

# 云直播

## 云端 API

### 产品文档



腾讯云

**【版权声明】**

©2013-2018 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

**【商标声明】**

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

**【服务声明】**

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

## 文档目录

### 云端 API

简介

API 概览

调用方式

请求结构

公共参数

接口鉴权

返回结果

更新历史

延播管理相关接口

延迟播放

恢复延播

直播流管理相关接口

查询在线推流信息列表

查询直播中的流

查询推过流的流列表

查询流状态

断开直播流

禁播直播流

恢复直播流

数据结构

云直播 API 2017

概述

直播码模式

概述

API 概览

操作类接口

事件消息通知

开启关闭推流

云端混流

暂停并延迟恢复

创建录制任务

结束录制任务

延播

查询类接口

查询直播状态

查询统计信息

查询录制文件

查询频道列表

查询直播中频道列表

获取计费带宽数据

统计类接口

获取推流历史信息

获取播放统计历史信息

频道模式 (旧)

调用方式

请求结构

后台 SDK

返回结果

签名方法

服务端API

API 概览

创建直播频道

修改直播频道

删除直播频道

批量启用直播频道

批量停止直播频道

查询直播频道列表

查询直播频道详情

创建录制任务

终止录制任务

删除录制任务

查询录制任务列表

查询录制文件(荐)

查询已录制分片列表

错误码表

# 云端 API 简介

最近更新时间：2018-10-12 16:25:31

**云直播 API 升级到 3.0 版本，全新的 API 接口文档更加规范和全面，统一的参数风格和公共错误码，统一的 SDK/CLI 版本与 API 文档严格一致，给您带来简单快捷的使用体验；支持全地域就近接入让您更快连接腾讯云产品。**

云直播（Live Video Broadcasting）依托腾讯强大的技术平台，将腾讯视频等核心业务底层能力开放给用户，为用户提供专业稳定快速的直播接入和分发服务，全面满足超低延迟和超大并发访问量的苛刻要求。

# API 概览

最近更新时间：2018-10-12 16:25:30

## 延播管理相关接口

接口名称	接口功能
<a href="#">AddDelayLiveStream</a>	延迟播放
<a href="#">ResumeDelayLiveStream</a>	恢复延播

## 直播流管理相关接口

接口名称	接口功能
<a href="#">DescribeLiveStreamOnlineInfo</a>	查询在线推流信息列表
<a href="#">DescribeLiveStreamOnlineList</a>	查询直播中的流
<a href="#">DescribeLiveStreamPublishedList</a>	查询推过流的流列表
<a href="#">DescribeLiveStreamState</a>	查询流状态
<a href="#">DropLiveStream</a>	断开直播流
<a href="#">ForbidLiveStream</a>	禁播直播流
<a href="#">ResumeLiveStream</a>	恢复直播流

# 调用方式

## 请求结构

最近更新时间：2018-10-12 16:25:31

### 1. 服务地址

API 支持就近地域接入，本产品就近地域接入域名为 `live.tencentcloudapi.com`，也支持指定地域域名访问，例如广州地域的域名为 `live.ap-guangzhou.tencentcloudapi.com`。

推荐使用就近地域接入域名。根据调用接口时客户端所在位置，会自动解析到**最近的**某个具体地域的服务器。例如在广州发起请求，会自动解析到广州的服务器，效果和指定 `live.ap-guangzhou.tencentcloudapi.com` 是一致的。

目前支持的域名列表为：

接入地域	域名
就近地域接入（推荐，只支持非金融区）	<code>live.tencentcloudapi.com</code>
华南地区(广州)	<code>live.ap-guangzhou.tencentcloudapi.com</code>
华东地区(上海)	<code>live.ap-shanghai.tencentcloudapi.com</code>
华北地区(北京)	<code>live.ap-beijing.tencentcloudapi.com</code>
西南地区(成都)	<code>live.ap-chengdu.tencentcloudapi.com</code>
西南地区(重庆)	<code>live.ap-chongqing.tencentcloudapi.com</code>
东南亚地区(首尔)	<code>live.ap-seoul.tencentcloudapi.com</code>
东南亚地区(新加坡)	<code>live.ap-singapore.tencentcloudapi.com</code>
亚太地区(孟买)	<code>live.ap-mumbai.tencentcloudapi.com</code>
美国西部(硅谷)	<code>live.na-siliconvalley.tencentcloudapi.com</code>
美国东部(弗吉尼亚)	<code>live.na-ashburn.tencentcloudapi.com</code>

**注意：**由于**金融区**和非金融区是隔离不互通的，因此当访问金融区服务时（公共参数 `Region` 为金融区地域），需要同时指定带金融区地域的域名，最好和 `Region` 的地域保持一致。

金融区接入地域	金融区域名
---------	-------

金融区接入地域	金融区域名
华东地区(上海金融)	live.ap-shanghai-fsi.tencentcloudapi.com
华南地区(深圳金融)	live.ap-shenzhen-fsi.tencentcloudapi.com

## 2. 通信协议

腾讯云 API 的所有接口均通过 HTTPS 进行通信，提供高安全性的通信通道。

## 3. 请求方法

支持 POST 和 GET 请求。POST 请求目前仅支持 Content-Type 类型为 application/x-www-form-urlencoded。

## 4. 字符编码

均使用UTF-8编码。



# 公共参数

最近更新时间：2018-10-12 16:25:31

公共参数是用于标识用户和接口鉴权目的的参数，如非必要，在每个接口单独的接口文档中不再对这些参数进行说明，但每次请求均需要携带这些参数，才能正常发起请求。

参数名称	类型	必选	描述
Action	String	是	具体操作的指令接口名称，例如想要调用云服务器的查询实例列表接口，则 Action 参数即为 DescribeInstances。
Region	String	是	地域参数，用来标识希望操作哪个地域的数据。
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在 <a href="#">云API密钥</a> 上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。
Version	String	是	API 的版本。例如 2017-03-12
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

假设用户想要查询广州地域的云服务器实例列表，则其请求链接的形式可能如下：

```
https://cvm.tencentcloudapi.com/?Action=DescribeInstances
&SecretId=xxxxxxx
&Region=ap-guangzhou
&Timestamp=1402992826
&Nonce=345122
&Signature=xxxxxxx
&Version=2017-03-12
```

# 接口鉴权

最近更新时间：2018-10-12 16:25:31

腾讯云 API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证请求者身份。签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往[云API密钥页面](#)申请，否则无法调用云API接口。

## 1. 申请安全凭证

在第一次使用云API之前，请前往[云API密钥页面](#)申请安全凭证。安全凭证包括 SecretId 和 SecretKey：

SecretId 用于标识 API 调用者身份 SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。 **用户必须严格保管安全凭证，避免泄露。**

申请安全凭证的具体步骤如下：

- 1) 登录[腾讯云管理中心控制台](#)。
- 2) 前往[云API密钥](#)的控制台页面
- 3) 在[云API密钥](#)页面，点击【新建】即可以创建一对SecretId/SecretKey

**开发商帐号最多可以拥有两对SecretId / SecretKey。**

## 2. 生成签名串

有了安全凭证SecretId 和 SecretKey后，就可以生成签名串了。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

```
SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE SecretKey:  
Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
```

**注意：这里只是示例，请根据用户实际申请的SecretId和SecretKey进行后续操作！**

以云服务器查看实例列表(DescribeInstances)请求为例，当用户调用这一接口时，其请求参数可能如下：

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	ap-guangzhou
InstanceIds.0	待查询的实例ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

## 2.1. 对参数排序

首先对所有请求参数按参数名的字典序（ASCII 码）升序排序。注意：1）只按参数名进行排序，参数值保持对应即可，不参与比大小；2）按 ASCII 码比大小，如 InstanceIds.2 要排在 InstanceIds.12 后面，不是按字母表，也不是按数值。用户可以借助编程语言中的相关排序函数来实现这一功能，如 php 中的 ksort 函数。上述示例参数的排序结果如下：

```
{
  'Action': 'DescribeInstances',
  'InstanceIds.0': 'ins-09dx96dg',
  'Limit': 20,
  'Nonce': 11886,
  'Offset': 0,
  'Region': 'ap-guangzhou',
  'SecretId': 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
  'Timestamp': 1465185768,
  'Version': '2017-03-12',
}
```

使用其它程序设计语言开发时，可对上面示例中的参数进行排序，得到的结果一致即可。

## 2.2. 拼接请求字符串

此步骤生成请求字符串。 将把上一步排序好的请求参数格式化成“参数名称”=“参数值”的形式，如对 Action 参数，其参数名称为 "Action"，参数值为 "DescribeInstances"，因此格式化后就为 Action=DescribeInstances。 **注**

意：“参数值”为原始值而非url编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为：

```
Action=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

### 2.3. 拼接签名原字符串

此步骤生成签名原字符串。签名原字符串由以下几个参数构成：

1) 请求方法: 支持 POST 和 GET 方式，这里使用 GET 请求，注意方法为全大写。 2) 请求主机: 查看实例列表 (DescribeInstances) 的请求域名为：cvm.tencentcloudapi.com。实际的请求域名根据接口所属模块的不同而不同，详见各接口说明。 3) 请求路径: 当前版本云API的请求路径固定为 /。 4) 请求字符串: 即上一步生成的请求字符串。

签名原字符串的拼接规则为：

```
请求方法 + 请求主机 + 请求路径 + ? + 请求字符串
```

示例的拼接结果为：

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

### 2.4. 生成签名串

此步骤生成签名串。首先使用 HMAC-SHA1 算法对上一步中获得的**签名原字符串**进行签名，然后将生成的签名串使用 Base64 进行编码，即可获得最终的签名串。

具体代码如下，以 PHP 语言为例：

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';  
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12';  
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));  
echo $signStr;
```

最终得到的签名串为：

```
EliP9YW3pW28FpsEdkXt/+WcGel=
```

使用其它程序设计语言开发时，可用上面示例中的原文进行签名验证，得到的签名串与例子中的一致即可。

### 3. 签名串编码

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。**注意：如果用户的请求方法是 GET，或者请求方法为 POST 同时 Content-Type 为 application/x-www-form-urlencoded，则对所有请求参数值均需要做 URL 编码。非 ASCII 字符在 URL 编码前需要先以 UTF-8 进行编码。**如上一步生成的签名串为 `EliP9YW3pW28FpsEdkXt/+WcGel=`，最终得到的签名串请求参数(Signature)为：  
`EliP9YW3pW28FpsEdkXt/+WcGel=`，它将用于生成最终的请求URL。

注意：有些编程语言的 http 库会自动为所有参数进行 urlencode，在这种情况下，就不需要对签名串进行 URL 编码了，否则两次 URL 编码会导致签名失败。

### 4. 鉴权失败

根据实际情况，存在以下鉴权失败的错误码，请根据实际情况处理

错误代码	错误描述
<code>AuthFailure.SignatureExpire</code>	签名过期
<code>AuthFailure.UnauthorizedOperation</code>	请求未 CAM 授权
<code>AuthFailure.SecretIdNotFound</code>	密钥不存在
<code>AuthFailure.SignatureFailure</code>	签名错误
<code>AuthFailure.TokenFailure</code>	token 错误
<code>AuthFailure.MFAFailure</code>	MFA 错误
<code>AuthFailure.InvalidSecretId</code>	密钥非法（不是云 API 密钥类型）

### 5. 签名演示

在实际调用 API 3.0 时，推荐使用配套的腾讯云 SDK 3.0，SDK 封装了签名的过程，开发时只关注产品提供的具体接口即可。详细信息参见 [SDK 中心](#)。当前支持的编程语言有：

- [Python](#)
- [Java](#)
- [PHP](#)

- [Go](#)
- [JavaScript](#)
- [.NET](#)

为了更清楚的解释签名过程，下面以实际编程语言为例，将上述的签名过程具体实现。请求的域名、调用的接口和参数的取值都以上述签名过程为准，代码只为解释签名过程，并不具备通用性，实际开发请尽量使用 SDK。

最终输出的 url 可能为：`https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=Elip9YW3pW28FpsEdkXt/+WcGel=&Timestamp=1465185768&Version=2017-03-12`

注意：由于示例中的密钥是虚构的，时间戳也不是系统当前时间，因此如果将此 url 在浏览器中打开或者用 curl 等命令调用时会返回鉴权错误：签名过期。为了得到一个可以正常返回的 url，需要修改示例中的 SecretId 和 SecretKey 为真实的密钥，并使用系统当前时间戳作为 Timestamp。

注意：在下面的示例中，不同编程语言，甚至同一语言每次执行得到的 url 可能都有所不同，表现为参数的顺序不同，但这并不影响正确性。只要所有参数都在，且签名计算正确即可。

注意：以下代码仅适用于 API 3.0，不能直接用于其他的签名流程，即使是旧版的 API，由于存在细节差异也会导致签名计算错误，请以对应的实际文档为准。

## Java

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Random;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TencentCloudAPIDemo {
    private final static String CHARSET = "UTF-8";

    public static String sign(String s, String key, String method) throws Exception {
        Mac mac = Mac.getInstance(method);
        SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
        mac.init(secretKeySpec);
        byte[] hash = mac.doFinal(s.getBytes(CHARSET));
        return DatatypeConverter.printBase64Binary(hash);
    }

    public static String getStringToSign(TreeMap<String, Object> params) {
        StringBuilder s2s = new StringBuilder("GETcvm.tencentcloudapi.com/?");
    }
}
```

```
// 签名时要求对参数进行字典排序，此处用TreeMap保证顺序
for (String k : params.keySet()) {
    s2s.append(k).append("=").append(params.get(k).toString()).append("&");
}
return s2s.toString().substring(0, s2s.length() - 1);
}

public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException {
    StringBuilder url = new StringBuilder("https://cvm.tencentcloudapi.com/?");
    // 实际请求的url中对参数顺序没有要求
    for (String k : params.keySet()) {
        // 需要对请求串进行urlencode，由于key都是英文字母，故此处仅对其value进行urlencode
        url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).append("&");
    }
    return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
    TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
    // 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_
    VALUE));
    params.put("Nonce", 11886); // 公共参数
    // 实际调用时应当使用系统当前时间，例如：params.put("Timestamp", System.currentTimeMillis() / 1000);
    params.put("Timestamp", 1465185768); // 公共参数
    params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 公共参数
    params.put("Action", "DescribeInstances"); // 公共参数
    params.put("Version", "2017-03-12"); // 公共参数
    params.put("Region", "ap-guangzhou"); // 公共参数
    params.put("Limit", 20); // 业务参数
    params.put("Offset", 0); // 业务参数
    params.put("InstanceId.0", "ins-09dx96dg"); // 业务参数
    params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE", "H
    macSHA1")); // 公共参数
    System.out.println(getUrl(params));
}
}
```

## Python

注意：如果是在 Python 2 环境中运行，需要先安装 requests 依赖包：`pip install requests`。

```
# -*- coding: utf8 -*-
import base64
import hashlib
import hmac
```

```
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

def get_string_to_sign(method, endpoint, params):
    s = method + endpoint + "/"
    query_str = "&".join("%s=%s" % (k, data[k]) for k in sorted(data))
    return s + query_str

def sign_str(key, s, method):
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
    return base64.b64encode(hmac_str)

if __name__ == '__main__':
    endpoint = "cvm.tencentcloudapi.com"
    data = {
        'Action': 'DescribeInstances',
        'InstanceId.0': 'ins-09dx96dg',
        'Limit': 20,
        'Nonce': 11886,
        'Offset': 0,
        'Region': 'ap-guangzhou',
        'SecretId': secret_id,
        'Timestamp': 1465185768, # int(time.time())
        'Version': '2017-03-12'
    }
    s = get_string_to_sign("GET", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
    print(data["Signature"])
    # 此处会实际调用，成功后可能产生计费
    # resp = requests.get("https://" + endpoint, params=data)
    # print(resp.url)
```



## 返回结果

最近更新时间：2018-10-12 16:25:31

### 正确返回结果

以云服务器的接口查看实例状态列表 (DescribeInstancesStatus) 2017-03-12 版本为例，若调用成功，其可能的返回如下为：

```
{
  "Response": {
    "TotalCount": 0,
    "InstanceStatusSet": [],
    "RequestId": "b5b41468-520d-4192-b42f-595cc34b6c1c"
  }
}
```

- Response 及其内部的 RequestId 是固定的字段，无论请求成功与否，只要 API 处理了，则必定会返回。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。
- 除了固定的字段外，其余均为具体接口定义的字段，不同的接口所返回的字段参见接口文档中的定义。此例中的 TotalCount 和 InstanceStatusSet 均为 DescribeInstancesStatus 接口定义的字段，由于调用请求的用户暂时还没有云服务器实例，因此 TotalCount 在此情况下的返回值为 0，InstanceStatusSet 列表为空。

### 错误返回结果

若调用失败，其返回值示例如下为：

```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated. Please check your signature is correct."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。

- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

## 公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误代码	错误描述
AuthFailure.InvalidSecretId	密钥非法（不是云API密钥类型）
AuthFailure.MFAFailure	MFA错误
AuthFailure.SecretIdNotFound	密钥不存在
AuthFailure.SignatureExpire	签名过期
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token错误
AuthFailure.UnauthorizedOperation	请求未CAM授权
DryRunOperation	DryRun操作，代表请求将会是成功的，只是多传了DryRun参数
FailedOperation	操作失败
InternalError	内部错误
InvalidAction	接口不存在
InvalidParameter	参数错误
InvalidParameterValue	参数取值错误
LimitExceeded	超过配额限制
MissingParameter	缺少参数错误
NoSuchVersion	接口版本不存在
RequestLimitExceeded	请求的次数超过了频率限制

错误代码	错误描述
ResourceInUse	资源被占用
ResourceInsufficient	资源不足
ResourceNotFound	资源不存在
ResourceUnavailable	资源不可用
UnauthorizedOperation	未授权操作
UnknownParameter	未知参数错误
UnsupportedOperation	操作不支持
UnsupportedProtocol	http(s)请求协议错误，只支持GET和POST请求
UnsupportedRegion	接口不支持所传地域

# 更新历史

最近更新时间：2018-10-12 16:25:30

## 第 1 次发布

发布时间：2018-10-11 19:44:12

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [AddDelayLiveStream](#)
- [DescribeLiveStreamOnlineInfo](#)
- [DescribeLiveStreamOnlineList](#)
- [DescribeLiveStreamPublishedList](#)
- [DescribeLiveStreamState](#)
- [DropLiveStream](#)
- [ForbidLiveStream](#)
- [ResumeDelayLiveStream](#)
- [ResumeLiveStream](#)

新增数据结构：

- [PublishTime](#)
- [StreamInfo](#)
- [StreamName](#)
- [StreamOnlineInfo](#)

# 延播管理相关接口

## 延迟播放

最近更新时间：2018-10-12 16:25:32

### 1. 接口描述

接口请求域名：live.tencentcloudapi.com。

对流设置延播

默认接口请求频率限制：100次/每秒。

### 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：AddDelayLiveStream
Version	是	String	公共参数，本接口取值：2018-08-01
Region	否	String	公共参数，本接口不需要传递此参数。
AppName	是	String	应用名称。
DomainName	是	String	您的加速域名。
StreamName	是	String	流名称。
DelayTime	是	Integer	延播时间，单位：秒，上限：600秒。

### 3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求ID，每次请求都会返回。定位问题时需要提供该次请求的RequestId。

## 4. 示例

### 示例1 请求示例

#### 输入示例

```
https://live.tencentcloudapi.com/?Action=AddDelayLiveStream
&DomainName=5000.livepush.myqcloud.com
&AppName=live
&StreamName=stream1
&DelayTime=30
&<公共请求参数>
```

#### 输出示例

```
{
  "Response": {
    "RequestId": "8e50cdb5-56dc-408b-89b0-31818958d424"
  }
}
```

## 5. 开发者资源

### API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

### SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

### 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
DryRunOperation	DryRun操作，代表请求将会是成功的，只是多传了DryRun参数
InternalError	内部错误
InternalError.CallOtherSvrError	调用内部服务错误。
InternalError.ConfigNotExist	配置不存在。
InternalError.GetBizidError	获取用户账号错误。
InternalError.GetStreamInfoError	获取流信息失败。
InternalError.NotPermmmitOperat	无权限操作。
InternalError.StreamStatusError	流状态异常。
InternalError.UpdateDataError	更新数据失败。
InvalidParameter	参数错误
InvalidParameterValue	参数取值错误
MissingParameter	缺少参数错误

# 恢复延播

最近更新时间：2018-10-12 16:25:32

## 1. 接口描述

接口请求域名：live.tencentcloudapi.com。

恢复延迟播放设置

默认接口请求频率限制：20次/每秒。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：ResumeDelayLiveStream
Version	是	String	公共参数，本接口取值：2018-08-01
Region	否	String	公共参数，本接口不需要传递此参数。
AppName	是	String	应用名称。
DomainName	是	String	您的加速域名。
StreamName	是	String	流名称。

## 3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求ID，每次请求都会返回。定位问题时需要提供该次请求的RequestId。

## 4. 示例

### 示例1 请求示例



### 输入示例

```
https://live.tencentcloudapi.com/?Action=ResumeDelayLiveStream
&DomainName=5000.livepush.myqcloud.com
&AppName=live
&StreamName=stream1
&<公共请求参数>
```

### 输出示例

```
{
  "Response": {
    "RequestId": "8e50cdb5-56dc-408b-89b0-31818958d424"
  }
}
```

## 5. 开发者资源

### API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

### SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

### 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
DryRunOperation	DryRun操作，代表请求将会是成功的，只是多传了DryRun参数
FailedOperation	操作失败
InternalError	内部错误
InternalError.CallOtherSvrError	调用内部服务错误。
InternalError.ConfigNotExist	配置不存在。
InternalError.GetBizidError	获取用户账号错误。
InternalError.GetStreamInfoError	获取流信息失败。
InternalError.GetUpstreamInfoError	获取直播源信息错误。
InternalError.NotPermmittOperat	无权限操作。
InternalError.StreamStatusError	流状态异常。
InternalError.UpdateDataError	更新数据失败。
InvalidParameter	参数错误
InvalidParameterValue	参数取值错误
LimitExceeded	超过配额限制
MissingParameter	缺少参数错误
ResourceInUse	资源被占用
ResourceInsufficient	资源不足
ResourceNotFound	资源不存在
ResourceUnavailable	资源不可用
UnauthorizedOperation	未授权操作
UnknownParameter	未知参数错误
UnsupportedOperation	操作不支持

# 直播流管理相关接口

## 查询在线推流信息列表

最近更新时间：2018-10-12 16:25:33

### 1. 接口描述

接口请求域名：live.tencentcloudapi.com。

查询在线推流信息列表

默认接口请求频率限制：100次/每秒。

### 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeLiveStreamOnlineInfo
Version	是	String	公共参数，本接口取值：2018-08-01
Region	否	String	公共参数，本接口不需要传递此参数。
PageNum	是	Integer	取得第几页。 默认值：1
PageSize	是	Integer	分页大小。 最大值：100。 取值范围：1~100 之前的任意整数。 默认值：10
Status	否	Integer	0:未开始推流 1:正在推流 2:服务出错 3:已关闭。
StreamName	否	String	流名称。

### 3. 输出参数

参数名称	类型	描述
PageNum	Integer	分页的页码。
PageSize	Integer	每页大小
TotalNum	Integer	符合条件的总个数。
TotalPage	Integer	总页数。
StreamInfoList	Array of <a href="#">StreamInfo</a>	流信息列表
RequestId	String	唯一请求ID，每次请求都会返回。定位问题时需要提供该次请求的RequestId。

## 4. 示例

### 示例1 请求示例

#### 输入示例

```
https://live.tencentcloudapi.com/?Action=DescribeLiveStreamOnlineInfo
&PageNum=1
&PageSize=10
&Status=1
&<公共请求参数>
```

#### 输出示例

```
{
  "Response": {
    "PageNum": 1,
    "PageSize": 10,
    "RequestId": "6d7080fe-7cc5-40eb-4d44-fecdd7a33cc5",
    "StreamInfoList": [
      {
        "AppName": [
          "live"
        ],
        "CreateMode": "1",
        "CreateTime": "2018-07-13 14:48:23",
        "Status": 1,
        "StreamId": "10905947996257011227",
```

```
"StreamName": "teststream",
"WaterMarkId": "0"
},
],
"TotalNum": 1,
"TotalPage": 1
}
}
```

## 5. 开发者资源

### API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

### SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

### 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
InternalServerError	内部错误
InternalServerError.CallOtherSvrError	调用内部服务错误。

错误码	描述
InternalServerError.ConfigNotExist	配置不存在。
InternalServerError.InvalidUser	账号信息错误。
InvalidParameter	参数错误
InvalidParameterValue	参数取值错误

# 查询直播中的流

最近更新时间：2018-10-12 16:25:33

## 1. 接口描述

接口请求域名：live.tencentcloudapi.com。

返回正在直播中的流列表

默认接口请求频率限制：100次/每秒。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeLiveStreamOnlineList
Version	是	String	公共参数，本接口取值：2018-08-01
Region	否	String	公共参数，本接口不需要传递此参数。
DomainName	是	String	您的加速域名。
AppName	否	String	应用名称。
PageNum	否	Integer	取得第几页，默认1。
PageSize	否	Integer	每页大小，最大100。 取值：1~100之前的任意整数。 默认值：10

## 3. 输出参数

参数名称	类型	描述
TotalNum	Integer	符合条件的总个数。
TotalPage	Integer	总页数。

参数名称	类型	描述
PageNum	Integer	分页的页码。
PageSize	Integer	每页显示的条数。
OnlineInfo	Array of <a href="#">StreamOnlineInfo</a>	正在推送流的信息列表
RequestId	String	唯一请求ID，每次请求都会返回。定位问题时需要提供该次请求的RequestId。

## 4. 示例

### 示例1 请求示例

#### 输入示例

```
https://live.tencentcloudapi.com/?Action=DescribeLiveStreamOnlineList
&DomainName=5000.livepush.myqcloud.com
&AppName=live
&PageNum=1
&PageSize=10
&<公共请求参数>
```

#### 输出示例

```
{
  "Response": {
    "OnlineInfo": [
      {
        "PublishTimeList": [
          {
            "PublishTime": "2017-10-24T12:00:00Z"
          }
        ],
        "StreamName": "5000_abcdefg"
      }
    ],
    "PageNum": 1,
    "PageSize": 10,
    "RequestId": "8e50cdb5-56dc-408b-89b0-31818958d424",
    "TotalNum": 100,
    "TotalPage": 10
  }
}
```



```
}  
}
```

## 5. 开发者资源

### API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

### SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

### 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
InternalError	内部错误
InternalError.CallOtherSvrError	调用内部服务错误。
InternalError.ConfigNotExist	配置不存在。
InternalError.GetBizidError	获取用户账号错误。
InternalError.GetStreamInfoError	获取流信息失败。

错误码	描述
InternalServerError.GetUpstreamInfoError	获取直播源信息错误。
InternalServerError.NotPermmmitOperat	无权限操作。
InternalServerError.StreamStatusError	流状态异常。
InternalServerError.UpdateDataError	更新数据失败。
InvalidParameter	参数错误
InvalidParameterValue	参数取值错误
MissingParameter	缺少参数错误

# 查询推过流的流列表

最近更新时间：2018-10-12 16:25:33

## 1. 接口描述

接口请求域名：live.tencentcloudapi.com。

返回已经推过流的流列表

默认接口请求频率限制：100次/每秒。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeLiveStreamPublishedList
Version	是	String	公共参数，本接口取值：2018-08-01
Region	否	String	公共参数，本接口不需要传递此参数。
DomainName	是	String	您的域名。
EndTime	是	String	结束时间。 UTC 格式，例如：2016-06-30T19:00:00Z。 EndTime 和 StartTime 之间的间隔不能超过 30 天。
StartTime	是	String	起始时间。 UTC 格式，例如：2016-06-29T19:00:00Z。
AppName	否	String	直播流所属应用名称。
PageNum	否	Integer	取得第几页。 默认值：1
PageSize	否	Integer	分页大小。 最大值：100。 取值范围：1~100 之前的任意整数。 默认值：10

### 3. 输出参数

参数名称	类型	描述
PublishInfo	Array of <a href="#">StreamName</a>	推流记录信息。
PageNum	Integer	分页的页码。
PageSize	Integer	每页大小
TotalNum	Integer	符合条件的总个数。
TotalPage	Integer	总页数。
RequestId	String	唯一请求ID，每次请求都会返回。定位问题时需要提供该次请求的RequestId。

### 4. 示例

#### 示例1 请求示例

##### 输入示例

```
https://live.tencentcloudapi.com/?Action=DescribeLiveStreamPublishedList
&DomainName=5000.livepush.myqcloud.com
&AppName=live
&PageNum=1
&PageSize=10
&StartTime=2015-06-25T03:30:50Z
&EndTime=2015-12-26T03:30:50Z
&<公共请求参数>
```

##### 输出示例

```
{
  "Response": {
    "PageNum": 1,
    "PageSize": 10,
    "PublishInfo": [
      {
        "StreamName": "5000_abcdefg"
      }
    ]
  }
}
```

```
],
"RequestId": "8e50cdb5-56dc-408b-89b0-31818958d424",
"TotalNum": 100,
"TotalPage": 10
}
}
```

## 5. 开发者资源

### API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

### SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

### 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
InternalServerError	内部错误
InternalServerError.CallOtherSvrError	调用内部服务错误。
InternalServerError.ConfigNotExist	配置不存在。

错误码	描述
InternalError.GetBizidError	获取用户账号错误。
InternalError.GetStreamInfoError	获取流信息失败。
InternalError.GetUpstreamInfoError	获取直播源信息错误。
InternalError.NotPermmmitOperat	无权限操作。
InternalError.StreamStatusError	流状态异常。
InternalError.UpdateDataError	更新数据失败。
InvalidParameter	参数错误
InvalidParameterValue	参数取值错误
MissingParameter	缺少参数错误

# 查询流状态

最近更新时间：2018-10-12 16:25:33

## 1. 接口描述

接口请求域名：live.tencentcloudapi.com。

返回直播中、无推流或者禁播等状态

默认接口请求频率限制：100次/每秒。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeLiveStreamState
Version	是	String	公共参数，本接口取值：2018-08-01
Region	否	String	公共参数，本接口不需要传递此参数。
AppName	是	String	应用名称。
DomainName	是	String	您的加速域名。
StreamName	是	String	流名称。

## 3. 输出参数

参数名称	类型	描述
StreamState	String	流状态
RequestId	String	唯一请求ID，每次请求都会返回。定位问题时需要提供该次请求的RequestId。

## 4. 示例

## 示例1 请求示例

### 输入示例

```
https://live.tencentcloudapi.com/?Action=DescribeLiveStreamState
&DomainName=5000.livepush.myqcloud.com
&AppName=live
&StreamName=stream1
&<公共请求参数>
```

### 输出示例

```
{
  "Response": {
    "RequestId": "8e50cdb5-56dc-408b-89b0-31818958d424",
    "StreamState": "active"
  }
}
```

## 5. 开发者资源

### API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

### SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

### 命令行工具

- [Tencent Cloud CLI 3.0](#)



## 6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
InternalServerError	内部错误
InternalServerError.CallOtherSvrError	调用内部服务错误。
InternalServerError.ConfigNotExist	配置不存在。
InternalServerError.GetBizidError	获取用户账号错误。
InternalServerError.GetStreamInfoError	获取流信息失败。
InternalServerError.GetUpstreamInfoError	获取直播源信息错误。
InternalServerError.NotPermmmitOperat	无权限操作。
InternalServerError.StreamStatusError	流状态异常。
InternalServerError.UpdateDataError	更新数据失败。
InvalidParameter	参数错误
InvalidParameterValue	参数取值错误
MissingParameter	缺少参数错误

# 断开直播流

最近更新时间：2018-10-12 16:25:32

## 1. 接口描述

接口请求域名：live.tencentcloudapi.com。

断开推流连接，但可以重新推流

默认接口请求频率限制：100次/每秒。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DropLiveStream
Version	是	String	公共参数，本接口取值：2018-08-01
Region	否	String	公共参数，本接口不需要传递此参数。
StreamName	是	String	流名称。
DomainName	是	String	您的加速域名。
AppName	是	String	应用名称。

## 3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求ID，每次请求都会返回。定位问题时需要提供该次请求的RequestId。

## 4. 示例

### 示例1 请求示例

### 输入示例

```
https://live.tencentcloudapi.com/?Action=DropLiveStream
&DomainName=5000.livepush.myqcloud.com
&AppName=live
&StreamName=stream1
&<公共请求参数>
```

### 输出示例

```
{
  "Response": {
    "RequestId": "8e50cdb5-56dc-408b-89b0-31818958d424"
  }
}
```

## 5. 开发者资源

### API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

### SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

### 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
InternalError	内部错误
InternalError.CallOtherSvrError	调用内部服务错误。
InternalError.ConfigNotExist	配置不存在。
InternalError.GetBizidError	获取用户账号错误。
InternalError.GetStreamInfoError	获取流信息失败。
InternalError.GetUpstreamInfoError	获取直播源信息错误。
InternalError.NotPermmmitOperat	无权限操作。
InternalError.StreamStatusError	流状态异常。
InternalError.UpdateDataError	更新数据失败。
InvalidParameter	参数错误
InvalidParameterValue	参数取值错误
MissingParameter	缺少参数错误

# 禁播直播流

最近更新时间：2018-10-12 16:25:32

## 1. 接口描述

接口请求域名：live.tencentcloudapi.com。

禁止某条流的推送，可以预设某个时刻将流恢复。

默认接口请求频率限制：100次/每秒。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：ForbidLiveStream
Version	是	String	公共参数，本接口取值：2018-08-01
Region	否	String	公共参数，本接口不需要传递此参数。
AppName	是	String	应用名称。
DomainName	是	String	您的加速域名。
StreamName	是	String	流名称。
ResumeTime	否	String	恢复流的时间。UTC 格式，例如：2018-11-29T19:00:00Z。 UTC 时间，格式：2018-08-08T17:37:00Z。

## 3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求ID，每次请求都会返回。定位问题时需要提供该次请求的RequestId。

## 4. 示例

### 示例1 请求示例

#### 输入示例

```
https://live.tencentcloudapi.com/?Action=ForbidLiveStream
&DomainName=5000.livepush.myqcloud.com
&AppName=live
&StreamName=stream1
&ResumeTime=2018-11-24T12:00:00Z
&<公共请求参数>
```

#### 输出示例

```
{
  "Response": {
    "RequestId": "8e50cdb5-56dc-408b-89b0-31818958d424"
  }
}
```

## 5. 开发者资源

### API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

### SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

### 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
InternalError	内部错误
InternalError.CallOtherSvrError	调用内部服务错误。
InternalError.ConfigNotExist	配置不存在。
InternalError.GetBizidError	获取用户账号错误。
InternalError.GetStreamInfoError	获取流信息失败。
InternalError.GetUpstreamInfoError	获取直播源信息错误。
InternalError.NotPermmmitOperat	无权限操作。
InternalError.StreamStatusError	流状态异常。
InternalError.UpdateDataError	更新数据失败。
InvalidParameter	参数错误
InvalidParameterValue	参数取值错误
MissingParameter	缺少参数错误

# 恢复直播流

最近更新时间：2018-10-12 16:25:32

## 1. 接口描述

接口请求域名：live.tencentcloudapi.com。

恢复某条流的推送。

默认接口请求频率限制：100次/每秒。

## 2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：ResumeLiveStream
Version	是	String	公共参数，本接口取值：2018-08-01
Region	否	String	公共参数，本接口不需要传递此参数。
AppName	是	String	应用名称。
DomainName	是	String	您的加速域名。
StreamName	是	String	流名称。

## 3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求ID，每次请求都会返回。定位问题时需要提供该次请求的RequestId。

## 4. 示例

### 示例1 请求示例



### 输入示例

```
https://live.tencentcloudapi.com/?Action=ResumeLiveStream
&DomainName=5000.livepush.myqcloud.com
&AppName=live
&StreamName=stream1
&<公共请求参数>
```

### 输出示例

```
{
  "Response": {
    "RequestId": "8e50cdb5-56dc-408b-89b0-31818958d424"
  }
}
```

## 5. 开发者资源

### API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

### SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

### 命令行工具

- [Tencent Cloud CLI 3.0](#)

## 6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
InternalError	内部错误
InternalError.CallOtherSvrError	调用内部服务错误。
InternalError.ConfigNotExist	配置不存在。
InternalError.GetBizidError	获取用户账号错误。
InternalError.GetStreamInfoError	获取流信息失败。
InternalError.GetUpstreamInfoError	获取直播源信息错误。
InternalError.NotPermmmitOperat	无权限操作。
InternalError.StreamStatusError	流状态异常。
InternalError.UpdateDataError	更新数据失败。
InvalidParameter	参数错误
InvalidParameterValue	参数取值错误
MissingParameter	缺少参数错误

# 数据结构

最近更新时间：2018-10-12 16:25:30

## PublishTime

推流时间

被如下接口引用：DescribeLiveStreamOnlineList.

名称	类型	描述
PublishTime	String	推流时间 UTC 格式，例如：2018-06-29T19:00:00Z。

## StreamInfo

推流信息

被如下接口引用：DescribeLiveStreamOnlineInfo.

名称	类型	描述
AppName	String	直播流所属应用名称
CreateMode	String	创建模式
CreateTime	String	创建时间，如：2018-07-13 14:48:23
Status	Integer	流状态
StreamId	String	流id
StreamName	String	流名称
WaterMarkId	String	水印id

## StreamName

流名称列表

被如下接口引用：DescribeLiveStreamPublishedList.

名称	类型	描述
StreamName	String	流名称。

## StreamOnlineInfo

查询当前正在推流的信息

被如下接口引用：DescribeLiveStreamOnlineList.

名称	类型	描述
StreamName	String	流名称。
PublishTimeList	Array of <a href="#">PublishTime</a>	推流时间列表

# 云直播 API 2017

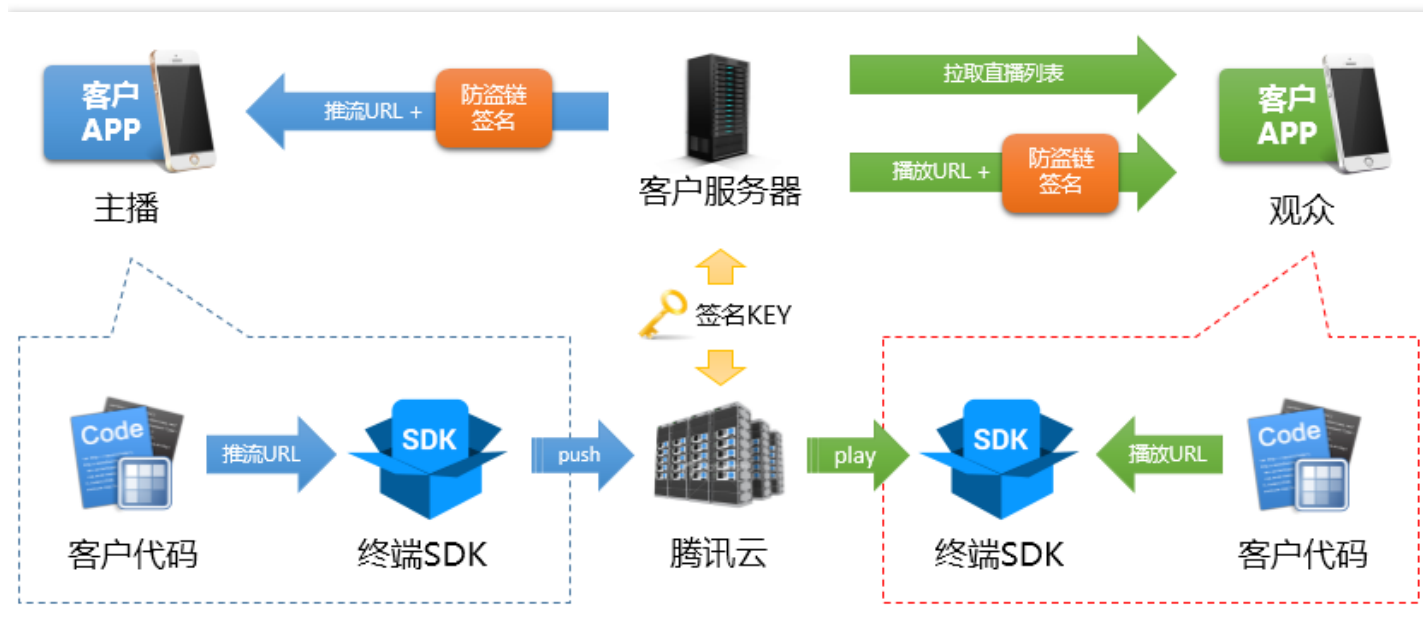
## 概述

最近更新时间：2018-07-23 19:32:45

云端 API 围绕的是如何让您能够对您的直播流进行管理，腾讯云提供了两种模式，即直播码模式（新）和频道模式（旧）。

### 直播码接入模式

直播码模式主要是为方便客户后台直接管理直播流而设计的，2016 年 5 月份之前只开放给了国内几个知名的直播平台，由于接入成本低且可靠性高而成为了腾讯云直播服务的主流接入方案，该方案的设计思路就是地址生成和直播流尽量由您自行掌握，因此有很强的自由度和定制型，是游戏直播、秀场直播等平台类直播场景的首选接入方案。



### 频道(托管)模式

频道模式的操作方式是为手动管理直播频道量身定制的，如果您是做单场次的直播活动，比较适合这种方式：

- **创建频道**：开始活动前，先创建一个频道，然后获得推流地址，并将播放地址传播出去。
- **关闭频道**：直播结束后，或者有意外情况发生时，可以随时关闭直播频道，也就是所谓的禁播。

该模式也提供了服务端 API 支持，但接入成本较直播码模式要高出很多，且稳定性不及直播码。

### 两种模式的区别

对比项	直播码模式	频道模式
-----	-------	------

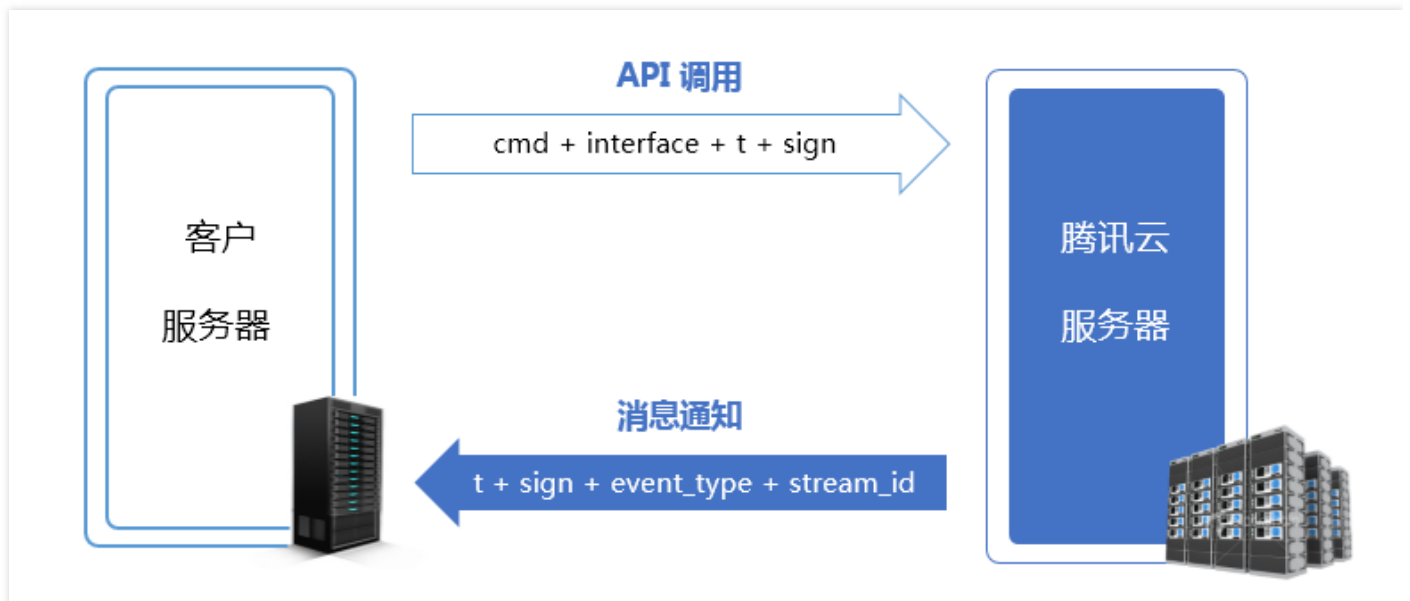
对比项	直播码模式	频道模式
推流地址	您可以自由指定，无需跟腾讯云后台通讯	通过创建频道的方式在腾讯云后台生成
防盗链	支持推流和播放防盗链（播放防盗链配置请联系客服）	支持推流防盗链，FLV 和 HLS 播放防盗链（rtmp 播放防盗链暂不支持）
API 列表	<a href="#">直播流管理 API</a>	适合手动操作，但也提供了 <a href="#">API</a>
消息通知	流状态变化时会以 JSON 格式的消息通知给您	不支持
适用场景	适合服务器对接，比如秀场直播	适合手工操作，比如一场直播活动

# 直播码模式概述

最近更新时间：2018-10-10 18:35:31

您的服务器与腾讯云服务器的信息同步可以通过两种方式组合实现：

- **API 调用**：腾讯云提供了一组直播码管理 API，包括状态查询和状态管理等功能，供您的后台服务器调用；
- **消息通知**：腾讯云在直播流状态变更、录制文件生成等一系列事件发生时，能够以事件消息（JSON）的形式主动通知您的后台服务器，您只需要在腾讯云注册接收事件通知的回调 URL 即可实现。



## 调用方法

在您的服务端采用 HTTP 协议的 GET 请求方式（即调用参数拼接在 URL 中）进行调用即可，详细的调用方法在每个 API 的说明文档中都有示例参考，没有任何对接难度。

## 安全机制

由于对 API 的调用采用的是普通的 HTTP 协议（出于性能考虑），这就需要一套行之有效的办法来确保您的服务器与腾讯云后台之间的通讯安全。

所有直播码相关的云端 API 都采用了同一种安全检查机制，t + sign 校验：

- **t（过期时间）**：如果一个 API 请求或者通知中的 t 值所规定的时间已经过期，则可以判定这个请求或者通知为无效的，这样做可以防止网络重放攻击。t 的格式为 UNIX 时间戳，即从 1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的秒数。
- **sign（安全签名）**：sign = MD5(key + t)，即把加密 key 和 t 进行字符串拼接后计算一下 md5 值，这里的 key 即 CGI 调用 key，您在腾讯云直播管理 [控制台](#) 中可以进行设置：



## • 安全原理

由于 MD5 是不可逆的 HASH 算法，所以只要确保 KEY 不泄露，即使攻击者拿到很多对 t 和 sign 也无法反算出 KEY 值，进而无法进行伪装攻击。

## • 计算示例

比如我们现在的的时间是 2016-08-22 15:16:27，我们希望有效期是 1 分钟，也就是 2016-08-22 15:17:27 以后再收到携带这个 t 的请求或者通知即判定为非法的：

```
t = "2016-08-22 15:17:27" = 1471850187
```

假设我们的 key 是 5d41402abc4b2a76b9719d911017c592，那么我们计算的签名结果就是：

```
sign = MD5(5d41402abc4b2a76b9719d911017c5921471850187) = b17971b51ba0fe5916ddcd96692e9fb3
```

## 消息通知

详情参考腾讯云事件 [消息通知](#) 服务。



# API 概览

最近更新时间：2018-09-03 10:50:36

腾讯云提供了一组直播码管理 API，包括状态查询和状态管理等功能，供您的后台服务器调用。

## API 列表

API	功能介绍
<a href="#">Get_LiveStat</a>	统计信息查询 - 查询推流和播放相关信息
<a href="#">Get_LivePushStat</a>	统计信息查询 - 查询推流相关信息
<a href="#">Get_LivePlayStat</a>	统计信息查询 - 查询播放相关信息
<a href="#">Get_LivePushStatHistory</a>	获取推流历史信息
<a href="#">Get_LivePlayStatHistory</a>	获取播放统计历史信息
<a href="#">Live_Channel_GetStatus</a>	仅查询某条流的状态信息（旧版本接口）
<a href="#">Live_Channel_SetStatus</a>	对某条流实行禁播操作，主要用于鉴黄场景
<a href="#">Live_Tape_GetFilelist</a>	查询某条流在直播过程中的录制文件列表
<a href="#">Live_Channel_GetChannelList</a>	查询频道列表
<a href="#">Live_Channel_GetLiveChannelList</a>	查询直播中频道列表
<a href="#">mix_streamv2.start_mix_stream_advanced</a>	云端混流操作接口
<a href="#">channel_manager</a>	暂停并延迟恢复——可针对某路流禁止推流
<a href="#">Live_Tape_Start</a>	创建录制任务——可实现定时录制任务或者实时视频录制
<a href="#">Live_Tape_Stop</a>	结束录制任务

## 错误码

### 1.http 错误

错误码	含义	备注
403	Forbidden	接口为了安全考虑开启了校验，若使用浏览器验证发现该错误，可检查下 cookie 里是否含有skey

错误码	含义	备注
404	Not Found	查看请求时是否带上 host

## 2.接口通用返回错误

错误码	含义	备注
appid is invalid	appid 不合法，表示未开通该功能	

### 注意：

以上错误码针对本文 API 列表中的 API，不包括 [消息事件通知](#)。

# 操作类接口

## 事件消息通知

最近更新时间：2018-09-20 17:59:15

## 功能介绍

一条直播流的状态变化、一个新录制文件的生成、一张新截图文件的生成，这类事件都在腾讯云内部管理，但您的后台服务器可能也同样有获知的需求，这时您可以使用腾讯云的事件通知服务获知这些事件。

您可以在控制台注册一个来自您后台服务器的回调 URL 给腾讯云，当有事件发生时，腾讯云会通过 HTTP POST 的方式将新的事件投递给您的服务器，事件内容以 JSON 格式组织。

## 配置 URL

在【直播控制台】>【直播码接入】>【接入配置】开启直播码模式时，可以指定一个接收腾讯云通知的 URL，如下图：



## 消息组织格式

通知信息是以 JSON 格式进行组织的，然后放在 HTTP POST 协议体中，注意这里的 POST 格式的 ContentType 是 application/json，而不是 multipart/form-data，所以不要使用 PHP 或者 Java 里读取表单字段的函数来读取信息。

## 公共的头信息

如下的字段是每种类型的通知消息都一定会携带的：

字段名称	类型	含义	备注
t	string	有效时间	UNIX 时间戳（十进制）
sign	string	安全签名	MD5(KEY+t)
event_type	int	事件类型	目前可能值为：0、1、100、200
stream_id	string	直播码	标示事件源于哪一条直播流
channel_id	string	直播码	同 stream_id

### • stream\_id | channel\_id (直播码)

在直播码模式下，stream\_id 和 channel\_id 两个字段都是同一个值，有两个不同的名字主要是历史原因所致。

### • t (过期时间)

来自腾讯云的消息通知的默认过期时间是10分钟，如果一条通知消息中的 t 值所指定的时间已经过期，则可以判定这条通知无效，进而可以防止网络重放攻击。t 的格式为十进制UNIX时间戳，即从1970年1月1日（UTC/GMT的午夜）开始所经过的秒数。

### • sign (安全签名)

sign = MD5(key + t)：腾讯云把加密 key 和 t 进行字符串拼接后通过 MD5 计算得出 sign 值，并将其放在通知消息里，您的后台服务器在收到通知消息后可以根据同样的算法确认 sign 是否正确，进而确认消息是否确实来自腾讯云后台。

这里的加密 key 即 CGI 调用 key，您在腾讯云直播管理控制台开通直播码服务时可以指定，如下图所示：



### • event\_type (通知类型)

目前腾讯云支持三种消息类型的通知：0 — 断流；1 — 推流；100 — 新的录制文件已生成；200 — 新的截图文件已生成。

## 各类型消息体

### (1)推流 (0)断流

event\_type = 0 代表断流，event\_type = 1 代表推流，同时消息体会额外包含如下信息：

字段名称	含义	类型	备注	是否必须
appname	推流路径	string		Y
app	推流域名	string		Y
event_time	消息产生的时间	int	单位 s	Y
sequence	消息序列号，标识一次推流活动，一次推流活动会产生相同序列号的推流和断流消息	string		Y
node	Upload 接入点的 IP	String		Y
user_ip	用户推流 IP	String	Client_ip	Y
errcode	断流错误码	Int		N
errmsg	断流错误信息	String		N
stream_param	推流 url 参数	String		Y
push_duration	推流时长	String	单位 ms	N

示例：腾讯云通知直播流 ( 1234\_15919131751 ) 发生断流 ( event\_type=0 ) 事件。

```
{
  "app": "3954.livepush.myqcloud.com",
  "appname": "live",
  "channel_id": "16093425727656502238",
  "event_type": 0,
  "sign": "ab86d22870427e3f25bb1d9446b8f924",
  "stream_id": "3954_ea88f7495ba711e6a2cba4dcbef5e35a",
  "t": 1471256100,
  "event_time": 1471256200,
  "sequence": "5911795891871911817",
```

```

"node": "123.10.1.1",
"user_ip": "127.0.0.1",
"errcode": 0,
"errmsg": "OK",
"stream_param": ""
}
    
```

### 推流断流错误码

错误码	错误描述	错误原因
1	recv rtmp deleteStream	主播端主动断流
2	recv rtmp closeStream	主播端主动断流
3	recv() return 0	主播端主动断开 TCP 连接
4	recv() return error	主播端 TCP 连接异常
7	rtmp message large than 1M	收到流数据异常
其他错误码	直播服务内部异常	如需处理请联系腾讯商务人员或者 <a href="#">提交工单</a> ，联系电话：4009-100-100

### ( 100 ) 新录制文件

**event\_type = 100** 代表有新的录制文件生成，同时消息体会额外包含如下信息：

字段名称	含义	类型	备注	是否必须
video_id	vid	string	点播用 vid，在点播平台可以唯一定位一个点播视频文件	Y
video_url	下载地址	string	点播视频的下载地址	Y
file_size	文件大小	string		Y
start_time	分片开始时间戳	int	开始时间（unix 时间戳，由于 I 帧干扰，暂时不能精确到秒级）	Y
end_time	分片结束时间戳	int	结束时间（unix 时间戳，由于 I 帧干扰，暂时不能精确到秒级）	Y
file_id	file_id	string		Y
file_format	文件格式	string	flv, hls, mp4	Y

字段名称	含义	类型	备注	是否必须
vod2Flag	是否开启点播 2.0	Int	0 表示未开启, 1 表示开启	N
record_file_id	录制文件 ID	String	点播 2.0 开启时, 才会有这个字段	N
duration	推流时长	Int		Y
stream_param	推流 url 参数	string		Y

示例：一个 ID 为 9192487266581821586 的新的 flv 录制分片已经生成，播放地址

为：`HTTP://200025724.vod.myqcloud.com/200025724_ac92b781a22c4a3e937c9e61c2624af7.f0.flv`。

```
{
  "appid": 1252033264,
  "channel_id": "2519_2500647",
  "duration": 272,
  "end_time": 1496220894,
  "event_type": 100,
  "file_format": "flv",
  "file_id": "9031868222958931071",
  "file_size": 30045521,
  "record_file_id": "9031868222958931071",
  "sign": "c2e3bdc344ddb62ab05229d01672a79e",
  "start_time": 1496220622,
  "stream_id": "2519_2500647",
  "stream_param": "bizid=2519&record=hls|flv&txSecret=d5569fb19d1e858bf683b30c10dec908&txTime=592FBDD9&mix=layer:b;session_id:709036962551160107;t_id:1",
  "t": 1496221502,
  "video_id": "200011683_481565e0befe4e44903839aebe370ef6",
  "video_url": "HTTP://1252033264.vod2.myqcloud.com/d7a4cabbvodgzp1252033264/0257ade99031868222958931071/f0.flv"
}
```

## ( 200 ) 新截图文件

**event\_type = 200** 代表有新的截图图片生成，同时消息体会额外包含如下信息：

字段名称	含义	类型	备注	是否必须
pic_url	图片地址	string	不带域名的路径	Y
create_time	截图时间戳	int	截图时间戳 ( unix 时间戳, 由于 I 帧干扰, 暂时不能精确到秒级 )	Y

字段名称	含义	类型	备注	是否必须
pic_full_url	截图全路径	String	需要带上域名	Y

示例：直播流“2016090090936”在腾讯云生成了一张新的截图图片：

```
{
  "channel_id": "2016090090936",
  "create_time": 1473645788,
  "event_type": 200,
  "pic_url": "/2016-09-12/2016090090936-screenshot-10-03-08-1280x720.jpg", //文件路径名
  "sign": "8704a0297ab7fdd0d8d94f8cc285cbb7",
  "stream_id": "2016090090936",
  "t": 1473646392
}
```

#### 图片下载地址：

1.由于历史原因，以前开启截图功能需要您自己申请 cos 服务，我们给您返回的 pic\_url 不是真正的图片下载地址只是下载路径，真正的下载地址是需要拼接的，拼接方法是：

- 下载前缀：`http://(cos_bucketname)-(cos_appid).file.myqcloud.com/`
- 下载路径：`/2016-09-12/2016090090936-screenshot-10-03-08-1280x720.jpg`
- 完整 URL：`http://(cos_bucketname)-(cos_appid).file.myqcloud.com/2016-09-12/2016090090936-screenshot-10-03-08-1280x720.jpg`

其中 cos\_appid 和 cos\_bucketname 是您在腾讯云开通 [对象存储服务](#) 的信息。

2.当前回调新信息中增一个图片URL 字段 pic\_full\_url 以便您直接得到完整的图片 URL，为了不影响原有业务，原有回调信息字段均无变化（即 pic\_url 字段依旧存在）。

3.以前您的腾讯云账号需要开通 cos 服务才能使用截图功能，当前该方式已经不适用，现在无需您自己申请 cos 服务，目前开通截图功能的方式有两种：a)控制台上开启截图功能；b)联系我们配置相关信息后再调用接口开启截图功能。

## 通知可靠性

很多客户会担心消息丢了怎么办，比如客户的服务器宕机了一下会儿，消息会不会丢失呢？

腾讯云后台目前的消息可靠性保证机制是基于简单重传实现的，即如果一条通知消息没有成功发送到您指定的回调 URL，腾讯云会反复重试 3 次，每次重试的时间间隔是 60 秒。



那怎么确认消息是已经送达您的服务器了呢？这里是需要您的协助的：当您的服务器成功收到一条 http 事件通知消息时，请回复：

```
json
```

```
// 在收到消息通知的http请求里返回错误码 0 以代表您已经成功收到了消息，从而避免腾讯云反复重复通知  
{ "code":0 }
```

代表：“嗯，我（客户服务器）已经接收到通知了，请（腾讯云）不要再不断地发消息给我”。

# 开启关闭推流

最近更新时间：2018-08-01 21:02:12

## 1. 接口描述

### • 接口

#### Live\_Channel\_SetStatus :

对一条直播流执行禁用、断流和允许推流操作。**禁用**表示不能再继续使用该流 ID 推流；如果正在推流，则推流会被中断，中断后不可再次推流。**断流**表示中断正在推的流，断流后可以再次推流。**允许推流**表示启用该流 ID，允许用该流 ID 推流。

### • 地址

API 调用地址为：`http://fcgi.video.qcloud.com/common_access`

### • 用途

主要用于鉴黄时的禁播场景，比如您如果在后台发现某个主播有涉黄或者反动内容，可以随时断流或者禁用这条流。

### • 说明

一条直播流一旦被设置为【禁用】状态，推流链路将被腾讯云主动断开，并且后续的推流请求也会被拒绝，一条流最长禁用 3 个月，超过 3 个月，禁用失效。

## 2. 输入参数

参数名	参数含义	类型	备注	是否必需
appid	客户 ID	int	即直播 APPID，用于区分不同客户的身份	Y
interface	接口名称	string	Live_Channel_SetStatus	Y
t	<a href="#">有效时间</a>	int	UNIX 时间戳（十进制）	Y
sign	<a href="#">安全签名</a>	string	MD5(key+t)	Y
Param.s.channel_id	直播码	string		Y
Param.n.status	开关状态	int	0 表示禁用，1 表示允许推流，2 表示断流	Y
Param.s.path	路径	string	多路径用户使用	N
Param.s.domain	域名	string	多域名用户使用	N

### 注意：

有些早期提供的 API 中直播码参数被定义为 channel\_id，新的 API 则称直播码为 stream\_id，仅历史原因而已。

### 3. 输出结果

参数名	参数含义	类型	备注
ret	返回码	int	0：成功；其他值：失败
message	错误信息	string	错误信息

### 4. 调用示例

目标：考虑到直播码为 8888\_test123 的直播流内容涉及违规内容，对其进行禁播操作。

组成部分	示例内容
接口URL	http://fcgi.video.qcloud.com/common_access?
appid	1234
interface	Live_Channel_SetStatus
Param.s.channel_id	8888_test123
Param.n.status	0
t	1471850187
sign	b17971b51ba0fe5916ddcd96692e9fb3

```
// copy时请去掉美化排版用的不可见换行符，否则可能出现“appid is invalid”等url拼装错误
URL = http://fcgi.video.qcloud.com/common_access?
appid=1234&interface=Live_Channel_SetStatus
&Param.s.channel_id=8888_test123
&Param.n.status=0
&t=1471850187&sign=b17971b51ba0fe5916ddcd96692e9fb3
```

# 云端混流

最近更新时间：2018-07-24 15:12:19

## 接口描述

### http 协议请求接口 cgi

```
http://fcgi.video.qcloud.com/common_access
```

### 通过 URI 传递鉴权参数

```
http://fcgi.video.qcloud.com/common_access?appid=1252500000&interface=Mix_StreamV2&t=t&sign=sign
```

#### 参数说明：

参数名称	参数含义	输入类型	备注	是否必填
appid	客户 ID	int	填写直播 APPID，用于区分不同客户的身份	Y
interface	接口名称	string	混流接口名称固定填写：Mix_StreamV2	Y
t	有效时间	int	UNIX 时间戳，即从 1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的秒数；这个字段表示的是请求过期时间，请您在获取当前时间（秒）的基础上加 60 秒偏移	Y
sign	安全签名	string	sign = MD5(key + t)，即把加密 key 和 t 进行字符串拼接后，计算一下 md5 值。这里的 key 是您在腾讯云直播管理控制台中设置的 API 鉴权 key	Y

#### 安全签名 sign 计算方法：

```
key = "40328529ca4381a80c6ecf2e6aa57438" //API鉴权key  
t = 1490858347 //t 过期时间  
key + t = "40328529ca4381a80c6ecf2e6aa574381490858347" //key 和 t 进行字符串拼接  
sign = MD5 ( key + t ) = "7f29ed83c61b77de1b0d66936fd4fd44" //对拼接后的字符串计算MD5
```

#### HTTP 请求说明

```
POST /common_access&interface=mix_streamv2.start_mix_stream_advanced&sign=xxxxxxxx&appid=1252500000 HTTP/1.0
```

```
Content-Length: xxx
```

```
...
```

## 通过 POST 方法传递混流 body

示例:

```
{
  "timestamp":int(time.time()), # UNIX 时间戳数
  "eventId":int(time.time()), # 取随机数即可, 标识一次网络请求
  "interface":
  {
    "interfaceName":"Mix_StreamV2", # 固定值
    "para":
    {
      "app_id": appid, # 填写直播 APPID
      "interface": "mix_streamv2.start_mix_stream_advanced", # 固定值
      "mix_stream_session_id": "lewis_room", # 标识一次网络请求
      "output_stream_id": "stream_lewis01", # 填输出流 id
      "output_stream_type": 0, # 填输出流类型
      "output_stream_bitrate":1500, # 输出码率
      "input_stream_list":
      [
        # 背景画面
        {
          "input_stream_id":"stream_lewis01", # 流ID
          "layout_params":
          {
            "image_layer": 1 # 图层号, 背景填1;
          }
        },
        # 小画面1
        {
          "input_stream_id":"stream_lewis02", # 流ID
          "layout_params":
          {
            "image_layer": 2, # 图层标识号
            "image_width": 160, # 画面宽度
            "image_height": 240, # 画面高度
            "location_x": 380, # x偏移: 相对于背景画面左上角的横向偏移
            "location_y": 630 # y偏移: 相对于背景画面左上角的纵向偏移
          }
        }
      ]
    }
  }
}
```

```

    }
}
    
```

**必填参数说明:**

参数名称	参数含义	输入类型	备注	是否必填
timestamp	当前时间	int	取当前时间（秒）即可	Y
eventId	标识一次网络请求	int	取随机值即可	Y
interfaceName	接口标识	string	Mix_StreamV2，固定值	Y
app_id	直播 APPID	int	直播 APPID	Y
interface	混流接口名称	string	填 mix_streamv2.start_mix_stream_advanced 或 mix_streamv2.cancel_mix_stream	Y
mix_stream_session_id	一次混流操作标识	string	填写 80 字节以内，仅含字母、数字以及下划线的字符串	Y
output_stream_id	输出流 ID	string	指定输出流 ID	Y
input_stream_id	输入源 ID	string	指定输入源 ID	Y
image_layer	图层标识号	int	支持1-16 注意： 1)背景流（即大主播画面或画布）的 image_layer 填 1 2)纯音频混流，该参数也需填	Y

**选填参数说明:**

参数名称	参数含义	输入类型	备注	是否必填
output_stream_type	输出流类型	int	不填默认为0 当输出流为输入流list中的一条时，填写0； 当输出流并非输入流list中的流时，该值为1。 注意：对于使用1的情况，请联系客服人员咨询使用方法。	N

参数名称	参数含义	输入类型	备注	是否必填
output_stream_bitrate	输出码率	int	不填系统会自动判断	N
mix_stream_template_id	输入模版ID	int	不填默认为0。 两输入源支持10、20、30、40、50； 三输入源支持310、390、391； 四输入源支持410； 五输入源支持510、590； 六输入源支持610。	N
input_type	输入源类型	int	目前支持： 不填默认为0； 0表示输入源为音视频； 2表示输入源为图片； 3表示输入源为画布； 4表示输入源为音频； 5表示输入源为纯视频。	N
image_width	输入画面的宽度	double	像素建议在0-3000以内。 百分比建议0.01-0.99。	N
image_height	输入画面的高度	double	像素建议在0-3000以内。 百分比建议0.01-0.99。	N
location_x	x偏移	double	相对于大主播背景画面左上角的横向偏移。 像素建议在0-3000以内。 百分比建议0.01-0.99	N
location_y	y偏移	double	相对于大主播背景画面左上角的纵向偏移。 像素建议在0-3000以内。 百分比建议0.01-0.99	N
color	颜色	string	使用画布时填写，常用的颜色有： 红色：0xcc0033 黄色：0xcc9900 绿色：0xcccc33 蓝色：0x99CCFF 黑色：0x000000 白色：0xFFFFFFFF 灰色：0x999999。	N

参数名称	参数含义	输入类型	备注	是否必填
picture_id	图片的cosid	int	使用图片时填写，填入将图片上传cos后生成的id	N
crop_width	裁剪源画面的宽度	int	建议在0-3000以内。	N
crop_height	裁剪源画面的高度	int	建议在0-3000以内。	N
crop_x	相对裁剪源画面左上角横向偏移。	int	建议在0-3000以内。	N
crop_y	相对裁剪源画面左上角纵向偏移。	int	建议在0-3000以内。	N

## 接口返回

```
{"code":0, "message":"Success!", "event_id": "1527240138", "timestamp":1490079362}
```

## 参数说明:

参数名	参数含义	类型	备注
code	返回错误码	int	0表示成功，其他表示失败
message	错误信息	string	返回错误信息
timestamp	时间戳	long int	返回时间
event_id	请求id	int	网络请求标识

## 常见错误码说明:

错误码	原因	排查建议
-1	解析输入参数错误	检查请求体body json格式是否正确 检查interface name是否为 mix_streamv2.start_mix_stream_advanced 检查input_stream_list是否为空
-2	输入参数错误	检查appid、eventid是否为0 检查画面参数是否溢出
-3	流数目错误	检查输入流数目是否在[1, 16]范围内



错误码	原因	排查建议
-4	流参数错误	检查输入输出长宽在 ( 0 , 3000 ) 范围内 检查输入流数目是否在 ( 0,16]范围内 检查输入流是否携带layout_params 检查input_type是否支持 ( 合法数值 : 0 , 2 , 3 , 4 , 5 ) 检查流id长度是否满足 ( 1 , 80 )
-11	图层错误	检查图层个数与输入流个数是否一致 检查图层id是否重复 检查图层id是否在 ( 0 , 16]之间
-20	输入参数与接口不匹配	检查输入流条数是否匹配模板id 检查颜色参数是否正确
-21	混流输入流条数错误	检查输入流的条数是否至少为两条
-28	获取背景长宽失败	如果设置画布，检查画布的长宽是否设置 检查背景流是否存在 ( 推流后需等待5s再混流 )
-29	裁剪参数错误	检查裁剪位置是否超出流的长宽
-33	水印图片id错误	检查输入图片id是否设置
-34	获取水印图片url失败	检查图片是否上传成功，是否已经生成url
-300	输出流id已经被使用	检查当前输出流是否已经是另一个混流的输出流
-505	输入流无法在upload查到	重新推流后再试
-507	流长宽参数查询失败	检查画布宽、高是否设置 检查推流是否已经成功，建议推流后5s再开始混流
-508	输出流id错误	检查是否存在同样sessionid使用不同输出流id的情况
-10031	触发混流失败	建议推流后等待5s再混流
-30300 -31001 -31002	取消混流时sessionid不存在	检查sessionid是否存在
-31003	输出流id与session中输出流id不匹配	检查取消混流时填入的输出流id
-31004	输出流码率不合法	检查输出流码率是否在[1 , 50000]之间
其他	其他错误，请联系客服提供技术支持	

## 取消混流

支持两种接口取消混流：

1. 使用mix\_streamv2.cancel\_mix\_stream接口取消混流

示例:

```
{
  "timestamp":int(time.time()), # UNIX时间戳数
  "eventId":int(time.time()), # 混流事件ID，取时间戳即可
  "interface":
  {
    "interfaceName":"Mix_StreamV2", # 固定取值"Mix_StreamV2"
    "para":
    {
      "app_id": appid, # 填写直播APPID
      "interface": "mix_streamv2.cancel_mix_stream", # 取消混流使用mix_streamv2.cancel_mix_stream接口
      "mix_stream_session_id": "lewis_room", # 标识一次网络请求
      "output_stream_id": "stream_lewis01" # 输出流id
    }
  }
}
```

2. 使用mix\_streamv2.start\_mix\_stream\_advanced接口取消混流，需要填输入流id信息

示例:

```
{
  "timestamp":int(time.time()), # UNIX时间戳
  "eventId":int(time.time()), # 混流事件ID，取时间戳即可
  "interface":
  {
    "interfaceName":"Mix_StreamV2", # 固定取值"Mix_StreamV2"
    "para":
    {
      "app_id": appid, # 填写直播APPID
      "interface": "mix_streamv2.start_mix_stream_advanced", #接口名为"mix_streamv2.start_mix_stream_
advanced"
      "mix_stream_session_id": "lewis_room", # 标识一次网络请求
      "output_stream_id": "stream_lewis01",
      "input_stream_list":
      [
        {
          "input_stream_id": "stream_lewis01", #流ID
          "layout_params":
          {
            "image_layer": 1
          }
        }
      ]
    }
  }
}
```

```
}  
]  
}  
}  
}
```

#### 注意事项:

- 使用当前 sessionid 操作当前输入流即可取消混流，半分钟后可使用该 sessionid 操作其他输出流

## 云端混流功能说明

### 功能支持列表

- 最大支持同时**16**条流混流
- 支持混入**5**种输入源类型(音视频, 纯音频, 纯视频, 图片, 画布)
- 支持混流合成全新流
- 支持裁剪, 水印功能
- 支持模板配置
- 支持混流录制
- 支持自动混流
- 支持实时混流种类与位置切换
- 混流启动与取消无缝平滑过渡

### 常用模版说明

常用的模版有10、20、30、40、310、410、510、610。使用这八种模版时，输入流不需要填写位置和长宽参数，为**原始画面的等比例缩放**。只需要传入模版id即可。

模版**10**效果图：



模版20效果图：



模版30效果图：



模版40效果图：



模版310效果图：



模版410效果图：



模版510效果图：



模版610效果图：



### 特殊模版的说明

为了便于部分用户的同屏对比需求，目前定制了三个特殊的模版390、391以及590。

模版390效果图：



模版391效果图：





模版590效果图：



**注意事项:**

当使用这三个模版时，图层id为1的那个输入源必须传入画布参数，颜色可以自定义指定

**自定义位置参数的填写规则说明**

示意图：



**注意事项:**

- 位置参数 location\_x、location\_y 为小画面左上角相对背景画面左上角的绝对像素距离。

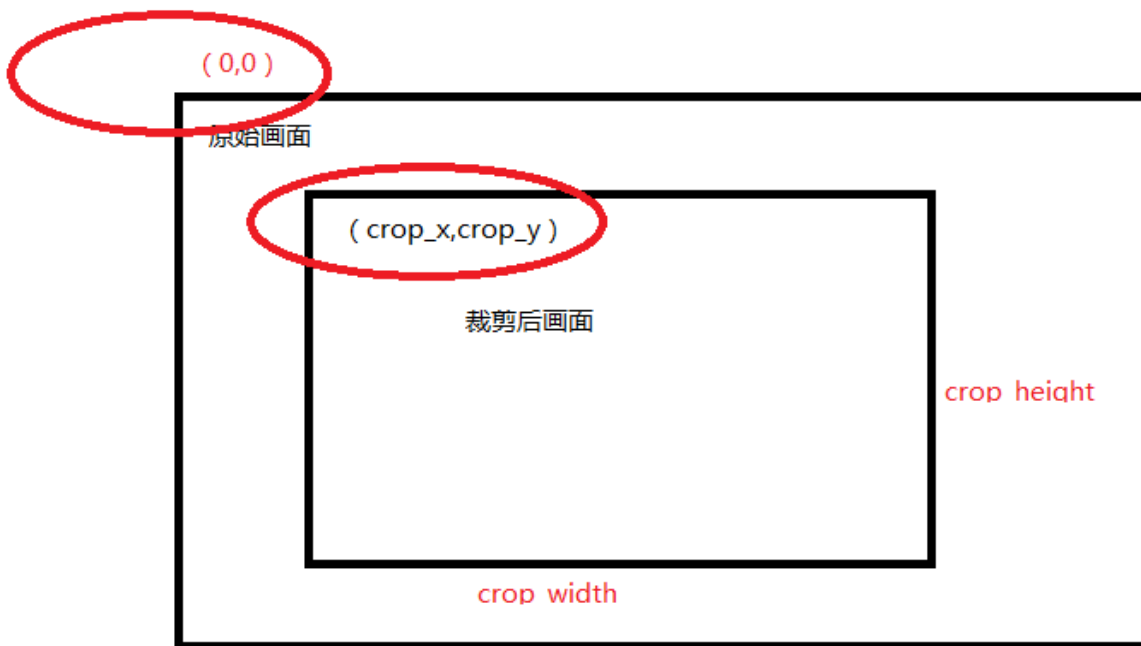
**云端裁剪**

**参数说明:**

```
{
  "input_stream_id": "5000_lewis03", # 流ID
  "layout_params":
  {
    "image_layer": 3, # 图层标识号
    "image_width": 320, # 画面宽度
    "image_height": 240, # 画面高度
    "location_x": 580, # x偏移：相对于背景画面左上角的横向偏移
    "location_y": 830 # y偏移：相对于背景画面左上角的纵向偏移
  }
  "crop_params":
  {
    "crop_width": 200, # 裁剪宽度
    "crop_height": 100, # 裁剪高度
  }
}
```

```
"crop_x":100, # 裁剪x偏移：相对于原画面左上角的横向偏移  
"crop_y":1 # 裁剪y偏移：相对于原画面左上角的纵向偏移  
}  
}
```

示意图：



注意事项：

- 输入的宽高、位置参数全部是像素绝对值，对于不同分辨率的流，需要不同处理。
- 位置参数 `crop_x`、`crop_y` 表示相对原始流左上角绝对像素的位置坐标。
- 如果 `layout_params` 中指定了 `image_width` 和 `image_height`，会将裁剪后的画面缩放到指定的宽高。示例中，会将从原始画面裁剪出的 `200 100` 的画面缩放成 `320 249` 输出。

## 自动混流

若需求较简单，只要实现一个大主播和一个小主播混流，可以使用自动混流功能。只需在原先的推流地址中加入自动混流参数 `mixv2=t_id:x;session_id:xxx;layer:x` 即可使用。

示例：

```
rtmp://1234.livepush.myqcloud.com/live/1234_lewis01?from=interactive&bizid=1234&mixv2=t_id:1;session_id:1234_lewis_room;layer:b
```

```
rtmp://1234.livepush.myqcloud.com/live/1234_lewis02?from=interactive&bizid=1234&mixv2=t_id:1;session_id:1234_lewis_room;layer:s
```

在两条流均推流成功后，会自动开始混流，输出流id为背景流id，示例中输出流id为1234\_lewis01

**参数说明**

参数	含义	取值	是否必填
t_id	模板id	只支持填1	Y
session_id	混流操作标识	填写 80 字节以内，仅含字母、数字以及下划线的字符串	Y
layer	图层	大主播图层，填 b 小主播图层，填 s	Y

**自动混流效果图**



**注意事项：**

- 自动混流只支持两条输入流
- 自动混流的取消方法与api取消混流方法一致
- 自动混流不支持自定义长宽和位置参数

## 混流问题 Q&A

### 1. 推流后混流，为什么会返回-505错误码？

推流后等待5s左右再混流。

### 2. 纯音频混流后听不到小主播声音？

检查纯音频流的input\_type是否填为4。

### 3. 连麦场景，输出流id不变，小主播流id变化，是否要取消混流后再重新混流？

不需要，直接用新的小主播id发新的混流请求即可。

### 4. 申请混流后，如果一直未取消混流，会出现什么情况？

混流会一直进行，直到收到取消混流命令。

### 5. 混流过程中，输入流突然断开会出现什么情况？

非背景流断开，断掉的流画面会停在最后一帧，背景流断开，则整个画面都会卡住。在15分钟内该流以同一流id重新推流成功，则自动恢复。

## DEMO下载

混流 demo 下载：请单击[此处](#)下载 demo。

# 暂停并延迟恢复

最近更新时间：2018-08-01 21:03:08

## 1. 接口描述

- **接口**

**channel\_manager**：暂停推流并延迟恢复接口。

- **地址**

API 调用地址为：[http://fcgi.video.qcloud.com/common\\_access](http://fcgi.video.qcloud.com/common_access)。

- **用途**

当某些原因不允许主播推流时，可调用此接口。

- **说明**

调用该接口可以暂停推某路流（即禁止推流），如果要恢复主播推流，可再次调用该接口或者设置一个恢复时间，到达指定时间后允许推流，最长禁止推流 3 个月，即禁止推流截止时间最长为当前时间往后 3 个月，如果超过 3 个月，则以 3 个月为准。

## 2. 输入参数

参数名	参数含义	类型	备注	是否必需
appid	客户 ID	int	即直播 appid，用于区分不同客户的身份	Y
interface	接口名称	string	如：channel_manager	Y
t	<a href="#">有效时间</a>	int	UNIX 时间戳（十进制）	Y
sign	<a href="#">安全签名</a>	string	MD5(key+t)	Y
Param.s.channel_id	直播码	string		Y
Param.n.abstime_end	禁播截止的时间戳	int	禁播截止的绝对时间，请填写 UNIX 时间戳（十进制），系统最多禁播三个月。	Y
Param.s.action	动作	string	断流：forbid；恢复推流：resume	Y
Param.s.path	路径	string	多路径用户使用	N
Param.s.domain	域名	string	多域名用户使用	N

### 3.输出结果

参数名	参数含义	类型	备注
ret	返回码	int	0：成功；其他值：失败
message	错误信息	string	错误信息

### 4.调用示例

目标：考虑到直播码为 8888\_test123 的直播流内容涉及违规内容，对其进行禁播操作。

组成部分	示例内容
接口 URL	http://fcgi.video.qcloud.com/common_access?
appid	1250000000
interface	channel_manager
Param.s.channel_id	8888_test123
Param.n.abstime_end	1499756910
t	1471850187
sign	b17971b51ba0fe5916ddcd96692e9fb3
Param.s.action	resume

```
// copy时请去掉美化排版用的不可见换行符，否则可能出现“appid is invalid”等url拼装错误
URL = http://fcgi.video.qcloud.com/common_access?
appid=1250000000
&interface=channel_manager
&Param.s.channel_id=8888_test123
&Param.n.abstime_end=1499756910
&t=1471850187
&sign=b17971b51ba0fe5916ddcd96692e9fb3
&Param.s.action=resume
```

# 创建录制任务

最近更新时间：2018-08-08 16:16:49

## 1. 接口描述

### • 接口

**Live\_Tape\_Start**：创建录制任务。

### • 地址

API 调用地址为：`http://fcgi.video.qcloud.com/common_access`

### • 说明

录制文件存放于点播平台。用户如需使用录制功能，需首先自行开通点播服务，录制文件存放后相关费用（含存储以及下行播放流量）按照点播平台计费方式收取，具体请参考 [对应文档](#)。

创建录制任务。该接口支持两种录制模式，定时录制任务和实时视频录制模式，定时任务录制由于需要传入任务开始时间导致可能会错过精彩视频的录制，实时视频录制功能则可满足出现精彩视频的瞬间同步录制。注意：调用接口超时设置应大于3秒，小于3秒重试以及频繁调用都有可能产生重复录制任务。

## 2. 输入参数

参数名	参数含义	类型	备注	是否必需
appid	客户 ID	int	即直播 appid，用于区分不同客户的身份	Y
interface	接口名称	string	填写Live_Tape_Start	Y
t	<a href="#">有效时间</a>	int	UNIX 时间戳（十进制）	Y
sign	<a href="#">安全签名</a>	string	MD5(key+t)	Y
Param.s.channel_id	频道 ID	string		Y
Param.s.start_time	任务开始时间	string	中国标准时间，需要 urlencode，如 2017-01-01 10:10:01	Y
Param.s.end_time	任务结束时间	string	中国标准时间，需要 urlencode，如 2017-01-01 10:10:01	Y



参数名	参数含义	类型	备注	是否必需
Param.n.task_sub_type	是否开启实时视频录制	int	1 - 开启；0 - 关闭，推荐使用实时视频录制。 a. 实时视频录制创建成功依赖主播活跃推流，调用接口成功则同步开始录制，并忽略任务开始时间参数。 b. 实时视频录制任务时长支持最大为 30 分钟，如果传入任务结束时间与当前时间差大于 30 分钟，则按 30 分钟计算。录制时长建议控制在 5 分钟以内。 c. 如果关闭实时录制，则需要传入任务开始时间参数，且结束时间与开始时间差不大于 1 天。	Y
Param.s.file_format	录制文件格式	string	默认 flv；可取值 flv、hls、mp4、aac	N
Param.s.record_type	录制文件类型	string	默认 video 当 record_type 取值"video"时，file_format 可以取值 "flv","hls", "mp4" 当 record_type 取值"audio"时，file_format 可以取值 "aac", "flv", "hls", "mp4"	N
Param.s.path	路径	string	多路径用户使用	N
Param.s.domain	域名	string	多域名用户使用	N
Param.s.mix_stream	是否开启输出流类型为1的混流录制	int	1-开启，0-关闭，该功能只针对于直播码模式，且混流输出流类型为1生效	N

### 3.输出结果

参数名	参数含义	类型	备注
ret	返回码	int	0：成功；其他值：失败
message	错误信息	string	错误信息
output	消息内容	array	详情见下

其中output的主要内容为：

字段名	含义	类型	备注

字段名	含义	类型	备注
task_id	任务 ID	int	64位无符整形

#### 4.调用示例

目标：查询直播码为 8888\_test123 的直播流在整个直播过程中所录制的文件列表。

组成部分	示例内容
接口 URL	http://fcgi.video.qcloud.com/common_access?
appid	1234
interface	Live_Tape_Start
Param.s.channel_id	8888_test123
Param.s.start_time	任务开始时间
Param.s.end_time	任务结束时间
t	1471850187
sign	b17971b51ba0fe5916ddcd96692e9fb3

// copy时请去掉美化排版用的不可见换行符，否则可能出现 “appid is invalid” 等url拼装错误

URL = http://fcgi.video.qcloud.com/common\_access?

appid=1234&interface=Live\_Tape\_Start

&Param.s.channel\_id=8888\_test123

&Param.s.start\_time=2017-05-20+10:00:00

&Param.s.end\_time=2017-05-20+10:30:00

&Param.n.task\_sub\_type=1

&t=1471850187&sign=b17971b51ba0fe5916ddcd96692e9fb3

注：Param.s.start\_time和Param.s.end\_time需要encode编码。2017-05-20+10:00:00是2017-05-20 10:00:00的encode编码;2017-05-20+10:30:00是2017-05-20 10:30:00的encode编码。

# 结束录制任务

最近更新时间：2018-07-10 15:43:02

## 1. 接口描述

### • 接口

**Live\_Tape\_Stop**：结束录制任务。

### • 地址

API 调用地址为：`http://fcgi.video.qcloud.com/common_access`

### • 说明

结束录制任务。

## 2. 输入参数

参数名	参数含义	类型	备注	是否必需
appid	客户 ID	int	即直播 appid，用于区分不同客户的身份	Y
interface	接口名称	string	如：Get_LivePushStat	Y
t	<a href="#">有效时间</a>	int	UNIX 时间戳（十进制）	Y
sign	<a href="#">安全签名</a>	string	MD5(key+t)	Y
Param.s.channel_id	频道 ID	string		Y
Param.n.task_id	任务 ID	int		Y
Param.n.task_sub_type	是否开启实时视频录制	int	默认 0，1 表示开启小视频录制	N

## 3. 输出结果

参数名	参数含义	类型	备注
ret	返回码	int	0：成功；其他值：失败
message	错误信息	string	错误信息

## 4. 调用示例

目标：查询直播码为 8888\_test123 的直播流在整个直播过程中所录制的文件列表。

组成部分	示例内容
接口 URL	http://fcgi.video.qcloud.com/common_access?
appid	1234
interface	Live_Tape_Stop
Param.s.channel_id	8888_test123
Param.n.task_id	123
t	1471850187
sign	b17971b51ba0fe5916ddcd96692e9fb3

```
// copy 时请去掉美化排版用的不可见换行符，否则可能出现 “appid is invalid” 等 URL 拼装错误
URL = http://fcgi.video.qcloud.com/common_access?
appid=1234&interface=Live_Tape_Stop
&Param.s.channel_id=8888_test123
&Param.n.task_id=123
&t=1471850187&sign=b17971b51ba0fe5916ddcd96692e9fb3
```

# 延播

最近更新时间：2018-09-17 15:13:01

## 1. 接口描述

### • 接口

Live\_Set\_Delay：对直播流进行延播设置。

### • 地址

API 调用地址：[http://fcgi.video.qcloud.com/common\\_access](http://fcgi.video.qcloud.com/common_access)

### • 用途

延播功能主要用于直播流的延迟播放。

### • 说明

延播上限为 10 分钟；

若需取消延播、可设置 time\_length=0；

推流前设置延播，则推流时生效；

推流中设置延播，则当前流会在 time\_length 时间内处于黑屏状态，达到延播时间后恢复播放；

## 2. 输入参数

参数名	参数含义	类型	备注	是否必需
appid	客户 ID	int	即直播 APPID，用于区分不同客户的身份	Y
interface	接口名称	string	Live_Set_Delay	Y
t	有效时间	int	UNIX 时间戳（十进制）	Y
sign	安全签名	string	MD5(key+t)	Y
Param.s.channel_id	直播码	string	直播码 id	Y
Param.n.time_length	延播时长	int	单位秒，上限 600s	Y

## 3. 输出结果

参数名	参数含义	类型	备注
ret	返回码	int	0：成功；其他值：失败
message	错误信息	string	错误信息

## 4.调用示例

目标：对直播码为 8888\_test123 的直播流进行延播设置。

组成部分	示例内容
接口 URL	<code>http://fcgi.video.qcloud.com/common_access?</code>
appid	4
interface	Live_Set_Delay
Param.s.channel_id	5000_123
Param.n.time_length	30
t	1532488286
sign	0b27965db3faa78913b4f7e6037c0780

URL:

```
http://fcgi.video.qcloud.com/common_access?Param.n.time_length=30&Param.s.channel_id=5000_123
&sign=0b27965db3faa78913b4f7e6037c0780&appid=4&t=1532488286&interface=Live_Set_Delay
```

```
{"errmsg":"","message":"","output":{},"ret":0,"retcode":0}
```

# 查询类接口

## 查询直播状态

最近更新时间：2018-07-11 14:51:57

### 1. 接口描述

- 接口

**Live\_Channel\_GetStatus**：用于查询某条流是否处于正在直播的状态。

- 地址

API 调用地址为：`http://fcgi.video.qcloud.com/common_access`

- 用途

用于查询某条流是否处于正在直播的状态，其内部原理是基于腾讯云对音视频流的中断检测而实现的，因此实时性上可能不如 App 的主动上报这么快速和准确，但在进行直播流定时清理检查的时候可以作为一种不错的补充手段。

- 说明

如果要查询的推流直播码从来没有推过流，会返回 20601 错误码。

### 2. 输入参数

参数名	参数含义	类型	备注	是否必需
appid	客户 ID	int	即直播 appid，用于区分不同客户的身份	Y
interface	接口名称	string	如：Get_LivePushStat	Y
t	<a href="#">有效时间</a>	int	UNIX 时间戳（十进制）	Y
sign	<a href="#">安全签名</a>	string	MD5(key+t)	Y
Param.s.channel_id	直播码	string	一次只能查询一条直播流	Y

**注意：**

有些早期提供的 API 中直播码参数被定义为 channel\_id，新的 API 则称直播码为 stream\_id，仅历史原因而已。

### 3. 输出结果

参数名	参数含义	类型	备注
-----	------	----	----

参数名	参数含义	类型	备注
ret	返回码	int	0 : 成功 ; 其他值 : 失败
message	错误信息	string	错误信息
output	消息内容	array	详情见下

output 的主要内容为 :

字段名	含义	类型	备注
rate_type	码率	int	0 : 原始码率 ; 10 : 普清 ; 20 : 高清
recv_type	播放协议	int	1 : rtmp/flv ; 2 : hls ; 3 : rtmp/flv+hls
status	状态	int	0 : 断流 ; 1 : 开启 ; 3 : 关闭

#### 4.调用示例

目标 : 查询直播码为 8888\_test123 的直播流的当前状态是否是【正在直播中】。

组成部分	示例内容
接口 URL	<code>http://fcgi.video.qcloud.com/common_access?</code>
appid	1234
interface	Live_Channel_GetStatus
Param.s.channel_id	8888_test123
t	1471850187
sign	b17971b51ba0fe5916ddcd96692e9fb3

*// copy时请去掉美化排版用的不可见换行符, 否则可能出现 "appid is invalid" 等url拼装错误*

```
URL = http://fcgi.video.qcloud.com/common_access?
appid=1234&interface=Live_Channel_GetStatus
&Param.s.channel_id=8888_test123
&t=1471850187&sign=b17971b51ba0fe5916ddcd96692e9fb3
```



# 查询统计信息

最近更新时间：2018-07-10 16:44:47

## 1. 接口描述

### • 接口

**Get\_LiveStat**：查询指定直播流的推流和播放信息；

**Get\_LivePushStat**：仅返回推流统计信息以提高查询效率；

**Get\_LivePlayStat**：仅返回播放统计信息以提高查询效率。

### • 地址

API 调用地址为：`http://statcgi.video.qcloud.com/common_access`

### • 用途

查询某条直播流的统计信息（如观看人数、带宽、码率、帧率等）；

查询当前正在直播状态中的若干条直播流的统计信息（建议采用分页查询避免单次回包数据过大）。

### • 说明

统计数据均为查询时间点的瞬时统计数据，而并非历史累计数据；

如果目标流不在直播中，则返回结果中的 output 字段为空；

推流信息的统计数据每 **5 秒钟** 更新一次，也就是快于 5 秒的查询频率是浪费的；

播放信息的统计数据每 **1 分钟** 更新一次，也就是快于 60 秒的查询频率是浪费的。

### • BETA

统计接口目前尚处于 Beta 阶段，并未全员放开，未开通即调用此接口会收到【cmd is invalid】提示，如您急需请联系我们。

## 2. 输入参数

参数名	参数含义	类型	备注	是否必需
cmd	客户 ID	int	即直播 APPID，用于区分不同客户的身份	Y
interface	接口名称	string	如：Get_LivePushStat	Y
t	<a href="#">有效时间</a>	int	UNIX 时间戳（十进制）	Y
sign	<a href="#">安全签名</a>	string	MD5(key+t)	Y
Param.n.page_no	分页页码	int	从 1 开始，默认为 1	N
Param.n.page_size	分页大小	int	1~300，默认为 300	N
Param.s.stream_id	直播码	string	如不设置 stream_id：查询所有正在直播中的流	N
Param.s.pull_domain	拉流域名	string	即播放域名，如果不填则返回所有域名的播放数据	N

### 3.输出结果

#### Get\_LiveStat

参数名	参数含义	类型	备注
ret	返回码	int	0：成功；其他值：失败
message	错误信息	string	错误信息
output	消息内容	array	详情见下

其中output的主要内容为：

字段名	含义	类型	备注
stream_count	所有在线的直播流数量	int	
stream_info	直播流统计信息	array	详情见下
total_bandwidth	当前账号在查询时间点的总带宽	double	单位：Mbps
total_online	当前账号在查询时间点的总在线人数	int	单位：人

其中 stream\_info 的主要内容为：

字段名	含义	类型	备注
stream_name	直播码	string	-
bandwidth	该直播流的瞬时带宽占用	double	单位：Mbps
online	该直播流的瞬时在线人数	int	单位：人
client_ip	推流客户端 IP	string	-
server_ip	接流服务器 IP	string	-
fps	瞬时推流帧率	int	-
speed	瞬时推流码率	int	bps

#### Get\_LivePushStat

Get\_LivePushStat 的 output 部分是 Get\_LiveStat 的一个子集：

字段名	含义	类型	备注
-----	----	----	----

字段名	含义	类型	备注
stream_count	所有在线的直播流数量	int	
stream_info	直播流统计信息	array	详情见下

其中 stream\_info 的主要内容为：

字段名	含义	类型	备注
stream_name	直播码	string	-
client_ip	推流客户端 IP	string	-
server_ip	接流服务器 IP	string	-
fps	瞬时推流帧率	int	-
speed	瞬时推流码率	int	bps

#### Get\_LivePlayStat

Get\_LivePushStat 的 output 部分是 Get\_LiveStat 的另一个子集：

字段名	含义	类型	备注
stream_count	所有在线的直播流数量	int	
stream_info	直播流统计信息	array	详情见下
total_bandwidth	当前账号在查询时间点的总带宽	double	单位：Mbps
total_online	当前账号在查询时间点的并发连接数	int	单位：人

其中 stream\_info 的主要内容为：

字段名	含义	类型	备注
stream_name	直播码	string	-
bandwidth	该直播流的瞬时带宽占用	double	单位：Mbps
online	该直播流的瞬时并发连接数	int	单位：人

## 4.调用示例

目标：查询当前账户名下所有正在直播的视频流的推流和播放信息。

组成部分	示例内容
接口 URL	http://statcgi.video.qcloud.com/common_access?
cmd	1234
interface	Get_LiveStat
Param.n.page_no	1
Param.n.page_size	20
t	1471850187
sign	b17971b51ba0fe5916ddcd96692e9fb3

```

// copy时请去掉美化排版用的不可见换行符，否则可能出现 “cmd is invalid” 等url拼装错误
URL = http://statcgi.video.qcloud.com/common_access?
cmd=1234&interface=Get_LiveStat
&Param.n.page_no=1
&Param.n.page_size=20
&t=1471850187&sign=b17971b51ba0fe5916ddcd96692e9fb3
    
```

# 查询录制文件

最近更新时间：2018-08-03 16:56:58

## 1. 接口描述

### • 接口

**Live\_Tape\_GetFilelist**：用于查询某条直播流某段时间内生成的录制文件。

### • 地址

API 调用地址为：`http://fcgi.video.qcloud.com/common_access`

### • 说明

由于文件的落地时间对您不可知，所以这种主动查询接口在调用时机上并不是特别好掌握，更推荐使用[被动事件通知](#) ( event\_type = 100 ) 机制。

注意事项：

1. 查询频道录制文件结果信息，包含文件名，任务id，文件id，录制消息。
2. 最长查询2个月内的录制文件。

## 2. 输入参数

参数名	参数含义	类型	备注	是否必需
appid	客户 ID	int	即直播 appid，用于区分不同客户的身份(无法识别大写)	Y
interface	接口名称	string	如：Get_LivePushStat	Y
t	<a href="#">有效时间</a>	int	UNIX 时间戳 ( 十进制 )	Y
sign	<a href="#">安全签名</a>	string	MD5(key+t)	Y
Param.s.channel_id	直播码	string		Y
Param.n.page_no	分页页码	int	从 1 开始，默认为 1	N
Param.n.page_size	分页大小	int	1~100，默认为 10	N
Param.s.sort_type	排序方式	string	asc 表示升序，desc 表示降序，默认 asc	N
Param.s.start_time	查询开始时间	string	格式为：2016-12-10 00:00:00	N
Param.s.end_time	查询结束时间	string	格式为：2016-12-10 00:00:00。结束时间距开始时间一天以内，且不能跨天	N

**注意：**

有些早期提供的 API 中直播码参数被定义为 channel\_id，新的 API 则称直播码为 stream\_id，仅历史原因而已。

### 3.输出结果

参数名	参数含义	类型	备注
ret	返回码	int	0：成功；其他值：失败
message	错误信息	string	错误信息
output	消息内容	object	详情见下

其中 output 的主要内容为：

字段名	含义	类型	备注
all_count	分片总个数	int	
file_list	分片文件信息	array	详情见下

其中 file\_list 的主要内容为：

字段名	含义	类型	备注
vid	点播的 vid	string	如果为空，则使用 record_file_url
start_time	分片开始时间	string	由于 I 帧位置原因，并不能精确到秒
end_time	分片结束时间	string	由于 I 帧位置原因，并不能精确到秒
file_id	点播 file_id	string	用点播 API 换取播放 URL 需要输入此参数
record_file_url	播放地址	string	如果不为空，则使用该地址；如果为空，采用 vid，按拼接格式拼接地址

**注意：**

vid 拼接格式，录制分片请求 url 格式：`http://(点播bizid).vod.myqcloud.com/(vid).f0.flv`

## 4.调用示例

目标：查询直播码为 8888\_test123 的直播流在整个直播过程中所录制的文件列表。

组成部分	示例内容
接口 URL	<code>http://fcgi.video.qcloud.com/common_access?</code>
appid	1234
interface	Live_Tape_GetFilelist
Param.s.channel_id	8888_test123
Param.n.page_no	1
Param.n.page_size	20
t	1471850187
sign	b17971b51ba0fe5916ddcd96692e9fb3

*// copy时请去掉美化排版用的不可见换行符，否则可能出现 “appid is invalid” 等url拼装错误*

URL = `http://fcgi.video.qcloud.com/common_access?`

`appid=1234&interface=Live_Tape_GetFilelist`

`&Param.s.channel_id=8888_test123`

`&Param.n.page_no=1`

`&Param.n.page_size=20`

`&t=1471850187&sign=b17971b51ba0fe5916ddcd96692e9fb3`

`&Param.s.start_time=2016-12-10 00:00:00`

`&Param.s.end_time=2016-12-10 01:00:00`

# 查询频道列表

最近更新时间：2018-10-09 15:54:52

## 1. 接口描述

### • 接口

**Live\_Channel\_GetChannelList**：查询频道列表。

### • 地址

API 调用地址为：[http://fcgi.video.qcloud.com/common\\_access](http://fcgi.video.qcloud.com/common_access)

### • 用途

在直播码模式下，用于查询当前频道列表。

### • 说明

可以查询特定状态的频道列表，如可过滤当前处于开启状态的频道。

## 2. 输入参数

参数名	参数含义	类型	备注	是否必需
appid	客户 ID	int		Y
interface	接口名	string		Y
t	有效截止时间戳	int		Y
sign	签名	string	md5(key+ 有效截止时间戳)	Y
Param.n.status	0：表示断流，1：表示开启，3：表示关闭	int	默认是不过滤	N
Param.n.page_no	分页页码	int	从 1 开始，默认为 1	N
Param.n.page_size	分页大小	int	10~100，默认为 10	N
Param.s.order_field	排序字段	string	可选字段：create_time，默认为 create_time	N
Param.n.order_by_type	排序方法	int	0：表示正序，1：表示倒序	N

## 3. 输出结果

参数名	参数含义	类型	备注
ret	返回码	int	0：表示成功；其他值：表示失败
message	错误信息	string	错误描述



参数名	参数含义	类型	备注
output	消息内容	array	

其中 output 部分结构为：

字段名	含义	类型	备注
all_count	总个数	int	
channel_list	列表	array	

其中 channel\_list 部分结构为：

参数名	参数含义	类型	备注
channel_id	频道 ID	string	
channel_name	频道名	string	
create_mode	创建模式	string	
create_time	创建时间	string	
path	推流路径	Array(String)	
status	状态	Int	0-无输入，1-正常，2-服务错误，3-关闭，4-禁止
watermark_id	水印id	String	

#### 4.调用示例

目标：查询该帐号下当开启的频道列表

组成部分	示例内容
接口 URL	http://fcgi.video.qcloud.com/common_access?
Param.n.status	1
Param.n.page_no	1
Param.n.page_size	20
Param.s.order_field	create_time
Param.n.order_by_type	0

```
// copy时请去掉美化排版用的不可见换行符，否则可能出现“appid is invalid”等url拼装错误  
URL = http://fcgi.video.qcloud.com/common_access?  
appid=1234&interface=Live_Channel_GetChannelList  
&t=1471850187&sign=b17971b51ba0fe5916ddcd96692e9fb3  
&Param.n.status=1&Param.n.page_no=1&Param.n.page_size=20  
&Param.s.order_field=create_time&Param.n.order_by_type=0
```

# 查询直播中频道列表

最近更新时间：2018-07-11 14:57:07

## 1. 接口描述

### • 接口

**Live\_Channel\_GetLiveChannelList**：查询频道列表。

### • 地址

API 调用地址为：[http://fcgi.video.qcloud.com/common\\_access](http://fcgi.video.qcloud.com/common_access)

### • 用途

在直播码模式下，用于查询直播中频道列表。

## 2. 输入参数

参数名	参数含义	类型	备注	是否必需
appid	客户 ID	int		Y
interface	接口名	string		Y
t	有效截止时间戳	int		Y
sign	签名	string	md5(key+ 有效截止时间戳)	Y

## 3. 输出结果

参数名	参数含义	类型	备注
ret	返回码	int	0：表示成功；其他值：表示失败
message	错误信息	string	错误描述
output	消息内容	array	

其中 output 部分结构为：

字段名	含义	类型	备注
all_count	总个数	int	
channel_list	列表	array	

其中 channel\_list 部分结构为：

参数名	参数含义	类型	备注
channel_id	直播码 ID	string	

#### 4.调用示例

目标：查询该帐号下直播中频道列表。

组成部分	示例内容
接口 URL	<code>http://fcgi.video.qcloud.com/common_access?</code>

```
// copy时请去掉美化排版用的不可见换行符，否则可能出现“appid is invalid”等url拼装错误
URL = http://fcgi.video.qcloud.com/common_access?
appid=1234&interface=Live_Channel_GetChannelList
&t=1471850187&sign=b17971b51ba0fe5916ddcd96692e9fb3
```

# 获取计费带宽数据

最近更新时间：2018-07-10 16:56:51

## 1. 接口描述

### • 接口

**Get\_BillingBandwidth**：用于查询帐号计费带宽数据。

### • 地址

API 调用地址为：[http://statcgi.video.qcloud.com/common\\_access](http://statcgi.video.qcloud.com/common_access)

◦ 使用该接口需要后台配置，如需调用该接口，请联系腾讯商务人员或者 [提交工单](#)，联系电话：4009-100-100。

## 2. 输入参数

参数名	参数含义	类型	备注	是否必需
cmd	业务appid	int	填写直播appid，用于区分不同客户身份	Y
interface	接口名称	string	Get_BillingBandwidth	Y
t	<a href="#">有效时间</a>	int	UNIX 时间戳（十进制）	Y
sign	<a href="#">安全签名</a>	string	MD5(key+t)	Y
Param.n.start_time	查询起始时间	int	时间戳	Y
Param.n.end_time	查询结束时间	int	时间戳	Y
Params.s.domain	域名	string	域名	N
Param.n.home_foreign	国内外的数据	uint	0：不区分国内外，查询总数据（default） 1：只查询国内数据 2：只查询国外数据	N
Param.n.get_top_bd	获取峰值带宽	Uint	0：不是获取峰值带宽和流量，而是列表（default） 1：获取峰值带宽和流量	N

## 3. 输出结果

参数名	参数含义	类型	备注
-----	------	----	----

参数名	参数含义	类型	备注
ret	返回码	int	0表示成功；其他值表示失败
message	错误信息	string	错误描述
output	消息内容	array	无

其中 output 的主要内容为：

字段名	含义	类型	备注
total_info	计费总带宽统计信息	array	5分钟粒度
domain	域名	string	传入参后才有

其中 total\_info 的主要内容为：

字段名	含义	类型	备注
time	统计时间	string	
bandwidth	带宽	double	Mbps
flux	流量	double	MB

## 4.调用示例

目标：查询帐号计费带宽数据。

组成部分	示例内容
接口 URL	statcgi.video.qcloud.com/common_access?
cmd	1251059556
interface	Get_BillingBandwidth
Param.n.start_time	1481299200
Param.n.end_time	1481302800
Param.n.home_foreign	1
Param.s.domain	8888.liveplay.myqcloud.com
t	1471850187

组成部分	示例内容
sign	b17971b51ba0fe5916ddcd96692e9fb3

```
// copy时请去掉美化排版用的不可见换行符，否则可能出现 “appid is invalid” 等url拼装错误
URL = http://statcgi.video.qcloud.com/common_access?
cmd=1251059556
&interface=Get_BillingBandwidth
&Param.s.domain=8888.liveplay.myqcloud.com
&t=1471850187&sign=b17971b51ba0fe5916ddcd96692e9fb3
&Param.n.start_time=1481299200
&Param.n.end_time=1481302800
```

# 统计类接口

## 获取推流历史信息

最近更新时间：2018-07-10 17:04:20

### 1. 接口描述

- 接口

**Get\_LivePushStatHistory**：获取推流历史信息。

- 地址

API 调用地址为：[http://statcgi.video.qcloud.com/common\\_access](http://statcgi.video.qcloud.com/common_access)

- 说明

可获取指定时间段内推流信息；

推流信息的统计数据每 5 秒钟更新一次。

- 使用该接口需要后台配置，如需调用该接口，请联系腾讯商务人员或者 [提交工单](#)，联系电话：4009-100-100。

### 2. 输入参数

参数名	参数含义	类型	备注	是否必需
cmd	业务 APPID	int	填写直播appid，用于区分不同客户身份	Y
interface	接口名	string		Y
t	有效截止时间戳	int		Y
sign	签名	string	md5(key+有效截止时间戳)	Y
Param.n.start_time	查询起始时间	int	3 天内的数据时间戳	Y
Param.n.end_time	查询终止时间	int	建议查询跨度不大于 2 小时时间戳	Y
Param.s.stream_id	流 ID	string		Y

### 3. 输出结果

参数名	参数含义	类型	备注	其他说明
ret	返回码	int	0：成功；其他值：失败	
message	错误信息	string	错误信息	
output	消息内容	array	详情见下	



其中 output 的主要内容为：

参数名	参数含义	类型	备注	其他说明
stream_info	推流时间流统计信息	list	数组（列表）	

其中 stream\_info 部分结构为：

参数名	参数含义	类型	备注	其他说明
time	推流时刻	string		
client_ip	推流客户端 IP	string		
server_ip	接流服务器 IP	string		
fps	推流帧率	int		
speed	推流码率	int		单位：bps

#### 4.调用示例

目标：查询直播码为 8888\_test123 的直播流在整个直播过程中所录制的文件列表。

组成部分	示例内容
接口 URL	<code>http://statcgi.video.qcloud.com/common_access?</code>
cmd	1234
interface	Get_LivePushStatHistory
Param.n.start_time	1453279831
Param.n.end_time	1453279835
Param.s.stream_id	1234_xxx
t	1471850187
sign	b17971b51ba0fe5916ddcd96692e9fb3

```
// copy时请去掉美化排版用的不可见换行符，否则可能出现“cmd is invalid”等url拼装错误
URL = http://statcgi.video.qcloud.com/common_access?
cmd=1234&interface=Get_LivePushStatHistory
&Param.n.start_time =1453279831
&Param.n.end_time =1453279835
```

```
&Param.s.stream_id =1234_xxx  
&t=1471850187  
&sign=b17971b51ba0fe5916ddcd96692e9fb3
```

# 获取播放统计历史信息

最近更新时间：2018-07-10 17:05:16

## 1. 接口描述

### • 接口

**Get\_LivePlayStatHistory**：获取播放统计历史信息。

### • 地址

API 调用地址为：`http://statcgi.video.qcloud.com/common_access`

### • 说明

- 可获取指定时间段内播放信息。
- 播放信息的统计数据每 1 分钟更新一次。
- 该接口支持直播码模式，同时也支持频道模式。
- 使用该接口需要后台配置，如需调用该接口，请联系腾讯商务人员或者 [提交工单](#)，联系电话：4009-100-100。

## 2. 输入参数

参数名	参数含义	类型	备注	是否必需
cmd	业务 ID	int	请填写直播appid，用于区分不同客户身份	Y
interface	接口名	string		Y
t	有效截止时间戳	int		Y
sign	签名	string	md5(key+有效截止时间戳)	Y
Param.n.start_time	查询起始时间	int	时间戳 15 天内的数据	Y
Param.n.end_time	结束时间	int	时间戳建议时间跨度不大于 2 小时	Y
Param.s.stream_id	流 ID	string	不填就是获取总带宽	N
Param.s.domain	域名	String	若不填，取这个 APPID 下的总数据，需要填写 cname 前的原始播放域名	N

## 3. 输出结果

参数名	参数含义	类型	备注	其他说明
ret	返回码	int	0：成功；其他值：失败	

参数名	参数含义	类型	备注	其他说明
message	错误信息	string	错误信息	
output	消息内容	array	详情见下	-

其中 output 的主要内容为：

参数名	参数含义	类型	备注	其他说明
stat_info	直播流的统计信息	array		
sum_info	流量和累加信息	array		必输出项
domain	域名	string		非必输出项
stream_id	流 ID	string		非必输出项

其中 stat\_info 部分结构为：

参数名	参数含义	类型	备注	其他说明
time	统计时间	string		
bandwidth	带宽	double		单位：Mbps
online	在线人数	int		
flux	流量	double		单位：MB

其中 sum\_info 部分结构为（汇总部分）：

参数名	参数含义	类型	备注	其他说明
sum_flux	流量总和	double	必输出项	单位：MB

#### 4.调用示例

目标：查询直播码为 8888\_test123 的直播流在整个直播过程中所录制的文件列表。

组成部分	示例内容
接口 URL	<code>http://statcgi.video.qcloud.com/common_access?</code>
cmd	1234

组成部分	示例内容
interface	Get_LivePlayStatHistory
Param.n.start_time	1453279831
Param.n.end_time	1453279835
Param.s.stream_id	1234_xxx
Param.s.domain	www.123test.com
t	1471850187
sign	b17971b51ba0fe5916ddcd96692e9fb3

```

// copy时请去掉美化排版用的不可见换行符，否则可能出现“cmd is invalid”等url拼装错误
URL = http://statcgi.video.qcloud.com/common_access?
cmd=1234&interface=Get_LivePlayStatHistory
&Param.n.start_time =1453279831
&Param.n.end_time =1453279835
&Param.s.stream_id =1234_xxx
&Param.s.domain=www.123test.com
&t=1471850187
&sign=b17971b51ba0fe5916ddcd96692e9fb3
    
```

# 频道模式（旧）

## 调用方式

## 请求结构

最近更新时间：2018-07-10 17:26:19

### 1. 服务地址

腾讯云直播服务使用的域名访问地址为：live.api.qcloud.com。

### 2. 通信协议

腾讯云API的所有接口均通过HTTPS进行通信，提供高安全性的通信通道。

### 3. 请求方法

同时支持 POST 和 GET 请求，需要注意不能混合使用。即如果使用 GET 方式，则参数均从 Querystring 取得；如果使用 POST 方式，则参数均从 Request Body 中取得，Querystring 中的参数将忽略。两种方式参数格式规则相同，一般使用GET，当参数字符串过长时使用POST，请见各接口详细描述。

### 4. 字符编码

均使用UTF-8编码。

### 5. API请求结构

名称	描述	备注
API入口	API调用的 Webservice 入口	https://live.api.qcloud.com/v2/index.php
公共参数	每个接口都包含的通用参数	详见本文第六节公共参数部分
指令名称	API要执行的指令的名称，这里使用Action指定，例如Action=CreateLVBChannel	完整的指令请参见 <a href="#">API概览</a>

指令参数

每个特定的指令需要的参数

详见每个指令的接口文档

## 6. 公共参数

公共参数是用于标识用户和接口鉴权目的的参数, 如非必要, 在每个接口单独的接口文档中不再对这些参数进行说明, 但每次请求均需要携带这些参数, 才能正常发起请求。

名称	类型	描述	必选
Action	String	接口指令的名称, 例如: DescribeDomains	是
Region	String	区域参数, 用来标识希望操作哪个区域的实例。可选: bj:北京 gz:广州 sh:上海 hk:香港 ca:北美	是
Timestamp	UInt	当前UNIX时间戳	是
Nonce	UInt	随机正整数, 与 Timestamp 联合起来, 用于防止重放攻击	是
SecretId	String	由腾讯云平台上申请的标识身份的 SecretId 和 SecretKey, 其中 SecretKey 会用来生成 Signature 具体参考 <a href="#">接口鉴权</a> 页面	是
Signature	String	请求签名, 用来验证此次请求的合法性, 具体参考 <a href="#">接口鉴权</a> 页面	是

一个典型的接口请求如下, Action=DescribeInstance表示查询云服务器实例的详情。

```
https://domain/v2/index.php?Action=DescribeInstances
&SecretId=xxxxxxx
&Region=gz
&Timestamp=1402992826
&Nonce=345122
&Signature=mysignature
&instanceId=101
```

其中instanceId为指令参数, 其余为通用参数。

# 后台 SDK

最近更新时间：2018-07-24 15:27:33

## ## 调用方法

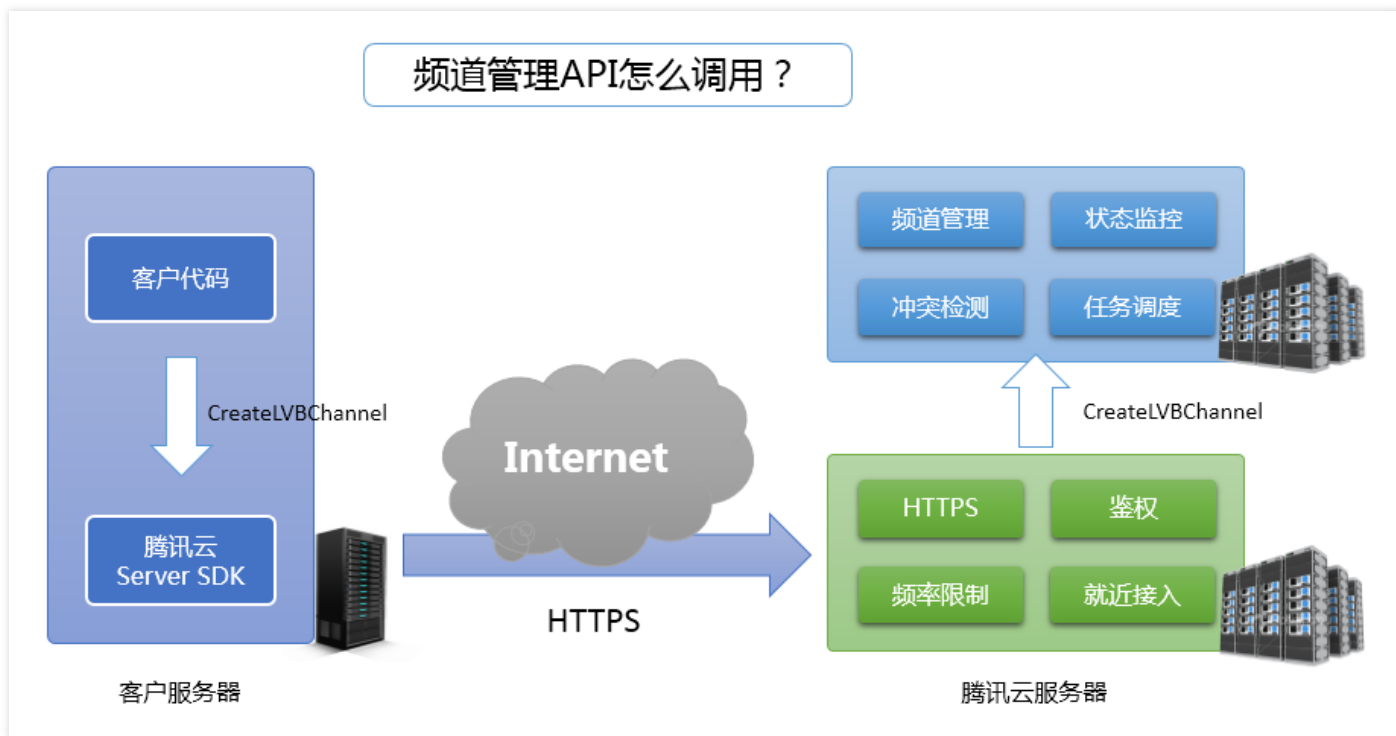
如何连接到腾讯云服务器是个首先要解决的问题，目前腾讯云服务端集群内部是分为两个层级的：

- **接入层：**

负责外部请求调用的接入，比如1分钟100次的频率调用限制，比如对于腾讯云客户的服务器身份验证（基于SecretId和SecretKey等等）。

- **功能层：**

躲在接入层后面，既然接入层已经完成了安全确认和攻击屏蔽，功能层就只需要做自己的功能处理就行了，频道管理、文件查询等基本功能都是在这个部分实现的。



您的服务器要调用腾讯云的服务端API，首先要过接入层这一关，她会要求：

- 您的请求协议必须是HTTPS
- 您的请求头必须符合腾讯云的接入格式
- 您的请求头必须要包含正确的SecretId和Signature

所以...



## 服务端SDK

推荐您使用[腾讯云服务端SDK](#)，目前提供了如下几种语言的版本：

- PHP
- Python
- Java
- .Net
- Node.js

这些SDK的目的只有一个：帮您以最简单的方式连上腾讯云服务器，而且确保中间链路安全可靠不被攻击。

## 错误排查

调用腾讯云服务端API，您可能会得到如下的这种错误码，它分成两个部分，前面一个code部分是腾讯云**接入层的错误码**，后面的**message**代表了错误的详细信息。

### unicode

```
{"code":4600,"message":"\u534f\u8bae\u4e0d\u652f\u6301\u00c0\u8bf7\u53c2\u8003\u6587\u6863\u8bf4\u660e\u3002"}
```

有些平台下拿到的message是如上的unicode形式的字符串，遇到这种情况打开chrome浏览器转义一下就行了。

打开chrome浏览器按F12进入开发者模式，在右下角的Console面板里输入：

```
"\u534f\u8bae\u4e0d\u652f\u6301\u00c0\u8bf7\u53c2\u8003\u6587\u6863\u8bf4\u660e\u3002"
```

就能得到转以后的结果"协议不支持，请参考文档说明。"

### 双错误码

当然，正常情况下，您看到的信息提示一般都是标准中文的，如下所示：

```
{"code":5100,"message":"20108：频道处于非编辑状态"}
```

您会看到这里有两个错误码：

- **5100**：前面这个错误码是接入层返给您的错误码，比如4100含义是鉴权出现问题（即您填写在请求头里的的SecretId和Signature有问题导致腾讯云后台系统认证失败），这时您的请求不会被投递给功能层，在接入层就直接被拦住了。

5100这个错误码是比较特殊的，它的含义是：“接入层这边安全检查都没问题，后面的功能层让我转告您，说您的请求有问题。”

- **20200**：这里是实打实的功能层错误码，是具体功能模块返回的，比如20108代表“频道处于非编辑状态”，代表您的请求已经给到直播系统在处理了，只是直播系统任务您的请求有问题，返回特定的提示性错误。

错误码的查询统一参考[错误码表](#)。

# 返回结果

最近更新时间：2018-07-10 17:26:53

## 返回值结构

如无特别说明, 每次请求的返回值中, 都会包含下面的字段：

名称	类型	描述	必选
code	Int	返回结果的错误码, 0表示成功, 其它值表示失败。具体错误码的含义可以参考 <a href="#">错误码</a> 页面	Yes
message	String	请求结果	Yes

例如：

使用公共参数部分的示例请求：

```
https://live/v2/index.php?Action=DescribeInstances&SecretId=xxxxxxx&Region=gz
&Timestamp=1402992826&Nonce=345122&Signature=mysignature&instanceId=101
```

可能的返回结果如下：

```
{
  "code":0,
  "message": "success",
  "instanceSet":
  [{
    "instanceId":"qcvm1234",
    "cpu":1,
    "mem":2,
    "disk":20,
    "bandwidth":65535,
    "os":"centos_62_64",
    "lanIp":"10.207.248.186",
    "wanIp":null,
    "status":0
  }]
}
```

## 错误码

响应包体内的返回码(code), 反映了腾讯云API调用和执行的概要结果.

当返回码不为 0 时, 表示请求未正常执行, 返回码也称为错误码, 错误描述(message)对该结果进行了细化补充, 用户可根据错误码判断API的执行情况.

message在部分终端(例如浏览器), 中文会显示成unicode编码, 需要解码.

**腾讯云 API 可能返回的错误码表如下:**

错误代码	错误类型	描述
4000	请求参数非法	缺少必要参数, 或者参数值格式不正确, 具体错误信息请查看错误描述 message 字段。
4100	鉴权失败	签名鉴权失败, 请参考文档中鉴权部分。
4200	请求过期	请求已经过期, 请参考文档中请求有效期部分。
4300	拒绝访问	帐号被封禁, 或者不在接口针对的用户范围内等。
4400	超过配额	请求的次数超过了配额限制, 请参考文档请求配额部分。
4500	重放攻击	请求的 Nonce 和 Timestamp 参数用于确保每次请求只会在服务器端被执行一次,所以本次的 Nonce 和上次的不能重复, Timestamp 与腾讯服务器相差不能超过 2 小时。
4600	协议不支持	协议不支持, 请参考文档说明。
5000	资源不存在	资源标识对应的实例不存在, 或者实例已经被退还, 或者访问了其他用户的资源。
5100	资源操作失败	对资源的操作失败, 具体错误信息请查看错误描述 message 字段, 稍后重试或者联系客服人员帮忙解决。
5200	资源购买失败	购买资源失败, 可能是不支持实例配置, 资源不足等等。
5300	资源购买失败	购买资源失败, 余额不足。
5400	部分执行成功	批量操作部分执行成功, 详情见方法返回值。
5500	用户资质审核未通过	购买资源失败, 用户资质审核未通过。
6000	服务器内部错误	服务器内部出现错误, 请稍后重试或者联系客服人员帮忙解决。
6100	版本暂不	本版本内不支持此接口或该接口处于维护状态等。注意: 出现这个错误时, 请先确定接口

	支持	的域名是否正确, 不同的模块, 域名可能不一样。
6200	接口暂时无法访问	当前接口处于停服维护状态, 请稍后重试。

## 异步任务接口返回格式

### 1. 普通异步任务接口返回格式

一次请求只能操作一个资源的异步任务接口, 例如创建负载均衡, 重置主机操作系统等。

名称	类型	描述	必选
code	Int	返回结果的错误码, 0表示成功, 其它值表示失败。	Yes
message	String	返回结果的错误信息	No
requestId	String	任务编号	Yes

### 2. 批量异步任务接口返回格式

一次请求能操作多个资源的异步任务接口, 例如修改密码, 启动机器, 停止机器等。

名称	类型	描述	必选
code	Int	返回结果的错误码, 0表示成功, 其它值表示失败。	Yes
message	String	返回结果的错误信息	No
detail	Array	以资源ID为key, 返回对该资源操作的code,message,requestId	Yes

例如：

```
{
  "code":0,
  "message": "success",
  "detail":
  {
    "qcv6a456b0d8f01d4b2b1f5073d3fb8ccc0":
    {
      "code":0,
      "message": "",
      "requestId": "1231231231231";
    }
  }
  "qcv6a456b0d8f01d4b2b1f5073d3fb8ccc0":
```

```
{
  "code":0,
  "message": "",
  "requestId": "1231231231232";
}
```

注意：

资源全部操作成功，则最外层code为0

资源全部操作失败，则最外层code会返回5100

资源部分操作失败，则最外层code会返回5400

在第3种情况下，终端可以通过detail得到失败部分的操作信息。

## 代码示例

### 1. SDK代码下载

[示例代码 PHP 版](#)

[示例代码 Java 版](#)

[示例代码 Python 版](#)

[示例代码 .NET 版](#)

将示例代码中的 YOUR\_SECRET\_ID 和 YOUR\_SECRET\_KEY 替换成实际的 SecretId 和 SecretKey

示例代码仅供参考，请根据实际情况使用。

### 2. PHP语言的示例代码

```
<?php
```

```
/******真实调用时，需要根据不同接口修改下面的参数******/
/******此处以DescribeInstances为例说明 如何获取指定 instanceId 的云服务器******/
```

```
/*DescribeInstances 接口的 URL地址为 cvm.api.qcloud.com，可从对应的接口说明 “1.接口描述” 章节获取该接口的地址*/
```

```
$HttpUrl="cvm.api.qcloud.com";
```

```
/*除非有特殊说明，如MultipartUploadVodFile，其它接口都支持GET及POST*/
```

```
$HttpMethod="GET";
```

```

/*是否https协议，大部分接口都必须为https，只有少部分接口除外（如MultipartUploadVodFile）*/
$isHttps =true;

/*需要填写您的密钥，可从 https://console.cloud.tencent.com/capi 获取 SecretId 及 $secretKey*/
$secretKey='XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX';

/*下面这五个参数为所有接口的公共参数；对于某些接口没有地域概念，则不用传递Region（如DescribeDeal
s）*/
$COMMON_PARAMS = array(
'Nonce'=> rand(),
'Timestamp'=>time(NULL),
'Action'=>'DescribeInstances',
'SecretId'=>'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX',
'Region'=>'gz',
);

/*下面这两个参数为 DescribeInstances 接口的私有参数，用于查询特定的云服务器列表*/
$PRIVATE_PARAMS = array(
'instanceIds.0'=>'qcvm00001',
'instanceIds.1'=>'qcvm00002',
);

/*****

CreateRequest($HttpUrl,$HttpMethod,$COMMON_PARAMS,$secretKey, $PRIVATE_PARAMS, $isHttps);

function CreateRequest($HttpUrl,$HttpMethod,$COMMON_PARAMS,$secretKey, $PRIVATE_PARAMS,
$isHttps)
{
$FullHttpUrl = $HttpUrl."/v2/index.php";

/*****对请求参数 按参数名 做字典序升序排列，注意此排序区分大小写*****/
$ReqParaArray = array_merge($COMMON_PARAMS, $PRIVATE_PARAMS);
ksort($ReqParaArray);

/*****生成签名原文*****/
* 将 请求方法, URI地址,及排序好的请求参数 按照下面格式 拼接在一起, 生成签名原文，此请求中的原文为
* GETcvm.api.qcloud.com/v2/index.php?Action=DescribeInstances&Nonce=345122&Region=gz
* &SecretId=AKIDz8krbsJ5yKBZQ·1pn74WFkmLPx3gnPhESA&Timestamp=1408704141
* &instanceIds.0=qcvm12345&instanceIds.1=qcvm56789
* *****/
$SigTxt = $HttpMethod.$FullHttpUrl."?";
    
```

```
$isFirst = true;
foreach ($ReqParamArray as $key => $value)
{
if (!$isFirst)
{
$SigTxt = $SigTxt."&";
}
$isFirst= false;

/*拼接签名原文时，如果参数名称中携带_，需要替换成.*/
if(strpos($key, '_'))
{
$key = str_replace('_', '.', $key);
}

$SigTxt=$SigTxt.$key."=".$value;
}

/*****根据签名原文字符串 $SigTxt，生成签名 Signature*****/
$Signature = base64_encode(hash_hmac('sha1', $SigTxt, $secretKey, true));

/*****拼接请求串,对于请求参数及签名，需要进行urlencode编码*****/
$Req = "Signature=".urlencode($Signature);
foreach ($ReqParamArray as $key => $value)
{
$Req=$Req."&".$key."=".$urlencode($value);
}

/*****发送请求*****/
if($HttpMethod === 'GET')
{
if($isHttps === true)
{
$Req="https://".$FullHttpUrl."?".$Req;
}
else
{
$Req="http://".$FullHttpUrl."?".$Req;
}

$Rsp = file_get_contents($Req);
}
else
```



```
{
if($isHttps === true)
{
$Rsp= SendPost("https://".$FullHttpUrl,$Req,$isHttps);
}
else
{
$Rsp= SendPost("http://".$FullHttpUrl,$Req,$isHttps);
}
}

var_export(json_decode($Rsp,true));
}

function SendPost($FullHttpUrl,$Req,$isHttps)
{

$ch = curl_init();
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $Req);

curl_setopt($ch, CURLOPT_URL, $FullHttpUrl);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
if ($isHttps === true) {
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);
}

$result = curl_exec($ch);

return $result;
}
```

# 签名方法

最近更新时间：2018-07-10 17:25:39

腾讯云 API 会对每个访问的请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息 (Signature) 以验证用户身份。签名信息由用户所执有的安全凭证生成，安全凭证包括 SecretId 和 SecretKey，若用户还没有安全凭证，则需要在腾讯云官网上自主申请，否则就无法调用云API接口。

## 1. 申请安全凭证

在第一次使用云API之前，用户需要在腾讯云CVM控制台上申请安全凭证。安全凭证包括 SecretId 和 SecretKey，其中SecretId 是用于标识 API 调用者身份的，而SecretKey是用于加密签名字符串和服务器端验证签名字符串的密钥。用户应严格保管其SecretKey，避免泄露。

申请安全凭证的具体步骤如下：

- 1) 登录[腾讯云管理中心控制台](#)。
- 2) 单击【云产品】，选择【监控与管理】栏下的【云API密钥】，进入云API密钥管理页面。



- 3) 在[云API访问密钥管理](#)页面，单击【新建】即可以创建一对SecretId/SecretKey，每个帐号最多可以拥有两对SecretId/SecretKey。

## 2. 生成签名串

有了安全凭证SecretId 和 SecretKey后，就可以生成签名串了。下面给出了一个生成签名串的详细过程。

假设用户的 SecretId 和 SecretKey 分别是：

SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA

SecretKey: Gu5t9xGARNpq86cd98joQYCN3Cozk1qA

**注意：这里只是示例，请用户根据自己实际的SecretId和SecretKey进行后续操作！**

以[查看实例列表](#)(DescribeInstances)请求为例，当用户调用这一接口时，其请求参数可能如下：

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	gz
instanceIds.0	待查询的实例ID	ins-09dx96dg
offset	偏移量	0
limit	最大允许输出	20

由上表可以看出，请求参数中的公共请求参数只有5个：Action、SecretId、Timestamp、Nonce和Region，而不是在“公共请求参数”中所述的6个，事实上，第6个参数Signature（签名串）正是由其他参数（包括指令请求参数）共同生成的，具体步骤如下：

## 2.1. 对参数排序

首先对所有请求参数按参数名做字典序升序排列，所谓字典序升序排列，直观上就如同在字典中排列单词一样排序，按照字母表或数字表里递增顺序的排列次序，即先考虑第一个“字母”，在相同的情况下考虑第二个“字母”，依此类推。您可以借助编程语言中的相关排序函数来实现这一功能，如php中的ksort函数。上述示例参数的排序结果如下：

```
{
  'Action': 'DescribeInstances',
  'Nonce': 11886,
  'Region': 'gz',
  'SecretId': 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA',
  'Timestamp': 1465185768,
  'instanceIds.0': 'ins-09dx96dg',
  'limit': 20,
  'offset': 0,
}
```

使用其它程序设计语言开发时, 可对上面示例中的参数进行排序, 得到的结果一致即可。

## 2.2. 拼接请求字符串

此步骤生成请求字符串。

将把上一步排序好的请求参数格式化“参数名称”=“参数值”的形式, 如对Action参数, 其参数名称为"Action", 参数值为"DescribeInstances", 因此格式化后就为Action=DescribeInstances。

**注意：1、“参数值”为原始值而非url编码后的值。2、若输入参数中包含下划线, 则需要将其转换为“.”。**

然后将格式化后的各个参数用"&"拼接在一起, 最终生成的请求字符串为:

```
Action=DescribeInstances&Nonce=11886&Region=gz&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3g  
nPhESA&Timestamp=1465185768&instanceIds.0=ins-09dx96dg&limit=20&offset=0
```

## 2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。

签名原文字符串由以下几个参数构成:

- 1) 请求方法: 支持 POST 和 GET 方式, 这里使用 GET 请求, 注意方法为全大写。
- 2) 请求主机:[查看实例列表](#)(DescribeInstances)的请求域名为: cvm.api.qcloud.com。实际的请求域名根据接口所属模块的不同而不同, 详见各接口说明。
- 3) 请求路径: 云API的请求路径固定为/v2/index.php。
- 4) 请求字符串: 即上一步生成的请求字符串。

签名原文串的拼接规则为:

```
请求方法 + 请求主机 + 请求路径 + ? + 请求字符串
```

示例的拼接结果为：

```
GETcvm.api.qcloud.com/v2/index.php?Action=DescribeInstances&Nonce=11886&Region=gz&SecretId  
=AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA&Timestamp=1465185768&instanceIds.0=ins-09dx96dg  
&limit=20&offset=0
```

## 2.4. 生成签名串

此步骤生成签名串。

首先使用 HMAC-SHA1 算法对上一步中获得的**签名原文字符串**进行签, 然后将生成的签名串使用 Base64 进行编码, 即可获得最终的签名串。

具体代码如下, 以 PHP 语言为例:

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3Cozk1qA';  
$srcStr = 'GETcvm.api.qcloud.com/v2/index.php?Action=DescribeInstances&Nonce=11886&Region=gz  
&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA&Timestamp=1465185768&instanceIds.0=ins-  
09dx96dg&limit=20&offset=0';  
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));  
echo $signStr;
```

最终得到的签名串为:

```
NSI3UqqD99b/UJb4tbG/xZpRW64=
```

使用其它程序设计语言开发时, 可用上面示例中的原文进行签名验证, 得到的签名串与例子中的一致即可。

### 3. 签名串编码

生成的签名串并不能直接作为请求参数, 需要对其进行 URL 编码。

**注意: 如果用户的请求方法是GET, 则对所有请求参数值均需要做URL编码。**

如上一步生成的签名串为NSI3UqqD99b/UJb4tbG/xZpRW64=, 则其编码后为

NSI3UqqD99b/UJb4tbG/xZpRW64%3D。因此, 最终得到的签名串请求参数(Signature)为:

NSI3UqqD99b/UJb4tbG/xZpRW64%3D, 它将用于生成最终的请求URL。

# 服务端API

## API 概览

最近更新时间：2018-07-10 17:08:00

接口功能	Action ID
创建直播频道	<a href="#">CreateLVBChannel</a>
查询直播频道列表	<a href="#">DescribeLVBChannelList</a>
查询直播频道详情	<a href="#">DescribeLVBChannel</a>
修改直播频道	<a href="#">ModifyLVBChannel</a>
批量启用直播频道	<a href="#">StartLVBChannel</a>
批量停止直播频道	<a href="#">StopLVBChannel</a>
删除直播频道	<a href="#">DeleteLVBChannel</a>
创建录制任务	<a href="#">CreateRecord</a>
终止录制任务	<a href="#">StopRecord</a>
查询录制文件(荐)	<a href="#">GetVodRecordFiles</a>
查询已录制分片列表	<a href="#">DescribeRecord</a>
创建截图任务	<a href="#">CreateLVBSHOT</a>
终止截图任务	<a href="#">StopLVBSHOT</a>
查看队列消息	<a href="#">DescribeQueueLog</a>
删除录制任务	<a href="#">DeleteRecord</a>
删除截图任务	<a href="#">DeleteLVBSHOT</a>
查询录制任务列表	<a href="#">DescribeRecordList</a>
查询截图任务列表	<a href="#">DescribeLVBSHOTList</a>

# 创建直播频道

最近更新时间：2018-07-10 17:11:02

## 创建频道

### 1. 接口描述

域名：live.api.qcloud.com

接口名：CreateLVBChannel

### 2. 输入参数

参数名称	必选	类型	描述
channelName	是	String	频道名称
channelDescribe	否	string	直播频道描述
outputSourceType	是	int	输出源选择（1表示只有RTMP输出，2表示只有HLS输出，3表示两者都有）
playerPassword	否	string	接收方播放器密码
sourceList	是	array	直播源列表，每个直播源可以包括类型、名称和地址（限于拉流） 注：目前只支持一个直播源
watermarkId	否	int	水印ID
outputRate	否	array	输出码率。注：参数数组，0表示原始码率；10表示550码率(即标准)；20表示900码率(即高清)。如需设置码率，0是必填

sourceList结构如下：

参数名称	必选	类型	描述
name	是	string	直播源名称
type	是	int	1 rtmp推流

### 3. 输出参数

参数名称	类型	描述
code	Int	错误码, 0: 成功, 其他值: 失败

参数名称	类型	描述
message	String	错误信息
channel_id	String	频道id

#### 4. 示例

输入 (rtmp推流示例)

```
https://domain/v2/index.php?Action=CreateLVBChannel&channelName=国际乒乓球锦标赛3&outputSourceType=2&
```

输出

```
{
  "code": 0,
  "message": "",
  "codeDesc": "Success",
  "channel_id": "XXX",
  "channelInfo": {
    "upstream_address": "rtmp://2000.livepush.myqcloud.com/live/YYYYYYYYYYYYYYYYYY?bizid2000",
    "downstream_address": [{
      "rate_type": 0,
      "hls_downstream_address": "http://2000.liveplay.myqcloud.com/live/XXX.m3u8",
      "rtmp_downstream_address": "rtmp://2000.liveplay.myqcloud.com/live/XXX",
      "flv_downstream_address": "http://2000.liveplay.myqcloud.com/live/XXX.flv"
    }]
  }
}
```

#### 5. 标准参数定义

频道状态定义：

值	状态
0	无输入流
1	直播中
2	异常
3	关闭



接收方类型定义：

类型	含义
1	RTMP输出
2	HLS输出
3	RTMP 和HLS输出

直播流协议定义：

类型	含义
1	RTMP推流
2	RTMP拉流
3	HLS拉流

# 修改直播频道

最近更新时间：2018-07-10 17:08:54

## 1. 接口描述

域名：live.api.qcloud.com

接口名: ModifyLVBChannel

修改直播频道的基本信息，包括频道的名称和描述信息。

## 2. 输入参数

参数名称	必选	类型	描述
channelId	是	无符号整型	频道的ID号
channelName	是	字符串型	直播频道名称
channelDescribe	否	字符串型	直播频道描述

## 3. 输出参数

参数名称	类型	描述
code	Int	错误码, 0: 成功, 其他值: 失败
message	String	错误信息

## 4. 示例

输入1

```
https://domain/v2/index.php?Action=ModifyLVBChannel&channelId=96171715553394807&channelName=test-15&channelDescribe=第15个测试频道用例&公共请求参数
```

输出1

```
{
  "code" : 0,
  "message" : "",
}
```

## 5.标准参数定义

频道状态定义：

值	状态
0	无输入流
1	直播中
2	异常
3	关闭

接收方类型定义：

类型	含义
1	RTMP输出
2	HLS输出
3	RTMP 和HLS输出

直播流协议定义：

类型	含义
1	RTMP推流
2	RTMP拉流
3	HLS拉流

# 删除直播频道

最近更新时间：2018-07-10 17:13:31

## 1. 接口描述

域名：live.api.qcloud.com

接口名: DeleteLVBChannel

输入待删除的直播频道的ID号（支持批量），删除该频道。

## 2. 输入参数

参数名称	必选	类型	描述
channelIds	是	数组型	频道的ID值，支持批量，具体用法见示例

## 3. 输出参数

参数名称	类型	描述
code	Int	错误码, 0: 成功, 其他值: 失败
message	String	错误信息

## 4. 示例

输入1

```
https://domain/v2/index.php?Action=DeleteLVBChannel&channelIds.1=96171715553394807&公共请求参数
```

输出1

```
{  
  "code": 0,  
  "message": "",  
}
```

## 5.标准参数定义

频道状态定义：

值	状态
0	无输入流
1	直播中
2	异常
3	关闭

接收方类型定义：

类型	含义
1	RTMP输出
2	HLS输出
3	RTMP 和HLS输出

直播流协议定义：

类型	含义
1	RTMP推流
2	RTMP拉流
3	HLS拉流

# 批量启用直播频道

最近更新时间：2018-07-10 17:15:39

## 1. 接口描述

域名：live.api.qcloud.com

接口名: StartLVBChannel

输入待启用直播频道的ID号（支持批量），启动这些直播频道。

## 2. 输入参数

参数名称	必选	类型	描述
channelIds	是	数组型	频道的ID值，支持批量，具体用法见示例

## 3. 输出参数

参数名称	类型	描述
code	Int	错误码, 0: 成功, 其他值: 失败
message	String	错误信息

## 4. 示例

输入1

```
https://domain/v2/index.php?Action=StartLVBChannel&channelIds.1=96171715553394807&公共请求参数
```

输出1

```
{
  "code": 0,
  "message": "",
}
```

## 5.标准参数定义

频道状态定义：

值	状态
0	无输入流
1	直播中
2	异常
3	关闭

接收方类型定义：

类型	含义
1	RTMP输出
2	HLS输出
3	RTMP 和HLS输出

直播流协议定义：

类型	含义
1	RTMP推流
2	RTMP拉流
3	HLS拉流

# 批量停止直播频道

最近更新时间：2018-07-10 17:14:16

## 1. 接口描述

域名：live.api.qcloud.com

接口名: StopLVBChannel

输入待停止直播频道的ID号（支持批量），停止这些直播频道。

## 2. 输入参数

参数名称	必选	类型	描述
channelIds	是	数组型	频道的ID值，支持批量，具体用法见示例

## 3. 输出参数

参数名称	类型	描述
code	Int	错误码, 0: 成功, 其他值: 失败
message	String	错误信息

## 4. 示例

输入1

```
https://domain/v2/index.php?Action=StopLVBChannel&channelIds.1=96171715553394810&公共请求参数
```

输出1

```
{  
  "code": 0,  
  "message": "",  
}
```



## 5.标准参数定义

频道状态定义：

值	状态
0	无输入流
1	直播中
2	异常
3	关闭

接收方类型定义：

类型	含义
1	RTMP输出
2	HLS输出
3	RTMP 和HLS输出

直播流协议定义：

类型	含义
1	RTMP推流
2	RTMP拉流
3	HLS拉流

# 查询直播频道列表

最近更新时间：2018-07-10 17:18:43

## 1. 接口描述

域名：live.api.qcloud.com

接口名: DescribeLVBChannelList

获取用户的全部直播频道信息，包括频道的ID、当前状态、名称和创建时间。

## 2. 输入参数

参数名称	必选	类型	描述
channelStatus	否	无符号整型	用作频道列表过滤，频道状态（0：无输入流;1：有输入流;2：异常;3：关闭;4：配置不完整）
ascDesc	否	无符号整型	结果排序，默认按频道创建时间排序，0：按时间升序 1：按时间降序
pageNo	否	无符号整型	分页页号，比如想查看第3页的列表，将该变量设置为3
pageSize	否	无符号整型	每一个分页上显示的频道的个数
orderBy	否	字符串型	默认以频道创建时间来排序

## 3. 输出参数

参数名称	类型	描述
code	Int	错误码, 0: 成功, 其他值: 失败
message	String	错误信息
all_count	int	总频道数
channelSet	数组	频道列表

## 4. 示例

输入1

```
https://domain/v2/index.php?Action=DescribeLVBChannelList&channelStatus=0&ascDesc=1&pageNo=1&pageSize=
```

输出1

```
{
  "code": 0,
  "message": "",
  "all_count": "2",
  "channelSet": [
    {
      "channel_id": "96171715553394811",
      "channel_name": "国际乒乓球锦标赛3",
      "channel_status": "0",
      "create_time": "2015-07-23 20:05:52"
    },
    {
      "channel_id": "96171715553394810",
      "channel_name": "国际乒乓球锦标赛2",
      "channel_status": "0",
      "create_time": "2015-07-23 19:54:05"
    }
  ]
}
```

## 5. 标准参数定义

频道状态定义：

值	状态
0	无输入流
1	直播中
2	异常
3	关闭

接收方类型定义：

类型	含义
1	RTMP输出
2	HLS输出
3	RTMP 和HLS输出

直播流协议定义：

类型	含义
1	RTMP推流
2	RTMP拉流
3	HLS拉流

# 查询直播频道详情

最近更新时间：2018-07-10 17:19:43

## 1. 接口描述

域名：live.api.qcloud.com

接口名: DescribeLVBChannel

输入待查询频道的ID号，获得该频道的当前状态、名称、描述、直播源信息和输出源信息。

## 2. 输入参数

参数名称	必选	类型	描述
channelId	是	无符号整型	频道的ID号

## 3. 输出参数

参数名称	类型	描述
code	Int	错误码, 0: 成功, 其他值: 失败
message	String	错误信息
channelInfo	数组型	频道的信息（当前状态、名称、描述、直播源信息和输出源信息）

## 4. 示例

输入1

```
https://domain/v2/index.php?Action=DescribeLVBChannel&channelId=96171715553394807&公共请求参数
```

输出1

```
{  
  "code": 0,  
  "message": "",  
}
```

```

"channelInfo": [
{
"channel_id": "XXX",
"channel_name": "XXX",
"channel_describe": "XXX",
"channel_status": "1",
"upstream_list": [
{
"sourceName": "RTMP",
"sourceID": "YYYYYYYYYYYY",
"sourceType": "1",
"sourceAddress": "rtmp://2000.livepush.myqcloud.com/live/YYYYYYYYYYYYYYYYYYYY?bizid2000"
}
],
"hls_downstream_address": "",
"rtmp_downstream_address": "rtmp://2000.liveplay.myqcloud.com/live/XXX",
"player_id": "226",
"resolution": null,
"password": null
}
]
}
    
```

## 5. 标准参数定义

频道状态定义：

值	状态
0	无输入流
1	直播中
2	异常
3	关闭

接收方类型定义：

类型	含义
1	RTMP输出
2	HLS输出

3	RTMP 和HLS输出
---	-------------

直播流协议定义：

类型	含义
1	RTMP推流
2	RTMP拉流
3	HLS拉流

# 创建录制任务

最近更新时间：2018-07-18 17:08:48

## 1. 接口描述

域名：live.api.qcloud.com

接口名: CreateRecord

说明：**录制文件存放于点播平台。用户如需使用录制功能，需首先自行开通点播服务。**录制文件存放后，相关费用（含存储以及下行播放流量）按照点播平台计费方式收取，请参考[对应文档](#)。注意：调用接口超时设置应大于3秒，小于3秒重试以及频繁调用都有可能产生重复录制任务。

录制规则

- 1) 任务时间到达时，即启动录制任务，直至直播结束
- 2) 录播按时间进行分片，最大大小为0.5小时；
- 3) 录播时，发生断流，则停止录播生成一个分片

断流恢复后启动新的录播分片任务，重复1) 2) 3) 的过程，直至整个任务结束

录制分片请求url格式：[http://\(点播bizid\).vod.myqcloud.com/\(vid\).f0.flv](http://(点播bizid).vod.myqcloud.com/(vid).f0.flv)

或直接到点播页面查看即可

## 2. 输入参数

参数名称	必选	类型	描述
channelId	是	String	频道id
startTime	是	String	开始时间（中国标准时间），如 2017-01-01 10:10:01
endTime	是	String	结束时间（中国标准时间），如 2017-01-01 10:10:01
tapeType	是	Int	是否开启实时视频录制。1 - 开启；0 - 关闭，推荐使用实时视频录制。 a. 实时视频录制创建成功依赖主播活跃推流，调用接口成功则同步开始录制，并忽略任务开始时间参数。 b. 实时视频录制任务时长支持最大为 30 分钟，如果传入任务结束时间与当前时间差大于 30 分钟，则按 30 分钟计算。录制时长建议控制在 5 分钟以内。 c. 如果关闭实时录制，则需要传入任务开始时间参数，且结束时间与开始时间差不大于 1 天。

## 3. 输出参数



参数名称	类型	描述
code	Int	错误码, 0: 成功, 其他值: 失败
message	String	错误信息
task_id	Int	任务ID, 64位无符整形

## 4. 示例

输入1

```
http://domain/v2/index.php?Action=CreateRecord&channelId=123&tapeType=1&endTime=2016-01-21 12:00:00&公
```

输出1

```
{  
  "code": 0,  
  "message": "",  
  "task_id": 1  
}
```

# 终止录制任务

最近更新时间：2018-07-10 17:20:33

## 1. 接口描述

域名：live.api.qcloud.com

接口名: StopRecord

说明：录制后的文件存放于点播平台。用户如需使用录制功能，需首先自行开通点播账号并确保账号可用。录制文件存放后，相关费用（含存储以及下行播放流量）按照点播平台计费方式收取，请参考[对应文档](#)。

## 2. 输入参数

参数名称	必选	类型	描述
channelId	是	String	频道id
taskId	是	Int	任务id

## 3. 输出参数

参数名称	类型	描述
code	Int	错误码, 0: 成功, 其他值: 失败
message	String	错误信息

## 4. 示例

输入1

```
http://domain/v2/index.php?Action=StopRecord&channelId=123&taskId=1&公共请求参数
```

输出1

```
{  
  "code": 0,  
  "message": "",  
}
```

# 删除录制任务

最近更新时间：2018-07-10 17:11:48

## 1. 接口描述

域名：live.api.qcloud.com

接口名：DeleteRecord

## 2. 输入参数

参数名称	必选	类型	描述
channelId	是	String	频道id
taskId	是	Int	任务id

## 3. 输出参数

参数名称	类型	描述
code	Int	错误码, 0: 成功, 其他值: 失败
message	String	错误信息

## 4. 示例

输入1

```
http://domain/v2/index.php?Action=DeleteRecord&channelId=123&taskId=1&公共请求参数
```

输出1

```
{  
  "code": 0,  
}
```

```
"message": "",  
}
```

# 查询录制任务列表

最近更新时间：2018-07-10 17:16:54

## 1. 接口描述

域名：live.api.qcloud.com

接口名: DescribeRecordList

注意事项：**最长可查询2个月内的录制任务。**

## 2. 输入参数

参数名称	必选	类型	描述
channelId	是	String	频道id
pageNo	否	Int	页码，默认为1
pageSize	否	Int	分页大小，默认为10

## 3. 输出参数

参数名称	类型	描述
code	Int	错误码, 0: 成功, 其他值: 失败
message	String	错误信息
totalCount	Int	总个数
taskSet	Array	任务结果集

文件信息结构如下：

参数名称	类型	描述
id	Int	任务id
startTime	String	分片起始时间

endTime	String	分片结束时间
status	Int	任务状态, 0-未启动, 1-启动中, 2-执行完, 3-异常等
count	Int	已生成分片数

## 4. 示例

输入1

```
http://domain/v2/index.php?Action=DescribeRecordList&channelId=16093104850681751595&pageSi ze=10&pageN
```

输出1

```
{  
  
  "code": 0,  
  "message": "",  
  "totalCount": 2,  
  "taskSet ": [  
    {  
      "id": "16093104850681751583",  
      "startTime": "7",  
      "endTime": "720",  
      "status": "0"  
    },  
    {  
      "id": "16093104850681751599",  
      "startTime": "8",  
      "endTime": "1806",  
      "status ": "2"  
      "count ": "5"  
    }  
  ]  
}
```

# 查询录制文件(荐)

最近更新时间：2018-07-10 17:17:34

## 1. 接口描述

域名：live.api.qcloud.com

接口名: GetVodRecordFiles

注意事项：

- 1.查询频道录制文件结果信息，包含文件名，任务id，文件id，录制消息。
- 2.最长查询2个月内的录制文件。

## 2. 输入参数

参数名称	必选	类型	描述
channelId	是	String	频道的ID号
startTime	是	String	填写YYYY-MM-DD HH:MM:SS经过encode编码后的内容 eg: 2016-01-01 00:00:00 ( 2016-01-01 00:00:00经过encode编码的内容 )
endTime	否	String	写YYYY-MM-DD HH:MM:SS经过encode编码后的内容, 默认不限截至时间
pageNum	否	Int	页码，默认值1
pageSize	否	Int	一页的记录数，默认值20

## 3. 输出参数

参数名称	类型	描述
code	Int	错误码, 0: 成功, 其他值: 失败
message	String	错误信息
total_count	int	总返回文件数
filesInfo	数组	[ { "fileId": "9896125784085567721", "fileName": "直播 -aaa-20160112-1455-20160112-1457", "reportMessage": null, "taskId": "57", "startTime": "7", "endTime": "128" } ]



## 4. 示例

输入1

```
https://domain/v2/index.php?Action=GetVodRecordFiles&channelId=9896125784085535840&startTime=2016-01-(
```

输出1

```
{
  "code": 0,
  "message": "",
  "codeDesc": "Success",
  "filesInfo": [
    {
      "fileId": "9896125784085567721",
      "fileName": "直播 -aaa-20160112-1455-20160112-1457",
      "reportMessage": null,
      "taskId": "57",
      "startTime": "7",
      "endTime": "128"
    },
    {
      "fileId": "9896125784085567722",
      "fileName": "直播 -aaa-20160112-1457-20160112-1459",
      "reportMessage": null,
      "taskId": "57",
      "startTime": "128",
      "endTime": "250"
    },
    {
      "fileId": "9896125784085567723",
      "fileName": "直播 -aaa-20160112-1459-20160112-1459",
      "reportMessage": null,
      "taskId": "57",
      "startTime": "250",
      "endTime": "299"
    },
    {
      "fileId": "9896125784085567727",
      "fileName": "直播 -aaa-20160112-1510-20160112-1512",
      "reportMessage": null,
      "taskId": "58",
      "startTime": "7",
      "endTime": "132"
    }
  ]
}
```

```
},
{
  "fileId": "9896125784085567736",
  "fileName": "直播 -aaa-20160112-1512-20160112-1514",
  "reportMessage": null,
  "taskId": "58",
  "startTime": "132",
  "endTime": "258"
},
{
  "fileId": "9896125784085567739",
  "fileName": "直播 -aaa-20160112-1514-20160112-1515",
  "reportMessage": null,
  "taskId": "58",
  "startTime": "258",
  "endTime": "301"
},
{
  "fileId": "9896125784085567756",
  "fileName": "直播 -aaa-20160112-1525-20160112-1526",
  "reportMessage": null,
  "taskId": "61",
  "startTime": "7",
  "endTime": "60"
}
],
"totalCount": 7
}
```

# 查询已录制分片列表

最近更新时间：2018-07-10 17:16:28

## 1. 接口描述

域名：live.api.qcloud.com

接口名: DescribeRecord

说明：录制后的文件存放于点播平台。用户如需使用录制功能，需首先自行开通点播账号并确保账号可用。录制文件存放后，相关费用（含存储以及下行播放流量）按照点播平台计费方式收取，请参考[对应文档](#)。

## 2. 输入参数

参数名称	必选	类型	描述
channelId	是	String	频道id
taskId	是	Int	任务id
pageNo	否	Int	页码，默认为1
pageSize	否	Int	分页大小，默认为10

## 3. 输出参数

参数名称	类型	描述
code	Int	错误码, 0: 成功, 其他值: 失败
message	String	错误信息
totalCount	Int	总个数
fileSet	Array	文件结果集

文件信息结构如下

参数名称	类型	描述
fileId	String	文件ID

startTime	String	分片起始时间
endTime	String	分片结束时间
fileName	String	文件名

## 4. 示例

输入1

```
http://domain/v2/index.php?Action=DescribeRecord&channelId=16093104850681751595&taskId=10&pageSi ze=100
```

输出1

```
{
  "code": 0,
  "message": "",
  "totalCount": 2,
  "fileSet": [
    {
      "fileId": "16093104850681751583",
      "startTime": "7",
      "endTime": "720",
      "fileName": "直播 -sparrow_hls-20160120-0958-20160120-1010"
    },
    {
      "fileId": "16093104850681751599",
      "startTime": "8",
      "endTime": "1806",
      "fileName": "直播 -sparrow_hls-20160120-1010-20160120-1040"
    }
  ]
}
```

在录制后，点播平台将生成对应文件，例如：

```
http://2527.vod.myqcloud.com/2527_000007d0b18*****bd98f9125ed6569ee9a90001.f0.mp4
```

完整访问方式，可通过fileId以及[点播API](#)获取

# 错误码表

最近更新时间：2018-07-10 17:27:24

## 错误码的分类

本页所列错误码分为**接入层错误码**和**视频云错误码**，接入层错误码是整个腾讯云接入层统一约定的，视频云错误码则只有您在直播和点播业务中才需要参考。

关于两种错误码类型的区分和内部原理，请参考[双错误码](#)。

## 接入层错误码

错误码	提示消息	具体含义
4000	请求参数非法	缺少必要参数，或者参数值格式不正确，具体错误信息请查看错误描述 message 字段。
4100	鉴权失败	签名鉴权失败，请参考文档中鉴权部分。
4200	请求过期	请求已经过期，请参考文档中请求有效期部分。
4300	拒绝访问	帐号被封禁，或者不在接口针对的用户范围内等。
4400	超过配额	请求的次数超过了配额限制，请参考文档请求配额部分。
4500	重放攻击	请求的 Nonce 和 Timestamp 参数用于确保每次请求只会在服务器端被执行一次,所以本次的 Nonce 和上次的不能重复, Timestamp 与腾讯服务器相差不能超过 2 小时。
4600	协议不支持	协议不支持，请参考文档说明。
5000	资源不存在	资源标识对应的实例不存在，或者实例已经被退还，或者访问了其他用户的资源。
5100	资源操作失败	功能层返回失败，具体错误信息请查看错误描述 message 字段
5200	资源购买失败	购买资源失败，可能是不支持实例配置，资源不足等等。
5300	资源购买失败	购买资源失败，余额不足。

错误码	提示消息	具体含义
5400	部分执行成功	批量操作部分执行成功, 详情见方法返回值。
5500	用户资质审核未通过	购买资源失败, 用户资质审核未通过。
6000	服务器内部错误	服务器内部出现错误, 请稍后重试或者联系客服人员帮忙解决。
6100	版本暂不支持	本版本内不支持此接口或该接口处于维护状态等。注意: 出现这个错误时, 请先确定接口的域名是否正确, 不同的模块, 域名可能不一样。
6200	接口暂时无法访问	当前接口处于停服维护状态, 请稍后重试。

## 视频云错误码

错误码	提示消息
1000	输入参数错误
1001	获取用户帐号错误
1002	用户不存在
1003	用户帐号错误
3000	网络错误
4001	json格式错误
20100	创建频道错误
20101	频道数超过上限
20102	获取频道信息错误
20103	修改频道信息错误
20104	获取频道列表错误
20105	删除频道错误

错误码	提示消息
20106	关闭频道错误
20107	开启频道错误
20108	频道处于非编辑状态
20109	频道处于非关闭状态
20111	频道无直播源
20112	直播源协议错误
20113	接收方协议错误
20114	开通服务失败
20200	拉流地址为空
20201	获取直播源信息错误
20202	创建直播源错误
20203	删除直播源错误
20204	拉流地址为空
20250	黑白名单为空错误
20251	黑白名单不存在
20252	获取黑白名单列表错误
20253	名单数量超过限制
20300	修改频道状态错误
20301	频道不存在
20320	创建录播任务失败
20321	获取录播任务列表失败
20322	录播任务时间冲突
20323	录播任务起始时间和结束时间不合法
20326	修改录播任务失败

错误码	提示消息
20327	录播任务不存在
20328	截图任务不存在
20500	水印不存在