

阿里云 音视频通信

SDK参考

文档版本：20191108

法律声明

阿里云提醒您阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云文档中所有内容，包括但不限于图片、架构设计、页面布局、文字描述，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 禁止： 重置操作将丢失用户配置数据。
	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告： 重启操作将导致业务中断，恢复业务时间约十分钟。
	用于警示信息、补充说明等，是用户必须了解的内容。	 注意： 权重设置为0，该服务器不会再接受新请求。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明： 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	单击设置 > 网络 > 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
##	表示参数、变量。	<code>bae log list --instanceid Instance_ID</code>
[]或者[a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ }或者{a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

法律声明.....	I
通用约定.....	I
1 SDK下载.....	1
2 Android SDK.....	2
2.1 数据类型.....	2
2.2 回调及监听.....	8
2.3 AliRtcEngine接口.....	14
3 iOS和Mac SDK.....	34
3.1 数据类型.....	34
3.2 回调及监听.....	39
3.3 AliRtcEngine接口.....	44
4 Windows SDK.....	63
4.1 数据类型.....	63
4.2 回调及监听.....	66
4.3 AliRtcEngine接口.....	70
4.4 AliMediaDeviceTestInterface接口.....	88
5 Web SDK.....	90
5.1 回调及监听.....	90
5.2 AliRtcEngine接口.....	91
6 客户端错误码表.....	102

1 SDK下载

本文为您介绍音视频通信客户端SDK的相关文档链接，帮助您快速了解并使用音视频通信相关功能。



说明：

目前音视频通信API支持版本：Java、Python、PHP、Go、.NET、Node.js。获取更多版本SDK，请单击[阿里云SDK中心](#)。

客户端SDK

客户端	集成SDK	API参考	SDK下载地址	示例代码
iOS	#unique_4	iOS接口说明	V1.14	iOS示例代码
Android	#unique_6	Android接口说明	V1.14	Android示例代码
Windows	#unique_8	Windows接口说明	V1.13	Windows示例代码
Mac	#unique_10	Mac接口说明	V1.13	Mac示例代码
Web	#unique_11	Web接口说明	V1.9.1	Web示例代码

2 Android SDK

2.1 数据类型

本文档为您列出了Android SDK的所有数据类型，您可以查看每一个数据类型下的枚举名。

目录

数据类型	描述	以上版本支持
AliRtcVideoTrack	视频track类型	1.1
AliRtcAudioTrack	音频track类型	1.1
AliRTCCameraType	摄像头类型	1.1
AliRtcNetworkQuality	网络质量	1.1
AliRtcRenderMode	渲染模式	1.1
AliRtcRenderMirrorMode	镜像模式	1.11
AliRtcLogLevel	log级别	1.1
AliVideoCanvas	渲染画布	1.1
AliRtcAuthInfo	鉴权信息	1.1
AliRtcVideoProfile	视频质量	1.1
AliRtcRemoteUserInfo	远端用户信息	1.1
AliRTCIImageFormat	图片格式枚举	1.14
AliStatusInfo	用户信息	1.14
AliStatusState	用户状态	1.14
AliDetectObserver	人脸识别预处理接口	1.14

详情

数据类型详情如下所示。

- **AliRtcVideoTrack**：视频track的类型。

枚举名	描述
AliRtcVideoTrackNo	无视频流
AliRtcVideoTrackCamera	相机流

枚举名	描述
AliRtcVideoTrackScreen	屏幕共享流
AliRtcVideoTrackBoth	相机流和屏幕共享流

- AliRtcAudioTrack：音频track的类型。

枚举名	描述
AliRtcAudioTrackNo	无音频流
AliRtcAudioTrackMic	麦克风流

- AliRTCCameraType：摄像头类型。

枚举名	描述
AliRTCCameraInvalid	无效
AliRTCCameraBack	后置摄像头
AliRTCCameraFront	前置摄像头
AliRTCCameraUsb	usb摄像头

- AliRtcNetworkQuality：网络状态。

枚举名	描述
Network_Excellent	极好
Network_Good	好
Network_Poor	一般
Network_Bad	差
Network_VeryBad	极差
Network_Disconnected	网络中断
Network_Unknow	未知原因

- AliRtcRenderMode：渲染模式。

枚举名	描述
AliRtcRenderModeAuto	自动
AliRtcRenderModeStretch	拉伸填充视图，不保持视频比例
AliRtcRenderModeFill	在保持视频宽高比的同时缩放，填充黑边
AliRtcRenderModeClip	在保持视频宽高比的同时缩放，并裁剪以适合视图

- **AliRtcRenderMirrorMode**: 镜像模式。

枚举名	描述
AliRtcRenderMirrorModeOnlyFront	只有前置摄像头预览镜像，其余不镜像
AliRtcRenderMirrorModeAllEnabled	全部镜像
AliRtcRenderMirrorModeAllDisable	全部不镜像

- **AliRtcLogLevel**: log级别。

枚举名	描述
AliRtcLogLevelDump	全量
AliRtcLogLevelDebug	调试
AliRtcLogLevelVerbose	详细
AliRtcLogLevelInfo	普通
AliRtcLogLevelWarn	警告
AliRtcLogLevelError	错误
AliRtcLogLevelFatal	严重
AliRtcLogLevelNone	无

- **AliVideoCanvas**: 渲染画布。

枚举名	描述
SophonSurfaceView	渲染视图
AliRtcRenderMode	渲染模式
AliRtcRenderMirrorMode	镜像模式

- **AliRtcAuthInfo**: 鉴权信息。



注意:

在1.9及以上版本移除session id字段。

字段名	类型	描述
mConferenceId	String	频道号
mUserId	String	用户ID
mAppid	String	应用ID
mNonce	String	随机串
mTimestamp	long	时间戳

字段名	类型	描述
mToken	String	令牌
mGslb	String[]	gslb地址

- AliRtcVideoProfile：视频流规格。



说明：

发布视频流规格，默认为AliRTCSdk_Video_Profile_Default，您可以在joinChannel之前或joinChannel之后设置，但是您必须在本地流发布之前通过setVideoProfile方法设置。

枚举名	描述
AliRTCSdk_Video_Profile_Default	默认，分辨率480 * 640，帧率15
AliRTCSdk_Video_Profile_180_320P_15	分辨率180 * 320，帧率15
AliRTCSdk_Video_Profile_180_320P_30	分辨率180 * 320，帧率30
AliRTCSdk_Video_Profile_360_640P_15	分辨率360 * 640，帧率15
AliRTCSdk_Video_Profile_360_640P_30	分辨率360 * 640，帧率30
AliRTCSdk_Video_Profile_480_640P_15	分辨率480 * 640，帧率15
AliRTCSdk_Video_Profile_480_640P_30	分辨率480 * 640，帧率30
AliRTCSdk_Video_Profile_720_1280P_15	分辨率720 * 1280，帧率15
AliRTCSdk_Video_Profile_720_1280P_30	分辨率720 * 1280，帧率30
AliRTCSdk_Video_Profile_Max	占位值

- AliRtcRemoteUserInfo：远端用户信息。

字段名	类型	描述
userID	String	远端用户userid
sessionID	String	远端用户sessionid
callID	String	远端用户caillid
displayName	String	远端用户名称

streamLabel	String	远端用户流标签
firstSubscribe	boolean	是否第一次订阅
muteAudioPlaying	boolean	是否静音远端用户
isOnline	boolean	远端用户是否在线
cameraCanvas	AliVideoCanvas	相机流显示画布
screenCanvas	AliVideoCanvas	屏幕流显示画布
hasAudio	boolean	远端用户是否推音频流
hasCameraMaster	boolean	远端用户是否推相机大流
hasCameraSlave	boolean	远端用户是否推相机小流
hasScreenSharing	boolean	远端用户是否推屏幕流
subAudio	boolean	是否订阅了远端用户的音频流
subCamera	boolean	是否订阅了远端用户的相机流
subCameraMaster	boolean	是否订阅了远端用户的相机大流
subScreenSharing	boolean	是否订阅了远端用户的屏幕流
requestAudio	boolean	是否请求订阅了远端用户的音频流
requestCamera	boolean	是否请求订阅了远端用户的相机流
requestCameraMaster	boolean	是否请求订阅了远端用户的相机大流
requestScreenSharing	boolean	是否请求订阅了远端用户的屏幕流

- **AliRTCImageFormat**: 图片格式枚举。

枚举名	描述
ALIRTC_IMAGE_FORMATUNKNOWN	格式未知
ALIRTC_IMAGE_FORMATARGB	格式ARGB
ALIRTC_IMAGE_FORMATBGRA	格式BGRA
ALIRTC_IMAGE_FORMATRGBA	格式RGBA
ALIRTC_IMAGE_FORMATYUV420P	格式YUV420P
ALIRTC_IMAGE_FORMATYUVYV12	格式YUVYV12
ALIRTC_IMAGE_FORMATYUVNV21	格式YUVNV21

ALIRTC_IMAGE_FORMATYUVNV12	格式YUVNV12
ALIRTC_IMAGE_FORMATYUVNV12_FULL_RANGE	格式YUVNV12_FULL_RANGE
ALIRTC_IMAGE_FORMATYUVJ420P	格式YUVJ420P
ALIRTC_IMAGE_FORMATYUV420SP	格式YUV420SP
ALIRTC_IMAGE_FORMATYUVJ420SP	格式YUVJ420SP
ALIRTC_IMAGE_FORMATYUVJ444P	格式YUVJ444P
ALIRTC_IMAGE_FORMATYUVJ422P	格式YUVJ422P
ALIRTC_IMAGE_FORMATYUV444P	格式YUV444P
ALIRTC_IMAGE_FORMATYUV2	格式YUV2
ALIRTC_IMAGE_FORMATMJPEG	格式MJPEG
ALIRTC_IMAGE_FORMATTEXTURE_2D	格式TEXTURE_2D
ALIRTC_IMAGE_FORMATTEXTURE_OES	格式TEXTURE_OES

- AliStatusInfo: 用户信息。

字段名	类型	描述
user_id	String	用户ID
status	<i>AliStatusState</i>	状态

- AliStatusState: 用户状态。

字段名	类型	描述
mic_open	boolean	mic是否开启
mic_interrupted	boolean	mic是否被打断
cam_open	boolean	相机是否打开
audio_disabled	boolean	音频是否不可用
video_disabled	boolean	视频是否不可用
background_mode	boolean	是否后台模式

- AliDetectObserver: 人脸识别预处理接口。

方法名	描述
onData	人脸识别回调接口

示例方法:

```
long onData(long dataFrameY, long dataFrameU, long dataFrameV,
ALI_RTC_INTERFACE.AliRTCImageFormat aliRTCImageFormat, int width,
```

```
int height, int strideY, int strideU, int strideV, int rotate, long
extraData);
```

参数	类型	描述
<code>dataFrameY</code>	<code>long</code>	Y分量指针
<code>dataFrameU</code>	<code>long</code>	U分量指针
<code>dataFrameV</code>	<code>long</code>	V分量指针, NV12和NV21该指针为null
<code>aliRTCImageFormat</code>	<code>ALI_RTC_INTERFACE.AliRTCImageFormat</code>	图像数据格式
<code>width</code>	<code>int</code>	图像宽度
<code>height</code>	<code>int</code>	图像高度
<code>strideY</code>	<code>int</code>	图像Y分量stride
<code>strideU</code>	<code>int</code>	图像U分量stride
<code>strideV</code>	<code>int</code>	图像V分量stride
<code>rotate</code>	<code>int</code>	图像旋转角度
<code>extraData</code>	<code>long</code>	附加字段 (非定制化可忽略)

2.2 回调及监听

本文档为您介绍Android SDK回调及监听的接口详情, AliRtcSDK为您提供AliRtcEngineEventListener (本地用户行为回调) 和AliRtcEngineNotify (远端用户行为回调)。

目录



说明:

回调接口都在子线程。

AliRtcEngineEventListener: 本地用户行为回调。

API	描述	以上版本支持
<i>onJoinChannelResult</i>	加入频道回调	1.1
<i>onLeaveChannelResult</i>	离开频道回调	1.1
<i>onPublishResult</i>	发布音视频流回调	1.1
<i>onUnpublishResult</i>	停止发布音视频流回调	1.1
<i>onSubscribeResult</i>	订阅回调	1.1

API	描述	以上版本支持
<i>onUnsubscribeResult</i>	取消订阅回调	1.1
<i>onNetworkQualityChanged</i>	网络状态变化时回调	1.1
<i>onOccurWarning</i>	warning回调	1.1
<i>onOccurError</i>	error回调	1.1
<i>onSubscribeChangedNotify</i>	订阅结果回调	1.1
<i>onBye</i>	被服务器踢出或者频道关闭时回调	1.1
<i>onFirstPacketSent</i>	首帧数据发送成功回调	1.1
<i>onFirstPacketReceived</i>	首包数据接收成功	1.13
<i>onConnectionLost</i>	连接断开回调	1.14
<i>onTryToReconnect</i>	重新尝试连接回调	1.14
<i>onConnectionRecovery</i>	连接恢复回调	1.14
<i>onParticipantStatusNotify</i>	远端用户状态改变通知	1.14

AliRtcEngineNotify: 远端用户行为回调。

API	描述	以上版本支持
<i>onRemoteUserUnPublish</i>	当远端用户停止发布时回调	1.1
<i>onRemoteUserOnLineNotify</i>	远端用户上线回调	1.1
<i>onRemoteUserOffLineNotify</i>	远端用户下线回调	1.1
<i>onRemoteTrackAvailableNotify</i>	远端用户音视频流发生变化时回调	1.1

接口详情

- **onJoinChannelResult: 加入频道回调。**

```
void onJoinChannelResult(int result)
```

参数:

参数	类型	描述
result	int	0为加入频道成功, 非0为失败。

- **onLeaveChannelResult**: 离开频道回调。

```
void onLeaveChannelResult(int result)
```

参数:

参数	类型	描述
result	int	0为离开频道成功, 非0为失败。

- **onPublishResult**: 发布音视频流回调。

```
void onPublishResult(int result, String publishId)
```

参数:

参数	类型	描述
result	int	0为发布成功, 非0为失败。
publishId	String	流ID。

- **onUnpublishResult**: 停止发布音视频流回调。

```
void onUnpublishResult(int result)
```

参数:

参数	类型	描述
result	int	0为停止发布成功, 非0为失败。

- **onSubscribeResult**: 订阅回调。

```
void onSubscribeResult(String uid, int result, AliRtcVideoTrack vt, AliRtcAudioTrack at)
```

参数:

参数	类型	描述
uid	String	用户ID, 从App server分配的唯一标示符。
result	int	0表示订阅成功, 非0表示失败。
vt	AliRtcVideoTrack	订阅成功的视频流。
at	AliRtcAudioTrack	订阅成功的音频流。

- **onUnsubscribeResult**: 取消订阅回调。

```
void onUnsubscribeResult(int result, String userId)
```

参数:

参数	类型	描述
result	int	0表示取消订阅成功, 非0表示失败。
userId	String	用户ID, 从App server分配的唯一标示符。

- **onNetworkQualityChanged**: 网络状况变化时回调。

建议在网络质量较差的时候可以在界面上提示用户。

```
void onNetworkQualityChanged(AliRtcNetworkQuality quality)
```

参数:

参数	类型	描述
quality	<i>AliRtcNetworkQuality</i>	网络状况。

- **onOccurWarning**: warning回调。

目前warnig回调可以忽略, 不会影响正常使用。

```
void onOccurWarning(int warn)
```

参数:

参数	类型	描述
warn	int	警告类型。

- **onOccurError**: error回调。

当回调的中参数error为0x0102020C或0x02010105时, 需要先leaveChannel, 再重新创建SDK实例, 然后重新joinChannel。

```
void onOccurError(int error)
```

参数:

参数	类型	描述
error	int	错误类型。

- **onSubscribeChangedNotify**: 订阅结果回调。

```
void onSubscribeChangedNotify(String uid, AliRtcAudioTrack
audioTrack, AliRtcVideoTrack videoTrack)
```

参数:

参数	类型	描述
uid	String	远端用户ID。
audioTrack	AliRtcAudioTrack	订阅成功的音频流。
videoTrack	AliRtcVideoTrack	订阅成功的视频流。

- **onBye**: 被服务器踢出或者频道关闭时回调。

```
void onBye(int code)
```

参数:

参数	类型	描述
code	int	消息类型。 <ul style="list-style-type: none"> - 1: 被服务器踢出。 - 2: 频道关闭。 - 3: 同一个userId在其他端登录, 被服务器踢出。

- **onFirstPacketSent**: 首帧数据发送成功回调。

```
public void onFirstPacketSent(String callId, String streamLabel,
String trackLabel, int timeCost) {
//首帧数据发送成功
}
```

参数:

参数	类型	描述
callId	String	远端callid。
streamLabel	String	stream标签。
trackLabel	String	流的标签。
timeCost	int	耗时 (ms) 。

- **onFirstPacketReceived**: 首包数据接收成功。

```
void onFirstPacketReceived(String callId, String streamLabel, String trackLabel, int timeCost)
```

参数:

参数	类型	描述
callId	String	远端callId。
streamLabel	String	stream标签。
trackLabel	String	流的标签。
timeCost	int	耗时 (单位ms)。

- **onConnectionLost**: 网络连接断开回调。

```
void onConnectionLost()
```

- **onTryToReconnect**: 重新尝试网络连接回调。

```
void onTryToReconnect()
```

- **onConnectionRecovery**: 网络连接恢复回调。

```
void onConnectionRecovery()
```

- **onParticipantStatusNotify**: 远端用户状态改变通知。

```
void onParticipantStatusNotify(AliStatusInfo[] status_info_list, int count)
```

参数:

参数	类型	描述
status_info_list	<i>AliStatusInfo</i> []	用户状态数组。
count	int	数组长度。

- **onRemoteUserUnPublish**: 当远端用户停止发布时回调。

```
void onRemoteUserUnPublish(AliRtcEngine rtcEngine, String userId)
```

参数:

参数	类型	描述
rtcEngine	AliRtcEngine	AliRtcEngine实例。
userId	String	远端用户ID。

- **onRemoteUserOnLineNotify**: 远端用户上线回调。

```
void onRemoteUserOnLineNotify(String uid)
```

参数:

参数	类型	描述
uid	String	远端用户ID。

- **onRemoteUserOffLineNotify**: 远端用户下线回调。

```
void onRemoteUserOffLineNotify(String uid)
```

参数:

参数	类型	描述
uid	String	远端用户ID。

- **onRemoteTrackAvailableNotify**: 远端用户音视频流发生变化时回调。

```
void onRemoteTrackAvailableNotify(String uid, AliRtcAudioTrack audioTrack, AliRtcVideoTrack videoTrack)
```

参数:

参数	类型	描述
uid	String	远端用户ID。
audioTrack	<i>AliRtcAudioTrack</i>	远端用户发生变化后的音频流。
videoTrack	<i>AliRtcVideoTrack</i>	远端用户发生变化后的视频流。

2.3 AliRtcEngine接口

本文档为您介绍Android SDK的AliRtcEngine接口详情。

目录

基础接口

API	描述	以上版本支持
<i>setH5CompatibleMode</i>	设置H5兼容模式	1.1
<i>getH5CompatibleMode</i>	检查当前是否兼容H5	1.1

API	描述	以上版本支持
<i>getInstance</i>	创建AliRTCEngine实例（同一时间只会存在一个实例），只能在主线程调用	1.1
<i>setRtcEngineEventListener</i>	设置本地用户行为的回调事件的监听	1.1
<i>setRtcEngineNotify</i>	设置远端用户行为的通知事件的监听	1.1
<i>destroy</i>	销毁引擎	1.1

频道相关接口

API	描述	以上版本支持
<i>setAutoPublish</i>	设置是否自动发布，是否自动订阅	1.1
<i>joinChannel</i>	加入频道	1.1
<i>leaveChannel</i>	离开频道	1.1
<i>isInCall</i>	检查当前是否在频道中	1.1

发布相关接口

API	描述	以上版本支持
<i>isAutoPublish</i>	查询当前是否为自动发布模式	1.1
<i>configLocalCameraPublish</i>	设置是否允许发布相机流	1.1
<i>isLocalCameraPublishEnabled</i>	查询当前是否允许发布相机流	1.1
<i>configLocalScreenPublish</i>	设置是否允许发布屏幕流	1.1
<i>isLocalScreenPublishEnabled</i>	查询当前是否允许发布屏幕流	1.1
<i>configLocalAudioPublish</i>	设置是否允许发布音频流	1.1
<i>isLocalAudioPublishEnabled</i>	查询当前是否允许发布音频流	1.1
<i>configLocalSimulcast</i>	设置是否允许发布次要视频流	1.1
<i>isLocalSimulcastEnabled</i>	查询当前是否允许发布次要视频流	1.1
<i>publish</i>	手动发布视频和音频流	1.1

订阅相关接口

API	描述	以上版本支持
<i>isAutoSubscribe</i>	查询当前是否为自动订阅模式	1.1
<i>configRemoteCameraTrack</i>	设置是否订阅远端相机流	1.1
<i>configRemoteScreenTrack</i>	设置是否订阅远端屏幕流	1.1
<i>configRemoteAudio</i>	设置是否订阅远端音频流	1.1
<i>subscribe</i>	手动订阅视频和音频流	1.1

视频相关接口

API	描述	以上版本支持
<i>setVideoProfile</i>	设置视频流参数	1.1
<i>getVideoProfile</i>	查询当前视频流参数	1.1
<i>setLocalViewConfig</i>	为本地预览设置渲染窗口以及绘制参数	1.1
<i>muteLocalCamera</i>	设置是否停止发布本地视频流	1.1
<i>setRemoteViewConfig</i>	为远端的视频设置渲染窗口以及绘制参数	1.1
<i>switchCamera</i>	切换前后摄像头	1.1
<i>getCurrentCameraType</i>	获取当前摄像头类型	1.1
<i>setPreCameraType</i>	预设值摄像头方向	1.1
<i>getPreCameraType</i>	获取预设值摄像头方向	1.1
<i>setCameraZoom</i>	设置摄像头参数	1.1
<i>isCameraOn</i>	检查摄像头是否打开	1.1
<i>isCameraSupportExposurePoint</i>	相机是否支持手动曝光	1.14
<i>isCameraSupportFocusPoint</i>	相机是否支持手动聚焦	1.14
<i>setCameraExposurePoint</i>	设置手动曝光的坐标点	1.14
<i>setCameraFocusPoint</i>	设置手动聚焦的坐标点	1.14
<i>isCameraFlash</i>	是否开启了闪光灯	1.14
<i>getCameraZoom</i>	获取相机的zoom值	1.14
<i>RegisterPreprocessVideoObserver</i>	注册人脸识别预处理	1.14
<i>RegisterVideoObserver</i>	注册视频观察者	1.14
<i>UnRegisterVideoObserver</i>	注销视频观察者	1.14

音频相关接口

API	描述	以上版本支持
<i>setAudioOnlyMode</i>	设置纯音频模式还是音视频模式	1.1
<i>isAudioOnly</i>	查询当前是否为纯音频模式	1.1
<i>muteLocalMic</i>	设置是否停止发布本地音频	1.1
<i>muteRemoteAudioPlaying</i>	设置是否停止播放远端音频流	1.1
<i>enableSpeakerphone</i>	切换听筒、扬声器输出	1.1
<i>isSpeakerOn</i>	查询是否开启扬声器	1.1
<i>startAudioCapture</i>	开启音频采集	1.11
<i>stopAudioCapture</i>	关闭音频采集	1.11
<i>startAudioPlayer</i>	开启音频播放	1.11
<i>stopAudioPlayer</i>	关闭音频播放	1.11

预览接口

API	描述	以上版本支持
<i>startPreview</i>	开始本地预览	1.1
<i>stopPreview</i>	停止本地预览	1.1

远端用户查询接口

API	描述	以上版本支持
<i>getOnlineRemoteUsers</i>	获取远端在线用户列表	1.1
<i>getUserInfo</i>	查询远端用户信息	1.1
<i>isUserOnline</i>	判断用户是否在线	1.1
<i>getMediaInfoWithUserId</i>	获取媒体流信息	1.9

其他接口

API	描述	以上版本支持
<i>setLogLevel</i>	设置log级别	1.1
<i>getSdkVersion</i>	获取SDK版本号	1.1

接口详情

- **setH5CompatibleMode**: 设置是否兼容H5, 当前版本不支持在创建AliRtcEngine实例之后更改H5兼容模式, 必须在创建实例之前就调用此方法, 默认不兼容H5。

```
AliRtcEngine.setH5CompatibleMode(int enable)
```

参数:

参数	类型	说明
enable	int	0表示不兼容H5, 1表示兼容H5。

- **getH5CompatibleMode**: 检查当前是否兼容H5, 返回1表示兼容, 0表示不兼容。

```
public static int getH5CompatibleMode()
```

- **getInstance**: 创建AliRTCEngine实例 (同一时间只会存在一个实例), 只能在主线程调用。

```
public static AliRtcEngineImpl getInstance(Context context)
```

参数:

参数	类型	说明
context	Context	上下文。

- **setRtcEngineEventListener**: 设置本地用户行为的回调事件的监听。

```
public abstract void setRtcEngineEventListener(AliRtcEngineEventListener listener)
```

参数:

参数	类型	说明
listener	AliRtcEngineEventListener	接收回调事件的监听器。

- **setRtcEngineNotify**: 设置远端用户行为的通知事件的监听。

```
public abstract void setRtcEngineNotify(AliRtcEngineNotify engineNotify)
```

参数:

参数	类型	说明
engineNotify	AliRtcEngineNotify	接收通知的监听器。

- **destroy**: 销毁引擎, 和leaveChannel中两者调用一个即可。

离开频道时, AliRtcEngine实例会被销毁, 如需继续joinChannel等操作, 需要先重新调用getInstance初始化AliRtcEngine实例。

```
public abstract void destroy();
```

- **setAutoPublish**: 设置是否自动发布, 是否自动订阅。默认是自动发布和订阅, 必须在joinChannel之前设置。

```
public int setAutoPublish(boolean autoPub, boolean autoSub);
```

参数:

参数	类型	说明
autoPub	boolean	true表示自动发布; false表示手动发布。
autoSub	boolean	true表示自动订阅; false表示手动订阅。

- **joinChannel**: 加入频道。

加入频道成功后, 如果中途需要加入其他频道, 必须先调用leaveChannel离开当前频道, 如果加入频道失败, 需要重试时, 无需先调用leaveChannel。

```
public abstract void joinChannel(AliRtcAuthInfo authInfo, String userName)
```

参数:

参数	类型	说明
authInfo	<i>AliRtcAuthInfo</i>	鉴权信息, 从App Server下发, APP Server可通过API获取。
userName	String	用户的显示名称, 不是uid。

- **leaveChannel**: 离开频道。

离开频道时, `AliRtcEngine`实例会被销毁, 如需继续`joinChannel`等操作, 需要先重新调用`getInstance`初始化`AliRtcEngine`实例。

- 对于版本号大于1.7的sdk, 请调用如下接口。

```
public abstract void leaveChannel()
```

- 对于版本号小于等于1.7的sdk, 请增加`timeout`参数, 一般建议设置为1000, 表示该接口的调用超时时间为1秒。

```
public abstract void leaveChannel(long timeout)
```

- **isInCall**: 检查当前是否在频道中, 返回`true`表示在频道中, `false`表示不在频道中。

```
public abstract boolean isInCall()
```

- **isAutoPublish**: 查询当前是否为自动发布模式, 返回`true`为自动发布, `false`为手动发布。

```
public abstract boolean isAutoPublish()
```

- **configLocalCameraPublish**: 设置是否允许发布相机流。默认为允许发布相机流, 手动发布时, 需要调用`publish`才能生效。

```
public abstract void configLocalCameraPublish(boolean enable)
```

参数:

参数	类型	说明
<code>enable</code>	<code>boolean</code>	<code>true</code> 为允许发布相机流; <code>false</code> 表示不允许。

- **isLocalCameraPublishEnabled**: 查询当前是否允许发布相机流，返回true为允许，false为不允许。

```
public abstract boolean isLocalCameraPublishEnabled()
```

- **configLocalScreenPublish**: 设置是否允许发布屏幕流。默认为不允许发布屏幕流，手动发布时，需要调用publish才能生效。

```
public abstract void configLocalScreenPublish(boolean enable)
```

参数:

参数	类型	说明
enable	boolean	true表示允许发布屏幕流；false表示不允许。

- **isLocalScreenPublishEnabled**: 查询当前是否允许发布屏幕流，返回true为允许，false为不允许。

```
public abstract boolean isLocalScreenPublishEnabled()
```

- **configLocalAudioPublish**: 设置是否允许发布音频流。默认为允许发布音频流，手动发布时，需要调用publish才能生效。

```
public abstract void configLocalAudioPublish(boolean enable)
```

参数:

参数	类型	说明
enable	boolean	true表示允许；false表示不允许。

- **isLocalAudioPublishEnabled**: 查询当前是否允许发布音频流, 返回true为允许, false为不允许。

```
public abstract boolean isLocalAudioPublishEnabled()
```

- **configLocalSimulcast**: 设置是否允许发布次要视频流。默认为允许发布次要视频流, 手动发布时, 需要调用publish才能生效。

```
public abstract int configLocalSimulcast(boolean enable, AliRtcVideoTrack track)
```

参数:

参数	类型	说明
enable	boolean	true 表示允许发布次要流, false 表示不允许。
track	<i>AliRtcVideoTrack</i>	流类型, 当前只支持相机流: AliVideoTrackCamera 。

- **isLocalSimulcastEnabled**: 查询当前是否允许发布次要视频流，返回true为允许，false为不允许。

```
public abstract boolean isLocalSimulcastEnabled()
```

- **publish**: 手动发布视频和音频流。
 - 调用publish的实际表现需要结合configLocalCameraPublish、configLocalScreenPublish、configLocalAudioPublish、configLocalSimulcast等接口才能确定。
 - 根据您的具体业务需求配置上述4个接口的参数，以发布相应的视频和音频流。
 - 发布和停止发布都是调用publish。
 - 如需停止发布，则需要上述4个配置接口的参数都置为false，再调用publish。

```
public abstract void publish()
```

- **isAutoSubscribe**: 查询当前是否为自动订阅模式，返回true为自动订阅，false为手动订阅。

```
public abstract boolean isAutoSubscribe()
```

- **configRemoteCameraTrack**: 设置是否订阅远端相机流。默认为订阅大流，手动订阅时，需要调用subscribe才能生效。

```
public abstract void configRemoteCameraTrack(String uid, boolean master, boolean enable)
```

参数:

参数	类型	说明
uid	String	用户ID，从App server分配的唯一标示符。
master	boolean	是否优先订阅大流。true为订阅大流；false为订阅次小流。
enable	boolean	true为订阅远端相机流，false为停止订阅远端相机流。

- **configRemoteScreenTrack**: 设置是否订阅远端屏幕流。默认为不订阅远端屏幕流，手动订阅时，需要调用**subscribe**才能生效。

```
public abstract void configRemoteScreenTrack(String uid, boolean enable)
```

参数:

参数	类型	说明
uid	String	用户ID，从App server分配的唯一标示符。
enable	boolean	true为订阅远端屏幕流，false为停止订阅远端屏幕流。

- **configRemoteAudio**: 设置是否订阅远端音频流。默认为订阅远端音频流，手动订阅时，需要调用**subscribe**才能生效。

```
public abstract void configRemoteAudio(String uid, boolean enable)
```

参数:

参数	类型	说明
uid	String	用户ID，从App server分配的唯一标示符。
enable	boolean	true为订阅远端音频流，false为停止订阅远端音频流。

- **subscribe**: 手动订阅视频和音频流。返回为0时说明接口执行正常, 但是否订阅成功还得看回调结果; 返回为非0时, 说明接口执行异常中断, 订阅失败。
 - 调用subscribe的实际表现需要结合configRemoteCameraTrack、configRemoteScreenTrack、configRemoteAudio等接口才能确定。
 - 根据您的具体业务需求配置上述3个接口的参数, 以订阅相应的视频和音频流。
 - 订阅和停止订阅都是调用subscribe。
 - 如需停止订阅, 则需要上述3个配置接口的参数都置为false, 再调用subscribe。

```
public abstract int subscribe(String uid)
```

参数:

参数	类型	说明
uid	String	用户ID, 从App server分配的唯一标示符。

- **setVideoProfile**: 设置视频流参数。

```
public abstract void setVideoProfile(AliRtcVideoProfile profile,
AliRtcVideoTrack track)
```

参数:

参数	类型	说明
profile	<i>AliRtcVideoProfile</i>	视频流参数。
track	<i>AliRtcVideoTrack</i>	需要设置的视频流类型。

- **getVideoProfile**: 查询当前视频流参数。

```
public abstract AliRtcVideoProfile getVideoProfile(AliRtcVideoTrack
track)
```

参数:

参数	类型	说明
track	<i>AliRtcVideoTrack</i>	需要查询的视频流类型。

返回值:

返回值	描述
<i>AliRtcVideoProfile</i>	视频流规格。

- **setLocalViewConfig**: 为本地预览设置渲染窗口以及绘制参数。
 - 支持joinChannel之前和之后切换窗口。如果viewConfig为NULL或者其成员渲染视图为NULL, 则停止渲染。
 - 如果在播放过程中需要重新设置渲染方式, 请保持viewConfig中其他成员变量不变, 仅修改renderMode。
 - viewConfig中渲染方式默认为AliRtcRenderModeAuto。

```
public abstract int setLocalViewConfig(AliVideoCanvas viewConfig,
AliRtcVideoTrack track)
```

参数:

参数	类型	说明
viewConfig	<i>AliVideoCanvas</i>	渲染参数, 包含渲染窗口以及渲染方式。
track	<i>AliRtcVideoTrack</i>	预览只允许AliVideoTrackCamera。

- **muteLocalCamera**: 设置是否停止发布本地视频流。不改变当前视频流的采集状态。

```
public abstract int muteLocalCamera(boolean mute, AliRtcVideoTrack
track)
```

参数:

参数	类型	说明
mute	boolean	true表示停止发布视频流; false表示恢复发布。
track	<i>AliRtcVideoTrack</i>	需要改变发布状态的videoTrack类型。

- **setRemoteViewConfig**: 为远端的视频设置渲染窗口以及绘制参数。
 - 支持joinChannel之前和之后切换窗口。如果canvas为NULL或者其成员渲染视图为NULL，则停止渲染相应的流。
 - 如果在播放过程中需要重新设置渲染方式，请保持canvas中其他成员变量不变，仅修改renderMode。
 - canvas中渲染方式默认为AliRtcRenderModeAuto。

```
public abstract int setRemoteViewConfig(AliVideoCanvas canvas,
String uid, AliRtcVideoTrack track)
```

参数:

参数	类型	说明
canvas	<i>AliVideoCanvas</i>	渲染参数，包含渲染窗口以及渲染方式。
uid	String	用户ID，从App server分配的唯一标示符。
track	<i>AliRtcVideoTrack</i>	需要设置的videoTrack类型。

- **switchCamera**: 切换前后摄像头，返回0为切换成功，其他为切换失败。

```
public abstract int switchCamera()
```

- **getCurrentCameraType**: 获取当前摄像头类型，返回摄像头的类型*AliRTCCameraType*。

```
public abstract AliRTCCameraType getCurrentCameraType()
```

- **setPreCameraType**: 预设值摄像头方向。

```
public abstract void setPreCameraType(int faceTo)
```

参数:

参数	类型	说明
faceTo	int	0表示后置，1表示前置。

- **getPreCameraType**: 获取预设值摄像头方向。返回0为后置摄像头，1为前置摄像头。

```
public abstract int getPreCameraType()
```

- **setCameraZoom**: 设置摄像头参数。返回0表示设置成功，其他表示设置失败。

```
public abstract int setCameraZoom(float zoom, boolean flash, boolean autoFocus)
```

参数:

参数	类型	说明
zoom	float	zoom变焦的级别。
flash	boolean	是否打开闪光灯。
autoFocus	boolean	是否打开自动对焦。

- **isCameraOn**: 检查摄像头是否打开。返回true表示摄像头已打开，false表示摄像头未打开。

```
public abstract boolean isCameraOn()
```

- **isCameraSupportExposurePoint**: 查看摄像头是否支持手动曝光，返回true表示支持，否则返回false。

```
public abstract boolean isCameraSupportExposurePoint();
```

- **isCameraSupportFocusPoint**: 查看是否支持手动聚焦，返回true表示支持，否则返回false。

```
public abstract boolean isCameraSupportFocusPoint();
```

- **setCameraExposurePoint**: 设置手动曝光点。返回0表示设置成功，其他表示设置失败。

```
public abstract int setCameraExposurePoint(float x, float y);
```

参数:

参数	类型	描述
x	float	x坐标。
y	float	y坐标。

- **setCameraFocusPoint**: 设置手动聚焦点, 返回0表示设置成功, 非0表示设置失败

```
public abstract int setCameraFocusPoint(float x, float y);
```

参数:

参数	类型	描述
x	float	x坐标。
y	float	y坐标。

- **isCameraFlash()**: 查看摄像头闪光灯是否开启, 返回true表示开启, 否则返回false。

```
public abstract boolean isCameraFlash();
```

- **getCameraZoom()**: 获取相机zoom值, 返回值范围: 1~相机支持的最大值。

```
public abstract float getCameraZoom();
```

- **RegisterPreprocessVideoObserver**: 注册人脸识别预处理。

```
public abstract void RegisterPreprocessVideoObserver(AliDetectObserver observer);
```

参数:

参数	类型	描述
observer	<i>AliDetectObserver</i>	人脸识别预处理。

- **RegisterVideoObserver**: 注册视频观察者。

```
public abstract void RegisterVideoObserver(String userId, AliRtcVideoTrack track, final AliVideoObserver observer);
```

参数:

参数	类型	描述
userId	String	用户id。
track	<i>AliRtcVideoTrack</i>	视频track的类型。
observer	AliVideoObserver	视频观察者。

- **UnRegisterVideoObserver**: 注销视频观察者。

```
public abstract void UnRegisterVideoObserver(String userId, AliRtcVideoTrack track);
```

参数:

参数	类型	描述
----	----	----

userId	String	用户id。
track	<i>AliRtcVideoTrack</i>	视频track的类型。

- **setAudioOnlyMode**: 设置纯音频模式还是音视频模式。返回0代表设置成功，其他代表设置失败。默认为音视频模式（非纯音频），必须在joinChannel之前设置。

```
public abstract int setAudioOnlyMode(boolean audioOnly)
```

参数:

参数	类型	说明
audioOnly	boolean	true表示只有音频发布和订阅；false表示音视频都支持。

- **isAudioOnly**: 查询当前是否为纯音频模式，返回true为纯音频，false为音视频。

```
public abstract boolean isAudioOnly()
```

- **muteLocalMic**: 设置是否停止发布本地音频。返回0表示设置成功，-1表示设置失败。不改变当前音频的采集状态。

```
public abstract int muteLocalMic(boolean mute)
```

参数:

参数	类型	说明
mute	boolean	true表示停止发布本地音频；false表示恢复发布。

- **muteRemoteAudioPlaying**: 设置是否停止播放远端音频流，返回0表示设置成功，-1表示设置失败。

```
public abstract int muteRemoteAudioPlaying(String uid, boolean mute)
```

参数:

参数	类型	说明
uid	String	用户ID，从App server分配的唯一标示符。
mute	boolean	true表示停止播放；false表示恢复播放。

- **enableSpeakerphone**: 切换听筒、扬声器输出。

```
public abstract int enableSpeakerphone(boolean enable)
```

参数:

参数	类型	说明
enable	boolean	true 为扬声器模式; false 为听筒模式。

- **isSpeakerOn**: 查询是否开启扬声器。返回true表示已开启扬声器，false表示未开启扬声器。

```
public abstract boolean isSpeakerOn()
```

- **startAudioCapture**: 开启音频采集。您可以控制提前打开音频采集，如果不设置，SDK会在开始推流的时候打开音频采集。

```
public int startAudioCapture();
```

- **stopAudioCapture**: 关闭音频采集。您可以控制关闭音频采集。

```
public int stopAudioCapture();
```

- **startAudioPlayer**: 开启音频播放。您可以控制提前打开音频播放，如果不设置，SDK会在订阅成功的时候打开音频播放。

```
public int startAudioPlayer();
```

- **stopAudioPlayer**: 关闭音频播放。您可以控制关闭音频播放。

```
public int stopAudioPlayer();
```

- **startPreview**: 开始本地预览。可以在joinChannel之前就开启预览。

```
public abstract int startPreview()
```

- **stopPreview**: 停止本地预览。

```
public abstract int stopPreview()
```

- **getOnlineRemoteUsers**: 获取远端在线用户列表，返回用户ID列表。

```
public abstract String[] getOnlineRemoteUsers()
```

- **getUserInfo**: 查询远端用户信息。

```
public abstract AliRtcRemoteUserInfo getUserInfo(String uid)
```

参数:

参数	类型	说明
uid	String	用户ID，从App server分配的唯一标示符。

返回值:

返回值	描述
AliRtcRemoteUserInfo	远程用户信息。

- **isUserOnline**: 查询用户是否在线, 返回true表示在线, false表示不在线。

```
public abstract boolean isUserOnline(String uid)
```

参数:

参数	类型	说明
uid	String	用户ID, 从App server分配的唯一标示符。

- **getMediaInfoWithUserId**: 获取媒体流信息。

```
public String getMediaInfoWithUserId(String userId, AliRtcVideoTrack track, String[] keys);
```

参数:

参数	类型	说明
userId	String	用户的userId。
track	<i>AliRtcVideoTrack</i>	需要查询的媒体流类型。
keys	String[]	查询key值数组。

- **setLogLevel**: 设置log级别。

```
public abstract void setLogLevel(AliRtcLogLevel logLevel)
```

参数:

参数	类型	说明
logLevel	<i>AliRtcLogLevel</i>	log级别。

- **getSdkVersion**: 获取SDK版本号。

```
public abstract String getSdkVersion()
```

3 iOS和Mac SDK

3.1 数据类型

本文档为您列出了iOS SDK和Mac SDK的所有数据类型。

数据类型	描述	以上版本支持
AliRtcVideoTrack	视频track类型	1.1
AliRtcAudioTrack	音频track类型	1.1
AliRtcNetworkQuality	网络质量	1.1
AliRtcRenderMode	渲染模式	1.1
AliRtcLogLevel	log级别	1.1
AliRtcVideoProfile	视频质量参数	1.1
AliRtcErrorCode	错误类型	1.9
AliRtcAuthInfo	鉴权信息	1.1
AliVideoCanvas	渲染画布	1.1
AliRtcRenderMirrorMode	镜像模式	1.11
AliRtcOnByeType	OnBye类型	1.11
AliRTCImageFormat	图片格式枚举（仅iOS）	1.14

- **AliRtcVideoTrack**：视频track的类型。

枚举名	描述
AliRtcVideoTrackNo	无视频流
AliRtcVideoTrackCamera	相机流
AliRtcVideoTrackScreen	屏幕共享流
AliRtcVideoTrackBoth	相机流和屏幕共享流

- **AliRtcAudioTrack**：音频track的类型。

枚举名	描述
AliRtcAudioTrackNo	无音频流
AliRtcAudioTrackMic	麦克风流

- **AliRtcNetworkQuality**: 网络质量。

枚举名	描述
AliRtcNetworkQualityExcellent	网络极好，流程度清晰度质量好
AliRtcNetworkQualityGood	网络好，流畅度清晰度和极好差不多
AliRtcNetworkQualityPoor	网络较差，音视频流畅度清晰度有瑕疵，不影响沟通
AliRtcNetworkQualityBad	网络差，视频卡顿严重，音频能正常沟通
AliRtcNetworkQualityVeryBad	网络极差，基本无法沟通
AliRtcNetworkQualityDisconnect	网络中断
AliRtcNetworkQualityUnknow	未知

- **AliRtcRenderMode**: 渲染模式。

枚举名	描述
AliRtcRenderModeAuto	自动模式
AliRtcRenderModeStretch	拉伸填充视图，不保持视频比例
AliRtcRenderModeFill	在保持视频宽高比的同时缩放，填充黑边
AliRtcRenderModeCrop	在保持视频宽高比的同时缩放，并裁剪以适合视图

- **AliRtcLogLevel**: log级别。

枚举名	描述
AliRtcLogLevelDump	全量
AliRtcLogLevelDebug	调试
AliRtcLogLevelVerbose	详细
AliRtcLogLevelInfo	普通
AliRtcLogLevelWarn	警告
AliRtcLogLevelError	错误
AliRtcLogLevelFatal	严重
AliRtcLogLevelNone	无

- **AliRtcVideoProfile**: 视频流规格。

发布视频流规格，默认为AliRTCSDK_Video_Profile_Default，可以在joinChannel之前或者之后设置，但必须在发布之前通过setVideoProfile方法设置。

枚举名	描述
AliRTCSDK_Video_Profile_Default	默认，分辨率480 * 640，帧率15
AliRTCSDK_Video_Profile_180_320P_15	分辨率180 * 320，帧率15
AliRTCSDK_Video_Profile_180_320P_30	分辨率180 * 320，帧率30
AliRTCSDK_Video_Profile_360_640P_15	分辨率360 * 640，帧率15
AliRTCSDK_Video_Profile_360_640P_30	分辨率360 * 640，帧率30
AliRTCSDK_Video_Profile_480_640P_15	分辨率480 * 640，帧率15
AliRTCSDK_Video_Profile_480_640P_30	分辨率480 * 640，帧率30
AliRTCSDK_Video_Profile_720_1280P_15	分辨率720 * 1280，帧率15
AliRTCSDK_Video_Profile_720_1280P_30	分辨率720 * 1280，帧率30
AliRTCSDK_Video_Profile_Max	占位值

- **AliRtcErrorCode**: 错误类型。

枚举名	描述
AliRtcErrorCodeNone	无
AliRtcErrorCodeHeartbeatTimeout	心跳超时
AliRtcErrorCodePollingError	信令错误

- **AliRtcAuthInfo**: 鉴权信息。



注意:

在1.9及以上版本移出session id字段。

枚举名	类型	描述
channel	NSString *	频道号

枚举名	类型	描述
user_id	NSString *	用户ID
appid	NSString *	应用ID
nonce	NSString *	随机串
token	NSString *	令牌
gslb	NSArray	gslb服务器地址
timestamp	long long	时间戳

- AliVideoCanvas: 渲染画布。

枚举名	类型	描述
view	AliRenderView *	渲染视图, 继承UIView类, 创建视图前需创建view
renderMode	AliRtcRenderMode	渲染模式, 默认为AliRtcRenderModeAuto
mirrorMode	AliRtcRenderMirrorMode	镜像模式, 默认为前置镜像

- AliRtcRenderMirrorMode: 镜像模式。

枚举名	描述
AliRtcRenderMirrorModeOnlyFrontCameraPreviewEnabled	只有前置摄像头预览镜像, 其余不镜像
AliRtcRenderMirrorModeAllEnabled	全部镜像
AliRtcRenderMirrorModeAllDisabled	全部不镜像

- AliRtcOnByeType: OnBye类型 (被服务器踢出频道的消息)。

枚举名	描述
AliRtcOnByeBeKickedOut	当前用户被踢出频道
AliRtcOnByeChannelDestroy	频道已结束, 需要离开会议
AliRtcOnByeUserReplaced	相同userID在其他设备加入频道, 当前设备被下线

- **ALIRTCImageFormat (仅iOS) : 图片格式枚举。**

枚举名	描述
ALIRTC_IMAGE_FORMAT_UNKNOWN	格式未知
ALIRTC_IMAGE_FORMAT_ARGB	格式ARGB
ALIRTC_IMAGE_FORMAT_BGRA	格式BGRA
ALIRTC_IMAGE_FORMAT_RGBA	格式RGBA
ALIRTC_IMAGE_FORMAT_YUV420P	格式YUV420P
ALIRTC_IMAGE_FORMAT_YUVV12	格式YUVV12
ALIRTC_IMAGE_FORMAT_YUVNV21	格式YUVNV21
ALIRTC_IMAGE_FORMAT_YUVNV12	格式YUVNV12
ALIRTC_IMAGE_FORMAT_YUVNV12_FULL_RANGE	格式YUVNV12_FULL_RANGE
ALIRTC_IMAGE_FORMAT_YUVJ420P	格式YUVJ420P
ALIRTC_IMAGE_FORMAT_YUV420SP	格式YUV420SP
ALIRTC_IMAGE_FORMAT_YUVJ420SP	格式YUVJ420SP
ALIRTC_IMAGE_FORMAT_YUVJ444P	格式YUVJ444P
ALIRTC_IMAGE_FORMAT_YUVJ422P	格式YUVJ422P
ALIRTC_IMAGE_FORMAT_YUV444P	格式YUV444P
ALIRTC_IMAGE_FORMAT_YUV2	格式YUV2
ALIRTC_IMAGE_FORMAT_MJPEG	格式MJPEG
ALIRTC_IMAGE_FORMAT_TEXTURE_2D	格式TEXTURE_2D
ALIRTC_IMAGE_FORMAT_TEXTURE_OES	格式TEXTURE_OES

3.2 回调及监听

本文档为您介绍iOS SDK和Mac SDK回调及监听的接口详情。

目录

AliRtcEngineDelegate回调如下所示。



说明:

回调接口都在子线程。

API	描述	以上版本支持
<i>onRemoteUserOnLineNotify</i>	远端用户上线回调	1.1
<i>onRemoteUserOffLineNotify</i>	远端用户下线回调	1.1
<i>onRemoteTrackAvailableNotify</i>	远端用户音视频流发生变化时回调	1.1
<i>onSubscribeChangedNotify</i>	订阅结果回调	1.1
<i>onNetworkQualityChanged</i>	网络质量变化回调	1.4
<i>onOccurWarning</i>	warning回调	1.1
<i>onOccurError</i>	error回调	1.1
<i>onBye</i>	被服务器踢出或者频道关闭时回调	1.1
<i>onFirstPacketSentWithAudioTrack</i>	首帧数据发送成功	1.11
<i>onFirstRemoteVideoFrameDrawn</i>	首帧渲染完回调	1.1
<i>onFirstPacketReceivedWithAudioTrack</i>	首包数据接收成功	1.13
<i>onVideoDetectCallback</i>	RTC采集视频数据回调（仅iOS）	1.14
<i>onUserAudioMuted</i>	用户muteAudio通知（仅iOS）	1.14
<i>onUserVideoMuted</i>	用户muteVideo通知（仅iOS）	1.14
<i>onUserAudioInterruptedBegin</i>	用户audio被中断通知（仅iOS）	1.14
<i>onUserAudioInterruptedEnded</i>	用户audio中断结束通知（仅iOS）	1.14
<i>onConnectionLost</i>	网络连接断开（仅iOS）	1.14

API	描述	以上版本支持
<i>onTryToReconnect</i>	网络连接正在尝试重连中（仅iOS）	1.14
<i>onConnectionRecovery</i>	网络连接重连成功（仅iOS）	1.14

接口详情

- **onRemoteUserOnLineNotify**: 远端用户上线回调。

```
- (void)onRemoteUserOnLineNotify:(NSString *)uid;
```

参数:

参数	类型	描述
uid	NSString *	远端用户ID。

- **onRemoteUserOffLineNotify**: 远端用户下线回调。

```
- (void)onRemoteUserOffLineNotify:(NSString *)uid;
```

参数:

参数	类型	描述
uid	NSString *	远端用户ID。

- **onRemoteTrackAvailableNotify**: 远端用户音视频流发生变化时回调。

```
- (void)onRemoteTrackAvailableNotify:(NSString *)uid audioTrack:(AliRtcAudioTrack)audioTrack videoTrack:(AliRtcVideoTrack)videoTrack;
```

参数:

参数	类型	描述
uid	NSString *	远端用户ID。
audioTrack	<i>AliRtcAudioTrack</i>	远端用户发生变化后的音频流。
videoTrack	<i>AliRtcVideoTrack</i>	远端用户发生变化后的视频流。

- **onSubscribeChangedNotify**: 订阅结果回调。

```
- (void)onSubscribeChangedNotify:(NSString *)uid audioTrack:(AliRtcAudioTrack)audioTrack videoTrack:(AliRtcVideoTrack)videoTrack;
```

参数:

参数	类型	描述
uid	NSString *	远端用户ID。
audioTrack	AliRtcAudioTrack	订阅成功的音频流。
videoTrack	AliRtcVideoTrack	订阅成功的视频流。

- **onNetworkQualityChanged**: 网络质量变化时回调，建议在网络质量较差的时候可以在界面上提示用户。

```
- (void)onNetworkQualityChanged:(NSString *)uid
    upNetworkQuality:(AliRtcNetworkQuality)upQuality
    downNetworkQuality:(AliRtcNetworkQuality)downQuality;
```

参数:

参数	类型	描述
uid	NSString *	网络质量发生变化的用户。
upQuality	AliRtcNetworkQuality	上行网络质量。
downQuality	AliRtcNetworkQuality	下行网络质量。

- **onOccurWarning**: Warning回调。目前Warning回调可以忽略，不会影响正常使用。

```
- (void)onOccurWarning:(int)warn;
```

参数:

参数	类型	描述
warn	int	警告类型。

- **onOccurError**: Error回调。当回调的中参数error为0x0102020C或0x02010105时，需要先leaveChannel，再重新创建SDK实例，然后重新joinChannel。

```
- (void)onOccurError:(int)error;
```

参数:

参数	类型	描述
error	int	错误类型。

- **onBye**: 被服务器踢出或者频道关闭时回调。

```
- (void)onBye:(int)code;
```

参数:

参数	类型	描述
code	int	消息类型。 - 1: 被服务器踢出。 - 2: 频道关闭。 - 3: 同一个userId在其他端登录，被服务器踢出。

- **onFirstPacketSentWithAudioTrack**: 首帧数据发送成功。

```
- (void)onFirstPacketSentWithAudioTrack:(AliRtcAudioTrack)audioTrack  
videoTrack:(AliRtcVideoTrack)videoTrack;
```

参数:

参数	类型	描述
audioTrack	<i>AliRtcAudioTrack</i>	发送成功的音频流类型。
videoTrack	<i>AliRtcVideoTrack</i>	发送成功的视频流类型。

- **onFirstRemoteVideoFrameDrawn**: 远端视频流首帧渲染完回调。

```
- (void)onFirstRemoteVideoFrameDrawn:(NSString *)uid videoTrack:(  
AliRtcVideoTrack)videoTrack;
```

参数:

参数	类型	描述
uid	NSString *	远端用户ID。
videoTrack	<i>AliRtcVideoTrack</i>	渲染的流类型。

- **onFirstPacketReceivedWithAudioTrack**：首包数据接收成功。

```
- (void)onFirstPacketReceivedWithAudioTrack:(AliRtcAudioTrack)
audioTrack videoTrack:(AliRtcVideoTrack)videoTrack;
```

参数：

参数	类型	描述
audioTrack	<i>AliRtcAudioTrack</i>	发送成功的音频流类型。
videoTrack	<i>AliRtcVideoTrack</i>	发送成功的视频流类型。

- **onVideoDetectCallback**（仅iOS可用）：RTC采集视频数据回调。

```
- (long)onVideoDetectCallback:(AliRtcVideoSource)type videoFrame:(
AliRtcVideoDataSample *)videoFrame;
```

参数：

参数	类型	描述
type	<i>AliRtcVideoSource</i>	视频流类型。
videoFrame	<i>AliRtcVideoDataSample *</i>	视频数据帧。

- **onUserAudioMuted**（仅iOS可用）：用户muteAudio通知。

```
- (void)onUserAudioMuted:(NSString *)uid audioMuted:(BOOL)isMute;
```

参数：

参数	类型	描述
uid	<i>NSString *</i>	执行muteAudio的用户。
isMute	BOOL	YES ：静音； NO ：未静音。

- **onUserVideoMuted**（仅iOS可用）：用户muteVideo通知。

```
- (void)onUserVideoMuted:(NSString *)uid videoMuted:(BOOL)isMute;
```

参数：

参数	类型	描述
uid	<i>NSString *</i>	执行muteVideo的用户。
isMute	BOOL	YES ：推流黑帧； NO ：正常推流。

- **onUserAudioInterruptedBegin** (仅iOS可用) : 用户audio被中断通知 (一般用户打电话等音频被抢占场景)。

```
- (void)onUserAudioInterruptedBegin:(NSString *)uid;
```

参数:

参数	类型	描述
uid	NSString *	audio被中断的用户。

- **onUserAudioInterruptedEnded** (仅iOS可用) : 用户audio中断结束通知 (对应onUserAudioInterruptedBegin)。

```
- (void)onUserAudioInterruptedEnded:(NSString *)uid;
```

参数:

参数	类型	描述
uid	NSString *	audio中断结束的用户。

- **onConnectionLost** (仅iOS可用) : 网络连接断开。

```
- (void)onConnectionLost;
```

- **onTryToReconnect** (仅iOS可用) : 网络连接正在尝试重连中。

```
- (void)onTryToReconnect;
```

- **onConnectionRecovery** (仅iOS可用) : 网络连接重连成功。

```
- (void)onConnectionRecovery;
```

3.3 AliRtcEngine接口

本文档为您介绍iOS SDK和Mac SDK的AliRtcEngine接口详情。

目录

基础接口

API	描述	以上版本支持
<i>setH5CompatibleMode</i>	设置H5兼容模式。	1.1
<i>getH5CompatibleMode</i>	检查当前是否兼容H5。	1.1
<i>sharedInstance</i>	创建AliRtcEngine实例 (同一时间只会存在一个实例)。	1.1

API	描述	以上版本支持
<i>destroy</i>	销毁SDK。	1.1

频道相关接口

API	描述	以上版本支持
<i>setAutoPublish</i>	设置是否自动发布，是否自动订阅。	1.1
<i>joinChannel</i>	加入频道。	1.1
<i>leaveChannel</i>	离开频道。	1.1
<i>isInCall</i>	检查当前是否在频道中。	1.1

发布相关接口

API	描述	以上版本支持
<i>isAutoPublish</i>	查询当前是否为自动发布模式。	1.1
<i>configLocalCameraPublish</i>	设置是否允许发布相机流。	1.1
<i>isLocalCameraPublishEnabled</i>	查询当前是否允许发布相机流。	1.1
<i>configLocalScreenPublish</i>	设置是否允许发布屏幕流（仅Mac）。	1.1
<i>isLocalScreenPublishEnabled</i>	查询当前是否允许发布屏幕流（仅Mac）。	1.1
<i>configLocalAudioPublish</i>	设置是否允许发布音频流。	1.1
<i>isLocalAudioPublishEnabled</i>	查询当前是否允许发布音频流。	1.1
<i>configLocalSimulcast</i>	是否允许发布次要视频流。	1.1
<i>isLocalSimulcastEnabled</i>	查询当前是否允许发布次要视频流。	1.1
<i>publish</i>	手动发布视频和音频流。	1.1

订阅相关接口

API	描述	以上版本支持
<i>isAutoSubscribe</i>	查询当前是否为自动订阅模式。	1.1
<i>configRemoteCameraTrack</i>	设置是否订阅远端相机流。	1.1
<i>configRemoteScreenTrack</i>	设置是否订阅远端屏幕流。	1.1
<i>configRemoteAudio</i>	设置是否订阅远端音频流。	1.1
<i>subscribe</i>	手动订阅视频和音频流。	1.1

视频相关接口

API	描述	以上版本支持
<i>setVideoProfile</i>	设置视频流的参数。	1.1
<i>setLocalViewConfig</i>	为本地预览设置渲染窗口以及绘制参数。	1.1
<i>muteLocalCamera</i>	设置是否停止发布本地视频流。	1.1
<i>setRemoteViewConfig</i>	为远端的视频设置渲染窗口以及绘制参数。	1.1
<i>getCameraList</i>	获取摄像头列表（仅Mac）。	1.1
<i>getCurrentCamera</i>	获取当前使用的摄像头名称（仅Mac）。	1.1
<i>setCurrentCamera</i>	选择摄像头（仅Mac）。	1.1
<i>switchCamera</i>	切换前后摄像头（仅iOS）。	1.1
<i>setCameraZoom</i>	设置摄像头参数（仅iOS）。	1.1
<i>isCameraOn</i>	检查摄像头是否打开（仅iOS）。	1.1
<i>isCameraFocusPointSupported</i>	摄像头是否支持手动聚焦（仅iOS）。	1.14
<i>isCameraExposurePointSupported</i>	摄像头是否支持设置曝光区域（仅iOS）。	1.14
<i>setCameraFocusPoint</i>	设置摄像头手动聚焦（仅iOS）。	1.14
<i>setCameraExposurePoint</i>	设置摄像头曝光点（仅iOS）。	1.14
<i>unsubscribeVideoData</i>	取消订阅视频数据（仅iOS）。	1.14
<i>subscribeVideoPreprocessData</i>	订阅采集视频前处理裸数据（仅iOS）。	1.14
<i>unsubscribeVideoPreprocessData</i>	取消采集订阅前处理裸数据（仅iOS）。	1.14
<i>enableHighDefinitionPreview</i>	是否允许高清预览，默认打开（仅iOS）。	1.14

音频相关接口

API	描述	以上版本支持
<i>setAudioOnlyMode</i>	设置是否为纯音频模式还是音视频模式。	1.1
<i>isAudioOnly</i>	查询当前是否为纯音频模式。	1.1
<i>muteLocalMic</i>	设置是否停止发布本地音频。	1.1

API	描述	以上版本支持
<i>muteRemoteAudioPlaying</i>	设置是否停止播放远端音频流。	1.1
<i>enableSpeakerphone</i>	切换听筒、扬声器输出（仅iOS）。	1.1
<i>getAudioCaptures</i>	获取音频采集设备列表（仅Mac）。	1.1
<i>getCurrentAudioCapture</i>	获取当前使用的音频采集设备名称（仅Mac）。	1.1
<i>setCurrentAudioCapture</i>	选择音频采集设备（仅Mac）。	1.1
<i>getAudioRenderers</i>	获取音频播放设备列表（仅Mac）。	1.1
<i>getCurrentAudioRenderer</i>	获取当前使用的音频播放设备（仅Mac）。	1.1
<i>setCurrentAudioRenderer</i>	选择音频播放设备（仅Mac）。	1.1
<i>startAudioCapture</i>	开启音频采集。	1.11
<i>stopAudioCapture</i>	关闭音频采集。	1.11
<i>startAudioPlayer</i>	开启音频播放。	1.11
<i>stopAudioPlayer</i>	关闭音频播放。	1.11

预览接口

API	描述	以上版本支持
<i>startPreview</i>	开始本地预览。	1.1
<i>stopPreview</i>	停止本地预览。	1.1

远端用户查询接口

API	描述	以上版本支持
<i>getOnlineRemoteUsers</i>	获取远端在线用户列表。	1.1
<i>getUserInfo</i>	查询远端用户信息。	1.1
<i>isUserOnline</i>	查询用户是否在线。	1.1
<i>getMediaInfoWithUserId</i>	获取当前的媒体流信息。	1.9

其他接口

API	描述	以上版本支持
<i>setLogLevel</i>	设置log级别。	1.1

API	描述	以上版本支持
<code>getSdkVersion</code>	获取SDK版本号。	1.1

接口详情

- **setH5CompatibleMode**: 设置H5兼容模式，当前版本不支持在创建SDK实例之后更改H5兼容模式，必须在创建实例之前就调用此方法，默认不兼容H5。

```
+ (void)setH5CompatibleMode:(BOOL)comp;
```

参数:

参数	类型	描述
comp	BOOL	YES为兼容H5模式，NO为不兼容H5。

- **getH5CompatibleMode**: 检查当前是否兼容H5，返回YES标识兼容H5，NO表示不兼容H5。

```
+ (BOOL)getH5CompatibleMode;
```

- **sharedInstance::** 创建AliRtcEngine实例（同一时间只会存在一个实例）。

```
+ (instancetype)sharedInstance:(id<AliRtcEngineDelegate>)delegate
extras:(NSString *)extras;
```

参数:

参数	类型	描述
delegate	<i>AliRtcEngineDelegate</i> 类型的代理	监听回调的代理。
extras	NSString *	SDK初始化配置，目前请使用@” ”。

- **destroy**: 销毁SDK, 在所有操作结束之后调用。

```
+ (void)destroy;
```

- **setAutoPublish**: 设置是否自动发布, 是否自动订阅。默认是自动发布和订阅, 必须在 **joinChannel** 之前设置。

```
- (int)setAutoPublish:(BOOL)autoPub withAutoSubscribe:(BOOL)autoSub;
```

参数:

参数	类型	描述
autoPub	BOOL	YES表示自动发布; NO表示手动发布。
autoSub	BOOL	YES表示自动订阅; NO表示手动订阅。

- **joinChannel**: 加入频道。加入频道成功后, 如果中途需要加入其他频道, 必须先调用 **leaveChannel** 离开当前频道; 如果加入频道失败, 需要重试时, 无需先调用 **leaveChannel**。

```
- (void)joinChannel:(AliRtcAuthInfo *)authInfo name:(NSString *)
userName onResult:(void(^)(NSInteger errorCode))onResult;
```

参数:

参数	类型	描述
authInfo	<i>AliRtcAuthInfo *</i>	鉴权信息, 从App Server下发, APP Server通过API获取。
userName	NSString *	用户的显示名称, 不是uid。
onResult	void(^)(NSInteger errorCode)	当joinChannel执行结束后回调。

- **leaveChannel**: 离开频道。离开频道时, AliRtcEngine实例会被销毁, 如需继续joinChannel等操作, 需要先重新调用sharedInstance初始化AliRtcEngine实例。

```
- (void)leaveChannel;
```

- **isInCall**: 检查当前是否在频道中, 返回YES表示在频道中, NO表示不在频道中。

```
- (BOOL)isInCall;
```

- **isAutoPublish**: 查询当前是否为自动发布模式, 返回YES为自动发布, NO为手动发布。

```
- (BOOL)isAutoPublish;
```

- **configLocalCameraPublish**: 设置是否允许发布相机流。默认为允许发布相机流, 手动发布时, 需要调用publish才能生效。

```
- (void)configLocalCameraPublish:(BOOL)enable;
```

参数:

参数	类型	描述
enable	BOOL	YES为允许发布相机流; NO为不允许。

- **isLocalCameraPublishEnabled**: 查询当前是否允许发布相机流, 返回YES为允许, NO为不允许。

```
- (BOOL)isLocalCameraPublishEnabled;
```

- **configLocalScreenPublish** (仅Mac可用): 设置是否允许发布屏幕流。默认为不允许发布屏幕流, 手动发布时, 需要调用publish才能生效。

```
- (void)configLocalScreenPublish:(BOOL)enable;
```

参数:

参数	类型	描述
enable	BOOL	YES为允许发布屏幕流, NO为不允许发布屏幕流。

- **isLocalScreenPublishEnabled** (仅Mac可用) : 查询当前是否允许发布屏幕流, 返回YES为允许, NO为不允许。

```
- (BOOL)isLocalScreenPublishEnabled;
```

- **configLocalAudioPublish**: 设置是否允许发布音频流。默认为允许发布音频流, 手动发布时, 需要调用publish才能生效。

```
- (void)configLocalAudioPublish:(BOOL)enable;
```

参数:

参数	类型	描述
enable	BOOL	YES为允许发布音频流, NO为不允许。

- **isLocalAudioPublishEnabled**: 查询当前是否允许发布音频流, 返回YES为允许, NO为不允许。

```
- (BOOL)isLocalAudioPublishEnabled;
```

- **configLocalSimulcast**: 设置是否允许发布次要视频流。默认为允许发布次要视频流, 手动发布时, 需要调用publish才能生效。

```
- (int)configLocalSimulcast:(BOOL)enabled forTrack:(AliRtcVideoTrack)track;
```

参数:

参数	类型	描述
enabled	BOOL	YES表示允许发布次要流, NO表示不允许。
track	<i>AliRtcVideoTrack</i>	流类型, 当前只支持相机流: AliVideoTrackCamera。

- **isLocalSimulcastEnabled**: 查询当前是否允许发布次要视频流，返回YES为允许，NO为不允许。

```
- (BOOL)isLocalSimulcastEnabled;
```

- **publish**: 手动发布视频和音频流。
 - 调用publish的实际表现需要结合configLocalCameraPublish、configLocalAudioPublish、configLocalSimulcast等接口才能确定。
 - 根据您的具体业务需求配置上述3个接口的参数，以发布相应的视频和音频流。
 - 发布和停止发布都是调用publish。
 - 如需停止发布，则需要上述3个配置接口的参数都置为NO，再调用publish。

```
- (void)publish:(void (^)(int errorCode))onResult;
```

参数:

参数	类型	描述
onResult	void (^)(int errorCode)	当publish执行结束后回调。

- **isAutoSubscribe**: 查询当前是否为自动订阅模式，返回YES为自动订阅，NO为手动订阅。

```
- (BOOL)isAutoSubscribe;
```

- **configRemoteCameraTrack**: 设置是否订阅远端相机流。默认为订阅大流，手动订阅时，需要调用subscribe才能生效。

```
- (void)configRemoteCameraTrack:(NSString *)uid preferMaster:(BOOL)master enable:(BOOL)enable;
```

参数:

参数	类型	描述
uid	NSString *	用户ID，从App server获取的唯一标示符。
master	BOOL	是否优先订阅大流。YES为订阅大流；NO为订阅次小流。
enable	BOOL	YES为订阅远端相机流，NO为停止订阅远端相机流。

- **configRemoteScreenTrack**: 设置是否订阅远端屏幕流。默认为不订阅远端屏幕流，手动订阅时，需要调用**subscribe**才能生效。

```
- (void)configRemoteScreenTrack:(NSString *)uid enable:(BOOL)enable;
```

参数:

参数	类型	描述
uid	NSString *	用户ID，从App server获取的唯一标示符。
enable	BOOL	YES为订阅远端屏幕流，NO为停止订阅远端屏幕流。

- **configRemoteAudio**: 设置是否订阅远端音频流。默认为订阅远端音频流，手动订阅时，需要调用**subscribe**才能生效。

```
- (void)configRemoteAudio:(NSString *)uid enable:(BOOL)enable;
```

参数:

参数	类型	描述
uid	NSString *	用户ID，从App server获取的唯一标示符。
enable	BOOL	YES为订阅远端音频流，NO为停止订阅远端音频流。

- **subscribe**: 手动订阅视频和音频流。
 - 调用**subscribe**的实际表现需要结合**configRemoteCameraTrack**、**configRemoteScreenTrack**、**configRemoteAudio**等接口才能确定。
 - 根据您的具体业务需求配置上述3个接口的参数，以订阅相应的视频和音频流。
 - 订阅和停止订阅都是调用**subscribe**。
 - 如需停止订阅，则需要上述3个配置接口的参数都置为false，再调用**subscribe**。

```
- (void)subscribe:(NSString *)uid onResult:(void (^)(NSString *uid, AliRtcVideoTrack vt, AliRtcAudioTrack at))onResult;
```

参数:

参数	类型	描述
uid	NSString *	用户ID，从App server获取的唯一标示符。

参数	类型	描述
onResult	<code>void (^)(NSString *uid, <i>AliRtcVideoTrack</i> vt, <i>AliRtcAudioTrack</i> at)</code>	当subscribe执行结束后回调。

- **setVideoProfile**: 设置视频流的参数。

```
- (void)setVideoProfile:(AliRtcVideoProfile)profile forTrack:(AliRtcVideoTrack)track;
```

参数:

参数	类型	描述
profile	<i>AliRtcVideoProfile</i>	视频流参数。
track	<i>AliRtcVideoTrack</i>	需要设置的videoTrack类型。

- **setLocalViewConfig**: 为本地预览设置渲染窗口以及绘制参数。
 - 支持joinChannel之前和之后切换窗口。如果viewConfig为NULL或者其成员渲染视图为NULL, 则停止渲染。
 - 如果在播放过程中需要重新设置渲染方式, 请保持viewConfig中其他成员变量不变, 仅修改renderMode。
 - viewConfig中渲染方式默认为AliRtcRenderModeAuto。

```
- (int)setLocalViewConfig:(AliVideoCanvas *)viewConfig forTrack:(AliRtcVideoTrack)track;
```

参数:

参数	类型	描述
viewConfig	<i>AliVideoCanvas</i> *	渲染参数, 包含渲染窗口以及渲染方式。
track	<i>AliRtcVideoTrack</i>	预览只允许AliVideoTrackCamera。

- **muteLocalCamera**: 设置是否停止发布本地视频流, 不改变当前视频流的采集状态。

```
- (int)muteLocalCamera:(BOOL)mute forTrack:(AliRtcVideoTrack)track;
```

参数:

参数	类型	描述
mute	BOOL	YES表示停止发布视频流; NO表示恢复发布。
track	<i>AliRtcVideoTrack</i>	需要改变发布状态的 videoTrack类型。

- **setRemoteViewConfig**: 为远端的视频设置渲染窗口以及绘制参数。

- 支持joinChannel之前和之后切换窗口。如果canvas为NULL或者其成员view为NULL, 则停止渲染相应的流。
- 如果在播放过程中需要重新设置渲染方式, 请保持canvas中其他成员变量不变, 仅修改renderMode。
- canvas中渲染方式默认为AliRtcRenderModeAuto。

```
- (int)setRemoteViewConfig:(AliVideoCanvas *)canvas uid:(NSString *)  
uid forTrack:(AliRtcVideoTrack)track;
```

参数:

参数	类型	描述
canvas	<i>AliVideoCanvas *</i>	渲染参数, 包含渲染窗口以及 渲染方式。
uid	NSString *	用户ID, 从App server获取 的唯一标示符。
track	<i>AliRtcVideoTrack</i>	需要设置的videoTrack类 型。

- **getCameraList** (仅Mac可用) : 获取摄像头列表。

```
- (NSArray<NSString *> *)getCameraList;
```

- **getCurrentCamera** (仅Mac可用) : 获取当前使用的摄像头名称。

```
- (NSString *)getCurrentCamera;
```

- **setCurrentCamera** (仅Mac可用) : 选择摄像头。必须先调用getCameraList接口获取设备列表后再调用此接口设置。

```
- (void)setCurrentCamera:(NSString *)camera;
```

参数:

参数	类型	描述
camera	NSString *	摄像头名称。

- **switchCamera** (仅iOS可用) : 切换前后摄像头, 返回0为切换成功, 其他为切换失败

```
- (int)switchCamera;
```

- **setCameraZoom** (仅iOS可用) : 设置摄像头参数, 返回0表示设置成功, 其他表示设置失败。

```
- (int)setCameraZoom:(float)zoom flash:(BOOL)flash autoFocus:(BOOL)autoFocus;
```

参数:

参数	类型	描述
zoom	float	zoom变焦的级别。
flash	BOOL	是否打开闪光灯。
autoFocus	BOOL	是否打开自动对焦。

- **isCameraOn**: 检查摄像头是否打开, YES表示摄像头已打开, NO表示摄像头没有打开。

```
- (BOOL)isCameraOn;
```

- **isCameraFocusPointSupported** (仅iOS可用) : 摄像头是否支持手动聚焦。

```
- (BOOL)isCameraFocusPointSupported;
```

该方法返回YES表示支持, NO表示不支持。

- **isCameraExposurePointSupported** (仅iOS可用) : 摄像头是否支持设置曝光区域。

```
- (BOOL)isCameraExposurePointSupported;
```

该方法返回YES表示支持，NO表示不支持。

- **setCameraFocusPoint** (仅iOS可用) : 设置摄像头手动聚焦。

```
- (int)setCameraFocusPoint:(CGPoint)point;
```

参数:

参数	类型	描述
point	CGPoint	聚焦点坐标。

该方法返回0表示成功，其他表示失败。

- **setCameraExposurePoint** (仅iOS可用) : 设置摄像头曝光点。

```
- (int)setCameraExposurePoint:(CGPoint)point;
```

参数:

参数	类型	描述
point	CGPoint	曝光点坐标。

该方法返回0表示成功，其他表示失败。

- **unsubscribeVideoData** (仅iOS可用) : 取消订阅视频数据。

```
- (void)unsubscribeVideoData:(NSString *)uid videoSource:(AliRtcVideoSource)videoSource;
```

参数:

参数	类型	描述
uid	NSString *	用户id。
videoSource	AliRtcVideoSource	视频流类型。

- **subscribeVideoPreprocessData** (仅iOS可用) : 订阅采集视频前处理裸数据。

```
- (void)subscribeVideoPreprocessData:(AliRtcVideoSource)videoSource;
```

参数:

参数	类型	描述
videoSource	AliRtcVideoSource	视频流类型。

- **unsubscribeVideoPreprocessData** (仅iOS可用) : 取消采集订阅前处理裸数据。

```
- (void)unsubscribeVideoPreprocessData:(AliRtcVideoSource)videoSource;
```

参数:

参数	类型	描述
videoSource	AliRtcVideoSource	视频流类型。

- **enableHighDefinitionPreview** (仅iOS可用) : 是否允许高清预览, 默认打开。



说明:

需要在开启预览和开启推流之前调用。

```
- (BOOL)enableHighDefinitionPreview:(BOOL)enable;
```

参数:

参数	类型	描述
enable	BOOL	YES表示允许, NO表示不允许。

该方法返回0表示成功, 其他表示失败。

- **setAudioOnlyMode**: 设置是否为纯音频模式还是音视频模式, 返回0代表设置成功, 其他代表设置失败。默认为音视频模式 (非纯音频), 必须在joinChannel之前设置。

```
- (int)setAudioOnlyMode:(BOOL)audioOnly;
```

参数:

参数	类型	描述
audioOnly	BOOL	YES表示只有音频发布和订阅; NO表示音视频都支持。

- **isAudioOnly**: 查询当前是否为纯音频模式, 返回YES为纯音频, NO为音视频。

```
- (BOOL)isAudioOnly;
```

- **muteLocalMic**: 设置是否停止发布本地音频, 返回0表示设置成功, -1表示设置失败。不改变当前音频的采集状态。

```
- (int)muteLocalMic:(BOOL)mute;
```

参数:

参数	类型	描述
mute	BOOL	YES表示停止发布本地音频; NO表示恢复发布。

- **muteRemoteAudioPlaying**: 设置是否停止播放远端音频流, 返回0表示设置成功, -1表示设置失败。

```
- (int)muteRemoteAudioPlaying:(NSString *)uid mute:(BOOL)mute;
```

参数:

参数	类型	描述
uid	NSString *	用户ID, 从App server获取的唯一标示符。
mute	BOOL	YES表示停止播放; NO表示恢复播放。

- **enableSpeakerphone** (仅iOS可用): 切换听筒、扬声器输出。

```
- (int)enableSpeakerphone:(BOOL)enable;
```

参数:

参数	类型	描述
enable	BOOL	YES为扬声器模式, NO为听筒模式。

- **getAudioCaptures** (仅Mac可用) : 获取音频采集设备列表。

```
- (NSArray<NSString *> *)getAudioCaptures;
```

- **getCurrentAudioCapture** (仅Mac可用) : 获取当前使用的音频采集设备名称。

```
- (NSString *)getCurrentAudioCapture;
```

- **setCurrentAudioCapture** (仅Mac可用) : 选择音频采集设备。必须先调用**getCurrentAudioCapture**接口获取设备列表后再调用此接口设置。

```
- (void)setCurrentAudioCapture:(NSString *)capture;
```

参数:

参数	类型	描述
capture	NSString *	音频采集设备名称。

- **getAudioRenderers** (仅Mac可用) : 获取音频播放设备列表。

```
- (NSArray<NSString *> *)getAudioRenderers;
```

- **getCurrentAudioRenderer** (仅Mac可用) : 获取当前使用的音频播放设备。

```
- (NSString *)getCurrentAudioRenderer;
```

- **setCurrentAudioRenderer** (仅Mac可用) : 选择音频播放设备。必须先调用**getAudioRenderers**接口获取设备列表后再调用此接口设置。

```
- (void)setCurrentAudioRenderer:(NSString *)renderer;
```

参数:

参数	类型	描述
renderer	NSString *	音频播放设备名称。

- **startAudioCapture**: 开启音频采集。您可以控制提前打开音频采集，如果不设置，SDK会在开始推流的时候打开音频采集。

```
- (void)startAudioCapture;
```

- **stopAudioCapture**: 关闭音频采集。您可以控制关闭音频采集。

```
- (void)stopAudioCapture;
```

- **startAudioPlayer**: 开启音频播放。您可以控制提前打开音频播放，如果不设置，SDK会在订阅成功的时候打开音频播放。

```
- (void)startAudioPlayer;
```

- **stopAudioPlayer**: 关闭音频播放。您可以控制关闭音频播放。

```
- (void)stopAudioPlayer;
```

- **startPreview**: 开始本地预览，可以在joinChannel之前就开启预览。

```
- (int)startPreview;
```

- **stopPreview**: 停止本地预览。

```
- (int)stopPreview;
```

- **getOnlineRemoteUsers**: 获取远端在线用户列表，返回用户ID列表。

```
- (NSArray<NSString * > *)getOnlineRemoteUsers;
```

- **getUserInfo**: 查询远端用户信息。

```
- (NSDictionary *)getUserInfo:(NSString *)uid;
```

参数:

参数	类型	描述
uid	NSString *	用户ID，从App server获取的唯一标示符。

- **isUserOnline**: 查询用户是否在线，YES表示在线，NO表示不在线。

```
- (BOOL)isUserOnline:(NSString *)uid;
```

参数:

参数	类型	描述
uid	NSString *	用户ID，从App server获取的唯一标示符。

- **getMediaInfoWithUserId**: 获取当前的媒体流信息。返回key-value的json格式字符串。

```
- (NSString *)getMediaInfoWithUserId:(NSString *)userId videoTrack:(AliRtcVideoTrack)videoTrack keys:(NSArray<NSString *> *)keys;
```

参数:

参数	类型	描述
userId	NSString *	需要查询的userId, self请赋值空字符串。
videoTrack	<i>AliRtcVideoTrack</i>	需要查询的媒体流类型。
keys	NSArray<NSString *> *	查询key值数组。

- **setLogLevel**: 设置log级别。

```
- (void)setLogLevel:(AliRtcLogLevel)logLevel;
```

参数:

参数	类型	描述
logLevel	<i>AliRtcLogLevel</i>	log级别。

- **getSdkVersion**: 获取SDK版本号。

```
+ (NSString *)getSdkVersion;
```

4 Windows SDK

4.1 数据类型

本文档为您列出了Windows SDK的所有数据类型。

目录

数据类型	描述	以上版本支持
AliRtcVideoTrack	视频track类型	1.1
AliRtcAudioTrack	音频track类型	1.1
AliRtcNetworkQuality	网络质量	1.1
AliRtcRenderMode	渲染模式	1.1
AliRtcLogLevel	log级别	1.1
AliRtcVideoProfile	视频质量参数	1.1
AliRtcAuthInfo	鉴权信息	1.1
AliVideoCanvas	渲染画布	1.1
AliRtcOnByeType	OnBye返回类型	1.11

详情

- **AliRtcVideoTrack**: 视频track的类型。

枚举名	描述
AliRtcVideoTrackNo	无视频流
AliRtcVideoTrackCamera	摄像头流
AliRtcVideoTrackScreen	屏幕共享流
AliRtcVideoTrackBoth	摄像头和屏幕共享

- **AliRtcAudioTrack**: 音频track的类型。

枚举名	描述
AliRtcAudioTrackNo	无音频流
AliRtcAudioTrackMic	麦克风流

- **AliRtcNetworkQuality**: 网络质量。

枚举名	描述
AliRtcNetworkQualityGood	网络质量好
AliRtcNetworkQualityPoor	网络质量差

- **AliRtcRenderMode**: 渲染模式。

枚举名	描述
AliRtcRenderModeAuto	自动模式
AliRtcRenderModeStretch	拉伸填充视图, 不保持视频比例
AliRtcRenderModeFill	在保持视频宽高比的同时缩放, 填充黑边
AliRtcRenderModeClip	在保持视频宽高比的同时缩放, 并裁剪以适合视图

- **AliRtcLogLevel**: log级别。

枚举名	描述
AliRtcLogLevelDump	全量
AliRtcLogLevelDebug	调试
AliRtcLogLevelVerbose	详细
AliRtcLogLevelInfo	普通
AliRtcLogLevelWarn	警告
AliRtcLogLevelError	错误
AliRtcLogLevelFatal	严重
AliRtcLogLevelNone	无

- **AliRtcVideoProfile**: 视频质量参数。

视频质量参数, 默认为AliRtcVideoProfile_Default, 可以在joinChannel之前或者之后设置, 但必须在发布之前通过setVideoProfile方法设置。

枚举名	描述
AliRtcVideoProfile_Default	默认, 分辨率480 * 640, 帧率15
AliRtcVideoProfile_180_320P_15	分辨率180 * 320, 帧率15
AliRtcVideoProfile_180_320P_30	分辨率180 * 320, 帧率30
AliRtcVideoProfile_360_640P_15	分辨率360 * 640, 帧率15
AliRtcVideoProfile_360_640P_30	分辨率360 * 640, 帧率30

枚举名	描述
AliRtcVideoProfile_720_1280P_15	分辨率720 * 1280, 帧率15
AliRtcVideoProfile_720_1280P_30	分辨率720 * 1280, 帧率30

- **AliRtcAuthInfo**: 鉴权信息。



注意:

在1.9及以上版本移除session id字段。

枚举名	类型	描述
channel	AliRtc::String	频道号
user_id	AliRtc::String	用户ID
appid	AliRtc::String	应用ID
nonce	AliRtc::String	随机串
token	AliRtc::String	令牌
gslb	AliRtc::String	gslb服务器地址
timestamp	long long	时间戳

- **AliVideoCanvas**: 渲染画布。

枚举名	类型	描述
hWnd	HWND	渲染窗口句柄
renderMode	AliRtcRenderMode	渲染模式
flip	bool	- true: 镜像画面 - false: 正常画面

- **AliRtcOnByeType**: OnBye返回类型。

枚举名	描述
AliRtcOnByeChannelTerminated	频道结束

4.2 回调及监听

本文档为您介绍Windows SDK回调及监听的接口详情。

目录



说明:

回调接口都在子线程。

AliRtcEventListener: AliRtcEngine回调。

API	描述	以上版本支持
<i>onRemoteUserOnLineNotify</i>	远端用户上线回调	1.1
<i>onRemoteUserOffLineNotify</i>	远端用户下线回调	1.1
<i>onRemoteTrackAvailableNotify</i>	远端用户音视频流发生变化时回调	1.1
<i>onSubscribeChangedNotify</i>	订阅结果回调	1.1
<i>onNetworkQualityChanged</i>	网络质量变化回调	1.1
<i>onFirstRemoteVideoFrameDrawn</i>	远端视频流首帧渲染完回调	1.1
<i>onOccurWarning</i>	warning回调	1.1
<i>onOccurError</i>	error回调	1.1
<i>onBye</i>	被服务器踢出或者频道关闭时回调	1.1
<i>onExternalDeviceStateChange</i>	外接设备状态变更	1.11
<i>onFirstPacketReceived</i>	首包数据接收成功	1.13
<i>onFirstPacketSent</i>	首包发送回调函数	1.11

AliMediaDeviceTestEventListener: 音频设备测试回调。

API	描述	以上版本支持
<i>OnAudioDeviceRecordLevel</i>	麦克风音量通知	1.11
<i>OnAudioDevicePlayoutLevel</i>	音频播放音量通知	1.11
<i>OnAudioDevicePlayoutEnd</i>	音频播放完成通知	1.11

接口详情

Windows端的回调及监听接口详情如下所示。

- **onRemoteUserOnLineNotify**: 远端用户上线回调。

```
void onRemoteUserOnLineNotify(const AliRtc::String& uid)
```

参数:

参数	类型	描述
uid	const AliRtc::String&	远端用户ID。

- **onRemoteUserOffLineNotify**: 远端用户下线回调。

```
void onRemoteUserOffLineNotify(const AliRtc::String& uid)
```

参数:

参数	类型	描述
uid	const AliRtc::String&	远端用户ID。

- **onRemoteTrackAvailableNotify**: 远端用户音视频流发生变化时回调。

```
void onRemoteTrackAvailableNotify(const AliRtc::String& uid,
AliRtcAudioTrack audioTrack, AliRtcVideoTrack videoTrack)
```

参数:

参数	类型	描述
uid	const AliRtc::String&	远端用户ID。
audioTrack	<i>AliRtcAudioTrack</i>	远端用户发生变化后的音频流。
videoTrack	<i>AliRtcVideoTrack</i>	远端用户发生变化后的视频流。

注解: 远端用户停止发布, 也会触发此回调。

- **onSubscribeChangedNotify**: 订阅结果回调。

```
void onSubscribeChangedNotify(const AliRtc::String& uid, AliRtcAudioTrack audioTrack, AliRtcVideoTrack videoTrack)
```

参数:

参数	类型	描述
uid	const AliRtc::String&	远端用户ID。
audioTrack	<i>AliRtcAudioTrack</i>	订阅成功的音频流。
videoTrack	<i>AliRtcVideoTrack</i>	订阅成功的视频流。

- **onNetworkQualityChanged**: 网络质量变化时回调，建议在网络质量较差的时候可以在界面上提示用户。

```
void onNetworkQualityChanged(AliRtcNetworkQuality quality)
```

参数:

参数	类型	描述
quality	<i>AliRtcNetworkQuality</i>	当前网络质量。

- **onFirstRemoteVideoFrameDrawn**: 远端视频流首帧渲染完回调。

```
void onFirstRemoteVideoFrameDrawn(const AliRtc::String& uid,
AliRtcVideoTrack videoTrack)
```

参数:

参数	类型	描述
uid	const AliRtc::String&	远端用户ID。
videoTrack	<i>AliRtcVideoTrack</i>	渲染的流类型。

- **onOccurWarning**: Warning回调。目前warning回调可以忽略，不会影响正常使用。

```
void onOccurWarning(int warn)
```

参数:

参数	类型	描述
warn	int	警告类型。

- **onOccurError**: Error回调。当回调的中参数error为0x0102020C或0x02010105时，需要先leaveChannel，再重新创建SDK实例，然后重新joinChannel。

```
void onOccurError(int error)
```

参数:

参数	类型	描述
error	int	错误类型。

- **onBye**: 被服务器踢出或者频道关闭时回调。

```
void onBye(int code)
```

参数:

参数	类型	描述
code	int	消息类型。 <ul style="list-style-type: none"> - 1: 被服务器踢出。 - 2: 频道关闭。 - 3: 同一个userId在其他端登录, 被服务器踢出。

- **onExternalDeviceStateChange**: 外接设备状态变更。

```
void onExternalDeviceStateChange(const AliRtc::String &deviceName,
AliRtcExternalDeviceState state)``
```

参数:

参数	类型	描述
deviceName	AliRtc::String	外接设备名称。
state	AliRtcExternalDeviceState	外接设备状态。

- **onFirstPacketReceived**: 首包发送回调函数。

```
void onFirstPacketReceived(AliRtcAudioTrack audioTrack, AliRtcVideoTrack videoTrack)
```

参数:

参数	类型	描述
audioTrack	AliRtcAudioTrack	音频ID。
videoTrack	AliRtcVideoTrack	视频ID。

- **onFirstPacketSent**: 首包发送回调函数。

```
void onFirstPacketSent(AliRtcAudioTrack audioTrack, AliRtcVideoTrack videoTrack)
```

参数	类型	描述
audioTrack	AliRtcAudioTrack	是否音频首包发送。
videoTrack	AliRtcVideoTrack	是否视频首包发送。

- **OnAudioDeviceRecordLevel**: 麦克风音量通知。

```
void OnAudioDeviceRecordLevel(int level)
```

参数:

参数	类型	描述
level	int	音量等级, 范围为0~10。

- **OnAudioDevicePlayoutLevel**: 音频播放音量通知。

```
void OnAudioDevicePlayoutLevel(int level)
```

参数:

参数	类型	描述
level	int	音量等级, 范围为0~10。

- **OnAudioDevicePlayoutEnd**: 音频播放完成通知。

```
void OnAudioDevicePlayoutEnd()
```

4.3 AliRtcEngine接口

本文档为您介绍Windows SDK的AliRtcEngine接口详情。

目录

基础接口

API	描述	以上版本支持
<i>setH5CompatibleMode</i>	设置H5兼容模式。	1.1
<i>getH5CompatibleMode</i>	检查当前是否兼容H5。	1.1
<i>sharedInstance</i>	创建AliRtcEngine实例（同一时间只会存在一个实例）。	1.1
<i>destroy</i>	销毁SDK。	1.1

频道相关接口

API	描述	以上版本支持
<i>setAutoPublishSubscribe</i>	设置是否自动发布, 是否自动订阅。	1.1
<i>joinChannel</i>	加入频道。	1.1

API	描述	以上版本支持
<i>leaveChannel</i>	离开频道。	1.1
<i>isInCall</i>	检查当前是否在频道中。	1.1

发布相关接口

API	描述	以上版本支持
<i>isAutoPublish</i>	查询当前是否为自动发布模式。	1.1
<i>configLocalCameraPublish</i>	设置是否允许发布相机流。	1.1
<i>isLocalCameraPublishEnabled</i>	查询当前是否允许发布相机流。	1.1
<i>configLocalScreenPublish</i>	设置是否允许发布屏幕共享流。	1.1
<i>isLocalScreenPublishEnabled</i>	查询当前是否允许发布屏幕共享流。	1.1
<i>configLocalAudioPublish</i>	设置是否允许发布音频流。	1.1
<i>isLocalAudioPublishEnabled</i>	查询当前是否允许推音频流。	1.1
<i>configLocalSimulcast</i>	设置是否允许发布次要视频流。	1.1
<i>isLocalSimulcastEnabled</i>	查询当前是否允许发布次要视频流。	1.1
<i>publish</i>	手动发布视频和音频流。	1.1

订阅相关接口

API	描述	以上版本支持
<i>isAutoSubscribe</i>	查询当前是否为自动订阅模式。	1.1
<i>configRemoteCameraTrack</i>	设置是否订阅远端相机流。	1.1
<i>configRemoteScreenTrack</i>	设置是否订阅远端屏幕流。	1.1
<i>configRemoteAudio</i>	设置是否订阅远端音频流。	1.1
<i>subscribe</i>	手动订阅视频和音频流。	1.1

视频相关接口

API	描述	以上版本支持
<i>setVideoProfile</i>	设置视频流的参数。	1.1
<i>setLocalViewConfig</i>	为本地预览设置渲染窗口以及绘制参数。	1.1
<i>muteLocalCamera</i>	设置是否停止发布本地视频流。	1.1
<i>setRemoteViewConfig</i>	为远端的视频设置渲染窗口以及绘制参数。	1.1
<i>getCameraList</i>	获取摄像头列表。	1.1
<i>getCurrentCamera</i>	获取当前使用的摄像头名称。	1.1
<i>setCurrentCamera</i>	选择摄像头。	1.1
<i>isCameraOn</i>	检查摄像头是否打开。	1.1

音频相关接口

API	描述	以上版本支持
<i>setAudioOnlyMode</i>	设置为纯音频模式还是音视频模式。	1.1
<i>isAudioOnly</i>	查询当前是否为纯音频模式。	1.1
<i>muteLocalMic</i>	设置是否停止发布本地音频。	1.1
<i>muteRemoteAudioPlaying</i>	设置是否停止播放远端音频流。	1.1
<i>getAudioCaptures</i>	获取音频采集设备列表。	1.1
<i>getCurrentAudioCapture</i>	获取当前使用的音频采集设备名称。	1.1
<i>setCurrentAudioCapture</i>	选择音频采集设备。	1.1
<i>getAudioRenderers</i>	获取音频播放设备列表。	1.1
<i>getCurrentAudioRenderer</i>	获取当前使用的音频播放设备。	1.1
<i>setCurrentAudioRenderer</i>	选择音频播放设备。	1.1
<i>SetAudioVolume</i>	设置系统音量。	1.1
<i>GetAudioVolume</i>	获取系统音量。	1.1
<i>startAudioCapture</i>	开启音频采集。	1.11
<i>stopAudioCapture</i>	关闭音频采集。	1.11

API	描述	以上版本支持
<i>startAudioPlayer</i>	开启音频播放设备。	1.11
<i>stopAudioPlayer</i>	关闭音频播放。	1.11

预览接口

API	描述	以上版本支持
<i>startPreview</i>	开始本地预览。	1.1
<i>stopPreview</i>	停止本地预览。	1.1

远端用户查询接口

API	描述	以上版本支持
<i>getOnlineRemoteUsers</i>	获取远端在线用户列表。	1.1
<i>getUserInfo</i>	查询远端用户信息。	1.1
<i>isUserOnline</i>	查询用户是否在线。	1.1
<i>getMediaInfoWithKeys</i>	获取媒体信息。	1.9

其他接口

API	描述	以上版本支持
<i>setLogLevel</i>	设置log级别。	1.1
<i>getSdkVersion</i>	获取SDK版本号。	1.1

接口详情

- **setH5CompatibleMode**: 设置是否兼容H5，当前版本不支持在创建AliRtcEngine实例之后更改H5兼容模式，必须在创建实例之前就调用此方法，默认不兼容H5。

```
void setH5CompatibleMode(BOOL comp)
```

参数：

参数	类型	说明
comp	BOOL	0表示不兼容H5，1表示兼容H5。

- **getH5CompatibleMode**: 检查当前是否兼容H5, 返回TRUE表示兼容H5; FALSE表示不兼容H5。

```
BOOL getH5CompatibleModeine()
```

- **sharedInstance**: 创建AliRTCEngine实例 (同一时间只会存在一个实例)。

```
AliRtcEngine* sharedInstance(AliRtcEventListener* listener, const AliRtc::String &extras)
```

参数:

参数	类型	描述
listener	<i>AliRtcEventListener*</i>	AliRtcEngine回调的监听器。
extras	const AliRtc::String &	SDK初始化配置, 目前请使用空字符串。

- **destroy**: 销毁SDK, 在所有操作结束之后调用。

```
AliRtcEngine::destroy();
```

- **setAutoPublishSubscribe**: 设置是否自动发布, 是否自动订阅。默认是自动发布和订阅, 必须在joinChannel之前设置。

```
int setAutoPublishSubscribe(bool autoPub, bool autoSub)
```

参数:

参数	类型	说明
autoPub	bool	true表示自动发布; false表示手动发布。
autoSub	bool	true表示自动订阅; false表示手动订阅。

· **joinChannel**: 加入频道。

加入频道成功后，如果中途需要加入其他频道，必须先调用leaveChannel离开当前频道，如果加入频道失败，需要重试时，无需先调用leaveChannel。

```
void joinChannel(const AliRtcAuthInfo& authInfo, const AliRtc::String& userName, void*(void* opaquePtr, int errCode) onResult, void* opaquePtr)
```

参数:

参数	类型	描述
authInfo	const <i>AliRtcAuthInfo</i> &	鉴权信息，从App Server下发，APP Server可通过API获取。
userName	const <i>AliRtc::String</i> &	用户的显示名称，不是uid。
onResult	void()(void opaquePtr, int errCode)	当joinChannel执行结束后回调。
opaquePtr	void*	App提供的UserData，在调用onResult时传回App。

注解：异步接口，是否成功入会，通过onResult判断。lambda表达式转换成onResult。

```
void (*foo)(void*, int);
foo = [](void* opaquePtr, int errCode) {
    ClassA *pThis = (ClassA *)opaquePtr;
    if(errCode != 0) {
        pThis->OutputError("Failed to exeucte joinChannel.");
    }
};
mpEngine->joinChannel(/*authInfo*/, /*userName*/, foo, /*opaquePtr*/)
*/)
```

· **leaveChannel**: 离开频道。

离开频道时，AliRtcEngine实例会被销毁，如需继续joinChannel等操作，需要先重新调用sharedInstance初始化AliRtcEngine实例。

```
void leaveChannel()
```

注解：如果当前不在频道内，leaveChannel不会有任何影响，leaveChannel会产生消息通知频道内其他用户。

- **isInCall**: 检查当前是否在频道中, 返回true表示在频道中, false表示不在频道中。

```
bool isInCall()
```

- **isAutoPublish**: 查询当前是否为自动发布模式, 返回true为自动发布, false为手动发布。

```
bool isAutoPublish()
```

- **configLocalCameraPublish**: 设置是否允许发布相机流。默认为允许发布相机流, 手动发布时, 需要调用publish才能生效。

```
void configLocalCameraPublish(bool enable)
```

参数:

参数	类型	说明
enable	bool	true为允许发布相机流; false表示不允许。

- **isLocalCameraPublishEnabled**: 查询当前是否允许发布相机流, 返回true为允许, false为不允许。

```
public abstract boolean isLocalCameraPublishEnabled()
```

- **configLocalScreenPublish**: 设置是否允许发布屏幕流。默认为不允许发布屏幕流, 手动发布时, 需要调用publish才能生效。

```
void configLocalScreenPublish(bool enable)
```

参数:

参数	类型	说明
enable	bool	true表示允许发布屏幕流; false表示不允许。

- **isLocalScreenPublishEnabled**: 查询当前是否允许发布屏幕流, 返回true为允许, false为不允许。

```
bool isLocalScreenPublishEnabled()
```

- **configLocalAudioPublish**: 设置是否允许发布音频流。默认为允许发布音频流, 手动发布时, 需要调用publish才能生效。

```
void configLocalAudioPublish(bool enable)
```

参数:

参数	类型	说明
enable	bool	true表示允许; false表示不允许。

- **isLocalAudioPublishEnabled**: 查询当前是否允许发布音频流, 返回true为允许, false为不允许。

```
bool isLocalAudioPublishEnabled()
```

- **configLocalSimulcast**: 设置是否允许发布次要视频流。默认为允许发布次要视频流, 手动发布时, 需要调用publish才能生效。

```
int AliRtcEngine::configLocalSimulcast(bool enabled, AliRtcVideoTrack track)
```

参数:

参数	类型	说明
enable	bool	true表示允许发布次要流, false表示不允许。
track	<i>AliRtcVideoTrack</i>	流类型, 当前只支持相机流: AliVideoTrackCamera。

- **isLocalSimulcastEnabled**: 查询当前是否允许发布次要视频流, 返回true为允许, false为不允许。

```
bool isLocalSimulcastEnabled()
```

- **publish**: 手动发布视频和音频流。
 - 调用publish的实际表现需要结合configLocalCameraPublish、configLocalScreenPublish、configLocalAudioPublish、configLocalSimulcast等接口才能确定。
 - 根据您的具体业务需求配置上述4个接口的参数, 以发布相应的视频和音频流。
 - 发布和停止发布都是调用publish。
 - 如需停止发布, 则需要上述4个配置接口的参数都置为false, 再调用publish。

```
void publish(void(*) (void* opaquePtr, int errCode) onResult, void* opaquePtr)
```

参数:

参数	类型	描述
onResult	void()(void opaquePtr, int errCode)	当publish执行结束后回调。
opaquePtr	void*	app提供的UserData, 在调用onResult时传回app。

注解: 异步接口, 通过onResult判断调用结果。lambda表达式转换成onResult。

```
void (*foo)(void *, int);
foo = [](void*opaquePtr, int errCode) {
  ClassA *pThis = (ClassA *)opaquePtr;
  pThis->OutputError("publish result: %d", errCode);
};
```

```
publish(foo, /*UserData*/);
```

- **isAutoSubscribe**: 查询当前是否为自动订阅模式，返回true为自动订阅，false为手动订阅。

```
bool isAutoSubscribe()
```

- **configRemoteCameraTrack**: 设置是否订阅远端相机流。默认为订阅大流，手动订阅时，需要调用subscribe才能生效。

```
void configRemoteCameraTrack(const AliRtc::String& uid, bool preferMaster, bool enable)
```

参数:

参数	类型	说明
uid	String	用户ID，从App server分配的唯一标示符。
preferMaster	bool	是否优先订阅大流。true为订阅大流；false为订阅次小流。
enable	bool	true为订阅远端相机流，false为停止订阅远端相机流。

- **configRemoteScreenTrack**: 设置是否订阅远端屏幕流。默认为不订阅远端屏幕流，手动订阅时，需要调用subscribe才能生效。

```
void configRemoteAudio(const AliRtc::String& uid, bool enable)
```

参数:

参数	类型	说明
uid	const AliRtc::String&	用户ID，从App server分配的唯一标示符。
enable	bool	true为订阅远端屏幕流，false为停止订阅远端屏幕流。

- **configRemoteAudio**: 设置是否订阅远端音频流。



说明:

默认为订阅远端音频流，手动订阅时，需要调用subscribe才能生效。

```
void configRemoteAudio(const AliRtc::String& uid, bool enable)
```

参数：

参数	类型	描述
uid	const AliRtc::String&	用户ID，从App server分配的唯一标示符。
enable	bool	true为订阅远端音频流；false为停止订阅远端音频流。

· **subscribe**：手动订阅视频和音频流。返回为0时说明接口执行正常，但是否订阅成功还得看回调结果；返回为非0时，说明接口执行异常中断，订阅失败。

- 调用subscribe的实际表现需要结合configRemoteCameraTrack、configRemoteScreenTrack、configRemoteAudio等接口才能确定。
- 根据您的具体业务需求配置上述3个接口的参数，以订阅相应的视频流和音频流。
- 订阅和停止订阅都是调用subscribe。
- 如果需停止订阅，则需要上述3个配置接口的参数都置为false，再调用subscribe。

```
void subscribe(const AliRtc::String& uid, void(*)(void* opaquePtr, const AliRtc::String& uid, AliRtcVideoTrack vt, AliRtcAudioTrack at) onResult, void * opaquePtr)
```

参数：

参数	类型	说明
uid	const AliRtc::String&	用户ID，从App server分配的唯一标示符。
onResult	void()(void opaquePtr, const AliRtc::String& uid, <i>AliRtcVideoTrack</i> vt, <i>AliRtcAudioTrack</i> at)	当subscribe执行结束后回调。
opaquePtr	void *	App提供的UserData，在调用onResult时传回App。

注解：异步接口，通过onResult判断结果。lambda表达式转换成onResult。

```
void (foo)(void*, const AliRtc::String&, AliRtcAudioTrack, AliRtcVideoTrack);
foo = [](void *opaquePtr, const AliRtc::String &uid, AliRtcAudioTrack publishedAudioTrack, AliRtcVideoTrack publishedVideoTrack)
{
```

```
ClassA *pThis = (ClassA*)opaquePtr;
OutputError("subscribe result: user: %s, audio: %d, video: %d", uid.
asCString(), publishedAudioTrack, publishedVideoTrack);
};
publish(foo, this);
```

- **setVideoProfile**: 设置视频流参数。可以在joinChannel之前或者之后设置。

```
void setVideoProfile(AliRtcVideoProfile profile, AliRtcVideoTrack
track)
```

参数:

参数	类型	说明
profile	<i>AliRtcVideoProfile</i>	视频流参数。
track	<i>AliRtcVideoTrack</i>	需要设置的视频流类型。

- **setLocalViewConfig**: 为本地预览设置渲染窗口以及绘制参数。
 - 支持joinChannel之前和之后切换窗口。如果canvas中的hWnd为NULL, 则停止渲染。
 - 如果在播放过程中需要重新设置渲染方式, 请保持canvas中其他成员变量不变, 仅修改renderMode。
 - canvas中渲染方式默认为AliRtcRenderModeFill。

```
int setLocalViewConfig(const AliVideoCanvas& canvas, AliRtcVide
oTrack track)
```

参数:

参数	类型	说明
canvas	const <i>AliVideoCanvas&</i>	渲染参数, 包含渲染窗口以及渲染方式。
track	<i>AliRtcVideoTrack</i>	预览只允许AliVideoTrackCamera。

- **muteLocalCamera**: 设置是否停止发布本地视频流。不改变当前视频流的采集状态。

```
int muteLocalCamera(bool mute, AliRtcVideoTrack track)
```

参数:

参数	类型	说明
mute	bool	true表示停止发布视频流; false表示恢复发布。
track	<i>AliRtcVideoTrack</i>	需要改变发布状态的videoTrack类型。

- **setRemoteViewConfig**: 为远端的视频设置渲染窗口以及绘制参数。
 - 支持joinChannel之前和之后切换窗口。如果canvas为NULL或者其成员渲染视图为NULL，则停止渲染相应的流。
 - 如果在播放过程中需要重新设置渲染方式，请保持canvas中其他成员变量不变，仅修改renderMode。
 - canvas中渲染方式默认为AliRtcRenderModeAuto。

```
public abstract int setRemoteViewConfig(AliVideoCanvas canvas
, String uid, AliRtcVideoTrack track) int setRemoteViewConfig(
AliVideoCanvas* canvas, const AliRtc::String& uid, AliRtcVideoTrack
track)
```

参数:

参数	类型	说明
canvas	<i>AliVideoCanvas</i>	渲染参数，包含渲染窗口以及渲染方式。
uid	const AliRtc::String&	用户ID，从App server分配的唯一标示符。
track	<i>AliRtcVideoTrack</i>	需要设置的videoTrack类型。

- **getCameraList**: 获取摄像头列表。

```
void getCameraList(AliRtc::StringArray& array)
```

参数:

参数	类型	描述
array	AliRtc::StringArray&	摄像头列表。

- **getCurrentCamera**: 获取当前使用的摄像头名称。

```
AliRtc::String getCurrentCamera()
```

- **setCurrentCamera**: 选择摄像头。必须先调用getCameraList接口获取设备列表后再调用此接口设置。

```
void setCurrentCamera(const AliRtc::String& camera)
```

参数:

参数	类型	说明
camera	const AliRtc::String&	摄像头名称。

- **isCameraOn**: 检查摄像头是否打开。true表示摄像头已打开; false表示摄像头未打开。

```
bool isCameraOn()
```

- **setAudioOnlyMode**: 设置为纯音频模式还是音视频模式, 返回0代表设置成功, 其他代表设置失败。

```
int setAudioOnlyMode(bool audioOnly)
```

参数:

参数	类型	说明
audioOnly	bool	true表示只有音频发布和订阅; false表示音视频都支持。

注解: 默认为音视频模式(非纯音频), 必须在joinChannel之前设置。

- **isAudioOnly**: 查询当前是否为纯音频模式, 返回true为纯音频, false为音视频。

```
bool isAudioOnly()
```

- **muteLocalMic**: 设置是否停止发布本地音频。返回0表示设置成功, -1表示设置失败。不改变当前音频的采集状态。

```
int muteLocalMic(bool mute)
```

参数:

参数	类型	说明
mute	bool	true表示停止发布本地音频; false表示恢复发布。

- **muteRemoteAudioPlaying**: 设置是否停止播放远端音频流，返回0表示设置成功，-1表示设置失败。

```
int muteRemoteAudioPlaying(const AliRtc::String& uid, bool mute)
```

参数:

参数	类型	说明
uid	const AliRtc::String&	用户ID，从App server分配的唯一标示符。
mute	bool	true表示停止播放；false表示恢复播放。

- **getAudioCaptures**: 获取音频采集设备列表。

```
void getAudioCaptures(AliRtc::StringArray& array)
```

参数:

参数	类型	描述
array	AliRtc::StringArray&	音频采集设备列表。

- **getCurrentAudioCapture**: 获取当前使用的音频采集设备名称。

```
AliRtc::String getCurrentAudioCapture()
```

- **setCurrentAudioCapture**: 选择音频采集设备。必须先调用getAudioCaptures接口获取设备列表后再调用此接口设置。

```
void setCurrentAudioCapture(const AliRtc::String& capture)
```

参数:

参数	类型	说明
capture	String	音频采集设备名称。

- **getAudioRenderers**: 获取音频播放设备列表。

```
void getAudioRenderers(AliRtc::StringArray& array)
```

参数:

参数	类型	描述
array	AliRtc::StringArray&	音频播放设备列表。

- **getCurrentAudioRenderer**: 获取当前使用的音频播放设备。

```
AliRtc::String getCurrentAudioRenderer()
```

- **setCurrentAudioRenderer**: 选择音频播放设备。必须先调用**getAudioRenderers**接口获取设备列表后再调用此接口设置。

```
void setCurrentAudioRenderer(const AliRtc::String &renderer)
```

参数:

参数	类型	说明
renderer	String	音频播放设备名称。

- **SetAudioVolume**: 设置系统音量。

```
int SetAudioVolume(int audio_device, int volume)
```

参数:

参数	类型	说明
audio_device	int	音频设备类型。 - 1: 麦克风。 - 3: 扬声器/耳机。
volume	int	音量大小范围0~100。

- **GetAudioVolume**: 获取系统音量。返回音量大小范围0~100。

```
int GetAudioVolume(int audio_device)
```

参数:

参数	类型	说明
audio_device	int	音频设备类型。 - 1: 麦克风。 - 3: 扬声器/耳机。

- **startAudioCapture**: 开启音频采集。您可以控制提前打开音频采集，如果不设置，SDK会在开始推流的时候打开音频采集。

```
void startAudioCapture() = 0;
```

- **stopAudioCapture**: 关闭音频采集。您可以控制关闭音频采集。

```
void stopAudioCapture();
```

- **startAudioPlayer**: 开启音频播放。您可以控制提前打开音频播放，如果不设置，SDK会在订阅成功的时候打开音频播放。

```
void startAudioPlayer();
```

- **stopAudioPlayer**: 关闭音频播放。您可以控制关闭音频播放。

```
void stopAudioPlayer();
```

- **startPreview**: 开始本地预览。可以在joinChannel之前就开启预览。

```
int startPreview()
```

- **stopPreview**: 停止本地预览。

```
int stopPreview()
```

- **getOnlineRemoteUsers**: 获取远端在线用户列表。

```
void getOnlineRemoteUsers(AliRtc::StringArray& array)
```

参数:

参数	类型	说明
array	AliRtc::StringArray&	用户列表，保存的是用户ID。

- **getUserInfo**: 查询远端用户信息。返回0表示成功获取，其他表示失败。

```
int getUserInfo(const AliRtc::String& uid, AliRtc::Dictionary& dict)
```

参数:

参数	类型	说明
uid	const AliRtc::String&	用户ID，从App server分配的唯一标示符。
dict	AliRtc::Dictionary&	用于存放用户数据。

dict当中key值包括: userID、isOnline、sessionID、callID、displayName、hasAudio、hasCameraMaster、hasCameraSlave、hasScreenSharing、requestAud

io,requestCameraMaster、requestCameraSlave、requestScreenSharing、preferCameraMaster subScribedAudio、subScribedCameraMaster,subScribed CamearSlave、subScribedScreenSharing、hasCameraView、hasScreenView、muteAudioPlaying。

- **isUserOnline**: 查询用户是否在线。返回true表示在线, false表示不在线。

```
bool isUserOnline(const AliRtc::String& uid)
```

参数:

参数	类型	说明
uid	const AliRtc::String&	用户ID, 从App server分配的唯一标示符。

- **getMediaInfoWithKeys**: 获取媒体信息。返回key-value格式的json字符串。

```
AliRtc::String getMediaInfoWithKeys(const AliRtc::String& call_id,
AliRtcVideoTrack track,const AliRtc::String key_list[],int length)
= 0;
```

参数:

参数	类型	说明
call_id	AliRtc::String	需要查询的userId。
track	<i>AliRtcVideoTrack</i>	需要查询的媒体流类型。
length	int	数组长度。
key_list	AliRtc::String	查询key值数组。

- **createMediaDeviceTestInterface**: 创建音视频设备测试实例。

```
AliMediaDeviceTestInterface * createMediaDeviceTestInterface(
AliMediaDeviceTestEventListener * pMediaDeviceEventListener)
```

参数:

参数	类型	说明
pMediaDeviceEventLis tener	<i>AliMediaDeviceTestEventListener</i> *	音频设备测试事件监听器。

- **setLogLevel**: 设置log级别。

```
void setLogLevel(AliRtcLogLevel logLevel)
```

参数:

参数	类型	说明
logLevel	<i>AliRtcLogLevel</i>	log级别。

- **getSdkVersion**: 获取SDK版本号。

```
const char* getSdkVersion()
```

4.4 AliMediaDeviceTestInterface接口

本文档为您介绍Windows SDK的AliMediaDeviceTestInterface接口。

目录

API	描述
<i>StartTestAudioRecord</i>	开始测试麦克风设备
<i>StopTestAudioRecord</i>	停止测试麦克风设备
<i>StartTestAudioPlayout</i>	开始测试音频播放设备
<i>StopTestAudioPlayout</i>	停止测试音频播放设备
<i>Release</i>	销毁音频设备测试实例

接口详情

- **StartTestAudioRecord**: 开始测试麦克风设备。

```
int StartTestAudioRecord(const char *deviceName, int nTimeInv)
```

参数:

参数	类型	说明
deviceName	const char*	麦克风设备名称, 调用getAudioCaptures获取。
nTimeInv	int	音量回调通知的时间间隔, 单位为ms, 必须大于0。

返回0表示成功, 其他表示失败。

- **StopTestAudioRecord**: 停止测试麦克风设备。

```
int StopTestAudioRecord()
```

返回0表示成功，其他表示失败。

- **StartTestAudioPlayOut**: 开始测试音频播放设备。

```
int StartTestAudioPlayOut(const char *deviceName, int nTimeInv,
const char *wavFile)
```

参数:

参数	类型	说明
deviceName	const char*	音频播放设备名称，从getAudioRenderers获取。
nTimeInv	int	音量回调通知的时间间隔，单位为ms，必须大于0。
wavFile	const char*	需要播放的音频文件，例如d:/test.wav。

返回0表示成功，其他表示失败。

- **StopTestAudioPlayOut**: 停止测试音频播放设备。

```
int StopTestAudioPlayOut()
```

- **Release**: 销毁音频设备测试实例。

```
void Release()
```

5 Web SDK

5.1 回调及监听

本文为您介绍Web SDK回调及监听的接口详情。

目录

API	描述	以上版本支持
<i>onPublisher</i>	发布流事件	1.2
<i>onUnPublisher</i>	取消发布流事件	1.2
<i>onMediaStream</i>	订阅流成功事件	1.2
<i>onJoin</i>	加入频道事件	1.2
<i>onLeave</i>	离开频道事件	1.2
<i>onError</i>	错误异常事件	1.2

接口详情

AliRtcSDK Web端的回调方法如下所示。

- **onPublisher**：发布流事件。当频道里的其他人发布本地流时，触发本事件。

```
aliwebrtc.on('onPublisher',(publisher) =>{
  //远程发布者userId
  console.log(publisher.userId);
  //远程发布名字
  console.log(publisher.displayName);
  //远程流内容，streamConfigs是数组。
  console.log(publisher.streamConfigs);
});
```

- **onUnPublisher**：取消发布流事件。当频道里的其他人取消发布本地流时，触发本事件。

```
aliwebrtc.on('onUnPublisher',(publisher) =>{
  //远程发布者userId
  console.log(publisher.userId);
  //远程发布名字
  console.log(publisher.displayName);
});
```

- **onMediaStream**：订阅流成功事件。当订阅远程流成功时触发，会返回远程流的Stream对象，通过H5 Video或Audio播放。

```
aliwebrtc.on('onMediaStream',(subscriber, stream) =>{
  var video = document.getElementsByTagName('video');
  aliwebrtc.setDisplayRemoteVideo(
```

```

        subscriber, // onMediaStream中返回的参数
        video, // html中用于显示stream对象的video元素
        stream // onMediaStream中返回的参数
    )
});
    
```

- **onJoin**: 加入频道事件。当有其他用户加入频道是触发onJoin事件。

```

aliwebrtc.on('onJoin',(data) =>{
    console.log(data.userId);
    console.log(data.displayName + " 加入频道");
});
    
```

- **onLeave**: 离开频道事件。当其他用户离开频道是触发onLeave事件。

```

aliwebrtc.on('onLeave',(data) =>{
    console.log(data.userId);
    console.log(data.displayName + " 离开频道");
});
    
```

- **onError**: 错误异常事件。当有错误发生时，触发onError事件。

```

aliwebrtc.on('onError',(error) =>{
    var msg = error && error.message ? error.message : error;
    if (msg && msg.indexOf('no session') > 0) {
        error = "请重新登录：" + msg;
    }
    console.log('错误：' + msg);
});
    
```

5.2 AliRtcEngine接口

本文为您介绍AliRtcSDK Web端的所有接口信息，您可以根据接口详细说明集成SDK。

目录

基础接口

API	描述	以上版本支持
isSupport	检测浏览器是否支持。	1.7
getDevices	获取设备信息。	1.2
getAvailableResolutions	获取可支持的分辨率。	1.2

频道相关接口

API	描述	以上版本支持
joinChannel	加入频道。	1.2
leaveChannel	离开频道。	1.2

发布相关接口

API	描述	以上版本支持
<i>configLocalAudioPublish</i>	设置是否允许发布音频流。	1.9
<i>configLocalCameraPublish</i>	设置是否允许发布相机流。	1.9
<i>configLocalScreenPublish</i>	设置是否允许发布屏幕共享流。	1.9
<i>publish</i>	发布本地视频流。	1.2
<i>unPublish</i>	结束发布本地流。	1.2

订阅相关接口

API	描述	以上版本支持
<i>configRemoteAudio</i>	设置是否订阅远端音频流。	1.9
<i>configRemoteCameraTrack</i>	设置是否订阅远端相机流。	1.9
<i>configRemoteScreenTrack</i>	设置是否订阅远端屏幕流。	1.9
<i>subscribe</i>	订阅远程发布流。	1.2
<i>unSubscribe</i>	取消订阅该用户所有的流。	1.2

视频相关接口

API	描述	以上版本支持
<i>currentCamera</i>	指定摄像头设备。	1.9
<i>videoProfile</i>	设置视频流参数。	1.9
<i>muteLocalCamera</i>	是否停止本地视频采集。	1.2
<i>setDisplayRemoteVideo</i>	为远端的视频设置渲染窗口以及绘制参数。	1.5

音频相关接口

API	描述	以上版本支持
<i>currentAudioCapture</i>	指定麦克风设备。	1.9
<i>muteLocalMic</i>	是否停止本地音频采集。	1.2

预览接口

API	描述	以上版本支持
<i>startPreview</i>	预览本地摄像头。	1.2

API	描述	以上版本支持
<i>stopPreview</i>	结束预览本地摄像头。	1.2

远端用户查询接口

API	描述	以上版本支持
<i>getUserList</i>	获取当前房间在线用户。	1.9
<i>getUserInfo</i>	获取用户信息。	1.9

其他接口

API	描述	以上版本支持
<i>on</i>	订阅事件。	1.2
<i>off</i>	取消订阅事件。	1.2

接口详情

AliRtcSDK Web端接口信息如下所示。

- **isSupport():** 检测浏览器是否支持。

```
AliRtcEngine.isSupport().then((re)=> {
  console.log(re);
}).catch(error => {
  console.log(error);
})
```

返回参数:

参数	类型	描述
audioDevice	Boolean	true表示音频设备可用, false表示音频设备不可用。
browser	String	浏览器名称。
browser_version	String	浏览器版本。
isSupported	Boolean	是否支持webRTC。
message	String	错误信息。
supportH264	Boolean	true表示支持H264, false表示不支持H264。
supportScreenShare	Boolean	true表示支持屏幕分享, false表示不支持屏幕分享。

参数	类型	描述
videoDevice	Boolean	true表示支持摄像头可用, false表示摄像头不可用。

- **getDevices():** 获取设备信息, 返回摄像头和音频输入设备, 在Safari浏览器下面, 如果外接设备重插拔后获取不到, 请尝试重新启动电脑。

```
aliwebrtc.getDevices().then((re)=>{
}).catch((error)=>{
    console.log(error.message)
});
```

- **getAvailableResolutions(deviceId):** 获取可支持的分辨率。返回支持分辨率的Array数组信息。

```
aliwebrtc.getAvailableResolutions(deviceId).then((re)=>{
}).catch((error)=>{
    console.log(error.message)
});
```

参数:

参数	类型	描述
deviceId	String	摄像头ID。

- **joinChannel(config,displayName):** 加入频道。

```
aliwebrtc.joinChannel({
    userid, // 用户ID, 只能由数字、字母、下划线组成
    channel, // 频道
    appid, // 应用ID
    nonce, // nonce
    timestamp, // 时间戳
    gslb, // gslb
},displayName).then(()=>{
    // 入会成功
} , (error)=>{
    // 入会失败, 这里console下error内容, 可以看到失败原因
    console.log(error.message);
});
```

参数:

参数	类型	描述
config	—	鉴权频道信息。
	userid	String 用户ID (只能包含数字和字母)。
	channel	String 频道。
	appid	String 应用ID。

参数		类型	描述
	nonce	String	nonce。
	timestamp	String	时间戳。
	gslb	Array	Global Server Load Balancing (简称gslb)。
displayName		String	用户名字。

- **leaveChannel():** 离开频道。

```
aliwebrtc.leaveChannel().then(()=>{
},(error)=>{
    console.log(error.message);
});
```

- **configLocalAudioPublish:** 设置是否允许发布音频流。

 **说明:**
默认为允许发布音频流，需要调用publish才能生效。

```
aliWebrtc.configLocalAudioPublish=enable;
```

参数:

参数	类型	描述
enable	Boolean	true表示允许发布音频流；false表示不允许。

- **configLocalCameraPublish:** 设置是否允许发布相机流。

 **说明:**
默认为允许发布相机流，需要调用publish才能生效。

```
aliWebrtc.configLocalCameraPublish=enable;
```

参数:

参数	类型	描述
enable	Boolean	true表示允许发布音频流，false表示不允许。

- **configLocalScreenPublish:** 设置是否允许发布屏幕共享流。

 **说明:**

默认为允许发布音频流，需要调用publish才能生效。

```
aliWebrtc.configLocalAudioPublish=enable;
```

参数：

参数	类型	描述
enable	Boolean	true表示允许发布音频流；false表示不允许。

- **publish()**：发布本地视频流，如果需要对远程订阅本地的流，需要调用publish接口，发布本地流，远程会接收到onPublisher事件。

```
aliwebrtc.publish().then(()=>{
  } ,(error)=>{
    console.log(error.message);
  });
```

- **unPublish()**：结束发布本地流，当您取消发布本地流时，远程会收到onUnPublisher事件。

```
aliwebrtc.unPublish().then(()=>{
  } ,(error)=>{
    console.log(error.message);
  });
```

- **configRemoteAudio**：设置是否订阅远端音频流。

 **说明：**
默认为订阅远端音频流，需要调用subscribe才能生效。

```
aliWebrtc.configRemoteAudio(userId,enable);
```

参数：

参数	类型	描述
userId	String	用户ID。
enable	Boolean	true为订阅远端音频流，false为停止订阅远端音频流。

- **configRemoteCameraTrack**：设置是否订阅远端相机流。

相机流包含大流和小流，大流分辨率更高，小流分辨率较低。

 **说明：**

默认为订阅大流，需要调用subscribe才能生效。

```
aliWebrtc.configRemoteCameraTrack(userId,preferMaster,enable);
```

参数：

参数	类型	描述
userId	String	用户ID。
preferMaster	Boolean	是否优先订阅大流。true为订阅大流，false为订阅次小流。
enable	Boolean	true为订阅远端相机流，false为停止订阅远端相机流。

- **configRemoteScreenTrack**：设置是否订阅远端屏幕流。



说明：

默认为不订阅远端屏幕流，需要调用subscribe才能生效。

```
aliWebrtc.configRemoteScreenTrack(userId,enable);
```

参数：

参数	类型	描述
userId	String	用户ID。
enable	Boolean	true为订阅远端屏幕流，false为停止订阅远端屏幕流。

- **subscribe(userId)**：订阅远程发布流，通过subscirbe方法可以订阅远程的流，默认订阅相机流和音频流，可以通过调用**configRemoteAudio**、**configRemoteCameraTrack**、**configRemoteScreenTrack**来设置订阅内容。

```
aliwebrtc.subscribe(userId).then((userId)=>{
},(error)=>{
    console.log(error.message);
});
```

参数：

参数	类型	描述
userId	String	用户ID。

- **unsubscribe(userId)**: 取消订阅该用户所有的流。

```
aliwebrtc.unsubscribe(userId).then(() => {
},(error)=>{
    console.log(error.message);
});
```

参数:

参数	类型	描述
userId	String	用户ID。

- **currentCamera**: 指定摄像头设备。暂不支持热切换, 需要在预览和推流之前设置。

```
//pc端
aliwebrtc.currentCamera = {
    deviceId: deviceId
};
//手机端 facingMode指定user时启用前置摄像头、指定environment时启用后置摄像头
aliwebrtc.currentCamera = {
    facingMode:"user"
};
```

参数:

参数	类型	描述
deviceId	String	摄像头ID。

- **videoProfile**: 设置视频流参数。您要设置的分辨率需要调用getAvailableResolutions()返回, 然后调用publish()才能生效。

 **说明:**
如果您设置的分辨率不合适, 系统会自动进行调整。

```
aliwebrtc.videoProfile = {
    frameRate:20,
    width: 640,
    height: 480
};
```

参数:

参数	类型	描述
frameRate	int	帧率 (5~30)。
width	int	视频宽度。
height	int	设备高度。

- **muteLocalCamera(mute)**: 是否停止本地视频采集。

```
aliwebrtc.muteLocalCamera(true);
```

参数:

参数	类型	描述
mute	Boolean	true为视频采集, false为恢复正常。

- **setDisplayRemoteVideo**: 为远端的视频设置渲染窗口以及绘制参数。

```
aliwebrtc.setDisplayRemoteVideo(
  subscriber,      // onMediaStream中返回的参数
  video,           // html中用于显示stream对象的video元素
  stream           // onMediaStream中返回的参数
)
```

- **currentAudioCapture**: 指定麦克风设备。暂不支持热切换, 需要在预览和推流之前设置。

```
aliwebrtc.currentAudioCapture = {
  deviceId: deviceId
};
```

参数:

参数	类型	描述
deviceId	String	麦克风ID。

- **muteLocalMic(mute)**: 是否停止本地音频采集。

```
aliwebrtc.muteLocalMic(true);
```

参数:

参数	类型	描述
mute	Boolean	true为停止视频采集, false为恢复正常。

- **startPreview(video)**: 预览本地摄像头, 通过video标签播放。

```
aliwebrtc.startPreview(
  video // html中的video元素
).then(()=>{
}).catch((error) => {
  // 预览失败
});
```

- **stopPreview()**: 结束预览本地摄像头。

```
aliwebrtc.stopPreview().then(()=>{
}).catch((error) => {
```

```
}); // 结束预览失败
```

- **getUserList():** 获取当前房间在线用户。返回当前房间的在线用户列表信息。

```
aliWebrtc.getUserList();
```

返回参数:

参数	类型	描述
displayName	String	用户名称。
userId	String	用户ID。

- **getUserInfo(userId):** 获取用户信息。

```
aliWebrtc.getUserInfo(userId);
```

返回参数:

参数	类型	描述	
displayName	String	用户名称。	
streamConfigs	—	Array	音视频流数组。
	label	String	流标签。
	state	String	active表示流可用, inactive表示流不可用。
	subscribed	Boolean	true表示已订阅, false表示未订阅。
	type	String	audio表示音频, video表示视频。
userId	String	用户ID。	

- **on(name, handler):** 订阅事件。

参数:

参数	类型	描述
name	String	事件名字。
handler	String	处理事件的方法。

- **off(name, handler)**: 取消订阅事件。

参数:

参数	类型	描述
name	String	事件名字。
handler	String	处理事件的方法。

6 客户端错误码表

本文为您介绍客户端常见错误码，通过查询对应的错误码，您可以快速定位客户端出现的错误，方便您更好的体验音视频通信相关服务。

十六进制	十进制	描述
0x00000009	9	状态错误，SDK实例不存在或已被销毁。
0x01030101	16974081	参数错误，authinfo中的字段为空或者timestamp<=0。
0x01030204	16974340	状态错误。比如在joinChannel时，当前状态为正在jioning中，已经jion成功，正在离开频道，或者SDK正在销毁中时，都会报这个错误码。
0x01030304	16974596	本地流不存在。
<ul style="list-style-type: none"> · 0x01030305 · 0x01030308 	<ul style="list-style-type: none"> · 16974597 · 16974600 	本地流错误。
0x01030403	16974851	远端流不存在。
<ul style="list-style-type: none"> · 0x01030404 · 0x01030405 	<ul style="list-style-type: none"> · 16974852 · 16974853 	远端流错误。
<ul style="list-style-type: none"> · 0x01010103 · 0x0101010C · 0x0101010D 	<ul style="list-style-type: none"> · 16843011 · 16843020 · 16843021 	初始化SDK异常。
0x01040403	17040387	音频设备异常，建议检查音频设备。
0x1020204	16908804	入会超时。
0x0102020C	16908812	心跳包超时，建议leavechannel之后再重新create、join。
0x01020401	16909313	信令请求错误。

十六进制	十进制	描述
<ul style="list-style-type: none"> · 0x01020210 · 0x01020211 · 0x01020212 · 0x01020213 · 0x01020214 · 0x01020215 · 0x01020216 · 0x01020217 · 0x01020218 	<ul style="list-style-type: none"> · 16908816 · 16908817 · 16908818 · 16908819 · 16908820 · 16908821 · 16908822 · 16908823 · 16908824 	信令请求超时。
0x02010201	33620481	加入频道失败, AppID不存在, 您可以在控制台创建应用。
0x02010202	33620482	加入频道失败, 应用已经失效, 您可以在控制台创建重新激活应用。
0x02010203	33620483	加入频道失败, 频道已经失效, 目前频道有效期为48小时。
0x02010204	33620484	加入频道失败, 频道不存在。
0x02010205	33620485	加入频道失败, token无效。
0x02010401	33620993	发布流hangup。
0x02010402	33620994	订阅流hangup。
0x02010403	33620995	订阅流失败, 远端流已经停止发布。