

目录

目录	1
直播转码服务概述	4
功能介绍	4
使用方式	4
API使用方法	4
签名机制	4
请求结构	4
SDK调用流程	5
返回结果	5
调用成功	5
调用失败	5
公共参数	5
参数说明	5
示例	5
拉流地址说明	6
1. 拉流触发转码	6
示例如下:	6
2. 推流触发转码	7
示例如下:	7
直播转码错误码定义	7
模板说明	8
直播模板结构	8
Output结构	8
format结构	8
Video结构	9
LOGO结构	9
转码输出格式表	9
模板使用示例	10
创建模板接口(Preset)	12
接口描述	12
请求方式:	12
请求参数:	12
返回参数:	12
请求示例:	12
返回示例:	13
更新模板接口(UpdatePreset)	13
接口描述	13
请求方式:	14
请求参数:	14
返回参数:	14
请求示例:	14
返回示例:	15
删除模板接口(DelPreset)	15
接口描述	15
请求方式:	15
请求参数:	15
返回参数:	15
请求示例:	15
返回示例:	16
查询模板列表接口(GetPresetList)	16

接口描述	16
请求方式:	16
请求参数:	16
返回参数:	16
请求示例:	16
返回示例:	17
查询模板详情接口 (GetPresetDetail)	18
接口描述	18
请求方式:	18
请求参数:	18
返回参数:	18
请求示例:	18
返回示例:	18
查询任务列表接口 (GetStreamTranList)	19
接口描述	19
请求方式:	19
请求参数:	19
返回参数:	19
请求示例:	20
返回示例:	20
发起外网拉流接口 (StartStreamPull)	21
功能描述:	21
请求方式:	21
请求参数:	21
返回参数:	21
请求示例:	21
返回示例:	22
停止外网拉流接口 (StopStreamPull)	22
功能描述:	22
请求方式:	23
请求参数:	23
返回参数:	23
请求示例:	23
返回示例:	23
发起轮播转码接口 (StartLoop)	24
接口描述	24
输入文件格式	24
请求方式:	24
请求参数:	24
返回参数:	24
请求示例:	25
返回示例:	25
停止轮播转码接口 (StopLoop)	26
接口描述	26
请求方式:	26
请求参数:	26
返回参数:	26
请求示例:	26
返回示例:	26
更新轮播任务接口 (UpdateLoop)	27
接口描述	27
请求方式:	27

请求参数:	27
返回参数:	27
请求示例:	27
返回示例:	28
查询轮播列表接口 (GetLoopList)	28
接口描述	28
请求方式:	28
请求参数:	28
返回参数:	28
请求示例:	29
返回示例:	29
发起拉转推接口 (RedirectStart)	30
接口描述	30
拉流协议	30
转推流协议	30
请求方式:	30
请求参数:	30
trans_params结构:	30
logos结构:	30
fops结构:	31
返回参数:	31
请求示例:	31
停止拉转推接口 (RedirectStop)	32
接口描述	32
第一种请求方式, 根据原始流和输出流停止任务:	32
请求参数:	33
第二种请求方式, 根据创建时返回的任务reqid停止任务:	33
请求参数:	33
返回参数:	33
请求示例:	33
返回示例:	33
查询拉转推接口 (GetTaskList)	34
接口描述	34
请求方式:	34
请求参数:	34
返回参数:	34
请求示例:	35
返回示例:	35

直播转码服务概述

功能介绍

金山云视频云对外提供直播转码服务，利用该服务客户可对金山云直播系统中的直播流进行多种实时转码处理，获取对应的转码流。具体实现的功能包括：视频分辨率、视频码率、视频帧率及音频相关参数的调整，另外还可实现对指定视频直播流的转推，外网拉取转推等。具体参数参考模板说明：<https://docs.ksyun.com/documents/2358>

使用方式

客户开通金山云直播服务时，可申请开通直播转码服务，利用服务提供的SDK或者API实现直播转码或转推等功能。使用时，客户需要首先使用其中的模板管理API预设自定义的转码通用参数，在启动直播推流或拉流时指定预先设定的模板名，则金山云直播系统会自动触发直播转码，客户按既定的规则使用转码流即可。

此外对于外网拉流轮播等功能客户也可自行调用对应的任务管理API实现指定功能。各API的说明详见API接口描述文档，对于API的使用我们也提供PHP/Java/Python语言版本的sdk供开发者便捷调用，详见各版本sdk使用说明文档。

根据使用场景不同，直播转码服务提供以下几种使用方式：

- 推流时触发转码
- 拉流时触发转码
- 外网拉流转推源站
- 直播流转推外网
- 多路直播流循环轮播
- 多文件生成节目单轮播

API使用方法

签名机制

直播转码openAPI的调用需要进行用户身份控制以防止非法访问。身份控制采用AWS签名算法进行签名认证，生成签名的方法详见[签名算法](#)

直播转码服务封装openAPI，对外提供三种语言（PHP/Python/JAVA）的SDK，主要设置转码模板，发起外网拉流、轮播任务。对于只使用直播转码的客户，请联系技术支持代为设置直播转码模板。

请求结构

客户调用金山云直播转码服务的openAPI接口是通过向指定服务地址发送请求，并按照openAPI文档说明在请求中添加相应的公共参数和接口参数来完成的。

直播转码openAPI的请求结构组成如下：

1. 服务地址

直播转码的服务线上正式接入地址为：`ket.cn-beijing-6.api.ksyun.com`

2. 通信协议

支持通过 HTTP 或 HTTPS 两种方式进行请求通信，推荐使用安全性更高的 HTTPS方式发送请求。

3. 请求方法

直播转码的openAPI支持GET或POST请求，具体查看接口说明。

注意

- 如果使用 GET 方式，参数均从 querystring 取得；如果使用 POST 方式，公共参数均从querystring取得，接口参数从请求Json Body中取得

4. 请求参数

金山云openAPI请求包含两类参数：**公共请求参数**和**接口请求参数**。其中，公共请求参数是每个接口都要用到的请求参数，具体可参见公共参数小节；接口请求参数是各个接口所特有的，具体见各个接口的**请求参数描述**。

5. 字符编码

请求及返回结果都使用UTF-8字符集进行编码。

SDK调用流程

1. SDK封装API请求，发往金山云IAM。
2. IAM对HTTP请求进行鉴权，鉴权失败则返回调用方；鉴权成功后，将HTTP请求中的业务相关信息下发给直播转码API服务端。IAM同时对客户的QPM (Requests Per Minute) 进行检查，超过QPM的请求会被直接拒绝，避免后端服务被少数客户的误操作打爆。
3. 直播转码API服务处理客户请求，将处理结果封装成JSON，返回给调用方。

返回结果

调用金山云的openAPI服务，调用成功，返回的HTTP状态码 (Status) 为200；调用失败，返回4xx 或5xx的HTTP状态码 (Status)。

金山云的媒体转码服务的调用返回的数据格式目前只支持Json格式，可通过设置HTTP Header `Accept=application/json`来指定返回数据格式。

调用成功

调用成功返回HTTP状态码200和Json格式的消息体，Json包含通用参数和特有参数，具体详见各接口。

调用失败

在转码API工作异常时，IAM会直接返回调用方。HTTP请求返回一个4xx或5xx的HTTP状态码，返回的HTTP消息体中包含具体的错误代码 (code) 及错误信息 (message)，详见[金山云身份与访问控制服务](#)。

请注意，请求ID (RequestId) 是用来唯一标识本次调用请求的，在调用方找不到错误原因时，可以联系金山云客服，并提供RequestId，以便我们尽快帮您解决问题。

Json格式示例

```
{
  "RequestId": "68093a99-2f63-4f39-8f70-3047ab8ecb5b",
  "Error": {
    "Type": "Sender",
    "Code": "InvalidParameterValue",
    "Message": "An invalid or out-of-range value was supplied for the input parameter PathPrefix."
  }
}
```

公共参数

参数说明

公共请求参数是每个身份与访问管理接口都需要使用到的请求参数。详见[KOP API公共参数](#)

示例

HTTPS GET方法请求示例：

```
GET /?Action=GetPresetList&Version=2017-01-01&UniqName=mytest&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20170517T131355Z&X-Amz-Credential=AKLT6UU6Vp_9SxWW5RAW6UHtdQ%2F20170517%2Fcn-beijing-6%2Fket%2Faws4_request&X-Amz-SignedHeaders=accept%3Bhost&X-Amz-Signature=1b55ca5217cf02ec38c9e19ecdcf2d5ebe9d37353b9da625461d514bfe79df57 HTTP/1.1
```

Host: ket.cn-beijing-6.api.ksyun.com

Accept: application/json

HTTPS POST方法请求示例:

```
POST /?Action=Preset&Version=2017-01-01&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20170517T132456Z&X-Amz-Credential=AKLT6UU6Vp_9SxWW5RAW6UHtdQ%2F20170517%2Fcn-beijing-6%2Fket%2Faws4_request&X-Amz-SignedHeaders=accept%3Bhost&X-Amz-Signature=a831782a11c587d529d779557e873e4a5fcc145788547a13cb4f9ad00d92e0d HTTP/1.1
```

Host: ket.cn-beijing-6.api.ksyun.com

Accept: application/json

Content-Type: application/json;charset=utf-8

Content-Length: %LEN

```
{
  %JSON Body
}
```

- 特别注意以上示例中的Action与Version。新版直播转码API的版本号：Version=2017-01-01，接口名Action值为各接口API。推荐使用新版SDK/API接入直播转码。
- GET/POST示例中，HTTP header中Host为必需参数，且需要参与签名计算。而querystring中X-Amz-SignedHeaders=accept;host指定了HTTP Header中Host/Accept均需要参与签名
- 老版直播转码API请联系商务支持。

拉流地址说明

直播转码为用户提供拉流触发转码和推流触发转码两种选择:

- **拉流触发转码（推荐）**：有观众拉转码流时启动转码，推流结束或无观众拉转码流时停止转码。
- **推流触发转码**：在直播推流开始时直接根据预先设置的模板进行转码，推流结束时停止转码。

1. 拉流触发转码

转码模板format参数组中的可选参数suffix，与模板名preset共同决定了拉流转码的转码流名，即：preset与suffix必须同时设置并且suffix值需和_preset保持一致。拉流转码模板一般对应一转一，即只需设置一组format参数，如果设置了多组format参数，则只有第一组format参数会生效。

推流地址：

```
rtmp://hostForStreamPush/live/{stream}
```

拉源流：

```
rtmp://hostForStreamPull/live/{stream}
```

拉转码流：

```
rtmp://hostForStreamPull/live/{stream}{suffix}
```

示例如下：

设置模板命名为mypreset，模板中设置了258一组转码output_format，流名为mystream，则

推流地址：

```
rtmp://hostForStreamPush/live/mystream
```

拉源流：

```
rtmp://hostForStreamPull/live/mystream
```

拉转码流，将suffix参数设为_mypreset:

```
rtmp://hostForStreamPull/live/mystream_mypreset
```

注: 如果preset值为mypreset, suffix值必须设置为_mypreset。即suffix值需和_preset值保持一致。

2. 推流触发转码

推流转码可设置多组不同的format参数, 每组设置不同的suffix用来区分不同的转码流。尽量保证每组format参数中的output_format值唯一(若设置了format_index参数, 则可设置多组相同output_format参数, 且output_index不能重复), suffix与output_format/format_index值共同决定了转码后的流名, 转码后流名命名规则如下:

- 若suffix已设置(推荐), 则使用suffix作后缀。
- 若suffix未设置(不推荐, 随着模板增加局限性增强), 使用下划线“_”加模板参数output_format参数(若设置了format_index, 则还需要在output_format后加上该两位数的format_index)作后缀。

成功设置转码模板之后, 把如下两种方式, 启动推流转码:

- 联系技术支持, 开启推流转码功能, 用户使用{stream}推流时, 自动触发转码。这种方式, 默认只能指定一个模板, 不够灵活, 不推荐使用。
- 用户自己设定一个流名{stream}进行推流, URL的querystring加上模板名称, 推流地址如下:

```
rtmp://hostForStreamPush/live/{stream}?preset={preset}
```

推流转码被触发后, 请使用如下方式拉流:

- 拉源流:

```
rtmp://hostForStreamPull/live/{stream}
```

- 拉转码流(未设置suffix参数, 未设置format_index参数):

```
rtmp://hostForStreamPull/live/{stream}_{output_format}
```

- 拉转码流(未设置suffix参数, 设置了format_index参数):

```
rtmp://hostForStreamPull/live/{stream}_{output_format}{format_index}
```

- 拉转码流(设置suffix参数为mysuffix):

```
rtmp://hostForStreamPull/live/{stream}{mysuffix}
```

示例如下:

设置模板命名为mypreset, 模板中设置了两组258(其中一个format_index=1)和一组259转码output_format, 其中258未设置suffix, 259设置suffix为my259, 且流名为mystream, 则

推流地址:

```
rtmp://hostForStreamPush/live/mystream?preset=mypreset
```

拉源流:

```
rtmp://hostForStreamPull/live/mystream
```

拉转码流:

```
rtmp://hostForStreamPull/live/mystream_258
```

```
rtmp://hostForStreamPull/live/mystream_25801
```

```
rtmp://hostForStreamPull/live/mystreammy259
```

直播转码错误码定义

直播转码API调用成功时会返回Json格式的信息，Json中均包含ErrNum/ErrMsg参数，用以描述处理结果，如下：

ErrNum	说明	ErrMsg
0	成功	success
1	未知错误	unknow issue
1000	参数缺失	lack of parameters
1001	参数不合法	parameter illegal
1002	无效的身份标识名称	lack of parameters
1003	无效的应用名称	invalid app name
1004	无效的流名称	invalid stream name
1005	无效的模板名称	invalid preset name
1006	无效的输出格式	invalid output format
1100	拒绝访问	access denied
1200	记录不存在	record is empty
1201	记录已经存在	record already exist
1202	插入记录错误	add record fail
1203	删除记录错误	drop record fail
1204	更新记录错误	update record fail
1300	任务失败	task fail
1301	任务不存在	task most exist

模板说明

转码模板是以Json格式预先设定一套转码参数，在拉流URL中携带转码参数，用来标识转码流，详情见[直播转码操作详解](#)

直播模板结构

名称	类型	必填项	示例值	描述
UniqName	string	是	"myuniqname"	客户域名标识，模板一级参数，由2~128位字母、数字、下划线组成。
App	string	是	"live"	客户频道名，模板一级参数，由2~32位字母、数字、下划线组成。
Preset	string	是	"test"	模板名称，模板一级参数，在用户直播转码模板命名空间唯一，由2~32位字母、数字、下划线组成。
Description	string	否	"demo"	模板描述，模板一级参数。
Output	json array	是	如下扩展	转码输出流媒体参数集，模板一级参数，用于设定一到多组参数集，每一组参数产生一路转码输出流。
Video	json object	否	如下扩展	可选参数：视频设置补充参数集，object，模板一级参数

Output结构

名称	类型	必填项	示例值	描述
format	json object	是	如下扩展	"format"转码输出流关键参数集。

format结构

名称	类型	必填项	示例值	描述
output_format	int	是	265	转码输出档标识，转码核心参数，指定转码输出媒体流的参数档次，见转码输出格式表。
width	int	否	1080	视频宽度，单位px，取值范围：0-3000
height	int	否	1920	视频高度，单位px，取值范围：0-3000
abr	int	否	64000	可选参数：自定义音频码率，int，取值范围20000~320000，单位换算 1k=1024byte 如：100k = 102400
vbr	int	否	800000	自定义视频码率，取值范围100000~10000000，单位换算 1k=1024byte 如：100k = 102400
fr	int	否	25	自定义帧率，取值范围1~30
gop	int	否	5	I帧间隔，单位：秒，默认5

suffix	String	是	"_mysuffix"	自定义拉流后缀名，string，由字母、数字、下划线组成，必须下划线开头。
remuxflag	int	否	0	音视频转码选项，默认0。0:音视频都转码；1:音频原样输出，视频转码；2:视频原样输出，音频转码；3:音视频都原样输出。
exstream	int	否	0	指定转出纯音频或纯视频：默认0。0:音视频都保留；1:只保留视频；2:只保留音频。
logo_swit ch	int	否	0	视频加水印LOGO开关选项，用来确定是否对视频添加水印，默认0。0:关闭；1:开启，开启时必须在Video->LOGO中配置LOGO图。

Output和format示例如下：

```
{
  "app": "demo",
  "preset": "for_RC-DSQ",
  "description": "live transcode for RC-DSQ",
  "output": [{
    "format": {
      "output_format": 265,
      "vbr": 100,
      "fr": 5
      ...
    }
  },
  {
    "format": {...}
  }
}]
```

Video结构

名称	类型	必填项	示例值	描述
LOGO	json array	是	如下扩展	水印图片参数集。

LOGO结构

名称	类型	必填项	示例值	描述
pic	String	是	"/bucketname/objectname"	水印图片所在的ks3中路径，必须是公共可访问，且与转码服务在同一region。
optimal_w	int	否	480	自定义水印图片宽度
optimal_h	int	否	640	自定义水印图片高度
offsetX	int	否	-10	水印图片横向偏移，基准位置为左上角顶点，向右为正，向左为负，默认10
offsetY	int	否	10	水印图片纵向偏移，基准位置为左上角顶点，向下为正，向上为负，默认10

Video和LOGO示例如下：

```
{
  "app": "demo",
  "preset": "for_RC-DSQ",
  "description": "live transcode for RC-DSQ",
  "output": [{
    "format": {...}
  }],
  "Video": {
    "LOGO": [{
      "pic": "/mybucketname/myobjectname",
      "optimal_w": 480,
      "optimal_h": 640,
      "offsetX": -10,
      "offsetY": 10
    }],
    ...
  }
}
```

转码输出格式表

为方便用户接入，直播转码使用一个output_format值作为一组转码默认参数的代号，即转码预设模板代号，是转码必选参

数。预设模板参见下表。

对一般使用场景，只设置output_format足矣，转码会首先对输入源流进入嗅探，根据结果自动决策一组折中的转码参数集，其过程远比下表中罗列的更为复杂。

如果发现预设模板方案不能满足自身业务需求，可以使用视频云提供的SDK/API，调整转码模板中各转码参数，比如帧率、码率、加水印等。如何使用模板与转码输出格式表，请参照[模板使用示例](#)。如果您在配置模板中有任何问题，请咨询技术支持。

output_format	container	vcodec	acodec	resolution	fps	vbitrate (bps)	samplerate (Hz)	channels	abitrage (bps)
256	flv	h264	aac	144P	10	150K	22050	2	24K
257	flv	h264	aac	240P	15	250K	22050	2	32K
258	flv	h264	aac	360P	18	500K	22050	2	48K
259	flv	h264	aac	480P	24	800K	44100	2	64K
260	flv	h264	aac	540P	25	900K	44100	2	64K
261	flv	h264	aac	720P	25	1100K	44100	2	96K
263	flv	h264	aac	960P	30	1700K	44100	2	128K
264	flv	h264	aac	1080P	30	2M	44100	2	128K
266	flv	h264	aac	1440P (2K)	30	3M	44100	2	128K
268	flv	h264	aac	2160P (4K)	30	6M	44100	2	128K
271	flv	h264	aac	COPY	self-adapt	self-adapt	44100	2	128K
274	flv	h265	aac	360P	18	400K	22050	2	48K
275	flv	h265	aac	480P	24	600K	44100	2	64K
276	flv	h265	aac	540P	25	650K	44100	2	64K
277	flv	h265	aac	720P	25	800K	44100	2	96K
279	flv	h265	aac	960P	30	1200K	44100	2	128K
280	flv	h265	aac	1080P	30	1400K	44100	2	128K
282	flv	h265	aac	1440P (2K)	30	2M	44100	2	128K
284	flv	h265	aac	2160P (4K)	30	4M	44100	2	128K
287	flv	h265	aac	COPY	self-adapt	self-adapt	44100	2	128K

模板使用示例

转码格式定义里面的360P、480P，是指输出分辨率最大是360P，480P，如果输入视频是320x240\15fps\300kb，用户设置了output_format=258(即输出最大360P)，转码的输出同样会是320x240。转码服务有一些总的规定如下：

1. 保持视频宽高比，转码过程不会对视频进行拉伸、加黑边。
2. 不允许小转大，包括转出分辨率不会比大于输入，转出视频或音频码率不会大于输入，输出视频帧率不会大于输入。
3. 音频转码不改变采样率。

需求：我想做视频监控墙，我不在意分辨率。

```
{
  "app": "demo",
  "preset": "for_RC-DSQ",
  "description": "live transcode for RC-DSQ",
  "output": [
    {
      "format": {
        "output_format": 256, // 转出144P
        "vbr": 100, // 单位: bps, 可选项, 没有会用默认值
        "abr": 20000, // 单位: bps, 可选项, 没有会用默认值
        "fr": 5 //可选项, 没有会用默认值
      }
    }
  ]
}
```

```
    }  
  }  
]
```

需求：音频转码会失真，希望只对视频转码，音频原样输出。

```
{  
  "app": "demo",  
  "preset": "copy_audio",  
  "description": "transcode vidoe, reserve audio",  
  "output": [  
    {  
      "format": {  
        // 转出360P  
        "output_format": 258,  
        // 1: 音频原样输出，视频转码  
        // 2: 视频原样输出，音频转码  
        // 3: 音视频都原样输出。（没有人会选这个的）  
        // 默认是0: 音视频都经过转码  
        "remuxflag": 1  
      }  
    }  
  ]  
}
```

需求：我想保留视频分辨率，但视频码率要转码成我设定的。

```
{  
  "app": "demo",  
  "preset": "for_RC-DSQ",  
  "description": "live transcode for RC-DSQ",  
  "output": [  
    {  
      "format": {  
        // 调用方不用告知分辨率大小，转码服务默认输出原始分辨率。  
        // 码率和帧率会根据用户设定或format默认值进行调整。  
        "output_format": 271,  
        // 可选项，但推荐用户设置  
        "vbr": 500000  
      }  
    }  
  ]  
}
```

创建模板接口(Preset)

接口描述

预设直播转码模板参数集

请求方式:

POST

请求参数:

参数名	类型	是否必需	备注
UniqName	string	是	客户的域名标识
App	string	是	客户的频道名
Preset	string	是	自定义模板名
Description	string	否	模板描述
Output	json array	是	转码输出参数集
Video	json object	否	转码输出视频补充参数集

返回参数:

参数名	类型	是否必需	备注
ErrNum	int	是	错误码, 0: 成功; 非0: 失败
ErrMsg	string	是	描述本次请求错误的原因

请求示例:

```
POST /?Action=Preset
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170517T134823Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx
HTTP/1.1
Host:ket.cn-beijing-6.api.ksyun.com
Accept:application/json
{
  "UniqName": "xxxxx",
  "App": "xxxxx",
  "Preset": "live_for_demo",
  "Description": "demo",
  "Output": [
    {
      "format": {
        "output_format": 256,
        "vbr": 800000,
```

```
        "logo_switch": 1
      },
      "switch": {
        "sn": 1,
        "sm": 1
      }
    },
    {
      "format": {
        "output_format": 256,
        "format_index": 1
      }
    }
  ],
  "Video": {
    "orientationAdapt": 0,
    "LOGO": [
      {
        "pic": "/test/mylogo.jpg",
        "optimal_w": 480,
        "offsetX": -10,
        "offsetY": 10
      }
    ]
  }
}
```

返回示例：

```
HTTP/1.1 200 OK
Server: ksyun-open-platform
Date: Fri, 12 May 2017 08:01:06 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
{
  "ErrNum":0,
  "ErrMsg":"success"
}
```

更新模板接口 (UpdatePreset)

接口描述

修改已设定的直播转码模板参数

请求方式:

POST

请求参数:

参数名	类型	是否必需	备注
UniqName	string	是	客户的域名标识
App	string	是	客户的频道名
Preset	string	是	模板名
Description	string	否	模板描述
Output	json array	是	转码输出参数集
Video	json object	否	转码输出视频补充参数集

返回参数:

参数名	类型	是否必需	备注
ErrNum	int	是	错误码, 0: 成功; 非0: 失败
ErrMsg	string	是	描述本次请求错误的原因

请求示例:

```
POST /?Action=UpdatePreset
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170518T070151Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxxx HTTP/1.1
Host: ket.cn-beijing-6.api.ksyun.com
Accept: application/json
{
  "UniqName": "xxxxx",
  "App": "xxxxx",
  "Preset": "live_for_demo",
  "Description": "demo",
  "Output": [
    {
      "format": {
        "output_format": 256,
        "suffix": "_mysuffix"
      }
    },
    {
      "format": {
        "output_format": 257
      }
    }
  ]
}
```

```
]
}
```

返回示例:

```
HTTP/1.1 200 OK
Server: ksyun-open-platform
Date: Fri, 12 May 2017 08:01:06 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
{
  "ErrNum":0,
  "ErrMsg":"success"
}
```

删除模板接口 (DelPreset)

接口描述

删除已设定的直播转码参数模板

请求方式:

GET

请求参数:

参数名	类型	是否必需	备注
UniqName	string	是	客户的域名标识
App	string	是	客户的频道名
Preset	string	是	模板名

返回参数:

参数名	类型	是否必需	备注
ErrNum	int	是	错误码, 0: 成功; 非0: 失败
ErrMsg	string	是	描述本次请求错误的原因

请求示例:

```
GET /?Action=DelPreset
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170517T140655Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx
&App=xxxxx
&Preset=live_for_demo
```

```
&UniqName=xxxxx HTTP/1.1
Host: ket.cn-beijing-6.api.ksyun.com
Accept: application/json
```

返回示例:

```
HTTP/1.1 200 OK
Server: ksyun-open-platform
Date: Fri, 12 May 2017 08:01:06 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked

{
  "ErrNum":0,
  "ErrMsg":"success"
}
```

查询模板列表接口 (GetPresetList)

接口描述

查询预设模板列表

请求方式:

GET

请求参数:

参数名	类型	是否必需	备注
UniqName	string	是	客户的域名标识
App	string	是	客户的频道名

返回参数:

参数名	类型	是否必需	备注
ErrNum	int	是	错误码, 0: 成功; 非0: 失败
ErrMsg	string	是	描述本次请求错误的原因
PresetList	json array	否	直播模板列表信息, ErrNum为0时不为空
Preset	string	否	模板名称
App	string	否	模板频道
FmtNum	int	否	单个模板设置的output_format个数
CreateTime	string	否	模板创建时间
UpdateTime	string	否	模板更新时间

请求示例:

```
GET /?Action=GetPresetList
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170518T070151Z
&X-Amz-Credential=xxxxx
```



```
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx
&App=xxxxx
&UniqName=xxxxx HTTP/1.1
Host: ket.cn-beijing-6.api.ksyun.com
Accept: application/json
```

返回示例：

```
HTTP/1.1 200 OK
Server: ksyun-open-platform
Date: Fri, 12 May 2017 08:01:06 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
{
  "PresetList": [
    {
      "Preset": "555",
      "App": "xxxxx",
      "FmtNum": "3",
      "Description": "descsss:555",
      "CreateTime": "2017-05-06 12:09:08",
      "UpdateTime": "2017-05-06 16:19:54"
    },
    {
      "Preset": "666",
      "App": "xxxxx",
      "FmtNum": "2",
      "Description": "testpreset",
      "CreateTime": "2017-05-06 16:07:21",
      "UpdateTime": ""
    },
    {
      "Preset": "testpreset1",
      "App": "xxxxx",
      "FmtNum": "2",
      "Description": "ttttt",
      "CreateTime": "2017-05-06 16:07:56",
      "UpdateTime": "2017-05-06 16:24:50"
    }
  ],
  "ErrNum": 0,
```

```
"ErrMsg": "success"  
}
```

查询模板详情接口 (GetPresetDetail)

接口描述

查询指定预设参数模板详情

请求方式:

GET

请求参数:

参数名	类型	是否必需	备注
UniqName	string	是	客户的域名标识
App	string	是	客户的频道名
Preset	string	是	模板名称

返回参数:

参数名	类型	是否必需	备注
ErrNum	int	是	错误码, 0: 成功; 非0: 失败
ErrMsg	string	是	描述本次请求错误的原因
PresetDetail	json object	否	直播模板详情信息, ErrNum为0时不为空, 二级参数见模板格式定义

请求示例:

```
GET /?Action=GetPresetDetail  
&Version=2017-01-01  
&X-Amz-Algorithm=AWS4-HMAC-SHA256  
&X-Amz-Date=20170518T064235Z  
&X-Amz-Credential=xxxxx  
&X-Amz-SignedHeaders=xxxxx  
&X-Amz-Signature=xxxxx  
&App=xxxxx  
&Preset=live_for_demo  
&UniqName=xxxxx HTTP/1.1  
Host: ket.cn-beijing-6.api.ksyun.com  
Accept: application/json
```

返回示例:

```
HTTP/1.1 200 OK  
Server: ksyun-open-platform  
Date: Fri, 12 May 2017 08:01:06 GMT  
Content-Type: application/json;charset=UTF-8  
Transfer-Encoding: chunked  
{
```

```
"PresetDetail":
{
  "Description": "demo",
  "Output":
  [
    {
      "format":{
        "output_format":256,
        "suffix": "_mysuffix"
      }
    },
    {
      "format":{
        "output_format":257,
      }
    }
  ],
  "Video": {},
  "Audio": {},
  "CreateTime": "2016-08-15 13:41:46",
  "IntelligentSwitch": 0,
  "IntelligentCurl": ""
}
"ErrNum": 0,
"ErrMsg": "success"
}
```

查询任务列表接口 (GetStreamTranList)

接口描述

流维度查询转码任务信息。

请求方式:

GET

请求参数:

参数名	类型	是否必需	备注
UniqName	string	是	客户的域名标识
App	string	是	客户的频道名
StreamID	string	是	流名称

返回参数:

参数名	类型	是否必须	备注
-----	----	------	----

ErrNum	int	是	错误码，0：成功；非0：失败
ErrMsg	string	是	描述本次请求错误的原因
TranList	json array	否	转码任务列表信息，ErrNum为0时不为空
ListCount	int	否	转码任务列表数
TaskID	string	否	转码任务ID
OutputFormat	int	否	对应模板格式定义中的output_format
TranStatus	int	否	转码任务状态
StreamName	string	否	直播源流名
CreateTime	string	否	任务创建时间

请求示例：

```
GET /?Action=GetStreamTranList
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170518T074413Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx
&UniqName=xxxxx
&StreamID=streamtest
&App=xxxxx HTTP/1.1
Host: ket.cn-beijing-6.api.ksyun.com
Accept: application/json
```

返回示例：

```
HTTP/1.1 200 OK
Server: ksyun-open-platform
Date: Fri, 12 May 2017 08:01:06 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
{
  "TranList": [
    {
      "Preset": "demo",
      "OutputFormat": "256",
      "TaskID": "7e8c97e0767050753612f70d245c6027",
      "TranStatus": 30,
      "StreamName": "streamtest",
      "CreateTime": "2017-05-18 12:20:01"
    }
  ],
  "ListCount": 1,
  "ErrNum": 0,
  "ErrMsg": "success"
```

}

发起外网拉流接口 (StartStreamPull)

功能描述:

发起从外网拉流推流到源站，模拟用户向源站推流。支持rtmp/http-flv/hls协议的外网拉流。需要提前开通金山云直播源站服务。

请求方式:

POST

请求参数:

参数名	类型	是否必需	备注
UniqName	string	是	客户的域名标识
App	string	是	客户的频道名
StreamID	string	是	流名称
SrcUrl	string	是	外网源流地址
Vdoid	string	否	用于标识同一流名的每次推流，开启直播转点播时，该值将作为点播文件名的一部分（见直播转点播部分说明）
Params	string	否	推流到源站使用的额外参数，形如url querystring，如：带转码模板时设置为preset=my preset

ExtParam 参数说明:

参数名	类型	是否必需	备注
preset	string	否	转码模版，需要对源流进行转码时设置，默认为空
interval	int	否	拉源流失败时，重试间隔时间，单位：秒，范围：[1, 20]，默认6秒。如果此次拉流时间已经花费n秒，则只sleep(interval - n)。最长重试时间不超过5分钟。
retrytimes	int	否	拉源流失败时，重试最大次数，范围：[1, 100]。最长重试时间不超过5分钟。

返回参数:

参数名	类型	是否必须	备注
ErrNum	int	是	
ErrMsg	string	是	
TaskID	string	否	任务ID

请求示例:

示例一:

```
POST /?Action=StartStreamPull
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170518T092721Z
&X-Amz-Credential=AKLT6UU6Vp_9SxWW5RAW6UHtdQ%2F20170518%2Fcn-beijing-6%2Fket%2Faws4_request
&X-Amz-SignedHeaders=accept%3Bhost
&X-Amz-Signature=a439d0f4a6275da64cb04ead02946a1ce6eede11c552e24504e116a4c5558aee HTTP/1.1
Host: ket.cn-beijing-6.api.ksyun.com
Accept: application/json
{
```

```
"UniqName": "mytest",
"StreamID": "teststreamid",
"SrcUrl": "rtmp://qa-ws.test-rtmplive.ks-cdn.com/live/20160819",
"App": "live",
"Vdoid": "123456",
"Params": "preset=testpreset",
}
```

示例二:

```
POST /?Action=StartStreamPull
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170518T092721Z
&X-Amz-Credential=AKLT6UU6Vp_9SxWW5RAW6UHtdQ%2F20170518%2Fcn-beijing-6%2Fket%2Faws4_request
&X-Amz-SignedHeaders=accept%3Bhost
&X-Amz-Signature=a439d0f4a6275da64cb04ead02946a1ce6eede11c552e24504e116a4c5558aee HTTP/1.1
Host: ket.cn-beijing-6.api.ksyun.com
Accept: application/json
{
  "UniqName": "mytest",
  "StreamID": "teststreamid",
  "SrcUrl": "rtmp://qa-ws.test-rtmplive.ks-cdn.com/live/20160819",
  "App": "live",
  "DstUrl": "rtmp://qa-ks.uplive.ks-cdn.com/live/20160819",
  "ExtParam": '{"timeout":5,"retrytimes":3}'
}
```

返回示例:

```
HTTP/1.1 200 OK
Server: ksyun-open-platform
Date: Fri, 12 May 2017 08:01:06 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
{
  "ErrNum":0,
  "ErrMsg":"success",
  "TaskID":"8291322e08e34d8bfd796e1ae9807dee"
}
```

停止外网拉流接口(StopStreamPull)

功能描述:

停止从外网拉流推流到源站的任务。

注：当外网直播源发生断流，超过60s后，拉流任务进行重试，每5s重试一次。正常情况下，客户不需要主动停止外网拉流，只要输入源停流，超过20分钟，后端服务会自动删除该拉流任务；特别的，用户想马上主动断流，可以使用该接口。

请求方式：

POST

请求参数：

参数名	类型	是否必需	备注
UniqName	string	是	客户的域名标识
App	string	是	客户的频道名
StreamID	string	是	流名称

返回参数：

参数名	类型	是否必须	备注
ErrNum	int	是	错误码，0：成功；非0：失败
ErrMsg	string	是	描述本次请求错误的原因

请求示例：

```
POST /?Action=StopStreamPull
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170518T094352Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx HTTP/1.1
Host: ket.cn-beijing-6.api.ksyun.com
Accept: application/json
{
  "UniqName": "xxxxx",
  "StreamID": "streamtest",
  "App": "xxxxx"
}
```

返回示例：

```
HTTP/1.1 200 OK
Server: ksyun-open-platform
Date: Fri, 12 May 2017 08:01:06 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
{
  "ErrNum": 0,
  "ErrMsg": "success"
```

}

发起轮播转码接口 (StartLoop)

接口描述

发起轮播任务，用于将指定的点播视频文件转换为源站的直播视频流；

适用于将客户在直播间内循环播放热门视频。

此接口需开通直播转码才能应用，请联系技术支持。

输入文件格式

mp4/flv/m3u8/mpegs

注：如果使用多文件轮播最好先用离线转码将所有视频参数统一，并转成mp4视频再发起轮播。支持的文件名字符：中文、英文大小写、-、\$、&、=、'、~、[]、{}、;、!、¥、……（——）；‘’“”，。《》？。限制说明：目前只支持音视频类文件轮播，不支持纯视频、纯音频类混合轮播类。

请求方式：

POST

请求参数：

参数名	类型	是否必需	描述
UniqName	string	是	客户的域名标识
App	string	是	客户的频道名
Preset	string	是	模板名称，支持配置：视频帧率、视频码率、音频码率、视频分辨率
StreamID	string	是	轮播流名
SrcInfo	json array	是	轮播源文件信息，数组，每项包括文件路径（金山云KS3内网URL）和顺序参数。发起轮播前请先确认轮播源文件是否存在。最多可以支持100个轮播任务。
PubDomain	string	是	推流域名
TaskStartTime	int	否	任务开始时间戳，指定轮播流启动播放时间，精确到秒，不填默认下发后立即开始轮播。只能填当前时间1分钟之后的时间。
TaskStopTime	int	否	任务结束时间戳，指定轮播流结束播放时间，精确到秒，不填默认结束时间为开始时间之后一年。TaskStopTime-TaskStartTime必须>30s。
LoopTimes	int	否	文件轮播次数，与TaskStopTime冲突时，以TaskStopTime为准，如果需要以轮播次数为准，TaskStopTime不填
SrcInfo	类型	是否必需	描述
Path	string	是	源文件在ks3的完整路
Index	int	是	文件播放顺序

备注：

- SrcInfo必须是金山云KS3内网URL，组合内网域名与视频文件的objectkey。使用外网URL会造成轮播失败。推荐使用region在北京的bucket做为SrcInfo的来源。在[KS3空间概况](#)可以显示该bucket的内网域名。
- 推荐bucket的权限设置为公开读，设置方式参见[KS3存储空间维护-权限设置](#)。如果bucket只能设置私密，则SrcInfo需要客户按[KS3接口文档-请求签名](#)的要求预先完成签名，再赋予SrcInfo。请注意，采用预先签名时，需要保证签名过期时间足够长，因为签名过期，会造成轮播失败。
- 新建轮播任务时，只能指定最大20天的轮播时长，可以调用[更新轮播时长接口](#)，每次更新时长也是最长20天。更新轮播时长接口可以多次调用。

返回参数：

参数名	类型	是否必需	描述
ErrNum	int	是	错误码
ErrMsg	string	是	错误信息
List	json array	否	轮播转码任务信息

TaskID	string	是	轮播转码单个任务ID
Format	int	是	转码任务使用的output_format

请求示例：

```
POST /?Action=StartLoop
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170518T090434Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxxx HTTP/1.1
Host: ket.cn-beijing-6.api.ksyun.com
Accept: application/json
{
  "PubDomain": "xxx.xxx.com",
  "UniqName": "xxxxx",
  "Preset": "testpreset",
  "SrcInfo": [
    {
      "Path": "http://xxx.xxx.com/ksyun.flv",
      "Index": 0
    }
  ],
  "StreamID": "myteststreamid",
  "App": "xxx",
}
```

返回示例：

```
HTTP/1.1 200 OK
Server: ksyun-open-platform
Date: Fri, 12 May 2017 08:01:06 GMT
Content-Type: application/json; charset=UTF-8
Transfer-Encoding: chunked
{
  "List": [
    {
      "TaskID": "5cf5b255d7cf7d8a16cdafe46ee38cca",
      "Format": 256
    },
    {
      "TaskID": "87a24f7fb1758f77b93756dce42082f0",
      "Format": 257
    }
  ]
}
```

```
    }  
  ],  
  "ErrNum": 0,  
  "ErrMsg": "success"  
}
```

停止轮播转码接口 (StopLoop)

接口描述

停止轮播任务。

请求方式：

POST

请求参数：

参数名	类型	是否必需	描述
UniqName	string	是	客户的域名标识
App	string	是	客户的频道名
StreamID	string	是	轮播流名

返回参数：

参数名	类型	是否必需	描述
ErrNum	int	是	错误码，0：成功；非0：失败
ErrMsg	string	是	描述本次请求错误的原因

请求示例：

```
POST /?Action=StopLoop  
&Version=2017-01-01  
&X-Amz-Algorithm=AWS4-HMAC-SHA256  
&X-Amz-Date=20170518T093750Z  
&X-Amz-Credential=xxxxx  
&X-Amz-SignedHeaders=xxxxx  
&X-Amz-Signature=xxxxx HTTP/1.1  
Host: ket.cn-beijing-6.api.ksyun.com  
Accept: application/json  
{  
  "App": "xxxxx",  
  "UniqName": "xxxxx",  
  "StreamID": "myteststreamid"  
}
```

返回示例：

```
HTTP/1.1 200 OK
```

```
Server: ksyun-open-platform
Date: Fri, 12 May 2017 08:01:06 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
{
  "ErrNum": 0,
  "ErrMsg": "success"
}
```

更新轮播任务接口 (UpdateLoop)

接口描述

更新轮播源文件信息及文件播放顺序，仅支持对进行中的轮播任务进行更新，未开始的任務請直接停止任務后重新下发。

请求方式：

POST

请求参数：

参数名	类型	是否必需	描述
UniqName	string	是	客户的域名标识
App	string	是	客户的频道名
StreamID	string	是	轮播流名
SrcInfo	json array	否	轮播源文件信息，数组，每项包括文件路径（金山云KS3内网URL）和顺序参数。

备注：DurationHour与SrcInfo参数至少选择一项。

返回参数：

参数名	类型	是否必需	描述
ErrNum	int	是	错误码，0：成功；非0：失败
ErrMsg	string	是	描述本次请求错误的原因

请求示例：

```
POST /?Action=UpdateLoop
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170518T095006Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx HTTP/1.1
Host: ket.cn-beijing-6.api.ksyun.com
Accept: application/json
{
  "App": "xxxxx",
  "UniqName": "xxxxx",
```

```

"StreamID": "myteststreamid",
"DurationHour":100,
"SrcInfo": [
  {
    "Path": "http://wangxxx.ks3-cn-beijing.ksyun.com/xxxx.flv",
    "Index": 0
  },
  ...
]
}

```

返回示例:

```

HTTP/1.1 200 OK
Server: ksyun-open-platform
Date: Fri, 12 May 2017 08:01:06 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
{
  "ErrNum": 0,
  "ErrMsg": "success"
}

```

查询轮播列表接口 (GetLoopList)

接口描述

查询轮播列表。

此接口需开通才能应用，请联系技术支持。

请求方式:

GET

请求参数:

参数名	类型	是否必需	描述
UniqName	string	是	客户的域名标识
App	string	是	客户的频道名
StreamID	string	否	轮播流名

返回参数:

参数名	类型	是否必需	描述
ErrNum	int	是	错误码，0：成功；非0：失败
ErrMsg	string	是	描述本次请求错误的原因
Total	int	是	轮播任务数
List	json array	是	轮播任务列表
OutputFormat	int	是	转码output_format
SrcInfo	json array	是	轮播源地址列表

App	string	是	轮播频道名
PubDomain	string	是	轮播发布目标域名
Preset	string	是	模板名
ContinueHour	int	是	已播放时长，单位小时
TaskID	string	是	任务ID
StreamID	string	是	轮播流名
CreateTime	string	是	任务创建时间

请求示例：

```
GET /?Action=GetLoopList
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170518T083314Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx
&UniqName=xxxxx
&StreamID=streamtest
&App=xxxxx HTTP/1.1
Host: ket.cn-beijing-6.api.ksyun.com
Accept: application/json
```

返回示例：

```
HTTP/1.1 200 OK
Server: ksyun-open-platform
Date: Fri, 12 May 2017 08:01:06 GMT
Content-Type: application/json; charset=UTF-8
Transfer-Encoding: chunked
{
  "Total": 1,
  "List": [
    {
      "OutputFormat": "257",
      "SrcInfo": [
        "http://xxx.xxx.com/ksyun.flv"
      ],
      "App": "xxxxx",
      "PubDomain": "xxx.xxx.com",
      "Preset": "testpreset00",
      "ContinueHour": "100",
      "TaskID": "e8baa00b12a7b118451f54c3b71be2b1",
      "StreamID": "myteststreamid",
      "CreateTime": "2017-05-05 15:15:00"
    }
  ]
}
```

```

    }
  ],
  "ErrNum": 0,
  "ErrMsg": "success"
}

```

发起拉转推接口 (RedirectStart)

接口描述

发起拉转推任务，用于将指定的直播流转推为源站的直播视频流，并支持该直播流转码；

此接口需开通直播转码才能应用，该服务分配资源与直播转码隔离，需要单独配置，请联系技术支持。

拉流协议

直播流: rtmp/flv/hls

转推流协议

直播流: rtmp

请求方式:

POST

请求参数:

参数名	类型	是否必需	描述
uniqname	string	是	客户的域名标识
app	string	是	客户的频道名
src	array	是	源流地址，支持多路。直播源流支持rtmp/hls/flv协议，最多支持20路。
durl	array	是	推流地址，长度必须和src相同，说明：src[0]推流到 → durl[0]，src[1]推流到 → durl[1]，如长度不一致或非地址类型：1030:invalid durl。
trans_params	json array	否	转码参数，详见下。说明：src[0]推流到 → durl[0]，使用trans_params[0]的转码参数。
task_start_time	int64	否	任务开始时间，定时任务用到。如果时间戳小于当前时间或是0或是空或者不填，代表创建拉转推任务即开始。例：想在今晚8:00启动一个任务，此参数为今晚的八点的时间戳：1594692000。仅支持提前10分钟创建。不支持临近时间点了创建。
task_stop_time	int64	否	任务自动停止时间，如果未手动调用停止拉转推接口，则任务会一直到这个时间结束。如果时间戳小于当前时间或是0或是空或者不填，代表任务会一直轮询下去，可以通过stop接口停止。如果设置该时间要大于task_start_time 时间的10分钟。

- 如果拉转推任务启动时失败，任务会每隔5s重试一次，重试次数无限制。

trans_params结构:

参数名	类型	是否必需	描述
logos	array	否	水印信息，详见下。支持多LOGO
fops	json	否	视频转码参数，详见下
trans	int	否	是否转码，0: 否，1: 是，不传默认0

logos结构:

参数名	类型	是否必需	描述
wm_image	string	否	可访问的图片网址，支持http，https格式地址。若带了logos字段，则该字段是必填
wm_image_back	string	否	可访问的图片网址，备用的logo地址。若上面wm_image图片下载失败，会下载该地址。支持http，https格式地址。

wm_gravity	string	否	logo位置。“TOP_LEFT”, // 左上角; “TOP_CENTER”, // 上部居中; “TOP_RIGHT”, // 右上角; “CENTER_LEFT”, // 中部靠左; “CENTER”, // 居中; “CENTER_RIGHT”, // 中部靠右; “BOTTOM_LEFT”, // 左下角; “BOTTOM_CENTER”, // 下部居中; “BOTTOM_RIGHT”, // 右下角。若带了logos字段, 则该字段是必
wm_width	int	否	logo的宽度
wm_height	int	否	logo的高度
wm_dx	int	否	表示logo的坐标或相对输入视频的水平偏移量, 值有两种形式: 整数型代表偏移像素, 范围[-4096, 4096], 单位px; 小数型范围(-1, 1) 小数时可添加 h 或 w 指定比率的参考是视频的宽或高, 如 0.26h 表示视频高的0.26, 如果只写为0.26, 那么就相当于0.26w
wm_dy	int	否	表示logo的坐标或相对输入视频的垂直偏移量, 值有两种形式: 整数型代表偏移像素, 范围[-4096, 4096], 单位px; 小数型范围(-1, 1) 小数时可添加 h 或 w 指定比率的参考是视频的宽或高, 如 0.26h 表示视频高的0.26, 如果只写为0.26, 那么就相当于0.26h

fops结构:

参数名	类型	是否必需	描述
bitrate	string	否	视频码率。比如“2000k”、“2M”
res	string	否	分辨率, 宽x高。比如: “1080×720”
fps	string	否	帧率, 使用范围1-60
vcodec	string	否	编码格式。可选: 264/265/copy/COPY/H264/H265
acodec	string	否	音频格式, 可选: aac/copy

返回参数:

参数名	类型	是否必需	描述
errno	int	是	错误码
errmsg	string	是	错误信息
uniqname	string	是	客户的域名标识
task_type	string	是	“redirect”
reqlist	array	是	返回具体任务id与状态
reqid	string	是	任务id
errno	array	是	任务状态

请求示例:

```
POST /?Action=RedirectStart
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170518T090434Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxxx HTTP/1.1
Host: ket.cn-beijing-6.api.ksyun.com
Accept: application/json
{
  "uniqname": "XXX",
  "app": "XXX",
  "src": [
    "rtmp://XXX/live/src_test00",
    "rtmp://XXX/live/src_test01"
  ],
  "durl": [
    "rtmp://XXX/live/lsw_test",
    "rtmp://XXX/live/lsw_test2"
  ],
  "trans_params": [
    {
      "logos": [
        {
```

```

        "wm_image": "https://ks3-cn-beijing.ksyun.com/XXX/XX.png",
        "wm_gravity": "TOP_LEFT",
        "wm_width": 200,
        "wm_height": 100,
        "wm_dx": 0,
        "wm_dy": 0
    }],
    "fops": {
        "bitrate": "200k",
        "res": "1080x720",
        "fps": "30",
        "vcodec": "h264",
        "acodec": "aac"
    },
    "ext_param": "",
    "trans": 1
},
{
    "logos": [
        {
            "wm_image": "https://ks3-cn-beijing.ksyun.com/XXX/XX.png",
            "wm_gravity": "TOP_LEFT",
            "wm_width": 200,
            "wm_height": 100,
            "wm_dx": 0,
            "wm_dy": 0
        }],
        "fops": {
            "bitrate": "200k",
            "res": "1080x720",
            "fps": "30",
            "vcodec": "h264",
            "acodec": "aac"
        },
        "ext_param": "",
        "trans": 1
    }],
    "type": 0,
    "task_start_time": 0,
    "task_stop_time": 0,
    "start_index": 2
}

```

接口返回值:

```

{
    "errno": 0,
    "errmsg": "success",
    "uniqname": "XX",
    "task_type": "redirect",
    "reqlist": [ // 返回成功任务的id
        {
            "reqid": "db3b17fa7e5c1f4ea269634cd30de254", // 第一个任务id
            "errno": 0
        }
        ...
    ]
}

```

停止拉转推接口（RedirectStop）

接口描述

停止拉转推任务。

此接口需开通直播转码才能应用，请联系技术支持。

第一种请求方式，根据原始流和输出流停止任务：

POST

请求参数：

参数名	类型	是否必需	描述
uniqname	string	是	客户的域名标识
app	string	是	客户的频道名
src	string	是	停止任务的输入流
durl	string	是	停止任务的输出流，与src对应关系，和创建时保持一致

第二种请求方式，根据创建时返回的任务reqid停止任务：

POST

请求参数：

参数名	类型	是否必需	描述
reqid	string	是	客户创建时reqid

返回参数：

参数名	类型	是否必需	描述
errno	int	是	错误码，0：成功；非0：失败
errmsg	string	是	描述本次请求错误的原因

请求示例：

```
POST /?Action=RedirectStop
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170518T093750Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx HTTP/1.1
Host: ket.cn-beijing-6.api.ksyun.com
Accept: application/json
```

第一种请求方式：

```
{
  "uniqname": "XX",
  "app": "XX",
  "src": "rtmp://XX/live/src1",
  "durl": "rtmp://XX/live/test1"
}
```

第二种请求方式：

```
{
  reqid: "d5c44c969a889d0d240c91e97a523c76"
}
```

返回示例：

HTTP/1.1 200 OK

Server: ksyun-open-platform

Date: Fri, 12 May 2017 08:01:06 GMT

Content-Type: application/json;charset=UTF-8

Transfer-Encoding: chunked

```
{
  "errno": 0,
  "errmsg": "success"
}
```

查询拉转推接口 (GetTaskList)

接口描述

查询拉转推接口。

请求方式:

GET

请求参数:

参数名	类型	是否必需	备注
UniqName	string	是	客户的域名标识
App	string	否	客户的频道名
TaskType	string	是	6代表转码转推，7直播轮播，8点播文件轮播
StartDate	string	是	查询任务创建开始时间
EndDate	string	是	查询任务创建结束时间
Marker	int	否	开始标志位，默认0，即表示从查询范围内最早第一个任务开始
Limit	int	否	每次查询返回的最大数，默认50，最大支持50

返回参数:

参数名	类型	是否必须	备注
ErrNum	string	是	错误码，0：成功；非0：失败
ErrMsg	string	是	描述本次请求错误的原因
StartDate	string	是	查询任务创建开始时间
EndDate	string	是	查询任务创建结束时间
Marker	int	是	开始标志位
Total	int	是	整体任务数
Count	int	是	本次返回任务总数
TaskInfo	array	是	任务详情
TaskID	string	是	任务id
UniqName	string	是	客户的域名标识
App	string	是	客户的频道名
Preset	string	是	无意义，等待扩展参数
Format	string	是	无意义，等待扩展参数
StreamID	string	是	无意义，等待扩展参数
SrcAddr	string	是	拉流地址
DstAddr	string	是	输出流地址
MsgType	string	是	任务处理区域
QuotaType	string	是	任务类型，6代表转码转推，7直播轮播，8点播文件轮播
UseQuota	string	是	使用的配额数
TranStatus	string	是	0已经成功结束、2异常结束、1正在执行
task_error_msg	string	是	任务错误信息
CreateTime	string	是	任务创建时间

请求示例:

```
GET /?Action=GetTaskList
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170518T074413Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx
&UniqName=xxxxx
&TaskType=6
&StartDate=2019-10-30
&EndDate=2019-11-30
&Marker=0
&Limit=50
&App=xxxxx HTTP/1.1
Host: ket.cn-beijing-6.api.ksyun.com
Accept: application/json
```

返回示例:

```
HTTP/1.1 200 OK
Server: ksyun-open-platform
Date: Fri, 12 May 2017 08:01:06 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
{
  "ErrNum": 0,
  "ErrMsg": "success",
  "StartDate": "2020-08-01",
  "EndDate": "2020-09-17",
  "Marker": 0,
  "Total": 2,
  "TaskInfo": [
    {
      "TaskID": "4f6fc15c2cff822827526de79b0a984b",
      "UniqName": "videoqa",
      "App": "live_tran",
      "Preset": "presetredirect",
      "Format": "18447",
      "StreamID": "1ed3bea16e66a4fec1a397c4cd23e395",
      "SrcAddr": "http://XXX/live/test1.flv?vhost=videoqa.live.ks-cdn.com",
      "DstAddr": "rtmp://XXX/qa-ws.test-uplive.ks-cdn.com/katest/test1",
      "MsgType": "beijing",
      "QuotaType": "6",
      "UseQuota": "1.0",
      "TranStatus": "0",
      "task_error_msg": "success",
      "CreateTime": "2020-09-16 16:42:13"
    },
    {
      "TaskID": "d960f07083898c0a28d7e6e5f6ef3461",
      "UniqName": "videoqa",
      "App": "live_tran",
      "Preset": "presetredirect",
      "Format": "18447",
      "StreamID": "85df9e85fafdabd26e13ef841c48bac2",
      "SrcAddr": "http://XXX/live/test1.flv?vhost=XXX",
      "DstAddr": "rtmp://XXX/katest/test1",
    }
  ]
}
```

```
    "MsgType": "beijing",
    "QuotaType": "6",
    "UseQuota": "1.0",
    "TranStatus": "0",
    "task_error_msg": "success",
    "CreateTime": "2020-09-16 16:44:53"
  }
],
"Count": 2
}
```