

ALIBABA CLOUD

阿里云

视频直播
开发指南

文档版本：20220705

 阿里云

法律声明

阿里云提醒您在使用或阅读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 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

- 1.概述 ----- 06
- 2.权限管理 ----- 10
 - 2.1. 权限管理概述 ----- 10
 - 2.2. 创建RAM用户并授权 ----- 12
 - 2.3. 自定义授权 ----- 14
 - 2.4. 直播加密服务关联角色 ----- 18
- 3.直播基础服务 ----- 20
 - 3.1. 直播基础服务概述 ----- 20
 - 3.2. 直播域名 ----- 22
 - 3.3. 推流地址和播放地址 ----- 26
 - 3.4. 直播推流 ----- 28
 - 3.5. 直播播放 ----- 30
 - 3.6. 直播基础服务FAQ ----- 32
- 4.直播加速 ----- 36
 - 4.1. 边缘推流 ----- 36
- 5.直播安全 ----- 37
 - 5.1. 直播安全概述 ----- 37
 - 5.2. 安全加速 ----- 41
 - 5.3. 访问控制 ----- 44
 - 5.4. URL鉴权 ----- 46
 - 5.5. 鉴权代码示例 ----- 49
 - 5.6. 阿里云视频加密 ----- 54
 - 5.7. DRM加密 ----- 56
 - 5.8. 内容安全 ----- 57
- 6.直播延迟 ----- 67
- 7.直播录制 ----- 70

7.1. 概述	70
7.2. 自动录制	71
7.3. 按需录制	73
7.4. 手动录制	75
7.5. 录制内容检索与管理	76
8.HTTP配置	79
9.直播流媒体处理	80
9.1. 直播转码	80
9.2. 直播时移	83
9.3. 直播录制	86
9.4. 直播截图	90
10.云端合流	92
10.1. 云端合流概述	92
10.2. 合流预设布局参考	95
10.3. 合流代码示例	97
11.直播管理	105
11.1. 直播流管理	105
11.2. 回调管理	107
11.3. 文件管理	121
12.数据统计	123
13.日志管理	129
14.配额管理	133
15.常见问题FAQ	135
15.1. 视频直播服务支持哪些直播流播放协议?	135
15.2. 视频直播支持的分辨率和对应合适的码率	135
15.3. 直播禁止推流功能相关问题	136
15.4. 视频直播鉴权功能相关问题	141

1. 概述

开发指南从视频直播功能实现的角度，介绍了视频直播的功能和使用方法。本文介绍开发指南包含的内容。

视频直播包含权限管理、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。

权限管理

功能	说明	操作支持	相关文档
RAM用户管理	支持创建多个RAM用户，使用系统授权策略或自定义授权策略对RAM用户进行操作权限的授权。	RAM访问控制台	权限管理概述

基础服务

功能	说明	操作支持	相关文档
域名管理	支持对域名进行添加、删除、启用、停用等管理操作。	<ul style="list-style-type: none"> 直播控制台 API 	直播域名
域名配置	支持配置CNAME，关联推流域名和播流域名，关联主、子播流域名。	<ul style="list-style-type: none"> 直播控制台 API 	直播域名
地址管理	支持查看和自动生成推流地址和播放地址。	直播控制台	推流地址和播放地址
推流、拉流与播流	直播流的接入和分发服务。支持接入第三方直播流。	<ul style="list-style-type: none"> 直播控制台 第三方推流工具和播放工具 	推流、拉流与播流

直播加速

功能	说明	操作支持	相关文档
边缘推流	优先将视频推流至最优CDN节点，保证用户访问的都是最佳的上行网络。	不涉及	边缘推流

直播安全

功能	说明	操作支持	相关文档
权限管理	支持创建RAM用户并授权。	RAM访问控制台	权限管理概述
安全加速	支持开启HTTPS安全加速。	<ul style="list-style-type: none"> 直播控制台 API 	安全加速

功能	说明	操作支持	相关文档
访问控制	支持Refer防盗链、User-Agent黑白名单、IP黑白名单。	<ul style="list-style-type: none"> 直播控制台 API User-Agent黑白名单仅支持工单 	访问控制
URL鉴权	支持通过加密URL进行鉴权。	直播控制台	URL鉴权
阿里云视频加密	支持对视频数据进行加密。	<ul style="list-style-type: none"> API 提交工单 	阿里云视频加密
内容安全	支持对音频和视频进行智能审核，并根据审核结果禁推直播流。	<ul style="list-style-type: none"> 直播控制台 API 	内容安全

直播延时

功能	说明	操作支持	相关文档
配置延时	支持根据不同的播流协议来配置延时时间。	直播控制台	直播延时配置
超低延时直播	提供易接入、毫秒级别延迟、高并发、高清流畅的音视频直播服务。	SDK	超低延时直播简介

HTTP配置

功能	说明	操作支持	相关文档
HTTP消息头	支持配置HTTP消息头。	直播控制台	HTTP配置

直播流媒体处理

功能	说明	操作支持	相关文档
直播转码	支持通用转码、自定义转码和RTS转码。	<ul style="list-style-type: none"> 直播控制台 API 	直播转码
直播时移	支持30天内的直播时移。	直播控制台	直播时移
直播录制	支持录制直播内容至OSS或VOD。	<ul style="list-style-type: none"> 直播控制台 API 	直播录制
直播截图	支持实时截图。	<ul style="list-style-type: none"> 直播控制台 API 	直播截图

云端合流

功能	说明	操作支持	相关文档
云端合流	支持将多路视频源推至直播中心，按照设置的布局进行合流。	API	云端合流概述

云导播台

功能	说明	操作支持	相关文档
云导播台	支持对直播流、视频文件、图片等多种素材进行处理，经过媒体处理和包装，输出新的直播流。	<ul style="list-style-type: none"> 直播控制台 API 	简介

直播管理

功能	说明	操作支持	相关文档
直播流管理	<ul style="list-style-type: none"> 支持直播在线流、历史流内容的查看及对视频流进行加入黑名单、中断推流操作。 支持流粒度的监控信息查询，支持快速检索指定的流。 	<ul style="list-style-type: none"> 直播控制台 API 	直播流管理
回调管理	支持直播推流状态、直播录制、按需录制、直播截图和智能审核的事件回调。		回调管理
文件管理	<ul style="list-style-type: none"> 可对录制的点播视频进行快速的索引剪辑及管理。 可对录制下来的视频进行内容预览。 可实时查看截图存储下来的图片。 	直播控制台	文件管理

数据统计

功能	说明	操作支持	相关文档
资源监控	支持流量、错误码的状态监控。	<ul style="list-style-type: none"> 直播控制台 API 	数据统计
实时监控	可对指定的视频流进行实时的推流监控。		
访问统计	可对访问数据进行监控。		
用量查询	可对使用视频直播的流量带宽、录制、截图、转码等用量数据进行查询。		

日志管理

功能	说明	操作支持	相关文档
日志下载	支持流量，转码的用量日志下载。	直播控制台	

功能	说明	操作支持	日志管理 相关文档
实时日志	支持实时日志推送，将直播域名日志实时推送至日志服务，并进行日志分析。	直播控制台	

2. 权限管理

2.1. 权限管理概述

视频直播会对每一次发起操作请求的用户身份进行验证，通过AccessKey验证该账号是否拥有相应的权限。视频直播支持主账号AK和RAM用户AK，本文为您介绍这两种方式和对比，以及阿里云提供的系统授权策略和自定义授权策略。

简介

您可以使用视频直播服务提供的API接口或SDK访问视频直播服务。视频直播会针对每一次发起的请求，根据当前的操作验证用户身份，验证该账号是否拥有相应的权限，验证用户身份都需要使用AccessKey。

通过阿里云访问控制服务（RAM），可为RAM用户授予相关权限，以达到RAM用户在授权范围使用视频直播控制台的目的。

访问控制基本概念

- 访问控制（RAM）

RAM（Resource Access Management）是阿里云提供的用户身份管理与资源访问控制服务。更多信息，请参见[什么是访问控制](#)。

 **说明** RAM服务提供的权限的分割和管理，而并非资源本身的隔离，也就是RAM用户从属于主账号，并且这些RAM用户下不能拥有实际的任何资源，所有资源都属于主账号。

- 阿里云账号（主账号）

阿里云账号（主账号）是阿里云资源归属、资源使用计量计费的基本主体。阿里云账号为其名下所拥有的资源付费，并对其名下所有资源拥有完全控制权限。

- RAM用户

根据阿里云的主账号创建的RAM用户，每个RAM用户拥有自己AccessKey，可以和阿里云主账号一样正常的完成有权限的操作。一般来说，这里的RAM用户可以理解为具有某种权限的用户，可以被认为是一个具有某些权限的操作发起者。

- 授权策略（RAM Policy）

授权策略使用语法结构描述一组权限，它可以精确地描述被授权的资源集、操作集以及授权条件。通过设置权限策略，并给用户或用户组附加授权，就可以精确控制用户访问您名下哪些资源或服务的权限，比如限制您的用户只拥有上传、播放或审核权限。

- 访问密钥（AccessKey）

AccessKey（简称AK），包括访问身份验证中用到的AccessKey ID和AccessKey Secret。云产品通过使用AccessKey ID和AccessKey Secret对称加密的方法，来验证某个请求的发送者身份。

- AccessKey ID：用于标识用户。
- AccessKey Secret：用户用于加密签名字符串，以及云产品用来验证签名字符串的密钥，AccessKey Secret必须保密。

 **说明** AccessKey Secret只在创建时显示，不支持查询，请妥善保管。

- AK对：指AccessKey ID和AccessKey Secret。

更多关于访问控制的概念，请参见[访问控制基本概念](#)。

不同验证方式及对比

目前访问视频直播使用的AK有如下2种类型：

- 主账号AK

主账号AK特指视频直播开通者（即阿里云网站注册的账号）的AK，每个阿里云主账号提供的AK对拥有的资源有完全的权限。每个阿里云主账号能够同时拥有不超过5个启用或者禁用 AK对。您可以登录[AccessKey管理控制台](#)，申请新增或删除AK对。每个AK对都有启用和禁用两种状态，只有启用的AK对才能在身份验证时使用。

 **警告** 由于主账号AK有完全的权限，一旦泄露风险巨大，不建议使用其访问视频直播服务。

- RAM用户AK

RAM是阿里云提供的资源访问控制服务。RAM用户AK指的是通过RAM被授权的AK。这组AK只能按照RAM定义的规则去访问视频直播的资源。通过RAM，您可以集中管理您的用户（比如员工、系统或应用程序），以及控制用户可以访问您名下哪些资源的权限。比如能够限制您的用户只拥有视频播放权限。RAM用户从属于主账号，并且这些账号下不能拥有实际的任何资源，所有资源都属于主账号。

您可以登录[RAM访问控制台](#)创建RAM用户，获取AK，并授予相应权限。具体操作，请参见[创建RAM用户并授权](#)。

不同验证方式的对比

验证方式	风险	权限	时效	适用场景
主账号AK	极大	管理和操作视频直播所有资源的权限	启用后一直有效	超级管理员进行操作。不建议在程序里使用，尤其不要放到客户端。
RAM用户AK	较小	根据授权策略获得相应的权限	启用后一直有效	授权进行具体的管理等操作。可准备多个RAM用户，如遇AK泄露（人员离职等）需要更换；建议在服务端使用。

授权策略

使用授权策略，可对RAM用户进行方便、精确的授权。阿里云提供了系统授权策略和自定义授权策略两种方式。

- 系统授权策略

使用视频直播涉及到如下三种系统授权策略。

策略名称	说明	操作权限
AliyunLiveFullAccess	管理视频直播（Live）的权限	视频直播所有控制台操作和API
AliyunLiveReadOnlyAccess	只读访问视频直播的权限	视频直播所有读取类操作和API，如以Describe开头的接口
AliyunMTSFullAccess	管理媒体转码服务（MTS）的权限	媒体转码服务所有控制台操作和API

- 自定义授权策略

如果系统策略无法满足您的需求，您可以通过创建自定义策略实现精细化权限管理。
使用视频直播服务涉及到OSS授权策略和Live授权策略，更多信息和具体操作，请参见[授权示例](#)。

说明 使用直播控制台，必须给您的子账号开通以下授权才能正常使用服务。

- Live（必选）授权使用直播服务，直接使用系统内置AliyunLiveFullAccess授权策略。
- OSS（必选）授权使用截图存储服务，可按需自定义，请参见[授权示例](#)。

若要将视频直播的录制文件和截图文件保存到用户OSS的Bucket中，需要对直播服务Live授权访问OSS，使用的是AliyunMTSDefaultRole角色。[去授权](#)。

2.2. 创建RAM用户并授权

使用RAM用户可以帮助您有效避免AccessKey或者密码泄露导致的安全问题。本文介绍如何创建一个RAM用户，并设置相应的权限。

前提条件

已经开通访问控制服务和视频直播服务。

创建RAM用户

1. 创建RAM用户。
创建RAM用户请参见[创建RAM用户](#)。
完成第5步输入登录名称后，后续步骤需要按照下文操作。
2. 在访问方式区域勾选OpenAPI调用访问，单击确定。
单击确定后会弹出手机验证窗口，完成验证码验证后，自动生成该RAM用户的AccessKey。
3. 单击用户信息右侧的复制，保存用户登录名称、登录密码、AK对等用户信息。

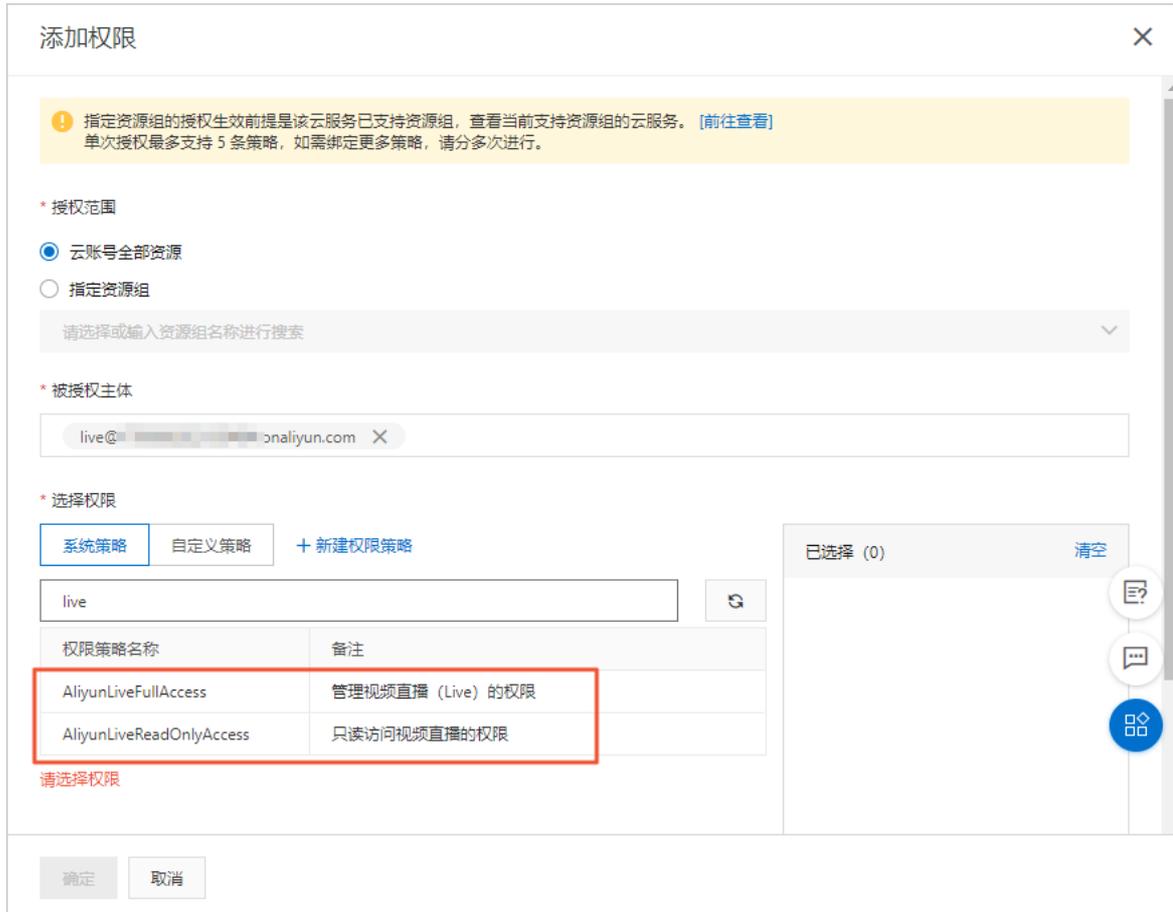
说明 请保存用户信息并妥善保管，用于后续的访问。

授权

1. 返回用户管理界面，这里显示上一步创建的账号。
创建完成之后，该RAM用户还是没有任何权限。
2. 单击用户信息右侧的添加权限。



3. 选择授权范围，在选择权限下输入框输入 `live`，选择筛选出的权限策略，单击确定。



RAM用户若要使用录制视频转码服务，需要授予RAM用户媒体转码服务使用权限。您可以直接使用系统内置 AliyunMTSFullAccess 授权策略。

您可以使用系统授权策略进行授权，也可以使用自定义的授权策略。授权策略的定义，请参见[授权策略](#)。自定义授权策略的具体操作，请参见[自定义授权](#)。

4. 授权完成之后如果该账号需要控制台登录等权限，也可以单击用户名称，在认证管理页面来完成操作。



2.3. 自定义授权

自定义授权可以满足用户个性化的授权策略需求，RAM控制台提供了可视化配置和脚本配置两种配置方式。本文为您介绍授权的基本概念、使用场景、基本步骤和授权的语法、示例。

使用场景

由于系统策略的授权粒度比较粗，很多时候并不能满足您的细粒度权限控制的需求，此时可以使用此时可以基于RAM Policy实现自定义授权。

授权操作

您可以在RAM控制台的新建自定义权限策略页面，新建自定义权限策略，请参见[创建自定义权限策略](#)。

可视化配置

选择可视化配置，单击添加授权语句，即可通过可视化的方式配置权利效力、产品/服务、操作名称、资源和限制条件。



授权语法

在开始设置前，需要先了解权限策略的基本元素和语法结构，更多信息，请参见[权限策略基本元素](#)和[权限策略语法和结构](#)。

以视频直播系统策略 `AliyunLiveReadOnlyAccess` 为例：

```

{
  "Version": "1",
  "Statement": [
    {
      "Action": "Live:Describe*",
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}

```

参数解释如下：

- Version**
 Version定义了Policy的版本，本文中Version可以固定设置为1。
- Statement**
 通过Statement描述授权语义，其中可以根据业务场景包含多条语义，每条包含对Action、Effect、Resource 和 Condition的描述。每次请求，系统会逐条依次匹配检查，所有匹配成功的Statement会根据Effect的设置不同分为通过（Allow）、禁止（Deny），其中禁止（Deny）的优先。如果匹配成功的都为通过，该条请求即鉴权通过。如果匹配成功有一条禁止，或者没有任何条目匹配成功，该条请求被禁止访问。
- Action**
 直播支持的Action操作与API一一对应，格式为 `Live:API名称`，如上面示例的播放操作：`Live:Describe*`，注意多个使用英文逗号分隔。通过指定授权的Action列表，就能组合出想要的权限分组。
 所有可用操作，请参见[API概览](#)。

● Resource

Resource指代的是Live上面的某个具体的资源或者某些资源（支持通配符*），Resource的规则是 `acs:live:{region}:*`。Resource也是一个列表，可以有多个Resource。其中的region字段暂不支持，请设置为 `*`。由于直播现在没有进一步区分资源，故一般建议授权直播时Resource填 `*`。

● Condition

Condition代表Policy授权的一些条件，对访问来源等进行限制，为可选项。

支持的条件如下：

Condition	功能	合法取值
acs:SourceIp	指定 IP 地址或网段	普通的IP, 支持通配符*
acs:SecureTransport	是否是 HTTPS 协议	true或者false
acs:MFAPresent	用户登录时是否使用了多因素认证	true或者false
acs:CurrentTime	指定合法的访问时间（云端接收到请求的时间）	ISO8601格式，例如：2012-11-11T23:59:59Z

下文示例表示只允许IP来源为42.160.1.0的请求者访问播放接口：

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": "Live:Describe*",
      "Resource": "*",
      "Effect": "Allow",
      "Condition": {
        "IpAddress": {
          "acs:SourceIp": "42.160.1.0"
        }
      }
    }
  ]
}
```

授权示例

根据API概览，可以得到所有对应的Action，进而分组控制权限，以下以OSS授权策略、Live授权策略为例。

● OSS授权策略

权限描述：

- 对指定的Bucket有所有操作权限；
- 有查看Bucket列表权限。

```

{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "oss:*"
      ],
      "Resource": [
        "acs:oss:*:*:$Bucket",
        "acs:oss:*:*:$Bucket/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "oss:ListBuckets"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}

```

• Live 授权策略

权限描述：

- 对指定的Live加速域名有所有权限；
- 有查询Live加速域名的权限。

```

{
  "Version": "1",
  "Statement": [
    {
      "Action": "live:*",
      "Resource": [
        "acs:cdn:*:$Uid:domain/$DomainName"
      ],
      "Effect": "Allow"
    },
    {
      "Action": "live:Describe*",
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}

```

② 说明

- 以上示例中，该API分组下如果增加了新的接口，则需要相应更新 `Action` 部分的列表。
- 各个服务的资源授权策略中使用了如下变量，请您替换为您实际的资源实例名称。

变量说明

- Uid
\$Uid：云账号 ID，可通过 [控制台 > 账号管理 > 安全设置](#) 进行查询。
- Bucket
\$Bucket：OSS bucket。
- Live
\$DomainName：Live加速域名名称。

2.4. 直播加密服务关联角色

本文介绍直播加密服务的关联角色：AliyunServiceRoleForLiveKes，以及如何删除该角色。

直播加密服务的关联角色AliyunServiceRoleForLiveKes，是视频加密服务为了创建加密使用的密钥对，需要获取[密钥管理服务KMS](#)的访问权限而提供的RAM角色。

 **说明** 更多关于服务关联角色的信息请参见[服务关联角色](#)。

应用场景

阿里云视频直播的视频加密功能，需要访问密钥管理服务KMS的资源时，可通过自动创建的直播加密服务关联角色AliyunServiceRoleForLiveKes获取访问权限。

权限说明

AliyunServiceRoleForLiveKes具备的访问权限如下：

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

删除AliyunServiceRoleForLiveKes

如果您使用了视频加密功能，然后您出于安全考虑，需要删除AliyunServiceRoleForLiveKes角色，则需要先明确删除后的影响：直播加密服务无法创建加密使用的密钥对，从而可能会影响直播加密功能的正常使用。

删除AliyunServiceRoleForLiveKes的步骤如下：

1. 登录[RAM控制台](#)，在左侧导航栏单击[RAM角色管理](#)。
2. 在[RAM角色管理](#)页面的搜索框中，输入AliyunServiceRoleForLiveKes，自动搜索到名称为

AliyunServiceRoleForLiveKes的RAM角色。

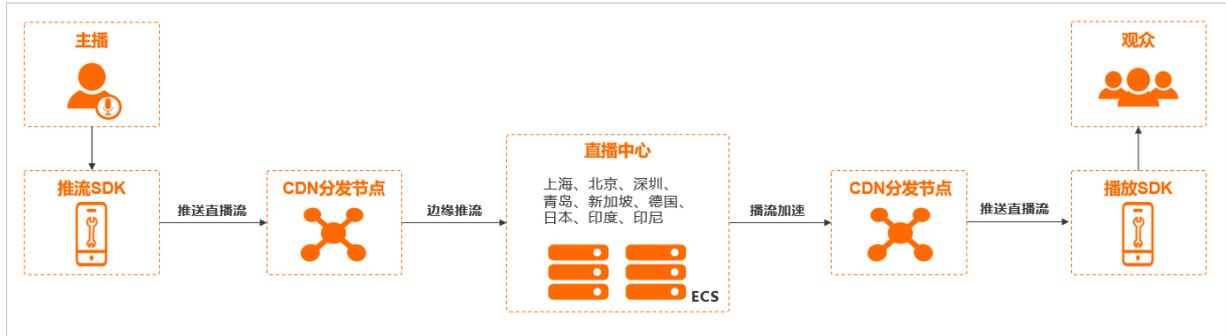
3. 在右侧操作列，单击删除。
4. 在删除RAM角色对话框，单击确定。

3.直播基础服务

3.1.直播基础服务概述

视频直播提供了完善的直播推流和播流服务。本文介绍完成视频直播流的推流与播流最基础的步骤。

视频直播基础流程



1. 主播使用采集设备采集直播内容，并通过推流SDK推送直播流到CDN分发节点。
2. 视频直播服务通过边缘推流的方式将直播流推送至阿里云直播中心。推送的视频流通过CDN边缘节点进行加速保证上行传输的稳定性。
3. 视频直播服务将直播流从阿里云直播中心推送到CDN分发节点。
4. 观众可以通过播放SDK观看推送的直播流。

操作流程



编号	节点	操作方式	操作	API参考	相关文档
1	准备工作	控制台	直播控制台 1. 注册阿里云账号 2. 个人实名认证或企业实名认证 3. 什么是阿里云域名服务 4. 开通与购买视频直播	不涉及	不涉及
2	添加域名	控制台	1. 添加域名 2. 验证域名归属权 3. 配置CNAME域名解析	添加直播域名	直播域名

编号	节点	操作方式	操作	API参考	相关文档
3	关联域名	<ul style="list-style-type: none"> 控制台 API 	<ul style="list-style-type: none"> 关联播流域名和推流域名 关联主、子播流域名 	<ul style="list-style-type: none"> 添加播流域名和推流域名的映射关系 添加主播流域名和子播流域名映射 	关联域名
4	配置访问控制	<ul style="list-style-type: none"> 控制台 API 	<ul style="list-style-type: none"> 配置URL鉴权 配置防盗链 配置IP黑白名单 	批量配置域名	<ul style="list-style-type: none"> URL鉴权 访问控制
5	生成地址	控制台	地址生成器	不涉及	推流地址和播放地址
6	推流	推流工具	<ol style="list-style-type: none"> 下载并安装OBS推流工具 填写推流URL和流名称 请参见 推流设置与操作 。	不涉及	推流、拉流与播流
7	播放	播放器	<ol style="list-style-type: none"> 下载并安装VLC播放器 填写播放地址 请参见 播放设置与查看 。	不涉及	推流、拉流与播流

支持协议

视频直播推流支持使用RTMP协议、SRT协议和ARTC协议（超低延时直播），播流支持使用RTMP协议、FLV协议、HLS协议和ARTC协议（超低延时直播）。

RTMP协议、FLV协议、HLS协议和ARTC协议（超低延时直播）四种协议对比如下：

协议类型	延时	优点	缺点	特点	适合端	场景推荐
RTMP	1s~3s	延时低	<ul style="list-style-type: none"> iOS平台要开发支持相关协议的播放器 使用非标准TCP端口 	TCP长连接	PC端	实时性要求不高的直播
FLV	1s~3s	<ul style="list-style-type: none"> 延时低 可通过HLS解封装播放 	需要集成SDK才能播放	TCP长连接	PC端	实时性要求不高的直播
HLS	> 10s	<ul style="list-style-type: none"> iOS、Android和H5原生支持良好 可通过HLS解封装播放 	延时高	HTTP短连接	PC端、移动端	实时性要求不高的直播；移动端和H5端

协议类型	延时	优点	缺点	特点	适合端	场景推荐
ARTC	1s	<ul style="list-style-type: none"> 超低延时 抗弱网能力强 	H5播放不支持B帧和AAC音频（可通过阿里云RTS转码功能去除B帧并将音频转为Opus）	UDP	PC端、移动端	实时性要求高的直播，如电商带货、在线教育、社交互动等

3.2. 直播域名

阿里云视频直播在服务端主要提供直播流接入、分发、实时流媒体处理服务，使用这些服务的都需要添加直播域名并进行配置。本文介绍直播域名的概念，如何添加域名，如何关联域名和直播域名功能配置生效表。

直播域名简介

直播域名包括推流域名和播流域名。

- 推流域名

推流域名是用于推送直播流的域名。进行添加域名操作时，将**业务类型**（LiveDomainType）配置为**推流域名**（liveEdge），则此域名就是推流域名。

- 播流域名

播流域名是用于播放直播流的域名。进行添加域名操作时，将**业务类型**（LiveDomainType）配置为**播流域名**（liveVideo），则此域名就是播流域名。

在使用直播服务前必须先添加至少一个推流域名和一个播流域名，并进行关联。一个域名只能设置为一种域名，不能同时添加为推流域名和播流域名。

使用限制

每个阿里云账户下，最多支持加速20个域名。如果您的域名的总带宽日均峰值大于50 Mbps，且业务无风险，则可[提交工单](#)申请增加域名个数。

添加域名

直播服务添加域名的流程如下：

1. 确认域名是否符合准入标准

视频直播依托于阿里云强大的CDN服务，为您提供视频分发加速服务。在您的域名接入阿里云视频直播服务前，可阅读[域名准入标准](#)快速了解阿里云CDN加速域名的接入条件和限制，防止因域名涉及违规内容而造成的损失。

2. 验证域名归属权

当您首次将一个新域名添加至阿里云视频直播时，需要您完成验证域名的归属权。验证通过后视为您拥有该域名。您在视频直播控制台再次添加该域名或子域名时，不需要再进行验证。阿里云为您提供DNS解析验证和文件验证两种方法。具体操作，请参见[验证域名归属权](#)。

3. 添加域名，支持通过[控制台](#)或[API](#)进行操作。

添加域名时需要配置**加速域名**、**直播中心**、**业务类型**和**CDN 加速区域**。

控制台参数	API参数	描述
加速域名	DomainName	<p>添加的域名。加速域名不允许重复添加，若已添加到其他云产品中，会出现域名已添加的提示。</p> <ul style="list-style-type: none"> 控制台：一般使用子域名，不支持泛域名。 API：支持泛域名，以英文句号（.）开头。
直播中心	Region	<p>直播中心所在地域。需要关联的推流域名和播流域名必须选择同一直播中心，初次配置后不可更改。</p> <p>关于如何选择直播中心，请参见如何选择直播中心？。</p>
业务类型	LiveDomainType	<p>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。</p>
CDN 加速区域	Scope	<p>指域名可执行加速的地域。</p> <ul style="list-style-type: none"> 中国内地、domestic： <p>当主播在中国内地时，推流域名应选择中国内地。</p> <p>当观众在中国内地时，播流域名应选择中国内地。</p> <p>域名需要在工信部备案。</p> 全球加速、global： <p>当主播在中国内地和海外及港澳台时，推流域名应选择全球加速。</p> <p>当观众在中国内地和海外及港澳台时，播流域名应选择全球加速。</p> <p>域名需要在工信部备案。</p> 海外及港澳台加速、overseas： <p>当主播在海外及港澳台时，推流域名应选择海外及港澳台加速。</p> <p>当观众在海外及港澳台时，播流域名应选择海外及港澳台加速。</p> <p>域名无需在工信部备案。</p> <div style="background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p>? 说明</p> <ul style="list-style-type: none"> 选择不同的加速区域覆盖，对应不同的直播CDN带宽或流量单价。以上只在不同区分有用户播放请求时才发生对应区域的分发加速，并产生计费。如果您选择了全球加速，但是仅仅中国内地用户播放，不会发生海外区域的分发加速计费。更多计费规则，请参见直播服务费用和云产品定价页。 </div>

4. 域名解析

域名添加成功后，如果您想启用视频直播加速服务，需要将您的域名指向阿里云视频直播分配的CNAME地址，这样访问加速域名的请求才能转发到视频直播节点上，达到加速效果。更多信息，请参见[配置CNAME域名解析](#)。

说明

- 如果您是初次接触域名解析，对“什么是域名解析”、“为什么要解析域名”、“如何进行域名解析”、“什么是A记录”、“什么是CNAME记录”、“CNAME记录与A记录的差别”有疑问，请参见[什么是域名解析?](#)
- 添加域名后，您需要分别对推流域名和播流域名进行CNAME解析才可使用域名。
- 由于Local DNS的解析记录存在缓存时间，因此配置了域名的CNAME解析记录之后CDN平台大约会延迟10分钟才会显示CNAME解析记录配置成功。

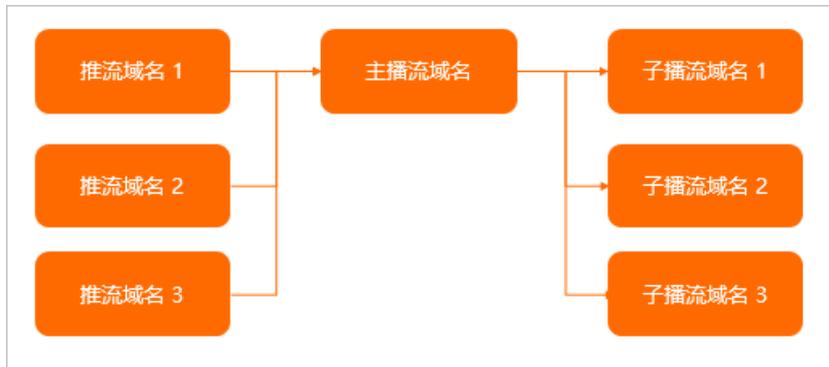
关联域名

在添加推流域名和播流后，您需要关联域名，才能使用视频直播服务。

视频直播采用边缘推流方式推送直播流，可以优先将视频推流至最优CDN节点，保证您访问的都是最佳的上行网络。因此需要将推流域名和播流域名进行映射，即关联推流域名和播流域名。

说明 视频直播已在逐步下线旧有的中心推流方式，2019年2月21日之后新增的播流域名，不再支持中心推流，都需要关联推流域名和播流域名。

视频直播支持关联多个推流域名与多个播流域名，形成的关联关系如下图所示。可通过关联推流和播流域名和关联主、子播流域名来实现。



- 关联推流和播流域名：**实现了一个播流域名对应多个推流域名。
一个推流域名只能配置一个主播流域名，若需要使用多个播流域名播放同一个推流，请把多个播流域名关联成主播流域名和子播流域名的关系。
您可以通过[控制台](#)，或调用AddLiveDomainMapping接口。
- 关联主、子播流域名：**实现了一个推流域名对应多个播流域名。
子播流域名会继承主播流域名的推流配置、转码配置，子播流域名配置这些信息无效，例如转码模板需要配置在主播流域名。更多功能配置是否生效，请参见[直播域名功能配置生效表](#)。
您可以通过[控制台](#)，或调用AddLiveDomainPlayMapping接口。

直播域名功能配置生效表

使用视频直播服务的都需要对直播域名进行配置，部分功能需要配置在主播流域名上，部分功能需要配置在子播流域名上。

功能项		主播流域名生效	子播流域名生效
直播流管理	查看直播流（在线流、历史流、禁推流）	√	-
	设置禁推流	√	-
文件管理	查询录制文件	√	-
	录制索引剪辑	√	-
	查询截图文件	√	-
域名管理	修改加速区域	-	√
	添加、删除推流信息	√	-
	添加子播放域名	√	子播流域名间不允许嵌套
	HLS回源HOST	√	子域名回源HOST头需配置为主域名
	延迟配置（高、中、低）	-	√
	HTTP头配置	-	√
	安全配置（HTTPS配置、Refer防盗链、URL鉴权、IP黑白名单）	-	√
	带宽峰值监控	-	√
	IPv6配置	-	√
直播处理配置	转码模板配置	√	-
	录制配置（录制模板配置、录制回调配置）	√	-
	截图配置（截图配置、截图回调配置）	√	-
	审核配置（审核配置、审核回调配置）	√	-
	直播时移	√	-
	直播延时配置	√	-
	拉流配置	√	-
	资源监控-流量带宽下行	-	√
	资源监控-回源统计	-	√

功能项 数据监控		主播流域名生效	子播流域名生效
	资源监控-HTTPCODE	-	√
	实时监控-流量带宽	-	√
	实时监控-质量监控	-	√
访问统计	独立访客数	-	√
	用户分布	-	√
用量查询	播放带宽/流量	-	√
	推流路数	√	-
	转码时长	√	-
	截图张数	√	-
	直播时移	√	-
日志管理	日志下载	-	√
	实时日志推送	-	√

3.3. 推流地址和播放地址

阿里云视频直播提供触发式推流与播放，您无需提前创建资源，只要添加经过备案的推流域名和播流域名并完成域名解析、鉴权等操作，即可快速拼接生成推流地址和播放地址。本文介绍推流地址和播放地址的拼接规则及生成方法。

使用限制

推流地址和播放地址可以根据规则进行多个创建，并同时执行直播活动。进行直播活动时需要注意直播每个域名对并发流是有限制的，详细限制信息请参见[使用限制](#)，进行直播活动前先确认当前限制是否满足您的需求。如不满足，您可[提交工单](#)。

生成推流地址和播放地址

您可以通过控制台或手动拼接生成推流地址和播放地址。

- 通过控制台生成，请参见[地址生成器](#)。
- 通过手动拼接生成推流地址和播放地址时，在拼接前需要自行判断直播流是否经过转码，是否进行鉴权，并准备好直播流的推流域名、播流域名、AppName（应用）、StreamName（直播流）、转码模板ID（如有）和鉴权串（如有）。拼接规则如下所示。

地址类	拼接规则	地址示例
-----	------	------

地址类	拼接规则	地址示例
推流地址	推流域名+AppName (应用) +StreamName (直播流) +鉴权串	<p>直播支持RTMP、RTS格式推流。RTS格式即超低延时直播RTS使用地址，需提前开通RTS服务。</p> <ul style="list-style-type: none"> RTMP: <code>rtmp://demo.aliyundoc.com/app/stream?auth_key={鉴权串}</code> RTS: <code>artc://demo.aliyundoc.com/app/stream?auth_key={鉴权串}</code> SRT: <code>srt://demo.aliyundoc.com:1105?streamid=#!::h=demo.aliyundoc.com,r=/app/stream?auth_key={鉴权串},m=publish</code>
播放地址	播流域名+AppName (应用) +StreamName (直播流) +鉴权串	<p>播放地址支持RTMP、FLV、M3U8、RTS格式。RTS格式即超低延时直播RTS使用地址，需提前开通RTS服务。</p> <ul style="list-style-type: none"> RTMP: <code>rtmp://example.aliyundoc.com/app/stream?auth_key={鉴权串}</code> FLV: <code>http://example.aliyundoc.com/app/stream.flv?auth_key={鉴权串}</code> M3U8: <code>http://example.aliyundoc.com/app/stream.m3u8?auth_key={鉴权串}</code> RTS: <code>artc://example.aliyundoc.com/app/stream?auth_key={鉴权串}</code>
转码流地址	播流域名+AppName (应用) +StreamName (直播流)_转码模板ID+鉴权串	<p>播放地址支持RTMP、FLV、M3U8、RTS格式，RTS格式即超低延时直播RTS使用地址，需提前开通RTS服务。不同的转码需要在 StreamName 后加 _模板ID。</p> <ul style="list-style-type: none"> RTMP: <code>rtmp://example.aliyundoc.com/app/stream_{转码模板ID}?auth_key={鉴权串}</code> FLV: <code>http://example.aliyundoc.com/app/stream_{转码模板ID}.flv?auth_key={鉴权串}</code> M3U8: <code>http://example.aliyundoc.com/app/stream_{转码模板ID}.m3u8?auth_key={鉴权串}</code> RTS: <code>artc://example.aliyundoc.com/app/stream_{转码模板ID}?auth_key={鉴权串}</code>

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。
- {}内容实际使用时, 需要将{}数据替换为实际数据。
- 如果是纯拉流场景, 您可以不需要生成推流域名。
- 鉴权串是根据鉴权算法得出的一个加密字符串, 需开启鉴权功能才可以生成。您可以在控制台的开启鉴权功能并获取鉴权串, 请参见[配置URL鉴权](#), 也可通过md5算法计算获取鉴权串, 请参见[鉴权URL组成](#)。
- 转码流地址需设置转码模板才可以生成。您可在控制台添加或查看转码模板的ID, 具体操作, 请参见[通用转码](#), 也可以通过调用AddCustomLiveStreamTranscode或DescribeLiveStreamTranscodeInfo接口添加或获取转码模板ID。

3.4. 直播推流

阿里云视频直播支持多种方式推流。通过阅读本文, 您可以了解不同方式的推流方法。

前提条件

您已经生成推流地址。具体操作, 请参见[推流地址](#)和[播放地址](#)。

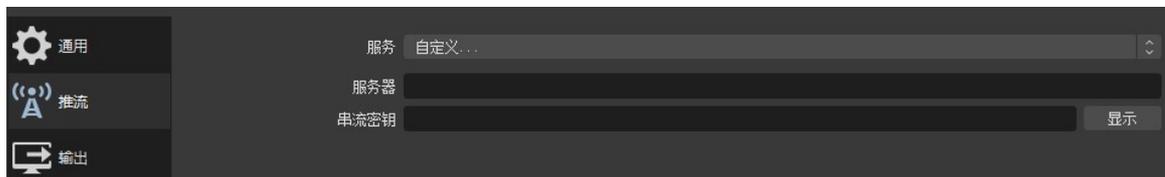
注意事项

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)。

使用OBS推流

OBS推流工具支持Windows、Mac和Linux平台。本文以Windows端OBS v27.2.3版本为例介绍使用OBS推流, 其他平台及版本操作类似。关于OBS更多使用信息, 请参见[OBS推流工具的使用方法](#)。

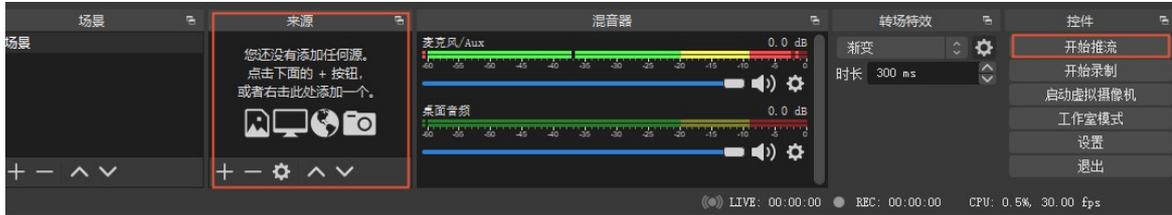
1. 下载并安装OBS推流工具。下载地址, 请参见[OBS Studio](#)。
2. 运行OBS推流工具。
3. 在菜单栏中选择文件 > 设置。
4. 在设置页面选择推流, 配置信息如下所示:



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: <code>rtmp://demo.aliyundoc.com/app/stream?auth_key=1543302081-0-0-9c6e7c8190c10bdfb3c0*****</code> 。

参数	描述
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。



使用阿里云直播Demo推流

1. 使用移动端扫描并安装阿里云直播Demo。



说明 iOS端扫码安装时如果提示未受信任的企业级开发者，需要在设置 > 通用 > 设备管理中 找到Taobao对应的信任描述，并选择信任。

- 2. 运行阿里云直播Demo。
- 3. 选择摄像头推流或录屏推流。

说明 移动端进行本地视频推流时仅支持视频图像原有yuv格式。

- 4. 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*****`），并按照实际需求完成其他参数的配置。
- 5. 单击开始推流，开始移动端推流。

使用直播推流SDK推流

iOS推流SDK

- 1. 集成iOS端推流SDK。具体操作，请参见[SDK集成](#)。
- 2. 根据实际需求配置推流参数并推流。具体操作，请参见[功能使用](#)。
在配置推流参数或推流时可能会出现异常情况，请参见[异常及特殊场景处理](#)。

Android推流SDK

- 1. 集成Android端推流SDK。具体操作，请参见[SDK集成](#)。

2. 根据实际需求配置推流参数并推流。具体操作，请参见[功能使用](#)。
在配置推流参数或推流时可能会出现异常情况，请参见[异常及特殊场景处理](#)。

后续步骤

- 进入[流管理](#)，在在线流中查看并管理已推的音视频流。更多信息，请参见[流管理](#)。
- 执行完直播推流之后，您可以进行直播播放。具体操作，请参见[直播播放](#)。

3.5. 直播播放

阿里云视频直播支持多种方式播放。通过阅读本文，您可以了解不同方式的播放方法。

前提条件

- 您已经完成直播推流。具体操作，请参见[直播推流](#)。
- 您已经生成播放地址。具体操作，请参见[推流地址和播放地址](#)。

通过VLC播放器播放

VLC播放器支持Windows、Mac、Linux、Android和iOS平台。本文以Windows端VLC v3.0.16版本为例介绍使用VLC播放器播放直播流，其他平台及版本操作类似。关于VLC更多使用信息，请参见[VLC播放器](#)。

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**。

通过阿里云直播Demo播放

1. 使用移动端扫描并安装阿里云直播Demo。



 **说明** iOS端扫码安装时如果提示未受信任的企业级开发者，需要在**设置 > 通用 > 设备管理**中找到Taobao对应的信任描述，并选择信任。

2. 运行阿里云直播Demo。
3. 选择**直播播放**。

4. 输入已经生成的播放地址，例如：`rtmp://example.aliyundoc.com/app/stream?auth_key=1543300311-0-0-d47ce016332bf280cf275*****`。

5. 单击拉流，开始移动端播放。

通过播放器SDK播放

播放器SDK分为直播播放器和阿里云播放器，各播放器SDK对应的播放流程如下所示：

播放器SDK		播放流程
直播播放器SDK	Android端直播播放器SDK	<ol style="list-style-type: none"> 1. 集成Android端直播播放器SDK。具体操作，请参见集成Android端直播播放器。 2. 根据实际需求完成播放器的开发并进行播放。具体操作，请参见实现Android端播放功能。 <p>更多Android端接口说明，请参见接口说明。</p>
	iOS端直播播放器SDK	<ol style="list-style-type: none"> 1. 集成iOS端直播播放器SDK。具体操作，请参见集成iOS端直播播放器。 2. 根据实际需求完成播放器的开发并进行播放。具体操作，请参见实现iOS端播放功能。 <p>更多iOS端接口说明，请参见接口说明。</p>
阿里云播放器SDK	Android端阿里云播放器SDK	<ol style="list-style-type: none"> 1. 集成Android端阿里云播放器SDK。具体操作，请参见快速集成。 2. 根据实际需求完成播放器的开发并进行播放。具体操作，请参见基础功能。 <p>更多Android端高阶功能，请参见进阶功能；接口说明，请参见接口说明。</p>
	iOS端阿里云播放器SDK	<ol style="list-style-type: none"> 1. 集成iOS端阿里云播放器SDK。具体操作，请参见快速集成。 2. 根据实际需求完成播放器的开发并进行播放。具体操作，请参见基础功能。 <p>更多iOS端高阶功能，请参见进阶功能；接口说明，请参见接口说明。</p>
	Windows端阿里云播放器SDK	<ol style="list-style-type: none"> 1. 集成Windows端阿里云播放器SDK。具体操作，请参见快速集成。 2. 根据实际需求完成播放器的开发并进行播放。具体操作，请参见常用功能。 <p>更多Windows端接口说明，请参见接口说明。</p>

播放器SDK	播放流程
Flutter框架阿里云播放器SDK	<ol style="list-style-type: none"> 1. 集成Flutter框架阿里云播放器SDK。具体操作，请参见快速集成。 2. 根据实际需求完成播放器的开发并进行播放。具体操作，请参见常用功能。
Web端阿里云播放器SDK	<ol style="list-style-type: none"> 1. 集成Web端阿里云播放器SDK。具体操作，请参见快速接入。 2. 根据实际需求完成播放器的开发并进行播放。具体操作，请参见基础功能。 <p>更多Web端高阶功能，请参见进阶功能；接口说明，请参见接口说明。</p>

相关文档

[视频直播生成推流地址后无法播放视频](#)

3.6. 直播基础服务FAQ

边缘推流和直播中心有什么关系？

阿里云视频直播在服务端主要提供直播流接入、分发、实时流媒体处理服务。接入和分发网络（直播CDN）使用边缘就近接入方案，上行推流会根据直播发生地点使用最近最优匹配的节点进行接入。

在北京、上海、深圳、青岛、日本、新加坡、印度、印尼、德国9个国家和地区部署了直播中心。直播中心主要提供直播流的录制、截图、转码、时移、视频AI、导播台等实时流媒体服务。在开始直播前，需要您配置一对直播推流、播流域名（并进行绑定）。创建推流和播流域名时，会让您选择域名使用的直播中心，这个直播中心就是实时流媒体处理所使用的区域，与直播流接入和分发位置无关。

如何选择直播中心？

- 如果您的直播流业务推流和播放都在中国内地，且直播流不需要对实时流进行媒体处理（截图、录制、导播台服务等）：

添加域名时选择中国内地哪个区域没有区别。不是说您的主播多在北京，您就一定要选择北京区域，因为即使您域名选择了上海区域，主播在北京，边缘接入会使用北京的接入节点进行接入。分发节点会通过您的观众在哪里进行调度，和实时流媒体处理中心所在的区域无关。

- 如果您的业务需要对实时流进行媒体处理（截图、录制、导播台服务等）：

需要考虑以上服务会涉及图片、视频的存储，视频存储的OSS Bucket必须选择和直播中心同区域（避免跨区域即region，网络等因素可能影响实时流处理的及时性和稳定性）。所以您的存储如果已经在某个区域，对应的直播中心需要选择在同区域（region）。

- 如果您的业务直播主播和观众都在海外：

建议您选择距离主播较近的一个直播中心，将视频流推至附近的直播中心。

- 边缘推流

阿里云的CDN节点遍布全球主流国家，使用边缘推流，将视频流优先推送到距主播最近的边缘节点上，再通过阿里优化后的网络将视频流传输至直播中心。

○ 实时监控网络状态

为实时观察主播的网络状态，您可以登录[视频直播控制台](#)，在直播管理 > 流管理 > 流监控 页面中查看实时的推流状态、检测主播的网络状态，保证推流的稳定性，请参见[流管理](#)。

您也可以通过API自主获取实时帧率和码率，请参见[查询推流域名流帧率和码率](#)。

如何选择域名的CDN加速区域？

创建域名时需要您选择CDN加速区域类型，CDN加速区域是指仅中国内地加速或者海外及港澳台加速。其中全球加速是指中国内地和海外及港澳台加速分别加速，并不是指中国内地和海外及港澳台之间的加速。

- 如果您的直播业务主播和观众都在中国内地，创建推流和播流域名时直播中心都选择中国内地区域，CDN加速区域选择中国内地。
- 如果您的直播业务主播和观众都在海外及港澳台（如新加坡），则推流和播流域名的直播中心都选择新加坡。CDN加速区域选择海外及港澳台。
- 海外直播中心选择使用就近原则，尽量选择主播和观众所在地附近的区域。如果您的区域无对应的直播中心，可以[提交工单](#)咨询，选择哪个中心更好。
- 如果您的直播业务有主播和观众在不同区域（在中国内地、海外及港澳台或都有），创建播流域名时候CDN加速区域要选择全球加速（包括中国内地和海外及港澳台加速）或者海外及港澳台加速（不包括中国内地）。

 说明

- 选择不同的加速区域覆盖，对应不同的直播CDN带宽或流量单价。以上只在不同区分有用户播放请求时才发生对应区域的分发加速，并产生计费。如果您选择了全球加速，但是仅仅中国内地用户播放，不会发生海外区域的分发加速计费。更多计费规则，请参见[直播服务费用](#)和[云产品定价页](#)。

选择直播中心场景枚举

主播所在地	观众所在地	域名直播中心选择与播流域名CDN加速配置	说明
中国内地	中国内地	<ul style="list-style-type: none"> ● 不使用导播台：可以选择北京、上海、深圳、青岛直播中心。 ● 如果需要使用导播台，目前只能选择上海。 ● 如果有录制、截图等使用OSS场景，OSS Bucket必须和直播中心同区域。 	无
中国内地	海外及港澳台	<ul style="list-style-type: none"> ● 推流域名和播流域名可以选择中国内地的4个直播中心。 ● 播流域名的CDN加速选择海外及港澳台加速。 	主播都在中国内地，但观众在海外及港澳台。

主播所在地	观众所在地	域名直播中心选择与播流域名CDN加速配置	说明
中国内地	新加坡	方案一： <ul style="list-style-type: none"> 推流域名和播流域名中国内地的4个直播中心。 播流域名CDN加速选择海外及港澳台加速。 方案二： <ul style="list-style-type: none"> 推流域名和播流域名选择新加坡直播中心。 播流域名CDN加速选择海外及港澳台加速。 	解决goChina跨境直播的需求。 新加坡和中国内地线路支持双向配置，所以无论客户域名使用新加坡中心还是中国内地的中心，都支持主播从中国内地进行推流，播放观众大多在新加坡的场景。
新加坡	中国内地	方案一： <ul style="list-style-type: none"> 推流域名和播流域名中国内地的4个直播中心。 播流域名CDN加速选择海外及港澳台加速。 方案二： <ul style="list-style-type: none"> 推流域名和播流域名选择新加坡直播中心。 播流域名CDN加速选择海外及港澳台加速。 	解决goChina跨境直播的需求。 新加坡和中国内地线路支持双向配置，所以无论客户域名使用新加坡中心还是中国内地的中心，都支持主播从新加坡进行推流，播放观众大多在中国内地的场景。 若为了兼顾可能仍有少量在海外播放的用户，播流域名CDN加速可选择 全球加速 。
中国内地	中国内地+海外及港澳台	<ul style="list-style-type: none"> 推流域名和播流域名可以选择中国内地的4个直播中心。 播流域名CDN加速选择全球加速。 	主播都在中国内地，观众在中国内地、海外及港澳台都有。
海外及港澳台	海外及港澳台	<ul style="list-style-type: none"> 推流域名和播流域名可以选择日本、新加坡、印尼、印度、德国直播中心。 播流域名CDN加速选择海外及港澳台加速。 	无
海外及港澳台	中国内地	<ul style="list-style-type: none"> 推流域名和播流域名可以选择中国内地的4个直播中心。 播流域名CDN加速选择全球加速。 	其中主播在海外，观众在中国内地的情况。 播流域名CDN加速选择 全球加速 是为了兼顾可能仍有少量在海外播放的用户。

为什么会出现播放卡顿的情况？

播放的卡顿，很大一部分原因是推流时出现了卡顿。导致推流的卡顿有以下主要因素，当出现推流卡顿时，您可以逐一排查问题。

- 采集设备配置

推流过程中会有一定的CPU损耗，硬件配置较差的低端手机，在推流过程中若整体CPU使用率超过80%，画面会出现不同程度的卡顿，花屏等现象，会影响到视频的采集，同时也会影响到用户端的观看。

- 视频采集参数的设置

能保证人眼识别流畅度的视频最低保证在1秒15帧以上，如果FPS低于1秒10帧，画面就会出现较明显的卡顿，如无特殊情况，尽量将视频帧率设置在1秒15帧之上。

虽然视频的帧率越高画面流畅感越强，但是帧率超过1秒30帧后，人眼就无法识别出画面的效果，帧率增加也增加视频传输的带宽成本，建议您合理设置视频的参数。

- 网络带宽

大多数画面出现卡顿的现象均来自网络因素，以下列出几条常见因素：

- 网络带宽大小：确认网络运营商提供给您的带宽大小，是否足够本次直播传输。
- 下行带宽占用：检查是否有下载数据占用网络带宽。
- 系统资源占用：检查后台是否运行了大量的程序，请您合理的删除和停止正在运行的程序，空出资源。

4.直播加速

4.1. 边缘推流

视频直播采用边缘推流的方式推送直播流，优先将视频推流至最优CDN节点。您无需进行其他配置，2019年2月21日之后新增的播流域名均采用边缘推流方式进行分发。本文介绍边缘推流、边缘推流的优势及相关操作。

- 边缘推流

利用阿里云丰富的CDN节点、覆盖范围广的优势，优先将流数据调度至距离用户最近的最优CDN节点，通过阿里智能调度系统将数据快速传输至直播中心进行内容分发，保证用户访问的都是最佳的上行网络，减少因上行传输带来的卡顿、拉流缓慢的问题。



- 中心推流（已逐步下线）

中心推流方式已在逐步下线，2019年2月21日之后新增的播流域名，均采用边缘推流的接入方式。如果您因为历史原因还在使用中心推流方式，希望您尽快[提交工单](#)联系我们，切换为边缘推流，否则很多新功能无法支持。

边缘推流优势

- 高覆盖：全球有2800+CDN节点。其中，海外有500+，中国内地有2300+，覆盖主流城市和地区。
- 智能化：优先接入距离用户最近的CDN节点，保证内容传输的稳定。

相关操作

2019年2月21日之后新增的播放域名默认均采用边缘推流方式，无需配置，您只需要关联推流域名和播流域名即可。

- 更多信息，请参见[关联域名](#)。
- 具体操作，请参见[关联播流域名和推流域名](#)。

5.直播安全

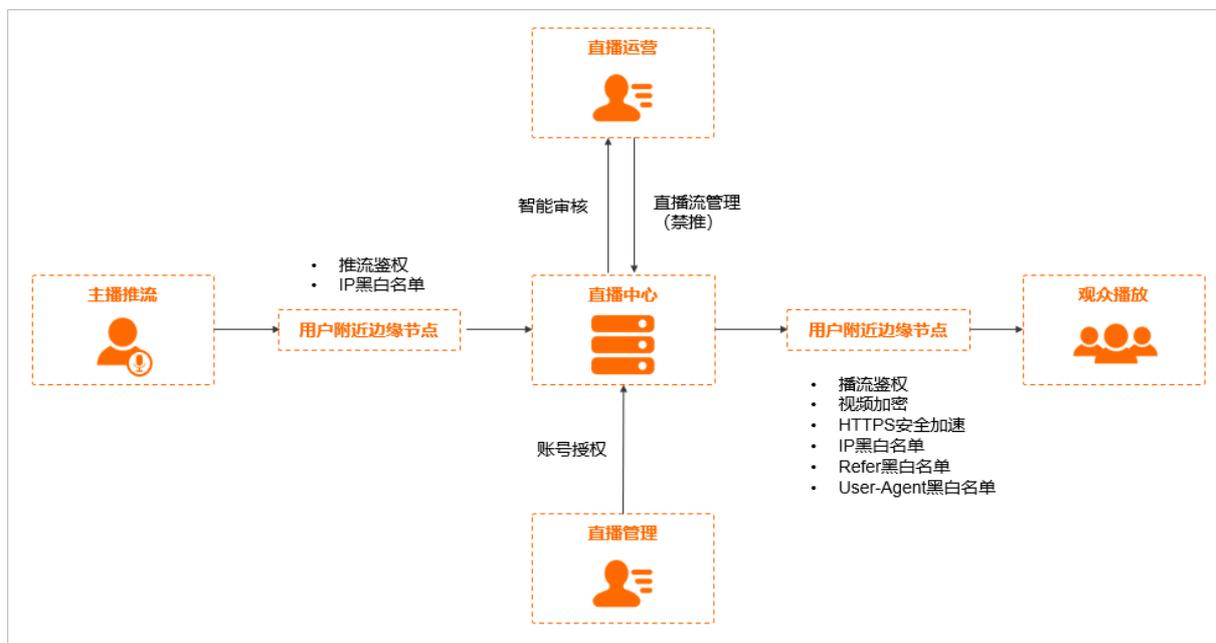
5.1.直播安全概述

视频直播提供了完善的内容安全保护机制，您可以根据不同业务场景的安全需求，选择RAM账号、HTTPS安全加速、访问控制等服务。本文介绍如何保护直播安全。

如何保护直播安全

为保证直播服务配置、直播内容生产、推送和播放全过程的安全性，视频直播提供了完善的内容安全保护机制，可用于保障直播内容不被盗链、非法下载和传播，可以满足不同业务场景的安全需求。

由下图我们可以了解到视频直播丰富的安全手段。



直播安全简介

视频直播提供如下安全机制：

安全机制	安全手段	特点	安全等级	使用门槛
账号授权	RAM用户	根据授权策略对RAM用户可授予相应的权限。	较低	低，仅需云端配置
安全加速	HTTPS安全加速	HTTPS协议是以安全为目标的HTTP通道，将HTTP用SSL/TLS协议进行封装。	高	低，仅需云端配置
	User-Agent黑白名单	基于HTTP Header跟踪来源，但极易伪造。	低	低，仅需云端配置

安全机制	安全手段	特点	安全等级	使用门槛
访问控制	Refer防盗链	基于HTTP Header跟踪来源，但极易伪造。	低	低，仅需云端配置
	IP黑白名单	拒绝或只允许特定IP访问，不适合大量C端用户的分发。	较低	低，仅需云端配置
推流播流URL鉴权	推流和播放的URL鉴权	支持自定义的鉴权Key和鉴权失效时间，动态生成鉴权URL。	中	较低，提供生成鉴权脚本
业务方远程鉴权	透传业务请求信息给客户自定义鉴权中心来判断合法性	客户添加自定义的业务请求信息，通过自建鉴权中心，更加精准识别合法请求。	高	较高，需部署鉴权中心，并确保高可用
视频安全	阿里云视频加密	云端一体的视频加密解决方案，采用私有加密算法对视频流加密，确保视频流安全传输。	高	较低，简单配置并集成阿里云播放器即可
	DRM加密	苹果Fairplay、谷歌Widevine的原生支持，安全级别很高，满足大的版权内容提供商的要求。	高	较高，按调用license次数进行收费，只需集成阿里云播放器SDK
内容安全	智能审核	对直播流的视频和音频进行审核，自动检测视频和语音内容是否涉嫌违规。	高	低，仅需云端配置
	禁推流	支持直播流进行禁止推流的处理，可自定义禁推时间。	高	低，仅需云端配置

账号授权

背景：由于阿里云账号AK有完全的权限，一旦泄露风险巨大。

介绍：视频直播会对每一次发起操作请求的用户身份进行验证，通过AccessKey验证该账号是否拥有相应的权限。视频直播支持对账号进行验证，并提供了系统授权策略，您还可以自定义授权策略。更多信息，请参见[权限管理概述](#)。

安全加速

背景：HTTP协议以明文方式发送内容，不提供任何方式的数据加密。

介绍：安全超文本传输协议（Hyper Text Transfer Protocol over Secure Socket Layer，简称 HTTPS），是以安全为目标的HTTP通道。简单来说，HTTPS是HTTP的安全版，即将HTTP用SSL/TLS协议进行封装，HTTPS的安全基础是SSL/TLS。更多信息，请参见[安全加速](#)。

访问控制

在云端配置视频直播的访问策略，可以达到基本保护。

主要手段有：

- Refer防盗链

基于HTTP协议支持的Referer机制，通过Referer跟踪来源。可配置拒绝访问的Referer黑名单，或仅允许访问的白名单。

- User-Agent黑白名单

基于HTTP协议支持的User-Agent请求头跟踪来源，可配置拒绝访问的User-Agent黑名单，或仅允许访问的白名单。

- IP黑白名单

您可以通过配置IP黑白名单来实现对访客身份的识别和过滤，从而限制访问视频直播资源的用户，提升Live的安全性。

更多信息，请参见[访问控制](#)。

推流播流URL鉴权

背景：固定不变的播放地址会导致视频内容被持久的非法扩散传播，且无法有效遏制。

介绍：URL鉴权通过生成动态的加密URL（包含权限验证、过期时效等信息）来区分合法请求，以达到保护视频资源的目的。

开启URL鉴权后：

- 直播推流URL和直播播流URL都会进行鉴权。
- 视频直播的播放器SDK、获取播放地址的API/SDK都会自动生成带时效的播放URL。如需要自己生成鉴权的动态URL，则可参见[URL鉴权](#)中的鉴权方法。

更多信息，请参见[URL鉴权](#)。

业务方远程鉴权

背景：目前播放中心鉴权对盗链等非法请求判断较为单一，业务方远程鉴权可以引入客户业务请求的问题，鉴权更加精准。

介绍：业务方远程鉴权是指直播CDN将用户的请求透传到客户的鉴权中心，由客户自己判定该请求是否合法，CDN根据客户的判断结果执行相应动作：允许或拒绝访问。

- 远程鉴权需要客户自己开发和部署鉴权中心，该鉴权中心的域名如果同时在CDN上加速，可以按照一定规则缓存客户的鉴权结果，以减轻客户鉴权中心的压力。
- 直播CDN会默认把用户请求的headers和request_uri透传到客户自定义的鉴权中心，并根据鉴权中心返回的结果执行相应的动作。
- 业务方可将其用户的登录Cookie或UUID等信息隐藏于播放请求中，进而透传到自己的鉴权中心以判定是否为合法用户。

 **说明** 远程鉴权使用门槛较高，需要客户自己开发和部署鉴权中心，如有需求可[提交工单](#)联系售后进行开通和配置。

视频安全

背景：防盗链安全机制能有效保障用户的合法访问，但对于付费观看直播的场景，用户只需通过一次付费行为便可拿到视频合法的防盗链播放URL，将视频下载到本地，进而实现二次分发。因此，防盗链方案对于视频版权保护是远远不够的。视频文件一旦泄露，会给付费观看模式造成严重的经济损失。

介绍：阿里云视频加密是对视频数据加密，即使下载到本地，视频本身也是被加密的，无法二次分发，可有效防止视频泄露和盗链问题。

• 阿里云视频加密

阿里云视频加密采用私有的加密算法和安全传输机制，提供云端一体的视频安全方案，核心部分包括加密转码和解密播放。

核心优势：

- 每个媒体文件拥有独立的加密密钥，能有效避免采用单一密钥时，一个密钥的泄露引起大范围的安全问题。
- 提供信封加密机制密文Key+明文Key，仅密文Key入库，明文Key不落存储，所有过程只在内存中，用完即销毁。
- 提供安全的播放器内核SDK，涵盖iOS、Android、H5、Flash等多平台，自动对加密内容进行解密播放。
- 播放器和云端使用私有加密协议进行密文传输，不传输明文Key，有效防止密钥被窃取。
- 提供安全下载，缓存到本地的视频会再次加密，在确保无网离线播放前提下，防止视频被拷贝窃取。

 **注意** 阿里云视频加密使用限制如下：

- 仅支持输出HLS格式。
- 只能使用阿里云播放器。
- 暂不支持网页端播放。

更多信息，请参见[阿里云视频加密](#)。

• DRM加密

高端的直播节目，需要满足内容版权和提供商的安全要求，如体育赛事、演唱会等。阿里视频云推出的云端DRM解决方案，支持Fairplay和Widevine加密，提供了一站式的视频加密、license下发、播放等能力。

更多信息，请参见[DRM加密](#)。

视频加密方案各有优劣。一般来说，越是标准、通用，灵活性越高，但安全性越低。您需要根据业务场景，有所取舍，选择适合的方案。

内容安全

背景：直播流是由直播的主播生产的，通过直播平台推送给观众，如果不对直播内容进行审核，可能导致有害信息传播，存在内容违规风险。

介绍：阿里云视频直播基于强大的视频AI能力，提供了智能审核功能，支持视频、音频、图片等全方位的媒体审核。并支持禁止推送违规的直播流。

- **智能审核：**基于海量标注数据和深度学习算法实现，从语音、视觉等多维度精准识别视频、封面、标题等媒体信息中包含的违禁内容。
- **禁推直播流：**针对于推流内容不合规或希望对推流进行封禁的场景，视频直播提供了直播流管理的入口，支持对直播流进行永久禁推和限时禁推。

更多信息，请参见[内容安全](#)。

5.2. 安全加速

视频直播支持HTTPS安全加速、强制跳转，本文介绍安全加速的原理、优势、使用方法和注意事项。

背景

HTTP协议以明文方式发送内容，不提供任何方式的数据加密。在安全性上，HTTP明文数据很容易在传输过程中的节点设备被截获，由于数据没有加密，很容易理解其含义。

功能简介

视频直播支持配置HTTPS安全加速以及强制跳转，使用强制跳转需要在已经完成HTTPS证书配置的前提下。

- HTTPS:

安全超文本传输协议（Hyper Text Transfer Protocol over Secure Socket Layer，简称 HTTPS），是以安全为目标的HTTP通道。简单来说，HTTP协议以明文方式发送内容，而HTTPS是HTTP的安全版，将HTTP用SSL/TLS协议进行封装，HTTPS的安全基础是SSL/TLS。

HTTPS提供了身份验证与加密通讯方法，被广泛用于万维网上安全敏感的通讯，根据2017年EFF（Electronic Frontier Foundation）发布的报告，目前全球已有超过一半的网页端流量采用了加密的HTTPS进行传输。

- 强制跳转:

将客户端至边缘节点的原请求方式强制重定向为HTTP或者HTTPS。

如果您的加速域名开启了HTTPS安全加速，您可以自定义设置，将终端用户的原请求方式进行强制跳转。例如，您开启HTTP -> HTTPS后，终端用户发起了一个HTTP请求，服务端返回301重定向响应，原来的HTTP请求强制重定向为HTTPS请求，如下图所示。

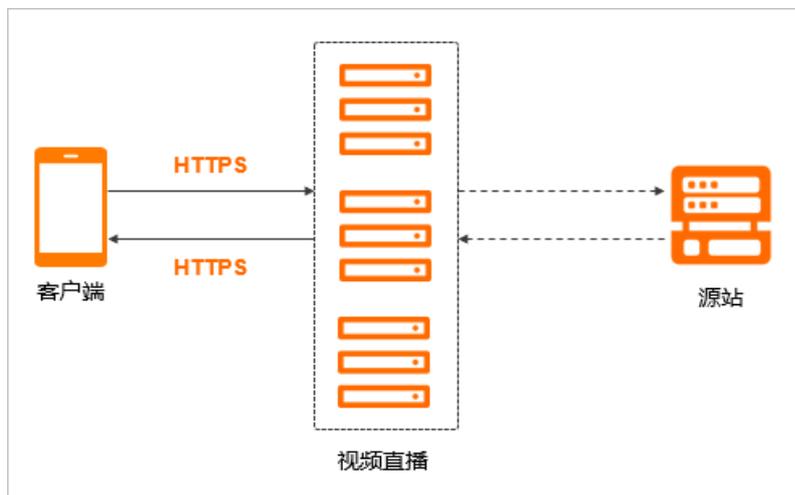
```
$ curl http://... -i
HTTP/1.1 301 Moved Permanently
Server: Tengine
Date: Mon, 03 Jun 2019 13:26:01 GMT
Content-Type: text/html
Content-Length: 278
Connection: keep-alive
Location: https://.../
Via: cache2.cn201[,0]
Timing-Allow-Origin: *
EagleId: 2a786b0215595683612635433e

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
<head><title>301 Moved Permanently</title></head>
<body bgcolor="white">
<h1>301 Moved Permanently</h1>
<p>The requested resource has been assigned a new permanent URI.</p>
<hr/>Powered by Tengine</body>
</html>
```

功能原理

在阿里云视频直播控制台开启的HTTPS协议，将实现客户端和阿里云视频直播节点之间请求的HTTPS加密。视频直播节点返回从源站获取的资源给客户端时，按照源站的配置方式进行。建议源站配置并开启HTTPS，实现全链路的HTTPS加密。

HTTPS加密流程如下图所示。



1. 客户端发起HTTPS请求。
2. 服务端生成公钥和私钥（可以自己制作，也可以向专业组织申请）。
3. 服务端把相应的公钥证书传送给客户端。
4. 客户端解析证书的正确性。
 - 如果证书正确，则会生成一个随机数（密钥），并用公钥随机数进行加密，传输给服务端。
 - 如果证书不正确，则SSL握手失败。

说明 正确性包括：

- 证书未过期；
- 发行服务器证书的CA可靠；
- 发行者证书的公钥能够正确解开服务器证书的发行者的数字签名；
- 服务器证书上的域名和服务器的实际域名相匹配。

5. 服务端用之前的私钥进行解密，得到随机数（密钥）。
6. 服务端用密钥对传输的数据进行加密。
7. 客户端用密钥对服务端的加密数据进行解密，拿到相应的数据。

功能优势

- 传输安全：HTTPS安全传输，有效防止HTTP明文传输中的窃听、篡改、冒充和劫持风险。
- 信息加密：数据传输过程中对您的关键信息进行加密，防止类似Session ID或者Cookie内容被攻击者捕获造成的敏感信息泄露等安全隐患。
- 数据完整：数据传输过程中对数据进行完整性校验，防止DNS或内容遭第三方劫持、篡改等中间人攻击（MITM）隐患，详情请参见[使用HTTPS防止流量劫持](#)。
- HTTPS是主流趋势：未来主流浏览器会将HTTP协议标识为不安全，谷歌浏览器Chrome 70以上版本以及Firefox已经在2018年将HTTP网站标识为不安全，若坚持使用HTTP协议，除了安全会埋下隐患外，终端客户在访问网站时出现的不安全标识，也将影响访问。

主流浏览器对HTTPS网站进行搜索加权，主流浏览器均支持HTTP/2，而支持HTTP/2必须支持HTTPS。无论从安全、市场或用户体验来看，普及HTTPS是未来的一个方向，所以强烈建议您将访问协议升级到HTTPS。

使用方法

视频直播支持通过控制台和API的方式配置HTTPS安全加速和强制跳转。

● 控制台：

功能	描述	参考文档
HTTPS安全加速	配置HTTPS证书。 您需要配置HTTPS证书，才能通过HTTPS方式访问资源，实现HTTPS安全加速。更多信息，请参见 证书格式说明 。	配置HTTPS安全加速
强制跳转	配置强制跳转类型。 配置强制跳转之前，请确保已成功配置HTTPS证书。	强制跳转

● API：

接口	描述	参考文档
SetLiveDomainCertificate	设置某域名下证书功能是否启用及修改证书信息。	设置域名证书
DescribeLiveCertificateList	获取证书列表信息。	查询证书列表
DescribeLiveCertificateDetail	获取证书详细信息。	查询证书信息
BatchDeleteLiveDomainConfigs	批量删除域名配置。	批量删除直播域名配置

注意事项

配置	说明
停用和启用HTTPS	<ul style="list-style-type: none"> 停用后，不支持HTTPS请求且将不再保留证书/私钥信息。 启用后，再次开启证书，需要重新上传证书/私钥。

配置	说明
上传证书/私钥	<ul style="list-style-type: none"> 开启HTTPS安全加速功能的加速域名，须上传证书，包含证书/私钥，均为PEM格式。 <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin: 5px 0;"> <p>? 说明 直播服务采用的Tengine服务是基于Nginx的，因此只支持Nginx能读取的证书，即PEM格式。</p> </div> <ul style="list-style-type: none"> 只支持带SN信息的SSL/TLS握手。 您上传的证书和私钥要匹配，否则会校验出错。 更新证书的生效时间是1小时。 不支持带密码的私钥。
查看证书	允许用户查看证书，但是只支持查看证书，由于私钥信息敏感不支持私钥查看，请您妥善保管证书相关信息。
修改编辑证书	支持修改编辑证书，但注意生效时间是1小时，请慎重操作。

5.3. 访问控制

视频直播支持通过设置Referer黑白名单、User-Agent黑白名单、IP黑白名单进行访问控制，本文介绍视频直播访问控制功能、适用场景和使用方式。

简介

访问控制是在云端配置视频资源的访问策略，达到基本的保护目的，具有使用门槛低（仅云端配置不需要额外开发）、快速生效等优点，主要手段有：

- Referer黑白名单
- User-Agent黑白名单
- IP黑白名单

? 说明 User-Agent黑白名单，由于配置繁琐且存在误操作风险，暂未开放控制台设置，如有需求可[提交工单](#)或联系阿里云售后处理。

适用场景

- 访问控制的使用门槛很低，只需简单配置即可使用，能起到一定的保护作用，特别是Web端。
- Referer和User-Agent都基于HTTP Header，容易被伪造，安全性低。
- IP黑白名单机制，无法实现内容分发给大量C端用户，不适合广泛的内容消费场景。

Referer黑白名单

- 介绍
 - 基于HTTP协议支持的Referer机制，通过Referer跟踪来源，对来源进行识别和判断，用户可配置访问的Referer黑、白名单来限制视频资源被访问的情况。
 - 访客对资源发起请求后，请求到达CDN节点，节点会根据用户预设的防盗链黑名单或白名单进行过滤：符合规则可顺利请求到视频数据；若不符合，请求会被拒绝，并返回403响应码。
 - Referer黑白名单是可选配置，默认不启用。

- 配置后会自动添加泛域名支持。例如填写 `example.com` ，最终配置生效的是 `*.example.com` ，所有子级域名都会生效。
- 支持设置是否允许空Referer字段访问资源，即允许通过浏览器地址栏直接访问资源URL。
- 使用方式
 - 控制台：具体操作请参见[防盗链](#)。
 - API：通过调用[BatchSetLiveDomainConfigs](#)接口，可批量配置直播域名，包括Referer黑白名单。
- 注意事项
 - 黑、白名单互斥，同一时间您只能选择一种方式。
 - 由于移动端一般拿不到Referer，当前默认支持空Referer访问，可选择关闭。如果设置为不允许空referer访问，在移动端可以配合阿里云播放器设置referer。
 - 当您设置不允许空Referer访问时，请务必配置HTTPS安全加速，并开启强制跳转HTTPS（HTTP>HTTPS）。部分浏览器处理HTTPS请求HTTP的资源时，会移除Referer，导致无法访问。
- 示例

当设置播流域名 `demo.developer.aliyundoc.com` 的Referer白名单为 `example.com` ，且不允许空Referer访问。

i. 请求数据：

```
curl -i 'http://demo.developer.aliyundoc.com/apptest/stream0000'
```

ii. 返回中包含：

```
X-Engine-Error:denied by Rererer ACL
```

iii. 当请求带上允许的Referer后即正常返回：

```
curl -i 'http://demo.developer.aliyundoc.com/apptest/stream0000' \  
-H 'Referer: http://www.example.com'
```

User-Agent黑白名单

- 介绍

User-Agent是一个特殊字符串头，帮助服务端识别用户使用的操作系统及版本、CPU类型、浏览器及版本、浏览器渲染引擎&语言和插件等。可通过User-Agent黑白名单来限制特定浏览器或终端的访问。
- 使用方式

[提交工单](#)或联系阿里云售后。
- 示例

○ 如PC端IE9浏览器：

```
User-Agent:Mozilla/5.0 (compatible;MSIE9.0;WindowsNT6.1;Trident/5.0;
```

○ 可模拟HTTP请求验证：

```
curl -i 'http://demo.developer.aliyundoc.com/apptest/stream0000' \  
-H 'User-Agent: iPhone OS;MI 5'
```

IP黑白名单

- 介绍

支持配置IP黑名单或白名单，拒绝或只允许特定IP的访问。

- 添加IP到黑名单，那么该IP无法访问当前加速域名。
- 添加IP到白名单，那么只有该IP能够访问当前加速域名。
- 支持IP列表添加，并支持添加IP网段。

例如：网段127.0.0.0/24，24表示采用子网掩码中的前24位为有效位，即用 $32-24=8$ bit来表示主机号，该子网可以容纳 $2^8-2=254$ 台主机，故可表示IP网段范围：127.0.0.0~127.0.0.254。

- 使用方式

- 控制台：具体操作，请参见[IP黑白名单](#)。
- API：通过调用[BatchSetLiveDomainConfigs](#)接口，可批量配置直播域名，包括IP黑白名单。

5.4. URL鉴权

URL鉴权功能旨在保护用户上传到视频直播的内容资源不被非法站点下载盗用，可通过控制台配置。本文介绍URL鉴权、原理、使用方式和注意事项。

背景

一般情况下，通过配置访问Referer黑名单和白名单来实现对访客身份的识别和过滤，保护站点资源，可以解决部分盗链问题。但由于Referer内容可以伪造，采用URL鉴权方式保护用户源站资源更为安全有效。

使用方法

- 控制台：URL鉴权支持默认鉴权和自定义鉴权两种方式。
 - 默认鉴权：默认为A方式。详细信息请参见[地址生成器](#)。
 - 自定义鉴权：如不采用默认配置可自定义配置，具体操作，请参见[配置URL鉴权](#)。

 注意

- 可通过控制台配置URL鉴权自定义主KEY和副KEY，两者拥有同样的效力，副KEY主要用于平滑更换。
- 若主KEY执行更换，所有使用主KEY生成的播放地址会立即失效。副KEY作为主KEY更换时，使用主KEY的播放地址不会马上中断，副KEY可以继续替代主KEY提供服务。
- 设置目标域名下URL地址的全局默认有效时长后，可对单个URL定制有效时长。此时视频直播会在timestamp后追加设置的默认有效时长。

- 代码拼接鉴权URL。

- 根据推流地址指定的AppName及StreamName得到未鉴权的播流地址，如 `rtmp://DomainName/AppName/StreamName`
- 根据鉴权算法，填入对应的播流地址计算鉴权URL，鉴权组成及算法请参见下表[鉴权URL组成](#)。获取鉴权URL的代码示例，请参见[鉴权代码示例](#)。

鉴权URL组成

鉴权URL适用于PC端、移动端、第三方推流和播放工具。由 `直播推流地址或播放地址+鉴权串` 组成，示例如下：

```
rtmp://DomainName/AppName/StreamName?auth_key=timestamp-rand-uid-md5hash
```

鉴权URL	描述
<code>rtmp://DomainName/AppName/StreamName</code>	直播推流地址或播流地址，更多信息，请参见 推流地址 和 播放地址 。
<code>auth_key=timestamp-rand-uid-md5hash</code>	鉴权串，包含timestamp、rand、uid、md5hash4个字段，字段描述如下。
timestamp	<p>生成鉴权URL的时间，与鉴权URL有效时长共同控制鉴权URL的失效时间。使用UNIX时间戳表示（单位：秒）。</p> <ul style="list-style-type: none"> 若使用控制台的地址生成器生成：timestamp=生成鉴权URL的时间。 若使用控制台的访问控制 > URL鉴权生成：timestamp（失效时间）=生成鉴权URL的时间+有效时间。 若使用代码拼接：您可以自己定义timestamp。以Python为例：<code>timestamp（失效时间） = int(time.time()) + 1 * 3600</code>。
rand	<p>随机数，一般为0。</p> <p>如要确保每次生成的URL不同，建议使用UUID格式，不能包含中划线（-），如：<code>477b3bbc253f467b8def6711128c7****</code>。</p>
uid	附加参数，暂未使用，一般为0。
md5hash	<p>通过md5算法计算出的验证串。数字和小写英文字母混合0~9、a~z，固定长度32。</p> <pre>sstring = "URI-timestamp-rand-uid-PrivateKey" md5hash = md5sum(sstring)</pre> <ul style="list-style-type: none"> URI：用户请求对象相对地址，即原始URL中的 <code>/AppName/StreamName</code>，其中StreamName包含扩展名。 生成FLV播放地址时，URI为 <code>/AppName/StreamName.flv</code>。 生成HLS播放地址时，URI为 <code>/AppName/StreamName.m3u8</code>。 生成FLV转码播放地址时，URI为 <code>/AppName/StreamName_ID.flv</code>。 生成HLS转码播放地址时，URI为 <code>/AppName/StreamName_ID.m3u8</code>。 PrivateKey：控制台配置的主KEY或副KEY，二者皆可。 md5sum：表示计算MD5值，请使用开发语言对应的函数。 <p>说明 使用控制台生成鉴权URL和使用代码拼接鉴权URL的 <code>md5hash</code> 值计算方法相同。</p>

功能原理

当开启URL鉴权功能后，服务器获取请求后会进行如下验证：

1. 判断请求中的失效时间是否小于当前时间。

- 如果小于，则认为过期失效并返回HTTP 403错误。
 - 如果大于当前时间，则构造出一个同样的字符串（参见下文sstring构造方式）。
2. 使用MD5算法算出HashValue，再和请求中带来的md5hash进行比对。
- 比对结果一致，则认为鉴权通过，返回直播流。
 - 否则鉴权失败，返回HTTP 403错误。

HashValue是通过以下字符串计算出来的：

```
sstring = "URI-timestamp-rand-uid-PrivateKey"
HashValue = md5sum(sstring)
```

示例说明

● 假设情况：

- i. 通过req_auth请求对象， `rtmp:// demo.aliyundoc.com /video/standard****`。
- ii. 鉴权KEY设置为：aliyunliveexp1234（控制台设置的主KEY或副KEY，由用户自行设置）。
- iii. 生成鉴权URL时间为：北京时间2021年05月28日16:49:57。
- iv. 控制台中，鉴权URL设置的有效时长为20分钟，生成鉴权URL时配置的有效时间为40分钟。
- v. rand和uid字段都设置为0。

● 则：

- i. 计算出来的鉴权URL的Unix时间戳为1622194197（北京时间2021年05月28日17:29:57）。
- ii. 服务器会构造一个用于计算Hashvalue的签名字符串。

```
/video/standard****-1622194197-0-0-aliyunliveexp1234
```

- iii. 服务器会根据该签名字符串计算HashValue。

```
HashValue = md5sum("/video/standard-1622194197-0-0-aliyunliveexp1234") = 5552ff52b5e4e20387c6dc18afce****
```

- iv. 请求时URL为 `rtmp:// demo.aliyundoc.com /video/standard****?auth_key=1622194197-0-0-5552ff52b5e4e20387c6dc18afce****`

 说明 auth_key即为鉴权URL所带的鉴权信息。

- v. 计算出来的HashValue与用户请求中带的 `md5hash = 5552ff52b5e4e20387c6dc18afce****` 值一致，于是鉴权通过。

注意事项

- 鉴权功能默认为开启状态，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。如果您想要关闭鉴权功能，需要了解盗刷风险并在控制台URL鉴权页面签署《直播服务关闭URL鉴权免责声明》。
- 鉴权功能默认为开启状态，但是配置鉴权auth_key参数必须需要业务端根据鉴权原理自行实现，暂时没有提供API接口计算鉴权auth_key参数。
- 开启鉴权后推流或播流地址均需要添加鉴权auth_key参数才可正常播放。暂时不支持仅推流或者播流一个地址开启鉴权的功能。
- 鉴权URL提供的是过期时间戳的限制，即在该时间段内该地址可以一直访问，不会有任何限制，鉴权还无法实现设置某个URL仅能够一次访问的功能。
- 鉴权auth_key参数是根据去掉queryString后的URI计算MD5值后拼接得到的，参见前述设置鉴权参数小

节。因此，推流和播流地址的URI均为AppName/StreamName；这样会导致推流和播流地址的鉴权auth_key参数其实是一致的。建议用户业务逻辑中如果推流地址有泄漏的情况下设置过期时间戳在业务许可范围内尽量短一些，避免播流地址被恶意访问。

- 鉴权仅会在推流或者播流开始的时候进行验证，在推流或者播流过程中即不会验证，也就是说推流或者播流过程中如果超过了鉴权时间戳也可以继续播放。

5.5. 鉴权代码示例

本文为您介绍URL鉴权的代码示例（Python、Java、Go、PHP和C#），您可以根据业务需要，方便的对URL进行鉴权处理。

URL鉴权规则请参见[直播鉴权](#)，以下示例包含直播所用的鉴权A方式。

说明

- 代码示例中的 `key` 参数对应控制台中URL鉴权的主KEY或备KEY。可在直播控制台的域名管理 > 直播管理 > 访问控制 > URL鉴权页面中的鉴权URL设置模块中配置或查看。
- 生成推流地址时，`key` 需要使用推流域名的主KEY或备KEY。
- 生成播放地址时，`key` 需要使用播流域名的主KEY或备KEY。

Python代码示例

```
import re
import time
import hashlib
import datetime
def md5sum(src):
    m = hashlib.md5()
    m.update(src)
    return m.hexdigest()
def a_auth(uri, key, exp):
    p = re.compile("^rtmp://?([/?]+)(/[?]*)?(\\?.*)?$")
    if not p:
        return None
    m = p.match(uri)
    scheme, host, path, args = m.groups()
    if not scheme: scheme = "rtmp://"
    if not path: path = "/"
    if not args: args = ""
    rand = "0" # "0" by default, other value is ok
    uid = "0" # "0" by default, other value is ok
    sstring = "%s-%s-%s-%s-%s" %(path, exp, rand, uid, key)
    hashvalue = md5sum(sstring.encode('utf-8'))
    auth_key = "%s-%s-%s-%s" %(exp, rand, uid, hashvalue)
    if args:
        return "%s%s%s&auth_key=%s" %(scheme, host, path, args, auth_key)
    else:
        return "%s%s%s?sauth_key=%s" %(scheme, host, path, args, auth_key)
def main():
    uri = "rtmp://example.aliyundoc.com/test/test?vhost=demo.aliyundoc.liucom" #
original uri
    key = "<input private key>" # private key of authorization
    exp = int(time.time()) + 1 * 3600 # expiration time: 1 hour after
current itme
    authuri = a_auth(uri, key, exp) # auth type:
    print("URL : %s\nAUTH: %s" %(uri, authuri))
if __name__ == "__main__":
    main()
```

Java代码示例

```
import java.math.BigInteger;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class AuthDemo {
    private static String md5Sum(String src) {
        MessageDigest md5 = null;
        try {
            md5 = MessageDigest.getInstance("MD5");
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        md5.update(StandardCharsets.UTF_8.encode(src));
        return String.format("%032x", new BigInteger(1, md5.digest()));
    }
    private static String aAuth(String uri, String key, long exp) {
        String pattern = "^((rtmp://)?(?:[^/?]+)\/(?:[^?]*)?(?:\\|\\|\\.*)?*$)";
        Pattern r = Pattern.compile(pattern);
        Matcher m = r.matcher(uri);
        String scheme = "", host = "", path = "", args = "";
        if (m.find()) {
            scheme = m.group(1) == null ? "rtmp://" : m.group(1);
            host = m.group(2) == null ? "" : m.group(2);
            path = m.group(3) == null ? "/" : m.group(3);
            args = m.group(4) == null ? "" : m.group(4);
        } else {
            System.out.println("NO MATCH");
        }
        String rand = "0"; // "0" by default, other value is ok
        String uid = "0"; // "0" by default, other value is ok
        String sString = String.format("%s-%s-%s-%s-%s", path, exp, rand, uid, key);
        String hashValue = md5Sum(sString);
        String authKey = String.format("%s-%s-%s-%s", exp, rand, uid, hashValue);
        if (args.isEmpty()) {
            return String.format("%s%s%s%s?auth_key=%s", scheme, host, path, args, authKey);
        } else {
            return String.format("%s%s%s%s&auth_key=%s", scheme, host, path, args, authKey);
        }
    }
    public static void main(String[] args) {
        String uri = "rtmp://example.aliyundoc.com/live/test****"; // original uri
        String key = "<input private key>"; // private key of authorization
        long exp = System.currentTimeMillis() / 1000 + 1 * 3600; // expiration time: 1 hour after current time
        String authUri = aAuth(uri, key, exp); // auth type:
        System.out.printf("URL : %s\nAuth: %s", uri, authUri);
    }
}
```

Go代码示例

```

package mainimport (
    "crypto/md5"
    "encoding/hex"
    "fmt"
    "regexp"
    "time"
)
func md5sum(src string) string {
    h := md5.New()
    h.Write([]byte(src))
    return hex.EncodeToString(h.Sum(nil))
}
func a_auth(uri, key string, exp int64) string {
    p, err := regexp.Compile("^rtmp://?([^/?]+)(/[^?]*)?(\\?.*)?$")
    if err != nil {
        fmt.Println(err)
        return ""
    }
    m := p.FindStringSubmatch(uri)
    var scheme, host, path, args string
    if len(m) == 5 {
        scheme, host, path, args = m[1], m[2], m[3], m[4]
    } else {
        scheme, host, path, args = "rtmp://", "", "/", ""
    }
    rand := "0" // "0" by default, other value is ok
    uid := "0" // "0" by default, other value is ok
    sstring := fmt.Sprintf("%s-%d-%s-%s-%s", path, exp, rand, uid, key)
    hashvalue := md5sum(sstring)
    auth_key := fmt.Sprintf("%d-%s-%s-%s", exp, rand, uid, hashvalue)
    if len(args) != 0 {
        return fmt.Sprintf("%s%s%s%s&auth_key=%s", scheme, host, path, args, auth_key)
    } else {
        return fmt.Sprintf("%s%s%s%s?auth_key=%s", scheme, host, path, args, auth_key)
    }
}
func main() {
    uri := "rtmp://example.aliyundoc.com/live/test****" // original uri
    key := "<input private key>" // private key o
    f authorization
    exp := time.Now().Unix() + 3600 // expiration ti
    me: 1 hour after current itme
    authuri := a_auth(uri, key, exp) // auth type:
    fmt.Printf("URL : %s\nAUTH: %s", uri, authuri)
}

```

PHP代码示例

```

<?php
function a_auth($uri, $key, $exp) {
    preg_match("/^(rtmp:\\/\\/)?([\^\\/?]+)?(\\/[^\?]*)?(\\?.*)?$/", $uri, $matches);
    $scheme = $matches[1];
    $host = $matches[2];
    $path = $matches[3];
    $args = $matches[4];
    if (empty($args)) {
        $args = "";
    }
    if (empty($scheme)) {
        $scheme = "rtmp://";
    }
    if (empty($path)) {
        $path = "/";
    }
    $rand = "0";
    // "0" by default, other value is ok
    $uid = "0";
    // "0" by default, other value is ok
    $sstring = sprintf("%s-%u-%s-%s-%s", $path, $exp, $rand, $uid, $key);
    $hashvalue = md5($sstring);
    $auth_key = sprintf("%u-%s-%s-%s", $exp, $rand, $uid, $hashvalue);
    if ($args) {
        return sprintf("%s%s%s&auth_key=%s", $scheme, $host, $path, $args, $auth_key);
    } else {
        return sprintf("%s%s%s?auth_key=%s", $scheme, $host, $path, $args, $auth_key);
    }
}
$uri = "rtmp://example.aliyundoc.com/live/test****";
$key = "<input private key>";
$exp = time() + 3600;
$authuri = a_auth($uri, $key, $exp);
echo "URL : " . $uri;
echo PHP_EOL;
echo "AUTH: " . $authuri;
?>

```

C#代码示例

```

using System;
using System.Text.RegularExpressions;
using System.Security.Cryptography;
using System.Text;
public class Test
{
    public static void Main()
    {
        string uri= "rtmp://example.aliyundoc.com/live/test****"; // original uri
        string key= "<input private key>"; // private key of authentication
        DateTime dateStart = new DateTime(1970, 1, 1, 8, 0, 0);
        string exp = Convert.ToInt64((DateTime.Now - dateStart).TotalSeconds+3600).ToStri

```

```
ng()); // expiration time: 1 hour after current time
    string authUri = aAuth(uri, key, exp);
    Console.WriteLine (String.Format("URL :{0}",uri));
    Console.WriteLine (String.Format("AUTH :{0}",authUri));
}
public static string aAuth(string uri, string key, string exp)
{
    Regex regex = new Regex("^rtmp://)?(?:[/?]+) ([^?]*)?(\\|\\.?)?$");
    Match m = regex.Match(uri);
    string scheme = "rtmp://", host = "", path = "/", args = "";
    if (m.Success)
    {
        scheme=m.Groups[1].Value;
        host=m.Groups[2].Value;
        path=m.Groups[3].Value;
        args=m.Groups[4].Value;
    }else{
        Console.WriteLine ("NO MATCH");
    }
    string rand = "0"; // "0" by default, other value is ok
    string uid = "0"; // "0" by default, other value is ok
    string u = String.Format("{0}-{1}-{2}-{3}-{4}", path, exp, rand, uid, key);
    string hashValue = Md5(u);
    string authKey = String.Format("{0}-{1}-{2}-{3}", exp, rand, uid, hashValue);
    if (args=="")
    {
        return String.Format("{0}{1}{2}{3}?auth_key={4}", scheme, host, path, args, authKey);
    } else
    {
        return String.Format("{0}{1}{2}{3}&auth_key={4}", scheme, host, path, args, authKey);
    }
}
public static string Md5(string value)
{
    MD5CryptoServiceProvider md5 = new MD5CryptoServiceProvider();
    byte[] bytes = Encoding.ASCII.GetBytes(value);
    byte[] encoded = md5.ComputeHash(bytes);
    StringBuilder sb = new StringBuilder();
    for(int i=0; i<encoded.Length; ++i)
    {
        sb.Append(encoded[i].ToString("x2"));
    }
    return sb.ToString();
}
}
```

5.6. 阿里云视频加密

阿里云视频加密可针对直播流进行加密，本文介绍优势、整体架构和使用方法。

背景

防盗链URL不能防止用户通过一次付费行为拿到付费直播合法的防盗链URL，进而实现二次分发。因此，防盗链方案对于直播的版权保护是远远不够的。

优势介绍

阿里云视频加密是对视频数据加密，即使下载到本地，视频本身也是被加密的，无法恶意二次分发。视频加密可有效防止视频泄露和盗链问题，广泛用于在线教育、财经金融、行业培训、独播剧等在线版权视频领域。

采用阿里云私有加密算法，安全级别高，能够便捷、高效、安全地保护视频资源。

- 每个媒体文件拥有独立的加密密钥，能有效避免采用单一密钥时，一个密钥的泄露引起大范围的安全问题。
- 视频直播提供完善的权限管理机制“子账号+播放凭证”。
- 视频直播提供信封加密机制“密文Key+明文Key”，明文Key不存储，所有过程只在内存中。
- 视频直播提供安全的播放内核SDK。

整体架构

阿里云视频加密方案包含两部分：加密转码+解密播放。

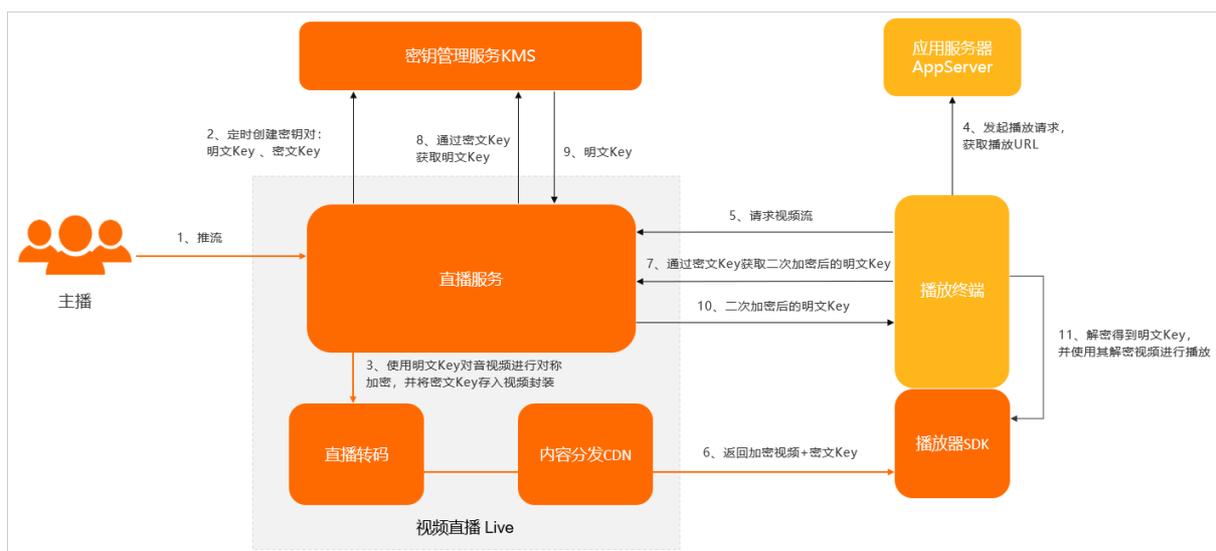
- 加密转码：流程1~3。

主播推流到直播中心后，直播服务负责通过密钥管理服务KMS生成明文Key和密文Key。通过直播转码服务，使用明文密钥对音视频进行对称加密，并将密文Key存入视频封装。

- 解密播放：流程4~11。

播放终端如需播放此直播流，需向应用服务器AppServer发起播放请求，获取播流URL。使用播流URL向直播服务请求视频流。直播服务将经过转码的加密视频和密文Key传输到播放器SDK。

播放终端用密文Key向直播服务获取二次加密后的明文Key，直播服务再通过密文Key向密钥管理服务KMS获取明文Key。播放终端将解密得到的明文Key传给播放器SDK，播放器SDK解密视频进行播放。



使用方式

在**密钥管理服务控制台**创建密钥并将密钥ID配置到直播转码模板中，使用**阿里云播放器**对加密后的视频流进行解密播放。

视频直播暂未开放控制台进行阿里云视频加密，您可以通过调用AddLiveStreamTranscode接口，配置 EncryptParameters 参数，或提交工单申请。

说明

- 仅支持输出HLS格式。
- 只能使用阿里云播放器。
- 暂不支持网页端播放。
- 使用视频加密，需要访问KMS服务，需要通过直播加密服务关联角色AliyunServiceRoleForLiveKms获取访问权限。

5.7. DRM加密

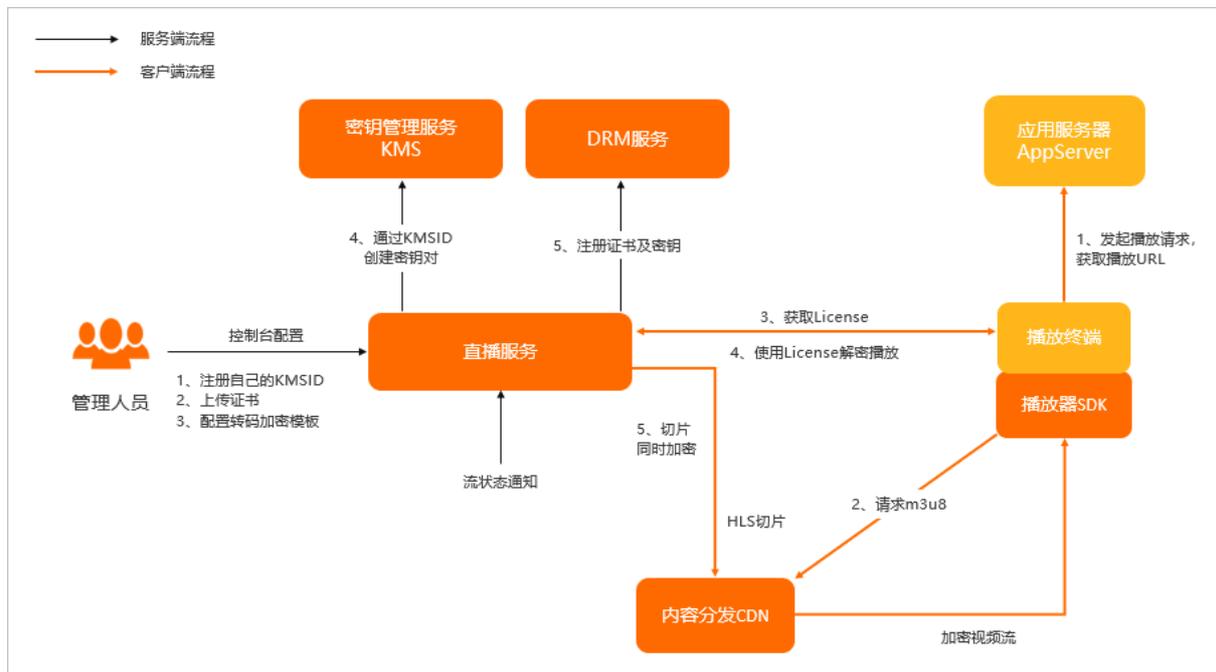
阿里云视频直播服务提供了行业通用的DRM（Digital Rights Management）加密方式，通过一站式添加和管理，轻松保护版权视频内容的安全。本文介绍DRM加密的功能、优势和使用限制。

功能说明

阿里云提供的DRM加密服务，包括如下功能：

- DRM加密：支持Widevine和Fairplay的DRM加密。
- 证书管理：支持对Fairplay加密证书的管理。
- DRM播放：阿里云播放器提供解密以及播放的功能。

架构图如下所示：



优势

- 一站式：提供了视频加密、证书管理、许可证派发、解密播放等功能，无须深入了解DRM架构和实现，使用门槛低。
- 低成本：支持一次转码即支持Widevine和Fairplay的加密，减少转码成本。支持灵活的按量计费模式，无

保底费用。

- 安全稳定、高并发：通过对权限精细管理和系统的安全设置，保护版权内容的存储安全。同时分布式的服务也满足了业务高并发的需求。

使用限制

DRM方案	移动端	浏览器
Widevine	Android	Chrome、Firefox (47+)、Edge、Opera
Fairplay	iOS	Safari

 说明 目前DRM功能仅支持控制台使用，具体操作请参见[配置DRM加密](#)。

5.8. 内容安全

视频直播提供智能审核服务，来对直播内容的合规进行审核，包括视频审核和语音审核的功能，配合直播流禁推功能，实现直播内容安全。本文介绍智能审核的背景、介绍、使用方法、收费方式和禁推直播流的功能。

内容安全机制是利用智能审核功能对直播内容进行审核，将审核结果返回给运营人员，由运营人员管理相应的直播流，可对直播流进行禁止推送的操作。视频直播还提供了高频率智能审核的方案供您参考，您可以参见[高频率智能审核解决方案](#)。

智能审核

● 背景

主播推送的直播流可能有内容违规风险，视频直播服务提供了支持视频、音频的智能审核，通过自动化、智能审核媒资库资源，极大降低色情、暴恐、涉政等内容违规风险，降低人工审核成本。

● 介绍

智能审核服务基于海量标注数据和深度学习算法实现，从语音、视觉等多维度精准识别直播流媒体信息中包含的违禁内容，支持涉黄识别、暴恐涉政识别、广告识别、无意义直播识别、音频审核等多个功能模块。

- 视频审核：采用截帧画面进行审核，自动检测视频直播的内容是否涉嫌违规，违规的内容将会通过回调的方式返回给用户，用户可对违规内容进行处理。
- 语音审核：采用识别直播流中音频流的内容，自动检测语音内容是否涉嫌违规，违规的内容将会通过回调的方式返回给用户，用户可对违规内容进行处理。

具体功能如下：

功能名称	Scene	描述	检测结果分类
鉴黄识别	porn	识别直播中的色情内容。	正常图片、性感图片、色情图片
暴恐涉政识别	terrorism	识别直播中的暴恐涉政内容。	正常图片、血腥、爆炸烟光、特殊装束、特殊标识、武器、涉政、打斗、聚众、游行、车祸现场、旗帜、地标、其他

功能名称	Scene	描述	检测结果分类
图文违规识别	ad	识别直播中的广告信息。	正常图片、含其他广告、含牛皮癣广告、含二维码、含小程序码
不良场景识别	live	识别直播中出现的黑屏、黑边、昏暗画面，画中画，抽烟，打架等不良场景图片。	正常图片、无意义图片、画中画、吸烟、车内直播
图片logo识别	logo	识别直播中出现受管控的logo和商标。	正常图片、含受管控的logo、含商标
语音审核	antispam	检测音频中是否包含垃圾信息、广告、涉政、暴恐、辱骂、色情、灌水、违规、无意义等内容。	正常文本、含垃圾信息、广告、涉政、暴恐、辱骂、色情、灌水、违禁、无意义、自定义（例如命中自定义关键词）

● 使用方法

○ 前置条件

- a. 需要开通直播服务，并配置一个有效的直播域名。
- b. 需要一个HTTP服务用于接收异常视频或语音的回调信息。

○ 使用流程

- a. 添加智能审核配置。
- b. 配置回调地址。
- c. 完成配置后，推流即开启语音审核。

 **说明** 审核只对于配置结束后的新推流生效，对于已有推流不会立刻生效。

○ 控制台

视频审核：支持配置视频审核的审核模板和回调地址，具体操作，请参见[配置审核](#)。

音频审核：不支持。

○ API

视频审核

接口	描述	参考文档
AddLiveSnapshotDetectPornConfig	添加直播视频审核配置。	添加直播审核配置
DeleteLiveSnapshotDetectPornConfig	删除直播视频审核配置。	删除直播审核配置
UpdateLiveSnapshotDetectPornConfig	更新直播视频审核配置。	更新直播审核配置
DescribeLiveSnapshotDetectPornConfig	查询直播视频审核配置。	查询直播审核配置

接口	描述	参考文档
AddLiveDetectNotifyConfig	添加直播视频审核回调配置。 默认格式请参见 视频审核默认回调格式说明 。	添加直播审核回调配置
DeleteLiveDetectNotifyConfig	删除直播视频审核回调配置。	删除直播审核回调配置
UpdateLiveDetectNotifyConfig	更新直播视频审核回调配置。	更新直播审核回调配置
DescribeLiveAudioAuditNotifyConfig	查询直播视频审核回调配置。	查询直播审核回调配置

语音审核

接口	描述	参考文档
AddLiveAudioAuditConfig	添加直播语音审核配置。	添加直播语音审核配置
DeleteLiveAudioAuditConfig	删除直播语音审核配置。	删除直播语音审核配置
UpdateLiveAudioAuditConfig	更新直播语音审核配置。	更新直播语音审核配置
DescribeLiveSnapshotDetectPornConfig	查询直播语音审核配置。	查询直播语音审核配置
AddLiveAudioAuditNotifyConfig	添加直播语音审核回调配置。用户可以根据自己需求自定义回调JSON格式。 默认格式请参见 语音审核默认回调格式说明 。	添加直播语音审核回调配置
DeleteLiveAudioAuditNotifyConfig	删除直播语音审核回调配置。	删除直播语音审核回调配置
UpdateLiveAudioAuditNotifyConfig	更新直播语音审核回调配置。	更新直播语音审核回调配置
DescribeLiveAudioAuditNotifyConfig	查询直播语音审核回调配置。	查询直播语音审核回调配置

- 计费规则：智能审核包括直播图片审核和直播语音审核，图片审核以您使用审核服务的图片扫描张数为结算标准，语音审核以您使用语音审核服务的总时长为结算标准。更多信息，请参见[智能审核费用](#)。

禁推直播流

视频直播提供了丰富的直播流管理功能，其中禁推直播流功能可帮助您在遇到下列情况的时候，对某路直播推流进行禁止推流的操作。

- 鉴权推流地址暴露。

- 用户恶意推流。
- 业务需求。

使用方法

- 控制台：支持在控制台查看禁推流、设置禁推流和恢复直播流。您可以在[视频直播控制台](#)的直播管理 > 流管理页面查看，请参见[流管理](#)。
- API

接口	描述	参考文档
ForbidLiveStream	<p>禁止某路流的推送。</p> <p>可以配置是否只断流不加入黑名单，预设某个时刻将流恢复。</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> 说明 禁止直播流的上限为10,000路，超出限制将禁用失败。调用频率限制为20次/秒。</p> </div>	禁止直播流
ResumeLiveStream	<p>恢复某路流的推送。</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> 说明 调用频率限制为30次/秒。</p> </div>	恢复直播流

视频审核默认回调格式说明

只会对于有问题的视频内容进行回调通知，通知内容包含问题视频截图的审核信息和存储信息。

```

{
  "DomainName": "example.com",
  "AppName": "liveApp****",
  "StreamName": "liveStream****",
  "OssEndpoint": "oss-cn-hangzhou.aliyuncs.com",
  "OssBucket": "liveBucket****",
  "OssObject": "liveObject****.jpg",
  "Result": [
    {
      "BizType": "example.com",
      "Result": [
        {
          "Label": "Porn", "Rate":11, "Suggestion": "review", "Scene":"porn", "Extent": {}
        },
        {
          "Label": "Ad", "Rate":11, "Suggestion": "review", "Scene":"ad", "Extent": {}
        }
      ]
    }
  ]
}
    
```

参数说明

参数	类型	描述
DomainName	String	播流域名。
AppName	String	应用名称。
StreamName	String	流名称。
OssEndpoint	String	存储对象Endpoint。
OssBucket	String	存储对象的Bucket。
OssObject	String	存储对象的文件名。
Result	JSONArray	检测结果。请参见Result。

Result

参数	类型	描述
BizType	String	业务类型。可用于选择模型，默认值为域名。
Scene	String	检测场景 <ul style="list-style-type: none">• porn: 鉴黄。• terrorism: 暴恐涉政。• ad: 图文违规。• live: 不良场景。• logo: 图片logo。

参数	类型	描述
Label	String	<p>检测结果的分类。不同检测场景的结果分类不同，具体如下：</p> <ul style="list-style-type: none"> • 图片智能鉴黄 (<i>porn</i>) 结果分类： <ul style="list-style-type: none"> ◦ <i>normal</i>: 正常。 ◦ <i>sexy</i>: 性感。 ◦ <i>porn</i>: 色情。 • 图片暴恐涉政 (<i>terrorism</i>) 结果分类： <ul style="list-style-type: none"> ◦ <i>normal</i>: 正常。 ◦ <i>bloody</i>: 血腥。 ◦ <i>explosion</i>: 爆炸烟光。 ◦ <i>outfit</i>: 特殊装束。 ◦ <i>logo</i>: 特殊标识。 ◦ <i>weapon</i>: 武器。 ◦ <i>politics</i>: 涉政。 ◦ <i>violence</i>: 打斗。 ◦ <i>crowd</i>: 聚众。 ◦ <i>parade</i>: 游行。 ◦ <i>carcrash</i>: 车祸现场。 ◦ <i>flag</i>: 旗帜。 ◦ <i>location</i>: 地标。 ◦ <i>others</i>: 其他。 • 图文违规 (<i>ad</i>) 结果分类： <ul style="list-style-type: none"> ◦ <i>normal</i>: 正常。 ◦ <i>ad</i>: 其他广告。 ◦ <i>npa</i>: 牛皮癣广告。 ◦ <i>qrcode</i>: 含二维码。 ◦ <i>programCode</i>: 含小程序码。 • 不良场景 (<i>live</i>) 结果分类： <ul style="list-style-type: none"> ◦ <i>normal</i>: 正常。 ◦ <i>meaningless</i>: 图片中无内容（例如，黑屏、白屏）。 ◦ <i>PIP</i>: 画中画。 ◦ <i>smoking</i>: 吸烟。 ◦ <i>drivelive</i>: 车内直播。 • 图片logo (<i>logo</i>) 结果分类： <ul style="list-style-type: none"> ◦ <i>normal</i>: 正常。 ◦ <i>TV</i>: 含受管控的logo。 ◦ <i>trademark</i>: 含商标。

参数	类型	置信度分数。取值范围：0（表示置信度最低）~100（表示置信度最高）。
Rate	Float	<p>② 说明 该值仅作为参考，强烈建议您不要在业务中使用。建议您参考Label结果用于内容违规判定。</p>
Extent	String	预留字段。

② 说明 默认新用户将使用此版本，老用户维持不变，如需使用此版本，请您提交工单。

语音审核默认回调格式说明

只会对于有问题的音频内容进行回调通知，通知内容包含问题语音的文本信息以及最近一分钟内的上下文信息。

```

{
  "domain": "example.com",
  "app": "liveApp****",
  "stream": "5d9747eba39b44769852d276f9ff****",
  "timestamp": 1572248095,
  "result": [
    {
      "scene": "antispam",
      "label": "ad",
      "suggestion": "block",
      "rate": 99.91,
      "details": [
        {
          "startTime": 1572248023,
          "endTime": 1572248040,
          "text": "大理陇川等酒店免费居住权120天居住权可与亲友共享来云栖社国际烧酒店接待中心地址芒市团结大街96号一一ktv旁咨询热线2285699",
          "label": "ad"
        },
        {
          "startTime": 1572248040,
          "endTime": 1572248070,
          "text": "典藏经典别墅尊享梁河金塔、旅游小镇二期滨湖苑地区11月2号盛大开盘天然龙窝、泉水入户户型多样,设计经典价位超值,值得您拥有开盘当天还有超值优惠,欢迎您前来品鉴,凭借地址,然和县遮岛镇先锋路229号分享热线069269557776955777",
          "label": "normal"
        },
        {
          "startTime": 1572248072,
          "endTime": 1572248077,
          "text": "快乐的时光,有你相伴惬意的日子,格外轻松。",
          "label": "normal"
        },
        {
          "startTime": 1572248078,
          "endTime": 1572248086,
          "text": "fme043忙里偷闲音乐时光有歌曲温暖,有你想念影院。",
          "label": "normal"
        }
      ]
    }
  ]
}

```

参数	类型	描述
domain	String	播流域名。
app	String	应用名称。
stream	String	流名称。

参数	类型	描述
timestamp	Int	回调时间戳。单位：秒。
result	JSONArray	检测结果。请参见Result。

Result

参数	类型	描述
scene	String	检测场景。
label	String	检测结果的分类。取值： <ul style="list-style-type: none"> • <i>normal</i>: 正常文本 • <i>spam</i>: 含垃圾信息 • <i>ad</i>: 广告 • <i>politics</i>: 涉政 • <i>terrorism</i>: 暴恐 • <i>abuse</i>: 辱骂 • <i>porn</i>: 色情 • <i>flood</i>: 灌水 • <i>contraband</i>: 违禁 • <i>meaningless</i>: 无意义 • <i>customized</i>: 自定义（例如命中自定义关键词）
suggestion	String	处理建议。取值： <ul style="list-style-type: none"> • <i>pass</i>: 结果正常，无需进行其余操作。 • <i>review</i>: 结果不确定，需要进行人工审核。 • <i>block</i>: 结果违规，建议直接删除或者限制公开。
rate	Float	置信度分数。取值范围：0（表示置信度最低）~100（表示置信度最高）。 <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> ? 说明 该值仅作为参考，强烈建议您不要在业务中使用。建议您参考Label结果用于内容违规判定。 </div>
details	JSONArray	语音对应的文本详情，每一句文本对应一个元素，可以包含一个或者多个元素。关于每个元素的结构描述，请参见Detail。

Detail

参数	类型	描述
startTime	Int	句子开始的时间戳，单位：秒。
endTime	Int	句子结束的时间戳，单位：秒。
text	String	问题语音转换成文本的结果。
label	String	检测结果的分类。取值： <ul style="list-style-type: none"> • <i>normal</i>: 正常文本 • <i>spam</i>: 含垃圾信息 • <i>ad</i>: 广告 • <i>politics</i>: 涉政 • <i>terrorism</i>: 暴恐 • <i>abuse</i>: 辱骂 • <i>porn</i>: 色情 • <i>flood</i>: 灌水 • <i>contraband</i>: 违禁 • <i>meaningless</i>: 无意义 • <i>customized</i>: 自定义（例如命中自定义关键词）

6. 直播延迟

本文主要介绍造成视频直播延迟的原因及降低延迟的解决方法。

造成直播延迟的原因

推流端

- GOP (Group Of Picture) 帧是视频的关键帧，是视频图像编码器和解码器存取的基本单位。直播会将每一帧数据打上时序标签，并进行网络传输。视频的关键帧数量过大会造成直播延迟。
- 多数第三方推流软件为解决卡顿问题会采用增大编码缓存的方法，而编码缓存过大，则会造成直播延迟。
- 由于硬件条件限制，无法满足编码端码率、帧率以及编码的高档位设置需求，造成编码延迟影响直播。

服务端

在视频播放前，为了保证直播秒开以及降低卡顿，会先缓存部分直播数据，这样能保证数据的流畅性，但缓存数据会造成一定的直播延迟；在视频播放中，由于网络抖动可能导致数据无法传送到客户端，因此会造成2秒~3秒的延迟。

播流端

多数不支持快进的播流端都是在接收缓存收满后才进行解码直播，而这部分接收的缓存会导致直播延迟。

如何降低直播延迟

您可通过以下配置来降低直播延迟：

- 推流端配置
 - 将GOP帧设置为1秒~2秒，减少播放器加载GOP帧所用的时间来降低直播延迟。控制台GOP设置，请参见[自定义转码](#)；API接口配置，请参见[添加直播流自定义转码配置](#)。
 - 由于编码缓存过大引起的直播延迟，建议您使用阿里云推流SDK降低延迟。
 - 对于推流设备的使用，推流iOS端建议您使用硬编码，其效率高且省电；Android端因机型复杂，CPU类型众多，采用硬编码可能出现兼容性问题，推流Android端建议您使用软编码。

- 服务端配置

将服务端缓存调小，来降低直播延迟。您可通过控制台进行直播延迟配置，根据不同的播流协议来配置延迟时间，延迟配置越低服务端缓存越小。如下图所示：



说明 将缓存调小后，当网络环境不稳定时，数据无法及时下载，会出现直播卡顿的现象。

● 选用播流协议

阿里云提供HTTP-FLV、HLS、RTMP三种播放协议，您可以根据直播场景选择使用。

- HTTP-FLV和RTMP延迟较小，适合低延迟播放的场景。HLS延迟较大但兼容性好，适合对延迟不敏感但需要更多播放终端可以播放的场景。
- HTTP-FLV和RTMP需要Flash播放器，HLS可以直接在浏览器中播放。
- 手机浏览器观看直播视频，仅可使用HLS播流协议。

说明 如果您的播流端使用的是HLS协议，感觉延迟较高，由于HLS协议的延迟在10秒~30秒左右为正常现象，您可以将HLS协议换为FLV协议即可解决此问题。

HTTP-FLV、HLS、RTMP协议对比如下表所示：

协议名称	协议简介	采用的传输协议	视频封装格式	推荐直播场景
HTTP-FLV	由Adobe公司推出，将流媒体数据封装成FLV格式，通过HTTP协议传输给客户端，延迟约2秒。支持使用HTTPS加密传输，支持Android移动端和iOS移动端。	HTTP	FLV、TAG	降低延迟

协议名称	协议简介	采用的传输协议	视频封装格式	推荐直播场景
HLS	由Apple公司推出，基于HTTP的流媒体传输协议。将流媒体数据切成连续的TS分片，每个分片有5秒以上的时长，分片数量一般为3~4个，所以总延迟约10秒~30秒，但流畅性较好。主要应用于iOS设备，提供音视频直播服务和录制内容等服务。	HTTP	M3U8、TS	跨终端
RTMP	由Adobe公司推出，传输的过程中消息会被拆分为更小的消息块（Chunk）单元，再将分割后的消息块通过TCP协议传输，接收端再反解接收的消息块恢复成流媒体数据将大文件拆分，因过程复杂可能导致不稳定。在iOS端，需要使用第三方解码器才能播放。	HTTP	FLV、TAG	互动式直播

超低延时直播

如果上述方法仍未帮您降低延时，您可以尝试使用超低延时服务RTS（Real-time Streaming）。RTS具备千万级并发场景下的毫秒级延时直播能力，适用于较大的互动直播场景，弥补了传统直播3秒~6秒延时的问題，保障超低延时、低卡顿、秒开流畅的直播观看体验。

- 超低延时直播费用

超低延时直播费用与标准直播费用不同。具体计费规则，请参见[阿里云产品定价页](#)

- 开通超低延时直播服务

如何开通RTS服务及具体使用方法，请参见[超低延时直播简介](#)。

7. 直播录制

7.1. 概述

本文档主要介绍直播录制的工作原理及其API提供的功能。

1 什么是直播录制

1.1 直播录制的原理

直播录制的过程，是通过拉取您推上来的直播流，将音视频封装成单独的一个个媒体切片（封装格式为TS），然后存入您指定的OSS bucket。直播录制在工作时遵循以下原则：

- 直播录制只修改音视频的封装格式（RTMP/FLV转成TS），绝不修改音视频内容（即编码层内容）。举个例子，如果您推上来的直播流是花屏的，那录制下来的也是花屏的。
- 直播录制有权限往用户bucket写入录制文件，但是不会也没有权限修改/删除用户bucket的任务文件（包括已写入的录制文件）。用户bucket中的录制文件，完全由您自己管理。

1.2 录制文件的封装格式

直播录制支持将直播流录制成以下封装格式的视频文件：

- TS/M3U8
- MP4
- FLV

但是不管是哪种封装格式，直播录制都会先将直播流切片成一个个的TS切片，然后再合成对应封装格式的录制文件。稍有区别的是，对于M3U8来说，直播录制会将TS和M3U8都写入用户的bucket（因为M3U8只是HLS协议中的索引文件，音视频数据仍然在TS中）；而对于MP4和FLV，用户bucket中只有合成后的MP4和FLV切片。

1.3 限制条件

1.3.1 OSS写入权限

正常情况下，开通直播服务时，您已自动授权“允许直播服务写入用户OSS”，因此直播录制写入您指定的bucket时不存在权限问题。如果该权限意外被删除，您可以重新配置：

- 通过控制台配置：您需要授权视频直播可将视频内容写入OSS产品的权限，授权后才能将视频存储至指定的OSS bucket中。详情参见[配置OSS](#)。
- 通过RAM进行权限配置：详情参见[创建RAM用户并授权](#)。

1.3.2 音视频编码格式

由于直播录制需要先将直播流切分成TS文件，因此您推上来的直播流的音视频编码格式必须满足TS封装格式的要求。根据FLV标准和ISO/IEC 13818-1标准，直播录制目前仅支持以下编码格式：

- 视频：H264、HEVC、MPEG4
- 音频：AAC、MP3

 **说明** 如果直播流包含了非以上格式编码的音视频，直播录制可能会出现：无法生成录制文件、录制黑屏、录制没有声音，以及其他一些无法预见的异常情况。

1.3.3 异常的直播流

对于正常的直播流（没有不支持的编码格式、帧率稳定、时间戳单调线性增长），直播录制均能正常生成录制文件。当然，对于偶尔帧率不稳定，时间戳跳变的流，直播录制也能做出一定的兼容，保证录制文件的正常生成。但是如果推上来的直播流存在严重异常（长时间没有视频帧，时间戳增长没有规律，音视频头缺失等等），直播录制可能无法保证正常生成录制文件。您如果对此存在疑问，可通过工单咨询。

直播录制API提供哪些功能

- 自动录制

一旦推流就启动录制，一旦断流就停止录制。您可选择对某个域名下所有流、某个appname下所有流、或者单独某条流开启自动录制。详细用法参见：[自动录制](#)。
- 按需录制

在自动录制的基础上，您可以动态决定具体某条流是否需要录制。详细用法参见：[按需录制](#)。
- 手动录制

如果不希望自动录制，您也可以选择通过调用接口触发直播录制的启动与停止。详细用法参见：[手动录制](#)。
- 录制内容检索与管理

直播录制每生成一个录制文件，都可以提供回调，方便您实现一些自定义的逻辑。另外，已经录制下来的内容，我们也提供检索与管理的功能。详细用法参见：[录制内容检索与管理](#)。

7.2. 自动录制

本文介绍自动录制的使用流程、注意事项和实践示例。

使用场景

- 只要有推流就会进行自动录制。同一条直播流推流，要么录制，要么不录制。
- 指定录制的时间段，如果该时间段内有推流，就启动自动录制。

使用流程

1. 配置自动录制规则，调用接口请参见[AddLiveAppRecordConfig](#)。
2. 进行推流，详细内容请参见[推流、拉流与播流](#)。

相关API参见：

API名称	说明
DeleteLiveAppRecordConfig	如果不再需要某项规则，可以通过该接口删除。
DescribeLiveRecordConfig	查询所有配置的规则

注意事项

- 修改配置后，新配置对修改之前的直播流不生效，必须重新推流才能生效。
- 如果指定了时间段，在该时间段内如果没有推流，自然不会录制。限定时间段的配置规则是一次性的，即当指定的时间段过去之后，该规则不会再触发。需要注意的是，StartTime和EndTime字段填的是UTC时间，请注意和本地时区的对应。
- AddLiveAppRecordConfig接口中的AppName和StreamName可以填为*，表示所有AppName和所有StreamName（即不限制AppName或StreamName）。

- 如果想知道自动录制是否生效，或者希望针对每个录制文件做实时处理，可以设置录制回调，详见：[录制内容检索与管理](#)。
- 自动录制每隔一定周期（周期时间通过RecordFormat.N.CycleDuration字段配置）会产生一个录制文件。如果在一个录制周期内，直播流发生了断流，但是在3分钟内，该直播流又推上来了，那么仍会在同一个录制文件中继续录制。这就意味着，一条直播流必须断流超过3分钟，才会生成最后一个录制文件。如果您希望修改这个默认的3分钟断流时间，可以提交工单在后台修改。
- 如果不指定录制周期，录制文件会在每6小时自动生成一个录制文件。
- 可以通过AddLiveAppRecordConfig配置多条规则，规则匹配时存在优先级：
 - 同时指定DomainName、AppName（不为*）、StreamName（不为*）的优先级最高。
 - 同时指定DomainName、AppName（不为*）的优先级次之。
 - 单独指定DomainName，AppName为*（即只限定域名）的优先级最低。

实践示例

指定某个appname下所有流自动录制

下例是最简单的用法，指定域名为 `example.com`，appname为liveApp****下的所有直播流自动录制，录制格式为M3U8，单个录制文件周期默认为1小时，录制到 `oss-cn-shanghai.aliyuncs.com` 名为liveBucket****的bucket中：

1. 调用AddLiveAppRecordConfig接口，参数如下：

```
/?AppName=liveApp****
&DomainName=example.com
&OssBucket=liveBucket****
&OssEndpoint=oss-cn-shanghai.aliyuncs.com
&RecordFormat.1.Format=m3u8
&<公共请求参数>
```

2. 推流。

同时录制M3U8和MP4

在上例的基础上，下例设置同时录制M3U8和MP4两种格式，同时设置MP4的切片周期为半小时：

1. 调用AddLiveAppRecordConfig接口，参数如下：

```
/?AppName=liveApp****
&DomainName=example.com
&OssBucket=liveBucket****
&OssEndpoint=oss-cn-shanghai.aliyuncs.com
&RecordFormat.1.Format=m3u8
&RecordFormat.2.Format=mp4
&RecordFormat.2.CycleDuration=1800
&<公共请求参数>
```

2. 推流。

限定自动录制的时间段

 **说明** 限制条件：该功能仅针对流级别的配置生效（即配置中StreamName字段不为空才生效）。

在上例的基础上，下例限定自动录制的时间段为UTC时间（北京时间-8） `2019-02-15 09:00:00` 到 `2019-02-15 21:00:00`。

1. 调用AddLiveAppRecordConfig接口，参数如下：

```

/?AppName=liveApp****
&DomainName=example.com
&StreamName=liveStream****
&OssBucket=liveBucket****
&OssEndpoint=oss-cn-shanghai.aliyuncs.com
&RecordFormat.1.Format=m3u8
&RecordFormat.2.Format=mp4
&RecordFormat.2.CycleDuration=1800
&StartTime=2019-02-15T01:00:00Z
&EndTime=2019-02-15T13:00:00Z
&<公共请求参数>

```

2. 推流。

7.3. 按需录制

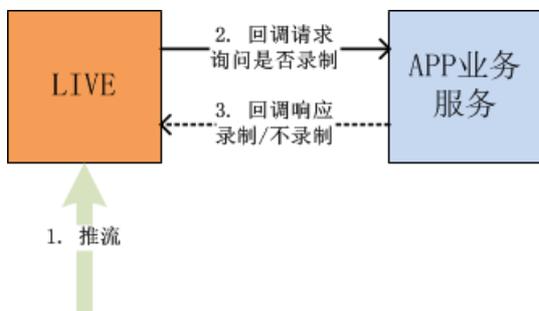
本文介绍按需录制的适用场景、使用流程、实践示例和注意事项。

适用场景

在自动录制的场景下，根据事先设定的配置，一条直播流要么录制，要么不录制，不可以动态修改录制行为。如果您想动态地控制录制的行为，可以使用按需录制的功能。

按需录制的原理是当直播流推流上来之后，直播录制服务会发送一条HTTP回调给您事先设定的后台服务，通过该回调的返回内容决定是否录制这条直播流，并且可以动态的修改录制格式和录制周期。

系统交互如下图所示：



在这里，您需要开发上图中的App业务服务，接收HTTP回调，并根据您自身的业务逻辑决定是否录制。

使用流程

1. 和自动录制一样，调用AddLiveAppRecordConfig接口。其中OnDemand字段填1，表示通过HTTP回调开启按需录制。
2. 调用AddLiveRecordNotifyConfig接口，设置OnDemandUrl字段。
3. 推流。
4. App业务服务接收到该推流的按需录制回调，根据自身业务决定是否录制，返回内容。
5. 直播录制服务根据按需录制回调返回的结果，启动录制或不启动录制。

涉及到的API包括：

API名称	说明
AddLiveAppRecordConfig	配置自动录制规则。
DeleteLiveAppRecordConfig	如果不再需要某项规则，可以通过该接口删除。
DescribeLiveRecordConfig	查询所有配置的规则。
AddLiveRecordNotifyConfig	配置某个域名下的录制回调，包括事件回调和按需录制回调。
DeleteLiveRecordNotifyConfig	如果不再需要某个域名的录制回调，可以通过该接口删除。
DescribeLiveRecordNotifyConfig	查询某个域名的录制回调。
UpdateLiveRecordNotifyConfig	更新某个域名的录制回调。
按需录制回调	按需录制回调给您的HTTP消息定义。

实践示例

- 调用AddLiveAppRecordConfig接口，配置指定域名为 `example.com`，appname为liveBucket****下的所有直播流自动录制：

```
/?AppName=liveApp****
&DomainName=example.com
&OssBucket=liveBucket****
&OssEndpoint=oss-cn-shanghai.aliyuncs.com
&RecordFormat.1.Format=m3u8
&<公共请求参数>
```

- 调用AddLiveRecordNotifyConfig接口，设置OnDemandUrl字段为http://example.aliyundoc.com：

```
/?DomainName=example.com
&OnDemandUrl=http://example.aliyundoc.com
&<公共请求参数>
```

- 推流。
- App业务服务接收到的按需录制回调为：（详见[按需录制回调](#)）

```
GET /?app=liveApp****&domain=example.com&stream=liveStream****&vbitrate=2000&codec=h264
HTTP/1.1
Host: push.aliyundoc.com
User-Agent: Go-http-client/1.1
```

- 根据回调返回的结果：（详见[按需录制回调](#)）

- 如果收到以下结果，则进行录制：

```
{
  "ApiVersion" : "1.0",
  "NeedRecord" : true
}
```

- 如果收到以下结果，则不录制：

```
{
  "ApiVersion" : "1.0",
  "NeedRecord" : false
}
```

注意事项

- 如果没有设置OnDemandUrl，就不会有按需录制回调，默认行为是不录制。
- 视频直播支持触发拉流，当使用触发拉流域名对应的播放地址进行播放时，会自动触发阿里云视频直播服务进行拉流直播，无人播放时则不会回源站进行拉流。自动录制、按需录制、手动录制功能应用在触发拉流场景时，如果触发拉流无人播放，则不会回源站拉流，自动录制、按需录制、手动录制同样也不会进行录制。

7.4. 手动录制

本文介绍手动录制的适用场景、使用限制、使用流程、实践示例和注意事项。

适用场景

您也可以事先配置录制配置，但是设置默认行为是不录制（设置ondemand=7），直接通过手动录制的接口启动某条直播流的录制。

如果某条直播流正在录制（可能是自动录制，也可能是手动录制启动的），您可以通过手动录制的接口停止该直播流的录制。

使用流程

1. 调用AddLiveAppRecordConfig接口配置录制规则。设置ondemand=7，表示推流后不自动录制。
2. 推流。
3. 调用RealTimeRecordCommand接口启动录制。
4. （如果需要）调用RealTimeRecordCommand接口停止录制。

涉及到的API包括：

API名称	说明
AddLiveAppRecordConfig	配置自动录制规则。
DeleteLiveAppRecordConfig	如果不再需要某项规则，可以通过该接口删除。
DescribeLiveRecordConfig	查询所有配置的规则。
RealTimeRecordCommand	实时录制命令，可控制手动启动录制和手动停止录制。

实践示例

- 手动启动录制

如果直播流 `example.com/liveApp****/liveStream****` 正在推流，可以按以下方式调用 RealTimeRecordCommand接口启动录制：

```
/?AppName=liveApp****
&DomainName=example.com
&StreamName=liveStream****
&Command=start
&<公共请求参数>
```

● 手动停止录制

如果直播流 `example.com/liveApp****/liveStream****` 正在录制，可以按以下方式调用 `RealTimeRecordCommand` 接口停止录制：

```
/?AppName=liveApp****
&DomainName=example.com
&StreamName=liveStream****
&Command=stop
&<公共请求参数>
```

注意事项

- 如果直播流不存在（没有推流），调用 `RealTimeRecordCommand` 接口手动启动录制会返回失败。
- 手动启动录制的直播流如果发生了断流，就会停止录制，并且重新推流后不会自动启动录制（如果没有配置自动录制）。
- 视频直播支持触发拉流，当使用触发拉流域名对应的播放地址进行播放时，会自动触发阿里云视频直播服务进行拉流直播，无人播放时则不会回源站进行拉流。自动录制、按需录制、手动录制功能应用在触发拉流场景时，如果触发拉流无人播放，则不会回源站拉流，自动录制、按需录制、手动录制同样也不会进行录制。

7.5. 录制内容检索与管理

本文介绍录制内容检索的适用场景、使用流程、实践示例和注意事项。

适用场景

对于录制下来的文件，您可以：

- 设置回调，每当一个录制文件生成，都能收到一个HTTP请求，便于您做一些业务上的实时处理。
- 查询已录制的内容，可按时间轴展示录制内容覆盖的时间区间。
- 将已录制的TS片段按时间段生成一个M3U8索引（写入您指定的Bucket，可以直接播放）。

使用流程

直接调用相应接口。涉及到的API包括：

API名称	说明
<code>AddLiveRecordNotifyConfig</code>	配置某个域名下的录制回调，包括事件回调和按需录制回调。
<code>DeleteLiveRecordNotifyConfig</code>	如果不再需要某个域名的录制回调，可以通过该接口删除。
<code>DescribeLiveRecordNotifyConfig</code>	查询某个域名的录制回调

API名称	说明
UpdateLiveRecordNotifyConfig	更新某个域名的录制回调
录制事件回调	录制事件回调给您的HTTP消息定义
DescribeLiveStreamRecordContent	查询录制内容包含的时间段
CreateLiveStreamRecordIndexFiles	根据时间段创建录制索引
DescribeLiveStreamRecordIndexFiles	根据时间段查询录制索引
DescribeLiveStreamRecordIndexFile	查询某个录制索引的信息

实践示例

- 回调录制文件生成事件

调用AddLiveRecordNotifyConfig接口，设置域名 `aliyundoc.com` 下直播录制文件生成事件都回调到 `http://example.aliyundoc.com` 这个地址：

```
/?DomainName=aliyundoc.com
&NotifyUrl=http://example.aliyundoc.com
&<公共请求参数>
```

设置之后，如果一条直播流 `aliyundoc.com/live/teststream` 生成录制文件（TS/MP4/FLV），您就会收到以下内容的回调：

```
POST / HTTP/1.1
Host: live.example.com
User-Agent: Go-http-client/1.1
{
  "domain": "aliyundoc.com",
  "app": "live",
  "stream": "teststream",
  "uri": "live/teststream/0_2017-03-08-23:09:46_2017-03-08-23:10:40.flv",
  "duration": 69.403,
  "start_time": 1488985786,
  "stop_time": 1488985840
}
```

- 查询录制内容覆盖的时间段

假设您希望查找这条直播流 `aliyundoc.com/live/teststream` 在 `2019-02-15 09:00:00` 到 `2019-02-15 21:00:00` 之间的录制内容，可按如下方式调用DescribeLiveStreamRecordContent接口：

```
/?AppName=live
&DomainName=aliyundoc.com
&StreamName=teststream
&StartTime=2019-02-15T01:00:00Z
&EndTime=2019-02-15T13:00:00Z
&<公共请求参数>
```

- 生成某个时间段的M3U8索引

假设您已录制了 2019-02-15 09:00:00 到 2019-02-15 21:00:00 的内容，但是想将10点到11点的内容单独生成一段M3U8录制索引，那么您可以按如下方式调用CreateLiveStreamRecordIndexFiles接口：

```
/?AppName=live
&DomainName=aliyundoc.com
&StreamName=teststream
&StartTime=2019-02-15T02:00:00Z
&EndTime=2019-02-15T03:00:00Z
&<公共请求参数>
```

生成的M3U8文件将会上传到您指定的OSS Bucket中（保存路径可通过AddLiveAppRecordConfig接口修改）。

注意事项

- 以上接口中的StartTime和EndTime字段填的均是UTC时间，请注意和本地时区的对应。
- 录制文件检索信息在视频直播系统中只保存6个月（过期会删除检索信息，但是不会删除用户录制文件）。
- ts分片信息在视频直播系统中只保存3个月，创建手动m3u8文件只能选择最近3个月的录制内容（过期会删除ts检索信息，但是不会删除用户录制ts文件）。

8.HTTP配置

本文介绍HTTP消息头的配置方式。

HTTP消息头

HTTP消息头是指，在超文本传输协议HTTP（Hypertext Transfer Protocol）的请求和响应消息中，协议头部的组件。HTTP消息头准确描述了正在获取的资源、服务器或客户端的行为。

您可以在[视频直播控制台](#)的[域名管理](#)页面，选择需要配置的播流域名，单击[域名配置](#)，进入[直播管理 > HTTP头配置](#)页面，配置HTTP消息头，请参见[配置HTTP消息头](#)。

 **说明** 当您要进行跨域播放时，需要在HTTP消息头配置时，找到Access-Control-Allow-Origin，并配置为 * 或需要访问的完整域名，例如www.aliyundoc.com。需要注意的是，跨域配置仅支持配置为 * 或单个域名，如需多域名跨域访问请配置为 * 。

9.直播流媒体处理

9.1.直播转码

直播转码可以解决推流端码率较高或较低而导致播放效果不好等问题。通过阅读本文，您可以快速了解直播转码的使用场景、使用限制、转码功能、模板参数、转码规则和宽度自适应等功能。

使用场景

- 当推流端码率较高而客户端带宽资源有限会导致视频播放卡顿；当推流端码率较低时会导致视频播放效果不好。使用直播转码服务可以快速帮您解决这些问题。
- 播放器插件需要实现多码率切换。前端播放器插件设置码率切换，需要同一路推流同时提供多种码率的播流地址。视频直播的实时转码功能能为同一路推流同时提供多种码率的播流地址服务。
 - 使用H.264编码进行推流，实时转码成H.265编码直播流进行播放，可以降低带宽。
 - 使用H.265编码进行推流，实时转码成低码率H.265编码的直播流进行播放。
 - 为了解决H.265在浏览器不兼容问题需要实时转码成H.264编码的直播流进行播放。

使用限制

- 根据域名+AppName进行转码模板设置时，同一个模板里转码模板类型不能同时配置标准模板和窄带高清™模板。
- 直播中心为北京、上海、深圳、新加坡时，每个域名下最多支持50路转码并发流，也就是一个域名下最多同时输出50路转码流。其他直播中心每个域名下最多支持10路转码并发流。如果一路高清分辨率直播流输入，同时转码成标清和流畅分辨率进行输出播放，算2路转码流。
- 当提供的直播服务不能满足您当前业务需求时，请您[提交工单](#)与我们联系并详述您的需求，或联系您的客服经理。

模板参数

转码模板支持通用模板、自定义模板、RTS转码模板和多码率转码模板（部分直播中心）。

通用模板中的模板为系统预设模板包括转码模板的ID和转码输出分辨率。转码输出的码率都是系统预设不支持修改。如果通用模板不能满足您的要求，您可以使用自定义模板或RTS转码模板，自定义转码模板支持编码格式、分辨率、码率、帧率、视频GOP、触发转码、转码模板ID的设置。

通用转码和自定义转码分别提供标准模板和窄带高清™模板。RTS转码在标准模板和窄带高清™模板基础上提供RTS超低延时播放SDK所需的B帧和OPUS参数设置。

- 通用转码
 - 标准模板

模板名称	模板ID	分辨率（高、宽自适应）	码率（kbps）	转码账单中输出规格
流畅	lld	360	460	LD
标清	lsd	432	640	SD
高清	lhd	648	1200	SD
超清	lud	1080	2100	HD

○ 窄带高清™模板

模板名称	模板ID	分辨率（高、宽自适应）	码率（kbps）	转码账单中输出规格
流畅	ld	360	≤420	LD
标清	sd	432	≤580	SD
高清	hd	648	≤1100	SD
超清	ud	1080	≤1900	HD

 注意

- 转码计费根据分辨率档位（LD、SD、HD、2K、4K）决定。分辨率档位与通用模板中标准模板的模板ID（包括lld、lsd、lhd、lud）和窄带高清™模板的模板ID（包括ld、sd、hd、ud）无关，模板ID仅作为业务调用时的标识使用，请勿混淆。
- 控制台不支持分辨率超过1080P的自定义设置，如果您的场景需要2K、4K规格分辨率，请[提交工单](#)。

● 自定义转码

自定义转码的模板参数由用户自行定义。

 说明 自定义转码中窄带高清™模板暂不支持选择H.265编码格式，如果您需要，请您[提交工单](#)联系我们。

● RTS转码

RTS转码的模板参数由用户自行定义。

● 多码率转码

多码率转码的模板参数由用户自行定义。仅支持新加坡、日本、印尼直播中心的播流域名。

转码功能介绍

● 转码配置

转码配置可以使用以下两种配置方式：

○ 控制台配置：

具体操作，请参见[通用转码](#)、[自定义转码](#)、[RTS转码](#)、[多码率转码](#)。

API和SDK配置：

API和SDK提供以下接口供用户使用：

- [添加转码配置](#)
- [删除转码配置](#)
- [查询直播流转码配置](#)

Java SDK添加转码配置的代码示例如下：

```
IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<AccessKeyId>", "<AccessKeySecret>");
IAcsClient client = new DefaultAcsClient(profile);
AddLiveStreamTranscodeRequest addLiveStreamTranscodeRequest = new AddLiveStreamTranscodeRequest();
addLiveStreamTranscodeRequest.setDomain("<DomainName>");
addLiveStreamTranscodeRequest.setApp("<AppName>");
addLiveStreamTranscodeRequest.setTemplate("hd");
addLiveStreamTranscodeRequest.setSnapshot("no");
addLiveStreamTranscodeRequest.setRecord("no");
try {
    AddLiveStreamTranscodeResponse addLiveStreamTranscodeResponse = client.getAcsResponse(addLiveStreamTranscodeRequest);
    System.out.println(addLiveStreamTranscodeResponse.getRequestId());
    // todo something.
} catch (ServerException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (ClientException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

 **说明** 如果更改转码配置，需要进行重新推流后配置才可生效。

● 转码使用

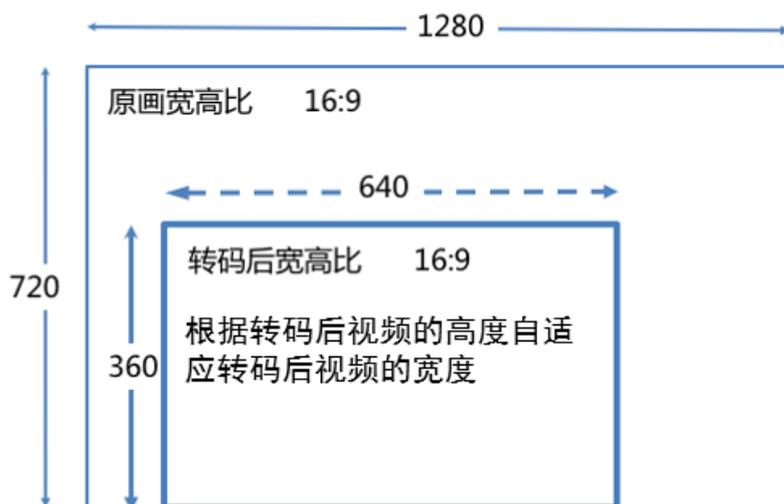
转码配置重新推流后提供多种码率的多条直播地址。您可以登录[视频直播控制台](#)，选择直播管理 > 流管理，在对应的推流记录中查看直播地址。其中原画（Original）即原始码率地址，自定义转码即直播转码后的地址，您可在播流地址上单击复制获取直播地址或单击播放直接播放。如果您需要在业务端代码中使用该地址，可直接在原始码率的 `StreamName` 后添加转码流。

转码规则

- 视频直播支持延时转码，系统监测到某一路直播流无人观看时，不会执行转码。检测到有人观看后启动转码，转码过程中5分钟内检测到无人观看即会停止转码。
- 每一路流仅执行一次转码。
- 转码不是必须服务，可按需配置。
- 转码目前支持FLV、RTMP、HLS、RTS格式。

宽度自适应

转码采用宽度自适应的算法，根据转码后视频的高度自适应转码后视频的宽度。



9.2. 直播时移

直播时移可以回看从直播开始时间到当前时间之间的直播视频。通过阅读本文，您可以了解直播时移的实现原理及请求方式。

实现原理

视频直播流会被切成TS切片，通过HLS协议向播放用户分发，用户请求的M3U8播放列表中包含不断刷新的TS切片地址。对于常规的HLS直播，TS切片地址及对应的TS文件无法保存，因此当前时间之前的视频直播内容无法进行回溯。在开启直播时移功能后，TS切片地址及对应的TS文件会分别在数据库和OSS中保存（最长配置30天，具体操作，请参见直播时移），可以回溯从直播开始时间到当前时间之间的视频。

使用限制

直播时移最大支持直播观看人数为10万人，如果需要支持更多人观看，请提交工单申请。

时移请求

一个完整的直播时移请求由HLS直播地址+时移参数组成，参数格式为 `lhs_{type}_{format}_{unit}_{zone}`，如下所示：

```
http://example.com/AppName/StreamName.m3u8?aliyunols=on&lhs_start_human_s_8=20171024160220&lhs_end_human_s_8=20171024160420
```

说明 `aliyunols=on` 为固定字段。

时移请求参数详情如下所示：

参数	描述
----	----

参数	描述
type	<p>时间类型。取值：</p> <ul style="list-style-type: none"> start：直播开始时间。 end：直播结束时间。 vodend：点播结束时间。 <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p>? 说明 设置vodend表示使用点播模式回看，一次返回指定时间段内所有的ts，包含endlist标签。</p> </div> <ul style="list-style-type: none"> offset：向前偏移的时间。
format	<p>时移的时间格式。取值：</p> <ul style="list-style-type: none"> unix：UNIX时间戳。 human：年月日时分秒。示例：20170809230130。
unit	<p>时移的时间单位。取值：</p> <ul style="list-style-type: none"> s：秒。 ms：毫秒。 us：微秒。 ns：纳秒。 <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p>? 说明 您可以在控制台中查询指定域名可时移的时间和数据量。更多信息，请参见用量查询。同时，您可以通过发送请求查询指定域名可时移的时间和数据量，详情请参见时移时间查询。</p> </div>
zone	<p>时区，取值：0~9，表示东*区。其中，0表示UNIX时间戳，8表示中国时区。</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p>? 说明 当format取值为unix时，zone取值为0。</p> </div>

时移参数示例如下所示：

- lhs_start_human_s_8=20170809200010
- lhs_start_unix_s_0=1502280113
- lhs_end_human_s_8=20170809200010
- lhs_vodend_unix_s_0=1502280113
- lhs_offset_unix_ms_0=1800000 （向前时移30分钟）

时移时间查询

直播时移时间查询请求由请求URL和参数组成。其中，请求URL格式为 `http://{domain}/openapi/timeline/query`，示例如下所示：

//不带鉴权

```
http://example.com/openapi/timeline/query?aliyunols=on&app=bbb&stream=cc&format=ts&lhs_end_unix_s_0=1513334270&lhs_start_unix_s_0=1513334255
```

//带鉴权

```
http://example.com/openapi/timeline/query?auth_key=1606204918-0-0-3e898a5f491fedb113d17aad7c091dde&aliyunols=on&app=zzb-test&stream=zzb-shiyi&format=ts&lhs_start_unix_s_0=1606200900&lhs_end_unix_s_0=1606202520
```

名称	描述
请求方式	GET
URL	请求URL，例如： <code>http://{domain}/openapi/timeline/query</code> ，其中， <code>{domain}</code> 为您的播流域名。
参数	<ul style="list-style-type: none"> aliyunols (required) : on。(固定字段) app (required) : 应用名。 stream (required) : 流名。 format (required) : 指定查询TS或FLV格式的时移数据。 lhs_start_unix_s_0 (required) : 开始时间的时间戳。单位：秒。 lhs_end_unix_s_0 (required) : 结束时间的时间戳。单位：秒。 <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p> 说明 如果您开启了CDN的鉴权，需要加上对应的鉴权参数 <code>auth_key</code>。<code>auth_key</code> 可以由控制台生成或通过代码自行拼接，详情请参见配置URL鉴权或鉴权代码示例。</p> </div>

返回示例如下所示：

```
{
  "retCode": 0,
  "description": "success",
  "content": {
    "current": 1514269063,
    "timeline": [
      {
        "start": 1514269054,
        "end": 1514269058
      },
      {
        "start": 1514269058,
        "end": 1514269062
      }
    ]
  }
}
```

参数	描述
current	当前系统时间，播放器可以基于这个字段对时。
timeline	有效的时移时间段，包含开始和结束时间戳。
start	有效片段的开始时间（UNIX时间戳），单位：秒。
end	有效片段的结束时间（UNIX时间戳），单位：秒。

9.3. 直播录制

视频直播录制可对当前直播实时录制并可回放历史直播视频。本文将详细为您介绍视频直播录制功能。

功能介绍

直播录制提供了录制直播内容、录制索引管理、录制文件管理、录制事件回调的功能。

- 录制直播内容

- 使用场景

视频直播是通过播放地址以实时流的形式播放推流端的数据，如果想要在直播结束后再次观看或修改直播内容，就必须使用视频直播的录制功能。

- 功能介绍

视频直播录制功能，就是将直播中心接收到的推流数据进行录制，将录制后的数据以HLS协议的封装格式，保存至您指定的存储位置。支持M3U8（同时会有.ts分片文件）、MP4、FLV格式。

直播录制支持将直播内容录制至VOD或OSS。录制到VOD可以联动使用VOD的云剪辑、媒资管理、播控管理等能力。

- 使用方法

请参见[直播录制使用方法](#)。

- 录制索引管理

 **说明** 您只有将视频存储在OSS中才可使用录制索引功能。

- 使用场景

视频直播录制在直播过程中会生成TS文件，但是仅在推流结束180秒后才会生成m3u8索引文件，无法在直播过程中生成，并且该索引文件时间为推流开始到结束，用户无法根据业务需要自定义索引范围。因此，视频直播提供录制索引管理功能，供用户对录制索引m3u8文件进行管理。

- 功能介绍

在一次推流结束时，会自动生成本次推流的录制索引文件。同时支持按照指定的录制开始时间和录制结束时间生成自定义录制索引文件。录制索引功能包括查看录制索引和剪辑录制索引两项功能。

- 查看录制索引：用户可以查看已有的录制索引文件，这里的索引文件包括系统在断流180秒后自动生成的索引文件，以及用户手动创建的索引文件。
- 剪辑录制索引：用户可以根据现有的录制索引文件进行剪辑，产生新的索引文件，这里的索引文件可以是对之前已有的索引文件进行合并或者拆分。

○ 使用方法

请参见[录制索引管理方法](#)。

● 录制文件管理

○ 功能介绍

- 查看和管理录制文件
- 过期删除录制文件

设置过期时间，超过配置的过期时间后，录制文件即可自动删除。

 **说明** 此功能默认不向用户开放，添加AliyunMTSVideoLifecycleRolePolicy授权后，再提交工单申请。

○ 使用方法

请参见[录制文件管理方法](#)。

● 录制事件回调

○ 功能介绍

直播录制回调包括录制状态回调、文件生成回调。

- 录制状态回调：在文件开始录制与结束录制时发出，主要表示文件已经开始和结束录制。
- 文件生成回调：在文件生成时发出，主要包括文件名称，开始结束时间与时长信息。

○ 使用方法

请参见[直播录制回调](#)。

直播录制规则

- 直播录制是针对推流的 `AppName` 为粒度设置的。您可以设置某个 `AppName` 下的所有 `StreamName` 的录制规则。并且 `AppName` 取值为星号 (*) 时，表示此录制规则适用于该直播域名下的所有推流。推流的 `AppName` 需要和配置的 `AppName` 相同。也可以针对单路流进行录制。

 **说明** 同时配置了 `AppName` 级别的录制和单路流录制，单路流录制配置的优先级更高。

- 录制设置仅会对配置完成后的新发起的推流记录生效，当前的推流需中断180秒后重新推流方可生效。
- 录制至VOD时，生成的M3U8和TS文件都存放在视频点播的输入媒体Bucket中。更多详情。请参见[添加媒体Bucket](#)。
- 如果使用录制至OSS，您需要事先创建好OSS Bucket。OSS Bucket必须与域名的直播中心同地域，不支持跨地域录制。
- 相同的 `AppName` 和 `StreamName` 不能同时存储至VOD和OSS。

直播录制使用方法

录制至...	使用流程	控制台操作	API操作
	准备工作：开通点播服务。	开通视频点播	不涉及

录制至...	使用流程	控制台操作	API操作
录制至VOD	录制配置： <ol style="list-style-type: none"> 1. 创建直播转点播录制模板。 2. 在点播服务中创建存储规则。 	<ol style="list-style-type: none"> 1. 录制存储至VOD 2. 视频点播转码配置 	AddLiveRecordVodConfig
	使用点播功能进行云剪辑、媒资管理、播控等操作。	控制台介绍	API概览
录制至OSS	准备工作：完成OSS配置。 <ol style="list-style-type: none"> 1. 创建OSS Bucket。 2. 配置直播写入OSS权限。 3. 配置CDN域名。 	配置OSS	不涉及
	录制配置： 配置录制信息。	录制存储至OSS	AddLiveAppRecordConfig

以下为使用Java SDK添加App录制配置的示例代码：

```

IClientProfile profile = DefaultProfile.getProfile("cn-shanghai", "<AccessKeyId>", "<AccessKeySecret>");
IAcsClient client = new DefaultAcsClient(profile);
AddLiveAppRecordConfigRequest addLiveAppRecordConfigRequest = new AddLiveAppRecordConfigRequest();
addLiveAppRecordConfigRequest.setDomainName("<DomainName>");
addLiveAppRecordConfigRequest.setAppName("<AppName>");
addLiveAppRecordConfigRequest.setOssEndpoint("<Endpoint>");
addLiveAppRecordConfigRequest.setOssBucket("<BucketName>");
addLiveAppRecordConfigRequest.setOssObjectPrefix("<ObjectPrefix>");
try {
    AddLiveAppRecordConfigResponse addLiveAppRecordConfigResponse = client.getAcsResponse(addLiveAppRecordConfigRequest);
    System.out.println(addLiveAppRecordConfigResponse.getRequestId());
    // todo something.
} catch (ServerException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (ClientException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
    
```

录制索引管理方法

功能	控制台操作	API操作
----	-------	-------

功能	控制台操作	API操作
查看录制索引	<p>在视频直播控制台的直播管理 > 录制文件管理页面，单击存储至OSS页签查看。</p> <p>这里的索引文件包括系统在断流180秒后自动生成的索引文件，以及用户手动创建的索引文件。</p>	<ul style="list-style-type: none"> 查询单个录制索引文件：<code>DescribeLiveStreamRecordIndexFile</code>。 查询某个时间段内的所有录制索引文件：<code>DescribeLiveStreamRecordIndexFiles</code>。
剪辑录制索引	剪辑录制索引文件	<code>CreateLiveStreamRecordIndexFiles</code>

录制文件管理方法

查看和管理录制文件：您可以在控制台进行查看和管理录制文件的操作。

- 录制至VOD
 - 在视频直播控制台查看和管理，请参见[录制文件管理](#)。
 - 在视频点播控制台查看和管理
 - a. 登录**视频点播控制台**，在左侧导航栏选择媒资库 > 音/视频。
 - b. 选择您存至VOD的录制视频，并单击右侧的管理。
 - c. 在**基础信息**页签中，您可以查看或修改视频的标题、描述、分类、标签、封面等信息。信息编辑完成后，单击**保存完成配置**。
- 录制至OSS
 - 在视频直播控制台查看和管理，请参见[录制文件管理](#)。
 - 在OSS控制台查看和管理：录制文件按照规则存储在OSS中，您可在OSS控制台中找到存储视频文件夹，并获取视频的播放地址。
 - a. 登录**OSS控制台**，在左侧Bucket列表中，选择您存储录制视频的Bucket。
 - b. 单击**文件管理**，选择存储录制视频的文件夹名称。
 视频完整的文件存储在 `record/{AppName}/{StreamName}` 下，与TS切片文件存储在一个文件夹中。
 - c. 选择您想要查看的录制视频，并单击右侧的**详情**即可获得录制视频的播放地址。

API和SDK提供[DescribeLiveStreamRecordContent](#)接口查询直播录制内容。

 **说明** 存储在OSS Bucket的录制文件，可以使用OSS的API进行生命周期管理。

过期删除录制文件：您需要按照如下步骤添加一个授权后，再提交工单申请。

1. 登录**访问控制 RAM 控制台**
2. 在左侧导航栏选择**RAM角色管理**。
3. 单击**创建RAM角色**，当前可信实体类型选择**阿里云服务**，单击下一步。
4. 在配置角色页签：
 - 角色类型：选择**普通服务角色**；
 - 角色名称：填写 `AliyunMTSVideoLifecycleRole`（请您务必保持名称一致）；
 - 选择受信服务：选择**多媒体转码服务**。

单击完成。

5. 在创建完成页签单击精准授权。

6. 在添加权限页签，添加权限

- 权限类型：选择系统策略；
- 输入策略名称：输入 `AliyunMTSVideoLifecycleRolePolicy`。

单击确定。

7. 请您[提交工单](#)。我们会在后台为您配置您需要的过期时间。配置过期时间后，录制文件过期删除功能授权操作即为完成。

9.4. 直播截图

直播截图提供实时截图功能，满足您在观看直播同时的截图需求。本文详细为您介绍视频直播截图功能。

前提条件

- 视频直播截图功能需要将截图文件存放在媒体Bucket中，因此需要开通OSS服务并且将Bucket设置为媒体Bucket。更多详情，请参见[开通OSS服务](#)和[添加媒体Bucket](#)。
- 配置截图模版后，需要重新推流才生效；新配置的截图模版对正在进行的直播流不生效。

直播截图规则

- 直播截图是针对推流的 `AppName` 为粒度设置的。您可以设置某个 `AppName` 下的所有 `StreamName` 的截图规则。当 `AppName` 取值为星号 (*) 时，表示此截图规则适用于该直播域名下的所有推流。推流的 `AppName` 需要和配置的 `AppName` 相同。
- 直播截图功能配置时，`ObjectName`可以定义为覆盖截图和实时截图两种类型。
 - 覆盖截图是每次新生成截图覆盖之前生成的截图文件。
 - 实时截图是每次新生成的截图不覆盖之前生成的截图文件，截图递增存储。截图文件使用 `{Sequence}` 区分，`{Sequence}` 通过1, 2,, n的方式表示。
 - 目前直播截图仅支持生成JPG格式。
- 使用API和SDK配置截图时，`OverwriteOssObject` 参数和 `SequenceOssObject` 参数必须配置其中之一，暂无默认值设置。

直播截图功能介绍

- 视频截图功能配置
配置视频直播截图功能可以通过以下两种方法：
 - 通过控制台配置。具体操作，请参见[配置截图](#)。

- 通过API和SDK配置。API和SDK提供了以下截图配置接口：
 - [添加直播截图配置](#)
 - [查询直播截图配置](#)
 - [更新直播截图配置](#)

以下为Java SDK添加截图配置的示例代码：

```
IClientProfile profile = DefaultProfile.getProfile("cn-shanghai", "<AccessKeyId>", "<AccessKeySecret>");
IAcsClient client = new DefaultAcsClient(profile);
AddLiveAppSnapshotConfigRequest addLiveAppSnapshotConfigRequest = new AddLiveAppSnapshotConfigRequest();
addLiveAppSnapshotConfigRequest.setDomainName("<DomainName>");
addLiveAppSnapshotConfigRequest.setAppName("<AppName>");
addLiveAppSnapshotConfigRequest.setTimeInterval(5);
addLiveAppSnapshotConfigRequest.setOssEndpoint("<Endpoint>");
addLiveAppSnapshotConfigRequest.setOssBucket("<BucketName>");
addLiveAppSnapshotConfigRequest.setOverwriteOssObject("{AppName}/{StreamName}.jpg");
;
try {
    AddLiveAppSnapshotConfigResponse addLiveAppSnapshotConfigResponse = client.getAcsResponse(addLiveAppSnapshotConfigRequest);
    System.out.println(addLiveAppSnapshotConfigResponse.getRequestId());
    // todo something.
} catch (ServerException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (ClientException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

● 视频截图管理

截图配置完成后通过新发起的推流生成截图，查看截图的方法包括以下两种：

- 您可以在[视频直播控制台](#)和[OSS控制台](#)中查看截图文件。截图文件存储在OSS的Bucket中，可以通过OSS Bucket管理视频截图。更多详情，请参见[截图管理](#)。

 **说明** 您只有在OSS控制台才能删除截图。

- API和SDK提供查询截图信息的接口。更多详情，请参见[查询截图内容](#)。

10. 云端合流

10.1. 云端合流概述

阿里云视频直播为您提供云端合流服务，本文介绍云端合流的功能特性和使用方法。

您可以调用开放接口，通过直播中心的合流服务器拉取多个视频源，依靠云端强大的处理能力将多路视频源按照您设置的布局进行合流，合成一路直播流后，通过边缘加速分发给观众观看。

说明

- 云端合流目前处于公测阶段。无需开通即可使用，且暂不收费。
- 每个UID下并发合流任务上限为10路，如需扩充请[提交工单](#)申请。
- 云端合流目前仅支持华北2和华东2地区。

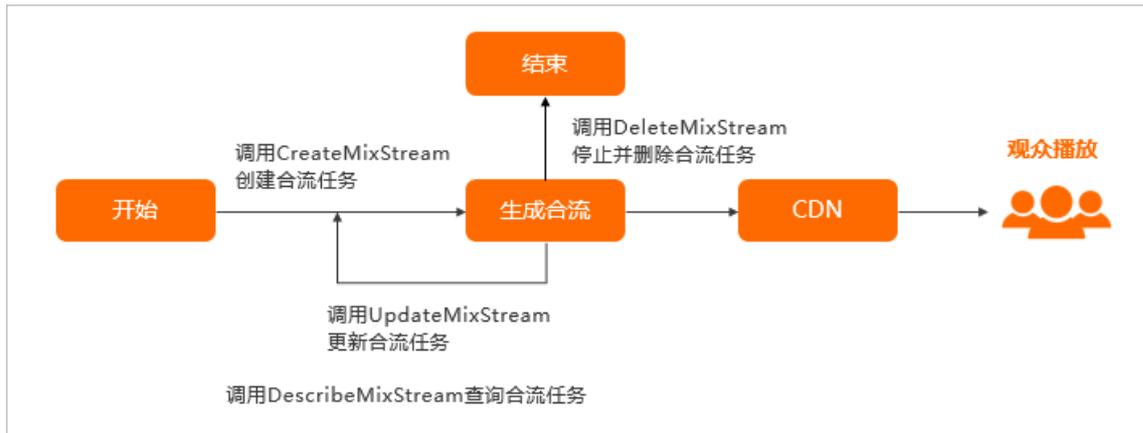
功能特性

功能	描述	特性
多路合流	将多路视频源推至直播中心，合流为一路直播流。	<ul style="list-style-type: none"> • 支持添加RTMP、FLV和ARTC协议的直播流。 <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p> 说明 超低延时直播使用协议为ARTC。</p> </div> <ul style="list-style-type: none"> • 支持添加文件流。 • 最多添加8个视频源。
预设布局	提供预设的合成直播流画面布局，满足不同场景下的直播。	提供7种预设布局，布局请参见 合流预设布局参考 。
自定义布局	根据需要，添加自定义合成后直播流的布局。	支持设置窗格个数、尺寸和位置。
合流输出	将合流的多路视频源以直播流的形式输出。	支持自定义合流输出域名，可用该域名观看合流后的内容。
合流查询	查询当前合流任务列表及相应配置信息。	支持实时查询。
合流更新	在合流观看过程中更新视频源、合流布局，且无缝切换。	不支持更新输出分辨率等内容。
自动删除	自动删除合流任务。	<p>支持在创建合流任务时设定自动删除时间，到期合流任务自动删除以节省流量，默认为7天。</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p> 注意 在合流创建成功后不可再修改。</p> </div>

功能	描述	特性
合流回调	合流任务状态回调。	支持在创建合流时向您的回调地址发送创建成功或创建失败的消息，以便让您及时获取合流任务创建的结果。

使用方法

视频直播支持通过调用API接口的方式，使用云端合流服务，包括创建合流、更新合流、查询合流和删除合流。调用次序如下：



具体调用方法和参数请参见如下API文档：

场景	描述	接口	代码示例
创建合流（预设布局）	<p>使用预设布局模板创建合流。</p> <p>以MixStreamLayout-2-1模板为例，最终效果为：</p> 	<p>CreateMixStream</p> <ul style="list-style-type: none"> 传入输入源配置（个数、类型、地址、叠加顺序、尺寸、位置、填充方式）。 传入合流播放地址（域名、Appname、Streamname）。 传入合流输出分辨率。 	<p>以MixStreamLayout-2-1模板为例，代码示例请参见合流代码示例。</p>

场景	描述	接口	代码示例
创建合流（自定义布局）	<p>使用自定义布局创建合流。</p> <p>以两路输入流为例，最终效果为：</p> 	<p>CreateMixStream</p> <ul style="list-style-type: none"> 传入输入源配置（个数、类型、地址、叠加顺序、尺寸、位置、填充方式）。 传入合流播放地址（域名、Appname、Streamname）。 传入合流输出分辨率。 	<p>以两路输入流为例，代码示例请参见合流代码示例。</p>
更新合流	<p>更新合流的视频源或布局。</p>	<p>UpdateMixStream</p> <p>传入输入源配置（个数、类型、地址、叠加顺序、尺寸、位置、填充方式）。</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff;"> <p>说明</p> <p>UpdateMixStream的参数为全量参数，调用时需要传入更新后的全量配置。</p> <p>例如：原合流为A、B、C、D共四路视频源合流，目标为删除其中C、D两路视频源，那么需要调用UpdateMixStream传入A、B两路视频源和相关配置。</p> </div>	<p>使用自定义布局更新合流，代码示例请参见更新合流任务。</p>
查询合流	<p>查询在线合流列表和相关配置。</p>	<p>DescribeMixStreamList</p> <ul style="list-style-type: none"> 返回在线合流列表。 返回合流配置。 返回合流创建时间、修改时间。 返回合流播放地址。 	<p>查询合流任务。</p>
删除合流	<p>当您不需要使用某条合流后，请删除此条合流，否则合流将一直处于输出状态。</p>	<p>DeleteMixStream</p>	<p>删除合流任务。</p>

开发支持

您在使用合流服务过程中有任何问题或建议，请通过搜索群号34935990加入钉钉群，或扫描下方二维码加入。



10.2. 合流预设布局参考

云端合流提供了7种预设布局，本文介绍预设布局的布局参数和样式。

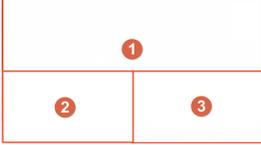
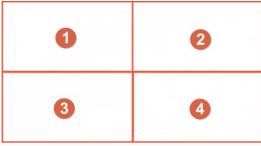
预设布局

云端合流为您提供7种预设布局以满足不同直播场景，使用时传入相应布局ID即可，无需额外配置。具体布局样式如下：

② 说明

- FillSizeNormalized：表示该Layer元素需要填充的尺寸大小。
[w,h]：w、h取值范围[0~1]。其中宽高都是进行了归一化计算。
例如：[0.2,0.3]代表水平占比20%，垂直占比30%。
- FillPositionNormalized：表示该Layer元素填充区位置归一化值。
[x,y]：x、y的取值范围[0~1]。
例如：[0.1,0.1]代表左上角水平偏移10%，垂直偏移10%。

布局ID	描述	布局方格对应ID	样式
MixStreamLayout-1-1	<p>1路视频源。布局参数如下：</p> <ul style="list-style-type: none"> • FillSizeNormalized: [1,1] • FillPositionNormalized: [0,0] 		
MixStreamLayout-2-1	<p>2路视频源。布局参数如下：</p> <p>第1路：</p> <ul style="list-style-type: none"> • FillSizeNormalized: [0.5,1] • FillPositionNormalized: [0,0] <p>第2路：</p> <ul style="list-style-type: none"> • FillSizeNormalized: [0.5,1] • FillPositionNormalized: [0.5,0] 		
MixStreamLayout-2-2	<p>2路视频源。布局参数如下：</p> <p>第1路：</p> <ul style="list-style-type: none"> • FillSizeNormalized: [1,1] • FillPositionNormalized: [0,0] <p>第2路：</p> <ul style="list-style-type: none"> • FillSizeNormalized: [0.25,0.25] • FillPositionNormalized: [0.7,0.7] 		
MixStreamLayout-2-3	<p>2路视频源。布局参数如下：</p> <p>第1路：</p> <ul style="list-style-type: none"> • FillSizeNormalized: [1,1] • FillPositionNormalized: [0,0] <p>第2路：</p> <ul style="list-style-type: none"> • FillSizeNormalized: [0.25,0.25] • FillPositionNormalized: [0.7,0.05] 		

布局ID	描述	布局方格对应ID	样式
MixStreamLayout-3-1	<p>3路视频源。布局参数如下：</p> <p>第1路：</p> <ul style="list-style-type: none"> • FillSizeNormalized: [1,1] • FillPositionNormalized: [0,0] <p>第2路：</p> <ul style="list-style-type: none"> • FillSizeNormalized: [0.5,0.5] • FillPositionNormalized: [0,0.5] <p>第3路：</p> <ul style="list-style-type: none"> • FillSizeNormalized: [0.5,0.5] • FillPositionNormalized: [0.5,0.5] 		
MixStreamLayout-3-2	<p>3路视频源。布局参数如下：</p> <p>第1路：</p> <ul style="list-style-type: none"> • FillSizeNormalized: [1,1] • FillPositionNormalized: [0,0] <p>第2路：</p> <ul style="list-style-type: none"> • FillSizeNormalized: [0.3,0.3] • FillPositionNormalized: [0.1,0.6] <p>第3路：</p> <ul style="list-style-type: none"> • FillSizeNormalized: [0.3,0.3] • FillPositionNormalized: [0.6,0.6] 		
MixStreamLayout-4-1	<p>4路视频源。布局参数如下：</p> <p>第1路：</p> <ul style="list-style-type: none"> • FillSizeNormalized: [0.5,0.5] • FillPositionNormalized: [0,0] <p>第2路：</p> <ul style="list-style-type: none"> • FillSizeNormalized: [0.5,0.5] • FillPositionNormalized: [0.5,0] <p>第3路：</p> <ul style="list-style-type: none"> • FillSizeNormalized: [0.5,0.5] • FillPositionNormalized: [0,0.5] <p>第4路：</p> <ul style="list-style-type: none"> • FillSizeNormalized: [0.5,0.5] • FillPositionNormalized: [0.5,0.5] 		

10.3. 合流代码示例

通过阅读本文，您可以查看合流代码的示例。

示例说明

本文以Java语言为例进行介绍，获取其他语言SDK，请参见[视频直播SDK](#)。

SDK引入

1. 引入阿里云核心SDK和视频直播SDK。此处版本仅供参考，获取最新的版本请参见[视频直播SDK](#)。

- i. 引入阿里云核心SDK。

```
<dependencies>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <version>4.5.17</version>
  </dependency>
</dependencies>
```

- ii. 引入视频直播SDK。

```
<dependencies>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-live</artifactId>
    <version>3.9.4</version>
  </dependency>
</dependencies>
```

2. 初始化IAcsClient实例。

视频直播SDK通过IAcsClient实例完成OpenAPI的调用。因此，在发起调用前需要先初始化IAcsClient实例。

```
public void init() throws ClientException {
    DefaultProfile profile = DefaultProfile.getProfile(REGION_ID, ACCESS_KEY_ID, ACCESS_KEY_SECRET);
    client = new DefaultAcsClient(profile);
}
```

3. 查看合流服务示例代码。

- o 创建合流任务（使用预设布局）

```
public void createDefault() throws ClientException {
    CreateMixStreamRequest createMixStreamRequest = new CreateMixStreamRequest();
    // 设置拉流域名(直播播流域名, 即用该域名观看合流后的内容)
    createMixStreamRequest.setDomainName("your domain name");
    // 设置预设的layoutId, 以"MixStreamLayout-2-1"为例
    createMixStreamRequest.setLayoutId("MixStreamLayout-2-1");
    // 设置输出配置
    JSONObject outputConfig = new JSONObject();
    outputConfig.put("AppName", "your app name"); // AppName
    outputConfig.put("StreamName", "your stream name"); // StreamName
    outputConfig.put("MixStreamTemplate", "lp_ld"); // 合流输出规格模板
    String outputConfigStr = outputConfig.toJSONString();
    createMixStreamRequest.setOutputConfig(outputConfigStr);
    // 初始化输入流
    JSONArray inputStreamList = new JSONArray();
    // 构建输入流A
    JSONObject inputStreamA = new JSONObject();
    inputStreamA.put("LayoutChildId", 1); // 布局id, 注意: 一定要从1开始, 并且所有id要连续
    inputStreamA.put("ResourceType", "url"); // 资源类型: url 或 live
    inputStreamA.put("ResourceValue", "your resource value"); // 资源地址
    inputStreamList.add(inputStreamA);
    // 构建输入流B(方法同A)
    JSONObject inputStreamB = new JSONObject();
    inputStreamB.put("LayoutChildId", 2);
    inputStreamB.put("ResourceType", "url");
    inputStreamB.put("ResourceValue", "your resource value");
    inputStreamList.add(inputStreamB);
    // 设置输入流
    String inputStreamListStr = inputStreamList.toJSONString();
    createMixStreamRequest.setInputStreamList(inputStreamListStr);
    // 发起openapi请求
    CreateMixStreamResponse createMixStreamResponse = client.getAcsResponse(createMixStreamRequest);
    // 打印合流id
    String mixStreamId = createMixStreamResponse.getMixStreamId();
    System.out.println(mixStreamId);
}
```

- 创建合流任务（自定义布局）

```
public void createMixStream() throws ClientException {
    CreateMixStreamRequest createMixStreamRequest = new CreateMixStreamRequest();
    // 设置拉流域名(直播播流域名, 即用该域名观看合流后的内容)
    createMixStreamRequest.setDomainName("your domain name");
    // 设置布局id, 自定义布局填写"USERDEFINED"
    createMixStreamRequest.setLayoutId("USERDEFINED");
    // 设置输出配置
    JSONObject outputConfig = new JSONObject();
    outputConfig.put("AppName", "your app name"); // AppName
    outputConfig.put("StreamName", "your stream name"); // StreamName
    outputConfig.put("MixStreamTemplate", "lp_ld"); // 合流输出规格模板
    String outputConfigStr = outputConfig.toJSONString();
    createMixStreamRequest.setOutputConfig(outputConfigStr);
    // 初始化输入流
    JSONArray inputStreamList = new JSONArray();
    // 构建输入流A
    JSONObject inputStreamA = new JSONObject();
    JSONObject layoutConfigA = new JSONObject(); // 自定义布局配置设置
    Float[] fillPositionNormalizedA = new Float[]{0f, 0f}; // 位置坐标
    Float[] fillSizeNormalizedA = new Float[]{1f, 1f}; // 大小归一化值
    layoutConfigA.put("FillMode", "fit");
    layoutConfigA.put("PositionRefer", "topLeft");
    layoutConfigA.put("FillPositionNormalized", fillPositionNormalizedA);
    layoutConfigA.put("FillSizeNormalized", fillSizeNormalizedA);
    inputStreamA.put("LayoutChildId", 1); // 布局id, 注意: 一定要从1开始, 并且所有id要连续
    inputStreamA.put("ResourceType", "url"); // 资源类型: url 或 live
    inputStreamA.put("ResourceValue", "your resource value"); // 资源地址
    inputStreamA.put("LayoutConfig", layoutConfigA); // 设置自定义的布局配置
    inputStreamList.add(inputStreamA);
    // 构建输入流B(方法同A)
    JSONObject inputStreamB = new JSONObject();
    JSONObject layoutConfigB = new JSONObject();
    Float[] fillPositionNormalizedB = new Float[]{0.7f, 0.03f};
    Float[] fillSizeNormalizedB = new Float[]{0.25f, 0.25f};
    layoutConfigB.put("FillMode", "fit");
    layoutConfigB.put("PositionRefer", "topLeft");
    layoutConfigB.put("FillPositionNormalized", fillPositionNormalizedB);
    layoutConfigB.put("FillSizeNormalized", fillSizeNormalizedB);
    inputStreamB.put("LayoutChildId", 2);
    inputStreamB.put("ResourceType", "url");
    inputStreamB.put("ResourceValue", "your resource value");
    inputStreamB.put("LayoutConfig", layoutConfigB);
    inputStreamList.add(inputStreamB);
    // 设置输入流
    String inputStreamListStr = inputStreamList.toJSONString();
    createMixStreamRequest.setInputStreamList(inputStreamListStr);
    // 发起openapi请求
    CreateMixStreamResponse createMixStreamResponse = client.getAcsResponse(createMixStreamRequest);
    // 打印合流id
    String mixStreamId = createMixStreamResponse.getMixStreamId();
    System.out.println(mixStreamId);
}
```

- 更新合流任务

```

public void updateMixStream() throws ClientException {
    UpdateMixStreamRequest updateMixStreamRequest = new UpdateMixStreamRequest();
    // 设置拉流域名(直播播流域名, 即用该域名观看合流后的内容)
    updateMixStreamRequest.setDomainName("your domain name");
    // 设置布局id, 自定义布局填写"USERDEFINED"
    updateMixStreamRequest.setLayoutId("USERDEFINED");
    // 设置要更新的合流id, 即之前创建过的合流任务
    updateMixStreamRequest.setMixStreamId("your mixstreamId");
    // 初始化输入流
    JSONArray inputStreamList = new JSONArray();
    // 构建输入流A
    JSONObject inputStreamA = new JSONObject();
    JSONObject layoutConfigA = new JSONObject(); // 自定义布局配置设置
    Float[] fillPositionNormalizedA = new Float[]{0f, 0f}; // 位置坐标
    Float[] fillSizeNormalizedA = new Float[]{1f, 1f}; // 大小归一化值
    layoutConfigA.put("FillMode", "fit");
    layoutConfigA.put("PositionRefer", "topLeft");
    layoutConfigA.put("FillPositionNormalized", fillPositionNormalizedA);
    layoutConfigA.put("FillSizeNormalized", fillSizeNormalizedA);
    inputStreamA.put("LayoutChildId", 1); // 布局id, 注意: 一定要从1开始, 并且所有id要连续
    inputStreamA.put("ResourceType", "url"); // 资源类型: url 或 live
    inputStreamA.put("ResourceValue", "your resource value"); // 资源地址
    inputStreamA.put("LayoutConfig", layoutConfigA); // 设置自定义的布局配置
    inputStreamList.add(inputStreamA);
    // 构建输入流B(方法同A)
    JSONObject inputStreamB = new JSONObject();
    JSONObject layoutConfigB = new JSONObject();
    Float[] fillPositionNormalizedB = new Float[]{0.5f, 0.5f};
    Float[] fillSizeNormalizedB = new Float[]{0.3f, 0.3f};
    layoutConfigB.put("FillMode", "fit");
    layoutConfigB.put("PositionRefer", "topLeft");
    layoutConfigB.put("FillPositionNormalized", fillPositionNormalizedB);
    layoutConfigB.put("FillSizeNormalized", fillSizeNormalizedB);
    inputStreamB.put("LayoutChildId", 2);
    inputStreamB.put("ResourceType", "url");
    inputStreamB.put("ResourceValue", "your resource value");
    inputStreamB.put("LayoutConfig", layoutConfigB);
    inputStreamList.add(inputStreamB);
    // 设置输入流
    String inputStreamListStr = inputStreamList.toJSONString();
    updateMixStreamRequest.setInputStreamList(inputStreamListStr);
    // 发起openapi请求
    UpdateMixStreamResponse updateMixStreamResponse = client.getAcsResponse(updateMixStreamRequest);
    String mixStreamId = updateMixStreamResponse.getMixStreamId();
    // 打印合流id
    System.out.println(mixStreamId);
}

```

- 查询合流任务

```
public void describeMixStreamList() throws ClientException {
    DescribeMixStreamListRequest describeMixStreamListRequest = new DescribeMixStream
ListRequest();
    // 设置拉流域名(直播播流域名, 即用该域名观看合流后的内容)
    describeMixStreamListRequest.setDomainName("your domain name");
    // 设置appName
    describeMixStreamListRequest.setAppName("your app name");
    // 设置查询开始时间
    describeMixStreamListRequest.setStartTime("2020-12-03T14:00:00Z");
    // 设置查询结束时间
    describeMixStreamListRequest.setEndTime("2020-12-05T14:00:00Z");
    // 发起openapi请求
    DescribeMixStreamListResponse describeMixStreamListResponse = client.getAcrespon
se(describeMixStreamListRequest);
    // 获取合流列表
    List<MixStreamInfo> result = describeMixStreamListResponse.getMixStreamList();
    // 打印合流id
    if (result != null && !result.isEmpty()) {
        for (MixStreamInfo mixStreamInfo : result) {
            String mixStreamId = mixStreamInfo.getMixstreamId();
            System.out.println(mixStreamId);
        }
    }
}
```

o 删除合流任务

```
public void deleteMixStream() throws ClientException {
    DeleteMixStreamRequest deleteMixStreamRequest = new DeleteMixStreamRequest();
    // 设置拉流域名(直播播流域名, 此前用该域名观看合流后的内容)
    deleteMixStreamRequest.setDomainName("your domain name");
    // 设置appName
    deleteMixStreamRequest.setAppName("your app name");
    // 设置流名
    deleteMixStreamRequest.setStreamName("your stream name");
    // 设置要删除的合流id
    deleteMixStreamRequest.setMixStreamId("your mixstreamId");
    // 发起openapi请求
    DeleteMixStreamResponse deleteMixStreamResponse = client.getAcresponse(deleteMix
StreamRequest);
    String mixStreamId = deleteMixStreamResponse.getMixStreamId();
    // 打印合流id
    System.out.println(mixStreamId);
}
```

更新合流任务

```
public void updateMixStream() throws ClientException {
    UpdateMixStreamRequest updateMixStreamRequest = new UpdateMixStreamRequest();
    // 设置拉流域名(直播播流域名,即用该域名观看合流后的内容)
    updateMixStreamRequest.setDomainName("your domain name");
    // 设置布局id,自定义布局填写"USERDEFINED"
    updateMixStreamRequest.setLayoutId("USERDEFINED");
    // 设置要更新的合流id,即之前创建过的合流任务
    updateMixStreamRequest.setMixStreamId("your mixstreamId");
    // 初始化输入流
    JSONArray inputStreamList = new JSONArray();
    // 构建输入流A
    JSONObject inputStreamA = new JSONObject();
    JSONObject layoutConfigA = new JSONObject(); // 自定义布局配置设置
    Float[] fillPositionNormalizedA = new Float[]{0f, 0f}; // 位置坐标
    Float[] fillSizeNormalizedA = new Float[]{1f, 1f}; // 大小归一化值
    layoutConfigA.put("FillMode", "fit");
    layoutConfigA.put("PositionRefer", "topLeft");
    layoutConfigA.put("FillPositionNormalized", fillPositionNormalizedA);
    layoutConfigA.put("FillSizeNormalized", fillSizeNormalizedA);
    inputStreamA.put("LayoutChildId", 1); // 布局id,注意:一定要从1开始,并且所有id要连续
    inputStreamA.put("ResourceType", "url"); // 资源类型: url 或 live
    inputStreamA.put("ResourceValue", "your resource value"); // 资源地址
    inputStreamA.put("LayoutConfig", layoutConfigA); // 设置自定义的布局配置
    inputStreamList.add(inputStreamA);
    // 构建输入流B(方法同A)
    JSONObject inputStreamB = new JSONObject();
    JSONObject layoutConfigB = new JSONObject();
    Float[] fillPositionNormalizedB = new Float[]{0.5f, 0.5f};
    Float[] fillSizeNormalizedB = new Float[]{0.3f, 0.3f};
    layoutConfigB.put("FillMode", "fit");
    layoutConfigB.put("PositionRefer", "topLeft");
    layoutConfigB.put("FillPositionNormalized", fillPositionNormalizedB);
    layoutConfigB.put("FillSizeNormalized", fillSizeNormalizedB);
    inputStreamB.put("LayoutChildId", 2);
    inputStreamB.put("ResourceType", "url");
    inputStreamB.put("ResourceValue", "your resource value");
    inputStreamB.put("LayoutConfig", layoutConfigB);
    inputStreamList.add(inputStreamB);
    // 设置输入流
    String inputStreamListStr = inputStreamList.toJSONString();
    updateMixStreamRequest.setInputStreamList(inputStreamListStr);
    // 发起openapi请求
    UpdateMixStreamResponse updateMixStreamResponse = client.getAcsResponse(updateMixStreamRequest);
    String mixStreamId = updateMixStreamResponse.getMixStreamId();
    // 打印合流id
    System.out.println(mixStreamId);
}
```

查询合流任务

```
public void describeMixStreamList() throws ClientException {
    DescribeMixStreamListRequest describeMixStreamListRequest = new DescribeMixStreamListRequest();
    // 设置拉流域名(直播播流域名, 即用该域名观看合流后的内容)
    describeMixStreamListRequest.setDomainName("your domain name");
    // 设置appName
    describeMixStreamListRequest.setAppName("your app name");
    // 设置查询开始时间
    describeMixStreamListRequest.setStartTime("2020-12-03T14:00:00Z");
    // 设置查询结束时间
    describeMixStreamListRequest.setEndTime("2020-12-05T14:00:00Z");
    // 发起openapi请求
    DescribeMixStreamListResponse describeMixStreamListResponse = client.getAcsResponse(describeMixStreamListRequest);
    // 获取合流列表
    List<MixStreamInfo> result = describeMixStreamListResponse.getMixStreamList();
    // 打印合流id
    if (result != null && !result.isEmpty()) {
        for (MixStreamInfo mixStreamInfo : result) {
            String mixStreamId = mixStreamInfo.getMixstreamId();
            System.out.println(mixStreamId);
        }
    }
}
```

删除合流任务

```
public void deleteMixStream() throws ClientException {
    DeleteMixStreamRequest deleteMixStreamRequest = new DeleteMixStreamRequest();
    // 设置拉流域名(直播播流域名, 此前用该域名观看合流后的内容)
    deleteMixStreamRequest.setDomainName("your domain name");
    // 设置appName
    deleteMixStreamRequest.setAppName("your app name");
    // 设置流名
    deleteMixStreamRequest.setStreamName("your stream name");
    // 设置要删除的合流id
    deleteMixStreamRequest.setMixStreamId("your mixstreamId");
    // 发起openapi请求
    DeleteMixStreamResponse deleteMixStreamResponse = client.getAcsResponse(deleteMixStreamRequest);
    String mixStreamId = deleteMixStreamResponse.getMixStreamId();
    // 打印合流id
    System.out.println(mixStreamId);
}
```

11.直播管理

11.1.直播流管理

视频直播提供丰富的流管理功能，包括禁推、禁播、中断直播流和恢复直播流，以及查看在线流、历史流、禁推流及其详细信息。本文介绍直播流管理的功能及方法。

禁推直播流

视频直播提供了丰富的直播流管理功能，其中禁推直播流功能可帮助您在遇到下列情况的时候，对某路直播推流进行禁止推流的操作。

- 鉴权推流地址暴露。
- 用户恶意推流。
- 业务需求。

使用方法

- 控制台：支持在控制台查看禁推流、设置禁推流和恢复直播流。您可以在[视频直播控制台](#)的直播管理 > 流管理页面查看，请参见[流管理](#)。
- API

接口	描述	参考文档
ForbidLiveStream	禁止某路流的推送。 可以配置是否只断流不加入黑名单，预设某个时刻将流恢复。 ? 说明 禁止直播流的上限为10,000路，超出限制将禁用失败。调用频率限制为20次/秒。	禁止直播流
ResumeLiveStream	恢复某路流的推送。 ? 说明 调用频率限制为30次/秒。	恢复直播流

禁播直播流

视频直播还支持对某域名的指定播放协议进行禁播。

使用方法

- 控制台
 通过控制台的域名管理页面，指定播流域名进入域名配置页面，通过直播管理 > 访问控制 > 协议禁播来管理是否禁止播放。具体操作，请参见[协议禁播](#)。
- API

接口	描述	参考文档
BatchSetLiveDomainConfigs	批量配置域名。 可以通过Functions参数的alilive记录进行配置。	批量配置域名

中断直播流

视频直播支持中断指定直播流的推流。

- 正在进行上行推流的直播流，被断流后，只需要推流端重新推流即可恢复直播流。
- 通过[拉流功能](#)拉取第三方直播流时，被断流后无需重新拉取，因为回源拉流功能具备自动重试能力，断流后会重新拉取直播流。

使用场景：在回源拉流的场景中，客户的流需要长时间拉取。有时候因长时间拉取会出现视频音画不同步等情况，此时需要断流，回源重新拉流保证音视频的同步。

使用方法

- 控制台：在[视频直播控制台](#)的直播管理 > 流管理页面，单击对应直播流的断流，请参见[流管理](#)。
- API

接口	描述	参考文档
ForbidLiveStream	禁止某路流的推送。 将 oneshot 参数配置为 yes ，可以配置为只断流不加入黑名单。	禁止直播流

查看直播流

- 控制台

视频直播将直播流分为在线流、历史流和禁推流三类，您可以在[视频直播控制台](#)的直播管理 > 流管理页面查看。

支持查看直播流的AppName、StreamName、推流时间、推流类型、转码模版、流监控、直播地址。

- API

接口	描述	参考文档
DescribeLiveStreamsOnlineList	获取指定域名或者指定域名下指定应用的所有在线流信息。	查询域名在线流列表
DescribeLiveStreamsPublishList	获取指定时间段内指定域名（或域名下指定应用或指定流）的历史推流记录。	查询域名历史流列表
DescribeLiveStreamsBlockList	获取指定域名下被禁推的直播流信息。	查询直播流播放黑名单
DescribeLiveStreamsControlHistory	获取指定域名或应用下的直播流操作历史。	查询直播流操作历史
DescribeLiveDomainOnlineUserNum	获取指定域名下所有流的在线人数信息。	查询域名在线人数

查看直播流质量

在直播环境中，对直播效果影响最大的就是直播的卡顿。

导致直播卡顿的因素很多，其中之一便是上行传输的流畅度。上行传输不佳对于观看体验影响非常大，会导致所有观众端看到的视频都是卡顿的。视频直播提供了对上行流量监控的功能，您可以及时查看直播流上行传输状态。

- 控制台

视频直播支持查看直播流的基本信息、不同时刻的视频帧率和音频帧率、不同时刻的码率，您可以在[视频直播控制台](#)的直播管理 > 流管理页面，单击[流监控查看](#)。

- API

接口	描述	参考文档
DescribeLiveDomainFrameRateAndBitRateData	查询推流域名下直播流流帧率和码率数据。	查询推流域名流帧率和码率
DescribeLiveStreamBitRateData	查询RTMP协议的直播流在设置时间范围内的一组帧率和码率，适用于获取历史数据。	查询RTMP协议直播流的帧率和码率

11.2. 回调管理

视频直播提供直播推流状态、直播录制、按需录制、直播截图和智能审核的事件回调，本文介绍这些回调的设置方式、回调参数和示例。

回调简介

当直播过程中某些事件被触发时，阿里云将主动发送请求到客户服务器，客户服务器负责应答请求，验证通过后，您可接收到事件回调信息的JSON数据包。

视频直播支持回调如下事件的信息：

- [直播推流状态](#)
- [直播录制](#)
- [按需录制](#)
- [直播截图](#)
- [智能审核](#)

回调方式

视频直播服务目前支持HTTP回调（兼容HTTPS）方式获取事件通知。您需要部署一个HTTP服务来接收回调消息，并在控制台或通过API配置回调地址。

- 当推流或断流事件产生时，直播服务端会向该地址发起HTTP GET 请求，具体内容将通过URL参数送达。
- 当其他事件产生时，直播服务端会向该地址发起HTTP POST 请求，具体内容将通过JSON Body送达。

注意事项

回调地址URL无需标识，只需可正常访问，URL的应答有如下要求：如果访问超时，会重试URL，目前超时时间是5秒，重试次数5次，重试间隔1秒。

直播推流状态回调

通知当前域名推流状态，包括推流成功、断流状态，支持通过控制台或API配置回调地址。

- 直播推流状态回调设置方式

- 控制台

在[视频直播控制台](#)的[域名管理](#)页面，选择需要配置的推流域名，单击[域名配置](#)，进入[直播管理 > 基本配置 > 推流信息](#)页面，添加或修改推流回调地址。

- API

接口	描述	参考文档
SetLiveStreamsNotifyUrlConfig	设置推流回调配置。	设置直播推流回调配置
DescribeLiveStreamsNotifyUrlConfig	查询推流回调配置。	查询直播推流回调配置
DeleteLiveStreamsNotifyUrlConfig	删除推流回调配置。	删除直播推流回调配置

- 直播推流状态回调

推断流状态回调会将参数封装在MultiDict内。

- 直播推流状态回调中开播（publish）状态回调逻辑

- RTMP推流在阿里云直播服务收到On Publish消息后2秒内，如果推流端不主动断开，阿里云直播服务就会发推流成功回调。如果您的业务需要阿里云直播收到On Publish建联并真正收到直播流数据后才发推流成功回调，请[提交工单](#)联系我们后台配置。
- 假设您有推流域名A和播流域名B，如果您的域名B使用拉流直播（固定拉流和触发回源拉流）需要回调，请在域名A配置[推流回调](#)。配置后域名B的拉流回调逻辑与上一条一致，建联后默认2秒无主动断开即认为拉流成功，需要真正收到数据才认为成功。请[提交工单](#)申请配置。

 **说明** 建议业务方不仅根据回调通知判断推流、拉流接入正常，同时配合在线流列表查询推流正常后，再下发直播流播放地址。

直播推流状态回调参数

参数	描述
action	事件。 <ul style="list-style-type: none"> publish: 推流。 publish_done: 断流。
ip	推流的客户端IP。
id	推流流名称。
app	推流域名。默认为自定义的推流域名，如果未绑定推流域名即为播流域名。
appname	推流应用名称。
time	Unix时间戳。单位：秒。
usrargs	用户推流的参数。
node	CDN接受流的节点或者机器名。
height	分辨率的高。单位：像素。 <div style="border: 1px solid #ccc; background-color: #e0f2f1; padding: 5px; margin-top: 5px;"> ? 说明 分辨率信息（height和width）仅在首次回调时产生。 </div>
width	分辨率的宽。单位：像素。

推流状态回调参数示例

```
http:// 1.1.1.1?action=publish&ip=192.168.0.1&id=world&app=example.aliyundoc.com&appname=liveApp****&time=1609220385&usrargs={用户参数}&node=cdnvideocenter01020711****.cm3&height=720&width=1280
```

断流流状态回调参数示例

```
http:// 1.1.1.1?action=publish_done&ip=192.168.0.0&id=world&app=example.aliyundoc.com&appname=liveApp****&time=1609220385&usrargs={用户参数}&node=cdnvideocenter01020711****.cm3&height=720&width=1280
```

直播推流状态回调鉴权

推流回调鉴权功能默认关闭，用户可在配置推流回调地址时启用鉴权。启用功能后，鉴权逻辑如下：

- a. 阿里云视频直播发起回调请求时在HTTP(S)请求头中带上 `ALI-LIVE-TIMESTAMP` 和 `ALI-LIVE-SIGNATURE` 字段，供回调消息接收服务端进行签名认证。其中，`ALI-LIVE-SIGNATURE` 的值由如下计算而得：

`ALI-LIVE-SIGNATURE=MD5SUM (MD5CONTENT)`

`MD5CONTENT=回调域名|ALI-LIVE-TIMESTAMP取值|鉴权KEY`

说明 回调域名指配置回调URL的推流域名。鉴权Key指用户为推流回调URL配置的鉴权Key。

- b. 回调消息接收服务端接收回调消息时，将回调域名、`ALI-LIVE-TIMESTAMP` 取值、鉴权Key进行拼接后计算MD5值，得到加密字符串，再将计算出的加密字符串与视频直播发起的HTTP(S)请求头中的 `ALI-LIVE-SIGNATURE` 字段值进行对比。如果不一致，则请求非法。

直播录制回调

直播录制回调包括录制状态回调、文件生成回调，支持通过控制台和API进行配置。

- 录制状态回调：在文件开始录制与结束录制时发出，主要表示文件已经开始和结束录制。
- 文件生成回调：在文件生成时发出，主要包括文件名称，开始结束时间与时长信息。
- 直播录制回调设置方式

控制台

在**视频直播控制台**的**域名管理**页面，选择需要配置的播流域名，单击**域名配置**，进入**模版配置 > 录制配置**页面，添加或修改回调地址。

API

接口	描述	参考文档
AddLiveRecordNotifyConfig	添加域名级别录制回调配置。 若需要开启录制任务状态回调，请将NeedStatusNotify参数值设置为true。	添加直播录制回调配置
DescribeLiveRecordNotifyConfig	查询域名级别录制回调配置。	查询直播录制回调配置
DeleteLiveRecordNotifyConfig	删除域名级别录制回调配置。	删除直播录制回调配置

- 录制状态回调

○ 录制状态回调参数

参数	描述
domain	录制的播流域名。
app	应用名。
stream	流名。
event	事件名。 <ul style="list-style-type: none"> ■ record_started: 录制已经成功开始。 ■ record_paused: 录制已经成功暂停。 ■ record_resumed: 录制已经成功恢复继续录制。

○ 录制状态回调参数示例

```
{
  "domain": "demo.aliyundoc.com",
  "app": "liveApp****",
  "stream": "liveStream****",
  "event": "record_started"
}
```

● 文件生成回调

○ 文件生成回调参数

参数	描述
domain	录制的播流域名。
app	应用名。
stream	流名。
uri	目标录制文件在用户录制OSS Bucket下的路径。
duration	目标录制文件录制内容时长，单位：秒。
start_time	录制开始时间。Unix时间戳，单位：秒。
stop_time	录制结束时间。Unix时间戳，单位：秒。
push_args	推流参数中以callback_为前缀的参数，如callback_arg1、callback_myid等。 <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>? 说明 同一个参数名只能有一个参数值，如果传入多个值，回调仅包含第一个值。如推流参数包含 <code>callback_args1=value1&callback_args1=value2</code>，回调中仍为 <code>"callback_args1": "value1"</code>。</p> </div>

o 文件生成回调参数示例

推流URL为：

```
rtmp://demo.aliyundoc.com/liveApp****/liveStream****?callback_args1=value1&callback_myid=1231389741
```

则回调内容为：

```
{
  "domain": "demo.aliyundoc.com",
  "app": "liveApp****",
  "stream": "liveStream****",
  "uri": "liveApp****/liveStream****/0_2017-03-08-23:09:46_2017-03-08-23:10:40.flv",
  "duration": 69.403,
  "start_time": 1488985786,
  "stop_time": 1488985840,
  "push_args": {
    "callback_args1": "value1",
    "callback_myid": "1231389741"
  }
}
```

 **说明** 以上回调示例，适用于所有没有定制回调模版的客户。

按需录制回调

按需录制回调是录制之前通过回调的方式，传递直播流的相关参数给用户，由用户判断是否需要录制的功能。

用户指定一个回调地址，并设置一个domain、app或stream级别的录制成为按需录制。当录制服务收到匹配domain、app或stream名称的流时，向用户的回调地址发送请求（请求包含5个参数），来询问用户是否需要录制这路流。

● 请求参数

参数	类型	描述
domain	String	主播流域名。
app	String	播流所属应用名称。
stream	String	播流名称。
codec	String	编码方式。取值： o h264 o h265
vbitrate	String	视频码率。单位：kbps。

● 返回参数

参数	类型	是否必选	描述
ApiVersion	String	否	API版本。默认为1.0版本。
NeedRecord	Bool	是	是否需要录制。
Interval	Json Object	否	具体格式的录制周期变化。取值范围：5~21600。单位：秒。
Format	Array	否	录制格式。支持MP4、FLV、M3U8。

● 请求示例

```
GET /?app=seq_all&domain=demo.aliyundoc.com&stream=ondemand8&vbitrate=2000&codec=h264 HTTP/1.1
Host: pull.aliyundoc.com
User-Agent: Go-http-client/1.1
Accept-Encoding: gzip
```

● 返回示例

```
{
  "ApiVersion" : "1.0",
  "NeedRecord" : true,
  "Interval" : {
    "Mp4": 300,
    "Flv": 120,
    "M3U8": 180
  },
  "Format" : ["mp4","flv"]
}
```

● 返回参数处理

录制服务会对用户返回的参数配置和既有配置进行覆盖。例如，显示指定了Interval，那么录制服务会覆盖数据库中的录制配里面的Interval，Format会和数据库中的配置取交集。如果用户指定的格式全不在既有配置之中，则不会录制这路直播流。

说明

- 所有非200 http status code，都会被认为是调用用户接口失败。
- 对于范围body长度有限制，超过2048会被截断，以防恶意攻击。

直播截图回调

直播提供截图文件回调，您可以通过调用UpdateLiveAppSnapshotConfig接口，配置Callback参数开启。更多信息，请参见[更新直播截图配置](#)。

● 截图文件回调参数

参数	描述
Event	事件。
DomainName	截图的播流域名。
AppName	应用名。
StreamName	流名。
OssBucket	存储截图的Bucket名。
OssEndpoint	存储截图的OSS。
OssObject	截图的文件名。
CreateTime	截图时间。
SnapshotUrl	截图文件在用户录制OSS Bucket下的路径。
Size	图片大小。
Width	图片宽。单位：像素。
Height	图片高。单位：像素。

● 截图文件回调参数示例

```
{
  "Event": "Snapshot",
  "DomainName": "demo.aliyundoc.com",
  "AppName": "liveApp****",
  "StreamName": "liveStream****",
  "OssBucket": "liveBucket****",
  "OssEndpoint": "oss-cn-shan****.aliyuncs.com",
  "OssObject": "1****.jpg",
  "CreateTime": "2015-12-01T17:36:00Z",
  "SnapshotUrl": "http://liveBucket****.oss-cn-shan****.aliyuncs.com/1****.jpg",
  "Size": "36291",
  "Width": "1280",
  "Height": "720"
}
```

- 直播截图支持截图回调鉴权配置。详细信息请参见[设置截图回调鉴权配置](#)、[截图回调鉴权说明](#)。

智能审核回调

视频直播支持视频审核和语音审核的结果回调，您可以在控制台或通过API配置回调地址。

- 视频审核回调设置方式

o 控制台

您可以在[视频直播控制台](#)的[域名管理](#)页面需要选择配置的播流域名，单击[域名配置](#)，进入[模版配置](#) > [审核配置](#)页面，添加或修改回调地址。

 **说明** 中国内地的直播中心中，青岛直播中心的域名暂不支持智能审核；海外的直播中心中，只有新加坡支持智能审核。

o API

接口	描述	参考文档
AddLiveDetectNotifyConfig	添加直播视频审核回调配置。	添加直播审核回调配置
DescribeLiveAudioAuditNotifyConfig	查询直播视频审核回调配置。	查询直播审核回调配置
UpdateLiveDetectNotifyConfig	更新直播视频审核回调配置。	更新直播审核回调配置
DeleteLiveDetectNotifyConfig	删除直播视频审核回调配置。	删除直播审核回调配置

● 视频审核回调

只会对于有问题的视频内容进行回调通知，通知内容包含问题视频截图的审核信息和存储信息。

o 视频审核回调参数

参数	类型	描述
DomainName	String	播流域名。
AppName	String	应用名称。
StreamName	String	流名称。
OssEndpoint	String	存储对象Endpoint。
OssBucket	String	存储对象的Bucket。
OssObject	String	存储对象的文件名。
Result	JSONArray	检测结果。请参见 Result 。

Result

参数	类型	描述
BizType	String	业务类型。可用于选择模型，默认值为域名。

参数	类型	描述
Scene	String	<p>检测场景</p> <ul style="list-style-type: none"> ▪ porn: 鉴黄。 ▪ terrorism: 暴恐涉政。 ▪ ad: 图文违规。 ▪ live: 不良场景。 ▪ logo: 图片logo。
Label	String	<p>检测结果分类。不同检测场景的结果分类不同，具体如下：</p> <ul style="list-style-type: none"> ▪ 图片智能鉴黄 (<i>porn</i>) 结果分类： <ul style="list-style-type: none"> ▪ <i>normal</i>: 正常。 ▪ <i>sexy</i>: 性感。 ▪ <i>porn</i>: 色情。 ▪ 图片暴恐涉政 (<i>terrorism</i>) 结果分类： <ul style="list-style-type: none"> ▪ <i>normal</i>: 正常。 ▪ <i>bloody</i>: 血腥。 ▪ <i>explosion</i>: 爆炸烟光。 ▪ <i>outfit</i>: 特殊装束。 ▪ <i>logo</i>: 特殊标识。 ▪ <i>weapon</i>: 武器。 ▪ <i>politics</i>: 涉政。 ▪ <i>violence</i>: 打斗。 ▪ <i>crowd</i>: 聚众。 ▪ <i>parade</i>: 游行。 ▪ <i>carcrash</i>: 车祸现场。 ▪ <i>flag</i>: 旗帜。 ▪ <i>location</i>: 地标。 ▪ <i>others</i>: 其他。 ▪ 图文违规 (<i>ad</i>) 结果分类： <ul style="list-style-type: none"> ▪ <i>normal</i>: 正常。 ▪ <i>ad</i>: 其他广告。 ▪ <i>npx</i>: 牛皮癣广告。 ▪ <i>qrcode</i>: 含二维码。 ▪ <i>programCode</i>: 含小程序码。

参数	类型	描述
		<ul style="list-style-type: none"> 不良场景 (live) 结果分类: <ul style="list-style-type: none"> normal: 正常。 meaningless: 图片中无内容 (例如, 黑屏、白屏)。 PIP: 画中画。 smoking: 吸烟。 drivelive: 车内直播。 图片logo (logo) 结果分类: <ul style="list-style-type: none"> normal: 正常。 TV: 含受管控的logo。 trademark: 含商标。
Rate	Float	置信度分数。取值范围: 0 (表示置信度最低) ~100 (表示置信度最高)。 <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> ? 说明 该值仅作为参考, 强烈建议您不要在业务中使用。建议您参考Label结果用于内容违规判定。 </div>
Extent	String	预留字段。

? 说明 默认新用户将使用此版本, 老用户维持不变, 如需使用此版本, 请您[提交工单](#)。

o 视频审核回调参数示例

```

{
  "DomainName": "demo.aliyundoc.com",
  "AppName": "liveApp****",
  "StreamName": "liveStream****",
  "OssEndpoint": "oss-cn-hang****.aliyuncs.com",
  "OssBucket": "liveBucket****",
  "OssObject": "example.jpg",
  "Result": [
    {
      "BizType": "demo.aliyundoc.com",
      "Result": [
        {"Label": "Porn", "Rate":11, "Suggestion": "review", "Scene":"porn", "Extent": {}},
        {"Label": "Ad", "Rate":11, "Suggestion": "review", "Scene":"ad", "Extent": {}},
        {}
      ]
    }
  ]
}
    
```

● 语音审核回调设置方式

目前不支持控制台方式设置语音审核，您可以通过API设置。

接口	描述	参考文档
AddLiveAudioAuditNotifyConfig	添加直播语音审核回调配置。  注意 直播语音审核回调返回为JSON字符串。	添加直播语音审核回调配置
DeleteLiveAudioAuditNotifyConfig	删除直播语音审核回调配置。	删除直播语音审核回调配置
UpdateLiveAudioAuditNotifyConfig	更新直播语音审核回调配置。	更新直播语音审核回调配置
DescribeLiveAudioAuditNotifyConfig	查询直播语音审核回调配置。	查询直播语音审核回调配置

● 语音审核回调

只会对于有问题的音频内容进行回调通知，通知内容包含问题语音的文本信息以及最近一分钟内的上下文信息。

○ 语音审核回调参数

参数	类型	描述
domain	String	播流域名。
app	String	应用名称。
stream	String	流名称。
timestamp	Int	回调时间戳。单位：秒。
result	JSONArray	检测结果。请参见Result。

Result

参数	类型	描述
scene	String	检测场景。

参数	类型	描述
label	String	<p>检测结果的分类。取值：</p> <ul style="list-style-type: none"> ▪ <i>normal</i>: 正常文本 ▪ <i>spam</i>: 含垃圾信息 ▪ <i>ad</i>: 广告 ▪ <i>politics</i>: 涉政 ▪ <i>terrorism</i>: 暴恐 ▪ <i>abuse</i>: 辱骂 ▪ <i>porn</i>: 色情 ▪ <i>flood</i>: 灌水 ▪ <i>contraband</i>: 违禁 ▪ <i>meaningless</i>: 无意义 ▪ <i>customized</i>: 自定义（例如命中自定义关键词）
suggestion	String	<p>处理建议。取值：</p> <ul style="list-style-type: none"> ▪ <i>pass</i>: 结果正常，无需进行其余操作。 ▪ <i>review</i>: 结果不确定，需要进行人工审核。 ▪ <i>block</i>: 结果违规，建议直接删除或者限制公开。
rate	Float	<p>置信度分数。取值范围：0（表示置信度最低）~100（表示置信度最高）。</p> <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p> 说明 该值仅作为参考，强烈建议您不要在业务中使用。建议您参考Label结果用于内容违规判定。</p> </div>
details	JSONArray	<p>语音对应的文本详情，每一句文本对应一个元素，可以包含一个或者多个元素。关于每个元素的结构描述，请参见Detail。</p>

Detail

参数	类型	描述
startTime	Int	句子开始的时间戳，单位：秒。
endTime	Int	句子结束的时间戳，单位：秒。
text	String	问题语音转换成文本的结果。

参数	类型	描述
label	String	检测结果的分类。取值： <ul style="list-style-type: none"> ▪ <i>normal</i>: 正常文本 ▪ <i>spam</i>: 含垃圾信息 ▪ <i>ad</i>: 广告 ▪ <i>politics</i>: 涉政 ▪ <i>terrorism</i>: 暴恐 ▪ <i>abuse</i>: 辱骂 ▪ <i>porn</i>: 色情 ▪ <i>flood</i>: 灌水 ▪ <i>contraband</i>: 违禁 ▪ <i>meaningless</i>: 无意义 ▪ <i>customized</i>: 自定义（例如命中自定义关键词）

- 语音审核回调参数示例

```
{
  "domain": "example.aliyundoc.com",
  "app": "liveApp****",
  "stream": "5d9747eba39b44769852d276f9ff****",
  "timestamp": 1572248095,
  "result": [
    {
      "scene": "antispam",
      "label": "ad",
      "suggestion": "block",
      "rate": 99.91,
      "details": [
        {
          "startTime": 1572248023,
          "endTime": 1572248040,
          "text": "丽江陇川等酒店免费居住权120天居住权可与亲友共享来云栖社国际烧酒店接待中心地址芒市团结大街96号邓朝ktv旁咨询热线2285699",
          "label": "ad"
        },
        {
          "startTime": 1572248040,
          "endTime": 1572248070,
          "text": "典藏经典别墅尊享隆多温泉、梁河金塔、温泉旅游小镇二期滨江苑地区11月2号盛大开盘天然龙窝、温泉水入户户型多样，设计经典价位超值，值得您拥有开盘当天还有超值优惠，欢迎您前来品鉴，凭借地址，然和县遮岛镇先锋路229号分享热线069269557776955777",
          "label": "normal"
        },
        {
          "startTime": 1572248072,
          "endTime": 1572248077,
          "text": "快乐的时光，有你相伴惬意的日子，格外轻松。",
          "label": "normal"
        },
        {
          "startTime": 1572248078,
          "endTime": 1572248086,
          "text": "fme043忙里偷闲音乐时光有歌曲温暖，有你想念影院。",
          "label": "normal"
        }
      ]
    }
  ]
}
```

11.3. 文件管理

视频直播具有直播录制和截图功能，录制和截图产生的文件可以通过控制台或存储的OSS进行管理。本文介绍管理方式。

录制文件管理

您可以在[视频直播控制台](#)的直播管理 > 录制文件管理页面，查看指定域名下的录制文件，请参见[录制文件管理](#)。

若存储至OSS，也可以通过录制文件生成的回调信息，查看对应目标录制文件在用户录制OSS Bucket下的路径。请参见[直播录制回调](#)。

截图管理

您可以在[视频直播控制台](#)的直播管理 > 截图管理页面，查看指定域名下的截图文件，请参见[截图管理](#)。

若您开启了截图回调，也可通过截图回调信息查看图片存储路径，请参见[直播截图回调](#)。

12. 数据统计

本文详细介绍如何查询视频直播中上行推流数据和下行播流数据。

您可以登录[视频直播控制台](#)，在[数据监控](#)、[访问统计](#)和[用量查询](#)页签中查询对应数据。

- [资源监控](#)可以监控流量带宽-上行、流量带宽-下行、回源带宽流量、HTTPCODE等数据。更多详情，请参见[资源监控](#)。
- [实时监控](#)可以监控推流、流量带宽等数据。更多详情，请参见[实时监控](#)。
- [访问统计](#)可以查询独立访客数、用户分布、域名排行等数据。更多详情，请参见[访问统计](#)。
- [用量查询](#)可以查询播放带宽/流量、推流路数、转码时长、截图张数等查询功能。更多详情，请参见[用量查询](#)。

API提供以下接口供您查询对应的数据：

- [查询直播域名带宽数据](#)
- [查询直播录制路数、转封装用量](#)
- [查询直播域名截图张数](#)
- [查询直播域名流量数据](#)
- [查询域名转码用量数据](#)
- [查询直播域名实时带宽数据](#)
- [查询直播域名实时HTTP返回码占比](#)
- [查询直播域名实时流量监控数据](#)
- [查询直播流历史在线人数](#)
- [查询在线直播流数量](#)
- [查询域名在特定计费区域的用量数据](#)
- [查询推流域名流帧率和码率](#)
- [查询域名实时日志投递信息](#)
- [查询直播域名PV、UV数据](#)

如何查看上行推流质量

实时监测上行推流的质量可以定位直播中的卡顿、延时等问题。推流监控是秒级监控，实时返回每秒钟的推流监控数据，包括主播到CDN节点的接收音视频帧率、音视频时间戳、接受头次数、音视频码率（kbps）、最大音视频帧间隔（毫秒）、推流断开错误码和实时日志。通过查看每秒中视频流的状态可判断推流的稳定性，及时对直播的内容进行监测，减少观看视频时的卡顿影响。

 **说明** 您可以按照以下步骤通过[视频直播控制台](#)查看上行推流质量。您还可以通过[查询推流域名流帧率和码率](#)接口中查询的推流实时码率和帧率来感知推流质量。

操作步骤

1. 登录[视频直播控制台](#)。
2. 在左侧导航栏中，选择[数据监控](#) > [实时监控](#)。
3. 选择正在推流的域名、时间段、应用名称和流名称，单击[查询](#)。
 - 如果您想要实时获取监控信息，可以选择[10分钟实时监控](#)。实时监控数据延迟为3分钟。
 - 如果查看历史的监控信息，也可以选择[近半小时](#)、[近1小时](#)，以及[自定义](#)的监控时间。查询的时间

跨度最大支持1天查询。

关于监控项的详细信息，请您参见以下内容。

流详情

流监控	含义	场景
推流地址	主播的推流地址。格式： rtmp://domainName/appName/ streamName。	查看推流地址。
AppName	流的应用名称。	查看应用名。
StreamName	主播推的流名。	查看流名。
开始推流时间	主播单次推流的开始推流时间。	查看开始推流时间点。
结束推流时间	主播单次推流的结束推流时间。	查看结束推流时间点。
推流主播IP	主播推流端的IP地址。	监控主播端IP。
编码信息	主播所推直播流的metadata。格式： {视频编码}{分辨率}/{音频编码} {音频采样率}。视频编码、分辨率、 音频编码和音频采样率分别采集自 meta的videocodecid,frame_rate, audiocodecid和aac header。	监控流的编码信息。
历史推流记录	主播针对同一路流名，其不同的推断 流记录。	监控主播的推断流时间点。

主播到CDN节点的接受音视频帧率

监控项	含义	场景
视频帧率	服务端每秒采集的接受视频帧数。	监控主播到服务器端的上行视频帧率 稳定性。
音频帧率	服务端每秒采集的接受的音频帧数。	监控主播到服务器端的上行音频帧率 稳定性。

音视频时间戳

监控项	含义	场景
-----	----	----

监控项	含义	场景
视频时间戳	服务器端采集的每秒最后一帧视频帧所携带的视频时间戳（单位：毫秒）。 ? 说明 采集的一秒周期内如果没有收到视频帧，则设置为-1。	监控视频时间戳的单调性。
音频时间戳	服务器端采集的每秒最后一帧音频帧所携带的音频时间戳（单位：毫秒）。 ? 说明 采集的一秒周期内如果没有收到音频帧，那么设置为-1。	监控音频时间戳的单调性。
音视频时间戳差值	以上采集的视频时间戳和音频时间戳所得出的差值（单位：毫秒）。	通过监控音视频时间戳差值的趋势进而判断主播推流音视频时间戳是否同步。

接受头次数

监控项	含义	场景
AVC	服务器端采集的接受主播端推流的AVC sequence header的累计次数。	监控主播端推的AVC sequence header 次数。
AAC	服务器端采集的接受主播推流的AAC sequence header的累计次数。	监控主播端推的AAC sequence header 次数。

音视频码率

监控项	含义	场景
视频码率	服务器端每秒采集的视频码率（单位：kbps）。	监控主播推流的视频码率。
音频码率	服务器端每秒采集的音频码率（单位：kbps）。	监控主播推流的音频码率。

最大音视频帧间隔

监控项	含义	场景
视频帧间隔	服务端每秒采集的最大的视频帧之间的时间戳间隔。	监控视频帧本身的时间戳是否出现跳变。
音频帧间隔	服务端每秒采集的最大的音频帧之间的时间戳间隔。	监控音频帧本身的时间戳是否出现跳变。

推流断开错误码

监控项	含义	场景
错误码	服务器端每秒采集的推流断开错误码。	<p>监控主播推流断开的原因。常见错误码如下：</p> <ul style="list-style-type: none"> • 2：推流端主动断开； • 1,3,4,6：客户端连接发送超时； • 101：被断流接口踢掉； • 103：推流端超过10秒没有发送数据； • 104,106：chunk stream id太大，数值超过32； • 105：推流一帧数据大小超过限制，超过5,242,880字节； • 122,123：推流鉴权不通过； • 125：推流时相同名称的流在中心已存在； • 其他错误码：若出现其他错误码，请您截图提交工单与我们联系。

实时日志

实时日志延时小（秒级延时），可以帮助您快速了解直播推流、访问详情。

 **说明** 您可以按照以下步骤通过视频直播控制台查询域名在指定时间的日志。您还可以通过[查询域名实时日志投递信息](#)接口来查询域名在指定时间的日志。

操作步骤

1. 登录[视频直播控制台](#)。
2. 在左侧导航栏中，选择[日志管理](#)。
3. 选择待查询的域名及时间，单击[查询](#)，查询该域名在指定时间的日志。
4. 上行推流日志字段，请参见[日志管理](#)。

如何进行下行播流数据分析

您可以通过播流的带宽流量、状态码、独立访客数、用户分布等数据进行播流数据分析。

名称	说明	查询方式
实时流量带宽	播流域名下某个区域、运营商、时间段内的实时带宽流量。	<p>说明 您可以按照以下步骤通过视频直播控制台查询播流域名下某个区域、运营商、时间段内的实时带宽流量。您还可以通过查询直播域名实时流量监控数据接口来查询播流域名下某个区域、运营商、时间段内的实时带宽流量。</p> <ol style="list-style-type: none"> 1. 登录视频直播控制台。 2. 在左侧导航栏中，选择数据监控 > 实时监控 > 流量带宽。 3. 选择播流域名、运营商、区域、时间段，单击查询。
播流带宽流量	查询指定域名的流量和带宽使用情况。	<p>说明 您可以按照以下步骤通过视频直播控制台查询指定域名的流量和带宽使用情况。您还可以通过查询直播域名实时带宽数据接口来查询指定域名的流量和带宽使用情况。</p> <ol style="list-style-type: none"> 1. 登录视频直播控制台。 2. 在左侧导航栏中，选择用量查询。 3. 您可以选择待查询的域名、区域、及查询的时间段（今天、昨天、近7天、近30天或自定义），查看域名的流量和带宽使用情况。
HTTP状态码	HTTP请求返回状态码。	<p>说明 您可以按照以下步骤通过视频直播控制台查看HTTP请求返回状态码，您还可以通过查询直播域名实时HTTP返回码占比接口来查看HTTP请求返回状态码。</p> <ol style="list-style-type: none"> 1. 登录视频直播控制台。 2. 在左侧导航栏中，选择数据监控 > 实时监控 > 质量监控。

名称	说明	查询方式
用户分布	观看视频用户所在地域分布。	<p> 说明 您可以按照以下步骤通过视频直播控制台观看视频用户所在地域分布、观看各个加速域名的访问排名、统计当前域名下一定时间段内独立请求的IP次数。您还可以通过查询直播域名PV、UV数据接口观看视频用户所在地域分布、观看各个加速域名的访问排名、统计当前域名下一定时间段内独立请求的IP次数。</p> <ol style="list-style-type: none"> 1. 登录视频直播控制台。 2. 在左侧导航栏中，选择访问统计。 3. 选择您所要查看 独立访客数、用户分布或域名排行。 4. 选择您想要查看的 域名和时间段，并单击查询。
域名排名	各个加速域名的访问排名。	
独立访客数 (UV)	统计当前域名下一定时间段内独立请求的IP次数。	

13. 日志管理

日志管理提供日志下载和实时日志推送功能。可以将日志推送至日志服务进行分析，帮您快速发现和定位问题。本文详细为您介绍实时日志推送和日志下载的功能。

日志下载

- 日志生成规则
 - 日志下载可以下载最近一个月（30天）的日志数据。日志文件按小时粒度分割。
 - 日志文件延迟时间：日志下载通常情况可下载4小时之前的日志，若数据量大、有堆积时可能超过24小时。
 - 日志命名规则：加速域名_年_月_日_时间开始_时间结束。示例：
www.example.com_2021_01_06_000000_010000.gz

- 使用方法

您可以在[视频直播控制台](#)日志管理中下载日志。具体操作，请参见[日志管理](#)。

- 日志下载字段格式说明

字段	参数
时间	[9/Jun/2015:01:58:09 +0800]
访问IP	192.168.0.1
代理IP	无
responsetime (单位: ms)	1542
referer	无
method	GET
访问URL	http://www.aliyun.com/index.html
httpcode	200
requestsize (单位: Byte)	191
responsesize (单位: Byte)	2830
cache命中状态	HIT (命中), MISS (未命中)
UA头	Mozilla/5.0 (compatible; AhrefsBot/5.0; +http://ahrefs.com/robot/)
文件类型	text、html

实时日志推送

- 功能介绍

您可以通过实时日志推送功能，将直播域名日志实时推送至日志服务，并进行日志分析，便于快速发现和定位主播推流问题，或者分析直播观看用户数据为运营策略提供依据。

此功能涉及阿里云日志服务产品，如果您想深入了解日志服务，请参考[什么是日志服务](#)。

● 计费说明

如果您开启了实时日志推送服务，将会产生额外的增值服务费用。您需要按实时日志推送成功条数进行付费，该费用仅包含日志采集推送服务的费用，不包含SLS日志服务的费用。更多详情，请参见[实时日志费用](#)。

● 实时日志推送服务的优势

传统的日志分析模式，需要将日志下载，重新上传至数据库后再进行数据分析，整个过程需耗费人力、时间维护。实时日志延时小（秒级延时），可帮您快速了解直播的推流、访问详情日志。开通实时日志推送服务，数据将自动推送到日志服务，可实时查看日志分析结果。

● 使用方法

实时日志推送可以在[视频直播控制台](#)中配置。具体操作，请参见[实时日志推送](#)。

 **说明** 目前仅支持配置播流域名，如需推送上行实时日志（即配置推流域名），请[提交工单](#)配置。

实时日志推送请参见[实时日志推送字段说明](#)。

实时日志推送与日志下载的区别

- 实时日志为实时采集的日志数据，日志数据延迟不超过3分钟。帮助您快速对日志进行分析，发现问题，及时决策。
- 日志下载通常情况可下载4小时之前的日志，若数据量大有堆积时可能超过24小时。

实时日志推送字段说明

在直播控制台开通、配置实时日志推送服务后，您可以根据下表查看日志中各字段的含义。

● 上行推流日志字段

名称	说明	默认是否创建索引
unixtime	请求时间	是
domain	请求域名	是
uri	请求资源	是
uri_param	请求参数	是
client_ip	和服务器建联客户端IP	是
client_port	和服务器建联客户端端口	是
request_size	请求大小	是
request_time	请求响应时间，单位：毫秒	是
response_size	请求返回大小，单位：字节	是

名称	说明	默认是否创建索引
server_addr	服务器IP	是
server_port	服务器端口	是
vfcnt	视频帧率	是
vfts	视频时间戳（1秒内最后一帧的 dts）	是
vfdropped	视频每秒丢帧数	是
afcnt	音频帧率	是
afts	音频时间戳（1秒内最后一帧的时间戳）	是
afdropped	音频每秒丢帧数	是
wait	应用层堆积数	是
first_frame_duration	首帧时间	是
v_header_count	收到的视频头个数	是
a_header_count	收到音频头个数	是
vbytes_merge	视频码率，单位：Bps ? 说明 常用单位是 kbps，例如100000 Bps=100000÷1000×8=800 kbps。 	是
abytes_merge	音频码率，单位：Bps	是
publish_type	上行接入协议类型	是

● 下行播放日志字段

名称	说明	默认是否创建索引
unixtime	请求时间	是
domain	请求域名	是
method	请求方法	是
scheme	请求协议	是
uri	请求资源	是

名称	说明	默认是否创建索引
uri_param	请求参数	是
client_ip	用户真实IP	是
proxy_ip	代理IP	是
remote_ip	和CDN节点建连IP	是
remote_port	和CDN节点建连客户端端口	是
refer_protocol	HTTP Refer中的协议	是
refer_domain	HTTP Refer中Domain信息	是
refer_uri	HTTP Refer中URI信息	是
refer_param	HTTP Refer中的参数信息	是
request_size	请求大小	是
request_time	请求响应时间，单位：毫秒	是
response_size	请求返回大小，单位：字节	是
return_code	请求响应码	是
sent_http_content_range	应答头里表示的Range信息（由源站创建），如Bytes: 0~99/200	是
server_addr	服务的CDN节点IP	是
server_port	服务的CDN节点服务端口	是
body_bytes_sent	实际发送Body大小，单位：字节	是
content_type	请求的资源类型	是
hit_info	用户请求中Header头中Range字段取值，如Bytes: 0~100	是
user_agent	用户代理信息	是
user_info	User Info信息	是
uuid	请求唯一标识	是
via_info	Via头信息	是
xforwardfor	请求头中XForwardFor字段	是

14. 配额管理

视频直播已在阿里云配额中心实现配额管理功能。您可以通过阿里云配额中心为视频直播服务申请配额、查询配额上限和配额使用情况，或者添加配额用量告警。本文介绍视频直播配额管理的使用说明和操作方法。

产品简介

配额中心是集中管理云服务配额的服务，配额中心的产品能力和产品优势如下。

产品能力

- 配额查询：查看云产品支持的配额项，配额是否可申请，支持申请的范围。
- 配额申请：自助式提交配额申请，通过申请历史进行申请状态跟踪。
- 配额用量告警：支持按配额用量的百分比和固定值预警，可提前申请调整配额。

产品优势

- 配额集中管理：配额中心提供了统一的配额管理功能，无需分散查找配额。
- 配额透明化：可查看产品默认配额值和申请后的配额值。
- 配额主动管理：支持配额告警功能，做到配额提前申请。
- 配额申请效率优化：配额自助申请，相对工单申请方式，申请链路更短。

使用说明

 **注意** 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。

- 如果您的业务扩展，配额需求大于阿里云视频直播**使用限制**中配额限制时，请参见**创建配额提升申请**提升配额，阿里云会根据您业务的实际需求评估后进行配置。
- 支持阿里云账号（主账号）和RAM用户账号（子账号）在配额中心完成相关操作，如果您为RAM用户需先完成授权，详情请参见**为RAM用户授权**。
- 原始流（推流/拉流）、转码流配额为峰值并发数值，若业务预期增长超出峰值，可申请提升配额（注意：需提供使用场景、未来的量级和时间等详细信息）。
- 如果您的上行推流与下行播放使用带宽的比例大于1：50（上下行不均衡业务），则需要收取上行推流费用，可通过**提交工单**或联系您的商务提交相关申请，再申请调整配额。
- 提交转码配额时，请先配置好转码模板或备注需使用的分辨率和清晰度具体信息。
- 若配额申请数值超出配额平台最大值，可通过**提交工单**或联系您的商务提交相关申请。

操作方法

1. **登录配额中心**。
2. 在左侧导航栏，选择**产品列表 > 通用配额**。
3. 在**通用配额产品列表**页面，**产品类目**下拉框里选择**视频与CDN**。
4. 单击**视频直播**进入产品配额申请页面，您可以根据需要完成如下操作：
 - **配额查询**：查询支持的配额项及配额详细信息，请参见**查询云服务支持的配额维度及其详情**。
 - **配额申请**：详细配额申请方法，请参见**创建配额提升申请**。
 - **申请历史**：如何查看申请历史，请参见**查看配额申请历史**。
 - **创建告警**：详细创建告警方法，请参见**创建配额告警**。

- 告警项：如何查询已经创建的告警项，请参见[查询配额告警列表及其详情](#)。

15.常见问题FAQ

15.1. 视频直播服务支持哪些直播流播放协议?

视频直播服务支持输出RTMP、FLV、HLS、ARTC四种协议

协议对比如下:

协议类型	延时	优点	缺点	特点	适合端	场景推荐
RTMP	1s~3s	延时低	<ul style="list-style-type: none"> iOS平台要开发支持相关协议的播放器 使用非标准TCP端口 	TCP长连接	PC端	实时性要求不高的直播
FLV	1s~3s	<ul style="list-style-type: none"> 延时低 可通过HTML5解封包播放 	需要集成SDK才能播放	TCP长连接	PC端	实时性要求不高的直播
HLS	> 10s	<ul style="list-style-type: none"> iOS、Android和H5原生支持良好 可通过HTML5解封包播放 	延时高	HTTP短连接	PC端、移动端	实时性要求不高的直播; 移动端和H5端
ARTC	1s	<ul style="list-style-type: none"> 超低延时 抗弱网能力强 	H5播放不支持B帧和AAC音频(可通过阿里云RTS转码功能去除B帧并将音频转为Opus)	UDP	PC端、移动端	实时性要求高的直播, 如电商带货、在线教育、社交互动等

15.2. 视频直播支持的分辨率和对应合适的码率

视频直播服务不限制推流码率, 支持常见分辨率以及对应的码率。

以下列举常见分辨率及对应码率:

- 640x480: 100kbps~800kbps。
- 1280x720: 200kbps~1500kbps。
- 1920x1080: 500kbps~4000kbps。
- 2K (2560x1440): 2000kbps~8000kbps。
- 4K (3840x2160): 4000kbps~30000kbps。

为避免推流卡顿, 建议:

- 推流不超过4Mbps
此建议是根据平常主播端带宽来判断的，视频直播没有这个限制。
- 4K、8K直播，推流码率超过25Mbps
单TS切片文件大小超过上限100MB，录制会失败。

15.3. 直播禁止推流功能相关问题

什么时候需要禁止推流？

当遇到如下业务场景，例如，对应的鉴权推流地址暴露、客户恶意推流该地址、业务需求禁止该推流URL进行推流等，就可以使用直播的禁止推流功能满足安全需求。

如何实现禁止推流？

- 直播控制台禁止推流
 - 获取正在推流地址并禁推

直播推流成功后，您可以在[视频直播控制台](#)的直播管理 > 流管理页面，选择对应的域名，单击查询，查看到正在推流的地址，单击禁推即可。



- 获取历史推流地址并禁推
直播推流结束后，您可以在[视频直播控制台](#)的直播管理 > 流管理页面，单击历史流页签，选择对应的域名，单击查询，查看到正在推流的地址，单击禁推即可。
- SDK实现禁止推流

视频直播的API是基于阿里云的OpenAPI实现的。因此，SDK的调用也是对OpenAPI封装后调用。下载服务端SDK，请参见[SDK下载](#)。如何使用SDK，请参见[SDK使用说明](#)。

以Java SDK为例：Java SDK建议使用Maven管理项目依赖，在pom.xml中添加如下的依赖项：

```
<dependencies>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-cdn</artifactId>
    <version>2.3.1</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <optional>true</optional>
    <version>2.3.8</version>
  </dependency>
</dependencies>
```

其中，aliyun-java-sdk-core是对阿里云OpenAPI封装的核心模块，在使用OpenAPI的产品调用Java SDK均需要添加该依赖项，而aliyun-java-sdk-cdn包括了视频直播接口的Request和Response类。其中，查询推流黑名单列表的Request和Response类为DescribeLiveStreamsBlockListRequest及DescribeLiveStreamsBlockListResponse，Java代码示例如下：

```
public static void main(String[] args) throws Exception {
    IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<accessId>", "<accessKey>");
    DefaultAcsClient client = new DefaultAcsClient(profile);
    DescribeLiveStreamsBlockListResponse res=DescribeLiveStreamsBlockListR(client);
    System.out.println(res.getStreamUrls());
}
static DescribeLiveStreamsBlockListResponse DescribeLiveStreamsBlockListR(DefaultAcsClient client) {
    DescribeLiveStreamsBlockListRequest request =new DescribeLiveStreamsBlockListRequest(
);
    request.setDomainName("<DomainName>");
    DescribeLiveStreamsBlockListResponse response=null;
    try {
        response = client.getAcsResponse(request);
    } catch (ServerException e) {
        e.printStackTrace();
    } catch (ClientException e) {
        e.printStackTrace();
    }
    return response;
}
```

Demo代码主要流程如下：

i. 初始化Client对象。

SDK需要通过IAcsClient对象完成对OpenAPI的调用，而初始化IAcsClient对象时根据IClientProfile对象配置region、AccessKeyId和AccessKeySecret参数。其中，region请一直写cn-hangzhou，因为这里指的是OpenAPI的服务器所在区域。AccessKeyId和AccessKeySecret参数即是账号权限验证，如果是主账号的AccessKeyId和AccessKeySecret，则可以正常使用。如果是子账号的AccessKeyId和AccessKeySecret参数，则需要保证该子账号有视频直播的权限。

ii. 初始化请求Request类。

SDK对每个接口实现了对应的Request类和Response类，对应的禁止推流Request类需要传入哪些参数。API接口定义请参见[禁止推流API](#)。

iii. 发起请求获取结果。

使用Client对象的getAcResponse方法发起Request对象请求，并根据该方法反馈的反序列化的结果实例化为对应接口的Response类，且在执行该方法时可以catch ServerException和ClientException异常类。

- 当http status>=300且<500，SDK抛ClientException，表示客户端错误。
- 当http status>=500，SDK抛ServerException，表示服务器端错误。
- 如果没有抛出异常，则表示接口调用成功。

如何查看推流黑名单？

• 直播控制台查看推流黑名单

禁推成功后，您可以在[视频直播控制台](#)的直播管理 > 流管理页面，选择对应的域名，单击禁推流页签，单击查询，即可查看被禁推的直播流。

• SDK获取推流黑名单

视频直播的API是基于阿里云的OpenAPI实现的。因此，SDK的调用也是对OpenAPI封装后调用。下载服务端SDK，请参见[SDK下载](#)。如何使用SDK，请参见[SDK使用说明](#)。

以Java SDK为例：Java SDK建议使用Maven管理项目依赖，在pom.xml中添加如下的依赖项：

```
<dependencies>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-cdn</artifactId>
    <version>2.3.1</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <optional>true</optional>
    <version>2.3.8</version>
  </dependency>
</dependencies>
```

其中，aliyun-java-sdk-core是对阿里云OpenAPI封装的核心模块，在使用OpenAPI的产品调用Java SDK均需要添加该依赖项，而aliyun-java-sdk-cdn包括了视频直播接口的Request和Response类。其中，查询推流黑名单列表的Request和Response类为DescribeLiveStreamsBlockListRequest及DescribeLiveStreamsBlockListResponse，Java代码示例如下：

```
public static void main(String[] args) throws Exception {
    IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<accessId>", "<accessKey>");
    DefaultAcsClient client = new DefaultAcsClient(profile);
    DescribeLiveStreamsBlockListResponse res=DescribeLiveStreamsBlockListR(client);
    System.out.println(res.getStreamUrls());
}
static DescribeLiveStreamsBlockListResponse DescribeLiveStreamsBlockListR(DefaultAcsClient client) {
    DescribeLiveStreamsBlockListRequest request =new DescribeLiveStreamsBlockListRequest(
);
    request.setDomainName("<DomainName>");
    DescribeLiveStreamsBlockListResponse response=null;
    try {
        response = client.getAcsResponse(request);
    } catch (ServerException e) {
        e.printStackTrace();
    } catch (ClientException e) {
        e.printStackTrace();
    }
    return response;
}
```

Demo代码主要流程如下：

i. 初始化Client对象。

SDK需要通过IAcsClient对象完成对OpenAPI的调用，而初始化IAcsClient对象时根据IClientProfile对象配置region、AccessKeyId和AccessKeySecret参数。其中，region请一直写cn-hangzhou，因为这里指的是OpenAPI的服务器所在区域。AccessKeyId和AccessKeySecret参数即是账号权限验证，如果是主账号的AccessKeyId和AccessKeySecret则可以正常使用，如果是子账号的AccessKeyId和AccessKeySecret参数则需要保证该子账号有视频直播的权限。

ii. 初始化请求Request类。

SDK对每个接口实现了对应的Request类和Response类，对应的禁止推流Request类需要传入哪些参数。API接口定义请参见[查询推流黑名单列表API](#)。

iii. 发起请求获取结果。

使用Client对象的getAcsResponse方法发起Request对象请求，并根据该方法反馈的反序列化的结果实例化为对应接口的Response类，并在执行该方法时可以catch ServerException和ClientException异常类。

- 当http status>=300且<500，SDK抛ClientException，表示客户端错误。
- 当http status>=500，SDK抛ServerException，表示服务器端错误。
- 如果没有抛出异常，则表示接口调用成功。

如何恢复黑名单直播流推送地址？

● 直播控制台恢复

禁推成功后，您可以在[视频直播控制台](#)的直播管理 > 流管理页面，选择对应的域名，单击禁推流页签，单击恢复，即可恢复被禁推的直播流。

● SDK恢复

视频直播的API是基于阿里云的OpenAPI实现的。因此，SDK的调用也是对OpenAPI封装后调用。下载服务端SDK，请参见[SDK下载](#)。如何使用SDK，请参见[SDK使用说明](#)。

以Java SDK为例：Java SDK建议使用Maven管理项目依赖，在pom.xml中添加如下的依赖项：

```
<dependencies>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-cdn</artifactId>
    <version>2.3.1</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <optional>true</optional>
    <version>2.3.8</version>
  </dependency>
</dependencies>
```

其中，aliyun-java-sdk-core是对阿里云OpenAPI封装的核心模块，在使用OpenAPI的产品调用Java SDK均需要添加该依赖项，而aliyun-java-sdk-cdn包括了视频直播接口的Request和Response类。其中，查询推流黑名单列表的Request和Response类为DescribeLiveStreamsBlockListRequest及DescribeLiveStreamsBlockListResponse，Java代码示例如下：

```
public static void main(String[] args) throws Exception {
    IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<accessId>", "<accessKey>");
    DefaultAcsClient client = new DefaultAcsClient(profile);
    DescribeLiveStreamsBlockListResponse res=DescribeLiveStreamsBlockListR(client);
    System.out.println(res.getStreamUrls());
}
static DescribeLiveStreamsBlockListResponse DescribeLiveStreamsBlockListR(DefaultAcsClient client) {
    DescribeLiveStreamsBlockListRequest request =new DescribeLiveStreamsBlockListRequest(
    );
    request.setDomainName("<DomainName>");
    DescribeLiveStreamsBlockListResponse response=null;
    try {
        response = client.getAcsResponse(request);
    } catch (ServerException e) {
        e.printStackTrace();
    } catch (ClientException e) {
        e.printStackTrace();
    }
    return response;
}
```

Demo代码主要流程如下：

- i. 初始化Client对象。

SDK需要通过IAcsClient对象完成对OpenAPI的调用，而初始化IAcsClient对象时根据IClient Profile对象配置region、AccessKeyId和AccessKeySecret参数。其中region请一直写cn-hangzhou，因为这里指的是OpenAPI的服务器所在区域。AccessKeyId和AccessKeySecret参数即是账号权限验证。如果是主账号的AccessKeyId和AccessKeySecret，则可以正常使用。如果是子账号的AccessKeyId和AccessKeySecret参数，则需要保证该子账号有视频直播的权限。

ii. 初始化请求Request类。

SDK对每个接口实现了对应的Request类和Response类，对应的恢复推流Request类需要传入哪些参数。API接口定义请参见[恢复直播流推送API](#)。

iii. 发起请求获取结果。

使用Client对象的getAcsResponse方法发起Request对象请求，并根据该方法反馈的反序列化的结果实例化为对应接口的Response类，且在执行该方法时可以catch ServerException和ClientException异常类。

- 当http status>=300且<500，SDK抛ClientException，表示客户端错误。
- 当http status>=500，SDK抛ServerException，表示服务器端错误。
- 如果没有抛出异常，则表示接口调用成功。

15.4. 视频直播鉴权功能相关问题

什么时候需要使用直播鉴权功能？

为了保护您的直播不被盗录、盗播，鉴权功能默认为开启状态，建议您保持默认开启状态。

视频直播的推流端和播流端均提供了地址供用户操作，但该地址为公网地址，即在没有配置安全控制时任何人获知该地址后均可操作。这就导致该地址可能被恶意盗用而导致用户大量计费的情况。

视频播放链接所在域名的根目录下，添加crossdomain.xml文件，其中添加播放器所在域名的权限，默认配置的所有域名均可跨域访问（配置如下）。

```
<!--播流域名crossdomain.xml内容-->
<cross-domain-policy>
<allow-access-from domain="*" />
</cross-domain-policy>
```

因此，为了对推流、播流地址进行安全限制，您可以通过设置鉴权功能的过期时间戳。详情参见[URL鉴权](#)。

视频直播还提供Referer防盗链和IP黑白名单的安全功能，您可以根据需要进行设置。详情参见[访问控制](#)。

如何配置鉴权功能？

控制台：URL鉴权支持默认鉴权和自定义鉴权两种方式。

- 默认鉴权中，鉴权KEY随机分配，有效时长30分钟。超过有效时间，鉴权失效。具体操作，请参见[地址生成器](#)。
- 自定义鉴权：如不采用默认配置，您也可以对鉴权主KEY、副KEY、有效时长进行自定义，并输入原始URL，然后再生成鉴权URL进行推流。具体操作，请参见[配置URL鉴权](#)。

您还可以通过代码获取鉴权URL。

鉴权算法请参见[鉴权URL组成](#)，根据推流地址指定的AppName及StreamName得到未鉴权的播流地址，如 `rtmp://DomainName/AppName/StreamName`；再根据鉴权算法代码实现，填入对应的播流地址计算鉴权URL。

获取鉴权URL的代码示例，请参见[鉴权代码示例](#)。

使用鉴权功能需要注意什么？

- 鉴权功能默认为开启状态，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。如果您想要关闭鉴权功能，需要了解盗刷风险并在控制台URL鉴权页面签署《直播服务关闭URL鉴权免责声明》。
- 鉴权功能默认为开启状态，但是配置鉴权auth_key参数必须需要业务端根据鉴权原理自行实现，暂时没有提供API接口计算鉴权auth_key参数。
- 开启鉴权后推流或播流地址均需要添加鉴权auth_key参数才可正常播放。暂时不支持仅推流或者播流一个地址开启鉴权的功能。
- 鉴权URL提供的是过期时间戳的限制，即在该时间段内该地址可以一直访问，不会有任何限制，鉴权还无法实现设置某个URL仅能够一次访问的功能。
- 鉴权auth_key参数是根据去掉queryString后的URI计算MD5值后拼接得到的，参见前述设置鉴权参数小节。因此，推流和播流地址的UR均为AppName/StreamName；这样会导致推流和播流地址的鉴权auth_key参数其实是一致的。建议用户业务逻辑中如果推流地址有泄漏的情况下设置过期时间戳在业务许可范围内尽量短一些，避免播流地址被恶意访问。
- 鉴权仅会在推流或者播流开始的时候进行验证，在推流或者播流过程中即不会验证，也就是说推流或者播流过程中如果超过了鉴权时间戳也可以继续播放。