



蚂蚁科技产品手册

音视频通话

产品版本：V20210430
文档版本：V20210430
蚂蚁科技技术文档

蚂蚁科技集团有限公司版权所有 © 2020，并保留一切权利。

未经蚂蚁科技事先书面许可，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。

商标声明



及其他蚂蚁科技服务相关的商标均为蚂蚁科技所有。
本文档涉及的第三方的注册商标，依法由权利人所有。

免责声明

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。蚂蚁科技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，并在蚂蚁科技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蚂蚁科技授权渠道下载、获取最新版的用户文档。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

目录

1 产品简介	1
2 快速开始	4
3 接入客户端	4
3.1 接入 Android	4
3.1.1 快速开始	5
3.1.2 Android 进阶功能	6
3.1.3 代码示例	9
3.2 接入 iOS	10
3.2.1 接入 iOS	10
3.2.2 使用 iOS SDK	11
3.2.3 iOS 进阶功能	16
3.2.4 代码示例	23
3.3 接入 Web	24
4 安全加签	24
4.1 接入步骤	24
4.2 代码示例	25
5 使用控制台	26
5.1 通话应用管理	26
5.1.1 创建通话应用	26
5.1.2 管理通话应用	27
5.2 签名校验	30
6 API 参考	31
6.1 Android API	31
6.1.1 权限及隐私说明	31
6.1.2 重要参数	32
6.1.3 公共函数	38
6.1.4 回调函数	50
6.1.5 错误码	62
6.2 iOS API	63
6.2.1 重要参数	63
6.2.2 公共函数	65
6.2.3 回调函数	75
6.3 Web SDK API	75
6.3.1 概述	75
6.3.2 主调接口	79
6.3.3 被调接口	93
6.3.4 错误码	112
6.3.5 Web SDK 发布说明	114

1 产品简介

音视频通话组件 (Mobile Real-Time Communication , 简称 MRTC) 是 mPaaS 提供的音频、视频通话组件。该组件功能丰富, 不仅支持一对一的双人视频通话, 还支持多人视频通话、在线会议等场景。支持独立语音通话。支持即时消息和文件传输、个性化的屏幕录制、屏幕共享、截图、点播、美颜等功能。支持 PC、手机、Pad、小程序等多终端接入, 不同终端和 App 之间能够互通。安全性高, 端到端全链路加密, 房间信令创建采用非对称加密, 符合国密标准。视频流传输采用对称加密, 兼顾安全和性能。视频录制加密传输、存储。通话过程中支持实时语音识别, 能够识别客户对风险提示的语音确认, 辅助坐席判断客户的意向。支持在视频通话过程中, 播放视频、PPT 等风险提示画面。

功能特点

- **视频通话**
 - 多种参与模式: 支持一对一视频通话、一对多视频通话、多对多视频通话。
 - 多平台: 支持 iOS、Android 以及 PC Web。
 - 多端互通: 支持手机、PC IoT 设备之间互联互通。
 - 会话保持: 网络短暂异常、网络切换时, 业务流程不中断, 保持会话的持续性。
 - 自定义视频规格、自适应视频规格: 支持自定义宽、高、最大帧率、最大码率, 并能在上限范围内根据网络状况自适应调整视频规格。
- **录制**
 - 多粒度录制控制: 纯语音录制、纯视频录制、音视频混合录制、一人录制、两人录制、多人录制。
 - 多端录制: 根据安全要求, 既可以在坐席侧录制, 也可以在服务端录制。
 - 多种画面组合模式: 支持画中画、九宫格模式。
 - 叠加水印: 支持实时叠加文字、图片水印、时间戳, 以防篡改。
 - 多种存储系统: 支持 NAS、OSS、AFTS、本地文件系统。
 - 多格式: 支持多种常见格式, 包括 FLV、MP4 等。
- **数据通信**
 - 即时消息: 支持简单的文字消息, 方便在视频通话前进行沟通。
 - 文件传输: 上传文件后, 传输文件链接到对端, 供对端下载获取文件。
 - 命令透传: 为业务提供透明可靠的命令传输通道。
- **自动化和智能化**
 - 自动语音风险提示: 自动播报风险提示等内容, 代替人工说明。
 - 视频点播: 在视频通话过程中, 播放视频、PPT 等风险提示画面。
 - 自动纪要: 实时识别视频通话的语音内容, 并按照时间分角色交替文字记录说话内容。
 - 自动确认: 识别客户对风险提示的语音确认, 辅助坐席判断客户的意向。
- **增强功能**
 - 屏幕共享: 将本段的屏幕内容投送到其他端。
 - 美颜: 美化面貌, 可调整美化等级。
 - 截图: 支持远程控制摄像头拍照、本端截图。
- **系统转接**
 - 旁路直播: 云端混音合图, 转接直播系统。支持推流到 CDN、第三方直播平台。

产品优势

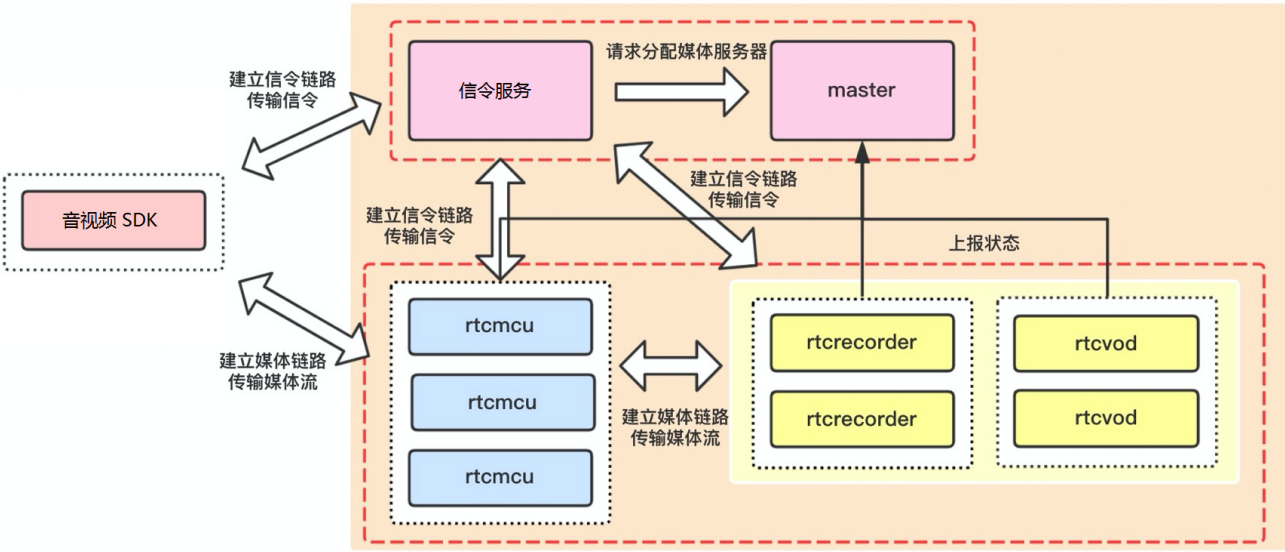
- **高安全性**
 - 动态加签验签。
 - 端到端全链路加密。
 - OpenAPI 动态密钥鉴权。
 - 录制加密传输、存储。
- **弱网环境下视频质量自适应**
 - 根据网络丢包类型判断、丢包率、RTT、网络抖动评估，来准确的评估网络有效带宽。
 - 经过优化的视频质量自适应算法，视频清晰度、流畅度、尺寸可随目标码率实时变化。
 - 优先保证感兴趣区域的视频质量。
- **高可靠的媒体流和信令传输**
 - 实时流媒体传输平台，确保传输稳定。
 - 冗余链路和优选，动态切换。
 - 信令在节点到节点、端到端实现了双层高可靠机制，既降低了时延，又提高了可靠性，使可靠性达到了 99.99%。
- **低时延**
 - 超低时延，远程操控领域可达到 200 ms。
 - 快速对焦。
 - 改进的 FEC 算法，抗丢包 40%。
 - H.264-highprofile 技术，可减少高达 50% 的带宽消耗。
- **多端接入**
 - 支持 PC、手机、Pad、小程序、IoT 等多终端接入，不同终端和 App 之间能够互通。
- **功能丰富**
 - 双录：云端双向录制、多粒度、多格式、个性化水印、多种存储方式。
 - 智能坐席：自动语音风险提示、视频点播、自动纪要。
- **接口简洁，接入方便**
 - 提供轻量版本，便于接入。
 - 已有业务最快接入纪录：一周开发上线。

产品架构

音视频通话的产品架构如下：



音视频通话的系统架构如下：



- 音视频 SDK：客户端 SDK，现有支持 Android、iOS、PC Web、小程序、Linux 以及 Windows。
- 信令服务：负责信令连接管理、信令转发、房间管理、生命周期维护。
- master：负责均衡调度服务器，负责媒体服务器的均衡调度、存活管理。
- rtcmcu：媒体转发服务，负责媒体流接收、转发、处理、状态维护。
- rtcrecorder：录制服务，负责录制存储。
- rtcvod：点播服务，负责音视频点播。

2 快速开始

本节引导您快速使用音视频通话：

1. 登录 mPaaS 控制台，点击 + 创建应用，输入应用名并创建 mPaaS 应用。



2. 进入 mPaaS 应用，在左侧导航栏中选择 音视频通话 > 通话应用管理 进入音视频通话产品页。
3. 创建通话应用。创建后可查看 bizName 和密钥，相同 bizName 下的用户可互相通话。
扫描下方二维码加入钉钉群，获取接入支持。除接入外，该群还可提供全方位的技术支持。

mPaaS音视频通话技术支持

7人



 扫一扫群二维码，立刻加入该群。

4. 生成签名。音视频通话中提供两种签名生成方法：
 - 服务端生成：通过安全加签生成签名。
 - 控制台生成：在控制台中生成临时签名，可用于临时体验视频通话产品。
5. 使用音视频通话。客户端 SDK 初始化时传入生成的签名和 API 参数，开始音视频通话。

3 接入客户端

3.1 接入 Android

3.1.1 快速开始

说明：音视频通话只在 10.1.68 及以上版本基线中提供支持。

前置条件

- 若采用 **原生 AAR 方式** 接入，需先完成 将 mPaaS 添加至您的项目 。并确保工程根目录 build.gradle 文件中，有如下依赖：

```
classpath 'com.android.boost.easyconfig:easyconfig:2.4.3'
```

确保主工程（ android main module ） 的 build.gradle 中有如下配置：

```
apply plugin: 'com.alipay.apollo.baseline.config'
```

- 若采用 **mPaaS Inside 方式** 接入，需先完成 mPaaS Inside 接入流程。
- 若采用 **组件化（ Portal&Bundle ）方式** 接入，需先完成 组件化接入流程。

添加 SDK

原生 AAR 方式

参考 AAR 组件管理，通过 **组件管理（AAR）** 在工程中安装 **音视频通话** 组件。

mPaaS Inside 方式

在工程中通过 **组件管理** 安装 **音视频通话** 组件。
更多信息，参考 管理组件依赖。

组件化方式

在 Portal 和 Bundle 工程中通过 **组件管理** 安装 **音视频通话** 组件。
更多信息，参考 管理组件依赖。

初始化 mPaaS

如果使用 **原生 AAR 接入** 或 **mPaaS Inside 接入**，需要初始化 mPaaS。

在 Application 中添加以下代码：

```
public class MyApplication extends Application {  
  
    @Override  
    protected void attachBaseContext(Context base) {  
        super.attachBaseContext(base);  
        // mPaaS 初始化回调设置。  
        QuinoxlessFramework.setup(this, new IInitCallback() {
```



```

@Override
public void onPostInit() {
    // 此回调表示 mPaaS 已经初始化完成，mPaaS 相关调用可在这个回调里进行。
}
});
}

@Override
public void onCreate() {
    super.onCreate();
    // mPaaS 初始化。
    QuinoxlessFramework.init();
}
}

```

使用 SDK

1. 初始化 engine，设置自动发布和订阅。
将 autoPublish/autoSubscribe 设置为 true。

```

AlipayRtcEngine rtcEngine = AlipayRtcEngine.getInstance( this );
rtcEngine.setRtcListenerAndHandler( engineEventListener, eventHandler );
rtcEngine.setImListener( imListener );
rtcEngine.setInviteListener( inviteListener );
rtcEngine.setServerAddr( SERVER_ONLINE );
rtcEngine.setAutoPublishSubscribe( autoPublish, autoSubscribe );
publishConfig = new PublishConfig();
rtcEngine.configAutoPublish( publishConfig );

```

2. 创建房间/加入房间。

```

CreateRoomParams createRoomParams = new CreateRoomParams();
rtcEngine.createRoom( createRoomParams );
JoinRoomParams joinRoomParams = new JoinRoomParams();
rtcEngine.joinRoom( joinRoomParams );

```

3. 监听本端和对端的 view。
 - 本端：onCameraPreviewInfo(final ARTVCView view)
 - 对端：void onRemoteViewFirstFrame(final FeedInfo info, final ARTVCView _view)

3.1.2 Android 进阶功能

设置音视频通话或纯音频通话

设置纯音频通话，在 初始化 engine 时，按如下方法配置 publishConfig：

```
publishConfig = new PublishConfig();
```

- 纯音频通话

```
publishConfig.videoSource = VIDEO_SOURCE_NULL;  
publishConfig.audioSource = AUDIO_SOURCE_MIC;
```

- 视频通话

```
publishConfig.videoSource = VIDEO_SOURCE_CAMERA;  
publishConfig.audioSource = AUDIO_SOURCE_MIC;
```

动态调整分辨率

在通话成功建立后，可按如下方法动态调整本端发送的分辨率：

```
public void updateVideoProfile( VideoProfile videoProfile, int maxBitrate )
```

其中：

- videoProfile 设置为新配置的分辨率配置。
- maxBitrate 设置为新配置的码率，0 为使用默认值。

通话质量监控

在通话成功建立后，监听以下两个回调，可以实时获取当前视频通话质量：

```
void onStatisticDebugInfo( StatisticInfoForDebug infoForDebug, FeedInfo feedInfo ); //获取通话过程调试信息  
void onRealTimeStatisticInfo( RealTimeStatisticReport report, FeedInfo feedInfo ); //获取实时监控信息
```

其中 feedInfo 为对应流的信息，包括对端和本端。

网络变化监控

在通话成功建立后，监听以下回调，获取带宽信息：

```
void onBandwidthImportanceChangeNotify( boolean isLow, double currentBandwidth, FeedInfo feedInfo ); //带宽不足通知
```

带宽过低会通过 isLow 通知，如果带宽过低，可能会断开或者通话质量差。

屏幕共享

开启屏幕共享功能，按如下设置后，可实现在视频通话的同时共享屏幕。

1. 在 初始化 engine 时，不能配置自动发布，但是可以配置自动订阅。即：

```
rtcEngine.setAutoPublishSubscribe( autoPublish, autoSubscribe );
```

其中：

- autoPublish 必须为 false。
- autoSubscribe 无特殊限制。

2. 初始化后，在 创建或加入房间 后，调用：

```
publishConfig = new PublishConfig();  
publishConfig.videoSource = VIDEO_SOURCE_SCREEN;  
rtcEngine.publish( publishConfig );
```

截屏功能

在通话建立后，需要截屏，可以通过以下方法进行截屏：

```
public void snapshot( FeedInfo info )
```

其中 feedInfo 为对应截屏的流信息。

截屏结果通过以下方法进行回调：

```
void onSnapShotComplete( Bitmap image, FeedInfo feedInfo ); //截屏图像回调
```

自定义推流

开启自定义推流。

1. 与屏幕共享相同，在 初始化 engine 时，不能配置自动发布，但是可以配置自动订阅。即：

```
rtcEngine.setAutoPublishSubscribe( autoPublish, autoSubscribe );
```

其中：

- autoPublish 必须为 false。
- autoSubscribe 无特殊限制。

2. 在 创建或加入房间 后，调用：

```
publishConfig = new PublishConfig();  
publishConfig.videoSource = VIDEO_SOURCE_CUSTOM;  
rtcEngine.publish( publishConfig );
```

3. 在发布成功之后，通过以下方法的结果不断的输入图像数据：

```
public void pushCustomVideoData(byte[] bytes, int width, int height, int rotation ) //自定义推流数据，目前仅支持  
nv21 格式数据
```

本地预览

如果需要在未进入视频通话时，提前开启预览，可以在 初始化 engine 后调用以下方法开启摄像头预览：

```
public void startCameraPreview()
```

预览的 view 可以参考 监听本端和对端的 view 。

3.1.3 代码示例

下载代码示例

[点击这里](#) 获取音视频通话代码示例。

使用示例代码

1. 替换示例工程中的配置文件。
 - 登录 mPaaS 控制台。
 - 创建 mPaaS 应用 。
 - 下载 mPaaS 应用的 .config 格式 配置文件 放入音视频通话示例工程的 app/ 目录下，用以替换已有的配置文件。

2. 创建通话应用 并获取 bizName。

在控制台左侧导航栏选择 **音视频通话** > **通话应用管理**，创建通话应用或进入已有的通话应用，获取通话应用的 bizName。

3. 控制台 生成临时签名 。

在控制台左侧导航栏选择 **音视频通话** > **签名校验**，在上方的 **生成临时签名** 区域，选择通话应用，输入 **UserID** 生成临时签名。

说明：

- 更换通话应用或 UserID 后需要重新生成临时签名。
- 注意临时签名的有效期，过期后需重新生成。

4. 配置示例工程。

运行示例工程，在 App 设置页填写从以上步骤获取的custome uid (userId)、业务名称 (bizName) 和临时签名 (signature) 。

说明：为保障您的流量安全，业务上线后务必 通过服务端生成签名 。

5. 发起或加入音视频通话。

您可在 App 中创建通话房间发起音视频通话，或者在页面上输入其他用户创建的视频通话房间号 (roomId) 和 token 加入音视频通话。

说明：在同一个通话应用中（使用同一个 bizName）的不同端的 userId，必须互不相同。

3.2 接入 iOS

3.2.1 接入 iOS

音视频通话支持 基于 mPaaS 框架、基于原生框架且使用 mPaaS 插件 以及 基于原生框架且使用 CocoaPods 3 种接入方式，您可以自由选择。

说明：

- 音视频通话只在 10.1.68 及以上版本基线中提供支持。
- 目前 iOS 音视频通话中暂不支持 ARMv7 架构。

前置条件

您已经接入工程到 mPaaS。更多信息，请参见以下内容：

- 基于 mPaaS 框架接入
- 基于已有工程且使用 mPaaS 插件接入
- 基于已有工程且使用 CocoaPods 接入

添加 SDK

根据您采用的接入方式，请选择相应的添加方式。

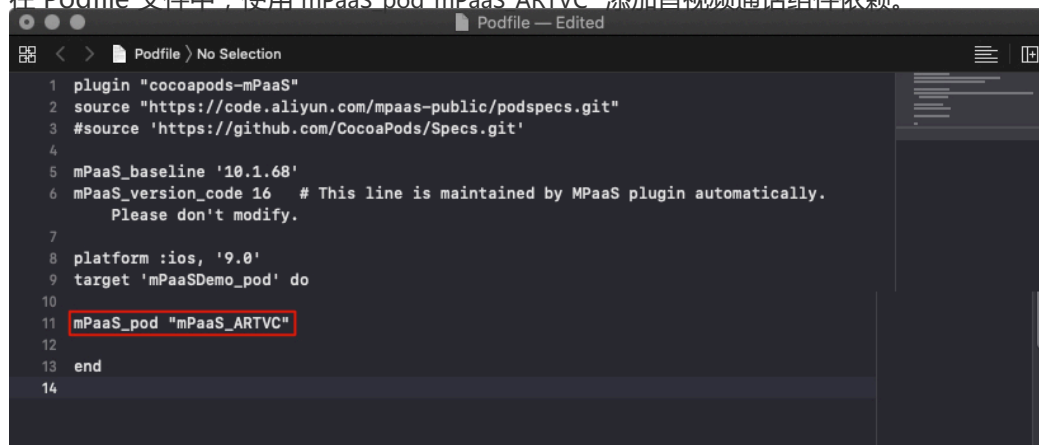
- 使用 mPaaS Xcode Extension。
此方式适用于采用了 基于 mPaaS 框架接入 或 基于已有工程且使用 mPaaS 插件接入 的接入方式。
 - 点击 Xcode 菜单项 **Editor > mPaaS > 编辑工程**，打开编辑工程页面。
 - 选择 **视频通话**，保存后点击 **开始编辑**，即可完成添加。



- 使用 cocoapods-mPaaS 插件。

此方式适用于采用了 **基于已有工程且使用 CocoaPods 接入** 的接入方式。

- 在 Podfile 文件中，使用 `mPaaS pod "mPaaS_ARTVC"` 添加音视频通话组件依赖。



- 执行 `pod install` 即可完成接入。

3.2.2 使用 iOS SDK

初始化 ARTVCEngine 并监听事件回调

- 初始化设置 uid 和 delegate。

```
_artvcEgnine = [[ARTVCEngine alloc] init];
_artvcEgnine.uid = self.uid;
_artvcEgnine.delegate = self;
```

- 监听事件回调（按需实现回调，比如错误处理是需要实现的）。

```
#pragma mark - ARTVCEngineDelegate
-(void)didReceiveRoomInfo:(ARTVCRoomInfo*)roomInfo{
}
-(void)didReceiveLocalFeed:(ARTVCFeed*)localFeed{
    self.feedForPreview = localFeed;
}
-(void)didEncounterError:(NSError *)error forFeed:(ARTVCFeed*)feed{
    [self showToastWith:[NSString stringWithFormat:@"%@", Error:%@",feed,error] duration:2.0];
}
.....
```

设置视频编码分辨率

设置视频编码分辨率代码如下：

```
//设置视频编码分辨率，默认是 ARTVCVideoProfileType_640x360_15Fps。
artvcEgnine.videoProfileType = ARTVCVideoProfileType_640x360_15Fps;
```

设置自动/手动推拉流开关

设置自动/手动推拉流开关代码如下，默认为自动推流/拉流。

```
_artvcEngine.autoPublish = YES;  
_artvcEngine.autoSubscribe = YES;
```

设置是音视频通话还是纯音频通话

- 音视频通话

```
ARTVCPublishConfig* config = [[ARTVCPublishConfig alloc] init];  
config.videoEnable = YES;//默认是 YES  
config.audioEnable = YES;//默认是 YES  
config.videoProfile = _artvcEngine.videoProfileType;  
_artvcEngine.autoPublishConfig = config;  
ARTVCSubscribeOptions* options = [[ARTVCSubscribeOptions alloc] init];  
_artvcEngine.autoSubscribeOptions = options;
```

- 纯音频通话

```
ARTVCPublishConfig* config = [[ARTVCPublishConfig alloc] init];  
config.videoEnable = NO;  
config.audioEnable = YES;//默认是 YES  
_artvcEngine.autoPublishConfig = config;  
ARTVCSubscribeOptions* options = [[ARTVCSubscribeOptions alloc] init];  
options.receiveVideo = NO;  
_artvcEngine.autoSubscribeOptions = options;
```

音视频通话下启动相机预览

对于纯音频通话，跳过此步骤。

```
//默认使用前置摄像头，如果设置为 YES 则使用后置摄像头。  
[_artvcEngine startCameraPreviewUsingBackCamera:NO];
```

- 启动相机预览后，如果本地 feed 没有被回调过。之后回调会返回一个 ARTVCFeed 对象，可用于关联后续返回的渲染 view。

```
-(void)didReceiveLocalFeed:(ARTVCFeed*)localFeed forPublishConfig:(ARTVCPublishConfig*)publishConfig{  
    self.feedForPreview = localFeed;  
}
```

- 预览 view 初始化的回调。

```

-(void)didVideoRenderViewInitialized:(UIView*)renderView forFeed:(ARTVCFeed*)feed{
    if([feed isEqual:self.feedForPreview]){
        [self showToastWith:@"video preview view created"duration:1.0];
    }else{
        self.feedForRemote = feed;
    };
    //可触发 UI 布局，把 renderView add 到 view 层级中去
}

```

- 预览首帧渲染的回调。

```

-(void)didFirstVideoFrameRendered:(UIView*)renderView forFeed:(ARTVCFeed*)feed{
}

```

创建或者加入房间

1. 作为主叫方，创建房间。

```

ARTVCCreateRoomParams* params = [[ARTVCCreateRoomParams alloc] init];
params.uid = self.uid;
params.bizName = DEMO_BIZ;
params.subBiz = DEMO_SUBBIZ;
params.signature = DEMO_SIGNATURE;
[_artvcEngine createRoom:params];

```

- 创建房间成功，会有房间信息回调。

```

-(void)didReceiveRoomInfo:(ARTVCRoomInfomation*)roomInfo{
}

```

- 创建房间失败，会有 Error 回调。

```

//error.code == ARTVCErrorCodeProtocolErrorCreateRoomFailed
-(void)didEncounterError:(NSError *)error forFeed:(ARTVCFeed*)feed{
}

```

- 其他人加入房间后，会有成员加入房间的回调。

```

-(void)didParticepantsEntered:(NSArray<ARTVCParticipantInfo*>*)participants{
}

```

2. 作为主被叫方，加入房间。


```
ARTVCJoinRoomParams* params = [[ARTVCJoinRoomParams alloc] init];
params.uid = self.uid;
params.bizName = DEMO_BIZ;
params.subBiz = DEMO_SUBBIZ;
params.roomId = self.roomId;
params.signature = DEMO_SIGNATURE;
params.rtoken = self.rtoken;
[_artvcEgine joinRoom:params];
```

- 加入房间成功，会有加入房间成功的回调以及房间已有成员的回调。

```
-(void)didJoinroomSuccess{
}
-(void)didParticepantsEntered:(NSArray<ARTVCParticipantInfo*>*)participants{
}
```

- 加入房间失败，会有 error 回调。

```
//error.code == ARTVCErrorCodeProtocolErrorJoinRoomFailed
-(void)didEncounterError:(NSError *)error forFeed:(ARTVCFeed*)feed{
}
```

- 后续其他人加入房间后，会有成员加入房间的回调。

```
-(void)didParticepantsEntered:(NSArray<ARTVCParticipantInfo*>*)participants{
}
```

创建或者加入房间成功后开始推流与拉流

默认是自动推流与拉流。

- 推流/拉流过程中，有如下相关状态回调。

```
-(void)didConnectionStatusChangedTo:(ARTVCConnectionStatus)status forFeed:(ARTVCFeed*)feed{
[self showToastWith:[NSString stringWithFormat:@"connection status:%d\nfeed:%@",status,feed]
duration:1.0];
if((status == ARTVCConnectionStatusClosed) && [feed.uid isEqualToString:[self uid]]){
[self.artvcEgine stopCameraPreview];//音视频通话下，停止摄像头。
[self.artvcEgine leaveRoom];
}
}
```

- ARTVCConnectionStatus 状态解释。

枚举	值	解释
ARTVCConnectionStatusC	2	开始发布和订阅时，首先回调该状态。

connecting	00	
ARTVCCConnectionStatusConnected	202	发布/订阅成功时回调该状态。
ARTVCCConnectionStatusDisConnected	203	出现闪断，底层媒体流断开，底层会做自动重连。业务拿到这个回调仅做用户提示，不需要做 leaveRoom 处理。
ARTVCCConnectionStatusFailed	204	底层 ICE 失败无法继续，是一个终极错误。业务拿到这个回调可做停止摄像头 leaveRoom 处理。
ARTVCCConnectionStatusClosed	206	每个发布和订阅结束时，回调该状态。它是最后的状态，业务拿到这个回调可做停止摄像头 leaveRoom 处理。

- 推流成功后，其他房间成员会收到新 feed 的回调。

```
-(void)didNewFeedAdded:(ARTVCFeed*)feed{
    [self showToastWith:[NSString stringWithFormat:@"new feed published by others:%@",feed] duration:2.0];
}
```

- 自动订阅模式下，会主动订阅该 feed。
- 订阅成功后，房间其他成员会收到如下回调。

```
-(void)didSubscriber:(NSString*)subscriber subscribedAFeed:(ARTVCFeed*)feed{
    [self showToastWith:[NSString stringWithFormat:@"subscriber subscribed :%@",feed] duration:2.0];
}
```

通话结束

- 音视频通话停止摄像头并离开房间。

```
[artvcEgnine stopCameraPreview];
[artvcEgnine leaveRoom];
```

- 纯音频通话离开房间。

```
[artvcEgnine leaveRoom];
```

- 成员离开房间后，房间其他成员会收到成员离开的回调。

```
-(void)didParticepant:(ARTVCParticipantInfo*)participant
leaveRoomWithReason:(ARTVCParticipantLeaveRoomReasonType)reason{
    [self showToastWith:[NSString stringWithFormat:@"participant left:%@ reason:%d",participant,reason]
```

```
duration:2.0];
}
```

- 自动发布订阅模式下离开房间，会自动取消发布本地的流以及取消订阅曾订阅过的流。
- 取消发布后，房间其他成员会收到取消发布的回调。

```
-(void)didFeedRemoved:(ARTVCFeed*)feed{
[self showToastWith:[NSString stringWithFormat:@"feed unpublished by others:%@",feed] duration:2.0];
}
```

- 取消订阅后，房间其他成员会收到取消订阅的回调。

```
-(void)didSubscriber:(NSString*)subscriber unsubscribedAFeed:(ARTVCFeed*)feed{
[self showToastWith:[NSString stringWithFormat:@"subscriber unsubscribed :%@",feed] duration:2.0];
}
```

3.2.3 iOS 进阶功能

渲染 View 相关

实现 ARTVCEngineDelegate 中部分和渲染相关的回调接口。

1. 渲染 View 对象被创建。

```
//渲染 View 对象被创建，同时 feed 相关联，业务收到此回调时可将此 view 加入到布局中，//并设置其 frame。
-(void)didVideoRenderViewInitialized:(UIView*)renderView forFeed:(ARTVCFeed*)feed{
}
```

2. 渲染首帧视频。

```
-(void)didFirstVideoFrameRendered:(UIView*)renderView forFeed:(ARTVCFeed*)feed{
}
```

3. 停止渲染视频。

```
-(void)didVideoViewRenderStopped:(UIView*)renderView forFeed:(ARTVCFeed*)feed{
}
```

统一错误处理

- 所有错误。通过统一的错误回调接口回调业务，业务根据不同的错误码，进行相应的错误处理。

```

-(void)didEncounterError:(NSError *)error forFeed:(ARTVCFeed*)feed{
//业务根据不同错误码，进行错误处理。
}

```

- 错误定义如下：

```

typedef NS_ENUM(int,ARTVCErrorCode){
/**
bad parameters passed to API
*/
ARTVCErrorCodeBadParameters = - 103,
/**
camera permission is denied by user
without this permission,video call cann't be continued,please advise user enable camera permission in settings.
*/
ARTVCErrorCodeCameraPermissionNOtAllowed = -104,
/**
microphone permission is denied by user
without this permission,video call cann't be continued,please advise user enable camera permission in settings.
*/
ARTVCErrorCodeMicrophonePermissionNOtAllowed = -105,
/**
timeout happened,publish/subscribe cann't be finished successfully
*/
ARTVCErrorCodeTimeout = -108,
/**
you has already published or subscribed a feed .
you cann't publish or subscribe the same feed once again when it has NOt been unpublished or unsubscribed.
*/
ARTVCErrorCodeAlreadyPublishedOrSubsribed = -111,
/**
you has NOt published or subscribed the feed .so you cann't do unpublish or unsubscribe operation.
*/
ARTVCErrorCodeFeedHasNOtBeenPublishedOrSubsribed = -119,
/**
internal webrtc-relative error when doing publish/subscribe,for example,setting sdp failed,creating sdp failed,e.g.
*/
ARTVCErrorCodeInternalError = -113,
/**
* current room has become invalid .most of all,it's because network's down,heartbeat abNormal.
* if you wanna continue,you MUST call createRoom again to get a new valid room.
*/
ARTVCErrorCodeCurrentRoomHasBecomeInvalid = -114,
/**
server error hanppened.CreateRoom request failed,it's a server internal error.
*/
ARTVCErrorCodeProtocolErrorCreateRoomFailed = -115,
/**
server error hanppened.JoinRoom request failed,it's a server internal error.maybe the room you joined has been
became invalid yet.
*/
ARTVCErrorCodeProtocolErrorJoinRoomFailed = -116,
/**
server error hanppened.Publish request failed,it's a server internal error.

```

```

*/
ARTVCErrCodeProtocolErrorPublishFailed = -117,
/**
server error hanppened.Subscribe request failed,it's a server internal error.maybe the stream you subscribed has
been unpunished yet or some error else.
*/
ARTVCErrCodeProtocolErrorSubscribeFailed = -118,
};

```

切换摄像头

切换摄像头代码如下：

```
[_artvcEgnie switchCamera];
```

mute 远端视频

mute 远端视频代码如下：

```

ARTVCFeed* feed = 要 mute 操作的远端 feed;
[_artvcEgnie muteRemoteVideo:YES forFeed:feed];

```

麦克风静音

麦克风静音代码如下：

```
[_artvcEgnie muteMicrophone:YES];
```

mute 远端音频

mute 远端音频代码如下：

```

ARTVCFeed* feed = 要 mute 操作的远端 feed;
[_artvcEgnie muteRemoteAudio:YES forFeed:feed];

```

听筒扬声器模式

- 切换模式：

```
[_artvcEgnie switchAudioPlayModeTo:ARTVCAudioPlayModeReceiver complete:nil];
```

变化通知：

```

-(void)didAudioPlayModeChangedTo:(ARTVCAudioPlayMode)audioPlayMode{
NSString *toast = nil;

```

```
switch (audioPlayMode) {
case ARTVCAudioPlayModeSpeaker:{
toast = @"扬声器模式";
}
break;
case ARTVCAudioPlayModeReceiver:{
toast = @"听筒模式";
}
break;
case ARTVCAudioPlayModeHeadphone:{
toast = @"耳机模式";
}
break;
case ARTVCAudioPlayModeBluetooth:{
toast = @"蓝牙设备模式";
}
break;
case ARTVCAudioPlayModeInit:{
toast = @"未知模式";
}
break;
}

[self showToastWith:toast duration:2.0];
}
```

网络变化通知

网络变化通知代码如下，对于移动网络，业务可做弹窗提示用户有流量风险。

```
-(void)didNetworkChangedTo:(APMNetworkReachabilityStatus)netStatus{
if(netStatus == APMNetReachabilityStatusReachableViaWiFi){
return ;
}
[self showToastWith:[NSString stringWithFormat:@"网络切换到:%@",[APMNetworkStatusManager
stringOfNetworkStatus:netStatus]] duration:2.0];
}
```

带宽不足通知

带宽不足通知如下：

```
-(void)didAvailabeSendBandwidthBecomeLow:(BOOL)isLow currentBandwidth:(double)bw
forFeed:(ARTVCFeed*)feed{
if(isLow){
[self showToastWith:@"当前通话质量不佳" duration:2.0];
}
}
```

截屏功能

截屏功能代码如下，可对任意流进行截屏。

```
ARTVCFeed* feed = 要截屏的feed;
[artvcEgnie snapshotForFeed:feed complete:^(UIImage* image){
//根据需要对截屏图片进行处理
}];
```

获取通话质量的 Debug 信息

通过回调来返回任意流的 Debug 信息。

```
/**
brief debug information is generated(including bitrate/cpu/codec,e.g.),you can show this on your debug
information view.
*/
- (void)didBriefDebugInformationGenerated:(NSString*)debugInfo forFeed:(ARTVCFeed*)feed{
//如果有需要，可以将 debugInfo 按流展示在 debug 窗口。
}
```

获取通话质量的实时监控信息

- 通过回调可获取任意流的实时监控信息，包括码率、帧率、CPU 等。

```
-(void)didRealtimeStatisticGenerated:(ARTVCRealtimeStatisticSummary*)summary
forFeed:(ARTVCFeed*)feed{
}
```

- 返回的数据内容如下：

```
@interface ARTVCRealtimeStatisticSummary : NSObject
//connection stats googCandidatePair
/** 实际总发码率 单位bps*/
@property(NOnatomic,copy) NSString* totalSendBitrate;
/** 实际总的收码率 单位bps*/
@property(NOnatomic,copy) NSString* totalRecvBitrate;
/** 网络延迟 ( 毫秒 ) */
/** 网络延迟 ( 毫秒 ) */
@property(NOnatomic,copy) NSString* rtt;
//video send
/** 视频码率发 单位bps*/
@property(NOnatomic,copy) NSString* videoSendBitrate;
/** 实际视频发帧率 */
@property(NOnatomic,copy) NSString* videoSendFps;
//video recv
/** 视频码率收 单位bps*/
@property(NOnatomic,copy) NSString* videoRecvBitrate;
/** 实际视频收帧率 */
@property(NOnatomic,copy) NSString* videoRecvFps;
/** 声音码率发 单位bps*/
@property(NOnatomic,copy) NSString* audioSendBitrate;
/** 声音码率收 单位bps*/
```

```
@property(NOnatomic,copy) NSString* audioRecvBitrate;
/** 视频发送丢包率*/
@property(NOnatomic,copy) NSString* videoLossRate;
//audio send
/** 音频发送丢包率*/
@property(NOnatomic,copy) NSString* audioLossRate;
/** cpu */
@property(NOnatomic,copy) NSString* cpu;
@end
```

通话过程中动态调整编码分辨率和相机 FPS (仅使用于内置 camera)

- 当前分辨率以及 FPS 分别为 640x360、15FPS 时，设置如下：

```
_artvcEgine.videoProfileType = ARTVCVideoProfileType_640x360_15Fps;
```

- 假设，因业务需求需要将分辨率以及 FPS 临时调高至 960x540、30FPS，设置如下：

```
_artvcEgine.videoProfileType = ARTVCVideoProfileType_960x540_30Fps;
```

- 处理完特定业务需求后，恢复至 640x360，15FPS。

```
_artvcEgine.videoProfileType = ARTVCVideoProfileType_640x360_15Fps;
```

通话过程中动态调整编码分辨率 (适应于所有 video source)

- 自定义推流中，当前分辨率为 640x360，若因业务需求需要临时调高至 960x540，设置如下：

```
[ _artvcEgine changeVideoProfileTo:ARTVCVideoProfileType_960x540_15Fps
forVideoSource:ARTVCVideoSourceType_Custom];
```

- 内置 camera，当前分辨率为 640x360，若因业务需求需要临时调高至 960x540，设置如下：

```
[ _artvcEgine changeVideoProfileTo:ARTVCVideoProfileType_960x540_30Fps
forVideoSource:ARTVCVideoSourceType_Camera];
```

自定义推视频流

自定义推视频流约束

- 只支持 NV12 格式的视频帧，以 CVPixelBufferRef 格式输入。
- 调用方目前需要保证 FPS，按照一定的 FPS 来输入。建议使用 15-24FPS。
- 只支持手动模式推流。

流程

自定义推视频流的流程如下：

```
//自定义推流中，必须设置为手动推流模式
[_artvcEgine setAutoPublish:NO];
//创建一个自定义推流类
ARTVCCreateCustomVideoCaputurerParams* params = [[ARTVCCreateCustomVideoCaputurerParams alloc] init];
//如果需要SDK帮忙渲染，需要设置provideRenderView为YES，SDK默认不帮业务渲染。
params.provideRenderView = YES;

self.customCapturer = [_artvcEgine createCustomVideoCapturer:params];
//按照一定的频率喂数据给SDK,格式是CVPixelBufferRef，只支持NV12格式。
[self.customCapturer provideCustomVideoFramePeriodlyWith:CVPixelBufferRef];
//主动调用推流
ARTVCPublishConfig* config = [[ARTVCPublishConfig alloc] init];
config.videoSource = ARTVCVideoSourceType_Custom;
config.videoProfile = ARTVCVideoProfileType_640x360_15Fps;
self.customPublishConfig = config;
[_artvcEgine publish:config];
```

屏幕共享

屏幕共享代码如下：

```
//必须设置为手动推流模式
[_artvcEgine setAutoPublish:NO];
//开始屏幕共享
-(void)startScreenSharing{
NSLog(@"start screen sharing");
ARTVCCreateScreenCaputurerParams* screenParams = [[ARTVCCreateScreenCaputurerParams alloc] init];
screenParams.provideRenderView = YES;
[_artvcEgine startScreenCaptureWithParams:screenParams complete:^(NSError* error){
NSLog(@"start screen sharing finish,error:%@",error);
if(error){
//这里的 code 见 启动屏幕捕捉的错误码 章节
}
}else{
//屏幕捕捉成功后再调用publish进行推流
ARTVCPublishConfig* config = [[ARTVCPublishConfig alloc] init];
config.videoSource = ARTVCVideoSourceType_Screen;
config.audioEnable = NO;
config.videoProfile = ARTVCVideoProfileType_1280x720_30Fps;
[_artvcEgine publish:config];
}
}
}
//停止屏幕共享
-(void)stopScreenSharing{
NSLog(@"stop screen sharing");
//停止屏幕捕捉
[_artvcEgine stopScreenCapture];
//取消发布屏幕共享流
ARTVCUnpublishConfig* config = [[ARTVCUnpublishConfig alloc] init];
config.feed = self.screenLocalFeed;
```

```
[_artvcEgnine unpublish:config];
}
```

启动屏幕捕捉的错误码

启动屏幕捕捉的错误码如下：

```
/**
screen capture already under running, you cann't start it again before you call stop
*/
ARTVCErrrorCodeScreenCapturerAlreadyUnderRunning = -1011,
/**
starting screen capture failed
*/
ARTVCErrrorCodeStartScreenCaptureFailed = -1012,
/**
start screen capture success,but may be encounter errors during the processing of the capure operation.
*/
ARTVCErrrorCodeScreenCaptureFailedInProcessing = -1013,
```

3.2.4 代码示例

下载代码示例

[点击这里](#) 获取音视频通话代码示例。

使用示例代码

- 在示例工程中添加配置文件。
 - 登录 mPaaS 控制台。
 - 创建 mPaaS 应用。
 - 下载 mPaaS 应用的 .config 格式 配置文件 放入音视频通话示例工程中。
 - 执行 `pod mpaas update 10.1.68 && pod install` 命令。
- 创建通话应用 并获取 bizName。
在控制台左侧导航栏选择 **音视频通话** > **通话应用管理**，创建通话应用或进入已有的通话应用，获取通话应用的 bizName。
- 在控制台 生成临时签名。
在控制台左侧导航栏选择 **音视频通话** > **签名校验**，在上方的 **生成临时签名** 区域，选择通话应用，输入 **UserID** 生成临时签名。

说明：

- 更换通话应用或 UserID 后需要重新生成临时签名。
 - 注意临时签名的有效期，过期后需重新生成。
- 配置示例工程。
运行示例工程，在 App 设置页填写从以上步骤获取的用户 ID (userId)、业务名称 (bizName) 和临时签名 (signature)。

说明：为保障您的流量安全，业务上线后务必 通过服务端生成签名。

5. 发起或加入音视频通话。

您可在 App 中创建通话房间发起音视频通话，或者在页面上输入其他用户创建的视频通话房间号（roomId）和 token 加入音视频通话。

说明：在同一个通话应用中（使用同一个 bizName）的不同端的 userId，必须互不相同。

3.3 接入 Web

前置条件

开发前的环境要求如下：

支持平台	浏览器	浏览器版本
Mac	Chrome	不低于 67
Windows	Chrome	不低于 67

说明：

- 集成 Web 客户端 SDK 必须使用 HTTPS 协议。
- 需尽快将 SDK 升级到 1.4.4 及以上版本，[点击这里](#) 下载最新版 SDK。

操作步骤

1. 下载 artvc-web-sdk。
2. 将 artvc-web-sdk 包保存到本地项目下。
3. 在项目相应的前端页面文件中，对 mcu.js 文件进行引用。
4. 实例化 SDK。

```
let test_controller = new McuController(); // 实例化 SDK
```

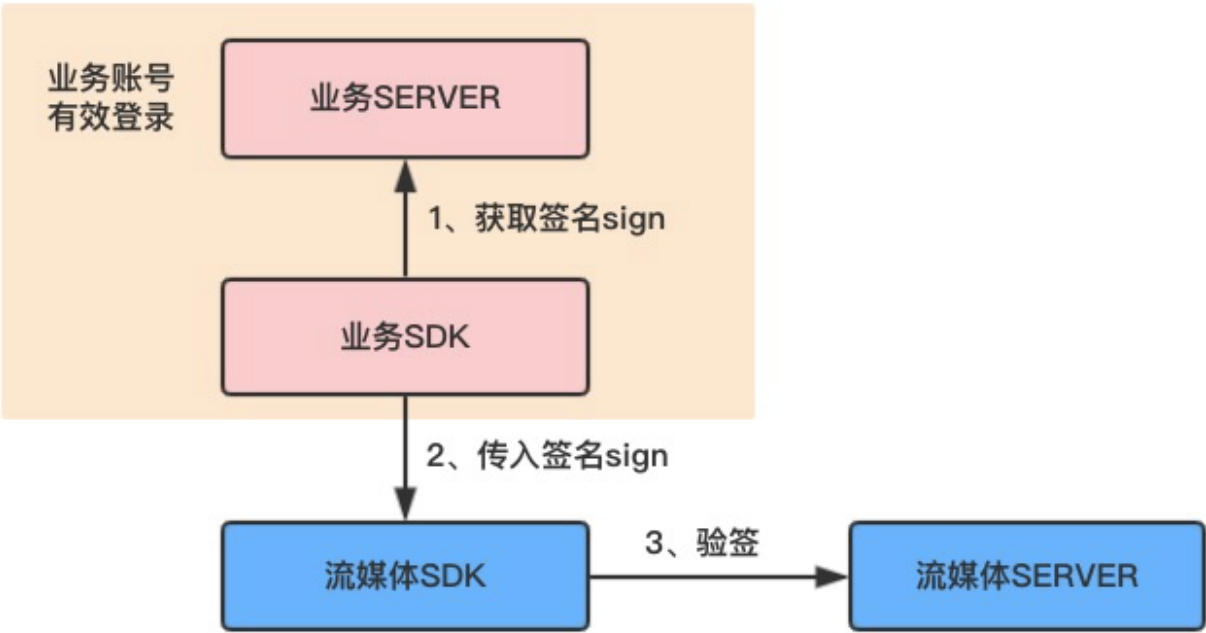
后续步骤

将音视频 SDK 接入 Web 端后，您可以实现音视频通信的功能，详情请参见 Web SDK API 说明。

4 安全加签

4.1 接入步骤

1. 创建 bizName 后，可以在控制台上获取对应密钥。
2. 在您自己的账号已登录的情况下，在您的服务端上按照规则使用密钥生成签名。
3. 客户端每次通话前，向业务服务端发送请求获取签名，然后传入流媒体 SDK。



4.2 代码示例

生成签名（示例）

生成签名代码示例如下：

```
bizName=demo
appId=default
workspaceId=default
uid=test1234
timestamp=1561014940059 ( 当前时间 + 有效期 N 毫秒 )
```

Java

安全加签的 Java 代码示例如下：

```
public class EncryptUtils {
    public static void main(String[] args) {
        // bizName
        String bizName = "demo";
        // appId
        String appId = "default";
        // workspaceId
        String workspaceId = "default";
        // uid
        String uid = "test1234";
        // 签名有效期(ms)，比如 5 分钟
        long expire = 5 * 60 * 1000L;
        // 签名生效截止时间: 当前时间 + 有效期
        long expireTime = System.currentTimeMillis() + expire;
        String str = bizName + appId + workspaceId + uid + expireTime;
```

```
// 分配的密钥
String key = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx";
// 生成签名
String sign = EncryptUtils.encryptByPrivate(str, EncryptUtils.getPrivateKey(key));
}
//RSA 私钥加密
public static String encryptByPrivate(String content, PrivateKey privateKey) throws Exception {
    Cipher cipher = Cipher.getInstance("RSA");
    cipher.init(Cipher.ENCRYPT_MODE, privateKey);
    return Base64.getEncoder().encodeToString(cipher.doFinal(content.getBytes("UTF-8")));
}
//将 base64 编码后的 RSA 私钥字符串转成 PrivateKey 实例
public static PrivateKey getPrivateKey(String privateKey) throws Exception {
    byte[] keyBytes = Base64.getDecoder().decode(privateKey);
    PKCS8EncodedKeySpec keySpec = new PKCS8EncodedKeySpec(keyBytes);
    KeyFactory keyFactory = KeyFactory.getInstance("RSA");
    return keyFactory.generatePrivate(keySpec);
}
}
```

5 使用控制台

5.1 通话应用管理

5.1.1 创建通话应用

在接入音视频通话 SDK 后，您可前往控制台创建通话应用，并对通话应用进行管理。

应用管理

创建通话应用

通话应用名称	bizName	创建时间	操作
		2020-06-16 10:14	查看密钥

共1条 < 1 >

操作步骤

登录控制台，在左侧导航栏中选择 **通话应用管理**。

点击页面右上方的 **创建通话应用**。

在弹出的窗口中，输入通话应用名称。

说明：创建通话应用时，后台会为通话应用自动生成 bizName 以及密钥。

创建通话应用

* 通话应用名称:

最多输入15个字符

取消 确定

4. 点击 确定，通话应用创建完成。

新创建的通话应用就会出现在应用列表中。

5.1.2 管理通话应用

在通话应用列表页面，您可查看通话应用的以下信息：

- 通话应用名称：创建应用时输入的名称。
- bizName：音视频通话场景码。使用同一个 bizName 的各个终端之间，可以相互进行音视频通话。
- 自动录制状态：在 功能配置 中可选择是否开启自动录制。
- 创建时间：当前通话应用的创建时间。
- 密钥：当前通话应用的密钥，并查看使用文档。

查看密钥

密钥使用说明 查看文档

密钥: [redacted]

功能配置

在通话应用列表页，点击应用右侧 操作 列中的 功能配置，可打开当前应用的功能配置页面。

云端自动录制

停用

开启后系统将进行自动录制

设置录制回调地址

请输入输入URL

保存 取消

- 云端自动录制**
开启该开关后，在音视频通话发起时服务端会自动对通话内容进行录制。通话结束时会同时结束录制，并生成录制文件。关闭该开关后，服务端不再自动录制。

说明：云端录制的默认状态为 **停用**，需手动开启。

- 设置录制回调地址**
录制回调地址为使用 HTTP 或 HTTPS 协议的 URL。在设置录制回调地址后，当开始录制、暂停录制、结束录制或录制文件出现异常时，服务端会将相关状态信息发送到该 URL。

录制信息回调

录制信息回调时请求数据字段如下：

字段	类型	必传	说明
bizRequestId	String	TRUE	请求id
bizName	String	TRUE	业务标识
appId	String	TRUE	mPaaS 应用的 id
workspaceId	String	TRUE	工作空间 id
roomId	String	TRUE	房间号id
recordId	String	TRUE	录制id
eventCode	int	TRUE	0：录制某条流成功
			10：初始化成功
			11：录制结束
			99：录制结果
			50：录制警告
			500301：流断开警告
			500302：低帧率警告
			1000xx：录制启动阶段失败
			100001：连接room失败
			100002：加入房间失败
			100003：订阅流失败

			100004 : 订阅路数缺失
			100005 : 连接mcu失败
			100006 : 视频长宽比异常
			100007 : 实际自定义混流数大于设置数目
			1001xx : 录制过程中失败
			100101 : 录制文件创建失败
			100102 : 录制文件格式转换失败
			100103 : 文件保存失败
			100104 : 音视频分离失败
			100105 : 录制子进程崩溃
			100106 : 磁盘满或者写文件失败
			1002xx : 录制结束阶段失败
			100201 : 上传文件失败

recordResult	json	eventCode=99	json body 如下 : • status : int , 录制状态 , 必传。2 : 录制成功 (持久化文件成功) ; 3 : 录制失败。 • fileType : int , 文件类型 , 当 status=2 时传。 1 : local ; 2 : OSS ; 3 : AFTS ; 4 : HDFS。 • filePath : string , 文件路径 , 当 status=2 时传。 • recordStartTime : long , 录制第一帧的绝对时间 (服务器时间 , ms) , 当 status=2 时传。 • mediaType : int , 录制文件的类型 , 当 status=2 时传。0 : 音视频 ; 1 : 纯音频 ; 2 : 纯视频。
--------------	------	--------------	--

业务服务器返回数据字段如下 :

字段	类型	必传	备注
bizRequestId	string	TRUE	回调请求中 bizRequestId
code	int	TRUE	状态码 , success=0

请求数据示例 :

```
请求 :
{
  "bizRequestId":"123456789",
  "bizName":"bizName",
  "appId":"appId",
  "workspaceId":"workspaceId",
  "roomId":"xxxxxx",
  "recordId":"record_xxx",
}
```



```
"eventCode":0,
"recordResult": {
  "status":2,
  "fileType":2,
  "filePath":"https://xxxxxx",
  "recordStartTime":1592817186122,
  "mediaType":0
}
}

返回数据示例：：
{
  "bizRequestId":"123456789",
  "code":0
}
```

5.2 签名校验

在左侧导航栏中选择 **签名校验**，可进入生成临时签名、验证签名的页面。

生成临时签名

在上方的 **生成临时签名** 区域，可对指定的 UserID 生成针对指定通话应用的临时签名。

生成临时签名

* 通话应用名称 (bizName) :

请选择通话应用

* UserID :

请输入UserID,最长64个字符

有效期 (TimeStamp) :

24小时

生成

- 1. 选择 **通话应用名称 (bizName)**，即已创建的通话应用。
- 2. 输入 **UserID**。
- 3. 选择 **TimeStamp**，即签名有效期。可选择 **24 小时**、**3 天** 或 **7 天**。
- 4. 点击 **生成**，即可在右侧区域展示生成的临时签名信息。

说明：如果生成临时签名区域中已有信息，您可点击该区域右上方的  来清空已有信息。

验证签名

在下方的 **验证签名** 区域，可验证上方生成的临时签名，也可以验证您在服务器中生成的签名。

验证签名

* 通话应用名称 (bizName) :

请选择通话应用

▼

* UserID :

请输入UserID,最长64个字符

* TimeStamp :

请输入TimeStamp

* 签名 (Signature) :

请输入完整签名

校验

1. 选择 **通话应用名称 (bizName)** ，即已创建的通话应用。
2. 输入 **UserID**。
3. 输入 **TimeStamp** ，即签名有效期。
4. 输入 **签名 (Signature)** ，可输入在上方生成临时签名区域生成的签名，也可输入您在服务器中生成的签名。
5. 点击 **校验** ，即可在右侧区域展示签名的校验结果。

说明：如果校验临时签名区域中已有信息，您可点击该区域右上方的  来清空已有信息。

6 API 参考

6.1 Android API

6.1.1 权限及隐私说明

权限申请

音视频通话功能需要以下三个权限：

- CAMERA：摄像头权限。
- RECORDAUDIO：麦克风权限。
- READPHONESTATE：监听系统的来电和去电广播权限，详见下方 隐私保护。

隐私保护

为了保护用户隐私，将在以下两种情况停止发送麦克风和摄像头数据，同时停止播放其他好友的视频和语音。

- 当用户拨打电话或者接听电话时（需要用户授予 READPHONESTATE 权限）。

说明：音视频通话需要用到上方提到的 READPHONESTATE 权限来监听系统的来电和去电广播。如果用户没有授予该权限，当拨打/接听电话时，不会终止摄像头和麦克风数据的发送，不利于隐私保护。

- 当应用切换至后台时。

当用户退出上述两种场景时，会继续发送麦克风和摄像头数据（如果有），同时继续播放其他好友的视频和语音。

6.1.2 重要参数

参数	说明
CreateRoomParams	创建房间。
JoinRoomParams	加入房间。
PublishConfig	发布媒体流。
UnpublishConfig	取消订阅。
SubscribeConfig	订阅媒体流。
UnsubscribeConfig	取消订阅媒体流。
LocalVideoInfo	本地视频信息，可用于暂停继续本地视频流的推送。
RemoteVideoInfo	远程视频信息，可以用于暂停继续远端视频流的渲染。
LocalAudioInfo	本地音频流信息，可以用于暂停继续本地音频流的推送。
RemoteAudioInfo	远端音频流信息，可以用于暂停继续远程音频的播放。
ParticipantInfo	视频通话参与者的用户信息，可用于订阅，发布等各种用户操作。
RoomInfo	创建房间返回的房间信息。
PublishEventCode	发布相关操作返回的事件码信息。
FeedInfo	可被订阅流的信息。
SubscribeEventCode	订阅相关操作返回的事件码信息。
Logger	业务可提供自定义 log，将 SDK 的日志输出到业务的 log 处。
ErrorCode	错误码。

CreateRoomParams

- **功能：**创建房间。
- **介绍：**作为创建房间的参数，只有该参数填写正确，才能顺利创建音视频房间。
- **属性列表：**

属性	类型	字段含义	推荐值	是否必填
----	----	------	-----	------

uid	String	用户 ID	-	是
bizName	String	业务名称，需要向服务提供方进行申请	-	是
signature	String	业务签名	-	是
extraInfo	String	额外信息，与业务相关	null	否

JoinRoomParams

- 功能：加入房间。
- 介绍：作为进房参数，只有该参数填写正确，才能顺利进入 roomId 所指定的音视频房间。
- 属性列表：

属性	类型	字段含义	推荐值	是否必填
uid	String	用户 ID	-	是
bizName	String	业务名称，需要向服务提供方进行申请	-	是
roomId	String	房间 ID	-	是
rtoken	String	进入房间所需要的 token	-	是
signature	String	业务签名	-	是
envType	int	服务提供类型，当前支持 alipay 和 aliyun alipay=0 (默认值) aliyun=1	0	否

PublishConfig

- 功能：发布媒体流。
- 介绍：配置需要发布媒体流的参数，包括音频和视频。
- 属性列表：

属性	类型	字段含义	推荐值	是否必填
videoSource	PublishVideoSource	待发布媒体流的视频源 VIDEO_SOURCE_CAMERA (默认值)	-	否
videoProfile	VideoProfile	视频属性配置 PROFILE_360_640P_15 (默认值)	PROFILE_540_960P_15	否
audioSource	PublishAudioSource	待发布媒体流的音频源 AUDIO_SOURCE_MIC (默认值)	-	否
videoCustomWidth	int	用户自定义视频输出 width 仅当 videoProfile = PROFILE_CUSTOM 时生效	-	否
VideoCustomHeight	int	用户自定义视频输出 height 仅当 videoProfile = PROFILE_CUSTOM 时生效	-	否
videoCustomFps	int	用户自定义视频输出 fps 仅当 videoProfile = PROFILE_CUSTOM 时生效	-	否
VideoCustomBitrate	int	用户自定义视频输出 bitrate 仅当 videoProfile = PROFILE_CUSTOM 时生效	-	否

说明：无论是横屏还是竖屏，640 × 360 还是 360 × 640，videoProfile 都选择 PROFILE_360_640P_15 或者 PROFILE_360_640_P30。

```
enum PublishVideoSource {
  VIDEO_SOURCE_CAMERA, //视频源是摄像头
  VIDEO_SOURCE_SCREEN, //视频源是屏幕共享
  VIDEO_SOURCE_CUSTOM, //视频源是自定义数据
  VIDEO_SOURCE_NULL //禁止输出视频
}
enum VideoProfile {
  PROFILE_360_640P_15, //分辨率360 * 640，帧率15
  PROFILE_360_640P_30, //分辨率360 * 640，帧率30
  PROFILE_540_960P_15, //分辨率540 * 960，帧率15
  PROFILE_540_960P_30, //分辨率540 * 960，帧率30
  PROFILE_720_1280P_15, //分辨率720 * 1280，帧率15
  PROFILE_720_1280P_30, //分辨率720 * 1280，帧率30
  PROFILE_1080_1920P_15, //分辨率1080 * 1920，帧率15
  PROFILE_1080_1920P_30, //分辨率1080 * 1920，帧率30
  PROFILED_EFAULT, //默认，分辨率360 * 640，帧率15
  PROFILE_CUSTOM //自定义模式，需要设置自定义视频属性
}
enum PublishAudioSource {
  AUDIO_SOURCE_MIC, //音频源是麦克风
  AUDIO_SOURCE_NULL //禁止输出音频
}
```

UnpublishConfig

- 功能：取消订阅。
- 介绍：配置需要取消发布的媒体流信息。
- 属性列表：

属性	类型	字段含义	推荐值	是否必填
videoSource	PublishVideoSource	待取消发布的媒体流视频源 VIDEO_SOURCE_CAMERA (默认值)	-	否
audioSource	PublishAudioSource	待取消发布媒体流的音频源 AUDIO_SOURCE_MIC (默认值)	-	否

SubscribeConfig

- 功能：订阅媒体流。
- 介绍：配置需要订阅的媒体流参数。
- 属性列表：

属性	类型	字段含义	推荐值	是否必填
info	FeedInfo	需要订阅的流信息，包含 uid 和流的 feedId，tag 信息	-	是

UnsubscribeConfig

- **功能**：取消订阅媒体流。
- **介绍**：配置需要取消订阅的媒体流参数。
- **属性列表**：

属性	类型	字段含义	推荐值	是否必填
info	FeedInfo	需要订阅的流信息，包含 uid 和流的 feedId，tag 信息	-	是

LocalVideoInfo

- **功能**：本地视频信息，可用于暂停继续本地视频流的推送。
- **介绍**：包含了视频源信息。
- **属性列表**：

属性	类型	字段含义	推荐值	是否必填
videoSource	PublishVideoSource	本地媒体流视频源信息 VIDEO_SOURCE_CAMERA（默认值）	-	否

RemoteVideoInfo

- **功能**：远程视频信息，可以用于暂停继续远端视频流的渲染。
- **介绍**：包含了远端视频流信息
- **属性列表**：

属性	类型	字段含义	推荐值	是否必填
feedId	String	远端媒体流 ID	-	是

LocalAudioInfo

- **功能**：本地音频流信息，可以用于暂停继续本地音频流的推送。
- **介绍**：包含了本地音频流源信息。
- **属性列表**：

属性	类型	字段含义	推荐值	是否必填
audioSource	PublishAudioSource	本地音频源信息 AUDIO_SOURCE_MIC（默认值）	-	否

RemoteAudioInfo

- **功能**：远端音频流信息，可以用于暂停继续远程音频的播放。
- **介绍**：包含了远程音频信息。
- **属性列表**：

属性	类型	字段含义	推荐值	是否必填
feedId	String	远端媒体流 ID	-	是

FeedInfo

- 功能：可被订阅流的信息。
- 介绍：一条可被订阅流的信息，包含了流所属用户的uid，流的feedId，流的额外属性tag
- 属性列表：

属性	类型	字段含义	推荐值	是否必填
uid	String	可被订阅流的所属用户 ID	-	是
feedId	String	可被订阅流的 ID	-	是
tag	String	可被订阅流的额外属性，目前是 PublishVideoSource 的字符串表现形式	-	否

ParticipantInfo

- 功能：视频通话参与者的用户信息，可用于订阅，发布等各种用户操作。
- 介绍：用户信息，包含用户 ID，用户发布的流信息，用户订阅的流信息。
- 属性列表：

属性	类型	字段含义
uid	String	用户 uid
feedList	List	用户推送的流信息，一个用户可以推送多路流供其他用户订阅
subscribeIdList	List	用户订阅的流信息，一个用户可以订阅多路流

RoomInfo

- 功能：创建房间返回的房间信息。
- 介绍：房间信息，通过该房间信息，可以加入对应的房间，进行音视频通话。
- 属性列表：

属性	类型	字段含义
roomId	String	房间 ID
rtoken	String	房间对应的 token

PublishEventCode

- 功能：发布相关操作返回的事件码信息。
- 介绍：在发布、取消发布后，返回的对应事件码信息。
- 属性列表：

属性	字段含义
PUBLISH_START	发布操作有效，服务器接受该发布请求，本地准备资源进行发布
PUBLISH_SUCCESS	发布操作成功，媒体流成功推送到服务器上
PUBLISH_FAIL	发布操作失败，可能是媒体流获取失败，也可能是媒体流推送失败
PUBLISH_DISCONNECT	发布断开，媒体流推送遇到错误，与服务器断开连接
UNPUBLISH_SUCCESS	取消发布成功
UNPUBLISH_FAIL	取消发布失败，参数有误或者服务器出错

SubscribeEventCode

- **功能：**订阅相关操作返回的事件码信息。
- **介绍：**在订阅，取消订阅后，返回的对应事件码信息。
- **属性列表：**

属性	字段含义
SUBSCRIBE_START	订阅操作有效，服务器接受该订阅请求，本地准备拉流操作
SUBSCRIBE_SUCCESS	订阅操作成功，成功从服务器拉取媒体流到本地
SUBSCRIBE_FAIL	订阅操作失败，可能是参数错误，也可能是服务器错误
SUBSCRIBE_DISCONNECT	订阅断开，媒体流推送遇到错误，与服务器断开连接
UNSUBSCRIBE_SUCCESS	取消订阅成功
UNSUBSCRIBE_FAIL	取消订阅失败，参数有误或者服务器出错

Logger

功能：业务可提供自定义 log，将 SDK 的日志输出到业务的 log 处，只需实现该 Logger 接口，然后通过 setLogger 设置到 SDK 即可。

```
interface Logger {
    int v(String tag, String msg);
    int d(String tag, String msg);
    int i(String tag, String msg);
    int w(String tag, String msg);
    int e(String tag, String msg);
    int e(String tag, String msg, Throwable tr);
}
```

ErrorCode

在 ErrorCode 类：

变量	值	说明
ERROR_NULL	0	无错误。
ERROR_CREATE_ROOM	-1	创建房间失败，可能参数为 null，或者参数在服务器端验证不通过，需要查阅参数正确配置后，再次创建房间。也可能是服务器内部处理出错（概率很低），如果遇到服务器内部错误，请通知服务提供

	15	商。
ERROR_JOIN_ROOM	-116	加入房间失败，可能参数为 null，或者参数在服务器端验证不通过，需要查阅参数正确配置后，再次加入房间。也可能是服务器内部处理出错（概率很低），如果遇到服务器内部错误，请通知服务提供商。
ERROR_LEAVE_ROOM	-1002	退出房间失败，可能当前不在房间内，或者服务器验证不通过。无论退出房间成功与否，都会释放所有的发布和订阅。
ERROR_INVITE_TO_JOIN_ROOM	-1003	邀请对方加入房间失败，可能参数为 null，或者参数在服务端验证不通过。
ERROR_NETWORK	-1004	网络错误，无法正常连接到服务器，此时，无法正常执行任何发布订阅等操作。
ERROR_CAMERA_PERMISSION	-104	打开相机预览失败，相机没有正确授权。
ERROR_MIC_PERMISSION	-105	打开相机预览失败，麦克风没有正确授权。
ERROR_OPEN_CAMERA	-1007	打开相机预览失败，无法正常打开摄像头。
ERROR_OPEN_MIC	-1008	打开相机预览失败，无法正常打开麦克风。

6.1.3 公共函数

接口类型	API	描述
基础接口	getInstance	创建 AlipayRtcEngine 实例（同一时间只会存在一个实例）。
	setRtcListenerAndHandler	设置回调事件的监听和对应的 Handler。
	setImListener	设置 IM 文本信息交互的 listener。
	setInviteListener	设置邀请模式下邀请信息 listener。
	setCustomPublishListener	设置自定义推流本地 view 交互 listener。
	destroy	销毁引擎。
房间相关接口	createRoom	创建房间。
	joinRoom	加入房间。

	leaveRoom	离开房间。
	setAutoPublishSubscribe	设置是否自动发布，是否自动订阅。
发布相关接口	isAutoPublish	查询当前是否为自动发布模式。
	configAutoPublish	设置是否允许发布视频流。
	getCurrentAutoPublishConfig	获取当前自动发布的配置。
	publish	手动发布设置的音频流和视频流。
	unpublish	取消发布流。
	updateVideoProfile	切换发布分辨率。
	pushCustomVideoData	自定义推流，推送自定义图像数据。
订阅相关接口	isAutoSubscribe	查询当前是否为自动订阅模式。
	subscribe	手动订阅视频和音频流。
	unsubscribe	取消订阅流。
视频相关接口	muteLocalVideo	设置是否暂停发布本地视频流。
	muteAllLocalVideo	设置是否暂停发布本地所有的视频流。
	switchCamera	切换前后摄像头。
	muteRemoteVideo	设置是否暂停播放远端视频流。
	muteAllRemoteVideo	设置是否暂停播放所有的远端视频流。
音频相关接口	muteLocalAudio	设置是否暂停发布本地音频。
	muteAllLocalAudio	设置是否暂停本地所有音频流。
	muteRemoteAudio	设置是否暂停播放远端音频流。
	muteAllRemoteAudio	设置是否暂停播放所有远端音频流。
	enableSpeaker	设置是否切换扬声器输出。
预览接口	startCameraPreview	开始本地预览。
	stopCameraPreview	停止本地预览。
查询远端用户接口	getRemoteUsers	获取远端用户信息。
其他接口	setLogger	设置自定义日志输出。
	invite	邀请对方加入。
	replyInvite	回复对方邀请。
	sendMessage	发送文本消息。
	snapshot	针对某路视频流进行截屏。
	onRequestPermissionsResult	如果需要处理权限申请，则业务需要将该权限申请结果通过该接口通知到 SDK 层进行对应的处理。
	onActivityResult	屏幕共享权限申请回调。

getInstance

- **声明**：public static AlipayRtcEngine getInstance(Context context)
- **说明**：
 - 创建 AlipayRtcEngine 实例（同一时间只会存在一个实例）。
 - 重要：由于涉及到权限申请，此处的 context 必须是 activity 的 context。
- **参数**：

参数	类型	说明
context	Context	上下文

- **返回值**：无。

setRtcListenerAndHandler

- **声明**：public void setRtcEngineEventListener(AlipayRtcEngineEventListener listener, Handler handler)
- **说明**：设置回调事件的监听。
- **参数**：

参数	类型	说明
listener	AlipayRtcEngineEventListener	接收回调事件的监听器。
handler	Handler	回调事件的 handler。 必须设置，不建议使用主线程的 handler，建议新建一个线程处理回调。 如果没有设置，则无法回调任何结果。

- **返回值**：无。

setImListener

- **声明**：public void setImListener(AlipayRtcEngineIMListener imListener)
- **说明**：设置文本信息交互的监听。
- **参数**：

参数	类型	说明
listener	AlipayRtcEngineIMListener	文本交互监听器。

- **返回值**：无。

setInviteListener

- **声明**：public void setInviteListener(AlipayRtcEngineInviteListener inviteListener)
- **说明**：设置邀请模式下的邀请信息监听。
- **参数**：

参数	类型	说明
----	----	----

listener	AlipayRtcEngineInviteListener	邀请模式下交互监听器。
----------	-------------------------------	-------------

- 返回值：无。

setCustomPublishListener

- 声明：public void setCustomPublishListener(AlipayRtcEngineCustomPublishListener listener)
- 说明：设置自定义推流模式下的本地预览 listener，当前包括屏幕共享和自定义推流。如果当前没有使用到自定义推流模式，可以不设置该参数。
- 参数：

参数	类型	说明
listener	AlipayRtcEngineCustomPublishListener	自定义推流的本地 view 交互 listener。

- 返回值：无。

destroy

- 声明：public void destroy()
- 说明：
 - 销毁引擎。
 - destroy 后，会退出房间，同时销毁本地引擎，再次使用功能，需要重新 getInstance() 初始化引擎。
- 参数：无。
- 返回值：无。

createRoom

- 声明：public void createRoom(CreateRoomParams params)
- 说明：
 - 创建房间，房间创建后可以通知其他人，邀请他们加入该房间。
 - 创建房间的结果由回调中的 void onRoomInfo(RoomInfo info) 返回，详情请参见 回调函数。
- 参数：

参数	类型	说明
params	CreateRoomParams	创建房间所需要的信息，详情请参见 重要参数。

- 返回值：无。

joinRoom

- **声明：** public void joinRoom(JoinRoomParams params)
- **说明：**
 - 加入房间，加入他人创建好的房间，若该房间失效，则加入失败。
 - 加入房间的结果由回调中的 void onEnterRoom(int result) 返回，详情请参见 回调函数 。
- **参数：**

参数	类型	说明
params	JoinRoomParams	加入房间所需要的信息，详情请参见 重要参数 。

- **返回值：**无。

leaveRoom

- **声明：** public void leaveRoom()
- **说明：**
 - 离开房间，如果是加入者（joinRoom 的方式加入），在离开房间后，可以再次加入，直到所有的与会者退出，房间没人后该房间被解散，无法再次进入。
 - 离开房间的结果由回调中的 void onLeaveRoom(int result) 返回，详情请参见 回调函数 。
- **参数：**无。
- **返回值：**无。

setAutoPublishSubscribe

- **声明：** public void setAutoPublishSubscribe(boolean autoPub, boolean autoSub)
- **说明：**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 createRoom 和 joinRoom 之前设置。
- **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。

- **返回值：**无。

isAutoPublish

- **声明：** public boolean isAutoPublish()
- **说明：**查询当前是否处于自动发布模式。
- **参数：**无。
- **返回值：**如果处于自动发布模式，返回 true，否则返回 false。

configAutoPublish

- **声明**：public void configAutoPublish(PublishConfig config)
- **说明**：配置自动发布音视频格式，如果没有配置，则使用默认配置进行发布，默认发布音视频，同时视频使用 PROFILE_360_640_P15 格式。
- **参数**：

参数	类型	说明
config	PublishConfig	发布的配置信息，详情请参见 重要参数。

- **返回值**：无。

getCurrentAutoPublishConfig

- **声明**：public PublishConfig getCurrentAutoPublishConfig()
- **说明**：获取当前自动发布的配置信息
- **参数**：无。
- **返回值**：PublishConfig，发布的配置信息，详情请参见 重要参数。

publish

- **声明**：public void publish(PublishConfig config)
- **说明**：
 - 发布媒体流。
 - 在没有设置自动发布的情况下，可以主动发布媒体流。
- **参数**：

参数	类型	说明
config	PublishConfig	本次发布媒体流的配置信息，详情请参见 重要参数。

- **返回值**：无。

unpublish

- **声明**：public void unpublish(UnpublishConfig config)
- **说明**：
 - 取消发布媒体流。
 - 用户在发布媒体流后，可以在期间取消发布，取消发布后，如果用户想要再次发布流，则需要调用 publish 接口进行重新发布。
- **参数**：

参数	类型	说明
----	----	----

config	UnpublishConfig	需要取消发布的流信息，详情请参见 重要参数。
--------	-----------------	------------------------

- 返回值：无。

updateVideoProfile

- 声明：public void updateVideoProfile(VideoProfile videoProfile, int maxBitrate, PublishVideoSource videoSource)
- 说明：用户在发布媒体流中间阶段，可以动态切换发布的分辨率。
- 参数：

参数	类型	说明
videoProfile	VideoProfile	目标分辨率的配置值，详情请参见 重要参数。
maxBitrate	int	目标分辨率对应的最大码率，如果设置为 0（或者 < 0），那么则使用默认的码率，否则，使用设置的最大的码率，单位为 kbps。
videoSource	PublishVideoSource	切换目标分辨率的媒体源。

- 返回值：无。

pushCustomVideoData

- 声明：public void pushCustomVideoData(byte[] bytes, int width, int height, int rotation)
- 说明：自定义推流图像数据输入，Android 仅支持 nv21 数据。
- 参数：

参数	类型	说明
bytes	byte[]	图像数据，NV21 格式
width	int	图像宽
height	int	图像高
rotation	int	图像旋转角度

- 返回值：无。

isAutoSubscribe

- 声明：public boolean isAutoSubscribe()
- 说明：查询是否处于自动订阅模式。
- 参数：无。
- 返回值：如果处于自动订阅模式，则返回 true，否则返回 false。

subscribe

- **声明**：public void subscribe(SubscribeConfig config)
- **说明**：订阅某路媒体流。
- **参数**：

参数	类型	说明
config	SubscribeConfig	被订阅流的配置信息，详情请参见 重要参数 。

- **返回值**：无。

unsubscribe

- **声明**：public void unsubscribe(UnsubscribeConfig config)
- **说明**：取消订阅某路媒体流。
- **参数**：

参数	类型	说明
config	UnsubscribeConfig	被订阅流的配置信息，详情请参见 重要参数 。

- **返回值**：无。

muteLocalVideo

- **声明**：public void muteLocalVideo(LocalVideoInfo info, boolean muted)
- **说明**：
 - 暂停本地视频流的发布。
 - 调用该接口，可以暂停继续本地视频流的推送，适用用临时暂停，后面继续的场景。
- **参数**：

参数	类型	说明
info	LocalVideoInfo	本地视频流信息，详情请参见 重要参数 。
muted	boolean	true，表示暂停本地视频推送。 false，表示继续本地视频推送

- **返回值**：无。

muteAllLocalVideo

- **声明**：public void muteAllLocalVideo(boolean muted)
- **说明**：
 - 暂停所有本地视频流的发布。
 - 当本地存在多路流需要同时暂停推送时，可以调用该接口进行暂停继续推送操作。

• 参数：

参数	类型	说明
muted	boolean	true，表示暂停所有本地视频流推送。 false，表示继续所有本地视频流推送

• 返回值：无。

switchCamera

- 声明：public void switchCamera()
- 说明：切换手机上的前后摄像头。
- 参数：无。
- 返回值：无。

muteRemoteVideo

- 声明：public void muteRemoteVideo(RemoteVideoInfo info, boolean muted)
- 说明：
 - 暂停远端视频流的订阅。
 - 调用该接口，可以暂停继续远端视频流的订阅，适用于临时暂停，后面继续的场景。
- 参数：

参数	类型	说明
info	RemoteVideoInfo	远端视频流信息，详情请参见 重要参数 。
muted	boolean	true，表示暂停远端视频流。 false，表示继续远端视频流

• 返回值：无。

muteAllRemoteVideo

- 声明：public void muteAllRemoteVideo(boolean muted)
- 说明：
 - 暂停所有远端的视频流。
 - 当本地存在多路流需要同时暂停订阅时，可以调用该接口进行操作。
- 参数：

参数	类型	说明
muted	boolean	true，表示暂停所有的远端视频流。 false，表示继续所有的远端视频流

- 返回值：无。

muteLocalAudio

- 声明：public void muteLocalAudio(LocalAudioInfo info, boolean muted)
- 说明：暂停本地音频流推送。
- 参数：

参数	类型	说明
info	LocalAudioInfo	本地音频流信息，详情请参见 重要参数 。 在音频时摄像头模式的情况下，可以填入null
muted	muted	true，表示暂停推送音频流。 false，表示继续推送音频流。

- 返回值：无。

muteAllLocalAudio

- 声明：public void muteAllLocalAudio(boolean muted)
- 说明：
 - 暂停所有本地音频流推送。
 - 当本地存在多路音频流推送时，可以使用该接口进行暂停继续操作。
- 参数：

参数	类型	说明
muted	boolean	true，表示暂停推送所有音频流。 false，表示继续推送所有音频流。

- 返回值：无。

muteRemoteAudio

- 声明：public void muteRemoteAudio(RemoteAudioInfo info, boolean muted)
- 说明：暂停远端音频流订阅。
- 参数：

参数	类型	说明
info	RemoteAudioInfo	远端音频流信息，详情请参见 重要参数 。
muted	boolean	true，表示暂停订阅音频流。 false，表示继续订阅音频流。

- 返回值：无。

muteAllRemoteAudio

- 声明：public void muteAllRemoteAudio(boolean muted)

- 说明：
 - 暂停远端所有音频流订阅。
 - 当订阅了远端多路音频流是，可以使用该接口进行暂停继续操作。
- 参数：

参数	类型	说明
muted	boolean	true，表示暂停所有音频流订阅。 false，表示继续所有音频流订阅

- 返回值：无。

enableSpeaker

- 声明：public void enableSpeaker(boolean enable)
- 说明：切换扬声器输出。
- 参数：

参数|类型|说明enable|boolean|true，切换至扬声器输出。
false，切换到非扬声器输出。

- 返回值：无。

startCameraPreview

- 声明：无。
- 说明：
 - 开始相机预览。
 - 相机预览需要 camera 权限，提前准备硬件资源。
- 参数：无。
- 返回值：无。

stopCameraPreview

- 声明：无。
- 说明：
 - 停止本地相机预览。
 - 相机预览停止后，会触发停止发布操作。如果想要再次发布，需先启动相机预览，成功方可进行发布操作。
- 参数：无。
- 返回值：无。

getRemoteUsers

- **声明**：public List<ParticipantInfo> getRemoteUsers()
- **说明**：获取房间中其他与会者的信息。
- **参数**：无。
- **返回值**：返回房间中其他成员列表，ParticipantInfo 详情请参见 重要参数。

setLogger

- **声明**：public void setLogger(Logger logger)
- **说明**：设置自定义日志输出。
- **参数**：

参数	类型	说明
logger	Logger	一个接口，用户使用自定义日志输出，需要实现该接口。

- **返回值**：无。

invite

- **声明**：public String invite(InviteParams params)
- **说明**：邀请对方加入。
- **参数**：

参数	类型	说明
params	InviteParams	邀请的配置信息，详情请参见 重要参数。

- **返回值**：String，本次邀请的 ID，后续的结果通知，以此 ID 进行标识。

replyInvite

- **声明**：public void replyInvite(ReplyOfInviteParam params)
- **说明**：回复对方邀请。
- **参数**：

参数	类型	说明
params	ReplyOfInviteParam	回复邀请的配置信息，详情请参见 重要参数。

- **返回值**：无。

sendMessage

- **声明**：public void sendMessage(Msg4Send msg4Send)
- **说明**：发送文本信息。

• 参数：

参数	类型	说明
msg4Send	Msg4Send	发送文本信息的配置信息，详情请参见 重要参数。

• 返回值：无。

snapshot

• 声明：public void snapshot(FeedInfo info)

• 说明：针对某路流进行截帧。

• 参数：

参数	类型	说明
info	FeedInfo	截帧流的信息，详情请参见 重要参数。

• 返回值：无。

onRequestPermissionsResult

• 声明：public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults)

• 说明：透传权限申请结果。

• 参数：

- 参考 Android 权限申请，直接将对应的结果透传到底层进行权限处理。
- 如果需要处理权限申请，则业务需要将该权限申请结果通过该接口通知到 SDK 层进行对应的处理。

• 返回值：无。

onActivityResult

• 声明：public void onActivityResult(int requestCode, int resultCode, Intent data)

• 说明：屏幕共享权限申请 activity 的结果回调。

• 参数：参考获取 Android activity 的结果回调，直接将结果透传到 SDK 底层进行处理。

• 返回值：无。

6.1.4 回调函数

回调类型	函数	说明
全局错误事件	void onError(int errorCode, String errorMessage, Bundle extra);	SDK 不可恢复的错误，一定要监听，并根据不同情况给用户适当的界面提示。 仅包含全局错误，不包括发布和订阅发生的错误。
房间事件	void onRoomInfo(RoomInfo info);	创建房间成功回调，包含房间信息，如果房间创建失败，则通过 onError 回调对应的错误。

	<code>void onEnterRoom(int result);</code>	加入房间回调，0 为成功，非 0 为失败，详情请参见 错误码。
	<code>void onLeaveRoom(int result);</code>	退出房间回调，0 为成功，非 0 为失败，详情请参见 错误码。
	<code>void onRecordInfo(String recordId)</code>	录制结果回调，返回录制 ID，通过该 ID 可以使用额外的查询接口进行录制文件操作。
预览事件	<code>void onCameraPreviewInfo(ARTVCView view);</code>	预览成功的回调，返回一个 view，业务可以将该 view 布局到界面上进行展示。如果预览失败，则通过 <code>onError</code> 返回错误信息。
	<code>void onCameraPreviewFirstFrame();</code>	预览首帧回调。
	<code>void onCameraPreviewStop();</code>	预览结束回调。
发布订阅事件	<code>void onPublishEvent(PublishConfig config, PublishEventCode eventCode, String eventDetail, FeedInfo feedInfo);</code>	发布相关事件通知，业务调用 <code>publish</code> 发布接口后，相关的结果从这里通知出去。
	<code>void onUnpublishEvent(UnpublishConfig config, PublishEventCode eventCode, String eventDetail);</code>	取消发布事件通知，业务调用 <code>unpublish</code> 取消发布接口后，相关的结果从这里通知出去。
	<code>void onSubscribeEvent(FeedInfo feedInfo, SubscribeEventCode eventCode, String eventDetail, ARTVCView view);</code>	订阅事件通知，业务调用 <code>subscribe</code> 订阅接口后，相关的结果从这里通知出去。渲染视频的 view 将从后续的第一帧数据的回调通知中返回。
	<code>void onUnsubscribeEvent(FeedInfo feedInfo, SubscribeEventCode eventCode, String eventDetail);</code>	取消订阅事件通知，当用户调用 <code>unsubscribe</code> 取消订阅某路媒体流时，相关结果从这里通知出去。
房间成员事件	<code>void onParticipantsEnter(List infos);</code>	用户进入房间通知。有新用户加入房间后，会将该用户的信息推送给房间的其他用户，同时，新用户也会通过该通知收到房间其他成员的信息。
	<code>void onParticipantsLeave(List<ParticipantLeaveInfo>uidList);</code>	用户退出房间通知。有用户退出房间时，会将该信息推送给房间的其他用户。
	<code>void onPublishNotify(FeedInfo publisherInfo);</code>	用户新发布通知。
	<code>void onUnpublishNotify(FeedInfo publisherInfo);</code>	用户取消发布通知。
	<code>void onSubscribeNotify(FeedInfo publisherInfo, ParticipantInfo subscriberInfo);</code>	用户订阅通知。
	<code>void onUnsubscribeNotify(FeedI</code>	用户取消订阅通知。

	nfo publisherInfo, ParticipantInfo subscribeInfo);	
视频开始渲染事件	void onRemoteViewFirstFrame(FeedInfo publisherInfo, ARTVCView view);	订阅的视频流开始渲染通知事件。
	void onRemoteViewStop(FeedInfo feedInfo, ARTVCView view);	订阅的视频流停止渲染通知事件。
当前网络状态	void onCurrentNetworkType(int type);	手机当前网络状态，当手机网络状态有变化的时候，会通过该回调通知到上层。同时，在用户通过 setRtcListenerAndHandler 接口设置整体回调的时候，也会通过该接口通知业务当前网络类型。
	void onBandwidthImportanceChangeNotify(boolean isLow, double currentBandwidth, FeedInfo feedInfo);	带宽变化通知，当带宽从正常变化到弱网，或者从弱网重新回到正常时，会回调该通知。
当前音频输出模式	void onCurrentAudioPlayoutMode(int mode);	手机当前声音输出模式，当手机声音输出模式有变化的时候，会通过该回调通知到上层。同时，在用户通过 setRtcListenerAndHandler 接口设置整体回调的时候，也会通过该接口通知业务当前声音输出模式。
截帧结果回调	void onSnapshotComplete(Bitmap image, FeedInfo feedInfo);	执行截帧后，在获取到结果后，会通过该回调通知。
调试信息回调	void onStatisticDebugInfo(StatisticInfoForDebug infoForDebug, FeedInfo feedInfo);	调试统计信息。
	void onRealTimeStatisticInfo(RealTimeStatisticReport report, FeedInfo feedInfo);	调试实时质量信息。
文本交互回调	void onMsgReceive(Msg4Receive msg4Receive);	接收到文本信息，详情请参见 重要参数 。
	void onMsgSend(int error, String errorMessage, long msgId);	文本发送结果。
邀请交互回调	void onInviteResponse(String inviteTaskId, int code, String msg);	发送邀请的结果，code=0 则发送成功。在 AlipayRtcEngineInviteListener 中，如果不使用邀请功能，可以不用设置和关注。
	void onReplyResponse(String inviteTaskId, int code, String msg);	回复邀请的结果，code=0 则回复成功。在 AlipayRtcEngineInviteListener 中，如果不使用邀请功能，可以不用设置和关注。
	void onReplyOfInviteNotify(ReplyOfInviteInfo replyInfo);	发送邀请之后，对方回复的结果，可以是接受或者拒绝之类的结果，详情请参见 重要参数 。在 AlipayRtcEngineInviteListener 中，如果不使用邀请功能，可以不用设置和关注。
自定义推流图像本地预览 view 交互回调	void onCustomPublishPreviewInfo(PublishVideoSource videoSource, ARTVCView view);	自定义推流的本地预览 view，通过 videoSource 区分视频源。在 AlipayRtcEngineCustomPublishListener 中，如果不使用自定义推流和屏幕共享，可以不用设置和关注。
	void onCustomPublishPreviewFirstFrame(PublishVideoSource	自定义推流的本地预览的第一帧显示事件，通过 videoSource 区分视频源。在 AlipayRtcEngineCustomPublishListener 中，如果不使用自定义推流和屏幕共享，可以不用设置和关注。

	videoSource);	自定义推流的本地预览停止显示事件，通过 videoSource 区分视频源。在 AlipayRtcEngineCustomPublishListener 中，如果不使用自定义推流和屏幕共享，可以不用设置和关注。
	void onCustomPublishPreviewStop(PublishVideoSource videoSource);	

onError

- **声明**：void onError(int errorCode, String errorMessage, Bundle extra);
- **说明**：
 - 错误回调：SDK 不可恢复的错误，必须监听，并按不同的毁掉结果给用户不同的界面提示。
 - 仅包含全局错误，不包括发布和订阅发生的错误。
- **参数**：

参数	类型	说明
errorCode	int	错误码，详情请参见附录错误码信息
errorMessage	String	错误信息
extra	Bundle	扩展信息字段，个别错误码可能会带额外的信息帮助定位问题。

- **返回值**：无。

onRoomInfo

- **声明**：void onRoomInfo(RoomInfo info);
- **说明**：创建房间成功回调，包含房间信息。如果房间创建失败，则通过 onError 回调对应的错误。
- **参数**：

参数	类型	说明
info	RoomInfo	房间信息，详情请参见 重要参数

- **返回值**：无。

onEnterRoom

- **声明**：void onEnterRoom(int result);
- **说明**：加入房间回调，0 为成功，非 0 为失败，详情请参见 错误码。
- **参数**：无。
- **返回值**：无。

onLeaveRoom

- **声明**：void onLeaveRoom(int result);

- **说明：**退出房间回调，0 为成功，非 0 为失败，详情请参见 错误码。
- **参数：**无。
- **返回值：**无。

void onRecordInfo(String recordId)

- **声明：**void onRecordInfo(String recordId);
- **说明：**录制结果回调，返回录制 ID，通过该 ID 可以使用额外的查询接口进行录制文件操作。
- **参数：**

参数	类型	说明
recordId	String	录制 ID

- **返回值：**无。

onCameraPreviewInfo

- **声明：**void onCameraPreviewInfo(ARTVCView view);
- **说明：**预览成功的回调，返回一个 view，业务可以将该 view 布局到界面上进行展示。如果预览失败，则通过 onError 返回错误信息。
- **参数：**

参数	类型	说明
view	ARTVCView	video 渲染控件

- **返回值：**无。

onCameraPreviewFirstFrame

- **声明：**void onCameraPreviewFirstFrame();
- **说明：**预览首帧回调。
- **参数：**无。
- **返回值：**无。

onCameraPreviewStop

- **声明：**void onCameraPreviewStop();
- **说明：**预览结束回调。
- **参数：**无。
- **返回值：**无。

onPublishEvent

- **声明**：void onPublishEvent(PublishConfig config, PublishEventCode eventCode, String eventDetail, FeedInfo feedInfo);
- **说明**：发布相关事件通知，业务调用 publish 发布接口后，相关的结果从这里通知出去。
- **参数**：

参数	类型	说明
config	PublishConfig	本次发布的配置信息
eventCode	PublishEventCode	事件码，详情请参见 重要参数
eventDetail	String	事件信息
feedInfo	FeedInfo	可以正常发布时，返回这个发布的 feedInfo 信息，即当 eventCode == PUBLISH_START 时，返回 feedInfo 信息，其它情况返回 null。

- **返回值**：无。

onUnpublishEvent

- **声明**：void onUnpublishEvent(UnpublishConfig config, PublishEventCode eventCode, String eventDetail);
- **说明**：取消发布事件通知，业务调用 unpublish 取消发布接口后，相关的结果从这里通知出去。
- **参数**：

参数	类型	说明
config	UnpublishConfig	本次取消发布的配置信息
eventCode	PublishEventCode	事件码，详情请参见 重要参数
eventDetail	String	事件信息

- **返回值**：无。

onSubscribeEvent

- **声明**：void onSubscribeEvent(FeedInfo feedInfo, SubscribeEventCode eventCode, String eventDetail, ARTVCView view);
- **说明**：订阅事件通知，业务调用 subscribe 订阅接口后，相关的结果从这里通知出去。渲染视频的 view 将从后续的第一帧数据的回调通知中返回。
- **参数**：

参数	类型	说明
feedInfo	FeedInfo	被订阅流的发布者信息，包含发布者 ID 和对应的流 ID 信息，详情请参见 重要参数。
eventCode	SubscribeEventCode	事件码，详情请参见 重要参数。
eventDetail	String	事件信息。

I		
view	ARTVCView	为 null。

- 返回值：无。

onUnsubscribeEvent

- 声明：void onUnsubscribeEvent(FeedInfo feedInfo, SubscribeEventCode eventCode, String eventDetail);
- 说明：取消订阅事件通知，当用户调用 unsubscribe 取消订阅某路媒体流时，相关结果从这里通知出去。
- 参数：

参数	类型	说明
feedInfo	FeedInfo	被订阅流的发布者信息，包含发布者 ID 和对应的流 ID 信息，详情请参见 重要参数。
eventCode	SubscribeEventCode	事件码，详情请参见 重要参数。
eventDetail	String	事件信息。

- 返回值：无。

onParticipantsEnter

- 声明：void onParticipantsEnter(List infos);
- 说明：用户进入房间通知。有新用户加入房间后，会将该用户的信息推送给房间的其他用户，同时，新用户也会通过该通知收到房间其他成员的信息。
- 参数：

参数	类型	说明
infos	List	成员数组信息，每个成员包含成员 ID，该成员所有的发布流和该成员所有的订阅流，详情请参见 重要参数

- 返回值：无。

onParticipantsLeave

- 声明：void onParticipantsLeave(List<ParticipantLeaveInfo> uidList);
- 说明：用户退出房间通知。有用户退出房间时，会将该信息推送给房间的其他用户。
- 参数：

参数	类型	说明
uidList	List<ParticipantLeaveInfo>	成员 uid 数组信息

- 返回值：无。

onPublishNotify

- 声明：void onPublishNotify(FeedInfo publisherInfo);
- 说明：用户新发布通知。
- 参数：

参数	类型	说明
publisherInfo	FeedInfo	新发布流的信息，包含发布者的 ID 和对应新发布流的流 ID 信息，详情请参见 重要参数

- 返回值：无。

onUnpublishNotify

- 声明：void onUnpublishNotify(FeedInfo publisherInfo);
- 说明：用户取消发布通知。
- 参数：

参数	类型	说明
publisherInfo	FeedInfo	取消发布流的信息，包含发布者的 ID 和对应取消发布流的流 ID 信息，详情请参见 重要参数

- 返回值：无。

onSubscribeNotify

- 声明：void onSubscribeNotify(FeedInfo publisherInfo, ParticipantInfo subscriberInfo);
- 说明：用户订阅通知。
- 参数：

参数	类型	说明
publisherInfo	FeedInfo	被订阅流的发布者信息，包含发布者 ID 和对应的流 ID 信息，详情请参见 重要参数
subscribeInfo	String	订阅者 uid 信息

- 返回值：无。

onUnsubscribeNotify

- 声明：void onUnsubscribeNotify(FeedInfo publisherInfo, ParticipantInfo subscribeInfo);
- 说明：用户取消订阅通知。
- 参数：

参数	类型	说明
----	----	----

publisherInfo	FeedInfo	被订阅流的发布者信息，包含发布者 ID 和对应的流 ID 信息，详情请参见 重要参数
subscribeInfo	String	订阅者 uid 信息

- 返回值：无。

onRemoteViewFirstFrame

- 声明：void onRemoteViewFirstFrame(FeedInfo publisherInfo);
- 说明：订阅的视频流开始渲染通知事件。
- 参数：

参数|类型|说明

publisherInfo|FeedInfo|被订阅流的发布者信息，包含发布者 ID 和对应的流 ID 信息，详情请参见 重要参数

- 返回值：无。

onRemoteViewStop

- 声明：void onRemoteViewStop(FeedInfo feedInfo, ARTVCView view);
- 说明：订阅的视频流停止渲染通知事件。
- 参数：

参数	类型	说明
feedInfo	FeedInfo	被订阅流的发布者信息，包含发布者ID 和对应的流 ID 信息，详情请参见 重要参数
view	ARTVCView	对应视频渲染的 view

- 返回值：无。

onCurrentNetworkType

- 声明：void onCurrentNetworkType(int type);
- 说明：手机当前网络状态。当手机网络状态有变化时，会通过该回调通知到上层。同时，在用户通过 setRtcListenerAndHandler 接口设置整体回调的时候，也会通过该接口通知业务当前网络类型。
- 参数：

参数	类型	说明
type	int	网络状态 0：无网络可用 1：Wi-Fi 网络 2：移动网络（该网络下，需要业务通知用户会消耗用户流量）

- 返回值：无。

onBandwidthImportanceChangeNotify

- **声明：** void onBandwidthImportanceChangeNotify(boolean isLow, double currentBandwidth, FeedInfo feedInfo);
- **说明：** 带宽变化通知，当带宽从正常变化到弱网，或者从弱网从新回到正常时，会回调该通知。
- **参数：**

参数	类型	说明
isLow	boolean	是否为弱网状态
currentBandwidth	double	当前有效带宽值
feedInfo	FeedInfo	带宽变化对应的流信息

- **返回值：** 无。

onCurrentAudioPlayoutMode

- **声明：** void onCurrentAudioPlayoutMode(int mode);
- **说明：** 手机当前声音输出模式，当手机声音输出模式有变化时，会通过该回调通知到上层。同时，在用户通过 setRtcListenerAndHandler 接口设置整体回调的时候，也会通过该接口通知业务当前声音输出模式。
- **参数：**

参数	类型	说明
mode	int	音频输出模式： 1：EARPIECE，听筒输出 2：SPEAKERPHONE，外放输出 3：WIREDHEADSET，耳机输出 4：BLUETOOTH，蓝牙输出 5：NONE，没有输出

- **返回值：** 无。

onSnapshotComplete

- **声明：** void onSnapshotComplete(Bitmap image, FeedInfo feedInfo);
- **说明：** 执行截帧后，在获取到结果后，会通过该回调通知。
- **参数：**

参数	类型	说明
image	Bitmap	截帧后的图像。
feedInfo	FeedInfo	截帧对应的流信息，可以是对方，也可以是本端。

- **返回值：** 无。

onStatisticDebugInfo

- **声明**：void onStatisticDebugInfo(StatisticInfoForDebug infoForDebug, FeedInfo feedInfo);
- **说明**：调试统计信息。
- **参数**：

参数	类型	说明
infoForDebug	StatisticInfoForDebug	统计信息，详情请参见 重要参数 。
feedInfo	FeedInfo	对应的流信息，可以是对方，也可以是本端。

- **返回值**：无。

onRealTimeStatisticInfo

- **声明**：void onRealTimeStatisticInfo(RealTimeStatisticReport report, FeedInfo feedInfo);
- **说明**：调试实时质量信息。
- **参数**：

参数	类型	说明
report	RealTimeStatisticReport	实时质量信息，详情请参见 重要参数 。
feedInfo	FeedInfo	对应的流信息，可以是对方，也可以是本端。

- **返回值**：无。

onMsgReceive

- **声明**：void onMsgReceive(Msg4Receive msg4Receive);
- **说明**：接收到文本信息，详情请参见 重要参数 。在 AlipayRtcEngineIMListener 回调中，如果不使用本文发送功能，则可以不用设置该回调。
- **参数**：

参数	类型	说明
msg4Receive	Msg4Receive	无

- **返回值**：无。

onMsgSend

- **声明**：void onMsgSend(int error, String errorMessage, long msgId);
- **说明**：文本发送结果。在 AlipayRtcEngineIMListener 回调中，如果不使用本文发送功能，则可以不用设置该回调。
- **参数**：

参数	类型	说明
----	----	----

error	int	0 为正常，其它则为失败。
errorMessage	String	如果 error 不为 0，则这里是对应的失败信息说明。
msgId	long	对应的消息 ID。

- 返回值：无。

onInviteResponse

- 声明：void onInviteResponse(String inviteTaskId, int code, String msg);
- 说明：发送邀请的结果，code=0 则发送成功。在 AlipayRtcEngineInviteListener 中，如果不使用邀请功能，可以不用设置和关注。
- 参数：

参数	类型	说明
inviteTaskId	String	无
code	int	无
msg	String	无

- 返回值：无。

onReplyResponse

- 声明：void onReplyResponse(String inviteTaskId, int code, String msg);
- 说明：回复邀请的结果，code=0 则回复成功。在 AlipayRtcEngineInviteListener 中，如果不使用邀请功能，可以不用设置和关注。
- 参数：

参数	类型	说明
inviteTaskId	String	无
code	int	无
msg	String	无

- 返回值：无。

onReplyOfInviteNotify

- 声明：void onReplyOfInviteNotify(ReplyOfInviteInfo replyInfo);
- 说明：发送邀请之后，对方回复的结果，可以是接受或者拒绝之类的结果，详情请参见 重要参数。在 AlipayRtcEngineInviteListener 中，如果不使用邀请功能，可以不用设置和关注。
- 参数：

参数	类型	说明
----	----	----

replyInfo	ReplyOfInviteInfo	无
-----------	-------------------	---

- 返回值：无。

onCustomPublishPreviewInfo

- 声明：void onCustomPublishPreviewInfo(PublishVideoSource videoSource, ARTVCView view);
- 说明：自定义推流的本地预览 view，通过 videoSource 区分视频源。在 AlipayRtcEngineCustomPublishListener 中，如果不使用自定义推流和屏幕共享，可以不用设置和关注。
- 参数：

参数	类型	说明
videoSource	PublishVideoSource	无
view	ARTVCView	无

- 返回值：无。

onCustomPublishPreviewFirstFrame

- 声明：void onCustomPublishPreviewFirstFrame(PublishVideoSource videoSource);
- 说明：自定义推流的本地预览的第一帧显示事件，通过 videoSource 区分视频源。在 AlipayRtcEngineCustomPublishListener 中，如果不使用自定义推流和屏幕共享，可以不用设置和关注。
- 参数：

参数	类型	说明
videoSource	PublishVideoSource	无

- 返回值：无。

onCustomPublishPreviewStop

- 声明：void onCustomPublishPreviewStop(PublishVideoSource videoSource);
- 说明：自定义推流的本地预览停止显示事件，通过 videoSource 区分视频源。在 AlipayRtcEngineCustomPublishListener 中，如果不使用自定义推流和屏幕共享，可以不用设置和关注。
- 参数：

参数	类型	说明
videoSource	PublishVideoSource	无

- 返回值：无。

6.1.5 错误码

错误码	错误信息
-103	参数错误
-104	相机权限错误，没有获得相机权限
-105	麦克风权限错误，没有获得麦克风权限
-106	获取手机状态权限失败（电话）
-108	超时错误
-114	房间失效
-115	创建房间失败
-116	加入房间失败
-117	发布失败
-118	订阅失败
-119	取消订阅或取消发布失败
-1002	退出房间失败
-1004	网络错误，无法连接服务器
-1005	room 地址设置无效
-1007	打开摄像头错误
-1008	打开麦克风错误
-1009	机器不支持，可能是其他 SO 库不支持
-1010	屏幕共享能力不支持，例如系统过低
-1011	屏幕共享正在进行中
-1012	屏幕共享启动失败
-1013	屏幕共享中系统出现异常，无法恢复

6.2 iOS API

6.2.1 重要参数

变量	说明
delegate	设置代理对象。
dynamicConfigProxy	动态配置代理。
uid	设置用户 id。
roomServerCustomUrl	设置服务器地址。
videoProfileType	设置编码分辨率。
autoPublish	设置自动发布。
autoPublishConfig	设置自动发布配置参数。
currentPublishConfig	获取自动订阅配置参数。
autoSubscribe	设置自动订阅。

autoSubscribeOptions	设置自动订阅配置参数。
----------------------	-------------

delegate

- **功能**：设置代理对象。
- **介绍**：通过该字段设置代理对象，weak 持有。只有先设置好 Delegate 才能收到底层事件回调。

dynamicConfigProxy

- **功能**：设置动态配置代理。
- **介绍**：通过该字段设置动态配置代理，可做一些策略动态配置。若没有动态配置需求，可不做设置。

uid

- **功能**：设置用户 ID。
- **介绍**：通过该字段设置用户 ID，调用 API 之前必须先设置好用户 ID。

roomServerCustomUrl

- **功能**：设置服务器地址。
- **介绍**：通过该字段设置服务器地址，调用 API 之前必须先设置好服务器地址。

videoProfileType

- **功能**：设置编码分辨率。
- **介绍**：通过该字段设置编码分辨率，调用 API 之前必须先设置好编码分辨率。

autoPublish

- **功能**：设置自动发布。
- **介绍**：通过该字段设置自动发布。可不做设置，默认是自动发布状态。

autoPublishConfig

- **功能**：设置自动发布参数。
- **介绍**：通过该字段设置自动发布配置参数，可设置发布音频、发布视频、编码分辨率 和超时等相关操作。可不做设置，底层有默认配置。

currentPublishConfig

- **功能**：获取自动订阅配置参数。

autoSubscribe

- **功能**：设置自动订阅。

- **介绍：**通过该字段设置自动订阅。可不做设置，默认是自动订阅。

autoSubscribeOptions

- **功能：**设置自动订阅配置参数。
- **介绍：**通过该字段设置自动订阅配置参数，可设置是否接受音频，接收视频，超时相关操作。可不做设置，底层有默认配置。

6.2.2 公共函数

接口类型	API	描述
房间相关接口	createRoom	创建视频通话房间。
	joinRoom	加入视频通话房间。
	leaveRoom	离开视频通话房间。
邀请应答接口	inviteWith	通知对方加入房间。
	replyWith	应答操作回复邀请动作。
发布订阅接口	publish	发布流。
	unpublish	取消发布流。
	subscribe	订阅流。
	unsubscribe	取消订阅流。
摄像头接口	startCameraPreviewUsingBackCamera	开启相机。
	stopCameraPreview	停止相机。
	switchCamera	切换相机摄像头。
静音接口	muteMicrophone	麦克风静音。
	muteRemoteAudio	静音单个远端音频。
	muteAllRemoteAudios	静音所有远端音频。
	muteRemoteVideo	静音单个远端视频。
	muteAllRemoteVideos	静音所有远端视频。
截图接口	snapshotForFeed	截图。
扬声器/听筒接口	currentAudioPlayMode	当前声音播放模式。
	switchAudioPlayModeTo	切换听筒扬声器模式。
IM 接口	sendMessage	发送 IM 到单个用户。
	sendMessage	发送 IM 到多个用户。
自定义推流接口	createCustomVideoCapturer	创建自定义推流类。
	destroyCustomVideoCapturer	销毁自定义推流类。
屏幕共享接口	isScreenCaptureSupported	是否支持屏幕捕捉。
	isScreenCaptureStarted	屏幕捕捉是否已启动。
	startScreenCaptureWithParams	开始屏幕捕捉。

	stopScreenCapture	结束屏幕捕捉。
--	-------------------	---------

createRoom

- **声明**：(void)createRoom:(ARTVCCreateRoomParams*) params
- **说明**：创建视频通话房间。
- **参数**：

参数	类型	说明	是否可为空
uid	NSString*	用户 uid	NO
bizName	NSString*	主业务名称	NO
signature	NSString*	签名	NO
extraInfo	NSDictionary*	扩展参数	YES

- **返回值**：无。

joinRoom

- **声明**：(void)joinRoom:(ARTVCJoinRoomParams*) params;
- **说明**：加入视频通话房间。
- **参数**：

参数	类型	说明	是否可为空
uid	NSString*	用户 uid	NO
bizName	NSString*	主业务名称	NO
signature	NSString*	签名	NO
roomId	NSString*	房间 ID	NO
rtoken	NSString*	房间 token	NO
extraInfo	NSDictionary*	扩展参数	YES

- **返回值**：无。

leaveRoom

- **声明**：(void)leaveRoom;
- **说明**：离开视频通话房间。
- **参数**：无。
- **返回值**：无。

inviteWith

- **声明**：(void)inviteWith:(ARTVCInviteParams*)params complete:(ARTVCInviteCallback)complete;

- **说明：**创建视频通话房间后，可调用该接口通知对方来加入房间。C2B 下可邀请 web 坐席。该接口暂不支持在 C2C 模式下调用。

参数：

- ARTVCInviteParams

参数	类型	说明	是否可为空	默认值
inviteId	NSString	邀请 ID，标志该次邀请。	NO	无
inviteeUid	NSString	对端 uid。	NO	无
inviteType	ARTVCInviteType	目前只支持 ARTVCInviteTypeWebsocket，被邀请方必须是 web 端。	NO	无
audioEnabled	BOOL	邀请对端开启音频。	NO	YES
videoEnabled	BOOL	邀请对端开启视频。	NO	YES
timeout	int	超时设置。	NO	60
extraInfo	NSDictionary*	扩展参数。	YES	无

- ARTVCInviteCallback

```
//error is nil means success,code see ARTVCInviteError
typedef void (^ARTVCInviteCallback)(NSError*_Nullable error);
```

- 错误码定义

```
typedef NS_ENUM(int,ARTVCInviteError){
    ARTVCInviteErrorBabParameters = 1,
    ARTVCInviteErrorNOtHaveValidRoom = 2,
    ARTVCInviteErrorPreviousInviteUnderProcessing = 3,
    ARTVCInviteErrorRequestFailed = 4,
    ARTVCInviteErrorTimeout = 5,
    ARTVCInviteErrorCanceledWhenLeaveRoom = 6,
};
```

返回值：无。

replyWith

- **声明：**-(void)replyWith:(ARTVCReplyParams*)params complete:(ARTVCReplyCallback)complete;
- **说明：**应答接口。

参数：

- ARTVCReplyParams

参数	类型	说明	是否可为空	默认值
uid	NSString	用户 uid	NO	无
inviterUid	NSString	邀请者 uid	NO	无
bizName	NSString	主业务名称	NO	无
roomId	NSString	房间 ID	NO	无
replyType	ARTVCReplyType	应答类型	NO	无
audioEnable	BOOL	开启音频	NO	YES
videoEnable	BOOL	开启视频	NO	YES
timeout	int	超时设置	NO	60
extraInfo	NSDictionary*	扩展参数	YES	无

- ARTVCReplyCallback

```
//error is nil means success,code see ARTVCReplyError
typedef void (^ARTVCReplyCallback)(NSError*_Nullable error);
```

- 错误码定义

```
typedef NS_ENUM(int,ARTVCReplyError){
    ARTVCReplyErrorBabParameters = 1,
    ARTVCReplyErrorPreviousReplyUnderProcessing = 3,
    ARTVCReplyErrorRequestFailed = 4,
    ARTVCReplyErrorTimeout = 5,
    ARTVCReplyErrorCanceledWhenLeaveRoom = 6,
};
```

返回值：无。

publish

- 声明：(void)publish:(ARTVCPublishConfig*)config
- 说明：发布流。
- 参数：
 - ARTVCPublishConfig

参数	类型	说明	默认值
videoEnable	BOOL	是否推视频。	YES

audioEnable	BOOL	是否推音频。	YES
videoSource	ARTVCVideoSourceType	视频源类型。	ARTVCVideoSourceType_Camera
videoProfile	ARTVCVideoProfileType	视频编码分辨率和 FPS。	ARTVCVideoProfileType640x36015Fps
videoCustomWidth	int	自定义视频宽，videoProfile 为 ARTVCVideoProfileType_Custom 时设置有效，不建议使用。	无
videoCustomHeight	int	自定义视频高，videoProfile 为 ARTVCVideoProfileType_Custom 时设置有效，不建议使用。	无
videoCustomFps	int	自定义 FPS，videoProfile 为 ARTVCVideoProfileType_Custom 时设置有效，不建议使用。	无
videoCustomBitrate	int	自定义码率，videoProfile 为 ARTVCVideoProfileType_Custom 时设置有效，不建议使用。	无
timeout	int	超时设置	60

- 返回值：无。

unpublish

- 声明：-(void)unpublish:(ARTVCUnpublishConfig*)config
- 说明：取消发布流。
- 参数：
- ARTVCUnpublishConfig

参数	类型	描述
feed	ARTVCFeed*	要取消发布的 Feed。

- ARTVCFeed

参数	类型	描述	是否可为空
uid	NSString*	Feed 归属的用户 uid。	NO
userIdType	ARTVCParticipantType	用户类型，默认是普通用户，ARTVCParticipantTypeVirtualVod 表示点播服务对应的虚拟用户。	NO
feedId	NSString*	feed 的标识 ID。	NO
tag	NSString*	feed 的标签类型。	YES

- 返回值：无。

subscribe

- 声明：-(void)subscribe:(ARTVCSubscribeConfig*)config

- 说明：订阅流。
- 参数：
- ARTVCSubscribeConfig

参数	类型	描述	是否可为空
feed	ARTVCFeed*	要订阅的 feed。	NO
options	ARTVCSubscribeOptions*	订阅参数。	NO

- ARTVCSubscribeOptions

参数	类型	描述	是否可为空
receiveAudio	BOOL	是否订阅音频流。	YES
receiveVideo	BOOL	是否订阅视频流。	YES
timeout	int	超时设置。	60

- 返回值：无。

unsubscribe

- 声明：(void)unsubscribe:(ARTVCUnsubscribeConfig*)config
- 说明：取消订阅流。
- 参数：
- ARTVCUnsubscribeConfig

参数	类型	描述
feed	ARTVCFeed*	要取消订阅的 feed。

- 返回值：无。

startCameraPreviewUsingBackCamera

- 声明：(void)startCameraPreviewUsingBackCamera:(BOOL)usingBackCamera
- 说明：开启相机。
- 参数：usingBackCamera 为 YES 时使用后置摄像头，否则使用前置摄像头。
- 返回值：无。

stopCameraPreview

- 声明：(void)stopCameraPreview
- 说明：停止使用相机。
- 参数：无。

- 返回值：无。

switchCamera

- 声明：(void)switchCamera
- 说明：切换相机摄像头。
- 参数：无。
- 返回值：无。

muteMicrophone

- 声明：(void)muteMicrophone:(BOOL)muted
- 说明：麦克风静音。
- 参数：muted 为 YES 时静音麦克风，否则取消静音麦克风。
- 返回值：无。

muteRemoteAudio

- 声明：(void)muteRemoteAudio:(BOOL)muted forFeed:(ARTVCFeed*)feed
- 说明：对单个远端音频静音。
- 参数：静音传入 feed 中的音频流。muted 为 YES 时静音，否则取消静音。feed 不能为空。
- 返回值：无。

muteAllRemoteAudios

- 声明：(void)muteAllRemoteAudios:(BOOL)muted
- 说明：静音所有远端音频。
- 参数：静音所有订阅 feed 中的音频流，muted 为 YES 时静音，否则取消静音。
- 返回值：无。

muteRemoteVideo

- 声明：(void)muteRemoteVideo:(BOOL)muted forFeed:(ARTVCFeed*)feed
- 说明：静音单个远端视频。
- 参数：静音传入 feed 中的视频流。Muted 为 YES 时做静音处理，否则取消。Feed 不能为空。
- 返回值：无。

muteAllRemoteVideos

- 声明：(void)muteAllRemoteVideos:(BOOL)muted
- 说明：静音所有远端视频。
- 参数：静音所有订阅 feed 中的视频流。muted 为 YES 时做静音处理，否则取消。
- 返回值：无。

snapshotForFeed

- **声明**：(void)snapshotForFeed:(ARTVCFeed*)feed complete:(void(^)(UIImage* image))complete
- **说明**：截图接口。截取视频帧返回 .jpg 格式的 UIImage，支持截取本地和对端任意 feed 的视频帧数据信息。
- **参数**：
 - Feed：见 unpublish 中的 ARTVCFeed。
 - complete：截图图片回调。若为空表示截图失败。
- **返回值**：无。

currentAudioPlayMode

- **声明**：(ARTVCAudioPlayMode)currentAudioPlayMode
- **说明**：当前声音的播放模式。
- **参数**：无。
- **返回值**：无。

switchAudioPlayModeTo

- **声明**：(void)switchAudioPlayModeTo:(ARTVCAudioPlayMode)audioPlayMode complete:(void(^_Nullable)(NSError*_Nullable error))callback
- **说明**：切换听筒扬声器模式。耳机模式下不作处理。
- **参数**：
 - ARTVCAudioPlayMode

```
/**
 * 声音播放模式
 */
typedef NS_ENUM(NSUInteger, ARTVCAudioPlayMode) {
    ARTVCAudioPlayModeInit = 0, // 初始化状态
    ARTVCAudioPlayModeReceiver, // 听筒
    ARTVCAudioPlayModeSpeaker, // 扬声器
    ARTVCAudioPlayModeHeadphone, // 耳机
    ARTVCAudioPlayModeBluetooth, // 蓝牙设备
};
```

- callback 回调解释：当 error 为空表示切换成功，否则切换失败。
- **返回值**：无。

sendMessage

- **声明**：(void)sendMessage:(ARTVCIMMessage*)message toParticipant:(NSString*)participant complete:(ARTVCIMCallback)complete

- 说明：发送 IM 到单个用户。
- 参数：

参数	类型	说明	是否可为空
message	ARTVCIMMessage*	IM 消息	NO
participant	NSString*	用户 uid	NO
complete	ARTVCIMCallback	消息发送结果回调	YES

- ARTVCIMMessage

参数	类型	说明	是否可为空
msg	NSString*	消息内容	NO
msgId	NSUInteger	消息 ID	YES
timestamp	NSTimeInterval	消息时间戳	YES

- ARTVCIMCallback

```
//error is nil means success
typedef void (^ARTVCIMCallback)(NSError* error);
```

- 返回值：无。

sendMessage

- 声明：(void)sendMessage:(ARTVCIMMessage*)message toParticipants:(NSArray<NSString*>*)participants complete:(ARTVCIMCallback)complete
- 说明：发送 IM 到多个用户。
- 参数：

参数	类型	说明	是否可为空
message	ARTVCIMMessage *	IM 消息	NO
participants	NSArray<NSString>	用户 uid 数组	NO
complete	ARTVCIMCallback	消息发送结果回调	YES

- 返回值：无。

createCustomVideoCapturer

- 声明：- (ARTVCCustomVideoCapturer*)createCustomVideoCapturer:(ARTVCCreateCustomVideoCaputurerParams*)params;
- 说明：创建自定义推流类。
- 参数：

参数	类型	说明	是否可为空
params	ARTVCCreateCustomVideoCaputurerParams*	创建自定义推流类参数	NO

ARTVCCreateCustomVideoCaputurerParams 定义：

参数	类型	说明	是否可为空
provideRenderView	BOOL	是否需要提供渲染 view	NO

- 返回值：无。

destroyCustomVideoCapturer

- 声明：-(void)destroyCustomVideoCapturer;
- 说明：销毁自定义推流类。
- 参数：无。
- 返回值：无。

isScreenCaptureSupported

- 声明：-(BOOL)isScreenCaptureSupported;
- 说明：是否支持屏幕捕捉，需注意 iOS11 及以上才支持。
- 参数：无。
- 返回值：无。

isScreenCaptureStarted

- 声明：-(BOOL)isScreenCaptureStarted;
- 说明：屏幕捕捉是否已启动。
- 参数：无。
- 返回值：无。

startScreenCaptureWithParams

- 声明：-(void)startScreenCaptureWithParams:(ARTVCCreateScreenCaputurerParams*)params
complete:(ARTVCErrCallback)callback;
- 说明：开始屏幕捕捉。
- 参数：

参数	类型	说明	是否可为空
params	ARTVCCreateCustomVideoCaputurerParams*	开始屏幕捕捉参数	NO
callback	ARTVCErrCallback	开始屏幕捕捉结果回调	YES

ARTVCCreateScreenCaputurerParams 定义：

参数	类型	说明	是否可为空
provideRenderView	BOOL	是否需要提供渲染 view	NO

- 返回值：无。

stopScreenCapture

- 声明：-(void)stopScreenCapture;
- 说明：结束屏幕捕捉。
- 参数：无。
- 返回值：无。

6.2.3 回调函数

didReceiveLocalFeed

- 说明：每一次本地发布，只返回一次。
 - 视频通话场景
开启摄像头会触发回调；publish 会触发回调；SDK 在底层做了限制处理，对于同一次发布，如果摄像头开启时已经返回了 feed，那么 publish 时不再返回。
 - 音频通话场景
不存在摄像头开启操作，在 publish 时会触发一次回调。
- 回调解释：

```
typedef NS_ENUM(int,ARTVCFeedType){
    ARTVCFeedTypeRemoteFeed = 0,
    //use builtin camera and microphone
    ARTVCFeedTypeLocalFeedDefault = 1,
    ARTVCFeedTypeLocalFeedCustomVideo = 2,
    ARTVCFeedTypeLocalFeedScreenCapture = 3,
};
```

6.3 Web SDK API

6.3.1 概述

MRTC-WEB-SDK 当前版本号为 1.4.4，主要提供多个 Client 之间的发布、订阅等服务。

其主要功能为：

- 音视频通话功能
- 录制功能
- 桌面共享功能

接入接口协议

接口协议的接入步骤如下：

引入依赖。

- 强依赖：
 - mcu.js
 - meeting_api.js
 - log4b.js
 - reconnecting-websocket.min.js
 - mtc_api.js
 - BandwidthHandler.js
 - adapter.js
- 可选依赖：
 - 若使用浏览器录制，需引入：
 - RecordRTC.min.js (<https://gw.alipayobjects.com/os/lib/recordrtc/5.5.9/RecordRTC.min.js>)
 - EBML.js
 - 若使用pdf投屏，需引入：
 - pdf.js
 - pdf.worker.js (必须和 meeting_api.js 处在 ./lib/pdf.worker.js 相对位置)
 - 若使用自定义区域投屏，需引入：
 - html2image.js
 - 若使用美颜，需引入：
 - beauty.js
 - demo相关：
 - dialogue.css
 - getMediaInfo.js
 - jquery.min.js (<https://gw.alipayobjects.com/os/lib/jquery/3.5.1/dist/jquery.min.js>)
 - iconfont.js

通过以下命令，实例化 SDK。

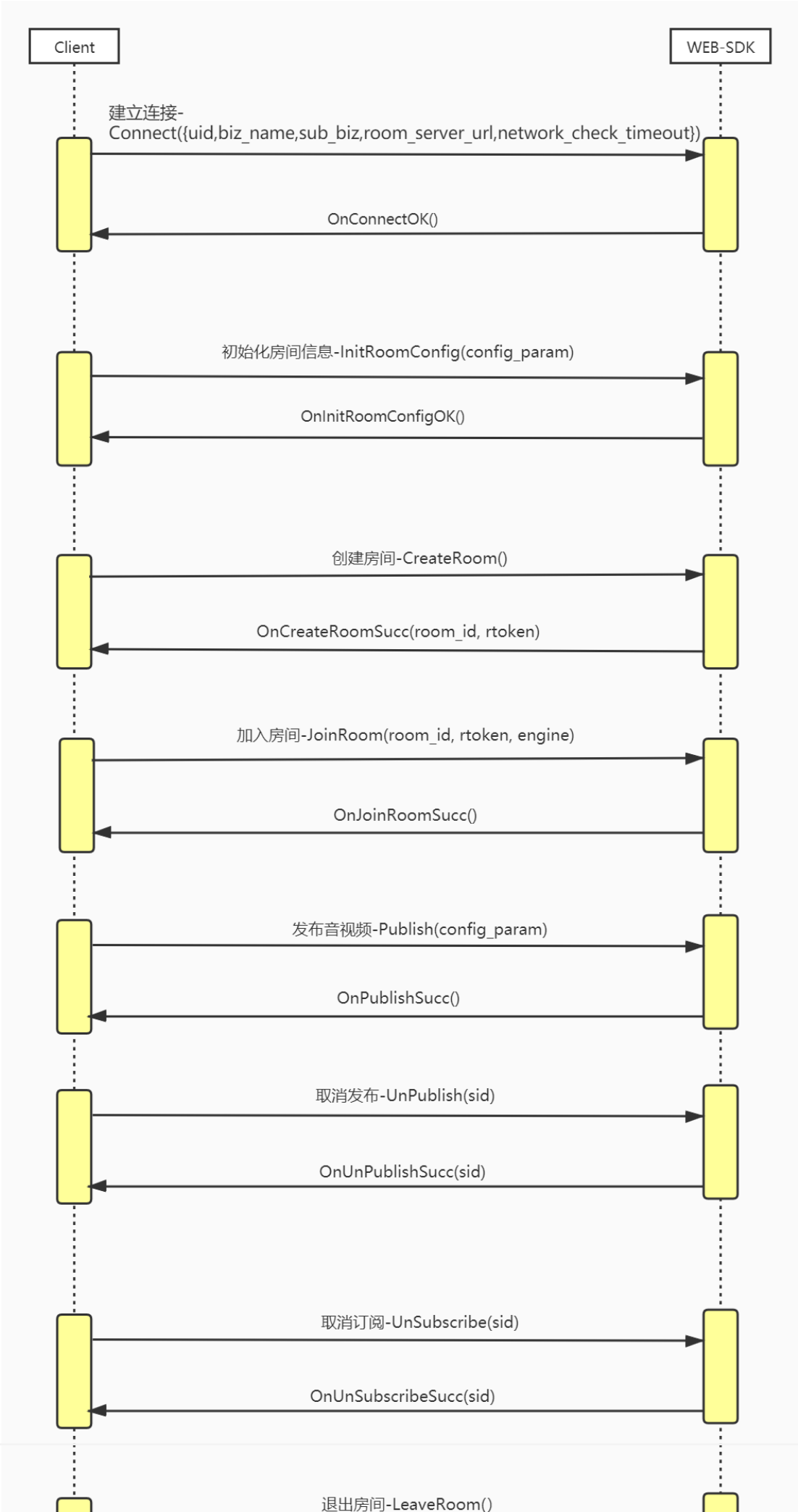
```
let test_controller = new McuController(); // 实例化 SDK
```

接口说明

所有 API 封装在 McuController 的类中，接口分为两类：

- 主调接口：
 - 供业务主动调用。
- 被调接口：
 - 主调接口的反馈回调。
 - 业务上的通知。

使用流程如下：



6.3.2 主调接口

1. Connect

- 说明
建立连接。建立连接时会回调 OnGetSign(bizName,subBiz,uid) 来获取签名。
- 参数

请求参数	类型	描述
config_param	{}	config_param 是一个结构体， 里面包含如下字段： - uid : int 类型，必填，用户唯一标识， 由客户端透传而来。 - room_server_url : String 类型，必填，房间服务器地址。 - biz_name : String 类型，必填，业务标识。 - sub_biz : String 类型，必填，业务标识。

- 回调接口异步返回
 - OnConnectOK()
 - OnConnectFailed(err_code, err_msg)

2. Disconnect

- 说明
断开连接。调用 Disconnect() 接口，如果正处于通话阶段，会触发关闭发布和订阅的流，并离开房间。
即：关闭房间所有流 > 离开房间 > 断开信令连接。
- 参数
无。
- 回调接口异步返回
 - OnConnectClose()

3. InitRoomConfig

说明
初始化房间。

请求参数	类型	描述
config_param	{}	auto_publish_subscribe : int 类型，必填，发布订阅模式 1 表示自动订阅。 2 表示自动发布。 3 表示自动订阅 + 自动发布。 4 表示均不自动，由业务自己操作。 third_id : String 类型，业务关联 id，默认为空。 need_volume_analyser : boolean 类型，默认为 false，是否需要实时音量值。 transport : String 类型，传输方式，可选为 all 或 relay，默认为 all。 defaultTurnServer : String 类型，自定义中转服务器。 ice_relay_ip_array : String 类型，中转服务器对应外网 IP，多个 IP 用 “;” 隔开，如

	: ip1;ip2;ip3 liveUrl : String 类型, 非必填 (直播模式必填), 直播地址。 publish_type : int 类型, 非必填 (直播模式必填)。 • 2 : RTC 模式 • 3 : 直播模式 defaultRecord : boolean 类型, 非必填, 服务端是否默认录制, 默认为 false。 recordStrongDepend : boolean 类型, 是否强依赖录制, 默认为 true。若为 true, 则录制如果没启动成功, 音视频通话也不能成功。 record_third_id : String 类型, 录制三方 ID, 用于业务方区分调用批次, 在启动成功或失败的回调中返回, 非必填。 recordParam : json 类型, 录制参数, 非必填。 silentRecord : boolean 类型, 静默录制, 默认为 false width : int 类型, 宽度, 默认为 640 height : int 类型, 高度, 默认为 360 fps : int 类型, 帧率, 默认为 20 sampleRate : int 类型, 音频采样率, 默认为 48000 channels : int 类型, 声道, 默认为 1 audioCodeRate : int 类型, 音频码率, 默认为 32000 startTimeout : int 类型, 录制开启超时时间, 默认为 10s recordTotalStream : int 类型, 最小成功开启录制流的条数, 默认为 0 splitType : int 类型 0 表示音视频 (默认) 1 表示纯音频 + 音视频 2 表示纯视频 + 音视频 splitFilePath : String 类型, 分离文件的存储路径, 默认为空 recordFormat : int 类型, 录制类型 1 表示mp4 2 表示flv layoutMode : 布局类型 0 表示双人模式 1 表示画中画模式 (双人) 2 表示多人录制模式 multiModeType : 多人录制模板类型 1 为默认, 表示奇特特殊偶平分 (横版) 2 表示奇特特殊偶平分 (竖版) rtWatermark : int 类型, 实时时间水印, 0 表示不设置, 1、2、3、4 表示设置位置分别为左上角, 右上角, 左下角, 右下角, 默认为 1 waterMark : int 类型, 图片水印, 0 表示不设置, 1、2、3、4 表示设置位置分别为左上角, 右上角, 左下角, 右下角, 默认为 1 waterMarkUrl : 图片水印 URL, 用于下载 textWatermark : int 类型, 文字水印, 0 表示不设置, 1、2、3、4 表示设置位置分别为左上角, 右上角, 左下角, 右下角, 默认为 0 text : String 类型, 文字水印文字内容 tagFilter : String 类型, 匹配上发布端的 tag 前缀才能订阅录制; 默认为空, 表示全匹配 tagPosition : String 类型, 指定 TAG 的布局位置, 例如 {"aa":1,"bb":2} mixPosition : jsonArray 类型, 录制窗布局, 例如 [{"index":1, &nbsp;"x_offset":1,"y_offset":2, "width":160,"height": 90}, {"index":2, &nbsp;"x_offset":1,"y_offset":2, "width":160,"height": 90}] endType : int 类型, 结束类型 0 表示调用者退出时结束 1 表示房间销毁时结束 filePath : String 类型, 非必填, 服务端录制路径 (若指定 filePath, 则需指定 defaultRecord 为 true) 若 auto_publish_subscribe=1, 需要以下参数: initSubscribe : Array 类型, 订阅参数。 subscribe_video_id : 必填, 订阅视频控制标签 ID, 用于展示视频流。 例如: <video id="video1"width="auto"height="auto"> video</video> subscribe_audio_id : 必填, 用于展示订阅音频流。 例如: <audio id="audio1"autoplay>音频</audio> feedId_id : 必填, label 标签, 用于展示 feedId。 例如: <label id="feedId"type="text"></label> subscribe_streamId_id : 必填, label 标签, 用于展示 streamId。 例如: <label id="streamId"type="text"></label> 整体举例: initSubscribe = [{ subscribe_video_id: "video1", subscribe_audio_id: "audio1", subscribe_streamId_id: "streamId", feedId_id: "feedId" },{}];
--	---

	若 auto_publish_subscribe=2，需要以下参数： media_type：int 类型，必填，音视频类型。 1 表示音视频 2 表示纯音频 3 表示纯视频 publish_bitrate：int 类型，选填，发布最大码率，默认为 600 Kbps publish_device：int 类型，必填，发布类型。 1 表示摄像头 2 表示共享桌面 3 表示发布文件（图片和视频） 4 表示发布自定义页面区域 当 publish_device=1 时： videoSource：String 类型，可选，摄像头的 deviceId，用于发布指定摄像头。 audioSource：String 类型，可选，麦克风的 deviceId，用于发布指定麦克风。 aspectRatioStrongDepend：boolean 类型，默认为 false，是否强制指定横纵比。 aspectRatio：String 类型，可选（当 aspectRatioStrongDepend=true 时必填） 1 表示 4:3 2 表示 16:9 video_profile_type：int 类型，可选，本地视频清晰度档位。 1 表示 720p(HD), 1280x720, 16:9 2 表示 VGA, 640x480, 4:3 3 表示 360p(nHD), 640x360, 16:9 4 表示 720p(HD), 720x1280, 9:16 5 表示 360p(nHD), 360x640, 9:16 initPublish：Array 类型，必填，发布参数。 publish_video_id：根据 media_type 选填，发布视频控制标签，用于展示视频流。 publish_streamId_id：必填，label 标签，用于展示 streamId。 publish_tag：必填，业务自定义。 <pre>initPublish = [{publish_video_id: "publish_video1", publish_streamId_id: "publish_streamId1", publish_tag: "tag1" }, {publish_video_id: "publish_video2", publish_streamId_id: "publish_streamId2", publish_tag: "tag2" }, {publish_video_id: "publish_video3", publish_streamId_id: "publish_streamId3", publish_tag: "tag3" }];</pre> 若 auto_publish_subscribe=3，需要以下参数： media_type：int 类型，必填，音视频类型。 1 表示音视频 2 表示纯音频 3 表示纯视 publish_bitrate：int 类型，选填，发布最大码率，默认为 600 Kbps publish_device：int 类型，必填，发布类型。 1 表示摄像头 2 表示共享桌面 3 表示发布文件（图片和视频） 4 表示发布自定义页面区域 video_profile_type：int 类型，可选，本地视频清晰度档位。 1 表示 720p(HD), 1280x720, 16:9 2 表示 VGA, 640x480, 4:3 3 表示 360p(nHD), 640x360, 16:9 4 表示 720p(HD), 720x1280, 9:16 5 表示 360p(nHD), 360x640, 9:16 initPublish：Array 类型，必填，发布参数。 publish_video_id：根据 media_type 选填，发布视频控制标签，用于展示视频流。 publish_streamId_id：必填，label 标签，用于展示 streamId。 publish_tag：必填，业务自定义。 例如： <pre>initPublish = [{publish_video_id: "publish_video1", publish_streamId_id: "publish_streamId1", publish_tag: "tag1" }, {publish_video_id: "publish_video2", publish_streamId_id: "publish_streamId2", publish_tag: "tag2" }, {publish_video_id: "publish_video3", publish_streamId_id: "publish_streamId3", publish_tag: "tag3" }];</pre> initSubscribe：Array 类型，订阅参数。 subscribe_video_id：必填，订阅视频控制标签，用于展示视频流。 subscribe_audio_id：必填，用于展示订阅音频流 feedId_id：必填，label 标签，用于展示 feedId。 subscribe_streamId_id：必填，label 标签，用于展示 streamId。 例如：
--	--

		<pre>initSubscribe = [{ subscribe_video_id: "video1", subscribe_audio_id: "audio1", subscribe_streamId_id: "subscribe_streamId1", feedId_id: "feedId1" },{}];</pre>
bizName	String	业务名
subBiz	String	子业务
uid	String	用户 ID

回调接口异步返回

- OnInitRoomConfigOK()
- OnInitRoomConfigFail(err_code, err_msg)

4. CreateRoom

- 说明
创建房间。
- 参数
无。
- 回调接口异步返回
 - OnCreateRoomSucc(room_id, rtoken)
 - OnCreateRoomFailed()

5. JoinRoom

- 说明
加入房间。
- 参数

请求参数	类型	描述
room_id	String	房间 ID
rtoken	String	房间 token，相当于房间密码
engine	int	0 表示 alipay，为默认值 1 表示 aliyun

- 回调接口异步返回
 - OnJoinRoomSucc()
 - OnJoinRoomFailed(err_code, err_msg)

6. Invite

- 说明
邀请加入房间。
- 参数

请求参数	类型	描述
invitee	String	被邀请方 uid
channelType	int	推送渠道 0 表示 WEB 1 表示 ALIPAY
extra	String	业务透传字段

- 回调接口异步返回
 - OnInviteOK()
 - OnInviteFail(err_code, err_msg)

7. ReplyInvite

- 说明
回复邀请加入房间。
- 参数

请求参数	类型	描述
roomId	String	房间 ID
inviter	String	邀请方 uid
reply	int	0 表示接听 2 表示拒绝

- 回调接口异步返回
 - OnReplyInviteOK()
 - OnReplyInviteFail(err_code, err_msg)

8. LeaveRoom

- 说明
退出房间。
- 参数
无。
- 回调接口异步返回
 - OnLeaveRoom()

9. Publish

- 说明
发布媒体流。
- 参数

请求参数	类型	描述
config_param	{}	media_type : int 类型，必填，音视频类型。 1 表示音视频。 2 表示纯音频。 3 表示纯视频。 publish_bitrate : int 类型，选填，发布最大码率，默认为 600 Kbps。 publish_device : int 类型，必填，发布类型。 1 表示摄像头; 2 表示共享桌面。 3 表示发布文件（图片和视频） 4 表示发布自定义页面区域。 video_profile_type : int 类型，可选，本地视频清晰度档位。 1 表示 720p(HD), 1280x720, 16:9 2 表示 VGA, 640x480, 4:3 3 表示 360p(nHD), 640x360, 16:9 4 表示 720p(HD), 720x1280, 9:16 5 表示 360p(nHD), 360x640, 9:16 publish_video_id : String 类型，根据 media_type 选填，发布视频控制标签，用于展示视频流。 publish_streamId_id : String 类型，必填，label 标签，用于展示 streamId。 publish_tag : String 类型，业务指定 tag。 need_volume_analyser : boolean 类型，默认为 false，是否需要实时音量值。 file : file，可选（publish_device=3 时必填），图片或视频文件。 例如：let file = document.getElementById('file').files[0]; part_of_screen_id : String 类型，可选（publish_device=4 时必填），需要共享区域的 div 的 id 值。 例如：<div id="part_of_screen_id"></div> 当 publish_device=1 时： videoSource : String 类型，可选，摄像头的 deviceId，用于发布指定摄像头。 audioSource : String 类型，可选，麦克风的 deviceId，用于发布指定麦克风。 aspectRatioStrongDepend : boolean 类型，可选，默认为 false，是否强制指定纵横比。 aspectRatio : String 类型，可选（aspectRatioStrongDepend=true 时必填）： 1 表示 4:3 2 表示 16:9 publish_bitrate : int 类型，选填，发布最大码率，默认 600 Kbps。

- 调接口异步返回：
 - OnPublishSucc(sid)
 - OnPublishFailed(sid,err_code, err_msg)

10. Subscribe

- 说明
订阅某路媒体流。
- 参数

请求参数	类型	描述
config_param	{}	subscribe_video_id : String 类型，必填，订阅视频控制标签，用于展示视频流。 subscribe_audio_id : String 类型，必填，用于展示订阅音频流。 subscribe_streamId_id : String 类型，必填，label 标签，用于展示 streamId。 feedId_id : String 类型，必填，label 标签，用于展示 feedId。 feedId : 订阅的流 id。

		need_volume_analyser : boolean 类型，默认为 false，是否需要实时音量值。
--	--	--

- 回调接口异步返回
 - OnSubscribeSucc(feedId,sid)
 - OnSubscribeFailed(sid,err_code, err_msg)

11. SetLocalAudioEnable

- 说明
开启或关闭本地麦克风。
- 参数

请求参数	类型	描述
enabled	int	1 表示开 0 表示关
sid	int	流 id

12. SetLocalVideoEnable

- 说明
开启或关闭本地摄像头。
- 参数

请求参数	类型	描述
enabled	int	1 表示开 0 表示关
sid	int	流 id

13. SetLocalCodecType

- 说明
设置视频编码格式。
- 参数

请求参数	类型	描述
type	String	H264 VP8

14. SetLocalProtoType

- 说明
设置音视频传输协议。
- 参数

请求参数	类型	描述
type	String	all tcp udp

15. TakePicture

- 说明
截图。
- 参数

请求参数	类型	描述
type	int	0 表示截取本地端 1 表示截取远端图像
width	int	图像宽度
height	int	图像高度
sid	int	流 id
picture_type	int	输出图片格式： 1 表示 png 2 表示 jpeg/jpg

17. StopRecord

- 说明
停止浏览器录制，与 StartBrowserRecord 接口配合使用。
离开房间之前需要调用此接口，否则无法获取 blob 数据。
- 参数

请求参数	类型	描述
clientRecordId	String	浏览器录制 id

- 回调接口异步返回
 - OnClientStopRecordSuccess(videoURL, clientRecordBlob, clientRecordId)
 - OnClientStopRecordFailed(clientRecordId, code, msg)

18. PauseRecord

- 说明
暂停浏览器录制，与 ResumeRecord 接口配合使用。
- 参数

请求参数	类型	描述
clientRecordId	String	浏览器录制 id

- 回调接口异步返回
- OnClientPauseRecordSuccess(clientRecordId)
- OnClientPauseRecordFailed(clientRecordId, code, msg)

19. ResumeRecord

- 说明
继续浏览器录制，与 PauseRecord 接口配合使用。
- 参数

请求参数	类型	描述
clientRecordId	String	浏览器录制 id

- 回调接口异步返回
- OnClientResumeRecordSuccess(clientRecordId)
- OnClientResumeRecordFailed(clientRecordId, code, msg)

20. DownloadRecord

- 说明
下载浏览器录制音视频，需要在离开房间之前下载。
- 参数

请求参数	类型	描述
clientRecordId	String	浏览器录制 id
fileName	String	指定文件名

- 回调接口异步返回
- OnClientDownloadRecordSuccess(clientRecordId)
- OnClientDownloadRecordFailed(clientRecordId, code, msg)

21. StartRemoteRecord

- 说明
开始服务端录制（远端录制，即在媒体服务器上记录），与 StopRemoteRecord 接口配合使用。
- 参数

请求参数	类型	描述
file_path	String	选填，服务端录制路径
record_param	json	录制参数，非必填 silentRecord : boolean 类型，静默录制，默认为 false width : int 类型，宽度，默认为 640

		height : int 类型, 高度, 默认为 360 fps : int 类型, 帧率, 默认为 20 sampleRate : int 类型, 音频采样率, 默认为 48000 channels : int 类型, 声道, 默认为 1 audioCodeRate : int 类型, 音频码率, 默认为 32000 startTimeout : int 类型, 录制开启超时时间, 默认为 10s recordTotalStream : int 类型, 最小成功开启录制流的条数, 默认为 0 splitType : int 类型 0 表示音视频 (默认) 1 表示纯音频 + 音视频 2 表示纯视频 + 音视频 splitFilePath : String 类型, 分离文件的存储路径, 默认为空 recordFormat : int 类型, 录制类型 1 表示 mp4 2 表示 flv layoutMode : 布局类型 0 表示双人模式 1 表示画中画模式 (双人) 2 表示多人录制模式 multiModeType : 多人录制模板类型 1 为默认, 表示奇特特殊偶平分 (横版) 2 表示奇特特殊偶平分 (竖版) rtWatermark : 实时时间水印, 0 表示不设置, 1/2/3/4 表示设置位置分别为左上角, 右上角, 左下角, 右下角, 默认为 1 waterMark : 图片水印, 0 表示不设置, 1/2/3/4 表示设置位置分别为左上角, 右上角, 左下角, 右下角, 默认为 1 waterMarkUrl : 图片水印 URL, 用于下载 textWatermark : int 类型, 文字水印, 0 表示不设置, 1/2/3/4 表示设置位置分别为左上角, 右上角, 左下角, 右下角, 默认为 0 text : String 类型, 文字水印文字内容 tagFilter : String 类型, 匹配发布端的 tag 前缀才能订阅录制; 默认为空, 表示全匹配 mixPosition : jsonArray 类型, 录制窗布局, 例如 [{"index":1,"x_offset":1,"y_offset":2,"width":160,"height": 90},{ "index":2,"x_offset":1,"y_offset":2,"width":160,"height": 90}] tagPosition : String 类型, 指定 TAG 的布局位置, 例如 {"aa":1,"bb":2} endType : int 类型, 结束类型 0 表示调用者退出时结束 1 表示房间销毁时结束
record_third_id	String	录制三方 ID, 用于业务方区分调用批次, 在启动成功或失败的回调中返回, 非必传。

- 回调接口异步返回
 - OnInitRemoteRecordSucc(record_id, record_third_id) : 服务端录制初始化成功。
 - OnStartRemoteRecordSucc(record_id, record_third_id) : 服务端录制启动成功, 晚于 OnInitRemoteRecordSucc。
 - OnStartRemoteRecordFailed(record_id, err_code, err_msg, record_third_id) : 服务端录制启动失败。

22. StopRemoteRecord

- 说明
 - 结束服务端录制, 与 StartRemoteRecord 接口配合使用。
- 参数

请求参数	类型	描述
record_id	String	服务端录制 id

- 回调接口异步返回
 - OnStopRecordSucc(recordId)
 - OnStopRecordFailed(recordId,err_code, err_msg)

23. GetRecordInfo

- 说明
获取服务端录制结果。调用此接口会回调 OnGetSign(bizName,subBiz,record_id) 来获取签名，同时 OnGetSign 中的 uid 会替换为此处的 record_id。
- 参数

请求参数	类型	描述
record_id	String	服务端录制 id
room_id	String	房间号
media_type	int	文件内容类型： 0 表示音视频（默认） 1 表示纯音频 2 表示纯视频

- 回调接口异步返回
 - OnRecordInfoSucc(recordInfo)
 - OnRecordInfoFailed(recordId,err_code, err_msg)

24. UnPublish

- 说明
取消发布媒体流。
- 参数

请求参数	类型	描述
sid	int	流 id

- 回调接口异步返回
 - OnUnPublishSucc(sid)

25. UnSubscribe

- 说明
取消订阅某路媒体流。
- 参数

请求参数	类型	描述
sid	int	流 id

- 回调接口异步返回
 - OnUnSubscribeSucc(sid)

26. SendTextMsg

- 说明
发送文本消息。
- 参数

请求参数	类型	描述
msg	String	消息文本
participants	Array	发送目标 id (为空则发给与会其他所有人)

28. SetPublishWeakBitrateLimit

- 说明
设置发布弱网码率告警下限，在 初始化 (InitRoomConfig) 时调用此接口。
- 参数

请求参数	类型	描述
weak_bitrate	int	发布弱网码率告警下限： 音频 + 视频：默认为 200 单独音频：默认为 20-25

29. SetSubscribeWeakBitrateLimit

- 说明
设置订阅弱网码率告警下限，在 初始化 (InitRoomConfig) 时调用此接口。
- 参数

请求参数	类型	描述
weak_bitrate	int	订阅弱网码率告警下限 弱网码率告警下限 音频+视频：默认为 200 单独音频：默认为 20-25

32. StartVod(vodParams)

- 说明
开始语音点播。
- 参数

请求参数	类型	描述
vodParams	json	vodFile：点播文件名 vodStartTimeout：开启点播超时时间 (单位为秒，默认为 5s)

- 回调接口异步返回
 - OnStartVodSuccess(file, vod_id)

- OnStartVodFail(file, vod_id, err_code, err_msg)

33. StopVod(vod_id)

- 说明
停止语音点播。
- 参数

请求参数	类型	描述
vod_id	String	点播 id

- 回调接口异步返回
 - OnStopVodSuccess(vod_id)
 - OnStopVodFail(vod_id, err_code, err_msg)

34. DownloadLog()

- 说明
日志上报，调用此接口可下载控制台日志到 PC 本机（默认最长 10000 行日志）。
- 参数
无。

37. ChangeMediaStream()

- 说明
自定义切流。
- 参数

请求参数	类型	描述
config	json	publish_device：必填，发布类型。 publish_bitrate：int 类型，选填，发布最大码率，默认为 600 Kbps media_type：必填，音视频类型。 sid：必填，已发布流 id。 publish_device=1 video_profile_type：选填，分辨率。 videoSource：String 类型，可选，摄像头的 deviceId，用于切换指定摄像头。 audioSource：String 类型，可选，麦克风的 deviceId，用于切换指定麦克风。 aspectRatioStrongDepend：boolean 类型，默认为 false，是否强制指定纵横比。 aspectRatio：String 类型，可选（aspectRatioStrongDepend=true 时必填） 1 表示 4:3 2 表示 16:9 file：选填，图片或视频文件，publish_device=3 时必填。 part_of_screen_id：选填，需要发布的自定义区域，publish_device=4 时必填。

- 回调接口异步返回
 - OnChangeMediaStreamSuccess(sid)
 - OnChangeMediaStreamFailed(sid,code, msg)

38. ChangeProfile()

- 说明
动态切换（分辨率、摄像头麦克风、横纵比）。
- 参数

请求参数	类型	描述
config	json	sid：必填，已发布流 id。 video_profile_type：选填，分辨率。 local_video_width：选填，宽度。 local_video_height：选填，高度。 videoSource：String 类型，可选，摄像头的 deviceId，用于切换指定摄像头。 audioSource：String 类型，可选，麦克风的 deviceId，用于切换指定麦克风。 aspectRatioStrongDepend：boolean 类型，默认为 false，是否强制指定横纵比。 aspectRatio：String 类型，可选（aspectRatioStrongDepend 为 true 时必填）。 1 表示 4:3 2 表示 16:9

- 回调接口异步返回
 - OnChangeMediaStreamSuccess(sid)
 - OnChangeMediaStreamFailed(sid,code, msg)

39. GetDevices()

- 说明
获取设备信息。
- 参数
无。
- 回调接口异步返回
 - OnGetDevicesSuccess(devicesInfo)
 - OnGetDevicesFailed(code, msg)

40. StartBrowserRecord

- 说明
开启浏览器录制（近端录制），指基于浏览器的 native 录制能力进行录制。目前只支持录制两条流，不支持录制点播机器人。
- 参数

请求参数	类型	描述
clientRecordId	String	浏览器录制 id
stream_list	array	流 id (sid)
record_config	json	media_type：int 类型，录制选项 1 表示音视频

		2 表示纯音频 3 表示纯视频 record_type : int 类型, 录制分离 1 表示多端合并 2 表示本地 3 表示远端 time_slice : int 类型, 录制 blob 通知间隔时间, 默认为 30, 单位为 min video_profile : json 类型 videoCodec : String 类型, 视频编码格式, h264/vp8/vp9 frameRate : int 类型, 帧率
--	--	---

- 回调接口异步返回
 - OnClientStartRecordSuccess(clientRecordId)
 - OnClientStartRecordFailed(clientRecordId, code, msg)

41. PauseRemoteRecord

- 说明
暂停服务端录制, 与 ResumeRemoteRecord 接口配合使用。
- 参数

请求参数	类型	描述
record_id	String	服务端录制 id

- 回调接口异步返回
 - OnPauseRemoteRecordSucc(record_id)
 - OnPauseRemoteRecordFailed(record_id, err_code, err_msg)

42. ResumeRemoteRecord

- 说明
恢复服务端录制, 与 PauseRemoteRecord 接口配合使用。
- 参数

请求参数	类型	描述
record_id	String	服务端录制 id

- 回调接口异步返回
 - OnResumeRemoteRecordSucc(record_id)
 - OnResumeRemoteRecordFailed(record_id, err_code, err_msg)

6.3.3 被调接口

1. OnConnectOK()

- 说明

连接成功。

- 参数
无。

2. OnConnectFailed(err_code, err_msg)

- 说明
连接失败。
- 参数

回调参数	类型	说明
err_code	String	错误码
err_msg	String	错误码对应的错误信息。 101：初始化超时 102：初始化失败

3. OnGetSign(bizName,subBiz,uid)

说明
获取签名。
说明：业务需要覆盖实现该方法，具体逻辑是到业务自己的服务端获取签名，在业务申请接入时会下发生成的签名。

参数

请求参数	类型	描述
bizName	String	业务名
subBiz	String	子业务
uid	String	用户 ID

4. OnInitRoomConfigOK()

- 说明
初始化房间成功
- 参数
无。

5. OnInitRoomConfigFail(err_code, err_msg)

- 说明
初始化房间失败。
- 参数

回调参数	类型	说明
------	----	----

err_code	String	错误码
err_msg	String	错误码对应的错误信息。 201：房间状态错误 202：参数错误

6. OnMediaCallSucc(sid)

- 说明
初始化音视频成功。
- 参数
无。

7. OnMediaCallFail(err_code, err_msg)

- 说明
初始化音视频失败。
- 参数

回调参数	类型	说明
err_code	int	错误码对应的错误信息： 5101：视频启动失败 5102：音频启动失败 5103：浏览器不支持 5104：房间状态不正确 5105：共享桌面启动失败 5106：发布获取文件流失败 5107：发布文件不存在 5108：发布区域不存在
err_msg	String	-

8. OnCreateRoomSucc(room_id, rtoken)

说明
创建房间成功。

参数

回调参数	类型	说明
room_id	String	房间号
rtoken	String	房间 token，相当于房间密码

9. OnCreateRoomFailed(err_code, err_msg)

- 说明
创建房间失败。
- 参数

回调参数	类型	说明
err_code	int	错误码对应的错误信息： 301：房间未初始化 302：服务器返回错误 303：创建超时 304：media 初始化超时
err_msg	String	-

10. OnJoinRoomSucc()

- 说明
加入房间成功。
- 参数
无。

11. OnJoinRoomFailed(err_code, err_msg)

- 说明
加入房间失败。
- 参数

回调参数	类型	说明
err_code	int	错误码对应的错误信息： 401：房间未初始化 402：参数错误 403：服务器返回错误 404：加入超时
err_msg	String	-

12. OnInviteOK()

- 说明
发送邀请加入房间成功。
- 参数
无。

13. OnInviteFail(err_code, err_msg)

- 说明
发送邀请加入房间失败。
- 参数

回调参数	类型	说明
err_code	int	错误码对应的错误消息： 501：房间状态错误 502：服务器返回错误

		503 : 参数错误
err_msg	String	-

14. OnReplyInviteOK()

- 说明
发送答复邀请加入房间成功。
- 参数
无。

15. OnReplyInviteFail(err_code, err_msg)

- 说明
发送答复邀请加入房间失败。
- 参数

回调参数	类型	说明
err_code	int	错误码对应的错误消息： 601：房间状态错误 602：服务器返回错误
err_msg	String	-

16. OnInviteRequest(roomId, rtoken, inviter, extra)

- 说明
被邀请者收到“加入房间”邀请消息。
- 参数

回调参数	类型	说明
roomId	String	房间 ID
rtoken	String	房间 token
inviter	String	邀请方 uid
extra	String	业务透传字段

17. OnInviteReply(invitee, reply)

- 说明
邀请者收到“邀请加入房间”回复消息。
- 参数

回调参数	类型	说明
invitee	String	被邀请方 uid
reply	int	0：接听 1：离线

		2 : 拒绝
--	--	--------

18. OnRoomAttendanceList(participants)

- 说明
推送“房间与会者列表”给新加入者。
- 参数

回调参数	类型	说明
participants	jsonarray	房间与会者列表。 示例： <pre>{ "uid": "1234", "publish": [{ "feedId": "1", "tag": "xxx" }, { "feedId": "2", "tag": "xxx" }], "subscribe": [{ "feedId": "3" }, { "feedId": "4" }] }</pre>

19. OnNewJoinerIn(participant)

- 说明
推送“新加入房间者”给与会者。
- 参数

回调参数	类型	说明
participant	String	新加入房间者 uid

20. OnPublishSucc(sid)

- 说明
发布媒体流成功。
- 参数

回调参数	类型	说明
sid	int	流 ID

21. OnPublishFailed(sid,err_code, err_msg)

- 说明
发布媒体流失败。

• 参数

回调参数	类型	说明
sid	int	流 ID
err_code	String	
err_msg	String	1071：服务器返回错误 1072：未设置 video 标签 1073：超出发布限额 1074：发布建立 ice 连接超时

22. OnSubscribeSucc(feed,sid)

• 说明

订阅媒体流成功。

• 参数

回调参数	类型	说明
sid	String	流 ID
feed	String	流信息。示例： { "feedId": ["xxx", "xxx"], "strategy": 1 } 其中： feedId：list 类型，表示订阅的流 ID strategy：int 类型，表示策略

23. OnSubscribeFailed(sid,err_code, err_msg)

• 说明

订阅媒体流失败。

• 参数

回调参数	类型	说明
sid	String	流 ID
err_code	int	错误码对应的错误信息。 1081：服务器返回错误 1082：视频窗口 videoId 已使用 1083：超出订阅限额 1084：订阅建立媒体链路失败 1085：订阅的窗口 videoId 为空
err_msg	String	-

24. OnNewPublish(feed)

• 说明

推送“有新发布”给与会者。

• 参数

回调参数	类型	说明
feed	json	新发布的流信息： uid：流所属的用户 ID feedId：发布流 ID tag：标签

25. OnNewSubscribe(subscriber,feed)

• 说明

推送“有新订阅”给与会者。

• 参数

回调参数	类型	说明
subscriber	String	新订阅者 ID
feed	json	新订阅的流信息： uid：流所属的用户 ID feedId：被订阅流 ID

26. OnUnPublishSucc(sid)

• 说明

取消发布成功。

• 参数

回调参数	类型	说明
sid	String	流 ID

27. OnUnSubscribeSucc(sid)

• 说明

取消订阅媒体流成功。

• 参数

回调参数	类型	说明
sid	String	流 ID

28. OnUnPublish(feed)

• 说明

推送“取消发布”给与会者。

• 参数

回调参数	类型	说明
feed	json	取消发布的流信息： uid：流所属的用户 ID feedId：发布流 ID

29. OnUnSubscribe(unsubscriber,feed)

- 说明
推送“取消订阅”给与会者。
- 参数

回调参数	类型	说明
unsubscriber	String	取消订阅者 ID
feed	json	取消订阅的流信息： uid：流所属的用户 ID feedId：被订阅流 ID

30. OnMediaClose(code , sid)

- 说明
音视频关闭通知。
- 参数

回调参数	类型	说明
code	int	5001：主动关闭 5002：离开房间被动通知关闭 5003：发布/订阅失败（服务器返回） 5004：建立音视频 ice 连接超时 5005：房间状态异常 5006：取消发布被动通知 5007：ice 连接异常 5008：退出房间关闭音视频
sid	int	流 ID

39. OnSendTextMsgSucc(msgId)

- 说明
发送文本消息成功。
- 参数

回调参数	类型	说明
msgId	int	消息 ID

40. OnSendTextMsgFailed(msgId, code, msg)

- 说明

发送文本消息失败。

• 参数

回调参数	类型	说明
msgId	int	消息 ID
code	int	10291：服务端返回错误
msg	String	-

43. OnReceiveTextMsg(uid, msg)

• 说明

推送文本消息。

• 参数

回调参数	类型	说明
uid	String	文本消息发送人
msg	String	消息文本

45. OnLeaveRoom(leaveType)

• 说明

退出房间回调。

• 参数

回调参数	类型	说明
leaveType	int	退出房间类型： 1：正常退出 2：异常退出

46. OnParticipantLeaveRoom(participant,exitType)

• 说明

推送“退出房间者”给与会者。

• 参数

回调参数	类型	说明
participant	String	退出者 ID
exitType	int	退出类型： 1：正常退出 2：异常退出

47. OnConnectClose()

• 说明

连接关闭。

- 参数
无。

48. OnNetworkWeak(bpsSend, bpsRecv,sid)

- 说明
弱网回调。
- 参数

回调参数	类型	说明
bpsSend	String	每秒发送数据（单位 bit）
bpsRecv	String	每秒接收数据（单位 bit）
sid	int	流 ID（便于区分哪一路流出现弱网）

55. OnStartVodSuccess(file, vod_id)

- 说明
开始语音点播成功。
- 参数

回调参数	类型	说明
file	String	点播文件
vod_id	String	点播 ID

56. OnStartVodFail(file, vod_id, err_code, err_msg)

- 说明
开始语音点播失败。
- 参数

回调参数	类型	说明
file	String	点播文件
vod_id	String	点播 ID
err_code	int	错误码对应的消息： 10371：服务端返回错误 10372：房间状态错误 10373：服务端通知开启失败 10374：点播参数有误 10375：开启点播超时
err_msg	String	-

57. OnStopVodSuccess(vod_id)

- 说明
停止语音点播成功。
- 参数

回调参数	类型	说明
vod_id	String	点播 ID

58. OnStopVodFail(vod_id, err_code, err_msg)

- 说明
停止语音点播失败。
- 参数

回调参数	类型	说明
vod_id	String	点播 ID
err_code	int	错误码对应的消息： 10391：服务端返回失败 10392：房间状态错误 10393：点播 ID 有误
err_msg	String	错误消息

59. OnVodOver(vod_id)

- 说明
语音点播结束通知。
- 参数

回调参数	类型	说明
vod_id	String	点播 ID

60. OnVolumeAnalyser(sid, analyser)

- 说明
实时音量数据回调。
- 参数

回调参数	类型	说明
sid	String	流 ID
analyser	AnalyserNode	返回 AnalyserNode 对象

67. OnClientStartRecordSuccess(clientRecordId)

- 说明
开启浏览器录制成功。
- 参数

回调参数	类型	说明
clientRecordId	String	浏览器录制 ID

68. OnClientStartRecordFailed(clientRecordId, code, msg)

- 说明
开启浏览器录制失败。
- 参数

回调参数	类型	说明
clientRecordId	String	浏览器录制 ID
code	int	20301 : room 状态错误 20302 : 已经在录制中 20303 : 录制 id 已经使用 20304 : 未发布或未订阅 20305 : mediaType 参数错误
msg	String	错误码对应的错误消息。

69. OnClientStopRecordSuccess(videoURL, clientRecordBlob, clientRecordId, targetType)

- 说明
停止浏览器录制成功。
- 参数

回调参数	类型	说明
videoURL	String	文件 URL
clientRecordBlob	blob	录制文件
clientRecordId	String	浏览器录制 ID

70. OnClientStopRecordFailed(clientRecordId, code, msg)

- 说明
停止浏览器录制失败。
- 参数

回调参数	类型	说明
clientRecordId	String	浏览器录制 ID
code	int	20331 : 房间状态错误

msg	String	错误码对应的错误消息。
-----	--------	-------------

71. OnClientPauseRecordSuccess(clientRecordId)

- 说明
暂停浏览器录制成功。
- 参数

回调参数	类型	说明
clientRecordId	String	浏览器录制 ID

72. OnClientPauseRecordFailed(clientRecordId, code, msg)

- 说明
暂停浏览器录制失败。
- 参数

回调参数	类型	说明
clientRecordId	String	浏览器录制 ID
code	int	20371：房间状态错误 20372：未开启浏览器录制
msg	String	错误码对应的错误消息。

73. OnClientResumeRecordSuccess(clientRecordId)

- 说明
继续浏览器录制成功。
- 参数

回调参数	类型	说明
clientRecordId	String	浏览器录制 ID

74. OnClientResumeRecordFailed(clientRecordId, code, msg)

- 说明
继续浏览器录制失败。
- 参数

回调参数	类型	说明
clientRecordId	String	浏览器录制 ID
code	int	20391：房间状态错误 20392：未开启浏览器录制
msg	String	错误码对应的错误消息。

75. OnClientDownloadRecordSuccess(clientRecordId)

- 说明
下载浏览器录制文件成功。
- 参数

回调参数	类型	说明
clientRecordId	String	浏览器录制 ID

76. OnClientDownloadRecordFailed(clientRecordId, code, msg)

- 说明
下载浏览器录制文件失败。
- 参数

回调参数	类型	说明
clientRecordId	String	浏览器录制 ID
code	int	20351：房间状态错误 20352：未开启浏览器录制 20353：未停止浏览器录制
msg	String	错误码对应的错误消息。

77. OnGetDevicesSuccess(devicesInfo)

- 说明
获取设备信息成功。
- 参数

回调参数	类型	说明
devicesInfo	json	设备信息

78. OnGetDevicesFailed(code, msg)

- 说明
获取设备信息失败。
- 参数

回调参数	类型	说明
code	int	30101：浏览器不支持音视频 30102：获取媒体错误
msg	String	错误码对应的错误消息。

79. OnChangeMediaStreamSuccess(sid)

- 说明
切流成功通知。
- 参数

回调参数	类型	说明
sid	int	流 ID

80. OnChangeMediaStreamFailed(sid,code, msg)

- 说明
切流失败通知。
- 参数

回调参数	类型	说明
sid	int	流 ID
code	int	1101：未发布 1102：参数错误 1103：media_type 错误 1104：获取音视频失败 1105：共享屏幕流为空 1106：获取视频失败 1107：获取文件流失败 1108：获取音频失败 1109：浏览器不支持音视频
msg	String	错误码对应的错误消息。

81. OnClientRecordBlob(clientRecordId, blob)

- 说明
浏览器录制推送录制 blob 数据。
- 参数

回调参数	类型	说明
clientRecordId	String	浏览器录制 ID
blob	Blob	-

82. OnInitRemoteRecordSucc(record_id, record_third_id)

- 说明
服务端录制初始化成功。
- 参数

回调参数	类型	说明
recordId	String	录制 id
record_third_id	String	启动录制时传入的 id，区分调用批次

83. OnStartRemoteRecordSucc(record_id, record_third_id)

- 说明
开始服务端录制成功。

• 参数

回调参数	类型	说明
record_id	String	服务端录制 id
record_third_id	String	启动录制时传入的 id，区分调用批次

84. OnStartRemoteRecordFailed

• 说明

开始服务端录制失败。

• 参数

回调参数	类型	说明
record_id	String	服务端录制 id
err_code	int	10311 服务端返回开启录制失败 10312：开启录制超时 10313：推送开启录制失败 eventCode=1 10314：推送开启录制失败 eventCode=2
err_msg	String	-
record_third_id	String	启动录制时传入的 id，区分调用批次

85. OnStopRemoteRecordSucc(record_id)

• 说明

停止服务端录制成功。

• 参数

回调参数	类型	说明
record_id	String	服务端录制 id

86. OnStopRemoteRecordFailed(record_id,err_code, err_msg)

• 说明

停止服务端录制失败。

• 参数

回调参数	类型	说明
record_id	String	服务端录制 id
err_code	String	10331：服务端返回停止失败
err_msg	String	-

87. OnRemoteRecordInfoSucc(recordInfo)

- 说明
获取服务端录制结果成功。
- 参数

说明：status 为 2 时，才会有 fileType 和 filePath。

回调参数	类型	说明
recordInfo	json	recordId：服务端录制 id status：录制状态 0：录制中 1：上传中 2：录制成功 3：录制失败 fileType 文件类型 1：本地文件 2：OSS 3：AFTS filePath：文件路径

88. OnRemoteRecordInfoFailed(record_id, err_code, err_msg)

- 说明
获取服务端录制结果失败。
- 参数

回调参数	类型	说明
record_id	String	服务端录制 id
err_code	int	10351：连接状态错误 10352：服务端返回错误 10353：请求参数错误
err_msg	String	-

89. OnPauseRemoteRecordSucc(record_id)

- 说明
暂停服务端录制成功。
- 参数

回调参数	类型	说明
record_id	String	服务端录制 id

90. OnPauseRemoteRecordFailed(record_id, err_code, err_msg)

- 说明
暂停服务端录制失败。

• 参数

回调参数	类型	说明
record_id	String	服务端录制 id
err_code	int	10341：房间状态错误 10342：录制状态错误 10343：服务端错误
err_msg	String	-

91. OnResumeRemoteRecordSucc(record_id)

• 说明

恢复服务端录制成功。

• 参数

回调参数	类型	说明
record_id	String	服务端录制 id

92. OnResumeRemoteRecordFailed(record_id, err_code, err_msg)

• 说明

恢复服务端录制失败

• 参数

回调参数	类型	说明
record_id	String	服务端录制 id
err_code	int	10361：房间状态错误 10362：录制状态错误 10363：服务端错误
err_msg	String	-

93. OnRemoteRecordOver(record_id)

• 说明

服务端录制结束通知。此时文件在持久化中，可通过拉取接口获取结果。

• 参数

回调参数	类型	说明
record_id	String	录制 id

94. OnRemoteRecordingFailed(recordId, feedId, code, msg)

• 说明

服务端录制失败通知。

• 参数

回调参数	类型	说明
recordId	String	录制 id
feedId	String	与事件相关的流 id
code	int	12481 : eventCode 为 1 12482 : eventCode 为 2
msg	String	-

6.3.4 错误码

错误码	错误信息
-101	信令连接初始化超时
-102/-1004	信令连接初始化错误
-201	初始化房间失败：房间状态非法
-202	初始化房间失败：初始化房间参数非法
-301	创建房间失败：房间未初始化
-302	创建房间失败：服务器返回错误
-303	创建房间失败：创建房间超时
-304	创建房间失败：media 初始化超时（创建房间后自动启动发布情况下）
-401	加入房间失败：房间未初始化
-402	加入房间失败：参数错误
-403	加入房间失败：服务器返回错误
-404	加入房间失败：加房间超时
-501	发送邀请加房间失败：房间状态非法
-502	发送邀请加房间失败：服务器返回错误
-503	发送邀请加房间失败：参数错误
-601	发送答复邀请加房间失败：房间状态错误
-602	发送答复邀请加房间失败：服务器返回错误
-1071	发布媒体流失败：服务器返回错误
-1072	发布媒体流失败：未设置 video 标签
-1073	发布媒体流失败：超出发布限额
-1074	发布媒体流失败：发布建立 ice 连接超时
-1081	订阅媒体流失败：服务器返回错误
-1082	订阅媒体流失败：视频窗口 videoId 已使用
-1083	订阅媒体流失败：超出订阅限额
-1084	订阅媒体流失败：订阅建立媒体链路失败
-1085	订阅媒体流失败：订阅的窗口 videoId 为空
-10311	服务端录制失败：服务端返回录制启动失败

-10312	服务端录制失败：开启录制超时
-10313/-10314	服务端录制失败
-10351	获取服务端录制结果失败：连接状态错误
-10352	获取服务端录制结果失败：服务端返回错误
-10353	获取服务端录制结果失败：请求参数错误
-10291	发送文本消息失败：服务端返回错误
-10371	开始语音点播失败：服务端返回错误
-10372	开始语音点播失败：房间状态错误
-10373	开始语音点播失败：服务端通知开启失败
-10374	开始语音点播失败：点播参数有误
-10375	开始语音点播失败：开启点播超时
-10391	停止语音点播失败：服务端返回失败
-10392	停止语音点播失败：房间状态错误
-10393	停止语音点播失败：点播 id 错误
-12481/-12482	服务端录制失败
-20301	开启浏览器录制失败：room 状态错误
-20302	开启浏览器录制失败：已经在录制中
-20303	开启浏览器录制失败：录制 id 已经使用
-20304	开启浏览器录制失败：未发布或未订阅
-20305	开启浏览器录制失败：mediaType 参数错误
-20331	停止浏览器录制失败：房间状态错误
-20371	暂停浏览器录制失败：房间状态错误
-20372	暂停浏览器录制失败：未开启浏览器录制
-20391	继续浏览器录制失败：房间状态错误
-20392	继续浏览器录制失败：未开启浏览器录制
-20351	下载浏览器录制文件失败：房间状态错误
-20352	下载浏览器录制文件失败：未开启浏览器录制
-20353	下载浏览器录制文件失败：未停止浏览器录制
-30101	获取设备信息失败：浏览器不支持音视频
-30102	获取设备信息失败：获取媒体错误
-1101	切流失败通知：未发布
-1102	切流失败通知：参数错误
-1103	切流失败通知：media_type 错误
-1104	切流失败通知：获取音视频失败
-1105	切流失败通知：共享屏幕流为空
-1106	切流失败通知：获取视频失败
-1107	切流失败通知：获取文件流失败

-1108	切流失败通知：获取音频失败
-1109	切流失败通知：浏览器不支持音视频
-10341	暂停服务端录制：房间状态错误
-10342	暂停服务端录制：录制状态错误
-10343	暂停服务端录制：服务端错误
-10361	恢复服务端录制：房间状态错误
-10362	恢复服务端录制：录制状态错误
-10363	恢复服务端录制：服务端错误
5101	发布启动失败：视频启动失败
5102	发布启动失败：音频启动失败
5103	发布启动失败：浏览器不支持媒体
5104	发布启动失败：房间状态非法
5105	发布启动失败：共享桌面启动失败
5106	发布启动失败：获取文件流失败
5107	发布启动失败：发布文件不存在
5108	发布启动失败：发布区域不存在
5001	视频关闭通知：主动关闭
5002	视频关闭通知：离开房间被动通知关闭
5003	视频关闭通知：发布/订阅失败（服务器返回）
5004	视频关闭通知：建立音视频 ice 连接超时
5005	视频关闭通知：房间状态异常
5006	视频关闭通知：取消发布被动通知
5007	视频关闭通知：ice 连接异常
5008	视频关闭通知：退出房间关闭音视频

6.3.5 Web SDK 发布说明

V 1.4.4 (2021-03-04)

- 浏览器录制支持4路
- 浏览器录制流与流之间增加白边分屏

V 1.4.3 (2021-02-24)

- 增加 1280P 分辨率档位。
- 调整各档位码率。
- 支持自定义设置多个中转代理外网 IP。
- bundlePolicy 设置为 max-bundle，多路流共用同一个 ice。

V 1.4.2 (2021-01-21)

- 修复自动订阅模式下自定义中转服务器设置问题。

V 1.4.1 (2021-01-14)

- 修复 SDK 初始化后首次进入房间心跳异常问题。
- 修复服务端推送消息部分没回ack问题。
- 优化自定义区域共享。
- 摄像头接口松动监测，支持自动恢复。
- 支持发布纯音频不传 video 标签，订阅纯视频（纯音频）只传 video（audio）标签。
- 提供预启动摄像头/麦克风功能。

V 1.4.0 (2020-12-17)

- 修复自定义推流发布成功后没有回调问题。
- 修复帧率切换失效问题。
- 支持服务端录制每个画面打上发布 tag 水印。
- 修复直播 rtmp 地址设置无效问题。

V 1.3.9 (2020-12-09)

- 修复自定义推流模糊问题。

V 1.3.8 (2020-12-02)

- 修复房间中无发布流情况下启动录制会报错的 bug。

V 1.3.7 (2020-11-23)

- 音视频通话功能支持 H5：Android Chrome 浏览器（VP8/部分机型支持 H264）/iOS Safari（H264）。

V 1.3.6 (2020-11-06)

- 废弃 ASR/TTS 功能。

V 1.3.5 (2020-11-02)

- 服务端录制支持自定义布局（见主调接口 3、21 中的 mixPosition 参数）。
- 支持同一时间进行多个发布。

V 1.3.4 (2020-09-29)

- 服务端录制支持两种结束类型（除主动调用结束）（见主调接口 3、21 中的 endType 参数）。
- 修复外接麦克风插拔导致没声音问题。

V 1.3.3 (2020-09-02)

- 服务端录制支持静默功能（见主调接口 3、21 中的 silentRecord 参数）。

V 1.3.2 (2020-08-25)

- 修复客户端录制视频高度不一致，导致拼接视频后高度小的视频下方出现黑块的问题。
- 修复服务端录制在全房间没有发布情况下会出现初始化超时问题。

V 1.3.1 (2020-08-14)

- 服务端录制回调接口命名调整（兼容旧接口）。
- 服务端录制增加暂停/恢复功能。

V 1.3.0 (2020-08-12)

- 服务端录制增加参数 record_third_id，用于业务方区分调用批次。启动录制时传入（见主调接口 3、21），在初始化成功、启动成功、启动失败的回调携带返回（见被调接口 33、34、63）。
- 支持并发启动录制。

V 1.2.9 (2020-08-11)

- 优化音视频质量统计。
- 服务端录制放开开发限制。

V 1.2.8 (2020-08-07)

- 浏览器录制：升级接口，丰富配置（原主调接口 16 废弃，新接口为主调接口 40）。
- 服务端录制：增加 tagFilter、tagPostion 配置（见主调接口 3、21）。
- 初始化房间增加三方业务 ID 属性（见主调接口 3）。

V 1.2.7 (2020-07-28)

- 发布添加最大码率参数 publish_bitrate：
 - InitRoomConfig 接口（见主调接口 3）。
 - Publish 接口（见主调接口 9）。
 - ChangeMediaStream 接口（见主调接口 37）。
- 自定义推流支持 PDF 文件。
- 服务端录制支持时间戳水印、图片水印，支持多人录制（见主调接口 3、21）。
- 支持设置自定义设置中转服务器 defaultTurnServer（见主调接口 3）。

V 1.2.6

- 修复 ice 建立连接后清除定时器。

V 1.2.5

- 修复离开房间销毁浏览器端录制信息。

V 1.2.4

- 浏览器录制相关接口调整
 - 主调接口调整
 - StartRecord 删除 targetType 参数，添加 timeSlice 参数 recordType 值调整为 1,2,3 (见主调接口 16)。
 - DownloadRecord 删除 returnType 参数 (见主调接口 20)。
 - 被调通知接口调整
 - OnClientStartRecordSuccess 删除 targetType 参数 (见被调接口 67)。
 - OnClientStopRecordSuccess 删除 targetType 参数 (见被调接口 69)。
 - OnClientDownloadRecordSuccess 删除 returnType, clientRecordBlob 参数 (见被调接口 75)。
 - 添加 OnClientRecordBlob 接口 (见被调接口 81)。

V 1.2.3

- 添加相关
 - 添加切流成功通知 OnChangeMediaStreamSuccess (见被调接口 79)。
 - 添加切流失败通知 OnChangeMediaStreamFailed (见被调接口 80)。
 - ChangeProfile 添加参数 local_video_width 和 local_video_height (见主调接口 38)。
- 删除相关
 - 浏览器录制，停止录制 StopRecord 接口删除 targetType 参数 (见主调接口 17)。
 - video_profile_type 参数去掉选项 4 (见主调接口 3、9)。
- 内部调整。
 - 创建房间和加入房间 添加用户浏览器信息 (SDK 内部调整)。
 - 修复浏览器录制结果 blob 为空 (SDK 内部优化)。
 - 发布流 tag 更新 (SDK 内部调整)。
 - 添加 web-sdk 业务接入说明。

V 1.2.2

- 修复浏览器录制多次回调的 bug。

V 1.2.1

- publish_device 新增 2 种发布类型，发布文件和发布自定义页面区域（见主调接口 3、9）。
- 发布和初始化接口调整。
 - 新增 videoSource 参数, 摄像头的deviceId, 用于发布指定摄像头（见主调接口 3、9）。
 - 新增 audioSource 参数, 麦克风的deviceId, 用于发布指定麦克风（见主调接口 3、9）。
 - 新增 aspectRatioStrongDepend 参数，是否指定横纵比（见主调接口 3、9）。
 - 新增 aspectRatio 参数，横纵比（见主调接口 3、9）。
 - 新增 file 参数，用于发布图片或视频文件（见主调接口 9）。
 - 新增 part_of_screen_id 参数，用于发布指定页面区域（见主调接口 9）。
- 新增自定义切流接口 ChangeMediaStream（见主调接口 37）。
- 新增动态切换（分辨率、摄像头麦克风、横纵比）接口 ChangeProfile（见主调接口 38）。
- 新增音视频设备相关接口。
 - 新增获取音视频设备接口 GetDevices（见主调接口 39）。
 - 获取设备信息成功 OnGetDevicesSuccess（见被调接口 77）。
 - 获取设备信息失败 OnGetDevicesFailed（见被调接口 78）。

V 1.2.0

- 浏览器录制
 - 主调接口
 - StartRecord 开启浏览器录制（见主调接口 16）。
 - StopRecord 停浏览器录制（见主调接口 17）。
 - PauseRecord 暂停浏览器录制（见主调接口 18）。
 - ResumeRecord 继续浏览器录制（见主调接口 19）。
 - DownloadRecord 下载浏览器录制音视频（见主调接口 20）。
 - 回调接口
 - OnClientStartRecordSuccess 开启浏览器录制成功（见被调接口 67）。
 - OnClientStartRecordFailed 开启浏览器录制失败（见被调接口 68）。
 - OnClientStopRecordSuccess 停止浏览器录制成功（见被调接口 69）。
 - OnClientStopRecordFailed 停止浏览器录制失败（见被调接口 70）。
 - OnClientPauseRecordSuccess 暂停浏览器录制成功（见被调接口 71）。
 - OnClientPauseRecordFailed 暂停浏览器录制失败（见被调接口 72）。
 - OnClientResumeRecordSuccess 继续浏览器录制成功（见被调接口 73）。
 - OnClientResumeRecordFailed 继续浏览器录制失败（见被调接口 74）。
 - OnClientDownloadRecordSuccess 下载浏览器录制文件成功（见被调接口 75）。
 - OnClientDownloadRecordFailed 下载浏览器录制文件失败（见被调接口 76）。

V 1.1.9

- ASR 回调接口添加 sessionId。
 - 开始语音识别成功 OnStartAsrSuccess (见被调接口 49)。
 - 开始语音识别失败 OnStartAsrFailed (见被调接口 50)。
 - 停止语音识别成功 OnStopAsrSuccess (见被调接口 51)。
 - 停止语音识别失败 OnStopAsrFailed (见被调接口 52)。
 - 通知识别结果 OnAsrMessage (见被调接口 53)。
 - 长时间没有返回结果通知 OnAsrMsgIdle (见被调接口 54)。

V 1.1.8

- 去掉 ASR 连接时 uid 带 asr 后缀 (SDK 内部更新)。

V 1.1.7

- ASR 相关。
 - 说明：
 - tag 是业务透传参数。
 - status 表示会话是否结束，status=0 表示会话未结束，status=1 表示会话结束后的最终结果。
 - StartAsr 接口 config 添加 tag 业务透传参数 (见主调接口 30)。
 - OnStartAsrSuccess 接口增加返回参数 tag (见被调接口 49)。
 - OnStartAsrFailed 接口增加返回参数 tag (见被调接口 50)。
 - OnStopAsrSuccess 接口增加返回参数 tag (见被调接口 51)。
 - OnStopAsrFailed 接口增加返回参数 tag (见被调接口 52)。
 - OnAsrMessage 接口增加两个参数 OnAsrMessage(msg, sid)→OnAsrMessage(msg, sid,tag,status) (见被调接口 53)。
 - OnAsrMsgIdle 接口增加返回参数 tag (见被调接口 54)。
 - 修复 ASR 并发启动。
- TTS相关。
 - 开始语音合成 StartTts (见主调接口 35)。
 - 停止语音合成 StopTts (见主调接口 36)。
 - 开启语音合成成功 OnStartTtsSuccess (见被调接口 64)。
 - 开启语音合成失败 OnStartTtsFailed (见被调接口 65)。
 - 语音合成结果 OnTtsResult (见被调接口 66)。
- 录制相关。
 - 录制默认参数调整 (见主调接口 3、21)。

- 如果 recordTotalStream=0 , startTimeout 默认为 10s。
- 如果 recordTotalStream!=0 , startTimeout 默认为 5 分钟。
- startTimeout 优先级：业务指定优先于默认值。

V 1.1.6

- 更新 ASR 生产环境域名。

V 1.1.5

- chrome 版本号低于 67 , 弹窗提示。
- 发布分辨率档位调整：
 - 1: 720p (HD) , 1280x720, 16:9。
 - 2: VGA, 640x480, 4:3。
 - 3: 360p (nHD) , 640x360, 16:9。
 - 4: CIF, 352x288, 4:3。
- 发布时优先按照上述分辨率采集, 如浏览器不支持, 则按浏览器默认分辨率采集。
- 访问设备失败时, alert 弹窗提示原因。
- 服务端录制调整：
 - 启动超时时间 startTimeout : 范围在 3 ~ 10s 之间, 默认 5s。
 - 录制启动成功标准 recordTotalStream : 成功订阅流, 订阅流数默认为 0 , 之前默认为 2 (2 即成功录下双方) 。
 - 服务端录制增加初始化成功接口 OnInitRecordSucc (见被调接口 63) 。说明：正常情况下, 开启录制后会先回调 OnInitRecordSucc , 之后再回调 OnStartRecordSucc , 两个回调之间的时间间隔视业务设置 recordTotalStream 而定。
- ASR 内部优化修复状态管理漏洞。

V 1.1.4

- ASR 内部优化修复识别不准的问题。

V 1.1.3

- 录制通知回调 SDK 内部优化
- ASR 识别结果：完善说明, 根据 procType 来确定格式 (见被调接口 53) 。

V 1.1.2

- 增加点播通知功能, StartVod 接口参数变更 vodParams (见主调接口 32) 。
- 增加录制通知功能, InitRoomConfig 和 StartRemoteRecord 接口中 recordParam 参数添加 2 个属性: startTimeout 和 recordTotalStream (见主调接口 3、21) 。

- 增加录制过程中录制中断接口 OnRecordingFailed (见被调接口 62)。
- 录制默认音频采样率 sampleRate 从 24000 修改为 48000 (见 InitRoomConfig 和 StartRemoteRecord)。
- 废弃 OnRecordInfo 接口, 通过 OnStartRecordSucc 替代 (见被调接口 32)。

V 1.1.1

- 修复 ASR 验签失败 (SDK 内部改动,接口未改动)。

V 1.1.0

- 修复离开房间未关闭摄像头 (SDK 内部改动, 接口未改动)。

V 1.0.9

- 增加开始语音识别失败接口 OnStartAsrFailed (见被调接口 50)。
- 语音识别结果接口名从 OnAsrMsg 修改为 OnAsrMessage (见被调接口 53)。
- 开始语音识别接口添加参数引擎类型 procType (见主调接口 30)。
- 服务端录制结束通知接口名从 OnRecorderOver 修改为 OnRecordOver (见被调接口 61)。
- 增加视频清晰度参数 video_profile_type (见主调接口 3、9)。
- 弱网回调接口添加 sid 参数 (见被调接口 48)。
- 废弃指令消息相关接口 (见主调接口 27 和被调接口 41 ~ 44)。

V 1.0.8

- 增加错误日志上报保存到本地 (见主调接口 34)。

V 1.0.7

- 增加语音识别功能 (见主调接口 30 ~ 31, 被调接口 49 ~ 54)。
- 增加语音点播功能 (见主调接口 32 ~ 33, 被调接口 55 ~ 59)。
- 增加自定义截屏文件格式 (见主调接口 15)。
- 增加实时音量可视化功能 (见主调接口 3、9、10 的 need_volume_analyser 字段, 被调接口 60 的回调接口)。
- 增加服务端录制结束通知回调 (见被调接口 61)。
- 音视频关闭通知接口添加 sid 参数 (见被调接口 30)
- 邀请加入房间添加 channelType 参数 (见主调接口 6)。

V 1.0.6

- 修改发布弱网频率告警 (见主调接口 28 SetPublishWeakBitrateLimit)。
- 添加订阅弱网频率告警 (见主调接口 29 SetSubscribeWeakBitrateLimit)。

- 添加退出回调参数，退出类型leaveType（见被调接口 45 OnLeaveRoom）。
- 启动服务端录制添加参数 recordParam（见主调接口 21 StartRemoteRecord）。
- 添加退出类型参数（见被调接口 46 OnParticipantLeaveRoom）。
- 添加服务端录制相关参数 recordParam 和 recordStrongDepend（见主调接口 3 InitRoomConfig）。

V 1.0.5

- 添加 获取录制信息接口（见 GetRecordInfo）。
- 添加 获取服务端录制结果成功（见 OnRecordInfoSucc）。
- 添加 获取服务端录制结果失败（见 OnRecordInfoFailed）。
- 添加 退出类型参数 OnParticipantLeaveRoom（见被调接口 46）。
- 添加 服务端是否录制 defaultRecord参数（见主调接口 3 InitRoomConfig）。

V 1.0.4

- 添加 发布共享桌面功能。
- 添加 弱网回调接口（见被调接口 46）。
- 添加 设置弱网码率告警下限（见主调接口 27）。
- 添加 参数 publish_device 标识发布类型（见主调接口 3 和主调接口 9）。

V 1.0.3

- 支持签名在页面重写（被调接口 3）。
- 截屏如果不指定宽高，则默认为视频窗口的宽高。
- TakePicture() 接口参数宽高设置默认值（主调接口 15）。
- 浏览器录制，不在录制中时，停止录制直接返回（主调接口 22）。
- StartRemoteRecord 接口添加可选参数服务端录制路径（主调接口 21）。
- publish_video_id 由必填更正为根据 media_type 选填（主调接口 3 和主调接口 9）。
- StartRemoteRecord 接口添加可选参数 filePath 服务端录制路径（主调接口 21）。
- OnSubscribeFailed(sid,err_code, err_msg) 接口去掉 feed 参数（被调接口 23）。
- 如果发布的是纯音频，则 sid 获取方式为 publish_streamId_id 的 name 值，具体参考 meeting.html。
- InitRoomConfig 接口添加服务端录制路径的可选参数 filePath，通过 OnRecordInfo 接口回调 recordId（见主调接口 3 和被调接口 32）。

V 1.0.2

- 添加录制功能：主调接口 16 ~ 22 和被调接口 31 ~ 35。



蚂蚁集团
ANT GROUP

