
远端首帧视频接收解码回调 (firstRemoteVideoDecodedOfUid).....	20
远端首帧视频显示回调 (firstRemoteVideoFrameOfUid)	20
用户加入回调 (didJoinedOfUid)	21
用户离线回调 (didOfflineOfUid)	21
用户音频静音回调 (didAudioMuted)	21
用户停止/重新发送视频回调 (didVideoMuted)	22
本地视频统计回调 (localVideoStatWithSentBitrate)	22
远端视频统计回调 (remoteVideoStatOfUid).....	22
摄像头启用回调 (rtcEngineCameraDidReady)	22
视频功能停止回调(rtcEngineVideoDidStop)	23
网络连接丢失回调 (rtcEngineConnectionDidLost).....	23
加入频道成功回调 (didJoinChannel).....	24
重新加入频道回调 (didRejoinChannel)	24
Rtc Engine 统计数据回调 (reportRtcStats)	24
语音质量回调 (audioQualityOfUid)	25
网络质量回调 (networkQuality).....	25
3. 回调方法	25
用户加入回调 (userJoinedBlock)	26
用户离线回调 (userOfflineBlock)	26
重新加入频道回调 (rejoinChannelSuccessBlock).....	26
离开频道回调 (leaveChannelBlock).....	26
Rtc Engine 统计数据回调 (rtcStatsBlock)	27
音量提示回调 (audioVolumeIndicationBlock)	27

本地首帧视频显示回调 (firstLocalVideoFrameBlock)	27
远端首帧视频显示回调 (firstRemoteVideoFrameBlock)	27
远端首帧视频接收解码回调 (firstRemoteVideoDecodedBlock)	27
用户音频静音回调 (userMuteAudioBlock)	28
用户停止/重新发送视频回调 (userMuteVideoBlock)	28
本地视频统计回调 (localVideoStatBlock)	28
远端视频统计回调 (remoteVideoStatBlock)	28
摄像头启用回调 (cameraReadyBlock)	29
语音质量回调 (audioQualityBlock)	29
网络质量回调 (networkQualityBlock)	29
网络连接丢失回调 (connectionLostBlock)	30

所需库

Agora Rtc Engine SDK 需要 iOS 6.0 SDK 或更新版本的 iOS SDK。主项目应当与 SDK 中的下列库有连接：

- AgoraMediaKit.framework
- AudioToolbox.framework
- AVFoundation.framework
- CoreMedia.framework
- CoreTelephony.framework
- CoreMotion.framework
- SystemConfiguration.framework
- libc++.dylib

注：Agora Rtc Engine SDK 默认使用 libc++ (LLVM)，如需使用 libstdc++ (GNU)，请联系 support@agora.io。

在源文件中，使用下列命令导入 AgoraRtcEngineKit：

- `#import <AgoraRtcEngineKit/AgoraRtcEngineKit.h>`

SDK 提供的库是 Fat Image，包含 32/64 位模拟器、32/64 位真机版本。

Agora Rtc Engine SDK 接口方法

1. AgoraRtcEngineKit 接口类

agora::rtc::IRtcEngine 是 Agora Rtc Engine SDK 的基本接口，通过生成 agora::rtc::IRtcEngine 对象，调用该方法使用 Agora Rtc Engine SDK 的通话能力。在之前的版本中，该类名为 IAgoraAudio，从 1.0 版本更改为 agora::rtc::IRtcEngine。

初始化 (initWithVendorKey)

```
(id) initWithVendorKey: (NSString*) vendorKey
delegate: ( id<AgoraRtcEngineDelegate>) delegate;
```

以上代码初始化 AgoraRtcEngineKit 类。使用 AgoraRtcEngineKit，必须先调用该接口进行初始化。

SDK 通过指定的 delegate 通知应用程序引擎运行时的事件。Delegate 中定义的所有方法都是可选实现的。

名称	描述
vendorKey	Agora 为应用程序开发者签发的厂商秘钥。如果没有，请向 Agora 申请。
delegate	

开启视频模式 (enableVideo)

```
(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。使用 `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`)

(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模式。使用 `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 (`joinChannelByKey`)

```
(int)joinChannelByKey:(NSString *)key channelName:(NSString *)channelName info:(NSString *)info uid:(NSUInteger)uid joinChannelSuccess:(void (^)(NSString* channel, NSUInteger uid, NSInteger elapsed))joinChannelSuccessBlock;
```

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，户必须调用

leaveChannel 退出当前通话，才能进入下一个频道。该方法是异步的，所以可以在主用户界面线程被调用。

注：SDK 在通话中使用 iOS 系统的 AVAudioSession 共享对象进行录音和播放，应用程序对该对象的操作可能会影响 SDK 的音频相关功能。在 joinChannel 时，SDK 调用 setCategory AVAudioSessionCategoryPlayAndRecord 将 AVAudioSession 设置到 PlayAndRecord 模式，应用程序不应将其设置到其他模式。设置该模式时，正在播放的声音会被打断（比如正在播放的响铃声）。

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”。
info	可选。程序员想要加入的任何附加信息。
uid	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。
joinSuccessBlock	
channel	频道名称
uid	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	频道进入延迟（毫秒）

离开频道 (leaveChannel)

```
(int)leaveChannel:(void(^)(AgoraRtcStats* stat))leaveChannelBlock;
```

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
leaveChannelBlock	离开频道成功的回调。
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(NSString*) 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 AgoraRtc_Error_DynamicKeyTimeout(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 (AgoraRtcChannelProfile) 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： <ul style="list-style-type: none">AgoraRtc_ChannelProfile_Free（默认）AgoraRtc_ChannelProfile_BroadcasterAgoraRtc_ChannelProfile_Audience
Return Value	0: 方法调用成功 <0: 方法调用失败

获取通话 ID (getCallId)

(NSString*) 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)

```
(int) rate:(NSString*)callId rating:(NSInteger) rating
description(NSString*) 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 方法获取的通话 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)

```
(int) complain:(NSString*) callId description(NSString*) description;
```

该方法能够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 方法获取的通话 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。

启动语音通话测试 (startEchoTest)

```
(int)startEchoTest :(void(^)(NSString* channel, NSUInteger uid,
NSInteger elapsed))successBlock;
```

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
successBlock	启动 echo test 成功回调，回调参数含义见 joinChannelByKey 的 joinSuccessBlock
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。
--	--------------------------------------

终止应答测试 (stopEchoTest)

(int) stopEchoTest;

该方法用于终止应答测试。

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)

(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 networkQualityBlock 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。网络测试默认为关闭状态。

名称	描述
Return Value	0: 方法调用成功 <0: 方法调用失败

禁用网络测试 (disableNetworkTest)

(int) disableNetworkTest;

该方法禁用网络链接质量测试。

名称	描述
Return Value	0: 方法调用成功 <0: 方法调用失败

报告通话质量 URL (makeQualityReportUrl)

**(NSString*) makeQualityReportUrl:(NSString*) channel
listenerUid:(NSUInteger) listenerUid
speakerUid:(NSUInteger) speakerUid
reportFormat:(AgoraRtcQualityReportFormat) reportFormat;**

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 方法中指定的频道名

listenerUid	听话方用户 ID
speakerUid	说话方用户 ID
reportFormat	报告格式类型 AgoraRtc_QualityReportFormat_Json (0): JSON。返回质量报告数据，应用程序根据需要展示给用户。 AgoraRtc_QualityReportFormat_Html (1): HTML。返回 HTML 格式报告，可以直接通过浏览器或 WebView 组件直接展示给用户。
Return Value	生成的 URL。

静音 (muteLocalAudioStream)

```
(int)muteLocalAudioStream: (BOOL)muted;
```

开启/取消客户端静音。

名称	描述
mute	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

静音所有远端音频流 (muteAllRemoteAudioStreams)

```
(int)muteAllRemoteAudioStreams: (BOOL)muted;
```

将所有远端用户静音或取消静音。

名称	描述
muted	Yes: 停止播放所有接收音频流 No: 继续播放接收到的音频流
Return Value	0: 方法调用成功 <0: 方法调用失败

静音指定远端音频流 (muteRemoteAudioStream)

```
(int)muteRemoteAudioStream : (NSUInteger)uid  
muted: (BOOL)muted;
```

将所有远端用户静音或取消静音。

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	Yes: 停止播放指定用户音频流 No: 继续播放指定用户音频流
Return Value	0: 方法调用成功 <0: 方法调用失败

扬声器通话 (setEnabledSpeakerphone)

(int)setEnabledSpeakerphone:(BOOL)enableSpeaker;

启用扬声器。

注：由于 SDK 调用 setCategory AVAudioSessionCategoryPlayAndRecord withOptions 设置听筒/扬声器状态，正在播放的声音会被打断。

名称	描述
enableSpeaker	Yes: 切换到扬声器 No: 切换到听筒
Return Value	0: 方法调用成功 <0: 方法调用失败

是否是扬声器状态 (isSpeakerphoneEnabled)

(BOOL)isSpeakerphoneEnabled;

当前是否是扬声器状态。

名称	描述
Return Value	Yes: 是 No: 否

设定扬声器音量 (setSpeakerphoneVolume)

(int)setSpeakerphoneVolume:(NSInteger)volume;

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)

**(int)enableAudioVolumeIndication:(NSInteger)interval
smooth:(NSInteger)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

设置视频 Profile (setVideoProfile)

(int)setVideoProfile: (AgoraRtcVideoProfile)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 AgoraRtcVideoProfile 的定义。
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

设置本地视频视图 (setupLocalVideo)

(int) setupLocalVideo: (AgoraRtcVideoCanvas*) 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	<p>设置视频属性。AgoraRtcVideoCanvas 类型。</p> <ul style="list-style-type: none"> view: 视频显示视窗。SDK 不维护 view 的生命周期，应用程序应保证 view 在通话中是有效的，否则可能会导致 SDK 崩溃。view 可以在 leaveChannel 调用返回后安全销毁。由于 view 的值会缓存在 SDK 中，切换 view 或者销毁 view 前，应调用 setupLocalVideo 指定 view 为 NULL 清除缓存。 renderMode: 视频显示模式。 AgoraRtc_Render_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。

	<p>AgoraRtc_Render_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。</p> <p>AgoraRtc_Render_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 AgoraRtc_Render_Hidden; 如果对方和自己一个竖屏一个横屏, 则使用 AgoraRtc_Render_Fit。</p> <ul style="list-style-type: none"> uid: 本地用户 ID, 设置为 0。
Return Value	<p>0: 方法调用成功</p> <p><0: 方法调用失败</p>

设置远端视频显示视图 (setupRemoteVideo)

(int) setupRemoteVideo: (AgoraRtcVideoCanvas*) 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 didJoinedOfUid 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	<p>设置视频属性。AgoraRtcVideoCanvas 类型。</p> <ul style="list-style-type: none"> view: 视频显示视窗。SDK 不维护 view 的生命周期, 应用程序应保证 view 在通话中是有效的, 否则可能会导致 SDK 崩溃。view 可以在 leaveChannel 调用返回后安全销毁。由于 view 的值会缓存在 SDK 中, 切换 view 或者销毁 view 前, 应调用 setupRemoteVideo 指定 view 为 NULL 清除缓存。 renderMode: 视频显示模式。 <p>AgoraRtc_Render_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。</p> <p>AgoraRtc_Render_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。</p> <p>AgoraRtc_Render_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 AgoraRtc_Render_Hidden; 如果对方和自己一个竖屏一个横屏, 则使用 AgoraRtc_Render_Fit。</p> <ul style="list-style-type: none"> uid: 用户 ID, 指定远程视频来自哪个用户。
Return Value	<p>0: 方法调用成功</p> <p><0: 方法调用失败</p>

设置本地视频显示模式 (setLocalRenderMode)

(int) setLocalRenderMode: (AgoraRtcRenderMode) 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。AgoraRtcRenderMode 类型。 <ul style="list-style-type: none">AgoraRtc_Render_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。AgoraRtc_Render_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。AgoraRtc_Render_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 AgoraRtc_Render_Hidden; 如果对方和自己一个竖屏一个横屏, 则使用 AgoraRtc_Render_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)

```
(int)setRemoteRenderMode: (NSUInteger)uid mode: (AgoraRtcRenderMode)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, 指定远程视频来自哪个用户。
mode	设置视频显示模式。AgoraRtcRenderMode 类型。 <ul style="list-style-type: none">AgoraRtc_Render_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。AgoraRtc_Render_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。AgoraRtc_Render_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 AgoraRtc_Render_Hidden; 如果对方和自己一个竖屏一个横屏, 则使用 AgoraRtc_Render_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)

```
(int)switchView: (NSUInteger)uid1 andAnother: (NSUInteger)uid2;
```


该方法用于切换不同用户视频流的显示视窗。

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。
Return Value	0: 方法调用成功 <0: 方法调用失败

切换前置/后置摄像头 (switchCamera)

(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)

(int)muteLocalVideoStream: (BOOL)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: 暂停发送本地视频流 False: 恢复发送本地视频流
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)

(int)muteAllRemoteVideoStreams: (BOOL)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)

**(int)muteRemoteVideoStream : (NSUInteger)uid
muted: (BOOL)muted;**

暂停/恢复指定用户视频流。本方法用于允许/禁止播放指定用户的视频流。

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
uid	用户 ID
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

开始客户端录音 (startAudioRecording)

(int) startAudioRecording: (NSString*) 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)

(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

设置日志文件 (setLogFile)

(int) setLogFile: (NSString*) 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

设置日志文件 (setLogFilter)

(int) setLogFilter: (NSUInteger) 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： <ul style="list-style-type: none"> 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

2. 代理方法 (AgoraRtcEngineDelegate)

SDK 会通过 Delegate 给 APP 上报一些运行时事件。从 1.1 版本开始，SDK 使用 Delegate 代替原有的部分 Block 回调。原有的 Block 回调被标为废弃，目前仍然可以使用，但是建议用相应的 Delegate 方法代替。如果同一个回调 Block 和 Delegate 方法都有定义，则 SDK 只回调 Block 方法。

发生警告回调 (didOccurWarning)

(void)rtcEngine:(AgoraRtcEngineKit *)engine
didOccurWarning:(AgoraRtcErrorCode)warningCode

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。比如和服务端失去连接时，SDK 可能会上报 AgoraRtc_Error_OpenChannelTimeout(106) 警告，同时自动尝试重连。

名称	描述
warningCode	警告代码

发生错误回调 (didOccurError)

(void)rtcEngine:(AgoraRtcEngineKit *)engine
didOccurError:(AgoraRtcErrorCode)errorCode

该回调方法表示 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 AgoraRtc_Error_StartCall(1002) 错误。APP 可以提示用户启动通话失败，并调用 leaveChannel 退出频道。

名称	描述
Engine	AgoraRtcEngineKit 对象
errorCode	错误代码

音量提示回调 ([reportAudioVolumeIndicationOfSpeakers](#))

```
(void)rtcEngine:(AgoraRtcEngineKit *)engine  
reportAudioVolumeIndicationOfSpeakers:(NSArray*)speakers  
totalVolume:(NSInteger)totalVolume
```

同 [audioVolumeIndicationBlock](#)。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() ： <ul style="list-style-type: none">uid：说话者的用户 IDvolume：说话者的音量（0~255）
totalVolume	（混音后的）总音量（0~255）

本地首帧视频显示回调 ([firstLocalVideoFrameWithSize](#))

```
(void)rtcEngine:(AgoraRtcEngineKit *)engine  
firstLocalVideoFrameWithSize:(CGSize)size elapsed:(NSInteger)elapsed
```

同 [firstLocalVideoFrameBlock](#)。提示第一帧本地视频画面已经显示在屏幕上。

名称	描述
size	视频流尺寸（宽度和高度）
elapsed	加入频道开始到该回调触发的延迟（毫秒）

远端首帧视频接收解码回调 ([firstRemoteVideoDecodedOfUid](#))

```
(void)rtcEngine:(AgoraRtcEngineKit *)engine  
firstRemoteVideoDecodedOfUid:(NSUInteger)uid size:(CGSize)size  
elapsed:(NSInteger)elapsed
```

同 [firstRemoteVideoDecodedBlock](#)。提示已收到第一帧远程视频流并解码。

名称	描述
uid	用户 ID，指定是哪个用户的视频流
size	视频流尺寸（宽度和高度）
elapsed	加入频道开始到该回调触发的延迟（毫秒）

远端首帧视频显示回调 ([firstRemoteVideoFrameOfUid](#))

```
(void)rtcEngine:(AgoraRtcEngineKit *)engine  
firstRemoteVideoFrameOfUid:(NSUInteger)uid size:(CGSize)size  
elapsed:(NSInteger)elapsed
```

同 [firstRemoteVideoFrameBlock](#)。提示第一帧远端视频画面已经显示在屏幕上。

名称	描述
uid	用户 ID，指定是哪个用户的视频流

size	视频流尺寸（宽度和高度）
elapsed	加入频道开始到该回调触发的延迟（毫秒）

用户加入回调 (didJoinedOfUid)

(void)rtcEngine:(AgoraRtcEngineKit *)engine didJoinedOfUid:(NSUInteger)uid elapsed:(NSInteger)elapsed

同 **userJoinedBlock**。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。

名称	描述
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）

用户离线回调 (didOfflineOfUid)

(void)rtcEngine:(AgoraRtcEngineKit *)engine didOfflineOfUid:(NSUInteger)uid reason:(AgoraRtcUserOfflineReason)reason

同 **userOfflineBlock**。提示有用户离开了频道（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。

名称	描述
uid	用户 ID。
reason	离线原因： <ul style="list-style-type: none"> AgoraRtc_UserOffline_Quit：用户主动离开 AgoraRtc_UserOffline_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。

用户音频静音回调 (didAudioMuted)

(void)rtcEngine:(AgoraRtcEngineKit *)engine didAudioMuted:(BOOL)muted byUid:(NSUInteger)uid

同 **userMuteAudioBlock**。提示有用户用户将通话静音/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	Yes: 静音 No: 取消静音
-------	---------------------

用户停止/重新发送视频回调 (**didVideoMuted**)

(void)rtcEngine:(AgoraRtcEngineKit *)engine didVideoMuted:(BOOL)muted
byUid:(NSUInteger)uid

同 **userMuteVideoBlock**。提示有其他用户暂停发送/恢复发送其视频流。

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	Yes: 该用户已暂停发送其视频流 No: 该用户已恢复发送其视频流

本地视频统计回调 (**localVideoStatWithSentBitrate**)

(void)rtcEngine:(AgoraRtcEngineKit *)engine
localVideoStatWithSentBitrate:(NSInteger)sentBitrate
sentFrameRate:(NSInteger)sentFrameRate

同 **localVideoStatBlock**。报告更新本地视频统计信息，该回调方法每两秒触发一次。

名称	描述
sentBytes	(上次统计后) 发送的字节数
sentFrames	(上次统计后) 发送的帧数

远端视频统计回调 (**remoteVideoStatOfUid**)

(void)rtcEngine:(AgoraRtcEngineKit *)engine
remoteVideoStatOfUid:(NSUInteger)uid delay:(NSInteger)delay
receivedBitrate:(NSInteger)receivedBitrate
receivedFrameRate:(NSInteger)receivedFrameRate

同 **remoteVideoStatBlock**。报告更新远端视频统计信息，该回调方法每两秒触发一次。

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	(上次统计后) 接收到的码率(kbps)
receivedFrameRate	(上次统计后) 接收帧率(fps)

摄像头启用回调 (**rtcEngineCameraDidReady**)

(void)rtcEngineCameraDidReady:(AgoraRtcEngineKit *)engine

同 **cameraReadyBlock**。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(**rtcEngineVideoDidStop**)

(void)rtcEngineVideoDidStop:(AgoraRtcEngineKit *)engine

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。

网络连接丢失回调 (**rtcEngineConnectionDidLost**)

(void)rtcEngineConnectionDidLost:(AgoraRtcEngineKit *)engine

同 **connectionLostBlock**。该回调方法表示 SDK 和服务器失去了网络连接。

加入频道成功回调 (didJoinChannel)

(void)rtcEngine:(AgoraRtcEngineKit *)engine
didJoinChannel:(NSString*)channel withUid:(NSUInteger)uid elapsed:(NSInteger)
elapsed

同 joinChannelByKey API 的 joinSuccessBlock。该回调方法表示该客户端成功加入了指定的频道。

名称	描述
channel	频道名
uid	用户 ID。如果 joinChannelByKey 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 (didRejoinChannel)

(void)rtcEngine:(AgoraRtcEngineKit *)engine
didRejoinChannel:(NSString*)channel withUid:(NSUInteger)uid elapsed:(NSInteger)
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，提示有用户重新加入了频道，且频道 ID 和用户 ID 均已分配。

名称	描述
channel	频道 ID
uid	用户 ID
elapsed	加入延迟（毫秒）

Rtc Engine 统计数据回调 (reportRtcStats)

(void)rtcEngine:(AgoraRtcEngineKit *)engine
reportRtcStats:(AgoraRtcStats*)stats

同 rtcStatsBlock。该回调定期上报 Rtc Engine 的运行时的状态，每两秒触发一次。

AgoraRtcStats	描述
duration	通话时长，累计值
txBytes	发送字节数，累计值
rxBytes	接收字节数，累计值

名称	描述
stat	见上表

语音质量回调 (audioQualityOfUid)

```
(void)rtcEngine:(AgoraRtcEngineKit *)engine  
audioQualityOfUid:(NSUInteger)uid quality:(AgoraRtcQuality)quality  
delay:(NSUInteger)delay lost:(NSUInteger)lost
```

同 `audioQualityBlock`。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。

名称	描述
audioQualityBlock	
uid	用户 ID。每次回调上报一个用户当前的语音质量（不包含自己），不同用户在不同的回调中上报。
quality	声音质量评分： AgoraRtc_Quality_Excellent(1) AgoraRtc_Quality_Good(2) AgoraRtc_Quality_Poor(3) AgoraRtc_Quality_Bad(4) AgoraRtc_Quality_VBad(5) AgoraRtc_Quality_Down(6)
delay	延迟（毫秒）
lost	丢包率（百分比）

网络质量回调 (networkQuality)

```
(void)rtcEngine:(AgoraRtcEngineKit *)engine  
networkQuality:(AgoraRtcQuality)quality
```

同 `networkQualityBlock`。报告网络质量，该回调方法每两秒触发一次。

名称	描述
quality	网络质量评分： AgoraRtc_Quality_Unknown = 0 AgoraRtc_Quality_Excellent = 1 AgoraRtc_Quality_Good = 2 AgoraRtc_Quality_Poor = 3 AgoraRtc_Quality_Bad = 4 AgoraRtc_Quality_VBad = 5 AgoraRtc_Quality_Down = 6

3. 回调方法

SDK 会通过 Block 给 APP 上报一些运行时事件，以下为 SDK 中的回调方法说明。

用户加入回调 (userJoinedBlock)

```
(void)userJoinedBlock: (void (^)(NSUInteger uid, NSInteger elapsed))userJoinedBlock;
```

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。

名称	描述
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）

用户离线回调 (userOfflineBlock)

```
(void)userOfflineBlock: (void (^)(NSUInteger uid))userOfflineBlock;
```

提示有用户离开了频道（或掉线）。注：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。

名称	描述
uid	用户 ID。

重新加入频道回调 (rejoinChannelSuccessBlock)

```
(void)rejoinChannelSuccessBlock: (void (^)(NSString* channel, NSUInteger uid, NSInteger elapsed))rejoinChannelSuccessBlock;
```

有时候由于网络原因，客户端可能会和服务端失去连接，SDK 会进行自动重连，自动重连成功后触发此回调方法，提示有用户重新加入了频道，且频道 ID 和用户 ID 均已分配。

名称	描述
channel	频道 ID
uid	用户 ID
elapsed	加入延迟（毫秒）

离开频道回调 (leaveChannelBlock)

```
(void)leaveChannelBlock: (void (^)(AgoraRtcStats* stat))leaveChannelBlock;
```

显示用户离开了频道，提供会话数据信息，包括通话时长和 tx/rx 字节数。

名称	描述
stat	见下表

AgoraRtcStats	描述
duration	通话时长，累计值
txBytes	发送字节数，累计值
rxBytes	接收字节数，累计值

Rtc Engine 统计数据回调 (rtcStatsBlock)

```
(void)rtcStatsBlock:(void (^)(AgoraRtcStats* stat))rtcStatsBlock;
```

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
stat	见上表

音量提示回调 (audioVolumeIndicationBlock)

```
(void)audioVolumeIndicationBlock:(void (^)(NSArray *speakers,
NSInteger totalVolume))audioVolumeIndicationBlock;
```

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()： <ul style="list-style-type: none">uid: 说话者的用户 IDvolume: 说话者的音量（0~255）
totalVolume	（混音后的）总音量（0~255）

本地首帧视频显示回调 (firstLocalVideoFrameBlock)

```
(void)firstLocalVideoFrameBlock:(void (^)(NSInteger width, NSInteger
height, NSInteger elapsed))firstLocalVideoFrameBlock;
```

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）

远端首帧视频显示回调 (firstRemoteVideoFrameBlock)

```
(void)firstRemoteVideoFrameBlock:(void (^)(NSUInteger uid, NSInteger
width, NSInteger height, NSInteger elapsed))firstRemoteVideoFrameBlock;
```

提示第一帧远端视频画面已经显示在屏幕上。

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）

远端首帧视频接收解码回调 (firstRemoteVideoDecodedBlock)

```
(void)firstRemoteVideoDecodedBlock:(void (^)(NSUInteger uid, NSInteger
width, NSInteger height, NSInteger
elapsed))firstRemoteVideoDecodedBlock;
```

提示已收到第一帧远程视频流并解码。

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）

用户音频静音回调 (userMuteAudioBlock)

```
(void)userMuteAudioBlock:(void (^)(NSUInteger uid, bool muted))userMuteAudioBlock;
```

提示有用户用户将通话静音/取消静音。

名称	描述
uid	用户 ID
muted	Yes: 静音 No: 取消静音

用户停止/重新发送视频回调 (userMuteVideoBlock)

```
(void)userMuteVideoBlock:(void (^)(NSUInteger uid, bool muted))userMuteVideoBlock;
```

提示有其他用户暂停发送/恢复发送其视频流。

名称	描述
uid	用户 ID
muted	Yes: 该用户已暂停发送其视频流 No: 该用户已恢复发送其视频流

本地视频统计回调 (localVideoStatBlock)

```
(void)localVideoStatBlock:(void (^)(NSInteger sentBytes, NSInteger sentFrames))localVideoStatBlock;
```

报告更新本地视频统计信息，该回调方法每两秒触发一次。

名称	描述
sentBytes	（上次统计后）发送的字节数
sentFrames	（上次统计后）发送的帧数

远端视频统计回调 (remoteVideoStatBlock)

```
(void)remoteVideoStatBlock:(void (^)(NSUInteger uid, NSInteger frameCount, NSInteger delay, NSInteger receivedBytes))remoteVideoStatBlock;
```

报告更新远端视频统计信息，该回调方法每两秒触发一次。

名称	描述
uid	用户 ID，指定是哪个用户的视频流

frameCount	（上次统计后）接收的总帧数
delay	延时(毫秒)
receivedBytes	（上次统计后）接收的总字节数

摄像头启用回调 (cameraReadyBlock)

(void)cameraReadyBlock:(void (^)(void))cameraReadyBlock;

提示已成功打开摄像头，可以开始捕获视频。

语音质量回调 (audioQualityBlock)

(void)audioQualityBlock:(void (^)(NSUInteger uid, AgoraRtcQuality quality, NSUInteger delay, NSUInteger lost))audioQualityBlock;

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。

名称	描述
audioQualityBlock	
uid	用户 ID。每次回调上报一个用户当前的语音质量（不包含自己），不同用户在不同的回调中上报。
quality	声音质量评分： AgoraRtc_Quality_Excellent(1) AgoraRtc_Quality_Good(2) AgoraRtc_Quality_Poor(3) AgoraRtc_Quality_Bad(4) AgoraRtc_Quality_VBad(5) AgoraRtc_Quality_Down(6)
delay	延迟（毫秒）
lost	丢包率（百分比）

网络质量回调 (networkQualityBlock)

(void)networkQualityBlock:(void (^)(AgoraRtcQuality quality))networkQualityBlock;

报告网络质量，该回调方法每两秒触发一次。

名称	描述
quality	网络质量评分： AgoraRtc_Quality_Unknown = 0 AgoraRtc_Quality_Excellent = 1 AgoraRtc_Quality_Good = 2 AgoraRtc_Quality_Poor = 3 AgoraRtc_Quality_Bad = 4 AgoraRtc_Quality_VBad = 5 AgoraRtc_Quality_Down = 6

网络连接丢失回调 (connectionLostBlock)

```
(void)connectionLostBlock: (void(^)())connectionLostBlock;
```

该回调方法表示 SDK 和服务器失去了网络连接。

