

LSS 文档



【版权声明】

版权所有©百度在线网络技术（北京）有限公司、北京百度网讯科技有限公司。 未经本公司书面许可，任何单位和个人不得擅自摘抄、复制、传播本文档内容，否则本公司有权依法追究法律责任。

【商标声明】



和其他百度系商标，均为百度在线网络技术（北京）有限公司、北京百度网讯科技有限公司的商标。 本文档涉及的第三方商标，依法由相关权利人所有。未经商标权利人书面许可，不得擅自对其商标进行使用、复制、修改、传播等行为。

【免责声明】

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导。 如您购买本文档介绍的产品、服务，您的权利与义务将依据百度智能云产品服务合同条款予以具体约定。本文档内容不作任何明示或暗示的保证。

目录	
目录	2
产品动态	5
产品简介	5
产品概述	5
产品优势	6
核心概念	6
功能特性	7
应用场景	9
产品计费	10
计费概述	10
直播基础服务费用	11
标准直播服务费用	11
低延时直播服务费用	13
直播增值服务费用	15
直播转码	15
直播截图	16
直播转推	16
直播时移	16
直播审核	17
直播录制	17
预付费资源包	18
购买流程	18
计费方式变更	19
余额不足提醒和欠费处理	20
账单	20
退款说明	21
快速入门	21
快速入门流程	21
开始使用LSS	22
创建推流和播放域名	23
低延时直播	24
低延时直播概述	24
低延时直播和标准直播区别	25
低延时直播快速入门	25
SDK集成指南	28
SDK集成指南概述	28
低延时传输层SDK	28
低延时全功能SDK	43
SDK接入说明	48
低延时直播RTL SDK开发者个人信息保护合规指引	49

Baidu 百度智能云文档	目录
低延时直播RTL SDK隐私政策	52
控制台操作指南	60
LSS概览	60
操作指引	60
开通LSS	63
域名管理	64
添加域名	64
开启域名	66
配置HTTPS	67
配置CNAME	68
配置转推功能	71
关闭域名	72
删除域名	73
时移&回看功能	73
流管理	75
新建流	75
查询流信息	77
质量监控	78
模板管理	80
转码模板	80
录制模板	82
水印模板	84
缩略图模板	87
通知模板	88
黄反审核	89
安全策略	92
统计分析	97
直播域名	97
用量查询	99
用户行为	100
日志管理	104
标签管理	105
多用户访问控制	112
Demo体验	116
推流端SDK	116
Android端推流SDK	116
简介	116
推流-SDK-开发指南	118
iOS端推流SDK	124
简介	124
推流-SDK-开发指南	125

播放器SDK	130
服务端SDK	131
Java-SDK	131
安装LSS-Java-SDK	131
LssClient	132
stream管理	134
统计功能	136
通知功能	138
录制功能	139
PHP-SDK	139
安装LSS-PHP-SDK	139
安装LssClient	140
LSS开发指南	146
异常处理	150
API参考	151
概述	151
使用须知	152
错误返回	153
公共头域	157
Stream接口	158
模板接口	178
统计接口	198
通知接口	240
轮播接口	246
录制视频裁剪	246
日志下载接口	249
文档更新记录	251
视频专区	251
产品简介	251
操作指南	252
常见问题	252
常见问题总览	252
使用类问题	254
性能类问题	261
计费类问题	266
API、SDK使用类问题	267
安全类问题	267
服务等级协议SLA	267
LSS服务等级协议SLA	268

产品动态

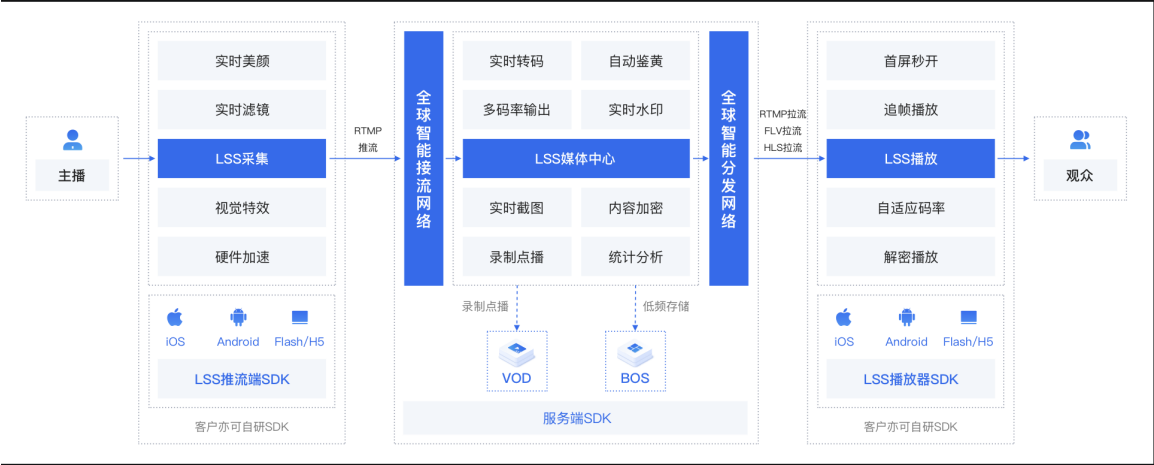
发布日期	动态名称	动态描述	相关文档
2023-12	低延时直播	低延时直播传输层SDK发布，支持轻量化接入	低延时直播
2023-11	直播转码	直播转码支持转码模板快速检索	直播转码
2023-09	直播智感超清转码	直播智感超清转码支持自适应分辨率	直播转码
2023-07	直播截图审核	直播截图审核功能优化，支持涉黄标签细分	直播截图审核
2023-05	直播数据监控	直播播放数据接入BCM，支持域名用量监控	用量监控
2023-04	低延时直播	新增低延时直播功能，提供更低延时、更强互动的直播体验。	低延时直播
2022-06	用量查询	用量查询支持转推和低延时直播数据查询。	用量查询
2022-05	智能审核	云导播台发布智能审核功能，支持对添加的每个视频源进行审核，实时检测直播过程中的政治敏感、色情、暴恐违禁等违规内容，支持一键切换备播，为直播安全保驾护航。	智能审核
2022-05	键盘快捷键	云导播台支持键盘快捷键操作，一键切换预览/预览，操作无忧。	快捷键
2022-02	云导播台	新增云导播台功能。在云端实现直播流的切换、实时编辑、多画面的混流播出等功能，为活动直播、赛事直播、广电节目直播等多场景提供智能导播服务。	云导播台
2021-11	录制计费	录制全面开始计费	录制计费
2018-10	推拉流多对多关联	直播支持多推流域名、多播放域名。不同业务场景下需要转推其他直播流时，LSS 能够支持添加一个或多个推流域名。	域名管理
2018-09	预付费流量包	发布预付费流量包，相同流量的流量包价格比后付费流量有优惠折扣。	流量包

产品简介

产品概述

音视频直播LSS（Live Streaming Service）为用户提供便捷接入、稳定流畅、超低延时、高并发的音视频直播服务。根据不同的直播场景需求，LSS提供标准直播、低延时直播、云导播台服务，为用户提供一站式的音视频直播解决方案。

 产品架构图



- 1. 主播通过推流SDK推送直播流到百度全球智能接流网络，LSS直播服务通过边缘推流的方式将直播流推送至媒体中心。
- 2. 直播流推送至媒体中心后，用户可按需对直播流进行转码、时移、录制、截图等云端处理。
- 3. 经媒体中心处理好的直播流通过全球智能分发网络下发至观众的设备中进行播放。

子产品简介

子产品	产品描述	应用场景
标准直播	依托百度智能云全球海量加速节点和领先的实时音视频 AI 技术，提供便捷接入、稳定流畅、高并发的直播接入和分发服务。	适用于秀场直播、游戏直播、广电媒体、电商直播
低延时直播	低延时直播是传统标准直播在超低延时场景下的延伸，比传统直播延迟更低，为观众提供毫秒级延迟的直播观看体验，显著提升直播的互动性。	适用于电商直播、在线教育、体育赛事等
云导播台	依托百度智能云领先的音视频直播、媒体处理和AI技术，将线下传统导播能力云端化，提供简单易用的云端导播服务，助力用户打造更好的直播体验。	适用于广电媒体、游戏赛事、电商直播、在线教育、活动直播。

产品优势

丰富的带宽储备和节点覆盖

依托百度智能云遍布全球的3000+CDN节点和150Tb+储备带宽，满足不同直播场景的上传分发需求，保障国内外的用户均可获得极致直播观看体验。

领先的人工智能核心技术

基于百度智能云的图像识别、语音识别、自然语言处理、人脸检测、表情迁移和图像融合等人工智能核心技术，提供内容审核、视觉特效和实时美颜能力。

端到端的全平台场景化SDK

提供从视频直播采集端到播放端一站式全套SDK及API，您可以根据实际业务需求将直播服务无缝对接到自己的业务平台，支持PC/Android/iOS/Web等各大主流平台。

精细化流控

实时统计流级别直播流量消耗、请求、并发等数据，通过详细的统计数据，实现精细化指标统计、实时数据监控和智能根因分析，助力用户快速进行业务拓展。

高效降本

提供完整的 H264和 H265智感超清转码方案，保证用户观看体验的同时，视频压缩率达到 20%~50%，实现高效降本。

核心概念

感谢您选择百度智能云音视频直播LSS服务，音视频直播服务 LSS 采用域名（Domain）、应用（App）、流（Stream）三级直

播管理单元，一个域名（Domain）下可包含多个应用（App），一个应用（App）下可包含多个流（Stream）。您可以通过本教程快速了解音视频直播LSS的基本概念。

🔗 基本概念

关键词	说明
推流	主播将采集到的视频流和音频流推送到百度全球智能接流网络就近的边缘服务器。
拉流	即直播流播放，是指观众通过指定地址从百度智能云边缘服务器拉取视频流和音频流。由于是实时直播，所以播放器在播直播视频的时候是没有进度条的。
推流域名	主播必须通过推流域名来推送直播流。推流域名必须在使用直播服务前完成注册并备案。配置完推流域名后，直播服务会生成对应的推流地址。
播放域名	观众用于播放直播流的域名，播放域名必须在使用直播服务前完成注册并备案。配置完播放域名后，直播服务会生成对应的播放地址。
域名 CNAME	CNAME是在百度智能云直播控制台接入加速域名后，系统给对应的域名分配一个以 .bcelive.com 为后缀的域名。用户需要在DNS解析服务商处配置一条 CNAME 记录，记录生效后，域名解析的工作就正式转向百度智能云云直播，该域名所有的请求都将转向百度智能云直播的边缘节点。
Stream Name	StreamName 是一路流的标识符，通常与域名、AppName一起唯一标识一路流。
AppName	直播的应用名称，通常与域名、StreamName一起唯一标识一路流。
直播转码	直播转码是将视频码流转换成另一个视频码流的功能。通过转码，可以改变原始码流的编码格式、分辨率和码率等参数，从而适应不同终端和网络环境的播放。
H.264	H.264 是由 ITU-T 视频编码专家组和 ISO/IEC 动态图像专家组联合提出的高度压缩数字视频编解码器标准，拥有低码率、图像质量高、容错能力强和网络适应性强等优点。
H.265	H.265 是在现有的 H.264 视频编码标准基础上继续技术优化，可以改善码流、编码质量、延时和算法复杂度之间的关系，达到最优化设置。
直播录制	在推流过程中，将实时的直播原始流经过转音视频封装（不修改音频、视频数据以及对应的时间戳等信息）得到的视频文件存储到音视频处理平台。
水印	为保障视频版权不受侵犯，在直播推流过程中，将设置好的水印合并到视频流中输出一个带有水印的视频流，水印内容可以为文字或图片。
截图	以固定时间间隔将直播推流视频画面截取下来，形成图片文件存储在对象存储 BOS 中。
鉴黄	系统通过推流域名已关联的截图鉴黄模板，对截图进行黄反识别和结果回调。
95峰值带宽	取计费周期内所有的5分钟带宽计数点，按大小排序后去掉前5%点，剩余95%的计数点取最大值即为月95峰值带宽。

功能特性

本文提供音视频直播LSS产品功能说明，具体功能描述和使用方法，请查询对应详细文档。

🔗 直播推流

功能名称	描述
推流协议	支持主流RTMP协议推流到LSS。
推流方式	支持百度智能云iOS、Android、Web等推流端SDK及Demo，也支持常见的第三方推流软件，如OBS/XSplit/FMLE等。
推流设备	支持常见的第三方RTMP推流硬件、编码器以及盒子等设备。

直播向第三方源站拉流

功能名称	描述
拉流协议	支持RTMP、FLV和HLS三种协议回第三方源站拉流。

直播播放

功能名称	描述
播放协议	支持RTMP、FLV和HLS三种播放协议。
播放方式	支持百度智能云iOS、Android、Web等推流端SDK及Demo，也支持常见的第三方播放软件，如VLC等。

直播流处理

功能名称	描述
转码	支持H264、H265标准转码，支持普通转码和智感超清转码。转码支持多码率播放。
录制	支持将直播流录制到百度智能云点播平台和云存储，录制格式支持MP4、M3U8和FLV格式，支持自定义录制文件名。
水印	支持对直播流添加时间水印和图片水印，可自由调整水印位置。
缩略图	支持直播截图到云存储，截图支持jpg和gif格式。
黄反审核	支持直播鉴黄，支持鉴黄后的回调通知。

直播管理

功能名称	描述
管理方式	支持直播控制台图形化管理和API管理。

直播控制台

功能名称	描述
概览	可查看用户账号下的带宽/流量、转码等数据。
域名管理	可对域名进行增删改查、域名停用等操作。
模板管理	可对转码、录制、黄反审核、缩略图、水印等模板的增删改查。
统计分析	可查看流量、带宽、请求数、转码时长等统计数据。
质量监控	可查看流级别的上下行质量数据。
流量包	可购买流量包、查看当前流量包使用状况。
日志管理	可对直播日志进行下载和管理。

直播安全

功能名称	描述
推流鉴权	支持推流URL鉴权，支持自行设置鉴权超时时间。
播放鉴权	支持播放URL鉴权、Refer防盗链、IP黑白名单防盗链。

统计分析

功能名称	描述
带宽	支持域名粒度的上下行带宽统计。
流量	支持域名粒度的上下行流量统计。
转码时长	支持域名级别转码时长统计。
用户行为	支持流级别的主播、观众数统计。

🔗 质量监控

功能名称	描述
流质量	支持流级别的上下行流畅度监控。
流信息	支持流级别的码率和帧率监控。

🔗 API管理

功能名称	描述
流管理API	支持新建、查询、删除、封禁、重置某条流；支持添加流的metadata信息；支持更新流的模板配置。
模板管理API	支持录制、水印、缩略图等模板的查询、创建、删除。
统计API	支持查询用户带宽、流量、请求数、转码时长；支持查询域名级别的上下行带宽、流量；支持查询流级别的上下行带宽、流量、实时在线人数。
通知API	支持流状态变更、录制、缩略图、黄反审核等通知接口的创建、删除和查询。
录制视频裁剪API	支持对已录制的视频进行裁剪。

🔗 直播SDK

功能名称	描述
推流端SDK	支持ios、andriod推流SDK，包含美颜、滤镜、推流等功能。
服务端SDK	支持java、php版本的SDK，包含流统计、录制、通知等功能。
播放端SDK	支持web、ios、andriod播放SDK。

🔗 海外直播

功能名称	描述
海外分发	目前已支持中国、香港、台湾、新加坡、美东、美西、德国、日本、泰国多个国家和地区的接流和分发，轻松实现跨境直播。如您有其他国家的直播需求，可提交 工单 咨询。

应用场景

百度智能云音视频直播服务适用于各类直播场景，比如娱乐互动、电商直播、媒体直播、教育直播、企业直播等。

🔗 娱乐互动

面向游戏、秀场、唱吧等直播场景，为直播提供直播秒开、色彩滤镜、主播观众直播连麦等功能，功能便捷接入，开启即用，提高直播人气，活跃直播气氛。

电商直播

电商直播方案可以使商家更加全面的传递商品信息，用户在观看直播中边看边买，降低营销成本，增加成交量。



媒体直播

面向媒体场景支持4K、8K等高清分辨率，配合智感超清转码功能，在相同码率下实现更高画质。使用云导播台功能，还可实现多视频分屏、添加台标、双语字幕和紧急备播等功能。



教育直播

超低延时直播能够提升在线教育的实时互动性，动态防盗链保障防止精品内容泄漏。



企业直播

提供IP黑白名单等直播内容保护功能，为企业直播内容保驾护航。同时配合直播录制，可将企业宣讲、企业培训等内容保存云端，方便回看。



产品计费

计费概述

计费总览

音视频直播LSS的服务费用包括直播基础服务费用、直播增值服务费用和其他云产品服务费用。费用组成如下：

- 基础服务费用：使用直播（包括标准直播、低延时直播）后产生的直播消耗费用，标准直播和低延时直播支持流量或带宽两种计费方式切换。



- 增值服务费用：使用直播转码、录制、截图、审核、时移、拉流转推等直播增值功能产生的云服务费用，此类功能默认关闭，使用才收费。



- 其他云产品费用：结合百度智能云其他产品一起提供的增值功能，由其他云产品根据各自的计费规则分别收取相关费用。

🔗 计费方式

直播支持按量计费和资源包 2 种计费方式。

- 按量计费：后付费方式，按照各计费项的实际用量结算费用；
- 资源包：预付费方式，预先购买资源包，在费用结算时，优先从资源包抵扣用量，超出资源包额度部分将根据按量计费规则进行结算。

🔗 计费说明

🔗 基础直播服务费用

计费项	计费方式	付费方式
标准直播计费	按使用流量计费	后付-按量计费、预付-资源包
标准直播计费	按日峰值带宽计费	后付-按量计费
低延时直播计费	按使用流量计费	后付-按量计费
低延时直播计费	按日峰值带宽计费	后付-按量计费

直播增值服务费用

计费项	计费方式	付费方式
转码计费	区分转码类型，按转码时长计费	后付-按量计费、预付-资源包
录制计费	按直播录制并发峰值路数计费	后付-按量计费
截图计费	按截图张数计费	后付-按量计费
时移计费	区分时移回看规格，按时移数据写入量计费	后付-按量计费
审核计费	按审核截图的张数计费	后付-按量计费
拉流转推计费	按转推带宽计费	后付-按量计费

直播基础服务费用

标准直播服务费用

🔗 标准直播服务费用

标准直播支持按流量计费、带宽峰值计费和月95带宽计费三种按量计费方式，您可以根据实际业务场景选择合适的计费方式。

计费方式	适用场景
按流量计费	适用于流量曲线波动较大，单日里仅某些时刻产生流量的场景。
按峰值带宽计费	适用于流量曲线比较平稳，单日里产生的播放流量在各个时间段分布比较均衡的业务场景。
按月95带宽计费	默认不提供该计费方式，若您月消费金额大于或等于10万元，可以联系商务洽谈使用该计费方式，变更后将在次月1日生效。

注意事项

- 新用户开通标准直播后，默认采用流量日结计费。
- 流量/带宽的单位换算进制为1000，例如：1TB = 1000GB。
- 默认只收取下行播放费用，针对上下行使用不均衡的业务场景（下行播放：上行推流 < 50:1），会按照实际推流用量额外收取推流费用。

按流量计费 计费价格 按流量计费的详细价格信息，请参考[“音视频直播LSS价格说明”](#) ->直播基础服务计费->标准直播->按流量计费”。

计费说明

项目	说明
计费规则	按日到达阶梯计费，统计一个自然日内的流量累计值乘以对应阶梯单；流量统计进制为 1000，例如：1GB = 1000MB。
付费类型	后付费
计费周期	按小时出账单，出账单时间是计费周期结束后1小时内。如：10:00-11:00的账单将在12:00之前生成，具体以系统出账时间为准。

计费示例 假设您在6月5日之前没有流量消耗，在6月5日消耗的流量为9TB，6月6日消耗流量为10TB；根据以月为周期，超额累进”原则计价，6月5日至6月6日的流量计费如下表所示

日期	流量用量	计费
6月5日	9TB	9TB 按0GB-10TB（不含）档位计费，即 $9 \times 1000GB \times 0.24 \text{ 元/GB} = 2160 \text{ 元}$
6月6日	10TB	10TB流量中，有1TB落在0GB-10TB（不含）档位，剩余的9TB落在10TB-50TB（不含）档位，即 $1 \times 1000GB \times 0.24 \text{ 元/GB} + 9 \times 1000GB \times 0.23 \text{ 元/GB} = 2310 \text{ 元}$

按峰值带宽计费 计费价格 按峰值带宽计费的详细价格信息，请参考[“音视频直播LSS价格说明”](#) ->直播基础服务计费->标准直播->按流量计费”。

计费说明

项目	说明
计费规则	按日到达阶梯计费，统计一个自然日内的带宽峰值乘以对应阶梯单价。带宽统计进制为 1000，例如：1Gbps = 1000Mbps。
付费类型	后付费
结算周期	按日结算

计费示例 客户6月5日的峰值带宽为300Mbps，6月6日的峰值带宽为600Mbps，6月5日至6月6日的安峰值带宽计费如下表所示

日期	峰值带宽用量	计费
6月5日	300Mbps	300Mbps 按0-500Mbps（不含）档位计费，即300× 0.6元/Mbps=180元
6月6日	600Mbps	600Mbps按500Mbps-5Gbps（不含）档位计费，即600× 0.58元/Mbps=348元

按 95 峰值带宽计费 计费价格 如果您想使用按 95 峰值带宽计费的方式，请联系商务进行变更。变更后将在次月 1 日生效。

计费说明

项目	说明
计费规则	按照推拉流所在地区统计当月的 95 峰值带宽。费用的计算方式为：95 峰值带宽 × 单价 × 有效因子 = 当月费用。 95 峰值带宽取值：在有效计费周期内，取每 5 分钟的带宽值，去除 5% 的最高带宽点，剩下的最高点即为 95 峰值带宽。有效因子 = 当月有效计费天数/自然月天数。提示带宽统计进制为 1000，例如，1Gbps = 1000Mbps。
付费类型	后付费
结算周期	按月结算

低延时直播服务费用

低延时直播服务费用

低延时直播支持按流量计费、带宽峰值计费和月95带宽计费三种按量计费方式，您可以根据实际业务场景选择合适的计费方式。

计费方式	适用场景
按流量计费	适用于流量曲线波动较大，单日里仅某些时刻产生流量的场景。
按峰值带宽计费	适用于流量曲线比较平稳，单日里产生的播放流量在各个时间段分布比较均衡的业务场景。
按月95带宽计费	默认不提供该计费方式，若您月消费金额大于或等于10万元，可以联系商务洽谈使用该计费方式，变更后将在次月1日生效。

注意事项

- 新用户开通低延时直播后，默认采用流量日结计费。**
- 流量/带宽的单位换算进制为1000，例如：1TB = 1000GB。
- 由于低延时直播采用超低延迟链路，流量/带宽费用略高于标准直播。
- 默认只收取下行播放费用，针对上下行使用不均衡的业务场景（下行播放：上行推流 < 50:1），会按照实际推流用量额外收取推流费用。

按流量计费 计费价格 按流量计费的详细价格信息，请参考["音视频直播LSS价格说明"](#) ->直播基础服务计费->低延时直播->按流量计费”。

计费说明

项目	说明
计费规则	按日到达阶梯计费，统计一个自然日内的流量累计值乘以对应阶梯单；流量统计进制为 1000，例如：1GB = 1000MB。
付费类型	后付费
计费周期	按小时出账单，出账单时间是计费周期结束后1小时内。如：10:00-11:00的账单将在12:00之前生成，具体以系统出账时间为准。

计费示例 假设您在6月5日之前没有流量消耗，在6月5日消耗的流量为9TB，6月6日消耗流量为10TB；根据'以月为周期，超额累进'原则计价，6月5日至6月6日的流量计费如下表所示

日期	流量用量	计费
6月5日	9TB	9TB 按0GB-10TB（不含）档位计费，即 $9 \times 1000GB \times 0.48 \text{ 元/GB}$ =4320元
6月6日	10TB	10TB流量中，有1TB落在0GB-10TB（不含）档位，剩余的9TB落在10TB-50TB（不含）档位，即 $1 \times 1000GB \times 0.48 \text{ 元/GB} + 9 \times 1000GB \times 0.46 \text{ 元/GB}$ =4620元

按峰值带宽计费 计费价格 按峰值带宽计费的详细价格信息，请参考["音视频直播LSS价格说明"](#) ->直播基础服务计费->低延时直播->按流量计费”。

计费说明

项目	说明
计费规则	按日到达阶梯计费，统计一个自然日内的带宽峰值乘以对应阶梯单价。带宽统计进制为 1000，例如：1Gbps = 1000Mbps。
付费类型	后付费
结算周期	按日结算

计费示例 客户6月5日的峰值带宽为300Mbps，6月6日的峰值带宽为600Mbps，6月5日至6月6日的安峰值带宽计费如下表所示

日期	峰值带宽用量	计费
6月5日	300Mbps	300Mbps 按0-500Mbps（不含）档位计费，即 $300 \times 1.2 \text{ 元/Mbps}$ =360元
6月6日	600Mbps	600Mbps按500Mbps-5Gbps（不含）档位计费，即 $600 \times 1.16 \text{ 元/Mbps}$ =696元

按 95 峰值带宽计费 计费价格 如果您想使用按 95 峰值带宽计费的方式，请联系商务进行变更。变更后将在次月 1 日生效。

计费说明

项目	说明
计费规则	按照推拉流所在地区统计当月的 95 峰值带宽。费用的计算方式为：95 峰值带宽 × 单价 × 有效因子 = 当月费用。 95 峰值带宽取值：在有效计费周期内，取每 5 分钟的带宽值，去除 5% 的最高带宽点，剩下的最高点即为 95 峰值带宽。有效因子 = 当月有效计费天数/自然月天数。提示带宽统计进制为 1000，例如，1Gbps = 1000Mbps。
付费类型	后付费
结算周期	按月结算

直播增值服务费用

直播转码

直播转码费用

音视频直播LSS转码目前提供普通转码和智感超清转码服务。与普通转码相比，智感超清转码采用更复杂的编码算法，在画质不变或增强的情况下，码率能比普通转码降低20%-50%左右。

注意事项

- 转码功能支持后付费和购买**预付费资源包**。
- 相同转码时长的转码包价格比后付费转码有优惠折扣。如果您业务转码长期稳定、使用时长可预估，建议购买转码包价格优惠。

后付费 计费价格 转码费用的详细信息，请参考["音视频直播LSS价格说明"](#) ->转码计费->后付费。

计费说明

项目	说明
计费规则	转码费用 = 普通转码费用 + 智感超清转码费用。 普通转码费用 = 普通转码总时长 x 普通转码单价。 智感超清转码费用 = 智感超清转码总时长 x 智感超清转码单价。
计费类型	后付费
计费周期	按小时出账单，出账单时间是当前计费周期结束后1小时内。例如，10:00-11:00的账单会在12:00之前生成，具体以系统出账时间为准。

计费示例 若您在2023年1月1日使用了转码服务，其中 A 直播流使用普通转码720P（H.264）线路 L1 转码60分钟，使用普通转码480P(H.264)线路 L2 转码30分钟，B 直播流使用智感超清转码720P(H.264)线路 L3 转码60分钟，则您1月2日需要支付的直播转码账单如下：

类型	计费
A普通转码费用	0.032（元/分钟）x 60（分钟）+ 0.015（元/分钟）x 30（分钟）=2.37元
B智感超清转码费用	0.125（元/分钟）x 60（分钟）=7.5元

预付费 计费价格 转码费用的详细信息，请参考["音视频直播LSS价格说明"](#) ->转码计费->转码包预付费。

计费说明

- 计费方式：预付费
- LSS转码产生的费用优先扣除转码包，超出部分按转码时长计费。
- 每个转码包只能扣除相应规格的转码时长费用。
- 转码包有效期为12个月，购买后立即生效。每个转码包的有效期独立计算，多个转码包有效期不会叠加。
- 用户可同时购买多个转码包，当用户购买多个转码包时，优先抵扣过期时间最早的转码包，其它的依次顺延，当所有转码包扣减完毕后，自动转为按量计费的方式扣减。
- 用户购买转码包7天内，若未使用、则可全额退款；若已使用或超出7天，则不支持退款。用户可在控制台 ->财务中心 -> 退订管理菜单直接操作退款。
- 账单周期：与当前LSS转码计费周期一致，实时从转码包扣除已使用转码时长。

直播截图

直播截图费用

百度音视频直播LSS可对直播流进行截图处理，截图将存储于对象存储BOS中。使用直播截图功能会产生费用，以当月累计截图数量为结算标准。

注意事项

- 截图功能默认关闭，可通过控制台或云API开启。
- 直播截图存储于对象存储BOS中，使用后将产生BOS存储费用，详情请参见[BOS产品计费定价](#)。

计费价格

截图费用的详细信息，请参考["音视频直播LSS价格说明"](#) ->增值服务计费->截图计费。

计费说明

项目	说明
计费规则	按截图张数计费
计费方式	后付费
计费周期	按小时出账单，出账单时间是计费周期结束后1小时内。如：10:00-11:00的账单将在12:00之前生成，具体以系统出账时间为准

直播转推

直播转推费用

转推是指直播流被推送到CDN节点后，由CDN节点将该流转推至您的源站或者您指定的直播平台。

注意事项

- 开启转推功能将产生转推服务费。
- 以所有转推第三方的并发带宽值进行计费，默认按日峰值带宽方式计费。
- 若账号下LSS为其他月带宽计费方式，则转推第三方的带宽计费方式会跟随LSS的计费方式。

计费价格

转推费用的详细信息，请参考["音视频直播LSS价格说明"](#) ->增值服务计费->转推计费。

计费说明

项目	说明
计费规则	按转推带宽计费
计费方式	后付费
计费周期	以月为单位出账，本月账单次月生成，具体以系统出账时间为准

直播时移

直播时移费用

百度智能云 LSS 基于 HLS 直播实现时移功能，用户可以实时选择从开播后的某个过往时间点开始进行回看，从而达到播放之前直播内容的效果。

注意事项

- 直播时移功能默认关闭，可通过控制台或云 API 开启。
- 播放时移内容产生的下行带宽、流量费用按 LSS下行播放带宽、流量进行收费。保存不同天数档位使用的存储量不额外计费。

计费价格

直播时移费用的详细信息，请参考["音视频直播LSS价格说明"](#) ->增值服务计费->时移计费。

计费说明 | 项目 | 说明 | | --- | --- | | 计费规则 | 按时移数据写入量计费 | | 计费方式 | 后付费 | | 计费周期 | 按日计费，实时扣费（每日零点后出前一日账单并扣费，具体出账时间以系统为准） |

直播审核

直播审核费用

LSS直播可对直播流进行鉴黄审核，因鉴黄需首先对直播流做截图处理，故使用鉴黄审核功能会产生 鉴黄 和 截图 两笔费用。直播审核以当月累计鉴黄审核张数为结算标准。

注意事项

- 截图功能可单独开启使用，但审核功能需开启截图后才能开启，不可单独使用。
- 截图与鉴黄为收费功能，启用后，截图功能按每千张 ¥ 0.1元收费；鉴黄功能按每千张 ¥ 1.1元收费。请参考["音视频直播LSS价格说明"](#) ->增值服务->黄反审核计费。
- 截图与鉴黄图片存储至您的对象存储BOS中，将产生BOS存储费用。详情请参见[BOS产品计费定价](#)。

计费价格 截图审核费用的详细信息，请参考["音视频直播LSS价格说明"](#) ->增值服务计费->黄反审核计费。

计费说明

项目	说明
计费规则	按直播审核张数计费
计费方式	后付费
计费周期	以月为单位出账，本月账单次月生成，具体以系统出账时间为准

直播录制

直播录制费用

百度音视频直播LSS服务可将直播流录制下来，并存储至对象存储 BOS 中。使用直播录制功能会产生费用，分为录制路数峰值费用和 BOS 存储服务费用。

注意事项

- 录制功能默认关闭，可通过控制台或API开启。
- 录制的视频文件默认保存至对象存储BOS，将产生对象存储费用。
- 1路流同时转成MP4和FLV格式录制时，按2路计算。

计费价格 直播录制费用的详细信息，请参考["音视频直播LSS价格说明"](#) ->增值服务计费->录制计费。

计费说明

项目	说明
计费规则	直播录制并发峰值路数
计费方式	后付费
计费周期	以月为单位出账，本月账单次月生成，具体以系统出账时间为准

预付费资源包

资源包类型

音视频直播LSS支持预付费资源包，包括直播流量包和直播转码包，所有资源包支持叠加使用。

预付费资源包	费用说明	计费价格
直播流量包	支持抵扣标准直播在中国大陆地区下行产生的日结/月结流量费用。	直播流量资源包
标准转码包	支持抵扣使用直播标准转码服务产生的转码时长费用。	直播转码资源包
智感超清转码包	支持抵扣使用智感超清转码服务产生的转码时长费用。	直播智感超清转码资源包

注意事项

- 资源包有效期为 12 个月，到期未使用的资源将自动作废。
- 资源包支持使用的区域为中国大陆，其他区域暂不支持。
- 您消耗的各项直播资源，将优先在相应的资源包内抵扣，超出资源包的部分将根据直播基础服务费用和转码计费规则计费。
- 直播流量资源包只支持标准直播的下行抵扣。
- 资源包支持叠加购买，即有效期内资源用量叠加，多个资源包按购买时间先后进行抵扣。
- 视频直播提供的资源包，仅可抵扣视频直播产品对应付费服务产生的用量。
- 资源包购买当日生效。当日购买的资源包，无法抵扣当日结算的（即前一计费周期产生的）费用。
- 用户购买资源包 7 天内，若未使用，可申请全额退款；若已使用或超出 7 天，则不支持退款。

购买流程

计费方式

音视频直播支持预付费和后付费两种方式，您可以根据业务情况灵活选择最优的计费方式。

1. 注册百度智能云账号

若您已经注册百度智能云账号，可忽略此步骤。若您未注册百度智能云账号，可参考[账号注册](#)进行注册。

2. 完成实名认证

若您已经在百度智能云完成实名认证，可忽略此步骤。若您未完成实名认证，可参考[实名认证](#)完成认证。完成个人实名或企业实名认证后，均可申请开通音视频直播服务。

3. 开通服务

登录[音视频直播 LSS 控制台](#)，在音视频直播 LSS 开通页面选择**计费方式**并同意 [《百度音视频直播服务LSS使用协议》](#)，点击**立即开通**即可完成购买。

开通音视频直播LSS

产品名称：

音视频直播LSS

分发计费方式：

按使用流量计费

按日峰值带宽计费

分发计费项：

下行流量

计费价格：

开通即送20G流量

点击查看计费文档>>

流量阶梯	价格（元/GB）
0GB-10TB（不含）	0.24
10TB-50TB（不含）	0.23
50TB-100TB（不含）	0.21
100TB-1PB（不含）	0.18
大于1PB	0.15

转码价格：

0.133元/分钟/转码线路

点击查看计费文档>>

关联开通：

使用音视频直播LSS服务时，可能会涉及直播录制，为方便录制视频的点播和存储，帮您同时开通如下服务（不使用相关服务的情况下不会产生任何费用）：

☒ BOS

☒ VOD

☒ 同意《百度音视频直播服务LSS使用协议》

立即开通

计费方式变更

🔗 注意事项

- 按流量计费和按日峰值带宽计费可以相互切换，提交切换申请后，次日零点生效。
- 目前变更计费方式只支持企业认证用户，如您需要变更计费方式，请提前完成企业认证。如您有其他计费变更需求，请提交[工单](#)。
- 如果您是百度智能云大客户，可以联系商务经理协助您变更计费方式。

🔗 计费方式变更

- 登录[音视频直播LSS控制台](#)。
- 点击左侧导航栏**域名管理**，在域名管理页面可以查看自己当前的计费方式。



- 如需变更计费方式，请点击**变更**按钮。
- 点击变更后，在弹出框中选择您需要变更的计费方式，如确认变更请点击**确认**。

变更计费方式

×

当前配置

计费方式：按日峰值带宽计费

计费项：日峰值带宽

0.75元 / Mbps / 日起，分段阶梯计费

▶

变更配置

计费方式：按使用流量计费

计费项：下行流量

0.23元 / GB起，分段阶梯计费

[查看详细计费规则](#)

确定

取消

余额不足提醒和欠费处理

欠费预警

当您的账户余额不足时，百度智能云会提前发送续费提醒，请您及时为百度智能云账号进行充值，避免欠费停服。

- 提前3天预警：根据您最近3天的账单金额来判断您的账户余额（含可用代金券）是否足够支付未来3天的费用，若不足以支付，百度智能云将发送续费提醒。
- 提前1天预警：根据您最近1天的账单金额来判断您的账户余额（含可用代金券）是否足以支付未来1天的费用，若不足以支付，百度智能云将发送续费提醒。

欠费停服

当您的账户余额不足0元无法支付LSS账单时，账号进入欠费状态无法正常使用音视频直播服务，百度智能云会发送欠费停服通知，您需补缴所有欠费账单后方可继续使用。

回收说明

针对停服超过12个月的欠费账号，百度智能云音视频直播自动回收该账号下的直播资源，定期清除该账号下直播相关数据和记录（包括域名配置、服务日志、资源包等），清除后不支持数据找回。

账单

若需要查看百度音视频直播LSS产生的账单及扣费明细，可前往财务中心 > [消费总览](#) 查看消费账单。

资源账单

1. 单击账单明细进入账单页。
2. 单击账单明细页面的资源账单页签。
3. 单击全部产品筛选框，选择音视频直播LSS，即可查看LSS的资源ID账单。

流水账单

资源账单

计费项账单

拆分项账单

操作指南

温馨提示:

资源账单可查询云资源每一计费周期用量及账单金额，如果您需要查询产品拆分项信息，如 BOS 产品"bucket"，请移步至“拆分项账单”查询。

2023年

1月

音视频直播 LSS

实例ID

请输入实例ID进行搜索

Q

↓

↺

资源ID	产品名称	区域	付费方式	账单金额	应付金额	资源标签	操作
-	音视频直播 LSS	全局	后付费	¥ 340.00	¥ 170.00	-	查看详情

每页显示

20

<

1

>

🔗 计费项账单

1. 单击账单明细进入账单页。
2. 单击账单明细页面的计费项账单页签。
3. 单击全部产品筛选框，选择音视频直播LSS，即可查看LSS的具体计费项账单。

流水账单

资源账单

计费项账单

拆分项账单

操作指南

温馨提示：

计费项账单仅包含后付费产品的分计费项账单，账单周期与该产品流水账单周期一致。

2023

1月

音视频直播 LSS LSS

区域-全部

↓

↺

资源ID	产品名称	区域	账单周期	计费项	用量	配置	单价	账单金额
-	音视频直播 LSS	全局	2023-01-01 00:00-2023-01-31 23:59	直播录制峰值	17路数	-	20元/路数	340

退款说明

🔗 7天无理由退款

为了方便您使用音视频直播，如果您7天内购买的资源包未被使用，百度智能云支持7天内无理由退款，其他情况不支持退款。

🔗 退款说明

对于单个账号而言，未使用的资源包新购之日起7天内（含7天），符合7天无理由退款 场景，可支持7天内申请退款，退款金额为购买时花费的全部消耗金额，包括现金账户金额、收益转入账户金额以及赠送账户金额。以下情况不在7天无理由退款范围内：

- 资源包新购之日到目前为止已超过7天（不包括7天），该资源包不支持退款。
- 已使用的资源包，不论是否使用完毕，均不支持7天内无理由退款。
- 赠送的资源包不支持7天内无理由退款。
- 若用户账户因内容违规导致停服，账户下所有资源包均不支持退款。

🔗 退款途径

抵扣或代金券不予以退还。 退还金额将全部原路退回到支付方的百度智能云账户。

🔗 退款方式

未使用的直播流量和转码资源包新购之日起7天内（含7天），可通过控制台 -> 财务中心 -> 退订管理菜单直接操作退款。

快速入门

快速入门流程

🔗 概述

本文介绍如何开通和使用百度智能云音视频直播LSS来完成一次直播过程，使用户能快速了解购买和操作的流程。

🔗 使用流程

基本操作流程如下图所示：



- 1. 注册及实名认证。注册百度智能云账号，并完成实名认证。
- 2. 开通音视频直播LSS服务。
- 3. 创建推流和播放域名，所有域名请确保已经[备案](#)。
- 4. 进行相关配置（可选）。
- 5. 推流：用户可以使用手机、电脑等设备推流到LSS。
- 6. 播放：用户通过LSS播放直播流。

开始使用LSS

🔗 概述

新用户首次使用LSS时，需要开通LSS服务，本文将详细介绍LSS的使用准备工作。

🔗 前提条件

🔗 注册及实名认证

使用百度智能云音视频直播LSS前，用户需要拥有一个百度智能云账号并完成实名认证，具体操作如下：

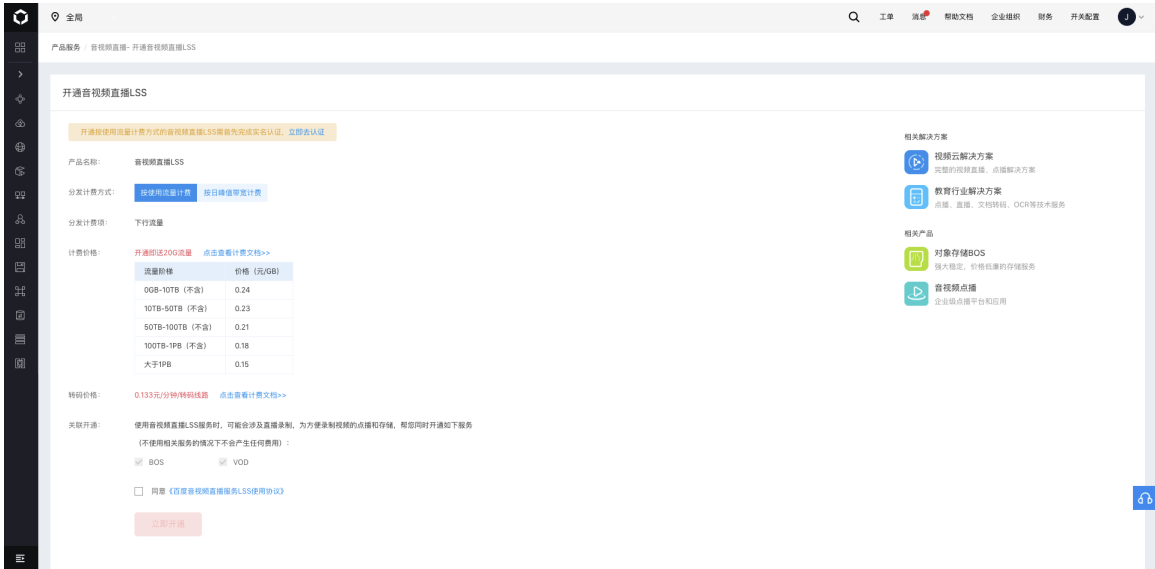
- 1. 注册并登录百度智能云平台，请参考[注册](#)和[登录](#)。
- 2. 完成实名认证，操作细节请参考[实名认证](#)。实名认证分个人认证和企业认证两种，个人认证用户流量上限为100G，企业认证用户无流量上限。如果您是商用，建议您完成企业认证。

🔗 费用说明

- 1. 百度智能云音视频直播服务LSS费用目前包含[分发计费](#)和[转码计费](#)两项。其中分发计费包含两种计费方式：按带宽或者按流量计费；转码计费按照转码时长计费。LSS还提供[流量包](#)方式的流量预付费购买。

🔗 开通服务

- 1. 成功登录控制台后，在[控制台](#)导航栏中选择“产品服务>智能视频>音视频直播LSS”。
- 2. 参照[购买流程](#)开通音视频直播LSS，即可完成服务开通。



创建推流和播放域名

概述

开通LSS服务后，用户就可以在LSS中进行创建推流和播放域名操作，使用百度智能云音视频直播LSS来完成一次直播过程。

创建推流和播放域名

1. 登录[控制台](#)，选择百度智能云音视频直播LSS，选择域名管理，点击添加域名，完成推流域名和播放域名创建。
2. 为直播进行相关配置（可选），包含一些访问控制、转码、录制、水印、缩略图等配置。
 - [实时转码](#)。灵活选择多个转码模板，实现多码率输出，最多可同时配置30路输出。
 - [同步录制](#)。在直播过程中可以同步录制音视频流。
 - [水印](#)。在直播音视频中添加图片水印和时间戳。
 - [缩略图](#)。可以为输出的直播视频生成缩略图。
 - [安全策略](#)。在直播过程中配置安全措施，包括推流/播放认证、防盗链三种安全保障机制。
3. 将系统生成的cname地址配置到相应域名cname上。
4. 获取您的推流地址，使用手机、电脑或其他直播设备将采集到的音视频画面推流到LSS。在移动端，百度智能云为您提供[推流端SDK](#)，支持ios、Android平台；PC端您可以用[OBS](#)软件推流。
 - 推流支持rtmp协议，推流地址拼接规则，鉴权信息参考[安全策略](#)：
 - rtmp://<push.your-domain.com>/{app-name}/{stream-name}?{鉴权信息}
5. 获取您的播放地址，使用手机、电脑等其他设备播放直播视频流，百度智能云为您提供[播放器SDK](#)
 - 播放支持rtmp/flv/hls三种播放协议，播放地址拼接规则，L0-L4代表每路转码流，鉴权信息参考[安全策略](#)：
 - rtmp播放地址
 - L0线路： rtmp://<play.your-domain.com>/{app-name}/{stream-name}?{鉴权信息}
 - L1线路： rtmp://<play.your-domain.com>/<your-app>/<your-stream>-L1?{鉴权信息}
 - L2线路： rtmp://<play.your-domain.com>/<your-app>/<your-stream>-L2?{鉴权信息}
 - L3线路： rtmp://<play.your-domain.com>/<your-app>/<your-stream>-L3?{鉴权信息}
 - L4线路： rtmp://<play.your-domain.com>/<your-app>/<your-stream>-L4?{鉴权信息}

- flv播放地址
 - L0线路：<http://<play.your-domain.com>/<your-app>/<your-stream>.flv?{鉴权信息}>
 - L1线路：<http://<play.your-domain.com>/<your-app>/<your-stream>-L1.flv?{鉴权信息}>
 - L2线路：<http://<play.your-domain.com>/<your-app>/<your-stream>-L2.flv?{鉴权信息}>
 - L3线路：<http://<play.your-domain.com>/<your-app>/<your-stream>-L3.flv?{鉴权信息}>
 - L4线路：<http://<play.your-domain.com>/<your-app>/<your-stream>-L4.flv?{鉴权信息}>
- hls播放地址
 - L0线路：<http://<play.your-domain.com>/<your-app>/<your-stream>.m3u8?{鉴权信息}>
 - L1线路：<http://<play.your-domain.com>/<your-app>/<your-stream>-L1.m3u8?{鉴权信息}>
 - L2线路：<http://<play.your-domain.com>/<your-app>/<your-stream>-L2.m3u8?{鉴权信息}>
 - L3线路：<http://<play.your-domain.com>/<your-app>/<your-stream>-L3.m3u8?{鉴权信息}>
 - L4线路：<http://<play.your-domain.com>/<your-app>/<your-stream>-L4.m3u8?{鉴权信息}>

🔗 相关文档

- 百度智能云音视频直播服务的基本概念，请参考[产品描述](#)。
- 更多有关LSS的操作指导请参考[操作指南](#)。

低延时直播

低延时直播概述

🔗 概述

低延时直播是在百度智能云视频直播的基础上通过链路传输协议优化，解决传统直播延迟过高的弊端，通过集成百度智能云播放器SDK，可以支持千万级并发场景的毫秒级延时直播，并为用户提供低卡顿、秒开流畅的直播观看体验。

🔗 产品架构



🔗 产品特性

🔗 超低延时播放

采用 UDP 协议解决传统直播使用TCP协议延迟过高的问题，同时兼顾秒开、卡顿率等核心指标，为用户提供超低延时、秒开流畅的直播体验。

平滑兼容

兼容传统直播现有的媒体处理和分发能力，包括推流、转码、录制、截图、鉴黄、播放等全部功能，业务可平滑从传统直播迁移至低延时直播。

简单易用，安全可靠

采用标准协议，对接简单，动态兼容B帧、SVC等流信息实现自适应播放，优化信令建联通道，保证信令建联成功率

适用场景

- 电商直播：增强观众和主播互动，促进商品销售。
- 体育直播：观众实时了解赛事现场情况，避免观众间画面不同步
- 教育直播：支持大规模数量学生同时在线超低延时与老师答疑互动。
- 泛娱乐：观众送礼时主播及时反馈，增强互动。

计费说明

低延时直播计费项包含基础服务费用和增值服务费用，基础服务费用为使用低延时直播服务时上行推流和播放产生的流量/带宽费用；增值服务费用为直播转码、录制、截图和鉴黄等增值功能产生的费用。低延时直播计费文档请参见低延时直播计费

功能开通

低延时直播接入相关请参考[低延时直播快速入门](#)

低延时直播和标准直播区别

低延时直播是传统标准直播在超低延时场景下的延伸，比传统直播延迟更低，为观众提供毫秒级延迟的直播观看体验，显著提升直播的互动性，常用于电商直播、在线教育、体育赛事直播、秀场直播等场景。低延时直播和传统标准直播的区别如下：

指标	低延时直播	传统直播
使用限制	播放端集成SDK	无
协议	支持WebRTC 协议播放，延迟在毫秒级别	支持RTMP、FLV、HLS格式的播放协议，这些播放协议都基于 TCP实现，延迟较高
延迟	WebRTC播放协议以UDP协议为基础，延迟在300ms到1000ms之间	RTMP、FLV协议的流延时为3~6秒，延迟因素主要是GOP大小和TCP弱网传输积压 HLS协议的流延时大于等于10秒，延迟因素主要是GOP大小和TS片大小
弱网抗性	观众端丢包30%的情况下仍可以流畅播放，比传统直播有更强的弱网抗性，更稳定流畅的播放体验	弱网抗性有限

低延时直播快速入门

本教程将指引您快速了解低延时直播服务，学习掌握低延时直播的配置方法，用时大约十分钟。在使用低延时直播服务前，建议您提前阅读计费概述，了解计费项目和价格，避免产生争议。

基本操作流程如下图所示：



1. 使用准备：注册百度智能云账号，并完成实名认证，开通LSS直播服务

- 2. 添加推流域名
- 3. 配置WebRTC域名
- 4. SDK接入

🔗 使用准备

- 1. [注册百度智能云账号](#)，并完成[实名认证](#)。
- 2. 参照[购买流程](#)，进入购买服务百度智能云视频直播LSS服务开通页，勾选同意《百度智能云服务协议》，并单击 **申请开通** 即可开通LSS直播服务。

🔗 添加推流域名

- 1. 准备自有域名，并完成[域名备案](#)。
- 2. 登录[音视频直播控制台](#)，进入域名管理，点击添加域名。
- 3. 在添加域名页完成域名添加，并点击下一步。

基本信息

媒体中心:

广州

北京

苏州

LSS支持最近最优边缘节点接流和分发，其中媒体中心负责流媒体实时处理，包括转码、拉流、录制、截图、水印、通知、审核和鉴权等。

类型:

RTMP推流

推流域名:

by-test-push.bigenemy.cn

已备案 最长50个字符

+ 添加域名

播放域名:

☐ RTMP

☒ FLV

test.sz.push1.qq.com

已备案

+ 添加域名

☐ HLS

域名描述:

最长128个字符

下一步

取消

- 4. 在高级配置页，可根据您的业务需求配置高级功能。完成配置后，您可直接点击完成，完成域名的添加，也可以点击下一步为域名添加标签。

产品服务 / 音视频直播LSS-域名管理 / 添加域名

添加域名

1 填写基本信息

2 填写高级配置

3 标签(选填)

4 创建完成

高级配置

可选配置项:

☐ 实时转码

☐ 同步录制

☐ 水印

☐ 缩略图

☐ 通知

上一步

下一步

完成

取消

- 5. （选填）在标签页，为域名添加标签键以及值。更多关于标签的介绍，请参见[标签管理](#)。完成添加后，点击完成。
- 6. 系统将自动跳转至域名列表页面，状态显示“创建中”，耐心等待约一分钟，即可完成域名创建。

🔗 配置WebRTC域名

- 1. 在域名管理页面点击相应域名的 **管理** 按钮进入域名管理页面。
- 2. 在已有flv播放域名时，才可创建低延时播放域名。在输出设置-WebRTC输出中点击添加播放域名即可添加。

27

基本信息

媒体中心：华北 - 北京

类型：RTMP推流

输入设置

推流域名1

推流域名2

+ 添加

推流域名：

推流CNAME：

推流地址：

安全策略：

转推：

已备案

推流CNAME未生效

`{app-name}/{stream-name}?{鉴权信息}`

default

开启

输出设置

FLV输出

RTMP输出

HLS输出

WebRTC输出

您还没创建此类播放域名！

+ 添加播放域名

添加播放域名

WebRTC播放域名：

输入您的播放域名，例如：play.yourdomain.com，请确保域名已备案

+ 添加域名

确定

取消

3. 在添加完成后可完善相应配置如：安全策略、HTTPS证书。

输出设置

FLV输出

RTMP输出

HLS输出

WebRTC输出

播放域名1：

播放CNAME：

安全策略：

HTTPS证书：

hls123.duanhuiyan.top

hls123.duanhuiyan.top.a.e-web.com.cn

default

无

L0：

L1：

L2：

L3：

+ 添加播放域名

webrtc://hls123.duanhuiyan.top/{app-name}/{stream-name}

webrtc://hls123.duanhuiyan.top/{app-name}/{stream-name}-960

webrtc://hls123.duanhuiyan.top/{app-name}/{stream-name}-1280

webrtc://hls123.duanhuiyan.top/{app-name}/{stream-name}-480

常见问题

1. 音频转码

Web 端浏览器不支持 AAC 音频格式，需要将推流的 AAC 音频格式转化为 OPUS 音频格式播放，因此会产生 音频转码 费用。

2. B 帧播放

web端低延时直播不支持B帧解码播放，当原始直播流存在B帧时，后台会自动进行转码去掉B帧，这样不仅会引入转码延迟同时也会产生转码费用。建议尽量不推包含B帧的流，用户可以通过调整推流端软件（如 OBS）的视频编码参数来去除B帧。如



下图：

SDK集成指南

SDK集成指南概述

低延时直播 RTL 提供三种SDK 支持低延时直播能力，满足您在已有直播架构下快速集成低延时直播能力的需求，分别为

- 低延时传输层SDK
- 百度云播放器（低延时直播版）
- 低延时全功能SDK

其特点及集成步骤如下， 您可根据需要选择合适 SDK 进行集成。

低延时播放引擎	特点	接入方式
低延时传输层SDK	仅从原生WebRTC分离其传输能力，实现媒体数据的低延时接收及接入侧回调，包体小，易于集成到现在播放内核中，对上层业务无感。	提供底层so库，您可以采用插件方式集成到播放内核，如用户播放内核基于FFmpeg实现， 则可直接集成rtl_demuxer快速接入。
百度云播放器（低延时直播版）	百度云播放器（低延时直播版）已实现上述低延时传输层SDK的接入，并完成核心能力调优， 除提供低延时播放能力外，百度云播放器（低延时直播版）还提供丰富的标准直播能力和低延时播放降级逻辑；帮您低成本接入全套直播功能。	接入或升级百度云播放器SDK，实现标准直播+低延时直播同步接入。
低延时全功能SDK	基于全功能WebRTC实现的低延时播放引擎，采用WebRTC原生媒体传输、解码、渲染逻辑，具备更好的性能，可实现更低的端到端延时；包体较大，对业务有一定的侵入性。	接入独立的低延时全功能SDK，其接口与标准直播播放器几乎一致。

低延时传输层SDK

Android端

一、概述 低延时传输层SDK（RTL，相对于低延时全功能SDK）是基于webrtc 实现的具备良好抖动缓冲、弱网抗性等能力的低延

时传输组件，为了更好的与通用的基于ffmpeg内核的播放器（如百度云xplayer、ijkplayer、vlc、ffplayer等）进行整合，低延时传输层SDK也实现了ffmpeg的demuxer接口（ff_rtl_demuxer），本文主要介绍ijkplayer集成低延时传输层sdk rtl的相关要点。

二、SDK产物说明

```

├── include // rtl 头文件
│   ├── rtl_api.h
│   └── rtl_def.h
├── libs // rtl 各架构低延时传输库，可集成相应的架构到自己工程
│   ├── arm64-v8a
│   │   └── librtl.so
│   ├── armeabi-v7a
│   │   └── librtl.so
└── source // ff_rtl_demuxer 参考实现，可直接集成到播放器 avforamt（如ijkplayer, ijkplayer/ijkavformat）下
    └── rtl_dec.c

```

三、集成步骤 ijkplayer 集成rtl 传输层sdk步骤：

• 导入头文件及动态库

将上述头文件（rtl_api.h、rtl_def.h）及库文件(\$arch/librtl.so)分别复制到工程架构路径：

ijkplayer/android/contrib/build/ffmpeg-\$arch/output/include 及ijkplayer/android/contrib/build/ffmpeg-\$arch/output 目录

• 修改编译脚本

RTL 回调音频格式为PCM, 修改，新增PCM 音频解码

```

export COMMON_FF_CFG_FLAGS="$COMMON_FF_CFG_FLAGS --enable-decoder=pcm_s16be_planar"
export COMMON_FF_CFG_FLAGS="$COMMON_FF_CFG_FLAGS --enable-decoder=pcm_s16le"
export COMMON_FF_CFG_FLAGS="$COMMON_FF_CFG_FLAGS --enable-decoder=pcm_s16le_planar"

```

修改ijkmedia/ijkSDL/Android.mk 编译脚本，引入librtl.so：

```

include $(CLEAR_VARS)
LOCAL_MODULE := ijkffmpeg
LOCAL_SRC_FILES := $(MY_APP_FFMPEG_OUTPUT_PATH)/libijkffmpeg.so
-include $(PREBUILT_SHARED_LIBRARY)
\ No newline at end of line
+include $(PREBUILT_SHARED_LIBRARY)
+
+include $(CLEAR_VARS)
+LOCAL_MODULE := rtl
+LOCAL_SRC_FILES := $(MY_APP_FFMPEG_OUTPUT_PATH)/librtl.so
+include $(PREBUILT_SHARED_LIBRARY)

```

• 添加rtl_dec.c(ff_rtl_demuxer)及动态库

将rtl_dec.c文件拷贝到工程源码目录：src/main/jni/ijkmedia/ijkplayer/ijkavformat，修改对应Android.mk 编译脚本如下：

```

@@ -75,7 +75,9 @@ LOCAL_SRC_FILES += ijkavformat/ijktree.c
LOCAL_SRC_FILES += ijkavformat/ijkfifo.c
LOCAL_SRC_FILES += ijkavformat/ijkstl.c

+LOCAL_SRC_FILES += ijkavformat/rtl_dec.c

-LOCAL_SHARED_LIBRARIES := ijkffmpeg ijkSDL
+LOCAL_SHARED_LIBRARIES := ijkffmpeg ijkSDL rtl
LOCAL_STATIC_LIBRARIES := android-ndk-profiler ijkSoundtouch

```

- **rtl so 加载**

```

static {
    System.loadLibrary("rtl");
}

```

- **注册rtl demuxer 协议**

修改ff_ffplayer.c, 声明 低延时拉流协议 ff_rtl_demuxer:

```

49 + #include "rtl_api.h"
49 50
50 51 #if CONFIG_AVDEVICE
51 52 #include "libavdevice/avdevice.h"

    块 2 : 行 148-155
147 148 }
148 149 #endif
149 150
151 + extern AVInputFormat ff_rtl_demuxer;
152 +

```

在read_thread()线程方法里针对 webrtc:// 前缀 url 指定 AVInputFormat 为 ff_rtl_demuxer:

```

- if (ffp->iformat_name)
+ if (strncmp(is->filename, "webrtc://", 9) == 0) {
+     av_register_input_format(&ff_rtl_demuxer);
+     is->iformat = &ff_rtl_demuxer;
+ } else {
+     is->iformat = av_find_input_format(ffp->iformat_name);
+ }
+

// customized flag settings
char *lastDot = strrchr(is->filename, '.');

```

- **配置修改, 适配低延时播放场景**

- 不限制拉流缓存大小, 开启无限读, 防止弱网下延迟累积;
- ijkMediaPlayer.setOption(ijkMediaPlayer.OPT_CATEGORY_PLAYER, "infbuf", 1); 指定rtl demuxer 为 realtime :

```

static int is_realtime(AVFormatContext *s) {
    if (!strcmp(s->iformat->name, "rtsp")
        || !strcmp(s->iformat->name, "rtsp")
        || !strcmp(s->iformat->name, "sdp")
        || !strcmp(s->iformat->name, "rtmp", 4)
        || !strcmp(s->iformat->name, "ffrtmp")
        || !strcmp(s->iformat->name, "live_flv")
        || !strcmp(s->iformat->name, "rtl"))
        return 1;

    if(s->pb && (!strcmp(s->url, "rtsp:", 4)
        || !strcmp(s->url, "udp:", 4)))
        return 1;
    return 0;
}

```

- 关闭buffering, 保证弱网下也能进行视频低帧率 音频流畅播放 ;
ijkMediaPlayer.setOption(IjkMediaPlayer.OPT_CATEGORY_PLAYER, "packet-buffering", 0);
- 最大缓存配置低于500ms, 建议使用200ms; setOption(IjkMediaPlayer.OPT_CATEGORY_PLAYER, "max-buffer-duration", 200);

四、低延时播放及RTL事件监听

- RTL 消息回调 经过上述步骤, ijkplayer 已具体低延时播放能力, 接下来将RTL 事件接入到上层。

修改 ijkmedia/ijkplayer/ff_ffmsg.h

```

// add rtl message
##### define FFP_MSG_RTL

```

修改ff_ffplay.c 文件 接收来自RTL 库的消息

```
static int av_format_message_cb(struct AVFormatContext *ic, int type,
                               void *data, size_t data_size) {
    FFPlayer *ffp = ic->opaque;
    // rtl modify start
    if (!ffp) {
        return 0;
    }

    // merge rtl info and error event post
    if ((type > RTL_EVENT_BASE && type < RTL_EVENT_BASE + 100)
        || (type > RTL_ERROR_BASE && type < RTL_ERROR_BASE + 100)) {
        ffp_notify_msg4(ffp, FFP_MSG_RTL, type, 0, data, data_size);
    } else if (ffp->inject_opaque) {
        return inject_callback(ffp->inject_opaque, type, data, data_size);
    }
    // rtl modify end
    return 0;
}
```

修改 ijkmedia/ijkplayer/android/ijkplayer_jni.c 新半RTL 消息post 到JAVA 层

```
static void message_loop_n(JNIEnv *env, BDCloudMediaPlayer *mp) {
    // ...
    while (1) {
        switch (msg.what) {
            // ...
            case FFP_MSG_RTL:
                MPTRACE ("FFP_MSG_RTL:\n");
                post_event(env, weak_this, MEDIA_RTL_MSG, msg.arg1, 0);
                break;
        }
    }
}
```

修改ijkMediaPlayer.java 接收来自native 的rtl 消息

```
private static class EventHandler extends Handler {
    // ...
    @Override
    public void handleMessage(Message msg) {
        switch (msg.what) {
            case MEDIA_RTL_MSG:
                player.notifyOnRtlMessage(msg.arg1);
                break;
        }
    }
}
```

修改AbstractMediaPlayer.java 新增RTL 消息监听

```
private OnRtlMessageListener mOnRtlMessageListener;

@Override
public void setOnRtlMessageListener(OnRtlMessageListener listener) {
    mOnRtlMessageListener = listener;
}

protected final void notifyOnRtlMessage(int what) {
    if (mOnRtlMessageListener != null) {
        mOnRtlMessageListener.onRtlMessage(this, what);
    }
}
```

- 低延时播放跑通验证

至此，已完成rtl在ijkplayer的集成，我们可以像播放普通视频一样调用ijkplayer 进行播放，具体调用过程同样是 创建ijkplayer->setOnXxxListener()->setSurface()-> setDataSource("webrtc://xxx")->prepare()/start()/stop(); 值得注意的是：

1. 低延时播放流地址没"webrtc://"前缀，该地址可以在域名申请时获取；

2. option 参数配置

```
// 关闭buffering缓冲逻辑
this.setOption(OPT_CATEGORY_PLAYER, "packet-buffering", 0);
// 设置信令模式
setOption(OPT_CATEGORY_FORMAT, "rtl_signaling_mode", signalingMode);
// 设置媒体服务地址，非必需，不设置则调用系统接口进行域名解析
setOption(OPT_CATEGORY_FORMAT, "rtl_mediaserver_host", mediaServerHost);
```

4. 监听新增的setOnRtlMessageListener接口获得并处理RTL 事件；

- RTL 事件处理

ijkplayer 通过setOnRtlMessageListener 监听RTL 事件，用户可针对特定的事件进行重连、降级播放处理。

```
// 设置rtl事件监听
mVV.setOnRtlMessageListener(this);

// rtl 事件处理
@Override
public void onRtlMessage(IMediaPlayer mp, int what) {
    Log.i(TAG, "on RTL MSG: " + what);
    switch (what) {
        case IMediaPlayer.RTL_EVENT_FIRST_VFRAME:
            bRtlRetried = false;
            break;
        case IMediaPlayer.RTL_ERROR_ICE_CHANNEL:
        case IMediaPlayer.RTL_ERROR_STREAMING_INTERRUPT:
            if (!bRtlRetried) {
                // 重试一次，依然失败则降级
                updateTextInfo("RTL play need retry since error : " + what);
                bRtlRetried = true;
                mVV.rtlReload(info.getUrl());
            } else {
                rtlFallback(what);
            }
            break;
        default:
            if (what > IMediaPlayer.RTL_ERROR_BASE) {
                // 直接降级
                rtlFallback(what);
            }
            break;
    }
}

// 降级播放
private void rtlFallback(int fallbackReason) {
    updateTextInfo("RTL play need fallback since error : " + fallbackReason);
    if (!TextUtils.isEmpty(info.getFallbackUrlForRtl())) {
        mVV.rtlReload(info.getFallbackUrlForRtl());
    } else {
        showError("低延时直播错误 : " + fallbackReason);
        mVV.stopPlayback();
    }
}
```

RTL 回调的事件定义如下：

```

public interface IMediaPlayer {
    ... ..
    // RTL 事件消息
    int RTL_EVENT_ICE_BASE          = 1200;
    // ICE 连接成功
    int RTL_EVENT_ICE_CONNECTED     = 1201;
    // 连接关闭
    int RTL_EVENT_CONNECTION_CLOSED = 1202;
    // ICE 连接断开
    int RTL_EVENT_ICE_DISCONNECTED  = 1206;
    // 没有检测到媒体流
    int RTL_EVENT_NO_STREAMING_DETECTED = 1207;
    // 获得 remote sdp
    int RTL_EVENT_REMOTE_SDP_ACQUIRED = 1210;
    // RTL首屏事件
    int RTL_EVENT_FIRST_VFRAME       = 1211;

    // RTL 错误消息
    int RTL_ERROR_BASE              = 12000;
    // ICE 连接错误, 建议重试一次, 重试失败降级为普通直播
    int RTL_ERROR_ICE_CHANNEL       = 12001;
    // Peer连接创建失败, 建议降级为普通直播
    int RTL_ERROR_CONNECTION       = 12003;
    // 远端媒体描述请求失败, 建议降级为普通直播
    int RTL_ERROR_REMOTE_SDP_REQUEST = 12006;
    int RTL_ERROR_REMOTE_SDP_SET    = 12007;
    // 播放状态错误, 建议降级为普通直播
    int RTL_ERROR_INVALID_STATE     = 12008;
    // 媒体流中断 一段时间未收媒体流(默认6S) SDK内部检测到断流错误后会立即停止播放. 重试失败降级为普通直播
    int RTL_ERROR_STREAMING_INTERRUPT = 12009;
    // URL格式错误, 降级为普通直播
    int RTL_ERROR_INVALID_URL       = 12010;
    // 约2s后返回, 用户网络MTU Size 检测失败 (< 1300 Byte) 无法支持RTC媒体传输, 建议降级为普通播放
    int RTL_ERROR_MTU_SIZE_CHECK_FAILED = 12014;
    // 非法状态异常, 降级为普通直播
    int RTL_ERROR_ILLEGAL_STATE     = 12015;
}

```

🔗 iOS端

一、概述

低延时传输层sdk (RTL, 相对于低延时全功能SDK) 是基于webrtc 实现的具备良好抖动缓冲、弱网抗性等能力的低延时传输组件, 为了更好的与通用的基于ffmpeg内核的播放器 (如百度云xplayer、ijkplayer、vlc、ffplayer 等) 进行整合, 低延时传输层SDK 也实现了ffmpeg的demuxer 接口 (ff_rtl_demuxer), 本文主要介绍ijkplayer集成低延时传输层sdk rtl 的相关要点。

二、SDK产物说明

提供基于ffmpeg的ff_rtl_demuxer 文件：rtl_dec.c 及其依赖库 RtlSDK.framework 集成到客户播放器程序
RtlSDK.framework结构如下所示：

```

├── Headers
│   ├── RtlSDK.h
│   ├── rtl_api.h
│   └── rtl_def.h
├── Info.plist
├── Modules
│   └── module.modulemap
├── RtlSDK
└── _CodeSignature
    ├── CodeDirectory
    ├── CodeRequirements
    ├── CodeRequirements-1
    ├── CodeResources
    └── CodeSignature

```

三、集成步骤

ijkplayer 集成rtl 传输层sdk步骤：

- 导入头文件及动态库

将上述头文件（rtl_dec.c）及库文件(RtlSDK.framework)分别复制到工程架构路径：添加依赖

▼ Link Binary With Libraries (24 items)

Name	Status
 RtlSDK.framework	Required ↕

- 注册rtl demuxer 协议

修改ff_ffplayer.c, 声明 低延时拉流协议 ff_rtl_demuxer:

```

49 + #include "rtl_api.h"
49 50
50 51 #if CONFIG_AVDEVICE
51 52 #include "libavdevice/avdevice.h"
    块 2：行 148-155
147 148 }
148 149 #endif
149 150
151 + extern AVInputFormat ff_rtl_demuxer;
152

```

在read_thread()线程方法里针对 webrtc:// 前缀 url 指定 AVInputFormat 为 ff_rtl_demuxer:

```

- if (ffp->iformat_name)
+ if (strncmp(is->filename, "webrtc://", 9) == 0) {
+     av_register_input_format(&ff_rtl_demuxer);
+     is->iformat = &ff_rtl_demuxer;
+ } else {
+     is->iformat = av_find_input_format(ffp->iformat_name);
+ }
+

// customized flag settings
char *lastDot = strrchr(is->filename, '.');

```

- 配置修改，适配低延时播放场景

1、不限制拉流缓存大小, 开启无限读, 防止弱网下延迟累积；

```
ijkMediaPlayer.setOption(ijkMediaPlayer.OPT_CATEGORY_PLAYER, "infbuf", 1);
```

指定rtl dumuxer 为 realtime :

```
static int is_realtime(AVFormatContext *s) {
    if (!strcmp(s->iformat->name, "rtsp")
        || !strcmp(s->iformat->name, "sdp")
        || !strcmp(s->iformat->name, "rtmp")
        || !strcmp(s->iformat->name, "ffrtmp")
        || !strcmp(s->iformat->name, "live_flv")
        || !strcmp(s->iformat->name, "rtl"))
        return 1;

    if(s->pb && (!strcmp(s->url, "rtsp:")
        || !strcmp(s->url, "udp:")))
        return 1;
    return 0;
}
```

2、关闭buffering, 保证弱网下也能进行视频低帧率 音频流畅播放 ;

```
ijkMediaPlayer.setOption(IjkMediaPlayer.OPT_CATEGORY_PLAYER, "packet-buffering", 0);
```

3、最大缓存配置低于500ms, 建议使用200ms;

```
setOption(IjkMediaPlayer.OPT_CATEGORY_PLAYER, "max-buffer-duration", 200);
```

四、低延时播放及RTL事件监听

- RTL 消息回调

经过上述步骤, ijkplayer 已具体低延时播放能力, 接下来将RTL 事件接入到上层。

1、修改 ijkmedia/ijkplayer/ff_ffmsg.h

```
// add rtl message
##### define FFP_MSG_RTL          30000
```

2、修改ff_ffplay.c 文件 接收来自RTL 库的消息

```
static int av_format_message_cb(struct AVFormatContext *ic, int type,
                               void *data, size_t data_size) {
    FFPlayer *ffp = ic->opaque;
    // rtl modify start
    if (!ffp) {
        return 0;
    }

    // merge rtl info and error event post
    if ((type > RTL_EVENT_BASE && type < RTL_EVENT_BASE + 100)
        || (type > RTL_ERROR_BASE && type < RTL_ERROR_BASE + 100)) {
        ffp_notify_msg4(ffp, FFP_MSG_RTL, type, 0, data, data_size);
    } else if (ffp->inject_opaque) {
        return inject_callback(ffp->inject_opaque, type, data, data_size);
    }
    // rtl modify end
    return 0;
}
```

3、修改 ijkmedia/ijkplayer/android/ijkplayer_jni.c 新半RTL 消息post 到JAVA 层

```
static void message_loop_n(JNIEnv *env, BDCloudMediaPlayer *mp) {
    // ...
    while (1) {
        switch (msg.what) {
            // ...
            case FFP_MSG_RTL:
                MPTRACE ("FFP_MSG_RTL:\n");
                post_event(env, weak_thiz, MEDIA_RTL_MSG, msg.arg1, 0);
                break;
        }
    }
}
```

4、修改 IJKFFMoviePlayerControler 接收来自native的rtl 消息

```
- (void)postEvent: (IJKFFMoviePlayerMessage *)msg

    // ...
    case FFP_MSG_RTL: {
        av_log(NULL, AV_LOG_DEBUG, "FFP_MSG_RTL: %d\n", avmsg->arg1);
        [[NSNotificationCenter defaultCenter]
         postNotificationName:IJKMPMoviePlayerRtlMsgNotification
         object:self.shell
         userInfo:@{IJKMPMoviePlayerRtlMsgNotificationKey: @(avmsg->arg1)}];
        break;
    }
}
```

• 低延时播放跑通验证

至此，已完成rtl 在ijkplayer的集成，我们可以像播放普通视频一样调用ijkplayer 进行播放，具体调用过程同样是 创建
setDataSource("webrtc://xxx")

```
prepare()/start()/stop();
```

值得注意的是：

1. 低延时播放流地址没"webrtc://"前缀，该地址可以在[域名申请](#)时获取；
2. 监听新增的setOnRtlMessageListener接口获得并处理RTL 事件；

- RTL 事件处理

ijkplayer 通过BDCloudMediaPlayerRtlMsgNotification 监听RTL 事件，用户可针对特定的事件进行重连、降级播放处理。

```
// rtl 事件处理
- (void)onRtlMsg:(NSNotification *)notification {
    if (notification.object != _player) {
        return;
    }
    NSNumber *rtlMsg = notification.userInfo[BDCloudMediaPlayerRtlMsgNotificationKey];
    NSLog(@"onRtlMsg %ld", (long)rtlMsg.integerValue);
    [self.actions appendVideoLog:[NSString stringWithFormat:@"onRtlMsg %ld\n", rtlMsg.integerValue]];

    switch (rtlMsg.integerValue) {
        case RTL_EVENT_FIRST_VFRAME:
            self.bRtlRetried = NO;
            break;
        case RTL_ERROR_ICE_CHANNEL:
        case RTL_ERROR_STREAMING_INTERRUPT:
            if (!self.bRtlRetried) {
                // 重试一次，依然失败则降级
                [self.actions appendVideoLog:[NSString stringWithFormat:@"RTL play need retry since error %ld",
                    rtlMsg.integerValue]];
                self.bRtlRetried = YES;

                [self stopPlayback];
                self.bSwitching = true;

            } else {
                [self rtlFallback:rtlMsg.integerValue];
            }
            break;
        default:
            if (rtlMsg.integerValue > RTL_ERROR_BASE) {
                // 直接降级
                [self rtlFallback:rtlMsg.integerValue];
            }
            break;
    }
}

// 降级播放
- (void)rtlFallback:(int)fallbackReason {
    [self.actions appendVideoLog:[NSString stringWithFormat:@"RTL play need fallback since error %d\n",
        fallbackReason]];
    if (self.model.fallbackUrlForRtl == nil || self.model.fallbackUrlForRtl.length == 0) {
        [self stopPlayback];
        self.tipLabel.text = [NSString stringWithFormat:@"低延时直播错误 : %d", fallbackReason];
        self.tipLabel.hidden = NO;
    } else {
        [self stopPlayback];
        self.model.url = self.model.fallbackUrlForRtl;
        self.bSwitching = true;
    }
}
```

RTL 回调的事件定义如下：

```
typedef NS_ENUM(NSUInteger, BDCloudRtlMsg) {
    RTL_EVENT_ICE_BASE            = 1200,
    // ICE 连接成功
    RTL_EVENT_ICE_CONNECTED       = 1201,
    // 连接关闭
    RTL_EVENT_CONNECTION_CLOSED   = 1202,
    // ICE 连接断开
    RTL_EVENT_ICE_DISCONNECTED    = 1206,
    // 没有检测到媒体流
    RTL_EVENT_NO_STREAMING_DETECTED = 1207,
    // 获得 remote sdp
    RTL_EVENT_REMOTE_SDP_ACQUIRED  = 1210,
    // RTL首屏事件
    RTL_EVENT_FIRST_VFRAME        = 1211,

    // RTL error event
    RTL_ERROR_BASE                = 12000,
    // ICE 连接错误, 建议重试一次, 重试失败降级为普通直播
    RTL_ERROR_ICE_CHANNEL         = 12001,
    // Peer连接创建失败, 建议降级为普通直播
    RTL_ERROR_CONNECTION         = 12003,
    // 远端媒体描述请求失败, 建议降级为普通直播
    RTL_ERROR_REMOTE_SDP_REQUEST  = 12006,
    RTL_ERROR_REMOTE_SDP_SET      = 12007,
    // 播放状态错误, 建议降级为普通直播
    RTL_ERROR_INVALID_STATE       = 12008,
    // 媒体流中断 一段时间未收媒体流(默认6S) SDK内部检测到断流错误后会立即停止播放. 重试失败降级为普通直播
    RTL_ERROR_STREAMING_INTERRUPT = 12009,
    // URL格式错误, 降级为普通直播
    RTL_ERROR_INVALID_URL        = 12010,
    // 约2s后返回, 用户网络MTU Size 检测失败 (< 1300 Byte) 无法支持RTC媒体传输, 建议降级为普通播放
    RTL_ERROR_MTU_SIZE_CHECK_FAILED = 12014,
    // 非法状态异常, 降级为普通直播
    RTL_ERROR_ILLEGAL_STATE       = 12015
};
```

🔗 SDK接口说明

一、概述

API	描述
GetRtlStreamModule	获取rtl API操作唯一句柄。
create	创建rtl 播放实例
open	通过url打开媒体流
read	读取一帧媒体数据
free_frame	释放一帧媒体数据
close	关闭媒体流

二、API详情

* GetRtlStreamModule: 获取rtl API操作唯一句柄。

```
const struct RtlStreamModule* GetRtlStreamModule(int version);
```

参数

参数	类型	描述
version	int	版本号

返回

返回操作其它API的 RtlStreamModule 函数指针。

*** create**：创建rtl 播放实例。

```
void* (*create)(RtlProfile profile);
```

参数

参数	类型	描述
profile	RtlProfile	播放配置

返回

返回播放操作句柄表示成功，返回nullptr表示失败。

1、RtlProfile 数据结构

```
typedef struct RtlProfile {
    RtlLogLevel log_level; // debug log level
    void* ctx;             // AVFormatContext usually
    RtlSignalingMode signaling_mode;
    RtlCallbacks callbacks; // rtl callbacks
    char *media_server_host;
} RtlProfile;
```

成员属性	描述
log_level	debug log 级别控制 包括：0：RTL_LOG_DEBUG、1：RTL_LOG_INFO、2：RTL_LOG_WARN、3：RTL_LOG_ERROR、4：RTL_LOG_NONE
signaling_mode	信令模式，1: http 信令、2:百度私有udp信令，暂不开放 3: udp over minisdp信令。
ctx	传入rtl 的上下文， ffmpeg 里一般传入AVFormatContext。
callbacks	rtl 回调
media_server_host	预设媒体服务地址，可优化启播，非必需。

2、RtlCallbacks 结构

```
typedef struct RtlCallbacks {
    OnLogSink log_sink; // log callback
    MsgCallback msg_callback; // message callback
    MediaInfoCallback media_info_callback; // media info callback
} RtlCallbacks;
```

成员属性	描述
log_sink	日志回调， rti 通过该接口向接入层回调debug日志信息；
msg_callback	消息回调， rti 通过该接口向接入层回调事件及异常消息；
media_info_callback	媒体信息回调， rti 通过该接口向接入层回调媒体信息；

3、MsgCallback 消息回调

```
typedef int(*MsgCallback)(void* ctx, int type, const void* data, int data_size);
```

参数	类型	描述
ctx	void*	上下文
type	int	消息类型
	void*	消息扩展信息
data_size	int	消息扩展信息长度

消息类型包含事件消息及异常消息。

事件消息定义：

事件消息	事件码	描述
RTL_EVENT_ICE_CONNECTED	1201	ICE 媒体通道连接成功。
RTL_EVENT_CONNECTION_CLOSED	1202	播放连接关闭。
RTL_EVENT_ICE_DISCONNECTED	1206	ICE 媒体通道连接断开。
RTL_EVENT_NO_STREAMING_DETECTED	1207	没有检测到媒体流。
RTL_EVENT_REMOTE_SDP_ACQUIRED	1210	获得远端媒体描述。
RTL_EVENT_FIRST_VFRAME	1211	视频首帧事件。

异常消息定义：

错误消息	错误码	描述
RTL_ERROR_ICE_CHANNEL	12001	ICE 连接错误, 建议重试一次, 重试仍失败降级为普通直播。
RTL_ERROR_CONNECTION	12003	播放连接创建失败。
RTL_ERROR_REMOTE_SDP_REQUEST	12006	远端媒体描述请求失败, 建议降级为普通播放。
RTL_ERROR_INVALID_STATE	12008	播放状态错误。
RTL_ERROR_STREAMING_INTERRUPT	12009	媒体流中断错误（默认10s未收到媒体流则反馈该错误）。
RTL_ERROR_INVALID_URL	12010	URL 格式错误。
RTL_ERROR_MTU_SIZE_CHECK_FAILED	12014	用户当前网络不支持低延时播放错误（MTU检测异常），建议降级为普通播放。

* open：通过url打开媒体流。

```
int (*open)(void* handle, const char* stream_url);
```

参数

参数	类型	描述
handle	void*	create 接口返回的播放句柄。
stream_url	char*	媒体流播放地址，前缀为“webrtc://”。

返回

成功返回0，失败返回负值。

*** read**：读取一帧音视频媒体数据。

```
int (*read)(void* handle, struct RtIframe** frame);
```

参数

参数	类型	描述
handle	void*	create 接口返回的播放句柄。
frame	struct RtIframe**	音视频媒体数据结构。

返回

返回 1 表示读取一帧到存入frame。 返回-1表示EOF。 返回其它负值表示发生错误；

*** free_frame**：释放一帧媒体数据。

```
void (*free_frame)(void* handle, struct RtIframe* frame);
```

参数

参数	类型	描述
handle	void*	create 接口返回的播放句柄。
frame	struct RtIframe**	待释放的一帧媒体数据。

返回

无返回

*** close**：关闭媒体流。

```
void (*close)(void* handle);
```

参数

参数	类型	描述
handle	void*	create 接口返回的播放句柄。

返回

无返回。

低延时全功能SDK

Android 端集成指引

概述 百度云超低延时直播（BRTCPlayer）全功能版本是在百度智能云视频直播的基础上通过链路传输协议优化，解决传统直播延迟过高的弊端，通过集成百度智能云播放器SDK，可以支持千万级并发场景的毫秒级延时直播，并为用户提供低卡顿、秒开

流畅的直播观看体验。

功能介绍 BRTCPlayer 主要功能如下：

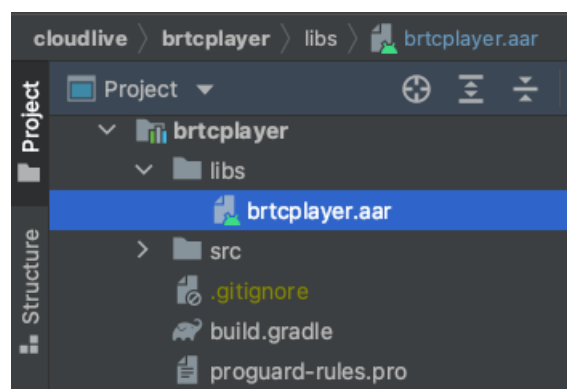
- 多信令模式：支持HTTP及基于Minisdp的udp信令模式；支持BD UDP私有信令模式：0 RTT 极速启播，有效提升秒开率；
- 支持AVC、HEVC视频硬解码；
- 支持视频B帧：引入B帧，可在清晰率不变的情况下降低码率；
- 支持AAC音频解码，支持MP4A-ADTS、LATM格式 48000 及 44100 采样率；
- 支持视频平滑渲染：网络抖动时，平滑渲染视频帧，降低卡顿；
- 支持最小延迟动态配置：可实现延迟换流畅，在允许范围内增大延迟可有效降低卡顿；
- 支持网络健康度实时检测；
- 支持 SEI 消息；

Demo体验 前往[DEMO下载](#)页面扫码安装DEMO

SDK 下载 前往[SDK下载](#)页面下载支持超低延时全功能SDK版本， 内含Demo源码；

快速集成 百度云超低延时直播由独立SDK提供支持，用户可通过在播放侧单独集成BRTCPlayer SDK实现超低延时直播，而推流侧则可复用当前使用的推流器，仅需在百度直播LSS平台针对当前域名[开通低延时播放能力](#)。BRTCPlayer SDK具体的集成步骤如下：

- 导入BRTCPlayer aar到工程，集成后的示例如下：



- 修改 build.gradle

```
dependencies {  
    implementation fileTree(dir: 'libs', include: ['brtcpayer.aar'])  
    implementation 'com.squareup.okhttp3:okhttp:3.10.0'  
    ...  
}
```

- 添加混淆规则

```
-keepclasseswithmembernames,allowshrinking,allowoptimization class * {  
    native <methods>;  
}  
  
-keep interface * {  
    public <fields>;  
    public <methods>;  
}  
  
-keep class com.baidu.rtc.** {*;}  
-keep class com.webrtc.** {*;}
```

快速开始

1. 创建播放渲染视窗

```
// 在布局文件中添加渲染控件  
<com.baidu.rtc.RTCVideoView  
    android:id="@+id/brtc_video_view"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" />  
  
// 寻回渲染控件对象  
mVideoView = (RTCVideoView) findViewById(R.id.brtc_video_view);
```

2. 创建播放器对象及播放器初始化

```
// 创建播放器对象  
mBRTCPlayer = new BRTCPlayerImpl(this);  
// 创建播放参数集，初始化播放器  
mPlayerParameters = new BRTCPlayerParameters();  
// 设置播放器信令服务地址  
mPlayerParameters.setPullUrl(mSignalUrl);  
// 设置信令模式  
mPlayerParameters.setSignalingMode(mSignalingMode);  
// 初始化播放器 传入播放参数及事件监听器  
mBRTCPlayer.initPlayer(mPlayerParameters, this);
```

3. 播放设置

```
// 设置播放渲染视图  
mBRTCPlayer.setSurfaceView(mVideoView);  
// 设置媒体流地址  
mBRTCPlayer.setStreamUri(mStreamUrl);
```

4. 播放控制

```
// 准备播放，获取播放依赖资源  
mBRTCPlayer.prepareAsync();  
// 开始播放  
mBRTCPlayer.startPlay();  
// 暂停播放  
mBRTCPlayer.pausePlay();  
// 恢复播放  
mBRTCPlayer.resumePlay();  
// 停止播放  
mBRTCPlayer.stopPlay();
```

5. 释放播放器

```
// 释放播放器
mBRTCPlayer.releasePlayer();
```

在智能视频SDK Demo中对上述流程有详细的展示，可以参考；

接口说明 BRTCPlayerParameters类 | 接口名 | 描述 | 备注 | |-----|-----|-----|

| void setEnableDebug(boolean enable) | 设置是否开启debug日志 | 无 | | void setPullUrl(String pullUrl) | 设置信令服务地址，格式为http[s]://domain/brtc/v3/pullstream | 仅HTTP信令 | | void void setSignalingMode(SignalingMode signalingMode) | 设置信令模式 | 默认HTTP信令 | | void setAutoPlay(boolean autoPlay) | 设置是否自动启播，默认false | 设置后调用prepareAsync后自动播放 | | void void setEnableSilentStart(boolean enable) | 设置静音启播 | 无 | | void setBaseDelayMs(int baseDelayMs) | 设置最小延时，可通过该参数增大延时降低卡顿提升播放流畅度 | 非必需 | | void setCpuType(String cpuType) | 指定下载的SO库CPU架构 | 非必需，开启so后下载配置 | | void setVideoDecodeFormat(int videoDecodeFormat) | 设置视频解码格式，由业务侧确定是否为H265格式，默认为H264格式 | 仅UDP_RTL信令 | | void setAudioSampleRate(int sampleRate) | 设置音频播放采样率[48000, 44100]，仅UDP_RTL信令模式 | 仅UDP_RTL信令 | | void setEnableVideoBFrame(boolean bFrame) | 是否开启B帧支持，由业务侧决定是否开启 | 默认开启 | | void setMediaServerIp(String mediaServerIp) | 预设媒体服务IP地址以加速启播 | 非必需，可配合HttpDns使用 | | void setEnableFirstFrameReconnect(boolean enable) | 当前播放超过2s没有出首帧则使用Http信令模式重试一次 | 默认开启 |

SignalingMode 枚举类 | 枚举类型 | 信令模式 | 备注 | |-----|-----|-----| | HTTP_RTL | HTTP信令 | 自协商媒体参数 | | UDP_RTL | 百度私有UDP信令 | 使用该模式时需通过BRTCPlayerParameters类预设音视频媒体类型参数 | | UDP_MINISDP | MINISDP UDP 信令 | 自协商媒体参数 |

BRTCPlayerEvents类 | 错误码 | 含义 | |-----|-----|-----|

BRTC_PLAYER_ERROR_INVALID_URL = 10000 | URL 格式错误 | | BRTC_PLAYER_ERROR_ICE_CHANNEL = 10001 | ICE 连接错误 | | BRTC_PLAYER_ERROR_RESERVED = 10002 | 替换预留错误码 | | BRTC_PLAYER_ERROR_CONNECTION = 10003 | 连接创建失败 | | BRTC_PLAYER_ERROR_LOCAL_SDP_REQUEST = 10004 | 媒体描述请求失败 | | BRTC_PLAYER_ERROR_LOCAL_SDP_SET = 10005 | 媒体描述设置失败 | | BRTC_PLAYER_ERROR_REMOTE_SDP_REQUEST = 10006 | 远端媒体描述请求失败，一般由于启播放时主播侧已停播 | | BRTC_PLAYER_ERROR_REMOTE_SDP_SET = 10007 | 远端媒体描述设置失败 | | BRTC_PLAYER_ERROR_INVALID_STATE = 10008 | 播放状态错误 | | BRTC_PLAYER_ERROR_STREAMING_INTERRUPT = 10009 | 媒体流中断，一段时间未收媒体流，SDK内部检测到断流错误后会立即停止播放，一般在播放中主播停播但业务侧未退出 | | BRTC_PLAYER_ERROR_LOAD_LIBRARIES = 10010 | SO库文件后下载失败 | | BRTC_PLAYER_ERROR_INVALID_LICENSE = 10011 | license错误 | | BRTC_PLAYER_ERROR_LICENSE_FEATURE_INVALID = 10012 | 当前license不包含低延时播放feature（废弃） | | BRTC_PLAYER_ERROR_DECODER_OPEN_FAILED = 10013 | 解码器打开失败 | | int BRTC_PLAYER_ERROR_MTU_SIZE_CHECK_FAILED = 10014 | 用户网络MTU Size 检测失败，无法支持RTC媒体传输，约2s后返回，建议业务侧降级为普通播放 | | int BRTC_PLAYER_ERROR_SET_EXTERNAL_SURFACE = 10015 | 设置外部渲染错误，重复设置相同外部Surface | | int BRTC_PLAYER_ERROR_ILLEGAL_STATE = 10016 | 非法状态错误 |

事件码	含义
BRTC_PLAYER_EVENT_REMOTE_RENDER = 1000	开始远端渲染
BRTC_PLAYER_EVENT_ICE_CONNECTED = 1001	ICE连接成功
BRTC_PLAYER_EVENT_PEER_CONNECTION_CLOSED = 1002	对端连接关闭
BRTC_PLAYER_EVENT_STATS_UPDATED = 1003	媒体流信息更新
BRTC_PLAYER_EVENT_BUFFERING_START = 1004	Buffering start事件
BRTC_PLAYER_EVENT_BUFFERING_END = 1005	Buffering end事件
BRTC_PLAYER_EVENT_ICE_DISCONNECTED = 1006	ICE连接断开
BRTC_PLAYER_EVENT_FIRST_AUDIO_PLAYOUT = 1007	音频首帧播放出事件, 若业务侧设置初始静音播放，则可在该事件恢复音频播放
BRTC_PLAYER_EVENT_PLAY_TIME_STATISTIC = 1008 BRTC_PLAYER_EVENT_LOCAL_SDP_SET = 1009 BRTC_PLAYER_EVENT_REMOTE_SDP_ACQUIRED = 1010 BRTC_PLAYER_EVENT_LIBS_DOWNLOAD_COMPLETED = 1011 BRTC_PLAYER_EVENT_LIBS_LOADED_SUCCESS = 1012 BRTC_PLAYER_EVENT_RENDERVIEW_VISIBLE = 1013	启播各阶段耗时统计
BRTC_PLAYER_EVENT_DECODER_OPENED = 1014	解码器成功打开
BRTC_PLAYER_EVENT_START_RECONNECT = 1015	内部播放重连
BRTC_PLAYER_EVENT_LINK_DOWN = 1016	播放链路带宽不足，建议业务侧尝试低码率流播放
BRTC_PLAYER_EVENT_FIRST_AUDIO_PLAYOUT = 1007	没有检测到媒体流

事件回调	含义
void onPrepared()	播放器准备就绪
void onFirstFrameRendered()	首帧渲染事件
void onResolutionChanged(int w, int h)	分辨率更新回调
void onError(int errCode, String msg)	错误错误回调，错误码定义见前文
void onInfoUpdated(int event, Object msg)	事件更新回调，事件码定义见前文
void onPlayerStateChanged(BRTCPlayer.PlayerState currentState)	播放状态更新
void onSEIRecv(ByteBuffer data)	接收到SEI 消息，回调在解码线程,不应在该回调里实现复杂业务

SEI 数据格式(h264):[nal_type(1)|sei_type(1)|sei_size(n+1)|sei_payload(n*255+m)|trailing_bits(1)]

sei_size: 需判断第3个字节起连续0xFF的个数n，加第1个不为0xFF的字节(如: 0x21)，该字段占字节数为 n + 1; 其值为:0xFF*n+0x21;

sei_payload: SEI字段的值（h265）；如:[0x06 0x05 0x18 0x54 0x80 0x83 0x97 0xF0 0x23 0x47 0x4B 0xB7 0xF7 0x4F 0x32 0xB5 0x4E 0x06 0xAC 0x27 0x11 0xFE 0x69 0x79 0x01 0x00 0x00 0x80]

注：h265 与 h264 的 sei 格式略有不同，h264 的nal_type 为 0x06, sei_type 为0x 05; h265 的 nal_type 为 0x4e, sei_type 为两字节 0x01、0x05;

RTCVideoView类 | 枚举类型 | 描述 | |-----|-----| | ScalingType.SCALE_ASPECT_FIT | 缩放模式：适应 | | ScalingType.SCALE_ASPECT_FILL | 缩放模式：拉伸 |

BRTCPlayer接口 | 接口名 | 描述 | |-----|-----| | long
initPlayer(BRTCPlayerParameters playParameters, BRTCPlayerEvents events) | 初始化播放器。
playParameters 播放参数集
events 播放事件监听 | | void setEventObserver(BRTCPlayerEvents events) | 设置播放事件监听 | | void
setSurfaceView(RTCVideoView surfaceView) | 设置播放渲染视窗,当前只支持RTCVideoView | | void
setScalingType(RTCVideoView.ScalingType scaleType) | 设置播放视窗缩放模式 | | void prepareAsync() | 准备播放 | | void
setPlayWhenReady(boolean autoPlay) | 设置是否自动启播，默认false | | void startPlay() | 开始播放 | | void pausePlay() | 暂停播
放,不停止拉流，仅暂停本地媒体流渲染 | | void resumePlay() | 恢复播放,从暂停状态恢复播放 | | void stopPlay() | 停止播放，停
止媒体拉流及渲染，再次播放需调用startPlay() | | boolean hasVideo() | 媒体流是否包含视频流 | | boolean hasAudio() | 媒体流
是否包含音频流 | | void releasePlayer() | 释放播放器 | | void setVolume(double volume) | 设备播放音量 | | void
setStreamUri(String streamPath) | 设置播放媒体流地址，媒体流URL格式为webrtc://domain/app/stream | | PlayerState
getPlayerState() | 获取播放状态 |

SDK接入说明

RTL低延时直播SDK及Demo下载

合规指南

- 开发者：北京百度网讯科技有限公司
- SDK名称：百度智能云RTL低延时直播SDK
- 版本号：Android SDK: v1.1.0; iOS SDK: v1.0.1;
- 主要功能：支持端到端延时小于1000ms超低延时直播，支持AVC/HEVC、AAC/OPUS等媒体格式；
- 个人信息处理规则：[百度云低延时RTL SDK隐私政策](#)
- 合规使用说明：[百度云低延时RTL SDK合规指南](#)

RTL Demuxer传输层SDK

- Android

版本号	更新记录	更新时间	下载
v1.1.0	1、支持基于Minisdp的udp媒体参数自适应信令模式。 2、支持AAC LATM 音频格式。 3、支持媒体服务地址预设。	2024/5/28	下载地址
v1.0.1	1、百度云超低延时传输层SDK首次发布。 2、支持端到端延迟小于1000ms。 3、支持首屏秒开。 4、视频格式支持H.264/H.265，支持B帧。 5、音频格式支持AAC/opus。	2023/12/08	下载地址

- iOS

版本号	更新记录	更新时间	下载
v1.0.1	1、百度云超低延时传输层SDK首次发布。 2、支持端到端延迟小于1000ms。 3、支持首屏秒开。 4、视频格式支持H.264/H.265，支持B帧。 5、音频格式支持AAC/opus。	2023/12/08	下载地址

RTL播放器SDK

基于百度云播放器SDK实现，播放url scheme及option配置相对普通略有不同，接入集成可参数百度云播放器文档及RTL Demuxer传输层SDK集成指引实现。

- Android

版本号	更新记录	更新时间	下载
v3.7.0	1、支持http、udp over MiniSdp。 2、支持端到端直播延迟小于1000ms。 3、支持首屏秒开。 4、视频格式支持H.264/H.265，支持B帧。 5、音频格式支持AAC（adts、latm）/opus。	2024/5/28	下载地址

RTL全功能SDK

- Android

版本号	更新记录	更新时间	下载
v3.3.0	1、支持http、udp over MiniSdp及百度私有udp 信令。 2、支持端到端直播延迟小于1000ms。 3、支持首屏秒开。 4、视频格式支持H.264/H.265，支持B帧。 5、音频格式支持AAC（adts、latm）/opus。	2024/5/28	下载地址

Demo体验

- Android



低延时直播RTL SDK开发者个人信息保护合规指引

亲爱的开发者：

感谢您在所开发的移动应用中集成并使用百度旗下软件开发工具包（SDK）！

百度非常重视用户个人信息保护，包括集成百度旗下低延时直播RTL软件开发工具包（SDK）（以下简称“百度SDK”）的移动应用的最终用户（以下简称“最终用户”）个人信息保护，特制定《低延时直播RTL SDK个人信息保护合规开发者指引》，以供您在所开发的移动应用（以下简称“移动应用”）中集成并使用百度SDK时参照执行。

一、个人信息保护合规基本要求

在所开发的移动应用中集成并使用百度SDK，您需要首先遵守以下个人信息保护合规基本要求：

1. 您应遵守收集、使用最终用户个人信息有关的所有可适用法律法规及规范性文件要求，包括但不限于《个人信息保护法》、《网络安全法》、《App违法违规收集使用个人信息行为认定方法》、《工业和信息化部关于开展APP侵害用户权益专项整治工作的通知》（工信部信管函〔2019〕337号）、《工业和信息化部关于开展纵深推进APP侵害用户权益专项整治行动的通知》（工信部信管函〔2020〕164号）、《工业和信息化部关于进一步提升移动互联网应用服务能力的通知》（工信部信管函〔2023〕26号）等，保护用户个人信息安全。
2. 您的APP需要制定一份独立的隐私政策。隐私政策的内容建议通俗易懂，对您的APP收集、使用个人信息的目的、方式、范围，清晰、完整、准确地向个人信息主体进行明示告知，并充分给予最终用户独立的选择权，确保在获得个人信息主体授权同意后方可进行个人信息的处理活动。《隐私政策》应由用户自主选择是否同意，不应以默认勾选同意的方式或是以欺骗诱导的方式取得用户授权。但请您注意，仅是改善服务质量、提升用户体验、定向推送信息、研发新产品还不足以能成为要求用户同意收集其个人信息的理由。
3. 您应将在移动应用中集成并使用百度SDK服务的情况，以及百度SDK对最终用户必要个人信息的收集、使用和保护规则（具体请见低延时直播RTL SDK隐私政策），在移动应用的显著位置或以其他可触达最终用户的方式告知最终用户（具体请参考本指引第二部分“如何告知最终用户”及第三部分“告知文案示例”），并获得最终用户对于百度SDK收集、使用最终用户相关个人信息的完整、合法、在使用百度SDK服务期间持续有效的授权同意。如果最终用户是未满14周岁的未成年人，请您务必确保获得最终用户的父母或其他监护人对于百度SDK收集、使用最终用户相关个人信息的完整、合法、在使用百度SDK服务期间持续有效的授权同意。
4. 您应向最终用户提供易于操作的访问、更正、删除其个人信息，撤销或更改其授权同意、注销其个人账号等用户权利实现机制。
5. 您应确保在移动应用首次运行时，应在最终用户阅读并同意移动应用隐私政策之后，方可初始化百度SDK进行最终用户信息采集。
6. 百度SDK会根据产品升级优化、提升安全性能、法律及监管要求等原因，不断升级迭代SDK版本，不同版本的SDK获取的字段信息可能会有差异。为了保证合法合规开展合作，并切实履行保护用户个人信息的责任与义务，请您务必确保您已将APP中的百度SDK升级至官方最新版本，以避免因使用旧版本SDK而出现违法违规的问题，导致您的APP遭受监管处罚的风险。百度SDK更新后会及时通过官网通知、站内信、公告、邮件、短信等有效方式对您进行告知，请您及时关注并尽快更新。

除了上述个人信息保护合规基本要求外，您还应遵守您所开发的移动应用所集成并使用的低延时直播RTL SDK隐私政策。

SDK基本业务功能与拓展业务功能：

低延时直播RTL SDK基本业务功能为提供音视频通话、屏幕共享、水印、数据统计、质量监控。低延时直播RTL SDK扩展业务功能为云端录制、旁路转推、云播放器、美颜、变声、背景分割、音视频审核等。

以下是低延时直播RTL SDK获取您的APP的最终用户的个人信息以及权限，由于不同SDK版本采集的字段与是否可选可能存在一定差异，具体采集情况以实际接入的SDK版本为准：

SDK收集个人信息情况：

功能及服务	个人信息类型	个人信息字段（区分必选信息和可选信息）
网络质量监测，根据网络状态调整网络策略	WiFi状态	WiFi状态（必选）
网络质量监测，在WiFi切换时实现快速断网重连	WiFi-SSID	WiFi-SSID（必选）
设备性能检测，为了更好的用户体验，进行机型分级，针对不同的机型的性能提供不同的功能模型及服务策略	设备型号	设备型号（必选）
SDK通信质量检测	IP地址	IP地址（必选）

SDK初始化设置：

请务必确保您已将低延时直播RTL SDK升级到最新版。上述版本调用初始化函数不会采集用户个人信息，也不会向百度联盟后台上报数据。请确保用户同意《隐私政策》后再进行start/autoTrace采集配置，方可采集用户个人信息并上报数据。

调用初始化函数文档：<https://cloud.baidu.com/doc/LSS/s/flputdwdg>

二、如何告知最终用户

为帮助您明确地告知最终用户百度SDK个人信息收集、使用和保护相关事宜，我们为您提供了以下告知方式，供您参考执行：

1. 在移动应用隐私政策中“**个人信息共享**”条款部分或“**所集成的第三方SDK**”条款部分告知最终用户相应功能/服务由百度SDK提供，并显示相应**百度SDK隐私政策链接**以告知最终用户，百度SDK收集、使用的最终用户个人信息类型、目的及用途。移动应用隐私政策在用户首次打开移动应用或者在移动应用的注册/登记界面通过**协议在线展示**的方式向用户展示，并获得最终用户**明示同意**（例如：点击“同意”，或勾选“√”）。
2. 在移动应用隐私政策中“**个人信息共享**”条款部分或“**所集成的第三方SDK**”条款部分告知最终用户相应功能/服务由百度SDK提供，并参考相应百度SDK隐私政策内容，以**条款或表格等形式列明**收集、使用的最终用户个人信息类型、目的及用途。移动应用隐私政策在用户首次打开移动应用或者在移动应用的注册/登记界面通过**协议在线展示**的方式向用户展示，并获得最终用户**明示同意**（例如：点击“同意”，或勾选“√”）。
3. 当最终用户在移动应用中首次打开/使用相应功能/服务时，以**弹窗、页面提示**方式显示相应**百度SDK隐私政策链接**，以告知最终用户相应功能/服务由百度SDK提供，百度SDK为提供相应功能/服务而收集、使用的最终用户个人信息类型、目的及用途，并获得最终用户**明示同意**（例如：点击“同意”，或勾选“√”）。

三、告知文案示例

为帮助您明确地告知最终用户百度SDK个人信息保护规则相关事宜，我们为您提供了以下告知方文案示例，供您参考执行：

文案示例A

为向您提供低延时直播功能/服务，我们集成了低延时直播RTL SDK。在为您提供低延时直播频功能/服务时，低延时直播RTL SDK需要收集、使用您必要的个人信息。关于低延时直播RTL SDK收集、使用的个人信息类型、目的及用途，以及低延时直播RTL SDK将如何保护所收集、使用的个人信息，请您仔细阅读《低延时直播RTL SDK隐私政策》了解。

文案示例B

为保障App相关功能的实现与应用安全稳定的运行，我们可能会接入由第三方提供的SDK实现相关目的，具体接入的相关第三方SDK列明如下：

第三方SDK名称	使用目的	官网链接/隐私政策链接
低延时直播RTL SDK	提供低延时直播功能/服务	https://cloud.baidu.com/doc/LSS/s/flputdwdg

文案示例C

为向您提供低延时直播功能/服务，我们集成了低延时直播RTL SDK。在为您提供低延时直播功能/服务时，低延时直播RTL SDK收集、使用您的wifi状态、设备型号、ip地址信息，用于网络质量、设备性能、SDK通信质量检测目的/用途。具体请您仔细阅读《低延时直播RTL SDK隐私政策》了解。

文案示例D

为保障App相关功能的实现与应用安全稳定的运行，我们可能会接入由第三方提供的SDK实现相关目的，具体接入的相关第三方SDK列明如下：

第三方SDK名称	业务功能	个人信息类型	调用权限类型	具体目的/用途
低延时直播RTL SDK	音视频通话	WIFI状态、WIFI-SSID、设备型号、IP地址	相机、录音、录屏、相册	用于通信质量和设备性能检测，保障用户正常使用音视频通话、本地录制及屏幕共享等功能

四、使用SDK服务的合规注意事项

1. 您接入百度SDK前，应当仔细阅读SDK的协议约定、本合规规范、用户协议、隐私政策等内容，并依据相关内容对您APP的《隐私政策》及您APP收集、存储、使用、共享等处理个人信息的情况进行合规自查。
2. 您知悉并认可百度SDK具备收集个人信息的功能，并认可该等信息的收集均为双方合作之必要目的所需。
3. 您承诺已制定并按照相关要求公示您APP的《隐私政策》，并已清晰、明确、显著地说明有关通过SDK收集个人信息的必要性、收集数据的范围、方式以及用途。同时，您应确保在APP首次运行时以弹窗等合规方式显著提示用户阅读您APP的《隐私政策》并取得用户的合法授权同意，经过合法授权后再初始化百度SDK进行个人信息的收集与处理。
4. 您已认真阅读并理解百度SDK平台协议、合作规范、隐私政策、接入文档等约定和要求，并承诺在您使用百度SDK服务期间，针对百度SDK收集、使用、处理、共享、转让相关个人信息，您已取得了用户持续有效的授权和同意，并保证您不会违反国家相关法律法规、相关国家标准以及双方约定的目的。
5. 如果您的APP面向不满十四周岁的儿童及未成年人用户提供服务，您承诺遵守儿童个人信息保护及未成年人保护相关的法律法规，您承诺已采取相关措施并保证已获得其监护人的授权同意。
6. 如因您违反百度SDK的平台协议、合作规范、隐私政策、接入文档等约定，导致您的用户或第三方对百度主张任何形式的索赔或权利要求，或导致百度因此产生任何法律纠纷的，您将负责解决并承担全部责任，如因此给百度及其关联主体造成损失的，您应赔偿因此给百度及其关联主体造成的全部损失。
7. 您保证对于您从百度SDK获取的数据，无论是在合作期间或是合作停止后，均承担保密义务，不得擅自提供、泄露、透露给任何第三方，并应采取一切合法措施以使上述数据免于散发、传播、披露、复制、滥用及被无关人员接触，避免导致相关数据被超出双方合作目的使用。

低延时直播RTL SDK隐私政策

欢迎使用低延时直播RTL SDK软件开发工具包（SDK）（简称“低延时直播RTL SDK”）服务！

低延时直播RTL SDK 是一款为移动应用开发者（以下简称“开发者”）提供稳定、超低延时、高质量的传输层软件开发工具包（SDK）。开发者在其移动应用内集成后，可通过低延时直播RTL SDK平台向其移动应用（以下简称“开发者应用”）的最终用户（以下简称“最终用户”）提供本隐私政策所说明的功能及服务。开发者在其移动应用集成并使用低延时直播RTL SDK服务时，委托低延时直播RTL SDK处理开发者应用相关数据信息，其中可能包括开发者应用最终用户（以下简称“最终用户”）的个人信息。此隐私政策旨在帮助开发者及最终用户了解我们收集最终用户个人信息的类型及我们如何利用和保护最终用户的个人信息。为了便于开发者及最终用户阅读及理解，我们将专门术语进行了定义，请参见本隐私政策“附录1：名词解释”来了解这些定义的具体内容。

特别说明：

- 1、如果开发者在其移动应用中集成并使用低延时直播RTL SDK服务，则开发者应承诺：
 - （1）开发者应遵守收集、使用最终用户个人信息有关的所有可适用法律、政策和法规，保护用户个人信息安全。
 - （2）开发者应将在其移动应用中集成并使用低延时直播RTL SDK服务的情况，以及低延时直播RTL SDK对最终用户必要个人信息的收集、使用和保护规则（即本隐私政策），在其移动应用的显著位置或以其他可触达最终用户的方式告知最终用户（包括但不限于：在其移动应用隐私政策显眼处提供最终用户可浏览本隐私政策的链接），并获得最终用户对于低延时直播RTL SDK

收集、使用最终用户相关个人信息的完整、合法、在使用低延时直播RTL SDK服务期间持续有效的授权同意。如果开发者的移动应用最终用户是未满14周岁的未成年人，请开发者务必确保获得最终用户的父母或其他监护人对于低延时直播RTL SDK收集、使用最终用户相关个人信息的完整、合法、在使用低延时直播RTL SDK服务期间持续有效的授权同意。

(3) 开发者应向最终用户提供易于操作的查阅、更正、补充、删除、复制或转移其个人信息，撤回或更改其授权同意，注销其个人账号，要求开发者就个人信息处理规则作出解释说明等用户权利实现机制。

(4) 关于上述承诺的具体落地执行可参考《低延时直播RTL SDK开发者个人信息保护合规指引》。

2、我们希望集成并使用低延时直播RTL SDK服务的开发者应用以合法合规的方式收集、使用最终用户的个人信息，但我们并不了解且无法控制任何开发者以及他们的移动应用如何使用他们所控制的最终用户个人信息，因此也不应为其行为负责。我们建议最终用户在认真阅读开发者应用相关隐私政策，在确认充分了解并同意他们如何收集、使用最终用户的个人信息后再使用开发者应用。

3、本隐私政策不适用于展示在、链接到或再封装我们的服务的那些适用第三方隐私政策、并由第三方提供的服务。虽然第三方展示在、链接到或再封装我们的服务，但我们并不了解或控制其行为，因此也不为其行为负责。请开发者及最终用户在已查看并接受其隐私政策之前，谨慎访问或使用其服务。

4、最终用户具体获得的低延时直播RTL SDK服务内容 by 开发者根据其移动应用需要进行选择，可能因为最终用户所使用的开发者应用不同而有所差异，低延时直播RTL SDK可能获得的个人信息取决于最终用户所使用的开发者应用的具体类型/版本以及最终用户所使用的功能。如果在部分开发者应用版本中不涵盖某些服务内容或未提供特定功能，则本隐私政策中涉及到上述服务/功能及相关个人信息的内容将不适用。

请开发者及最终用户务必认真阅读本隐私政策，在确认充分了解并同意后再集成并使用低延时直播RTL SDK服务。

本隐私政策将帮助开发者及最终用户了解以下内容：

1. 我们如何收集和使用最终用户的个人信息
2. 我们如何使用 Cookie 和同类技术
3. 我们如何共享、转让、公开披露最终用户的个人信息
4. 我们如何保存及保护最终用户的个人信息
5. 我们如何保障最终用户的个人信息相关权利行使
6. 我们如何处理未成年人的个人信息
7. 隐私政策的修订
8. 如何联系我们

我们珍视最终用户在向我们提供最终用户个人信息时对我们的信任，我们将按照本隐私政策处理最终用户的个人信息并保障最终用户信息的安全。

一、我们如何收集和使用最终用户的个人信息

(一) 为帮助开发者向最终用户提供相应功能及服务

为了帮助开发者向最终用户提供相应功能及服务，当最终用户使用相应功能及服务时，我们会通过开发者应用向系统申请最终用户设备的相应权限。开发者应确保最终用户可以随时通过取消系统授权开发者应用获取相应设备权限或其他开发者应用提供的授权设置，停止我们对最终用户个人信息的收集，之后最终用户可能将无法使用基于相应个人信息而提供的相关服务或功能，或者无法达到基于相应个人信息提供的相关服务拟达到的效果，但不会影响最终用户正常使用低延时直播RTL SDK 的其他不基于相应个人信息即可实现的业务功能。

1. 各项功能及服务涉及的个人信息

序号	功能及服务	个人信息类型	收集方式	适用系统版本
1	网络质量监测，根据网络状态调整网络策略	WIFI状态（必选）	SDK直接采集	iOS及Android
2	网络质量监测，在WIFI切换时实现快速断网重连	WIFI-SSID（必选）	SDK直接采集	iOS及Android
3	设备性能检测，为了更好的用户体验，进行机型分级，针对不同的机型的性能提供不同的功能模型及服务策略	设备型号（必选）	SDK直接采集	iOS及Android
4	SDK通信质量检测	IP地址（必选）	SDK直接采集	iOS及Android

（二）保证服务安全、优化和改善服务目的

为了帮助开发者向最终用户提供上述功能及服务，同时为了更准确定位并解决开发者以及最终用户在使用低延时直播RTL SDK产品和服务时遇到的问题，改进及优化低延时直播RTL SDK产品和服务在开发者侧以及最终用户侧的双重体验，更准确定位并解决最终用户在使用低延时直播RTL SDK服务时遇到的问题，改进及优化低延时直播RTL SDK的服务体验，提高低延时直播RTL SDK服务的安全性，预防、发现、调查欺诈、危害安全、非法或违反与我们的协议、政策或规则的行为，以保护开发者、最终用户、我们或我们的关联公司、合作伙伴及社会公众的合法权益，我们会收集最终用户的设备信息、位置信息、日志信息及其他与登录环境相关的信息。

（三）个人信息的匿名化处理

在不公开披露、对外提供最终用户个人信息的前提下，百度公司有权对匿名化处理后的用户数据库进行挖掘、分析和利用（包括商业性使用），有权对产品/服务使用情况进行统计并与公众/第三方共享匿名化处理后的统计信息。

（四）事先征得授权同意的例外 请注意，在以下情形中，收集、使用个人信息无需事先征得最终用户的授权同意：

- 1. 与国家安全、国防安全直接相关的；
- 2. 为订立、履行个人作为一方当事人的合同所必需；
- 3. 为履行法定职责或者法定义务所必需；
- 4. 为应对突发公共卫生事件，或者紧急情况下为保护自然人的生命健康和财产安全所必需；
- 5. 与犯罪侦查、起诉、审判和判决执行等直接有关的；
- 6. 出于维护最终用户或其他个人的生命、财产等重大合法权益但又很难得到本人同意的；
- 7. 依照法律法规的规定在合理的范围内收集最终用户自行向社会公众公开或其他已经合法公开的个人信息；
- 8. 依照法律法规的规定在合理的范围内从合法公开披露的信息中收集最终用户的个人信息，如合法的新闻报道、政府信息公开等渠道；
- 9. 为公共利益实施新闻报道、舆论监督等行为，在合理的范围内处理个人信息；
- 10. 学术研究机构基于公共利益开展统计或学术研究所必要，且对外提供学术研究或描述的结果时，对结果中所包含的个人信息进行去标识化处理的；
- 11. 法律法规规定的其他情形。

提示最终用户注意，当我们要将信息用于本隐私政策未载明的其它用途时，会事先征求最终用户的同意。

 二、我们如何使用 Cookie 和同类技术

Cookie是支持服务器端（或者脚本）在客户端上存储和检索信息的一种机制。当最终用户使用低延时直播RTL SDK SDK产品或

服务时，我们会向最终用户的设备发送一个或多个Cookie或匿名标识符。当最终用户与低延时直播RTL SDK服务进行交互时，我们允许Cookie或者匿名标识符发送给百度公司服务器。Cookie 通常包含标识符、站点名称以及一些号码和字符。运用Cookie技术，百度公司能够了解最终用户的使用习惯，记住最终用户的偏好，省去最终用户输入重复信息的步骤，为最终用户提供更加周到的个性化服务，或帮最终用户判断最终用户账户的安全性。Cookie还可以帮助我们统计流量信息，分析页面设计和广告的有效性。

我们不会将 Cookie 用于本政策所述目的之外的任何用途。最终用户可根据自己的偏好管理或删除 Cookie。有关详情，请参见 [AboutCookies.org](https://www.aboutcookies.org)。最终用户可以清除计算机上保存的所有 Cookie，大部分网络浏览器都设有阻止 Cookie 的功能。但如果最终用户这么做，则需要在每一次访问我们的网站时亲自更改用户设置，但最终用户可能因为该等修改，无法登录或使用依赖于 Cookie的百度公司提供的服务或功能。

三、我们如何共享、转让、公开披露最终用户的个人信息

（一）共享

除非经过您本人事先单独同意或符合其他法律法规规定的情形，我们不会向除百度公司以外的第三方共享您的个人信息，但经过处理无法识别特定个人且不能复原的除外。

对我们与之共享个人信息的公司、组织和个人，我们会对其数据安全环境进行调查，与其签署严格的保密协定，要求他们按照依法采取保密和安全措施来处理个人信息。

1. 在下列情况下，经过最终用户的授权同意，我们可能会共享的个人信息：

仅为实现本隐私政策中声明的目的，我们的某些服务将由授权合作伙伴提供。我们可能会与合作伙伴共享最终用户的某些个人信息，以提供更好的客户服务和用户体验。我们仅会出于合法、正当、必要、特定、明确的目的共享最终用户的个人信息，并且只会共享与提供服务相关的个人信息。我们的合作伙伴无权将共享的个人信息用于任何其他用途。

目前，我们的授权合作伙伴包括以下类型：

(1) 服务平台或服务提供商。百度各产品接入了丰富的第三方服务。当最终用户选择使用该第三方服务时，最终用户授权我们将该信息提供给第三方服务平台或服务提供商，以便其基于相关信息为最终用户提供服务。

(2) 软硬件/系统服务提供商。当第三方软硬件/系统产品或服务与百度的产品或服务结合为最终用户提供服务时，经最终用户授权，我们会向第三方软硬件/系统服务提供商提供最终用户必要的个人信息，以便最终用户使用服务，或用于我们分析产品和服务使用情况，来提升最终用户的使用体验。

(3) 广告、咨询类服务商/广告主。未经最终用户授权，我们不会将最终用户的个人信息与提供广告、咨询类服务商共享。但我们可能会将经处理无法识别最终用户的身份且接收方无法复原的信息，例如经匿名化处理的画像，与广告或咨询类服务商或广告主共享，以帮助其在不识别最终用户个人的前提下，提升广告有效触达率，以及分析我们的产品和服务使用情况等。

对我们与之共享个人信息的公司、组织和个人，我们会对其数据安全环境进行调查，与其签署严格的保密协定，要求他们按照我们的说明、本隐私政策以及其他任何相关的保密和安全措施来处理个人信息。

（二）转让

我们不会将最终用户的个人信息转让给除关联公司外的任何公司、组织和个人，但以下情况除外：

1. 事先获得最终用户的明确授权或同意；
2. 满足法律法规、法律程序的要求或强制性的政府要求或司法裁定；
3. 如果我们或我们的关联公司涉及合并、分立、清算、资产或业务的收购或出售等交易，最终用户的个人信息有可能作为此类交易的一部分而被转移，我们将确保该等信息在转移时的机密性，并尽最大可能确保新的持有最终用户个人信息的公司、组织继续受此隐私政策的约束，否则我们将要求该公司、组织重新向最终用户征求授权同意。

（三）公开披露

我们仅会在以下情况下，公开披露最终用户的个人信息：

1. 获得最终用户的单独同意；
2. 基于法律法规、法律程序、诉讼或政府主管部门强制性要求下。

（四）共享、转让、公开披露个人信息时事先征得授权同意的例外

在以下情形中，共享、转让、公开披露个人信息无需事先征得最终用户的授权同意：

1. 与国家安全、国防安全直接相关的；
2. 为订立、履行个人作为一方当事人的合同所必需；
3. 为履行法定职责或者法定义务所必需；
4. 与公共安全、公共卫生、重大公共利益直接相关的。例如：为应对突发公共卫生事件，或者紧急情况下为保护自然人的生命健康和财产安全所必需；
5. 与犯罪侦查、起诉、审判和判决执行等直接相关的；
6. 出于维护个人信息主体或其他个人的生命、财产等重大合法权益但又很难得到本人同意的；
7. 依照法律法规的规定在合理的范围内处理个人自行公开或者其他已经合法公开的个人信息，例如：合法的新闻报道、政府信息公开等渠道等；
8. 法律、行政法规另有规定的情形。

四、我们如何保存及保护最终用户的个人信息

（一）保存期限

我们将在低延时直播RTL SDK自身提供服务以及开发者应用提供服务所需的期限内保存您的个人信息，但法律法规对保存期限另有规定、您同意留存更长的期限、保证服务的安全与质量、实现争议解决目的、技术上难以实现等情况下，在前述保存期限到期后，我们将依法、依约或在合理范围内延长保存期限。

在超出保存期限后，我们将根据法律规定删除您的个人信息或进行匿名化处理。

（二）保存地域

原则上，我们在中华人民共和国境内收集和产生的个人信息，将存储在中华人民共和国境内。如您的个人信息可能会被转移到您使用产品或服务所在国家/地区的境外管辖区，或者受到来自这些管辖区的访问，我们会严格履行法律法规规定的义务并**按照法律规定事先获得您的单独同意**。此类管辖区可能设有不同的数据保护法，甚至未设立相关法律。在此类情况下，我们会按照中国现行法律的规定传输您的个人信息，并确保您的个人信息得到在中华人民共和国境内足够同等的保护。例如，我们会请求您对跨境转移个人信息的同意，或者在跨境数据转移之前实施数据去标识化等安全举措。

（三）安全措施

1. 我们会以“最小化”原则收集、使用、存储和传输用户信息，并通过用户协议和隐私政策告知您相关信息的使用目的和范围。
2. 我们非常重视信息安全。我们成立了专责团队负责研发和应用多种安全技术和程序等，我们会对安全管理负责人和关键安全岗位的人员进行安全背景审查，我们建立了完善的信息安全管理制度和内部安全事件处置机制等。我们会采取适当的符合业界标准的安全措施和技术手段存储和保护您的个人信息，以防止您的信息丢失、遭到被未经授权的访问、公开披露、使用、毁损、丢失或泄漏。我们会采取一切合理可行的措施，保护您的个人信息。我们会使用加密技术确保数据的保密性；我们会使用受信赖的保护机制防止数据遭到恶意攻击。
3. 我们会对员工进行数据安全的意识培养和安全能力的培训和考核，加强员工对于保护个人信息重要性的认识。我们会对处理个人信息的员工进行身份认证及权限控制，并会与接触您个人信息的员工、合作伙伴签署保密协议，明确岗位职责及行为准则，确保只有授权人员才可访问个人信息。若有违反保密协议的行为，会被立即终止与百度公司的合作关系，并会被追究相关法律责任，对接触个人信息人员在离岗时也提出了保密要求。
4. 我们提醒您注意，互联网并非绝对安全的环境，当您通过低延时RTL SDK中嵌入的第三方社交软件、电子邮件、短信等与其

他用户交互您的地理位置或行踪轨迹信息时，不确定第三方软件对信息的传递是否完全加密，注意确保您个人信息的安全。

5. 我们也请您理解，在互联网行业由于技术的限制和飞速发展以及可能存在的各种恶意攻击手段，即便我们竭尽所能加强安全措施，也不可能始终保证信息的百分之百安全。请您了解，您使用我们的产品和/或服务时所用的系统和通讯网络，有可能在我们控制之外的其他环节而出现安全问题。
6. 根据我们的安全管理制度，个人信息泄露、毁损或丢失事件被列为最特大安全事件，一旦发生将启动公司最高级别的紧急预案，由安全部、公共事务部、法务部等多个部门组成联合应急响应小组处理。

（四）安全事件通知

1. 我们会制定网络安全事件应急预案，及时处置系统漏洞、计算机病毒、网络攻击、网络侵入等安全风险，在发生危害网络安全的事件时，我们会立即启动应急预案，采取相应的补救措施，并按照规定向有关主管部门报告。
2. 个人信息泄露、毁损、丢失属于公司级特大安全事件，我们会负责定期组织工作组成员进行安全预案演练，防止此类安全事件发生。若一旦不幸发生，我们将按照最高优先级启动应急预案，组成紧急应急小组，在最短时间内追溯原因并减少损失。
3. 在不幸发生个人信息安全事件后，我们将按照法律法规的要求，及时向您告知安全事件的基本情况和可能的影响、我们已采取或将要采取的处理措施、您可自主防范和降低的风险的建议、对您的补救措施等。我们将及时将事件相关情况以站内通知、短信通知、电话、邮件等您预留的联系方式告知您，难以逐一告知时我们会采取合理、有效的方式发布公告。同时，我们还将按照监管部门要求，主动上报个人信息安全事件的处置情况。请您理解，根据法律法规的规定，如果我们采取的措施能够有效避免信息泄露、篡改、丢失造成危害的，除非监管部门要求向您通知，我们可以选择不向您通知该个人信息安全事件。

🔗 五、我们如何处理未成年人的个人信息

百度公司非常重视对未成年人信息的保护。

低延时RTL SDK的产品和服务主要面向成年人。如果最终用户是未满14周岁的未成年人，请务必在使用已集成低延时直播RTL SDK的开发者应用前，在父母或其他监护人监护、指导下共同仔细阅读开发者应用隐私政策、本隐私政策以及《[百度儿童个人信息保护声明](#)》，并在征得监护人同意的前提下使用开发者应用或提供个人信息。

如果我们发现在未事先获得可证实的父母同意的情况下收集了未成年人的个人信息，将会采取措施尽快删除相关信息。

如果任何时候监护人有理由相信我们在未获监护人同意的情况下收集了未成年人的个人信息，请通过工单联系我们，我们会采取措施尽快删除相关数据。

🔗 六、我们如何保障最终用户的个人信息相关权利行使

按照中国相关的法律、法规、标准，以及其他国家、地区的通行做法，我们将尽力协调、支持并保障最终用户对自己的个人信息行使以下权利：

1. 查阅权、更正及补充权、复制权、帐号注销权

鉴于最终用户通过开发者应用使用低延时直播RTL SDK服务，并且使用服务时并不会注册/登录百度帐号，为保障最终用户查阅、更正、补充、复制其个人信息以及注销其开发者应用帐号的权利实现，我们在本隐私政策以及其他与开发者的约定中，要求开发者承诺提供便于操作的实现方式。如最终用户要查阅、更正、补充、复制其个人信息或注销其开发者应用帐号，应通过开发者提供的上述权利实现方式，如开发者未按照承诺进行提供，最终用户可通过【八、如何联系我们】中的方式与我们取得联系，我们将尽力协调、支持并保障最终用户的上述权利实现。

2. 删除权

为保障最终用户删除其个人信息的权利实现，我们在本隐私政策以及其他与开发者的约定中，要求开发者承诺提供便于操作的实现方式。如最终用户要删除其个人信息，应通过开发者提供的上述权利实现方式，如开发者未按照承诺进行提供，在以下情形中，最终用户可以通过【八、如何联系我们】中的方式与我们取得联系，向我们提出删除个人信息的请求：

1. 如果我们违反法律法规或与最终用户的约定处理其个人信息；
2. 如果我们的处理目的已实现、无法实现或者为实现处理目的不再必要；

3. 如果我们停止提供产品或者服务，或者保存期限已届满；
4. 如果最终用户撤回同意；
5. 法律、行政法规规定的其他情形。

当最终用户从我们的服务中删除信息后，我们可能不会立即在备份系统中删除相应的信息，但会在备份更新时删除这些信息。请最终用户知晓并理解，如果法律、行政法规规定的或本隐私政策说明的保存期限未届满，或者删除个人信息从技术上难以实现的，我们会停止除存储和采取必要的安全保护措施之外的处理。

3. 撤回同意

每个业务功能需要一些基本的个人信息才能得以完成。对于额外收集的个人信息的收集和使用，最终用户可以随时给予或收回其授权同意。

最终用户可以在设备系统中直接关闭本隐私政策说明的我们可能调用的设备系统权限，或开发者应用提供的其他授权设置（如适用），改变同意范围或撤回其授权。

当最终用户撤回同意后，我们无法继续为最终用户提供撤回同意所对应的服务，也将不再使用最终用户相应的个人信息。但最终用户撤回同意的决定，不会影响此前基于其同意而开展的个人信息处理。

4. 可携带权

为保障最终用户转移其个人信息的权利实现，我们在本隐私政策以及其他与开发者的约定中，要求开发者承诺提供便于操作的实现方式。如最终用户要转移其个人信息，应通过开发者提供的上述权利实现方式，如开发者未按照承诺进行提供，最终用户可以通过【八、如何联系我们】中的方式与我们取得联系，我们将尽力协调、支持并保障其上述权利实现。

在法律法规规定的条件下，同时符合国家网信部门规定指令和条件的，如果技术可行，最终用户也可以要求我们将其个人信息转移至最终用户指定的其他主体。

5. 提前获知产品和服务停止运营权。

低延时直播RTL SDK愿一直陪伴开发者和最终用户，若因特殊原因导致低延时直播RTL SDK产品停止运营，我们将在合理期间内在产品或服务的主页面或站内信或向开发者和最终用户发送电子邮件或其他合适的能触达其方式通知开发者和最终用户，并将停止对最终用户个人信息的收集，同时会按照法律规定对已收集的最终用户的个人信息进行删除或匿名化处理。

6. 获得解释的权利

最终用户有权要求我们就个人信息处理规则作出解释说明。最终用户可以通过【八、如何联系我们】中的方式与我们取得联系。

七、隐私政策的修订与通知

我们的隐私政策可能变更。

未经最终用户明确同意，我们不会削减最终用户按照本隐私政策所应享有的权利。我们会在本页面上发布对本隐私政策所做的任何变更。

对于重大变更，我们会在低延时RTL SDK官方网站的主要曝光页面向开发者以及最终用户公示，并以任何可触达的方式通知开发者。若开发者不同意该等变更可以停止集成并使用低延时RTL SDK产品和服务，若开发者继续集成并使用低延时RTL SDK产品和/或服务，即表示开发者同意受修订后的本隐私政策的约束，并将此变更通知最终用户，获取最终用户对此变更的完整、合法、在使用低延时RTL SDK服务期间持续有效的授权同意。若最终用户不同意该等变更，可以停止使用低延时直播RTL SDK产品和服务，开发者应向用户提供相应实现机制；若最终用户继续使用低延时RTL SDK产品和/或服务，即表示最终用户同意受修订后的本隐私政策的约束。

本政策所指的重大变更包括但不限于：

1. 我们的服务模式发生重大变化。如处理个人信息的目的、处理的个人信息类型、个人信息的使用方式等；
2. 我们在所有权结构、组织架构等方面发生重大变化。如业务调整、破产并购等引起的所有者变更等；

3. 个人信息共享、转让或公开披露的主要对象发生变化；
4. 最终用户参与个人信息处理方面的权利及其行使方式发生重大变化；
5. 我们负责处理个人信息安全的责任部门、联络方式及投诉渠道发生变化时；
6. 个人信息安全影响评估报告表明存在高风险时。

本政策更新后，我们会将本政策的旧版本存档，供最终用户查阅。

如有本隐私政策未尽事宜，以《百度隐私权保护声明》为准。

八、如何联系我们

低延时RTL SDK的成长离不开各方开发者及最终用户的共同努力，我们非常感谢开发者及最终用户对低延时直播RTL SDK数据更新、使用反馈方面的贡献。

开发者及最终用户可以通过工单反馈开发者及最终用户对低延时直播RTL SDK产品和服务的建议以及在使用过程中遇到的问题，以帮助我们优化产品功能及服务，使更多用户更加便捷的使用我们的产品和服务。

开发者及最终用户可以通过个人信息保护问题反馈平台(<http://help.baidu.com/personalinformation>) 同我们联系。

开发者及最终用户也可以通过如下联络方式同我们联系：

中国北京市海淀区上地十街10号

北京百度网讯科技有限公司 法务部

邮政编码：100085

为保障我们高效处理开发者/最终用户的问题并及时向开发者/最终用户反馈，需要开发者/最终用户提交身份证明、有效联系方式和书面请求及相关证据，我们会在验证开发者/最终用户的身份后处理开发者/最终用户的请求。

如果开发者/最终用户对我们的回复不满意，特别是最终用户认为我们的个人信息处理行为损害了最终用户的合法权益，开发者/最终用户还可以通过以下外部途径寻求解决方案：向北京市海淀区人民法院提起诉讼。

附录1：名词解释 个人信息是指以电子或者其他方式记录的与已识别或者可识别的自然人有关的各种信息，不包括匿名化处理后的信息。个人信息包括姓名、出生日期、身份证件号码、个人生物识别信息、住址、通信通讯联系方式、通信记录和内容、帐号密码、财产信息、征信信息、行踪轨迹、住宿信息、健康生理信息、交易信息等。敏感个人信息是指一旦泄露或者非法使用，容易导致自然人的人格尊严受到侵害或者人身、财产安全受到危害的个人信息，包括生物识别、宗教信仰、特定身份、医疗健康、金融账户、行踪轨迹等信息，以及不满十四周岁未成年人的个人信息。

设备是指可用于使用百度产品和/或服务的装置，例如桌面设备、平板电脑或智能手机。

设备信息可能包括最终用户用于安装、运行低延时直播RTL SDK的终端设备的设备属性信息（例如最终用户的硬件型号，操作系统版本，设备配置，国际移动设备身份码IMEI、国际移动用户识别码IMSI、网络设备硬件地址MAC、广告标识符IDFA、供应商标识符IDFV、移动设备识别码MEID、匿名设备标识符OAID、集成电路卡识别码ICCID、Android ID、硬件序列号等唯一设备标识符）、设备连接信息（例如浏览器的类型、电信运营商、使用的语言、WIFI信息）以及设备状态信息（例如设备应用安装列表）。

日志信息是指我们的服务器所自动记录最终用户在访问低延时RTL SDK时所发出的请求，例如最终用户的IP 地址、浏览器的类型和使用的语言、硬件设备信息、操作系统的版本、网络运营商的信息、最终用户访问服务的日期、时间、时长等最终用户在使用我们的产品或服务时提供、形成或留存的信息。

Cookie是指支持服务器端（或者脚本）在客户端上存储和检索信息的一种机制，通过增加简单、持续的客户端状态来扩展基于Web的客户端/服务器应用。服务器在向客户端返回HTTP对象的同时发送一条状态信息，并由客户端保存。状态信息中说明了该状态下有效的URL范围。此后，客户端发起的该范围内的HTTP请求都将把该状态信息的当前值从客户端返回给服务器，这个状态信息被称为Cookie。

位置信息是指通过GPS信息，WLAN接入点、蓝牙、基站以及其他传感器信息所获取的**精确位置信息**，也包括通过IP地址或其他网络信息等获取的**粗略地理位置信息**。

用户画像是指通过收集、汇聚、分析个人信息，对某特定自然人个人特征，如其职业、经济、健康、教育、个人喜好、信用、行为等方面做出分析或预测，形成其个人特征模型的过程。直接使用特定自然人的个人信息，形成该自然人的特征模型，称为直接用户画像。使用来源于特定自然人以外的个人信息，如其所在群体的数据，形成该自然人的特征模型，称为间接用户画像。

去标识化是指个人信息经过处理，使其在不借助额外信息的情况下无法识别特定自然人的过程。

匿名化是指通过对个人信息的技术处理，使得个人信息主体无法被识别，且处理后的信息不能被复原的过程。个人信息经匿名化处理后所得的信息不属于个人信息。

百度平台是指百度公司旗下各专门频道或平台服务（包括百度搜索、百度APP及衍生版、百度百科、百度知道、百度贴吧、百度手机助手及其他百度系产品<https://www.baidu.com/more/>）等网站、程序、服务、工具及客户端。

关联公司是指低延时RTL SDK的经营者【北京百度网讯科技有限公司】及其他与百度公司存在关联关系的公司的单称或合称。“关联关系”是指对于任何主体（包括个人、公司、合伙企业、组织或其他任何实体）而言，即其直接或间接控制的主体，或直接或间接控制其的主体，或直接或间接与其受同一主体控制的主体。前述“控制”指，通过持有表决权、合约或其他方式，直接或间接地拥有对相关主体的管理和决策作出指示或责成他人作出指示的权力或事实上构成实际控制的其他关系。

再次感谢开发者以及最终用户对低延时RTL SDK的信任和使用！

北京百度网讯科技有限公司

【2023】年【12】月【15】日更新

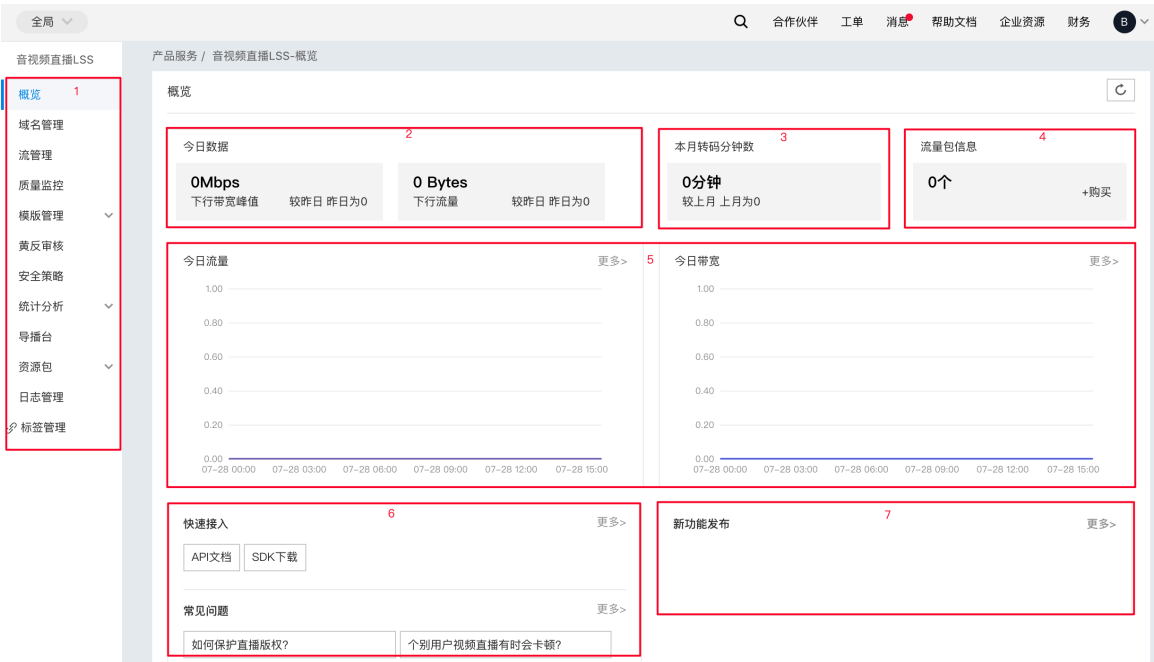
【2023】年【12】月【15】日生效

控制台操作指南

LSS概览

为了帮助用户了解LSS运行的整体情况，实时观察当前业务状态，LSS提供了概览页面。

🔗 控制台指引



操作指引

LSS控制台为您提供方便快捷的LSS操作体验，在控制台，您可以完成登录LSS、开通LSS、配置LSS等操作。

序号	区域	说明
1	左侧导航栏	LSS导航栏。详细功能介绍，请参见 操作指引 功能列表。
2	今日数据	为您展示 LSS 今日宽带、流量使用情况的数据。
3	本月转码分钟数	为您展示 LSS 本月及上月的转码分钟数。
4	流量包信息	为您展示您的流量包信息，您可以点击 购买 按钮进行购买操作。
5	数据图	“今日流量”和“今日带宽”的波形图。
6	快速接入&常见问题	为您提供API接口、SDK的快速入口，展示常见问题链接。
7	新功能发布	为您展示 LSS 最新功能发布动态。

操作	包含操作说明
域名管理	域名管理包含以下操作: • 添加域名 • 删除域名 • 开启域名 • 关闭域名 • 配置CNAME • 配置HTTPS
	流管理包含以下操作: • 新建流

流管理	● 查询流信息：您可以查看已经成功推流的流（Stream）和当前的直播状态，以及流监控、禁播和删除流操作。
质量监控	质量监控包含以下操作： ● 查看监测数据：您可以查看某路流在近7天内任意时间段的质量监控信息。
模版管理	模版管理包含以下操作： ● 转码模版 ● 录制模版 ● 水印模版 ● 缩略图模版 ● 通知模版
黄反审核	黄反审核包含以下操作： ● 配置黄反审核策略：配置黄色、反动相关内容的审核策略，目前提供黄色、性感内容的审核。
安全策略	安全策略包含以下操作： ● 配置安全策略：用于配置直播流安全机制的一组策略，包括在直播过程中使用的推流/播放认证、内容加密（即将开放）、防盗链三种安全保障机制。
统计分析	统计分析包含以下操作： ● 直播域名 ● 流量宽带

	• 用户行为
导播台	依托百度智能云领先的音视频直播、媒体处理和AI技术，云导播台将线下传统导播能力云端化，提供简单好用的直播在线编辑工具，助力用户打造更好的直播体验。您可以根据需求进行开通。
资源包	资源包包含以下操作： • 流量包：展示您购买的流量包消息。 • 转码包：展示您购买的转码包消息。
日志管理	LSS为用户提供了近180天的国内完善的日志管理与下载功能，您可以根据实际情况查询、下载或自动存储LSS的日志文件，便于查询直播记录。
标签管理	您可以根据项目和场景需求，为域名、模板等资源添加标签，便于对它们进行分类和识别管理。

开通LSS

概述

在使用LSS之前，您需要先开通LSS服务，本文详细介绍了快速开通LSS服务的方法。

前提条件

开通LSS服务前，您需要完成 [账号注册](#)、[实名认证](#) 和待接入域名的ICP [备案](#)。

- **实名认证**：每个账户只需认证一次即可。请根据实际情况选择 [企业认证](#) 或者 [个人认证](#)。点击 **立即认证** 完成实名认证。在工作时间内提交认证，认证会在30分钟内完成；非工作时间提交，认证将于次日上午10点前完成。
- **域名备案**：域名备案需要一定的时间（大约15个工作日左右），建议您提前完成相关域名的备案工作。

注意事项

- **个人用户**：限制总流量100GB。如需取消流量限制，请先完成 [企业认证](#) 升级为企业用户。
- **企业用户**：如果企业用户有《网络文化经营许可证》，建议在认证时在 [资质申请页面](#) 提交《网络文化经营许可证》。

开通LSS服务

完成实名认证和备案后，可参考以下操作步骤进行开通LSS服务。

1. 登录 [LSS管理控制台](#)，进入 [开通音视频直播LSS](#) 页面。
2. 您可根据业务需求选择 [按使用流量计费](#) 或 [按日峰值带宽计费](#)，计费详情请参见 [计费概述](#)。

开通音视频直播LSS

产品名称：

音视频直播LSS

分发计费方式：

按使用流量计费

按日峰值带宽计费

分发计费项：

下行流量

计费价格：

开通即送20G流量

点击查看计费文档>>

流量阶梯	价格（元/GB）
0GB-10TB（不含）	0.24
10TB-50TB（不含）	0.23
50TB-100TB（不含）	0.21
100TB-1PB（不含）	0.18
大于1PB	0.15

转码价格：

0.133元/分钟/转码线路

点击查看计费文档>>

关联开通：

使用音视频直播LSS服务时，可能会涉及直播录制，为方便录制视频的点播和存储，帮您同时开通如下服务

（不使用相关服务的情况下不会产生任何费用）：

☒ BOS

☒ VOD

☒ 同意《[百度音视频直播服务LSS使用协议](#)》

立即开通

相关解决方案

视频云解决方案

完整的视频直播、点播解决方案

教育行业解决方案

点播、直播、文档转码、OCR等技术服务

相关产品

对象存储BOS

强大稳定，价格低廉的存储服务

音视频点播

企业级点播平台和应用

3. 勾选 同意《百度音视频直播服务LSS使用协议》，并点击 立即开通。

4. 进入 音视频直播LSS-概览 页面，服务开通成功。

域名管理

添加域名

概述

域名是LSS特有的核心概念，分为推流域名和播放域名，本文将介绍如何使用LSS管理控制台添加域名。

LSS支持两种方式直播，直推和拉流。

1. 直推：主播推流到LSS源站，观众从LSS播放观看直播

2. 拉流：LSS向客户源站或者第三方源站拉流，观众从LSS播放观看直播

前提条件

- 已准备域名，并完成域名备案。
 - 若您需要购买自有域名，可通过 [域名服务](#) 完成域名购买。
 - 若您的域名未完成备案，您可前往百度智能云的 [备案](#) 完成域名备案。

添加域名

1. 登录 [LSS管理控制台](#)。

2. 在左侧导航栏点击**域名管理**，进入域名管理页面。

3. 点击 **添加域名**。

65



4. 在添加域名页完成域名添加，并点击下一步。

添加域名

1 填写基本信息

2 填写高级配置

3 标签(选填)

4 创建完成

基本信息

媒体中心：

广州

北京

苏州

LSS支持最近最优边缘节点接流和分发，其中媒体中心负责流媒体实时处理，包括转码、拉流、录制、截图、水印、通知、审核和鉴权等。

类型：

RTMP推流

RTMP拉流

HTTP-FLV拉流

HTTP-HLS拉流

推流域名：

请输入域名，请确保域名已备案

最长50个字符

播放域名：

☐ RTMP

☐ FLV

☐ HLS

域名描述：

最长128个字符

下一步

取消

配置项	说明
媒体中心	指进行流媒体处理的服务器所在区域，流媒体处理包括转码、拉流、录制、截图、水印、通知、审核和鉴权等。
类型	根据您的业务场景不同，您可以选择不同的域名类型。 - 推流：适用于直推LSS业务场景，可以选择RTMP推流，需要添加推流和播放域名。 - 拉流：适用于LSS拉流场景，可以选择RTMP拉流、HTTP-FLV拉流、HTTP-HLS拉流，需要添加播放域名和源站。
推流域名	当您选择推流时，您需要添加用于进行推流的域名。最多可同时创建 5 个推流域名。
源站	当您选择拉流时，您需要输入您的源站IP或域名。源站分为主源站和备源站，当主源站无法正常工作，LSS会拉备源站。最多可同时创建5个备源站
播放域名	支持RTMP、FLV和HLS三种播放协议。您可以创建多个播放域名，以将不同的播放域名应用于多种业务场景。每种协议可创建 5 个播放域名。 注意：HLS播放域名暂不支持和RTMP/FLV一致。

5. 在高级配置页，可根据您的业务需求配置高级功能。完成配置后，您可直接点击完成，完成域名的添加，也可以点击下一步为域名添加标签。

66

添加域名

1 填写基本信息 > 2 填写高级配置 > 3 标签(选填) > 4 创建完成

高级配置

可选配置项：

☐ 实时转码 ☐ 同步录制 ☐ 水印 ☐ 缩略图 ☒ 安全策略 ☐ 通知

安全策略

安全策略是控制直播会话安全的一组策略，目前支持推流认证、播放认证、防盗链。[查看帮助](#)

安全策略名称：

default

 开启了推流认证

上一步

下一步

完成

取消

高级配置	说明
实时转码	LSS支持视音频实时转码处理，您可以通过转码模板选择适合自己编码标准、分辨率、码率、输出流类型和多码率等流处理参数。可通过“添加线路”配置多个转码模板实现多码率输出，最多可同时配置30路输出。
同步录制	LSS支持直播过程中同步录制，同时支持录制到BOS和录制到VOD，方便您随时重温精彩的直播内容。
水印	LSS支持为直播视频添加图片或时间戳水印，以满足您品牌推广、版权保护等需求。
缩略图	LSS支持在直播过程中实时截取一帧或多帧生成缩略图，以满足您视频预览、特殊帧提取、内容审核及制作直播间封面等需求。
安全策略	LSS提供推流认证、播放认证、内容加密（即将开放）、防盗链等安全机制，为您的视频资源提供全方位保护。
通知	输入通知名称和接口地址，直播状态转换时，系统会主动向您服务器推送消息。
回源Host	当您选择HTTP-HLS直播拉流时，可选是否设置回源Host。回源Host表示CDN节点回源请求头中带有的Host字段，Host字段的值表示的是原始URL给出的服务器或者网关的命名授权。当源站设置解析得到的IP对应的服务器上有多个站点配置了不同的命名Host，那么CDN回源就会根据回源Host字段决定是由哪个站点提供服务。更多信息请参见 回源配置。 注意： 如果源站为IP地址，回源Host默认值与加速域名一致；若源站为域名，回源Host默认值与源站域名一致；您也可手动自定义回源host。

6. （选填）在标签页，为域名添加标签键以及值。更多关于标签的介绍，请参见[标签管理](#)。
- 完成添加后，点击**完成**。
7. 系统将自动跳转至域名列表页面，状态显示“创建中”，耐心等待约一分钟，即可完成域名创建。

常见问题

- 创建直播域名时出现报错 push should not equals to play

错误原因：LSS：push should not equals to play 是由于您的推流域名和播放域名是同一个域名导致，推流域名不可以和播放域名相同，您只需要修改为不同的域名即可。

开启域名

概述

当您想要开启已关闭的域名时，可使用开启域名操作。您可通过开启域名恢复推流和播放能力，本文将为您介绍如何使用开启域名操作。

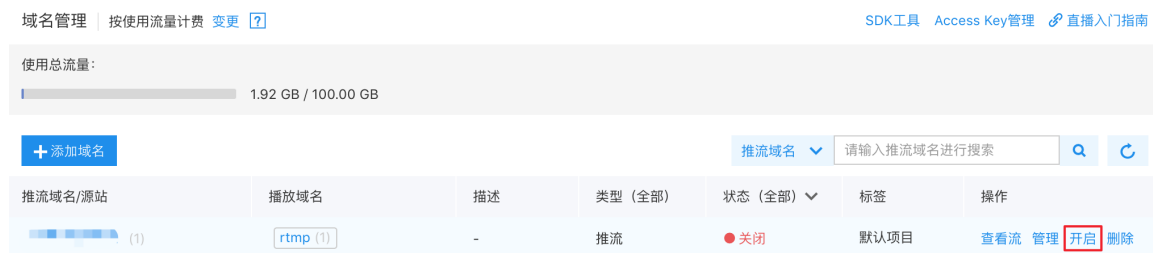
前提条件

待开启的域名处于**关闭**状态。

关闭域名

- 登录 [LSS管理控制台](#)。

2. 在左侧导航栏点击**域名管理**，进入域名管理页面。
3. 选择您想要启用的域名，并在右侧的操作栏点击**开启**。

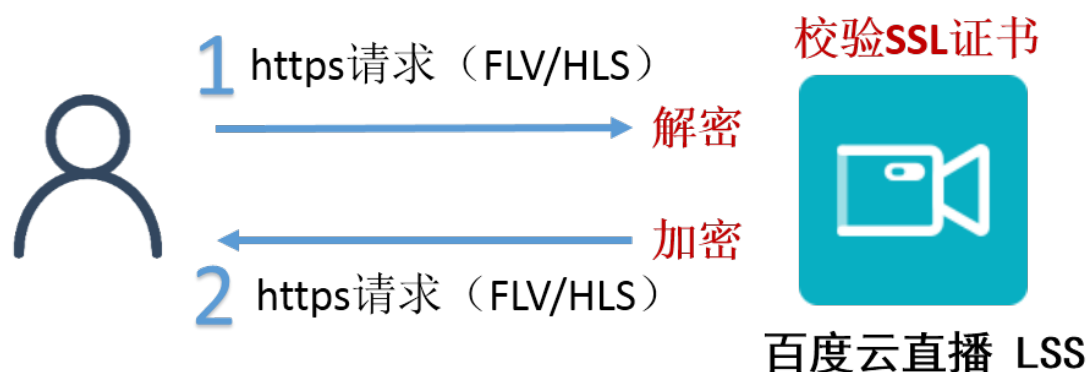


4. 此时该域名状态变为**操作中**状态，当状态变成**启用**时，则域名开启成功。

配置HTTPS

概述

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。直播播放域名配置SSL证书后，收到HTTPS播放请求时（FLV或HLS），LSS将使用客户配置的SSL证书去校验请求的合法性并返回内容。实现流程如下图：



前提条件

配置 HTTPS 的播放域名，您可以上传已具备的证书，请参考[证书管理](#)；也可以使用百度智能云[SSL证书服务](#)（提供免费证书）。

配置HTTPS

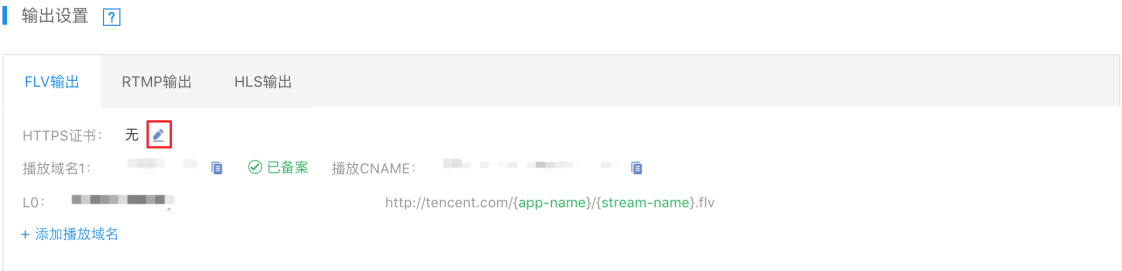
1. 登录 [LSS管理控制台](#)。
2. 在左侧导航栏点击**域名管理**，进入域名管理页面。



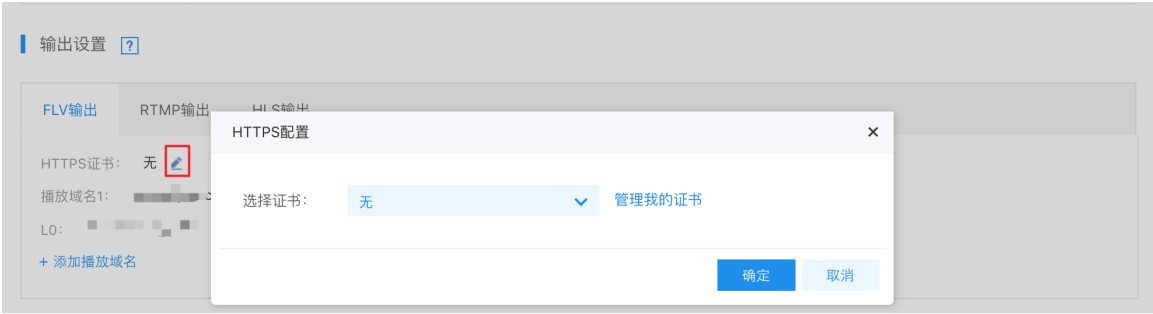
3. 选择包含 FLV 或 HLS 播放域名的域名，点击右侧操作栏的**管理**进入域名详情页。



4. 在FLV输出页签或HLS输出页签点击  。



5. 在弹框中选择您的HTTPS证书。如还未创建证书，您可点击**管理我的证书**进行证书的添加或购买，详情请参见[证书管理](#)。



6. 点击**确认**，完成HTTPS证书的配置。

配置CNAME

概述

CNAME即指别名记录，通过配置CNAME，您可将域名指向另一个域名，再由另一个域名提供IP。

当您使用百度智能云LSS完成域名创建后，您指定的推流域名和播放域名分别生成 **推流CNAME** 和 **播放CNAME**，CNAME域名不能直接访问，您需要在域名服务提供商处完成CNAME配置，配置生效后，即可享受云直播服务。

注意事项

- 推流域名和播放域名均需完成CNAME解析。
- 如果您的域名是在百度智能云域名服务申请，请按照以下步骤操作。
- 如果您在其他云厂商购买的域名，建议您通过 [域名转入](#) 将域名转至百度智能云，再配置CNAME。
- 您也可以直接在域名服务商那里配置CNAME，具体操作请咨询您的域名解析服务提供商。

前提条件

- 已在 [域名服务](#) 申请域名，并备案成功。
- 已在LSS管理控制台的**域名管理**中成功添加域名，且域名CNAME地址状态为  推流CNAME未生效 或  播放CNAME未生效。

配置CNAME

- 音视频直播LSS

概览
域名管理
流管理
质量监控
模版管理
黄反审核
安全策略

产品服务 / 音视频直播LSS- 域名管理

域名管理

按使用流量计费 变更

SDK工具 Access Key管理 直播入门指南

使用总流量:
1.92 GB / 100.00 GB

+ 添加域名

推流域名 请输入推流域名进行搜索

推流域名/源站	播放域名	描述	类型 (全部)	状态 (全部)	标签	操作
 (1)	rtmp flv	-	推流	启用	默认项目	查看流 管理 关闭
 (1)	rtmp flv	-	拉流	启用	默认项目	查看流 管理 关闭

- 域名管理 | 按使用流量计费

变更 ?

SDK工具 Access Key管理 直播入门指南

使用总流量:

1.92 GB / 100.00 GB

+ 添加域名

推流域名 请输入推流域名进行搜索 🔍 ↺

推流域名/源站	播放域名	描述	类型 (全部)	状态 (全部) ▼	标签	操作
(1)	rtmp(1)	-	推流	● 启用	默认项目	查看流 管理 关闭

- 
 启用

 关闭
 

媒体中心： 华南 - 广州

类型： RTMP推流

推流域名1

+ 添加

推流域名: w m 已备案

推流CNAME: 推流CNAME未生效

推流地址: rtmp:// `bm/{app-name}/{stream-name}?{鉴权信息}`

FLV输出

RTMP输出

HLS输出

播放域名1:

☒ 已备案

播放CNAME:

☒ 播放CNAME未生效

L0:

.com/{app-name}/{stream-name}"/>

+ 添加播放域名



7. 选择您需要配置CNAME的域名，点击右侧操作栏下的解析。



8. 在解析页面点击添加解析。



9. 在解析页面完成CNAME类型的配置。

解析域名:

主机记录: play ?

* 记录类型: CNAME记录 ?

* 解析线路: 默认

* 记录值:

* TTL: 10分钟 ?

描述: 最多200个字符
0/200

确定 取消

配置项	说明
主机记录	填写二级域名，即域名前缀。若播放域名为play.baidulive.com，则添加play；若需要直接解析主域名baidulive.com，则输入@；若需要解析泛域名，则输入*。
记录类型	选择CNAME记录。
解析路线	建议选择默认。
记录值	LSS管理控制台域名管理页域名对应的CNAME值，即第四步复制的CNAME值。
TTL	建议填写10分钟。

10. 点击**确定**，完成CNAME配置。

🔗 验证CNAME是否生效

1. 登录 [LSS管理控制台](#)。
2. 左侧导航栏点击域名管理，进入域名管理页面。
3. 选择您需要配置CNAME的域名，点击**域名名称**或右侧操作栏中的**管理**。
4. 在域名详情页查看配置CNAME的域名是否变为 推流CNAME已生效 或 播放CNAME已生效。



配置转推功能

转推是指直播流被推送到CDN节点后，由CDN节点将该流转推至您的源站或者您指定的直播平台。

1. 登录[LSS管理控制台](#)。
2. 点击左侧导航栏“域名管理”，进入域名管理页面。
3. 点击指定的推流域名/源站右侧的“管理”，进入域名详情页。
4. 在输入设置中，点击“转推”右侧的编辑标志，进入配置页面。



5. 点击“开启”，打开转推功能。
默认情况下，转推功能关闭。

编辑转推配置

转推：☒ 开启 ☐ 关闭

ⓘ

开启转推功能将产生转推服务费用，详细定价请参考[转推计费](#)

* 转推域名：

请输入转推域名，最多添加5个

+ 添加域名

确定

取消

6. 添加转推域名，点击确定。 即可开启直播流转推。
- 同一个域名下，转推域名最多可添加五个。

关闭域名

概述

当您想要关闭域名时，可使用关闭域名操作。使用关闭域名操作后，域名将无法推流和播放，本文将为您介绍如何使用关闭域名操作。

注意事项

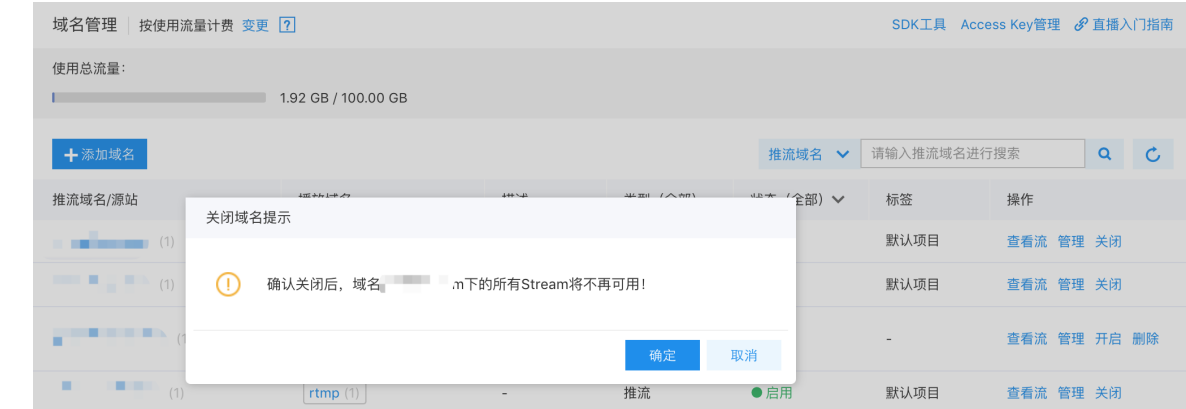
关闭域名后，该域名下的所有流将不可用。

关闭域名

1. 登录 [LSS管理控制台](#)。
2. 在左侧导航栏点击**域名管理**，进入域名管理页面。
3. 选择您想要停用的域名，并在右侧的操作栏点击**关闭**。



4. 在弹出的关闭域名提示框中点击**确定**。



5. 在 **状态** 栏下可见域名当前为 **关闭** 状态。



删除域名

概述

如果您想要删除不需要的域名，可使用删除域名操作。删除域名后，域名将无法找回。本文将为您介绍如何使用删除域名操作。

前提条件

域名需要在关闭状态下才能删除。因此，您需要先关闭域名，再进行删除操作。参见 [关闭域名](#)。

注意事项

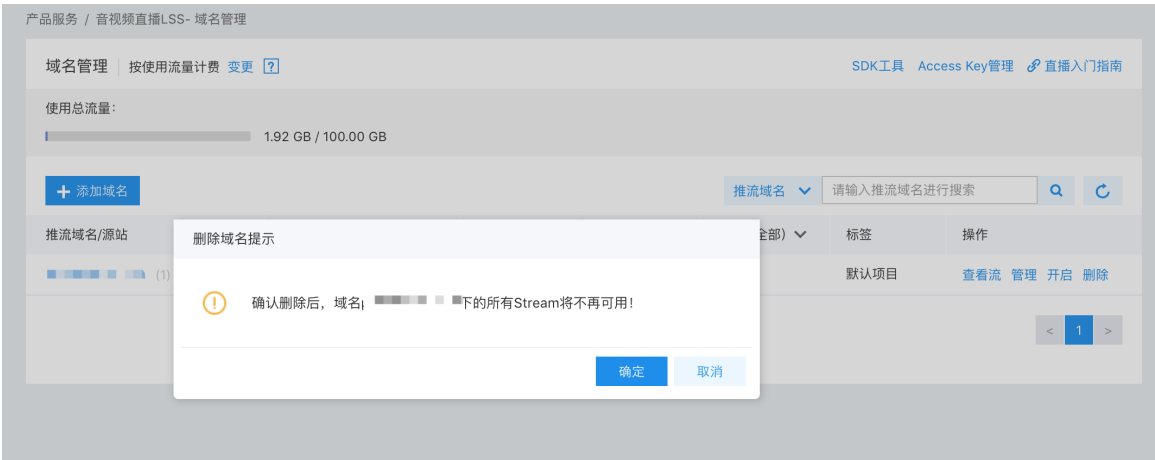
删除域名后，该域名下的所有流将不可用。

关闭域名

- 1. 登录 [LSS管理控制台](#)。
- 2. 在左侧导航栏点击**域名管理**，进入域名管理页面。
- 3. 选择您想要删除的域名，并在右侧的操作栏点击**删除**。



- 4. 在弹出的关闭域名提示框中点击**确定**。



- 5. 此时，该域名已从您的域名列表删除。

时移&回看功能

概述

所谓时移/回看，是指观众可以任意回放过去时间的直播内容。时移/回看是直播与点播相结合而形成的一种新业务，是直播业务的补充。用户在观看直播节目时，中途可以暂停，过后可以从暂停处继续收看，以免错过某些重要情节。

百度智能云 LSS 基于 HLS 直播实现时移/回看功能。HLS 直播时，将 TS 分片持久化到百度智能云对象存储 BOS 上，将音视频保存下来，方便时移/回看播放。

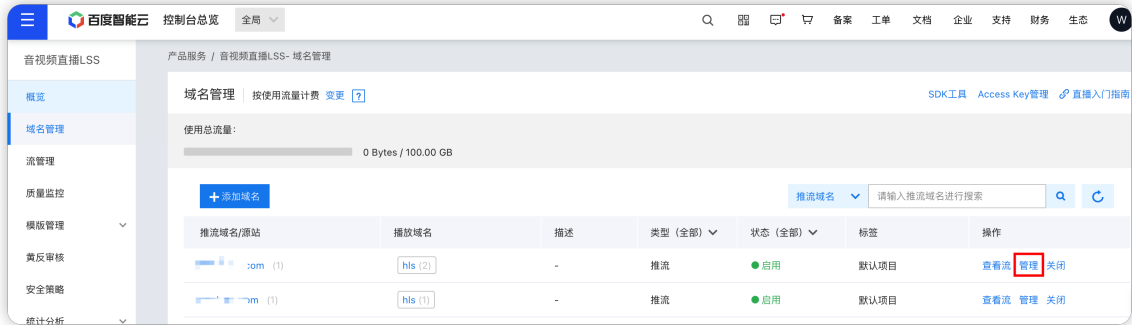
您可以在控制台上自助开通时移/回看功能。

配置时移

- 1. 登录 [LSS管理控制台](#)。
- 2. 在左侧导航栏点击**域名管理**，进入域名管理页面。



- 3. 选择 HLS 播放域名的域名，点击右侧操作栏的管理进入域名详情页。

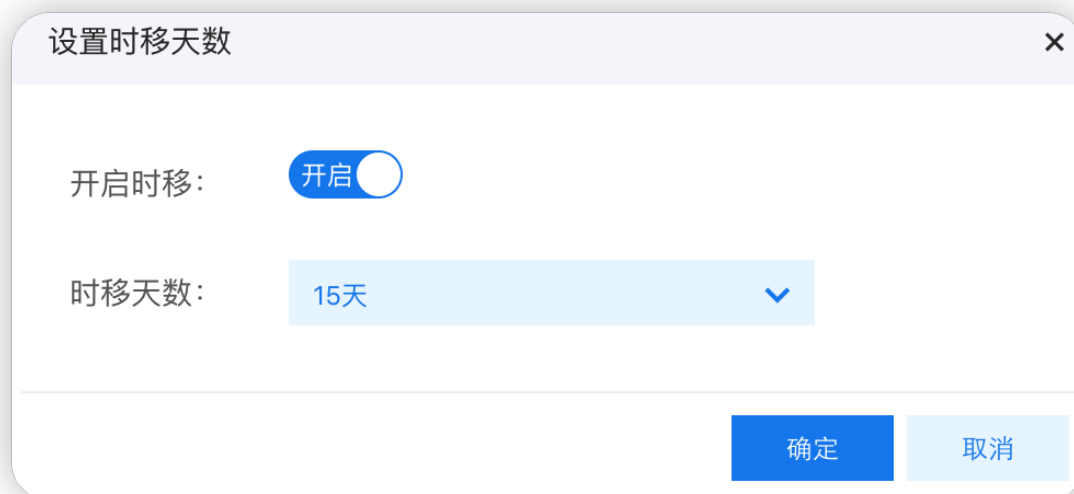


- 4. 在高级设置页签的【时移】行点击 。

注：该功能开启后，对应的所有播放域名的时移/回看功能都生效。



- 5. 在弹框中配置时移/回看相关配置。



6. 点击**确定**，完成时移/回看配置。

🔗 请求格式

常规 HLS 直播地址形如 `http://domain/app/stream.m3u8`，为实现时移/回看播放需要在此直播地址基础上追加相关参数。

时移请求格式

在观看直播内容过程中，把进度条往回拖动到过去的某个时间点开始播放。

时移请求需要带上请求时移内容的时间点 `starttime` 和发起时移请求时的系统时间 `requesttime`，以unix时间戳表示，形如 `http://domain/app/stream.m3u8?starttime=1545978109&requesttime=1545981709`，表示2018-12-28 15:21:49发起时移请求，请求时移内容的时间点从2018-12-28 14:21:49开始。

回看请求格式

请求过去某段时间的直播内容。

回看请求需要带上 `starttime` 和 `endtime` 参数，`starttime` 表示开始时间，`endtime` 表示结束时间，以unix时间戳表示，形如 `http://domain/app/stream.m3u8?starttime=1545978109&endtime=1545981709`，表示请求2018-12-28 14:21:49到2018-12-28 15:21:49这段时间内的回看内容

流管理

新建流

🔗 概述

流 (Stream) 是最基本的直播管理单元。一个域名 (Domain) 下可以包含多个应用 (App)，一个应用 (App) 下可包含多个流 (Stream)。App name 和 Stream name 由用户自定义，当第一次推流时LSS 即会在流管理界面自动创建一个流 (Stream)。

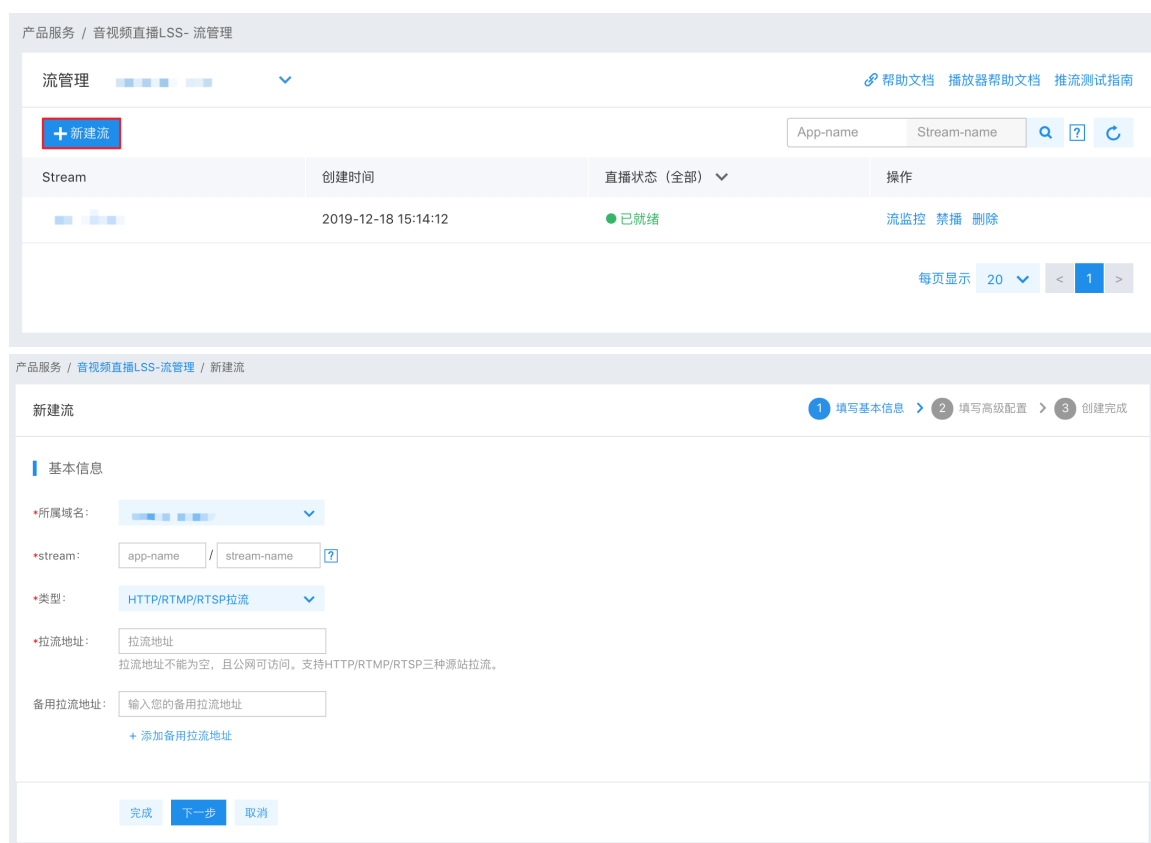
默认一个域名下所有流的配置都跟域名配置保持一致，如果您需要对域名下面的流进行特殊配置，请使用新建流功能添加特殊配置；如果域名下面的流没有特殊配置请直接推流即可正常使用LSS，无需进行新建流操作。

🔗 新建流

1. 登录[音视频直播LSS操作控制台](#)。
2. 在左侧导航栏点击 **流配置管理**，进入**流配置管理**页面。



3. 先打开下拉框选中需要创建流的播放域名，然后点击**新建流**，进入新建流页面。



4. 选择流（Stream）所属的**域名**，自定义**App**和**Stream**名称，拼装推流地址。

5. 选择**推流**或**拉流**类型。

- 选择RTMP推流，系统自动会将App和Stream拼接成推流地址。
- 选择HTTP/RTMP/RTSP拉流，由用户输入拉流地址。为了避免拉流地址错误导致拉流失败，您可以添加备用拉流地址。当主拉流地址失效后，优先切换到第一个备用拉流地址。最多可添加10个备用拉流地址，拉流失败时按顺序切换地址。

6. 点击**下一步**，自定义高级配置。流会自动继承其所属域名的高级配置，但您也可以根据需要对流进行自由配置。

可配置项	描述
实时转码	转码模板是一组音视频流处理参数的集合，可通过配置多个转码模板实现多码率输出，最多可同时配置30路输出。
同步录制	开启同步录制功能，可将直播过程中的音视频流录制存储，通过录制模板可配置详细录制参数。
水印	直播中添加水印，最多可同时支持4个图片水印和1个时间戳水印。 温馨提示： 使用forward_only模板的原始流不会添加水印！
缩略图	通过缩略图模板可配置缩略图图片格式、大小、尺寸伸缩策略和截图间隔等参数。
安全策略	安全策略是控制直播会话安全的一组策略，目前支持推流认证、播放认证、防盗链。
通知	配置会话通知后，当直播会话状态发生变化或者直播过程中完成某个指定任务时，系统将通过HTTP请求将该通知消息POST到您所配置的接口地址。

7. 点击**创建完成**，页面提示“您的直播流创建成功，即将跳转至流管理列表”。在流管理列表，可查看新创建的直播流，点击stream 名称，获取推流地址和播放地址。

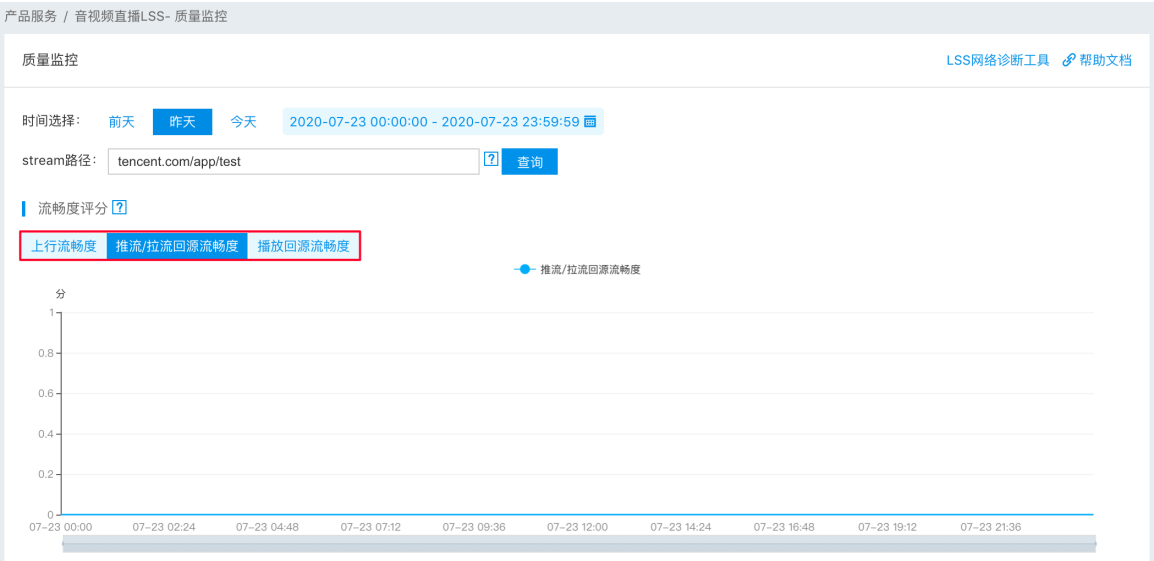
查询流信息

- 1. 登录[音视频直播LSS操作控制台](#)。
- 2. 在左侧导航栏点击 **流管理**，进入“流管理”页面。
- 3. 在**流管理**列表页面，您可以查看已经成功推流的流（Stream）和当前的直播状态，以及**流监控**、**禁播**和**删除**流操作。



• 流监控

点击**流监控**，进入质量监控页面，可以查看查看监测数据。



• 禁播

点击**禁播**，直播状态变为已禁播，此时若想恢复直播点击操作列的**恢复**即可。

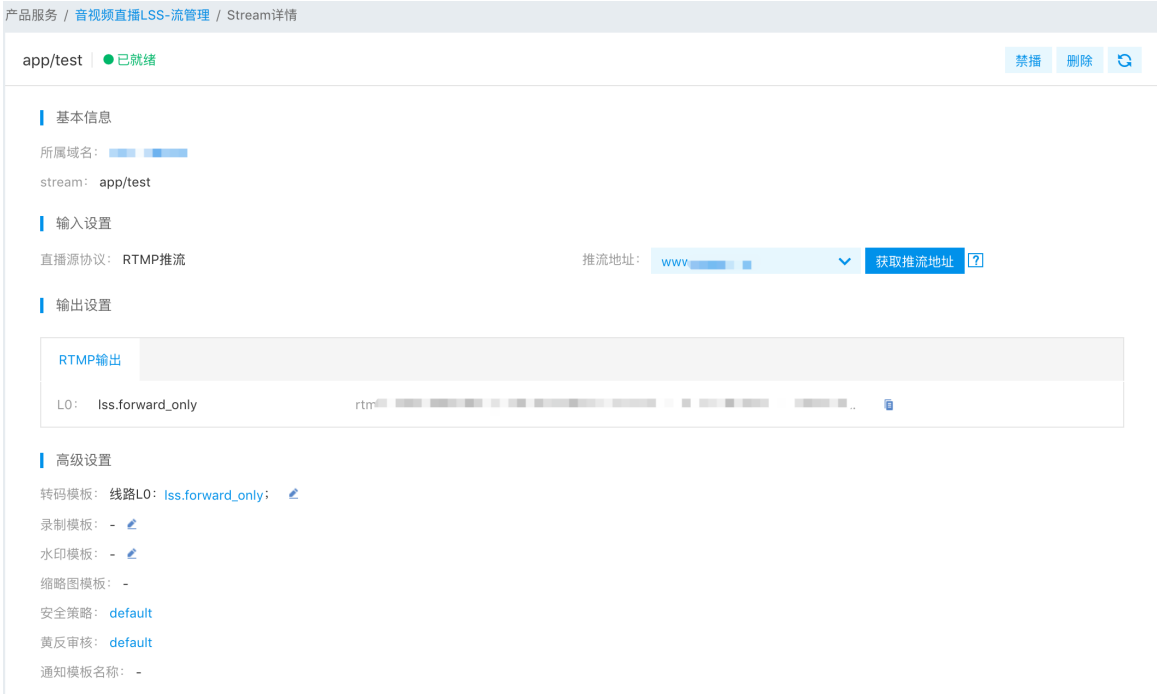


● 删除

点击**删除**，页面提示删除成功，**直播流立刻删除**，请谨慎操作。

● 查看详情

点击**流名称**查看详情，包括基本信息、输入输出设置（直播源协议、推流地址）及高级设置（转码、录制、水印、缩略图、安全策略）。



质量监控

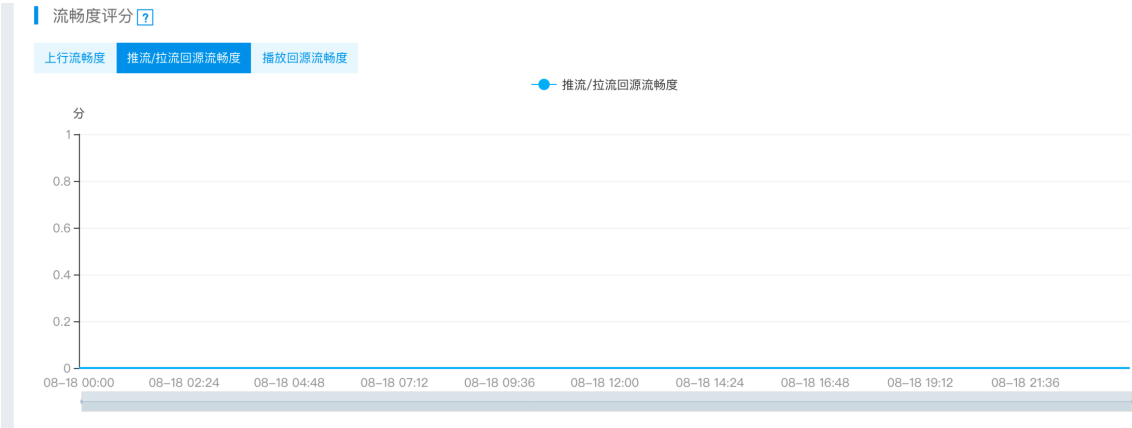
🔗 概述

直播过程中，卡顿现象会直接影响到直播效果，影响原因可能是上行传输流畅度不佳，网络限速等。为了便于用户排查到问题出现在哪一环节，LSS 提供了质量监控功能，能够快速定位卡顿原因。

🔗 查看监测数据

您可以查看某路流在近7天内任意时间段的质量监控信息。

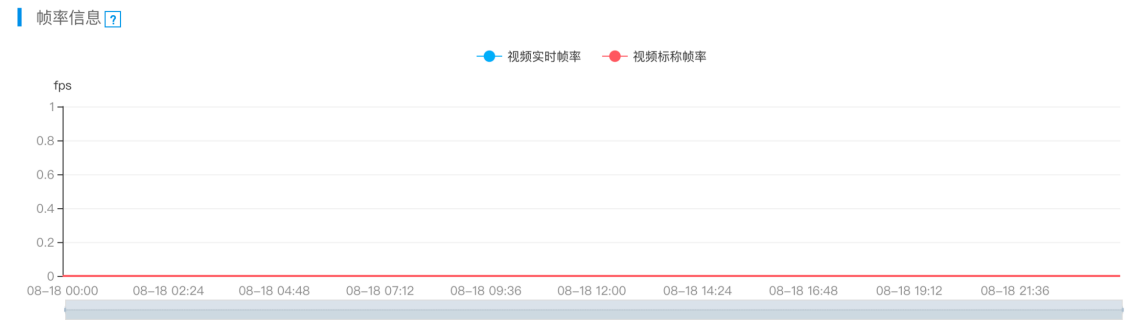
1. 登录[音视频直播LSS操作控制台](#)。
2. 在左侧导航栏点击 **质量监控**，进入质量监控页面。
3. 查询某路直播流的质量信息：单击输入框依次选择 **Domain>App>Stream**，针对多码率输出的情况，Stream 可精确到具体线路，例如：Domain>App>Stream-L1。

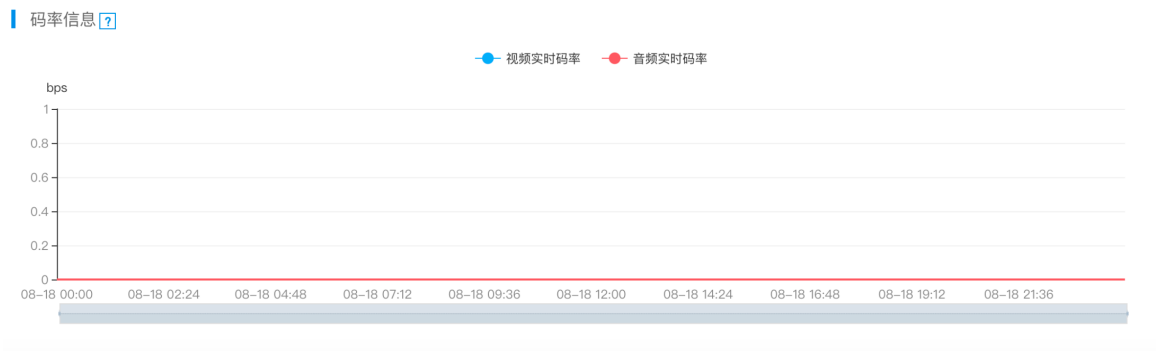


4. 选择时间跨度（近7天内的任意时间段），需要监测的环节，包括上行流畅度、推流/拉流回源流畅度和播放回源流畅度情况，以及源站帧率/码率信息。



数据监测环节	说明	流畅度评分（每隔10秒采集）
上行流畅度	从主播端推流至cdn的流畅度。	- 流畅度评分=实时视频帧率/标称视频帧率x100 - 当数据显示状态为高评分，并且曲线平滑状态时，表示视频流回源传输较稳定，若出现较大的抖动，表明cdn接流节点到LSS源站不稳定，可能是主播上行流畅度差所致，也可能是LSS CDN节点到LSS源站链路问题，可申请工单排查。
推流/拉流回源流畅度	管理控制台提供了对CDN节点到LSS源站监控的功能，可以及时查看直播推流回源状态。目前，仅支持查询推流状态回源流畅度数据。	
播放回源流畅度	不同线路实时播放视频的帧率和码率信息。	
帧率	LSS实时接收到的音视频帧率信息。	帧率(fps)=1分钟媒体帧个数(frame) /60(秒)
码率	LSS实时接收到的音视频码率信息。	码率(kbps)=1分钟内的文件大小\ (MB) /60(秒)





模板管理

转码模板

概述

您可以通过对域名绑定转码模板实现直播转码功能，转码模板可以配置音视频编码标准、码率、输出流类型等基本参数，包括两类：

- 样例模板：系统内置，方便您直接使用从而简化域名配置。
- 用户模板：自定义模板，根据需求量身定制从而更好地适配用户业务场景。

注意事项

- 水印会转码，所以水印会产生转码费用，具体参考[转码计费](#)。

转码模板

1. 登录[音视频直播LSS操作控制台](#)。
2. 在左侧导航栏点击 **模板管理** -> **转码**，进入转码模板页面。

音视频直播LSS

产品服务 / 音视频直播LSS- 模板管理-转码模板

转码模板 [转码模板帮助>>](#)

[+ 新建转码模板](#) 模板搜索: [C](#)

模板名称	模板描述	转码模式	编码标准	模板类型	标签	操作
lss.lss_1280x720_cae	智能超清720p,16:9	智能超清	H.264	系统内置	-	详情 添加域名
lss.lss_320x240_cae	智能超清240p,4:3	智能超清	H.264	系统内置	-	详情 添加域名
lss.lss_480x360_cae	智能超清360p,4:3	智能超清	H.264	系统内置	-	详情 添加域名
lss.lss_640x360_cae	智能超清360p,16:9	智能超清	H.264	系统内置	-	详情 添加域名
lss.lss_640x480_cae	智能超清480p,4:3	智能超清	H.264	系统内置	-	详情 添加域名
lss.lss_854x480_cae	智能超清480p,16:9	智能超清	H.264	系统内置	-	详情 添加域名
lss.lss_960x720_cae	智能超清720p,4:3	智能超清	H.264	系统内置	-	详情 添加域名

3. 点击[新建转码模板](#)按钮，进入“创建转码模板”页面，基于[样例模板](#)或[空模板](#)创建用户模板。

产品服务 / 音视频直播LSS- 模板管理-转码模板

转码模板 [转码模板帮助>>](#)

[+ 新建转码模板](#) 模板搜索: [C](#)

模板名称	模板描述	转码模式	编码标准	模板类型	标签	操作
lss.lss_1280x720_cae	智能超清720p,16:9	智能超清	H.264	系统内置	-	详情 添加域名
lss.lss_320x240_cae	智能超清240p,4:3	智能超清	H.264	系统内置	-	详情 添加域名
lss.lss_480x360_cae	智能超清360p,4:3	智能超清	H.264	系统内置	-	详情 添加域名

目前LSS提供下述样例模板：

preset	description
lss.lss_1280x720_cae	智感超清转码，高清720p，16:9，RTMP/FLV/HLS输出
lss.lss_960x720_cae	智感超清转码，高清720p，4:3，RTMP/FLV/HLS输出
lss.lss_854x480_cae	智感超清转码，标清480p，16:9，RTMP/FLV/HLS输出
lss.lss_640x480_cae	智感超清转码，标清480p，4:3，RTMP/FLV/HLS输出
lss.lss_640x360_cae	智感超清转码，标清360p，16:9，RTMP/FLV/HLS输出
lss.lss_480x360_cae	智感超清转码，标清360p，4:3，RTMP/FLV/HLS输出
lss.lss_320x240_cae	智感超清转码，低清240p，4:3，RTMP/FLV/HLS输出
lss.lss_1280x720	高清720p，16:9，RTMP/FLV/HLS输出
lss.lss_960x720	高清720p，4:3，RTMP/FLV/HLS输出
lss.lss_854x480	标清480p，16:9，RTMP/FLV/HLS输出
lss.lss_640x480	标清480p，4:3，RTMP/FLV/HLS输出
lss.lss_640x360	标清360p，16:9，RTMP/FLV/HLS输出
lss.lss_480x360	标清360p，4:3，RTMP/FLV/HLS输出
lss.lss_320x240	低清240p，4:3，RTMP/FLV/HLS输出
lss.all_as_input	保持输入分辨率与码率不变，RTMP/FLV/HLS输出

说明：

- 1. 上述模板中，除了forward_only只转发不做转码，使用其他模板均收取转码费用。
- 2. forward_only模板对应的stream不支持加水印，如需添加水印，建议使用all_as_input模板，该模板会保持原有的输入参数再次做转码并添加水印。
- 3. 模板名称中带_cae后缀的为智感超清转码模板，码率压缩更大，画质更清晰。您可以前往[智感超清解决方案详情页](#)进行开通。

4. 填写模板**基本信息**和**高级配置项**后点击**完成并创建**。

基本信息参数详情：

分类	参数名称	描述	可选值	必要性
基本信息	样例模板	为用户提供参考模板，包括内置转码模板和空模板。	内置转码模板、已创建的用户模板、空模板	必填
	模板名称	命名自定义模板，命名规则见控制台，用户拥有的模板之间不能重名。	-	必填
	模板描述	自定义模板的类型描述，包括清晰度、分辨率等	-	可选
	直播类型	包括两种类型的转码模板，仅用于转发的模板和实时转码的模板。	仅转发、实时转码	必填
	绑定标签	标签支持您按各种标准（如用途、所有者或项目）对资源进行分类，每个标签包含键和值两部分。	可以通过下拉菜单选择已创建的标签键值对，也可以直接输入内容来创建新的标签。	可选

转码模板的高级配置项参见下表，其中视频和音频项至少配置一种。

分类	参数名称	描述	可选值	必要性
视频	编码标准	视频编解码	h264	可选
	编码规格	指定目标视频所用编码规格	baseline, main, high	可选
	码率	视频目标码率（单位 bps）	-	可选
	最大帧率	目标视频最大帧率	10, 15, 20, 23.97, 24, 25, 29.97, 30, 50, 60	可选
	最大宽度	目标视频的最大宽度	128 ~ 4096，必须为2的倍数	可选
	最大高度	目标视频的最大高度	96 ~ 3072，必须为2的倍数	可选
	尺寸伸缩策略	视频图像的尺寸伸缩策略	keep，保持原始视频尺寸（默认值）； shrinkToFit，原始视频尺寸自适应； stretch，表示拉伸原始视频尺寸。	可选
音频	编码标准	音频编码方式	目前音频编码标准只支持AAC	可选
	采样率	音频采样率	22050, 32000, 44100, 48000, 96000，(auto)与输入保持一致	可选
	码率	音频目标码率	大于等于1000	可选
	声道	音频声道数目	单声道、双声道、与输入一致	可选

5. 创建模板完成后，可在转码模板列表查看。

录制模板

概述

LSS提供直播视频录制功能，通过在创建录制模板时指定视频存储路径，并在域名配置中绑定适当的录制模板，即可将直播内容同步录制到[对象存储BOS](#)或[音视频点播VOD](#)。

录制策略如下：

- 主播端连接不断开且网络稳定，则录制为一个视频文件。
- 主播端断开，或者网络不稳定均有可能导致录制多个视频文件，具体策略如下：
 - 主播端断开，每断开一次就录制一个视频文件；
 - 主播端网络不稳定，无流时长在30秒内不会触发新的录制操作，超过30秒才会重新录制一个视频文件。

🔗 注意事项

- 录制到VOD点播时，仅支持 MP4 格式;录制到BOS 时，支持MP4、M3U8、FLV格式。
- 使用BOS Bucket存储时，用户需要对录制模板中使用的Bucket授予LSS服务账号WRITE权限（目前在用户创建录制模板时，LSS服务会自动添加该授权）。
- 请不要删除录制文件的Bucket和WRITE权限，否则在直播过程中无法正常录制视频流。



🔗 录制模板

1. 登录[音视频直播LSS操作控制台](#)。
2. 在左侧导航栏点击 **模板管理 -> 录制**，进入录制模板页面。



3. 点击[新建录制模板](#)按钮，进入“创建录制模板”页面，基于[样例模板](#)或[空模板](#)创建用户模板。



4. 填写模板基本信息和配置项后点击**完成并创建**。

参数名称	描述	必要性
模板名称	录制模板的名称	必填
模板描述	当前模板的相关信息	可选
存储模式	同步录制到对象存储BOS或音视频点播VOD	必选
录制模式	与输出流一致	必选
所属地域	北京、广州、苏州	必选
选择BOS Bucket	选择录制文件保存的BOS Bucket路径	必选
录制视频格式	MP4、M3U8、FLV	必选
单个录制文件时长	设置录制的视频文件时长，输入1~360分钟	必选
自动合并时间间隔	当选择录制到BOS（包括标准及低频）并选择录制视频格式为m3u8时，支持自定义自动合并视频时间间隔	必选
录制文件命名模式	系统默认pattern=%d%a/%s/recording_%.f； - 其中，%d: domain %a: App名称 %s: stream名称 %t: 录制开始时间，CST时区，精确到秒 %T: 录制结束时间，CST时区，精确到秒 %f: 录制文件格式 - 例如，生成的录制文件名形如“domain/appName/streamName/recording_20160213120931.mp4”	可选

5. 创建模板完成后，可在录制模板列表查看，在操作列可以对模板查看详情，添加域名，编辑或删除。



🔗 相关 API

- [查询指定录制模板](#)
- [查询录制模板列表](#)

水印模板

🔗 概述

在直播流中，可以通过水印模板（支持图片水印和时间戳水印）对输出视频添加水印，水印模板中定义水印的详细参数集合用于完成水印添加，包括水印形态（图片、时间戳）、水印大小、位置等。

🔗 注意事项

水印仅支持在转码线路里播放。

🔗 水印模板

- 1. 登录[音视频直播LSS操作控制台](#)。
- 2. 在左侧导航栏点击 **模板管理 -> 水印**，进入水印模板页面。



- 3. 点击 **点击新建水印模板** 按钮，进入“创建水印模板”页面。



- 4. 填写模板基本信息和配置项后点击**完成并创建**。

根据您的需求，您可选择**图片水印**或**时间戳水印**。

• 图片水印

参数	描述	是否必须
模板名称	字符串，可由小写字母、数字、下划线组成，必须以小写字母开头，最长40个字符	是
类型	图片水印	是
图片	支持PNG、JPG、GIF三种图片格式，推荐使用PNG和GIF透明背景的图标。有效值：大小<500kb，尺寸< 1280*720像素	是
最大宽度	有效值：[10-4096]间的偶数。不设置时，如果设置了最大高度则会根据原始图片宽高比计算水印宽度，否则和原始图片宽度保持一致	否
最大高度	有效值：[10-3072]间的偶数。不设置时，如果设置了最大宽度则会根据原始图片宽高比计算水印高度，否则和原始图片高度保持一致	否
伸缩策略	目前仅支持：keep（与原图片宽高比例一致）	否
显示位置	九宫格形式，分左上、中上、右上、左中、居中、右中、左下、中下、右下	否
水平偏移	显示位置 有水平偏移时有效。单位：像素，有效值：[0-4096]，默认值：10	否
垂直偏移	显示位置 有垂直偏移时有效。单位：像素，有效值：[0-3072]，默认值：10	否

模板名称:

请输入模板名称

可由小写字母、数字、下划线组成，必须以小写字母开头，最长40个字符。

类型:

图片水印

图片水印支持PNG、JPG、GIF三种图片格式，推荐使用PNG和GIF透明背景的图标。要求：大小< 500kb，尺寸< 1280*720像素。

选择图片

最大宽度:

最大高度:

伸缩策略:

Keep（与原图片宽高比例一致）

显示位置:

左上

中上

右上

左中

居中

右中

左下

中下

右下

垂直偏移:

10

px

水平偏移:

10

px

效果预览(可能与真实效果存有差异)

关于水印在视频画面上的定位策略，请参考[帮助](#)。

时间戳水印

参数	描述	是否必须
模板名称	字符串，可由小写字母、数字、下划线组成，必须以小写字母开头，最长40个字符	是
类型	时间戳水印	是
时区	支持CST（中国标准时间）、UTC（世界统一时间）两种类型	是
透明度	有效值：[0-100]%，0表示完全透明，100%表示完全不透明	否
文字样式	设置字体、大小、颜色。字体目前仅支持Arial；大小有效值：[1-72]，默认值：16；颜色默认值：白色	否
背景颜色	默认值：无	否
显示位置	九宫格形式，分左上、中上、右上、左中、居中、右中、左下、中下、右下	否
水平偏移	显示位置有水平偏移时有效。单位：像素，有效值：[0-4096]，默认值：10	否
垂直偏移	显示位置有垂直偏移时有效。单位：像素，有效值：[0-3072]，默认值：10	否

模板名称:

请输入模板名称

可由小写字母、数字、下划线组成，必须以小写字母开头，最长40个字符。

类型:

时间戳水印

时区:

CST（中国标准时间）

透明度:

0%（透明）

50%

（不透明）100%

100%

文字样式:

Arial

16

背景颜色:

无

显示位置:

左上

中上

右上

左中

居中

右中

左下

中下

右下

垂直偏移:

10

px

水平偏移:

10

px

绑定标签:

温馨提示：标签支持您按各种标准（如用途、所有者或项目）对资源进行分类；每个标签包含键和值两部分

标签键:

默认项目

值:

请选择已有或手动输入

×

2020-08-17 16:34:56

效果预览(可能与真实效果存有差异)

关于水印在视频画面上的定位策略，请参考[帮助](#)。

5. 创建模板完成后，可在水印模板列表查看，在模板右上角可以查看或删除。

水印模板

[水印模板帮助>>](#)

1. 支持图片水印和时间戳水印。
2. 图片水印支持PNG、JPG、GIF三种图片格式，推荐使用PNG和GIF透明背景的图标，要求大小 < 500kb，尺寸 < 1280*720像素。



相关 API

- [水印模板接口](#)
- [创建时间戳水印](#)

缩略图模板

概述

在直播流中，可以通过缩略图模板为输出视频生成缩略图，缩略图模板中定义缩略图的详细参数集合用，包括图片格式、伸缩策略及尺寸大小、BOS Bucket及所属区域等。

缩略图模板

1. 登录[音视频直播LSS操作控制台](#)。
2. 在左侧导航栏点击 **模板管理** ->**缩略图**，进入缩略图模板页面。



3. 点击**新建缩略图模板**按钮，进入“创建缩略图模板”页面，



4. 填写模板基本信息和配置项后点击**完成并创建**。

参数	描述	是否必须
模板名称	字符串，可由小写字母、数字、下划线组成，必须以小写字母开头，最长40个字符	是
模板描述	模板说明信息	否
图片格式	目前仅支持jpg、gif	是
尺寸伸缩策略	目前仅支持(Strech)拉伸原视频尺寸	是
所属地域	支持北京、广州、苏州	是
选择BOS Bucket	用于存储截到的缩略图的Bucket。指定目标BOS Bucket后，系统会授权LSS服务访问此Bucket	是
缩略图宽度	[10-4096]间的偶数，输入值超过视频实际宽度时按输入值输出	否
缩略图高度	[10-3072]间的偶数，输入值超过视频实际高度时按输入值输出	否
截图间隔	[1-21600]，单位：秒	否
文件命名模式	系统默认pattern=%d%a/%s/thumbnails/%t.%f；每个参数仅设置一次 - 其中，%d：domain %a: App名称 %s: Stream流名称 %t: 缩略图截图时间，CST时区，精确到秒；建议设置%t，否则同一个流的每个截图会覆盖前一张，最终该流中只保存了最后一张截图 %f: 缩略图文件格式 - 例如，生成的缩略图文件名形如'domain/appName/streamName/thumbnails/20160213120931.jpg'	否

5. 创建模板完成后，可在缩略图模板列表查看，在操作列可以对模板查看详情，添加域名，编辑或删除。

产品服务 / 音视频直播LSS- 模板管理-缩略图模板

缩略图模板

[缩略图模板帮助>>](#)

+ 新建缩略图模板

模板搜索:

模板名称	模板描述	缩略图存储路径	标签	操作
thumbnail_template_for...		bos://ainput	-	详情 添加域名 编辑 删除

- 相关 API
- [查询缩略图模板](#)
 - [缩略图列表](#)

通知模板

概述

您可以通过配置通知模板来实现对直播流的自动监控。在直播状态转变时，或者特定事件发生时，配置了通知模板的直播流会向通知模板的接口地址发送通知。

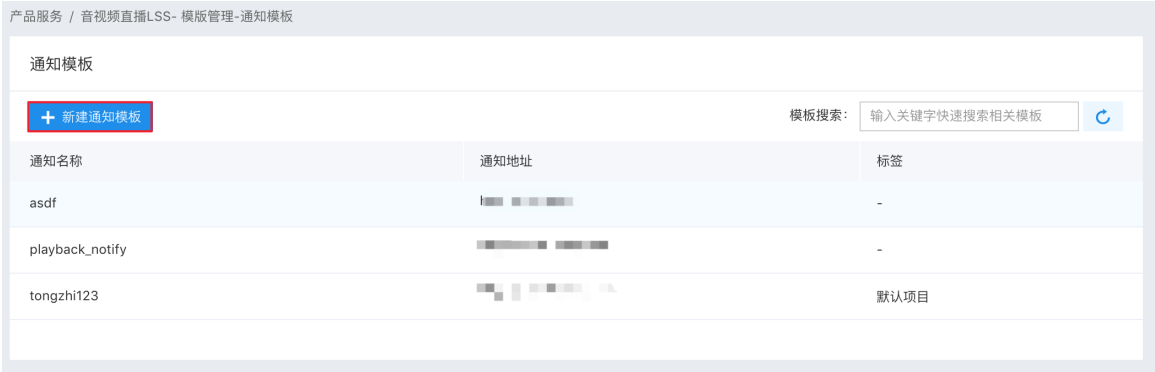
通知模板

1. 登录[音视频直播LSS操作控制台](#)。

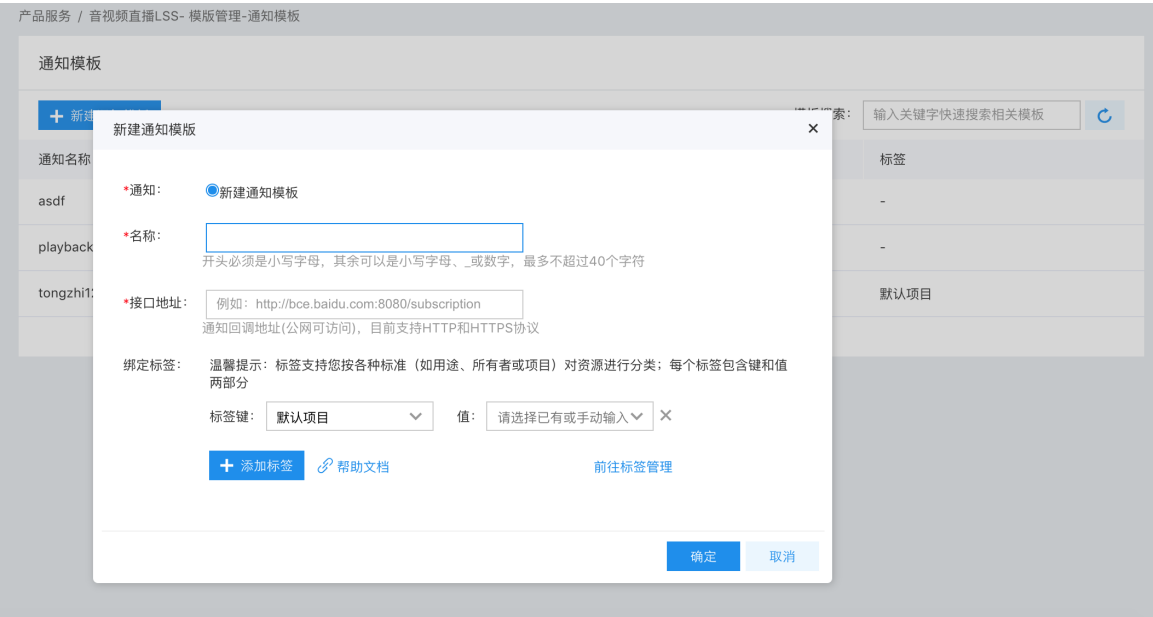
2. 在左侧导航栏点击 **模板管理** ->**通知**，进入通知模板页面。



3. 点击**新建通知模板**按钮，弹出“新建通知模板”弹窗。



4. 在 **新建通知模板** 弹框中完成模板配置。



参数	描述
名称	通知模板的名称， 开头必须是小写字母，其余可以是小写字母、_或数字，最多不超过40个字符。
接口地址	通知的回调地址，需要是公网可以访问的地址，目前支持HTTP和HTTPS协议。
标签	（可选）标签支持您按各种标准（如用途、所有者或项目）对资源进行分类；每个标签包含键和值两部分。

5. 点击**确定**，创建完成后可在通知模板列表查看已创建的模板。

通知模板默认可以在直播流开启/停止时向接口地址发送通知。您也可以在[黄反审核模板](#)处指定通知模板。

黄反审核

LSS直播提供截图审核功能，支持通过控制台配置的截图审核模板，在主播推流过程中截取直播画面进行内容审核，并将直播截图或鉴黄数据存储至百度云对象存储中。LSS直播推流默认关闭截图审核功能，本文将向您介绍如何对指定推流域名配置截图审核功能。

🔗 注意事项

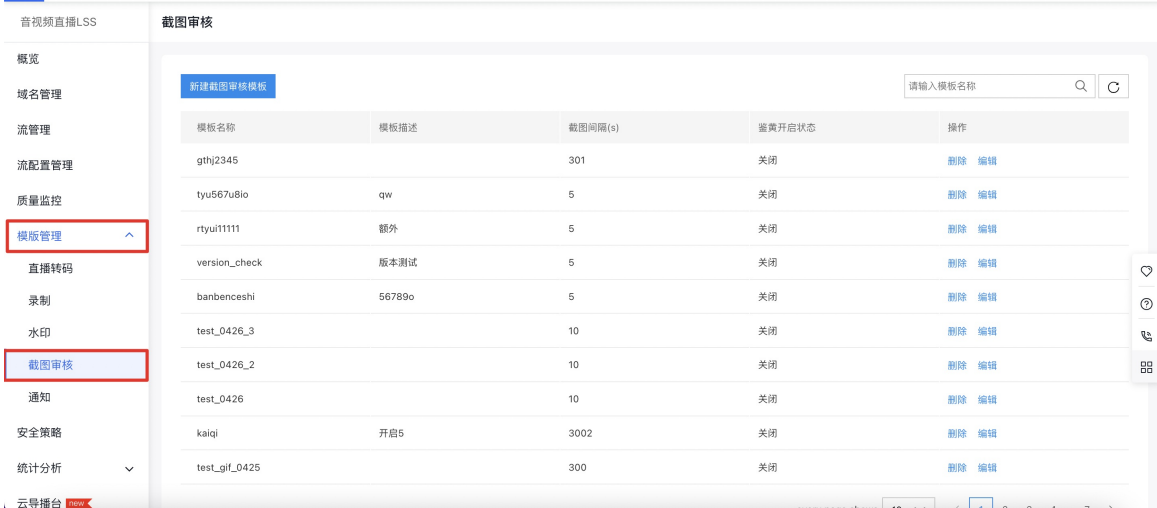
- 截图功能可单独开启使用，但鉴黄功能需开启截图后才能开启，不可单独使用。
- 截图与鉴黄为收费功能，启用后，截图功能按每千张 ¥ 0.1元收费；鉴黄功能按每千张 ¥ 1.1元收费。详情请参见 [音视频直播LSS价格说明](#)。
- 截图与鉴黄图片存储至您的对象存储BOS中，将产生BOS存储费用。详情请参见：

🔗 前提条件

- 已开通百度云LSS直播服务，并[添加推流域名](#)。
- 已[创建BOS Bucket对象存储服务](#)。

🔗 创建截图审核模版

1. 登录[音视频直播LSS操作控制台](#)。
2. 在左侧导航栏点击**模板管理**-**截图审核**。



3. 点击新建截图审核模板，可编辑配置项，并单击确认即可

* 模板名称：

Please input

可由小写字母、数字和下划线组成，必须以小写字母开头，最长40个字符。

模板描述：

0/128

* 图片格式：

jpg

* 截图间隔：

5

秒

取值范围1-3600，且必须为整数

* 图片分辨率：

保持原视频尺寸比率

文件命名模式：

Please input

智能鉴黄：

OFF

所属区域：

北京

广州

苏州

* BOS Bucket：

请选择

绑定标签

温馨提示：标签支持您按各种标准（如用途、所有者或项目）对资源进行分类；
每个标签包含键和值两部分；

标签键：

请选择已有或手动输入

值：

请选择已有或手动输入

+ 添加标签

[帮助文档](#)

[前往标签管理](#)

确认

取消

- 您可以调整截图间隔、关联截图审核模版、绑定[通知接口](#)。设置通知后，系统会在检测到黄反内容后向指定的回调地址推动通知信息。
- 点击域名管理-管理-高级设置即可关联截图审核模版

 修改模版

1. 在左侧导航栏点击**模板管理-截图审核**。
2. 选择您已创建成功的截图鉴黄模板，并单击右侧的 **编辑**，即可进入修改模板信息。

3. 单击**确认**即可。

🔗 删除模版

1. 在左侧导航栏点击**模板管理-截图审核**。
2. 选择您已创建成功的截图鉴黄模板，单击右侧的删除按钮。
3. 确认是否删除当前截图鉴黄配置模板，单击**OK**即可成功删除。

注：当模板为空闲模板时才可以删除，已关联模板不可删除。已关联模版需先解除关联才可以删除。

安全策略

安全策略是用于配置直播流安全机制的一组策略，包括在直播过程中使用的推流/播放认证、内容加密（即将开放）、防盗链三种安全保障机制。

当前系统默认不开启安全策略，需要您手动开启安全策略。

🔗 推流认证

为确保推流地址不被非法占用，LSS提供token认证和有效期限相结合的推流地址。开启推流认证后，您可以通过”推流地址+超时时间+密钥”的方式获取完整的加密推流地址。

1. 登录 [LSS管理控制台](#)。
2. 在左侧导航栏点击**安全策略**，进入安全策略页面。
3. 点击**新建安全策略**。



4. 在新建安全策略页将**推流认证**按钮打开，并填写安全策略名称。

新建安全策略

[帮助文档](#)

*名称:

可由小写字母、数字和下划线组成，必须以小写字母开头，最长40个字符。

*推流认证:

ON

开启推流认证后，您可以设置推流地址的有效期，过了有效期该地址将失效，再次推流时需重新获取推流地址。

*播放认证:

ON

开启播放认证后，播放地址附带token且仅在用户设定的有效期内可访问。

密钥:

保存安全策略后，服务器会生成密钥。

绑定标签:

温馨提示：标签支持您按各种标准（如用途、所有者或项目）对资源进行分类；每个标签包含键和值两部分

标签键:

默认项目

值:

请选择已有或手动输入

×

+ 添加标签

[帮助文档](#)

[前往标签管理](#)

防盗链

Referer

IP

☒ 允许Referer为空

类型:

☒ 白名单生效

☐ 黑名单生效

黑名单与白名单同一时间只能生效一种

+ 添加refer白名单

删除

☐ 全选

没有数据

确定

取消

5. 点击 **确定** 完成创建。
6. 创建完成后，您可以在安全策略管理页找到创建的安全策略点击“编辑”，即可查看密钥key的值：

编辑安全策略

*名称:

demo

*推流认证:

ON

开启推流认证后，您可以设置推流地址的有效期，过了有效期该地址将失效，再次推流时需重新获取推流地址。

*播放认证:

ON

开启播放认证后，播放地址附带token且仅在用户设定的有效期内可访问。

密钥:

显示

7. 在域名或者单个流的详情页面，均可配置安全策略。

高级设置

以下模板更新后在下一次推流时生效，对当前处于直播中有输入流的Stream暂不起作用。

转码模板：线路L0：lss.forward_only；

录制模板：-

水印模板：-

缩略图模板：-

安全策略：default

黄反审核：default

通知接口：-

假设您对默认安全组进行了更新，同时打开了推流认证，并从控制台获取到密钥key，例如值为asdfghjkl。对于推流地址rtmp://<push.your-domain.com>/<your-app>/<your-stream>，token计算方式及合法推流地址如下所示：

• token计算方式：

```
sha256_hex('rtmp://<push.your-domain.com>/<your-app>/<your-stream>;2017-08-15T01:00:00Z','asdfghjkl')
```

其中2017-08-15T01:00:00Z为token的超时时间，您可以根据此格式自行进行设置时间。举例，当前时间为北京时间2017年08月15日上午06:00:00，设置的超时时间为3小时，则超时时间为2017年08月15日9点，因token中时间为utc时间（utc时间=北京本地时间-时区差8小时），则token中的超时时间为2017-08-15T01:00:00Z。

提供四种语言的计算sample code作为参考：

Java

Python

PHP

Node.js

public String sha256Hex(String stringToSign, String signingKey) {
 try {
 Mac mac = Mac.getInstance("HmacSHA256");
 mac.init(new SecretKeySpec(signingKey.getBytes(Charset.forName("UTF-8")), "HmacSHA256"));
 return new String(Hex.encodeHex(mac.doFinal(stringToSign.getBytes(Charset.forName("UTF-8")))));
 } catch (Exception e) {
 e.printStackTrace();
 }
}

• 合法推流地址：

```
rtmp://<push.your-domain.com>/<your-app>/<your-stream>?token=计算出来的token值&expire=2017-08-15T01:00:00Z
```

🔗 播放认证

为确保视频在播放时不被非法获取，LSS提供token认证和有效期限相结合的播放地址。开启播放认证后，您可以通过“播放地址+timestamp+secret”的方式获取完整的加密播放地址。

操作步骤如下：

1. 登录 [LSS管理控制台](#)。
2. 在左侧导航栏点击**安全策略**，进入安全策略页面。
3. 点击**新建安全策略**。



4. 在新建安全策略页将**播放认证**按钮打开，并填写安全策略名称。

新建安全策略

帮助文档

*名称:

可由小写字母、数字和下划线组成，必须以小写字母开头，最长40个字符。

*推流认证:

ON

开启推流认证后，您可以设置推流地址的有效期，过了有效期该地址将失效，再次推流时需重新获取推流地址。

*播放认证:

ON

开启播放认证后，播放地址附带token且仅在用户设定的有效期内可访问。

密钥:

保存安全策略后，服务器会生成密钥。

绑定标签:

温馨提示：标签支持您按各种标准（如用途、所有者或项目）对资源进行分类；每个标签包含键和值两部分

标签键：

默认项目

 值：

请选择已有或手动输入

 X

+ 添加标签

帮助文档

前往标签管理

防盗链

Referer

IP

☒ 允许Referer为空

类型:

☒白名单生效

☐黑名单生效

黑名单与白名单同一时间只能生效一种

+ 添加refer白名单

删除

☐ 全选

没有数据

确定

取消

5. 点击 **确定** 完成创建。
6. 创建完成后，您可以在安全策略管理页找到创建的安全策略点击“编辑”，即可查看密钥key的值：

编辑安全策略

*名称:

demo

*推流认证:

ON

开启推流认证后，您可以设置推流地址的有效期，过了有效期该地址将失效，再次推流时需重新获取推流地址。

*播放认证:

ON

开启播放认证后，播放地址附带token且仅在用户设定的有效期内可访问。

密钥:

显示

7. 在域名或者单个流的详情页面，均可配置安全策略。

高级设置

以下模板更新后在下一次推流时生效，对当前处于直播中有输入流的Stream暂不起作用。

转码模板: 线路L0: lss.forward_only ;

录制模板: -

水印模板: -

缩略图模板: -

安全策略: default

黄反审核: default

通知接口: -

8. 计算secret，方式为：md5（密钥+播放地址+timestamp）。

- 密钥：在播放认证开关可以获取到播放密钥，按照步骤一已获取。
- 播放地址: /<play.your-domain.com>/{app-name}/{stream-name}。
- timestamp：用户指定播放的超时时间，格式需要转换为十进制 Unix 时间戳，推荐一个[在线转换网址](#)。
- 将（密钥+播放地址+timestamp）拼接完成后，加密为32位小写的md5码。
假设key=111， domain = play.domain.com， app=live， stream=ghs, timestamp=149055612，md5加密后的secret为：md5(111/play.domain.com/live/ghs149055612)=3b927f3f31dd31613a537fad7a640a76

9. 拼接完整的加密播放地址，格式为：

http://{playdomain}/{app}/{stream}.flv?timestamp=149055612&secret=3b927f3f31dd31613a537fad7a640a76>

🔗 设置防盗链

应用场景

- 通过添加Referer/IP 黑白名单方式可以解决部分盗链问题，保护资源不被非法下载盗用。
- LSS 支持用户自己配置 Referer/IP 黑白名单，用户可在更新安全策略页面的“防盗链”选项卡中设置防盗链。
- 用户最多可配置200个Referer/IP黑白名单，每项最多可输入4096字节（包含空格和回车符）。
- 黑名单和白名单同一时间只能生效一种。
- 由于referer内容可以伪造，referer防盗链方式对于资源的保护仍有泄漏风险，我们推荐您使用播放认证方式保护您的源站资源。

注意事项

黑名单与白名单同一时间只能生效一种。

配置方法

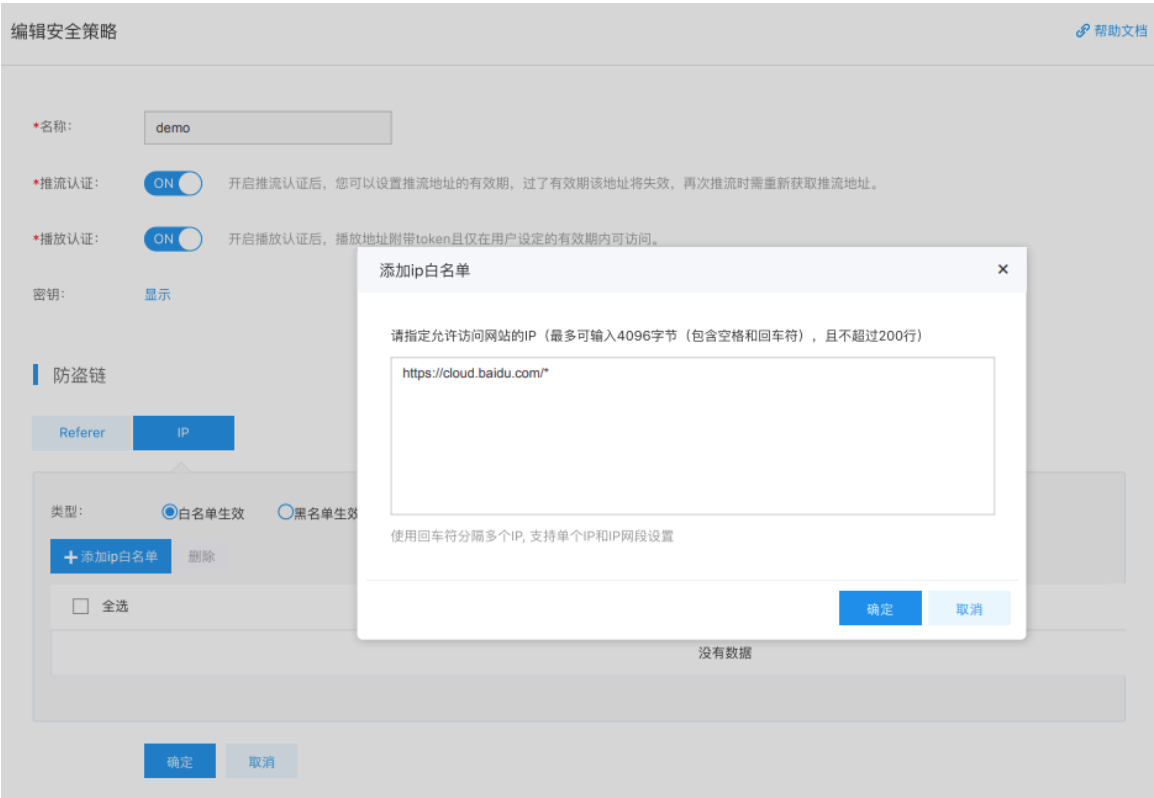
1. 设置Referer黑白名单

选择“白名单生效”或“黑名单生效”，输入允许或禁止访问直播流播放的URL源站地址。

默认允许Referer为空。空Referer与Referer黑白名单是分别配置，如果允许Referer为空，空referer即可访问，否则禁止访问。

2. 设置IP黑白名单

选择“白名单生效”或“黑名单生效”，输入允许或禁止访问直播流播放地址的IP名单。



说明

点击删除，可自定义删除域名Referer/IP的黑白名单，数据清除后不可被还原，需谨慎操作。

统计分析

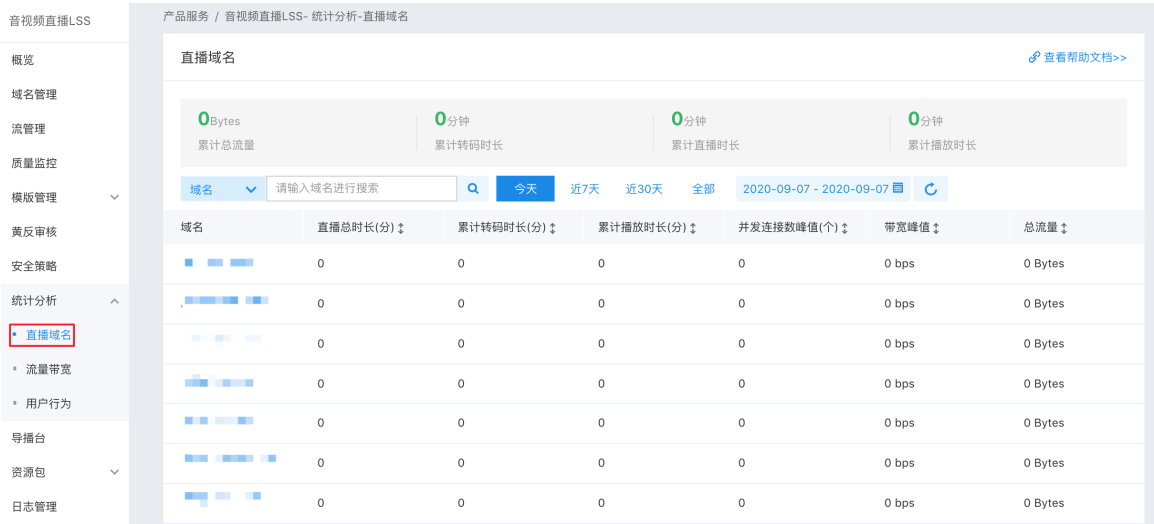
直播域名

概述

LSS服务会统计当前账户所有/特定域名的直播数据，包括直播总时长（推流时长）、累计转码时长、累计播放时长（用户观看时长，不同用户累加计时）、并发连接数峰值、带宽峰值和总流量，您可以查看域名下所有流的统计信息。

查看直播域名

- 1. 登录[音视频直播LSS操作控制台](#)。
- 2. 在左侧导航栏点击 **统计分析 -> 直播域名**，进入直播域名页面。



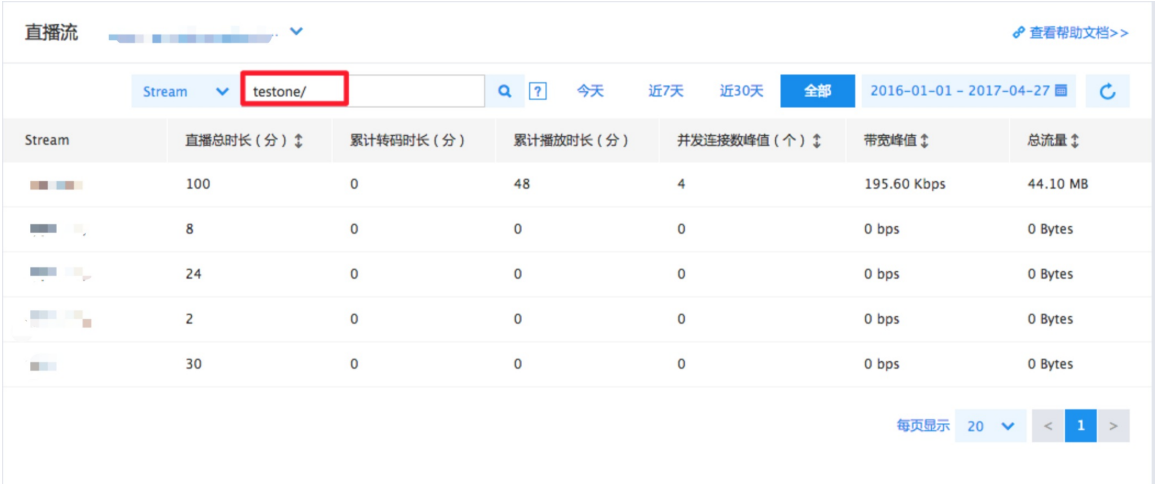
3. 在检索框输入域名关键字，选择查询时间，查询当前域名下直播数据。每个维度支持排序，方便从不同维度分析和对比域名的直播数据。



4. 点击目标域名，可以查看当前域名下stream的直播数据。



5. 在检索框输入 App 名称，和需要筛选的 Stream，格式为：App-Name/Stream-Name。



6. 您可以按照不同的时间跨度（全部、今天、近7天、近30天）筛选数据，查看相关统计信息。

🔗 相关 API

- [查询统计数据](#)
- [查询所有Stream数据](#)
- [查询特定Stream数据](#)

用量查询

方便用户了解直播在不同时间段的流量、带宽和请求数据，LSS提供了对指定域名进行不同时长、多个维度的直播数据统计。

1. 登录[音视频直播LSS操作控制台](#)。
2. 在左侧导航栏点击“统计分析”->“用量查询”，进入用量查询页面。

在此页面您可以查看不同的时间跨度内所有/特定域名的用量情况。



3. 在“时间选择”栏，您可以切换不同的时间跨度。不同的时间跨度，统计的粒度也不同。
 - 所选时间区间<=31天时，统计粒度支持：5分钟、1小时、1天。
 - 所选时间区间>31天时，统计粒度支持：1小时、1天。
4. 选择页面中部的功能按钮，可以查看带宽、流量、请求数、直播录制、直播时移、直播截图、转推的用量统计。

统计数据如下图带宽统计数据所示。 点击右上角[下载](#)图标，可以将数据以Excel的形式下载到本地查看。



🔗 相关 API

- [查询总带宽](#)
- [查询带宽](#)
- [查询总流量](#)
- [查询流量](#)
- [查询总请求数](#)
- [查询请求数](#)

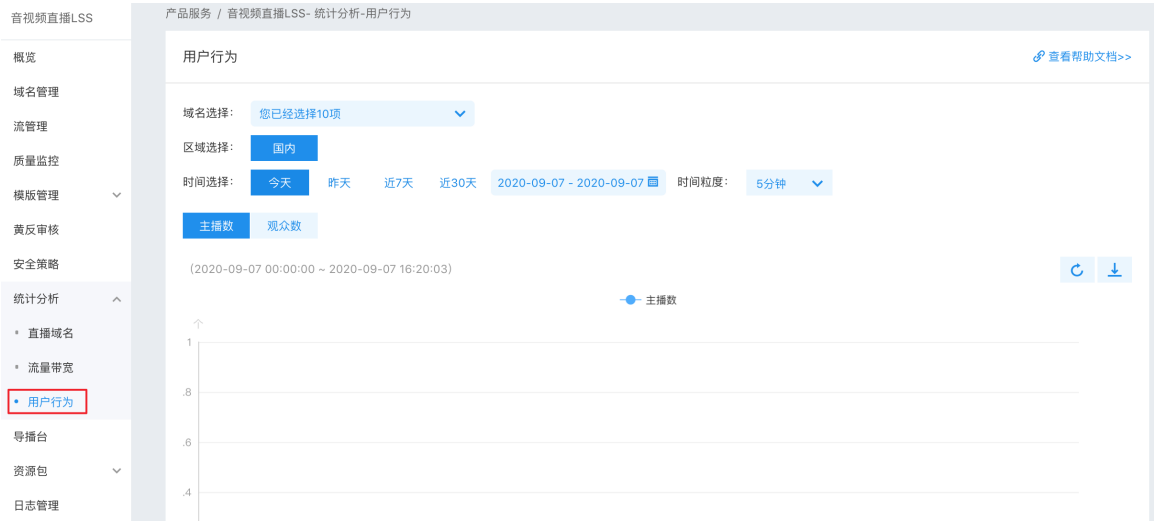
用户行为

🔗 概述

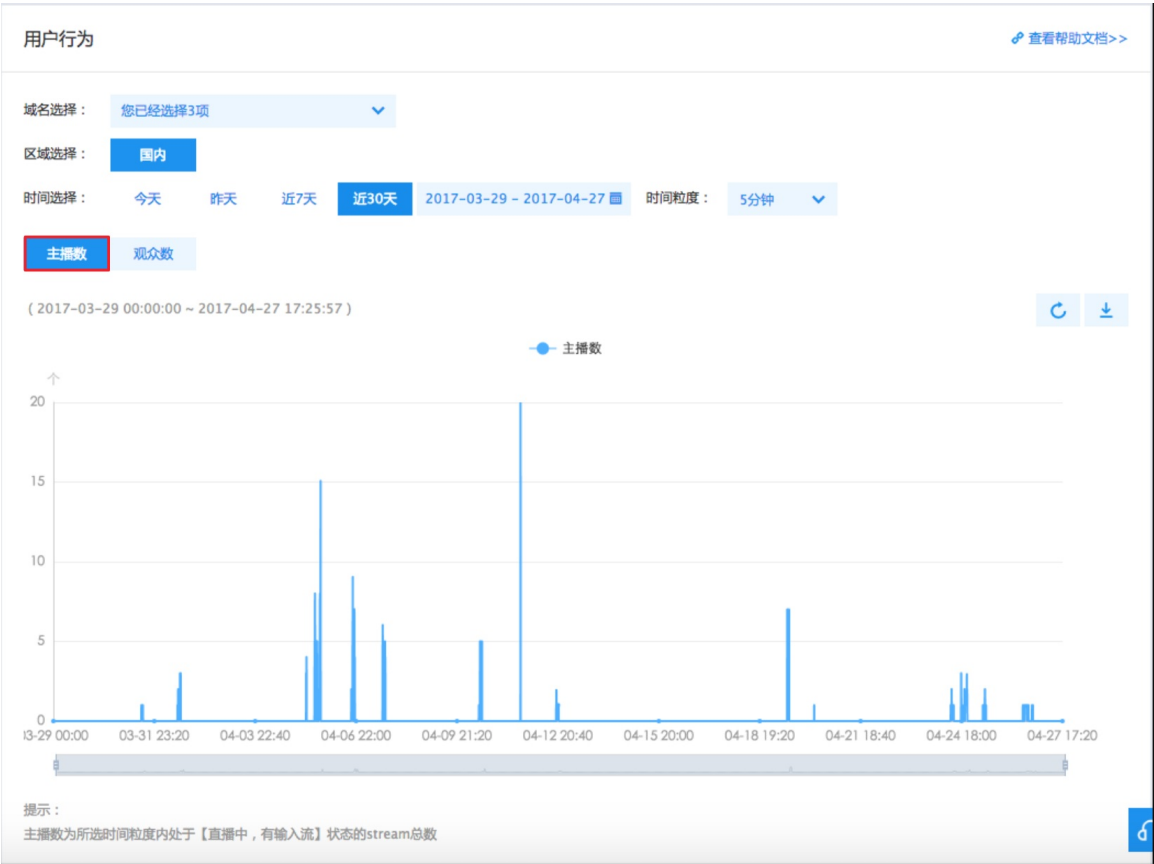
实时统计和展示用户的行为，是直播需要监测的一项重要指标。LSS 通过实时监控主播数量和观众数量，为企业分析用户行为提升了良好的体验。

🔗 主播数

1. 登录[音视频直播LSS操作控制台](#)。
2. 在左侧导航栏点击 **统计分析 -> 用户行为**，进入用户行为页面。



3. 选择页面中的主播数按钮，查看主播数统计。



4. 在此页面您可以查看近3个月内所有/特定域名的主播数量情况。
5. 在**时间选择**：栏，切换不同的时间跨度，统计的粒度也不同。另外，当数据有延时情况，点击右上角刷新按钮，数据会重新加载。
- 所选时间区间<=31天时：统计粒度支持5分钟、1小时、1天。
 - 所选时间区间>31天时：统计粒度支持1小时、1天。
6. 主播数统计数据参见下图，点击右上角 ，还可以将excel数据下载到本地查看。

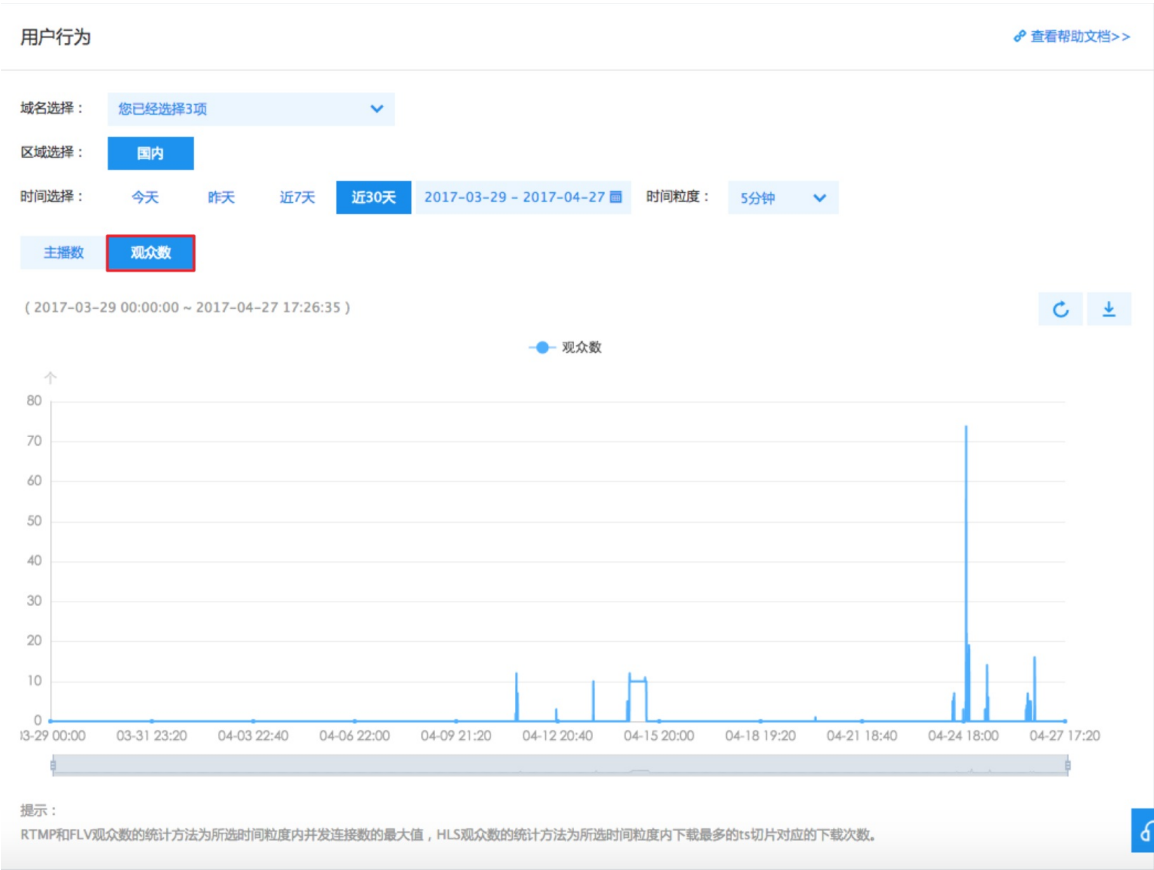


观众数

1. 登录[音视频直播LSS操作控制台](#)。
2. 在左侧导航栏点击 **统计分析 -> 用户行为**，进入用户行为页面。



3. 选择页面中的**观众数**按钮，查看观众数统计。



4. 在此页面您可以查看近3个月内所有/特定域名的观众数量情况。
5. 在时间选择：栏，切换不同的时间跨度，统计的粒度也不同。当数据有延时情况，点击右上角刷新按键，数据会重新加载。
- 所选时间区间<=31天时：统计粒度支持5分钟、1小时、1天。
 - 所选时间区间>31天时：统计粒度支持1小时、1天。
6. 观众数统计数据参见下图，点击右上角 [↓](#)，还可以将excel数据下载到本地查看。



注意：当出现下列三种情况时，观众数可能会有重复统计情况：

- 1. 在开始推流前，观众首先打开播放器播放；
- 2. 推流过程中，出现流中断情况，观众再次接入直播；
- 3. 直播推流结束后，无流状态时，观众持续或再次打开播放器播放。

相关 API

- API 接口：[统计实时播放人数和带宽](#)

日志管理

概述

LSS 管理控制台为用户保存每个域名近180天的国内 CDN 日志，便于查询直播记录。

日志字段说明

下载的日志格式包含如下字段：

字段	含义
remote_addr	客户端IP
remote_user	登录用户名，一般为空“-”
host	域名
time_local	时间戳
request_time	请求处理时间，单位s
request	请求的方法、路径、版本，例如"GET /path/of/url?args&a=1&b=2 HTTP/1.1"
status	HTTP状态码
bytes_sent	返回的字节长度
body_bytes_sent	返回的body长度
referer	请求Referer
cookie	客户端Cookie
user_agent	请求UA，即请求Header中的User-Agent部分
x_forwarded_for	即请求Header中的x_forwarded_for部分
connection_id	实际为requestId#silent_limit_req_detail，requestId 为请求标识符，多级节点唯一标识该请求，格式为timestamp-connection-ip；打印域名计数详情：silent_limit_req_detail，格式为"status-number"，比如"503-10"，代表这条日志之前记录的503有10个。如为空，则自动用“-”代替。
udf_hit	命中状态，例如:MISS/HIT
server_ip	服务端IP

🔗 日志下载

1. 登录 [LSS管理控制台](#)，选择 **日志管理**。
2. 百度智能云将保存180天的日志数据，您可以根据需要进行**域名选择**和**时间选择**，筛选出目标日志，选择需要下载的日志，点击 **下载**，将日志下载到本地。

产品服务 / 音视频直播LSS- 日志管理

日志管理 [?](#) [帮助文档](#)

域名选择: play8111.tjsh.com.cn

日期选择: 2018-01-10 00:00:00 - 2018-01-16 13:21:00 百度云将为您保存90天的日志数据

日志名称	日志开始时间	日志结束时间	大小	操作
play8111.tjsh.com.cn/20180112/play8111.tjsh.com.cn_2018_01_12_	2018-01-12 13:00:00	2018-01-12 13:59:59	426 Bytes	下载
play8111.tjsh.com.cn/20180112/play8111.tjsh.com.cn_2018_01_12_	2018-01-12 19:00:00	2018-01-12 19:59:59	677 Bytes	下载

标签管理

概述

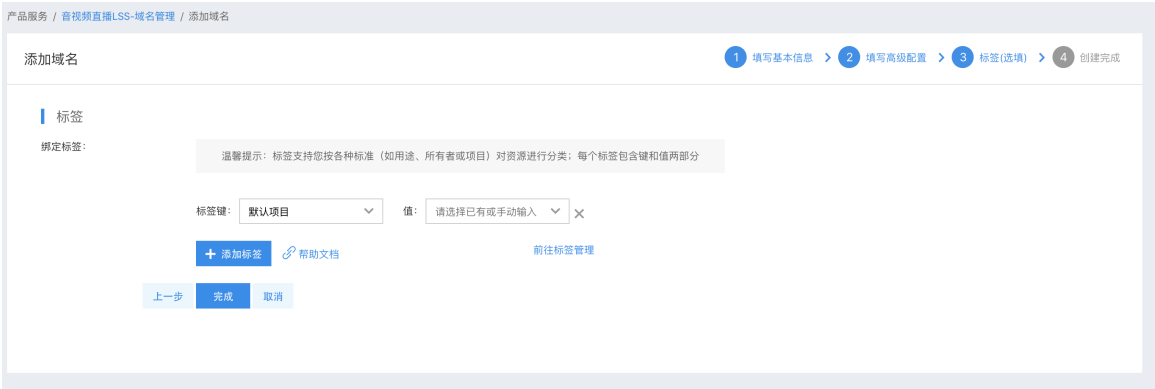
百度智能云提供标签管理功能，通过给每个云资源添加标签，从而快速分类和识别管理这些资源，LSS的资源分为域名和各类模板。每个标签由键和值两部分组成，标签（键+值）唯一。用户可以根据项目和场景需求，为域名、模板等资源添加标签，便于对它们进行分类和识别管理。

注意：每个用户最多可以创建200个标签。

添加标签

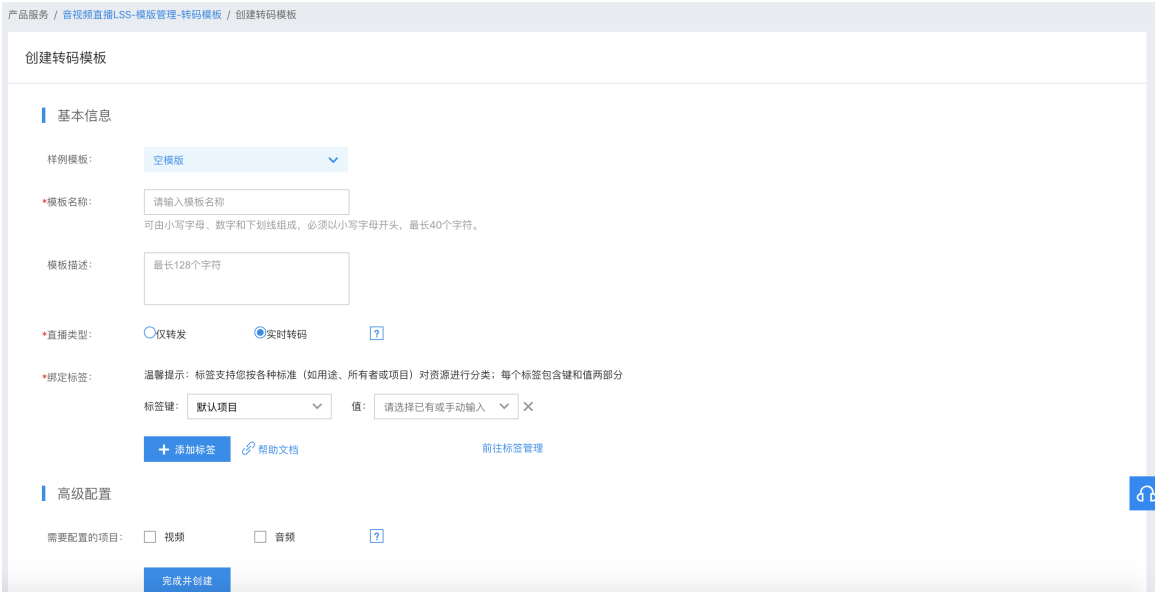
添加域名标签

- 1. 登录 [LSS管理控制台](#)，进入“音视频直播LSS”页面。
- 2. 点击[域名管理](#)，进入“域名管理”页面。
- 3. 您可以在添加域名的操作中进行[标签（选填）](#)。



添加模板标签

- 1. 登录 [LSS管理控制台](#)，进入“音视频直播LSS”页面。
- 2. 点击[模板管理](#)模块。
- 3. 以转码模板为例：点击左侧导航栏[转码](#)，进入“转码模板”页面，您可以在创建转码模板的操作中进行[添加标签](#)。



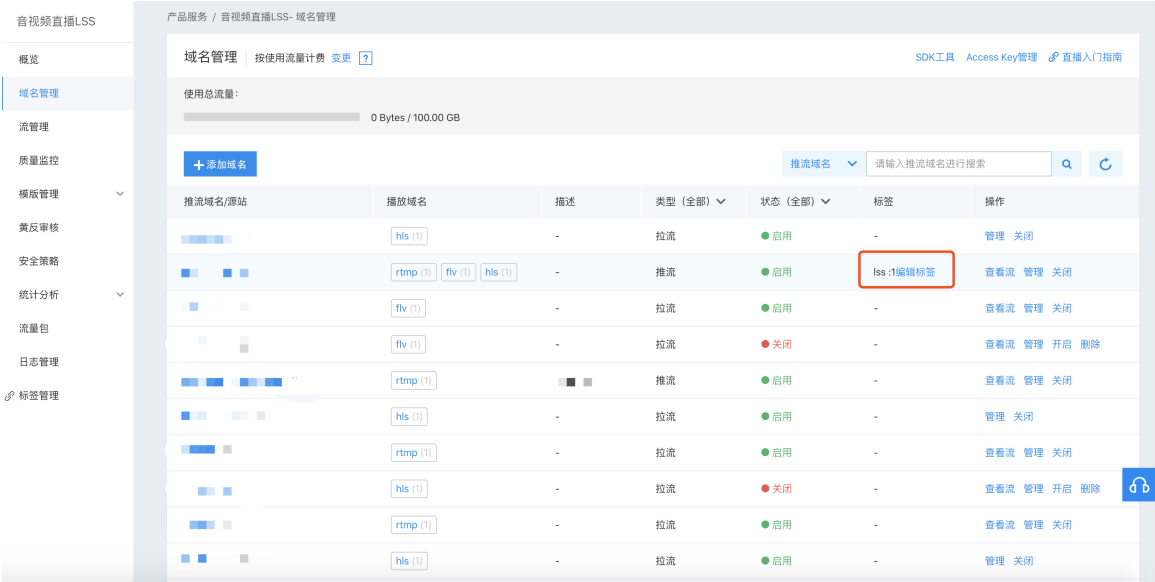
注意：主用户在给资源绑定标签时，如果不选择任何标签或者直接跳过标签选择，那么这个资源的标签将会是空标签；子用户在给资源绑定标签时，如果不选择任何标签或者直接跳过标签选择，那么这个资源的标签将会被默认绑定上标签库里

面所有的标签。

编辑标签

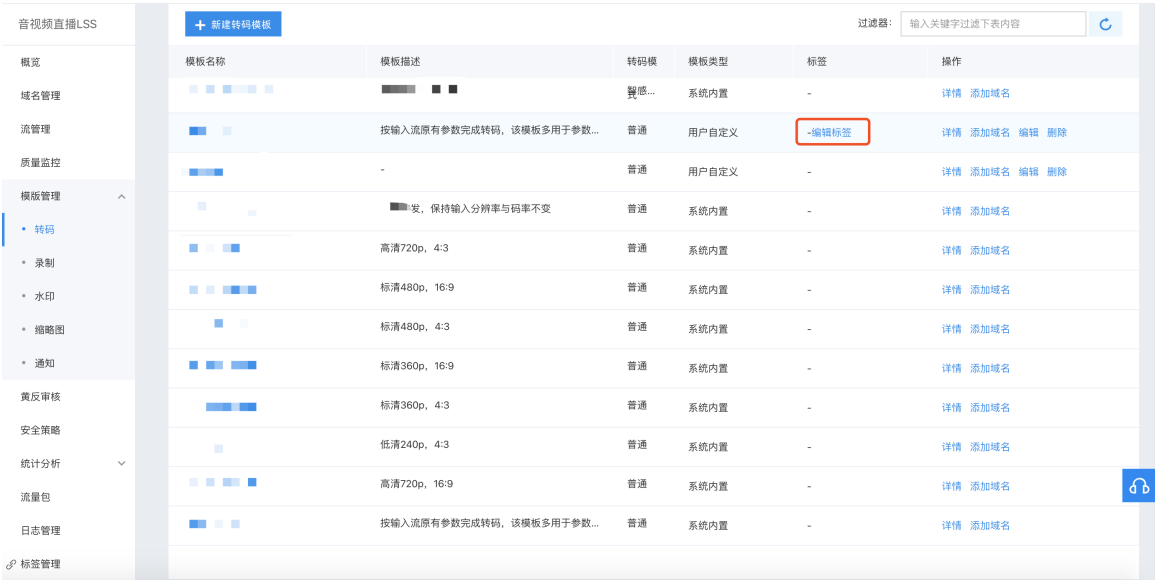
编辑域名标签

- 1. 登录 [LSS管理控制台](#)，进入“音视频直播LSS”页面。
- 2. 点击[域名管理](#)，进入“域名管理”页面。
- 3. 点击目标域名[标签列](#)中的[编辑标签](#)按钮，对目标域名标签进行编辑。



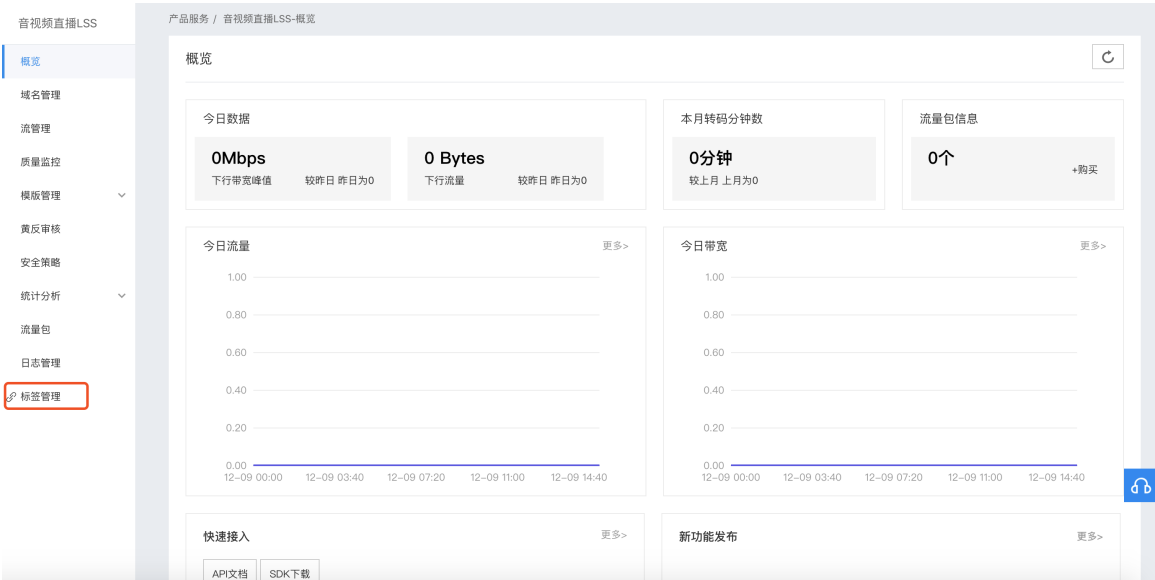
编辑模板标签

- 1. 登录 [LSS管理控制台](#)，进入“音视频直播LSS”页面。
- 2. 点击[模板管理](#)模块下的[转码](#)，进入“转码模板”页面。
- 3. 点击目标模板[标签列](#)中的[编辑标签](#)按钮，对目标模板标签进行编辑。
- 4. 以转码模板为例：



删除管理

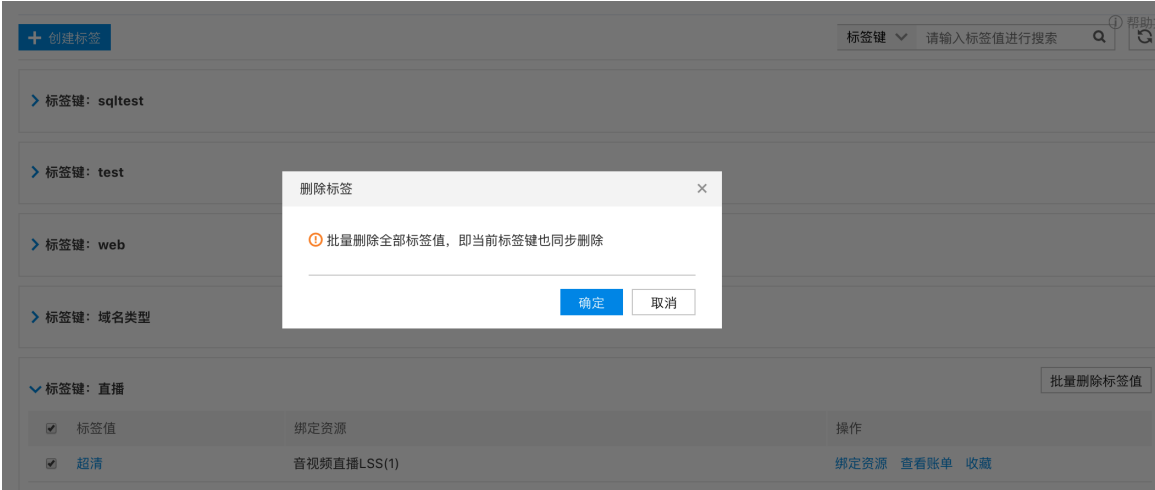
- 1. 登录 [LSS管理控制台](#)，进入“音视频直播LSS”页面。
- 2. 点击左侧导航栏[标签管理](#)。



3. 在“标签管理页面”，您可以进行标签管理，勾选需要删除的目标标签，点击**批量删除标签值**按钮，进行删除操作。



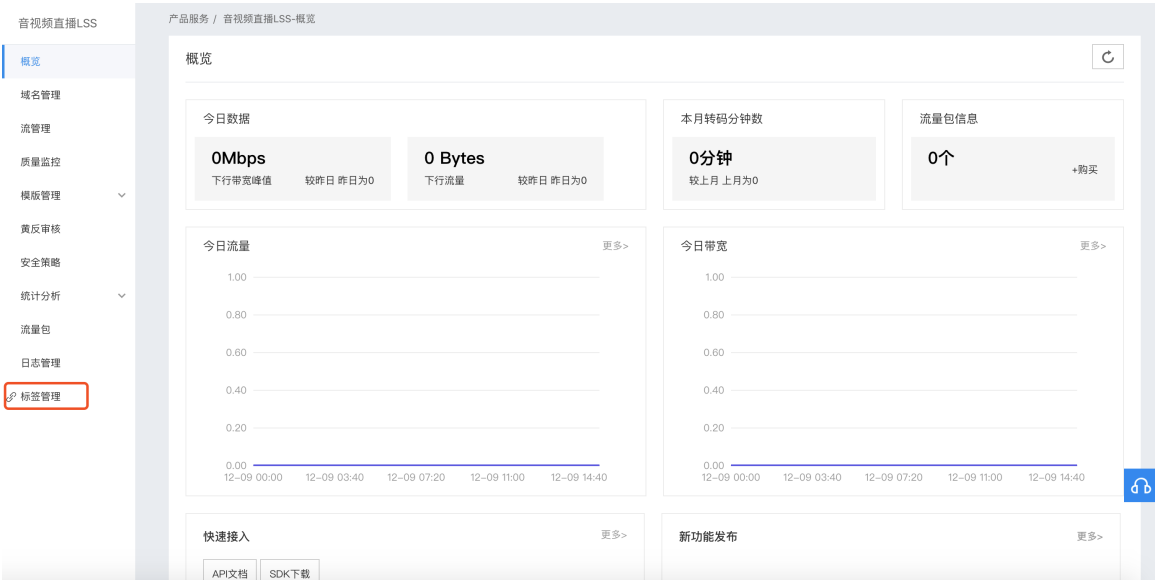
4. 确认信息之后，点击**确认**，进行删除。



注意：子用户不能删除被绑定的标签，要想删除被绑定资源的标签，请登录主用户操作。

绑定资源

- 1. 登录 [LSS管理控制台](#)，进入“音视频直播LSS”页面。
- 2. 点击左侧导航栏**标签管理**。



3. 在“标签管理页面”，您可以进行标签管理，勾选需要绑定资源的目标标签，点击操作列绑定资源按钮。

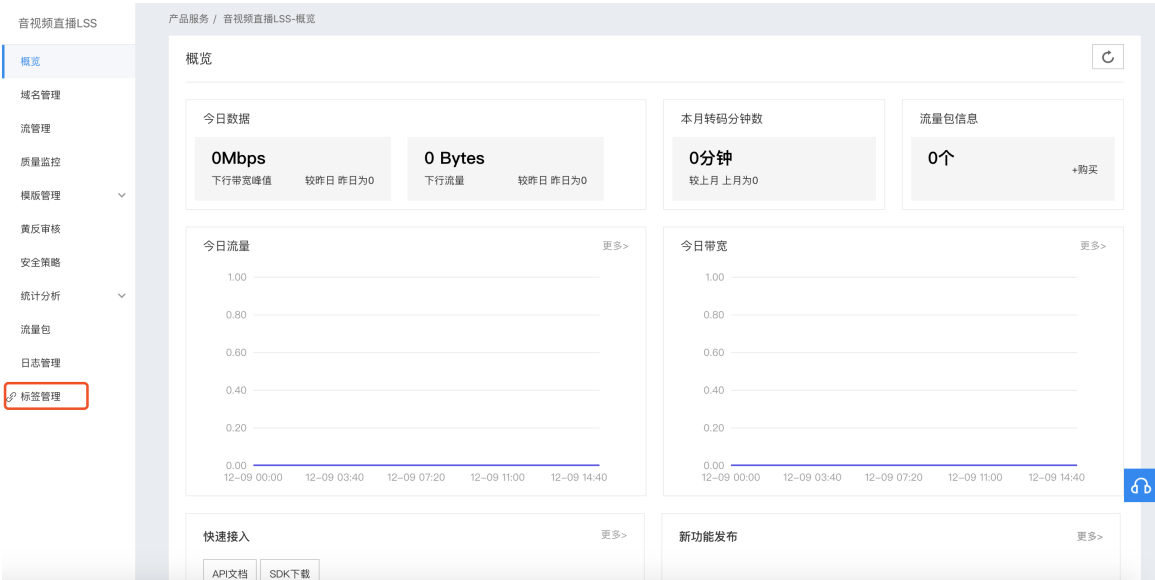


4. 选择资源、地域，勾选目标实例，点击确定按钮进行绑定操作。



绑定资源

- 1. 登录 [LSS管理控制台](#)，进入“音视频直播LSS”页面。
- 2. 点击左侧导航栏[标签管理](#)。



3. 在“标签管理页面”，您可以进行标签管理，勾选需要绑定资源的目标标签，点击操作列绑定资源按钮。

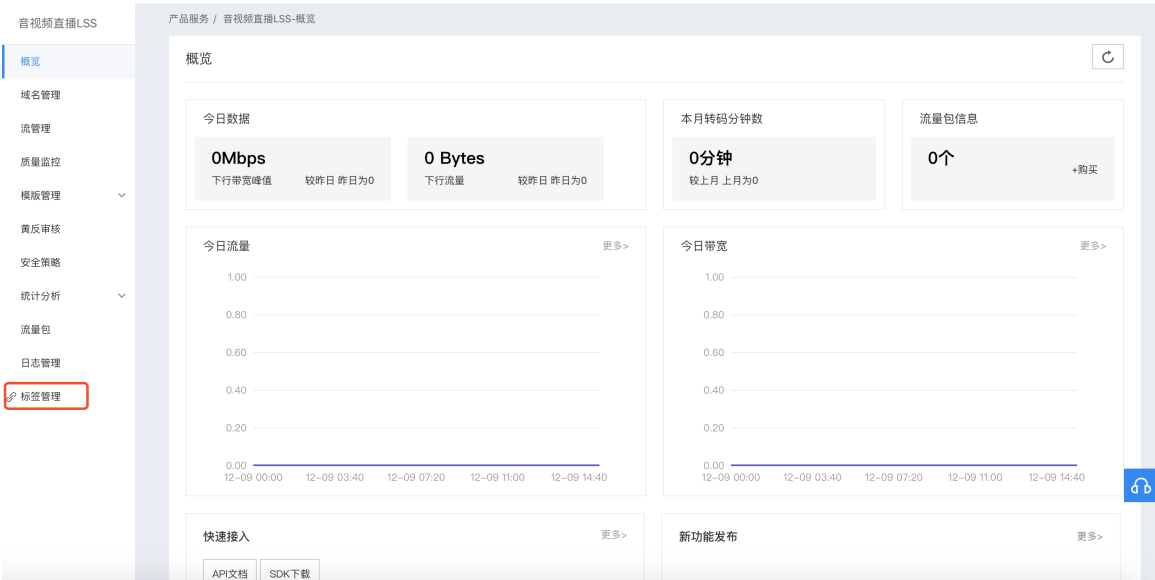


4. 选择资源、地域，勾选目标实例，点击确定按钮进行绑定操作。

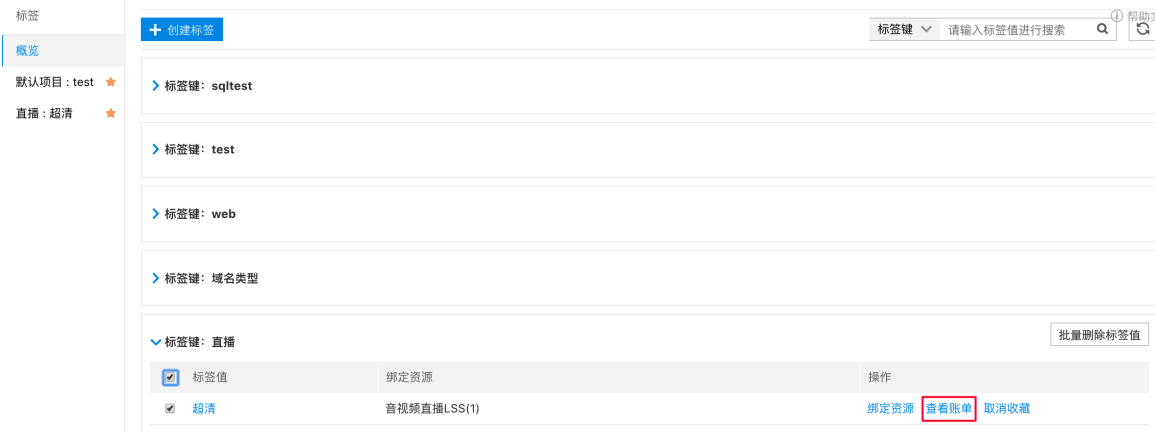


查看账单&收藏

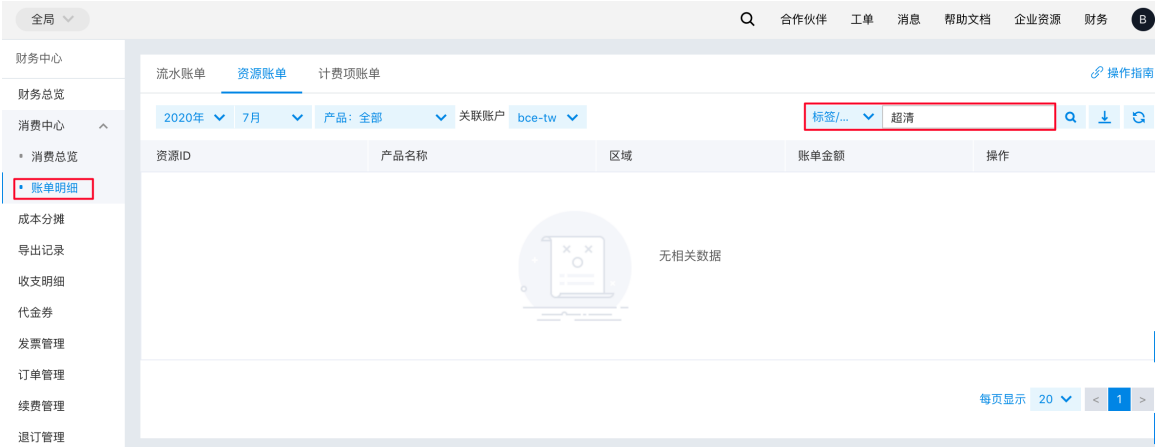
- 1. 登录 LSS管理控制台，进入“音视频直播LSS”页面。
- 2. 点击左侧导航栏标签管理。



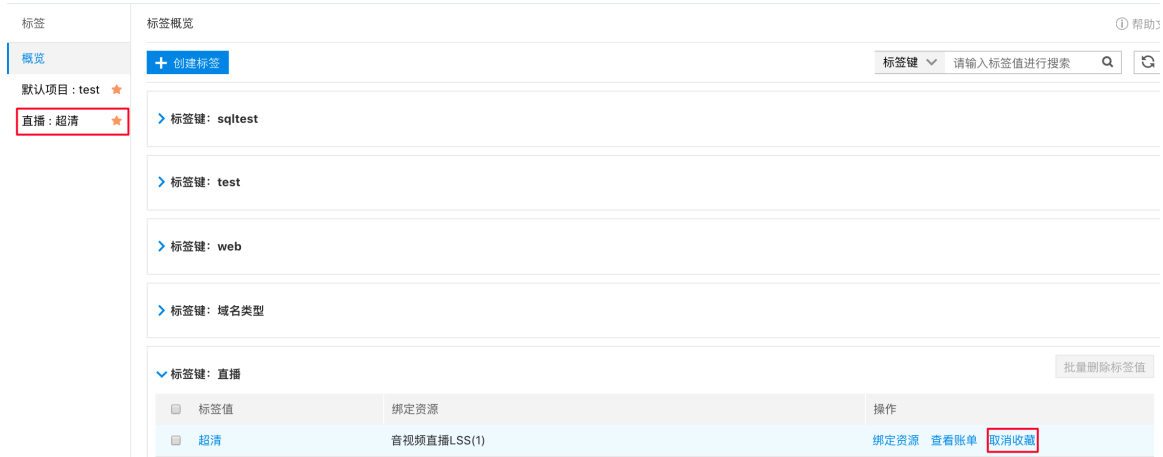
3. 在“标签管理页面”，您可以进行标签管理，勾选目标标签。



4. 点击操作列查看账单按钮，进入“账单明细页面”，显示所选标签相关账单信息。



5. 点击操作列收藏按钮，所选标签就会出现在收藏列表。若您要进行取消工作，点击操作列取消收藏按钮，可取消收藏此标签。



多用户访问控制

概述

百度智能云多用户访问控制（IAM）可为用户提供权限控制功能，目前音视频直播LSS已经接入IAM，并支持对LSS整个产品和产品中更细粒度资源（域名、模板）的权限管理。

通过IAM，用户主用户可以创建多个子用户，通过对子用户授权，可以控制子用户对产品资源、功能的访问。[了解多用户访问控制](#)。

使用场景

- 为子用户分配产品权限。
- 为子用户添加系统策略，参考 [子用户绑定策略](#)。

注意： 当添加多个策略时，这些策略都会生效，即取他们的并集权限。

- 为子用户添加某个具体资源的权限，参考[按策略生成器创建策略](#)。

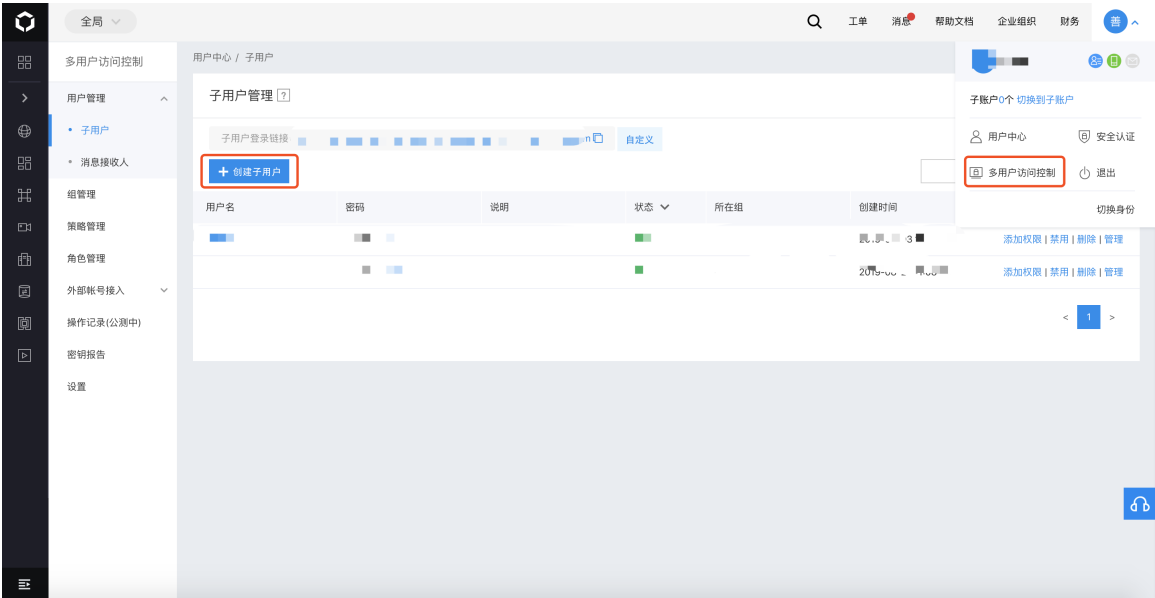
说明： 权限配置只能对选定的推拉流域名有查询权限，即无法修改他们的配置。

- 为某个子用户分配某一个业务部门资源权限，参考 [按标签创建策略](#)。

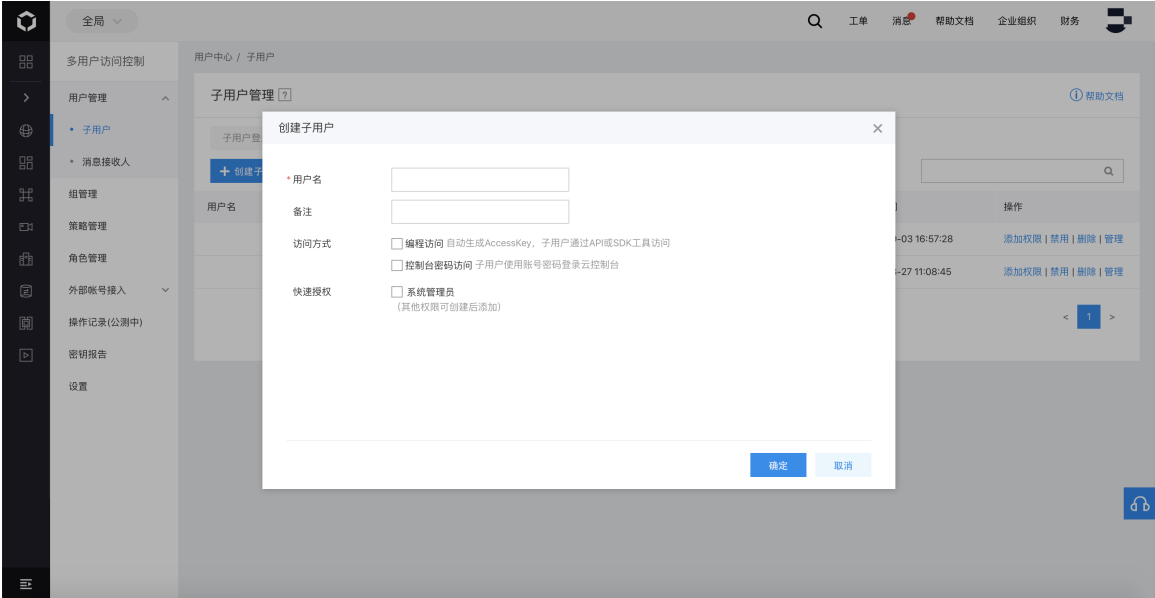
说明： 配置的策略允许对权限配置中选择的标签资源提供管理权限。

创建子用户

1. 主账号用户登录后在控制台选择“多用户访问控制”进入用户管理页面。



- 2. 在左侧导航栏点击“用户管理”，在“子用户”页，点击“创建子用户”。
- 3. 在弹出的“创建子用户”对话框中，完成填写“用户名”和确定，



- 4. 返回“子用户管理列表”区可以查看到刚刚创建的子用户。

配置策略

权限策略分为系统策略和自定义策略。

- **系统策略**：只能控制整个产品的权限，为产品粗粒度权限控制，属于内置策略，用户无法创建。
- **自定义策略**：可以控制产品下单个或批量资源的权限，可由用户创建，包含按策略生成器创建和按标签创建。

系统策略

系统策略包含读写权限和只读权限2种策略，权限范围详细如下：

策略名称	权限说明	权限范围
LSSFullControlAccessPolicy	整个产品的读写权限	用户可对所有域名、模板进行创建、删除、修改、查询
LSSReadAccessPolicy	整个产品的读权限	用户可对所有域名、模板进行查询

自定义策略

自定义策略分为按策略生成器创建和按标签创建。

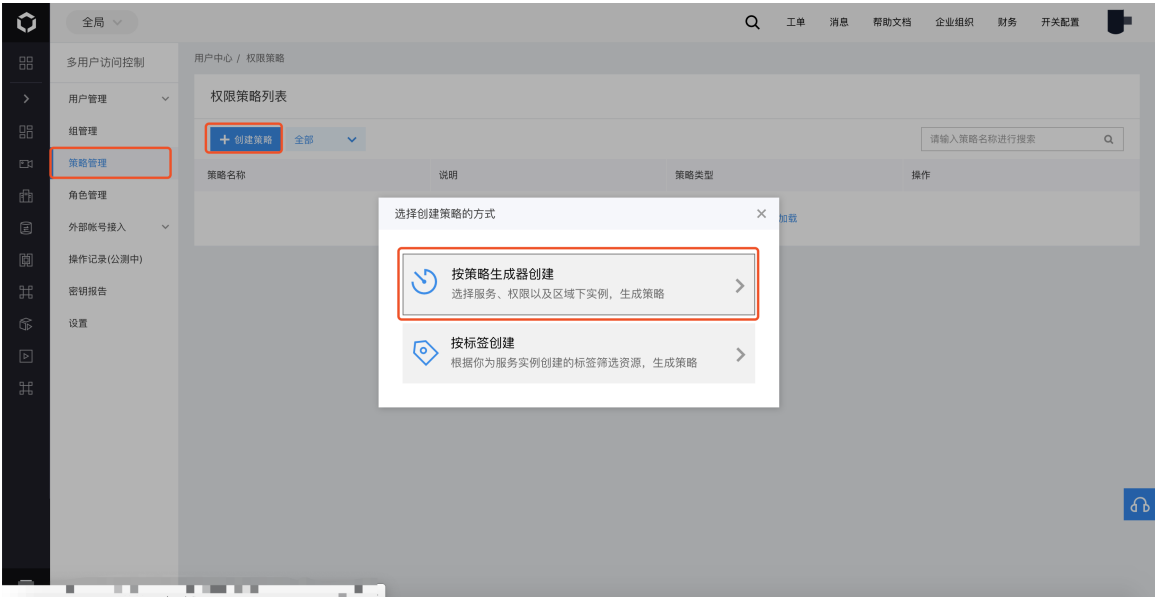
- 按策略生成器创建的策略表示对LSS的单个或多个资源（域名、模板）进行权限控制；涉及到的权限为管理权限和只读权限。
- 按标签创建的策略表示对LSS打上某类标签的资源进行批量权限控制；涉及到的权限为管理权限、只读权限和运维权限（暂时不生效，用户可以不勾选）。关于标签说明，请查看[标签管理](#)。
- 权限说明，如下表所示：

类型	权限说明	权限范围
管理权限	单个或批量资源的读写权限	用户可对单个或批量域名、模板进行创建、删除、修改、查询
只读权限	单个或批量资源的读权限	用户可对单个或批量域名、模板进行查询

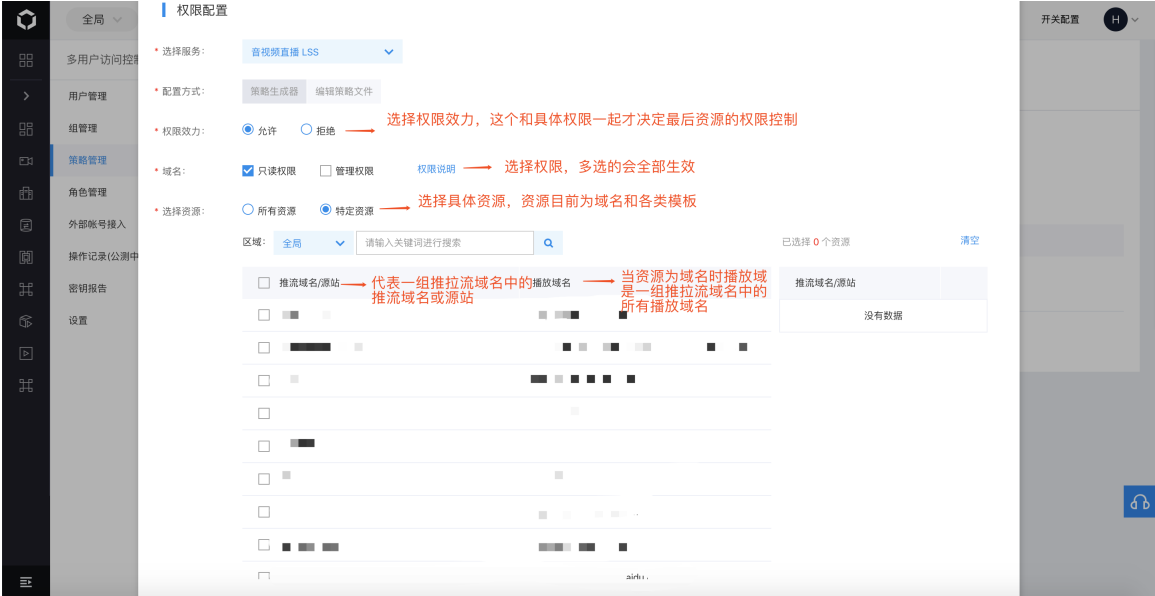
创建自定义策略

按照策略生成器创建策略

1. 进入策略管理模块，点击创建策略，选择按策略生成器创建。

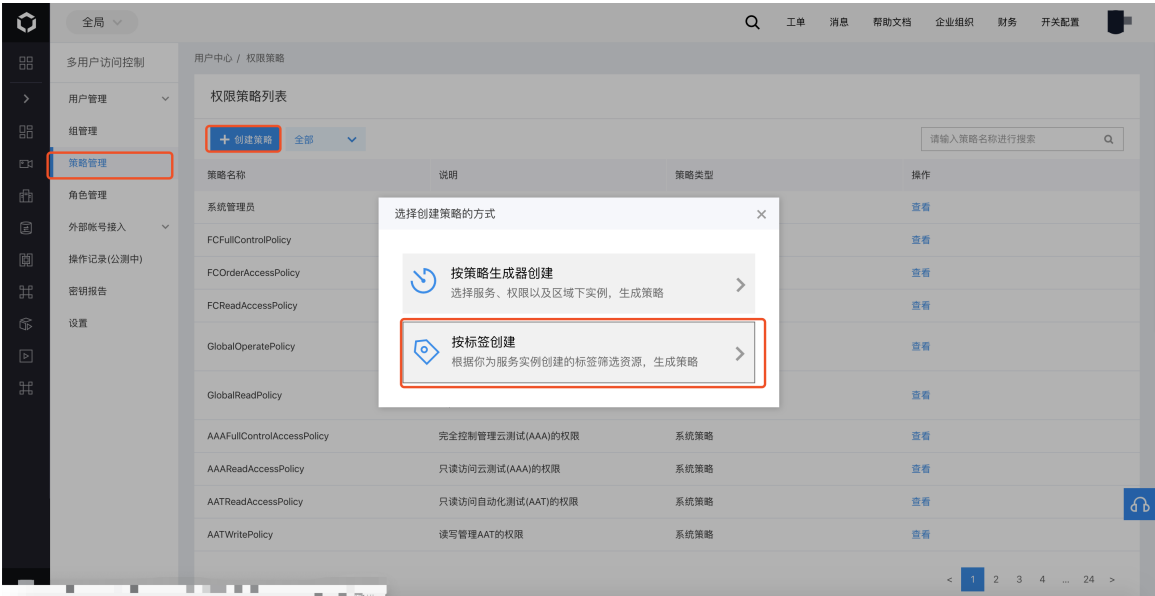


2. 为策略添加具体权限。

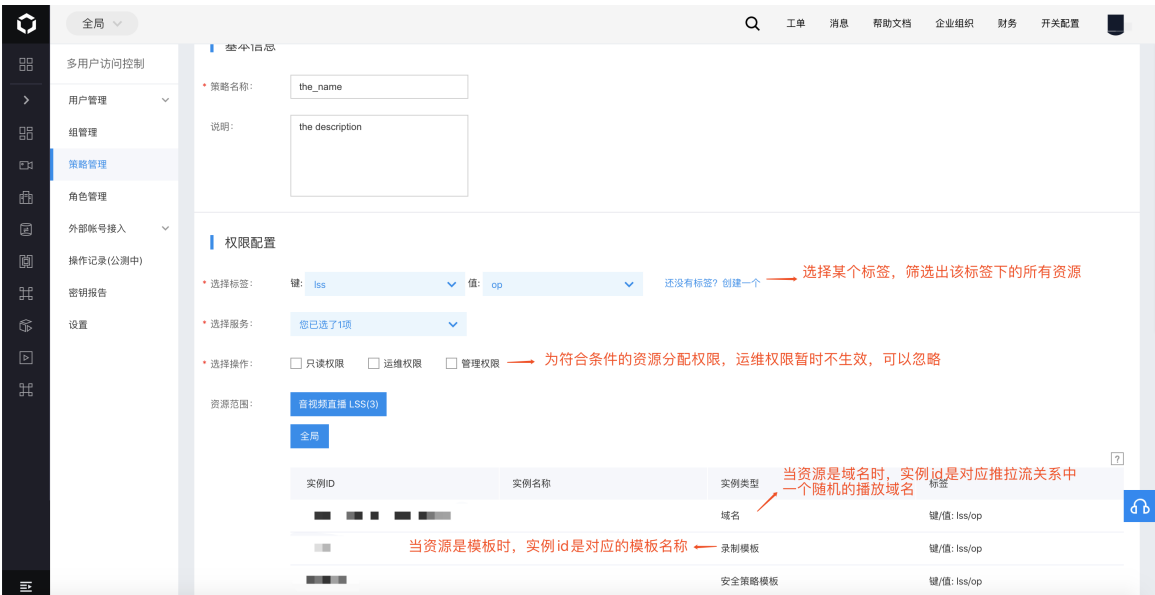


按照标签创建策略

1. 进入策略管理模块，点击创建策略，选择按标签创建策略。

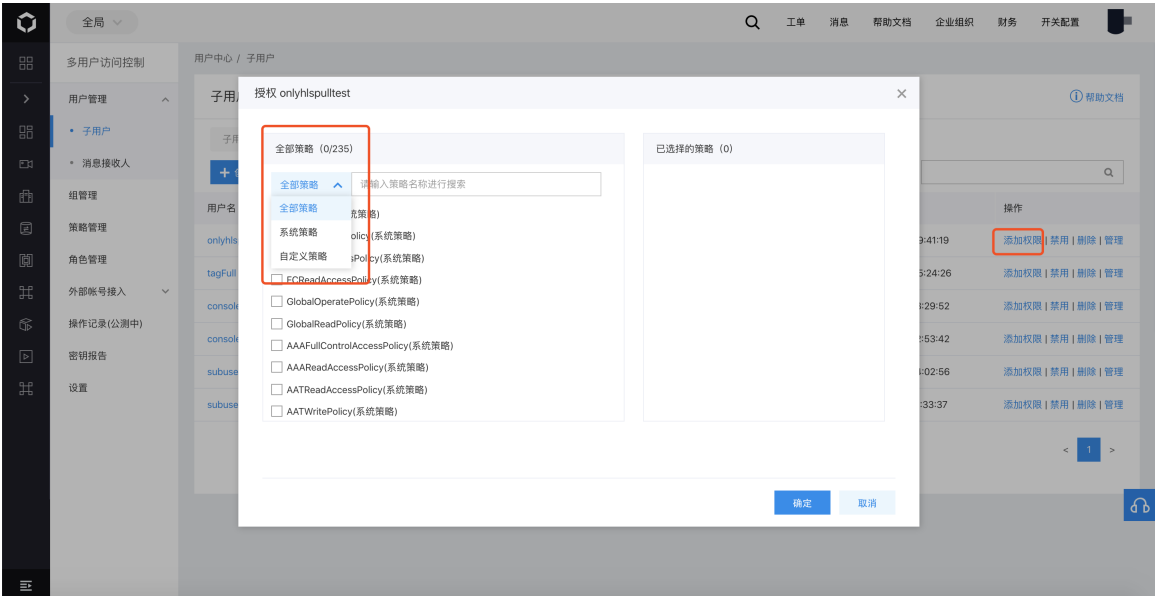


2. 为策略添加具体权限。



子用户绑定策略

- 1. 进入 用户管理>子用户模块，展示子用户列表。
- 2. 选择需要绑定策略的子用户，点击操作列的添加权限，进行策略的绑定。



Demo体验

百度智能云音视频直播Demo集成百度智能云推流和播放SDK，为直播用户提供美颜、AR特效功能，用户可以扫码下载最新的直播Demo体验直播推流和观看功能。

平台	Demo下载	下载密码
iOS		bdcloud
Android		bdcloud

推流端SDK

Android端推流SDK

简介

百度智能云 [Android 推流 SDK](#) (以下简称“推流 SDK”) 是百度官方推出的 Android 平台用于推流的软件开发工具包，为您提供简单、便捷的开发接口，助您快速实现音视频直播推流功能。

优势

- 压缩比例大，带宽占用小
- 硬件兼容性好，可支持 arm64 指令集
- 良好的软件版本兼容性，最低可支持 Android 4.2版本

特点

- **支持主流 RTMP 服务器**
支持目前几乎所有 RTMP 服务器 (simple-rtmp-server、crtmpserver、FMS 等)。
- **支持所有的 RTMP 协议**
支持所有 RTMP 协议及变种 (RTMP、RTMPT、RTMPE、RTMPS、RTMPTE、RTMPTS 等)。
- **性能强大**
CPU/内存资源占用率低，压缩比例大，延时小。
- **完备的编程接口提供极大的灵活性**
通过提供完备的编程接口，方便您在采集音视频过程中对相机的对焦模式、测光模式等参数进行灵活控制。同时提供开发示

例，为您说明各接口调用细节。

● 高兼容性、稳定性和快速响应

软编码支持 Android 4.2及以上平台，硬编码支持 Android 4.3 及以上平台，结合百度开发云直播服务，可实现端到端平均仅 2秒延时的实时直播功能。

🔗 组件及资源

推流 SDK 的完整下载包中包含 lib、doc、demo 三部分：

● lib 目录

存放推流 SDK 的 Java 依赖包和两类底层动态链接库：

- armeabi-v7a：适配 armv7/armv7a 指令集的底层动态链接库
- arm64-v8a：适配 arm64 指令集的底层动态链接库

具体文件如下表所示：

文件名	含义
BRecorder.jar	Java 依赖包，处理音视频采集逻辑
librecorder.so	处理音视频数据的动态链接库，同时提供 JNI 接口
libffmpeg.so	FFmpeg 动态链接库，处理编码与推流逻辑
librtmp_jni.so	librtmp 动态链接库，处理 RTMP 协议交互

● doc 目录

存放 BRecorder.jar 依赖包的 javadoc 文档，该文档详细说明了推流 SDK 各接口 API 类的用法。

● demo 目录

存放 Android 示例工程，展示了如何使用 LiveSession 类来实现 Android 端推流应用。

解压后的目录结构如下所示：

```
Baidu-Capture-SDK-Android-x.x.x
├── libs
│   ├── BRecorder.jar           // 推流 SDK Java依赖包
│   ├── armeabi-v7a            // armv7/armv7a 架构的动态链接库
│   │   ├── librecorder.so
│   │   ├── libffmpeg.so
│   │   └── librtmp_jni.so
│   └── arm64-v8a              // arm64 架构的动态链接库
│       ├── librecorder.so
│       ├── libffmpeg.so
│       └── librtmp_jni.so
├── doc                        // BRecorder.jar 对应的 javadoc 文档
└── demo                      // 示例工程，用于演示推流 API 的使用
    ├── res                   // 工程资源文件目录
    ├── libs                  // Java 依赖包以及动态链接库文件目录
    ├── src                   // 源代码文件目录
    └── ... ...               // 其他工程文件
```

推流-SDK-开发指南

准备工作

- 开通 [百度智能云音视频直播服务](#)，添加推拉流域名并获取推流地址及观看地址。
- 下载最新的[Android 推流 SDK](#)。
- 准备 Android 开发环境：安装有 ADT 插件的 Eclipse 开发环境。
- 准备 Android 运行环境：软编码支持Android 4.2及以上的所有系统，硬编码支持Android 4.3 及以上的所有系统
- 适配 CPU 指令集：armv7/armv7a、arm64。
- 添加依赖：将Baidu-Capture-Android-iOS-x.x.x/libs 目录复制到 Eclipse 工程根目录，注意根据所支持的 arm 版本选择对应的动态链接库。
- 权限声明：请您在 Android App 的 AndroidManifest.xml 中声明如下权限：

初始化 LiveSession

LiveSession是对视频采集、编码、推流等功能模块接口进行封装后的接口类，为您提供友好的编程接口。 LiveSession有两个子类，LiveSessionHW 跟 LiveSessionSW，分别是硬编码和软编码方案对应的实现类。

推流 SDK 使用LiveConfig类完成初始化：

```
LiveConfig liveConfig = new LiveConfig.Builder()
    .setCameraId(LiveConfig.CAMERA_FACING_FRONT) // 选择摄像头为前置摄像头
    .setCameraOrientation(orientation) // 设置摄像头为竖向
    .setVideoWidth(mVideoWidth) // 设置推流视频宽度, 需传入长的一边
    .setVideoHeight(mVideoHeight) // 设置推流视频高度, 需传入短的一边
    .setVideoFPS(mFrameRate) // 设置视频帧率
    .setInitVideoBitrate(mBitrate) // 设置视频码率, 单位为bit per seconds
    .setAudioBitrate(64 * 1000) // 设置音频码率, 单位为bit per seconds
    .setAudioSampleRate(LiveConfig.AUDIO_SAMPLE_RATE_44100) // 设置音频采样率
    .setGopLengthInSeconds(2) // 设置帧间隔, 单位为秒
    .setQosEnabled(true) // 开启码率自适应, 默认为true, 即默认开启
    .setMinVideoBitrate(200 * 1000) // 码率自适应, 最低码率
    .setMaxVideoBitrate(1024 * 1000) // 码率自适应, 最高码率
    .setQosSensitivity(5) // 码率自适应, 调整的灵敏度, 单位为秒, 可接受[5, 10]之间的整数值
    .build();
Log.d(TAG, "Calling initRTMPSession..." + liveConfig.toString());
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.JELLY_BEAN_MR2) {
    mLiveSession = new LiveSessionHW(this, liveConfig);
} else {
    mLiveSession = new LiveSessionSW(this, liveConfig);
}
```

设置预览 View

初始化成功后，必须为 LiveSession 对象设置预览 View，否则无法启动相机。

以下示例代码演示了通过调用 bindPreviewDisplay 接口设置预览 View。

```
SurfaceView cameraView = (SurfaceView) findViewById(R.id.cameraView);
mLiveSession.bindPreviewDisplay(cameraView.getHolder());
```

启动音视频采集设备

在开始推流前，我们还需要启动音视频采集设备（即相机和 MIC）。

```
mLiveSession.prepareSessionAsync();
```

开始推流

当采集设备成功启动后（即收到 onSessionPrepared 回调后，参考[处理推流 SDK 状态变化事件](#)），可以通过调用 startRtmpSession 方法设置推流地址并开始推流。

以下示例代码使用 startRtmpSession 设置推流路径并开始推流，推流 URL 的具体值为：

```
rtmp://push.bj.bcelive.com/live/fjqgewdr17gquje235。
```

```
final String url1 = "rtmp://push.bj.bcelive.com/live/fjqgewdr17gquje235";
if (mLiveSession.startRtmpSession(url1)) {
    Log.d(TAG, "Starting Streaming in right state!");
} else {
    Log.e(TAG, "Starting Streaming in wrong state!");
}
```

暂停视频预览和推流

成功开启推流后，您随时可以暂停视频预览和推流。

以下示例代码使用 stopPreview 停止预览以及推流。

```
mLiveSession.stopPreview();
```

恢复视频预览和推流

暂停视频预览和推流后，您随时可以恢复视频预览和推流。

以下示例代码使用 startPreview 恢复预览以及推流。

```
mLiveSession.startPreview();
```

结束推流

成功开启推流后，您随时可以结束推流。

以下示例代码使用 stopRtmpSession 结束推流。

```
mLiveSession.stopRtmpSession();
```

处理推流SDK状态变化事件

在推流 SDK 中，我们定义了名为 SessionStateListener 的 interface，您可以实现该接口并将其传递给 LiveSession 对象（通过调用 setStateListener 接口实现），这样就可以实时接收到推流 SDK 的一些属性及状态更新信息。

interface 定义如下：

```

public interface SessionStateListener {

    /**
     * 录制设备准备完毕
     * @param code 固定为RESULT_CODE_OF_OPERATION_SUCCEEDED
     */
    void onSessionPrepared(int code);

    /**
     * 推流开始后的回调
     * @param code 固定为RESULT_CODE_OF_OPERATION_SUCCEEDED
     */
    void onSessionStarted(int code);

    /**
     * 推流结束后的回调
     * @param code 固定为RESULT_CODE_OF_OPERATION_SUCCEEDED
     */
    void onSessionStopped(int code);

    /**
     * 推流 SDK 出错后的回调
     * @param code 错误类型如下：
     *      ERROR_CODE_OF_OPEN_MIC_FAILED
     *      ERROR_CODE_OF_OPEN_CAMERA_FAILED
     *      ERROR_CODE_OF_PREPARE_SESSION_FAILED
     *      ERROR_CODE_OF_CONNECT_TO_SERVER_FAILED
     *      ERROR_CODE_OF_DISCONNECT_FROM_SERVER_FAILED
     *      ERROR_CODE_OF_UNKNOWN_STREAMING_ERROR
     *      ERROR_CODE_OF_WEAK_CONNECTION_ERROR
     *      ERROR_CODE_OF_SERVER_INTERNAL_ERROR
     *      ERROR_CODE_OF_LOCAL_NETWORK_ERROR
     */
    void onSessionError(int code);
}

```

注意：prepareSessionAsync、startRtmpSession、stopRtmpSession 均为异步接口，即接口被调用后，不会立即得到结果，需要实现 SessionStateListener interface 侦听回调结果。如果执行成功，onSessionPrepared、onSessionStarted、onSessionStopped 方法将分别被调用，且参数固定为 RESULT_CODE_OF_OPERATION_SUCCEEDED。如果执行失败，只有 onSessionError 方法将被调用，参数及意义如下：

```

ERROR_CODE_OF_OPEN_MIC_FAILED // MIC设备无法打开
ERROR_CODE_OF_OPEN_CAMERA_FAILED // 相机设备无法打开
ERROR_CODE_OF_PREPARE_SESSION_FAILED // onSessionPrepared 接口调用失败，原因只能是 MIC 或相机打开失败
ERROR_CODE_OF_CONNECT_TO_SERVER_FAILED // startRtmpSession 接口调用失败，原因通常为连接不上推流服务器
ERROR_CODE_OF_DISCONNECT_FROM_SERVER_FAILED // stopRtmpSession 接口调用失败，原因通常是网络异常

```

另外，在推流过程中，如果遇到数据包发送失败的情况时，onSessionError 方法也将被调用，参数及意义如下：

```

ERROR_CODE_OF_UNKNOWN_STREAMING_ERROR // 未知的推流错误，建议收到此参数后立即结束推流
ERROR_CODE_OF_WEAK_CONNECTION_ERROR // 弱网错误，建议收到此参数后提示用户网络不稳定
ERROR_CODE_OF_SERVER_INTERNAL_ERROR // 服务器错误，建议收到此参数后立即结束推流，并在服务器恢复后再尝试推流
ERROR_CODE_OF_LOCAL_NETWORK_ERROR // 本地网络连接错误，建议收到此参数后提示用户已经断网

```

以下示例代码演示了如何实现 SessionStateListener interface 对 LiveSession 状态变化事件进行处理：

```

// 其中 mEventHandler 为各 UI 事件处理器

```

```
mStateListener = new SessionStateListener() {
    @Override
    public void onSessionPrepared(int code) {
        if (code == SessionStateListener.RESULT_CODE_OF_OPERATION_SUCCEEDED) {
            isSessionReady = true;
        }
    }

    @Override
    public void onSessionStarted(int code) {
        if (code == SessionStateListener.RESULT_CODE_OF_OPERATION_SUCCEEDED) {
            Log.d(TAG, "Starting Streaming succeeded!");
            isSessionStarted = true;
            if (mEventHandler != null) {
                mEventHandler.sendEmptyMessage(EVENT_RECORDER_STARTED);
            }
        } else {
            Log.e(TAG, "Starting Streaming failed!");
        }
    }

    @Override
    public void onSessionStopped(int code) {
        if (code == SessionStateListener.RESULT_CODE_OF_OPERATION_SUCCEEDED) {
            Log.d(TAG, "Stopping Streaming succeeded!");
            isSessionStarted = false;
            if (mEventHandler != null) {
                mEventHandler.sendEmptyMessage(EVENT_RECORDER_STOPPED);
            }
        } else {
            Log.e(TAG, "Stopping Streaming failed!");
        }
    }

    @Override
    public void onSessionError(int code) {
        switch (code) {
            case SessionStateListener.ERROR_CODE_OF_OPEN_MIC_FAILED:
                Log.e(TAG, "Error occurred while opening MIC!");
                onOpenDeviceFailed();
                break;
            case SessionStateListener.ERROR_CODE_OF_OPEN_CAMERA_FAILED:
                Log.e(TAG, "Error occurred while opening Camera!");
                onOpenDeviceFailed();
                break;
            case SessionStateListener.ERROR_CODE_OF_PREPARE_SESSION_FAILED:
                Log.e(TAG, "Error occurred while preparing recorder!");
                onPrepareFailed();
                break;
            case SessionStateListener.ERROR_CODE_OF_CONNECT_TO_SERVER_FAILED:
                Log.e(TAG, "Error occurred while connecting to server!");
                // As we can not start session successfully, we need to take it as stopped
                onStartOrStopFailed();
                break;
            case SessionStateListener.ERROR_CODE_OF_DISCONNECT_FROM_SERVER_FAILED:
                Log.e(TAG, "Error occurred while disconnecting from server!");
                // Although we can not stop session successfully, we still need to take it as stopped
                onStartOrStopFailed();
                break;
            case SessionStateListener.ERROR_CODE_OF_WEAK_CONNECTION_ERROR:
                Log.e(TAG, "Your network connection is too weak to streaming!");
                onStreamingError();
                break;
        }
    }
}
```

```

        break;
    case SessionStateListener.ERROR_CODE_OF_SERVER_INTERNAL_ERROR:
        Log.e(TAG, "There is something wrong with the server!");
        onStreamingError();
        break;
    case SessionStateListener.ERROR_CODE_OF_LOCAL_NETWORK_ERROR:
        Log.e(TAG, "Connection Error while Streaming! Please check your network!");
        onStreamingError();
        break;
    default:
        if (mEventHandler != null) {
            mEventHandler.sendMessage(EVENT_RECORDER_UNKNOWN_ERROR);
            onStreamingError();
        }
        break;
    }
}
};

```

推流效果控制

在推流过程中，你还可以通过我们的接口灵活控制推流过程中的音视频效果，详情如下：

- 切换摄像头

推流过程中，您可以通过调用 switchCamera 接口来动态切换摄像头，该函数原型如下：

```

void switchCamera(int cameraId);
// 作用：使用 cameraId 所代表的摄像头采集视频
// 参数：cameraId 只能为 Camera.CameraInfo.CAMERA_FACING_FRONT 或者
Camera.CameraInfo.CAMERA_FACING_BACK
// 注意：使用该接口前，您需要调用 canSwitchCamera 接口来判断当前设备是否支持切换摄像头，如果不支持，您将无法开始设备，并收到对应的错误消息 ERROR_CODE_OF_OPEN_CAMERA_FAILED

```

- 判断是否支持切换摄像头

推流过程中，您可以通过调用 canSwitchCamera 接口来判断设备是否支持动态切换摄像头，该函数原型如下：

```

boolean canSwitchCamera();
// 作用：判断设备是否支持动态切换摄像头
// 返回值：返回值为 true 表示支持切换，否则不支持切换

```

- 开关闪光灯

推流过程中，您可以通过调用 toggleFlash 接口来动态开关闪光灯，该函数原型如下：

```

void toggleFlash(boolean flag);
// 作用：根据传入的参数值开关闪光灯
// 参数：flag 为 true 时表示开启闪光灯，为 false 时表示关闭闪光灯
// 注意：使用该接口前，您需要确认此时的摄像头是拥有闪光灯，如果没有（前置摄像头通常不具备闪光灯），调用该接口将无法开启闪光灯

```

- 触发自动对焦

推流过程中，您可以通过调用 focusToPosition 接口来触发自动对焦，该函数原型如下：

```

void focusToPosition(int x, int y);
// 作用：触发摄像头自动对焦功能
// 参数：(x, y) 为对焦中心点在屏幕上的坐标

```

- 放大视频画面

推流过程中，您可以通过调用 `zoomInCamera` 接口来放大当前视频画面，该函数原型如下：

```
void zoomInCamera();  
// 作用：将摄像头所采集的当前视频画面放大一级（视频原始大小被定义为第0级）  
// 注意：视频被放大后，高宽像素保持不变，超出画面的部分将裁剪掉；如果放大级别超过了最大级别（通过  
getMaxZoomFactor 接口获取），调用该接口将不起作用
```

- 缩小视频画面

推流过程中，您可以通过调用 `zoomOutCamera` 接口来缩小当前视频画面，该函数原型如下：

```
void zoomOutCamera();  
// 作用：将摄像头所采集的当前视频画面缩小一级  
// 注意：视频被缩小后，高宽像素保持不变；如果当前放大级别（通过 getCurrentZoomFactor 接口获取）为第0级，调用  
该接口将不起作用
```

- 指定视频画面放大级别

推流过程中，您可以通过调用 `setCameraZoomLevel` 接口来指定当前视频画面的放大级别，该函数原型如下：

```
void setCameraZoomLevel(int zoomlevel);  
// 作用：将摄像头所采集的当前视频画面放大到 zoomlevel 所指定的级别  
// 注意：zoomlevel 的取值不能小于0，也不能大于最大级别（通过 getMaxZoomFactor 接口获取），否则该接口将不起作用
```

- 取消视频画面放大效果

推流过程中，您可以通过调用 `cancelZoomCamera` 接口来取消当前视频画面的放大效果，该函数原型如下：

```
void cancelZoomCamera();  
// 作用：取消当前视频画面的放大效果
```

- 使用美颜功能（仅当硬编码时有效）

收到推流 SDK 的 `onSessionPrepared` 回调后，即可使用 `enableDefaultBeautyEffect` 和 `setBeautyEffectLevel` 两个接口设置美颜效果。

- `setBeautyEffectLevel`：用于设置美颜效果，参数为 0 表示没有效果，参数为 1 表示最大效果。调用该接口后美颜效果会立即生效，无需再次调用 `enableBeautyEffect`。函数原型如下：

```
/**  
 * 开启或关闭默认美颜效果  
 * 注意：此方法只对LiveSessionHW对象有效  
 * @param isEnabled: true 则开启默认美颜效果； false 则关闭所有美颜效果（包括自定义的美颜效果）  
 */  
public abstract void enableDefaultBeautyEffect(boolean isEnabled);
```

- `enableDefaultBeautyEffect`：用于开启或关闭美颜效果。开启前若未使用 `setBeautyEffectLevel` 接口设置过美颜参数，则使用默认美颜参数；否则使用最近设置的美颜参数。函数原型如下：

```

/**
 * 设置自定义美颜效果，设置后立即开启
 * 注意：此方法只对LiveSessionHW对象有效
 * @param brightLevel: 美白级别，取值范围为[0, 1]，0为无效果，1为最大效果
 * @param smoothLevel: 磨皮级别，取值范围为[0, 1]，0为无效果，1为最大效果
 * @param pinkLevel: 粉嫩级别，取值范围为[0, 1]，0为无效果，1为最大效果
 */
public abstract void setBeautyEffectLevel(float brightLevel, float smoothLevel, float pinkLevel);

```

- 控制音视频输出

推流开始以后，可以通过调用 `setAudioEnabled` 和 `setVideoEnabled` 来控制是否推送音频流或者视频流。默认音视频数据都推送。

`setAudioEnabled` 函数原型如下：

```

/**
 * 开启或关闭音频推流接口
 * @param isEnabledAudio
 */
public abstract void setAudioEnabled(boolean isEnabledAudio);

```

`setVideoEnabled` 函数原型如下：

```

/**
 * 开启或关闭视频推流接口
 * @param isEnabledVideo
 */
public abstract void setVideoEnabled(boolean isEnabledVideo);

```

iOS端推流SDK

简介

百度智能云 [iOS 端推流 SDK](#) (以下简称“推流 SDK”) 是百度官方推出的 iOS 平台用于推流的软件开发工具包，为您提供简单、便捷的开发接口，助您快速实现音视频直播推流功能。

优势

- 支持 GPU 硬件加速，CPU 占用低，能明显增加手机续航时间
- 压缩比例大，能对视频进行 High Profile 的H264编码
- 支持高斯模糊等多种实时滤镜效果
- 支持视频图片水印功能
- 良好的向上兼容，最低可支持 iOS 8.0 版本

特点

- **支持主流 RTMP 服务器**
支持目前所有主流的 RTMP 服务器 (simple-rtmp-server、crtmpserver、FMS 等)。
- **支持所有的 RTMP 协议**
支持所有 RTMP 协议及变种 (RTMP、RTMPT、RTMPE、RTMPS、RTMPTE、RTMPTS 等)。
- **性能强大**

资源 (CPU/内存) 占用率低, 使用硬件加速进行编码, 压缩比例大, 延时小。

- **支持实时滤镜**

支持多种实时滤镜效果 (Gray、InvertColors、Sepia、Fisheye、Glow、GaussianBlur)。

- **完备的编程接口提供极大的灵活性**

通过提供完备的编程接口, 方便您在采集音视频过程中对相机的对焦模式、测光点、焦点等参数进行灵活控制。同时提供开发示例, 为您说明各接口调用细节。

- **高兼容性、稳定性和快速响应**

支持 iOS 8.0 及以上平台, 结合百度开发云直播服务, 可实现端到端平均仅2秒延时的实时直播功能。

🔗 组件及资源

推流 SDK 的完整下载包中包含 sdk 和 LiveDemo 两部分:

SDK 目录

存放推流 SDK 的动态库 framework, 有用于开发和发布的2个版本, 包含的 arch 有 armv7、arm64、i386、x86_64。

LiveDemo 目录

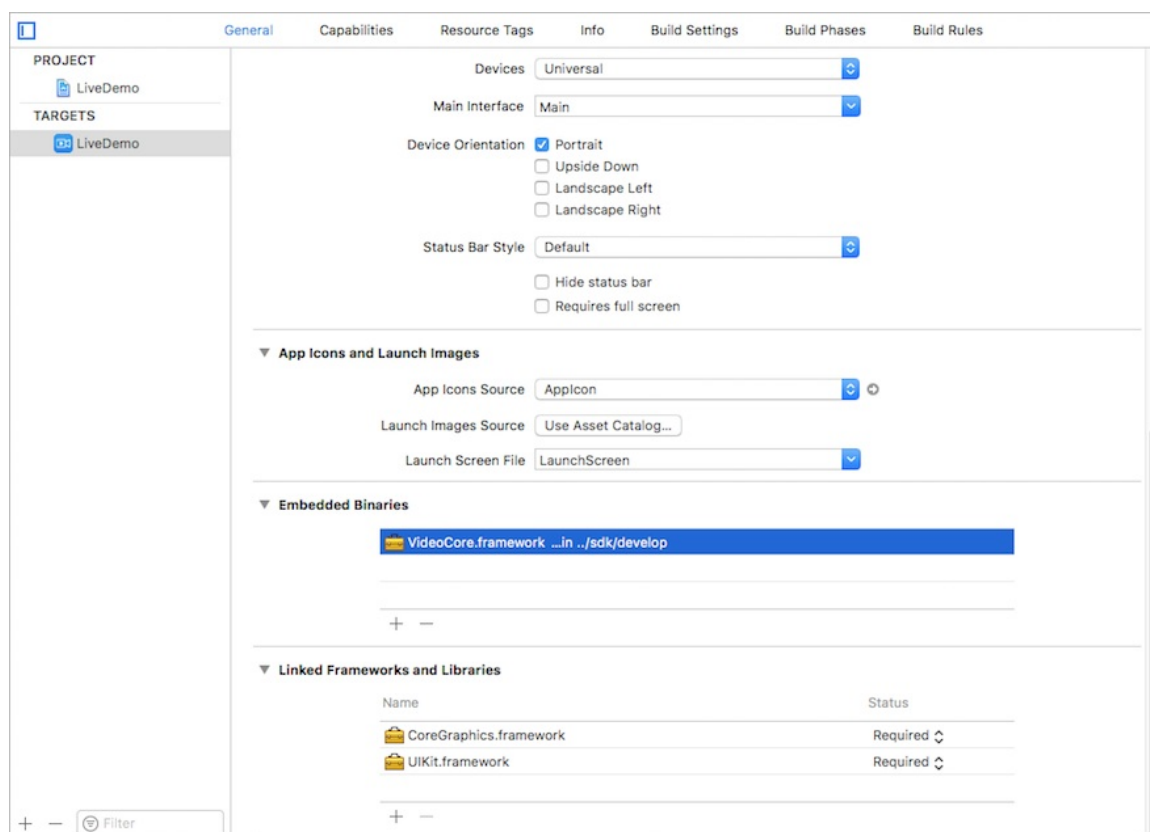
存放 iOS 示例工程, 展示了如何使用 VCSimpleSession 来实现 iOS 端推流。

推流-SDK-开发指南

****准备工作****

- 开通 [百度智能云音视频直播服务](#), 添加推拉流域名并获取推流地址及观看地址。
- 登录百度智能云官网, 在[安全认证页面](#) 获取 [AK/SK](#)。
- 下载最新的[iOS 端推流 SDK](#)。
- 准备 iOS 运行环境: iOS 8.0 及以上的所有系统。
- Xcode版本: 推荐使用Xcode 8.1。
- 适配 CPU 指令集: armv7/armv7s、arm64、i386/x86_64。
- 项目配置:
 - (1) 支持cocoapods接入方式, 具体请参见[Github百度开发云页面](#)。
 - (2) 将Baidu-Capture-SDK-iOS-x.x.x/sdk 目录下的 VideoCore.framework 添加到 Xcode 工程。
 - (3) 添加 VideoCore.framework 到 Embed Frameworks 中。

如下图所示:



**** 初始化 VCSimpleSession****

推荐使用 VCSimpleSessionConfiguration 类来进行初始化（原有的初始化接口已被废弃，在未来的版本中将被删除），具体用法请参考如下代码段：

```
VCSimpleSessionConfiguration* configuration = [[VCSimpleSessionConfiguration alloc] init];
configuration.cameraOrientation = AVCaptureVideoOrientationPortrait;
configuration.videoSize = CGSizeMake(720, 1280);
configuration.bitrate = 1200 * 1000;
configuration.cameraDevice = VCCameraStateFront;
...

self.session = [[VCSimpleSession alloc] initWithConfiguration:configuration];
```

参数详情请参见[参数控制](#)。

**** 开始推流****

初始化成功后，可以通过 startRtmpSessionWithURL 方法设置推流地址并开始推流。

以下示例代码使用 startRtmpSessionWithURL 设置推流路径并开始推流，推流 URL 的具体值为：

rtmp://push.bj.bcelive.com/live/fjqgewdr17gquje235。

```
[session startRtmpSessionWithURL:@"rtmp://push.bj.bcelive.com/live/" andStreamKey:@"fjqgewdr17gquje235"];
或者
[session startRtmpSessionWithURL:@"rtmp://push.bj.bcelive.com/live/fjqgewdr17gquje235"];
```

**** 结束推流****

开始推流成功后，您随时可以结束推流。

以下示例代码使用 endRtmpSession 结束推流。

```
[session endRtmpSession];
```

*****## 监听推流过程中的通知 (v2.0.0版本后失效) *****

- 侦听 name 为 RTMP_Started 的系统消息以获取推流开始的事件

示例代码：

```
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(handleRTMPStartedMessage:)
name:RTMP_Started object:nil];

(void) handleRTMPStartedMessage:(NSNotification*)notification { NSLog(@"RTMP Streaming was started..\n"); }
```

- 侦听 name 为 RTMP_Error 的系统消息以获取推流异常的事件

示例代码：

```
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(handleRTMPErrorMessage:) name:RTMP_Error
object:nil];

(void) handleRTMPErrorMessage:(NSNotification*)notification { NSLog(@"Error occurred in Streaming..\n"); }
```

注意：在收到 RTMP_Started 消息之前，请勿调用 endRtmpSession，否则将造成资源释放异常。推荐侦听状态变化事件代替通知事件。

*****## 处理推流 SDK 状态变化事件****

在 VCSimpleSession 接口中，您可以通过实现名为 VCSessionDelegate 的 protocol，实时接收到推流 SDK 的一些属性及状态更新信息，其中 onError 接口用于在发生错误时通知用户。

protocol 定义如下：

```
@protocol VCSessionDelegate <NSObject>
@required
- (void) connectionStatusChanged: (VCSessionState) sessionState;
@optional
- (void) didAddCameraSource:(VCSimpleSession*)session;
- (void) onError:(VCErrCode)error;
@end
```

以下示例代码演示了如何实现 protocol 对 session 状态变化事件进行处理：

```
- (void) connectionStatusChanged:(VCSessionState) state
{
    switch(state) {
        case VCSessionStateStarting:
            NSLog(@"Current state is VCSessionStateStarting\n");
            [self.btnConnect setImage:[UIImage imageNamed:@"block.png"] forState:UIControlStateNormal];
            break;
        case VCSessionStateStarted:
            NSLog(@"Current state is VCSessionStateStarted\n");
            [self.btnConnect setImage:[UIImage imageNamed:@"to_stop.png"] forState:UIControlStateNormal];
            break;
        case VCSessionStateError:
            NSLog(@"Current state is VCSessionStateError\n");
            [self.btnConnect setImage:[UIImage imageNamed:@"to_start.png"] forState:UIControlStateNormal];
            break;
        default:
            NSLog(@"Current state is VCSessionStateEnded\n");
            [self.btnConnect setImage:[UIImage imageNamed:@"to_start.png"] forState:UIControlStateNormal];
            break;
    }
}
```

注意：在 protocol 定义中，connectionStatusChanged 为必选方法，当推流 SDK 状态发生改变时，该方法会被调用，参数

sessionState 即为当前推流 session 所处的状态。

sessionState 参数为 VCSessionState 类型，定义如下：

```
typedef NS_ENUM(NSInteger, VCSessionState)
{
    VCSessionStateNone, //推流 SDK 的初始状态
    VCSessionStatePreviewStarted, //推流 SDK 开始出现预览画面
    VCSessionStateStarting, //推流 SDK 开始连接服务器
    VCSessionStateStarted, //推流已经开始
    VCSessionStateEnded, //推流已经结束
    VCSessionStateError //推流 SDK 出错
};
```

在状态 VCSessionStateStarting 与 VCSessionStateStarted 之间，推流 SDK 正在后台连接 RTMP 服务器，此时请勿对 VCSimpleSession 对象进行任何操作，特别是请勿调用 endRtmpSession，否则将造成资源释放异常。状态变为 VCSessionStatePreviewStarted 后，方可使用美颜接口开启美颜功能。

其中，didAddCameraSource 为可选方法。

当推流 SDK 创建 CameraSource（即相机被占用）后，didAddCameraSource 方法会被调用，参数 session 为 VCSimpleSession 对象。

***## 使用美颜功能**

当推流 SDK 的状态变为 VCSessionStatePreviewStarted 时，可以使用 setBeautyEffect 和 enableBeautyEffect 两个接口设置美颜效果。

- 设置美颜级别 beautyLevel: 预设美颜的一些参数，分为5个级别：

```
/**
 * @brief 美颜级别定义
 */
typedef NS_ENUM(NSInteger, VCBeautyLevel) {
    /**
     * 不开启美颜。
     */
    VCBeautyLevelNone,
    /**
     * 自然。基本去皱，适当美白提亮
     */
    VCBeautyLevelNatural,
    /**
     * 美白。去皱但不失真，以美白为主粉嫩为辅
     */
    VCBeautyLevelWhiten,
    /**
     * 粉嫩。去皱但不失真，以粉嫩为主美白为辅
     */
    VCBeautyLevelPink,
    /**
     * 梦幻。最大值
     */
    VCBeautyLevelMagic
};
```

- 设置具体的美颜参数 setBeautyEffect：用于设置美颜效果，参数为 0 表示没有效果，参数为 1 表示最大效果。调用该接口后美颜效果会立即生效，无需再次调用 enableBeautyEffect。接口定义如下：

```
/**
 * 设置美颜参数
 * bright 美白参数，取值范围[0, 1]
 * smooth 磨皮参数，取值范围[0, 1]
 * pink 粉嫩参数，取值范围[0, 1]
 */
- (void) setBeautyEffect:(float) bright
    withSmooth:(float) smooth
    withPink:(float) pink;
```

- enableBeautyEffect：用于开启或关闭美颜效果。开启前若未使用 setBeautyEffect 接口设置过美颜参数，则使用默认美颜参数；否则使用最近设置的美颜参数。接口定义如下：

```
// 开启或关闭美颜效果
- (void) enableBeautyEffect:(BOOL) isEnabled;
```

****## 参数控制****

推流 SDK 提供丰富的参数供您灵活控制推流过程中的音视频效果，详情如下：

参数名	参数类型	含义	推流后是否可变	默认值	取值范围
videoSize	CGSize	视频分辨率	否	无(必须初始化)	宽高均大于0
bitrate	int	视频编码码率(单位为bps)	否	无(必须初始化)	大于100000
fps	int	视频采集帧率	否	无(必须初始化)	大于1，小于等于30
useInterfaceOrientation	BOOL	是否使用应用的垂直方向作为视频的垂直方向	否	NO	YES, NO
cameraState	VCCameraState	视频采集所用摄像头种类	是(调用switchCamera)	VCCameraStateBack	VCCameraStateFront, VCCameraStateBack
orientationLocked	BOOL	是否锁定视频正方向	是	YES	YES, NO
torch	BOOL	是否开启闪光灯	是	NO	YES, NO
videoZoomFactor	float	视频缩放比例	是	1.0	大于0，小于等于1.0
audioChannelCount	int	音频采集编码过程中所使用的声道数	否	2	1, 2
audioSampleRate	float	音频采集编码过程中所使用的采样率	否	44100.0	44100.0, 22050.0
micGain	float	音量增益因子	是	1.0	大于等于0，小于等于1.0
focusPointOfInterest	CGPoint	自动对焦时摄像头的对焦焦点	是	(0.5, 0.5)	(0, 0)到(1, 1)，(0, 0)表示左上角，(1, 1)表示右下角
exposurePointOfInterest	CGPoint	摄像头的测光中心点	是	(0.5, 0.5)	(0, 0)到(1, 1)，(0, 0)表示左上角，(1, 1)表示右下角
continuousAutofocus	BOOL	是否开启自动对焦	是	YES	YES, NO
continuousExposure	BOOL	是否开启动态测光	是	YES	YES, NO
useAdaptiveBitrate	BOOL	是否开启可变码率选项(暂不支持)	否	NO	YES, NO
aspectMode	VCAsspectMode	视频画面对窗口的填充方式	是	VCAsspectModeFit	VCAsspectModeFit, VCAsspectModeFill
filter	VCFilter	实时滤镜效果	是	VCFilterNormal	VCFilterNormal, VCFilterGray, VCFilterInvertColors, VCFilterSepia, VCFilterFisheye, VCFilterGlow, VCFilterGussBlur

播放器SDK

百度智能云提供Web、Android及iOS平台的播放器SDK，为开发者提供简单、便捷的开发接口，帮助开发者在各类终端设备上实现媒体播放功能。

- Web平台
 - [播放器Web SDK帮助手册](#)
 - [播放器Web SDK下载](#)
- Android平台
 - [播放器Android SDK帮助手册](#)
 - [播放器Android SDK下载](#)
- iOS平台
 - [播放器iOS SDK帮助手册](#)
 - [播放器iOS SDK下载](#)

服务端SDK

Java-SDK

安装LSS-Java-SDK

直接使用JAR包

操作步骤如下：

1. 下载最新版[Java SDK](#)压缩工具包。
2. 将下载的bce-java-sdk-version.zip解压后，复制到工程文件夹中。
3. 在Eclipse右键“工程 -> Properties -> Java Build Path -> Add JARs”。
4. 添加SDK工具包lib/bce-java-sdk-{version}.jar和第三方依赖工具包third-party/*.jar。

其中，{version}为版本号，添加完成后，用户就可以在工程中使用LSS Java SDK。

使用Maven

操作步骤如下：

1. 在[Maven官网](#)搜索bce-java-sdk，进入最新版sdk页面。
2. 将Dependency Information中的Apache Maven代码添加到工程的pom.xml文件中。

LSS Java SDK目录结构

com.baidubce	
├── auth	//BCE签名相关类
├── http	//BCE的Http通信相关类
├── internal	//SDK内部类
├── model	//BCE公用model类
├── services	
│ ├── lss	//LSS服务相关类
│ ├── model	//LSS内部model，如Request或Response
│ └── LssClient.class	//LSS客户端入口类
├── util	//BCE公用工具类
├── BceClientConfiguration.class	//对BCE的HttpClient的配置
├── BceClientException.class	//BCE客户端的异常类
├── BceServiceException.class	//与BCE服务端交互后的异常类
├── ErrorCode.class	//BCE通用的错误码
└── Region.class	//BCE提供服务的区域

LssClient

新建LssClient

LssClient是Live Streaming Service的Java客户端，为使用Live Streaming Service的开发人员提供了一系列的交互方法，用户可以参考如下代码新建一个LssClient：

```
public class Sample {
    public static void main(String[] args) {

        //设置AK/SK
        String ACCESS_KEY_ID = "your-access-key-id";
        String SECRET_ACCESS_KEY = "your-secret-access-key";

        // 初始化一个LssClient
        BceClientConfiguration config = new BceClientConfiguration();
        config.setCredentials(new DefaultBceCredentials(ACCESS_KEY_ID, SECRET_ACCESS_KEY));
        LssClient client = new LssClient(config);

    }
}
```

在上面代码中，变量ACCESS_KEY_ID与SECRET_ACCESS_KEY是由系统分配给用户的，均为字符串，用于标识用户，为访问LSS做签名验证。其中ACCESS_KEY_ID对应控制台中的“Access Key ID”，SECRET_ACCESS_KEY对应控制台中的“Access Key Secret”，获取方式请参考[获取AK/SK](#)。

上面的方式使用默认域名作为LSS的服务地址，如果用户需要自己制定域名，可以通过传入ENDPOINT参数来指定。

```
String ACCESS_KEY_ID = "your-access-key-id";
String SECRET_ACCESS_KEY = "your-secret-access-key";
String ENDPOINT = "http://lss.bj.baidubce.com";

BceClientConfiguration config = new BceClientConfiguration();
config.setCredentials(new DefaultBceCredentials(ACCESS_KEY_ID, SECRET_ACCESS_KEY));
config.setEndpoint(ENDPOINT);
LssClient client = new LssClient(config);
```

配置LssClient

如果用户需要配置LssClient的一些细节的参数，可以在构造LssClient的时候传入BceClientConfiguration对象。

BceClientConfiguration是LSS服务的配置类，可以为客户端配置代理、最大连接数等参数。

使用代理

下面一段代码可以让客户端使用代理访问LSS服务：

```
String ACCESS_KEY_ID = "your-access-key-id";
String SECRET_ACCESS_KEY = "your-secret-access-key";
String ENDPOINT = "http://lss.bj.baidubce.com";

// 创建BceClientConfiguration实例
BceClientConfiguration config = new BceClientConfiguration();

// 配置代理为本地8080端口
config.setProxyHost("127.0.0.1");
config.setProxyPort(8080);

// 配置认证秘钥和服务器信息
config.setCredentials(new DefaultBceCredentials(ACCESS_KEY_ID, SECRET_ACCESS_KEY));
config.setEndpoint(ENDPOINT);

// 创建LSS客户端
LssClient client = new LssClient(config);
```

使用上面的代码段，客户端的所有操作都会通过127.0.0.1地址的8080端口做代理执行。

对于有用户验证的代理，可以通过下面的代码段配置用户名和密码：

```
// 创建BceClientConfiguration实例
BceClientConfiguration config = new BceClientConfiguration();

// 配置代理为本地8080端口
config.setProxyHost("127.0.0.1");
config.setProxyPort(8080);

// 设置用户名和密码
config.setProxyUsername("username");
config.setProxyPassword("password");
```

设置网络参数

用户可以用BceClientConfiguration对基本网络参数进行设置：

```
BceClientConfiguration config = new BceClientConfiguration();

// 设置HTTP最大连接数为10
config.setMaxConnections(10);

// 设置TCP连接超时为5000毫秒
config.setConnectionTimeout(5000);

// 设置Socket传输数据超时的时间为2000毫秒
config.setSocketTimeout(2000);
```

BceClientConfiguration参数说明

通过BceClientConfiguration指定的所有参数如下表所示：

参数	说明
UserAgent	用户代理，指HTTP的User-Agent头
Protocol	连接协议类型
ProxyDomain	访问NTLM验证的代理服务器的Windows域名
ProxyHost	代理服务器主机地址
ProxyPort	代理服务器端口
ProxyUsername	代理服务器验证的用户名
ProxyPassword	代理服务器验证的密码
ProxyPreemptiveAuthenticationEnabled	是否设置用户代理认证
ProxyWorkstation	NTLM代理服务器的Windows工作站名称
LocalAddress	本地地址
ConnectionTimeoutInMillis	建立连接的超时时间（单位：毫秒）
SocketTimeoutInMillis	通过打开的连接传输数据的超时时间（单位：毫秒）
MaxConnections	允许打开的最大HTTP连接数
RetryPolicy	连接重试策略
SocketBufferSizeInBytes	Socket缓冲区大小

stream管理

createStream

在特定Domain下新建Stream。

```
CreateStreamRequest request = new CreateStreamRequest();
request.withPlayDomain(yourPlayDomain) // yourPlayDomain是用户提前创建的播放域名
.withApp(yourApp) // yourApp是用户自定义的app名称
.withPublish(new CreateStreamRequest.PublishInfo().withPushStream(yourStream)); // yourStream是用户自定义的
Stream名称
CreateStreamResponse response = IssClient.createStream(request);
```

listStream

查询特定Domain下的所有Stream信息。

```
ListStreamRequest request = new ListStreamRequest();
String status = "READY";
request.withPlayDomain(yourPlayDomain).withMaxSize(10).withStatus(status); // status表示Stream的状态，有效值有
READY、ONGOING、PAUSED
ListStreamResponse response = IssClient.listStream(request);
```

listDomainApp

查询特定Domain下所有App（不包括默认App）。

```
ListDomainAppRequest request = new ListDomainAppRequest();
request.setPlayDomain(yourPlayDomain);
ListDomainAppResponse response = IssClient.listDomainApp(request);
```

getStream

查询特定Domain下的特定Stream信息。

```
GetStreamRequest request = new GetStreamRequest();
request.withPlayDomain(yourPlayDomain).withApp(yourApp).withStream(yourStream);
GetStreamResponse response = IssClient.getStream(request);
```

🔗 pauseDomainStream

封禁特定domain下的特定stream。

```
PauseDomainStreamRequest request = new PauseDomainStreamRequest();
request.withDomain(yourDomain).withApp(yourApp).withStream(yourStream);
IssClient.pauseDomainStream(request);
```

🔗 resumeDomainStream

解禁特定Domain下的特定Stream。

```
ResumeDomainStreamRequest request = new ResumeDomainStreamRequest();
request.withDomain(yourDomain).withApp(yourApp).withStream(yourStream);
IssClient.resumeDomainStream(request);
```

🔗 deleteStream

删除特定stream

```
DeleteStreamRequest request = new DeleteStreamRequest()
    .withDomain(domain)
    .withApp(app)
    .withStream(stream);
IssClient.deleteStream(request);
```

🔗 updateStreamWatermark

更新特定stream的水印模板

```
Watermarks watermarks = new Watermarks(); //如果watermarks为null，将继承域名下的水印模板配置。
watermarks.setImage(Collections.singletonList("lss_sdk_java_imw"));
watermarks.setTimestamp(Collections.singletonList("lss_sdk_java_tsw"));
UpdateStreamWatermarkRequest request = new UpdateStreamWatermarkRequest()
    .withDomain(domain)
    .withApp(app)
    .withStream(stream)
    .withWatermarks(watermarks);
IssClient.updateStreamWatermark(request);
```

🔗 updateStreamRecording

更新特定stream的录制模板

```
String recording = "lss_java_sdk_m3u8"; //如果recording为null，将继承域名下的录制模板配置。
UpdateStreamRecordingRequest request = new UpdateStreamRecordingRequest()
    .withDomain(domain)
    .withApp(app)
    .withStream(stream)
    .withRecording(recording);
IssClient.updateStreamRecording(request);
```

🔗 updateStreamPresets

更新特定stream的转码模板

```
Map<String, String> presets = new HashMap<String, String>();  
// 这里不能更新LO对应的模板  
// 如果presets为null，将继续域名下的转码模板配置。  
presets.put("L1", "lss.lss_960x720");  
presets.put("L2", "lss.lss_1280x720");  
UpdateStreamPresetsRequest request = new UpdateStreamPresetsRequest()  
    .withDomain(domain)  
    .withApp(app)  
    .withStream(stream)  
    .withPresets(presets);  
IssClient.updateStreamPresets(request);
```

🔗 updateDestinationPushUrl

更新特定stream目标推流地址

```
String destinationPushUrl = "rtmp://push.testsdk.com";  
UpdateStreamDestinationPushUrlRequest request = new UpdateStreamDestinationPushUrlRequest()  
    .withDomain(domain)  
    .withApp(app)  
    .withStream(stream)  
    .withDestinationPushUrl(destinationPushUrl);  
IssClient.updateStreamDestinationPushUrl(request);
```

统计功能

查询统计相关的请求和响应参数，请参考[统计接口](#)。

🔗 getDomainStatistics

查询特定Domain的统计数据。

```
GetDomainStatisticsRequest request = new GetDomainStatisticsRequest();  
request.withDomain(yourDomain).withStartDate("20161018").withEndDate("20161019").withAggregate(Boolean.TRUE);  
GetDomainStatisticsResponse response = IssClient.getDomainStatistics(request);
```

🔗 getDomainSummaryStatistics

查询当前用户所有Domain的统计概要。

```
GetDomainSummaryStatisticsRequest request = new GetDomainSummaryStatisticsRequest();  
request.withStartTime("2016-10-18T01:00:00Z").withEndTime("2016-10-19T00:00:00Z");  
GetDomainSummaryStatisticsResponse response = IssClient.getDomainSummaryStatistics(request);
```

🔗 getAllDomainsPlayCount

查询当前用户所有Domain的总请求数。

```
GetAllDomainsStatisticsRequest request = new GetAllDomainsStatisticsRequest();  
request.withStartTime("2016-10-17T16:00:00Z").withEndTime("2016-10-19T00:00:00Z").withTimeInterval("LONG_TERM");  
GetAllDomainsPlayCountResponse response = IssClient.getAllDomainsPlayCount(request);
```

🔗 getOneDomainPlayCount

查询特定Domain的请求数。

```
GetOneDomainStatisticsRequest request = new GetOneDomainStatisticsRequest();
request.withDomain(domain).withStartTime("2016-09-18T00:00:00Z")
.withEndTime("2016-09-18T08:00:00Z").withTimeInterval("MID_TERM");
GetOneDomainPlayCountResponse response = IssClient.getOneDomainPlayCount(request);
```

getAllDomainsBandwidth

查询当前用户所有Domain的总带宽。

```
GetAllDomainsStatisticsRequest request = new GetAllDomainsStatisticsRequest();
request.withStartTime("2016-10-18T00:00:00Z")
.withEndTime("2016-10-18T00:20:00Z").withTimeInterval("SHORT_TERM");
GetAllDomainsBandwidthResponse response = IssClient.getAllDomainsBandwidth(request);
```

getOneDomainBandwidth

查询特定Domain的带宽。

```
GetOneDomainStatisticsRequest request = new GetOneDomainStatisticsRequest();
request.withDomain(domain).withStartTime("2016-10-18T00:00:00Z")
.withEndTime("2016-10-18T00:20:00Z").withTimeInterval("SHORT_TERM");
GetOneDomainBandwidthResponse response = IssClient.getOneDomainBandwidth(request);
```

getAllDomainsBandwidthV2

查询当前用户所有Domain的总带宽。

```
GetAllDomainsStatisticsRequest request = new GetAllDomainsStatisticsRequest();
request.withStartTime("2016-10-18T00:00:00Z")
.withEndTime("2016-10-18T00:20:00Z").withTimeInterval("SHORT_TERM").withAllDomain("true");
GetAllDomainsBandwidthResponse response = IssClient.getAllDomainsBandwidth(request);
```

getOneDomainBandwidthV2

查询特定Domain的带宽。

```
GetOneDomainStatisticsRequest request = new GetOneDomainStatisticsRequest();
request.withDomain(domain).withStartTime("2016-10-18T00:00:00Z")
.withEndTime("2016-10-18T00:20:00Z").withTimeInterval("SHORT_TERM");
GetOneDomainBandwidthResponse response = IssClient.getOneDomainBandwidth(request);
```

getAllDomainsTraffic

查询当前用户所有Domain的总流量。

```
GetAllDomainsStatisticsRequest request = new GetAllDomainsStatisticsRequest();
request.withStartTime("2016-10-17T16:00:00Z").withEndTime("2016-10-20T04:00:00Z").withTimeInterval("LONG_TERM");
GetAllDomainsTrafficResponse response = IssClient.getAllDomainsTraffic(request);
```

getOneDomainTraffic

查询特定Domain的流量。

```
GetOneDomainStatisticsRequest request = new GetOneDomainStatisticsRequest();
request.withDomain(domain).withStartTime("2016-10-18T04:00:00Z").withTimeInterval("LONG_TERM");
GetOneDomainTrafficResponse response = lssClient.getOneDomainTraffic(request);
```

listDomainStatistics

查询当前用户的所有Domain数据。

```
ListDomainStatisticsRequest request = new ListDomainStatisticsRequest();
request.withStartTime("2016-10-18T04:00:00Z")
.withEndTime("2016-10-20T04:00:00Z")
.withKeywordType("domain")
.withKeyword("test")
.withOrderBy("duration desc");
ListDomainStatisticsResponse response = lssClient.listDomainStatistics(request);
```

listStreamStatistics

查询当前用户特定Domain下的所有Stream数据。

```
ListStreamStatisticsRequest request = new ListStreamStatisticsRequest();
request.withDomain(domain).withApp("test")
.withStartTime("2016-10-18T04:00:00Z").withEndTime("2016-10-20T04:00:00Z")
.withKeyword("stream").withOrderBy("downstream")
.withPageNo(1).withPageSize(5);
ListStreamStatisticsResponse response = lssClient.listStreamStatistics(request);
```

getStreamStatistics

查询特定Domain下的特定Stream数据。

```
GetStreamStatisticsRequest request = new GetStreamStatisticsRequest();
request.withDomain(domain).withApp(app).withStream(stream).withAggregate(false)
.withStartDate("20161019").withEndDate("20161020");
GetStreamStatisticsResponse response = lssClient.getStreamStatistics(request);
```

listRealTimeStreamStatistics

查询特定Domain下实时stream的观看人数与带宽数据。

```
ListRealtimeStreamStatisticsRequest request = new ListRealtimeStreamStatisticsRequest();
request.withPlayDomain(playDomain).withApp(app);
ListRealtimeStreamStatisticsResponse response = lssClient.listRealTimeStreamStatistics(request);
```

通知功能

当用户成功创建消息通知，并在直播会话中配置Notification的接口名称和接口地址，LSS会在直播会话状态转换时主动向用户推送消息，以便用户随时了解直播stream的情况

创建Notification

新建Notification，参考如下代码：

```
public void createNotification(LssClient client, String name, String endpoint) {
    client.createNotification(name, endpoint);
}
```

查询所有Notification

用户可以通过如下代码查询所有的Notification：

```
public void listNotifications(LssClient client) {  
    ListNotificationsResponse resp = client.listNotifications();  
    System.out.println(resp.toString());  
}
```

查询指定的Notification信息

用户可以查询指定名称的Notification：

```
public void getNotification(LssClient client, String name) {  
    GetNotificationResponse resp = client.getNotification(name);  
    System.out.println("name: " + resp.getName());  
    System.out.println("endpoint: " + resp.getEndpoint());  
}
```

删除Notification

删除一个指定名称的Notification，参考如下代码：

```
public void deleteNotification(LssClient client, String name) {  
    client.deleteNotification(name);  
}
```

录制功能

录制模板Recording包含直播视频录制功能的各类参数配置。通过录制模板可以配置录制视频的存储模式，存储到BOS或存储到VOD、录制视频格式(MP4/M3U8)、录制模式、单个录制文件时长等。

目前Java sdk仅支持查询指定录制模板和查询模板列表接口，用户如需新建和删除录制模板，请在控制台操作。

查询指定的Recording

用户可以查询指定名称的录制模板：

```
public void getRecording(LssClient client, String recording) {  
    GetRecordingResponse resp = client.getRecording(recording);  
    System.out.println(resp.toString());  
}
```

查询Recording列表

用户可以通过如下代码查询所有的录制模板列表：

```
public void listRecordings(LssClient client) {  
    ListRecordingsResponse resp = client.listRecordings();  
    System.out.println(resp.toString());  
}
```

PHP-SDK

安装LSS-PHP-SDK

- 当您将HTTP请求发送到百度智能云时，您需要对您的请求进行签名计算，以便百度智能云可以识别您的身份。您可以通过[API认证机制](#)了解鉴权认证机制。
- 成功登录管理控制台后，系统会自动分配一对AK/SK，您可通过控制台申请并管理自己的访问密钥。

🔗 安装SDK包

1. 下载最新版[BCE PHP SDK](#)。
2. 解压安装包并浏览SDK目录：

```
BaiduBce.phar
├── src
│   ├── BaiduBce
│   │   ├── Auth          //BCE签名相关
│   │   ├── Exception     //BCE客户端的异常
│   │   ├── Http          //BCEHttp通信相关
│   │   ├── Log           //BCE日志
│   │   ├── Services
│   │   └── Lss            //LSS主目录，此目录必须保留
│   │       └── LssClient.php //LSS操作类，所有操作可以通过LssClient类可以完成
│   └── Util              //BCE公用工具
└── vendor                //第三方库
```

3. 在脚本文件中添加以下代码并保存：

```
include 'BaiduBce.phar';
require 'YourConf.php';
```

有关配置文件的引用，请参考[配置LssClient](#)。

安装LssClient

🔗 配置LssClient

LssClient是LSS服务的PHP客户端，为开发者与LSS服务进行交互提供了一系列的方法。

在新建LssClient前，您需要先创建配置文件对LssClient进行配置，以下将此配置文件命名为YourConf.php，具体配置信息如下：

```
// 报告所有 PHP 错误
error_reporting(-1);

define('__LSS_CLIENT_ROOT', dirname(__DIR__));

// 设置LssClient的Access Key ID、Secret Access Key和ENDPOINT
$LSS_TEST_CONFIG =
    array(
        'credentials' => array(
            'ak' => '<AK>',
            'sk' => '<SK>',
        ),
        'endpoint' => 'http://lss.bj.baidubce.com',
    );

// 设置log的格式和级别
$__handler = new \Monolog\Handler\StreamHandler(STDERR, \Monolog\Logger::DEBUG);
$__handler->setFormatter(
    new \Monolog\Formatter\LineFormatter(null, null, false, true)
);
\BaiduBce\Log\LogFactory::setInstance(
    new \BaiduBce\Log\MonoLogFactory(array($__handler))
);
\BaiduBce\Log\LogFactory::setLogLevel(\Psr\Log\LogLevel::DEBUG);
```

新建LssClient

在完成上述配置之后，参考如下代码新建一个LssClient：

```
//使用PHP SDK，并且使用自定义配置文件
include 'BaiduBce.phar';
require 'YourConf.php';

use BaiduBce\Services\Lss\LssClient;

//调用配置文件中的参数
global $LSS_TEST_CONFIG;
//新建LssClient
$client = new LssClient($LSS_TEST_CONFIG);
```

参数说明

PHP SDK在 \BaiduBce\Bce.php中默认设置了一些基本参数，如需修改参数值，可参考此文件自定义参数配置函数，并在构造LssClient的时候传入，传入代码参考如下：

```
public function CustomizedConfig() {
    $customizedConfig = array(
        BceClientConfigOptions::PROTOCOL => 'http',
        BceClientConfigOptions::CONNECTION_TIMEOUT_IN_MILLIS => 120 * 1000,
        BceClientConfigOptions::SOCKET_TIMEOUT_IN_MILLIS => 300 * 1000,
        BceClientConfigOptions::SEND_BUF_SIZE => 5 * 1024 * 1024,
        BceClientConfigOptions::RECV_BUF_SIZE => 5 * 1024 * 1024,
        BceClientConfigOptions::CREDENTIALS => array(
            'ak' => '<AK>',
            'sk' => '<SK>',
        ),
        'endpoint' => 'http://lss.bj.baidubce.com',
    );

    //利用自定义配置创建LssClient
    $customizedClient = new LssClient($customizedConfig);

    //通过自定义配置调用方法
    $options = array('config'=>$customizedConfig);
    $this->client->listSessions($options);
}
```

参数说明如下：

参数	说明	默认值
PROTOCOL	协议	http
CONNECTION_TIMEOUT_IN_MILLIS	请求超时时间（单位：毫秒）	50 * 1000
SOCKET_TIMEOUT_IN_MILLIS	通过打开的连接传输数据的超时时间（单位：毫秒）	0（表示无限等待，若设置非0数值需要对文件大小和网速进行评估，否则上传大文件时会产生超时）
SEND_BUF_SIZE	发送缓冲区大小	1024 * 1024
RECV_BUF_SIZE	接收缓冲区大小	10 ** 1024 1024

🔗 相关说明

LssClient将可选的参数封装到\$options中，每一个方法具有的可选参数详见具体的接口使用方法介绍，以createSession方法为例，请参考如下代码设置可选参数：

```
//创建Session无任何必选参数，通过$options传入可选参数
$options = array(
    'description' => 'This is a test session',
    'preset' => '<Preset_Name>',
    'notification' => '<Notification_Name>',
);
$client->createSession($options);
```

注意：请不要将null传入\$options中，否则调用时会抛出异常。

🔗 录制模板Recording

录制模板是对直播过程中音视频进行录制的参数配置集合，包括录制文件存储模式、录制文件格式、录制文件切割时长等。

目前API仅支持查询指定录制模板和查询模板列表接口，用户如需新建录制模板，请在控制台创建。

查询录制模板

下述代码示例用于查询指定的Recording：

```
$name = '<Recording_Name>';  
$response = $client->getRecording($name);  
print json_encode($response);
```

录制模板列表

下述代码示例用于查询所有的Recording：

```
$response = $client->listRecordings();  
foreach ($response->recordings as $recording) {  
    print json_encode($recording);  
}
```

🔗 安全策略SecurityPolicy

安全策略是直播过程中安全配置参数的集合，包括推流认证、播放认证、内容加密、防盗链（Referer/IP 黑白名单）三类安全配置。

目前LSS为每个用户创建了一个默认安全策略，尚不支持创建自定义安全策略。

查询安全策略

下述代码示例用于查询指定的SecurityPolicy：

```
$name = 'default';  
$response = $client->getSecurityPolicy($name);  
print json_encode($response);
```

安全策略列表

下述代码示例用于查询所有的SecurityPolicy：

```
$response = $client->listSecurityPolicies();  
foreach ($response->securityPolicies as $securityPolicy) {  
    print json_encode($securityPolicy);  
}
```

🔗 通知Notification

您可以通过提供通知接口名称和接口地址来创建Notification，然后在创建直播会话时配置直播通知。LSS会在直播会话状态改变，直播过程中HLS直播m3u8文件等生成时主动向您推送消息。

创建通知

下述代码示例用于创建一个Notification：

```
//指定notification name & endpoint  
$name = '<Notification_Name>';  
$endpoint = 'http://notification.domain.com';  
$client->createNotification($name, $endpoint);
```

通知列表

下述代码示例用于查询所有的Notification：

```
$response = $client->listNotifications();  
foreach ($response->notifications as $notification) {  
    print json_encode($notification);  
}
```

查询通知

下述代码示例用于查询指定的Notification：

```
//指定notification name  
$name = '<Notification_Name>';  
$response = $client->getNotification($name);  
print json_encode($response);
```

删除通知

下述代码示例用于删除Notification：

```
//指定notification name  
$name = '<Notification_Name>';  
$client->deleteNotification($name);
```

注意：不能删除使用中的Notification。

🔗 水印Watermark

LSS支持为直播会话添加水印，包括图片水印和时间戳水印两种类型。

图片水印

为直播会话添加图片水印需要您先创建图片水印，然后在创建直播会话时配置图片水印。

创建图片水印

通过定义水印的详细参数集合（大小、位置等）来创建图片水印。代码示例如下：

```
$path = '<Image_Path>';  
$fp = fopen($path, 'rb');  
$content = fread($fp, filesize($path));  
fclose($fp);  
$content = base64_encode($content);  
$options = array(  
    'maxWidthInPixel' => <Image_Width_Limit>,  
    'maxHeightInPixel' => <Image_Height_Limit>,  
    'sizingPolicy' => '<Sizing_Policy>',  
);  
$name = '<Image_Watermark_Name>';  
$client->createImageWatermark($name, $content, $options);
```

查询指定图片水印

通过指定水印名称查询特定图片水印的详细信息，包括图片URL、大小、位置、创建时间等。代码示例如下：

```
//指定ImageWatermark name  
$name = '<Image_Watermark_Name>';  
$response = $client->getImageWatermark($name);  
print json_encode($response);
```

查询图片水印列表

查询用户的所有图片水印的详细信息。代码示例如下：

```
$resp = $client->listImageWatermarks();  
foreach ($response->imageWatermarks as $imageWatermark) {  
    print json_encode($imageWatermark);  
}
```

删除图片水印

通过指定水印名称删除特定图片水印。代码示例如下：

```
//指定ImageWatermark name  
$name = '<Image_Watermark_Name>';  
$client->deleteImageWatermark($name);
```

注意：不能删除使用中的图片水印。

时间戳水印

为直播会话添加时间戳水印需要您先创建时间戳水印，然后在创建直播会话时配置时间戳水印。

创建时间戳水印

通过定义水印的详细参数集合（时区、文字、背景、位置等）来创建时间戳水印。代码示例如下：

```
$options = array(  
    'timezone' => '<Timezone>',  
    'alpha' => <Alpha>,  
    'fontFamily' => '<Font_Family>',  
    'fontSizeInPoint' => <Font_Size>,  
    'fontColor' => '<Font_Color>',  
    'backgroundColor' => '<Background_Color>',  
);  
$name = '<Timestamp_Watermark_Name>';  
$client->createTimestampWatermark($name, $options);
```

查询指定时间戳水印

通过指定水印名称查询特定时间戳水印的详细信息，包括时间戳的时区、文字、位置、创建时间等。代码示例如下：

```
//指定TimestampWatermark name  
$name = '<Timestamp_Watermark_Name>';  
$response = $client->getTimestampWatermark($name);  
print json_encode($response);
```

时间戳水印列表

查询用户的所有时间戳水印的详细信息。代码示例如下：

```
$resp = $client->listTimestampWatermarks();  
foreach ($response->timestampWatermarks as $timestampWatermark) {  
    print json_encode($timestampWatermark);  
}
```

删除时间戳水印

通过指定水印名称删除特定时间戳水印。代码示例如下：

```
//指定TimestampWatermark name
$name = '<Timestamp_Watermark_Name>';
$client->deleteTimestampWatermark($name);
```

注意：不能删除使用中的时间戳水印。

LSS开发指南

音视频直播服务 LSS 采用域名（Domain）、应用（App）、流（Stream）三级直播管理单元，一个域名（Domain）下可包含多个应用（App），一个应用（App）下可包含多个流（Stream）。每个应用（App）继承所属域名（Domain）的所有配置，同理，每个流（Stream）继承所属应用（App）的所有配置。

Stream流管理

新建Stream

在特定Domain下新建推流Stream，用户只需要在域名下指定app、stream名称，就可以拼接推流地址。

用户无需创建推流地址，仅需要在推流时，将推流地址输入在推流工具即可开始推流。

查询特定Domain下的特定Stream

代码示例如下：

```
//指定Domain, App, Stream
$domain = '<Domain_Name>';
$app = '<App_Name>';
$stream = '<Stream_Name>';
$response = $client->getStream($domain, $app, $stream);
print json_encode($response);
```

查询特定Domain下的所有Stream

代码示例如下：

```
//指定Domain
$domain = '<Domain_Name>';
$response = $client->listStreams($domain);
foreach ($response->streams as $stream) {
    print json_encode($stream);
}
```

封禁特定Domain下的特定stream

代码示例如下：

```
//指定Domain, App, Stream
$domain = '<Domain_Name>';
$app = '<App_Name>';
$stream = '<Stream_Name>';
$response = $client->pauseStream($domain, $app, $stream);
print json_encode($response);
```

解禁特定Domain下的特定Stream

代码示例如下：

```
//指定Domain, App, Stream
$domain = '<Domain_Name>';
$app = '<App_Name>';
$stream = '<Stream_Name>';
$response = $client->resumeStream($domain, $app, $stream);
print json_encode($response);
```

更新特定Stream的水印

代码示例如下：

```
//指定Domain, 参数params
$domain = '<Domain_Name>';
$app = '<App_Name>';
$stream = '<Stream_Name>';
$options = array(
    'watermarks' => '<Watermarks_Object>',
);
$response = $client->updateWatermarksOfStream($domain, $app, $stream, $options);
print json_encode($response);
```

查询特定Stream的实时直播源信息

代码示例如下：

```
//指定Domain, App, Stream
$domain = '<Domain_Name>';
$app = '<App_Name>';
$stream = '<Stream_Name>';
$response = $client->getRealtimeStreamSourceInfo($domain, $app, $stream);
print json_encode($response);
```

删除Stream

删除特定Domain下的特定Stream

代码示例如下：

```
//指定Domain, App, Stream
$domain = '<Domain_Name>';
$app = '<App_Name>';
$stream = '<Stream_Name>';
$response = $client->deleteStream($domain, $app, $stream);
print json_encode($response);
```

🔗 安全策略

查询安全策略

查询指定安全策略的相关信息，代码示例如下：

```
//指定安全策略名称
$security_policy = '<SecurityPolicy_Name>';
$response = $client->querySecurityPolicy($security_policy);
print json_encode($response);
```

🔗 统计

查询所有Domain的统计概要

代码示例如下：

```
//指定参数params
$params = array(
    "startTime" => "2016-09-20T08:00:00Z",
    "endTime" => "2016-09-21T08:00:00Z",
);
$response = $client->getDomainSummaryStatistics($params);
print json_encode($response);
```

查询所有Domain的统计数据

代码示例如下：

```
//指定参数params
$params = array(
    "startTime" => "2016-09-20T08:00:00Z",
    "endTime" => "2016-09-21T08:00:00Z",
    "orderBy" => "domain",
);
$response = $client->listDomainStatistics($params);
print json_encode($response);
```

查询特定Domain下特定Stream的统计数据

代码示例如下：

```
//指定Domain, App, Stream, 参数params
$domain = '<Domain_Name>';
$app = '<App_Name>';
$stream = '<Stream_Name>';
$params = array(
    "startDate" => "20160920",
    "endDate" => "20160921",
);
$response = $client->getStreamStatistics($domain, $app, $stream, $params);
print json_encode($response);
```

查询特定Domain下所有Stream的统计数据

代码示例如下：

```
//指定Domain, 参数params
$domain = '<Domain_Name>';
$params = array(
    "startTime" => "2016-09-20T08:00:00Z",
    "endTime" => "2016-09-21T08:00:00Z",
);
$response = $client->listStreamStatistics($domain, $params);
print json_encode($response);
```

查询所有Domain的总流量

代码示例如下：

```
//指定参数params
$params = array(
    "startTime" => "2016-09-20T08:00:00Z",
    "endTime" => "2016-09-21T08:00:00Z",
    "timeInterval" => "MID_TERM",
);
$response = $client->getAllDomainTrafficStatistics($params);
print json_encode($response);
```

查询特定Domain的流量

代码示例如下：

```
//指定Domain, 参数params
$domain = '<Domain_Name>';
$params = array(
    "startTime" => "2016-09-20T08:00:00Z",
    "endTime" => "2016-09-21T08:00:00Z",
    "timeInterval" => "MID_TERM",
);
$response = $client->getDomainTrafficStatistics($domain, $params);
print json_encode($response);
```

查询所有Domain的总带宽

代码示例如下：

```
//指定参数params
$params = array(
    "startTime" => "2016-09-20T08:00:00Z",
    "endTime" => "2016-09-21T08:00:00Z",
    "timeInterval" => "MID_TERM",
);
$response = $client->getAllDomainBandwidthStatistics($params);
print json_encode($response);
```

查询特定Domain的带宽

代码示例如下：

```
//指定Domain, 参数params
$domain = '<Domain_Name>';
$params = array(
    "startTime" => "2016-09-20T08:00:00Z",
    "endTime" => "2016-09-21T08:00:00Z",
    "timeInterval" => "MID_TERM",
);
$response = $client->getDomainBandwidthStatistics($domain, $params);
print json_encode($response);
```

查询所有Domain的总请求数

代码示例如下：

```
//指定参数params
$params = array(
    "startTime" => "2016-09-20T08:00:00Z",
    "endTime" => "2016-09-21T08:00:00Z",
    "timeInterval" => "MID_TERM",
);
$response = $client->getAllDomainPlayCountStatistics($params);
print json_encode($response);
```

查询特定Domain的请求数

代码示例如下：

```
//指定Domain, 参数params
$domain = '<Domain_Name>';
$params = array(
    "startTime" => "2016-09-20T08:00:00Z",
    "endTime" => "2016-09-21T08:00:00Z",
    "timeInterval" => "MID_TERM",
);
$response = $client->getDomainPlayCountStatistics($domain, $params);
print json_encode($response);
```

查询实时Stream统计数据

查询指定domain、app下所有stream的实时统计数据。仅对streamingStatus=STREAMING的stream有效，否则会报Exception。

代码示例如下：

```
//指定Domain, App
$domain = '<Domain_Name>';
$app = '<App_Name>';
$response = $client->listRealtimeStreamStatistics($domain, $app);
print json_encode($response);
```

异常处理

****LSS异常提示****

异常方法	说明
BceBaseException	异常总集
BceClientException	客户端异常
BceServerException	服务器异常
InvalidArgumentException	系统自带异常，参数错误

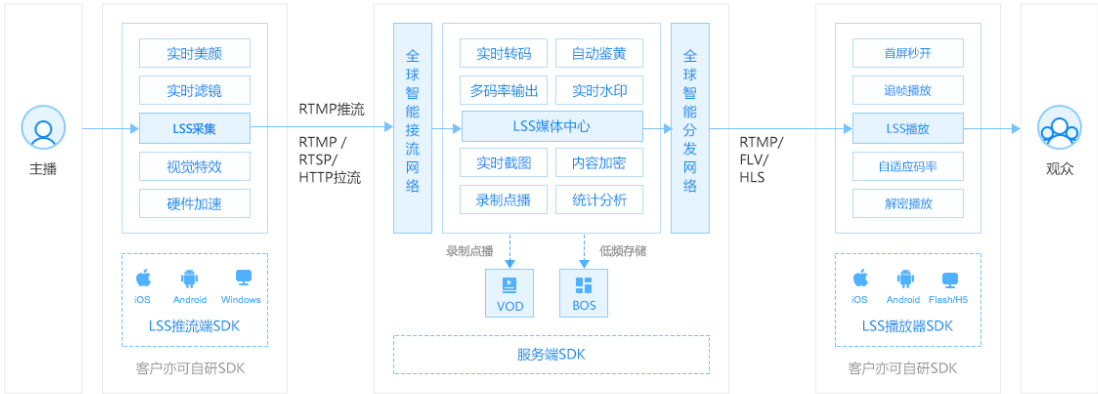
****异常事件获取**** 您可以使用try-catch获取某个事件所产生的异常：

```
try {
    $client->listSessions();
}catch (\BaiduBce\Exception\BceBaseException $e) {
    print $e->getMessage();
    if (stcmp(gettype($e), "BceClientException") == 0) {
        print "Catch a client exception";
    }
    if (stcmp(gettype($e), "BceServerException") == 0) {
        print "Catch a server exception";
    }
}
```

API参考

概述

音视频直播LSS（Live Streaming Service）是一个直播PaaS服务平台，旨在帮助企业及个人开发者客户快速搭建自己的直播平台及应用，LSS的产品架构图如图所示：



LSS提供直播“采集端-服务端-播放端”全套定制开发能力，采集端从直播源获取直播推流送到LSS服务端，LSS服务端完成直播流的处理(包括转码、加密、添加水印、生成缩略图等)，播放端进行播放。

如果您是初次调用百度智能云产品的API，可以观看[API入门视频指南](#)，快速掌握调用API的能力。也可以通过可视化API调试工具 - [示例代码中心](#)，进行学习与调用测试。

本文档介绍 LSS API 的用法，主要包括：

- 使用须知
- 错误返回
- 公共头域
- Stream接口
- 模板接口
- 统计接口
- 通知接口
- 录制视频裁剪
- 直播回看和时移接口
- 日志下载接口

使用须知

API调用遵循HTTP协议，采用REST风格。服务域名为 lss.bj.baidubce.com。数据交换格式为JSON，所有request/response body内容均采用UTF-8编码。

如果您刚刚接触LSS产品，请参阅[LSS产品描述](#)进行详细了解。

注意：“域名”分为推流域名和直播域名，分别cname至LSS所生成的地址。用户往推流域名上推流，即可在直播域名上进行播放。

🔗 API认证机制

所有API的安全认证一律采用Access Key与请求签名机制。Access Key由Access Key ID和Secret Access Key组成，均为字符串。对于每个HTTP请求，使用下面所描述的算法生成一个认证字符串。提交认证字符串放在Authorization头域里。服务端根据生成算法验证认证字符串的正确性。认证字符串的格式为bce-auth-

v{version}/{accessKeyId}/{timestamp}/{expirationPeriodInSeconds}/{signedHeaders}/{signature}。

- version是正整数。
- timestamp是生成签名时的UTC时间。
- expirationPeriodInSeconds表示签名有效期限。
- signedHeaders是签名算法中涉及到的头域列表。头域名之间用分号（;）分隔，如host;x-bce-date。列表按照字典序排列。（本API签名仅使用host和x-bce-date两个header）
- signature是256位签名的十六进制表示，由64个小写字母组成。

当百度智能云接收到用户的请求后，系统将使用相同的SK和同样的认证机制生成认证字符串，并与用户请求中包含的认证字符串进行比对。如果认证字符串相同，系统认为用户拥有指定的操作权限，并执行相关操作；如果认证字符串不同，系统将忽略该操作并返回错误码。

鉴权认证机制的详细内容请参见[鉴权认证](#)。

AK/SK的获取方法请参见[获取AK/SK](#)

🔗 LSS服务域名

- LSS 的服务域名是 lss.bj.baidubce.com。
- LSS API支持HTTP和HTTPS两种协议。为了提升数据的安全性，建议使用HTTPS协议。

🔗 日期与时间规范

日期与时间的表示有多种方式。为统一起见，除非约定俗成或有相应规范，凡是HTTP标准中规定的表示日期和时间字段用GMT，其他日期时间表示的地方一律采用UTC时间，遵循ISO 8601，并做以下约束：

- 表示日期一律采用YYYY-MM-DD方式，例如2016-07-07表示2016年7月7日。
- 表示时间一律采用hh:mm:ss方式，并在最后加一个大写字母Z表示UTC时间。例如23:00:10Z表示UTC时间23点0分10秒。
- 凡涉及日期和时间合并表示时，在两者中间加大写字母T，例如2016-07-07T23:00:10Z表示UTC时间2016年7月7日23点0分10秒。

🔗 API版本号

参数	类型	参数位置	描述	是否必须
version	String	URL参数	API版本号，不同的接口的版本号有所不同，请以具体接口文档中的请求示例中的版本号为准。	是

- 排版约定

排版格式	含义
< >	变量
[]	可选项
{ }	必选项
	互斥关系
等宽字体Courier New	屏幕输出

错误返回

错误返回格式

通用错误返回格式如下：

```
{
  "requestId": "<bce-request-id>",
  "code": "<error-code>",
  "message": "<error-message>",
  "success": false
}
```

LSS专用错误返回码

错误码	错误信息	描述	HTTP状态码
AccessDenied	You have no permission to the live streaming service	不可以使用音视频直播服务，请联系工单开通	403
UserOverdue	your account is overdue, please renew it	用户账号欠费，请续费	403
InvalidSession	Invalid session parameter	创建流参数错误	400
NoSuchSession	the requested session does not exist	流不存在	404
OperationNotPermitted	Invalid operation	非法操作	403
InvalidPreset	Invalid preset parameter	创建模板参数错误	400
DuplicatePreset	duplicate preset name	模板名称已被使用	400
NoSuchPreset	The requested preset does not exist	模板不存在	404
DeleteActiveResource	Preset is in use	资源使用中，无法删除	403
InvalidSecurityPolicy	Invalid security policy parameter	更新模板参数错误	400
DuplicateSecurityPolicy	duplicate security policy name	安全策略名称已被使用	400
NoSuchSecurityPolicy	No such security policy	安全策略不存在	404
SecurityPolicyInUse	security policy is used by some sessions	安全策略使用中，无法更新	403
InvalidNotification	Invalid notification parameter	创建通知参数错误	400
DuplicateNotification	duplicate notification name	通知名称已被使用	400
NoSuchNotification	notification does not exist	通知不存在	404
NotificationInUse	notification is used by some sessions	通知使用中，无法删除	403

通用错误返回码

错误返回码	错误消息	状态码	说明
AccessDenied	Access denied.	403 Forbidden	无权限访问对应的资源。
InappropriateJSON	The JSON you provided was well-formed and valid, but not appropriate for this operation.	400 Bad Request	请求中的JSON格式正确，但语义上不符合要求。如缺少特定必需项，或值类型不匹配等。出于兼容性考虑，对于所有无法识别的项应直接忽略，不应该返回这个错误。
InternalError	We encountered an internal error. Please try again.	500 Internal Server Error	所有未定义的其他错误。在有明确对应的其他类型的错误时（包括通用的和服务自定义的）不应该使用。

InvalidAccessKeyId	The Access Key ID you provided does not exist in our records.	403 Forbidden	Access Key ID不存在。
InvalidHTTPOAuthHeader	The Access Key ID you provided does not exist in our records.	400 Bad Request	Authorization头域格式错误。
InvalidHTTPRequest	There was an error in the body of your HTTP request.	400 Bad Request	HTTP body格式错误。例如不符合特定的 Encoding等。
InvalidURI	Could not parse the specified URI.	400 Bad Request	URI形式不正确。例如一些服务定义的关键词不匹配等。对于ID不匹配的问题，应定义更加具体的错误码，如NoSuchKey。
MalformedJSON	The JSON you provided was not well-formed.	400 Bad Request	JSON格式不合法。
InvalidVersion	The API version specified was invalid.	404 Not Found	URI的版本号不合法。
OptInRequired	A subscription for the service is required.	403 Forbidden	没有开通对应的服务。
PreconditionFailed	The specified If-Match header doesn't match the ETag header.	412 Precondition Failed	详见Etag。
RequestExpired	Request has expired. Timestamp date is <Data> .	400 Bad Request	请求超时。<Data>要改成x-bce-date。若请求中只有Date，需将Date转成datetime。
IdempotentParameterMismatch	The request uses the same client token as a previous, but non-identical request.	403 Forbidden	clientToken对应的API参数不一样。
SignatureDoesNotMatch	The request signature we calculated does not match the signature you provided. Check your Secret Access Key and signing method. Consult the service documentation for details.	400 Bad Request	Authorization头域中附带的签名和服务端验证不一致。
错误返回码	错误消息	状态码	说明
AccessDenied	Access denied	403 Forbidden	无权限访问对应的资源

enied	Access denied.	Forbidden	权限限制访问对应的资源。
InappropriateJSON	The JSON you provided was well-formed and valid, but not appropriate for this operation.	400 Bad Request	请求中的JSON格式正确，但语义上不符合要求。如缺少特定必需项，或值类型不匹配等。出于兼容性考虑，对于所有无法识别的项应直接忽略，不应该返回这个错误。
InternalError	We encountered an internal error. Please try again.	500 Internal Server Error	所有未定义的其他错误。在有明确对应的其他类型的错误时（包括通用的和服务自定义的）不应该使用。
InvalidAccessKeyId	The Access Key ID you provided does not exist in our records.	403 Forbidden	Access Key ID不存在。
InvalidHTTPTPAuthHeader	The Access Key ID you provided does not exist in our records.	400 Bad Request	Authorization头域格式错误。
InvalidHTTPRequest	There was an error in the body of your HTTP request.	400 Bad Request	HTTP body格式错误。例如不符合特定的Encoding等。
InvalidURI	Could not parse the specified URI.	400 Bad Request	URI形式不正确。例如一些服务定义的关键词不匹配等。对于ID不匹配的问题，应定义更加具体的错误码，如NoSuchKey。
MalformedJSON	The JSON you provided was not well-formed.	400 Bad Request	JSON格式不合法。
InvalidVersion	The API version specified was invalid.	404 Not Found	URI的版本号不合法。
OptInRequired	A subscription for the service is required.	403 Forbidden	没有开通对应的服务。
PreconditionFailed	The specified If-Match header doesn't match the ETag header.	412 Precondition Failed	详见Etag。
RequestExpired	Request has expired. Timestamp date is <Data>.	400 Bad Request	请求超时。<Data>要改成x-bce-date。若请求中只有Date，需将Date转成datetime。
IdempotentParameterMismatch	The request uses the same client token as a previous, but non-identical request.	403 Forbidden	clientToken对应的API参数不一样。
SignatureDoesNotMatch	The request signature we calculated does not match the signature you provided. Check your Secret Access Key and signing method. Consult the service	400 Bad Request	Authorization头域中附带的签名和服务端验证不一致。

	documentation for details.	t	
--	----------------------------	---	--

公共头域

公共请求头域

公共头域	描述	是否必选
host	服务域名，目前固定为lss.bj.baidubce.com	是
x-bce-date	表示日期的字符串，符合API规范。	否
x-bce-request-id	请求ID。	否
authorization	包含Access Key与请求签名。	是
content-type	application/json; charset=utf-8。	是
content-length	用于描述HTTP消息实体的传输长度the transfer-length of the message-body。	否

下面是一个标准的用户获取特定Stream信息时的请求头域内容：

```
GET /v5/domain/<playDomain>/app/<app>/stream/<stream> HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

LSS的API服务要求使用JSON格式（“创建文档”除外）的结构体来描述一个请求的具体内容。 下面是一个标准的用户创建通知时的请求消息体格式：

```
{
  "description": "this is a demo session",
  "preset": "live.rtmp_hls_forward_only"
}
```

LSS的API服务采用JSON格式的消息体作为响应返回的格式。下面是一个标准的用户查询特定Stream时的完整的请求返回示例：

```
{
  "app": "demo_app",
  "audit": "default",
  "createTime": "2016-07-14T08:20:02Z",
  "description": "",
  "play": {
    "flvUrl": "http://play.hello.com/test_app/test_stream.flv",
    "rtmpUrl": "rtmp://play.hello.com/test_app/test_stream"
  },
  "playDomain": "demo.domain.com",
  "preset": "live.rtmp_hls_forward_only",
  "presets": {
    "LO": "live.rtmp_hls_forward_only"
  },
  "publish": {
    "pullUrl": "http://testpull.baidu.com",
    "pushStream": "demo_stream",
    "region": "bj"
  },
  "scene": "normal",
  "securityPolicy": "default",
  "sessionId": "p-gggs1qiiin8b0a3z",
  "statistics": {
    "bandwidthInBps": 0,
    "playCount": 0
  },
  "status": "ONGOING"
  "streamingStatus": "STREAMING"
}
```

Stream接口

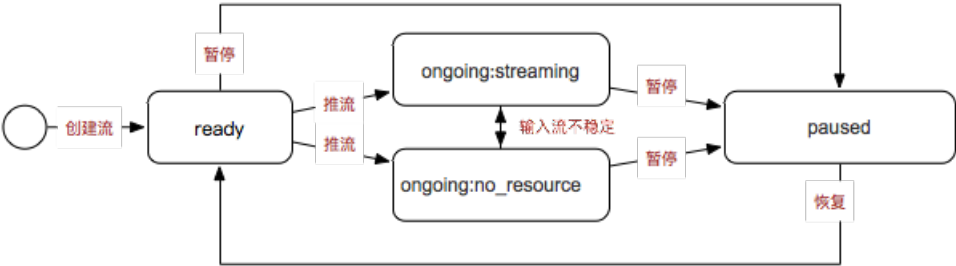
新建Stream

推流类型stream，用户只需要在域名下指定app、stream名称，就可以拼接推流地址。

推流

直播推流的状态包括下列几种：

- 已就绪 ready
- 直播中有输入流 ongoing:streaming；无输入流 ongoing:no_resource。streaming，no_resource为ongoing的子状态，会因为网络等因素导致直播不稳定而切换状态。
- 已暂停 paused



查询所有Stream

接口描述

查询特定Domain下的所有Stream。

请求语法

```
GET /v<version>/domain/<playDomain>/stream HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
playDomain	String	直播域名	是
status	String	Stream状态，有效值：READY、ONGOING、PAUSED	否
marker	String	本次请求的marker，标记查询的起始位置	否
maxSize	Integer	本次请求的Stream数目，不超过200。默认值：200	否

请求示例

```
GET /v5/domain/demo.domain.com/stream?status=READY HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/f1a2705d3cf8448cb917684c4f40ac1f/2016-01-28T07:19:17Z/3600/host;x-bce-account;x-bce-console-rpc-id;x-bce-date;x-bce-request-id/31b7dce808571441fb78077e74c048495945536909bbebd37e4037b7c796bf15
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
marker	String	本次请求的marker，标记查询的起始位置
isTruncated	Boolean	数据是否截断。True表示后面还有数据，False表示已经是最后一页
nextMarker	String	获取下一页所需要传递的marker值，仅当 "isTruncated": true时有效
streams	Array	流列表
+sessionId	String	stream 对应的sessionId
+app	String	app信息
+playDomain	String	直播域名
+description	String	流描述
+publish	Object	直播源信息
++region	String	流所属区域
++pushStream	String	推流Stream
++pullUrl	String	拉流url，仅对拉流有效
+status	String	直播状态，有效值：READY/ONGOING/PAUSED
+streamingStatus	String	直播流状态，仅当 status=ONGOING时存在，有效值：STREAMING、NO_SOURCE、FAILED
+error	Object	直播流错误信息，仅当 streamingStatus=FAILED时存在
++code	String	错误码
++message	String	错误详情
++image	String	图片水印模版名称列表
++timestamp	String	时间戳水印模版名称，单模版支持创建一个时间戳水印
+createTime	Date	流创建时间

响应示例

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
x-bce-request-id: 8dbde9a9-34e8-4071-afa0-b6d4c4e974b2
Cache-Control: no-cache
Server: BWS
Date: Thu, 28 Jan 2016 07:19:17 GMT
Content-Type: application/json;charset=UTF-8
```

```
{
  "isTruncated": true,
  "marker": "p-ga3h4ihs3dededaa",
  "nextMarker": "p-ga3h9q00hu7ayxmp",
  "streams" : [
    {
      "app": "demo_app",
      "createTime": "2016-07-14T08:20:02Z",
      "description": "",
      "playDomain": "demo.domain.com",
      "publish": {
        "pullUrl": "http://testpull.baidu.com",
        "pushStream": "demo_stream",
        "region": "bj"
      },
      "sessionId": "p-gggs1qiiin8b0a3z",
      "statistics": {
        "bandwidthInBps": 0,
        "playCount": 0
      },
      "status": "ONGOING"
      "streamingStatus": "STREAMING"
    }, ...
  ]
}
```

查询所有App

接口描述

查询特定Domain下所有App（不包括默认App）。

请求语法

```
GET /v<version>/domain/<playDomain>/app HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

N/A

请求示例

```
GET /v5/domain/demo.domain.com/app HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

响应头域

无特殊响应头域。

响应参数

N/A

响应示例

```
{
  "appList": [
    "demo_app",
    "live0",
    "test_app"
  ]
}
```

🔗 查询特定Stream

接口描述

查询特定Domain下的特定Stream。

请求语法

```
GET /v5/domain/<playDomain>/app/<app>/stream/<stream> HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

N/A

请求示例

```
GET /v5/domain/demo.domain.com/app/demo_app/stream/demo_stream HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/f1a2705d3cf8448cb917684c4f40ac1f/2016-01-28T07:19:19Z/3600/host;x-bce-account;x-bce-console-rpc-id;x-bce-date;x-bce-request-id/47a7d2dc00dce815ed39b5420b05a102bf070568c36c30066ce06ea900159bd3
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
----	----	----

参数	类型	说明
sessionId	String	流的唯一标识
playDomain	String	直播域名
app	String	App名称
description	String	流描述
preset	String	转码模板
presets	Object	多码率下转码模板列表
+L0	String	L0模式下使用的转码模板名称
+L1	String	L1模式下使用的转码模板名称
+L2	String	L2模式下使用的转码模板名称
+L3	String	L3模式下使用的转码模板名称
+L4	String	L4模式下使用的转码模板名称
publish	Object	直播源信息
+region	String	流所属区域
+pushStream	String	推流Stream
+ pushUrl	String	推流地址，仅对推流有效
+ pullUrl	String	拉流地址，仅对拉流有效
play	Object	直播播放信息
+hlsUrl	String	hls直播播放地址
+rtmpUrl	String	rtmp直播播放地址
+flvUrl	String	flv直播播放地址
+hlsUrls	Object	多码率下hls直播播放地址列表
++L0	String	L0模式hls直播播放地址
++L1	String	L1模式hls直播播放地址
++L2	String	L2模式hls直播播放地址
++L3	String	L3模式hls直播播放地址
++L4	String	L4模式hls直播播放地址
+rtmpUrls	Object	多码率下rtmp直播播放地址列表
++L0	String	L0模式rtmp直播播放地址
++L1	String	L1模式rtmp直播播放地址
++L2	String	L2模式rtmp直播播放地址
++L3	String	L3模式rtmp直播播放地址
++L4	String	L4模式rtmp直播播放地址
+flvUrls	Object	多码率下flv直播播放地址列表
++L0	String	L0模式flv直播播放地址
++L1	String	L1模式flv直播播放地址
++L2	String	L2模式flv直播播放地址
++L3	String	L3模式flv直播播放地址
++L4	String	L4模式flv直播播放地址
status	String	流状态，有效值：READY/ONGOING/PAUSED

recording	String	录制模板
thumbnail	String	缩略图模板
watermarks	Object	水印模版，包括图片水印模版和时间戳水印模版，每个流不能超过五个水印模版
+image	String	图片水印模版名称列表
+timestamp	String	时间戳水印模版名称，单模版支持创建一个时间戳水印
notification	String	通知
securityPolicy	String	安全策略
audit	String	内容审核名称
scene	String	直播场景名称
createTime	Date	流创建时间
statistics	Object	即时流统计信息
+bandwidthInBps	Number	当前流的占用带宽(bps)
+playCount	Number	当前流的播放人数

响应示例

```
{
  "app": "demo_app",
  "audit": "default",
  "createTime": "2016-07-14T08:20:02Z",
  "description": "",
  "play": {
    "flvUrl": "http://play.hello.com/test_app/test_stream.flv",
    "rtmpUrl": "rtmp://play.hello.com/test_app/test_stream"
  },
  "playDomain": "demo.domain.com",
  "preset": "live.rtmp_hls_forward_only",
  "presets": {
    "LO": "live.rtmp_hls_forward_only"
  },
  "publish": {
    "pullUrl": "http://testpull.baidu.com",
    "pushStream": "demo_stream",
    "region": "bj"
  },
  "scene": "normal",
  "securityPolicy": "default",
  "sessionId": "p-ggps1qiiin8b0a3z",
  "statistics": {
    "bandwidthInBps": 0,
    "playCount": 0
  },
  "status": "ONGOING"
  "streamingStatus": "STREAMING"
}
```

 查询活跃的Stream

接口描述

查询指定的播放domain下所有推流中的Stream（活跃Stream）。

请求语法

```
GET /v{version}/domain/{playDomain}/streaming HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求参数

参数	类型	是否必须	描述	可选值	默认值
playDomain	String	是	查询的域名	-	-

请求示例

```
GET /v5/domain/demo.domain.com/streaming HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/f1a2705d3cf8448cb917684c4f40ac1f/2016-01-28T07:19:19Z/3600/host;x-bce-account;x-bce-console-rpc-id;x-bce-date;x-bce-request-id/47a7d2dc00dce815ed39b5420b05a102bf070568c36c30066ce06ea900159bd3
```

响应参数

参数	类型	描述
streams	List	活跃流的信息
+ sessionId	String	会话ID
+ app	String	app信息
+ playDomain	String	所属播放域名
+ stream	String	推流stream
+ userId	String	用户Id

响应示例

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
x-bce-request-id: 8dbde9a9-34e8-4071-afa0-b6d4c4e974b2
Cache-Control: no-cache
Server: BWS
Date: Thu, 28 Jan 2016 07:19:17 GMT
Content-Type: application/json;charset=UTF-8

{
  "streams" : [
    {
      "app": "demo_app",
      "playDomain": "demo.domain.com",
      "stream": "demo_stream",
      "sessionId": "f-23ps11iin8b0a3z",
      "userId": "aabbccddeeffhh112233445566778899"
    }, ...
  ]
}
```

🔒 封禁特定Stream

接口描述

封禁播放domain下特定播放stream时，将直接封禁该条直播流，无法推流和播放。

请求语法

```
PUT /v5/domain/<playDomain>/app/<app>/stream/<stream>?pause HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

N/A

请求示例

```
PUT /v5/domain/demo.domain.com/app/demo_app/stream/demo_stream?pause HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

N/A

响应示例

```
HTTP/1.1 200 OK
```

🔗 解禁特定Stream

接口描述

解禁特定Domain下的特定Stream。

请求语法

```
PUT /v5/domain/<playDomain>/app/<app>/stream/<stream>?resume HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

N/A

请求示例

```
PUT /v5/domain/demo.domain.com/app/demo_app/stream/demo_stream?resume HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

N/A

响应示例

```
HTTP/1.1 200 OK
```

删除特定Stream

请求语法

```
DELETE /v5/domain/{playDomain}/app/{app}/stream/{stream} HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: {utc-date-string}
host: lss.bj.baidubce.com
accept: */*
connection: keep-alive
x-bce-request-id: {bce-request-id}
content-type: application/json
authorization: {bce-authorization-string}
```

请求头域

无特殊请求头域。

请求参数

N/A

请求示例

```
DELETE /v5/domain/demo.domain.com/app/demo_app/stream/demo_stream HTTP/1.1
content-length: 0
accept-encoding: gzip, deflate
x-bce-date: 2015-07-03T09:28:13Z
connection: keep-alive
accept: */*
user-agent: python-requests/2.4.0 CPython/2.7.9 Darwin/14.3.0
host: lss.bj.baidubce.com
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

N/A

响应示例

```
HTTP/1.1 200 OK
```

🔗 重置特定stream

中断当前直播流，并保证下次可以再次推流成功。

请求语法

```
PUT /v5/domain/{playDomain}/app/{app}/stream/{stream}?reset HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: {utc-date-string}
host: lss.baidubce.com
accept: */*
connection: keep-alive
x-bce-request-id: {bce-request-id}
content-type: application/json
authorization: {bce-authorization-string}
```

请求头域

无特殊请求头域。

请求参数

N/A

请求示例

```
PUT /v5/domain/demo.domain.com/app/demo_app/stream/demo_stream?reset HTTP/1.1
content-length: 0
accept-encoding: gzip, deflate
x-bce-date: 2015-07-03T09:28:13Z
connection: keep-alive
accept: */*
user-agent: python-requests/2.4.0 CPython/2.7.9 Darwin/14.3.0
host: lss.baidubce.com
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

N/A

响应示例

```
HTTP/1.1 200 OK
```

更新Stream水印模版

接口描述

更新Stream水印模版的配置。

- 默认继承域名下配置的水印模板，也可以通过本接口，对某个Stream单独设置水印模板。
- 如果watermarks为空，则更新结果为null，继承域名下的水印模板配置。

请求语法

```
POST /v5/domain/{playDomain}/app/{app}/stream/{stream}?watermark HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: {utc-date-string}
host: lss.bj.baidubce.com
accept: */*
connection: keep-alive
x-bce-request-id: {bce-request-id}
content-type: application/json
authorization: {bce-authorization-string}
```

请求头域

无特殊请求头域。

请求参数

参数	字段类型	是否必须	字段描述
watermarks	Object	是	更新的水印模版配置
+ image	Array	否	图片水印模版名称列表
+ timestamp	Array	否	时间戳水印模版名称列表

请求示例

```
POST /v5/domain/demo.domain.com/app/demo_app/stream/demo_stream?watermark HTTP/1.1
content-length: 0
accept-encoding: gzip, deflate
x-bce-date: 2015-07-03T09:28:13Z
connection: keep-alive
accept: */*
user-agent: python-requests/2.4.0 CPython/2.7.9 Darwin/14.3.0
host: lss.bj.baidubce.com
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715

{"watermarks": {
  "image":["demo_image_watermark1", "demo_image_watermark2"],
  "timestamp": ["demo_timestamp_watermark"]
}}
```

响应头域

无特殊响应头域。

响应参数

N/A

响应示例

```
HTTP/1.1 200 OK
```

更新Stream录制模版

接口描述

更新Stream录制模版的配置。

- 默认继承域名下配置的录制模版，也可以通过本接口，对某个Stream单独设置录制模版。
- 如果本接口recording为空，则更新结果为null，继承域名下的录制模板配置。

请求语法

```
PUT /v5/domain/{playDomain}/app/{app}/stream/{stream}?recording={recording} HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: {utc-date-string}
host: lss.bj.baidubce.com
accept: */*
connection: keep-alive
x-bce-request-id: {bce-request-id}
content-type: application/json
authorization: {bce-authorization-string}
```

请求头域

无特殊请求头域。

请求参数

参数	字段类型	是否必须	字段描述
recording	String	是	更新录制模板

请求示例

```
PUT /v5/domain/demo.domain.com/app/demo_app/stream/demo_stream?recording=recording_gz HTTP/1.1
content-length: 0
accept-encoding: gzip, deflate
x-bce-date: 2015-07-03T09:28:13Z
connection: keep-alive
accept: */*
user-agent: python-requests/2.4.0 CPython/2.7.9 Darwin/14.3.0
host: lss.bj.baidubce.com
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

N/A

响应示例

```
HTTP/1.1 200 OK
```

更新stream转码模版

接口描述

更新stream转码模版的配置。

- 默认继承域名下配置的转码模版，也可以通过本接口，对某个Stream单独设置转码模板。
- 如果本接口presets为空，则更新结果为null，继承域名下的转码模板配置。

请求语法

```
POST /v5/domain/{playDomain}/app/{app}/stream/{stream}?presets HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: {utc-date-string}
host: lss.bj.baidubce.com
accept: */*
connection: keep-alive
x-bce-request-id: {bce-request-id}
content-type: application/json
authorization: {bce-authorization-string}
```

请求头域

无特殊请求头域。

请求参数

参数	字段类型	是否必须	字段描述
presets	Map	是	更新的转码模版配置
+L1	String	否	L1线路下使用的转码模板名称
+L2	String	否	L2线路下使用的转码模板名称
+L3	String	否	L3线路下使用的转码模板名称
+L4	String	否	L4线路下使用的转码模板名称

注意：

- presets更新的合法keys仅限于L1，L2，L3，L4线路。
- L0为系统默认字段，不能自定义模版。

请求示例

```
PUT /v5/domain/demo.domain.com/app/demo_app/stream/demo_stream?presets HTTP/1.1
content-length: 0
accept-encoding: gzip, deflate
x-bce-date: 2015-07-03T09:28:13Z
connection: keep-alive
accept: */*
user-agent: python-requests/2.4.0 CPython/2.7.9 Darwin/14.3.0
host: lss.bj.baidubce.com
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715

{"presets": {
  "L1": "preset_l1"
}}
```

响应头域

无特殊响应头域。

响应参数

N/A

响应示例

HTTP/1.1 200 OK

更新stream目标推流地址

接口描述

更新stream目标推流地址

请求语法

```
PUT /v5/domain/{playDomain}/app/{app}/stream/{stream}?destinationPushUrl={destinationPushUrl} HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: {utc-date-string}
host: lss.bj.baidubce.com
accept: */*
connection: keep-alive
x-bce-request-id: {bce-request-id}
content-type: application/json
authorization: {bce-authorization-string}
```

请求头域

无特殊请求头域。

请求参数

参数	字段类型	是否必须	字段描述
destinationPushUrl	String	是	更新目标推流地址

请求示例

```
PUT /v5/domain/demo.domain.com/app/demo_app/stream/demo_stream?
destinationPushUrl=rtmp://demo.baidu.com/destination.flv HTTP/1.1
content-length: 0
accept-encoding: gzip, deflate
x-bce-date: 2015-07-03T09:28:13Z
connection: keep-alive
accept: */*
user-agent: python-requests/2.4.0 CPython/2.7.9 Darwin/14.3.0
host: lss.bj.baidubce.com
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-
bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

N/A

响应示例

```
HTTP/1.1 200 OK
```

实时直播源信息

接口描述

查询某条正在直播推流的直播源详细信息，本接口仅对streamingStatus=STREAMING的stream有效。

请求语法

```
GET /v{version}/domain/{playDomain}/app/{app}/stream/{stream}?sourceInfo HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

N/A

请求示例

```
GET /v5/domain/demo.domain.com/app/demo_app/stream/demo_stream?sourceInfo HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/f1a2705d3cf8448cb917684c4f40ac1f/2016-01-28T07:19:17Z/3600/host;x-bce-account;x-
bce-console-rpc-id;x-bce-date;x-bce-request-
id/31b7dce808571441fb78077e74c048495945536909bbebd37e4037b7c796bf15
```

响应头域

无特殊响应头域。

响应参数

参数	字段类型	字段描述
sessionId	String	流的唯一ID
sourceIP	String	返回将直播流推向媒体中心CDN的对应IP
publishIP	String	推流端IP地址
score	Number	推流拉流质量评分[0,100]，100表示质量最好，0 表示质量最差
captureTime	Date	直播源信息采时间
inputBitRateInBps	Number	实时输入总码率,包括视频/音频/数据
video	Object	视频信息
+ codec	String	视频codec，NA表示无视频或未识别
+ profile	String	视频profile，NA表示无视频或未识别
+ level	String	视频level，NA表示无视频或未识别
+ widthInPixel	Number	视频宽度，0表示无视频或未识别
+ heightInPixel	Number	视频高度，0表示无视频或未识别
+ frameRate	Number	视频标称帧率，0表示无视频或未识别
+ bitRateInBps	Number	视频实时码率
+ realFPS	Number	视频实时帧率
audio	Object	音频信息
+ codec	String	音频codec，NA表示无音频或未识别
+ profile	String	音频profile， NA表示无音频或未识别
+ sampleRateInHz	Number	音频采样率，0表示无音频或未识别
+ channels	Number	音频声道数，0表示无音频或未识别
+ bitRateInBps	Number	音频实时码率

响应示例

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
x-bce-request-id: cde8bf42-8021-46de-8e9c-5cd088bfcee2
Cache-Control: no-cache
Server: BWS
Date: Thu, 04 Feb 2016 03:00:58 GMT
Content-Type: application/json;charset=UTF-8
```

```
{
  "sessionId" : "p-ga3pk3t3t9wc1h60",
  "sourceIP" : "10.111.65.220",
  "score" : 100,
  "captureTime" : "2016-02-04T03:00:56Z",
  "inputBitRateInBps" : 5980249,
  "video" : {
    "codec" : "H264",
    "profile" : "High",
    "level" : "3.1",
    "widthInPixel" : 1280,
    "heightInPixel" : 720,
    "frameRate" : 29.1,
    "bitRateInBps" : 5828568,
    "realFPS" : 28.9269
  },
  "audio" : {
    "codec" : "AAC",
    "profile" : "Main",
    "sampleRateInHz" : 44100,
    "channels" : 2,
    "bitRateInBps" : 151680
  }
}
```

🔗 域名下实时直播源信息

接口描述

查询域名下所有正在直播推流过程的直播源详细信息，本接口仅对streamingStatus=STREAMING的stream有效。

请求语法

```
GET /v{version}/domain/{playDomain}/sourceInfo HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

N/A

请求示例

```
GET /v5/domain/demo.domain.com/sourceInfo HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/f1a2705d3cf8448cb917684c4f40ac1f/2016-01-28T07:19:17Z/3600/host;x-bce-account;x-bce-console-rpc-id;x-bce-date;x-bce-request-id/31b7dce808571441fb78077e74c048495945536909bbebd37e4037b7c796bf15
```

响应头域

无特殊响应头域。

响应参数

字段名称	字段类型	字段描述
sourceInfoList	Array	sourceInfo List
+ sessionId	String	会话ID，用户无需关注
+ playDomain	String	播放域名
+ app	String	app名称
+ stream	String	stream名称
+ sourceIP	String	cdn节点IP地址，NA表示未知
+ score	Number	推流拉流质量评分[0,100]，100表示质量最好，0 表示质量最差
+ captureTime	Date	直播源信息采时间
+ inputBitRateInBps	Number	实时输入总码率，包括视频/音频/数据
+ video	Object	视频信息
++ codec	String	视频codec，NA表示无视频或未识别
++ profile	String	视频profile，NA表示无视频或未识别
++ level	String	视频level，NA表示无视频或未识别
++ widthInPixel	Number	视频宽度，0表示无视频或未识别
++ heightInPixel	Number	视频高度，0表示无视频或未识别
++ frameRate	Number	视频标称帧率，0表示无视频或未识别
++ bitRateInBps	Number	视频实时码率
++ realFPS	Number	视频实时帧率
+ audio	Object	音频信息
++ codec	String	音频codec，NA表示无音频或未识别
++ profile	String	音频profile，NA表示无音频或未识别
++ sampleRateInHz	Number	音频采样率，0表示无音频或未识别
++ channels	Number	音频声道数，0表示无音频或未识别
++ bitRateInBps	Number	音频实时码率

响应示例

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
x-bce-request-id: cde8bf42-8021-46de-8e9c-5cd088bfcee2
Cache-Control: no-cache
Server: BWS
Date: Thu, 04 Feb 2016 03:00:58 GMT
Content-Type: application/json;charset=UTF-8
```

```
[
  {
    "sessionId": "p-ga3pk3t3t9wc1h60",
    "playDomain": "play.demo.com",
    "app": "demoapp",
    "stream": "demostream",
    "sourceIP": "10.111.65.220",
    "score": 100,
    "captureTime": "2016-02-04T03:00:56Z",
    "inputBitRateInBps": 5980249,
    "video": {
      "codec": "H264",
      "profile": "High",
      "level": "3.1",
      "widthInPixel": 1280,
      "heightInPixel": 720,
      "frameRate": 29.1,
      "bitRateInBps": 5828568,
      "realFPS": 28.9269
    },
    "audio": {
      "codec": "AAC",
      "profile": "Main",
      "sampleRateInHz": 44100,
      "channels": 2,
      "bitRateInBps": 151680
    }
  }, ...
]
```

[查看实时流推流url参数](#)

接口描述

查询某条正在直播推流的推流url参数信息

请求语法

```
GET /v{version}/domain/{pushDomain}/app/{app}/stream/{stream}/params HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

N/A

请求示例

```
GET /v5/domain/demo.domain.com/app/{app}/stream/{stream}/params HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/f1a2705d3cf8448cb917684c4f40ac1f/2016-01-28T07:19:17Z/3600/host;x-bce-account;x-bce-console-rpc-id;x-bce-date;x-bce-request-id/31b7dce808571441fb78077e74c048495945536909bbebd37e4037b7c796bf15
```

响应头域

无特殊响应头域。

响应参数

字段名称	字段类型	字段描述
params	Object	推流参数列表
+ param1	String	参数名，值为参数值
+ param2	String	参数名，值为参数值

响应示例

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
x-bce-request-id: cde8bf42-8021-46de-8e9c-5cd088bfcee2
Cache-Control: no-cache
Server: BWS
Date: Thu, 04 Feb 2016 03:00:58 GMT
Content-Type: application/json;charset=UTF-8

{
  "params": {
    "param1": "paramValue1",
    "param2": "paramValue2"
  }
}
```

模板接口

🔗 录制模板接口

此接口用于配置直播过程中视频录制功能的参数集合。通过录制模板可以配置录制视频的存储模式，存储到BOS或存储到VOD、录制视频格式(MP4/M3U8)、单个录制文件时长等。

目前API仅支持查询指定录制模板和查询模板列表接口，用户如需新建录制模板，请在控制台创建，参见[录制模板](#)。

🔗 查询录制模板

接口描述

查询用户指定录制模板的详情。

请求语法

```
GET /v{version}/recording/{name} HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: {bce-authorization-string}
```

请求头域

无特殊Header参数

请求参数

无

请求示例

```
GET /v5/recording/demo_recording HTTP/1.1
```

响应头域

无特殊Header参数

响应参数

参数	类型	描述
name	String	录制模板名称
description	String	录制模板描述
bos	Object	录制到BOS的配置信息，与“录制到VOD”互斥
+ bucket	String	录制文件存储的bos bucket名称
+ region	String	录制文件存储的bos bucket区域
+ type	String	录制文件存储的bos 文件类型
vod	Object	录制到VOD的配置信息，与“录制到BOS”互斥
+ enabled	Boolean	是否录制到VOD
format	String	录制文件格式。录制到BOS时，有效值：mp4、m3u8，默认值：mp4；录制到VOD时，有效值：m3u8
pattern	String	录制文件名模式
periodInMinute	Number	单个录制文件时长，1~360分钟
inUse	Boolean	录制模板是否在用
alwaysMerge	Boolean	录制m3u8时一直合并
avMode	String	
createTime	Date	录制模板创建时间
lastUpdateTime	unix timestamp	录制模板最后更新时间（毫秒）

响应示例

```
HTTP/1.1 200 OK

{
  "name": "lss_demo_recording",
  "createTime": "2021-10-13T02: 59: 59Z",
  "lastUpdateTime": 1634094083000,
  "description": "api_recording_test",
  "bos": {
    "bucket": "live-bucket",
    "region": "bj",
    "type": "standard"
  },
  "format": "mp4",
  "pattern": "%d/%a/%s/recording_%.%f",
  "periodInMinute": 360,
  "inUse": false,
  "alwaysMerge": false
}
```

🔗 录制模板列表

接口描述

查询用户已创建的所有录制模板详情。

请求语法

```
GET /v{version}/recording HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: {bce-authorization-string}
```

请求头域

无特殊Header参数

请求参数

无

请求示例

```
GET /v5/recording HTTP/1.1
```

响应头域

无特殊Header参数

响应参数

参数	类型	描述
recordings	Array	录制模板列表
+ name	String	录制模板名称
+ description	String	录制模板描述
+ bos	Object	录制到BOS的配置信息，与“录制到VOD”互斥
++ bucket	String	录制文件存储的bos bucket名称
++ region	String	录制文件存储的bos bucket区域
++ type	String	录制文件存储的bos 文件存储类型
+vod	Object	录制到VOD的配置信息，与“录制到BOS”互斥
++ enabled	Boolean	是否录制到VOD
+ format	String	录制文件格式。录制到BOS时，有效值：mp4、m3u8，默认值：mp4；录制到VOD时，有效值：m3u8
+ pattern	String	录制文件名模式
+ periodInMinute	Number	单个录制文件时长
+ inUse	Boolean	录制模板是否在用
+ alwaysMerge	Boolean	录制m3u8时一直合并
+ createTime	Date	录制模板创建时间
+ lastUpdateTime	Date	录制模板最后更新时间
+ tagList	Array	标签列表
++ tagKey	String	标签键
++ tagValue	String	标签值

- 响应示例：

HTTP/1.1 200 OK

```
{
  "recordings": [
    {
      "name": "lss_api_recordingdemo01",
      "createTime": "2021-04-20T07: 28: 14Z",
      "lastUpdateTime": 1618903694000,
      "description": "api_recording_test",
      "bos": {
        "bucket": "live-bucket",
        "region": "bj",
        "type": "standard"
      },
      "format": "m3u8",
      "periodInMinute": 10,
      "inUse": false,
      "alwaysMerge": true,
      "tagList": [{
        "tagKey": "默认项目",
        "tagValue": ""
      }]
    },
    {
      "name": "lss_api_recordingdemo02",
      "createTime": "2021-04-20T07: 28: 14Z",
      "lastUpdateTime": 1618903694000,
      "description": "api_recording_test",
      "bos": {
        "bucket": "live-bucket",
        "region": "bj",
        "type": "standard"
      },
      "format": "m3u8",
      "periodInMinute": 10,
      "inUse": false,
      "alwaysMerge": true,
      "tagList": [{
        "tagKey": "默认项目",
        "tagValue": ""
      }]
    }
  ]
}
```

🔗 水印模板接口

在直播过程中，可以通过水印模板接口对输出视频添加水印，水印模版中定义水印的详细参数集合用于完成水印添加，包括水印类型（图片、时间戳）、水印大小、位置等。

🔗 创建图片水印模板

通过定义水印的详细参数集合（大小、位置等）来创建图片水印模板。

请求语法

```
POST /v<version>/watermark/image HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
name	String	水印名称	是
content	String	图片文件base64编码后字符串	是
size	Object	水印尺寸	否
+maxWidthInPixel	Number	水印最大宽度，size子参数。有效值：[10-4096]间的偶数。不设置时，如果设置了最大高度则会根据原始图片宽高比计算水印宽度，否则和原始图片宽度保持一致	否
+maxHeightInPixel	Number	水印最大高度，size子参数。有效值：[10-3072]间的偶数。不设置时，如果设置了最大宽度则会根据原始图片宽高比计算水印高度，否则和原始图片高度保持一致	否
+sizingPolicy	String	水印的尺寸伸缩策略，size子参数。有效值：keep	否
position	Object	水印位置	否
+verticalAlignment	String	垂直对齐，position子参数。可选值：top、center、bottom，默认值：top	否
+horizontalAlignment	String	水平对齐，position子参数。可选值：left、center、right，默认值：left	否
+verticalOffsetInPixel	Number	垂直偏移量，position子参数，与verticalAlignment配合，仅当verticalAlignment设置为top或bottom时有效。单位：像素，有效值：[0-3072]，默认值：10	否
+horizontalOffsetInPixel	Number	水平偏移量，position子参数，与horizontalAlignment配合，仅当horizontalAlignment设置为left或right时有效。单位：像素，有效值：[0-4096]，默认值：10	否

请求示例

```
POST /v5/watermark/image HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715

{
  "name": "demo_watermark",
  "content": "<demo_watermark_content>"
  "size": {
    "maxWidthInPixel": 200,
    "maxHeightInPixel": 200,
    "sizingPolicy": "keep"
  }
  "position": {
    "verticalAlignment": "top",
    "horizontalAlignment": "left",
    "verticalOffsetInPixel": 10,
    "horizontalOffsetInPixel": 10
  }
}
```

响应头域

无特殊响应头域。

响应参数

N/A

响应示例

```
HTTP/1.1 200 OK
```

🔗 查询图片水印

通过指定水印名称查询特定图片水印的详细信息，包括图片URL、大小、位置、创建时间等。

请求语法

```
GET /v<version>/watermark/image/<name> HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
name	String	水印名称	是

请求示例

```
GET /v5/watermark/image/demo_watermark HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
name	String	水印名称
imageUrl	String	存于BOS Bucket中的图片文件URL
size	Object	水印尺寸
+maxWidthInPixel	Number	水印最大宽度，size子参数
+maxHeightInPixel	Number	水印最大高度，size子参数
+sizingPolicy	String	水印的尺寸伸缩策略，size子参数
position	Object	水印位置
+verticalAlignment	String	垂直对齐，position子参数
+horizontalAlignm ent	String	水平对齐，position子参数
+verticalOffsetInPi xel	Number	垂直偏移量，position子参数，与verticalAlignment配合，仅当verticalAlignment设置为top或bottom时有效。单位：像素
+horizontalOffsetIn nPixel	Number	水平偏移量，position子参数，与horizontalAlignment配合，仅当horizontalAlignment设置为left或right时有效。单位：像素
createTime	Date	水印模板创建时间

响应示例

```
{
  "name": "demo_image_watermark",
  "imageUrl": "http://lss-watermarks.bos.bce.baidu.com/wmk-gdwmbvvs7f1igmjs.jpg"
  "size": {
    "maxWidthInPixel": 200,
    "maxHeightInPixel": 200,
    "sizingPolicy": "keep"
  },
  "position": {
    "verticalAlignment": "top",
    "horizontalAlignment": "left",
    "verticalOffsetInPixel": 10,
    "horizontalOffsetInPixel": 10
  },
  "createTime": "2016-04-22T12:51:34Z"
}
```

🔗 查询图片水印列表

查询用户的所有图片水印模板的详细信息。

请求语法

```
GET /v<version>/watermark/image HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

N/A

请求示例

```
GET /v5/watermark/image HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
imageWatermarks	Array	水印模板列表
+name	String	水印名称
+imageUrl	String	存于BOS Bucket中的图片文件URL
+size	Object	水印尺寸
++maxWidthInPixel	Number	水印最大宽度，size子参数
++maxHeightInPixel	Number	水印最大高度，size子参数
++sizingPolicy	String	水印的尺寸伸缩策略，size子参数
+position	Object	水印位置
++verticalAlignment	String	垂直对齐，position子参数
++horizontalAlignment	String	水平对齐，position子参数
++verticalOffsetInPixel	Number	垂直偏移量，position子参数，与verticalAlignment配合，仅当verticalAlignment设置为top或bottom时有效。单位：像素
++horizontalOffsetInPixel	Number	水平偏移量，position子参数，与horizontalAlignment配合，仅当horizontalAlignment设置为left或right时有效。单位：像素
+createTime	Date	水印模板创建时间

响应示例

```
{
  "imageWatermarks": [
    {
      "name": "demo_image_watermark",
      "imageUrl": "http://lss-watermarks.bos.bce.baidu.com/wmk-gdwmcbvs7f1igmjs.jpg"
      "size": {
        "maxWidthInPixel": 200,
        "maxHeightInPixel": 200,
        "sizingPolicy": "keep"
      },
      "position": {
        "verticalAlignment": "top",
        "horizontalAlignment": "left",
        "verticalOffsetInPixel": 10,
        "horizontalOffsetInPixel": 10
      },
      "createTime": "2016-04-22T12:51:34Z"
    },
    {
      "name": "demo_image_watermark_2",
      "imageUrl": "http://lss-watermarks.bos.bce.baidu.com/wmk-gdwmcbvs7f1igmjs.jpg"
      "size": {
        "maxWidthInPixel": 200,
        "maxHeightInPixel": 200,
        "sizingPolicy": "keep"
      },
      "position": {
        "verticalAlignment": "top",
        "horizontalAlignment": "left",
        "verticalOffsetInPixel": 10,
        "horizontalOffsetInPixel": 10
      },
      "createTime": "2016-04-22T12:51:34Z"
    }
  ]
}
```

 删除图片水印

通过指定水印名称删除特定图片水印模板。

请求语法

```
DELETE /v<version>/watermark/image/<name> HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
name	String	水印名称	是

请求示例

```
DELETE /v5/watermark/image/{name} HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

N/A

响应示例

```
HTTP/1.1 200 OK
```

创建时间戳水印

通过定义水印的详细参数集合（时区、文字、背景、位置等）来创建时间戳水印模板。

请求语法

```
POST /v<version>/watermark/timestamp HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
name	String	水印名称	是
timezone	String	时区。有效值：UTC、CST，默认值：CST	否
alpha	String	水印透明度。有效值：[0-1]间的小数，小数点后最多保留两位，0表示完全透明，1表示完全不透明	否
font	Object	水印文字	否
+family	String	文字字体，font子参数。有效值：Arial	否
+sizeInPoint	Number	文字大小，font子参数。有效值：[1-72]，默认值：16	否
+color	String	文字颜色，font子参数。有效值：#开头的十六进制颜色代码，默认值：#FFFFFF	否
background	Object	水印背景	否
+color	String	背景颜色，background子参数。有效值：#开头的十六进制颜色代码，默认值：#000000	否
position	Object	水印位置	否
+verticalAlignment	String	垂直对齐，position子参数。可选值：top、center、bottom，默认值：top	否
+horizontalAlignment	String	水平对齐，position子参数。可选值：left、center、right，默认值：left	否
+verticalOffsetInPixel	Number	垂直偏移量，position子参数，与verticalAlignment配合，仅当verticalAlignment设置为top或bottom时有效。单位：像素，有效值：[0-3072]，默认值：10	否
+horizontalOffsetInPixel	Number	水平偏移量，position子参数，与horizontalAlignment配合，仅当horizontalAlignment设置为left或right时有效。单位：像素，有效值：[0-4096]，默认值：10	否

请求示例

```
POST /v5/watermark/timestamp HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715

{
  "name": "demo_timestamp_watermark",
  "timezone": "CST",
  "alpha": 1,
  "font": {
    "family": "Arial",
    "sizeInPoint": 16,
    "color": "#FFFFFF"
  },
  "background": {
    "color": "#000000"
  },
  "position": {
    "verticalAlignment": "top",
    "horizontalAlignment": "left",
    "verticalOffsetInPixel": 10,
    "horizontalOffsetInPixel": 10,
  }
}
```

响应头域

无特殊响应头域。

响应参数

N/A

响应示例

```
HTTP/1.1 200 OK
```

🔗 查询时间戳水印

通过指定水印名称查询特定时间戳水印的详细信息，包括时间戳的时区、文字、位置、创建时间等。

请求语法

```
GET /v<version>/watermark/timestamp/<name> HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
name	String	水印名称	是

请求示例

```
GET /v5/watermark/timestamp/demo_watermark HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
name	String	水印名称
timezone	String	时区
alpha	String	水印透明度
font	Object	水印文字
+family	String	文字字体，font子参数
+sizeInPoint	Number	文字大小，font子参数
+color	String	文字颜色，font子参数
background	Object	水印背景
+color	String	背景颜色，background子参数
position	Object	水印位置
+verticalAlignment	String	垂直对齐，position子参数
+horizontalAlignm ent	String	水平对齐，position子参数
+verticalOffsetInPi xel	Number	垂直偏移量，position子参数，与verticalAlignment配合，仅当verticalAlignment设置为top或bottom时有效。单位：像素
+horizontalOffsetI nPixel	Number	水平偏移量，position子参数，与horizontalAlignment配合，仅当horizontalAlignment设置为left或right时有效。单位：像素
createTime	Date	水印模板创建时间

响应示例

```
{
  "name": "demo_timestamp_watermark",
  "timezone": "CST",
  "alpha": 1,
  "inUse" : true,
  "font": {
    "family": "Arial",
    "sizeInPoint": 16,
    "color": "#FFFFFF"
  },
  "background": {
    "color": "#000000"
  },
  "position": {
    "verticalAlignment": "top",
    "horizontalAlignment": "left",
    "verticalOffsetInPixel": 10
    "horizontalOffsetInPixel": 10
  },
  "createTime": "2016-04-22T12:51:34Z"
}
```

🔗 查询时间戳列表

查询用户的所有时间戳水印模板的详细信息。

请求语法

```
GET /v<version>/watermark/timestamp HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

N/A

请求示例

```
GET /v5/watermark/timestamp HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
timestampWatermarks	Array	水印模板列表
+name	String	水印名称
+timezone	String	时区
+alpha	String	水印透明度
+font	Object	水印文字
++family	String	文字字体，font子参数
++sizeInPoint	Number	文字大小，font子参数
++color	String	文字颜色，font子参数
+background	Object	水印背景
++color	String	背景颜色，background子参数
+position	Object	水印位置
++verticalAlignment	String	垂直对齐，position子参数
++horizontalAlignment	String	水平对齐，position子参数
++verticalOffsetInPixel	Number	垂直偏移量，position子参数，与verticalAlignment配合，仅当verticalAlignment设置为top或bottom时有效。单位：像素
++horizontalOffsetInPixel	Number	水平偏移量，position子参数，与horizontalAlignment配合，仅当horizontalAlignment设置为left或right时有效。单位：像素
+createTime	Date	水印模板创建时间
+tagList	Array	标签列表
++tagKey	String	标签键
++tagValue	String	标签值

响应示例

```
{
  "timestampWatermarks": [{
    "name": "demo",
    "createTime": "2020-01-06T09: 19: 29Z",
    "lastUpdateTime": 1578302369000,
    "position": {
      "verticalAlignment": "top",
      "horizontalAlignment": "left",
      "verticalOffsetInPixel": 3,
      "horizontalOffsetInPixel": 3
    },
    "timezone": "CST",
    "alpha": 0.19,
    "font": {
      "family": "Arial",
      "sizeInPoint": 9,
      "color": "#FFFFFF"
    },
    "inUse": false,
    "tagList": [{
      "tagKey": "默认项目",
      "tagValue": ""
    }]
  }]
}
```

```
    }, {
      "name": "test",
      "createTime": "2019-06-16T05: 05: 26Z",
      "lastUpdateTime": 1568801460000,
      "position": {
        "verticalAlignment": "top",
        "horizontalAlignment": "left",
        "verticalOffsetInPixel": 10,
        "horizontalOffsetInPixel": 10
      },
      "timezone": "CST",
      "alpha": 1,
      "font": {
        "family": "Arial",
        "sizeInPoint": 16,
        "color": "#FFFFFF"
      },
      "inUse": false
    }, {
      "name": "time",
      "createTime": "2018-08-30T09: 05: 17Z",
      "lastUpdateTime": 1618464853000,
      "position": {
        "verticalAlignment": "top",
        "horizontalAlignment": "left",
        "verticalOffsetInPixel": 10,
        "horizontalOffsetInPixel": 10
      },
      "timezone": "CST",
      "alpha": 1,
      "font": {
        "family": "Arial",
        "sizeInPoint": 16,
        "color": "#FFFFFF"
      },
      "inUse": true,
      "tagList": [{
        "tagKey": "默认项目",
        "tagValue": ""
      }]
    }
  ]
}
```

🔗 删除时间戳

通过指定水印名称删除特定时间戳水印模板。

请求语法

```
DELETE /v<version>/watermark/timestamp/<name> HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
name	String	水印名称	是

请求示例

```
DELETE /v5/watermark/timestamp/<name> HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

N/A

响应示例

```
HTTP/1.1 200 OK
```

缩略图模板接口

查询缩略图模板

请求语法

```
GET /v<version>/thumbnail/<name> HTTP/1.1
host:lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
name	String	缩略图模板名称	是

请求示例

```
GET /v5/thumbnail/demo_thumbnail HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
name	String	缩略图模板名称
description	String	缩略图模板描述
pattern	String	缩略图文件命名模式
bos	Object	缩略图文件存储至BOS的配置信息
+bucket	String	存储缩略图文件的BOS Bucket名称
+region	String	存储缩略图文件的BOS区域
target	Object	缩略图目标信息
+format	String	缩略图文件格式
+sizingPolicy	String	缩略图拉伸策略
+maxWidthInPixel	Number	缩略图宽度尺寸
+maxHeightInPixel	Number	缩略图高度尺寸
capture	Object	缩略图生成规则
+mode	String	缩略图生成模式
+startTimeInSecond	Number	缩略图开始时间
+endTimeInSecond	Number	缩略图结束时间
+intervalInSecond	Number	缩略图截图间隔，有效值：[1-21600]
createTime	Date	缩略图模版创建时间

响应示例

```
{
  "createTime": "2020-01-06T09: 28: 45Z",
  "skipFrame": true,
  "bos": {
    "bucket": "demo_thumbnail_bucket",
    "region": "bj"
  },
  "name": "demo_thumbnail",
  "pattern": "%d/%a/%s/thumbnails/%t.%f",
  "inUse": false,
  "description": "缩略图",
  "capture": {
    "mode": "manual",
    "startTimeInSecond": 0,
    "intervalInSecond": 5
  },
  "lastUpdateTime": 1578302926000,
  "target": {
    "sizingPolicy": "stretch",
    "format": "jpg"
  }
}
```

缩略图列表

请求语法

```
GET /v<version>/thumbnail HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

N/A

请求示例

```
GET /v5/thumbnail HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
thumbnails	Array	缩略图模板列表
+name	String	缩略图模板名称
+description	String	缩略图模板描述
+bos	Object	缩略图文件存储至BOS的配置信息
++bucket	String	存储缩略图文件的BOS Bucket名称
++region	String	存储缩略图文件的BOS区域
+target	Object	缩略图目标信息
++format	String	缩略图文件格式
++sizingPolicy	String	缩略图拉伸策略
++maxWidthInPixel	Number	缩略图宽度尺寸
++maxHeightInPixel	Number	缩略图高度尺寸
+capture	Object	缩略图生成规则
++mode	String	缩略图生成模式
++startTimeInSecond	Number	缩略图开始时间
++endTimeInSecond	Number	缩略图结束时间
++intervalInSecond	Number	缩略图截图间隔，有效值：[1-21600]
+createTime	Date	缩略图模版创建时间
+pattern	String	缩略图文件命名模式
+tagList	Array	标签列表
++tagKey	String	标签键
++tagValue	String	标签值

响应示例

```
{
  "thumbnails": [
    {
      "name": "demo_thumbnail",
      "createTime": "2020-01-06T09: 28: 45Z",
      "lastUpdateTime": 1578302926000,
      "description": "缩略图",
      "bos": {
        "bucket": "demo_thumbnail_bucket",
        "region": "bj"
      },
      "target": {
        "format": "jpg",
        "sizingPolicy": "stretch"
      },
      "capture": {
        "mode": "manual",
        "startTimeInSecond": 0,
        "intervalInSecond": 5
      },
      "pattern": "%d/%a/%s/thumbnails/%t.%f",
      "inUse": false,
      "skipFrame": true,
      "tagList": [{
        "tagKey": "默认项目",
        "tagValue": ""
      }]
    },
    {
      "name": "test",
      "createTime": "2019-06-16T05: 06: 46Z",
      "lastUpdateTime": 1573545608000,
      "description": "",
      "bos": {
        "bucket": "demo_thumbnail_bucket",
        "region": "bj"
      },
      "target": {
        "format": "jpg",
        "sizingPolicy": "stretch"
      },
      "capture": {
        "mode": "manual",
        "startTimeInSecond": 0,
        "intervalInSecond": 5
      },
      "pattern": "%d/%a/%s/thumbnails/%t.%f",
      "inUse": false,
      "skipFrame": true
    }
  ]
}
```

统计接口

查询统计数据

接口描述

查询特定Domain的统计数据。

请求语法

```
GET /v<version>/statistics/domain/<playDomain>?startDate=<startDate>&endDate=<endDate>&aggregate={true | false} HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
playDomain	String	直播域名	是
startDate	String	起始时间	是
endDate	String	结束时间	是
aggregate	boolean	指定是否聚合，即数据聚合统计或按日统计。有效值：true、false，默认值: false。为true时聚合统计；否则按日统计	否

请求示例

```
GET /v5/statistics/domain/play.e-web.com.cn?startDate=20160202&endDate=20160205 HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
domain	String	直播域名
startDate	String	起始日期
endDate	Number	结束日期
aggregate	Object	聚合统计数据，仅当 aggregate=true时存在
+durationInMinute	Number	直播总时长
+peakPlayCount	Number	当天峰值播放人数
+peakBandwidthInBps	Number	峰值带宽
+downstreamInByte	Number	总下行流量
+playCount	Number	累计播放请求数，等于用户累计播放时长，单位：分钟
statistics	Object	按日统计数据
+date	String	数据日期
+durationInMinute	Number	直播总时长
+peakPlayCount	Number	当天峰值播放人数
+peakBandwidthInBps	Number	峰值带宽
+downstreamInByte	Number	总下行流量
+playCount	Number	累计播放请求数，等于用户累计播放时长，单位：分钟

响应示例

- aggregate值为true

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "domain": "play.e-web.com.cn",
  "startDate": 20160101,
  "endDate": 20160201,
  "aggregate": {
    "durationInMinute": 200,
    "peakPlayCount": 100,
    "peakBandwidthInBps": 100,
    "downstreamInByte": 10000,
    "playCount": 200
  }
}
```

- aggregate值为false

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "domain": "play.e-web.com.cn",
  "startDate": 20160101,
  "endDate": 20160201,
  "statistics":[
    {
      "date": "20160202",
      "durationInMinute": 100,
      "peakPlayCount": 100,
      "peakBandwidthInBps": 100,
      "downstreamInByte": 10000,
      "playCount": 100
    },{
      "date": "20160203",
      "durationInMinute": 100,
      "peakPlayCount": 100,
      "peakBandwidthInBps": 100,
      "downstreamInByte": 10000,
      "playCount": 50
    }...
  ]
}
```

🔗 查询统计概要

接口描述

查询当前用户所有Domain的统计概要。

请求语法

```
GET /v<version>/statistics/domain/summary?startTime=<startTime>&endTime=<endTime> HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
startTime	String	起始时间	是
endTime	String	结束时间	否

请求示例

```
GET /v5/statistics/domain/summary/?startTime=2016-01-18T00%3a00%3a00Z&endTime=2016-01-18T00%3a00%3a00Z HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
startTime	String	统计数据起始时间
endTime	String	统计数据截止时间
summary	Object	统计概要
+downStreamInByte	Number	累计总流量
+durationInMinute	Number	累计直播时长
+playCount	Number	累计播放时长
+transcoding	Number	累计转码时长（分钟）

响应示例

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "startTime": "2016-01-01T00:00:00Z",
  "endTime": "2016-01-01T10:00:00Z",
  "summary": {
    "downStreamInByte": 12300000,
    "durationInMinute":1234,
    "playCount": 200
  }
}
```

🔗 查询总请求数

接口描述

查询当前用户所有Domain的总请求数。

请求语法

```
GET /v<version>/statistics/table/domain/playcount?startTime=<startTime>&endTime=<endTime>&timeInterval={LONG_TERM | MID_TERM | SHORT_TERM}&allDomain={true | false} HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
startTime	String	起始时间	是
endTime	String	结束时间	是
timeInterval	enum	时间间隔，有效值：LONG_TERM（每天）、MID_TERM（每小时）、SHORT_TERM（每5分钟）	是
allDomain	boolean	true查询全部域名，默认为false	是

请求示例

```
GET /v5/statistics/table/domain/playcount?startTime=2016-01-18T00%3a00%3a00Z&endTime=2016-01-19T00%3a00%3a00Z&timeInterval=LONG_TERM&allDomain=true
HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
domain	String	直播域名
startTime	String	统计数据起始时间
endTime	String	统计数据截止时间
timeInterval	enum	LONG_TERM, SHORT_TERM, MID_TERM
hlsStatistics	Object	hls请求数数组
+timestamp	String	数据时间点
+playCount	Number	播放请求数
\flvStatistics	Object	flv请求数数组
+timestamp	String	数据时间点
+playCount	Number	播放请求数
rtmpStatistics	Object	rtmp请求数数组
+timestamp	String	数据时间点
+playCount	Number	播放请求数
totalStatistics	Object	总请求数数组
+timestamp	String	数据时间点
+playCount	Number	播放请求数

响应示例

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "startTime": "2016-01-01T00:00:00Z",
  "endTime": "2016-01-04T00:00:00Z",
  "timeInterval": "LONG_TERM",
  "hlsStatistics":[
    {
      "timestamp": "2016-01-01T00:00:00Z",
      "playCount": 50
    },{
      "timestamp": "2016-01-02T00:00:00Z",
      "playCount": 0
    },{
      "timestamp": "2016-01-03T00:00:00Z",
      "playCount": 20
    }
  ],
  "flvStatistics":[
    {
      "timestamp": "2016-01-01T00:00:00Z",
      "playCount": 40
    },{
      "timestamp": "2016-01-02T00:00:00Z",
      "playCount": 0
    },{
      "timestamp": "2016-01-03T00:00:00Z",
      "playCount": 10
    }
  ],
  "rtmpStatistics":[
    {
      "timestamp": "2016-01-01T00:00:00Z",
      "playCount": 30
    },{
      "timestamp": "2016-01-02T00:00:00Z",
      "playCount": 0
    },{
      "timestamp": "2016-01-03T00:00:00Z",
      "playCount": 40
    }
  ],
  "totalStatistics":[
    {
      "timestamp": "2016-01-01T00:00:00Z",
      "playCount": 120
    },{
      "timestamp": "2016-01-02T00:00:00Z",
      "playCount": 0
    },{
      "timestamp": "2016-01-03T00:00:00Z",
      "playCount": 70
    }
  ]
}
```

接口描述

查询特定Domain的请求数。

请求语法

```
GET /v<version>/statistics/domain/<playDomain>/playcount?startTime=<startTime>&endTime=<endTime>&timeInterval={LONG_TERM | MID_TERM | SHORT_TERM}
HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
playDomain	String	直播域名	是
startTime	String	起始时间	是
endTime	String	结束时间	是
timeInterval	enum	时间间隔，有效值：LONG_TERM（每天）、MID_TERM（每小时）、SHORT_TERM（每5分钟）	是

请求示例

```
GET /v5/statistics/table/domain/demo.domain.com/playcount?startTime=2016-01-18T00%3a00%3a00Z&endTime=2016-01-19T00%3a00%3a00Z&timeInterval=LONG_TERM
HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
domain	String	直播域名
startTime	String	统计数据起始时间
endTime	String	统计数据截止时间
timeInterval	enum	LONG_TERM, SHORT_TERM, MID_TERM
hlsStatistics	Object	hls请求数数组
+timestamp	String	数据时间点
+playCount	Number	播放请求数
flvStatistics	Object	flv请求数数组
+timestamp	String	数据时间点
+playCount	Number	播放请求数
rtmpStatistics	Object	rtmp请求数数组
+timestamp	String	数据时间点
+playCount	Number	播放请求数
totalStatistics	Object	总请求数数组
+timestamp	String	数据时间点
+playCount	Number	播放请求数

响应示例

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "domain": "play.e-web.com.cn",
  "startTime": "2016-01-01T00:00:00Z",
  "endTime": "2016-01-04T00:00:00Z",
  "timeInterval": "LONG_TERM",
  "hlsStatistics":[
    {
      "timestamp": "2016-01-01T00:00:00Z",
      "playCount": 50
    },{
      "timestamp": "2016-01-02T00:00:00Z",
      "playCount": 0
    },{
      "timestamp": "2016-01-03T00:00:00Z",
      "playCount": 20
    }
  ],
  "flvStatistics":[
    {
      "timestamp": "2016-01-01T00:00:00Z",
      "playCount": 40
    },{
      "timestamp": "2016-01-02T00:00:00Z",
      "playCount": 0
    },{
      "timestamp": "2016-01-03T00:00:00Z",
      "playCount": 10
    }
  ],
  "rtmpStatistics":[
    {
      "timestamp": "2016-01-01T00:00:00Z",
      "playCount": 30
    },{
      "timestamp": "2016-01-02T00:00:00Z",
      "playCount": 0
    },{
      "timestamp": "2016-01-03T00:00:00Z",
      "playCount": 40
    }
  ],
  "totalStatistics":[
    {
      "timestamp": "2016-01-01T00:00:00Z",
      "playCount": 120
    },{
      "timestamp": "2016-01-02T00:00:00Z",
      "playCount": 0
    },{
      "timestamp": "2016-01-03T00:00:00Z",
      "playCount": 70
    }
  ]
}
```

🔗 查询总带宽

接口描述

查询当前用户所有Domain的总带宽。

请求语法

```
GET /v<version>/statistics/table/domain/bandwidth?startTime=<startTime>&endTime=<endTime>&timeInterval={LONG_TERM | MID_TERM | SHORT_TERM}&allDomain={true | false} HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
startTime	String	起始时间	是
endTime	String	结束时间	是
timeInterval	enum	时间间隔，有效值：LONG_TERM（每天）、MID_TERM（每小时）、SHORT_TERM（每5分钟）	是
allDomain	boolean	true查询全部域名，默认为false，	是

请求示例

```
GET /v5/statistics/table/domain/bandwidth?startTime=2016-01-18T00%3a00%3a00Z&endTime=2016-01-19T00%3a00%3a00Z&timeInterval=LONG_TERM&allDomain=true HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
startTime	String	统计数据起始时间
endTime	String	统计数据截止时间
timeInterval	String	时间间隔粒度
statistics	Object	统计数据数组
+timestamp	String	数据时间点
+bandwidthInBps	Number	平均带宽，单位：bps (bit per second)

响应示例

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "startTime": "2016-01-01T00:00:00Z",
  "endTime": "2016-01-01T10:00:00Z",
  "timeInterval" : "SHORT_TERM",
  "statistics":[
    {
      "timestamp": "2016-01-01T00:00:00Z",
      "bandwidthInBps": 100
    },{
      "timestamp": "2016-01-01T00:05:00Z",
      "bandwidthInBps": 0
    },{
      "timestamp": "2016-01-01T00:10:00Z",
      "bandwidthInBps": 150
    }
  ]
}
```

🔗 查询带宽

接口描述

查询特定Domain的带宽。

请求语法

```
GET /v<version>/statistics/table/domain/<playDomain>/bandwidth?startTime=<startTime>&endTime=
<endTime>&timeInterval={LONG_TERM | MID_TERM | SHORT_TERM}
HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
playDomain	String	直播域名	是
startTime	String	起始时间	是
endTime	String	结束时间	是
timeInterval	enum	时间间隔，有效值：LONG_TERM（每天）、MID_TERM（每小时）、SHORT_TERM（每5分钟）	是

请求示例

```
GET /v5/statistics/table/demo.domain.com/bandwidth?startTime=2016-01-18T00%3a00%3a00Z&endTime=2016-01-19T00%3a00%3a00Z&timeInterval=LONG_TERM&allDomain=true
HTTP/1.1
host: lss.bj.baidubce.comss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
startTime	String	统计数据起始时间
endTime	String	统计数据截止时间
timeInterval	String	时间间隔粒度
statistics	Object	统计数据数组
+timestamp	String	数据时间点
+bandwidthInBps	Number	平均带宽，单位：bps (bit per second)

响应示例

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "startTime": "2016-01-01T00:00:00Z",
  "endTime": "2016-01-01T10:00:00Z",
  "timeInterval" : "SHORT_TERM",
  "statistics":[
    {
      "timestamp": "2016-01-01T00:00:00Z",
      "bandwidthInBps": 100
    },{
      "timestamp": "2016-01-01T00:05:00Z",
      "bandwidthInBps": 0
    },{
      "timestamp": "2016-01-01T00:10:00Z",
      "bandwidthInBps": 150
    }
  ]
}
```

 查询总流量

接口描述

查询当前用户所有Domain的总流量。

请求语法

```
GET /v<version>/statistics/table/domain/traffic?startTime=<startTime>&endTime=<endTime>&timeInterval={LONG_TERM | MID_TERM | SHORT_TERM}&allDomain={true | false} HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
startTime	String	起始时间，采用UTC时间	是
endTime	String	结束时间，采用UTC时间	是
timeInterval	enum	时间间隔，有效值：LONG_TERM（每天）、MID_TERM（每小时）、SHORT_TERM（每5分钟）	是
allDomain	boolean	true查询全部域名，默认为false，	是

请求示例

```
GET /v5/statistics/table/domain/traffic?startTime=2016-01-18T00%3a00%3a00Z&endTime=2016-01-18T00%3a00%3a00Z&timeInterval=SHORT_TERM&allDomain=true HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
domain	String	直播域名
startTime	String	统计数据起始时间
endTime	String	统计数据截止时间
timeInterval	String	时间间隔
statistics	Object	统计数据数组
+timestamp	String	数据时间点
+downstreamInByte	Number	下行流量

响应示例

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "startTime": "2016-01-01T00:00:00Z",
  "endTime": "2016-01-01T10:00:00Z",
  "timeInterval": "SHORT_TERM",
  "statistics":[
    {
      "timestamp": "2016-01-01T00:00:00Z",
      "downstreamInByte": 100
    },{
      "timestamp": "2016-01-01T00:10:00Z",
      "downstreamInByte": 0
    },{
      "timestamp": "2016-01-01T00:20:00Z",
      "downstreamInByte": 150
    }
  ]
}
```

 查询流量

接口描述

查询特定Domain的流量。

请求语法

```
GET /v<version>/statistics/table/domain/<playDomain>/traffic?startTime=<startTime>&endTime=
<endTime>&timeInterval={LONG_TERM | MID_TERM | SHORT_TERM} HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
playDomain	String	直播域名	是
startTime	String	起始时间	是
endTime	String	结束时间	是
timeInterval	enum	时间间隔，有效值：LONG_TERM（每天）、MID_TERM（每小时）、SHORT_TERM（每5分钟）	是

请求示例

```
GET /v5/statistics/table/domain/demo.domain.com/traffic?startTime=2016-01-18T00%3a00%3a00Z&endTime=2016-01-18T00%3a00%3a00Z&timeInterval=SHORT_TERM HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
domain	String	直播域名
startTime	String	统计数据起始时间，精确到秒级，按输入的时间间隔查询数据。
endTime	String	统计数据截止时间，精确到秒级，按输入的时间间隔查询数据。
timeInterval	String	时间间隔
statistics	Object	统计数据数组
+timestamp	String	数据时间点
+downstreamInByte	Number	下行流量

响应示例

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "domain": "play.e-web.com.cn",
  "startTime": "2016-01-01T00:00:00Z",
  "endTime": "2016-01-01T10:00:00Z",
  "timeInterval": "SHORT_TERM",
  "statistics":[
    {
      "timestamp": "2016-01-01T00:00:00Z",
      "downstreamInByte": 100
    },{
      "timestamp": "2016-01-01T00:10:00Z",
      "downstreamInByte": 0
    },{
      "timestamp": "2016-01-01T00:20:00Z",
      "downstreamInByte": 150
    }
  ]
}
```

 查询所有Domain数据

接口描述

查询当前用户的所有Domain数据。

请求语法

```
GET /v<version>/statistics/domain/list?startTime=<startTime>&endTime=<endTime>&[orderBy=
<orderRules>&keywordType=<keywordType>&keyword=<keyword>] HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
startTime	Date	开始时间，精确到秒级，按输入的时间间隔查询数据。	是
endTime	Date	结束时间，精确到秒级，按输入的时间间隔查询数据。	否
orderBy	String	排序关键字。有效值：domain，downstream，duration，peak_bandwidth，peak_play_count，play_count。默认按流量降序排列	否
keywordType	String	搜索关键字类型。默认值：domain	否
keyword	String	搜索关键字，"",表示匹配所有	否

请求示例

```
GET /v5/statistics/domain/list?startTime=2016-01-18T00%3a00%3a00Z&endTime=2016-01-
18T00%3a00%3a00Z&orderBy=domain&keywordType=domain&keyword={keyword} HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-
bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
domainStatisticsList	Object	domain数据列表
+domain	String	直播域名
+startTime	String	统计数据起始时间
+endTime	String	统计数据截止时间
+aggregate	Object	统计数据聚合
++durationInMinute	Number	直播总时长
++peakPlayCount	Number	当天峰值播放人数
++peakBandwidthInBps	Number	峰值带宽
++downstreamInByte	Number	总下行流量
++playCount	Number	累计播放请求数（每用户每播放一分钟算一次请求），等于用户累计播放时长，单位：分钟
++transcoding	Number	累计转码时长（分钟）

响应示例

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "domainStatisticsList": [
    {
      "domain": "play.e-web.com.cn",
      "startDate": "2016-01-01T00:00:00Z",
      "endDate": "2016-01-05T00:00:00Z",
      "aggregate": {
        "durationInMinute": 200,
        "peakPlayCount": 100,
        "peakBandwidthInBps": 100,
        "downstreamInByte": 10000,
        "playCount": 200
      },
    },
    {
      "domain": "test.e-web.com.cn",
      "startDate": "20160101",
      "endDate": "20160201",
      "aggregate": {
        "durationInMinute": 200,
        "peakPlayCount": 100,
        "peakBandwidthInBps": 100,
        "downstreamInByte": 10000,
        "playCount": 200
      }
    }
  ]
}
```

 查询所有Stream数据

接口描述

查询当前用户特定Domain下的所有Stream数据。

请求语法

```
GET /v{version}/statistics/domain/{domain}/stream?app={app}&startTime=2016-01-18T00%3a00%3a00Z&endTime=2016-01-18T00%3a00%3a00Z&orderBy=app&keywordType=stream&keyword={keyword}
HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须	可选值	默认值
Domain	String	直播域名	是	-	-
app	String	app 名称	是	-	-
startTime	Date	起始时间	是	-	-
endTime	Date	结束时间	是	-	-
orderBy	String	排序关键字,默认升序，加desc可以降序	是	downstream,duration,peak_bandwidth, peak_play_count,play_count, "downstream desc", ...	-
keywordType	String	搜索关键字类型	否	stream	stream
keyword	String	搜索关键字	否	-	"" ,空表示匹配所有
pageNo	Int	页数	否	大于0的整数，例如1	1
pageSize	Int	每页的条数	否	大于0的整数，例如20	20

请求示例

```
GET /v5/statistics/domain/demo.domain.com/stream?app=myapp&startTime=2016-01-18T00%3a00%3a00Z&endTime=2016-01-18T00%3a00%3a00Z&orderBy=app&keywordType=stream&keyword={keyword} HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
streamStatisticsList	Object	stream数据列表
+startDate	String	统计数据起始日期
+endDate	String	统计数据截止日期
+app	String	应用名称
+stream	String	流名称
+aggregate	Object	统计数据聚合
++durationInMinute	Number	直播总时长
++peakPlayCount	Number	当天峰值播放人数
++peakBandwidthInBps	Number	峰值带宽
++downstreamInByte	Number	总下行流量
++playCount	Number	累计播放请求数，等于用户累计播放时长，单位：分钟
++transcoding	Number	累计转码时长（分钟）
totalCount	Number	总共stream的数量，并不一定等于该列表的长度

响应示例

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "totalCount": 236,
  "streamStatisticsList": [
    {
      "app": "testapp",
      "stream": "teststream",
      "startDate": 20160101,
      "endDate": 20160201,
      "aggregate": {
        "durationInMinute": 200,
        "peakPlayCount": 100,
        "peakBandwidthInBps": 100,
        "downstreamInByte": 10000,
        "playCount": 200,
        "transcoding": 0
      },
    },
    {
      "app": "testapp1",
      "stream": "teststream1",
      "startDate": 20160101,
      "endDate": 20160201,
      "aggregate": {
        "durationInMinute": 200,
        "peakPlayCount": 100,
        "peakBandwidthInBps": 100,
        "downstreamInByte": 10000,
        "playCount": 200,
        "transcoding": 0
      },
    }
  ]
}
```

查询特定Stream数据

接口描述

查询特定Domain下的特定Stream数据。

请求语法

```
GET /v<version>/statistics/domain/<playDomain>/app/<app>/stream/<stream>?startDate=<startDate>&endDate=
<endDate>&aggregate={true | false} HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
playDomain	String	直播域名	是
app	String	应用名称	是
stream	String	流名称	是
startDate	String	起始时间，默认为当天时间，若参数错误也默认为当天	是
endDate	String	结束时间，，默认为当天时间，若参数错误也默认为当天	是
aggregate	boolean	指定是否聚合，即数据聚合统计或按日统计。有效值：true、false，默认值: false。为true时聚合统计；否则按日统计	否

请求示例

```
GET /v5/statistics/domain/demo.domain.com/app/myapp/stream/mystream?
startDate=20160118&endDate=20160128&aggregate=true HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
app	String	应用名称
stream	String	流名称
startDate	String	起始日期
endDate	Number	结束日期
statistics	Object	统计数据数组
+date	String	数据日期
+durationInMinute	Number	直播总时长
+peakPlayCount	Number	当天峰值播放人数
+peakBandwidthInBps	Number	峰值带宽
+downstreamInByte	Number	总下行流量
+playCount	Number	累计播放请求数
aggregate	Object	聚合数据，只有请求参数设置 aggregate=true时才返回。
+durationInMinute	Number	直播总时长
+peakPlayCount	Number	当天峰值播放人数
+peakBandwidthInBps	Number	峰值带宽
+downstreamInByte	Number	总下行流量
+playCount	Number	累计播放请求数

响应示例

aggregate 和 statistics数据仅返回一份，当 aggregate=true时，返回aggregate类别数据；当 aggregate=false时，返回statistics类别数据。

- aggregate值为true

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "domain": "play.e-web.com.cn",
  "app": "test_stream",
  "stream": "test_stream",
  "startDate": 20160101,
  "endDate": 20160201,
  "aggregate": {
    "durationInMinute": 200,
    "peakPlayCount": 100,
    "peakBandwidthInBps": 100,
    "downstreamInByte": 10000,
    "playCount": 200
  }
}
```

- aggregate值为false

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "domain": "play.e-web.com.cn",
  "app": "test_stream",
  "stream": "test_stream",
  "startDate": 20160101,
  "endDate": 20160201,
  "statistics":[
    {
      "date": "20160202",
      "durationInMinute": 100,
      "peakPlayCount": 100,
      "peakBandwidthInBps": 100,
      "downstreamInByte": 10000,
      "playCount": 100
    },{
      "date": "20160203",
      "durationInMinute": 100,
      "peakPlayCount": 100,
      "peakBandwidthInBps": 100,
      "downstreamInByte": 10000,
      "playCount": 50
    }...
  ]
}
```

🔗 统计实时播放人数和带宽

接口描述

查询app下所有stream的实时播放人数和实时带宽，数据源采集频率每分钟一次，可能会有几分钟延迟情况。

请求语法

```
GET /v5/statistics/realtime/domain/{domain}/app/{app} HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: {utc-date-string}
host: lss.bj.baidubce.com
accept: */*
connection: keep-alive
x-bce-request-id: {bce-request-id}
content-type: application/json
authorization: {bce-authorization-string}
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必填	示例
domain	String	播放域名	是	play.bcelive.com
app	String	app名称	是	live

请求示例

```
GET /v5/statistics/realtime/domain/play.bcelive.com/app/live HTTP/1.1
content-length: 87
accept-encoding: gzip, deflate
x-bce-date: 2015-07-03T09:28:13Z
connection: keep-alive
accept: */*
user-agent: python-requests/2.4.0 CPython/2.7.9 Darwin/14.3.0
host: 10.105.97.172
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
bandwidthInBps	Long	实时带宽数据，单位Byte
playCount	Long	实时播放人数（每分钟下行带宽/上行码率）
date	Date	当前时间
stream	String	stream名称

响应示例

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
[
  {
    "bandwidthInBps": 272,
    "date": "2017-02-20T11:11:32Z",
    "playCount": 8,
    "stream": "stream1"
  },
  {
    "bandwidthInBps": 326874087,
    "date": "2017-02-20T11:11:32Z",
    "playCount": 174,
    "stream": "stream2"
  },
  {
    "bandwidthInBps": 541274250,
    "date": "2017-02-20T11:11:32Z",
    "playCount": 325,
    "stream": "stream3"
  }
]
```

特定domain上行流量

接口描述

查询多个domain(播放域名)的上行流量。

请求语法

```
GET /v5/statistics/domain/{domain}/uptraffic?startTime=2016-01-18T00%3a00%3a00Z&endTime=2016-01-18T00%3a00%3a00Z&timeInterval=SHORT_TERM HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: {utc-date-string}
host: 10.105.97.172
accept: */*
connection: keep-alive
x-bce-request-id: {bce-request-id}
content-type: application/json
authorization: {bce-authorization-string}
```

请求头域

无特殊请求头域。

请求参数

参数	类型	是否必须	描述	可选值	默认值
domain	String	是	播放域名，多个域名以,分割	-	-
startTime	Date	是	起始时间	-	-
endTime	Date	是	结束时间	-	-
timeInterval	String	是	时间间隔粒度	LONG_TERM/MID_TERM/SHORT_TERM；LONG_TERM:天粒度；MID_TERM:小时粒度；SHORT_TERM:5min粒度	-

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
startTime	String	统计数据起始时间
endTime	String	统计数据截止时间
timeInterval	String	时间间隔粒度;时间间隔粒度SHORT_TERM:五分钟平均带宽;MID_TERM:小时内的max五分钟平均带宽;LONG_TERM:一天内的max五分钟平均带宽。
statistics	Object	统计数据数组
+timestamp	String	数据时间点
+upstreamInByte	Number	上行流量

响应示例：

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "startTime": "2016-01-01T00:00:00Z",
  "endTime": "2016-01-01T10:00:00Z",
  "timeInterval ": LONG_TERM,
  "statistics":[
    {
      "timestamp": "2016-01-01T00:00:00Z",
      "upstreamInByte": 100
    },{
      "timestamp": "2016-01-01T00:10:00Z",
      "upstreamInByte": 0
    },{
      "timestamp": "2016-01-01T00:20:00Z",
      "upstreamInByte": 150
    }
  ]
}
```

所有domain上行流量

接口描述

查询所有播放域名上行流量。

请求语法

```
GET /v5/statistics/domain/uptraffic?startTime=2016-01-18T00%3a00%3a00Z&endTime=2016-01-18T00%3a00%3a00Z&timeInterval=SHORT_TERM HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: {utc-date-string}
host: 10.105.97.172
accept: */*
connection: keep-alive
x-bce-request-id: {bce-request-id}
content-type: application/json
authorization: {bce-authorization-string}
```

请求头域

无特殊请求头域。

请求参数

参数	类型	是否必须	描述	可选值	默认值
startTime	Date	是	起始时间	-	-
endTime	Date	是	结束时间	-	-
timeInterval	String	是	时间间隔粒度	LONG_TERM/MID_TERM/SHORT_TERM ; LONG_TERM:天粒度 ; MID_TERM:小时粒度 ; SHORT_TERM:5min粒度	-

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
domain	String	播放域名
startTime	String	统计数据起始时间
endTime	String	统计数据截止时间
timeInterval	String	时间间隔粒度;时间间隔粒度SHORT_TERM:五分钟平均带宽; MID_TERM:小时内的max五分钟平均带宽; LONG_TERM:一天内的max五分钟平均带宽。
statistics	Object	统计数据数组
+timestamp	String	数据时间点
+upstreamInByte	Number	上行流量

响应示例

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "domain": "play.bcelive.com",
  "startTime": "2016-01-01T00:00:00Z",
  "endTime": "2016-01-01T10:00:00Z",
  "timeInterval ": LONG_TERM,
  "statistics":[
    {
      "timestamp": "2016-01-01T00:00:00Z",
      "upstreamInByte": 100
    },{
      "timestamp": "2016-01-01T00:10:00Z",
      "upstreamInByte": 0
    },{
      "timestamp": "2016-01-01T00:20:00Z",
      "upstreamInByte": 150
    }
  ]
}
```

特定domain上行带宽

接口描述

查询多个domain(播放域名)的上行带宽

请求语法

```
GET /v5/statistics/domain/{domain}/upbandwidth?startTime=2016-01-18T00%3a00%3a00Z&endTime=2016-01-18T00%3a00%3a00Z&timeInterval=SHORT_TERM HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: {utc-date-string}
host: 10.105.97.172
accept: */*
connection: keep-alive
x-bce-request-id: {bce-request-id}
content-type: application/json
authorization: {bce-authorization-string}
```

请求头域

无特殊请求头域。

请求参数

参数	类型	是否必须	描述	可选值	默认值
domain	String	是	播放域名，多个域名以,分割	-	-
startTime	Date	是	起始时间	-	-
endTime	Date	是	结束时间	-	-
timeInterval	String	是	时间间隔粒度	LONG_TERM/MID_TERM/SHORT_TERM ; LONG_TERM:天粒度； MID_TERM:小时粒度； SHORT_TERM:5min粒度	-

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
domain	String	播放域名
startTime	String	统计数据起始时间
endTime	String	统计数据截止时间
timeInterval	String	时间间隔粒度;时间间隔粒度SHORT_TERM:五分钟平均带宽; MID_TERM:小时内的max五分钟平均带宽; LONG_TERM:一天内的max五分钟平均带宽。
statistics	Object	统计数据数组
+timestamp	String	数据时间点
+bandwidthInBps	Number	上行带宽

响应示例：

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "domain": "play.bcelive.com",
  "startTime": "2016-01-01T00:00:00Z",
  "endTime": "2016-01-01T10:00:00Z",
  "timeInterval ": LONG_TERM,
  "statistics":[
    {
      "timestamp": "2016-01-01T00:00:00Z",
      "bandwidthInBps": 100
    },{
      "timestamp": "2016-01-01T00:10:00Z",
      "bandwidthInBps": 0
    },{
      "timestamp": "2016-01-01T00:20:00Z",
      "bandwidthInBps": 150
    }
  ]
}
```

🔗 所有domain上行带宽

接口描述

查询用户下所有domain的上行带宽

请求语法

```
GET /v5/statistics/domain/upbandwidth?startTime=2016-01-18T00%3a00%3a00Z&endTime=2016-01-18T00%3a00%3a00Z&timeInterval=SHORT_TERM HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: {utc-date-string}
host: 10.105.97.172
accept: */*
connection: keep-alive
x-bce-request-id: {bce-request-id}
content-type: application/json
authorization: {bce-authorization-string}
```

请求头域

无特殊请求头域。

请求参数

参数	类型	是否必须	描述	可选值	默认值
startTime	Date	是	起始时间	-	-
endTime	Date	是	结束时间	-	-
timeInterval	String	是	时间间隔粒度	LONG_TERM/MID_TERM/SHORT_TERM ; LONG_TERM:天粒度 ; MID_TERM:小时粒度 ; SHORT_TERM:5min粒度	-

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
domain	String	播放域名
startTime	String	统计数据起始时间
endTime	String	统计数据截止时间
timeInterval	String	时间间隔粒度;时间间隔粒度 SHORT_TERM:五分钟平均带宽; MID_TERM:小时内的max五分钟平均带宽; LONG_TERM:一天内的max五分钟平均带宽。
statistics	Object	统计数据数组
+timestamp	String	数据时间点
+bandwidthInBps	Number	上行带宽

响应示例

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "domain": "play.bcelive.com",
  "startTime": "2016-01-01T00:00:00Z",
  "endTime": "2016-01-01T10:00:00Z",
  "timeInterval ": LONG_TERM,
  "statistics":[
    {
      "timestamp": "2016-01-01T00:00:00Z",
      "bandwidthInBps": 100
    },{
      "timestamp": "2016-01-01T00:10:00Z",
      "bandwidthInBps": 0
    },{
      "timestamp": "2016-01-01T00:20:00Z",
      "bandwidthInBps": 150
    }
  ]
}
```

🔗 查询录制路数

接口描述

查询特定Domain的录制路数/查询当前用户下所有Domain的总录制路数

🔗 查询特定Domain的录制路数

请求语法

```
GET /v<version>/statistics/table/domain/recordCount?domainType=<domainType>&domain=<domain>&startTime=
<startTime>&endTime=<endTime>&timeInterval={LONG_TERM | MID_TERM | SHORT_TERM} HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
domainType	String	域名类型，有'play" 和 "push"两种类型，分别对应按照 播放域名 和 推流域名 来查询录制路数	是
domain	String	直播域名(查询用户级别时不填)，若是查询多个域名，域名间以英文逗号分隔	否
startTime	String	起始时间， UtcTime	是
endTime	String	结束时间， UtcTime	是
timeInterval	enum	时间间隔粒度 默认SHORT_TERM(5min) SHORT_TERM（每5分钟）：5min内的录制路数峰值 MID_TERM（每小时）：小时内Max 5min录制路数 LONG_TERM（每天）：一天内Max 5min录制路数	否
recordType	String	查询录制文件类型，有效值：all，mp4，flv，hls 默认为"all",查询所有录制格式路数	否
allDomain	Boolean	默认为false，表示查询特定domain的录制路数，为true表示查询用户级别录制路数	否

注：查询用户级别录制数据时，domain不填，allDomain为true
查询域名级别录制数据时，domain为对应查询域名，allDomain不填

请求示例

```
GET /v5/statistics/table/domain/recordCount?domainType=play&domain=demo1.com&startTime=2021-08-13T01:00:00Z&endTime=2021-08-13T02:00:00Z&timeInterval=SHORT_TERM HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2021-08-13T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
startTime	String	统计数据起始时间,UtcTime
endTime	String	统计数据截止时间,UtcTime
domain	String	直播域名
timeInterval	enum	SHORT_TERM(每5分钟), MID_TERM(每小时),LONG_TERM(每天)
recordType	String	查询录制文件类型 默认为all，表示查询所有类型 mp4: 查询mp4类型录制路数 flv: 查询 flv类型录制路数 hls：查询hls类型录制路数
statistics	Object	统计数据数组
+type	String	录制文件类型(flav / mp4 / hls)
+timestamp	String	数据时间点
+count	Long	录制路数

响应示例

- **recordType**值为all 表示查询所有录制类型

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Aug, 14 Jul 2021 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "startTime": "2021-08-13T01:00:00Z",
  "endTime": "2021-08-13T01:05:00Z",
  "timeInterval": "SHORT_TERM",
  "domain": "demo1.com",
  "recordType": "all"
  "statistics": [
    {
      "type": "flv",
      "timestamp": "2021-08-13T01:00:00Z",
      "count": 60
    },
    {
      "type": "mp4",
      "timestamp": "2021-08-13T01:00:00Z",
      "count": 60
    },
    {
      "type": "hls",
      "timestamp": "2021-08-13T01:00:00Z",
      "count": 10
    }
  ]
}
```

- **recordType**值为mp4/flv/hls 表示查询特定录制类型(这里以mp4为例)

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a2860966
Date: Aug, 14 Jul 2021 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "startTime": "2021-08-13T01:00:00Z",
  "endTime": "2021-08-13T01:05:00Z",
  "timeInterval": "SHORT_TERM",
  "domain": "demo1.com",
  "recordType": "mp4"
  "statistics": [
    {
      "timestamp": "2021-08-13T01:00:00Z",
      "count": 60
    }
  ]
}
```

🔗 查询用户下所有域名总录制路数

请求语法

```
GET /v<version>/statistics/table/domain/recordCount?domainType=<domainType>&allDomain=
<allDomain>&startTime=<startTime>&endTime=<endTime>&timeInterval={LONG_TERM | MID_TERM | SHORT_TERM}
HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
domainType	String	域名类型，有"play" 和 "push"两种类型，分别对应按照 播放域名 和 推流域名 来查询录制路数	是
domain	String	直播域名(查询用户级别时不填)，若是查询多个域名，域名间以英文逗号分隔	否
startTime	String	起始时间， UtcTime	是
endTime	String	结束时间， UtcTime	是
timeInterval	enum	时间间隔粒度 默认SHORT_TERM(5min) SHORT_TERM（每5分钟）：5min内的录制路数峰值 MID_TERM（每小时）：小时内Max 5min录制路数 LONG_TERM（每天）：一天内Max 5min录制路数	否
recordType	String	查询录制文件类型，有效值：all，mp4，flv，hls 默认为"all",查询所有录制格式路数	否
allDomain	Boolean	默认为false，表示查询特定domain的录制路数，为true表示查询用户级别录制路数	否

注：查询用户级别录制数据时，domain不填，allDomain为true
查询域名级别录制数据时，domain为对应查询域名，allDomain不填

请求示例

```
GET /v5/statistics/table/domain/recordCount?domainType=play&allDomain=true&startTime=2021-08-
13T01:00:00Z&endTime=2021-08-13T02:00:00Z&timeInterval=SHORT_TERM HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2021-08-13T09:28:13Z/1800/content-type;host;x-
bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
startTime	String	统计数据起始时间,UtcTime
endTime	String	统计数据截止时间,UtcTime
domain	String	直播域名
timeInterval	enum	SHORT_TERM(每5分钟), MID_TERM(每小时),LONG_TERM(每天)
recordType	String	查询录制文件类型 默认为all，表示查询所有类型 mp4: 查询mp4类型录制路数 flv: 查询 flv类型录制路数 hls : 查询hls类型录制路数
statistics	Object	统计数据数组
+type	String	录制文件类型(flav / mp4 / hls)
+timestamp	String	数据时间点
+count	Long	录制路数

响应示例

- recordType值为all 表示查询所有录制类型

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Aug, 14 Jul 2021 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "startTime": "2021-08-13T01:00:00Z",
  "endTime": "2021-08-13T01:05:00Z",
  "timeInterval":"SHORT_TERM",
  "recordType":"all"
  "statistics":[
    {
      "type": "flv",
      "timestamp":"2021-08-13T01:00:00Z",
      "count": 100
    },
    {
      "type": "mp4",
      "timestamp":"2021-08-13T01:00:00Z",
      "count": 200
    },
    {
      "type": "hls",
      "timestamp":"2021-08-13T01:00:00Z",
      "count": 55
    }
  ]
}
```

- recordType值为mp4/flv/hls 表示查询特定录制类型(这里以mp4为例)

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a2860966
Date: Aug, 14 Jul 2021 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "startTime": "2021-08-13T01:00:00Z",
  "endTime": "2021-08-13T01:05:00Z",
  "timeInterval": "SHORT_TERM",
  "recordType": "mp4"
  "statistics": [
    {
      "timestamp": "2021-08-13T01:00:00Z",
      "count": 200
    }
  ]
}
```

🔗 转推带宽数据

接口描述

查询用户级别或者域名级别的转推（LSS推三方）带宽数据。

请求语法

```
GET /v<version>/statistics/table/domain/forwardBandwidth?domain=<domain>&allDomain=<allDomain>&startTime=
<startTime>&endTime=<endTime>&timeInterval={LONG_TERM | MID_TERM | SHORT_TERM} HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
startTime	String	起始时间，UtcTime	是
endTime	String	结束时间，UtcTime	是
domain	String	推流域名(查询用户级别时不填)，若是查询多个域名，域名间以英文逗号分隔	否
timeInterval	enum	时间间隔粒度 默认SHORT_TERM(5min) SHORT_TERM(每5分钟)、 MID_TERM(每小时)、 LONG_TERM(每天)	否
allDomain	Boolean	默认为false，表示查询特定domain的转推带宽，为true表示查询用户级别转推带宽	否

注：查询用户级别转推带宽数据时，domain不填，allDomain为true
查询域名级别转推带宽数据时，domain为对应查询域名，allDomain不填

请求示例

```
GET /v5/statistics/table/domain/forwardBandwidth?startTime=2022-05-27T05:00:00Z&endTime=2022-05-27T05:30:01Z&timeInterval=SHORT_TERM&domain=push.xxx.com HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
Authorization: bce-auth-v1/7dbfa52b34a24282ac026e49f96987d3/2022-05-27T07:30:53Z/1800/host/****
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
startTime	String	统计数据起始时间，UtcTime
endTime	String	统计数据截止时间，UtcTime
timeInterval	enum	SHORT_TERM(每5分钟), MID_TERM(每小时), LONG_TERM(每天)
statistics	List	统计数据数组
+timestamp	String	数据时间点
+bandwidthInBps	Long	平均带宽，单位：bps (bit per second)

响应示例

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Content-Type: application/json;charset=UTF-8
Date: Fri, 27 May 2022 07:30:53 GMT
X-Bce-Request-Id: 9eba0ee3-c419-43ee-bf8f-e802c929987e
Transfer-Encoding: chunked
```

```
{
  "startTime": "2022-05-27T05:00:00Z",
  "endTime": "2022-05-27T05:30:00Z",
  "allDomain": false,
  "statistics": [
    {
      "timestamp": "2022-05-27T05:00:00Z",
      "bandwidthInBps": 35559471999
    },
    {
      "timestamp": "2022-05-27T05:05:00Z",
      "bandwidthInBps": 35986738623
    },
    {
      "timestamp": "2022-05-27T05:10:00Z",
      "bandwidthInBps": 36271014665
    },
    {
      "timestamp": "2022-05-27T05:15:00Z",
      "bandwidthInBps": 36696126181
    },
    {
      "timestamp": "2022-05-27T05:20:00Z",
      "bandwidthInBps": 37144444535
    },
    {
      "timestamp": "2022-05-27T05:25:00Z",
      "bandwidthInBps": 37546103252
    }
  ],
  "timeInterval": "SHORT_TERM"
}
```

🔗 上行推流带宽数据

接口描述

查询用户级别或者域名级别的上行推流（用户推LSS）带宽数据。

请求语法

```
GET /v<version>/statistics/table/domain/publishBandwidth?domain=<domain>&allDomain=<allDomain>&startTime=
<startTime>&endTime=<endTime>&timeInterval={LONG_TERM | MID_TERM | SHORT_TERM} HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
startTime	String	起始时间，UtcTime	是
endTime	String	结束时间，UtcTime	是
domain	String	推流域名(查询用户级别时不填)，若是查询多个域名，域名间以英文逗号分隔	否
timeInterval	enum	时间间隔粒度 默认SHORT_TERM(5min) SHORT_TERM(每5分钟)、MID_TERM(每小时)、LONG_TERM(每天)	否
allDomain	Boolean	默认为false，表示查询特定domain的推流带宽，为true表示查询用户级别推流带宽	否

注：查询用户级别上行推流带宽数据时，domain不填，allDomain为true
查询域名级别上行推流带宽数据时，domain为对应查询域名，allDomain不填

请求示例

```
GET /v5/statistics/table/domain/publishBandwidth?startTime=2022-05-27T05:00:00Z&endTime=2022-05-27T05:30:01Z&timeInterval=SHORT_TERM&domain=push.xxx.com HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
Authorization: bce-auth-v1/7dbfa52b34a24282ac026e49f96987d3/2022-05-27T07:30:53Z/1800/host/****
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
startTime	String	统计数据起始时间，UtcTime
endTime	String	统计数据截止时间，UtcTime
timeInterval	enum	SHORT_TERM(每5分钟), MID_TERM(每小时), LONG_TERM(每天)
statistics	List	统计数据数组
+timestamp	String	数据时间点
+bandwidthInBps	Long	平均带宽，单位：bps (bit per second)

响应示例

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Content-Type: application/json;charset=UTF-8
Date: Fri, 27 May 2022 07:30:53 GMT
X-Bce-Request-Id: b4316e96-931e-44ad-b601-03031aae9037
Transfer-Encoding: chunked
```

```
{
  "startTime": "2022-05-27T05:00:00Z",
  "endTime": "2022-05-27T05:30:00Z",
  "statistics": [
    {
      "timestamp": "2022-05-27T05:00:00Z",
      "bandwidthInBps": 0
    },
    {
      "timestamp": "2022-05-27T05:05:00Z",
      "bandwidthInBps": 0
    },
    {
      "timestamp": "2022-05-27T05:10:00Z",
      "bandwidthInBps": 0
    },
    {
      "timestamp": "2022-05-27T05:15:00Z",
      "bandwidthInBps": 26054856445
    },
    {
      "timestamp": "2022-05-27T05:20:00Z",
      "bandwidthInBps": 32903641589
    },
    {
      "timestamp": "2022-05-27T05:25:00Z",
      "bandwidthInBps": 33260493588
    }
  ],
  "timeInterval": "SHORT_TERM"
}
```

🔗 低延迟直播带宽数据

接口描述

查询用户级别或者域名级别的低延迟带宽数据。

请求语法

```
GET /v<version>/statistics/rtc/domain/bandwidth?domain=<domain>&allDomain=<allDomain>&startTime=
<startTime>&endTime=<endTime>&timeInterval={LONG_TERM | MID_TERM | SHORT_TERM} HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
startTime	String	起始时间，UtcTime	是
endTime	String	结束时间，UtcTime	是
domain	String	直播域名(查询用户级别时不填)，若是查询多个域名，域名间以英文逗号分隔	否
timeInterval	enum	时间间隔粒度 默认SHORT_TERM(5min) SHORT_TERM(每5分钟)、 MID_TERM(每小时)、 LONG_TERM(每天)	否
allDomain	Boolean	默认为false，表示查询特定domain的低延迟带宽数据，为true表示查询用户级别低延迟带宽数据	否

注：查询用户级别低延迟带宽数据时，domain不填，allDomain为true
查询域名级别低延迟带宽数据时，domain为对应查询域名，allDomain不填

请求示例

```
GET /v5/statistics rtc/domain/bandwidth?startTime=2022-05-27T05:00:00Z&endTime=2022-05-27T05:30:01Z&timeInterval=SHORT_TERM&domain=play.xxx.com HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
Authorization: bce-auth-v1/7dbfa52b34a24282ac026e49f96987d3/2022-05-27T07:30:53Z/1800/host/****
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
startTime	String	统计数据起始时间，UtcTime
endTime	String	统计数据截止时间，UtcTime
timeInterval	enum	SHORT_TERM(每5分钟), MID_TERM(每小时), LONG_TERM(每天)
statistics	List	统计数据数组
+timestamp	String	数据时间点
+bandwidthInBps	Long	平均带宽，单位：bps (bit per second)

响应示例

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Content-Type: application/json;charset=UTF-8
Date: Fri, 27 May 2022 07:30:53 GMT
X-Bce-Request-Id: 9eba0ee3-c419-43ee-bf8f-e802c929987e
Transfer-Encoding: chunked
```

```
{
  "startTime": "2022-05-27T05:00:00Z",
  "endTime": "2022-05-27T05:30:00Z",
  "allDomain": false,
  "statistics": [
    {
      "timestamp": "2022-05-27T05:00:00Z",
      "bandwidthInBps": 35559471999
    },
    {
      "timestamp": "2022-05-27T05:05:00Z",
      "bandwidthInBps": 35986738623
    },
    {
      "timestamp": "2022-05-27T05:10:00Z",
      "bandwidthInBps": 36271014665
    },
    {
      "timestamp": "2022-05-27T05:15:00Z",
      "bandwidthInBps": 36696126181
    },
    {
      "timestamp": "2022-05-27T05:20:00Z",
      "bandwidthInBps": 37144444535
    },
    {
      "timestamp": "2022-05-27T05:25:00Z",
      "bandwidthInBps": 37546103252
    }
  ],
  "timeInterval": "SHORT_TERM"
}
```

🔗 低延迟直播流量数据

接口描述

查询用户级别或者域名级别的低延迟流量数据。

请求语法

```
GET /v<version>/statistics/rtc/domain/traffic?domain=<domain>&allDomain=<allDomain>&startTime=
<startTime>&endTime=<endTime>&timeInterval={LONG_TERM | MID_TERM | SHORT_TERM} HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
startTime	String	起始时间，UtcTime	是
endTime	String	结束时间，UtcTime	是
domain	String	直播域名(查询用户级别时不填)，若是查询多个域名，域名间以英文逗号分隔	否
timeInterval	enum	时间间隔粒度 默认SHORT_TERM(5min) SHORT_TERM(每5分钟)、 MID_TERM(每小时)、 LONG_TERM(每天)	否
allDomain	Boolean	默认为false，表示查询特定domain的低延迟流量数据，为true表示查询用户级别低延迟流量数据	否

注：查询用户级别低延迟流量数据时，domain不填，allDomain为true
查询域名级别低延迟流量数据时，domain为对应查询域名，allDomain不填

请求示例

```
GET /v5/statistics rtc/domain/traffic?startTime=2022-05-27T05:00:00Z&endTime=2022-05-27T05:30:01Z&timeInterval=SHORT_TERM&domain=play.xxx.com HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
Authorization: bce-auth-v1/7dbfa52b34a24282ac026e49f96987d3/2022-05-27T07:30:53Z/1800/host/****
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
startTime	String	统计数据起始时间，UtcTime
endTime	String	统计数据截止时间，UtcTime
timeInterval	enum	SHORT_TERM(每5分钟), MID_TERM(每小时), LONG_TERM(每天)
statistics	List	统计数据数组
+timestamp	String	数据时间点
+downstreamInByte	Long	流量，单位：byte

响应示例

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Content-Type: application/json;charset=UTF-8
Date: Fri, 27 May 2022 07:30:53 GMT
X-Bce-Request-Id: 9eba0ee3-c419-43ee-bf8f-e802c929987e
Transfer-Encoding: chunked
```

```
{
  "startTime": "2022-05-27T05:00:00Z",
  "endTime": "2022-05-27T05:30:00Z",
  "allDomain": false,
  "statistics": [
    {
      "timestamp": "2022-05-27T05:00:00Z",
      "downstreamInByte": 35559471999
    },
    {
      "timestamp": "2022-05-27T05:05:00Z",
      "downstreamInByte": 35986738623
    },
    {
      "timestamp": "2022-05-27T05:10:00Z",
      "downstreamInByte": 36271014665
    },
    {
      "timestamp": "2022-05-27T05:15:00Z",
      "downstreamInByte": 36696126181
    },
    {
      "timestamp": "2022-05-27T05:20:00Z",
      "downstreamInByte": 37144444535
    },
    {
      "timestamp": "2022-05-27T05:25:00Z",
      "downstreamInByte": 37546103252
    }
  ],
  "timeInterval": "SHORT_TERM"
}
```

通知接口

LSS提供通知功能方便您实时掌握直播状态。通过在创建域名或创建流时指定通知，您可以在直播状态改变等情形下第一时间收到LSS推送的通知消息。

通知类型及格式

通知类型

通知类型	描述	结构体
SessionStatusChanged	直播流状态改变时触发	"body":{"previous":"READY","current":"ONGOING"}，其中： previous: 直播流改变前状态 current: 直播流改变后状态
StreamingStatusChanged	直播流状态改变时触发	"body":{"previous":null,"current":"STREAMING"}，其中： previous: 直播流改变前状态 current: 直播流改变后状态 当直播流状态为ONGOING时，流才会有直播流状态，null表示没有直播流状态
RecordFileGenerated	录制文件生成时触发	同步录制到VOD（视频点播）消息体： "body":{"createTime":"2016-11-22T02:47:54Z","mediaId":"mda-gkxkuqznz28u812y","mediaTitle":"lss-gkxk2fwy3wqyh4r/20161122104721"}，其中： mediaId:系统生成的媒资的唯一标识mediaId mediaTitle:媒资名称 同步录制到BOS（对象存储）消息体： "body":{"bucket":"live-bucket","key":"lss-fkig2241twm00z6b/recording_20151109143507.mp4","createTime":"2016-03-09T06:36:10Z","format":"mp4"}，其中： bucket: 录制文件存储的BOS bucket key: 录制文件的object key createTime: 录制文件的生成时间 format: 录制文件的格式： mp4(选择mp4录制模板只有mp4视频生产的回调) m3u8(如果未配置录制自动合并，则只有m3u8格式的回调；如果配置了自动合并，则除了m3u8的回调，还会在录制最后有一个ts切片生成的回调，用户可以忽略这个类型的回调。) flv(选择flv录制模板只有flv视频生产的回调)
ThumbnailFileGenerated	缩略图生成时触发	"body":{"bucket":"<bucket>","key":"<key>","createTime":"2016-07-12T08:20:16Z","mode":"manual","format":"jpg"}，其中： bucket: 存储缩略图文件的BOS Bucket名称 key: 缩略图文件在BOS存储中的Key createTime: 缩略图生成时间 mode: 缩略图生成模式 format: 缩略图文件格式
IllegalContentDetected	开启黄反审核后，检测到非法内容时触发	"body":{"alarmType":"PORN","confidence":"90","imageUrl":"<image_url>}，其中： alarmType: 非法内容类型 confidence: 置信度 imageUrl: 非法图片链接

通知格式

LSS通过HTTP请求将通知消息POST到您所配置的通知接口，用户通知接口收到HTTP POST请求BODY示例。具体内容和格式与通知类型相对应，这里以“缩略图生成时触发”时通知接口为例：

```
{
  "sessionId": "lss-gjwmww88txi41gqt",
  "playDomain": "play.bcelive.com",
  "app": "live",
  "stream": "onszf1j2gn8gb80462",
  "requestId": "2585d6cb-9edd-4961-bab0-3a9c568da0f0",
  "body": {
    "bucket": "video-live-thumbnail",
    "key": "/lss-gjwmww88txi41gqt.jpg",
    "createTime": "2016-10-21T07:44:26Z",
    "mode": "manual",
    "format": "jpg"
  },
  "notifyTime": "2016-12-21T07:44:26Z",
  "type": "ThumbnailFileGenerated"
}
```

messageBody字段的值为字符串类型，可以通过常用的JSON库转为标准的JSON格式。

字段说明：

字段	描述
sessionId	直播流ID
playDomain	直播域名
app	应用名称
stream	流名称
body	直播通知内容，body的具体内容和格式与通知类型相对应，即不同的通知类型，body中包含的子字段也不同。
notifyTime	文件生成时间
type	直播通知类型

🔗 创建通知

接口描述

通过用户提供的回调地址创建通知，在创建域名时可以指定通知接口，则直播过程中直播状态改变等情况下，LSS会向您指定的回调地址推送通知消息。默认情况下，同一个域名下的流使用同一个通知接口。

说明：如果客户端响应超时或返回500，LSS 会重新发送通知。

请求语法

```
POST /v<version>/notification HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: <utc-date-string>
host: lss.bj.baidubce.com
x-bce-request-id: <bce-request-id>
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	字段类型	必要性	字段描述	可选值	默认值
name	String	必选	接口名称。开头必须是小写字母，其余可以是小写字母、_或数字组成，最多不超过40个字符	-	-
endpoint	String	必选	通知消息接口地址，不超过256字符	-	-

请求示例

```
POST /v5/notification HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: 2015-07-03T09:28:13Z
user-agent: python-requests/2.4.0 CPython/2.7.9 Darwin/14.3.0
host:lss.bj.baidubce.com
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
content-type: application/json
authorization: bce-authorization-string

{
  "name": "live_notification",
  "endpoint": "http://live.notificaitonDomain.com/"
}
```

响应头域

无特殊响应头域。

响应参数

N/A

响应示例

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
```

🔗 查询通知

接口描述

查询通知的接口名称和接口地址。

请求语法

```
GET /v<version>/notification/<name> HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: <utc-date-string>
host:lss.bj.baidubce.com
x-bce-request-id: <bce-request-id>
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	字段类型	必要性	字段描述	可选值	默认值
name	String	必选	通知名称	-	-

请求示例

```
GET /v5/notification/live_notification HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: 2015-07-03T09:28:13Z
host: lss.bj.baidubce.com
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
content-type: application/json
authorization: bce-authorizati-on-string
```

响应头域

无特殊响应头域。

响应参数

参数	字段类型	字段描述
name	String	接口名称
endpoint	String	通知消息接口地址

响应示例

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8

{
  "name": "live_notification",
  "endpoint": "http://live.notificationDomain.com/"
}
```

通知列表

接口描述

获取已创建的全部通知。

请求语法

```
GET /v<version>/notification HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: <utc-date-string>
host: lss.bj.baidubce.com
x-bce-request-id: <bce-request-id>
content-type: application/json
authorization: <bce-authorizati-on-string>
```

请求头域

无特殊请求头域。

请求参数

N/A

请求示例

```
GET /v5/notification HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: 2015-07-03T09:28:13Z
host: lss.bj.baidubce.com
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
content-type: application/json
authorization: bce-authorization-string
```

响应头域

无特殊响应头域。

响应参数

参数	字段类型	字段描述
notifications	Object	通知列表
+ name	String	通知名称
+ endpoint	String	通知消息接收地址

响应示例

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8

{
  "notifications": [ {
    "name": "live_notification",
    "endpoint": "http://live.notificaitonDomain.com/"
  }, {
    "name": "my_notification",
    "endpoint": "http://my.notificaitonDomain.com/"
  } ]
}
```

删除通知

接口描述

删除指定的直播通知。

请求语法

```
DELETE /v<version>/notification/<name> HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: <utc-date-string>
host: lss.bj.baidubce.com
x-bce-request-id: <bce-request-id>
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

N/A

请求示例

```
DELETE /v5/notification/live_notification HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: 2015-07-03T09:28:13Z
user-agent: python-requests/2.4.0 CPython/2.7.9 Darwin/14.3.0
host: lss.bj.baidubce.com
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
content-type: application/json
authorization: bce-authorizati-on-string
```

响应头域

无特殊响应头域。

响应参数

N/A

响应示例

```
HTTP/1.1 200 OK
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
Date: Fri, 03 Jul 2015 09:28:13 GMT
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
```

轮播接口

轮播是将存放在百度对象存储BOS里的点播文件按节目单重新编排、打包成直播流的形式，实现将文件按一定间隔循环播放的播出效果。这一形式将随时点播、随意切换的播放模式和传统电视频道不间断播出的优点融合在一起。在用户不知道看什么内容时，为给他们提供更多选择。

轮播功能正在邀测中，您可[提交申请](#)测试本功能，也可[提交工单](#)咨询功能。开通轮播功能将产生费用，详情请见[计费文档](#)。

录制视频裁剪

接口描述

裁剪录制视频接口，输入绝对开始时间和结束时间，可以获取视频回放。

- 本接口仅支持已经录制的m3u8视频，裁剪后输出可以是m3u8或mp4格式。
- 只能剪裁已经录制完成的视频，不能干预直播中正在录制的或即将录制的视频。

请求语法

```
POST /v5/recording/clip HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: {utc-date-string}
host: lss.bj.baidubce.com
accept: */*
connection: keep-alive
x-bce-request-id: {bce-request-id}
content-type: application/json
authorization: {bce-authorization-string}
```

请求头域

无特殊请求头域。

请求参数

参数	类型	是否必须	描述	可选值	默认值
playDomain	String	是	播放域名名称	-	-
app	String	是	app名称	-	-
stream	String	是	stream名称	-	-
filename	String	否	指定裁剪后的文件名称	-	格式为clip_<stream名称>_<startTime>_<endTime>，startTime与endTime如果未指定也用对应的默认值
format	String	否	裁剪后视频的格式	m3u8, mp4	m3u8
startTime	Number	否	指定录制视频开始时间（unix时间戳，单位秒）	-	直播录制的最开始时间，不指定或0 表示从直播开始录制。
endTime	Number	否	指定录制视频结束时间（unix时间戳，单位秒）	-	直播录制结束最后时间，不指定或0 表示录制到直播结束。
sourceFile	String	是	指定要裁剪的m3u8文件名，需要包括bucket下目录的路径	-	-
pipeline	String	format=m p4时必选	指定MCT中的转码队列	-	-
preset	String	format=m p4时必选	指定MCT中的mp4转码模板	-	-
clipId	String	可选	查询裁剪MP4结果的id	-	-

请求示例

m3u8裁剪：

```
POST /v5/recording/clip HTTP/1.1
content-length: 444
accept-encoding: gzip, deflate
x-bce-date: 2015-07-03T09:28:13Z
connection: keep-alive
accept: */*
user-agent: python-requests/2.4.0 CPython/2.7.9 Darwin/14.3.0
host: lss.bj.baidubce.com
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715

{
  "playDomain": "play.test.com",
  "app": "testapp",
  "stream": "teststream",
  "filename": "teststream_clip"
  "format": "m3u8",
  "startTime": 1510627184,
  "endTime": 1510628184,
  "sourceFile": "yourfile/teststream_clip.m3u8"
}
```

MP4裁剪：

```
POST /v5/recording/clip HTTP/1.1
content-length: 444
accept-encoding: gzip, deflate
x-bce-date: 2015-07-03T09:28:13Z
connection: keep-alive
accept: */*
user-agent: python-requests/2.4.0 CPython/2.7.9 Darwin/14.3.0
host: lss.bj.baidubce.com
x-bce-request-id: 8776558c-81d9-4f97-8e2c-f977a286095d
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715

{
  "playDomain": "play.test.com",
  "app": "testapp",
  "stream": "teststream",
  "filename": "teststream_clip_mp4"
  "format": "mp4",
  "startTime": 1510627184,
  "endTime": 1510628184,
  "pipeline": "test_pipeline",
  "preset": "test_preset",
  "sourceFile": "yourfile/teststream_clip.m3u8"
}
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
fileUrl	String	m3u8剪裁：完成裁剪的视频的BOS地址
clipId	String	MP4裁剪：用于查询裁剪的MP4结果的id，与最终返回的jobId对应

说明：

m3u8剪裁：

- 1. 录制模板需要是m3u8格式，开启一直合并。
- 2. 录制的bucket请设置为公共读，否则返回的链接不能直接访问。
- 3. 如果正在推流时调用本接口，则只能裁剪本次推流之前的录制视频，所以注意指定对应的时间。
- 4. 如果录制的视频时间过长，本接口会有延时。

MP4剪裁：

- 1. preset尺寸伸缩策略需要设置为自适应伸缩（ShrinkToFit）。
- 2. 通过jobId可以从 MCT 查询裁剪后MP4的地址。
- 3. 转码队列的bucket要与LSS直播中存放录制视频的bucket一致。
- 4. 获取clipId后，需要再次请求该接口并带上clipId参数，以获取 MCT 服务的jobId。

响应示例

m3u8裁剪：

```
{
  "fileUrl": "http://bos-recording-test.bos.gz.qasandbox.bcetest.baidu.com/teststream_clip.m3u8"
}
```

mp4裁剪：

```
{
  "clipId": "job-hkvsk1j9gnc19cit"
}
```

日志下载接口

🔗 日志下载接口

接口描述

提供日志下载。

请求语法

```
GET /v<version>/statistics/domain/<playDomain>/originallogs?startTime=<startTime>&endTime=<endTime> HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: <bce-authorization-string>
```

请求头域

无特殊请求头域。

请求参数

参数	类型	描述	是否必须
playDomain	String	直播域名	是
startTime	String	起始时间，UTC时间。如：2017-02-03T15:00:00Z	是
endTime	String	结束时间，UTC时间。如：2017-02-03T15:00:00Z，默认为当前时间	是

请求示例

```
GET /v5/statistics/domain/domain.demo.com/originallogs?startTime=2017-02-03T15:00:00Z&endTime=2017-02-03T15:00:00Z HTTP/1.1
host: lss.bj.baidubce.com
content-type: application/json
authorization: bce-auth-v1/e8e4a9ced6794355a9a1b8a20b58d37b/2015-07-03T09:28:13Z/1800/content-type;host;x-bce-date/4a1692dc4bab84f5801f79ea0c1fece3601cf73ecd94409d2a94b3942b971715
```

响应头域

无特殊响应头域。

响应参数

参数	类型	描述
playdomain	String	查询的域名
totalCount	Integer	日志文件总个数
logEntries	List	日志
+key	String	日志文件名
+url	String	日志文件下载url
+startTime	String	日志文件开始时间，UTC时间
+endTime	String	日志文件结束时间，UTC时间
+size	Integer	日志大小

响应示例

```
HTTP/1.1 200 OK
x-bce-request-id: 8dbde9a9-34e8-4071-afa0-b6d4c4e974b2
Date: Thu, 28 Jan 2016 07:19:17 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache
{
  "domain": "domain.demo.com",
  "totalCount": "1",
  "logEntries":[
    {
      "url":
"http://bos.qasandbox.bcetest.baidu.com/v1/originlog/domain.demo.com/20170919/domain.demo.com_2017_09_19_2
?authorization=bce-auth-v1%2F6107480b80ae42c0a821a5fe53ba615e%2F2017-10-
12T14%3A21%3A45Z%2F360%2F%2Fbes26ad6db26328eb46edc8307d02a83dc9b82e3365e5d6320f2b09d51f75c6f",
      "key": "originlog/domain.demo.com/20170919/domain.demo.com_2017_09_19_21_00.gz",
      "startTime": "2017-09-19T13:00:00Z",
      "endTime": "2017-09-19T13:59:59Z",
      "size": 233
    }
  ]
}
```

文档更新记录

-  2018-06-28
- 录制视频裁剪接口支持sourceFile字段，可以指定裁剪文件名称。
-  2018-03-01
- 新增重置特定stream接口。
-  2018-01-23
- 新增为直播中的stream添加metadata信息接口。
-  2018-01-10
- 支持查询domain上行流量和带宽接口。
 - 新增域名下实时直播源信息接口。
-  2017-11-30
- 新增查询活跃的Stream接口。
-  2017-11-23
- 新增录制视频裁剪接口。
 - 实时直播源信息接口新增publishIP字段。

视频专区

产品简介

-  百度智能云直播LSS简介
- 百度智能云音视频直播LSS介绍，帮助您快速了解LSS。



操作指南

音视频直播LSS新手入门

- 指导用户快速在百度智能云LSS平台上开启直播,演示完整的推流和播放操作



音视频直播LSS服务开通

- 帮助用户快速开通音视频直播LSS服务



常见问题

常见问题总览

使用类问题

- [创建直播域名时出现报错 push should not equals to play](#)
- [LSS海外节点的支持情况](#)

- [hls、flv支持https访问么？](#)
- [如何利用百度智能云服务快速开发互动直播功能？](#)
- [如何实现二个不同的直播房间，使用同一个推流地址？](#)
- [LSS如何视频截图？](#)
- [如何监控直播内容？](#)
- [有没有美颜功能？](#)
- [视频剪辑功能怎么实现？](#)
- [怎么通过使用摄像设备进行视频源采集、直播？](#)
- [如果想在百度web播放器上做清晰度选择应该怎么做？](#)
- [LSS支持RTSPhttps拉流么？](#)
- [直播视频能否嵌入到自己建设的网站？](#)
- [直播中断多次生成多个视频文件，能自动合并成一个吗？](#)
- [用LSS推流直播必须用转码模板吗？](#)
- [是否支持纯音频直播？](#)
- [LSS支持弹幕吗？](#)
- [怎么查看是哪个流在跑流量？](#)
- [什么是转发模式和转码模式？](#)
- [转发模式和转码模式的区别是什么？我该如何选择？](#)
- [现在支持哪些直播源？](#)
- [如何保护直播版权，同时防止非授权用户访问？](#)
- [我应该设置多大的推流码率？](#)
- [如何录制直播视频？](#)
- [个别用户视频直播有时会有卡顿？](#)
- [如何完成LSS推流域名和播放域名的CName设置？](#)
- [LSS可以使用未备案的域名吗？](#)
- [使用LSS如何新建一个直播流？](#)
- [查询实时Stream统计数据时，报Exception是为什么？](#)

🔗 性能类问题

- [会话详情中的并发连接数峰值是什么意思？](#)
- [直播画面在播放的时候不清晰](#)
- [直播视频的同时在线观看人数有限制么？](#)
- [直播视频的延迟如何？](#)
- [如何优化端到端的直播延迟？](#)

- [如何加速RTMP首屏加载时间？](#)
- [直播视频的清晰度如何？](#)
- [是否需要启用CDN加速？](#)

🔗 计费类问题

- [如何更改计费方式？](#)
- [直播消耗的流量是不是包含了用户观看的流量和拉取摄像头图像的流量？](#)
- [百度智能云LSS收费是什么样的？](#)
- [流量和带宽计费为什么有默认带宽上限？](#)
- [欠费后数据会丢失吗？](#)

🔗 API、SDK使用类问题

- [LSS API的回调地址在哪里配置？](#)
- [有web端的推流SDK吗？](#)
- [百度SDK集成失败](#)
- [百度智能云web播放器SDK可以隐藏暂停键按钮么？](#)
- [安卓、IOS推流端SDK是否支持手机屏幕推流直播？](#)
- [API和SDK不能调用缩略图？](#)
- [通过API接口，怎样获得每个直播会话ID的每日使用流量？](#)

🔗 安全类问题

- [Access Key 在哪查看？](#)
- [如何防止别人往我的地址推流？](#)

使用类问题

🔗 创建直播域名时出现报错 push should not equals to play

错误原因：LSS：push should not equals to play是由于您的推流地址和播放地址是同一个地址导致，推流地址不可以和播放地址是同一个域名的，更正地址即可。

🔗 LSS海外节点的支持情况

目前已支持中国大陆、香港、台湾、新加坡、美东、美西、德国、日本、泰国多个国家和地区的接流和分发，轻松实现跨境直播。如您有其他国家的直播需求，可提交[工单](#)咨询。

🔗 hls、flv支持HTTPS访问么？

目前flv、hls支持https播放地址。rtmp暂时不支持https播放地址。如需实现https，需要登录控制台-[lss-域名管理](#)-点击相应的域名-输出设置-点击https配置对应证书即可。

🔗 如何利用百度智能云服务快速开发互动直播功能？

百度智能云RTC产品提供直播连麦等互动功能，客户可用RTC实现主播和观众的连麦互动，再通过LSS进行直播分发。

🔗 如何实现二个不同的直播房间，使用同一个推流地址？

您可通过设置不同的Stream来实现二个不同的直播房间，使用同一个推流地址，可参考[文档](#)。

🔗 LSS如何视频截图？

LSS截图（缩略图）功能目前只支持两种方式：

1. 可以在模版中进行设置截图的具体时间间隔、高度宽度，参考[缩略图模板](#)。
2. 可以通过API的方式，详细的接口介绍和代码示例可以参考[缩略图模板接口](#)。

🔗 如何监控直播内容？

LSS提供黄反审核功能，助您定制自己的黄反审核策略，从而有效过滤黄色、反动相关内容。

🔗 有没有美颜功能？

LSS通过推流端SDK提供美颜滤镜，支持美白、磨皮、粉嫩等多项美颜功能，目前只有推流安卓SDK和推流IOS SDK支持。

🔗 视频剪辑功能怎么实现？

LSS支持将直播过程中的视频流录制存储到VOD(音视频点播)或BOS(对象存储)，在新建直播会话时，可选择同步录制功能，参考：[录制模板](#)。如果您的视频将用于点播回放推荐存储到VOD，如果仅用于备份存储亦可选择存储到BOS。如果不需要录制取消绑定录制模板即可。若是保存在vod中，只支持mp4格式，bos中支持m3u8以及flv格式。

🔗 怎么通过使用摄像设备进行视频源采集、直播？

如果是想要通过LSS对摄像头所拍摄的视频进行直播，我们的LSS支持多协议，输入：支持RTMP推流，RTMP/RTSP/HLS拉流，输出：支持RTMP、HLS、FLV输出音视频流。需要您自己确认设备是否支持RTMP推流或RTMP/RTSP/HLS拉流，现在以拉流为例，可在LSS页面创建一个RTMP协议的拉流域名，拉流源站为推流播放域名，创建完成后，基于该拉流域名新建流，拉流地址为推流播放地址，而推流播放地址和域名您可在摄像头上查看到，或者联系摄像头出厂商，相关的说明和创建步骤，您可参考[相关文档](#)。

🔗 如果想在百度WEB播放器上做清晰度选择应该怎么做？

如果要使用web播放器的清晰度选择功能，需要在创建直播模板的时候，打开hls的adaptive参数，这样系统会在直播过程中给您转换出3个清晰度的视频，当web播放器播放直播的url时，会让您选择相应的清晰度。

🔗 LSS支持RTSP HTTPS拉流么？

目前LSS支持RTMP推流，在stream级别支持RTMP/HTTP/RTSP拉流；域名级别支持RTMP/HTTP-FLV/HTTP-HLS拉流。其他暂不支持，后续关注官网更新即可。

🔗 直播视频能否嵌入到自己建设的网站？

直播可以嵌入到网站中，请参考[WEB SDK文档](#)。

🔗 直播中断多次生成多个视频文件，能自动合并成一个吗？

直播由于网络的原因，或者其他的一些原因，被迫中断，从而形成了很多视频碎片，想要合并，可以在录制模板中选择bos存储M3U8格式，即可选择自动合并时间间隔。如果是直接放置在VOD中，那么是不支持在线合并的。详细参考 [录制模板](#)。

🔗 用LSS推流直播必须用转码模板吗？

转码并非硬性要求，转码模板可以配置音视频编码标准、码率、输出流类型等基本参数，可以使用forward_only的模板即可不转码（该模板不收费）。具体转码模式使用场景参考[模板管理](#)，转码计费详情请参考[转码计费](#)。

🔗 是否支持纯音频直播？

目前只有Android推流SDK有提供sdk接口修改参数实现纯音频直播。

🔗 怎么查看是哪个流在跑流量

您可以查看一下LSS管理控制台的统计分析中，看看是那个域名在进行推流，以及可以通过日志管理中查看一下具体使用中的流地址。

🔗 什么是转发模式和转码模式？

LSS包括两种直播模式类型：

1. 转发模式（新建直播流时选择live.rtmp_hls_forward_only或live.rtmp_forward_only模板），该模式不作实时转码处理，不产生转码计费；
2. 转码模式（新建直播流时选择的转码模板），该模式对输入码流进行实时转码处理，包括码率、分辨率、编码标准等流处理功能。

转发模式时forwardOnly设置为true；转码模式时forwardOnly设置为false，同时配置video/audio等转码相关参数。

LSS针对不同场景内置了转码模板，可以直接选取使用。

🔗 转发模式和转码模式的区别是什么？我该如何选择？

如果您使用下列直播场景时，建议采用转码模式：

1. 需要多码率/多清晰度支持；
2. 需要添加水印（图片水印、时间戳水印）；
3. 需要对直播流加密；
4. 需要将高码率视频转码成低码率视频流的；
5. 采集源端编码器性能较差，压缩率较低，需要服务器端进一步压缩的（建议将上传码率设置为转码输出码率的120%~130%）；
6. 输入流是非H.264（视频）或者非AAC（音频）编码；
7. 需要更精确更流畅更稳定的码率控制；
8. UGC场景中需要统一的输出分辨率码率等（输入流会因为设备、平台不同而不同）；

🔗 现在支持哪些直播源？

LSS支持RTMP推流，在stream级别支持RTMP/HTTP/RTSP拉流；域名级别支持RTMP/HTTP-FLV/HTTP-HLS拉流。

1. RTMP推流：

RTMP推流的地址可以通过百度智能云LSS控制台或者LSS的SDK/API来创建一个直播流Stream获得。推流包括以下几种形式：

- 台式计算机或笔记本，使用摄像头或者桌面

可以使用LSS的Windows端的推流SDK（或工具）或者第三方软件采集摄像头视频或桌面图像，将视频或桌面推流至LSS的RTMP推流地址。相关的第三方推流软件包括：OBS(推荐),FMLE,XSplit等。

- Android/iOS, 使用手机摄像头

可通过使用第三方软件或SDK采集摄像头视频，并推送视频流至LSS的RTMP推流地址。iOS端的RTMP推流工具包括EZWebPlayer, broadcast me等；Android端的RTMP推流工具包括arutcam、video broadcaster+、os broadcaster、broadcastme、android broadcast、tikilive、rtmp streamer mini等。百度智能云LSS也推出了基于Android/iOS的推流SDK。

- 视频采集设备如高清摄像机

该类型设备如果具备HDMI或者SDI输出接口，可以接入编码器，以RTMP推流的方式向LSS发布直播内容。您需要将LSS提供的推流地址配置到编码器的RTMP发布地址。

- 嵌入式Linux平台

Linux平台可以使用ffmpeg来采集视频源，并推送至LSS的RTMP推流地址，比如通过如下命令行：

```
ffmpeg -f v4l2 -r 25 -s 640x480 -i /dev/video0 -f flv {your_rtmp_address}
```

参考链接：<https://trac.ffmpeg.org/wiki/Capture/Webcam#Linux>

您也可以在程序中调用ffmpeg的API来进行RTMP推流。

- 播放视频文件

通过读取某个视频文件作为视频源来向LSS的RTMP推流地址进行视频发布。使用ffmpeg命令来实现(Windows/Linux/Mac均适用)，比如通过如下命令行：

```
ffmpeg -re -i {your_video_file} -r 25 -s 640x480 -f flv {your_rtmp_address}
```

2. RTMP/HTTP/RTSP拉流:

使用RTMP/HTTP/RTSP拉流，用户只需填写已有直播源的拉流地址(公网可访问)，开启拉流操作后，系统会从该地址自行拉取音视频流。

- RTMP既可以推流又可以用来直播拉流，一般传输的是FLV，F4V格式流。
- HTTP将所有的数据作为文件做处理，没有特定的传输流。
- RTSP主要用来控制具有实时特性的数据发送协议，一般传输TS，MP4格式的流。

🔗 如何保护直播版权，同时防止非授权用户访问？

LSS支持推流认证、播放认证、内容加密和防盗链等多种安全策略，保护您的直播版权不受侵犯。

- 推流认证

为确保推流频道不被非法的占用，防止非法推流，音视频直播LSS支持采用token认证方式与有效期限制结合的推流地址。

- 播放认证

为了避免播放地址长期暴露带来风险，LSS支持播放地址加密策略。开启播放认证后，用户可设置播放地址的有效期，并且需要加入播放token才可以进行播放。

- 内容加密

针对HLS直播视频流LSS提供基于AES128加密视频流和密钥访问控制的DRM方案；RTMP直播视频流暂不支持DRM方案。

- 防盗链

支持用户在LSS控制台设置防盗链Referer/IP黑白名单功能。

🔗 我应该设置多大的推流码率？

根据实践经验列举了分辨率与推荐上传码率的对应关系，供参考：

分辨率	推荐视频码率 (kbps)	推荐音频码率 (kbps)
240p (426X240)	240	40
360p (640X360)	440	80
480p (852X480)	800	80
720p (1280X720)	1200	128
1080p (1920X1080)	2000	128

说明：

视频内容中不同运动强度的直播需要的码率有所差别，对低运动（low motion）的直播内容（比如固定摄像机的讲座内容）等可以适当减小码率，针对高运动（high motion）的直播内容（比如足球比赛或者娱乐活动等）可以适当增大码率。

🔗 如何录制直播视频？

LSS支持将直播过程中的视频流录制存储到音视频点播VOD或BOS，在新建直播流时，可选择同步录制功能。如果您的视频将用于点播回放推荐存储到VOD，如果仅用于备份存储亦可选择存储到BOS。

🔗 个别用户视频直播有时会有卡顿？

影响直播卡顿的因素包括，直播源处理能力、上传网络稳定性、播放端所在设备处理能力、播放端所在网络稳定性等。

您可以通过以下方式解决直播卡顿现象：

- 改变直播源编码参数
- 减少上传码率
- 改变上传网络环境
- 改变播放端设备配置
- 改变播放端网络环境等

如果问题依然存在，请提[工单](#)，我们会尽快处理。

🔗 如何完成LSS推流域名和播放域名的CNAME设置？

请您参考[配置CNAME](#)。

🔗 LSS可以使用未备案的域名吗？

不能。请您先参考[备案指南](#)完成备案。

🔗 使用LSS如何新建一个直播流？

在您的域名下按需定义推流路径即可，无需显式新建一个直播流。

🔗 查询实时Stream统计数据时，报Exception是什么原因？

查询指定domain、app下所有stream的实时统计数据时，仅对streamingStatus=STREAMING的stream有效，否则会报Exception。

🔗 直播回看需要开通吗？

LSS直播服务支持开通直播回看和时移功能，请提交[LSS直播工单](#)，工单内提供userid、hls_pull域名和rtmp域名。

🔗 如何利用百度智能云服务快速开发互动直播功能？

目前LSS直播不支持音视频互动，只支持单向。百度智能云[RTC产品](#)提供直播连麦等互动功能，客户可用RTC实现主播和观众的连麦互动。

🔗 百度智能云如何拉流？

我方的LSS支持RTMP推流，在stream级别支持RTMP/HTTP/RTSP拉流；域名级别支持RTMP/HTTP-FLV/HTTP-HLS拉流。其他暂不支持，后续您关注官网更新即可。

🔗 百度智能云直播是否提供录制功能？

LSS支持将直播过程中的视频流录制存储到VOD(音视频点播)或BOS(对象存储)，在新建直播会话时，可选择同步录制功能，参考：[录制模版](#)。如果您的视频将用于点播回放推荐存储到VOD，如果仅用于备份存储亦可选择存储到BOS。如果不需要录制取消绑定录制模板即可。若是保存在VOD中，只支持mp4格式，BOS中支持m3u8以及flv格式，关于BOS的价格介绍请查看：[产品定价](#)，关于VOD存储收费参考：[产品定价](#)。

🔗 LSS支持将录制文件上传到第三方存储吗？

目前暂不支持转储到第三方存储。

🔗 服务器接收到远程摄像头的视频流推送到百度智能云上应该怎么做？

如果是想要通过LSS对您摄像头所拍摄的视频进行直播的话，我们的LSS支持多协议，输入：支持RTMP推流，RTMP/RTSP/HLS拉流，输出：支持RTMP、HLS、FLV输出音视频流。需要您自己确认设备是否支持RTMP推流或RTMP/RTSP/HLS拉流，现在以拉流为例，可在LSS页面创建一个RTMP协议的拉流域名，拉流源站为推流播放域名，创建完成后，基于该拉流域名新建流，拉流地址为推流播放地址，而推流播放地址和域名您可在摄像头上查看到，或者联系摄像头出厂商，相关的说明和创建步骤，您可以参考文档：[操作指南](#)。推荐您使用 [EVS 视频监控产品](#)，更适用于监控场景。能如果在使用过程中遇到了问题建议您提交音视频直播/LSS工单，提供：1.推流地址（拉流地址）2.播放地址3.具体报错截图我方会有工程师为您处理。

🔗 企业用户没有,网络许可证,只做企业实名,是否可以以音视频屏直播？

可以的，目前用户完成企业认证就没有流量上限了，也建议您最好完成资质认证。

🔗 音视频直播LSS直播可以直接录制吗？

音视频直播LSS，在直播时是可以直接录制的，如图所示，您先在控制台创建录制模板，然后在您的域名下绑定这个模板即可。录制功能2022年1月起开始计费，具体计费规则请参考：[录制计费](#)。





🔗 LSS 查询所有Stream数据接口中downstreamInByte参数是什么单位？

音视频直播LSS中，[查询所有Stream数据](#)接口的downstreamInByte参数单位是字节，转换是按1000进制转换。

🔗 音视频直播推流和播放域名需要做CDN加速吗？

音视频直播LSS，添加上域名后，默认就会进行CDN加速的，无需单独配置CDN。

🔗 有接口可以查询推流域名对应的播放域名吗？

十分抱歉，LSS的API暂不支持获取推流域名对应的播放域名，后续您可以关注百度智能云官网。

🔗 LSS中flv和hls是否支持https方式进行播放？

音视频直播LSS中，针对flv和hls播放流可以开启[https功能](#)，开启后，可以通过https方式进行播放。

🔗 音视频直播LSS,对于已经录制到bos的文件是否有多文件合并以及裁剪的API？

目前音视频直播LSS暂时没有对于已经录制到BOS的文件是否有多文件合并以及裁剪的API，后续您可以关注官网。

🔗 播放器PC端还是flash插件吗？

目前浏览器播放rtmp的流是要用到flash插件，无法通过其他方式播放，建议可以评估以下方案处理：

1. 可以固定使用支持flash的浏览器；
2. 更换以hls 或 flv协议播放

🔗 如何在手机浏览器中实现推流？

百度智能云LSS直播服务，目前百度智能云没有手机浏览器推流的工具。手机移动端推流可参考：[短视频SDK直播推流版](#) 如还有问题请提交 [LSS直播服务工单](#)。

🔗 如何获取LSS流状态？

如您想查询对应的流是否是直播的状态，可以使用查询特定stream接口进行获取对应stream状态，具体请参考：[查询特定stream](#)

🔗 BCC服务器是否支持通过内网向LSS直播推流？

目前并不支持bcc通过内网向lss进行推流。

🔗 LSS是否可以提供测试域名进行测试直播？

目前LSS并没有测试域名提供进行使用，需要您绑定自己的域名进行测试，注：使用Iss的前提必备条件：已经备案的域名（任何厂商域名均可），请您知悉。

🔗 如何使用obs进行推流？

目前音视频直播Iss是提供了obs进行实践文档，您可以参考：[OBS推流使用指南](#)

🔗 LSS流都有哪些状态？

目前Iss流状态包含：已就绪、直播中 有输入流、直播中 无输入流、已禁播、失败、拉流尝试中。

🔗 做了一个5分钟的安全策略有效期限限制，但实际无效，一直在推流？

关于此问题，其有效期是指在5分钟内可以发起推流操作，超过5分钟后，该url就无法进行推流。并非是指您推流过程中，达到5分钟断开推流。

性能类问题

🔗 直播画面在播放的时候不清晰

针对于模糊、不清晰的问题，首先检查下本地的录像设备是否正常，更换录像设备后再次尝试，另外视频内容中不同运动强度的直播需要的码率有所差别，对低运动（lowmotion）的直播内容（比如固定摄像机的讲座内容）等可以适当减小码率，针对高运动（highmotion）的直播内容（比如足球比赛或者娱乐活动等）可以适当增大码率。

🔗 直播视频的同时在线观看人数有限制么？

目前我们的直播服务没有人数限制，理论上可以支持百万级到千万级用户同时在线观看。

🔗 直播视频的延迟如何？

直播视频的端到端延迟影响因素包括采集端、网络传输、播放端以及采用的输入输出协议。

直播中延迟的影响因素包括下列几种：

1. 推流端设备的性能与网络状况。
2. 播放端设备的性能与网络状况。
3. 所使用的播放协议：一般来说，RTMP的延迟在2~4秒左右，HLS的延迟在10秒~20秒左右。RTMP输出流延迟会显著低于HLS输出流。

🔗 如何优化端到端的直播延迟？

在保障高质量流畅播放的前提下并取得较低延迟，您可以采用下列方法，对推流端、服务端、播放端分别设置。

推流端：

在推流端我们建议使用如下参数：

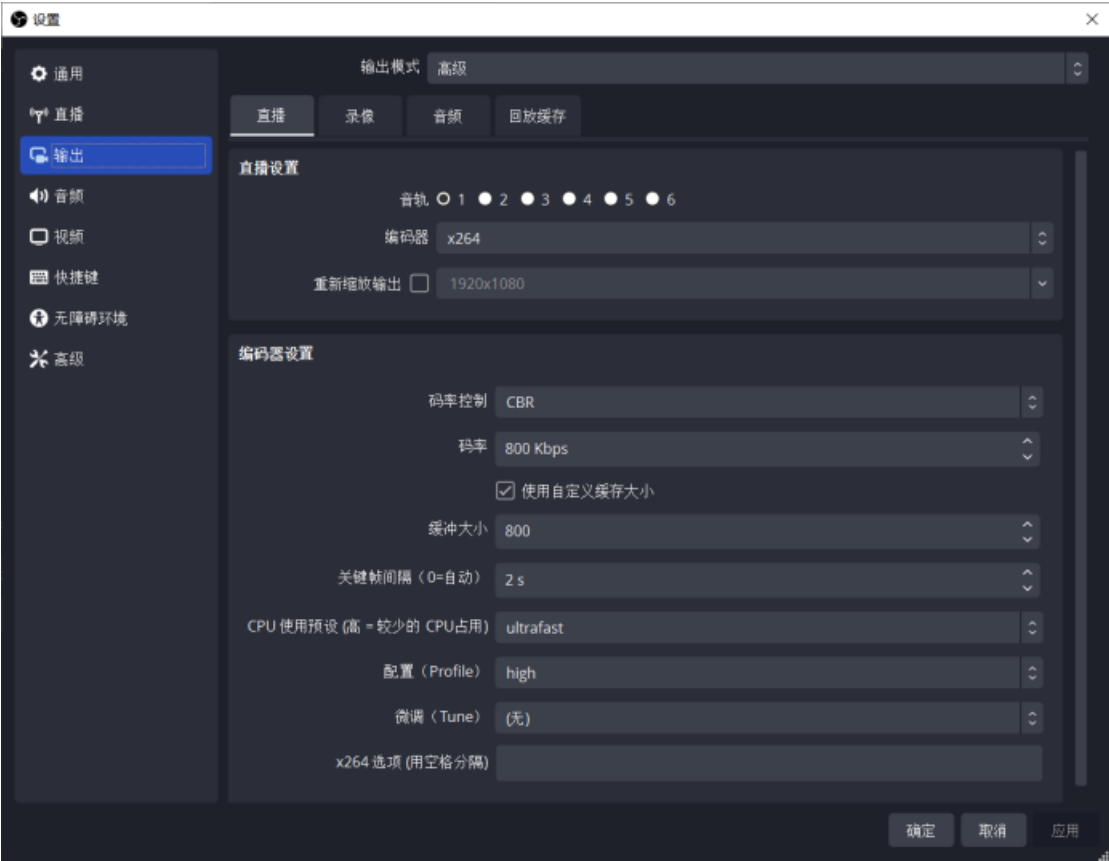
配置参数	推荐值
码率控制	不使用CBR，设置自定义缓冲区等于编码码率。
x264 CPU 预设（x264 preset）	ultrafast
编码配置文件（H.264 profile）	high
关键帧间隔（GOP length）	3秒或者（75~90帧）

推荐按照如下分辨率和码率的关系设置编码参数：

清晰度	分辨率	视频码率 (kbps)	音频码率 (kbps)	帧率 (fps)
超高清1080p	1920X1080	2000	128	30
高清720p	1280X720	1200	128	30
标清480p	852X480	800	96	25
标清360p	640X360	440	96	20

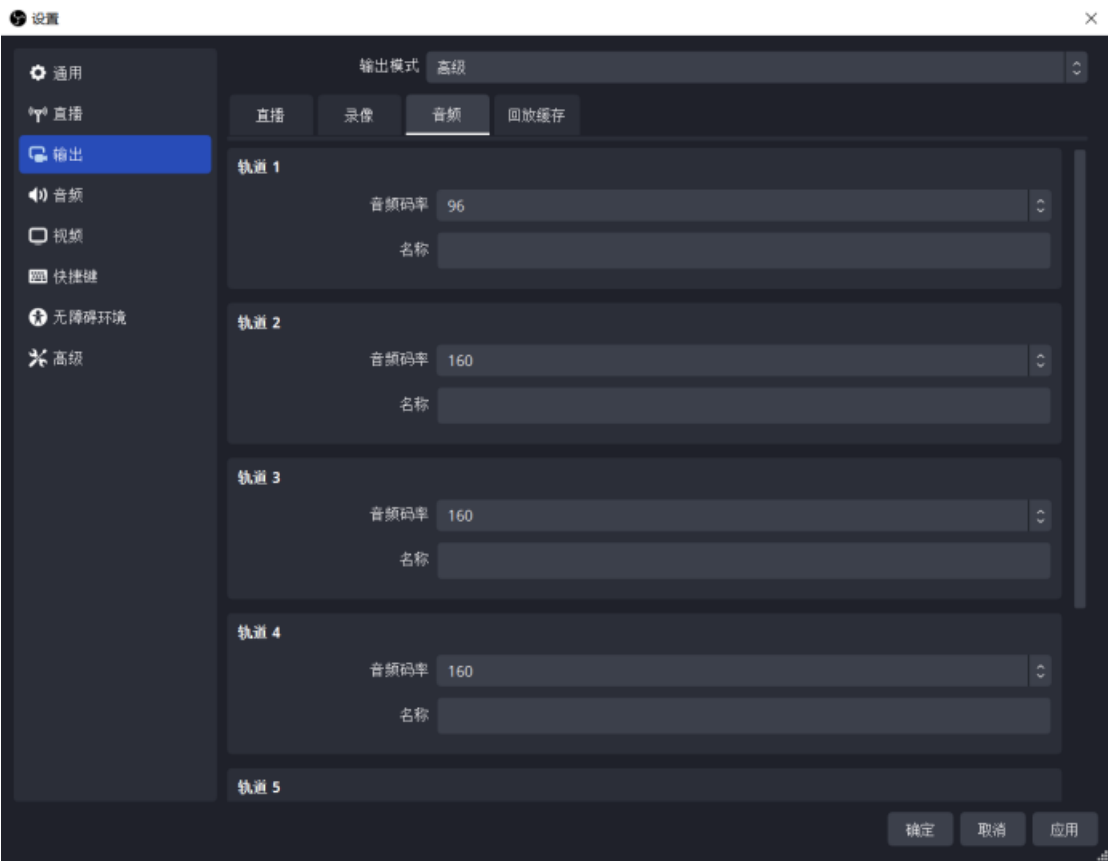
例如采用OBS进行推流时，设置如下参数：

- 1.以直播采用标清480p为例，菜单栏选择“文件”->“设置”->“输出”，输出模式选择“高级”，设置视频码率，设置码率为800kbps，缓冲大小800，关键帧间隔为2秒，X264 CPU预设为ultrafast, 编码配置文件为high，如下图所示：



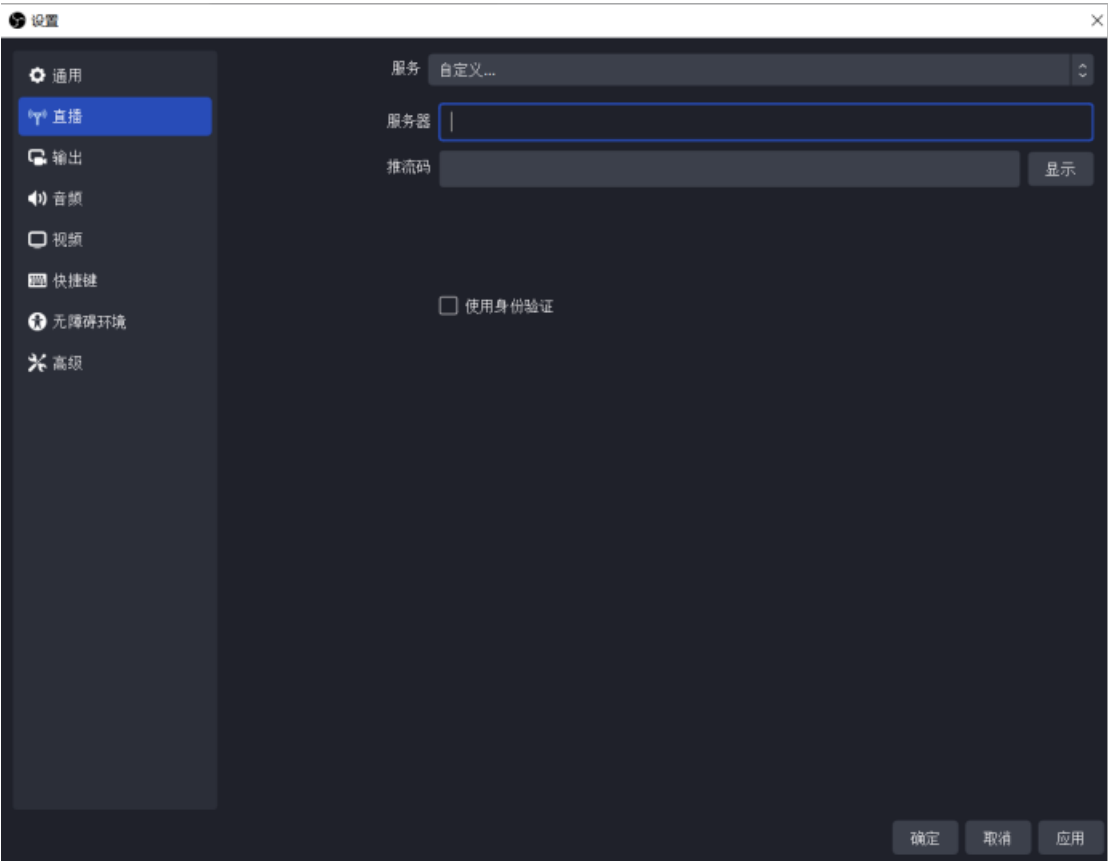
在音频选项

卡中选择推荐的音频码率，本例期望音频码率96，如下图所示：



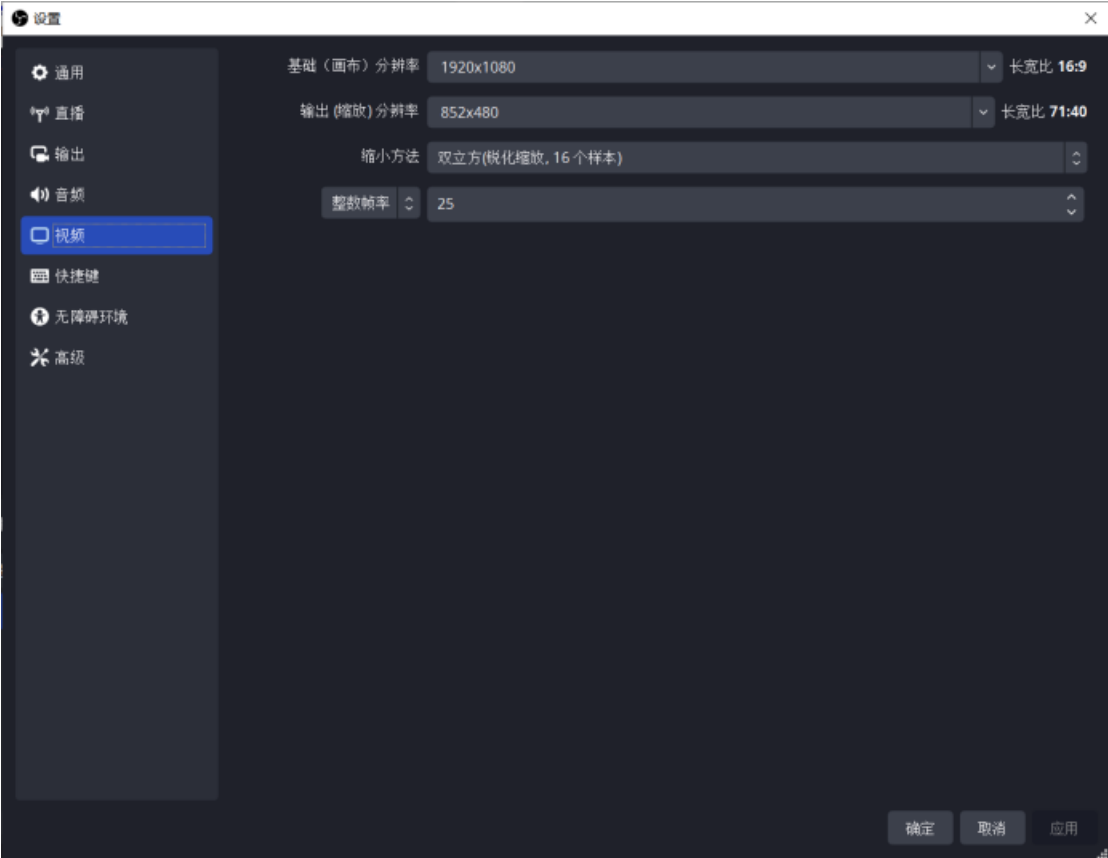
2.菜单栏“文

件”->“设置”->“直播”，服务器输入推流地址的URL，推流码输入token密钥，如下图所示：





3、菜单栏“文件”->“设置”->“视频”中设置压缩分辨率，您可以从压缩分辨率选择与自己期望最接近的一项。本例期望分辨率为852x480，期望帧率为25，下图所示：



说明：移动端用户建议您采用百度推出的[推流端SDK](#)，可以更好的减少端到端的延迟。至此，推流端(OBS)完成参数配置。

服务端：

您还可以在创建转码模板时，对模板参数进行设置，从而降低延迟。

HLS直播输出流，对HLS切片设置，分片时长设为3秒，TS切片数为3~4个，如下图所示：

基本信息

样例模板：

live.hls_854x480

*模板名称：

hls_854x480

可由小写字母、数字和下划线组成，必须以小写字母开头，最长40个字符。

*模板描述：

标清480p，16:9，HLS输出

*直播类型：

☐ 仅转发

☒ 实时转码

*直播输出流：

☐ RTMP

☒ HLS

*HLS分片时长：

3

秒 默认10秒。

*TS切片数：

4

个 默认5个。

m3u8文件中TS切片个数，默认5个

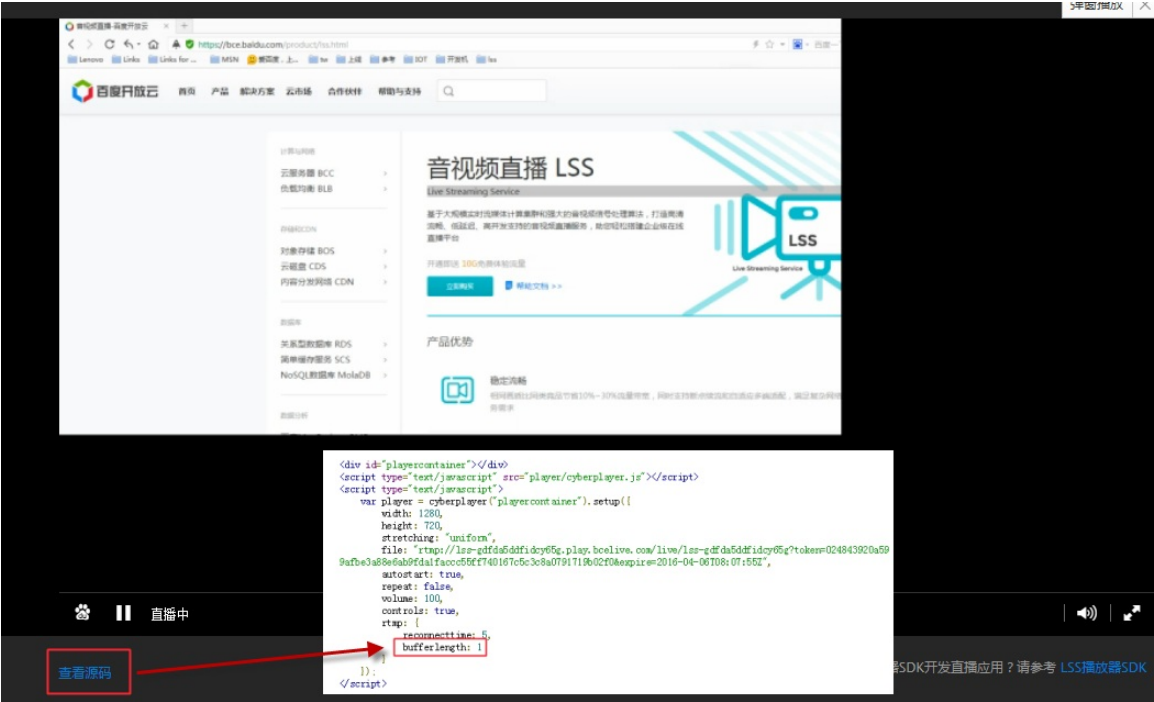
*支持多码率：

☒ 否

开启多种码率输出，播放端可根据接入网络状况实现多码率无缝切换。

播放端：

播放器缓存越长，端到端延迟越大。Flash播放器支持对最低缓冲长度的设置，建议用户在RTMP直播中使用1~2秒的缓冲长度，以百度Web播放器为例，建议设置缓冲长度为1s，如下图所示：



移动端用户采用下列播放设备，可以有效减少直播延迟时间：

- IOS用户推荐采用iphone 5及以上设备，系统支持IOS 8.1以上。
- Andriod用户推荐使用1~2年内生产的主流机型。

如何加速RTMP首屏加载时间？

LSS支持在CDN节点缓存最新的GOP数据（从关键帧开始的音视频数据），使得播放端能够快速加载视频画面。

系统默认开启缓存GOP，无需手动配置。凭借百度智能云的“追赶播放”专利技术，用于追赶播放当前最新帧之前的GOP缓存帧，

以达到最小的端到端延迟，可在加速RTMP首屏加载时间的同时，不增加端到端的延迟。

直播视频的清晰度如何？

转发模式的直播，清晰度取决于视频输入端的清晰度，服务本身对清晰度未做处理；转码模式的直播，清晰度取决于视频输入端及模板中转码的设置。

是否需要启用CDN加速？

不需要。LSS已默认启用CDN加速。

LSS是否支持srt推流？

百度智能云音视频直播LSS暂不支持srt推流，目前只有rtmp推流。

计费类问题

如何更改计费方式？

如果需要更改LSS计费方式，可以登录下百度智能云控制台-音视频直播LSS-域名管理-点击变配即可，次日零点生效。

直播消耗的流量是不是包含了用户观看的流量和拉取摄像头图像的流量？

直播消耗的流量只包含了用户观看的流量，拉取摄像头图像的流量这边是不包含的，上传流量是不计费的。

百度智能云LSS收费是什么样的？

LSS直播费用分为分发费用、转码费用和增值服务费用。直播服务推流和拉流的流量是不收费的。分发费用分为按流量计费、按日峰值带宽计费、大客户按月计费，参考[分发计费](#)。转码费用是后付费，除了forward_only只转发不做转码，使用其他模板均收取转码费用，详情参考[转码计费](#)。录制模板可以选择将直播内容同步录制到对象存储BOS或音视频点播VOD，录制的路数和缩略图模板等属于增值服务，详情参考[增值服务计费](#)。您选择录制存储到BOSbucket时，会收取您BOS的费用，详情参考[BOS存储价格](#)、[VOD存储价格](#)。

流量和带宽计费为什么有默认带宽上限？

LSS 按流量计费和日峰值带宽计费默认带宽上限，主要是为了避免非法内容传播，如果您的您有更高的带宽需求，并且具备相应的网络试听资质，请提交工单申请，LSS可支持百万级并发观看。

欠费后数据会丢失吗？

欠费处理：

- 当您的账户余额不足0元无法支付LSS账单，即为欠费，欠费时系统会发送欠费通知。
- 欠费后立即停服，系统会发送欠费停服通知，cache资源被回收，请求指向源站，配置信息保留12个月。

LSS推流会收取流量费用吗？

LSS直播服务流量是只计算下行流量费用，也就是观看直播产生的流量。推流属于上行流量，不会计算流量费用。参考[LSS分发计费文档](#)，如还有问题请提交[LSS直播工单](#)。

LSS流量包是指用于 LSS 客户端拉流时所产生的的下行流量么？

目前LSS流量包是抵消客户端播放LSS播放地址产生的下行流量费用。

LSS是否会对创建直播路数数量收费？

音视频直播LSS是后付费产品，分为分发费用、转码费用和增值服务费用。直播路数收费可以参考：[增值服务计费](#)。同时，分发和转码分别提供了对应预付费资源包，可以进行抵消费用，具体请参考：[计费概述](#)。

有多路的直播流需要转码，是产生多次转码费用，还是产生一次？

多路进行直播，且每路设置转码模版，那么每路都会产生转码费用和录制费用，您可以根据每路不同的转码规则进行购买对应转码资源包。转码费用具体不同规格价格请参考：[价格](#)。

API、SDK使用类问题

🔗 LSS API的回调地址在哪里配置？

回调地址在LSS创建session时进行配置。页面中有个“通知”，需要勾选。它是根据http的post和get请求来完成的。在配置中需要配置一个绝对路径。当server get到这个状态变化后，会生成一个post.log文件。可以通过这个log文件，获取信息。

🔗 有web端的推流SDK吗？

目前提供推流SDK（安卓，ios），播放器SDK（web，安卓，ios），服务端SDK（java，php），其他语言SDK暂时还不支持。

🔗 百度SDK集成失败

LSS的SDK对于集成没有限制。如您使用demo正常，集成报错，优先建议您按照demo集成，或把项目集成到demo工程中。如您使用demo异常，建议您提交音视频直播/LSS工单，并详细描述您的问题：1.提供使用的SDK语言和类别 2.提供具体报错截图 3.提供最小复现代码。4.是否稳定复现5.如果非稳定复现只在个别机型复现，提供下该机型的具体型号以及操作系统版本，我方工程师会为您排查指导。

🔗 百度智能云web播放器SDK可以隐藏暂停键按钮么？

web播放器SDK取消暂停按钮和取消进度条按钮可以通过配置和修改代码完成，可参考文档 [取消暂停按钮](#)、[取消进度条](#)。

🔗 API和SDK不能调用缩略图？

目前缩略图功能只能通过console来调用了，API只有查询和列出功能，sdk目前还没法使用createPreset接口来使用缩略图，详情请您参考[缩略图模板](#)。

🔗 通过API接口，怎样获得每个直播会话ID的每日使用流量？

您可以调用统计接口中的[特定domain](#)接口。

安全类问题

🔗 Access Key 在哪查看？

可在百度智能云页面右上角的账户信息--安全认证--Accesskey里面创建或获取AK/SK。

🔗 如何防止别人往我的地址推流？

为确保推流地址不被非法占用，LSS提供token认证和有效期限相结合的推流地址。开启推流认证后，您可以通过“推流地址+超时时间+密钥”的方式获取完整的加密推流地址。详情可参考 [推流认证](#)。

🔗 LSS如何关联安全策略？

目前LSS仅支持域名级别关联安全策略，并不支持stream级别关联安全策略，具体请参考：[高级配置](#)。

🔗 推流异常怎么处理？

LSS直播服务推流出现失败，

1、请检查下您的推流域名是否有添加cname解析，域名是否生效，参考[CNAME解析文档](#)

2、请检查您的推流域名是否有开启推流认证，推流认证是否过期失效，参考 [推流认证文档](#)

如还有问题请提交 [LSS音视频直播工单](#)。

服务等级协议SLA

LSS服务等级协议SLA

在使用百度智能云音视频直播LSS服务时，您应当阅读并遵守《LSS服务等级协议》（以下简称“本协议”或“SLA”）。本协议包含音视频直播LSS服务的术语、服务可用性/服务成功率等级指标、赔偿方案等相关内容。请您务必审慎阅读并理解各条款内容、限制、免责条款或者其他涉及您重大权益的条款。百度智能云会适时修改SLA协议，您可以在百度智能云官网查阅相关协议条款。如您不同意百度智能云对协议所做的修改，您有权停止使用音视频直播LSS服务，如您继续使用LSS服务，则视为您接受修改后的协议。

1. 术语

1.1 服务周期

以一个自然月作为一个服务周期，不满一个月则以实际使用日至月底最后一天计算为一个服务周期。假设您于9月19日开始使用直播，则第一个服务周期为9月19日到9月31日。服务可用性按照服务月度单独核算，新一轮服务周期开始后会重新计算当月的服务可用性。

1.2 服务区域

中国大陆地区（不含港澳台地区）。

1.3 失败请求

因音视频直播故障导致的未成功响应的请求，视为失败请求，包括

- (1) RTMP：维持链接20秒未返回数据；
- (2) FLV-HTTP：5XX类错误请求；
- (3) HLS：5XX类错误请求。

注意：错误请求不包含以下类型：

- (1) 客户开启防盗链鉴权，防盗链过期导致的不可播放请求
- (2) 内容违规或其它原因而导致域名被封禁而产生的错误
- (3) 推流端异常导致的播放错误
- (4) LSS服务发起的合理升级、变更、停机而导致的错误请求或服务不可用情况

1.4 有效请求

有效请求指的是LSS服务器接收到的音视频直播基础服务请求，不包含非客户应用程序的黑客攻击而发起的请求。

1.5 每5分钟错误率

百度智能云音视频直播服务统计以5分钟为一个单位，00:00:00时间点代表的时间区间为00:00:00 - 00:04:59，以此类推。单位时间内的错误率计算公式如下：

$$\text{每5分钟错误率} = \frac{\text{每5分钟错误请求}}{\text{每5分钟有效总请求}} \times 100\%$$

1.6 月度服务费用

客户的百度智能云账号在一个服务月度内使用音视频直播LSS服务产生的月度服务费用，主要分为

- 基础服务的月度服务费用：指按流量或带宽统计的流量费用或带宽费用。
- 增值服务的月度服务费用：指除基础服务外的转码、录制、截图和鉴黄等费用。

2. 服务可用性

2.1 服务可用性计算公式

以账号维度分别统计服务周期内的音视频直播基础服务及增值服务的服务可用性。服务可用性将根据服务周期内每5分钟错误率之和除以服务周期内5分钟单位时间总个数计算出每5分钟错误率的平均值，从而计算得出服务可用性，即：

$$\text{服务可用性} = \left(1 - \frac{\text{服务周期内} \Sigma \text{每} 5 \text{ 分钟错误率}}{\text{服务周期内} 5 \text{ 分钟总个数}}\right) \times 100\%$$

2.2 服务可用性指标标准

音视频直播LSS服务可用性不低于99.9%，如LSS服务未达到上述服务可用性承诺，客户可以根据本协议第3条约定获得赔偿。赔偿范围不包括以下原因所导致的请求失败或服务不可用：

- (1) 百度智能云预先通知用户后进行系统维护所引起的，包括合理升级、变更、停机、割接、维修和模拟故障演练；
- (2) 任何百度智能云所属设备以外的网络、设备故障或配置调整引起的不可用；
- (3) 用户内容违规或其他原因导致域名被封禁而产生的错误；
- (4) 用户大规模流量突发增长未提前书面告知百度智能云所导致的可用性降低；
- (5) 用户的应用程序或数据信息受到黑客攻击而引起的；
- (6) 用户维护不当或保密不当致使数据、口令、密码等丢失或泄漏所引起的；
- (7) 用户的疏忽或由用户授权的操作所引起的；
- (8) 非中国大陆内的请求所导致的错误；
- (9) 不可抗力以及意外事件引起的；
- (10) 其他非百度智能云原因所造成的不可用。

3. 赔偿方案

3.1 赔偿标准

- (1) 赔偿以百度智能云发放音视频直播代金券的形式实现，您应当遵守代金券的使用规则（包括使用期限等，具体以百度智能云官网发布的代金券相关规则为准）。发放的代金券不能折现、不开具发票，也不能转让、赠予等，仅限您通过您的百度智能云账户购买音视频直播服务，不能购买其他的百度智能云服务，
- (2) 按基础服务及增值服务分别来计算服务可用性并计算相应赔偿。如果某服务月度没有达到服务可用性标准，赔偿额按照相应未达标服务月度单独计算，赔偿总额不超过相应未达标服务月度内您就本服务支付的相应月度服务费（此处的月度服务费不含用代金券、优惠券、服务费减免等抵扣的费用）。

服务可用性	赔偿代金券金额
低于99.90%但等于或高于 95.00%	月度服务费用的 5%
低于 95.00%	月度服务费用的 10%

3.2 赔偿申请时限

- (1) 客户可以在每月第五（5）个工作日后对上个月没有达到可用性的服务提出赔偿申请。客户提出赔偿申请后百度智能云会进行核实，对于服务月度的服务可用性的计算，若双方出现争议的，应以百度智能云的后台记录为准。
- (2) 客户最晚提出赔偿申请的时间不应超过未达标的相应服务月度结束后60个自然日，未在60个自然日内提出赔偿申请的，视为自动放弃要求赔偿的权利及向百度智能云主张其他权利的权利，百度智能云有权不受理超限的赔偿申请，不进行任何赔偿或补偿。

4. 其他

- (1) 在法律法规允许的范围内，百度智能云负责对本协议进行解释说明。
- (2) 本协议一经公布立即生效，百度智能云有权适时对协议内容进行修改，修改后的结果公布于百度智能云网站上。如果不同意百度智能云对本协议相关协议所做的修改，用户有权停止使用百度智能云音视频直播服务LSS。若用户继续使用音视频直播服务LSS，则视为用户接受百度智能云对本协议相关协议所做的修改。

- (3) 本协议项下百度智能云对于用户所有的通知均可通过网页公告、站内信、电子邮件、手机短信或其他形式等方式进行；该等通知于发送之日视为已送达收件人。百度智能云不对用户承担因此造成的任何损失。
- (4) 本协议的订立、执行和解释及争议的解决均应适用中国法律并受中国法院管辖。如双方就本协议内容或其执行发生任何争议，双方应尽量友好协商解决；协商不成时，任何一方均可向北京市海淀区人民法院提起诉讼。
- (5) 本协议构成双方对本协议之约定事项及其他有关事宜的完整协议，除本协议规定的之外，未赋予本协议各方其他权利。
- (6) 如本协议中的任何协议无论因何种原因完全或部分无效或不具有执行力，本协议的其余协议仍应有效并且有约束力。
- (7) 关于用户约束条款，详见《[用户服务协议](#)》中的"用户的权利与义务"相关条款内容。
- (8) 关于服务商免责条款，详见《[用户服务协议](#)》中的"免责声明"相关条款内容。