

目录

目录	1
公共错误	2
转码结果回调用户	5
POST 回调接口	5
GET 回调接口	16
提取音频波形图	16
接口描述	16
请求方式:	16
请求参数:	16
返回参数	17
请求示例	17
返回示例	17

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	jsonp 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 twoker 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 timesta
          mp 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
dstpath	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thumbnailInfo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中的存路路径, 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
    "segmenlisttype": "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
"items": [
  {
    "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":"0c33c9c372ff87fe197cb0dcf236e18d20161223",
  "Status":3,
  "Type":"avinfo",
  // 为兼容ks3老客户，新用户接入时，不带有itmes字段，请忽略
  "itmes":[
    {
      "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",  
      "nb_frames": "15575",  
    }  
  ]  
}
```


字段名	类型	是否必有	描述
dstpath	string	是	任务在流式中的执行顺序
metainfo	string	是	任务类型
snapshotNum	int	是	子任务执行结果: 3为成功, 4为失败
outMediainfo	string	否	任务错误码
segmentsinfo	string	否	任务错误信息
thumbnailsinfo	string	否	输出文件的bucket
segmentlisttype	string	是	输出信息详情

备注: 对metainfo, outMediainfo等的说明, 见前文单独任务中该字段详情。

- 两个avtrans类型的任务流式执行, 结束后回调内容格式示例

```
{
  //任务id type:string
  "TaskID": "92e3745751aee5662f9f06ff3f03076920160914",
  //处理中:2 成功:3 失败: 4 type:int
  "Status": 3,
  //视频云离线转码返回信息
  "Details": [
    {
      "Order": 0,
      "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":"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

提取音频波形图

接口描述

通过模板对文件提取音频波形元数据创建波形图任务

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，固定传:wave，注意小写
SrcInfo	json array	是	参考下面srcInfo表格
DstBucket	string	是	输出目标Bucket
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
CbUrl	string	否	回调用户地址

points int 是 指定多少个打点数据。不允许小于0，取值范围:1~4095

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

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=CreateWaveTask
&Version=2017-01-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20170519T081513Z
&X-Amz-Credential=xxxx
&X-Amz-SignedHeaders=xxxxx
&X-Amz-Signature=xxxxx HTTP/1.1
Host: kvs.cn-beijing-6.api.ksyun.com
Accept: application/json
{
  "Preset": "wave",
  "SrcInfo": {
    "path": "/mysrcbucket/myobject/mykey.mp4",
  },
  "DstBucket": "mydstbucket",
  "DstObjectKey": "myobject/mykey.mp4",
  "DstAcl": "public-read",
  "CbUrl": "xxxxx",
  "points": 1000
}
```

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
{
  \t"TaskID": "40d309d3b2bf373cd3f08e5b5e1bddf720160816",
  \t"ErrNum": 0,
  \t"ErrMsg": "success"
}
```