

## 目录

目录	1
点播转码概述	4
功能介绍	4
使用场景	4
使用方式	4
API调用方式	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
示例	5
公共错误	5
转码结果回调用户	8
POST 回调接口	8
GET 回调接口	15
模板说明	15
模板类型说明	15
公有参数说明	15
公有参数格式定义:	15
四类模板Param详细说明	15
1. avtrans模板Param格式定义:	15
转码音频参数详情 (AUDIO)	16
转码视频参数详情 (VIDEO)	16
转码水印参数详情 (LOGOS)	16
2. avsample模板Param格式定义:	17
3. avsnapshot模板Param格式定义:	17
4. avinfo模板Param格式定义:	17
模板 (preset) 使用示例	18
创建模板接口 (Preset)	20
功能描述:	20
请求方式:	20
请求参数:	21
返回参数 :	21
请求示例:	21
返回示例:	21
更新模板接口 (UpdatePreset)	22
接口描述	22
请求方式	22
请求参数	22
返回参数:	22
请求示例	22
返回示例	22
删除模板接口 (DelPreset)	22
接口描述	22
请求方式	22
请求参数	22
返回参数	23
请求示例	23

返回示例	23
查询模板列表接口 (GetPresetList)	23
接口描述	23
请求方式	23
请求参数:	23
返回参数	23
请求示例	23
返回示例	24
查询模板详情接口 (GetPresetDetail)	24
接口描述	24
请求方式	24
请求参数	24
返回参数	24
请求示例	24
返回示例	24
任务队列说明	25
描述	25
暂停/激活任务队列接口 (UpdatePipeline)	25
功能描述:	25
请求方式:	25
请求参数:	25
返回参数:	25
请求示例:	25
返回示例:	26
查询任务队列信息接口 (QueryPipeline)	26
功能描述:	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:	27
创建任务接口 (CreateTask)	27
接口描述	27
请求方式:	27
请求参数:	27
返回参数	28
请求示例	28
返回示例	29
模板组合	29
拼接功能	29
创建流式任务接口 (CreateFlowTask)	29
功能描述:	29
请求方式:	29
请求参数:	29
返回参数:	30
请求示例:	30
返回示例:	31
删除任务接口 (DelTaskByTaskID)	31
功能描述:	31
请求方式:	31
请求参数:	31
返回参数:	31

请求示例:	31
返回示例:	31
置顶任务接口 (TopTaskByTaskID)	31
功能描述:	31
请求方式:	31
请求参数:	32
返回参数:	32
请求示例:	32
返回示例:	32
查询任务列表接口 (GetTaskList)	32
功能描述:	32
请求方式:	32
请求参数:	32
返回参数:	32
请求示例:	33
返回示例:	33
查询任务详情接口 (GetTaskByTaskID)	34
功能描述:	34
请求方式:	34
请求参数:	34
返回参数:	34
请求示例:	35
普通任务返回示例:	35
流式任务返回示例:	35
查询META信息接口 (GetTaskMetaInfo)	36
功能描述:	36
请求方式:	36
请求参数:	36
返回参数:	37
请求示例:	37
返回示例:	37
同步获取META信息接口 (FetchMetaInfo)	38
接口描述	38
请求方式	38
请求参数:	39
返回参数	39
请求示例	39
返回示例	39

## 点播转码概述

### 功能介绍

金山云视频云对外提供媒体转码服务，利用该服务客户可对存储在金山云存储平台（简称KS3）的音视频媒体文件进行多种媒体处理，获取对应的媒体信息。具体实现的功能包括：

- 文件格式及编码格式转换

支持在各种常见音视频格式与编码格式间进行转换。支持格式参考：<https://docs.ksyun.com/documents/1195>

- 视频帧嵌入水印

支持在视频帧上的自定义位置嵌入一个或多个自定义的水印图案

- 音视频切片处理

支持将指定的视频文件以指定的时长切分为多个媒体文件片段

- 视频帧截图和采样

支持获取指定视频文件的单张截图，或等时间间隔获取一系列视频帧截图

- 音视频信息提取

支持获取指定媒体文件的音视频参数信息

- 音视频拼接

支持不同编码类型的音视频拼接为一个指定编码格式和文件格式的音视频文件，前提是多个源的视频轨道数相同。

### 使用场景

当客户使用金山云存储平台（KS3）存储音视频媒体文件时，可开通并调用媒体转码服务，按指定的方式和参数对存储空间的媒体文件进行相关多媒体批量处理，获取对应的转码媒体文件或媒体文件参数信息。

### 使用方式

- 完成服务对接准备工作

点播转码服务的输入输出文件都在KS3的bucket上，因此，使用点播转码之前，主要包括开通bucket、申请AK/SK、将bucket授权给转码服务等步骤，详见：[点播转码服务-接入说明](#)

- 开通点播转码服务

开通服务之前，请确保已完成**服务对接准备工作**。请联系我们的技术支持，我们会第一时间为您开通试用。

- 获取对应的SDK

KS3用户开通点播转码服务后，推荐使用服务提供的各语言SDK对bucket媒体文件的多媒体处理。主要包含PHP/Python/JAVA三种语言SDK。如果您使用了其他的开发语言，请直接对接我们的OpenAPI，完成签名、HTTP调用、返回解析等操作。我们力争将API的描述得尽可能详细，但可能也存在遗漏，如有其他问题，请联系技术支持。

- 设置转码模板

转码模板是一套转码参数，同时也指定了转码流程，客户使用SDK调用模板管理API创建并设置转码模板。

- 创建多媒体处理任务

客户使用SDK，指定预设已经创建好的转码模板，并配置对应必备的任务参数、任务完成后的回调URL等信息，调用创建任务API，完成任务的创建。

**一定注意**，需要首先使用SDK的模板管理API预设自定义的转码通用参数，才可以调用任务管理API针对具体媒体文件创建单独的多媒体处理任务。

系统完成多媒体处理任务后将目标文件存入指定的存储空间，并根据预先设置的回调URL进行回调。

- 接收任务完成或出错的回调通知

点播转码是一个费时，也费资源的操作，推荐客户使用SDK创建任务时，带上回调通知URL，任务完成或执行失败后，服务会带上任务信息，回调这个URL。

## API调用方式

### 签名机制

点播转码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](#)

点播转码服务封装OpenAPI，对外提供三种语言(PHP/Python/JAVA)的SDK，推荐客户使用。

### 请求结构

客户调用金山云点播转码服务的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。

点播转码openAPI的请求结构组成如下：

1. 服务地址

Host	Service	Region
kvs.cn-beijing-6.api.ksyun.com	KVS	北京
kvs.cn-shanghai-2.api.ksyun.com	KVS	上海

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

点播转码的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点播转码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. 点播转码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。

## 调用成功

转码API服务端会返回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."
  }
}
```

## 示例

HTTP GET方法请求示例:

```
GET /?Action=GetPresetList&Version=2017-01-01&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20170519T060855Z&X-Amz-Credential=AKLT6UU6Vp_9SxWW5RAW6UhtdQ%2F20170519%2Fcn-beijing-6%2Fkvs%2Faws4_request&X-Amz-SignedHeaders=accept%3Bhost&X-Amz-Signature=0bfbb4876db7cf5b57206dbdcacf7f939e1ca395581e76d70a080b8c9c7bcd HTTP/1.1
```

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

Accept: application/json

HTTP 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%2Fkvs%2Faws4_request&X-Amz-SignedHeaders=accept%3Bhost&X-Amz-Signature=a8317822a11c587d529d779557e873e4a5fcc145788547a13cb4f9ad00d92e0d HTTP/1.1
```

Host: kvs.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。新版媒体转码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接入媒体转码**
- 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

## 公共错误

错误码	错误码说明
0	成功
1000	缺少参数
1001	参数不合法
1002	无效用户ID
1005	无效模板名
1009	无效任务ID
1010	内容不合法
1011	无效bucket
1012	非北京bucket
1013	bucket无法访问
1014	无效文件名
1015	无效模板类型
1016	源信息不合法
1017	源文件个数超限
1018	扩展参数不合法
1019	查询时间跨度太大
1020	无效目录名

1022	无效队列名
1023	队列未被初始化
1024	队列状态不合法
1101	用户未开通转码服务
1200	记录不存在
1201	记录已存在
1202	新增记录失败
1203	删除记录失败
1204	更新记录失败
1205	任务已完成或已删除
1210	文件不存在
1212	未授权桶策略
1213	源文件路径太长

## 回调用户错误码

错误码	错误码说明
0	Success
2001	Unknown Error
2002	machine reach max rss
2003	input param illegal
2004	task not exist
2005	task shutdown by user
2006	task sniff error
2007	channel not exist
2008	CommonChan task type illegal
2009	Parse url FAIL
2010	String length illegal
2011	Can not find what you want
2012	Daemon exit
2013	Call os function Error
2014	no idle daemon for task
2015	protobuf or json Unmarshal Error
2016	jsonpb Marshal Error
2017	variable in struct not set
2018	object count illegal
2019	this is a fake
2020	proto marshal to byte Error
2021	Slave res illegal
2022	Retrytimes illegal or reach thres
2023	GlobalValue content illegal
2024	Module fail to report to tasks_loop
2025	output_format parameter illegal
2026	taskvalue privMsg nil or illegal
2027	job operate data already exist
2028	mkdir FAIL
2029	doHttp input param lack or illegal
2030	doHttp canceled by user
2031	doHttp unknown error
2032	HTTP download file fail
2033	HTTP upload file fail
2034	http callback fail
2035	Slave already exist
2036	input file not ready
2037	not file or directory, or maybe wrong type
2038	operation time out
2039	httpd receive content illegal type
2040	file or dir mode not proper
2041	file not exist
2042	Notify to kts fail
2043	zmq socket allready exist
2044	set paramset fail
2045	form material illegal when POST
2046	Pem file decode fail
2047	init instance fail
2048	multiUpload fail
2049	Ban small to large
2050	operation specified illegal
2051	Extend Task Fail
2052	do not have daemon ID in doneMsgRes
2053	daemon response format illegal
2054	Module not exist
2055	type of body illegal
2056	can not get disk name of current directory
2057	slave fail to report to excutor or reach maxprocdura, maybe blocked
2058	sniffer result has AV headers but no audio and video data
2301	Snapshot Internal Error: read input file EOF
2302	Snapshot Internal Error: encode video error
2303	Snapshot Internal Error: codec context error
2304	Snapshot Internal Error: write frame fail
2305	Snapshot Internal Error: apply bitfilter fail
2306	Snapshot Internal Error: slave task empty
2307	Snapshot Internal Error: slave task full
2310	Snapshot Internal Error: parse json fail
2311	Snapshot Internal Error: json object null
2312	Snapshot Internal Error: zmq send fail

2313 Snapshot Internal Error: zmq recv error  
 2314 Snapshot Internal Error: notify to tworker fail  
 2315 Snapshot Internal Error: param check error  
 2319 Snapshot Internal Error: add input stream fail  
 2320 Snapshot Internal Error: open input file error  
 2321 Snapshot Internal Error: find streaminfo fail  
 2322 Snapshot Internal Error: new output stream fail  
 2323 Snapshot Internal Error: new output file fail  
 2324 Snapshot Internal Error: open output file fail  
 2325 Snapshot Internal Error: input/output stream null  
 2326 Snapshot Internal Error: dec/enc context null  
 2330 Snapshot Internal Error: malloc error  
 2331 Snapshot Internal Error: Sws GetContext error  
 2332 Snapshot Internal Error: Sws Scale error  
 2333 snapshot input lack param  
 2334 snapshot input alter param  
 2335 snapshot output path not set  
 2336 snapshot output Num larger than 3  
 2337 snapshot seek time maybe out of video duration  
 2350 Snapshot Internal Error: Unknown error  
 2401 probe inter error: read input file EOF  
 2403 probe inter error: codec context error  
 2406 probe inter error: slave task empty  
 2407 probe inter error: slave task full  
 2410 probe inter error: parse json fail  
 2411 probe inter error: json object null  
 2412 probe inter error: zmq send fail  
 2413 probe inter error: zmq recv error  
 2414 probe inter error: notify to tworker fail  
 2415 probe inter error: param check error  
 2420 probe inter error: open input file error  
 2421 probe inter error: find streaminfo fail  
 2422 probe inter error: new output stream fail  
 2423 probe inter error: new output file fail  
 2424 probe inter error: open output file fail  
 2425 probe inter error: input/output stream null  
 2426 probe inter error: dec/enc context null  
 2450 probe inter error: unknown error  
 2501 unknown request method  
 2502 allocate channel failed  
 2503 module not exist or unready  
 3022 Trans Internal Error: Invalid Argument  
 3140 Trans Internal Error: Bitstream filter not found  
 3141 Trans Internal Error: Internal bug  
 3142 Trans Internal Error: Buffer too small  
 3143 Trans Internal Error: Decoder not found  
 3144 Trans Internal Error: Demuxer not found  
 3145 Trans Internal Error: Encoder not found  
 3146 Trans Internal Error: End of file  
 3147 Trans Internal Error: Immediate exit was requested; the called function should not be restarted  
 3148 Trans Internal Error: Generic error in an external library  
 3149 Trans Internal Error: Filter not found  
 3150 Trans Internal Error: Invalid data found when processing input  
 3151 Trans Internal Error: Muxer not found  
 3152 Trans Internal Error: Option not found  
 3153 Trans Internal Error: Not yet implemented in segtran, patches welcome  
 3154 Trans Internal Error: Protocol not found  
 3155 Trans Internal Error: Stream not found  
 3156 Trans Internal Error: Internal bug2  
 3157 Trans Internal Error: Unknown error, typically from an external library  
 3158 Trans Internal Error: Requested feature is flagged experimental. Set strict\_std\_compliance if you really want to use it  
 3159 Trans Internal Error: Input changed between calls. Reconfiguration is required.  
 3160 Trans Internal Error: Output changed between calls. Reconfiguration is required.  
 3161 Trans Internal Error: Http Request 400 Bad Request  
 3162 Trans Internal Error: Http Request 401 Unauthorized  
 3163 Trans Internal Error: Http Request 403 Forbidden  
 3164 Trans Internal Error: Http Request 404 Not Found  
 3165 Trans Internal Error: Http Request 4xx Other Error  
 3166 Trans Internal Error: Http Request 5xx Server Error  
 3175 Trans Internal Error: Internal Other Error  
 3177 Trans Internal Error: first decoded frame pts\_time > start\_time. Maybe lead to non-precise timestamp seek  
 3178 Trans Internal Error: Video/Audio Illegal: pts of first packet is more than zero!  
 3180 slave connect to zmq error  
 3181 slave argv too short  
 3182 slave get bin path error  
 3183 slave move mp4 file header error  
 3184 slave input authentication expired  
 3186 sniffer file unsupport seekable in AVIOContext structure  
 3187 sniffer sniff time out  
 3188 src m3u8 file has illegal segment num  
 3189 sniffer pthread\_create() fail!  
 3190 sniffer reach proc duration thres  
 3200 slave param error  
 3201 the time seeking backward too large, exit!  
 3202 input stream format changed, frame parameters mismatch context

```

3203 slave open output file fail
3204 error reinitializing filters!
3205 error applying options to the filter
3235 pthread_create fail!
3236 init_msgchannel_context fail!
3240 connection timeout
3241 connection reset by peer
3242 connection refused by server
3243 server terminate the connection
3255 Trans Internal Error: System Signal is caught and Slave proces is killed

```

## 转码结果回调用户

### POST 回调接口

视频云提供HTTP POST的回调Body如下：

字段名	类型	是否必有	描述
TaskID	string	是	任务ID
Status	int	是	任务状态：3为成功，4为失败
Type	string	是	任务类型，与创建任务时使用的模板类型一致
Details	json object	是	更详细的信息

- 对Details更详细的说明：

字段名	类型	是否必有	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
bucket	string	是	输出目标文件的Bucket
snapshotNum	int	否	截图的个数，截图任务有效
status	string	否	任务当前所处的状态：transcoding、uploading。
progress	int	否	任务当前Status下的进度百分比。仅适用于长视频，且未设置时默认为0
items	json object	是	回调中用户定制的输出文件信息等。

备注：status、progress只有在用户在创建任务时，设置了定时回调进度信息才会有效。

- 对items更详细的说明：

字段名	类型	是否必有	描述
dspath	string	是	输出文件在ks3的相对路径
metainfo	string	否	avinfo任务的回调字段，源文件信息。string类型，string内部是json串
outMediainfo	string	否	avtrans任务的回调字段，输出视频的信息。string类型，string内部是json串
segmentlisttype	string	否	avtrans任务的字段，切片文件列表的类型，不为空“”，表示为切片任务，否则为普通转码任务，默认为空。
segmentsinfo	string	否	当任务是切片任务时（非HLS切片），每个切片的文件信息。string类型，string内部是json串，数组形式
thumbnailsinfo	string	否	生成雪碧图时，每张雪碧图的文件信息。string类型，string内部是json串，数组形式

- 对输出文件信息metainfo的说明

字段名	类型	是否必有	描述
streams	json array	是	流信息列表
format	json object	是	文件信息

streams部分：

字段名	类型	是否必有	描述
index	int	是	流索引值
codec_name	string	是	编码器名称
codec_long_name	string	是	编码器全称
codec_type	string	是	流类型，video/audio/data/subtitle等
codec_time_base	string	否	编码器的每帧时长，时间戳计算基础单位，分数表示
codec_tag_string	string	否	编码器的标签名
codec_tag	string	否	编码器标签的二进制代号
time_base	string	是	每帧时长，时间戳计算基础单位，分数表示
start_time	string	是	首帧时间，单位：秒
duration	string	否	该路流时长，单位：秒
duration_ts	int	否	该路流时长，以time_base为单位
bit_rate	string	是	该路流平均码率
max_bit_rate	string	否	该路流最大码率
nb_frames	string	否	音视频帧数
width	int	是	视频帧的高，文件必须有视频流
height	int	是	视频帧的宽，文件必须有音频流
coded_width	int	否	编码器对齐的帧宽
coded_height	int	否	编码器对齐的帧高
has_b_frames	int	否	包含B帧信息
sample_aspect_ratio	string	否	图像采样率
display_aspect_ratio	string	否	图像显示比例
pix_fmt	string	是	帧的图像色彩格式
profile	string	否	编码的profile
level	int	否	编码的level
refs	int	否	B和P帧向前预测参考的帧数
is_avc	string	否	封装格式是否是avc 1代表avc 0表示annexb格式
rotate	string	否	输出为转码后文件的rotate，与源文件无关，一般系统会自动旋转并删除rotate防止终端播放二次旋转
sample_fmt	string	是	音频采样格式
sample_rate	string	是	音频采样率
channels	int	是	音频声道数，对于音轨数>2的文件，可能该值不准
channel_layout	string	否	声道设计，是指的单声道，双声道，立体声

format部分：



字段名	类型	是否必有	描述
nb_streams	int	是	文件包含的流个数
format_name	string	是	文件封装格式
format_long_name	string	是	文件封装格式全称
duration	string	是	文件总时长, 单位: 秒
size	int	是	文件大小, 单位: bytes
bit_rate	string	是	文件平均码率, 单位bps

- 对输出文件信息outMediaInfo的说明

字段名	类型	是否必有	描述
outMediaInfo	string	是	视频信息
metadata	json object	是	视频metadata
streams	json array	是	视频流信息列表
format	json object	是	视频文件信息

#### metadata部分

字段名	类型	是否必需	描述
rotate	string	否	旋转信息

#### streams部分

字段名	类型	是否必需	描述
index	int	是	序号
codec_type	string	是	流类型
codec_name	string	是	编码格式
bit_rate	int	是	码率, 单位bps
width	int	否	宽
height	int	否	高
r_frame_rate	string	否	帧率
channels	int	否	声道数
sample_rate	int	否	音频采样率

#### format部分

字段名	类型	是否必需	描述
duration	string	是	文件时长, 单位秒
format_name	string	是	文件容器格式
size	int	是	文件大小, 单位字节
bit_rate	string	是	文件平均码率, 单位bps

- 对切片文件信息segmentsInfo的说明:

字段名	类型	是否必需	描述
fileindex	int	是	切片文件的顺序, 从0开始
filename	string	是	切片文件的名字
duration	string	是	每个切片的实际时长, 字符串内用小数表示, 单位为秒, 如"10.000", 表示10s
filesize	int	是	文件大小, 单位字节

- 对雪碧图信息thumbnailsInfo的说明:

字段名	类型	是否必需	描述
fileindex	int	是	雪碧图文件的顺序, 从0开始
thumbnailpath	string	是	雪碧图在ks3中的存储路径, 不包括bucket信息
filesize	int	是	文件大小, 单位字节

#### POST回调接口的Body的格式示例:

注: offline老用户回调内容格式, json一级参数没有驼峰

```
{
  //任务id type:string
  "taskId": "92e3745751aee5662f9f06ff3f03076920160914",
  //处理中:2 成功:3 失败: 4 type:int
  "status": 2,
  //任务类型:与创建请求的任务类型一致
  "type": "avtrans",
  //视频云媒体转码返回信息
  "details": {
    ...
  }
}
```

- avtrans 任务进度回调内容格式示例

```
{
  //任务id type:string
  "TaskID": "92e3745751aee5662f9f06ff3f03076920160914",
  //处理中:2 成功:3 失败: 4 type:int
  "Status": 2,
  //任务类型:与创建请求的任务类型一致
  "Type": "avtrans",
```

```
//视频云媒体转码返回信息
"Details": {
  //转码处理状态 transcoding 转码中, uploading 上传中 type:string
  "status": "transcoding",
  //转码进度百分比 type:int
  "progress": 5
}
}
```

- avtrans任务中segment切片任务（非HLS）回调内容格式示例

```
{
  //任务id type:string
  "TaskID": "92e3745751aee5662f9f06ff3f03076920160914",
  //处理中:2 成功:3 失败: 4 type:int
  "Status": 3,
  //任务类型:与创建请求的任务类型一致
  "Type": "avtrans",
  //视频云媒体转码返回信息
  "Details": {
    //成功: 0 失败: 其他 type:int
    "errnum": 0,
    //bucket地址 type:string
    "bucket": "wangshuai9",
    "items": [
      [
        //文件存储地址 type:string
        "dstpath": "m3u8/test/filelist.m3u8"
        //输出文件视频信息
        "segmenglisttype": "m3u8",
        "segmentsinfo": [
          [
            {
              "fileindex": 0,
              "filename": "segmentfile-000.mp4",
              "duration": "10.0000",
              "filesize": 67778
            },
            {
              "fileindex": 1,
              "filename": "segmentfile-001.mp4",
              "duration": "11.0000",
              "filesize": 76578
            },
            {
              "fileindex": 2,
              "filename": "segmentfile-002.mp4",
              "duration": "10.0000",
              "filesize": 54776
            },
            {
              "fileindex": 3,
              "filename": "segmentfile-003.mp4",
              "duration": "12.0000",
              "filesize": 76564
            }
          ]
        ]
      ]
    ]
  }
}
```

- avtrans、avsample、avsnapshot 等普通任务结束回调内容格式示例

```
{
  //任务id type:string
  "TaskID": "92e3745751aee5662f9f06ff3f03076920160914",
```

```
//处理中:2 成功:3 失败: 4 type:int
"Status": 3,
//任务类型:与创建请求的任务类型一致
"Type": "avtrans",
//兼容ks3老用户需求, avtrans任务没有该参数
"times": [
  {
    "cmd": "tag=avop|tag=saveas&bucket=wangshuai9&object=bTN10C90ZXN0LzEubXA",
    "desc": "success",
    "keys": [
      "m3u8/test/shouyin1.mp4"
    ],
    "code": 3
  }
],
//视频云媒体转码返回信息
"Details": {
  //成功: 0 失败: 其他 type:int
  "errnum": 0,
  //bucket地址 type:string
  "bucket": "wangshuai9",
  //截图总量
  //该参数只有avsample任务有效
  "snapshotNum": 0,
  "items": [
    {
      //文件存储地址 type:string
      "dstpath": "m3u8/test/shouyin1.mp4"
      //输出文件视频信息
      "outMediaInfo": "{
        "metadata": {
          "rotate": "10"
        },
        "streams": [
          {
            "index": 0,
            "codec_name": "h264",
            "codec_type": "video",
            "width": 1280,
            "height": 720,
            "bit_rate": "500000",
            "r_frame_rate": "30.1"
          },
          {
            "index": 1,
            "channels": 2,
            "sample_rate": 44100,
            "codec_name": "aac",
            "codec_type": "audio",
            "bit_rate": "64000"
          }
        ],
        "format": {
          "duration": "198.123",
          "format_name": "mp4",
          "size": 198
        }
      }"
    }
  ]
}
```

## • avinfo 任务回调格式示例

```
{
  "TaskID": "0c33e9c372ff87fe197cb0dcf236e18d20161223",
  "Status": 3,
  "Type": "avinfo",
  // 为兼容ks3老客户, 新用户接入时, 不带有items字段, 请忽略
  "items": [
    {
      "cmd": "tag=avinfo",
      "desc": "success",
      "keys": [
        ""
      ],
      "code": 3
    }
  ],
  "Details": {
    "errnum": 0,
    "bucket": "",
    "items": [
      {
        "dstpath": "",
        "metainfo": {
          "streams": [
            {
              "codec_name": "aac",
              "codec_long_name": "AAC (Advanced Audio Coding)",
              "profile": "LC",
              "codec_type": "audio",
              "codec_time_base": "1/48000",
              "codec_tag_string": "mp4a",
              "codec_tag": "0x6134706d",
              "r_frame_rate": "0/0",
              "avg_frame_rate": "0/0",
              "time_base": "1/48000",
              "start_time": "0.000000",
              "duration_ts": 61673448,
              "duration": "1284.863500",
              "bit_rate": 47325,
              "nb_frames": 60228,
              "sample_fmt": "fltp",
              "sample_rate": 48000,
              "channels": 2,
              "channel_layout": "stereo"
            },
            {
              "index": 1,
              "codec_name": "h264",
              "codec_long_name": "H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10",
              "profile": "High",
              "codec_type": "video",
              "codec_time_base": "642153/15575000",
              "codec_tag_string": "avc1",
              "codec_tag": "0x31637661",
              "r_frame_rate": "13/1",
              "avg_frame_rate": "7787500/642153",
              "time_base": "1/16000",
              "start_pts": 10176,
              "start_time": "0.636000",
              "duration_ts": 20548896,
              "duration": "1284.306000",
              "bit_rate": 781679,
            }
          ]
        }
      }
    ]
  }
}
```



```
//子任务的执行结果状态码
"Status": 3,
//成功: 0 失败: 其他 type:int
"Errnum": 0,
"Errmsg": "",
//bucket地址 type:string
"Bucket": "xxxxx",
"OutInfo": {
  //文件存储地址 type:string
  "dstpath": "xxxx/xxx/xxx.mp4",
  //输出文件视频信息
  "outMediaInfo": "{
    "metadata":{
      "rotate":"10"
    },
    "streams":[
      {
        "index":0,
        "codec_name":"h264",
        "codec_type":"video",
        "width":1280,
        "height":720,
        "bit_rate":"500000",
        "r_frame_rate":"30.1"
      },
      {
        "index":1,
        "channels":2,
        "sample_rate":44100,
        "codec_name":"aac",
        "codec_type":"audio",
        "bit_rate":"64000"
      }
    ],
    "format":{
      "duration":"198.123",
      "format_name":"mp4",
      "size":198
    }
  }"
}
},
{
  "Order": 1,
  "Type": "avtrans",
  //子任务的执行结果状态码
  "Status": 3,
  //成功: 0 失败: 其他 type:int
  "Errnum": 0,
  "Errmsg": "",
  //bucket地址 type:string
  "Bucket": "xxxxx",
  "OutInfo": {
    //文件存储地址 type:string
    "dstpath": "xxxx/xxx/xxx.mp4",
    //输出文件视频信息
    "outMediaInfo": "{
      "metadata":{
        "rotate":"10"
      },
      "streams":[
        {
```

```

        "index":0,
        "codec_name":"h264",
        "codec_type":"video",
        "width":1280,
        "height":720,
        "bit_rate":"200000",
        "r_frame_rate":"25"
    },
    {
        "index":1,
        "channels":2,
        "sample_rate":44100,
        "codec_name":"aac",
        "codec_type":"audio",
        "bit_rate":"64000"
    }
],
"format":{
    "duration":"220.456",
    "format_name":"flv",
    "size":328
}
}
]
}

```

## GET 回调接口

目前不提供

## 模板说明

- [1. 模板类型说明](#)
- [2. 公有参数说明](#)
- [3. 四类模板Param详细说明](#)
  - [avtrans模板](#)
  - [avsample模板](#)
  - [avsnapshot模板](#)
  - [avinfo模板](#)

### 模板类型说明

媒体转码模板是以Json格式预先设定的一组或多组转码参数，用来指定实施转码之后的目标媒体文件的特性参数以及转码的处理方式，包括视频分辨率、码率、帧率以及加水印、截图等相关参数，客户使用媒体转码服务时需事先定义好转码模板，然后再根据模板来执行相关任务。

根据转码任务类型的不同媒体转码模板分为以下四种类型：

- avtrans：转码/转码拼接/格式转换/水印/切片
- avsnapshot：单张截图
- avinfo：音视频信息获取

### 公有参数说明

名称	类型	必填项	描述
Preset	string	是	模板名称，由2~32位字母、数字、下划线组成
Description	string	否	模板描述
PresetType	string	是	模板类型，目前支持avtrans/avsample/avsnapshot/avinfo，媒体转码模板命名空间模板名+模板类型必须唯一
Param	json object	是	模板详情参数集，用于设定具体的转码参数，不同类型的模板具有不同的Param参数形式

公有参数格式定义：

所有媒体转码模板具有统一的格式：

```

{
  "Preset": "mypreset",
  "Description": "demo",
  "PresetType": "avtrans",
  "Param": {
    ...
  }
}

```

## 四类模板Param详细说明

### 1. avtrans模板Param格式定义：

名称	类型	必填项	描述
f	string	是	转码输出的容器格式 (ContainerFormat)，可选值：mp4, flv, hls, mpegts, mp3, adts, dash, gif。

AUDIO	json object	否	指定输出文件音频流的参数，详见下方AUDIO参数详情
VIDEO	json object	否	指定输出文件视频流的参数，详见下方VIDEO参数详情
LOGOS	json array	否	往输出文件中加一张logo。
mulrate	int	否	Dash多码率转码开关，0:关闭 1:开启，开启后支持dash多码率转码，仅适配mulAb（音频多码率）、mulVb（视频多码率）、mulRes（视频多分辨率）参数设定，音频多码率与视频多码率位置一一对应例如mulVb为 2m:1200k:800k，mulres为1920x1080:1280x720:600x360，那么2m与1920x1080即为映射关系

## 转码音频参数详情 (AUDIO)

名称	类型	必填项	描述
ab	string	否	音频码率，支持k表示，如：128k。取值范围 20000~320000
mulAb	string	否	在mulrate=1情况下有效，音频多码率，格式：128k:64k:32k
ar	string	否	音频采样频率，取值范围 8000~96000
an	int	否	是否去除音频流，0：不去除（默认）；1：去除
acodec	string	否	音频编码器，支持：aac（默认）、mp3、copy。设置copy时，请保证容器格式与codec兼容。当f输出格式为mp3时，acodec必须设置为mp3
profile	string	否	音频编码预置，当acodec为aac时，范围aac_low、aac_he，默认逻辑：码率小于64k为aac_he，否则为aac_low，设置为aac_low采样率保持不变 设置为aac_he采样率会变小大约一半

## 转码视频参数详情 (VIDEO)

名称	类型	必填项	描述
vr	string	否	视频帧率，取值范围 [1,60]
vb	string	否	视频码率，支持k、m 如：400k。取值范围 30000~50000000
vcodec	string	否	视频编码器，支持：h264(默认)、h265
width	int	否	视频分辨率宽，如果只指定任一，会按原视频宽高比缩放到指定值，需要在as=1情况下使用
height	int	否	视频分辨率高，如果只指定任一，会按原视频宽高比缩放到指定值，需要在as=1情况下使用
mulVb	string	mulrate=1时，必填	在mulrate=1情况下有效，视频多码率，支持k、m格式：2m:1200k:800k
mulRes	string	mulrate=1时，必填	在mulrate=1情况下有效，格式： 1920x1080:1280x720:600x360，视频多分辨率与多码率参数一一对应，支持按原比例等比缩放，自动计算另一边长格式为例如-1x360，待计算边长用-1替代
mulVr	string	mulrate=1时，必填	在mulrate=1情况下有效，格式： 24:23:30，视频帧率与视频分辨率、视频码率一一对应
shortSide	int	否	设置输出视频短边，程序按照原始比例自动缩放。如果需要模板支持横竖屏视频自动判断转码，则设置该参数。该参数需要在as=1情况下使用
as	int	否	是否按原比例进行缩放，取值范围 0：不自动缩放（默认）；1：自动缩放
vn	int	否	是否去除视频流，取值范围 0：不去除（默认）；1：去除
intervalframes	int	否	转gif等动态图时，指定抽帧间隔，单位为帧，不设默认不抽帧
loop	int	否	gif与webp循环次数，默认不循环
finaldelay	int	否	gif的loop间隔，单位为百分之一秒，比如500代表间隔为5s

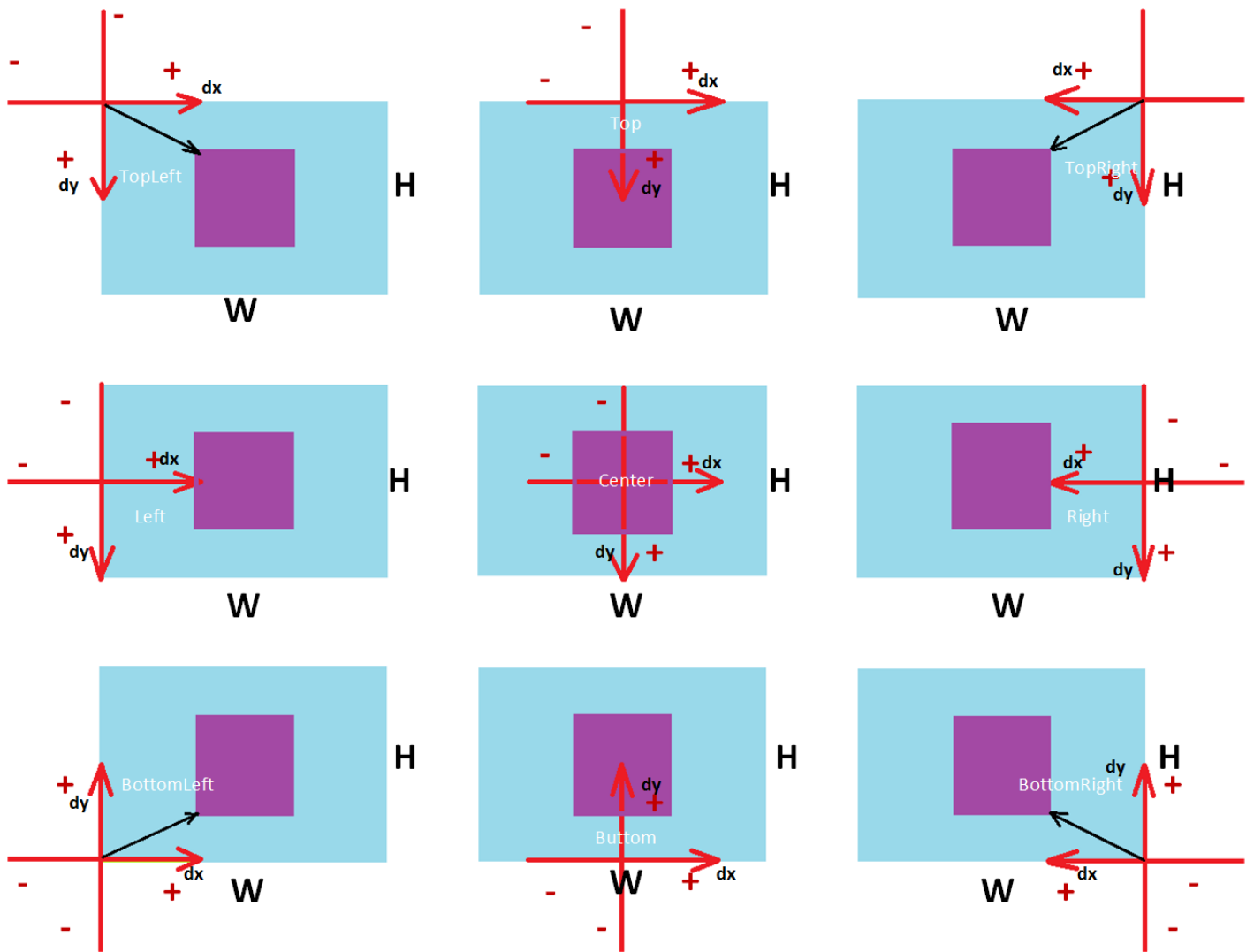
## 转码水印参数详情 (LOGOS)

名称	类型	必填项	描述
url	string	是	必填，水印图片地址，需要ks3的内网地址。该pic必须是不需要签名就可以访问的，public。该pic必须是与转码服务在同一region，即转码源文件在上海bucket，pic也要在上海bucket。
offsetX	int	否	pic宽的偏移值，向右为正。可选值，默认把pic放在左上角，offsetX=10,offsetY=10。如果offsetX=-10,offsetY=10，则Pic放在视频右上角。
offsetY	int	否	pic高的偏移值，向下为正。可选值，默认把pic放在左上角，offsetX=10,offsetY=10。如果offsetX=-10,offsetY=10，则Pic放在视频右上角。
ss	int	否	设置logo的进入时间，单位为秒。不设默认为0：从头开始
duration	int	否	设置logo的显示持续时间，单位为秒。不设默认为0：持续到结束，video类型logo持续到video结束。
referPos (推荐设置)	string	否	水印的位置，每个位置有对应的锚点，TopRight、TopLeft、BottomRight、BottomLeft、Top、Bottom、Left、Right、Center。默认TopLeft。
width	string	否	string类型，值有两种形式：整数型代表水印图片宽的像素值，范围[8,4096]，单位px；小数型代表相对输出视频分辨率宽/高的比率，范围(0,1]，支持4位小数，如0.9999，超出部分系统自动丢弃。小数时可添加 h 或 w 指定比率的参考是视频的宽或高，如 0.26*h 表示占视频高的0.26。
height	string	否	同上。
opacity	string	否	不透明度设置，范围(0-1]，值越大不透明度越高，1表示完全不透明。不设置默认为1。支持1位小数，比如0.8，超出部分自动丢弃。
loop	int	否	动态logo轮询次数，目前只支持gif格式的水印，0为无限循环，不设置默认为0。

水印宽、高参数设置说明：

水印位置坐标说明：





## 2. avsample模板Param格式定义:

avsample模板支持处理截图视频格式包含 (MP4, flv, m3u8, mov, wmv, avi, mpg, mpeg) 输出格式为根据dstobjectkey指定, 支持jpg、png、默认为jpg

名称	类型	描述	必填项
width	int	图片分辨率宽。如果只指定width/height任一, 会按原比例缩放	否
height	int	图片分辨率高。如果只指定width/height任一, 会按原比例缩放	否
interval	int	间隔, 单位s	是

avsample模板的param示例如下:

```
{
  "width": xxx,
  "height": xxx,
  "interval": xxx,
}
```

## 3. avsnapshot模板Param格式定义:

avsnapshot模板支持处理截图视频格式包含 (MP4, flv, m3u8, mov, wmv, avi, mpg, mpeg) 输出格式为根据dstobjectkey指定, 支持jpg、png、默认为jpg

名称	类型	必填项	描述
width	int	否	图片分辨率宽。宽高必须全部设定, 除非设置了shortSide, 否则会按源分辨率生成截图
height	int	否	图片分辨率高。宽高必须全部设定, 除非设置了shortSide, 否则会按源分辨率生成截图
shortSide	int	否	设置输出截图短边, 程序按照原始比例自动缩放。如果源视频既可能横屏, 也可能竖屏, 需要模板支持自动判断生成截图, 则设置该参数。该参数优先级最高
ss	string	是	设置截图的时间点, 单位s

avsnapshot的param示例如下:

```
{
  "width": xxx,
  "height": xxx,
  "shortSide": 360,
  "ss": "30"
}
```

## 4. avinfo模板Param格式定义:

```
{
  //avinfo无需参数
}
```

注：以上模板格式中，参数值均为示例值。

## 模板（preset）使用示例

需求场景：我想对mp4文件做hls切片。

```
{
  "Preset": "mypreset_hls",
  "Description": "demo_hls",
  "PresetType": "avtrans",
  "Param": {
    "f": "hls",
    "segment_time": 6,
    "segfile_acl": "public-read",
    "AUDIO": {
      "acodec": "copy"
    },
    "VIDEO": {
      "vcodec": "copy"
    }
  }
}
```

需求场景：我只想在源视频上加水印，但不打算改变源视频的分辨率、码率等参数。

```
{
  "Preset": "mypreset_0",
  "Description": "demo_logo",
  "PresetType": "avtrans",
  "Param": {
    "f": "mp4",
    "LOGOS": [
      [
        {
          "url": "http://ks3-cn-beijing-internal.ksyun.com/{bucket_name}/{object_name}", //logo路径需是ks3内网地址，属性为public
          "logotype": "image",
          "usage_scenario": "ALL",
          "shortSide": 480, // 或者"optimal_w": 640, 或者"optimal_h": 480,
          "disable_scale": 0,
          "offsetX": -10,
          "offsetY": 10
        }
      ]
    ]
  }
}
```

需求场景：我为了水印效果更好，不想水印被过度缩放，所以准备了不同尺寸的2个水印logo文件，想根据转出视频不同尺寸加相应的水印logo（只加一个）。

```
{
  "Preset": "mypreset_logoselect",
  "Description": "demo_logoselect",
  "PresetType": "avtrans",
  "Param": {
    "f": "mp4",
    "AUDIO": {
      "acodec": "copy"
    },
    "VIDEO": {
      "vcodec": "h264",
      "preset": "veryfast"
    },
    "LOGOS": [
      [
        {
          "url": "http://ks3-cn-beijing-internal.ksyun.com/{bucket_name}/{object_name0}",
          "logotype": "image",

```

```

        "usage_scenario": "ALL",
        "shortSide": 480, // 或者"optimal_w": 640, 或者"optimal_h": 480,
        "disable_scale": 0,
        "offsetX": -10,
        "offsetY": 10
    },
    {
        "url": "http://ks3-cn-beijing-internal.ksyun.com/{bucket_name}/{object_name1}",
        "logotype": "image",
        "usage_scenario": "ALL",
        "shortSide": 720, // 或者"optimal_w": 1280, 或者"optimal_h": 720,
        "disable_scale": 0,
        "offsetX": -15,
        "offsetY": 15
    }
]
}
}
}

```

需求场景：我想加多个水印logo，以及水印logo要在视频开始后某个时间点开始显示。

```

{
    "Preset": "mypreset_multilogo",
    "Description": "demo_multilogo",
    "PresetType": "avtrans",
    "Param": {
        "f": "mp4",
        "AUDIO": {
            "acodec": "copy"
        },
        "VIDEO": {
            "vr": "25",
            "vb": "500k",
            "vcodec": "h265",
            "height": 480,
            "as": 1,
            "preset": "veryfast"
        },
        "LOGOS": [
            [
                {
                    "url": "http://ks3-cn-beijing-internal.ksyun.com/{bucket_name}/{object_name_0}", //第一个logo路径
                    "logotype": "image",
                    "usage_scenario": "ALL",
                    "shortSide": 480, // 或者"optimal_w": 640, 或者"optimal_h": 480,
                    "disable_scale": 0,
                    "offsetX": -10,
                    "offsetY": 10
                }
            ],
            [
                {
                    "url": "http://ks3-cn-beijing-internal.ksyun.com/{bucket_name}/{object_name_1}", //第二个logo路径
                    "logotype": "image",
                    "usage_scenario": "ALL",
                    "shortSide": 480, // 或者"optimal_w": 640, 或者"optimal_h": 480,
                    "disable_scale": 0,
                    "offsetX": -10,
                    "offsetY": -10,
                    "ss": 5 //视频开始5秒后开始显示此logo
                }
            ]
        ]
    }
}
}

```

}

需求场景：我想对源视频拼接片头片尾的同时加多个水印，且第一个水印只显示在源视频上，第二个水印的显示贯穿全程。

```
{
  "Preset": "mypreset_multilogo",
  "Description": "demo_multilogo",
  "PresetType": "avtrans",
  "Param": {
    "f": "mp4",
    "AUDIO": {
      "acodec": "copy"
    },
    "VIDEO": {
      "vr": "25",
      "vb": "500k",
      "vcodec": "h265",
      "height": 480,
      "as": 1,
      "preset": "veryfast"
    },
    "LOGOS": [
      [
        {
          "url": "http://ks3-cn-beijing-internal.ksyun.com/{bucket_name}/{object_name_0}", //第一个logo路径
          "logotype": "image",
          "usage_scenario": "ALL",
          "effect_objects": "1", //表示logo加在拼接前的第二个视频上。
          "shortSide": 480, // 或者"optimal_w": 640, 或者"optimal_h": 480,
          "disable_scale": 0,
          "offsetX": -10,
          "offsetY": 10
        }
      ],
      [
        {
          "url": "http://ks3-cn-beijing-internal.ksyun.com/{bucket_name}/{object_name_1}", //第二个logo路径
          "logotype": "image",
          "usage_scenario": "ALL",
          "effect_objects": "", //设为空, 表示logo加在拼接片头片尾后的视频上。
          "shortSide": 480, // 或者"optimal_w": 640, 或者"optimal_h": 480,
          "disable_scale": 0,
          "offsetX": -10,
          "offsetY": -10,
          "ss": 5 //视频开始5秒后开始显示此logo
        }
      ]
    ]
  }
}
```

需求场景：我想对源视频拼接片头片尾

拼接任务，模板设为普通的avtrans类型，在创建任务的时候SrcInfo设置为数组即可。

需求场景：集智高清

```
"VIDEO": {"codecParams": {"rctype": 3, "qmax": 34, "iper": 184, "crf": 23, "preset": "medium", "vcodec": "h265"},
"f": "mp4",
"zdParams": {"preprocess": "video_type=custom:thread_count=16:filter_para=DnnEnhance 3|CommonSharpen 1",
kshd=1
```

## 创建模板接口(Preset)

### 功能描述：

提供预设媒体转码参数模板的功能

### 请求方式：

POST

## 请求参数：

参数名	类型	是否必须	描述
Preset	string	是	模板名
PresetType	string	是	模板类型，详见 <a href="#">模板说明</a> 对类型的描述
Param	JSON	是	模板定义，详见 <a href="#">模板说明</a> 的参数定义
Description	string	否	描述字段

## 返回参数：

参数名	类型	是否必须	描述
ErrNum	int	是	错误码。0：成功；非0：失败
ErrMsg	string	是	错误信息

## 请求示例：

```
POST /?Action=Preset
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170519T043000Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=accept%3Bhost
&X-Amz-Signature=xxxx HTTP/1.1
Host: kvs.cn-beijing-6.api.ksyun.com
Accept: application/json
Content-Type: application/json;charset=utf-8
Content-Length: %LEN
{
  "Preset": "myAvtransPreset",
  "Description": "test preset",
  "PresetType": "avtrans",
  "Param": {
    "f": "mp4",
    "intelligentSwitch": 0,
    "discontinue": 1,
    "AUDIO": {
      "ab": "164k",
      "ar": "44100",
      "acodec": "aac",
      "an": 0,
      "silenceFrame": 1
    },
    "VIDEO": {
      "vr": "25",
      "vb": "400k",
      "vcodec": "h264",
      "width": 1280,
      "height": 720,
      "as": 0,
      "shortSide": 360,
      "rotate": "auto",
      "vn": 0,
      "orientationAdapt": 1
    }
  }
}
```

## 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

提供修改已设定的离线转码参数模板的功能

### 请求方式

POST

### 请求参数

参数名	类型	是否必需	描述
Preset	string	是	模板名称
PresetType	string	是	模板类型, 见 <a href="#">模板说明文档</a>
Param	JSON	是	模板内容, 见 <a href="#">模板说明文档</a>
Description	string	否	模板描述

### 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-Credential=xxxxx
```

```
&X-Amz-Date=20160914T114902Z
```

```
&X-Amz-SignedHeaders=host
```

```
&X-Amz-Signature=xxxxx
```

```
Host: kvs.cn-beijing-6.api.ksyun.com
```

```
Accept: application/json
```

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

```
Content-Length: %LEN
```

```
{
  "Preset": "myAvsnapshotPreset",
  "Description": "test preset",
  "PresetType": "avsnapshot",
  "Param": {
    "width": xxx,
    "height": xxx,
    "shortSide": 360,
    "ss": "30"
  }
}
```

### 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

删除已设定的媒体转码参数模板

### 请求方式

GET

### 请求参数

参数名	参数类型	必选项	描述
Preset	string	是	模板名称

## 返回参数

参数名	参数类型	是否必须	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

```
GET /?Action=DelPreset
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170519T063845Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx
&Preset=testpreset HTTP/1.1
Host: kvs.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

提供查询预设参数模板列表的功能

### 请求方式

GET

### 请求参数：

参数名	类型	是否必须	描述
WithDetail	int	否	是否查询模板详情，1-是 0-否
PresetType	string	否	模板类型，多种模板类型以逗号隔开
Presets	string	否	模板名称，多个模板名称以逗号隔开

### 返回参数

参数名	类型	是否必须	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
PresetList	json array	是	模板列表信息
Preset	string	是	模板名称
PresetType	string	是	模板类型
Description	string	否	模板描述
CreateTime	string	是	模板创建时间
UpdateTime	string	是	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=20170519T060855Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx HTTP/1.1
HOST: kvs.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":"myAvtransPreset",
      "PresetType":"avtrans",
      "Description":"desc:preset_avtrans",
      "CreateTime":"2016-08-15 13:41:46",
      "UpdateTime":"2016-08-15 13:42:12"
    },
    {
      "Preset":"myAvsnapshotPreset",
      "PresetType":"avsnapshot",
      "Description":"desc:preset_avsnapshot",
      "CreateTime":"2016-08-16 10:55:10",
      "UpdateTime":""
    }
  ],
  "ErrNum":0,
  "ErrMsg":"success"
}

```

## 查询模板详情接口(GetPresetDetail)

### 接口描述

提供查询预设参数模板详情的功能

### 请求方式

GET

### 请求参数

参数名	类型	是否必须	描述
Preset	string	是	模板名称

### 返回参数

参数名	类型	是否必须	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
PresetDetail	Json Object	是	模板详情, 详见 <a href="#">模版说明</a>
Param	Json Object	是	是
PresetType	string	是	模板类型
Description	string	否	模板描述
CreateTime	string	是	模板创建时间
UpdateTime	string	否	模板更新时间

### 请求示例

```

GET /?Action=GetPresetDetail
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx
&Preset=testpreset HTTP/1.1
Host: kvs.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": "test preset",
    "PresetType": "avtrans",
    "Param": {
      "f": "mp4",
      "intelligentSwitch": 0,
      "discontinue": 1,
      "AUDIO": {
        "ab": "164k",
        "ar": "44100",
        "acodec": "acc",
        "an": 0,
        "silenceFrame": 1
      },
      "VIDEO": {
        "vr": "25",
        "vb": "400k",
        "vcodec": "h264",
        "width": 1280,
        "height": 720,
        "as": 0,
        "shortSide": 360,
        "rotate": "auto",
        "vn": 0,
        "orientationAdapt": 1
      }
    }
  },
  "CreateTime": "2016-08-16 10:55:10",
  "UpdateTime": ""
},
"ErrNum": 0,
"ErrMsg": "success"
}

```

## 任务队列说明

### 描述

离线转码任务队列是一种作业队列，转码作业加入到任务队列中，才会被媒体转码服务调度到转码执行；如果任务队列中作业数量过多，会排队等待。任务队列有激活、暂停两种状态，任务队列被暂停时，媒体转码服务不再调度此任务队列中的作业执行转码，新提交到客户队列中的任务也会暂停，直到任务队列被重新激活为止，已经在执行转码的作业不受影响。

## 暂停/激活任务队列接口(UpdatePipeline)

### 功能描述：

更新任务队列的激活/暂停状态

### 请求方式：

POST

### 请求参数：

参数名	类型	是否必需	备注
PipelineName	string	是	队列名，默认为：usual
State	string	否	Active: 激活, Paused: 暂停

### 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	错误信息

### 请求示例：

```

POST /?Action=UpdatePipeline
&Version=2017-01-01

```

```
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170519T043000Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=accept%3Bhost
&X-Amz-Signature=xxxx HTTP/1.1
Host: kvs.cn-beijing-6.api.ksyun.com
Accept: application/json
Content-Type: application/json;charset=utf-8
Content-Length: %LEN
{
  "PipelineName": "usual",
  "State": "Active",
}
```

## 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"
}
```

## 查询任务队列信息接口(QueryPipeline)

### 功能描述:

查询任务队列设置信息和任务统计数

### 请求方式:

GET

### 请求参数:

参数名	类型	是否必需	备注
PipelineName	string	是	队列名, 默认: usual

### 返回参数:

参数名	类型	是否必需	备注
ErrNum	int	是	错误码
ErrMsg	string	是	错误描述
Quota	int	是	队列配额数
State	string	是	Active: 激活, Paused: 暂停
RegularStart	string	是	每
CreateTime	string	是	队列创建时间
UpdateTime	string	是	队列更新时间
TaskStatus	json array	是	队列任务数统计
Total	int	是	总任务数
Running	int	是	正在执行的任务数
Pending	int	是	等待的任务数
Success	int	是	执行成功的任务数
Failed	int	是	执行失败的任务数

### 请求示例:

```
GET /?Action=QueryPipeline
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170519T043000Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=accept%3Bhost
&X-Amz-Signature=xxxxx
&PipelineName=usual HTTP/1.1
Host: kvs.cn-beijing-6.api.ksyun.com
Accept: application/json
Content-Type: application/json;charset=utf-8
Content-Length: %LEN
```

## 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",
  "Quota":20,
  "Description": "low priority pipeline",
  "State": "Active",
  "RegularStart":"01:00:00",
  "RegularDuration":6400,
  "TaskStatus":{
    "Total":10,
    "Running":5,
    "Pending":5,
    "Success":0,
    "Failed":0
  }
}

```

## 创建任务接口(CreateTask)

### 接口描述

根据已设置的模板创建媒体转码任务

### 请求方式：

POST

### 请求参数：

参数名	类型	是否必须	描述
Preset	string	是	模板名称
SrcInfo	json array	是	参考下面srcInfo表格
DstBucket	string	是	输出目标Bucket
DstDir	string	否	截图上传目录，avsample类型专属参数，对其他任务类型不起作用
DstObjectKey	string	是	目录及文件名，例mydirectory/myfilename.mp4
DstAcl	string	否	转码视频访问权限设置，参数：private、public-read 默认：public-read
IsTop	int	否	是否置顶 1-是 0-否
CbMethod	string	否	回调用户请求方法
CbUrl	string	否	回调用户地址
ExtParam	string	否	附加参数。注意，此值是一个字符串，是由json序列化而成。下方有示例。

#### srcInfo参数说明：

参数名	类型	是否必须	描述
path	string	是	源文件在ks3的相对路径
index	int	否	多文件拼接场景使用，注意第一个文件序号为 0，填1导致转码失败
type	string	否	源文件类型（默认为video，方便日后扩展）

#### ExtParam 参数说明

用来在创建任务时传递不适用于放入模板里的字段，每种任务类型，extParam内容略有不同，如下：

```

avtrans

"ExtParam":{"Callback":{"progress_cburl":"xxx","progress_cbmethod":"xxx","progress_interval":2},"ss":"xxx","duration":"xxx","segment_filename":"xxx","vol_percent":100,"vol_dB":10,"mulHlsRate":1,"masterSrcIndex":0,"passParams":{"max_muxing_queue_size":"1000000"}}

avsnapshot

"ExtParam":{"ss":"xxx"}

avsample

"ExtParam":{"ss":"xxx","duration":"xxx","pattern":"xxx"}

avinfo

"ExtParam":{" //avinfo暂时没有ExtParam，所以可以选择不传，或传空字符串

```

表格中的参数ss与duration支持三种时间格式，选任其一即可：

```

"55"           : 55 seconds
"12:03:45"    : 12 hours, 03 minutes and 45 seconds

```

"23.189" : 23.189 seconds

avsample扩展参数（采样截图）：

名称	类型	必填项	示例值	描述
ss	string	否	"2"	截图的开始时间点
segfile_acl	string	否	"private"	private、public-read 默认：public-read
duration	string	否	"23"	指定截图持续时间长度

avsnapshot扩展参数（定点截图）：

名称	类型	必填项	示例值	描述
ss	string	否	"2"	截图的开始时间点
segfile_acl	string	否	"private"	private、public-read 默认：public-read
pattern	string	否	"test-%5d"	指定avsample各张截图名字格式，如 xxx-%05d.jpg。含义同上。如果不设，会默认用"snapshot-%5d"

avtrans扩展参数（转码）：

名称	类型	必填项	示例值	描述
ss	string	否	"2"	指定转码的开始时间点
segfile_acl	string	否	"private"	private、public-read 默认：public-read
duration	string	否	"23"	指定转码持续时间长度
segment_filename	string	否	"test-%03d.ts"	切片文件的名字规则，格式如右 xxxx-%03d.ts，%03d表示占位宽度为3，从000开始增长
seg_duration	int	否	5	指定hls或dash切片大小，单位s
masterSrcIndex	int	否	2	拼接时才需要设置，表示拼接时主文件序号，序号和SrcInfo里的index对应；其作用为指定转码时以哪个源文件作为决策依据；默认为0，即第一个源文件
vol_percent	int	否	233	根据百分比调整音量大小，负值为减小，正值为增加。取值范围：-99至700 单位：%
265qua	string	否	"output264"	265同画质转码必填参数，可选值 "output264":被参照的264转码；"output265":同画质的265转码。
use_template	int	否	0	按照模板切片(dash转码时有效) 0-关闭, 1-开启, 默认: 1
use_timeline	int	否	1	设置切片模板为时间模板(dash转码时有效) 0-关闭 1-开启, 默认: 1
init_segment_name	string	否	"test"	设置切片初始命名模板前缀(dash转码时有效), 注意不要包含特殊字符, 如不设置default= init-stream\$RepresentationID\$.m4s, 如果设置前缀为test, 那么初始模板全名为test-0.m4s 和 test-1.m4s
mulHlsRate	int	否	1	在模板配置mulrate=1情况下有效, 需要输出hls多码率文件 0-关闭 1-开启
mulHlsFileName	string	否	"test"	在mulHlsRate=1情况下, hls多码率文件名, 路径与mpd文件一致, 若不设置该参数则按照mpd文件名定义, 如xxx.m3u8
media_segment_name	string	否	"test"	设置切片文件名模板前缀(dash转码时有效), 注意不要包含特殊字符, 如不设置default= chunk-stream\$RepresentationID\$-\$Number%05d\$.m4s, 如果前缀为media, 那么切片文件名模板全名为media-0-00001.m4s 和 media-1-00001.m4s等
hlsKey	string	否		需要使用hls加密功能时必填; base64_urisafe编码后的密钥, 支持数字、字母、下划线
hlsKeyUrl	string	否		需要使用hls加密功能时必填; 指定了密钥放置的url, 经过base64_urisafe编码, 这是生成m3u8 Playlist会使用到的, 支持数字、字母、下划线

Callback:

名称	类型	必填项	示例值	描述
progress_cbmethod	string	否	"POST"	转码进度回调用户请求方法, 默认POST, 当前只支持 POST, 用户不需要设置
progress_cburl	string	否	"http://10.89.**.***:8899"	转码进度回调用户地址
progress_interval	string	否	"2"	指定转码进度上报时间间隔, 单位:min 默认5分钟
timeout	int	否	30	超时时间 单位:s
interval	int	否	3	重试间隔时间 单位:s
retrytimes	int	否	3	重试次数

subtitles:

名称	类型	必填项	示例值	描述
subtitlesUrl	string	否	"http://ks3-cn-beijing-internal.ksyun.com/xx/xxx/xxx.ass"	subtitlesUrl : 字幕存在在ks3内网地址, 支持格式:ass/srt 如:http://ks3-cn-beijing-internal.ksyun.com/xxx/xxx.ass

subtitles参数说明:

{"ExtParam":{"\`subtitles\`":[{"subtitlesUrl":`http://ks3-cn-beijing-internal.ksyun.com/xxx/xxx.ass`}]}"

返回参数

参数名	类型	是否必须	描述
ErrNum	int	是	错误码。0: 成功; 非0: 失败
ErrMsg	string	是	错误描述
TaskID	string	否	任务ID

请求示例

```
POST /?Action=CreateTask
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170519T081513Z
&X-Amz-Credential=xxxx
&X-Amz-SignedHeaders=xxxx
&X-Amz-Signature=xxxx HTTP/1.1
Host: kvs.cn-beijing-6.api.ksyun.com
Accept: application/json
{
```

```

"Preset": "mypreset",
"Pipeline": "xxxx",
"SrcInfo": [
  {
    "path": "/mysrcbucket/myobject/mykey.mp4",
    "type": "video",
    "index": 0
  }
],
"DstBucket": "mydstbucket",
"DstObjectKey": "myobject/mykey.mp4",
"DstDir": "",
"DstAcl": "public-read",
"CbMethod": "POST",
"CbUrl": "xxxxx",
"ExtParam": "{}"
}

```

## 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
{
  "TaskID": "40d309d3b2bf373cd3f08e5b5e1bddf720160816",
  "ErrNum": 0,
  "ErrMsg": "success"
}

```

## 模板组合

### 拼接功能

音视频拼接功能需要设置多个源文件为前提（创建任务时指定），且存在以下限制：

- 不转码拼接，速度快，但限制也多（暂不提供）：

不同编码方案的音频，比如libmp3lame, libfaac, 音频不能拼接；

视频拼接，编码方案相同，分辨率不同，码率相同，帧率相同，可以拼接成功，但播放器不支持；

视频拼接，编码方案相同，分辨率相同，码率不同，帧率相同，拼接成功之后码率取平均值；

视频拼接，编码方案相同，分辨率相同，码率相同，帧率不同，拼接成功，但播放器不支持；

视频拼接，编码方案不同，分辨率不同，码率相同，帧率相同，有错误提示“Could not concat between 1280x720and 1920x1080”；

- 转码拼接，用avtrans：

视频间的音视频轨道数不同，拼接失败；

纯音频拼接，拼接失败；

目前支持纯视频拼接和A,V轨道都为1的音视频拼接；

## 创建流式任务接口(CreateFlowTask)

### 功能描述：

根据已设置的模板创建流式任务

### 请求方式：

POST

### 请求参数：

参数名	类型	是否必须	描述
FlowData	json array	是	流式任务信息，json格式
IsTop	int	否	是否置顶 1-是 0-否
CbMethod	string	是	回调用户请求方法
CbUrl	string	是	回调用户地址

#### FlowData参数说明：

参数名	类型	是否必须	描述
Preset	string	是	模板名称
SrcInfo	json array	是	参考下面srcInfo表格

DstBucket	string	是	输出目标Bucket
DstDir	string	否	截图上传目录，avsample类型专属参数，对其他任务类型不起作用
DstObjectKey	string	否	输出目标文件名, 若为切片任务, 表示输出的切片列表名字
DstAcl	string	否	转码视频访问权限设置, 参数: private、public-read 默认: public-read
ExtParam	string	否	附加参数, json格式

**SrcInfo参数说明:**

参数名	类型	是否必须	描述
path	string	是	源文件在ks3的相对路径
index	int	是	顺序, 对拼接等功能有效
type	string	否	源文件类型 (默认为video, 方便日后扩展)

**ExtParam 参数说明:**

用来在创建任务时传递不适用于放入模板里的字段, 每种任务类型, extParam内容略有不同, 参见《创建任务接口》

**返回参数:**

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

**请求示例:**

```
POST /?Action=CreateFlowTask
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170519T081513Z
&X-Amz-Credential=xxxx
&X-Amz-SignedHeaders=accept%3Bhost
&X-Amz-Signature=xxxx HTTP/1.1
Host: kvs.cn-beijing-6.api.ksyun.com
Accept: application/json
{
  "FlowData":[
    {
      "Preset":"xxxx",
      "SrcInfo":[
        {
          "path":"/xxxx/xxxx.mp4",
          "type":"video",
          "index":0
        }
      ],
      "DstBucket":"xxxx",
      "DstObjectKey":"xxxx/xxxx.mp4",
      "DstDir":"",
      "DstAcl":"public-read",
      "ExtParam":""
    },
    {
      "Preset":"xxxx",
      "SrcInfo":[
        {
          "path":"/xxxx/xxxx.mp4",
          "type":"video",
          "index":0
        }
      ],
      "DstBucket":"xxxx",
      "DstObjectKey":"xxxx/xxxx.mp4",
      "DstDir":"",
      "DstAcl":"public-read",
      "ExtParam":""
    }
  ],
  "Pipeline":"usual",
  "IsTop":0,

```

```
    "CbMethod": "POST",
    "CbUrl": "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

{
  "TaskID": "6ba8e05f80831902f6315c0265e8e00a20170815",
  "ErrNum": 0,
  "ErrMsg": "success"
}
```

## 删除任务接口 (DelTaskByTaskID)

### 功能描述:

根据任务ID删除已经创建的任务

### 请求方式:

GET

### 请求参数:

参数名	类型	是否必须	描述
TaskID	string	是	任务ID

### 返回参数:

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

### 请求示例:

```
GET /?Action=DelTaskByTaskID
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170519T082122Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx
&TaskID=40d309d3b2bf373cd3f08e5b5e1bddf720160816 HTTP/1.1
Host: kvs.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"
}
```

## 置顶任务接口 (TopTaskByTaskID)

### 功能描述:

根据任务ID置顶已经创建的任务

### 请求方式:

GET

**请求参数：**

参数名	类型	是否必须	描述
TaskID	string	是	任务ID

**返回参数：**

参数名	类型	是否必须	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：**

```
GET /?Action=TopTaskByTaskID
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170519T094959Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx
&TaskID=40d309d3b2bf373cd3f08e5b5e1bddf720160816 HTTP/1.1
Host: kvs.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"
}
```

## 查询任务列表接口(GetTaskList)

**功能描述：**

根据时间、任务状态、出错码等范围查询已创建的任务列表

注意：可以查询最近30天的任务信息

**请求方式：**

GET

**请求参数：**

参数名	类型	是否必须	描述
StartDate	int	否	开始时间，默认为当前月的第一天；格式：20160919
EndDate	int	否	截止时间，默认为开始时间加30天；若大于当前时间，则默认为当前时间；格式：20160930
Marker	int	否	请求起始游标，默认为0
Limit	int	否	单次请求的记录数，默认为100，最大值为100
ErrorCode	string	否	出错码限制，若要查询多个错误码，则用逗号隔开，形如1001,1002
TaskStatus	string	否	任务状态限制(init,exec,succ, fail)，若要查询多个状态则用逗号隔开，形如init,exec
StartTime	int	否	查询的起始时间戳，若设置则StartDate失效
EndTime	int	否	查询的截止时间戳，若设置则EndDate失效

**返回参数：**

参数名	类型	是否必须	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
StartDate	int	是	开始时间
EndDate	int	是	截止时间
Marker	int	是	请求起始游标，默认为0
Count	int	是	本次返回记录条数
Total	int	是	该时间区间内总记录条数
TaskInfo	json array	是	任务列表
TaskID	string	是	任务ID
IsFlow	int	是	是否为流式任务，1-是，0-否
FlowData	json array	否	流式任务信息
Preset	string	是	模板名
SrcInfo	json array	是	源文件信息
DstBucket	string	是	目标Bucket
DstDir	string	是	目标目录



DstObjectKey	string	是	目标ObjectKey
DstAcl	string	是	目标文件访问类型
IsTop	int	是	是否被置顶
CbUrl	string	是	任务回调地址
CbMethod	string	是	回调方式
ExtParam	string	是	扩展参数
TaskStatus	string	是	单个任务执行状态
ErrorCode	string	是	出错码
ErrorMsg	string	是	出错描述
CreateTime	string	是	任务创建时间
path	string	是	源文件路径(KS3)
index	string	否	任务拼接顺序
type	string	是	文件类型

### 请求示例:

```
GET /?Action=GetTaskList
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170519T085226Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxx HTTP/1.1
Host: kvs.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
{
  "StartDate": "20160815",
  "EndDate": "20160816",
  "Marker": 0,
  "Count": 2,
  "Total": 6,
  "TaskInfo": [
    [
      {
        "TaskID": "65909f7144cfc2de10cf656989eca37520160815",
        "IsFlow": 0,
        "Preset": "preset_avop1",
        "SrcInfo": [
          [
            {
              "path": "/wangshuai9/ksyun.flv",
              "index": 0,
              "type": "video"
            }
          ]
        ],
        "DstBucket": "wangshuai9",
        "DstDir": "",
        "DstObjectKey": "ksyun_1.flv",
        "DstAcl": "public-read",
        "IsTop": 0,
        "CbUrl": "http://www.ksyun.com",
        "CbMethod": "callback",
        "ExtParam": "",
        "TaskStatus": "succ",
        "ErrorCode": 0,
        "ErrorMsg": "",
        "CreateTime": "2016-08-15 16:24:12"
      },
      {
        "TaskID": "40d309d3b2bf373cd3f08e5b5e1bddf720160816",
        "IsFlow": 1,
```

```

"FlowData":[
  (
    "Preset":"preset_avop2",
    "SrcInfo":[
      (
        "path":"/wangshuai9/ksyun2.flv",
        "index":0,
        "type":"video"
      )
    ],
    "DstBucket":"wangshuai9",
    "DstDir":"",
    "DstObjectKey":"ksyun_2.flv",
    "DstAcl":"public-read",
    "ExtParam":""
  ),
  (
    "Preset":"preset_avop3",
    "SrcInfo":[
      (
        "path":"/wangshuai9/ksyun3.flv",
        "index":0,
        "type":"video"
      )
    ],
    "DstBucket":"wangshuai9",
    "DstDir":"",
    "DstObjectKey":"ksyun_3.flv",
    "DstAcl":"public-read",
    "ExtParam":""
  )
],
"IsTop":"1",
"CbUrl":"http://www.ksyun.com",
"CbMethod":"callback",
"ExtParam":"",
"TaskStatus":"succ",
"ErrorCode":"0",
"ErrMsg":"",
"CreateTime":"2016-08-16 11:13:29"
}
],
"ErrNum":0,
"ErrMsg":"success"
}

```

## 查询任务详情接口(GetTaskByTaskID)

### 功能描述：

根据任务ID查询已创建的任务。因为点播转码服务是异步的，所以请您在创建任务3~5秒后再调用此接口查询该任务

### 请求方式：

GET

### 请求参数：

参数名	类型	是否必须	描述
TaskID	string	是	任务ID

### 返回参数：

参数名	类型	是否必须	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
TaskInfo	json object	是	任务详情信息
TaskID	string	是	任务ID
IsFlow	int	是	是否为流式任务，1-是，0-否

FlowData	json array	否	流式任务信息
Preset	string	是	模板名
SrcInfo	json array	是	源文件信息
DstBucket	string	是	目标Bucket
DstDir	string	是	目标目录
DstObjectKey	string	是	目标ObjectKey
DstAcl	string	是	目标文件访问类型
IsTop	int	是	是否被置顶
CbUrl	string	是	任务回调地址
CbMethod	string	是	回调方式
ExtParam	string	是	扩展参数
TaskStatus	string	是	单个任务执行状态
ErrorCode	string	是	出错码
ErrorMsg	string	是	出错描述
CreateTime	string	是	任务创建时间
path	string	是	源文件路径(KS3)
index	string	否	任务拼接顺序
type	string	是	文件类型

### 请求示例：

```
GET /?Action=GetTaskByTaskID
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20170519T082842Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx
&TaskID=40d309d3b2bf373cd3f08e5b5e1bddf720160816 HTTP/1.1
Host: kvs.cn-beijing-6.api.ksyun.com
Accept: application/json
```

### 普通任务返回示例：

```
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
{
  "TaskInfo": {
    "TaskID": "65909f7144cfe2de10cf656989cca37520160815",
    "IsFlow": 0,
    "Preset": "preset_avop1",
    "SrcInfo": [
      {
        "path": "/wangshuai9/ksyun.flv",
        "index": 0,
        "type": "video"
      }
    ],
    "DstBucket": "wangshuai9",
    "DstDir": "",
    "DstObjectKey": "ksyun_1.flv",
    "DstAcl": "public-read",
    "IsTop": 0,
    "CbUrl": "http://www.ksyun.com",
    "CbMethod": "callback",
    "ExtParam": "",
    "TaskStatus": "succ",
    "ErrorCode": 0,
    "ErrorMsg": "",
    "CreateTime": "2016-08-15 16:24:12"
  },
  "ErrNum": 0,
  "ErrMsg": "success"
}
```

### 流式任务返回示例：

```
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

```
{
  "TaskInfo": {
    "TaskID": "40d309d3b2bf373cd3f08e5b5e1bddf720160816",
    "IsFlow": 1,
    "FlowData": [
      {
        "Preset": "preset_avop2",
        "SrcInfo": [
          {
            "path": "/wangshuai9/ksyun2.flv",
            "index": 0,
            "type": "video"
          }
        ],
        "DstBucket": "wangshuai9",
        "DstDir": "",
        "DstObjectKey": "ksyun_2.flv",
        "DstAcl": "public-read",
        "ExtParam": ""
      },
      {
        "Preset": "preset_avop3",
        "SrcInfo": [
          {
            "path": "/wangshuai9/ksyun3.flv",
            "index": 0,
            "type": "video"
          }
        ],
        "DstBucket": "wangshuai9",
        "DstDir": "",
        "DstObjectKey": "ksyun_3.flv",
        "DstAcl": "public-read",
        "ExtParam": ""
      }
    ],
    "IsTop": 1,
    "CbUrl": "http://www.ksyun.com",
    "CbMethod": "callback",
    "TaskStatus": "succ",
    "ErrorCode": 0,
    "ErrorMsg": "",
    "CreateTime": "2016-08-16 11:13:29"
  },
  "ErrNum": 0,
  "ErrMsg": "success"
}
```

## 查询META信息接口 (GetTaskMetaInfo)

### 功能描述:

根据任务ID查询单个任务META信息，或根据时间范围查询任务META信息列表，转码模板类型必须为avinfo才可通过GetTaskMetaInfo查询，主要用于查询转码后文件的meta信息

注意：可以查询最近90天的任务数据

### 请求方式:

GET

### 请求参数:

参数名	类型	是否必须	描述
-----	----	------	----

TaskID	string	否	任务ID, 若使用该参数查询, 则其他参数忽略
StartDate	int	否	开始时间, 默认为当前月的第一天; 格式: 20160919
EndDate	int	否	截至时间, 默认为开始时间加30天; 若大于当前时间, 则默认为当前时间; 格式: 20160930
Marker	int	否	请求起始游标, 默认为0
Limit	int	否	单次请求的记录数, 默认为100, 最大值为100

## 返回参数:

参数名	类型	是否必须	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
Count	int	是	本次返回记录条数
Total	int	是	该时间区间内总记录条数
MetaList	json array	是	META信息列表

## MetaList说明

参数名	类型	是否必须	描述
TaskID	string	是	任务ID
MetaInfo	string	是	源文件信息, 是序列化的json串。定义详见“用户回调说明->Details->items->metainfo字段”

## 请求示例:

```
GET /?Action=GetTaskMetaInfo
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170519T085908Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx
&StartDate=20160918
&EndDate=20160919
&Marker=0
&Limit=1000 HTTP/1.1
Host: kvs.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
{
  "StartDate": "20160929",
  "EndDate": "20160930",
  "Marker": 0,
  "Count": 1,
  "Total": 6,
  "MetaList": [
    {
      "TaskID": "c3697fdd9c4c1240c0e5f29b7a63cba820160929",
      "MetaInfo": {
        "streams": [
          {
            "codec_name": "aac",
            "codec_long_name": "AAC (Advanced Audio Coding)",
            "profile": "LC",
            "codec_type": "audio",
            "codec_time_base": "1/48000",
            "codec_tag_string": "mp4a",
            "codec_tag": "0x6134706d",
            "r_frame_rate": "0/0",
            "avg_frame_rate": "0/0",
            "time_base": "1/48000",
            "start_time": "0.000000",
            "duration_ts": 61673448,
            "duration": "1284.863500",
            "bit_rate": "47325",
```

```

        "nb_frames": "60228",
        "sample_fmt": "fltp",
        "sample_rate": "48000",
        "channels": 2,
        "channel_layout": "stereo"
    },
    {
        "index": 1,
        "codec_name": "h264",
        "codec_long_name": "H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10",
        "profile": "High",
        "codec_type": "video",
        "codec_time_base": "642153/15575000",
        "codec_tag_string": "avc1",
        "codec_tag": "0x31637661",
        "r_frame_rate": "13/1",
        "avg_frame_rate": "7787500/642153",
        "time_base": "1/16000",
        "start_pts": 10176,
        "start_time": "0.636000",
        "duration_ts": 20548896,
        "duration": "1284.306000",
        "bit_rate": "781679",
        "nb_frames": "15575",
        "width": 500,
        "height": 600,
        "coded_width": 512,
        "coded_height": 608,
        "rotate": "90", //当rotate有值时，返回值中的分辨率是rotate值旋转后的。当rotate为空时，分辨率为原始分辨率。
        "has_b_frames": 2,
        "sample_aspect_ratio": "0:1",
        "display_aspect_ratio": "0:1",
        "pix_fmt": "yuv420p",
        "level": 22,
        "chroma_location": "left",
        "refs": 1,
        "is_avc": "true",
        "bits_per_raw_sample": "8"
    }
],
"format": {
    "nb_streams": 2,
    "format_name": "mov, mp4, m4a, 3gp, 3g2, mj2",
    "format_long_name": "QuickTime / MOV",
    "duration": "1284.864000",
    "size": "133778683",
    "bit_rate": "832951",
    "probe_score": 100
}
},
"CreateTime": "2016-09-29 18:20:29"
}
],
"ErrNum": 0,
"ErrMsg": "success"
}

```

## 同步获取META信息接口（FetchMetaInfo）

### 接口描述

同步获取文件meta信息的接口，用于获取源文件的meta信息，仅支持mp4、flv、m3u8、mp3、dash格式

### 请求方式

POST

## 请求参数：

参数名	类型	是否必须	描述
SrcPath	string	是	源文件在KS3的存储路径

## 返回参数

参数名	类型	是否必须	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
MetaInfo	json string	是	meta信息，与回调结果中的meta信息结构一致，参数说明详见 <a href="#">转码结果回调</a>

## 请求示例

```
POST /?Action=FetchMetaInfo
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170519T060855Z
&X-Amz-Credential=xxxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx HTTP/1.1
HOST: kvs.cn-beijing-6.api.ksyun.com
Accept: application/json
Content-Type: application/json;charset=utf-8
Content-Length: %LEN
{
  "SrcPath": "/wangshuai9/ksyun1.flv"
}
```

## 返回示例

```
HTTP/1.1 200 OK
Server: ksyun-open-platform
Date: Sun, 7 Jan 2018 12:28:06 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
{
  "MetaInfo": {
    "streams": [
      {
        "codec_name": "h264",
        "codec_long_name": "H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10",
        "profile": "Baseline",
        "codec_type": "video",
        "codec_time_base": "103931/5018000",
        "codec_tag_string": "avc1",
        "codec_tag": "0x31637661",
        "r_frame_rate": "97/4",
        "avg_frame_rate": "2509000/103931",
        "time_base": "1/90000",
        "start_time": "0.000000",
        "duration_ts": "9353790",
        "duration": "103.931000",
        "bit_rate": "1969387",
        "nb_frames": "2509",
        "width": 1280,
        "height": 720,
        "coded_width": 1280,
        "coded_height": 720,
        "rotate": "90", //当rotate有值时，返回值中的分辨率是rotate值旋转后的。当rotate为空时，分辨率为原始分辨率。
        "sample_aspect_ratio": "0:1",
        "display_aspect_ratio": "0:1",
        "pix_fmt": "yuv420p",
        "level": 41,
        "chroma_location": "left",
        "refs": 1,
      }
    ]
  }
}
```

```
    "is_avc": "true",
    "bits_per_raw_sample": "8"
  },
  {
    "index": 1,
    "codec_name": "mp3",
    "codec_long_name": "MP3 (MPEG audio layer 3)",
    "codec_type": "audio",
    "codec_time_base": "1/44100",
    "codec_tag_string": "mp4a",
    "codec_tag": "0x6134706d",
    "r_frame_rate": "0/0",
    "avg_frame_rate": "0/0",
    "time_base": "1/44100",
    "start_time": "0.000000",
    "duration_ts": 4643712,
    "duration": "105.299592",
    "bit_rate": "63999",
    "max_bit_rate": "64000",
    "nb_frames": "4031",
    "sample_fmt": "s16p",
    "sample_rate": "44100",
    "channels": 2,
    "channel_layout": "stereo"
  }
],
"format": {
  "nb_streams": 2,
  "format_name": "mov, mp4, m4a, 3gp, 3g2, mj2",
  "format_long_name": "QuickTime / MOV",
  "duration": "105.300000",
  "size": "26499154",
  "bit_rate": "2013231",
  "probe_score": 100
}
},
"ErrNum": 0,
"ErrMsg": "success"
}
```