

## 目录

目录	1
签名机制	4
请求结构	4
1. 服务地址	4
2. 通信协议	4
3. 请求方法	4
4. 请求参数	4
5. 字符编码	4
返回结果	4
调用成功	4
调用失败	4
公共错误	4
特殊错误	5
接口调用示例	5
查询主播推拉流信息接口 (ListRealtimeStreamsInfo)	8
接口说明	8
请求参数	8
返回参数	8
请求示例	9
返回示例	9
错误码	9
查询推流信息接口 (ListRealtimePubStreamsInfo)	9
接口说明	9
请求参数	10
返回参数	10
请求示例	10
返回示例	10
错误码	10
查询流历史信息接口 (ListHistoryPubStreamsInfo)	10
接口说明	10
请求参数	10
返回参数	11
断流原因错误码及英文描述	11
请求示例	11
返回示例	11
错误码	11
查询流历史错误信息接口 (ListHistoryPubStreamsErrInfo)	11
接口说明	11
请求参数	11
返回参数	12
断流原因错误码及英文描述	12
请求示例	12
返回示例	12
错误码	12
查询主播流时长接口 (ListStreamDurations )	12
接口说明	12
请求参数	13
返回参数	13
请求示例	13
返回示例	13

错误码	13
禁止单路直播流推送 (ForbidStream)	14
接口描述	14
请求参数	14
返回参数	14
请求示例	14
返回示例	14
错误码	14
恢复单路直播流推送 (ResumeStream)	14
接口描述	14
备注	14
请求参数	14
返回参数	14
请求示例	14
返回示例	14
错误码	15
查询黑名单列表 (GetBlacklist)	15
接口描述	15
备注	15
请求参数	15
返回参数	15
请求示例	15
返回示例	15
错误码	15
检查流是否在黑名单内 (CheckBlacklist)	15
接口描述	15
备注	15
请求参数	15
返回参数	16
请求示例	16
返回示例	16
错误码	16
视频云直播SDK for java使用指南	16
云直播 Java SDK 概述 :	16
安装java sdk	16
1.git 安装	16
2.mvn 中央仓库安装	16
3.工程使用sdk录	16
新建maven项目,进入到pom文件下,进行如下配置	16
4.通过文件配置及管理密钥	16
本地文件配置:	16
该文件包含下述内容:	16
如不通过本地文件读入ak和sk信息,则需要执行的类文件中增加以下代码:	16
调用方式	16
在调用接口前,先初始化 KSCOFFJsonClient,如下用例:	16
接口目录说明	16
视频云直播SDK for php 使用指南	16
云直播 PHP SDK 概述	17
1.初始化	17
1.1 ak/sk 配置	17
1.2 composer安装	17
1.3 调试demo	17

1.4 调用示例	17
2.Post 方法调用示例	17
3.Get 方法调用示例	17
视频云直播SDK for Python 使用指南	18
视频云直播 Python SDK 概述	18
1.安装python sdk	18
1.1 git 安装	18
1.2 pip 安装	18
2.配置管理密钥	18
2.1 通过文件配置密钥	18
2.2 在代码中配置密钥	18
3.Post 方法调用示例	18
4.Get 方法调用示例	18

## 签名机制

金山云OpenAPI目前支持以下签名算法

(1) AWS签名算法版本4，具体可以参考[AWS文档](#)

- [公共参数](#)
- [签名算法](#)

注意：

- Region取值：cn-beijing-6
- Service取值：kls
- KLS的Region及服务固定填以上值即可，如您需要调用CDN的接口，Region及服务会稍有不同，详情请见[CDN鉴权方式](#)

## 请求结构

客户调用云直播服务的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。

云直播openAPI的请求结构组成如下：

### 1. 服务地址

正式接入地址为：kls.api.kyyun.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

云直播的查询类API支持GET请求，操作类API支持POST请求。

注意

- 不能混合使用两种请求方式。如果使用 GET 方式，参数均从 querystring 取得；如果使用 POST 方式，参数均从请求Body中取得（其中Action、Version在querystring中取得），详细调用方式请参考请求示例。

### 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[签名机制](#)；接口请求参数是每个接口所特有的，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。

## 返回结果

云直播的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）。

### 调用成功

详见各接口示例

### 调用失败

调用接口失败，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)；与调用成功一样还包含请求ID(RequestId)，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."
  }
}
```

### 公共错误

错误代码(Code)	错误消息(Message)	HTTP 状态码	中文描述（语义）
MissingParameter	The input parameter \"%s\" that is mandatory for processing this request is not supplied.	400	必填参数没有填，请您检查调用时是否填写了此参数
InvalidParameter	The input parameter \"%s\" is not valid.	400	参数名称错误，请您检查是否按接口使用文档填写请求参数
InvalidParameterValue	The input parameter value \"%s\" is not valid.	400	参数value值错误，请您检查是否按使用文档填写请求参数
InvalidParameterValue	The input parameter value type error.	400	参数value值类型错误，请您参照接口使用文档更新参数，再重新发起请求
LimitExceeded	Request was rejected because the request speed of this openAPI is beyond the current flow control limit.	409	请求被拒绝，因该openAPI接口访问速度已达到流控上限
StreamNotExist	The Live stream does not exist.	404	目标直播流不存在，请确认主播是否已经开播，再重新发起请求
UnknownError	The request processing has failed due to some unknown error.	400	未知错误。请您重试该操作，若再出现此问题请联系售后排查
InvalidParameterValue	Uniquename, pubdomain and app do not match.	400	UniqueName、Pubdomain、App不匹配，请您检查匹配关系后，再重新发起请求
MissingKs3Bucket	There is no Ks3Bucket for storing files.	403	没有为截图文件选择Ks3Bucket，请先选择Ks3Bucket，再重新发起请求
MissingKs3Bucket	Ks3Bucket does not exist.	404	所选择的Ks3Bucket不存在，请选择一个已经创建好的Ks3Bucket，再重新发起请求
ServiceUnavailable	The service is unavailable due to an internal exception.	500	系统内部错误。请您重试该操作，若再出现此问题请联系售后排查
UnavailableRegion	The region %s is unavailable for the openapi	400	此openapi不具备%s区域的服务
InaccessibleOpenapi	Without access to call Inner Openapi.	401	不具备访问inner openapi的权限
UnavailableAccountID	The Account ID is invalid.	401	账户id不合法，请调整您的id后再重新发起请求
ServiceTimeout	Internal Service is unavailable because of time out.	500	内部服务异常或超时暂不可用，请您重试该操作，若再出现此问题请联系售后排查
ErrorDryRun	Request would have succeeded, but DryRun flag is set.	412	请求本可成功，但由于设置DryRun标记未成功
InvalidAuthenticationCode	the policy deny to [%s] the resources[%s]	403	权限拒绝操作%s资源%s的行为
InvalidAuthenticationCode	lack of policy to [%s] the resources[%s]	403	权限不足，您缺少操作%s资源%s的权限

InvalidAuthenticationCode	Request was rejected because of lack of policy.	403	权限不足, 请确认您的相关权限
SignatureNotMatch	The Signature is not matched	400	签名不匹配, 请检查签名算法
InvalidAccesskey	Invalid Accesskey(%s)	400	非法秘钥(%s)
MissingAccesskey	Access Key is Missing	400	请求被拒绝, 因为没有秘钥
InvalidTimestamp	Invalid Timestamp(%s). Pls in 15mins	400	请求时间戳不合法(%s), 请保持请求时间戳在15分钟以内
InvalidActionOrVersion	The action %s or version %s is not valid for this web service	400	action 或 version 不合法, 请您调整后重新发起请求
ServiceUnavailable	Invalid service(%s)	404	请求被拒绝, 因其引用的服务(%s)不存在, 如确认没问题请联系售后排查
MissingSignature	The Request is missing Signature Parameter.	400	请求被拒绝, 因为没有合法的签名, 请您调整后重新发起请求
MissingTimestamp	The request is missing timestamp parameter.	400	请求timestamp不合法, 请您调整后重新发起请求
MissingParameter	The request is missing %s parameter.	400	请求没有合法%s, 请您调整后重新发起请求
UndefinedError	Undefine Internal Error. Pls Contact With Admin.	500	系统错误, 请您稍后重试该操作, 若再出现此问题请联系售后排查

### 特殊错误

金山云v1版本接口(“直播流信息查询”、“直播流封禁”目录里的全量接口)的HTTP状态码为200时, 代表服务端已经接收到了您的请求, 但此时仍可能出现调用失败的情况, 可能由于您的请求参数不符合接口的业务逻辑, 此时错误信息我们会在body体内发送给您, 请调整后重新发起调用请求, 形式如下:

#### json格式示例

```
{
  "Data": {
    "RetCode": 8201,
    "RetMsg": "stream is not on the blacklist"
  },
  "RequestId": "36d7f12e-2345-4e5e-a5f4-7hsjklff27eb"
}
```

错误码	RetCode	RetMsg	含义
1000	lack of parameters	缺少必须参数	
1002	invalid uniquenessname	域名空间错误	
1003	invalid app name	应用名错误	
1004	invalid stream name	流名错误	
1005	invalid preset name	转码模板名错误	
1200	task does not exist	任务不存在	
1201	task already exists	任务已经存在	
1202	failed to insert records	内部错误	
1203	failed to delete records	删除任务失败	
1204	failed to update records	更新任务失败	
3802	stream has been banned and no change forbid information	不支持在同一秒内重复封禁同一条流	
4000	invalid pubdomain	域名错误	
4001	invalid time	时间格式错误或查询的时间间隔超过限制	
4201	get result error	uid、uniquename、domain不匹配	
4999	parameters other error	参数错误, 请检查: 1. 参数名大小写是否正确 2. 参数value类型是否正确	
5000	the user does not exist	用户不存在	
6000	no files(m3u8) during the task	没有录像文件	
6010	splash started, but no online stream	短视频开始录制但流不在线上	
6400	failed to get the number of tasks	获取任务数量失败	
6401	recording number reach upper limit	任务数量超过上限	
6402	get stream record status error	获取流任务状态失败	
6403	stream is recording	该流正在录制	
6404	stream record has been stopped	已经停止过该流	
6405	RecID does not exist	RecID不存在	
6406	unique_name and accountId not match	域名空间不存在	
6701	get ej_jobName empty	查询定时任务信息为空	
6702	ej interface use or return fail	创建定时任务失败	
6703	start time error	开始时间错误	
6704	end time error	结束时间错误	
6705	file name or path error	文件名或路径不合法	
6706	HLS vod should be enable	HLS 不可用	
6707	Hls or mp4 Bucket error	Bucket 错误	
6708	Record Status error	任务状态不可取消	
6709	Recording time not be less than 1 minute	录制时间小于一分钟	
8200	failed to insert redis	系统内部错误	
8201	stream is not on the blacklist	目标流不在黑名单上	
8500	error nodeip	系统内部错误	
8501	kill stream fail	流封禁失败	
8502	stream is not exist	流不存在	
9014	PullProtocol error	拉流协议错误	
9015	Type error	返回信息类型错误	
9016	Over the speed limit	调用频率超过了接口限制	
1	other error	其他错误, 若您重试后仍不能解决, 请联系售后为您排查问题	

## 接口调用示例

建议正式使用接口前, 先用Postman进行调试, 本文以调用ListRealtimeStreamsInfo为例, 为您演示:

- [Postman下载链接](#)

- 打开Postman, 请求方法设置为Post, 在请求URL中填入 <http://kls.api.ksyun.com?Action=ListRealtimeStreamsInfo&Version=2017-01-01>
- 添加请求方法的Authorization

Type 类型选择AWS Signature  
Accesskey 和 SecretKey 分别输入自己的ak, sk  
AWS Region 输入cn-beijing-6  
Service Name 输入 kls

添加完成后的效果如下图所示, 点击右上角的Update Request:

example × + ... No Environment

example

POST http://kls.api.ksyun.com?Action=ListRealtimeStreamsInfo&Version=2017-01-01 Params

Authorization Headers (5) Body Pre-request Script Tests

Type AWS Signature Clear

AccessKey \*\*\*\*\*

SecretKey \*\*\*\*\*

AWS Region cn-beijing-6

Service Name kls

Save helper data to request

Response

Hit the Send button to get a response.

Do more with requests

点击完成后 Request Headers里面会有如下图的内容:

example × + ... No Environment

example

POST http://kls.api.ksyun.com?Action=ListRealtimeStreamsInfo&Version=2017-01-01 Params

Authorization Headers (5) Body Pre-request Script Tests

Key	Value	Description
<input checked="" type="checkbox"/> Content-Type	application/x-www-form-urlencoded	
<input checked="" type="checkbox"/> Host	kls.api.ksyun.com	
<input checked="" type="checkbox"/> Content-Length	70	
<input checked="" type="checkbox"/> X-Amz-Date	20171208T064330Z	
<input checked="" type="checkbox"/> Authorization		

New key Value Description

Response

Hit the Send button to get a response.

- 注意: 如果原来的Postman中Headers中有内容, 可能update Request 有时会不太奏效, 建议先将Headers中的所有全部清空, 再点击Update Request \*

3. 点击Body选项卡, 选择raw并将类型选择为Json, 然后如下图所示输入请求的body, 如果输入的参数错误或者缺少, 下方响应消息会提供失败原因, 成功返回success

调用成功如下图:

POST http://kls.api.ksyun.com?Action=ListRealtimeStreamsInfo&Version=2017-01-01

Authorization Headers (5) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```

1- {
2  "UniqueName": "your uniqueName",
3  "App": "your app"
4 }

```

Body Cookies Headers (5) Test Results Status:

Pretty Raw Preview JSON

```

1- {
2  "Data": {
3    "UniqueName": "your uniqueName",
4    "App": "your app",
5    "Count": 0,
6    "Bw": 0,
7    "OnlineUser": 0,
8    "RetCode": 0,
9    "RetMsg": "success",
10   "Streams": [
11     {
12       "Stream": "your stream",
13       "ClientIP": "192.168.1.1",
14       "Node": "your node",
15       "PubTime": "2023-04-28 10:00:00",
16       "FrameRate": 30,
17       "Curfps": 30,
18       "FLR": 0,
19       "Curbitrate": 1000,
20       "BwInAudIn": 1000,
21       "BwIn": 1000,
22       "BwReal": 1000,
23       "BwDiff": 1000,
24       "Width": 1920,
25       "Height": 1080

```

调用失败如下图:

POST http://kls.api.ksyun.com?Action=ListRealtimeStreamsInfo&Version=2017-01-01

Authorization Headers (5) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```

1- {
2  "UniqueName": "your uniqueName",
3  "App": "your app"
4 }

```

Body Cookies Headers (5) Test Results Status:

Pretty Raw Preview JSON

```

1- {
2  "Data": {
3    "UniqueName": "",
4    "App": "",
5    "Count": 0,
6    "Bw": 0,
7    "OnlineUser": 0,
8    "RetCode": 6406,
9    "RetMsg": "unique_name and accountId not match",
10   "Streams": []
11 },
12 "RequestId": "549189da-8d79-4cff-b132-6c3678bde840"
13 }

```

quanmin

AKSK鉴权失败 如下图:

example

POST http://kls.api.ksyun.com?Action=ListRealtimeStreamsInfo&Version=2017-01-01 Params

Authorization Headers (5) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary Text

```

1 {
2   "UniqueName": "your uniqueName",
3   "App": "your app"
4 }

```

Body Cookies Headers (5) Test Results Status: 400

Pretty Raw Preview XML

```

1 <ErrorResponse>
2   <RequestId>1bf04e2a-a0a1-4bda-af17-73cbc9726e10</RequestId>
3   <Error>
4     <Type>Sender</Type>
5     <Code>MissingAccesskey</Code>
6     <Message>Access Key is Missing</Message>
7   </Error>
8 </ErrorResponse>

```

## 查询主播推拉流信息接口 (ListRealtimeStreamsInfo)

### 接口说明

此接口汇总了主播的实时推流质量状态、主播热度和各码率观看热度

#### 1. 主播推流质量状态信息

- 推流列表：查询时间点的瞬时推流列表
- 质量信息：码率、帧率、丢帧率、客户端IP，CDN收流节点IP等

#### 2. 主播热度信息（注意：此接口不能用来查询推流触发转码的主播热度信息，即统计的在线人数和带宽不包括推流转码的转码流数据，如需全量查询带宽、人数请参考[GetLivePlayStatData](#)）

- 观众在线人数：即原流和各转码流观众在线人数的加和
- 观众拉流带宽：即原流和各转码流拉流带宽的加和

#### 3. 各码率观看热度（注意：此接口不能用来查询推流触发转码的转码流信息，即统计的流名列表、在线人数、拉流带宽不包括推流转码的转码流数据，如需全量查询带宽、人数请参考[GetLivePlayStatData](#)）

- 各码率拉流流名列表：包含原流名和转码后的流名
- 在线人数：返回原流/转码流的在线人数
- 拉流带宽：返回原流/转码流的带宽

#### 注意：

- 在线人数及带宽目前支持查询的地区：中国大陆
- 推流质量信息为查询时刻的瞬时信息，因数据统计汇总，拉流数据会有5-10min的延时
- 限制访问频率：50次/分钟
- 数据粒度：1min
- 单次返回数据最多10000条，如需增加数据量请联系售后评估
- 调用方式：HTTP POST

### 请求参数

参数名	类型	是否必须	描述
Action	String	Y	查询主播推拉流实时信息接口 (ListRealtimeStreamsInfo)
Version	String	Y	版本号，取值：2017-01-01
UniqueName	String	Y	域名空间
App	String	Y	应用名
Stream	String	N	流名，只支持输入单个流名
DomainIds	String	N	拉流域名ID，缺省为UniqueName下全部拉流域名，可输入需要查询的拉流域名ID，支持批量域名查询，多个域名ID用逗号（半角）分隔
PullProtocol	String	N	返回拉流信息协议类型，默认http+flv，http flv协议在线人数和带宽；rtmp，rtmp协议在线人数和带宽；hls：返回hls协议在线人数和带宽
Type	String	N	返回信息类型，缺省：返回推拉流信息的交集，push，只返回推流信息，pull：只返回拉流信息

### 返回参数

参数名	类型	描述
UniqueName	String	域名空间
App	String	应用名
Count	Int	在线主播数/人
Bw	Long	所有主播的观众拉流带宽加和/bps，如果输入流名查询，则查询单主播的观众拉流带宽
OnlineUser	Long	所有主播的在线观众数量，如果输入流名查询，则查询单主播的在线观众数量
RetCode	Int	响应码 0成功 其他失败
RetMsg	String	响应信息描述

Streams DataByStream[] 在线直播列表（不包含查询时间点在推流）

DataByStream

参数名	类型	描述
Stream	String	推流流名
ClientIP	String	推流客户端IP
Node	String	CDN收流节点IP
PubTime	Int	开始推流的UTC时间, 10位UTC时间戳
FrameRate	Int	视频理论帧率/fps
Curfps	Int	视频真实帧率/fps
FLR	Int	丢帧率/%=(1-帧率/原始帧率)*100%
Curbitrate	Double	视频码率/kbps
BwInAudio	Double	音频码率/kbps
BwIn	Double	理论码率/kbps=视频码率+音频码率
BwReal	Double	真实码率/kbps, 即推流传输码率
BwDiff	Double	码率差值/kbps=理论码率-真实码率
Width	Int	视频宽度/P
Height	Int	视频高度/P
SampleRate	Int	音频采样率/Hz
Vcodec	String	视频编码格式
Acodec	String	音频编码格式
OnlineUser	Long	主播在线人数/人, 原流名+所有转码流的在线人数
Bw	Long	主播拉流带宽/bps, 原流名+所有转码流的拉流带宽
Pulls	DataByPull[]	在线直播列表（不包含查询时间点）

DataByPull

参数名	类型	描述
Stream	String	拉流流名, 包括原流名拉流, 转码拉流流名
OnlineUser	Long	在线人数/人
Bw	Long	带宽/bps

## 请求示例

```
POST /?Action=ListRealtimeStreamsInfo&Version=2017-01-01 HTTP/1.1
Content-Type: application/json
X-Amz-Date: YOURDATE
Authorization: AWS4-HMAC-SHA256 Credential=YOURAK/YYYYMMDD/cn-beijing-6/kls/aws4_request, SignedHeaders=content-length;content-type;host;x-amz-date, Signature=SIG
Host: kls.api.ksyun.com
```

```
{
  "UniqueName": "xxx",
  "App": "xx"
}
```

## 返回示例

```
{
  "Data": [
    {
      "UniqueName": "xxxx",
      "App": "live",
      "Bw": 50000,
      "OnlineUser": 500,
      "Count": 1,
      "Streams": [
        {
          "Node": "xxxxx",
          "FLR": 52,
          "BwDiff": 555.5,
          "Vcodec": "h264",
          "Acodec": "mp3",
          "OnlineUser": 500,
          "Bw": 50000,
          "BwInAudio": 21043,
          "BwIn": 442409,
          "BwReal": 223,
          "BwDiff": 442186,
          "Width": 368,
          "Clientip": "114.255.44.132",
          "Curbitrate": 52666,
          "Curfps": 31,
          "FrameRate": 30,
          "Height": 614,
          "Pubtime": 1491033051,
          "SampleRate": 44100,
          "Stream": "s222",
          "Pulls": [
            {
              "Stream": "s222",
              "Bw": 20000,
              "OnlineUser": 200
            },
            {
              "Stream": "s222_1500",
              "Bw": 30000,
              "OnlineUser": 300
            }
          ]
        }
      ]
    }
  ],
  "RetCode": 0,
  "RetMsg": "success"
}
```

## 错误码

为了帮您快速定位问题, 云直播将错误码集中归类, 请到[错误码中心](#)查看详情。

## 查询推流信息接口 (ListRealtimePubStreamsInfo)

### 接口说明

- 用来查询域名下某应用或某条流的推流信息
- 限制访问频率: 50次/分钟
- 调用方式: HTTP GET
- 数据粒度: 100s

## 请求参数

参数名	类型	是否必须	描述
Action	String	Y	操作接口名，取值：ListRealtimePubStreamsInfo
Version	String	Y	版本号，取值：2017-01-01
UniqueName	String	Y	域名空间
App	String	Y	应用名
Pubdomain	String	Y	推流域名
Stream	String	N	流名
OrderTime	Int	N	排序类型，0：按推流开始时间降序排列，1：按推流开始时间升序排列
Marker	Int	N	起始游标，默认为0。0代表第1页的数据，以此类推
Limit	Int	N	每页返回的记录数，默认1000。单页最多返回5000条数据

## 返回参数

参数名	类型	描述
UniqueName	String	域名空间
App	String	应用名
Pubdomain	String	推流域名
Stream	String	流名
Count	Int	本页返回的记录数
Marker	Int	起始游标，默认为0。0代表第1页的数据，以此类推
OrderTime	Int	排序类型，0：按推流开始时间降序排列，1：按推流开始时间升序排列
Total	Int	符合查询条件的总记录数
ClientIP	String	推流客户端IP
Width	String	视频宽度
Height	String	视频高度
FrameRate	String	视频理论帧率
Curfps	Int	视频实时帧率
Curbitrate	Int	视频码率/B
BwInAudio	String	音频码率/B
SampleRate	String	音频采样率/Hz
BwReal	String	总真实码率/B
BwIn	String	总理论码率/B
UpdateTime	Int	流信息采集UTC时间
PubTime	Int	开始推流的UTC时间
RetCode	Int	响应码 0成功 其他失败
RetMsg	String	响应信息描述

## 请求示例

```
GET /?Action=ListRealtimePubStreamsInfo&Version=2017-01-01&UniqueName=xxx&App=live&Pubdomain=xxx.xxx.com&Marker=0&Limit=2&OrderTime=0 HTTP/1.1
Content-Type: application/json
X-Amz-Date: YOURDATE
Authorization: AWS4-HMAC-SHA256 Credential=YOURAK/YYYYMMDD/cn-beijing-6/kls/aws4_request, SignedHeaders=content-length;content-type;host;x-amz-date, Signature=SIG
Host: kls.api.ksyun.com
```

## 返回示例

```
{
  "Data": {
    "App": "live",
    "Count": 1,
    "Marker": 0,
    "OrderTime": 0,
    "Pubdomain": "xxx.xxx.com",
    "Result": [
      {
        "BwIn": "52666",
        "BwInAudio": "20814",
        "BwReal": "52227",
        "Clientip": "114.255.44.132",
        "Curbitrate": 52666,
        "Curfps": 31,
        "FrameRate": "30",
        "Height": "614",
        "Pubtime": 1491033051,
        "SampleRate": "44100",
        "Stream": "yb668",
        "UpdateTime": 1491036296,
        "Width": "1092"
      }
    ],
    "RetCode": 0,
    "RetMsg": "success",
    "Total": 1,
    "UniqueName": "xxx"
  }
}
```

## 错误码

为了帮您快速定位问题，云直播将错误码集中归类，请到[错误码中心](#)查看详情。

## 查询流历史信息接口(ListHistoryPubStreamsInfo)

## 接口说明

- 此接口用来查询某App下所有或单条流的历史信息
- 时间范围：28天
- 查询数据间隔：开始和结束时间的间隔不能超过5天
- 限制访问频率：50次/分钟
- 调用方式：HTTP GET

## 请求参数

参数名	类型	是否必须	描述
Action	String	Y	操作接口名，取值：ListHistoryPubStreamsInfo
Version	String	Y	版本号，取值：2017-01-01
UniqueName	String	Y	域名空间
App	String	Y	应用名
Pubdomain	String	Y	推流域名

Stream	String	N	流名
OrderTime	Int	N	排序类型（0：按推流开始的时间降序排列，1：按推流开始的时间升序排列），默认为0
Marker	Int	N	起始游标，默认为0。0代表第1页的数据，以此类推
Limit	Int	N	每页返回的记录数，默认1000。单页最多返回5000条数据
StartUnixTime	Int	N	UTC时间戳；默认5天前的0点，开始和结束时间间隔不能超过5天，支持查28天之内的数据
EndUnixTime	Int	N	UTC时间戳；默认当前时间，开始和结束时间间隔不能超过5天，支持查28天之内的数据

## 返回参数

参数名	类型	描述
UniqueName	String	域名空间
App	String	应用名
Pubdomain	String	推流域名
Stream	String	流名
StartUnixTime	Int	开始时间，UTC时间戳
EndUnixTime	Int	结束时间，UTC时间戳
Count	Int	本页返回的记录数
OrderTime	Int	排序类型
Total	Int	符合查询条件的记录总数
ClientIP	String	推流客户端IP
PubTime	Int	推流开始时间，UTC时间戳
FinTime	Int	推流结束时间，UTC时间戳
StatusCode	Int	断流原因错误码
StatusMessage	String	断流原因描述
RetCode	Int	响应码 0成功 其他失败
RetMsg	String	响应信息描述

## 断流原因错误码及英文描述

错误码	英文描述
200	Success operation!
422	Already exist stream!
423	Forbidden by blacklist!
425	Url expired!
426	Signature does not match!
428	Bad params!

## 请求示例

```
GET /?Action=ListHistoryPubStreamsInfo&Version=2017-01-01&UniqueName=xxx&App=live&Pubdomain=xxx.xxx.com&Marker=0&Limit=2&OrderTime=0 HTTP/1.1
Content-Type: application/json
X-Amz-Date: YOURDATE
Authorization: AWS4-HMAC-SHA256 Credential=YOURAK/YYYYMMDD/cn-beijing-6/kls/aws4_request, SignedHeaders=content-length;content-type;host;x-amz-date, Signature=SIG
Host: kls.api.ksyun.com
```

## 返回示例

```
{
  "Data": {
    "App": "live",
    "Count": 2,
    "EndUnixtime": 1492070336,
    "Marker": 0,
    "OrderTime": 0,
    "Pubdomain": "xxx.xxx.com",
    "Result": [
      {
        "Detail": [
          {
            "Clientip": "11.11.11.11",
            "Fintime": 1492070325,
            "Pubtime": 1492070324,
            "StatusCode": 200,
            "StatusMessage": "Success operation!"
          }
        ],
        "Stream": "stream5"
      },
      {
        "Detail": [
          {
            "Clientip": "11.11.11.11",
            "Fintime": 1492070332,
            "Pubtime": 1492070321,
            "StatusCode": 200,
            "StatusMessage": "Success operation!"
          }
        ],
        "Stream": "stream2"
      }
    ],
    "RetCode": 0,
    "RetMsg": "success",
    "StartUnixtime": 1491580800,
    "Total": 164745,
    "UniqueName": "xxx"
  }
}
```

## 错误码

为了帮您快速定位问题，云直播将错误码集中归类，请到[错误码中心](#)查看详情。

## 查询流历史错误信息接口(ListHistoryPubStreamsErrInfo)

### 接口说明

- 此接口用来查询某App下所有流或单条流的历史流错误信息
- 时间范围：28天
- 查询数据间隔：开始与结束时间的间隔不能超过5天
- 限制访问频率：50次/分钟
- 调用方式：HTTP GET

### 请求参数

参数名	类型	是否必须	描述
Action	String	Y	操作接口名, 取值: ListHistoryPubStreamsErrInfo
Version	String	Y	版本号, 取值: 2017-01-01
UniqueName	String	Y	域名空间
App	String	Y	应用名
Pubdomain	String	Y	推流域名
Stream	String	N	流名
OrderTime	Int	N	排序类型 (0: 按推流开始的时间降序排列, 1: 按推流开始的时间升序排列), 默认为0
Marker	Int	N	起始游标, 默认为0。0代表第1页的数据, 以此类推
Limit	Int	N	每页返回的记录数, 默认1000。单页最多返回5000条数据
StartUnixTime	Int	N	开始时间, UTC时间戳; 默认5天前的0点, 开始和结束时间间隔不能超过5天, 支持查28天内的数据
EndUnixTime	Int	N	结束时间, UTC时间戳; 默认当前时间, 开始和结束时间间隔不能超过5天, 支持查28天内的数据

## 返回参数

参数名	类型	描述
UniqueName	String	域名空间
App	String	应用名
Pubdomain	String	推流域名
Stream	String	流名
StartUnixTime	Int	开始时间, UTC时间戳
EndUnixTime	Int	结束时间, UTC时间戳
Count	Int	本页返回的记录数
OrderTime	Int	排序类型
Total	Int	符合查询条件的记录总数
ClientIP	String	推流客户端IP
PubTime	Int	推流开始时间, UTC时间戳
FinTime	Int	推流结束时间, UTC时间戳
StatusCode	Int	断流原因错误码
StatusMessage	String	断流原因英文描述
RetCode	Int	响应码 0成功 其他失败
RetMsg	String	响应信息描述

## 断流原因错误码及英文描述

错误码	英文描述
200	Sucess operation!
422	Already exist stream!
423	Forbidden by blacklist!
425	Url expired!
426	Signature does not match!
428	Bad params!

## 请求示例

```
GET /?Action=ListHistoryPubStreamsErrInfo&Version=2017-01-01&UniqueName=xxx&App=live&Pubdomain=xxx.xxx.com&Marker=0&Limit=2&OrderTime=0 HTTP/1.1
Content-Type: application/json
X-Amz-Date: YOURDATE
Authorization: AWS4-HMAC-SHA256 Credential=YOURAK/YYYYMMDD/cn-beijing-6/kls/aws4_request, SignedHeaders=content-length;content-type;host;x-amz-date, Signature=SIG
Host: kls.api.ksyun.com
```

## 返回示例

```
{
  "Data": {
    "App": "live",
    "Count": 2,
    "EndUnixTime": 1492070399,
    "Marker": 0,
    "OrderTime": 0,
    "Pubdomain": "xxx.xxx.com",
    "Result": [
      {
        "Detail": [
          {
            "Clientip": "localhost",
            "Fintime": 1492070338,
            "Pubtime": 1492070338,
            "StatusCode": 422,
            "StatusMessage": "Unknown"
          },
          {
            "Clientip": "localhost",
            "Fintime": 1492070338,
            "Pubtime": 1492070338,
            "StatusCode": 422,
            "StatusMessage": "Unknown"
          }
        ]
      }
    ],
    "Stream": "stream"
  },
  "RetCode": 0,
  "RetMsg": "success",
  "StartUnixTime": 1491580800,
  "Total": 5275,
  "UniqueName": "xxx"
}
```

## 错误码

为了帮您快速定位问题, 云直播将错误码集中归类, 请到[错误码中心](#)查看详情。

## 查询主播流时长接口(ListStreamDurations)

### 接口说明

- 此接口用来统计某App下所有流或单条流的推流时长信息
- 使用前提: 此接口开通后才能使用, 请联系技术支持, 开通维度: UniqueName
- 时间范围: 60天
- 查询时间间隔: 开始和结束时间不能为同一天
- 统计方式: 以0点为统计的开始和结束点, 举例说明: 输入的开始时间在前天, 结束时间在今天, 则返回前天0点到今天0点这条流的推流时长信息

- 限制访问频率：50次/分钟
- 调用方式：HTTP GET

## 请求参数

参数名	类型	是否必须	描述
Action	String	Y	操作接口名，取值：ListStreamDurations
Version	String	Y	版本号，取值：2017-01-01
UniqueName	String	Y	域名空间
App	String	Y	频道
Pubdomain	String	Y	推流域名
Stream	String	N	流名，如果携带则返回单条流的推流时长；否则返回整个App下所有流的推流时长
StartUnixTime	Int	Y	查询开始时间，UTC时间戳，以当天00:00点为开始时间
EndUnixTime	Int	Y	查询结束时间，UTC时间戳，以当天00:00点为结束时间

## 返回参数

参数名	类型	描述
UniqueName	String	域名空间
App	String	频道
Pubdomain	String	推流域名
Stream	String	流名
StartTime	Int	推流开始时间，UTC时间戳
EndTime	Int	推流结束时间，UTC时间戳
Duration	Int	每段推流的总时长
Count	Int	单次返回记录数

## 请求示例

```
GET /?Action=ListStreamDurations&Version=2017-01-01&UniqueName=xxx&App=xxx&Pubdomain=xxx&Stream=xxx&StartUnixTime=xxx&EndUnixTime=xxx&公共参数 HTTP/1.1
Content-Type: application/json
X-Amz-Date: YOURDATE
Authorization: AWS4-HMAC-SHA256 Credential=YOURAK/YYYYMMDD/cn-beijing-6/kls/aws4_request, SignedHeaders=content-length;content-type;host;x-amz-date, Signature=SIG
Host: kls.api.ksyun.com
```

## 返回示例

请求参数不携带 Stream 时，返回 Pubdomain 下所有流的推流时长 注意：如果同一条流存在断流情况，那么返回多条记录，如下：

```
{
  "Data": {
    "App": "live",
    "Count": 5,
    "Pubdomain": "xxx.xxxxx.com",
    "Result": [
      {
        "Duration": 11,
        "EndTime": 1490069354,
        "StartTime": 1490069283,
        "Stream": "10817737"
      },
      {
        "Duration": 11,
        "EndTime": 1490069430,
        "StartTime": 1490069371,
        "Stream": "10817737"
      },
      {
        "Duration": 22,
        "EndTime": 1490069534,
        "StartTime": 1490069468,
        "Stream": "10817737"
      },
      {
        "Duration": 33,
        "EndTime": 1490069816,
        "StartTime": 1490069777,
        "Stream": "10817737"
      },
      {
        "Duration": 555,
        "EndTime": 1490078283,
        "StartTime": 1490078218,
        "Stream": "10817737"
      }
    ]
  },
  "RetCode": 0,
  "RetMsg": "success",
  "UniqueName": "xxx"
},
"RequestId": "2125d244-7f40-45f0-91e8-05e046b3a61c"
```

请求参数携带 Stream 时，只返回单条流的推流时长，如下

```
{
  "Data": {
    "App": "live",
    "Count": 2,
    "Pubdomain": "xxx.xxxxx.com",
    "Result": [
      {
        "Duration": 11,
        "EndTime": 1490072527,
        "StartTime": 1490072515,
        "Stream": "10817737"
      },
      {
        "Duration": 11,
        "EndTime": 1490078283,
        "StartTime": 1490078218,
        "Stream": "10817737"
      }
    ]
  },
  "RetCode": 0,
  "RetMsg": "success",
  "UniqueName": "xxx"
},
"RequestId": "2125d244-7f40-45f0-91e8-05e046b3a61c"
```

## 错误码

为了帮您快速定位问题，云直播将错误码集中归类，请到[错误码中心](#)查看详情。

## 禁止单路直播流推送（ForbidStream）

### 接口描述

- 功能1：封禁单路直播流
- 功能2：断开单路直播流，且支持立即重新推流
- 接口请求频率限制：200次/秒
- 调用方式：HTTP POST

### 请求参数

参数名	类型	是否必须	描述	示例
Action	String	Y	操作接口名，取值：ForbidStream	ForbidStream
Version	String	Y	版本号，取值：2017-01-01	2017-01-01
UniqueName	String	Y	域名空间	testUniquename
App	String	Y	应用名	testApp
Pubdomain	String	Y	推流域名	www.yourdomain.com
Stream	String	Y	流名	testSteam
ForbidTillUnixTime	Int	N	流恢复的时间，Unix秒级时间戳 注意：默认禁播90天，且最长支持禁播90天 传1，立即断开直播流 传-1，立即断开直播流并禁播90天	1546300800

### 返回参数

参数名	类型	描述
RetCode	Int	响应码。0 成功，其他失败
RetMsg	String	响应信息描述

### 请求示例

```
POST /?Action=ForbidStream&Version=2017-01-01 HTTP/1.1
Content-Type: application/json
X-Amz-Date: YOURDATE
Authorization: AWS4-HMAC-SHA256 Credential=YOURAK/YYYYMMDD/cn-beijing-6/kls/aws4_request, SignedHeaders=content-length;content-type;host;x-amz-date, Signature=SIG
Host: kls.api.ksyun.com
```

```
{
  "UniqueName": "YOUR UNIQUENAME",
  "App": "YOUR APP",
  "Pubdomain": "YOUR PUBLISH DOMAIN",
  "Stream": "YOUR STREAM NAME"
}
```

### 返回示例

```
{
  "Data": {
    "RetCode": 0,
    "RetMsg": "success"
  }
}
```

### 错误码

为了帮您快速定位问题，云直播将错误码集中归类，请到[错误码中心](#)查看详情。

## 恢复单路直播流推送（ResumeStream）

### 接口描述

- 用于解除单路流的封禁
- 接口请求频率限制：100次/秒
- 调用方式：HTTP POST

### 备注

由于历史原因，该接口只能解封新版接口（ForbidStream）封禁的流。

### 请求参数

参数名	类型	是否必须	描述
Action	String	Y	操作接口名，取值：ResumeStream
Version	String	Y	版本号，取值：2017-01-01
UniqueName	String	Y	域名空间
App	String	Y	应用名
Pubdomain	String	Y	推流域名
Stream	String	Y	流名

### 返回参数

参数名	类型	描述
RetCode	Int	响应码 0成功 其他失败
RetMsg	String	响应信息描述

### 请求示例

```
POST /?Action=ResumeStream&Version=2017-01-01 HTTP/1.1
Content-Type: application/json
X-Amz-Date: YOURDATE
Authorization: AWS4-HMAC-SHA256 Credential=YOURAK/YYYYMMDD/cn-beijing-6/kls/aws4_request, SignedHeaders=content-length;content-type;host;x-amz-date, Signature=SIG
Host: kls.api.ksyun.com
```

```
{
  "UniqueName": "xxx",
  "App": "live",
  "Pubdomain": "xxx.xxx.com",
  "Stream": "xxx"
}
```

### 返回示例

```
{
  "Data": {
    "RetCode": 0,
    "RetMsg": "success"
  }
}
```

## 错误码

为了帮您快速定位问题，云直播将错误码集中归类，请到[错误码中心](#)查看详情。

## 查询黑名单列表（GetBlacklist）

### 接口描述

- 用于查询推流域名下某一App的封禁流信息
- 接口请求频率限制：100次/秒
- 调用方式：HTTP GET

### 备注

由于历史原因，该接口只能查询新版接口（ForbidStream）封禁的流。

### 请求参数

参数名	类型	是否必须	描述
Action	String	Y	操作接口名，取值：GetBlacklist
Version	String	Y	版本号，取值：2017-01-01
UniqueName	String	Y	域名空间
App	String	Y	应用名
Pubdomain	String	Y	推流域名

### 返回参数

参数名	类型	描述
UniqueName	String	域名空间
App	String	应用名
Pubdomain	String	推流域名
Total	Int	记录总数
Stream	String	流名
ForbidTillUnixTime	Int	Unix 时间戳，流禁推的结束时间；-1表示永久禁播
ForbidUnixTime	Int	Unix 时间戳，最后一次拉入黑名单时间
RetCode	Int	响应码 0成功，其他失败
RetMsg	String	响应信息描述

### 请求示例

http://kls.api.ksyun.com?Action=GetBlacklist&Version=2017-01-01&UniqueName=xxxx&App=live&Pubdomain=xxx.xxx.com

```
GET /?Action=GetBlacklist&Version=2017-01-01&UniqueName=xxxx&App=live&Pubdomain=xxx.xxx.com HTTP/1.1
```

Content-Type: application/json

X-Amz-Date: YOURDATE

Authorization: AWS4-HMAC-SHA256 Credential=YOURAK/YYYYMMDD/cn-beijing-6/kls/aws4\_request, SignedHeaders=content-length;content-type;host;x-amz-date, Signature=SIG

Host: kls.api.ksyun.com

### 返回示例

```
{
  "Data": {
    "App": "live",
    "Pubdomain": "xxx.xxx.com",
    "Recs": [
      {
        "ForbidTillUnixTime": -1,
        "ForbidUnixTime": 1497434114,
        "Stream": "aaa"
      }
    ],
    "RetCode": 0,
    "RetMsg": "success",
    "Total": 1,
    "UniqueName": "xxx"
  },
  "RequestId": "1680b7b1-a7ee-4906-8ff1-8ee66b451dc6"
}
```

## 错误码

为了帮您快速定位问题，云直播将错误码集中归类，请到[错误码中心](#)查看详情。

## 检查流是否在黑名单内（CheckBlacklist）

### 接口描述

- 用于查询单条流是否被封禁
- 接口请求频率限制：100次/秒
- 调用方式：HTTP GET

### 备注

由于历史原因，该接口只能查询新版接口（ForbidStream）封禁的流。

### 请求参数

参数名	类型	是否必须	描述
Action	String	Y	操作接口名，取值：CheckBlacklist
Version	String	Y	版本号，取值：2017-01-01
UniqueName	String	Y	域名空间
App	String	Y	应用名
Pubdomain	String	Y	推流域名
Stream	String	Y	流名

## 返回参数

参数名	类型	是否必须	描述
CheckResult	Int	Y	检查流名是否在黑名单内，1：在黑名单内，0：不在黑名单内
RetCode	Int	Y	响应码 0成功 其他失败
RetMsg	String	Y	响应信息描述

## 请求示例

```
GET /?Action=CheckBlacklist&Version=2017-01-01&UniqueName=xxx&App=live&Pubdomain=xxx.xxx.com&Stream=bbb HTTP/1.1
Content-Type: application/json
X-Amz-Date: YOURDATE
Authorization: AWS4-HMAC-SHA256 Credential=YOURAK/YYYYMMDD/cn-beijing-6/cls/aws4_request, SignedHeaders=content-length;content-type;host;x-amz-date, Signature=SIG
Host: cls.api.ksyun.com
```

## 返回示例

```
{
  "Data": {
    "CheckResult": 1,
    "RetCode": 0,
    "RetMsg": "success"
  },
  "RequestId": "bd297352-1ecf-4731-a190-a64111583890"
}
```

## 错误码

为了帮您快速定位问题，云直播将错误码集中归类，请到[错误码中心](#)查看详情。

# 视频云直播SDK for java使用指南

## 云直播 Java SDK 概述：

1. SDK支持的直播中心：华北1（北京）、华北2（北京）
2. SDK下载地址为：<https://github.com/KscSDK>

## 安装java sdk

### 1. git 安装

```
git clone https://github.com/KscSDK/ksc-sdk-java.git
```

```
mvn clean install
```

### 2. mvn 中央仓库安装

设置pom文件即可

### 3. 工程使用sdk录

建议使用Maven构建自己的项目，导入ksc-sdk-java，添加ksc-sdk-java-cls的依赖

新建maven项目，进入到pom文件下，进行如下配置

```
<dependencies>
  <dependency>
    <groupId>com.ksyun</groupId>
    <artifactId>ksc-sdk-java-cls</artifactId>
    <version>0.3.6</version>
  </dependency>
</dependencies>
```

## 4. 通过文件配置及管理密钥

本地文件配置：

```
~/aws/credentials Linux, MacOS和Unix系统
C:\Users\{USERNAME}\.aws\credentials Windows系统
```

该文件包含下述内容：

```
[default]
aws_access_key_id = your_access_key_id
aws_secret_access_key = your_secret_access_key
```

如不通过本地文件读入ak和sk信息，则需要执行的类文件中增加以下代码：

```
AWSCredentials credentials = new BasicAWSCredentials(aws_access_key_id, aws_secret_access_key);
```

## 调用方式

在调用接口前，先初始化 KSCOFFJsonClient，如下用例：

```
KSCSClient kls_client = new KSCSClient();
//可传入AWSCredentials参数 如:KSCSClient kls_client = new KSCSClient(credentials);
kls_client.setEndpoint("http://cls.api.ksyun.com");
字符集: utf-8
```

注：为了更方便您调用SDK，我们准备了实际的SDK调用示例，您只需要替换其中的AKSK和请求参数即可，详见[SDK调用示例](#)

## 接口目录说明

- [ForbidStream](#) 禁止单路直播流推送接口
- [ResumeStream](#) 恢复单路直播流推送接口
- [GetBlacklist](#) 查询黑名单列表接口
- [CheckBlacklist](#) 检查流名是否在黑名单内
- [ListStreamDurations](#) 查询主播流时长接口
- [ListHistoryPubStreamsInfo](#) 查询流历史信息接口
- [ListHistoryPubStreamsErrInfo](#) 查询流历史错误信息接口
- [ListRealtimeStreamsInfo](#) 查询主播推流实时信息接口
- [ListRealtimePubStreamsInfo](#) 查询推流实时信息接口

# 视频云直播SDK for php 使用指南

## 云直播 PHP SDK 概述

1. SDK 适用于PHP 5.5 及以上版本
2. SDK支持的直播中心: 华北1(北京)、华北2(北京)
3. SDK下载地址为: <https://github.com/KscSDK>

### 1. 初始化

#### 1.1 ak/sk 配置

在金山云控制台获取到ak/sk后创建以下文件 (如有问题请联系商务或者技术支持):

```
....  
mkdir ~/.ksyun && vim ~/.ksyun/config  
....  
config文件内容为:  
....  
{  
  "ak": "*****",  
  "sk": "*****"  
}
```

#### 1.2 composer安装

```
....  
mkdir test && cd test  
composer require kscsdk/ksyun_sdk  
....
```

如果需要使用最新版本, 安装完之后, 修改composer.json为

```
....  
{  
  "require": {  
    "kscsdk/ksyun_sdk": "dev-master"  
  }  
}
```

然后再执行更新操作

```
....  
composer update  
....
```

#### 1.3 调试demo

```
....  
cp vendor/kscsdk/ksyun_sdk/examples/demo_K1s.php .  
php demo_K1s.php GetBlacklist  
....
```

#### 1.4 调用示例

```
....  
<?php  
require('vendor/autoload.php');  
use Ksyun\Service\K1s;  
$response = K1s::getInstance()->request('GetBlacklist');  
echo $response->getBody();  
....
```

注:为了更方便您调用SDK, 我们准备了实际的SDK调用示例, 详见[SDK调用示例](#)

## 2. Post 方法调用示例

```
// Post 方法的调用均使用与下面示例相同的方式  
// 文档中仅以禁止单路直播流推送的接口为例  
// 调用示例  
....  
<?php  
require('vendor/autoload.php');  
use Ksyun\Service\K1s;  
  
$app = 'live'; // 频道名  
$uniqueName = 'test'; // 用户名  
$pubdomain = 'test.ksyun.com'; // 推流域名  
$stream = 'test1234'; // 流名  
$forbidTillUnixTime = -1 // 禁流时间  
$forbid_stream_data = [  
  'App' => $app,  
  'UniqueName' => $uniqueName,  
  'Pubdomain' => $pubdomain,  
  'Stream' => $stream,  
  'ForbidTillUnixTime' => $forbidTillUnixTime  
];  
  
$response = K1s::getInstance()->request('ForbidStream', 'json' => $forbid_stream_data);  
echo $response->getBody();  
....
```

## 3. Get 方法调用示例

Get 方法的调用均使用与下面示例相同的方式

```
// 文档中仅以查询录像任务状态的接口为例  
// 调用示例  
....  
<?php  
require('vendor/autoload.php');  
use Ksyun\Service\K1s;
```

```
$rec_id = 1: //录像任务ID
$response = Ket::getInstance()->request('GetRecordTask', 'json' => ['query' => ['RecID' => $rec_id]]);
echo $response->getBody();
...
```

## 视频云直播SDK for Python 使用指南

### 视频云直播 Python SDK 概述

1. SDK适用于2.6、2.7的Python版本
2. SDK支持的直播中心：华北1（北京）、华北2（北京）
3. SDK下载地址为：<https://github.com/KscSDK>

#### 1. 安装python sdk

##### 1.1 git 安装

从GitHub上获取SDK并安装

```
...
git clone https://github.com/KscSDK/ksc-sdk-python.git
cd ksc-sdk-python
python setup.py install
...
```

##### 1.2 pip 安装

```
...
pip install ksc-sdk-python
...
```

#### 2. 配置管理密钥

##### 2.1 通过文件配置密钥

- 1. 所在位置：'/etc/kscore.cfg' 或 './.kscore.cfg'
- 2. 注意：使用相对路径时，需与运行目录保持一致
- 3. ks\_access\_key\_id和ks\_secret\_access\_key是金山云控制台身份与管理里面生成密钥对(如有问题请联系商务或者技术支持)

```
...
[Credentials]
ks_access_key_id=xxxxxxxxxxxxxxxxxxxxx
ks_secret_access_key=xxxxxxxxxxxxxxxxxxxxx
...
```

##### 2.2 在代码中配置密钥

```
...
from kscore.kls import getKlsClient
// 若使用文件配置过密钥请使用这种方式:
client = getKetClient("kls", "cn-beijing-6", use_ssl=False)
//参数说明:
service_name: 服务名, 填写"kls"
region_name: 区域名, 填写"cn-beijing-6"
use_ssl: 是否https访问, 填写False
// 若准备使用代码中配置密钥, 请使用这种方式:
ks_access_key_id=xxxxxxxxxxxxxxxxxxxxx
ks_secret_access_key=xxxxxxxxxxxxxxxxxxxxx
//参数: 服务service_name, 大区region_name
client = getKetClient("kls", "cn-beijing-6", use_ssl=False, ks_access_key_id=ks_access_key_id, ks_secret_access_key=ks_secret_access_key)
...
```

注:为了更方便您调用SDK, 我们准备了实际的SDK调用示例, 详见[SDK调用示例](#)

#### 3. Post 方法调用示例

```
// Post 方法的调用均使用与下面示例相同的方式
// 文档中仅以禁止单路直播流推送的接口为例*
// 调用示例
...
#!/usr/bin/python
# -*- coding: UTF-8 -*-
from kscore.kls import getKlsClient
import json

if __name__ == "__main__":
    #ks_access_key_id=xxxxxxxxxxxxxxxxxxxxx
    #ks_secret_access_key=xxxxxxxxxxxxxxxxxxxxx
    client = getKlsClient("kls", "cn-beijing-6", use_ssl=False, ks_access_key_id=ks_access_key_id, ks_secret_access_key=ks_secret_access_key)

    param = {
        "UniqueName": "YOUR UNIQUENAME",
        "App": "YOUR APP",
        "Pubdomain": "YOUR PUBLISH DOMAIN",
        "Stream": "YOUR STREAM NAME",
        "ForbidTillUnixTime": -1
    }

    #禁止单路直播流推送: ForbidStream
    res = client.ForbidStream(param)
    print json.dumps(res)
...
```

#### 4. Get 方法调用示例

```
// Get 方法的调用均使用与下面示例相同的方式
// 文档中仅以查询录像任务状态的接口为例
// 调用示例
...
#!/usr/bin/python
# -*- coding: UTF-8 -*-
from kscore.kls import getKlsClient
```

```
import json

if __name__ == "__main__":
    #ks_access_key_id='xxxxxxxxxxxxxxxxxxxxxxxxxxxx'
    #ks_secret_access_key='xxxxxxxxxxxxxxxxxxxxxxxxxxxx'
    client = getKlsClient("kls", "cn-beijing-6", use_ssl=False, ks_access_key_id=ks_access_key_id, ks_secret_access_key=ks_secret_access_key)

    #查询录像任务状态接口(GetRecordTask)
    res = client.GetRecordTask(RecID = 2017)
    print json.dumps(res)

...
```