ALIBABA CLOUD

阿里云

视频直播

视频直播公共云合集

文档版本：20220712

阿里云

法律声明

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

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

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

目录

1.快速入门 06
2.广目监播（邀测） 16
2.1. 功能简介 16
2.2. 广目监播使用指南 17
3.SDK参考 21
3.1. 推流SDK（旧版） 21
3.1.1. 关于推流SDK V4.0.2升级的公告 21
3.1.2. Android推流SDK 21
3.1.2.1. 基础概念 21
3.1.2.2. 集成说明 21
3.1.2.3. 使用说明 23
3.1.2.4. Demo编译 26
3.1.2.5. API参考及配置列表 27
3.1.3. iOS推流SDK 52
3.1.3.1. 基础概念 52
3.1.3.2. 集成说明 52
3.1.3.3. 使用说明 55
3.1.3.4. Demo编译 58
3.1.3.5. API参考及配置列表 59
3.2. 播放器SDK 83
3.2.1. 直播播放器SDK简介 83
3.2.2. SDK下载 85
3.2.3. Android端直播播放器SDK 86
3.2.3.1. 集成Android端直播播放器 86
3.2.3.2. 实现Android端播放功能 88
3.2.3.3. 接口说明 90

3.2.4. iOS端直播播放器SDK 97
3.2.4.1. 集成iOS端直播播放器 97
3.2.4.2. 实现iOS端播放功能 99
3.2.4.3. 接口说明 100
4.常见问题 109

1.快速入门

通过阅读本文，您可以了解视频直播的基本流程。

前提条件

- 您已经注册了阿里云账号并完成账号实名认证。注册地址请参见阿里云官网。注册指引请参见注册阿里云账号。实名认证指引请参见个人实名认证或企业实名认证。
- 您已经准备好推流域名和播放域名，并完成备案。具体操作，请参见备案。

> **说明**
> - 如果您的直播中心在海外且加速区域采用**海外及港澳台加速**，则域名无需经过备案。
> - 如果推流域名和播流域名是同一个一级域名下的两个二级域名，则只需要将一级域名进行备案。

视频直播基本流程

1. 开通服务
2. 添加推流域名和播流域名
3. 配置CNAME
4. 关联推流域名和播流域名
5. 配置自定义鉴权（可选）
6. 生成推流地址和播放地址
7. 直播推流
8. 直播播放

开通服务

> **说明** 您可以观看视频讲解学习如何开通直播服务，详情请参见开通直播服务。

1. 登录阿里云官网，在顶部菜单栏选择**产品 > 视频服务 > > 视频直播**，进入**视频直播**产品首页。
2. 单击**立即开通**，进入开通页面。
3. 在**视频直播开通页**，选择计费方式为**按使用流量计费**或**按带宽峰值计费**，并选中**视频直播服务协议**。
4. 单击**立即开通**。

添加推流域名和播流域名

使用视频直播服务前，您需要分别添加推流域名和播流域名。

> **说明** 您可以观看视频讲解学习如何添加域名，详情请参见添加域名。

1. 登录视频直播控制台。
2. 在左侧导航栏单击**域名管理**，进入**域名管理**页面。
3. 单击**添加域名**。

4. 根据实际需求配置域名详情。

参数	参数分类	说明
加速域名	无	输入您需要加速的域名。注意事项如下： ○ 加速域名一般使用子域名，例如：`demo.aliyundoc.com`。 ○ 直播服务暂不支持添加泛域名，例如：`*.aliyundoc.com`。 ○ 加速域名不允许重复添加，如果出现**域名已添加**的提示，请检查您的域名是否已经添加到其他云产品中，例如视频点播、全站加速、SCDN和视频监控，您也可以提交工单处理。 ○ 开通视频直播服务后，每个账户最多可添加20个域名，如果您域名的总带宽日均峰值大于50 Mbps，且业务无风险，可提交工单申请增加域名个数。 ○ 加速内容必须合法且符合业务规范，详情请参见域名准入标准。 **说明** 子域名必须在同一个账号下。您在添加域名时，直播服务会进行检查，如果子域名被添加到不同账号，系统会报错。如果您无法自行解决，请提交工单。
直播中心	无	直播中心用来提供云上直播流的媒体处理服务，如截图、录制、转码、云导播等。 需要关联的直播推流域名与播流域名的直播中心所在地域要保持一致，初次配置后不可更改。 关于如何选择直播中心，请参见如何选择直播中心？。

参数	参数分类	说明
业务类型	○ **播流域名** ○ **推流域名**	指域名的加速类型，推流域名与播流域名分别对应推流加速与播流加速。初次配置后不可更改。 说明 ○ 添加推流域名时，**业务类型**请选择**推流域名**；添加播流域名时请选择**播流域名**。 ○ 您无法将同一域名既添加为推流域名，又添加为播流域名。
CDN 加速区域	**中国内地** **全球加速** **海外及港澳台加速**	指域名可执行加速的地域。 ○ **中国内地** 当主播在中国内地时，推流域名应选择**中国内地**。 当观众在中国内地时，播流域名应选择**中国内地**。 域名需要在工信部备案，推荐您进入阿里云ICP代备案管理系统进行备案。 ○ **全球加速** 当主播在中国内地、中国香港、中国澳门、中国台湾以及其他国家和地区时，推流域名应选择**全球加速**。 当观众在中国内地、中国香港、中国澳门、中国台湾以及其他国家和地区时，播流域名应选择**全球加速**。 域名需要在工信部备案，推荐您进入阿里云ICP代备案管理系统进行备案。 ○ **海外及港澳台加速** 当主播在海外及中国香港、中国澳门、中国台湾时，推流域名应选择**海外及港澳台加速**。 当观众在海外及中国香港、中国澳门、中国台湾时，播流域名应选择**海外及港澳台加速**。 如果选择**海外及港澳台加速**且直播中心在海外，则域名无需在工信部备案。 如果选择**海外及港澳台加速**且直播中心在中国境内，则域名需要在工信部备案。推荐您进入阿里云ICP代备案管理系统进行备案。 说明 关于如何选择CDN加速区域，请参见如何选择域名的CDN加速区域？。

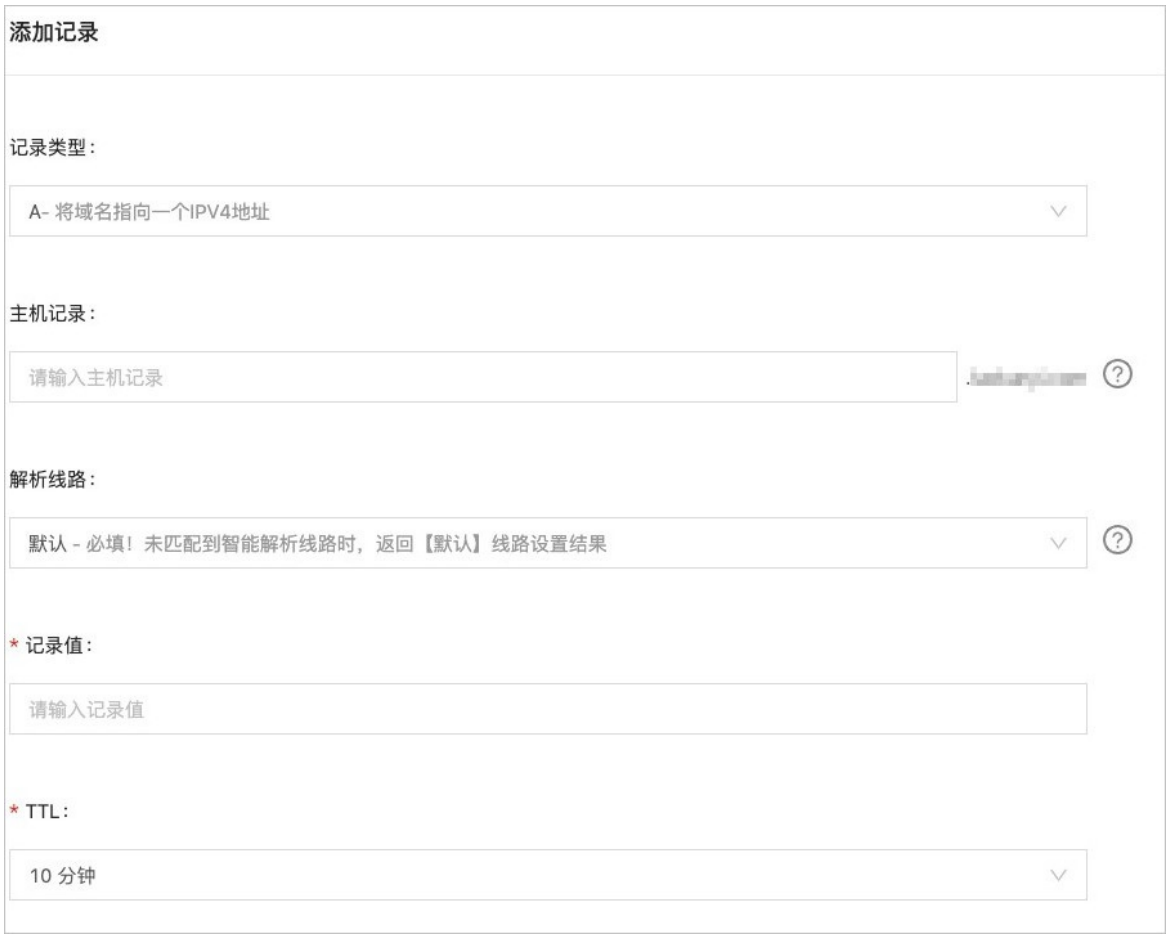
5. 单击**提交**。

配置CNAME

启用视频直播加速服务，需要将您的域名（推流域名和播流域名）指向阿里云视频直播分配的CNAME地址。此处以DNS服务商为阿里云（原万网）为例介绍配置CNAME的方法。更多DNS服务商及操作详情，请参见配置CNAME域名解析。

> **说明**
>
> - 您可以观看视频讲解学习如何解析CNAME，详情请参见解析CNAME。
> - 由于阿里云CDN校验域名的DNS解析记录的服务器部署在中国内地（大陆）。如果您对域名做了分区域DNS解析配置，例如仅对域名的中国内地以外区域（中国香港、中国澳门、中国台湾、其他国家和地区）配置了阿里云CDN的CNAME地址，校验服务器将无法解析到该CNAME地址，且在CDN控制台该域名的CNAME状态会显示为**待配置**，这种情况不影响CDN的加速服务。
> - 阿里云CDN、全站加速、直播、点播以及SCDN产品的CNAME域名仅可以作为阿里云CDN的调度解析使用，对于恶意使用CNAME域名的行为，阿里云有权清退对应的域名和账号。

1. 登录云解析DNS控制台。
2. 在左侧导航栏选择**域名解析**，进入域名解析页面。
3. 单击待设置域名**操作**列的**解析设置**。
4. 单击**添加记录**，配置信息如下所示：



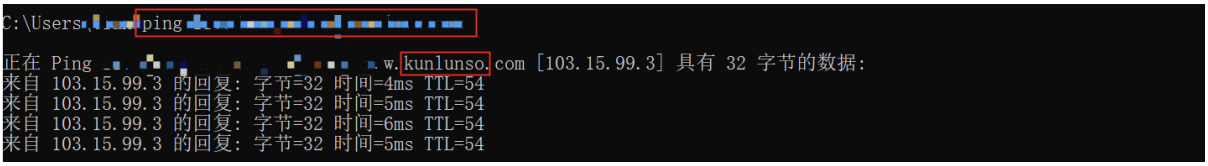
参数	描述
记录类型	选择**CNAME**，将域名指向另外一个域名。

参数	描述
主机记录	输入域名的前缀，常见前缀和对应的示例域名如下表所示。
解析路线	选择默认值。
记录值	输入记录的CNAME值。
TTL	选择默认值。

前缀	示例域名	说明
demo	demo.aliyundoc.com	解析子域名。
@	aliyundoc.com	解析主域名。
*	*.aliyundoc.com	泛解析。

5. 单击**确定**，完成CNAME配置。

验证生效：执行**ping <域名>**或**dig <域名>**命令，如果显示解析至 `*.*kunlun*.com` ，表示CNAME配置已生效。



关联推流域名和播流域名

目前视频直播采用边缘推流方式，您需要关联推流域名和播流域名，才能进行直播的推流和播流。

> **说明** 您可以观看视频讲解学习如何关联推流域名与播流域名，详情请参见关联推流域名与播流域名。

1. 在**域名管理**页面，选择您添加的播流域名，单击**操作**列的**域名配置**，进入**直播管理 > 基本配置**页面。
2. 单击**基础信息**页签。
   在此页面，您可以查看该域名的CNAME、创建时间、业务类型、直播中心、加速区域、流数限制。

> **说明**
>
> - 直播配额管理目前仅支持华北2（北京）、华东2（上海）、华南1（深圳）、亚太东南1（新加坡）中心，其余中心提升配额可提交工单。
> - 关于直播配额管理（流数限制），更多信息请参见配额管理。
> - 如需要修改加速区域设置，请参见修改加速区域。

3. 单击**推流信息**页签。
4. 单击**添加推流信息**或**推流域名**后的编辑图标，选择需要关联的推流域名。

配置自定义鉴权（可选）

URL鉴权功能旨在保护用户站点的内容资源不被非法站点下载盗用。建议您保持默认开启状态，否则会出现直播被盗录、盗播的风险。

当鉴权功能开启时，有默认鉴权和自定义鉴权两种鉴权方式，您可按需选择默认鉴权或自定义鉴权。采用默认鉴权无需设置。如不采用默认配置，您也可以对鉴权**主KEY**、**备KEY**、**有效时长**进行自定义，并输入原始URL。

> **说明** 您可以观看视频讲解学习如何配置URL鉴权，详情请参见配置URL鉴权。

1. 在**域名管理**页面，选择您要配置的播流域名，单击**操作**列的**域名配置**。
2. 选择**直播管理 > 访问控制**，进入**访问控制**页面。
3. 单击**URL鉴权**页签，然后单击**修改配置**。
4. 配置URL鉴权信息。

   配置项和说明如下表所示。

配项置	说明

配项置	说明
鉴权类型	视频直播播流域名仅支持使用鉴权A，来实现对源站资源的有效保护。 **说明** URL鉴权错误，都会返回403报错，请重新计算。 ○ MD5计算类错误 例如：`X-Tengine-Error:denied by req auth: invalid md5hash=de7bfdc915ced05e17380a149bd760be` ○ 时间类报错 例如：`X-Tengine-Error:denied by req auth: expired timestamp=1439469547`
主KEY	当您添加域名时，控制台会随机生成一个主用密码。您可以通过在直播控制台左侧的导航栏选择**域名管理**，选择您需要配置的域名，单击**域名配置 > 访问控制 > URL鉴权**，在URL鉴权页面查看主用密码。您还可以重新输入自定义的鉴权方式对应的主用密码。
备KEY	输入自定义的鉴权方式对应的备用密码。
有效时长	有效时长是指该URL鉴权仅在有效期内可用来发起推流或者播放。直播推流和播放都是长链接行为，在有效期内已经发起的直播推流或者播放行为，不会因为有效期过期而中止，但新发起的推流和播放请求会因为有效时长过期会失败。 初始化时，默认为30分钟。你可以自定义输入鉴权方式的默认有效时长，单位：分钟。

生成推流地址和播放地址

推流地址和播放地址是用于推送直播流和拉取直播流播放的域名，您可参考以下步骤生成推流地址和播放地址。

说明 您可以观看视频讲解学习如何生成推流地址和播放地址，详情请参见生成推流地址与播流地址。

1. 登录视频直播控制台。
2. 在左侧导航栏选择**地址生成器**。
3. 完成以下配置获取推流地址和播放地址。

配置项	描述
播流域名	在**播流域名**下拉框选择播流域名。
鉴权	默认为A方式，有效时长30分钟。 **说明** 如需修改有效时长，请参见配置URL鉴权。
关联的推流域名	○ 已经关联的推流域名会自动填充推流域名，未关联推流域名您需先完成域名关联。 ○ 如果是纯拉流的场景，可跳过此配置项不需选择关联的推流域名。
AppName	在编辑框输入AppName。 **说明** AppName不超过256字符，支持数字、大小写字母、短划线（-）、下划线（_）、等号（=）。
StreamName	在编辑框输入StreamName。 **说明** StreamName不超过256字符，支持数字、大小写字母、短划线（-）、下划线（_）、等号（=）。

配置项	描述
转码模版	选择您已配置的**转码模版**，不填则默认为原画。
开始生成	单击**开始生成**，获取推流地址和播放地址。 **说明** • 视频直播根据流名称（StreamName）而不是根据应用名称（AppName）来判断直播流是否唯一。如果您设置了不同的应用名称，还要保证直播流名称不同，才能保证最终的直播流不同。如果需要一个域名下不同的应用名称（AppName）推相同的流（StreamName），需开启App隔离功能，您可提交工单申请配置。 • 关联播流域名和推流域名则会生成推流地址。 • 未配置转码模板，则生成原画播放地址。已配置转码模板，则生成转码播放地址。

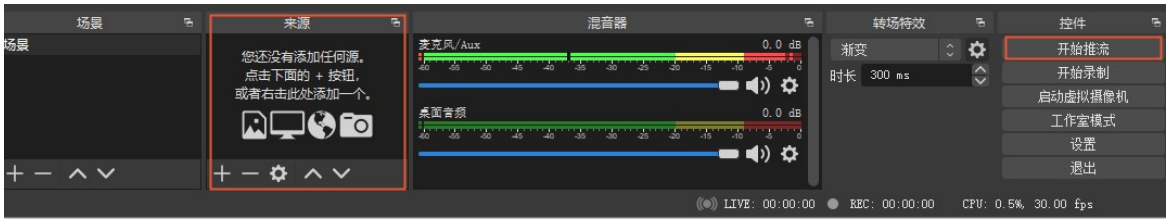
直播推流

直播推流是把采集阶段封装好的音视频直播流推送到阿里云直播服务中心的过程。此处以使用OBS推流工具推流为例进行介绍。更多推流工具操作详情，请参见直播推流。

1. 下载并安装OBS推流工具。下载地址，请参见OBS Studio。
2. 运行OBS推流工具。
3. 在菜单栏中选择**文件 > 设置**。
4. 在设置页面选择**推流**，配置信息如下所示：

参数	描述
服务	选择**自定义**。
服务器	输入已经生成的鉴权推流地址，例如：`rtmp://demo.aliyundoc.com/app/stream?auth_key=1543302081-0-0-9c6e7c8190c10bdfb3c0**********`。
串流密钥	串流密钥为空。

5. 在OBS主界面**来源**区域添加推流源，然后单击**开始推流**。



直播播放

直播播放是将直播服务中心已有直播内容分发到播放器进行播放的过程。此处以通过VLC播放器播放为例进行介绍，更多播放器操作详情，请参见直播播放。

1. 下载并安装VLC播放器。下载地址，请参见VLC media player。
2. 运行VLC播放器。
3. 在菜单栏中选择**媒体 > 打开网络串流**。
4. 在**网络**页签中输入网络URL，即播放地址，例如：`rtmp://example.aliyundoc.com/app/stream?auth_key=1543300311-0-0-d47ce016332bf280cf275********`。
5. 单击**播放**。

其他功能

视频直播提供了权限管理、基础服务、直播加速、直播安全、直播延时、HTTP配置、直播流媒体处理、云端合流、云导播台、直播管理、数据统计和日志管理等丰富的功能。您可以阅读开发指南概述，了解以上功能的使用方法。

2.广目监播（邀测）

2.1. 功能简介

广目系统为各类直播项目提供实时监播功能，并对帧率码率变化、音视频同步、延迟和卡顿等异常情况时进行告警，为各类专业直播保障护航。本文主要介绍了广目系统的核心功能、使用场景等相关内容。

> **说明** 目前广目监播已开始邀测，邀测期间功能免费。使用广目监播功能，请参见广目监播使用指南。

产品Demo

序号	描述
①	流时间戳，显示位置：左上角。
②	音频音量，区分左右声道，音量使用Peak计算标准。显示位置：左下角。
③	实时帧率和码率。显示位置：右上角。

序号	描述
④	告警详情，显示位置：右下角。 支持的告警类型： • vfps：视频帧率告警。 • afps：音频帧率告警。 • eof：断流告警。 • a-v：音视频不同步告警。 • wc（warning count）：表示告警总数。

核心功能

- 支持4分屏、8分屏、12分屏的盯屏功能。
- 实时看板以及数据化显示直播流帧率、码率等信息。
- 出现帧率异常、码率异常、音视频同步异常、卡顿、延迟等情况支持实时告警。

使用场景

- 视频平台：为平台开展的各类直播，尤其是重大直播、重要场次直播提供监播保障。
- 企业服务：多场次并发监播，为企业客户的直播节目提供保障。
- 机构服务：多场次并发监播，为机构主播们的日常直播提供保障。

问题咨询

广目系统产品使用过程中有任何问题或建议，请通过搜索群号：35080444，或扫码加入钉钉群进行沟通。

2.2. 广目监播使用指南

阅读本文，您可以了解通过视频直播控制台添加广目监播场次和管理的方法。

前提条件

在直播监播控制台上播放监播流，需要完成对监播输出地址的拉流域名进行HTTPS证书和跨域访问配置。

- 跨域访问配置：配置HTTP头Access-Control-Allow-Origin，取值为 `https://live.console.aliyun.com`，操作步骤请参见配置HTTP消息头。
- HTTPS配置：配置有效HTTPS证书，操作步骤请参见配置HTTPS安全加速。

背景信息

视频直播提供广目监播功能，可随时监控线上直播效果与质量，并对帧率码率变化、音视频同步、延迟和卡顿等异常情况时进行告警，为各类专业直播保障护航。

操作步骤

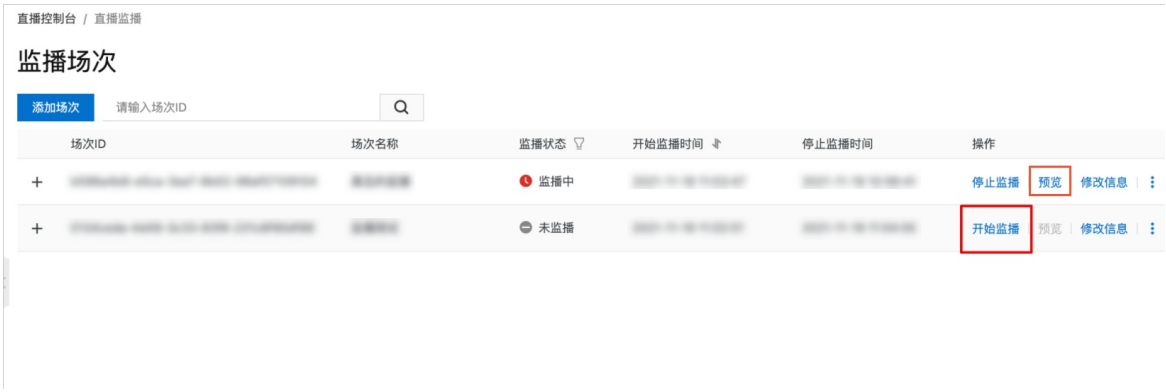
1. 登录直播控制台。
2. 在左侧导航栏单击**广目监播 > 监播场次 > 添加场次**可进行添加监播场次，同时可以在**操作**列管理监播场次。
3. 添加场次。

配置项	配置描述
场次名称	在输入框填写场次的名称。
输出分辨率	在下拉框选择您需要的输出分辨率，支持横竖屏多种分辨率格式。

配置项	配置描述
直播中心	在下拉框选择对应的直播中心。
合流域名	在下拉框选择合流输出的域名，如没有合适的合流域名可点击**去添加**，具体操作请参见添加域名。
监播流配置	支持手动添加或从直播控制台选取添加，最多可添加12路。
手动添加	单击**手动添加**，输入直播流名称和流地址，同时支持排序和删除功能。
从直播控制台选取	单击**从直播控制台选取**，在域名下拉框选择对应的域名以及勾选对应的在线直播流。
确定	单击**确定**完成添加场次配置。

4. 开始监播。

   配置添加场次完成，从监播场次操作列中单击**开始监播**即可启动。单击**预览**即可进入监播页面。


直播控制台 / 直播监播
监播场次
添加场次
请输入场次ID
场次ID
场次名称
监播状态
开始监播时间
停止监播时间
操作
监播中
停止监播
预览
修改信息
未监播
开始监播
预览
修改信息

3.SDK参考

3.1. 推流SDK（旧版）

3.1.1. 关于推流SDK V4.0.2升级的公告

自2021年7月起，阿里云视频直播服务将逐步停止对推流SDK V4.0.2的支持与维护，请升级并使用最新版本的SDK。

新版推流SDK，请参见产品介绍。

3.1.2. Android推流SDK

3.1.2.1. 基础概念

通过阅读本文，您可以了解Android推流SDK中一些专业术语的基本概念。

- 码率控制：一种编码的优化算法，用于控制视频流码率的大小。同样的视频编码格式，码率越大，包含的信息越多，对应的图像也就越清晰，反之亦然。
- 视频丢帧：发送视频帧时，如因网络较差导致视频帧堆积严重，可以通过丢弃视频帧来缩短推流的延时。
- 耳返：指主播可以通过耳机实时听到自己的声音。例如，当主播带上耳机唱歌时，需要把握音调，这时就需要开启耳返功能。因为声音通过网络传入耳朵和通过空气传入耳朵差异很大，而主播需要直接听到观众端的效果。
- 混音：把多种来源的声音整合至一个立体音轨或单音音轨中。Android推流SDK支持音乐和人声的混音。
- 合流：把多种来源的视频图像数据根据位置叠加到同一个视频画面中。

3.1.2.2. 集成说明

通过阅读本文，您可以了解Android推流SDK的集成方法以及配置权限的方法。

通过本地依赖方式集成SDK

1. 下载并解压Android端推流SDK。

   > **说明** 此版本推流SDK已停止使用，推荐使用新版推流SDK。

2. 新建项目，复制推流SDK文件AliLiveSdk.aar和播放器相关文件到App模块下的libs文件夹中。

3. 修改工程的build.gradle文件，在 `allprojects` 的 `repositories` 节点中增加 `flatDir` 的设置以及

阿里Maven地址。

```
maven {
    url 'https://maven.aliyun.com/repository/releases'
}
flatDir {
    dirs 'libs'
}
```

4. 修改App中的build.gradle文件。

```
dependencies {
    // 添加对libs文件夹中aar文件的依赖
    implementation fileTree(include: ['*.jar', '*.aar'], dir: 'libs')
}
```

通过Maven依赖方式集成SDK

1. 新建项目。
2. 修改工程的build.gradle文件，在 `allprojects` 的 `repositories` 节点中增加 `flatDir` 的设置以及阿里Maven地址。

```
maven {
    url 'https://maven.aliyun.com/repository/releases'
}
flatDir {
    dirs 'libs'
}
```

3. 修改App中的build.gradle文件，在 `dependencies` 节点中增加推流SDK、播放器SDK及Conan的引用。

```
dependencies {
    // 添加对libs文件夹中aar文件的依赖
    implementation fileTree(include: ['*.jar', '*.aar'], dir: 'libs')
    // 添加推流SDK的依赖
    implementation 'com.alivc.live:AliLiveSdk:4.0.2'
    // 添加播放器core的依赖
    implementation 'com.aliyun.video.android:core:1.2.2'
    // 添加播放器SDK的依赖
    implementation 'com.aliyun.sdk.android:AliyunPlayer:5.2.3-full'
    // 添加conan的依赖
    implementation 'com.alivc.conan:AlivcConan:1.0.4'
    // 添加artcdemuxer的依赖
    implementation 'com.aliyun.sdk.android:AlivcArtc:5.2.3'
    // 添加netsdk的依赖
    implementation 'com.aliyun.rts.android:RtsSDK:1.5.0'
}
```

配置权限

修改工程的AndroidManifest.xml文件，添加如下权限配置。

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

3.1.2.3. 使用说明

通过阅读本文，您可以了解Android推流SDK的使用方法。

RTMP推流

> **说明** 阿里云视频直播不允许同一时间向同一个推流URL进行多路推流（第二路推流会被拒绝）。

1. 创建AliLiveEngine。

```
//创建RTMP相关配置对象
AliLiveRTMPConfig rtmpConfig = new AliLiveRTMPConfig();
//初始化码率配置
rtmpConfig.videoInitBitrate = 1000;
rtmpConfig.videoTargetBitrate = 1500;
rtmpConfig.videoMinBitrate = 600;
//创建直播推流配置
AliLiveConfig mAliLiveConfig = new AliLiveConfig(rtmpConfig);
//初始化分辨率、帧率、是否开启高清预览、暂停后默认显示图片
mAliLiveConfig.videoFPS = 20;
mAliLiveConfig.videoPushProfile = AliLiveConstants.AliLiveVideoPushProfile.AliLiveVideo
Profile_540P;
mAliLiveConfig.enableHighDefPreview = false;
mAliLiveConfig.pauseImage = bitmap;
mAliLiveConfig.accountId = "";
AliLiveEngine mAliLiveEngine = AliLiveEngine.create(PushActivity.this, mAliLiveConfig);
```

2. 开始预览。

```
//创建预览显示窗口
AliLiveRenderView mAliLiveRenderView = mAliLiveEngine.createRenderView(false);
//添加预览显示窗口到布局中
addSubView(mAliLiveRenderView);
//设置预览显示模式
mAliLiveEngine.setPreviewMode(AliLiveRenderModeAuto, AliLiveRenderMirrorModeOnlyFront);
//开始预览
mAliLiveEngine.startPreview(mAliLiveRenderView);
```

3. 开始推流。

```
mAliLiveEngine.startPush(mPushUrl);
```

4. 停止拉流。

```
//停止预览
mAliLiveEngine.stopPreview();
//停止推流
mAliLiveEngine.stopPush();
//销毁liveEngine
mAliLiveEngine.destroy();
mAliLiveEngine = null;
```

RTMP拉流

1. 创建播放器。

```
//创建播放器
mAliPlayer = AliPlayerFactory.createAliPlayer(this.getApplicationContext());
//设置自动播放
mAliPlayer.setAutoPlay(true);
//设置mMaxDelayTime，建议范围500~5000，值越小，直播时延越小，卡顿几率越大，根据需要自行决定
PlayerConfig config = mAliPlayer.getConfig();
config.mMaxDelayTime = 1000;
mAliPlayer.setConfig(config);
```

2. 添加显示的SurfaceView。

```
    mSurfaceView.getHolder().addCallback(this);
    @Override
    public void surfaceCreated(SurfaceHolder holder) {
        if (mAliPlayer!=null){
            mAliPlayer.setSurface(holder.getSurface());
        }
    }
    @Override
    public void surfaceChanged(SurfaceHolder holder, int format, int width, int height)
{
        if (mAliPlayer!=null){
            mAliPlayer.surfaceChanged();
        }
    }
    @Override
    public void surfaceDestroyed(SurfaceHolder holder) {
        if (mAliPlayer!=null){
            mAliPlayer.setSurface(null);
        }
    }
```

3. 开始拉流。

```
//给播放器设置拉流地址
UrlSource source = new UrlSource();
source.setUri(mPullUrl);
mAliPlayer.setDataSource(source);
//开始拉流
mAliPlayer.prepare();
```

4. 停止拉流。

```
mAliPlayer.stop();
mAliPlayer.setSurface(null);
mAliPlayer.release();
mAliPlayer = null;
```

主播PK

1. 主播A发送请求至appserver，获取目前在线且符合要求的主播列表。

```
mSocketHandler.send(getSendMessage(CMD_GET_ROOM_LIST));
```

获取到主播列表后渲染到页面。

```
mOnlineRoomView.setRoomList(onlineRoomList);
```

2. 主播A在列表中找到一个主播B，请求PK。

```
mSocketHandler.send(getSendMessage(CMD_APPLY_PK,room.getUserId(),room.getRoomId()));
```

3. 主播B收到主播A的申请，并弹出提示。

```
showApplyPkDialog(applyPkNoticeBean);
```

4. 主播B同意了主播A的申请后，开始拉RTC流，并发送同意PK信令。

```
//开始订阅主播A
startPk(mLastApplyPkNotice.getFromRtcPullUrl());
//开始发送同意PK信令
mSocketHandler.send(getSendMessage(CMD_APPROVE_PK,true ,mLastApplyPkNotice.getFromUserI
d(),mLastApplyPkNotice.getFromRoomId() ));
```

5. 主播A收到主播B的同意申请后，开始拉RTC流。

```
//开始PK
startPk(approvePkNoticeBean.getRtcPullUrl());
```

6. `onSubscribeResult` 回调中收到订阅成功的通知，调整预览和订阅画面的布局。

```
showPkSurfaceView(true);
```

7. 观众端播放的RTMP地址有变化，收到NotifyPublish信令后，重新构建播放器播放新地址。

```
//判断是否和上次的主播拉流地址一致，不一致则重新拉流
if (!mRoomInfo.getRtmpPullUrl().equals(noticePublish.getAnchorRtmpPullUrl())){
    mRoomInfo.setRtmpPullUrl(noticePublish.getAnchorRtmpPullUrl());
    startPlayRTMP();
}
```

8. 主播B停止PK，发送停止PK信令，停止拉RTC流，并恢复PK前画面。

```
//发送停止PK信令
mSocketHandler.send(getSendMessage(CMD_CANCEL_PK));
//恢复PK之前的UI
showPkSurfaceView(false);
```

9. 观众端播放的RTMP地址有变化，收到NotifyPublish信令后，重新构建播放器播放新地址。

```
//判断是否和上次的主播拉流地址一致，不一致则重新拉流
if (!mRoomInfo.getRtmpPullUrl().equals(noticePublish.getAnchorRtmpPullUrl())){
    mRoomInfo.setRtmpPullUrl(noticePublish.getAnchorRtmpPullUrl());
    startPlayRTMP();
}
```

3.1.2.4. Demo编译

通过阅读本文，您可以了解Android端Demo的运行环境要求及编译方法。

环境要求

硬性要求

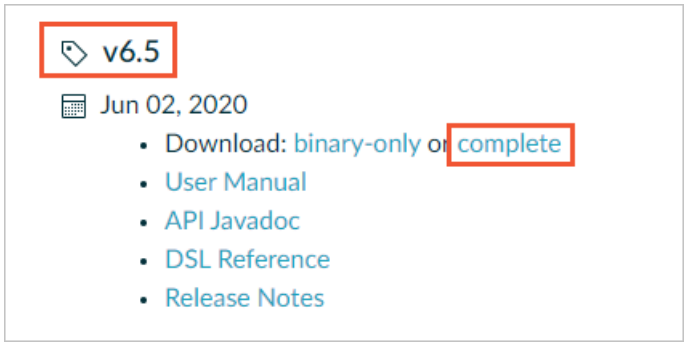
名称	要求
Android系统版本	Android 5.0及以上版本
Android API版本	Lollipop（API 20）及以上版本
CPU架构支持	ARM64、ARMV7
集成工具	Android Studio

非硬性要求（开发此Demo的环境，仅供编码的人员参考）

名称	要求
Android Studio版本	4.1.1
JRE	1.8.0_152-release-1024-b01 x86_64
JVM	OpenJDK 64-Bit
compileSdkVersion	29
buildToolsVersion	29.0.0
minSdkVersion	20
targetSdkVersion	29
gradle version	gradle-6.5-all
gradle plugin version	com.android.tools.build:gradle:4.1.1

编译Demo

1. 下载gradle v6.5版本并解压，详情请参见Gradle build tool。



2. 导入工程。

   打开**Android Studio**，单击**Open an Existing Project**并选择Demo目录下的*alilivesdk_demo*工程。

3. 编译运行。

   i. 编译成功后，单击**运行**，安装Demo到Android终端上。

   ii. 在Android终端上运行Demo，在首页进入房间模块，修改房间号、用户ID、昵称等。

   iii. 进入**主播**和**观众**页面体验连麦功能。

> **说明** 该Demo仅支持真实终端上运行，暂不支持虚拟机。

3.1.2.5. API参考及配置列表

通过阅读本文，您可以了解Android推流SDK的API及配置列表。

API

按照功能划分，API可以分为基础接口、推流基础接口、RTC拉流基础接口等，各功能对应的API如下所示：

基础接口

API	描述	所属类及说明
getSdkVersion	获取版本号。	AliLiveEngine：直播推流核心类，实现RTMP和RTC两种模式的推流功能、RTC模式下拉流功能、视频渲染、美颜开关等，是直播推流一体化SDK最重要的类。
create	创建AliLiveEngine实例。	
destroy	销毁推流。	
setRtsCallback	设置RTC推流相关回调。	
setStatsCallback	设置推流数据回调。	
setStatusCallback	设置推流状态相关回调。	
setNetworkCallback	设置推流网络状态相关回调。	
setVidePreProcessDelegate	设置视频前处理回调。	

API	描述	所属类及说明
setLogDirPath	设置SDK日志文件保存路径。如需调用，请在调用所有API之前先调用此接口，避免日志出现丢失，同时保证指定的目录已存在且可写入。	
setLogLevel	设置日志输出级别。	

推流基础接口

API	描述	所属类及说明
createRenderView	创建渲染的view。	AliLiveEngine：直播推流核心类，实现RTMP和RTC两种模式的推流功能、RTC模式下拉流功能、视频渲染、美颜开关等，是直播推流一体化SDK最重要的类。
startPreview	开始预览（主播端接口）。	
stopPreview	停止预览（主播端接口）。	
pausePush	暂停摄像头采集并进入垫片推流状态（仅支持RTMP模式推流）。需要先调用startPush后才可以调用pausePush，否则调用顺序会出错。	
resumePush	恢复摄像头采集并结束垫片推流状态（仅支持RTMP模式推流）。需要先调用pausePush后才可以调用resumePush，否则调用顺序会出错。	
startPush	开始推流。	
stopPush	停止推流。	
isPublishing	查询是否正在推流。	
getPublishUrl	获取当前推流的地址。	

RTC拉流基础接口

API	描述	所属类及说明
subscribeStream	订阅RTC方式连麦的成员视频流。	AliLiveEngine：直播推流核心类，实现RTMP和RTC两种模式的推流功能、RTC模式下拉流功能、视频渲染、美颜开关等，是直播推流一体化SDK最重要的类。
unSubscribeStream	取消订阅RTC方式的连麦成员视频流。	
renderRemoteStreamWithView	为连麦成员视频流设置渲染窗口。	
renderRemoteStreamWithView	为连麦成员视频流设置渲染窗口以及绘制参数。	

视频相关接口

API	描述	所属类及说明
setPreviewMode	设置预览模式。	AliLiveEngine：直播推流核心类，实现RTMP和RTC两种模式的推流功能、RTC模式下拉流功能、视频渲染、美颜开关等，是直播推流一体化SDK最重要的类。
switchCamera	切换前后摄像头。	
getCurrentCameraPosition	获取当前相机位置。	
isCameraOn	检查相机是否打开。	
setCameraZoom	设置摄像头缩放及是否允许闪光灯。	
isCameraFocusPointSupported	摄像头是否支持手动聚焦。	
isCameraExposurePointSupported	摄像头是否支持设置曝光区域。	
setCameraFocusPoint	设置摄像头手动聚焦。	
setCameraExposurePoint	设置摄像头曝光点。	

音频相关接口

API	描述	所属类及说明
setMute	设置本地音频采集是否为静音帧。	AliLiveEngine：直播推流核心类，实现RTMP和RTC两种模式的推流功能、RTC模式下拉流功能、视频渲染、美颜开关等，是直播推流一体化SDK最重要的类。
isAudioOnly	查询是否纯音频推流。	
enableSpeakerphone	设置音频输出为听筒或扬声器。	
isEnableSpeakerphone	获取当前音频输出为听筒或扬声器。	
setRecordingVolume	设置录音音量。	
setPlayoutVolume	设置播放音量。	
enableEarBack	启用耳返。建议在插入耳机后开启耳返，否则可能会引入回声。	
setEarBackVolume	设置耳返音量。	
playBGM	播放背景音乐。	
stopBGM	停止播放背景音乐。	
pauseBGM	暂停播放背景音乐。	
resumeBGM	恢复播放背景音乐。	
setBGMVolume	设置背景音乐音量。	
getBGMDuration	获取背景音乐文件的总时长。单位：ms。	

API	描述	所属类及说明
getBGMCurrentPosition	获取背景音乐播放进度。单位：ms。	
setBGMPosition	设置音频文件的播放位置。	
setPicthValue	设置声音音调高低。	
setReverbMode	设置混响模式。	
setVoiceChangerMode	设置变声模式。	

美颜相关接口

API	描述	所属类及说明
getBeautyManager	获取美颜接口。	AliLiveEngine：直播推流核心类，实现RTMP和RTC两种模式的推流功能、RTC模式下拉流功能、视频渲染、美颜开关等，是直播推流一体化SDK最重要的类。
enable	美颜开关。	AliLiveBeautyManager：美颜功能控制类，即美颜类型，包含美白、磨皮。
getEnableType	获取当前的美颜开关类型。	
setBeautyParam	设置某个美颜的参数。	
destroy	销毁美颜。	

回调接口

API	描述	所属类及说明
onLiveTotalStats	累计数据回调（仅支持RTC流）。	StatsCallback：推流过程实时数据回调。指标：当推RTC流时，2秒回调一次；推RTMP流时，10秒回调一次。
onLiveLocalVideoStats	本地视频统计信息（支持RTMP和RTC流）。	
onLiveRemoteVideoStats	远端视频统计信息（仅支持RTC流）。	
onLiveRemoteAudioStats	远端音频统计信息（仅支持RTC流）。	
onSubscribeResult	订阅某一个URL后的状态回调。当调用subscribeStream后，推流引擎会回调此URL的订阅状态，在连麦场景下，一个UID唯一标示一个粉丝的流信息，订阅表示接受该粉丝的音视频流。	

API	描述	所属类及说明
onUnSubscribeResult	取消订阅某一个URL后的状态回调。当调用unSubscribeStream后，推流引擎会回调此URL的取消订阅状态，在连麦场景下，一个UID唯一标示一个粉丝的流信息，取消订阅表示不再接受该粉丝的音视频流。	RtsCallback：此回调只适用于RTC推流，表示连麦方式下订阅粉丝流的状态回调，RTMP推流没有该回调信息。
onFirstPacketReceivedWithUid	成功订阅某个URL后开始接收其媒体数据，当接收到第一个包后触发此回调，一般在此回调后开始调用渲染renderRemoteStreamWithView来展示视频流。	
onFirstRemoteVideoFrameDrawn	订阅的第一帧视频帧显示时触发此消息。	
onLiveSdkError	如果AliLiveEngine出现Error，通过此回调通知App。	StatusCallback：回调推流状态的相关回调，RTMP和RTC推流都使用这个回调获取推流状态。
onLiveSdkWarning	如果AliLiveEngine出现Warning，通过此回调通知App。	
onPreviewStarted	开始预览回调。	
onPreviewStopped	停止预览回调。	
onFirstVideoFramePreviewed	渲染第一帧视频回调。	
onLivePushStarted	推流成功回调，表示开始推流。	
onLivePushStopped	推流停止回调，表示停止推流。	
onAudioFocusChanged	音频焦点变化的结果通知。	
onBGMStateChanged	音频音效播放状态回调。	
onTexture	在OpenGL线程中回调，进行采集图像的二次处理，如美颜。	AliLiveVideoPreProcessCallback：视频前处理回调，可以用来自定义视频图像处理，如自定义美颜等。
onTextureDestroy	Texture销毁前在OpenGL线程中回调。	
onVideoData	采集过程中实时进行视频数据回调。	
onNetworkPoor	弱网回调，当网络变成弱网状态时会触发此回调。	
onNetworkStatusChange	网络状态变化回调，例如WiFi切换成4G，或网络中断等会触发此回调。	
onConnectRecovery	网络状态恢复回调。	

API	描述	所属类及说明
onReconnectStart	推流引擎开始重连回调。当网络非常不好的情况下可能会出现推流中断的情况，推流中断后会尝试开始重连，此时触发此回调。	AliLivePushNetworkCallback：推流过程中网络状态变化相关回调，通知接入方当前推流引擎的网络状态。
onConnectionLost	推流中断回调，推流引擎中断推流。	
onReconnectSuccess	推流引擎重连成功回调，继续开始推流。	
onReconnectFail	重连失败回调。	

配置列表

- AliLiveConfig：阿里云直播SDK配置RTMP和RTC推流初始化参数配置模块，创建直播引擎时需要将配置对象通过参数的形式传入，配置只能在推流开始前设置，推流开始后重新设置配置无效。

参数	类型	默认值	说明
accountId	String	NULL	连麦域名HTTPDNS解析账号ID（连麦场景下必须设置）。
rtmpConfig	AliLiveRTMPConfig	NULL	RTMP推流参数属性。
cameraPosition	AliLiveCameraPosition	AliLiveCameraPositionFront	相机位置。
audioOnly	boolean	false	纯音频连麦。
autoFocus	boolean	false	是否支持自动对焦。
videoPushProfile	AliLiveVideoPushProfile	AliLiveVideoProfile_540P	视频推流分辨率。
videoFPS	int	20	视频帧率。
enableHighDefPreview	boolean	false	是否开启高清预览。如果采集分辨率低于720P时，本地预览以720P的方式呈现，采集分辨率高于720P以实际分辨率预览。
enableVideoEncoderHWAcceleration	boolean	true	视频硬编码。
enableVideoDecoderHWAcceleration	boolean	true	视频硬解码。
videoGopSize	AliLivePushVideoEncodeGOP	AliLivePushVideoEncodeGOP_2	视频编码GOP大小，单位：s。

参数	类型	默认值	说明
pauseImage	Bitmap	NULL	可以设置推流一张图片，此时观众只能看到推的这张图片，常用于App切换至后台。设置后调用pausePush推送图片。
videoRenderMode	AliLiveRenderMode	AliLiveRenderModeAuto	视频显示模式。
pushMirror	AliLiveRenderMirrorMode	AliLiveRenderMirrorModeOnlyFront	推流镜像。
flash	boolean	false	开启闪光灯。
zoom	float	1.0	画面缩放。
extra	String	NULL	辅助字段，用户可以传入直播成员唯一标示，用于SDK问题的日志排查。

- AliLiveRTMPConfig：阿里云直播SDK配置RTMP推流初始化参数配置模块，创建直播引擎时需要将配置对象通过参数的形式传入，配置只能在推流开始前设置，推流开始后重新配置无效。

参数	类型	默认值	说明
enableAudioHWAcceleration	boolean	false	音频软编码。
videoInitBitrate	int	1000	视频编码初始编码码率，单位：kbps。
videoTargetBitrate	int	1500	视频编码目标编码码率，单位：kbps。
videoMinBitrate	int	600	视频编码最小编码码率，单位：kbps。
audioChannel	AliLivePushAudioChannel	AliLivePushAudioChannel_1	音频声道数。
audioSampleRate	AliLivePushAudioSampleRate	AliLivePushAudioSampleRate44100	音频采样率。
audioEncoderProfile	AliLiveAudioEncoderProfile	AliLiveAudioEncoderProfile_AAC_LC	音频编码格式。
autoReconnectRetryCount	int	5	推流自动重连次数。
autoReconnectRetryInterval	int	1000	推流自动重连间隔，单位：ms。

基础接口

- getSdkVersion：获取版本号。

```
static String getSdkVersion()
```

返回说明

返回SDK版本号。

- create：创建AliLiveEngine实例。

```
static AliLiveEngine create(Context context, AliLiveConfig config)
```

参数说明

参数	类型	说明
context	Context	上下文。
config	AliLiveConfig	直播配置。

返回说明

返回AliLiveEngine实例。

- destroy：销毁推流。

```
void destroy()
```

- setRtsCallback：设置RTC推流相关回调。

```
void setRtsCallback(AliLiveCallback.RtsCallback rtsCallback)
```

参数说明

参数	类型	说明
rtsCallback	RtsCallback	RTC推流回调。

- setStatsCallback：设置推流数据回调。

```
void setStatsCallback(AliLiveCallback.StatsCallback statsCallback)
```

参数说明

参数	类型	说明
statsCallback	statsCallback	推流数据回调。

- setStatusCallback：设置推流状态相关回调。

```
void setStatusCallback(AliLiveCallback.StatusCallback statusCallback)
```

参数说明

参数	类型	说明
statusCallback	StatusCallback	推流状态回调。

- setNetworkCallback：设置推流网络状态相关回调。

```
void setNetworkCallback(AliLiveCallback.NetworkCallback networkCallback)
```

参数说明

参数	类型	说明
networkCallback	NetworkCallback	推流网络状态回调。

- setVidePreProcessDelegate：设置视频前处理回调。

```
void setVidePreProcessDelegate(AliLiveCallback.AliLiveVideoPreProcessCallback callback)
```

参数说明

参数	类型	说明
callback	AliLiveVideoPreProcessCallback	视频前处理回调。

- setLogDirPath：设置SDK日志文件保存路径。如需调用，请在调用所有API之前先调用此接口，避免日志出现丢失，同时保证指定的目录已存在且可写入。

```
int setLogDirPath(String logDirPath)
```

参数说明

参数	类型	说明
logDirPath	String	SDK日志文件保存路径。

- setLogLevel：设置日志输出级别。

```
void setLogLevel(AliLiveConstants.AliLiveLogLevel level)
```

参数说明

参数	类型	说明
level	AliLiveLogLevel	日志级别。

推流基础接口

- createRenderView：创建渲染的view。

```
AliLiveRenderView createRenderView(boolean overlay)
```

参数说明

参数	类型	说明
overlay	boolean	是否覆盖其他surfaceView。

返回说明

返回渲染的view。

- startPreview：开始预览（主播端接口）。

```
AliLiveError startPreview(AliLiveRenderView surfaceView)
```

参数说明

参数	类型	说明
surfaceView	AliLiveRenderView	渲染的view。

返回说明

返回NULL表示调用成功，否则表示调用失败。

- stopPreview：停止预览（主播端接口）。

```
AliLiveError stopPreview()
```

返回说明

返回NULL表示调用成功，否则表示调用失败。

- pausePush：暂停摄像头采集并进入垫片推流状态。需要先调用startPush后才可以调用pausePush，否则调用顺序会出错。

```
void pausePush()
```

- resumePush：恢复摄像头采集并结束垫片推流状态。需要先调用pausePush后才可以调用resumePush，否则调用顺序会出错。

```
void resumePush()
```

- startPush：开始推流。

```
AliLiveError startPush(String url)
```

参数说明

参数	类型	说明
url	String	推流地址。可以通过地址头（rtmp://或artc://）区分是RTMP或RTC推流。

返回说明

返回NULL表示调用成功，否则表示调用失败。

- stopPush：停止推流。

```
AliLiveError stopPush()
```

返回说明

返回NULL表示调用成功，否则表示调用失败。

- isPublishing：查询是否正在推流。

```
boolean isPublishing()
```

返回说明

返回true表示正在推流中，false表示没有在推流。

- getPublishUrl：获取当前推流的地址。

```
String getPublishUrl()
```

返回说明

返回当前推流的地址。

RTC拉流基础接口

- subscribeStream：订阅RTC方式连麦的成员视频流。

```
AliLiveError subscribeStream(String url)
```

参数说明

参数	类型	说明
url	String	需要被订阅的连麦成员的唯一URL标示。

返回说明

返回NULL表示调用成功，否则表示调用失败。

- unSubscribeStream：取消订阅RTC方式的连麦成员视频流。

```
AliLiveError unSubscribeStream(String url)
```

参数说明

参数	类型	说明
url	String	需要取消被订阅的连麦成员的唯一URL标示。

返回说明

返回NULL表示调用成功，否则表示调用失败。

- renderRemoteStreamWithView：为连麦成员视频流设置渲染窗口。

```
int renderRemoteStreamWithView(AliLiveRenderView remoteView, String url)
```

参数说明

参数	类型	说明
remoteView	AliLiveRenderView	需要渲染视频流的窗口。
url	String	连麦成员的唯一URL标示。

返回说明

返回0表示调用成功，-1表示调用失败。

- renderRemoteStreamWithView：为连麦成员视频流设置渲染窗口以及绘制参数。

```
int renderRemoteStreamWithView(AliLiveRenderView remoteView, String url, AliLiveConstants
.AliLiveRenderMode renderMode,
                                                    AliLiveConstants.AliLiveRenderMirrorMo
de mirrorMode)
```

参数说明

参数	类型	说明
remoteView	AliLiveRenderView	需要渲染视频流的窗口。
url	String	连麦成员的唯一URL标示。
renderMode	AliLiveRenderMode	视频渲染方式。
mirrorMode	AliLiveRenderMirrorMode	是否需要镜像。

返回说明

返回0表示调用成功，-1表示调用失败。

视频相关接口

- setPreviewMode：设置预览模式。

```
AliLiveError setPreviewMode(AliLiveConstants.AliLiveRenderMode renderMode, AliLiveConstan
ts.AliLiveRenderMirrorMode mirrorMode)
```

参数说明

参数	类型	说明
renderMode	AliLiveRenderMode	视频渲染方式。
mirrorMode	AliLiveRenderMirrorMode	是否需要镜像。

返回说明

返回NULL表示调用成功，否则表示调用失败。

- switchCamera：切换前后摄像头。

```
AliLiveError switchCamera()
```

返回说明

返回NULL表示调用成功，否则表示调用失败。

- getCurrentCameraPosition：获取当前相机位置。

```
AliLiveConstants.AliLiveCameraPosition getCurrentCameraPosition()
```

返回说明

返回当前相机位置。

- isCameraOn：检查相机是否打开。

```
boolean isCameraOn()
```

返回说明

返回true表示摄像头已打开，false表示摄像头没有打开。

- setCameraZoom：设置摄像头缩放及是否允许闪光灯。

```
int setCameraZoom(float zoom, boolean flashOn)
```

参数说明

参数	类型	说明
zoom	float	缩放的级别。
flashOn	boolean	是否允许闪光灯。

返回说明

返回0表示调用成功，否则表示调用失败。

- isCameraFocusPointSupported：摄像头是否支持手动聚焦。

```
boolean isCameraFocusPointSupported()
```

返回说明

返回true表示支持，false表示不支持。

- isCameraExposurePointSupported：摄像头是否支持设置曝光区域。

```
boolean isCameraExposurePointSupported()
```

返回说明

返回true表示支持，false表示不支持。

- setCameraFocusPoint：设置摄像头手动聚焦。

```
int setCameraFocusPoint(AliLivePoint point)
```

参数说明

参数	类型	说明
point	AliLivePoint	聚焦点。

返回说明

返回0表示调用成功，否则表示调用失败。

- setCameraExposurePoint：设置摄像头曝光点。

```
int setCameraExposurePoint(AliLivePoint point)
```

参数说明

参数	类型	说明
point	AliLivePoint	曝光点。

返回说明

返回0表示调用成功，否则表示调用失败。

音频相关接口

- setMute：设置本地音频采集是否为静音帧。

```
int setMute(boolean mute)
```

参数说明

参数	类型	说明
mute	boolean	是否为静音帧。取值： ○ true：表示本地音频采集静音帧。 ○ false：表示恢复成正常帧。

返回说明

返回0表示成功放入队列，-1表示被拒绝。

- isAudioOnly：查询是否纯音频推流。

```
boolean isAudioOnly()
```

返回说明

返回true表示纯音频推流，false表示非纯音频推流。

- enableSpeakerphone：设置音频输出为听筒或扬声器。

```
int enableSpeakerphone(boolean enable)
```

参数说明

参数	类型	说明
enable	boolean	音频输出为听筒或扬声器。取值： ○ true：表示扬声器模式。 ○ false（默认值）：表示听筒模式。

返回说明

返回0表示调用成功，否则表示调用失败。

- isEnableSpeakerphone：获取当前音频输出为听筒或扬声器。

```
boolean isEnableSpeakerphone()
```

返回说明

返回true表示扬声器模式，false表示听筒模式。

- setRecordingVolume：设置录音音量。

```
int setRecordingVolume(int volume)
```

参数说明

参数	类型	说明
volume	int	录音音量。默认值为100。取值范围： ○ 0：静音。 ○ [0,100）：减小音量到某值。 ○ (100,400]：增大音量到某值。

返回说明

返回0表示调用成功，否则表示调用失败。

- setPlayoutVolume：设置播放音量。

```
int setPlayoutVolume(int volume)
```

参数说明

参数	类型	说明
volume	int	播放音量。默认值为100。取值范围： ○ 0：静音。 ○ [0,100）：减小音量到某值。 ○ (100,400]：增大音量到某值。

返回说明

返回0表示调用成功，否则表示调用失败。

- enableEarBack：启用耳返。建议在插入耳机后开启耳返，否则可能会引入回声。

```
int enableEarBack(boolean enable)
```

参数说明

参数	类型	说明
enable	boolean	是否启用耳返。

- setEarBackVolume：设置耳返音量。

```
int setEarBackVolume(int volume)
```

参数说明

参数	类型	说明
volume	int	设置耳返音量。

- playBGM：播放背景音乐。

```
int playBGM(String path, boolean publish, boolean loop)
```

参数说明

参数	类型	说明
path	String	本地音乐文件路径（仅支持mp3、wav格式音频，且不支持中文路径）。
publish	boolean	是否推流。取值： ○ true：推流到远端。 ○ false：仅本地试听，不推流到远端。
loop	boolean	是否循环播放。取值： ○ true：循环。 ○ false：不循环。

- stopBGM：停止播放背景音乐。

```
int stopBGM()
```

- pauseBGM：暂停播放背景音乐。

```
int pauseBGM()
```

- resumeBGM：恢复播放背景音乐。

```
int resumeBGM()
```

- setBGMVolume：设置背景音乐音量。

```
int setBGMVolume( int volume)
```

参数说明

参数	类型	说明

参数	类型	说明
volume	int	背景音乐音量。取值： ○ 0：静音。 ○ (0,100]：设置音量为某值。

- getBGMDuration：获取背景音乐文件的总时长。单位：毫秒。

```
int getBGMDuration()
```

- getBGMCurrentPosition：获取背景音乐播放进度。单位：毫秒。

```
int getBGMCurrentPosition()
```

- setBGMPosition：设置音频文件的播放位置。

```
int setBGMPosition(int posMs)
```

参数说明

参数	类型	说明
posMs	int	进度条位置。单位：毫秒。

- setPicthValue：设置声音音调高低。

```
int setPicthValue(float pitch)
```

参数说明

参数	类型	说明
pitch	float	设置音调，默认值为1.0f，范围为[0.5, 2.0]。

- setReverbMode：设置混响模式。

```
int setReverbMode(AliLiveConstants.AliLiveReverbMode mode)
```

参数说明

参数	类型	说明

参数	类型	说明
mode	AliLiveReverbMode	混响模式。取值： ○ AliLiveReverbMode_Off：关闭。 ○ AliLiveReverbMode_Vocal_I：人声 I。 ○ AliLiveReverbMode_Vocal_II：人声 II。 ○ AliLiveReverbMode_Bathroom：澡堂。 ○ AliLiveReverbMode_Small_Room_Bright：明亮小房间。 ○ AliLiveReverbMode_Small_Room_Dark：黑暗小房间。 ○ AliLiveReverbMode_Medium_Room：中等房间。 ○ AliLiveReverbMode_Large_Room：大房间。 ○ AliLiveReverbMode_Church_Hall：教堂走廊。 ○ AliLiveReverbMode_Cathedral：大教堂。

- setVoiceChangerMode：设置变声模式。

```
int setVoiceChangerMode(AliLiveConstants.AliLiveVoiceChangerMode mode)
```

参数说明

参数	类型	说明
mode	AliLiveVoiceChangerMode	变声模式。取值： ○ AliLiveVoiceChanger_OFF：关闭。 ○ AliLiveVoiceChanger_Oldman：老人。 ○ AliLiveVoiceChanger_Babyboy：男孩。 ○ AliLiveVoiceChanger_Babygirl：女孩。 ○ AliLiveVoiceChanger_Robot：机器人。 ○ AliLiveVoiceChanger_Daimo：大魔王。 ○ AliLiveVoiceChanger_Ktv：KTV。 ○ AliLiveVoiceChanger_Echo：回声。

美颜相关接口

- getBeautyManager：获取美颜接口。

```
AliLiveBeautyManager getBeautyManager()
```

返回说明

返回美颜接口。

- enable：开关美颜。

```
void enable(EnableType enableType)
```

参数说明

参数	类型	说明
enableType	EnableType	美颜功能的开关。取值： ○ Off（默认值）：关闭美颜。 ○ Basic：基础美颜。

- getEnableType：获取当前的美颜开关类型。

```
EnableType getEnableType()
```

返回说明

返回美颜开关。

- setBeautyParam：设置某个美颜的参数。

```
void setBeautyParam(BeautyParam param, float value)
```

参数说明

参数	类型	说明
param	BeautyParam	美颜参数。取值： ○ SkinBuffing_SkinBuffing：磨皮。 ○ SkinWhiting_SkinWhiting：美白。
value	float	美颜参数值。取值：0~1。

- destroy：销毁美颜。

```
void destroy()
```

回调接口

- onLiveTotalStats：累计数据回调（仅支持RTC流）。

```
void onLiveTotalStats(AliLiveStats stats)
```

参数说明

参数	类型	说明
stats	AliLiveStats	实时数据。

实时数据包含以下内容：

参数	类型	说明
sentKbitrate	long	总发送码率，单位：kbps。（暂未支持）
rcvdKbitrate	long	总接收码率，单位：kbps。（暂未支持）
sentBytes	long	总发送数据量，单位：Byte。（暂未支持）
rcvdBytes	long	总接收数据，单位：Byte。（暂未支持）
videoRcvdKbitrate	long	视频发送码率，单位：kbps。（暂未支持）
videoSentKbitrate	long	视频接收码率，单位：kbps。（暂未支持）
callDuration	long	通话时长，单位：s。（暂未支持）
cpuUsage	float	CPU使用量，单位：%。（暂未支持）

- onLiveLocalVideoStats：本地视频统计信息（支持RTMP和RTC流）。

```
void onLiveLocalVideoStats(AliLiveLocalVideoStats localVideoStats)
```

参数说明

参数	类型	说明
localVideoStats	AliLiveLocalVideoStats	本地视频统计信息。

本地视频统计信息包含以下内容：

参数	类型	说明
track	AliLiveVideoTrack	视频流track类型。（暂未支持）
sentBitrate	int	发送码率。
sentFps	int	发送帧率。
encodeFps	int	编码帧率。

- onLiveRemoteVideoStats：远端视频统计信息（仅支持RTC流）。

```
void onLiveRemoteVideoStats(AliLiveRemoteVideoStats remoteVideoStats)
```

参数说明

参数	类型	说明
remoteVideoStats	AliLiveRemoteVideoStats	远端视频统计信息。

远端视频统计信息包含以下内容：

参数	类型	说明
userId	String	远端用户UserID。
track	AliLiveVideoTrack	视频流track类型。
width	int	宽度。
height	int	高度。
decodeFps	int	解码帧率。
renderFps	int	渲染帧率。
frozenTimes	int	卡顿次数。

- onLiveRemoteAudioStats：远端音频统计信息（仅支持RTC流）。

```
void onLiveRemoteAudioStats(AliLiveRemoteAudioStats remoteAudioStats)
```

参数说明

参数	类型	说明
remoteAudioStats	AliLiveRemoteAudioStats	远端音频统计信息。

远端音频统计信息包含以下内容：

参数	类型	说明
userId	String	远端用户UserID。
track	AliLiveAudioTrack	音频流track类型。
audioLossRate	int	回调周期内的音频丢帧率。
rcvdBitrate	int	接收流的瞬时码率，单位：kbps。
totalFrozenTimes	int	远端用户加入频道后音频卡顿的累计时长，单位：ms。

- onSubscribeResult：订阅某一个URL后的状态回调。当调用subscribeStream后，推流引擎会回调此URL的

订阅状态，在连麦场景下，一个UID唯一标示一个粉丝的流信息，订阅表示接受该粉丝的音视频流。

```
void onSubscribeResult(AliLiveResult result, String uid)
```

参数说明

参数	类型	说明
result	AliLiveResult	订阅状态，成功或者失败。
uid	String	订阅成员的唯一标识。

- onUnSubscribeResult：取消订阅某一个URL后的状态回调。当调用unSubscribeStream后，推流引擎会回调此URL的取消订阅状态，在连麦场景下，一个UID唯一标示一个粉丝的流信息，取消订阅表示不再接受该粉丝的音视频流。

```
void onUnSubscribeResult(AliLiveResult result, String uid)
```

参数说明

参数	类型	说明
result	AliLiveResult	订阅状态，成功或者失败。
uid	String	订阅成员的唯一标识。

- onFirstPacketReceivedWithUid：成功订阅某个URL后开始接收其媒体数据，当接收到第一个包后触发此回调，一般在此回调后开始调用渲染renderRemoteStreamWithView来展示视频流。

```
void onFirstPacketReceivedWithUid(String url)
```

参数说明

参数	类型	说明
url	String	订阅成员的唯一标识。

- onFirstRemoteVideoFrameDrawn：订阅的第一帧视频帧显示时触发此消息。

```
void onFirstRemoteVideoFrameDrawn(String uid, AliLiveConstants.AliLiveVideoTrack videoTra
ck)
```

参数说明

参数	类型	说明
uid	String	订阅成员的唯一标识。
videoTrack	AliLiveVideoTrack	流类型。

- onLiveSdkError：如果AliLiveEngine出现Error，通过此回调通知App。

```
void onLiveSdkError(AliLiveEngine publisher, AliLiveError error)
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。
error	AliLiveError	错误信息。

- onLiveSdkWarning：如果AliLiveEngine出现Warning，通过此回调通知App。

```
void onLiveSdkWarning(AliLiveEngine publisher, int warn)
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。
warn	int	警告码。

- onPreviewStarted：开始预览回调。

```
void onPreviewStarted(AliLiveEngine publisher)
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。

- onPreviewStopped：停止预览回调。

```
void onPreviewStopped(AliLiveEngine publisher)
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。

- onFirstVideoFramePreviewed：渲染第一帧视频回调。

```
void onFirstVideoFramePreviewed(AliLiveEngine publisher)
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。

- onLivePushStarted：推流成功回调，表示开始推流。

```
void onLivePushStarted(AliLiveEngine publisher)
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。

- onLivePushStopped：推流停止回调，表示停止推流。

```
void onLivePushStopped(AliLiveEngine publisher)
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。

- onAudioFocusChanged：音频焦点变化的结果通知。

```
void onAudioFocusChanged(int focusChange)
```

参数说明

参数	类型	说明
focusChange	int	请参见Android源码android.media.AudioManager中的定义。

- onBGMStateChanged：音频音效播放状态回调。

```
void onBGMStateChanged(AliLiveEngine publisher, AliLiveConstants.AliLiveAudioPlayingStateCode playState,
                                    AliLiveConstants.AliLiveAudioPlayingErrorCode errorCode)
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。
playState	AliLiveAudioPlayingStateCode	播放状态。
errorCode	AliLiveAudioPlayingErrorCode	播放状态对应的错误码。

- onTexture：在OpenGL线程中回调，进行采集图像的二次处理，如美颜。

```
int onTexture(int textureId, int width, int height, int stride, int rotate)
```

参数说明

参数	类型	说明
textureId	int	纹理ID。

参数	类型	说明
width	int	纹理宽度。
height	int	纹理高度。
stride	int	对齐偏移。
rotate	int	纹理角度。

- onTextureDestroy：Texture销毁前在OpenGL线程中回调。

```
int onTextureDestroy()
```

- onVideoData：采集过程中实时进行视频数据回调。

```
void onVideoData(long dataFrameY, long dataFrameU, long dataFrameV, AliLiveConstants.AliLiveImageFormat format, int width, int height, int strideY, int strideU, int strideV, int rotate)
```

参数说明

参数	类型	说明
dataFrameY	long	Y分量。
dataFrameU	long	U分量。
dataFrameV	long	V分量。
format	AliLiveImageFormat	图像格式。
width	int	图像宽。
height	int	图像高。
strideY	int	Y偏移。
strideU	int	U偏移。
strideV	int	V偏移。
rotate	int	角度。

- onNetworkPoor：弱网回调，当网络变成弱网状态时会触发此回调。

```
void onNetworkPoor()
```

- onNetworkStatusChange：网络状态变化回调，例如WiFi切换成4G，或网络中断等会触发此回调。

```
void  void onNetworkStatusChange(AliLiveConstants.AliLiveNetworkStatus status)()
```

参数说明

参数	类型	说明
status	AliLiveNetworkStatus	网络连接状态。取值： ○ 4G。 ○ WiFi。 ○ 无网络。

- onConnectRecovery：网络状态恢复回调。

```
void onConnectRecovery()
```

- onReconnectStart：推流引擎开始重连回调。当网络非常不好的情况下可能会出现推流中断的情况，推流中断后会尝试开始重连，此时触发此回调。

```
void onReconnectStart()
```

- onConnectionLost：推流中断回调，推流引擎中断推流。

```
void onConnectionLost()
```

- onReconnectSuccess：推流引擎重连成功回调，继续开始推流。

```
void onReconnectSuccess()
```

- onReconnectFail：重连失败回调。

```
void onReconnectFail()
```

3.1.3. iOS推流SDK

3.1.3.1. 基础概念

通过阅读本文，您可以了解iOS推流SDK中一些专业术语的基本概念。

- 码率控制：一种编码的优化算法，用于控制视频流码流的大小。同样的视频编码格式，码流越大，包含的信息越多，对应的图像也就越清晰，反之亦然。
- 视频丢帧：发送视频帧时，如果网络非常差，导致视频帧堆积严重，可以通过丢弃视频帧来缩短推流的延时。
- 耳返：指主播可以通过耳机实时听到自己的声音。例如，当主播带上耳机唱歌时，需要把握音调，这时就需要开启耳返功能。因为声音通过网络传入耳朵和通过空气传入耳朵差异很大，而主播需要直接听到观众端的效果。
- 混音：把多种来源的声音整合至一个立体音轨或单音音轨中。iOS推流SDK支持音乐和人声的混音。
- 动态库：即动态链接库，与常用的静态库相反。动态库在编译时并不会被拷贝到目标程序中，目标程序中只会存储指向动态库的引用。在程序运行时，动态库才会被真正加载进来。

> **说明** Xcode加载动态库需要加载到Embedded Binaries中，而不是加载到Linked Frameworks and Libraries中。

3.1.3.2. 集成说明

通过阅读本文，您可以了解iOS推流SDK的集成及配置方法。

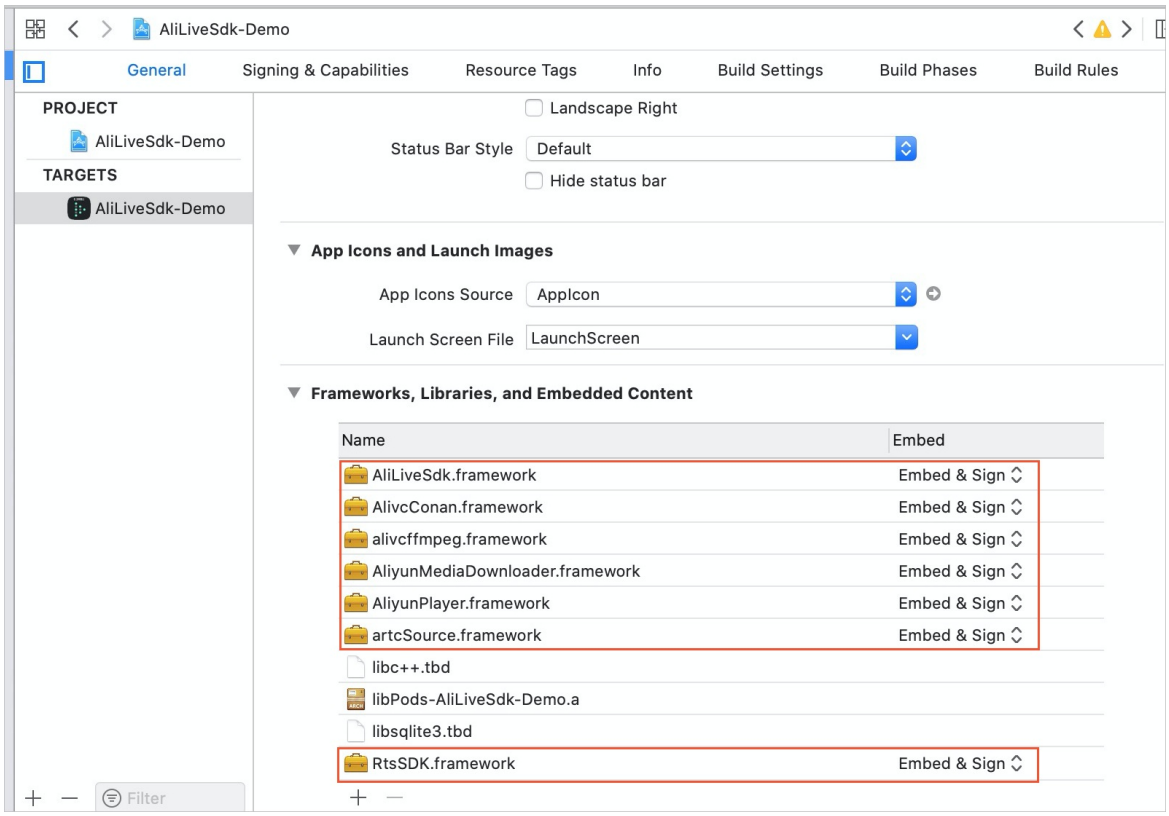
SDK集成

手工集成

1. 下载并解压iOS端推流SDK。

   > 说明 此版本推流SDK已停止使用，推荐使用新版推流SDK。

2. 新建工程并导入framework，然后将导入的framework的Embed属性设置成Embed & Sign。



3. Podfile添加第三方库依赖。

```
pod 'AFNetworking'
pod 'SocketRocket'
pod 'SVProgressHUD'
pod 'Masonry'
```

4. 安装第三方库依赖。

   pod install

Pod集成

1. Podfile添加播放器及第三方库依赖。

```
#播放器
pod 'AliLiveSDK_iOS', "4.0.2"
pod 'AliPlayerSDK_iOS', "5.2.3"
pod 'AliPlayerSDK_iOS_ARTC', "5.2.3"
pod 'RtsSDK','1.5.0'
#第三方库
pod 'AFNetworking'
pod 'SocketRocket'
pod 'SVProgressHUD'
pod 'Masonry'
```

2. 安装iOS推流SDK。

   pod install

配置工程

1. 配置Pod。

   Podfile中增加如下代码。

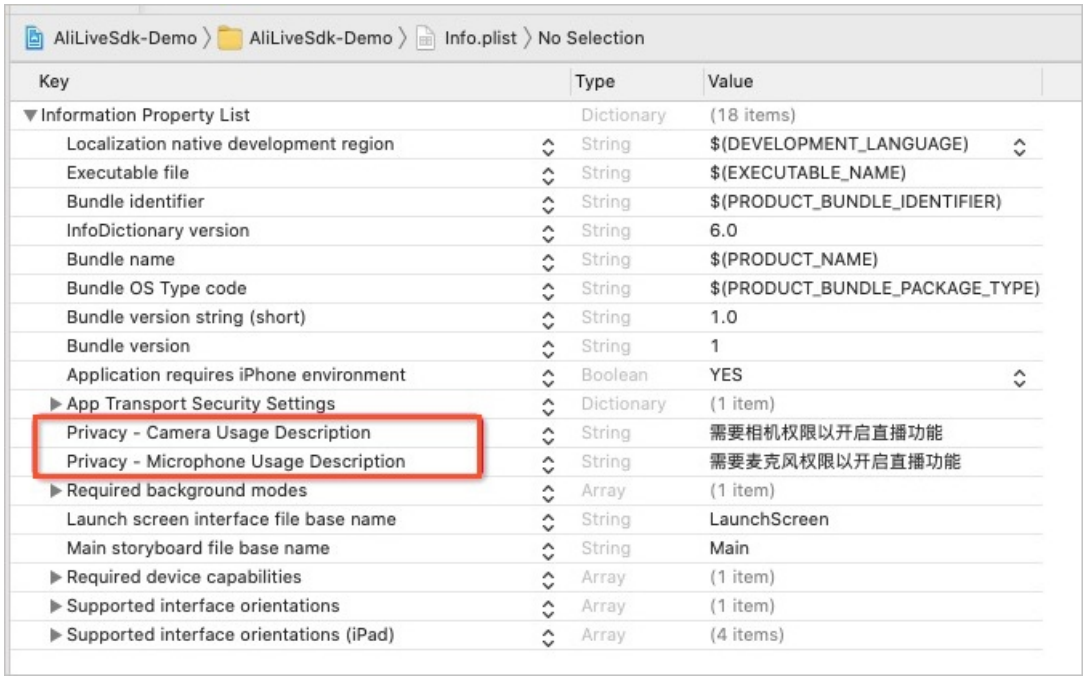
```
pod 'RtsSDK','1.5.0'
pod 'AliPlayerSDK_iOS','5.2.3'
pod 'AliPlayerSDK_iOS_ARTC','5.2.3'
```

2. 配置工程。
   i. 打开之前创建好的工程。
   ii. 在主菜单中选择**Build Setting > Build Options**。
   iii. 设置**Enable**为**No**。
   iv. 配置info.Plist文件。

   配置info.Plist有两种方式，如下所示：

- 添加如下字段：

```
Privacy - Camera Usage Description
Privacy - Microphone Usage Description
```



- 把info.Plist以SourceCode的形式打开，添加如下代码：

```
<key>NSMicrophoneUsageDescription</key>
<string></string>
<key>NSCameraUsageDescription</key>
<string></string>
<key>NSAppTransportSecurity</key>
<dict>
    <key>NSAllowsArbitraryLoads</key>
    <true/>
</dict>
```

3.1.3.3. 使用说明

通过阅读本文，您可以了解iOS推流SDK的使用方法。

RTMP推流

> **说明** 阿里云视频直播不允许同一时间向同一个推流URL进行多路推流（第二路推流会被拒绝）。

1. 导入推流头文件。

```
#import <AliLiveSdk/AliLiveSdk.h>
```

2. 创建AliLiveEngine。

```
AliLiveConfig *config = [[AliLiveConfig alloc] init];
config.videoProfile = AliLiveVideoProfile_540P;
config.videoFPS = 20;
myConfig.pauseImage = [UIImage imageNamed:@"background_img.png"];
myConfig.accountID = @"";
AliLiveEngine *engine = [[AliLiveEngine alloc] initWithConfig:myConfig];
[engine setAudioSessionOperationRestriction:AliLiveAudioSessionOperationRestrictionDeac
tivateSession];
[engine setRtsDelegate:self];
[engine setStatusDelegate:self];
```

3. 开始预览。

```
[self.engine startPreview:self.renderView];
```

4. 开始推流。

```
[self.engine startPushWithURL:self.pushUrl];
```

5. 结束推流。

```
[self.engine stopPush];
[self.engine stopPreview];
[self.engine destorySdk];
self.engine = nil;
```

RTMP拉流

1. 导入头文件。

```
#import <AliyunPlayer/AliyunPlayer.h>
```

2. 创建播放器。

```
self.player = [[AliPlayer alloc] init];
self.player.autoPlay = YES;
self.player.delegate = self;
self.player.playerView = self.renderView;
```

3. 开始拉流。

```
AVPUrlSource *source = [[AVPUrlSource alloc] urlWithString:self.playurl];
[self.player setUrlSource:source];
[self.player prepare];
```

4. 停止拉流。

```
[self.player stop];
[self.player destroy];
```

主播PK

1. 主播A发送请求至appserver，获取目前在线且符合要求的主播列表。

```
[self.roomClient sendCmd:@"GetOnlineRoomList"];
```

获取到主播列表后渲染到页面。

```
- (void)onReceiveCommandGetOnlineRoomList:(NSDictionary *)msg {
    NSArray *roomArray = [OnlineRoom roomArrayWithOnlineRoomArray:msg[@"onlineRoomList"
]];
    if (self.delegate && [self.delegate respondsToSelector:@selector(anchor:didReceiveO
nlineRoomArray:)]) {
        [self.delegate anchor:self didReceiveOnlineRoomArray:roomArray];
    }
}
```

2. 主播A在列表中找到一个主播B，请求PK。

```
[self.roomClient sendCmd:@"ApplyPk" param:@{@"toUserId":@(toUserId),@"toRoomId":@(toRoo
mId)}];
```

3. 主播B收到主播A的申请，并弹出提示。

```
- (void)onReceiveCommandApplyPkNotice:(NSDictionary *)msg {
    // 这里拿到申请方信息，并弹出提示
}
```

4. 主播B同意了主播A的申请后，开始拉RTC流，并发送同意PK信令。

```
[self.engine subscribeStream:self.pkOnlineRoom.rtcPullUrl preferMaster:YES];
```

```
[self.roomClient sendCmd:@"ApprovePk" param:@{@"fromUserId":@(fromUserId), @"fromRoomId
":@(fromRoomId), @"approve":@(approve)}];
```

5. 主播A收到主播B的同意申请后，开始拉RTC流。

```
[self.engine subscribeStream:self.pkOnlineRoom.rtcPullUrl preferMaster:YES];
```

6. 观众端播放的RTMP地址有变化，收到NotifyPublish信令后，重新构建播放器播放新地址。

```
self.rtmpUrl = anchorRtmpPullUrl;
[self playRtmp];
```

7. 主播B停止PK，发送停止PK信令，停止拉RTC流，并恢复PK前画面。

```
[self.roomClient sendCmd:@"CancelPk"];
```

```
[self.engine unSubscribeStream:self.pkOnlineRoom.rtcPullUrl];
self.pkOnlineRoom = nil;
self.pkRenderView = nil;
```

8. 主播A收到停止PK信令，结束拉流，并恢复PK前画面。

```
- (void)onReceiveCommandCancelPkNotice:(NSDictionary *)msg {
    if (self.delegate && [self.delegate respondsToSelector:@selector(anchorDidReceiveCa
ncelPk:)]) {
        [self.delegate anchorDidReceiveCancelPk:self];
    }
}
```

9. 观众端播放的RTMP地址有变化，收到NotifyPublish信令后，重新构建播放器播放新地址。

```
self.rtmpUrl = anchorRtmpPullUrl;
[self playRtmp];
```

3.1.3.4. Demo编译

通过阅读本文，您可以了解iOS端Demo的运行环境要求及编译方法。

环境要求

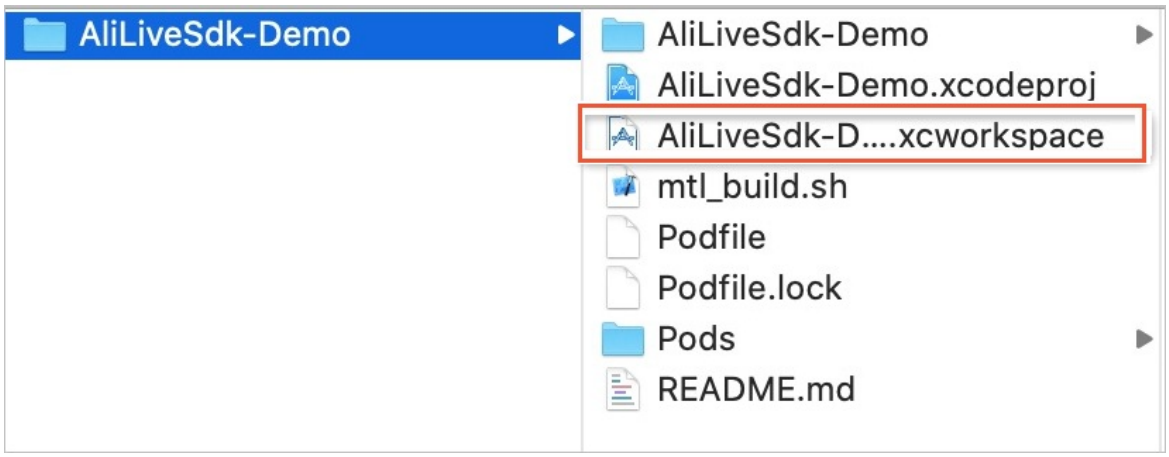
名称	要求
Xcode版本	Xcode 11.7及以上版本
CPU架构支持	ARM64、ARMV7、ARMV7s
运行系统	iOS 9.0及以上版本

编译Demo

1. 登录终端，进入到Podfile目录，引入iOS端SDK。

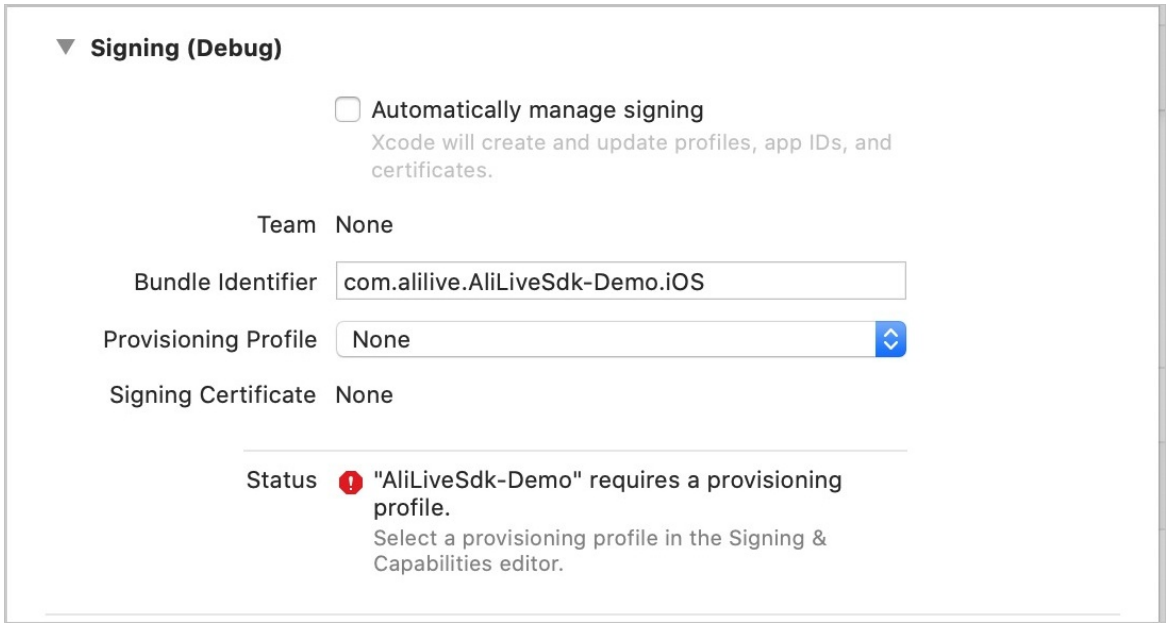
   ```
   pod install
   ```

2. 双击**AliLiveSdk-Demo.xcworkspace**，打开Demo工程。

3. 配置调试环境。

   如果环境中没有对应的profile文件，系统会报错，添加对应的profile文件或以开发者账号登录进行配置即可解决。



4. 按Command+R键运行Demo。

> 说明 该Demo仅支持真实终端上运行，暂不支持虚拟机。

3.1.3.5. API参考及配置列表

通过阅读本文，您可以了解iOS推流SDK的API及配置列表。

API

按照功能划分，API可以分为基础接口、音视频基础接口、推拉流基础接口等，各功能对应的API如下所示：

基础接口

API	描述	所属类及说明
getSdkVersion	获取SDK的当前版本号。	AliLiveEngine：直播推流核心类，实现RTMP和RTC两种模式的推流功能、RTC模式下拉流功能、视频渲染、美颜开关等，是直播推流一体化
initWithConfig	设置AliLiveConfig推流初始化配置项。	
destorySdk	销毁AliLiveEngine 对象，释放相关资源。	
setStatusDelegate	设置直播推流状态回调，推流状态回调请参见AliLivePushInfoStatusDelegate。	
setNetworkDelegate	设置直播网络状态回调，推流网络状态回调请参见AliLiveNetworkDelegate。	

API	描述	SDK最重要的类。 所属类及说明
setRtsDelegate	RTC推流模式下订阅某个粉丝媒体相关回调，详情请参见AliLiveRtsDelegate。	
setVidePreProcessDelegate	设置视频前处理回调。	
setDataStatsDelegate	设置直播媒体参数回调。	
setLogDirPath	设置SDK日志文件保存路径。如需调用，请在调用所有API之前先调用此接口，避免日志出现丢失，同时保证指定的目录已存在且可写入。	
setLogLevel	设置日志输出级别。	

本地预览接口

API	描述	所属类及说明
startPreview:renderMode:mirrorMode	开启摄像头本地预览，可以指定本地渲染模式和镜像模式。开启本地预览不会立刻开始推送媒体流信息到CDN上，需要调用startPushWithURL才真正开始推流。本地预览成功会触发onPreviewStarted回调。	AliLiveEngine：直播推流核心类，实现RTMP和RTC两种模式的推流功能、RTC模式下拉流功能、视频渲染、美颜开关等，是直播推流一体化SDK最重要的类。
startPreview	开启摄像头本地预览。renderMode默认为AliLiveRenderModeAuto，mirroeMode默认为AliLiveRenderMirrorModeOnlyFrontCameraPreviewEnabled。开启本地预览不会立刻开始推送媒体流信息到CDN上，需要调用startPushWithURL才真正开始推流。本地预览成功会触发onPreviewStarted回调。	
stopPreview	停止摄像头本地预览，结束本地预览成功会触发onPreviewStoped回调。	

推流基础接口

API	描述	所属类及说明
startPushWithURL	开始推流。推流成功后会触发onLivePushStarted回调。	
stopPush	停止推流。停止推流成功后会触发onLivePushStoped回调。	

API	描述	所属类及说明
pausePush	暂停摄像头采集并进入垫片推流状态（仅支持RTMP模式推流）。需要先调用startPushWithURL后才可以调用pausePush，否则调用顺序会出错。	AliLiveEngine：直播推流核心类，实现RTMP和RTC两种模式的推流功能、RTC模式下拉流功能、视频渲染、美颜开关等，是直播推流一体化SDK最重要的类。
resumePush	恢复摄像头采集并结束垫片推流状态（仅支持RTMP模式推流）。需要先调用pausePush后才可以调用resumePush，否则调用顺序会出错。	
isPublishing	查询是否正在推流。	
livePushURL	获取当前推流的地址。	

拉流基础接口

API	描述	所属类及说明
subscribeStream	订阅RTC方式连麦的成员视频流。	AliLiveEngine：直播推流核心类，实现RTMP和RTC两种模式的推流功能、RTC模式下拉流功能、视频渲染、美颜开关等，是直播推流一体化SDK最重要的类。
unSubscribeStream	取消订阅RTC方式的连麦成员视频流。	
renderRemoteStreamWithView:url:renderMode:mirrorMode	为连麦成员视频流渲染窗口。当该接口调用后，会触发onFirstRemoteVideoFrameDrawn回调，表示开始渲染连麦成员视频流。	
renderRemoteStreamWithView:url	为连麦成员视频流渲染窗口。renderMode默认为AliLiveRenderModeAuto，mirrorMode默认为AliLiveRenderMirrorModeAllDisabled。当该接口调用后，会触发onFirstRemoteVideoFrameDrawn回调，表示开始渲染连麦成员视频流。	

美颜相关接口

API	描述	所属类及说明
getRaceBeautyManager	获取美颜管理类AliLiveRaceBeautyManager的对象。	AliLiveEngine：直播推流核心类，实现RTMP和RTC两种模式的推流功能、RTC模式下拉流功能、视频渲染、美颜开关等，是直播推流一体化SDK最重要的类。

API	描述	所属类及说明
enableBeautyEffect	美颜开关。	AliLiveRaceBeautyManager：美颜功能控制类，即美颜类型，包含美白、磨皮。
setRaceBeautyParams:value	设置某个美颜的参数。	

音视频相关接口

API	描述	所属类及说明
isCameraOn	查询摄像头是否打开。	AliLiveEngine：直播推流核心类，实现RTMP和RTC两种模式的推流功能、RTC模式下拉流功能、视频渲染、美颜开关等，是直播推流一体化SDK最重要的类。
switchCamera	切换前后摄像头。	
setDeviceOrientationMode	设置设备横竖屏方向，可以设置横竖屏推流模式。当接入层没有调用该方法时，默认开启自适应模式，即横屏方向推横屏流，竖屏方向推竖屏流。	
setCameraZoom:flash	设置摄像头缩放倍数及是否开启闪光灯。	
isCameraFocusPointSupported	查询摄像头是否支持手动对焦。	
setCameraFocusPoint	设置摄像头手动聚焦点。	
isCameraExposurePointSupported	查询摄像头是否支持设置曝光区域。	
setCameraExposurePoint	设置摄像头曝光点。	
isAudioOnly	查询是否是纯音频推流。	
setMute	设置本地音频采集是否为静音帧。	
enableSpeakerphone	设置音频输出为听筒或扬声器。	
isEnableSpeakerphone	查询当前音频输出为听筒或扬声器。	
setPlayoutVolume	设置播放音量。	
setRecordingVolume	设置录音音量。	
setAudioSessionOperationRestriction	设置SDK对AVAudioSession的控制权限。	
enableEarBack	启用耳返。建议在插入耳机后开启耳返，否则可能会引入回声。	
setEarBackVolume	设置耳返音量。	
playBGM:publish:loop	播放背景音乐。	
stopBGM	停止播放背景音乐。	

API	描述	所属类及说明
pauseBGM	暂停播放背景音乐。	
resumeBGM	恢复播放背景音乐。	
setBGMVolume	设置背景音乐音量	
getBGMDuration	获取背景音乐文件的总时长。单位：ms。	
getBGMCurrentPosition	获取背景音乐播放进度。单位：ms。	
setBGMPosition	设置背景音乐的播放位置。	
setPicthValue	调整声音音调高低。	
setReverbMode	设置混响模式。	
setVoiceChangerMode	设置变声模式。	

回调接口

API	描述	所属类及说明
onSubscribe:result:url	订阅某一个URL后的状态回调。当调用subscribeStream后，推流引擎会回调此URL的订阅状态，在连麦场景下，一个URL唯一标示一个粉丝的流信息，订阅表示接受该粉丝的音视频流。	AliLiveRtsDelegate：此回调只适用于RTC推流，表示连麦方式下订阅粉丝流的状态回调，RTMP推流没有该回调信息。
onUnSubscribe:result:url	取消订阅某一个URL后的状态回调。当调用unSubscribeStream后，推流引擎会回调此URL的取消订阅状态，在连麦场景下，一个UID唯一标示一个粉丝的流信息，取消订阅表示不再接受该粉丝的音视频流。	
onFirstPacketReceived:url	成功订阅某个URL后开始接收其媒体数据，当接收到第一个包后触发此回调，一般在此回调后开始调用渲染renderRemoteStreamWithView:url来展示视频流。	
onFirstRemoteVideoFrameDrawn	订阅的第一帧视频帧显示时触发此消息。	
onLiveTotalStats:stats	RTC推流中实时数据回调（2秒回调一次）。	AliLiveDataStatsDelegate：统计直播推流媒体相关信息的回调。
onLiveLocalVideoStats:stats	RTC推流中发送本地视频的统计信息（2秒回调一次）。	

API	描述	所属类及说明
onLiveRemoteVideoStats:stats	RTC推流中接收远端视频的统计信息（2秒回调一次）。	
onNetworkStatusChange	网络状态变化回调，例如WiFi切换成4G，或网络中断等会触发此回调。	AliLiveNetworkDelegate：推流过程中网络状态变化相关回调，通知接入方当前推流引擎的网络状态。
onNetworkPoor	弱网回调，当网络变成弱网状态时会触发此回调。	
onNetworkRecovery	网络状态恢复回调。	
onReconnectStart	推流引擎开始重连回调。当网络非常不好的情况下可能会出现推流中断的情况，推流中断后会尝试开始重连，此时触发此回调。	
onReconnectSuccess	推流引擎重连成功回调，继续开始推流。	
onConnectionLost	推流中断回调，推流引擎中断推流。	
onTexture:width:height:rotate	在OpenGL线程中回调，进行采集图像的二次处理，如美颜。	AliLiveVidePreProcessDelegate：视频前处理回调，可以用来自定义视频图像处理，如自定义美颜等。
onTextureDestoryed	在OpenGL线程中回调，可以在这里释放创建的OpenGL资源。	
onVideoPixelBuffer	视频采集对象回调，进行采集图像的二次处理。	
onPreviewStarted	开始预览回调。	AliLivePushInfoStatusDelegate：回调推流状态的相关回调，RTMP和RTC推流都使用这个回调获取推流状态。
onPreviewStoped	停止预览回调。	
onFirstVideoFramePreviewed	渲染第一帧视频回调。	
onLivePushStarted	推流成功回调，表示开始推流。	
onLivePushStoped	推流停止回调，表示停止推流。	
onLiveSdkError	如果AliLiveEngine出现Error，通过此回调通知App。	
onBGMStateChanged	音频音效播放状态回调。	

配置列表

AliLiveConfig：阿里云直播SDK配置RTMP和RTC推流初始化参数配置模块，创建直播引擎时需要将配置对象通过参数的形式传入，配置只能在推流开始前设置，推流开始后重新设置配置无效。

参数	类型	默认值	说明
cameraPosition	AliLiveCameraPosition	AliLiveCameraPositionFront	相机位置。
beautyOn	BOOL	true	是否开启美颜。
enablePureAudioPush	BOOL	false	是否开启纯音频推流。
autoFocus	BOOL	false	是否支持自动对焦。
videoProfile	int	AliLiveVideoProfile_540P	视频推流分辨率。
videoFPS	int	20	视频帧率。
enableHighDefPreview	BOOL	false	是否开启高清预览。如果采集分辨率低于720P时，本地预览以720P的方式呈现，采集分辨率高于720P以实际分辨率预览。
enableVideoHWAcceleration	BOOL	true	视频硬编码。
enableVideoDecodeHWAcceleration	BOOL	true	视频硬解码。
enableAudioHWAcceleration	BOOL	false	音频软编码。
videoGopSize	AliLivePushVideoEncodeGOP	AliLivePushVideoEncodeGOP_2	视频编码GOP大小，单位：s。
pauseImage	UIImage	NULL	RTMP推流模式下暂停推流，推送默认图片，目前支持png格式。
videoInitBitrate	int	1000	视频编码初始编码码率，单位：Kbps。
videoTargetBitrate	int	1500	视频编码目标编码码率，单位：Kbps。
videoMinBitrate	int	600	视频编码最小编码码率，单位：Kbps。
audioChannel	AliLivePushAudioChannel	AliLivePushAudioChannel_1	音频声道数。
audioSampleRate	AliLivePushAudioSampleRate	AliLivePushAudioSampleRate44100	音频采样率。
audioEncoderProfile	AliLiveAudioEncoderProfile	AliLiveAudioEncoderProfile_AAC_LC	音频编码格式。

参数	类型	默认值	说明
autoReconnectRetryCount	int	5	推流自动重连次数。
autoReconnectRetryInterval	int	1000	推流自动重连间隔，单位：ms。
accountID	NSString	NULL	连麦域名HTTPDNS解析账号ID（连麦场景下必须设置）。
extra	NSString	NULL	辅助字段，用户可以传入直播成员唯一标示，用于SDK问题的日志排查。

基础接口

- getSdkVersion：获取SDK的当前版本号。

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

返回说明

返回SDK版本号。

- initWithConfig：设置AliLiveConfig推流初始化配置项。

```
- (instancetype)initWithConfig:(AliLiveConfig*)config;
```

参数说明

参数	类型	说明
config	AliLiveConfig	直播推流配置。

返回说明

返回AliLiveEngine实例。

- destorySdk：销毁AliLiveEngine 对象，释放相关资源。

```
- (void)destorySdk;
```

- setStatusDelegate：设置直播推流状态回调，推流状态回调请参见AliLivePushInfoStatusDelegate。

```
- (void) setStatusDelegate:(id<AliLivePushInfoStatusDelegate>)delegate;
```

参数说明

参数	类型	说明
delegate	AliLivePushInfoStatusDelegate	推流状态回调。

- setNetworkDelegate：设置直播网络状态回调，推流网络状态回调请参见AliLiveNetworkDelegate。

```
- (void) setNetworkDelegate:(id<AliLiveNetworkDelegate>)delegate;
```

参数说明

参数	类型	说明
delegate	AliLiveNetworkDelegate	推流网络状态回调。

- setRtsDelegate：RTC推流模式下订阅某个粉丝媒体相关回调，详情请参见AliLiveRtsDelegate。

```
- (void) setRtsDelegate:(id<AliLiveRtsDelegate>)delegate;
```

参数说明

参数	类型	说明
delegate	AliLiveRtsDelegate	RTC回调。

- setVidePreProcessDelegate：设置视频前处理回调。

```
- (void)setVidePreProcessDelegate:(id<AliLiveVidePreProcessDelegate>)delegate;
```

参数说明

参数	类型	说明
delegate	AliLiveVidePreProcessDelegate	视频前处理回调。

- setDataStatsDelegate：设置直播媒体参数回调。

```
- (void) setDataStatsDelegate:(id<AliLiveDataStatsDelegate>)delegate;
```

参数说明

参数	类型	说明
delegate	AliLiveDataStatsDelegate	直播媒体参数回调。

- setLogDirPath：设置SDK日志文件保存路径。如需调用，请在调用所有API之前先调用此接口，避免日志出现丢失，同时保证指定的目录已存在且可写入。

```
- (int)setLogDirPath:(NSString *)logDirPath;
```

参数说明

参数	类型	说明
logDirPath	String	SDK日志文件保存路径。

- setLogLevel：设置日志输出级别。

```
- (void)setLogLevel:(AliLiveLogLevel) level;
```

参数说明

参数	类型	说明
level	AliLiveLogLevel	日志级别。

本地预览接口

- startPreview:renderMode:mirrorMode：开启摄像头本地预览，可以指定本地渲染模式和镜像模式。开启本地预览不会立刻开始推送媒体流信息到CDN上，需要调用startPushWithURL才真正开始推流。本地预览成功会触发onPreviewStarted回调。

```
- (int)startPreview:(AliLiveRenderView *)previewView renderMode:(AliLiveRenderMode)rMode
mirrorMode:(AliLiveRenderMirrorMode)mMode;
```

参数说明

参数	类型	说明
previewView	AliLiveRenderView	预览view。
rMode	AliLiveRenderMode	视频渲染方式。
mMode	AliLiveRenderMirrorMode	是否需要镜像。

- startPreview：开启摄像头本地预览。renderMode默认为AliLiveRenderModeAuto，mirroeMode默认为AliLiveRenderMirrorModeOnlyFrontCameraPreviewEnabled。开启本地预览不会立刻开始推送媒体流信息到CDN上，需要调用startPushWithURL才真正开始推流。本地预览成功会触发onPreviewStarted回调。

```
/**
 * @brief 启动摄像头预览
 * @param previewView 预览view
 * @return 0:success  非0:failure
*/
- (int)startPreview:(AliLiveRenderView *)previewView;
```

参数说明

参数	类型	说明
previewView	AliLiveRenderView	预览view。

- stopPreview：停止摄像头本地预览，结束本地预览成功会触发onPreviewStoped回调。

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

推流基础接口

- startPushWithURL：开始推流。推流成功后会触发onLivePushStarted回调。

```
- (void)startPushWithURL:(NSString *)pushURL;
```

参数说明

参数	类型	说明
pushURL	NSString	推流地址。可以通过地址头（rtmp://或artc://）区分是RTMP或RTC推流。

- stopPush：停止推流。停止推流成功后会触发onLivePushStoped回调。

```
- (void)stopPush;
```

- pausePush：暂停摄像头采集并进入垫片推流状态。需要先调用startPushWithURL后才可以调用pausePush，否则调用顺序会出错。

```
- (void)pausePush;
```

- resumePush：恢复摄像头采集并结束垫片推流状态。需要先调用pausePush后才可以调用resumePush，否则调用顺序会出错。

```
- (void)resumePush;
```

- isPublishing：查询是否正在推流。

```
- (BOOL)isPublishing;
```

- livePushURL：获取当前推流的地址。

```
@property (nonatomic, readonly) NSString *livePushURL;
```

拉流基础接口

- subscribeStream：订阅RTC方式连麦的成员视频流。

```
- (void)subscribeStream:(NSString *)url;
```

参数说明

参数	类型	说明
url	NSString	需要被订阅的连麦成员的唯一URL标示。

- unSubscribeStream：取消订阅RTC方式的连麦成员视频流。

```
/**
 * @brief 取消订阅RTC 方式的连麦成员媒体流
 * @param url 需要取消订阅的连麦成员的唯一url标示
 */
- (void)unSubscribeStream:(NSString *)url;
```

参数说明

参数	类型	说明
url	NSString	需要取消订阅的连麦成员的唯一URL标示。

- renderRemoteStreamWithView:url:renderMode:mirrorMode：为连麦成员视频流渲染窗口。当该接口调用后，会触发onFirstRemoteVideoFrameDrawn回调，表示开始渲染连麦成员视频流。

```
- (void)renderRemoteStreamWithView:(AliLiveRenderView *)renderView url:(NSString *)url re
nderMode:(AliLiveRenderMode)rMode mirrorMode:(AliLiveRenderMirrorMode)mMode;
```

参数说明

参数	类型	说明
renderView	AliLiveRenderView	待渲染的view。
url	NSString	连麦成员的唯一标示。
rMode	AliLiveRenderMode	视频渲染方式。
mMode	AliLiveRenderMirrorMode	是否需要镜像。

- renderRemoteStreamWithView:url：为连麦成员视频流渲染窗口。renderMode默认为AliLiveRenderModeAuto，mirrorMode默认为AliLiveRenderMirrorModeAllDisabled。当该接口调用后，会触发onFirstRemoteVideoFrameDrawn回调，表示开始渲染连麦成员视频流。

```
- (void)renderRemoteStreamWithView:(AliLiveRenderView *)renderView url:(NSString *)url;
```

参数说明

参数	类型	说明
renderView	AliLiveRenderView	待渲染的view。
url	NSString	连麦成员的唯一标示。

美颜相关接口

- getRaceBeautyManager：获取美颜管理类AliLiveRaceBeautyManager的对象。

```
/**
 * @brief 设置美颜
 * @return 获取美颜管理对象
*/
- (AliLiveRaceBeautyManager *)getRaceBeautyManager;
```

返回说明

返回美颜管理对象。

- enableBeautyEffect：美颜开关。

```
- (void)enableBeautyEffect:(BOOL)isOpen;
```

参数说明

参数	类型	说明
isOpen	BOOL	打开或关闭美颜。

- setRaceBeautyParams:value：设置某个美颜的参数。

```
- (void)setRaceBeautyParams:(AliLiveRaceBeautyParams) param value:(float) value;
```

参数说明

参数	类型	说明
param	AliLiveRaceBeautyParams	美颜参数。取值： ○ kAliLiveRaceBeautyParamsSkinBuffing：磨皮。 ○ kAliLiveRaceBeautyParamsWhitening：美白。
value	float	美颜参数值。取值：0~1。

音视频相关接口

- isCameraOn：查询摄像头是否打开。

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

返回说明

返回true表示摄像头已打开，false表示摄像头没有打开。

- switchCamera：切换前后摄像头。

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

- setDeviceOrientationMode：设置设备横竖屏方向，可以设置横竖屏推流模式。当接入层没有调用该方法时，默认开启自适应模式，即横屏方向推横屏流，竖屏方向推竖屏流。

```
- (int)setDeviceOrientationMode:(AliLiveOrientationMode)mode;
```

参数说明

参数	类型	说明
mode	AliLiveOrientationMode	设备方向。

- setCameraZoom:flash：设置摄像头缩放倍数及是否开启闪光灯。

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

参数说明

参数	类型	说明
zoom	float	缩放倍数。
flash	BOOL	是否允许闪光灯。

- isCameraFocusPointSupported：查询摄像头是否支持手动对焦。

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

返回说明

返回true表示支持，false表示不支持。

- setCameraFocusPoint：设置摄像头手动聚焦点。

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

参数说明

参数	类型	说明
point	CGPoint	聚焦点。

- isCameraExposurePointSupported：查询摄像头是否支持设置曝光区域。

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

返回说明

返回true表示支持，false表示不支持。

- setCameraExposurePoint：设置摄像头曝光点。

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

参数说明

参数	类型	说明
point	CGPoint	曝光点。

- isAudioOnly：查询是否是纯音频推流。

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

返回说明

返回true表示纯音频推流，false表示非纯音频推流。

- setMute：设置本地音频采集是否为静音帧。

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

参数说明

参数	类型	说明
mute	BOOL	是否为静音帧。取值： ○ true：表示本地音频采集静音帧。 ○ false：表示恢复成正常帧。

- enableSpeakerphone：设置音频输出为听筒或扬声器。

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

参数说明

参数	类型	说明
enable	BOOL	音频输出为听筒或扬声器。取值： ○ true：表示扬声器模式。 ○ false（默认值）：表示听筒模式。

- isEnableSpeakerphone：查询当前音频输出为听筒或扬声器。

```
- (BOOL)isEnableSpeakerphone;
```

返回说明

返回true表示扬声器模式，false表示听筒模式。

- setPlayoutVolume：设置播放音量。

```
- (int)setRecordingVolume:(NSInteger)volume;
```

参数说明

参数	类型	说明
volume	NSInteger	播放音量。默认值为100。取值范围： ○ 0：静音。 ○ [0,100）：减小音量到某值。 ○ (100,400]：增大音量到某值。

- setRecordingVolume：设置录音音量。

```
- (int)setRecordingVolume:(NSInteger)volume;
```

参数说明

参数	类型	说明
volume	NSInteger	录音音量。默认值为100。取值范围： ○ 0：静音。 ○ [0,100）：减小音量到某值。 ○ (100,400]：增大音量到某值。

- setAudioSessionOperationRestriction：设置SDK对AVAudioSession的控制权限。

```
- (int)setAudioSessionOperationRestriction:(AliLiveAudioSessionOperationRestriction)restriction;
```

参数说明

参数	类型	说明
restriction	AliLiveAudioSessionOperationRestriction	设置AVAudioSession对应的权限。

- enableEarBack：启用耳返。建议在插入耳机后开启耳返，否则可能会引入回声。

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

参数说明

参数	类型	说明
enable	BOOL	是否启用耳返。

- setEarBackVolume：设置耳返音量。

```
- (int)setEarBackVolume:(NSInteger)volume;
```

参数说明

参数	类型	说明
volume	NSInteger	设置耳返音量。

- playBGM:publish:loop：播放背景音乐。

```
- (int)playBGM:(NSString *)path publish:(BOOL)publish loop:(BOOL)loop;
```

参数说明

参数	类型	说明
path	NSString	本地音乐文件路径（仅支持mp3、wav格式音频，且不支持中文路径）。
publish	BOOL	是否推流。取值： ○ true：推流到远端。 ○ false：仅本地试听，不推流到远端。
loop	BOOL	是否循环播放。取值： ○ true：循环。 ○ false：不循环。

- stopBGM：停止播放背景音乐。

```
- (int)stopBGM;
```

- pauseBGM：暂停播放背景音乐。

```
- (int)pauseBGM;
```

- resumeBGM：恢复播放背景音乐。

```
- (int)resumeBGM;
```

- setBGMVolume：设置背景音乐音量。

```
- (int)setBGMVolume:(NSInteger)volume;
```

参数说明

参数	类型	说明
volume	NSInteger	背景音乐音量，默认值为50。取值： ○ 0：静音。 ○ (0,100]：设置音量为某值。

- getBGMDuration：获取背景音乐文件的总时长。单位：毫秒。

```
- (int)getBGMDuration;
```

- getBGMCurrentPosition：获取背景音乐播放进度。单位：毫秒。

```
- (int)getBGMCurrentPosition;
```

- setBGMPosition：设置音频文件的播放位置。

```
- (int)setBGMPosition:(NSInteger)pos;
```

参数说明

参数	类型	说明
posMs	NSInteger	进度条位置。单位：毫秒。

- setPicthValue：设置声音音调高低。

```
- (int)setPicthValue:(float)pitch;
```

参数说明

参数	类型	说明
pitch	float	设置音调，默认值为1.0f，范围为[0.5, 2.0]。

- setReverbMode：设置混响模式。

```
- (int)setReverbMode:(AliLiveReverbMode)mode;
```

参数说明

参数	类型	说明
mode	AliLiveReverbMode	混响模式。取值： ○ AliLiveReverb_Off：关闭。 ○ AliLiveReverb_Vocal_I：人声 I。 ○ AliLiveReverb_Vocal_II：人声 II。 ○ AliLiveReverb_Bathroom：澡堂。 ○ AliLiveReverb_Small_Room_Bright：明亮小房间。 ○ AliLiveReverb_Small_Room_Dark：黑暗小房间。 ○ AliLiveReverb_Medium_Room：中等房间。 ○ AliLiveReverb_Large_Room：大房间。 ○ AliLiveReverb_Church_Hall：教堂走廊。 ○ AliLiveReverb_Cathedral：大教堂。

- setVoiceChangerMode：设置变声模式。

```
- (int)setVoiceChangerMode:(AliLiveVoiceChangerMode)mode;
```

参数说明

参数	类型	说明
mode	AliLiveVoiceChangerMode	变声模式。取值： ○ AliLiveVoiceChanger_OFF：关闭。 ○ AliLiveVoiceChanger_OLD_MAN：老人。 ○ AliLiveVoiceChanger_BABYBOY：男孩。 ○ AliLiveVoiceChanger_BABYGILR：女孩。 ○ AliLiveVoiceChanger_ROBOT：机器人。 ○ AliLiveVoiceChanger_DAIMO：大魔王。 ○ AliLiveVoiceChanger_KTV：KTV。 ○ AliLiveVoiceChanger_ECHO：回声。

回调接口

- onSubscribe:result:url：订阅某一个URL后的状态回调。当调用subscribeStream后，推流引擎会回调此URL的订阅状态，在连麦场景下，一个URL唯一标示一个粉丝的流信息，订阅表示接受该粉丝的音视频流。

```
- (void)onSubscribe:(AliLiveEngine *)publisher result:(AliLiveResult *)result url:(NSString *)url;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。
result	AliLiveResult	订阅状态，成功或者失败。
url	NSString	订阅成员的唯一标识。

- onUnSubscribe:result:url：取消订阅某一个URL后的状态回调。当调用unSubscribeStream后，推流引擎会回调此URL的取消订阅状态，在连麦场景下，一个UID唯一标示一个粉丝的流信息，取消订阅表示不再接受该粉丝的音视频流。

```
- (void)onUnSubscribe:(AliLiveEngine *)publisher result:(AliLiveResult *)result url:(NSString *)url;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。
result	AliLiveResult	订阅状态，成功或者失败。
url	NSString	订阅成员的唯一标识。

- onFirstPacketReceived:url：成功订阅某个URL后开始接收其媒体数据，当接收到第一个包后触发此回调，一般在此回调后开始调用渲染renderRemoteStreamWithView:url来展示视频流。

```
- (void)onFirstPacketReceived:(AliLiveEngine *)publisher url:(NSString *)url;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象
url	NSString	订阅成员的唯一标识。

- onFirstRemoteVideoFrameDrawn：订阅的第一帧视频帧显示时触发此消息。

```
- (void)onFirstRemoteVideoFrameDrawn:(AliLiveEngine *)publisher url:(NSString *)url;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。
url	NSString	订阅成员的唯一标识。

- onLiveTotalStats:stats：RTC推流中实时数据回调（2秒回调一次）。

```
- (void)onLiveTotalStats:(AliLiveEngine *)publisher stats:(AliLiveStats *)stats;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。
stats	AliLiveStats	实时数据。

实时数据包含以下内容：

参数	类型	说明
sent_kbitrate	long	总发送码率，单位：Kbps。（暂未支持）
rcvd_kbitrate	long	总接收码率，单位：Kbps。（暂未支持）
sent_bytes	long	总发送数据量，单位：Byte。（暂未支持）
rcvd_bytes	long	总接收数据，单位：Byte。（暂未支持）
video_rcvd_kbitrate	long	视频发送码率，单位：Kbps。（暂未支持）
video_sent_kbitrate	long	视频接收码率，单位：Kbps。（暂未支持）
call_duration	long	通话时长，单位：s。（暂未支持）
cpu_usage	float	CPU使用量，单位：%。（暂未支持）

- onLiveLocalVideoStats:stats：RTC推流中发送本地视频的统计信息（2秒回调一次）。

```
/**
 * @brief 本地视频统计信息(2s触发一次)
 * @param localVideoStats 本地视频统计信息
 * @note SDK每两秒触发一次此统计信息回调
 */
- (void)onLiveLocalVideoStats:(AliLiveEngine *)publisher stats:(AliLiveLocalVideoStats *)localVideoStats;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。
localVideoStats	AliLiveLocalVideoStats	本地视频统计信息。

本地视频统计信息包含以下内容：

参数	类型	说明
track	AliLiveVideoTrack	视频流track类型。（暂未支持）
sent_bitrate	int	发送码率。
sent_fps	int	发送帧率。
encode_fps	int	编码帧率。

- onLiveRemoteVideoStats:stats：RTC推流中接收远端视频的统计信息（2秒回调一次）。

```
/**
 * @brief 远端视频统计信息(2s触发一次)
 * @param remoteVideoStats 远端视频统计信息
 */
- (void)onLiveRemoteVideoStats:(AliLiveEngine *)publisher stats:(AliLiveRemoteVideoStats *)remoteVideoStats;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。
remoteVideoStats	AliLiveRemoteVideoStats	远端视频统计信息。

远端视频统计信息包含以下内容：

参数	类型	说明
userId	NSString	远端用户UserID。
track	AliLiveVideoTrack	视频流track类型。

参数	类型	说明
width	int	宽度。
height	int	高度。
decode_fps	int	解码帧率。
render_fps	int	渲染帧率。
frozen_times	int	卡顿次数。

- onNetworkStatusChange：网络状态变化回调，例如WiFi切换成4G，或网络中断等会触发此回调。

```
- (void)onNetworkStatusChange:(AliLiveEngine *)publisher status:(AliLiveNetworkStatus)netStatus;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。
status	AliLiveNetworkStatus	网络连接状态。取值： ○ 4G。 ○ WiFi。 ○ 无网络。

- onNetworkPoor：弱网回调，当网络变成弱网状态时会触发此回调。

```
- (void)onNetworkPoor:(AliLiveEngine *)publisher;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。

- onNetworkRecovery：网络状态恢复回调。

```
- (void)onNetworkRecovery:(AliLiveEngine *)publisher;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。

- onReconnectStart：推流引擎开始重连回调。当网络非常不好的情况下可能会出现推流中断的情况，推流中断后会尝试开始重连，此时触发此回调。

```
- (void)onReconnectStart:(AliLiveEngine *)publisher;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。

- onReconnectSuccess：推流引擎重连成功回调，继续开始推流。

```
- (void)onReconnectSuccess:(AliLiveEngine *)publisher;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。

- onConnectionLost：推流中断回调，推流引擎中断推流。

```
- (void)onConnectionLost:(AliLiveEngine *)publisher;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。

- onTexture:width:height:rotate：在OpenGL线程中回调，进行采集图像的二次处理，如美颜。

```
- (int)onTexture:(int)texture width:(int)width height:(int)height rotate:(int)rotate;
```

参数说明

参数	类型	说明
texture	int	纹理ID。
width	int	纹理宽度。
height	int	纹理高度。
rotate	int	纹理角度。

- onTextureDestoryed：在OpenGL线程中回调，可以在这里释放创建的OpenGL资源。

```
- (void)onTextureDestoryed;
```

- onVideoPixelBuffer：视频采集对象回调，进行采集图像的二次处理。

```
- (CVPixelBufferRef)onVideoPixelBuffer:(CVPixelBufferRef)pixelBuffer;
```

参数说明

参数	类型	说明
pixelBuffer	CVPixelBufferRef	采集图像。

- onPreviewStarted：开始预览回调。

```
- (void)onPreviewStarted:(AliLiveEngine *)publisher;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。

- onPreviewStoped：停止预览回调。

```
- (void)onPreviewStoped:(AliLiveEngine *)publisher;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。

- onFirstVideoFramePreviewed：渲染第一帧视频回调。

```
- (void)onFirstVideoFramePreviewed:(AliLiveEngine *)publisher;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。

- onLivePushStarted：推流成功回调，表示开始推流。

```
- (void)onLivePushStarted:(AliLiveEngine *)publisher;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。

- onLivePushStoped：推流停止回调，表示停止推流。

```
- (void)onLivePushStoped:(AliLiveEngine *)publisher;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。

- onLiveSdkError：如果AliLiveEngine出现Error，通过此回调通知App。

```
- (void)onLiveSdkError:(AliLiveEngine *)publisher error:(AliLiveError *)error;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。
error	AliLiveError	错误信息。

- onBGMStateChanged：音频音效播放状态回调。

```
- (void)onBGMStateChanged:(AliLiveEngine *)publisher
                playState:(AliLiveAudioPlayingStateCode)playState
                errorCode:(AliLiveAudioPlayingErrorCode)errorCode;
```

参数说明

参数	类型	说明
publisher	AliLiveEngine	推流实例对象。
playState	AliLiveAudioPlayingStateCode	播放状态。取值： ○ AliLiveAudioPlayingStarted：开始播放。 ○ AliLiveAudioPlayingStopped：停止播放。 ○ AliLiveAudioPlayingPaused：播放暂停。 ○ AliLiveAudioPlayingResumed：播放恢复。 ○ AliLiveAudioPlayingEnded：播放完毕。 ○ AliLiveAudioPlayingBuffering：正在缓冲。 ○ AliLiveAudioPlayingBufferingEnd：缓冲结束。 ○ AliLiveAudioPlayingFailed：播放失败。
errorCode	AliLiveAudioPlayingErrorCode	播放状态对应的错误码。

3.2. 播放器SDK

3.2.1. 直播播放器SDK简介

本文主要介绍阿里云直播播放器SDK（AliLivePlayer SDK）具备的功能和核心优势。

产品简介

阿里云直播播放器SDK是一款轻量级的专注于直播场景的客户端开发工具，裁减了复杂的播放功能并以极简的包大小来实现直播的基础播放能力。

> 说明
>
> 视频直播支持直播播放器、阿里云播放器以及第三方播放器工具进行播放。
>
> 直播播放器仅支持Android和iOS端播放，阿里云播放器支持Android、iOS、Web、Windows、Flutter多种端播放。如需使用下载请参见相关文档。

功能说明

阿里云直播播放器SDK主要定位于纯直播场景的轻量级播放器SDK，具体功能如下。

功能点	功能说明
URL播放	支持直播播流URL方式播放。
开始与暂停	支持直播流的开始播放或暂停播放。
协议和解码支持	• 支持RTMP、HTTP-FLV和ARTC三种协议。 • 支持视频H264、音频AAC解码。
直播截图	支持截取播放画面的任意一帧。
SEI解析	支持解析SEI（Supplemental Enhancement Information）消息，可用于直播答题等场景。
自定义音视频数据处理	支持边渲染边输出音频原始数据PCM和视频原始数据YUV，可用于变声和美颜的后处理。
缓存设置	支持自定义缓存区大小。
渲染镜像	支持无镜像、水平镜像和垂直镜像三种镜像模式设置。
渲染角度	支持0°、90°、180°和270°四个视频画面渲染角度设置。
填充模式	支持画面填充和画面裁剪两种填充模式。
音量调节	支持实时调节系统音量。
静音	支持开启和关闭静音功能。

> 注意
>
> • 直播播放器SDK不对播放地址来源进行限制。
> • 直播播放器SDK播放地址只支持RTMP、HTTP-FLV和ARTC三种协议。

核心优势

优势项	描述
极简大小	包大小相比阿里云播放器SDK缩减百分之六十左右。
简单、易集成	Android和iOS提供统一接口和错误码，接近系统API的接口设计保证每位开发者能快速集成。
接口兼容	直播播放器精简版与直播播放器完整版的接口兼容，即升级为完整版SDK后无需更换接口。

使用场景

播放器版本不同，播放能力不同，您可根据自己业务需求和使用场景来选择合适的直播播放器版本。

需求场景	推荐使用	相关文档
场景一： 仅需要基础的直播播放能力和直播协议的播放场景下，并且上方功能说明内容可以满足您的所有业务需求。	直播播放器精简版	• Android端：直播播放器精简版集成 • iOS端：直播播放器精简版集成
场景二： 同时需要直播播放能力、点播播放能力和DRM加密、动态多码率等直播高级播放功能。	直播播放器完整版	• Android端：直播播放器完整版集成 • iOS端：直播播放器完整版集成
场景三： 初期：需要基础的直播播放能力基础上选择下载使用直播播放器精简版。 后期：由于业务需要进行扩充点播播放等能力时需升级为直播播放器完整版。	直播播放器完整版	• Android端：直播播放器完整版集成 • iOS端：直播播放器完整版集成

相关文档

使用下载直播播放器和阿里云播放器，请参见SDK下载。

3.2.2. SDK下载

本文提供播放器SDK的下载地址。

直播播放器

使用下载直播播放器SDK，请参见下表。

平台	下载地址	说明
Android	V1.0.0	直播播放器集成方式分为精简版集成和完整版集成，详细内容，请参见集成Android端直播播放器。

平台	下载地址	说明
iOS	V1.0.0	直播播放器集成方式分为精简版集成和完整版集成，详细内容，请参见集成iOS端直播播放器。

阿里云播放器

使用下载阿里云播放器SDK，请参见下表。

播放端	相关文档	说明
Android播放器SDK	Android播放器	阿里云播放器SDK在V5.4.7.1做了较大升级，在稳定性、播放性能和增值服务方面均有较大提升，并已实现License授权。现阿里云播放器SDK的已有功能依然免费提供。 为了改善对客户的技术支持能力，集成方使用V5.4.7.1及之后版本的播放器SDK前，需要先申请License授权。您可以点击申请授权，填写应用信息后提交申请，我们将在48小时内免费向您颁布License文件和密钥。获取到License证书文件后，您还需要配置License，详细操作请参见Android端配置License或iOS端配置License。
iOS播放器SDK	iOS播放器	
Web播放器SDK	Web播放器	-
Windows播放器SDK	Windows播放器	-
Flutter播放器SDK	Flutter播放器	-

3.2.3. Android端直播播放器SDK

3.2.3.1. 集成Android端直播播放器

本文为您介绍了Android端直播播放器SDK的环境要求以及集成方式。

环境要求

类别	说明
系统版本	支持Android 4.3及以上版本。
手机芯片	架构要求： • armv7 • arm64
开发工具	推荐使用Android Studio，本文操作步骤基于Android Studio开发。下载地址：Android Studio。

版本选择

请根据使用场景选择集成直播播放器精简版或完整版。不同版本的使用场景请参见使用场景。

直播播放器精简版集成

集成SDK时，需要先下载Android端直播播放器SDK包（包含了播放器SDK及Demo源码），下载地址请参见SDK下载。

1. 复制*AliLivePlayer-1.0.0.aar*、*AliyunPlayer-5.4.4.0-simplify.aar*包到工程的*libs*目录下。

   > **说明** 如果没有，手动创建libs文件夹即可。

2. 修改Project下的*build.gradle*文件，在allprojects的repositories节点中增加flatDir的设置。

```
flatDir {
   dirs 'libs'
}
```

3. 修改App的build.gradle文件，在dependencies节点中增加对 `aar` 和 `maven` 的引用。

```
dependencies {
      implementation fileTree(dir: 'libs', include: ['*.aar'])
      //当需要支持ARTC协议，需额外添加如下2个依赖
      implementation 'com.aliyun.sdk.android:AlivcArtc:5.4.4.0'
      implementation 'com.aliyun.rts.android:RtsSDK:2.1.0'
}
```

4. 在App的Proguard-rules.pro文件中添加以下混淆配置。

```
-keep class com.alivc.**{*;}
-keep class com.aliyun.**{*;}
-keep class com.cicada.**{*;}
-dontwarn com.alivc.**
-dontwarn com.aliyun.**
-dontwarn com.cicada.**
```

5. 调用接口实现播放功能，请参见实现Android端播放功能。

直播播放器完整版集成

1. 复制*AliLivePlayer-1.0.0.aar*包到工程的*libs*目录下。

   > **说明** 如果没有，手动创建*libs*文件夹即可。

2. 修改Project下的build.gradle文件，在allprojects的repositories节点中增加flatDir的设置。

```
flatDir {
   dirs 'libs'
}
```

3. 修改App的build.gradle文件，在dependencies节点中增加对 `aar` 和 `maven` 的引用。

```
dependencies {
      implementation fileTree(dir: 'libs', include: ['*.aar'])
      //需要依赖点播播放器
      implementation 'com.aliyun.sdk.android:AliyunPlayer:5.4.4.0-full'
      //当需要支持ARTC协议，需额外添加如下2个依赖
      implementation 'com.aliyun.sdk.android:AlivcArtc:5.4.4.0'
      implementation 'com.aliyun.rts.android:RtsSDK:2.1.0'
}
```

4. 在App的*Proguard-rules.pro*文件中添加以下混淆配置。

```
-keep class com.alivc.**{*;}
-keep class com.aliyun.**{*;}
-keep class com.cicada.**{*;}
-dontwarn com.alivc.**
-dontwarn com.aliyun.**
-dontwarn com.cicada.**
```

5. 调用接口实现播放功能，请参见实现Android端播放功能。

参考文档

- 实现Android端播放功能
- 接口说明

3.2.3.2. 实现Android端播放功能

直播播放器SDK开放了客户端接口供用户实现播放功能，同时可使用直播播放器其它功能。本文提供实现直播播放器功能的示例代码及接口说明。

背景信息

直播播放器提供一系列接口支持基础的播放功能，您参考以下内容完成接口的调用以实现播放功能。

操作步骤

1. 创建直播播放器，设置渲染View。

```
final boolean enableLog = false; //是否开启日志
AliLivePlayer livePlayer = AliLivePlayerFactory.createAliLivePlayer(getApplicationConte
xt(), enableLog);
SurfaceView surfaceView;
livePlayer.setRenderView(surfaceView);
```

2. 设置播放器监听（可选）。

```
AliLivePlayerConfig config = new AliLivePlayerConfig()
config.enableFrameInfo = true; //可选：设置音视频数据回调
config.enableSEI = true;       //可选：设置SEI数据回调
livePlayer.setConfig(config);
//可选：设置AliLivePlayer回调
mLivePlayer.setObserver(new AliLivePlayerObserver() {
    @Override
    public void onError(AliLivePlayer player, ErrorInfo errorInfo) {
          //错误事件
        Log.e(TAG, "LivePlayer onError:" + "0x" + Integer.toHexString(errorInfo.getCode
()) + " " + errorInfo.getExtra() + " " + errorInfo.getMsg());
    }
    @Override
    public void onRenderingStart(AliLivePlayer player) {
          //首帧渲染事件
        Log.e(TAG, "LivePlayer onRenderingStart...");
    }
    @Override
    public void onInfo(AliLivePlayer player, InfoBean info) {
        //其他信息的事件，type包括了：循环播放开始，缓冲位置，当前播放位置，自动播放开始等
        Log.e(TAG, "LivePlayer onInfo:" + "0x" + Integer.toHexString(info.getCode()) +
" " + info.getExtraMsg() + " " + info.getExtraValue());
    }
    @Override
    public void onVideoSizeChanged(AliLivePlayer player, int width, int height) {
        //视频分辨率变化回调事件
        Log.e(TAG, "LivePlayer onVideoSizeChanged:" + width + " " + height);
    }
    @Override
    public void onSnapShot(AliLivePlayer player, Bitmap image, int with, int height) {
        //截图事件
        super.onSnapShot(player, image, with, height);
    }
    @Override
    public boolean onRenderFrame(AliLivePlayer player, FrameInfo frameInfo) {
        Log.e(TAG, "LivePlayer onRenderFrame:" + frameInfo.toString());
        return false;
    }
    @Override
    public void onSeiData(AliLivePlayer player, int type, byte[] data) {
        Log.e(TAG, "LivePlayer onSeiData:" + type + " , content:" + new String(data));
    }
});
```

3. 提供URL进行播放控制。

> **说明**
>
> - 播放地址支持RTMP、HTTP-FLV、ARTC协议的在线音视频播放。
> - FLV协议目前不支持HTTPS-FLV。

```
String url = "rtmp://example.aliyundoc.com";
//开始播放
livePlayer.start(url);
//停止播放
livePlayer.stop();
```

4. 销毁直播播放器。

```
AliLivePlayerFactory.releaseAliLivePlayer(livePlayer);
```

接口说明

如需实现直播播放器提供的其它功能，请参考接口说明。详情请参见接口说明。

3.2.3.3. 接口说明

通过阅读本文，您可以了解阿里云直播播放器的接口详情。

目录

直播播放器接口（类：AliLivePlayer）

接口	描述
setObserver	设置播放器回调。
setRenderView	设置播放器的视频渲染。
setMirrorMode	设置镜像模式。
setRotateMode	设置画面旋转模式。
setScaleMode	设置画面缩放模式。
start	开始播放。
pause	暂停播放。
resume	恢复播放。
stop	停止播放。
setVolume	设置音量（非系统音量）。
getVolume	获取音量。
setMute	设置静音。
isMute	返回是否静音状态。
snapshot	截取当前画面，截图结果通过 `AliLivePlayerObserver.onSnapShot` 回调。
setConfig	设置配置。

接口	描述
getConfig	获取配置。

直播播放器工厂（类：AliLivePlayerFactory）

接口	描述
createAliLivePlayer	创建播放器实例。
releaseAliLivePlayer	销毁播放器实例。

直播播放器回调（类：AliLivePlayerObserver）

接口	描述
onError	出错回调。
onRenderingStart	渲染开始回调。
onInfo	信息回调。
onVideoSizeChanged	分辨率变化回调。
onSnapShot	截屏回调。
onRenderFrame	音视频帧回调。
onSeiData	SEI信息回调。

接口详情

- setObserver：设置播放器回调。

```
public abstract void setObserver(AliLivePlayerObserver playerObserver);
```

参数说明

参数	类型	描述
playerObserver	AliLivePlayerObserver	通过设置回调，可以监听AliLivePlayer播放器回调事件，包括播放器状态、音视频首帧回调、警告和错误信息等。 详情请参见直播播放器SDK包中的*api.txt*文件。

- setRenderView：设置播放器的视频渲染。

```
public abstract void setRenderView(SurfaceView view);
```

参数说明

参数	类型	描述
view	SurfaceView	播放器的视频渲染。 view：该控件负责显示视频内容。

- setMirrorMode：设置镜像模式。

```
public abstract void setMirrorMode(MirrorMode mirrorMode);
```

参数说明

参数	类型	描述
mirrorMode	MirrorMode	镜像模式，取值： ○ MIRROR_MODE_HORIZONTAL：水平镜像。 ○ MIRROR_MODE_NONE：无镜像。 ○ MIRROR_MODE_VERTICAL：垂直镜像。

- setRotateMode：设置画面旋转模式。

```
public abstract void setRotateMode(RotateMode rotateMode);
```

参数说明

参数	类型	描述
rotateMode	RotateMode	画面旋转模式，取值： ○ ROTATE_0：顺时针旋转0度。 ○ ROTATE_180：顺时针旋转180度。 ○ ROTATE_270：顺时针旋转270度。 ○ ROTATE_90：顺时针旋转90度。

- setScaleMode：设置画面缩放模式。

```
public abstract void setScaleMode(ScaleMode scaleMode);
```

参数说明

参数	类型	描述

参数	类型	描述
scaleMode	ScaleMode	画面缩放模式，取值： ○ SCALE_ASPECT_FILL：宽高比填充。 ○ SCALE_ASPECT_FIT：宽高比适应。 ○ SCALE_TO_FILL：拉伸填充。

- start：开始播放。

```
public abstract void start(String url);
```

参数说明

参数	类型	描述
url	String	播放URL地址，支持RTMP、HTTP-FLV、ARTC协议的在线音视频播放。

- pause：暂停播放。

```
public abstract void pause();
```

- resume：恢复播放。

```
public abstract void resume();
```

- stop：停止播放。

```
public abstract void stop();
```

- setVolume：设置音量（非系统音量）。

```
public abstract void setVolume(float value);
```

参数说明

参数	类型	描述
value	float	音量数值（非系统音量）。参数范围[0.0, 1.0]。

- getVolume：获取音量。

```
public abstract float getVolume();
```

参数说明

参数	类型	描述

参数	类型	描述
value	float	音量返回数值（非系统音量）。返回值范围[0.0, 1.0]。

- setMute：设置静音。

```
public abstract void setMute(boolean value);
```

参数说明

参数	类型	描述
value	boolean	是否设置静音。取值： ○ false：默认值，未开启静音。 ○ true：开启静音。

- isMute：返回是否静音状态，默认返回值false。

```
public abstract boolean isMute();
```

- snapshot：截取当前画面，截图结果通过AliLivePlayerObserver.onSnapShot回调。

```
public abstract void snapshot();
```

- setConfig：设置播放器。

```
public abstract void setConfig(AliLivePlayerConfig liveConfig);
```

参数说明

参数	类型	描述
liveConfig	AliLivePlayerConfig	播放器参数。详情请参见直播播放器SDK包中的*api.txt*文件。

- getConfig：获取配置。

```
public abstract AliLivePlayerConfig getConfig();
```

- createAliLivePlayer：创建播放器实例。

```
public static AliLivePlayer createAliLivePlayer(Context context, boolean enableLog);
```

参数说明

参数	类型	描述
context	Context	应用程序上下文内容。
enableLog	boolean	日志开关。

- releaseAliLivePlayer：销毁播放器实例。

```
public static void releaseAliLivePlayer(AliLivePlayer player);
```

参数说明

参数	类型	描述
player	AliLivePlayer	来自createAliLivePlayer创建的实例player。

- onError：出错回调。

```
public void onError(AliLivePlayer player, ErrorInfo errorInfo) {}
```

参数说明

参数	类型	描述
player	AliLivePlayer	来自createAliLivePlayer创建的实例player。
errorInfo	ErrorInfo	回调错误信息。详情请参见直播播放器SDK包中的*api.txt*文件。

- onRenderingStart：渲染开始回调。

```
public void onRenderingStart(AliLivePlayer player) {}
```

参数说明

参数	类型	描述
player	AliLivePlayer	回调createAliLivePlayer创建的实例player。

- onInfo：信息回调。

```
public void onInfo(AliLivePlayer player, InfoBean info){}
```

参数说明

参数	类型	描述
player	AliLivePlayer	信息回调createAliLivePlayer创建的实例player。
info	InfoBean	信息回调内容。详情请参见直播播放器SDK包中的*api.txt*文件。

- onVideoSizeChanged：分辨率变化回调。

```
public void onVideoSizeChanged(AliLivePlayer player, int width, int height) {}
```

参数说明

参数	类型	描述
player	AliLivePlayer	回调createAliLivePlaye创建的实例player。
width	int	分辨率的宽。
height	int	分辨率的长。

- onSnapShot：截屏回调。

```
public void onSnapShot(AliLivePlayer player, Bitmap image, int width, int height) {}
```

参数说明

参数	类型	描述
player	AliLivePlayer	截屏回调createAliLivePlaye创建的实例player。
image	Bitmap	截图。
width	int	截图的宽。
height	int	截图的长。

- onRenderFrame：音视频帧回调。

```
public void onRenderFrame(AliLivePlayer player, FrameInfo frameInfo) {}
```

参数说明

参数	类型	描述
player	AliLivePlayer	音视频帧回调createAliLivePlayer创建的实例player。
frameInfo	FrameInfo	音视频帧回调内容，默认为false。 详情请参见直播播放器SDK包中的*api.txt*文件。

- onSeiData：SEI信息回调。

```
public void onSeiData(AliLivePlayer player, int type, byte[] data) {}
```

参数说明

参数	类型	描述
player	AliLivePlayer	SEI信息回调createAliLivePlaye创建的实例player。
type	int	信息类型。

参数	类型	描述
data	byte	信息内容。详情请参见直播播放器SDK包中的*api.txt*文件。

3.2.4. iOS端直播播放器SDK

3.2.4.1. 集成iOS端直播播放器

本文为您介绍了iOS端直播播放器SDK的环境要求以及集成方式。

环境要求

类别	说明
系统版本	支持iOS 9.0及以上。
开发工具	建议使用Xcode，本文操作步骤基于Xcode开发。下载Xcode。

版本选择

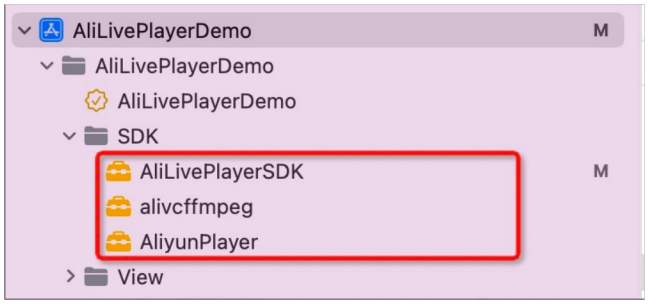
请根据使用场景选择集成直播播放器精简版或完整版。不同版本的使用场景请参见使用场景。

直播播放器精简版集成

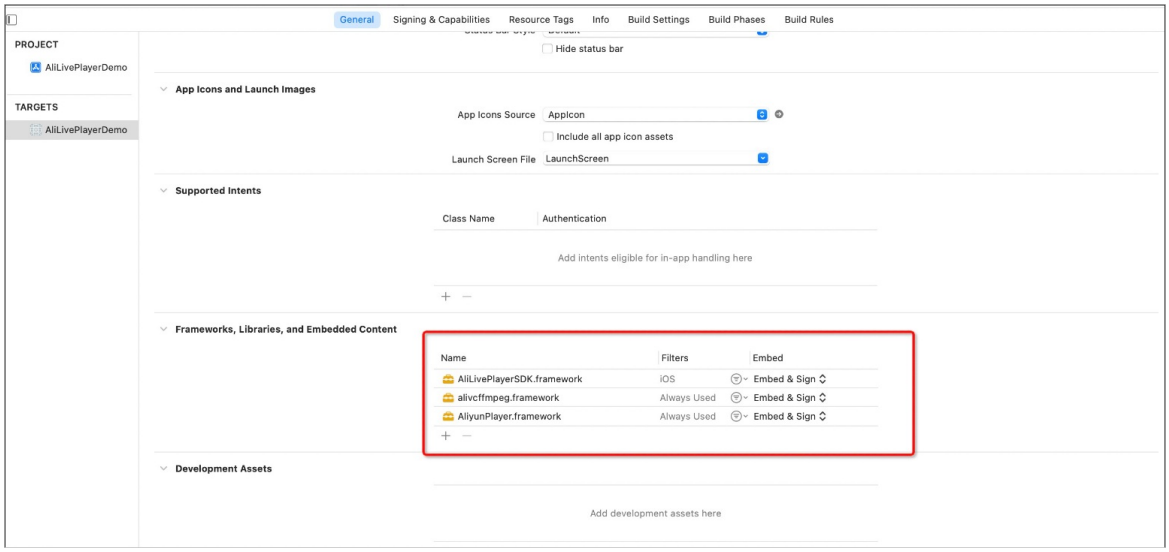
集成SDK时，需要先下载iOS端直播播放器SDK包（包含了播放器SDK及Demo源码），下载地址请参见SDK下载。

1. 复制AliLivePlayer-1.0.0-simplify.zip包里的AliLivePlayerSDK.framework、alivcffmpeg.framework、AliyunPlayer.framework文件到工程的 `SDK` 目录下。

   > **说明** 如果没有，手动创建 `SDK` 文件夹即可。

2. 在Target > General下修改动态库的Embed方式为Embed & Sign。



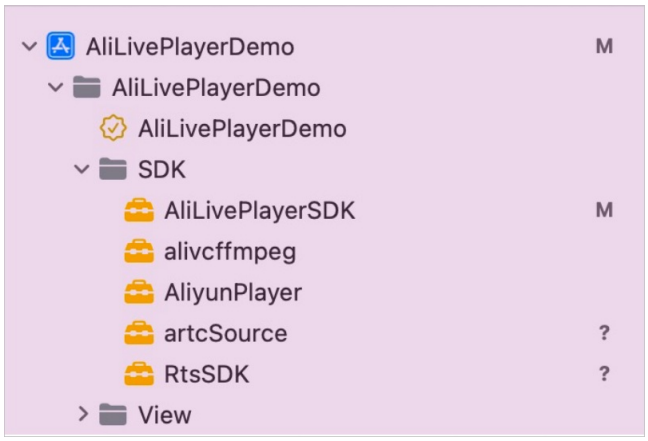
3. 调用接口实现播放功能，请参见实现iOS端播放功能。

直播播放器完整版集成

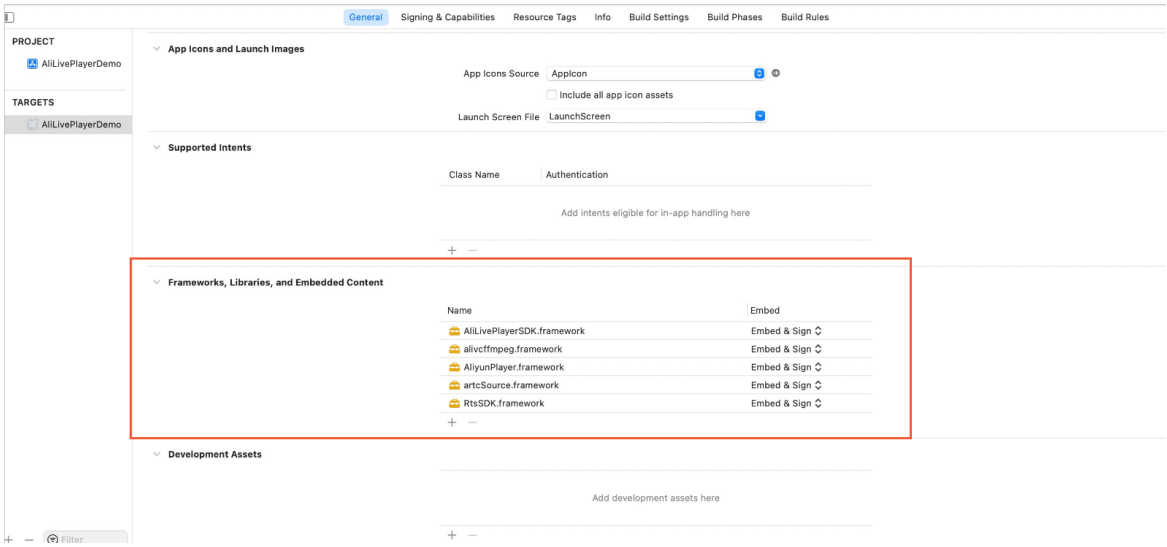
集成SDK时，需要先下载iOS端直播播放器SDK包（包含了播放器SDK及Demo源码），下载地址请参见SDK下载。

1. 复制AliLivePlayer-1.0.0-full.zip包里的AliLivePlayerSDK.framework、alivcffmpeg.framework、AliyunPlayer.framework、artcSource.framework、RtsSDK.framework文件到工程的SDK目录下。

   > 说明 如果没有，手动创建 `SDK` 文件夹即可。



2. 在Target > General下修改动态库的Embed方式为Embed & Sign。



3. 调用接口实现播放功能，请参见实现iOS端播放功能。

参考文档

- 实现iOS端播放功能
- 接口说明

3.2.4.2. 实现iOS端播放功能

直播播放器SDK开放了客户端接口供用户实现播放功能，同时可使用直播播放器其它功能。本文提供实现直播播放器功能的示例代码及接口说明。

背景信息

直播播放器提供一系列接口支持基础的播放功能，您参考以下内容完成接口的调用以实现播放功能。

操作步骤

1. 创建直播播放器，设置渲染View。

```
// 初始化 livePlayer
self.livePlayer = [[AliLivePlayer alloc] init];
// 设置 livePlayer 代理
self.livePlayer.delegate = self;
// 设置 livePlayer 的容器
[self.livePlayer setRenderView:self.playView];
```

2. 设置播放器代理 （可选）。

```
@interface AliLivePlayerViewController () <AliLivePlayerDelegate>
```

```
#pragma mark - AliLivePlayerDelegate
- (void)onRenderingStart:(AliLivePlayer *)player {
}
- (void)onPlayerInfoUpdate:(AliLivePlayer *)player andInfo:(AliLivePlayerInfo *)info {
}
- (void)onRenderFrame:(AliLivePlayer *)player timeMs:(int64_t)timeMs pts:(int64_t)pts {
}
- (void)onVideoSizeChanged:(AliLivePlayer *)player width:(int)width height:(int)height
rotation:(float)rotation {
}
- (void)onSnapShot:(AliLivePlayer *)player image:(UIImage *)image {
}
- (void)onSeiData:(AliLivePlayer *)player type:(int)type data:(NSData *)data {
}
- (void)onErrorWithPlayer:(AliLivePlayer *)player andErrorInfo:(AliLivePlayerErrorInfo
*)errorInfo {
}
```

3. 通过URL播放地址进行播放控制。

> 说明
> - 播放地址支持RTMP、HTTP-FLV、ARTC协议的在线音视频播放。
> - FLV协议目前不支持HTTPS-FLV。

```
/// 开始播放
[self.livePlayer startWithURL:@"rtmp:example.aliyundoc.com"];
```

接口说明

如需实现直播播放器提供的其它功能，请参考接口说明。详情请参见接口说明。

3.2.4.3. 接口说明

通过阅读本文，您可以了解阿里云直播播放器的接口详情。

目录

直播播放器接口（类：AliLivePlayer）

接口	描述
init	初始化播放器。
initWithConfiguration	初始化播放器，并对播放器进行设置。
setRenderView	设置播放器的容器。
startWithURL	通过URL地址开始播放。
pause	暂停播放。

接口	描述
resume	恢复播放。
stop	停止播放。
setRotateMode	设置直播播放器的旋转模式。
setMirrorMode	设置直播播放器的镜像模式。
setScaleMode	设置直播播放器的填充模式。
setVolume	设置直播播放器音量。
getCurrentVolume	获取直播播放器音量。
setMute	设置静音。
isMute	获取静音状态。
snapshot	截取当前画面。
setConfiguration	设置播放器的初始化配置。
getCurrentConfiguration	获取播放器的配置。

直播播放器渲染与信息回调（类：AliLivePlayer）

接口	描述
onRenderingStart	播放器开始渲染。
onPlayerInfoUpdate	播放器信息更新。
onRenderFrame	播放器渲染信息回调。
onVideoSizeChanged	视频大小变化回调。
onSnapShot	获取截图回调。
onSeiData	SEI回调。
onErrorWithPlayer	错误代理回调。

直播播放器工厂（类：AliLivePlayer）

接口	描述
createAliLivePlayer	创建播放器实例。
releaseAliLivePlayer	销毁播放器实例。

直播播放器配置（类：AliLivePlayerConfiguration）

接口	描述
enableSEI	SEI数据回调开关。
enableFrameInfo	音视频帧回调开关。
maxDelayTime	最大延迟时间，单位：ms。
maxBufferDuration	播放器加载缓冲数据最大缓冲区时长，单位：ms。
highBufferDuration	最高加载数据的缓冲时长，单位：ms。
startBufferDuration	起播缓冲区时长，单位：ms。
maxProbeSize	最大probe大小，单位：byte 。

接口详情

- init：初始化播放器。

```
- (instancetype)init;
```

- initWithConfiguration：初始化播放器，并对播放器进行设置。

```
- (instancetype)initWithConfiguration:(AliLivePlayerConfiguration *)configuration;
```

参数说明

参数	类型	描述
configuration	AliLivePlayerConfiguration	直播播放器配置，详细信息请参见直播播放器SDK下载包中*API.md*文件。

- setRenderView：设置播放器的容器。

```
- (void)setRenderView:(UIView *)view;
```

参数说明

参数	类型	描述
view	UIView	直播播放器的容器。

- startWithURL：开始播放。

```
- (void)startWithURL:(NSString *)url;
```

参数说明

参数	类型	描述

参数	类型	描述
url	NSString	播放URL地址，支持RTMP、HTTP-FLV、ARTC协议的在线音视频播放。

- pause：暂停播放。

```
- (void)pause;
```

- resume：恢复播放。

```
- (void)resume;
```

- stop：停止播放。

```
- (void)stop;
```

- setRotateMode：设置直播播放器画面旋转模式。

```
- (void)setRotateMode:(AliLivePlayerRotateMode)rotateMode;
```

参数说明

参数	类型	描述
rotateMode	AliLivePlayerRotateMode	设置画面旋转模式，取值： ○ AliLivePlayerRotateModeZero：无旋转。 ○ AliLivePlayerRotateMode90：旋转90度。 ○ AliLivePlayerRotateMode180：旋转180度。 ○ AliLivePlayerRotateMode270：旋转270度。

- setMirrorMode：设置直播播放器镜像模式。

```
- (void)setMirrorMode:(AliLivePlayerMirrorMode)mirrorMode;
```

参数说明

参数	类型	描述
mirrorMode	AliLivePlayerMirrorMode	设置镜像模式，取值： ○ AliLivePlayerMirrorModeNone：无镜像。 ○ AliLivePlayerMirrorModeHorizontal：水平镜像。 ○ AliLivePlayerMirrorModeVertical：垂直镜像。

- setScaleMode：设置直播播放器画面填充模式。

```
- (void)setScaleMode:(AliLivePlayerScaleMode)scaleMode;
```

参数说明

参数	类型	描述
scaleMode	AliLivePlayerScaleMode	设置画面填充模式，取值： ○ AliLivePlayerScaleAspectFit：比例填充。 ○ AliLivePlayerScaleAspectFill：比例全屏。 ○ AliLivePlayerScaleToFill：拉伸全屏。

- setVolume：设置直播播放器音量。

```
- (void)setVolume:(float)value;
```

参数说明

参数	类型	描述
value	float	音量数值。

- getCurrentVolume：获取直播播放器音量。

```
- (float)getCurrentVolume;
```

参数说明

参数	类型	描述
getCurrentVolume	float	音量返回数值。

- setMute：设置静音。

```
- (void)setMute:(BOOL)mute;
```

参数说明

参数	类型	描述
mute	BOOL	静音开启关闭。

- isMute：获取静音状态。

```
- (BOOL)isMute;
```

- snapshot：截取当前画面。

```
- (void)snapshot;
```

- setConfiguration：设置播放器的初始化配置。

```
- (void)setConfiguration:(AliLivePlayerConfiguration *)configuration;
```

参数说明

参数	类型	描述
configuration	AliLivePlayerConfiguration	播放器初始化配置，详细信息请参见直播播放器SDK下载包中*API.md*文件。

- getCurrentConfiguration：获取播放器的配置。

```
- (AliLivePlayerConfiguration *)getCurrentConfiguration;
```

参数说明

参数	类型	描述
getCurrentConfiguration	AliLivePlayerConfiguration	播放器初始化配置，详细信息请参见直播播放器SDK下载包中*API.md*文件。

- onRenderingStart：播放器开始渲染。

```
- (void)onRenderingStart:(AliLivePlayer *)player;
```

参数说明

参数	类型	描述
player	AliLivePlayer	播放器player指针。

- onPlayerInfoUpdate：播放器信息更新。

```
- (void)onPlayerInfoUpdate:(AliLivePlayer *)player andInfo:(AliLivePlayerInfo *)info;
```

参数说明

参数	类型	描述
player	AliLivePlayer	播放器player指针。
info	AliLivePlayerInfo	播放器信息，详细信息请参见直播播放器SDK下载包中*API.md*文件。

- onRenderFrame：播放器渲染信息回调。

```
- (void)onRenderFrame:(AliLivePlayer *)player timeMs:(int64_t)timeMs pts:(int64_t)pts;
```

参数说明

参数	类型	描述
player	AliLivePlayer	播放器player指针。

参数	类型	描述
timeMs	int64_t	渲染时的系统时间。
pts	int64_t	视频帧pts。

- onVideoSizeChanged：视频大小变化回调。

```
- (void)onVideoSizeChanged:(AliLivePlayer *)player width:(int)width height:(int)height rotation:(float)rotation;
```

参数说明

参数	类型	描述
player	AliLivePlayer	播放器player指针。
width	int	视频宽度。
height	int	视频高度。
rotation	float	视频旋转角度。

- onSnapShot：截屏回调。

```
- (void)onSnapShot:(AliLivePlayer *)player image:(UIImage *)image;
```

参数说明

参数	类型	描述
player	AliLivePlayer	播放器player指针。
image	UIImage	图像。

- onSeiData：SEI信息回调。

```
- (void)onSeiData:(AliLivePlayer *)player type:(int)type data:(NSData *)data;
```

参数说明

参数	类型	描述
player	AliLivePlayer	播放器player指针。
type	int	信息类型
data	NSData	信息数据，详细信息请参见直播播放器SDK下载包中*API.md*文件。

- onErrorWithPlayer：错误代理回调。

```
- (void)onErrorWithPlayer:(AliLivePlayer *)player andErrorInfo:(AliLivePlayerErrorInfo *)errorInfo;
```

参数说明

参数	类型	描述
player	AliLivePlayer	播放器player指针。
errorInfo	AliLivePlayerErrorInfo	播放器错误描述信息，详细信息请参见直播播放器SDK下载包中*API.md*文件。

- createAliLivePlayer：创建一个播放器实例。

```
+ (AliLivePlayer *)createAliLivePlayer;
```

- releaseAliLivePlayer：销毁一个播放器实例。

```
+ (void)releaseAliLivePlayer:(AliLivePlayer *)player;
```

- enableSEI：SEI数据回调开关，默认：NO。

```
@property (nonatomic, assign) BOOL enableSEI;
```

- enableFrameInfo：音视频帧回调开关，默认：NO。

```
@property (nonatomic, assign) BOOL enableFrameInfo;
```

- maxDelayTime：最大延迟时间，单位：ms。

```
@property (nonatomic, assign) int maxDelayTime;
```

> **说明** 当延时比较大时，播放器SDK内部会进行追帧，保证播放器的延时在最大延迟时间范围内。

- maxBufferDuration：播放器加载缓冲数据最大缓冲区时长，单位：ms。

```
@property (nonatomic, assign) int maxBufferDuration;
```

> **说明** 在配置缓冲区时长内，播放器单次最多加载的缓冲数据。

- highBufferDuration：最高加载数据的缓冲时长，单位：ms。

```
@property (nonatomic, assign) int highBufferDuration;
```

> **说明** 当网络不好导致加载数据时，如果加载的缓冲时长达到配置值，则结束加载状态。

- startBufferDuration：起播缓冲区时长，单位：ms。

```
@property (nonatomic, assign) int startBufferDuration;
```

> **说明** 时间设置越短，起播越快。可能出现播放之后很快进入加载状态的情况。

- maxProbeSize：最大probe大小，单位：byte 。

```
@property (nonatomic, assign) int maxProbeSize;
```

4.常见问题

您在使用视频直播产品时，如果收到系统提示的错误信息，可查阅视频直播错误代码汇总匹配错误信息。对于其他问题，您可根据下列分类匹配问题场景和解决方案。

> **说明** 本页面列举的文档较多，您可以通过搜索框搜索相关文档或者在本页面按Ctrl+F搜索关键字。

基础问题

- 视频直播服务支持哪些推流协议？
- 视频直播服务支持哪些直播流播放协议？
- 常见的推流包括哪几种来源？
- 视频直播支持的分辨率和对应合适的码率
- 视频直播生成推流和播放地址的PHP代码示例
- 视频直播生成推流和播放地址的Java代码示例
- 购买中国内地视频直播服务的实名登记
- 视频直播配置CNAME域名解析后无法正常访问的解决方法
- CNAME解析常见问题
- 边缘推流和直播中心有什么关系？
- 如何选择直播中心？
- 如何选择域名的CDN加速区域？

推流

- 直播推流异常排查
- iOS推流横屏方案建议
- 切换加速区域的影响是什么？

播流/播放

- 直播播流相关问题
- 控制台无法观看直播流和录制文件
- 直播无法观看或播流404问题排查
- 直播RTMP和FLV播放正常而HLS播放失败
- Flash播放器无法拖拽
- 播放器播放m3u8提示跨域错误
- 使用播放器播放时画面花屏
- 使用阿里云视频直播时视频画面存在黑边间隙
- 直播出现音视频不同步
- 视频直播出现卡顿现象

功能使用

- 直播禁止推流功能相关问题
- 视频直播鉴权功能相关问题

- 如何保护直播版权，防止非授权用户访问？
- 视频直播鉴权常见问题
- 转码出的文件与转码模板的宽高参数相反
- 无法使用HTTPS访问问题排查

第三方工具介绍

> **说明** 以下文档包含的第三方产品信息，仅供参考。阿里云对第三方产品的性能、可靠性以及操作可能带来的潜在影响，不做任何暗示或其他形式的承诺。

- 第三方推流工具常见问题
- OBS推流工具
- VLC播放器