# Live Video Broadcasting

# Cloud API

# Product Introduction

Copyright Notice

Trademark Notice

Tencent Cloud

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

# Contents

- Cloud API
  - Live API 2017
    - summarize
      - LVB Code Access
        - summarize
        - API overview
          - Operation API
            - Event Message Notification
            - Enable Close Push
            - Cloud Mixed Streaming
            - Pause and Resume after Latency
            - Create Recording Task
            - End Recording Task
          - Query API
            - Query Live Status
            - Query Statistical Info
            - Query Recording File
            - Query Channel List
            - Query Live Channel List
            - Getting Billing Bandwidth Data
          - Statistics API
            - Get Push History Info
            - Get Play Statistics History Info
      - Channel Hosting (old)
          - Call Method
            - Request Body
            - Backend SDK
            - Return Results
            - Signature Method
          - Server API
            - API Overview
            - Create Live Channel
            - Change Live Channel
            - Delete Live Channel
            - Enable Live Channel
            - Disable Live Channel
            - Query Live Channel List

- Query Live Channel Details
- Create Recording Task
- End Recording Task
- Delete Recording Task
- Query Recording Task List
- Query Recording File (Recommended)
- Query Recorded Shard List

Error Code List

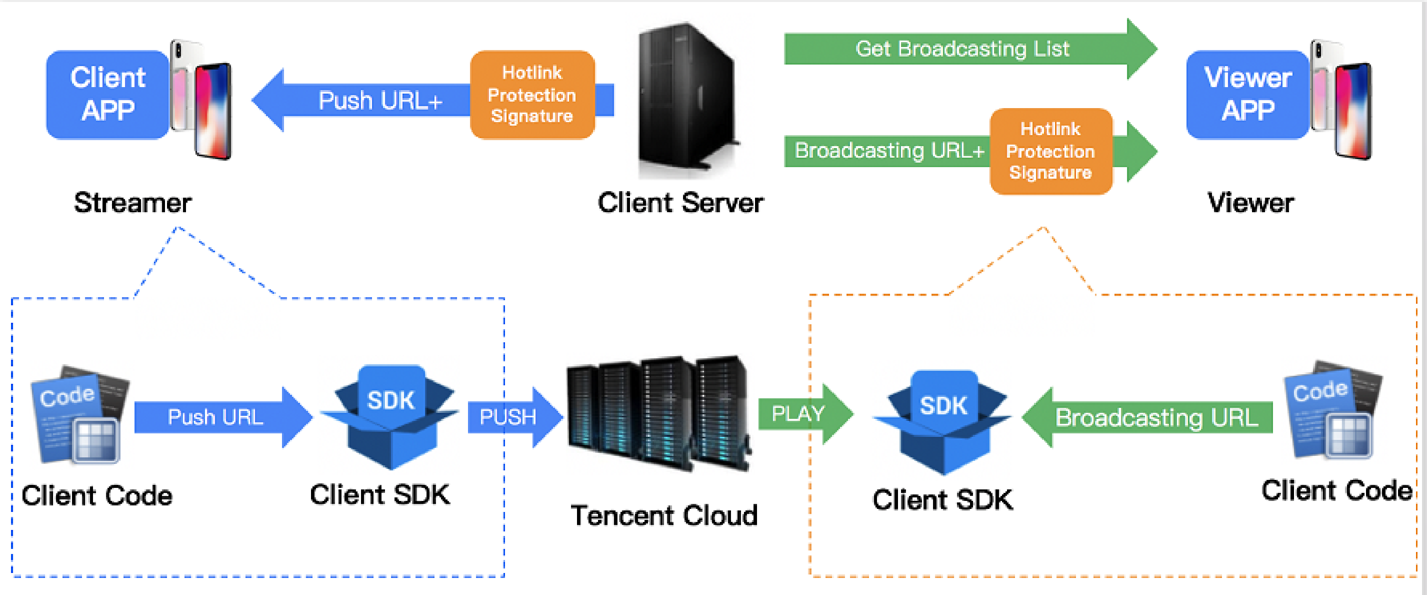
# Cloud API
# Live API 2017
# summarize

Last updated：2018-07-23 19:32:45

Cloud API is designed to help you manage LVB streams. Tencent Cloud offers two modes: LVB Code mode (new) and channel mode (old).

### LVB Code access mode

The LVB Code mode is mainly designed to facilitate the direct management of LVB streams by customers in the backend. It was only open to several well-known LVB platforms in China before May 2016, and now it has become a mainstream access solution of Tencent Cloud LVB service owing to its low access cost and high reliability. This solution is designed in such a way that the URL generation and LVB streams are basically controlled by customers, which delivers higher flexibility and customizability. It is the preferred access solution for platform LVB scenarios such as Game LVB and Live Show.



### Channel (hosting) mode

The channel mode is customized for manual management of LVB channels. It is more suitable for a single event LVB:

- **Create a channel**: Before starting an event, you can create a channel, then get the push URL, and propagate the playback URL.

- **Disable a channel**: After the LVB is finished, or when something unexpected happens, you can disable or ban the LVB channel at any time.

This mode also provides server API support, but the access cost is much higher than LVB Code mode, while the stability is lower.

## The difference between the two modes

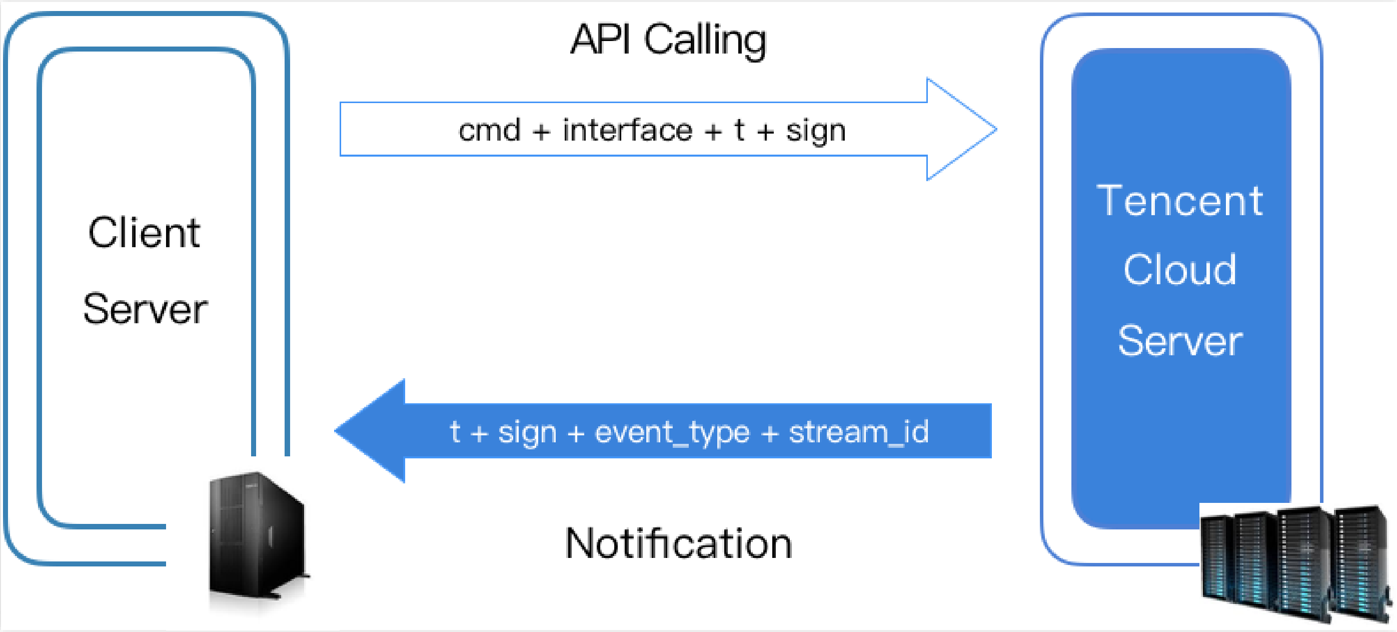
| Item | LVB Code mode | Channel mode |
|---|---|---|
| Push URL | You are free to specify it without communicating with Tencent Cloud backend | You need to generate it in Tencent Cloud backend by creating a channel first |
| Hotlink protection | Supports push and playback hotlink protection (for playback hotlink protection configuration, please contact customer service) | Supports push hotlink protection, and FLV and HLS playback hotlink protection (RTMP playback hotlink protection is not supported) |
| API list | LVB Stream Management API | Suitable for manual operation. API is also provided |
| Message notification | When the stream status changes, you will be notified by a message in JSON format | Not supported |
| Applicable scenario | Suitable for server interfacing, such as Live Show | Suitable for manual operation, such as an LVB event |

# LVB Code Access summarize

Last updated：2018-10-10 18:35:31

Information synchronization between your server and Tencent Cloud's CVM can be implemented by combining the following two ways:

- **API call:** Tencent Cloud provides a set of LVB code management APIs for your backend server, including status query, status management and other features.
- **Message notification:** As long as you register a callback URL for receiving event notifications in Tencent Cloud, Tencent Cloud will actively notify your backend server of such events in case of change of LVB stream status and generation of recording files in the form of event message (JSON).



## Calling method

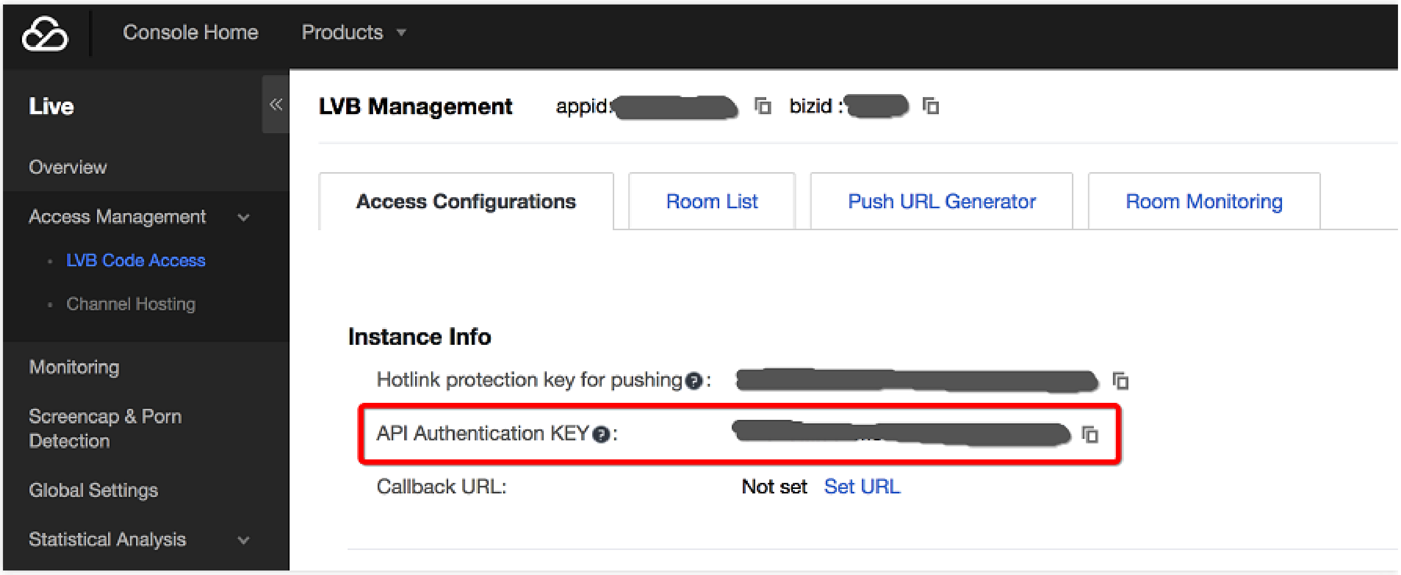
You can call relevant API using GET request method of HTTP protocol on your server, that is, the parameters in the call are directly added in the URL. Example on how to call the API is provided in every API's documentation, so there is no difficulty in interfacing.

## Security mechanism

To ensure performance, we use HTTP protocol to call the API. Therefore, an effective measure is required to ensure the communication security between your server and Tencent Cloud backend.

All LVB Code-related cloud APIs employ the same security check mechanism - "t + sign verification":

- **t (Expiration time):** If the t value specified in an API request or notification has expired, the request or notification is invalid. This can prevent replay attacks. The format of t is the UNIX timestamp, that is, the number of seconds that have elapsed since January 1, 1970 (midnight in UTC/GMT).
- **sign (security signature):** sign = MD5 (key + t). This means computing the MD5 value by concatenating the encryption key and t. The encryption key here is the CGI call key, which can be specified on the Tencent Cloud LVB Console:



LVB Management appid: [redacted] bizid : [redacted]

Instance Info

Hotlink protection key for pushing: [redacted]

API Authentication KEY: [redacted]

Callback URL: Not set Set URL

- **How it works**
  MD5 is an irreversible HASH algorithm. As long as the key is not disclosed, attackers cannot calculate the key or launch spoofing attacks even if they get many pairs of t and sign.
- **Example**
  If the current time is 2016-08-22 15:16:27 and the validity period is 1 minute, the request or notification that comes with this t received later than 2016-08-22 15:17:27 is invalid:

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

Assume that the key is **5d41402abc4b2a76b9719d911017c592**, and after calculation we can get the following signature:

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

## Message notification

For more information, please see Tencent Cloud Event Message Notification service.

# API overview

Last updated：2018-09-03 10:50:36

Tencent Cloud provides a set of LVB Code management APIs for your backend server, delivering status query, status management and other features.

## APIs

| API | Feature Description |
|---|---|
| Get_LiveStat | Statistics query - queries the push and playback information |
| Get_LivePushStat | Statistics query - queries the push information |
| Get_LivePlayStat | Statistics query - queries the playback information |
| Get_LivePushStatHistory | Obtains the historical statistics of push |
| Get_LivePlayStatHistory | Obtains the historical statistics of playback |
| Live_Channel_GetStatus | Queries only the status information of a stream (old version API) |
| Live_Channel_SetStatus | Bans an LVB stream. It is mainly used for porn detection. |
| Live_Tape_GetFilelist | Queries the list of videos recorded during LVB for a certain stream |
| Live_Queue_Get | Queries the list of screenshots captured during LVB for a certain stream |
| Live_Channel_GetChannelList | Queries the channel list |
| Live_Channel_GetLiveChannelList | Queries the channel list in LVB |
| mix_streamv2.start_mix_stream_advanced | API for stream mixing on the cloud |
| channel_manager | Stops pushing stream and delays the availability of API. It can disable push for a specified stream. |
| Live_Tape_Start | Creates a recording task. It supports scheduled recording and real-time recording. |
| Live_Tape_Stop | Ends a recording task |

## Error codes

1. HTTP error codes

| Error Code | Description | Note |
|---|---|---|
| 403 | Forbidden | Verification is enabled for the API to ensure security. If this error occurs when you perform verification via a browser, you can check whether skey is contained in cookie. |
| 404 | Not Found | Check whether the request comes with host |

1. Returned error codes

| Error Code | Description | Note |
|---|---|---|
| appid is invalid | Invalid appid, which indicates the feature is not activated | |

**Note:**
The above error codes apply to the APIs listed in this document and do not include Event Message Notification.

# Operation API

# Event Message Notification

Last updated：2018-09-20 17:59:15

## Feature Description

Events such as state change of an LVB stream, generation of a new recording file and generation of a screenshot file are internally managed in Tencent Cloud. But your backend server may also need to be informed of these events, in which case you can use the event notification service of Tencent Cloud to obtain these events.

You can register a callback URL from your backend server in Tencent Cloud console. When an event occurs, Tencent Cloud will deliver the event to your server using the HTTP POST method, with the event content organized in JSON format.

## Configuring a URL

When you enable the LVB Code mode in the **LVB Console** -> **LVB Code Access** -> **Access Configuration**, you can specify a URL for receiving Tencent Cloud notifications, as shown below:

## Message Organizing Format

Notification messages are organized in JSON format and then placed in the HTTP POST protocol body. Note that the ContentType of the POST format here is application/json instead of multipart/form-data. Therefore, do not use the function for reading form fields in PHP or Java to read the messages.

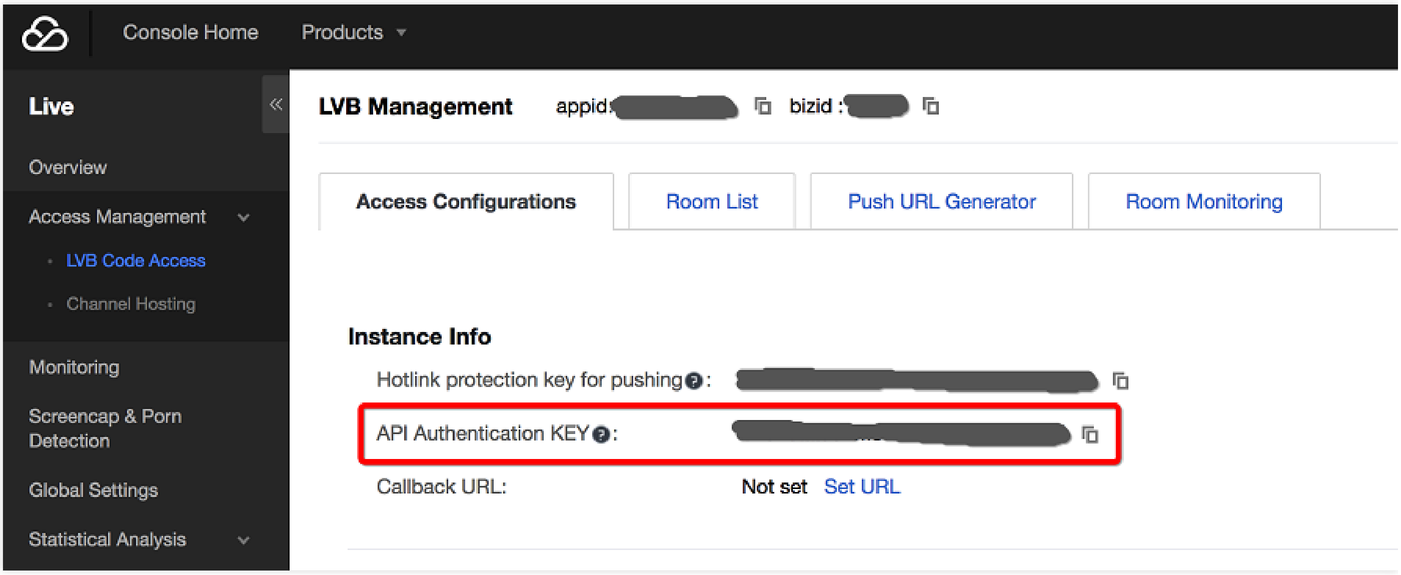
## Common Header Information

The following fields are included in all types of notification messages:

| Field Name | Type | Description | Note |
|---|---|---|---|
| t | string | Validity period | UNIX timestamp (decimal) |

| Field Name | Type | Description | Note |
|---|---|---|---|
| sign | string | Security signature | MD5(KEY+t) |
| event_type | int | Event type | Current available values: 0, 1, 100, 200 |
| stream_id | string | LVB Code | Indicates the LVB stream from which the event is derived |
| channel_id | string | LVB Code | Same as stream_id |

- **stream_id | channel_id (LVB Code)**:
  In the LVB Code mode, the fields stream_id and channel_id have the same value. They have different names due to historical reasons.
- **t (expiration time)**
  The default expiration time of notifications from Tencent Cloud is 10 minutes. If the time specified by t in a notification has expired, the notification is invalid, which can prevent network replay attacks. The format of t is a decimal UNIX timestamp, that is, the number of seconds that have elapsed since January 1, 1970 (Midnight in UTC/GMT).
- **sign (security signature)**
  sign = MD5(key + t): Tencent Cloud computes the value of "sign" using MD5 algorithm after concatenating the strings of the encryption key and t, then places the value in a notification. Upon receiving the notification, your backend server can verify whether the "sign" value is correct by using the same algorithm so as to check whether the notification is truly from Tencent Cloud backend.
  The encryption key here is the CGI call key, which can be specified when you activate the LVB Code service on the Tencent Cloud LVB console, as shown below:



appid: [redacted] bizid : [redacted]

Instance Info

Hotlink protection key for pushing: [redacted]

API Authentication KEY: [redacted]

Callback URL: Not set Set URL

- **event_type (notification type)**
  Tencent Cloud supports the following types of notifications: 0 - Stream interruption; 1 - Push; 100 -

Generation of a new recording file; 200 - Generation of a new screenshot file.

## Different Types of Message Bodies

### (1) Push (0) Stream Interruption

**event_type = 0** indicates stream interruption, and **event_type = 1** indicates push. These types of message bodies also contain the following information:

| Field Name | Description | Type | Note | Required |
|---|---|---|---|---|
| appname | Push path | string | | Y |
| app | Push domain name | string | | Y |
| event_time | Generation time of the message | int | In second | Y |
| sequence | Sequence number of the message, which indicates a push event. A push event generates push and stream interruption messages with the same sequence number | string | | Y |
| node | Upload access point IP | String | | Y |
| user_ip | User push IP | String | Client_ip | Y |
| errcode | Stream interruption error code | Int | | N |
| errmsg | Stream interruption error message | String | | N |
| stream_param | Push URL parameter | String | | Y |
| push_duration | Duration of the push | String | In millisecond | N |

Example: Tencent Cloud notifies that a stream interruption (event_type=0) occurred for the LVB stream (1234_15919131751).

```
{
"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":""
}
```

**Stream interruption error codes**

| Error Code | Error Description | Reason |
|---|---|---|
| 1 | recv rtmp deleteStream | Active interruption at the VJ end |
| 2 | recv rtmp closeStream | Active interruption at the VJ end |
| 3 | recv() return 0 | Active TCP disconnection at the VJ end |
| 4 | recv() return error | TCP connection exception at the VJ end |
| 7 | rtmp message large than 1M | Exception in received stream data |
| Other error codes | Internal LVB service exceptions | To solve these errors, contact Tencent's service personnel or submit a ticket. Tel: 4009-100-100 |

## (100) New recording file

**event_type = 100** indicates that a new recording file is generated. This type of message body also contains the following information:

| Field Name | Description | Type | Note | Required |
|---|---|---|---|---|
| video_id | vid | string | VID for VOD. It can be used to locate a unique video file on the VOD platform | Y |
| video_url | Download address | string | Download address of a VOD video | Y |
| file_size | File size | string |  | Y |
| start_time | Starting time stamp of the part | int | Start time (UNIX timestamp. The time cannot be accurate to seconds due to interference of the I frame position) | Y |

| Field Name | Description | Type | Note | Required |
|---|---|---|---|---|
| end_time | Ending time stamp of the part | int | End time (UNIX timestamp. The time cannot be accurate to seconds due to interference of the I frame position) | Y |
| file_id | file_id | string | | Y |
| file_format | File format | string | flv, hls, mp4 | Y |
| vod2Flag | Indicates whether VOD 2.0 is enabled | Int | 0: Disabled; 1: Enabled | N |
| record_file_id | ID of the recording file | String | This field only exists when VOD 2.0 is enabled | N |
| duration | Duration of the push | Int | | Y |
| stream_param | Push URL parameter | string | | Y |

Example: A new FLV recording part is generated with the ID of 9192487266581821586. The playback address is `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=5
92FBDD9&mix=layer:b;session_id:709036962551160107;t_id:1",
"t": 1496221502,
"video_id": "200011683_481565e0befe4e44903839aebe370ef6",
"video_url": "http://1252033264.vod2.myqcloud.com/d7a4cabbvodgzp1252033264/0257ade99031868222
958931071/f0.flv"
}
```

### (200) New screenshot file

**event_type = 200** indicates that a new screenshot image is generated. This type of message body also contains the following information:

| Field Name | Description | Type | Note | Required |
|---|---|---|---|---|
| pic_url | Image address | string | Path without the domain name | Y |
| create_time | Time stamp of the screenshot | int | Screenshot timestamp (UNIX timestamp. The time cannot be accurate to seconds due to interference of the I frame position) | Y |
| pic_full_url | Full path of the screenshot | String | The domain name needs to be included | Y |

Example: The LVB stream "2016090090936" generated a new screenshot image on Tencent Cloud:

```
{
"channel_id": "2016090090936",
"create_time": 1473645788,
"event_type": 200,
"pic_url": "/2016-09-12/2016090090936-screenshot-10-03-08-1280x720.jpg", //Path name of the file
"sign": "8704a0297ab7fdd0d8d94f8cc285cbb7",
"stream_id": "2016090090936",
"t": 1473646392
}
```

**Image download address:**

1. Previously, you needed to apply for the COS service to enable the screenshot feature, and the pic_url we returned to you was not the actual image download address but a download path. The actual download address was generated by combining the following:
2. Download prefix: `http://(cos_bucketname)-(cos_appid).file.myqcloud.com/`
3. Download path: `/2016-09-12/2016090090936-screenshot-10-03-08-1280x720.jpg`
4. Complete URL: `http://(cos_bucketname)-(cos_appid).file.myqcloud.com/2016-09-12/2016090090936-screenshot-10-03-08-1280x720.jpg`

cos_appid and cos_bucketname are only available after you activate Cloud Object Storage on Tencent Cloud. Previously, you could only use the screenshot feature after activating the COS service and binding it to the LVB screenshot service. Now, you can use the feature without applying for the COS service. You can activate the screenshot feature in the console, or by calling related API after contacting us to configure relevant information.

1. Now, you don't need to activate the COS service, and we can call back the complete screenshot URL to you with an additional complete image URL field of `pic_full_url` so that you can directly get the complete image URL. In order not to affect the original business, the original callback information fields will remain unchanged (that is, the field pic_url still exists).

## Notification Reliability

Many customers are worried about message loss. For example, if a customer's server goes down for a while, will the messages be lost?

The message reliability guarantee mechanism of Tencent Cloud backend is implemented based on simple retransmission, that is, if a notification is not successfully sent to the specified callback URL, Tencent Cloud will retry 3 times with an interval of 60 seconds.

So how to tell if the message has been sent to your server successfully? This requires your assistance: When your server receives an HTTP event notification successfully, please make a reply as follows:

```
json
// Return error code 0 in the HTTP request for which a message notification is received to indicate that the message is received successfully, so as to prevent Tencent Cloud from sending the notification repeatedly
{ "code":0 }
```

This indicates: "I (customer server) have received your notification. You (Tencent Cloud) should not keep sending the message to me."

# Enable Close Push

Last updated：2018-08-01 21:02:12

## 1. API Description

- **API**
  **Live_Channel_SetStatus**:
  This API is used to disable, interrupt and enable an LVB stream. **Disable** means the stream ID is no longer available for push. If the push is in progress, it will be interrupted and cannot be resumed. **Interrupt** means suspending the stream being pushed, which can be re-pushed later. **Enable** means enabling the stream ID and allowing it to push streams.
- **URL**
  URL for calling API: `http://fcgi.video.qcloud.com/common_access`
- **Purpose**
  This API is used to ban an LVB during porn detection. For example, if a VJ is found to play porny or rebellious content, this LVB stream can be interrupted or disabled at any time.
- **Note**
  Once an LVB stream is set as **disabled**, Tencent Cloud actively disconnects the push linkage and rejects subsequent push requests. A stream can be disabled for up to 3 months. The disabling expires after 3 months.

## 2. Input Parameters

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| appid | Customer ID | int | LVB APPID used for identifying customers | Y |
| interface | API name | string | Live_Channel_SetStatus | Y |
| t | Validity period | int | UNIX timestamp (decimal) | Y |
| sign | Security signature | string | MD5(key+t) | Y |
| Param.s.channel_id | LVB Code | string | | Y |
| Param.n.status | Status | int | 0: Disabled; 1: Enabled for push; 2: Stream interrupted | Y |

**Note:**
For historical reasons, the LVB Code parameter was defined as channel_id in some earlier APIs, and is defined as stream_id in new APIs.

### 3. Output Parameters

| Parameter Name | Description | Type | Note |
|---|---|---|---|
| ret | Error code | int | 0: Successful; other values: Failed. |
| message | Error message | string | Error message |

### 4. Example

Purpose: To ban the LVB stream with an LVB Code of 8888_test123 whose content has violated relevant regulations.

| Component | Example |
|---|---|
| API 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 |

```
//When copying them, remove the invisible line breaks used for improving layout. Otherwise, URL constr
uction errors may occur, such as "appid is invalid".
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
```

# Cloud Mixed Streaming

Last updated：2018-07-24 15:12:19

## 1. API Description

### 1.1 Send a request to the API CGI using HTTP protocol

**Address:** `http://fcgi.video.qcloud.com/common_access` .
**Purpose:** To mix several input streams into one stream on the cloud for output.

### 1.2 Pass the authentication parameter using URI

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

**Parameter description:**

| Parameter | Description | Type | Note | Required |
|---|---|---|---|---|
| appid | Customer ID | int | The LVB APPID, which is used to identify different customers | Y |
| interface | API name | string | The stream mixing API name is always Mix_StreamV2 | Y |
| t | Validity period | int | UNIX timestamp, which is the number of seconds that have elapsed since January 1, 1970 (Midnight in UTC/GMT). This field indicates the expiration time of requests. You need to add 60 seconds of offset to the current time (in sec) | Y |
| sign | Security signature | string | sign = MD5(key + t). This means computing the MD5 value by concatenating the encryption key and t. The encryption key is the API authentication key set on the Tencent Cloud LVB console. | y |

Example on how to compute the security signature "sign":

```
key = "40328529ca4381a80c6ecf2e6aa57438" //API authentication key
t = 1490858347 //t (expiration time)
key + t = "40328529ca4381a80c6ecf2e6aa574381490858347" //Concatenate the strings of the key and t
sign = MD5(key + t) = "7f29ed83c61b77de1b0d66936fd4fd44" //Compute the MD5 value for concatenated string
```

HTTP request description:

```
POST /common_access?interface=Mix_StreamV2&sign=xxxxxxxxx&appid=125250000 HTTP/1.0 Content-Length: 741
```

### 1.3 Send the stream mixing body using POST method

**Example:**

```
{
"timestamp":int(time.time()), # UNIX timestamp, which is the number of seconds that have elapsed since January 1, 1970 (Midnight in UTC/GMT)
"eventId":int(time.time()), # You can use a random number to identify a network request
"interface":
{
"interfaceName":"Mix_StreamV2", # Fixed value: "Mix_StreamV2"
"para":
{
"app_id": appid, # Enter the LVB APPID
"interface": "mix_streamv2.start_mix_stream_advanced", # Fixed value"mix_streamv2.start_mix_stream_advanced"
"mix_stream_session_id" : "5000_enson", # Enter the stream ID of the primary VJ
"output_stream_id": "5000_enson11", # Enter the stream ID of the primary VJ
"output_stream_type": 0, # Enter the output stream type
"input_stream_list":
[
# Primary VJ: Background image
{
"input_stream_id":"5000_enson11", # Stream ID
"layout_params":
{
"image_layer": 1 # Image layer ID: Primary VJ: 1; Secondary VJ: 2, 3, 4, 5, 6 in sequence
}
},
# Secondary VJ 1
{
"input_stream_id":"5000_enson22", # Stream ID
"layout_params":
{
"image_layer": 2, # Image layer ID
"image_width": 160, # Secondary VJ image width
"image_height": 240, # Secondary VJ image height
"location_x": 380, # x offset: Lateral offset from the top left corner of the primary VJ's background image
"location_y": 630 # y offset: Longitudinal offset from the top left corner of the primary VJ's background image
}
},
# Secondary VJ 2
{
"input_stream_id":"5000_enson33",
"layout_params":
{
"image_layer": 3,
"image_width": 160,
"image_height": 240,
"location_x": 380,
"location_y": 390
}
},
# Secondary VJ 3
{
"input_stream_id":"5000_enson44",
"layout_params":
{
"image_layer": 4,
"image_width": 160,
"image_height": 240,
"location_x": 380,
"location_y": 150
}
}
]
}
}
}
```

**Parameter description**

Required parameter description

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| timestamp | Current time | int | Enter the current time (in sec) | Y |
| eventId | Identifies a network request | int | Enter a random number | Y |
| interfaceName | API ID | string | Mix_StreamV2, a fixed value | Y |
| app_id | LVB APPID | int | LVB APPID | Y |

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| interface | Stream mixing API name | string | Enter mix_streamv2.start_mix_stream_advanced, which is fixed | Y |
| mix_stream_session_id | Stream mixing operation ID | string | Enter a string with a length limited to 80 bytes, containing letters, numbers and underscores only | Y |
| output_stream_id | Output stream ID | string | Specified output stream ID.<br>It must be a string with a length limited to 80 bytes containing letters, numbers and underscores only.<br>If this parameter is one of the input streams, output_stream_type =0; if this parameter is not one of the input streams, output_stream_type =1 | Y |
| input_stream_id | Input source ID | string | Specified input source ID | Y |
| image_layer | Image layer ID | int | 1-6 is supported for video input sources, and 1-8 for audio input sources | Y |

Optional parameter description

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| output_stream_type | Output stream type | int | Enter 0 if the output stream is a stream in the list of input streams;<br>enter 1 if the output stream is not in the list of input streams.<br>It defaults to 0 if left empty.<br>Note: Please consult with customer service if you select 1 for this parameter. | N |

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| mix_stream_template_id | Input template ID | int | Available values for two input sources are: 10, 20, 30, 40;<br>three input sources: 310, 390, 391;<br>four input sources: 410;<br>five input sources: 510, 590; and six input sources: 610.<br>It defaults to 0 if left empty. | N |
| input_type | Input source type | int | Available values:<br>0 indicates the input source is an audio or video;<br>3 indicates canvas;<br>4 indicates an audio; and<br>5 indicates a video. | N |
| image_width | Input image width | double | The recommended pixel is between 0-3000, and percentage between 0.01-0.99. | N |
| image_height | Input image height | double | The recommended pixel is between 0-3000, and percentage between 0.01-0.99. | N |
| location_x | x offset | double | Lateral offset from the top left corner of the primary VJ's background image. The recommended pixel is between 0-3000, and percentage between 0.01-0.99. | N |
| location_y | y offset | double | Longitudinal offset from the top left corner of the primary VJ's background image. The recommended pixel is between 0-3000, and percentage between 0.01-0.99. | N |
| color | Color | string | The commonly used colors of canvas are:<br>Black: 0x000000;<br>White: 0xFFFFFF;<br>Blue: 0x99CCFF. | N |
| crop_width | Cropped source image width | int | The recommended pixel is between 0-3000. | N |

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| crop_height | Cropped source image height | int | The recommended pixel is between 0-3000. | N |
| crop_x | Lateral offset from the top left corner of the cropped source image. | int | The recommended pixel is between 0-3000. | N |
| crop_y | Longitudinal offset from the top left corner of the cropped source image. | int | The recommended pixel is between 0-3000. | N |

**Common colors:**

Red: 0xcc0033
Yellow: 0xcc9900
Green: 0xcccc33
Blue: 0x99CCFF
Black: 0x000000
White: 0xFFFFFF
Gray: 0x999999

### 1.4 Message returned by API

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

**Parameter description**

| Parameter Name | Description | Type | Note |
|---|---|---|---|
| code | Returned error code | int | 0: Successful. Other values: Failed |
| message | Error message | string | Returned error message |
| timestamp | Timestamp | Long int | Returned time |

**Common error codes:**

| Error Code | Description |
|---|---|

| Error Code | Description |
|---|---|
| -505 | The current query for input stream failed |
| -21 | The parameter for the secondary screen position entered is invalid. The secondary screen falls out of the boundary of the primary screen |
| -30xxx | The sessionid is used to operate on other output streams before the mixing of input streams under the sessionid is cancelled. |
| Other | Other errors. Please contact customer service for technical support. |

### 1.5 Cancellation of stream mixing

```
{
"timestamp":int(time.time()), # UNIX timestamp, which is the number of seconds that have elapsed since January 1, 1970 (Midnight in UTC/GMT)
"eventId":int(time.time()), # Stream mixing event ID. Use the timestamp at the backend
"interface":
{
"interfaceName":"Mix_StreamV2", # Fixed value: "Mix_StreamV2"
"para":
{
"app_id": appid, # Enter the LVB APPID
"interface": "mix_streamv2.start_mix_stream_advanced", # Fixed value"mix_streamv2.start_mix_stream_advanced"
"mix_stream_session_id" : "5000_enson", # Enter the stream ID of the primary VJ
"output_stream_id": "5000_enson11", # Enter the stream ID of the primary VJ
"input_stream_list":
[
# Primary VJ: Background image
{
"input_stream_id":"5000_enson11", # Stream ID
"layout_params":
{
"image_layer": 1 # Image layer ID: Primary VJ: 1; Secondary VJ: 2, 3, 4 in sequence
}
}
]
}
}
}
```

**Note:**
To cancel a stream mixing task, you can operate on the current input stream using the current sessionid. Operations on other output streams with this sessionid can be performed after half a minute.

# 2. Scenarios and Notes on Special Operations

## 2.1 Supported features

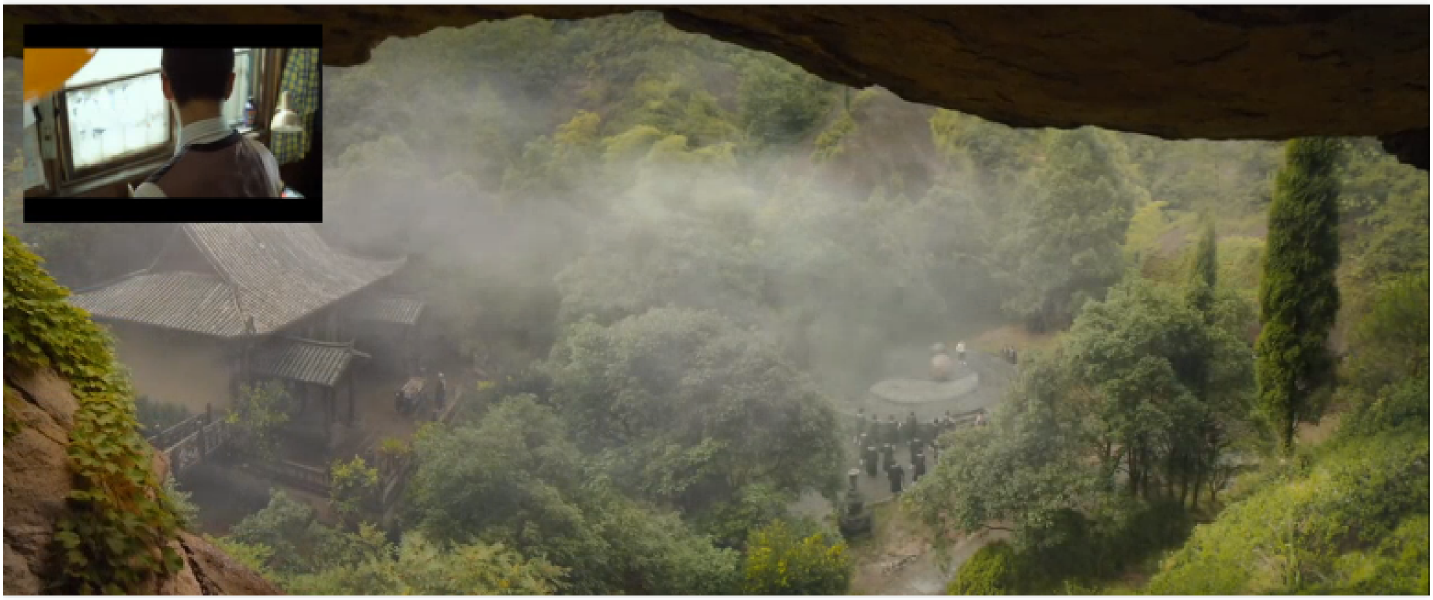
**Features**

| Feature Description | Using Condition | demo File Name |
|---|---|---|
| Mixing of 2-6 audio-video streams | The input sources are audios and videos | test_video_and_audio.py |
| Mixing of 2-6 audio-video streams and canvas | The input sources are audios, videos and canvas | test_stream_and_canvas.py |
| Mixing of 2-6 audio-video streams and video-only streams | The input sources are audio-video streams and video-only streams | test_stream_and_video.py |
| Mixing of 2-6 audio-video streams and audio-only streams | The input sources are audio-video streams and audio-only streams | test_stream_and_audio.py |
| Cropping on the cloud | The input sources are audio-video streams or video-only streams | test_stream_crop.py |
| Mixing of 2-8 audio-only streams | The input sources are audios only | test_audio_only.py |

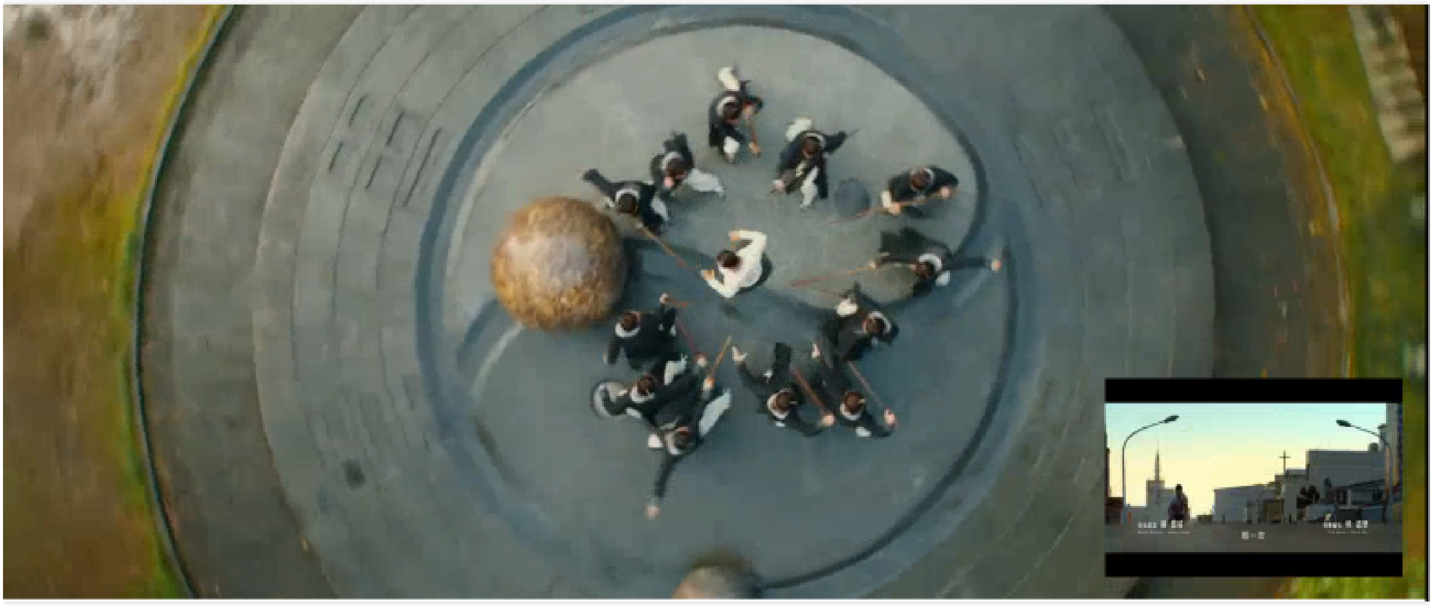
## 2.2 Common templates

Common templates include 10, 20, 30, 40, 310, 410, 510, and 610. When using the above templates, you don't need to enter position and length and width parameters for input streams, which are scaled in proportion to the original images. Only template ID needs to be passed.

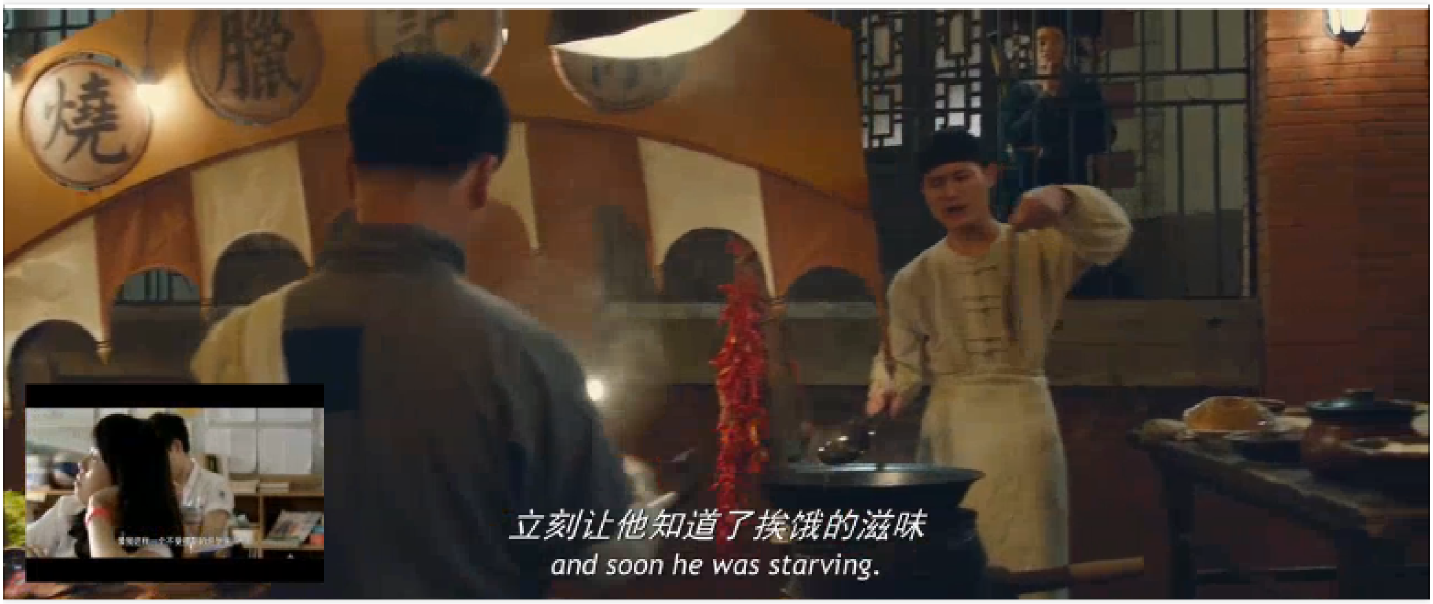
Template 10 preview:



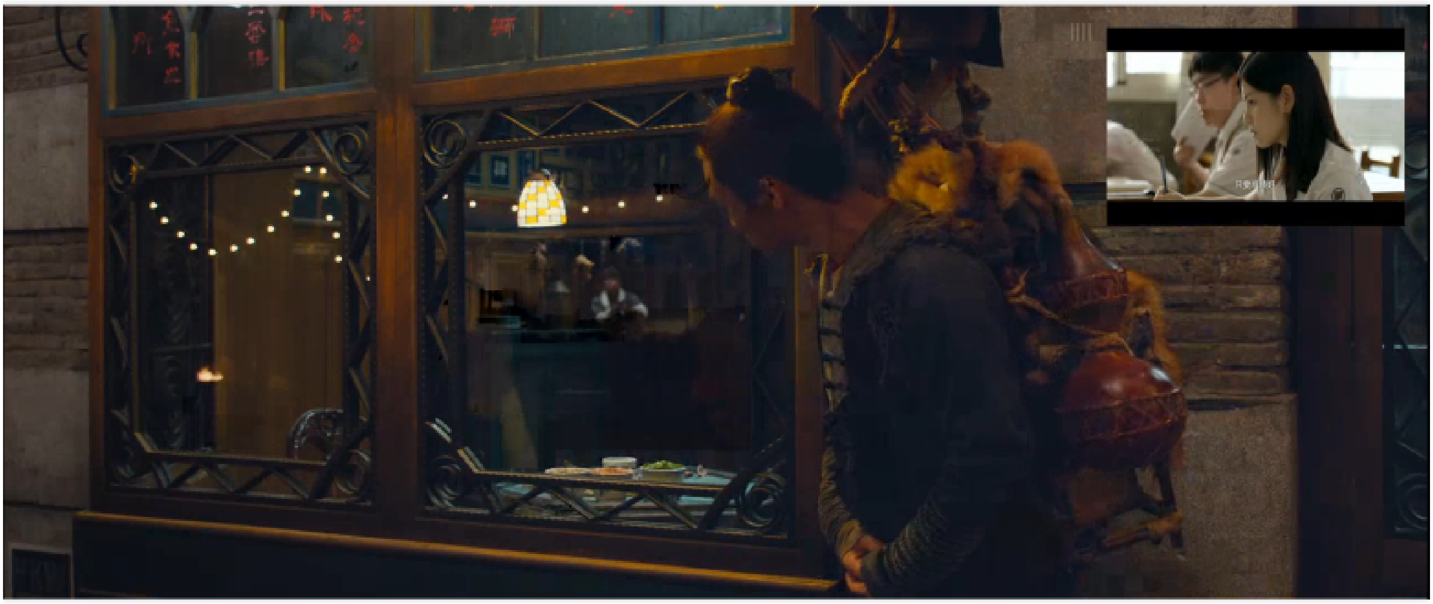
Template 20 preview:



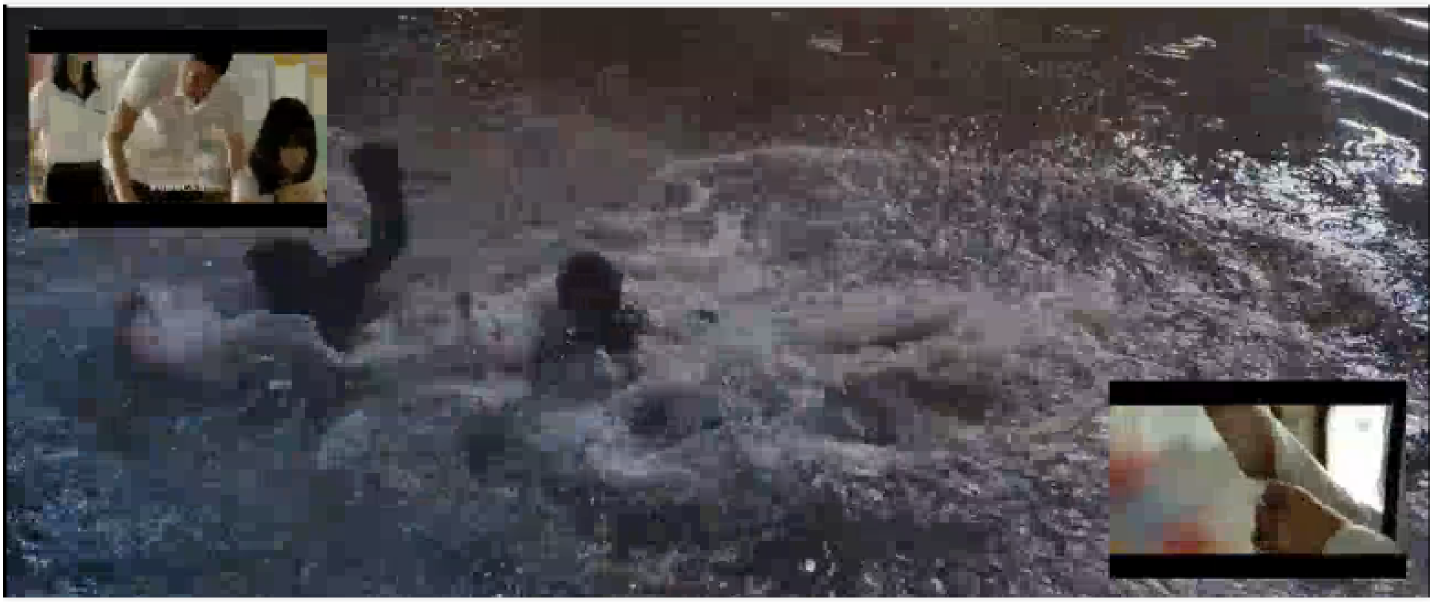
Template 30 preview:



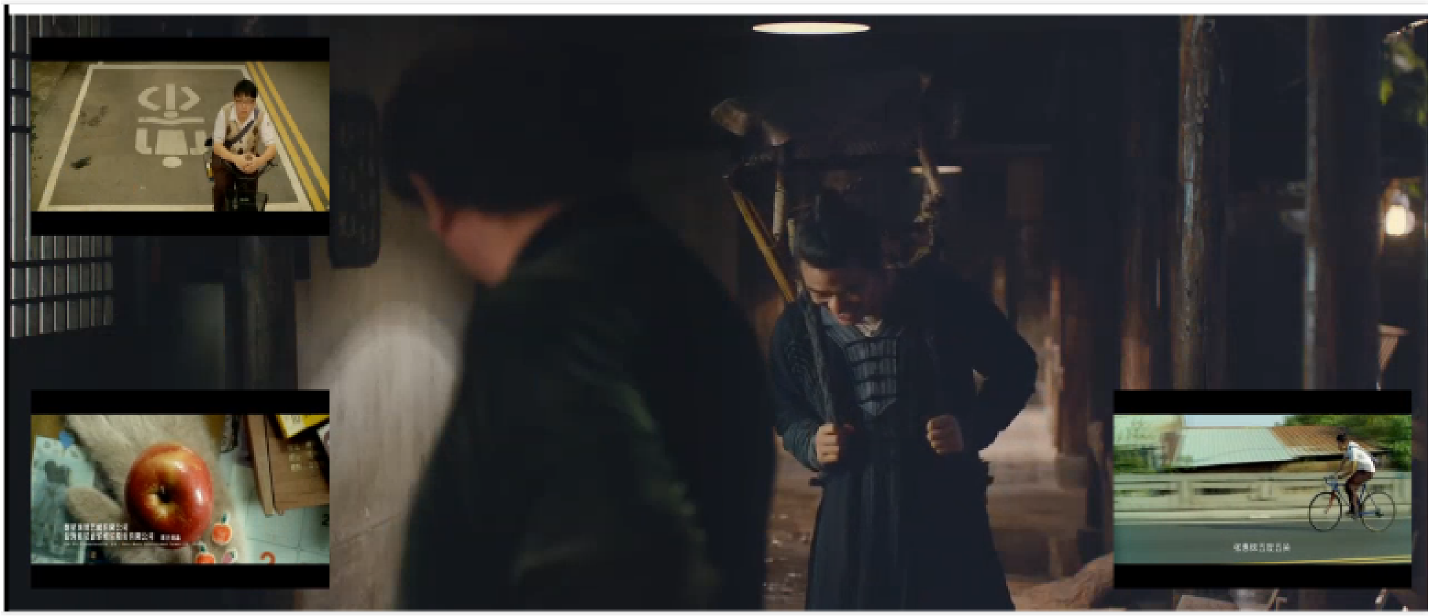
Template 40 preview:



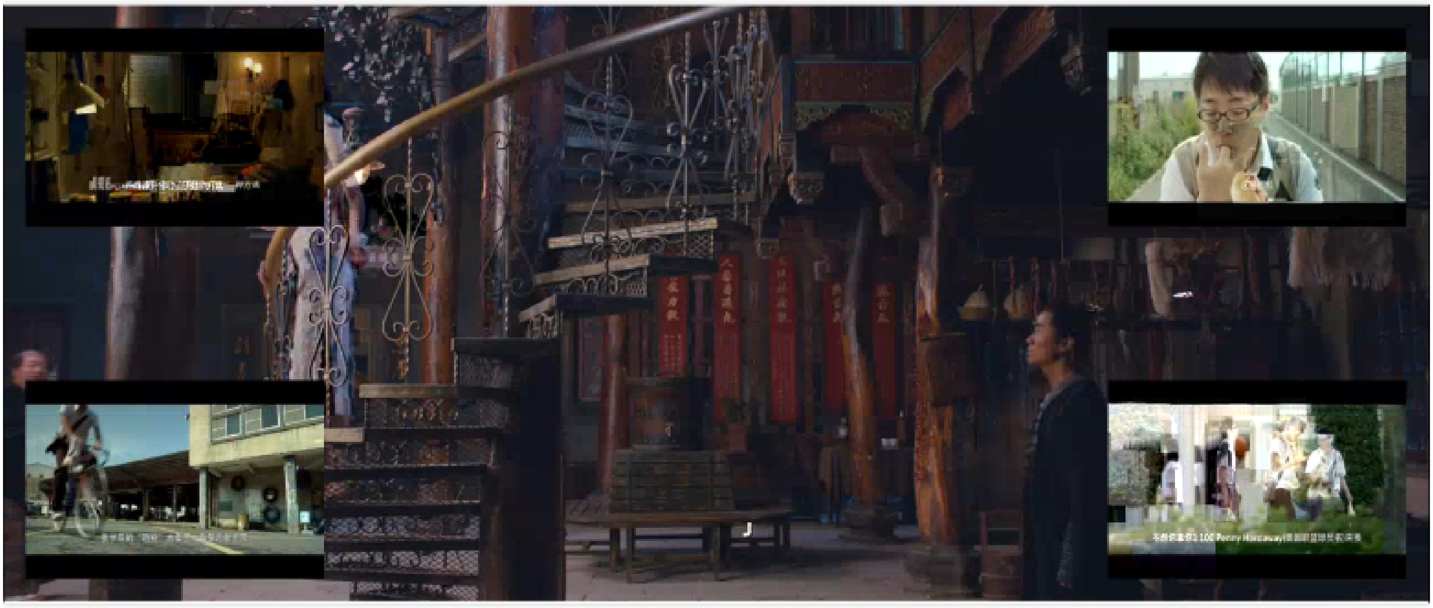
Template 310 preview:



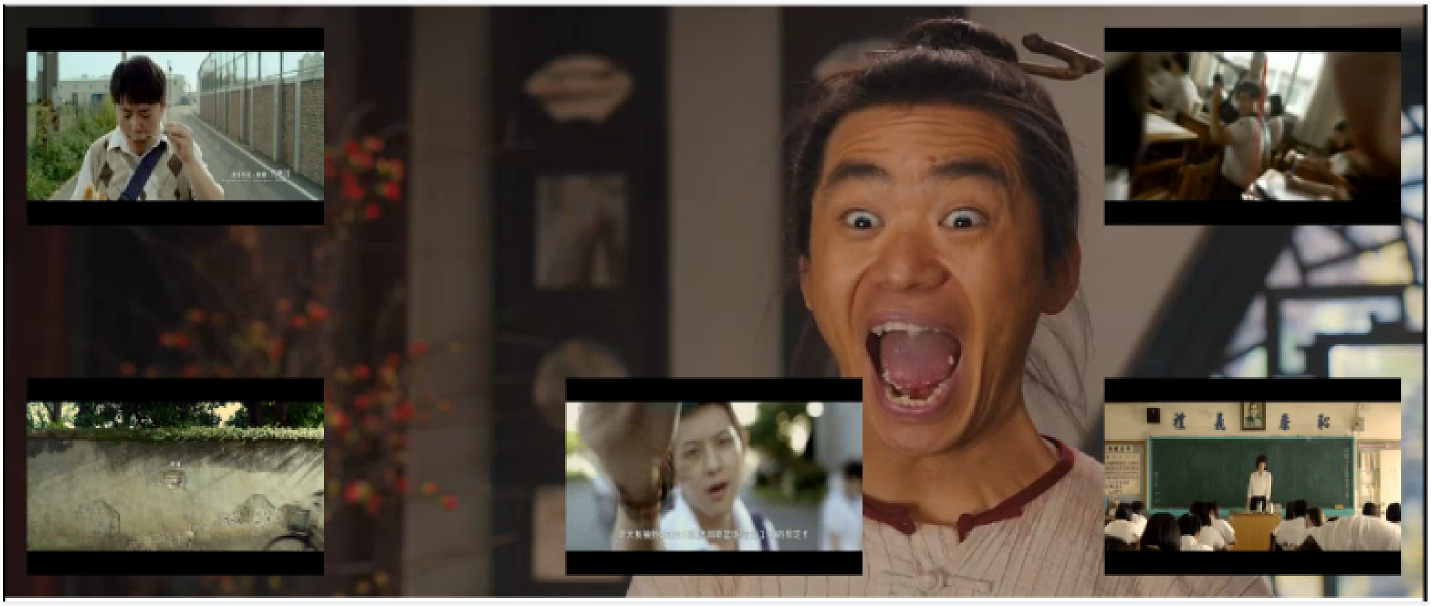
Template 410 preview:



Template 510 preview:



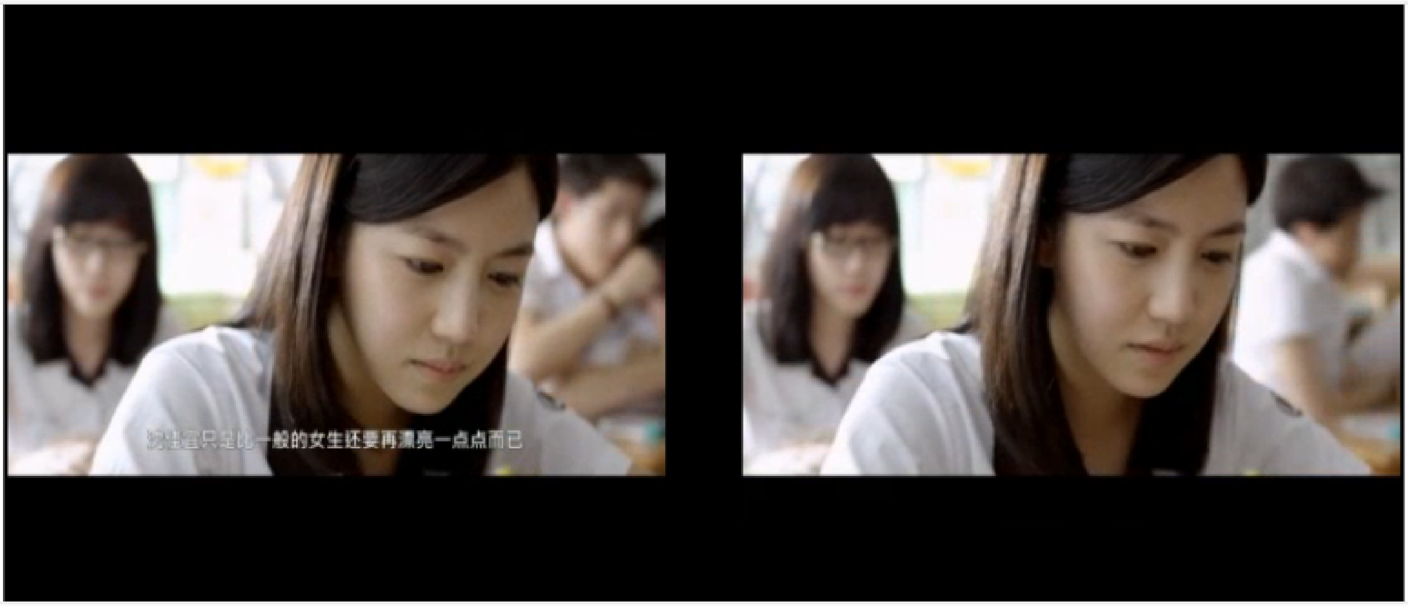
Template 610 preview:



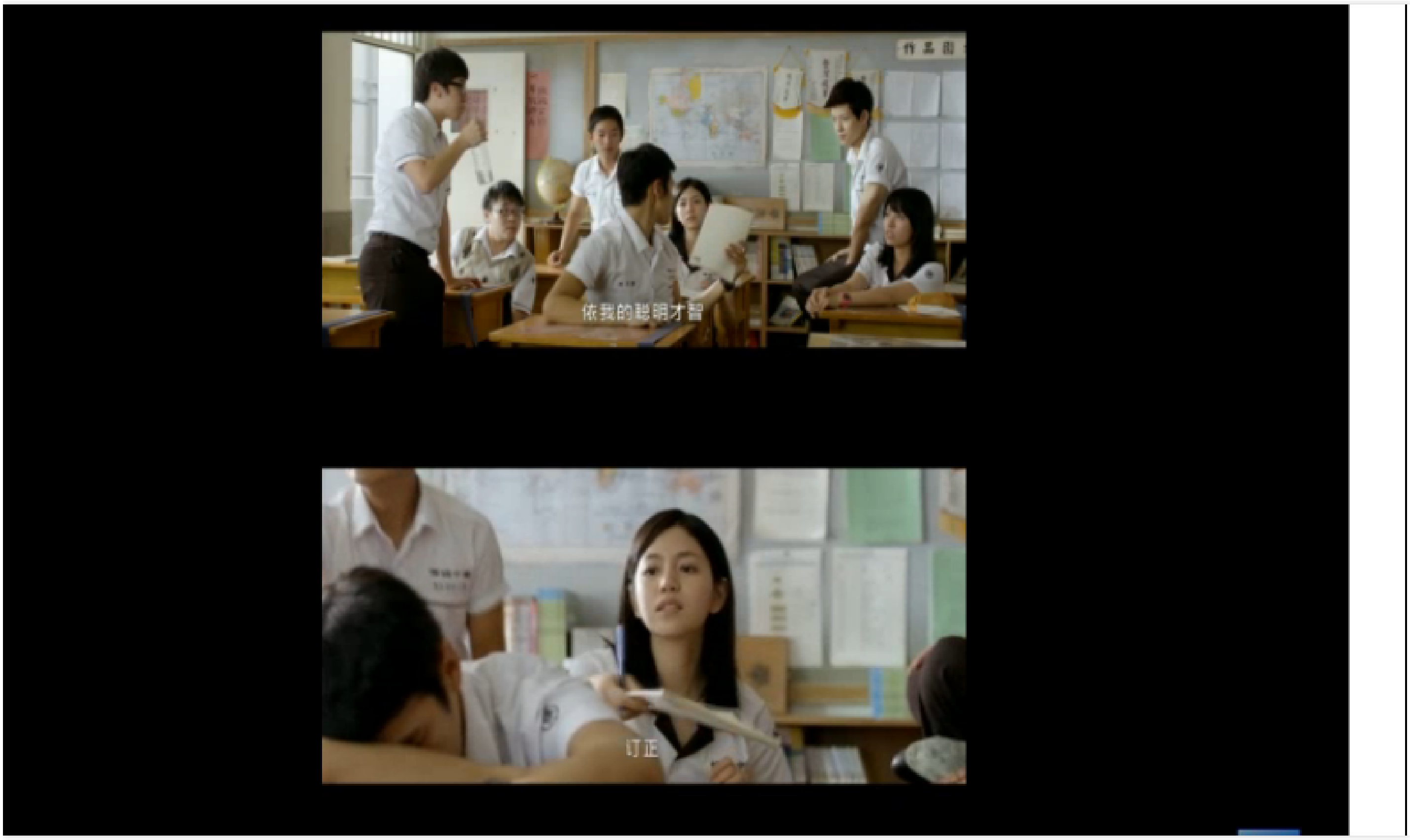
## 2.3 Notes on special templates

To meet the needs of some users for split screen comparison, we provide three special templates: 390, 391 and 590.

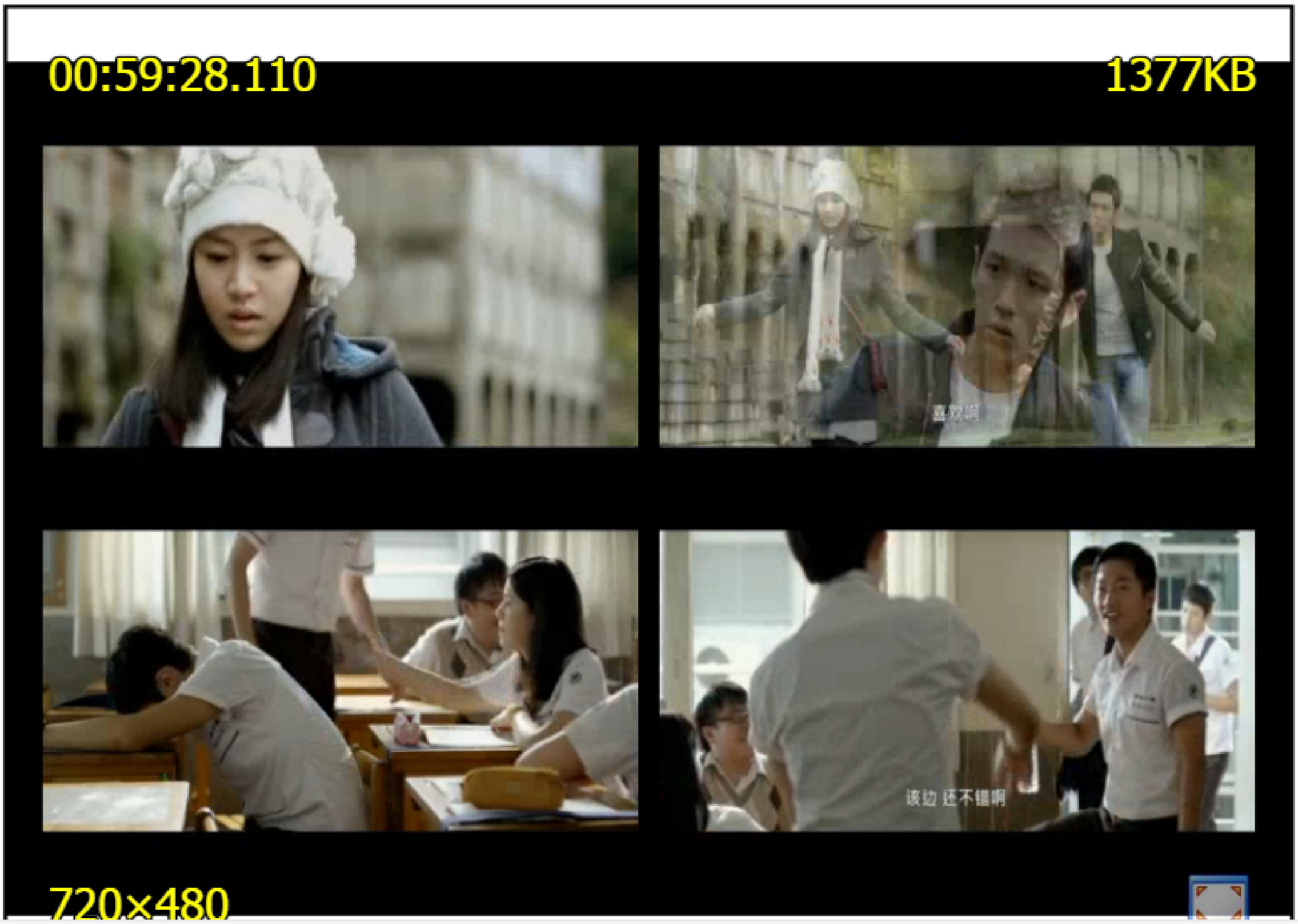
Template 390 preview:



Template 391 preview:



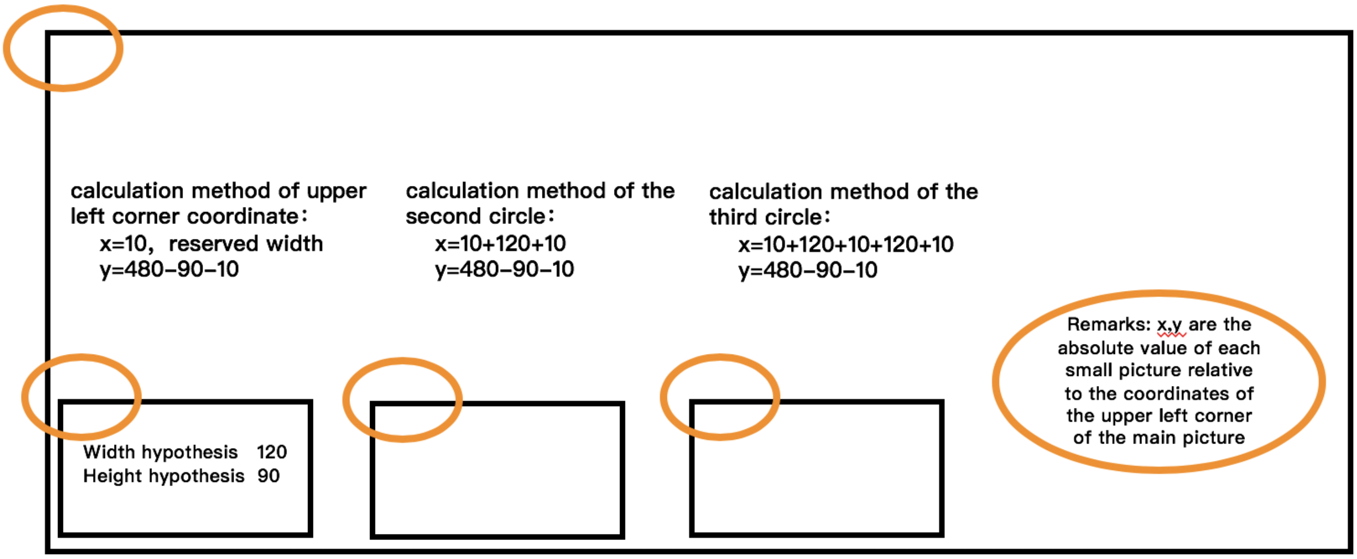
Template 590 preview:



**Note:**
When you use these templates, the input source with the image layer ID being 1 must be canvas, and its color can be customized.

## 2.4 How to enter custom position parameters

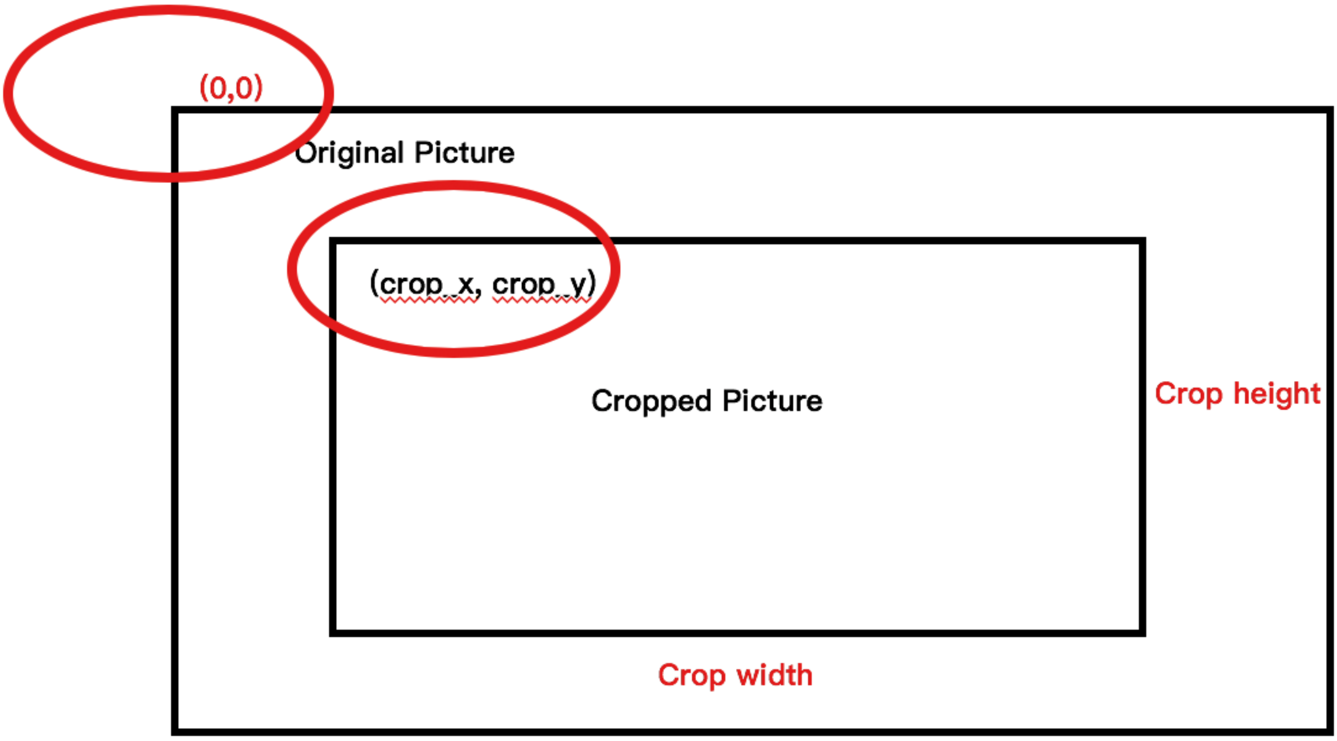
**Diagram:**


calculation method of upper left corner coordinate：
x=10，reserved width
y=480−90−10
calculation method of the second circle：
x=10+120+10
y=480−90−10
calculation method of the third circle：
x=10+120+10+120+10
y=480−90−10
Remarks: x,y are the absolute value of each small picture relative to the coordinates of the upper left corner of the main picture
Width hypothesis 120
Height hypothesis 90

As shown in the diagram above, the position parameters location_x and location_y are the absolute pixel distance from the top left corner of the secondary screen to that of the background image.

## 2.5 How to enter cropping position parameters

**Diagram:**



As shown in the diagram above:

a. Both the width and height and the position parameters you entered are absolute pixel values. Streams of different pixels are handled differently.
B. The position parameters crop_x and crop_y represent the absolute pixel distance from the top left corner of the original stream.

## 2.6 Demo downloading

**Download the stream mixing demo:** Click here to download the demo.

# Pause and Resume after Latency

Last updated：2018-08-01 21:03:08

## 1. API Description

- **API**
  **channel_manager**: This API is used to stop pushing stream and delay the availability of API.
- **URL**
  URL for calling API: `http://fcgi.video.qcloud.com/common_access` .
- **Purpose**
  This API can be called when VJ push is not allowed for certain reasons.
- **Note**
  This API is called to stop pushing a stream (which means to disable the push). To resume VJ push, you can call this API again or set a time for resuming push, and the push will be enabled at the specified time. A stream can be disabled for up to 3 months (3 months after the current time). If the suspension period is set to more than 3 months, 3 months shall prevail.

## 2. Input Parameters

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| APPID | Customer ID | int | LVB APPID used for identifying customers | Y |
| interface | API name | string | For example: Get_LivePushStat | Y |
| t | Validity period | int | UNIX timestamp (decimal) | Y |
| sign | Security signature | string | MD5(key+t) | Y |
| Param.s.channel_id | LVB Code | string | | Y |
| Param.n.abstime_end | The timestamp for the end time of stream suspension | int | The absolute time of stream suspension. Enter a UNIX timestamp (decimal). A stream can be disabled for up to 3 months. | |
| Y | | | | |

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| Param.s.action | Action | string | Interrupt stream: Forbid; Resume push: Resume | Y |

## 3. Output Parameters

| Parameter Name | Description | Type | Note |
|---|---|---|---|
| ret | Error code | int | 0: Successful; other values: Failed. |
| message | Error message | string | Error message |

## 4. Example

Purpose: To ban the LVB stream with an LVB Code of 8888_test123 whose content has violated relevant regulations.

| Component | Example |
|---|---|
| API URL | http://fcgi.video.qcloud.com/common_access? |
| APPID | 1234 |
| interface | Live_Channel_SetStatus |
| Param.s.channel_id | 8888_test123 |
| Param.n.abstime_end | 1499756910 |
| t | 1471850187 |
| sign | b17971b51ba0fe5916ddcd96692e9fb3 |
| Param.s.action | resume |

```
//When copying them, remove the invisible line breaks used for improving layout. Otherwise, URL constr
uction errors may occur, such as "appid is invalid".
URL = http://fcgi.video.qcloud.com/common_access?
appid=1234&interface=Live_Channel_SetStatus
&Param.s.channel_id=8888_test123
&Param.n.abstime_end=1499756910
```

```
&t=1471850187&sign=b17971b51ba0fe5916ddcd96692e9fb3
&Param.s.actio=resume
```

# Create Recording Task

Last updated：2018-08-08 16:16:49

## 1. API Description

- **API**
  **Live_Tape_Start**: This API is used to create a recording task.
- **URL**
  URL for calling API: `http://fcgi.video.qcloud.com/common_access`
- **Note**
  Recorded files are stored on the VOD platform. If you want to use the recording feature, you need to activate the VOD service. When recording files are stored, the charges (including charges for storage and downstream playback traffic) are calculated with VOD billing method. For more information, please see relevant document.
  Create a recording task. This API supports two recording modes: Scheduled Recording and Real-time Recording. In Scheduled Recording, importing the start time of a task is needed, which may cause the missing of wonderful moments in a video. In Real-time Recording, recording is synchronous with playback of a video, making it possible to capture all of the marvelous moments in a video. Note: API calling timeout should be greater than 3 seconds, because retries within 3 seconds and frequent calls may lead to duplicate recording tasks.

## 2. Input Parameters

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| APPID | Customer ID | int | LVB APPID used for identifying customers | Y |
| interface | API name | string | Enter Live_Tape_Start | Y |
| t | Validity period | int | UNIX timestamp (decimal) | Y |
| sign | Security signature | string | MD5(key+t) | Y |
| Param.s.channel_id | Channel ID | string | | Y |
| Param.s.start_time | Start time of the task | string | China Standard Time, which needs urlencode, such as: 2017-01-01 10:10:01 | Y |
| Param.s.end_time | End time of the task | string | China Standard Time, which needs urlencode, such as: 2017-01-01 10:10:01 | Y |

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| Param.n.task_sub_type | Whether to enable real-time recording | int | 1 - Enable; 0 - Disable. 1 is recommended.<br>a. Creating real-time recording tasks requires that VJs push streams actively. The video recording starts at the moment when this API is successfully called. In this case, the parameter of task start time is ignored.<br>b. The real-time recording task supports a maximum length of 30 minutes. If the time difference between the passed task end time and the current time is longer than 30 minutes, only a length of 30 minutes is recorded. The recommended recording length is within 5 minutes.<br>c. If real-time recording is disabled, the parameter of task start time is required, and the time difference between the end time and the start time should not be longer than 1 day. | Y |
| Param.s.file_format | Format of recorded files | string | Available values include flv (default), hls, mp4, and aac | N |
| Param.s.record_type | Type of recorded files | string | Default is video<br>When record_type is "video", file_format can be "flv", "hls", or "mp4"<br>When record_type is "audio", file_format can be "aac", "flv", "hls", or "mp4" | N |

### 3. Output Parameters

| Parameter Name | Description | Type | Note |
|---|---|---|---|
| ret | Error code | int | 0: Successful; other values: Failed. |
| message | Error message | string | Error message |
| output | Message content | array | For more information, please see the description below. |

"output" is composed as follows:

| Field Name | Description | Type | Note |
|---|---|---|---|
| task_id | Task ID | int | 64-bit unsigned integer |

### 4. Example

Purpose: To query the list of files recorded during the LVB for the LVB stream with an LVB Code of 8888_test123.

| Component | Example |
|---|---|
| API URL | `http://fcgi.video.qcloud.com/common_access?` |
| appid | 1234 |
| interface | Live_Tape_Start |
| Param.s.channel_id | 8888_test123 |
| Param.s.start_time | Start time of the task |
| Param.s.end_time | End time of the task |
| t | 1471850187 |
| sign | b17971b51ba0fe5916ddcd96692e9fb3 |

```
//When copying them, remove the invisible line breaks used for improving layout. Otherwise, URL construction errors may occur, such as "appid is invalid".
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
Note: Param.s.start_time and Param.s.end_time need to be encoded. After being encoded, 2017-05-20 10:00:00 is converted to 2017-05-20+10:00:00, and 2017-05-20 10:30:00 to 2017-05-20+10:30:00.
```

# End Recording Task

Last updated：2018-07-10 15:43:02

## 1. API Description

- **API**
  **Live_Tape_Stop**: This API is used to end a recording task.
- **URL**
  URL for calling API: `http://fcgi.video.qcloud.com/common_access`
- **Note**
  End a recording task.

## 2. Input Parameters

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| APPID | Customer ID | int | LVB APPID used for identifying customers | Y |
| interface | API name | string | For example: Get_LivePushStat | Y |
| t | Validity period | int | UNIX timestamp (decimal) | Y |
| sign | Security signature | string | MD5(key+t) | Y |
| Param.s.channel_id | Channel ID | string | | Y |
| Param.s.task_id | Task ID | string | | Y |
| Param.n.task_sub_type | Whether to enable real-time recording | int | Default is 0, and 1 means enabling real-time recording | N |

## 3. Output Parameters

| Parameter Name | Description | Type | Note |
|---|---|---|---|
| ret | Error code | int | 0: Successful; other values: Failed. |
| message | Error message | string | Error message |

## 4. Example

Purpose: To query the list of files recorded during the LVB for the LVB stream with an LVB Code of 8888_test123.

| Component | Example |
|---|---|
| API 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 |

```
//When copying them, remove the invisible line breaks used for improving layout. Otherwise, URL constr
uction errors may occur, such as "appid is invalid".
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
```

# Query API

# Query Live Status

Last updated：2018-07-11 14:51:57

## 1. API Description

- **API**
  **Live_Channel_GetStatus**: This API is used to query whether a stream has a status of Broadcasting.

- **URL**
  URL for calling API: `http://fcgi.video.qcloud.com/common_access`

- **Purpose**
  To query whether a stream has a status of Broadcasting. It is implemented based on Tencent Cloud's detection of audio/video stream interruption, and thus may not be as fast and accurate as the active reporting of application in terms of real-time capability. But it can serve as a good supplementary means for checking and clearing up LVB streams regularly.
- **Note**
  If the push LVB Code to be queried is never used to push streams, the 20601 error code is returned.

## 2. Input Parameters

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| APPID | Customer ID | int | LVB APPID used for identifying customers | Y |
| interface | API name | string | For example: Get_LivePushStat | Y |
| t | Validity period | int | UNIX timestamp (decimal) | Y |
| sign | Security signature | string | MD5(key+t) | Y |
| Param.s.channel_id | LVB Code | string | A single LVB stream can be queried at one time | Y |

**Note:**
For historical reasons, the LVB Code parameter was defined as channel_id in some earlier APIs, and is defined as stream_id in new APIs.

## 3. Output Parameters

| Parameter Name | Description | Type | Note |
|---|---|---|---|
| ret | Error code | int | 0: Successful; other values: Failed. |
| message | Error message | string | Error message |
| output | Message content | array | For more information, please see the description below. |

"output" is composed as follows:

| Field Name | Description | Type | Note |
|---|---|---|---|
| rate_type | Bitrate | int | 0: original bitrate; 10: LD; 20: HD |
| recv_type | Playback protocol | int | 1: RTMP/FLV; 2: HLS; 3: RTMP/FLV+HLS |
| status | Status | int | 0: Interrupted; 1: Enabled; 3: Disabled |

## 4. Example

Purpose: To query whether the LVB stream with the LVB Code of 8888_test123 is in a status **Broadcasting**.

| Component | Example |
|---|---|
| API URL | http://fcgi.video.qcloud.com/common_access? |
| APPID | 1234 |
| interface | Live_Channel_GetStatus |
| Param.s.channel_id | 8888_test123 |
| t | 1471850187 |
| sign | b17971b51ba0fe5916ddcd96692e9fb3 |

```
//When copying them, remove the invisible line breaks used for improving layout. Otherwise, URL constr
uction errors may occur, such as "appid is invalid".
URL = http://fcgi.video.qcloud.com/common_access?
appid=1234&interface=Live_Channel_GetStatus
&Param.s.channel_id=8888_test123
&t=1471850187&sign=b17971b51ba0fe5916ddcd96692e9fb3
```

# Query Statistical Info

Last updated：2018-07-10 16:44:47

## 1. API Description

- **API**
  **Get_LiveStat**: This API is used to query the push and playback information about a specified LVB stream.
  **Get_LivePushStat**: Return only the push statistics to improve query efficiency.
  **Get_LivePlayStat**: Return only the playback statistics to improve query efficiency.
- **URL**
  URL for calling API: `http://statcgi.video.qcloud.com/common_access`
- **Purpose**
  To query the statistics (such as number of viewers, bandwidth, bit rate, and frame rate) of an LVB stream.
  To query the statistics of multiple LVB streams that are being broadcast (paged query is recommended to avoid excessive returned data packet in each time)
- **Note**
  The statistics are instantaneous statistics at the query time point rather than historically accumulated data.
  If the target stream is not in "Broadcasting" status, the output field in the returned result is blank.
  The push statistics are updated once every **5 seconds**, which means there's no need to query at an interval shorter than 5 seconds.
  The playback statistics are updated once every **1 minute**, which means there's no need to query at an interval shorter than 60 seconds.
- **BETA**
  The statistic API is still in Beta and is not open to all customers. If you call this API when it is not activated, you will receive a prompt "cmd is invalid". Contact us if there's an urgent need.

## 2. Input Parameters

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| cmd | Customer ID | int | LVB APPID used for identifying customers | Y |
| interface | API name | string | For example: Get_LivePushStat | Y |
| t | Validity period | int | UNIX timestamp (decimal) | Y |
| sign | Security signature | string | MD5(key+t) | Y |
| Param.n.page_no | Page number | int | Starts from 1. Default value is 1 | N |

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| Param.n.page_size | Page size | int | 1-300. Default is 300 | N |
| Param.s.stream_id | LVB Code | string | If stream_id is not set, all streams that are in LVB will be queried | N |
| Param.s.pull_domain | Domain name for pulling stream | string | Playback domain name. If it is left empty, the playback statistics for all domain names are returned. | N |

## 3. Output Parameters

**Get_LiveStat**

| Parameter Name | Description | Type | Note |
|---|---|---|---|
| ret | Error code | int | 0: Successful; other values: Failed. |
| message | Error message | string | Error message |
| output | Message content | array | For more information, please see the description below. |

"output" is composed as follows:

| Field Name | Description | Type | Note |
|---|---|---|---|
| stream_count | Total number of online LVB streams | int | |
| stream_info | Statistics for the LVB stream | array | For more information, please see below. |
| total_bandwidth | Total bandwidth for the current account upon query | double | Unit: Mbps |
| total_online | Total number of online viewers for the current account upon query | int | |

"stream_info" is composed as follows:

| Field Name | Description | Type | Note |
|---|---|---|---|
| stream_name | LVB Code | string | - |
| bandwidth | Instantaneous bandwidth usage of the LVB stream | double | Unit: Mbps |

| Field Name | Description | Type | Note |
|---|---|---|---|
| online | Instantaneous number of live viewers for the LVB stream | int | |
| client_ip | Push client IP | string | - |
| server_ip | IP of the server that receives the stream | string | - |
| fps | Instantaneous frame rate of the push | int | - |
| speed | Instantaneous bit rate of the push | int | bps |

**Get_LivePushStat**

The output of Get_LivePushStat is a subset of Get_LiveStat:

| Field Name | Description | Type | Note |
|---|---|---|---|
| stream_count | Total number of online LVB streams | int | |
| stream_info | Statistics for the LVB stream | array | For more information, please see below. |

"stream_info" is composed as follows:

| Field Name | Description | Type | Note |
|---|---|---|---|
| stream_name | LVB Code | string | - |
| client_ip | Push client IP | string | - |
| server_ip | IP of the server that receives the stream | string | - |
| fps | Instantaneous frame rate of the push | int | - |
| speed | Instantaneous bit rate of the push | int | bps |

**Get_LivePlayStat**

The output of Get_LivePushStat is another subset of Get_LiveStat:

| Field Name | Description | Type | Note |
|---|---|---|---|
| stream_count | Total number of online LVB streams | int | |
| stream_info | Statistics for the LVB stream | array | For more information, please see below. |

| Field Name | Description | Type | Note |
|---|---|---|---|
| total_bandwidth | Total bandwidth for the current account upon query | double | Unit: Mbps |
| total_online | Total number of online viewers for the current account upon query | int | |

"stream_info" is composed as follows:

| Field Name | Description | Type | Note |
|---|---|---|---|
| stream_name | LVB Code | string | - |
| bandwidth | Instantaneous bandwidth usage of the LVB stream | double | Unit: Mbps |
| online | Instantaneous number of live viewers for the LVB stream | int | |

## 4. Example

Purpose: To query the push and playback information of all video streams that are being broadcasted under the current account

| Component | Example |
|---|---|
| API 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 |

```
//When copying them, remove the invisible line breaks used for improving layout. Otherwise, URL constr
uction errors may occur, such as "cmd is invalid".
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
```

# Query Recording File

Last updated：2018-08-03 16:56:58

## 1. API Description

- **API**
  **Live_Tape_GetFilelist**: This API is used to query the recorded files generated in a certain LVB stream during a certain period of time.
- **URL**
  URL for calling API: `http://fcgi.video.qcloud.com/common_access`
- **Note**
  You cannot determine when to call the API due to the uncertainty of the file generation time. Therefore, you are recommended to use Passive Event Notification (event_type = 100) mechanism.

## 2. Input Parameters

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| APPID | Customer ID | int | LVB APPID used for identifying customers | Y |
| interface | API name | string | For example: Get_LivePushStat | Y |
| t | Validity period | int | UNIX timestamp (decimal) | Y |
| sign | Security signature | string | MD5(key+t) | Y |
| Param.s.channel_id | LVB Code | string | | Y |
| Param.n.page_no | Page number | int | Starts from 1. Default value is 1 | N |
| Param.n.page_size | Page size | int | 1-100. Default is 10 | N |
| Param.s.sort_type | Sorting method | string | "asc" indicates ascending order, and "desc" indicates descending order. Default is "asc". | N |
| Param.s.start_time | Start time for the query | string | Format is: 2016-12-10 00:00:00 | N |
| Param.s.end_time | End time for the query | string | Format: 2016-12-10 00:00:00. The end time must be less than 24 hours later than the start time in the same day | N |

**Note:**
For historical reasons, the LVB Code parameter was defined as channel_id in some earlier APIs, and is defined as stream_id in new APIs.

### 3. Output Parameters

| Parameter Name | Description | Type | Note |
|---|---|---|---|
| ret | Error code | int | 0: Successful; other values: Failed. |
| message | Error message | string | Error message |
| output | Message content | array | For more information, please see the description below. |

"output" is composed as follows:

| Field Name | Description | Type | Note |
|---|---|---|---|
| all_count | Number of fragments | int | |
| file_list | Information of the fragment | array | For more information, please see the description below. |

"file_list" is composed as follows:

| Field Name | Description | Type | Note |
|---|---|---|---|
| vid | vid of the VOD file | string | "record_file_url" is used if the field is left empty. |
| start_time | Start time of the fragment | string | The time cannot be accurate to seconds due to interference of the I frame position |
| end_time | End time of the fragment | string | The time cannot be accurate to seconds due to interference of the I frame position |
| file_id | ID of the VOD file | string | This parameter is required to obtain playback URL with VOD API |

| Field Name | Description | Type | Note |
|---|---|---|---|
| record_file_url | Playback address | string | This field is used as the address. If left empty, the address will be the vid which is assembled based on assembly format |

**Note:**
vid assembly format (URL format for video fragment recording requests):
`http://(VODbizid).vod.myqcloud.com/(vid).f0.flv`

## 4. Example

Purpose: To query the list of files recorded during the LVB for the LVB stream with an LVB Code of 8888_test123.

| Component | Example |
|---|---|
| API 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 |

```
//When copying them, remove the invisible line breaks used for improving layout. Otherwise, URL constr
uction errors may occur, such as "appid is invalid".
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
```

# Query Channel List

Last updated：2018-10-09 15:54:52

## 1. API Description

- **API**
  **Live_Channel_GetChannelList**: This API is used to query channel list.
- **URL**
  URL for calling API: `http://fcgi.video.qcloud.com/common_access`
- **Purpose**
  To query the current channel list in LVB Code mode.
- **Note**
  You can query a list of channels with a certain status, such as channels that are active.

## 2. Input Parameters

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| APPID | Customer ID | int | | Y |
| interface | API name | string | | Y |
| t | Expiration time stamp | int | | Y |
| sign | Signature | string | md5(key+expiration time stamp) | Y |
| Param.n.status | 0: stream interrupted; 1: enabled; 3: disabled | int | Filter is not used by default | N |
| Param.n.page_no | Page number | int | Starts from 1. Default value is 1 | N |
| Param.n.page_size | Page size | int | 10-100. Default is 10 | N |
| Param.s.order_field | Sorting field | string | Available value: create_time. Default is create_time. | N |
| Param.n.order_type | Sorting method | int | 0 indicates ascending order. 1 indicates descending order. | N |

## 3. Output Parameters

| Parameter Name | Description | Type | Note |
|---|---|---|---|
| ret | Error code | int | 0: Successful. Other values: Failed |

| Parameter Name | Description | Type | Note |
|---|---|---|---|
| message | Error message | string | Error description |
| output | Message content | array | |

"output" is composed as follows:

| Field Name | Description | Type | Note |
|---|---|---|---|
| all_count | Total number | int | |
| channel_list | List | array | |

"channel_list" is composed as follows:

| Parameter Name | Description | Type | Note |
|---|---|---|---|
| channel_id | LVB Code ID | string | |

## 4. Example

Purpose: Query the list of active channels under the account

| Component | Example |
|---|---|
| API 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_type | 0 |

```
//When copying them, remove the invisible line breaks used for improving layout. Otherwise, URL constr
uction errors may occur, such as "appid is invalid".
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_type=0
```

# Query Live Channel List

Last updated：2018-07-11 14:57:07

## 1. API Description

- **API**
  **Live_Channel_GetLiveChannelList**: This API is used to query channel list.
- **URL**
  URL for calling API: `http://fcgi.video.qcloud.com/common_access`
- **Purpose**
  To query channel list for LVB in LVB Code mode.

## 2. Input Parameters

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| APPID | Customer ID | int | | Y |
| interface | API name | string | | Y |
| t | Expiration time stamp | int | | Y |
| sign | Signature | string | md5(key+expiration time stamp) | Y |

## 3. Output Parameters

| Parameter Name | Description | Type | Note |
|---|---|---|---|
| ret | Error code | int | 0: Successful. Other values: Failed |
| message | Error message | string | Error description |
| output | Message content | array | |

"output" is composed as follows:

| Field Name | Description | Type | Note |
|---|---|---|---|
| all_count | Total number | int | |
| channel_list | List | array | |

"channel_list" is composed as follows:

| Parameter Name | Description | Type | Note |
|---|---|---|---|
| channel_id | LVB Code ID | string | |

### 4. Example

Purpose: To query the channel list in LVB under the account.

| Component | Example |
|---|---|
| API URL | `http://fcgi.video.qcloud.com/common_access?` |

```
//When copying them, remove the invisible line breaks used for improving layout. Otherwise, URL construction errors may occur, such as "appid is invalid".
URL = http://fcgi.video.qcloud.com/common_access?
appid=1234&interface=Live_Channel_GetChannelList
&t=1471850187&sign=b17971b51ba0fe5916ddcd96692e9fb3
```

# Getting Billing Bandwidth Data

Last updated：2018-07-10 16:56:51

## 1. API Description

- **API**
  **Get_BillingBandwidth**: This API is used to query billing bandwidth data of accounts.

- **URL**
  URL for calling API: `http://statcgi.video.qcloud.com/common_access`

- Backend configuration is required to use the API. To call the API, contact Tencent service personnel or submit a ticket. Tel: 4009-100-100.

## 2. Input Parameters

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| cmd | Business appid | Enter the LVB appid, which is used to identify different customers | Y | |
| interface | API name | string | Get_BillingBandwidth | Y |
| t | Validity period | int | UNIX timestamp (decimal) | Y |
| sign | Security signature | string | MD5(key+t) | Y |
| Param.n.start_time | Start time for the query | int | Timestamp | Y |
| Param.n.end_time | End time for the query | int | Timestamp | Y |
| Param.s.domain | Domain name | string | Domain name | N |

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| Param.n.home_foreign | Domestic and abroad data | uint | 0: Query total data despite of domestic or abroad (default)<br>1: Query domestic data only<br>2: Query abroad data only | N |
| Param.n.get_top_bd | Obtain peak bandwidth | Uint | 0: Obtain the list instead of peak bandwidth and traffic (default)<br>1: Obtain peak bandwidth and traffic | N |

### 3. Output Parameters

| Parameter Name | Description | Type | Note |
|---|---|---|---|
| ret | Error code | int | 0: Successful. Other values: Failed |
| message | Error message | string | Error description |
| output | Message content | array | None |

"output" is composed as follows:

| Field Name | Description | Type | Note |
|---|---|---|---|
| total_info | Statistics information for total billing bandwidth | array | Granularity of 5 minutes |
| domain | Domain name | string | Available only after input parameters are passed |

"total_info" is composed as follows:

| Field Name | Description | Type | Note |
|---|---|---|---|
| time | Statistical time | string | |
| bandwidth | Bandwidth | double | Mbps |
| flux | Traffic | double | MB |

### 4. Example

Purpose: To query the billing bandwidth data of accounts.

| Component | Example |
|---|---|
| API 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 |
| sign | b17971b51ba0fe5916ddcd96692e9fb3 |

```
//When copying them, remove the invisible line breaks used for improving layout. Otherwise, URL constr
uction errors may occur, such as "appid is invalid".
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
```

# Statistics API

# Get Push History Info

Last updated：2018-07-10 17:04:20

## 1. API Description

- **API**
  This API (**Get_LivePushStatHistory**) is used to obtain the historical statistics of push.
- **URL**
  URL for calling API: `http://statcgi.video.qcloud.com/common_access`
- **Note**
  Used to obtain the push information for specified time period.
  The push statistics is updated every 5 seconds.
- Backend configuration is required to use the API. To call the API, contact Tencent service personnel or submit a ticket. Tel: 4009-100-100.

## 2. Input Parameters

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| cmd | Business appid | int | The LVB appid, which is used to identify different customers | Y |
| interface | API name | string | | Y |
| t | Expiration timestamp | int | | Y |
| sign | Signature | string | md5 (key+expiration timestamp) | Y |
| Param.n.start_time | Start time for the query | int | Limited to the last 3 days from the timestamp | Y |
| Param.n.end_time | End time for the query | int | It is recommended to limit the time span for query between start and end times to 2 hours | Y |
| Param.s.stream_id | Steam ID | string | | Y |

## 3. Output Parameters

| Parameter Name | Description | Type | Remarks | Note |
|---|---|---|---|---|
| ret | Error code | int | 0: Successful; other values: Failed. | |
| message | Error message | string | Error message | |
| output | Message content | array | For more information, please see the description below. | |

"output" is composed as follows:

| Parameter Name | Description | Type | Remarks | Note |
|---|---|---|---|---|
| stream_info | Statistics of the time stream of push | list | Array (list) | |

"stream_info" is composed as follows:

| Parameter Name | Description | Type | Remarks | Note |
|---|---|---|---|---|
| time | Time when push starts | string | | |
| client_ip | Push client IP | string | | |
| server_ip | IP of the server that receives the stream | string | | |
| fps | Frame rate of the push | int | | |
| speed | Bitrate of the push | int | | In bps |

## 4. Example

Purpose: To query the list of files recorded during the LVB for the LVB stream with an LVB Code of 8888_test123.

| Component | Example |
|---|---|
| API URL | `http://statcgi.video.qcloud.com/common_access?` |
| cmd | 1234 |
| interface | Get_LivePushStatHistory |
| Param.n.start_time | 1453279831 |
| Param.n.end_time | 1453279835 |

| Component | Example |
|---|---|
| Param.s.stream_id | 1234_xxx |
| t | 1471850187 |
| sign | b17971b51ba0fe5916ddcd96692e9fb3 |

```
//When copying them, remove the invisible line breaks used for improving layout. Otherwise, URL constr
uction errors may occur, such as "cmd is invalid".
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
```

# Get Play Statistics History Info

Last updated：2018-07-10 17:05:16

## 1. API Description

- **API**
  This API (**Get_LivePlayStatHistory**) is used to obtain the historical statistics of playback.
- **URL**
  URL for calling API: `http://statcgi.video.qcloud.com/common_access`
- **Note**
  - Used to obtain the playback information for specified time period.
  - The playback statistics is updated every 1 minute.
  - This API supports LVB code mode and channel mode.
  - Backend configuration is required to use the API. To call the API, contact Tencent service personnel or submit a ticket. Tel: 4009-100-100.

## 2. Input Parameters

| Parameter Name | Description | Type | Note | Required |
|---|---|---|---|---|
| cmd | Business ID | int | The LVB appid, which is used to identify different customers | Y |
| interface | API name | string | | Y |
| t | Expiration timestamp | int | | Y |
| sign | Signature | string | md5 (key+expiration timestamp) | Y |
| Param.n.start_time | Start time for the query | int | Limited to the last 15 days from the timestamp | Y |
| Param.n.end_time | End time | int | It is recommended to limit the timestamp between start and end times to 2 hours | Y |
| Param.s.stream_id | Steam ID | string | If it is left empty, the total bandwidth is obtained | N |
| Param.s.domain | Domain name | String | If it is left empty, all the data under the APPID is obtained. The original playback domain name before cname is required | N |

## 3. Output Parameters

| Parameter Name | Description | Type | Remarks | Note |
|---|---|---|---|---|
| ret | Error code | int | 0: Successful; other values: Failed. | |
| message | Error message | string | Error message | |
| output | Message content | array | For more information, please see the description below. | - |

"output" is composed as follows:

| Parameter Name | Description | Type | Remarks | Note |
|---|---|---|---|---|
| stat_info | Statistics for the LVB stream | array | | |
| sum_info | Traffic and summation information | array | | Required output parameter |
| domain | Domain Name | string | | Optional output parameter |
| stream_id | Steam ID | string | | Optional output parameter |

"stat_info" is composed as follows:

| Parameter Name | Description | Type | Remarks | Note |
|---|---|---|---|---|
| time | Statistical time | string | | |
| bandwidth | Bandwidth | double | | In Mbps |
| online | Number of online users | int | | |
| flux | Traffic | double | | In MB |

"sum_info" (summary) is composed as follows:

| Parameter Name | Description | Type | Remarks | Note |
|---|---|---|---|---|
| sum_flux | Traffic summation | double | Required output parameter | In MB |

## 4. Example

Purpose: To query the list of files recorded during the LVB for the LVB stream with an LVB Code of 8888_test123.

| Component | Example |
|---|---|
| API 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 |

```
//When copying them, remove the invisible line breaks used for improving layout. Otherwise, URL constr
uction errors may occur, such as "cmd is invalid".
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
```

# Channel Hosting (old)

# Call Method

# Request Body

Last updated：2018-07-10 17:26:19

## 1. Service Address

The domain name for the Tencent Cloud LVB service is live.api.qcloud.com.

## 2. Communication Protocol

All the Tencent Cloud APIs communicate over HTTPS to provide high-security channels.

## 3. Request Methods

POST and GET requests are both supported, but cannot be used at the same time. If the GET method is used, the parameters are obtained from Querystring. If the POST method is used, the parameters are obtained from Request Body, and the parameters in the Querystring are ignored. The request parameters are organized in the same way in both types of requests. Generally, GET method is used. If the parameter strings are too long, POST method is used. For more information, please see the relevant API description.

## 4 Character Encoding

UTF-8 encoding is always used.

## 5. API Request Structure

| Name | Description | Notes |
|---|---|---|
| API entry | The WebService entry for calling API | https://live.api.qcloud.com/v2/index.php |
| Common parameters | The parameters common to all the APIs | For more information, please see "6. Common Parameters" section in this document. |

| | | |
|---|---|---|
| Instruction name | Name of the instruction to be executed by API, specified with Action.<br>For example, Action=CreateLVBChannel | For more information on complete instructions, please see API Overview |
| Instruction parameters | Parameters required for each specific instruction | For more information, please see the API document of each instruction. |

## 6. Common Parameters

Common parameters are used for user identification and API authentication. Unless necessary, these parameters will not be discussed in each API document. A request that comes with these parameters can be initiated successfully.

| Name | Type | Description | Required |
|---|---|---|---|
| Action | String | Name of API instruction, for example: DescribeDomains | Yes |
| Region | String | Region parameter, which is used to identify the region to which the instance you want to work with belongs. Available values:<br>bj: Beijing<br>gz: Guangzhou<br>sh: Shanghai<br>hk: Hong Kong<br>ca: North America | Yes |
| Timestamp | UInt | Current Unix timestamp | Yes |
| Nonce | UInt | A random positive integer, which is used in conjunction with Timestamp to prevent replay attacks. | Yes |
| SecretId | String | SecretId and SecretKey used for identification and applied for through the Tencent Cloud platform. SecretKey is used to generate a Signature.<br>For more information, please see API Authentication page. | Yes |
| Signature | String | Request signature, which is used to verify the validity of the request.<br>For more information, please see API Authentication page. | Yes |

The following shows a typical API request. "Action=DescribeInstance" indicates a query for the details of the CVM instance.

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

instanceId is an instruction parameter, and others are common parameters.

# Backend SDK

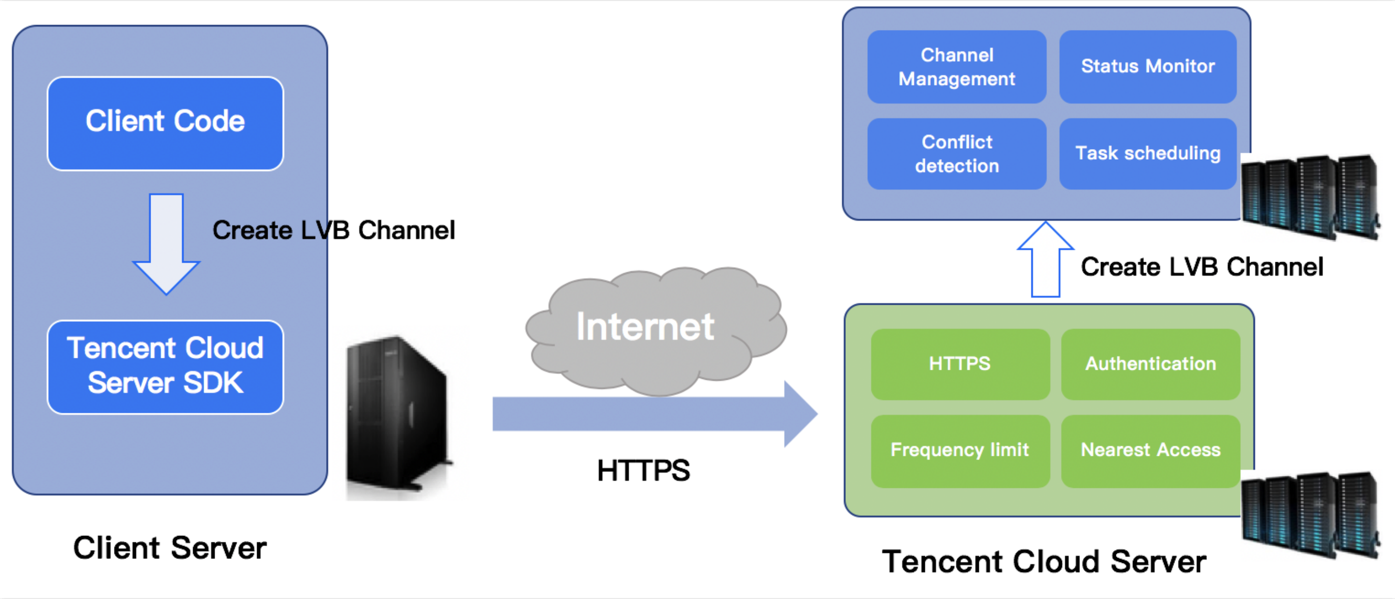
Last updated：2018-07-24 15:27:33

## Calling Method

How to connect to Tencent Cloud CVM is a problem we need to solve first. Tencent Cloud server cluster contains two layers:

- **Access layer:**
  It is responsible to access the external request for calling. For example, to limit the call frequency to 100 times per minute, or to verify the identity of the Tencent Cloud client server (based on SecretId and SecrectKey).

- **Function layer:**
  It "hides" behind the access layer. Since the security confirmation and attack blocking have been implemented at the access layer, all the function layer needs to do is functional processing. The basic features such as channel management and file query are implemented at this layer.



To call the Tencent Cloud server API, your server needs to meet the requirements of the access layer:

- The request protocol must be HTTPS
- The request header must conform to Tencent Cloud's access format
- The request header must contain correct SecretId and Signature

Therefore...

## Server SDK

Tencent Cloud Server SDK is recommended. Versions compiled in the following languages are available:

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

These SDKs are designed to help you connect to the Tencent Cloud CVM in the simplest way and to protect the intermediate linkage from attack by ensuring its security and reliability.

## Troubleshooting

An error code may be returned if you call the Tencent Cloud server API. It consists of "code", which is Tencent Cloud's **access layer error code**, and "**message**", which represents the detailed description of the error code.

### unicode

```
{"code":4600,"message":"\u534f\u8bae\u4e0d\u652f\u6301\uff0c\u8bf7\u53c2\u8003\u6587\u6863\u8bf4\u660e\u3002"}
```

"message" is displayed as an Unicode string in some platforms, which can be directly escaped in Chrome browser.

> Open Chrome, press F12 to enter the developer mode, and input the following content in Console panel at the bottom right corner:
> "\u534f\u8bae\u4e0d\u652f\u6301\uff0c\u8bf7\u53c2\u8003\u6587\u6863\u8bf4\u660e\u3002"
> "The protocol is not supported. For more information, please see the relevant document." is returned after escape.

### Dual Error Codes

Generally, messages are written in standard English, as shown below:

```
{"code":5100,"message":"20108:Channel is not in an editable status"}
```

You can see two error codes here:

- **5100**: This error code is returned from the access layer. For example, 4100 refers to authentication error (that is, verification failed in Tencent Cloud backend system because SecretId and Signature you entered in the request header is incorrect). In this case, your request is blocked at the access layer instead of being sent to the function layer.

  > The error code 5100 is special. It means that "No error occurs during security check at the access layer. A message from the function layer indicates that there is an error with your request."

- **20200**: This error code is returned via a specific function module from the function layer. For example, 20108 indicates "Channel is not in an editable status", which means that your request is being processed in LVB system. However, a specific error notification is returned because the system considers that there is an error with your request.

For more information, please see Error Codes.

# Return Results

Last updated：2018-07-10 17:26:53

## Composition of Returned Values

Unless otherwise specified, the returned values of each request contain the following fields:

| Name | Type | Description | Required |
|---|---|---|---|
| code | Int | Error code on the result. 0: Successful; other values: Failed. For meanings of error codes, please see the Error Codes page. | Yes |
| message | String | Request result | Yes |

For example:
Example requests that use common parameters:

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

Possible returned result is as follows:

```
{
"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
}]
}
```

## Error Codes

The error code in the response body summarizes the result of the calling and execution of a Tencent Cloud API.

Any error code other than 0 indicates the request is not properly executed. An error message describes the error in details. You can get the API execution result based on the error code.

On some terminals, such as browsers, messages in Chinese are displayed in Unicode and need to be decoded.

**The following error codes may be returned by Tencent Cloud APIs:**

| Error Code | Error Type | Description |
|---|---|---|
| 4000 | Invalid request parameter | Required parameter is missing, or parameter value is in an incorrect format. For error message, please see the "message" field in error description. |
| 4100 | Authentication failed | Signature authentication failed. For more information, please see the "Authentication" section in the document. |
| 4200 | Request expired | The request has expired. For more information, please see the "Request Validity" section in the document. |
| 4300 | Access denied | Account is blocked or not within the user range of the API. |
| 4400 | Quota exceeded | The number of requests exceeded the quota limit. For more information, please see the "Request Quota" section in the document. |
| 4500 | Replay attack | The Nonce and Timestamp parameters ensure that each request is executed only once on the server. Therefore, the Nonce value cannot be the same as the last one, and the difference between Timestamp and Tencent server time cannot be greater than 2 hours. |
| 4600 | Unsupported protocol | The protocol is not supported. For more information, please see the relevant document. |
| 5000 | Resource does not exist | The instance corresponding to the resource ID does not exist, or the instance has been returned, or another user's resource is accessed. |
| 5100 | Resource operation failed | The operation performed on the resource failed. For error message, please see the "message" field in error description. Try again later or contact customer service. |
| 5200 | Failed to purchase the resource | The resource purchase failed. This may be caused by unsupported instance configuration or insufficient resource. |
| 5300 | Failed to purchase the resource | The resource purchase failed because of insufficient balance. |
| 5400 | Part of | The batch operation was successful on some resources. For more |

| | | |
|---|---|---|
| | operations performed successfully | information, please see the returned value of the method. |
| 5500 | User failed to pass identity verification | The resource purchase failed because the user failed to pass identity verification. |
| 6000 | Internal error with the server | An internal error occurred with the server. Try again later or contact customer service. |
| 6100 | Not supported in the version | This API is not supported in this version or is under maintenance. Note: When this error occurs, check whether the domain name for the API is correct. Domain name may vary with different modules. |
| 6200 | API is unavailable | The API is under maintenance and is unavailable. Try again later. |

# Format of Returned Results for Asynchronous Task APIs

### 1. Format of returned results for common asynchronous task APIs

With such asynchronous task APIs, only one resource can be operated for each request, for example, creating load balancer or resetting OS for server.

| Name | Type | Description | Required |
|---|---|---|---|
| code | Int | Error code on the result. 0: Successful; other values: Failed. | Yes |
| message | String | Error message on the result | No |
| requestId | String | Task ID | Yes |

### 2. Format of returned results for batch asynchronous task APIs

With such asynchronous task APIs, multiple resources can be operated for each request, for example, changing passwords, starting or shutting down servers.

| Name | Type | Description | Required |
|---|---|---|---|
| code | Int | Error code on the result. 0: Successful; other values: Failed. | Yes |
| message | String | Error message on the result | No |
| detail | Array | The code, message, and requestId for an operation performed on the resource based on the resource ID (key). | Yes |

For example:

```
{
"code":0,
"message": "success",
"detail":
{
"qcvm6a456b0d8f01d4b2b1f5073d3fb8ccc0":
{
"code":0,
"message":"",
"requestId":"1231231231231":,
}
"qcvm6a456b0d8f01d4b2b1f5073d3fb8ccc0":
{
"code":0,
"message":"",
"requestId":"1231231231232":,
}
}
}
```

> Note:
> If the operations are successful for all resources, the outermost code is 0.
> If the operations fail for all resources, the outermost code returns 5100.
> If the operations fail for some resources, the outermost code returns 5400.
> In the third case, the terminal can obtain the information about the failed operations via "detail" field.

## Sample Code

### 1. Download SDK code

**Sample Code in PHP**
**Sample Code in JAVA**
**Sample Code in Python**
**Sample Code in .NET**

Replace the YOUR_SECRET_ID and YOUR_SECRET_KEY in the sample code with the actual SecretId and SecretKey.
The sample code is for reference only. Please use the code based on your actual needs.

### 2. Sample code in PHP

```
<?php


/***************In practice, the following parameters need to be changed based on the APIs called.*********************************/
/***************The DescribeInstances is taken as an example to describe how to obtain the cloud server with the specified instanceId.**********/

/*The URL of the API DescribeInstances is cvm.api.qcloud.com, which can be obtained from the "1. API Description" section of the API document.*/
$HttpUrl="cvm.api.qcloud.com";

/*Unless otherwise specified, all APIs other than MultipartUploadVodFile support GET and POST methods.*/
$HttpMethod="GET";

/*Most APIs are based on HTTPS protocol, except such APIs as MultipartUploadVodFile.*/
$isHttps =true;

/*Your key is required. You can obtain SecretId and $secretKey from https://console.cloud.tencent.com/capi.*/
$secretKey='XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX';


/*The following five parameters are the common parameters of all APIs. For some APIs that are not region-specific (e.g. DescribeDeals), the Region parameter is not required.*/
$COMMON_PARAMS = array(
'Nonce'=> rand(),
'Timestamp'=>time(NULL),
'Action'=>'DescribeInstances',
'SecretId'=> 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX',
'Region' =>'gz',
);

/*The following two parameters are the ones specific to the API DescribeInstances and are used to query the specific CVM list.*/
$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";

/**************Sort the request parameters in ascending lexicographical order by their names on a case-sensitive basis.*************/
$ReqParaArray = array_merge($COMMON_PARAMS, $PRIVATE_PARAMS);
ksort($ReqParaArray);

/*********************************Generate the original signature text.**********************************
* Combine the request method, URL and sorted request parameters into the following format to generate the original signature text. In this example, the original signature text is as follows:
* 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."?";

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

/*In the combination of original signature text, any "_" in a parameter name should be replaced with ".".
*/
if(strpos($key, '_'))
{
$key = str_replace('_', '.', $key);
}

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

/********************Generate a Signature based on the original signature string $SigTxt*****************/
$Signature = base64_encode(hash_hmac('sha1', $SigTxt, $secretKey, true));


/**************Combine the request strings together. The request parameters and the signature string need to be encoded using urlencode.*******************/
```

```
$Req = "Signature=".urlencode($Signature);
foreach ($ReqParaArray as $key => $value)
{
$Req=$Req."&".$key."=".urlencode($value);
}

/*********************************Send the request********************************/
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;
}
```

# Signature Method

Last updated：2018-07-10 17:25:39

Tencent Cloud API authenticates each access request, so each request is required to include the Signature in the common request parameters for user identity authentication. The signature is generated with user's security credentials, which consist of a SecretId and a SecretKey. If you don't have security credentials, apply for the credentials from the Tencent Cloud official website. Otherwise, you will not be able to call the cloud APIs.

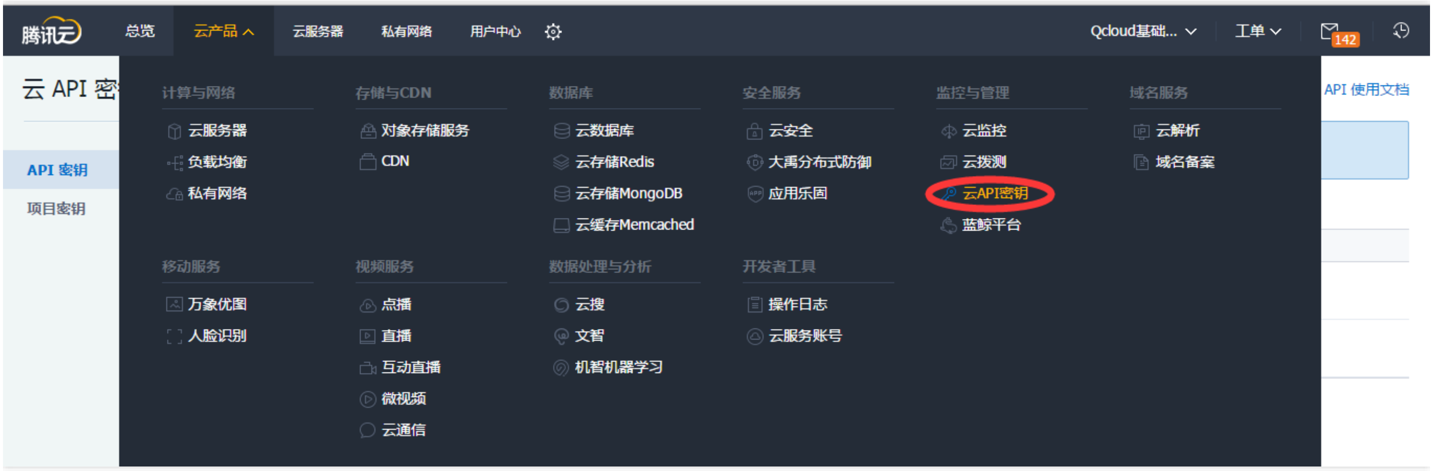
## 1. Apply for Security Credentials

Before calling the Cloud API for the first time, user needs to apply for security credentials on the Tencent Cloud console. Security credentials consists of a SecretId, which is used to identify an API caller, and a SecretKey, which is used to encrypt the signature string and verify its key on the server. Users must keep their SecretKeys private to avoid disclosure.

Apply for security credentials by following the steps below:

(1) Log in to the Tencent Cloud Console.

(2) Click **Cloud Products**, and select **Cloud API Key** under **Monitor and Management** to go to the cloud API key management page.



(3) On the Cloud API Key Management page, click **New** to create a pair of SecretId/SecretKey. Each account can have two pairs of SecretId/SecretKey at most.

## 2. Generate the Signature String

With the SecretId and SecretKey, a signature string can be generated. The following shows how to generate a signature:

Suppose that you have the following SecretId and SecretKey:

```
SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA
SecretKey: Gu5t9xGARNpq86cd98joQYCN3Cozk1qA
```

**Note: This information is only for demonstration purpose. Make sure you proceed with your actual SecretId and SecretKey.**

Take View Instance List (DescribeInstances) as an example. The possible request parameters are as follows when this API is called:

| Parameter Name | Description | Parameter Value |
|---|---|---|
| Action | Method name | DescribeInstances |
| SecretId | Key ID | AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA |
| Timestamp | Current timestamp | 1465185768 |
| Nonce | A random positive integer | 11886 |
| Region | The region where the instance resides | gz |
| instanceIds.0 | ID of the instance to be queried | ins-09dx96dg |
| offset | Offset | 0 |
| limit | Maximum number of output results | 20 |

According to the above table, among the request parameters, there are only 5 common request parameters (Action, SecretId, Timestamp, Nonce and Region), instead of 6 ones as described in "Common Request Parameters". Actually, Signature (the sixth one) is generated from other parameters (including the instruction request parameters) using the following procedure:

## 2.1. Sort parameters

First, sort all the request parameters in an ascending lexicographical order by their names, just like sorting words in a dictionary in ascending alphabetical order or numerical order. That is to say, sort the parameters by their first letters, then by their second letters if their first letters are the same, and so on. You can complete the sorting process using relevant sorting functions in programming language, such as the ksort function in PHP. The parameters in the example are sorted as follows:

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

Any other programming language can be used to sort these parameters as long as the same result is produced.

## 2.2. Construct the request string

This step is used to generate the request string.
Format the request parameters sorted in the previous step as "parameter name"="parameter value". For example, if the parameter value of "Action" is "DescribeInstances", the resulting format is Action=DescribeInstances.
**Note: 1. "Parameter value" is the original value, instead of the URL encoded value. 2. Any underscore in the Key of input parameter needs to be replaced with ".".**

Then, join the formatted parameters together with "&" to generate the final request string:

```
Action=DescribeInstances&Nonce=11886&Region=gz&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3gn
PhESA&Timestamp=1465185768&instanceIds.0=ins-09dx96dg&limit=20&offset=0
```

## 2.3. Construct the original signature string

This step is used to generate the original signature string.
The original signature string is composed of the following parameters:

(1) Request method: The POST and GET methods are supported. In this case, a GET request is used. The methods must be in upper-case.
(2) Request CVM: The request domain name for View Instance List (DescribeInstances) is cvm.api.qcloud.com. The actual request domain name varies with the module to which the API belongs. For more information, please see the relevant API description.
(3) Request path: The request path of cloud API is always /v2/index.php.
(4) Request string: The request string generated in the previous step.

The original signature string is constructed as follows:

```
Request Method + Request CVM + Request Path + ? + Request String
```

The resulting string is:

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

### 2.4. Generate the signature string

This step is to generate the signature string.
Sign the **original signature string** obtained in the previous step using HMAC-SHA1 algorithm, and then encode the signature string using Base64 to obtain the final signature string.

For example, the code is as follows if written in PHP:

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3Cozk1qA';
$srcStr = 'GETcvm.api.qcloud.com/v2/index.php?Action=DescribeInstances&Nonce=11886&Region=gz&
SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA&Timestamp=1465185768&instanceIds.0=ins-09d
x96dg&limit=20&offset=0';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

The resulting signature string is as follows:

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

If another programming language is used, the original signature string in the above example can be used for verification, as long as the signature string generated is the same as the one in the example.

## 3. Encode the Signature String

The generated signature string cannot be directly used as the request parameter, and needs to be URL encoded.
**Note: If the GET method is used, all request parameters need to be encoded with URL encoding.**
For example, the signature string generated in the previous step NSI3UqqD99b/UJb4tbG/xZpRW64= is converted to NSI3UqqD99b/UJb4tbG/xZpRW64= after being encoded. The resulting request parameter for signature string (Signature) is NSI3UqqD99b/UJb4tbG/xZpRW64=, which will be used to generate the final request URL.

# Server API

## API Overview

Last updated：2018-07-10 17:08:00

| Feature | Action ID |
|---|---|
| Creates an LVB channel | CreateLVBChannel |
| Queries LVB channel list | DescribeLVBChannelList |
| Queries LVB channel details | DescribeLVBChannel |
| Modifies an LVB channel | ModifyLVBChannel |
| Starts LVB channels in batch | StartLVBChannel |
| Stops LVB channels in batch | StopLVBChannel |
| Deletes an LVB channel | DeleteLVBChannel |
| Creates a recording task | CreateRecord |
| Terminates a recording task | StopRecord |
| Queries recorded files (recommended) | GetVodRecordFiles |
| Queries the list of recorded fragments | DescribeRecord |
| Creates a screenshot task | CreateLVBShot |
| Terminates a screenshot task | StopLVBShot |
| Queries queue messages | DescribeQueueLog |
| Deletes a recording task | DeleteRecord |
| Deletes a screenshot task | DeleteLVBShot |
| Queries the list of recording tasks | DescribeRecordList |
| Queries the list of screenshot tasks | DescribeLVBShotList |

# Create Live Channel

Last updated：2018-07-10 17:11:02

## Creating a Channel

### 1. API description

Domain name: live.api.qcloud.com

### 2. Input parameters

| Parameter Name | Required | Type | Description |
|---|---|---|---|
| channelName | Yes | String | Channel name |
| channelDescribe | No | string | LVB channel description |
| outputSourceType | Yes | int | Select an output source. 1: RTMP output only; 2: HLS output only; 3: Both RTMP and HLS output. |
| playerPassword | No | string | Password for the receiver's player. |
| sourceList | Yes | array | An LVB source list. Each LVB source may contain the type, name, and address (limited to stream pulling). Note: Only one LVB source is supported. |
| watermarkId | No | int | Watermark ID |
| outputRate | No | array | Output bitrate. Note: For the parameter array, 0 indicates the original bitrate, 10 indicates a bitrate of 550 (standard definition), and 20 indicates a bitrate of 900 (high definition). The value 0 is required if you need to set the bitrate. |

The sourceList structure is as follows:

| Parameter Name | Required | Type | Description |
|---|---|---|---|
| name | Yes | string | LVB source name |
| type | Yes | int | 1: RTMP push |

### 3. Output parameters

| Parameter Name | Type | Description |
|---|---|---|
| code | Int | Error code. 0: Successful; other values: Failed |
| message | String | Error message |
| channel_id | String | Channel ID |

## 4. Examples

Input (RTMP push)

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

Output

```
{
"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. Standard parameter definitions

Channel status definitions:

| Value | Status |
|---|---|
| 0 | No input stream |
| 1 | LVB in progress |
| 2 | Exception |
| 3 | Disable |

Receiver type definitions:

| Type | Description |
|---|---|
| 1 | RTMP output |
| 2 | HLS output |
| 3 | RTMP and HLS output |

LVB stream protocol definitions:

| Type | Description |
|---|---|
| 1 | RTMP push |
| 2 | RTMP pull |
| 3 | HLS pull |

# Change Live Channel

Last updated：2018-07-10 17:08:54

## 1. API Description

Domain name: live.api.qcloud.com

This API (ModifyLVBChannel) is used to modify the basic information of LVB channels, including the channel name and description.

## 2. Input Parameters

| Parameter Name | Required | Type | Description |
|---|---|---|---|
| channelId | Yes | Unsigned integer | Channel ID |
| channelName | Yes | String | LVB channel name |
| channelDescribe | No | String | LVB channel description |

## 3. Output Parameters

| Parameter Name | Type | Description |
|---|---|---|
| code | Int | Error code. 0: Successful; other values: Failed |
| message | String | Error message |

## 4. Example

Input 1

```
https://domain/v2/index.php?Action=ModifyLVBChannel&channelId=96171715553394807&channelName=test-15&cha
Describe=第15个测试频道用例&Common request parameters
```

Output 1

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

## 5. Standard Parameter Definitions

Channel status definitions:

| Value | Status |
|---|---|
| 0 | No input stream |
| 1 | LVB in progress |
| 2 | Exception |
| 3 | Disable |

Receiver type definitions:

| Type | Description |
|---|---|
| 1 | RTMP output |
| 2 | HLS output |
| 3 | RTMP and HLS output |

LVB stream protocol definitions:

| Type | Description |
|---|---|
| 1 | RTMP push |
| 2 | RTMP pull |
| 3 | HLS pull |

# Delete Live Channel

Last updated：2018-07-10 17:13:31

## 1. API Description

Domain name: live.api.qcloud.com

This API (DeleteLVBChannel) is used to enter the ID of an LVB channel to delete the channel (channels can be deleted in batch).

## 2. Input Parameters

| Parameter Name | Required | Type | Description |
|---|---|---|---|
| channelIds | Yes | Array | Channel ID value, batch operation supported. See examples for how to use |

## 3. Output Parameters

| Parameter Name | Type | Description |
|---|---|---|
| code | Int | Error code. 0: Successful; other values: Failed |
| message | String | Error message |

## 4. Example

Input 1

```
https://domain/v2/index.php?Action=DeleteLVBChannel&channelIds.1=96171715553394807&Common request param
```

Output 1

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

## 5. Standard Parameter Definitions

Channel status definitions:

| Value | Status |
|---|---|
| 0 | No input stream |
| 1 | LVB in progress |
| 2 | Exception |
| 3 | Disable |

Receiver type definitions:

| Type | Description |
|---|---|
| 1 | RTMP output |
| 2 | HLS output |
| 3 | RTMP and HLS output |

LVB stream protocol definitions:

| Type | Description |
|---|---|
| 1 | RTMP push |
| 2 | RTMP pull |
| 3 | HLS pull |

# Enable Live Channel

Last updated：2018-07-10 17:15:39

## 1. API Description

Domain name: live.api.qcloud.com

This API (StartLVBChannel) is used to enter the ID of an LVB channel to start the channel (channels can be started in batch).

## 2. Input Parameters

| Parameter Name | Required | Type | Description |
|---|---|---|---|
| channelIds | Yes | Array | Channel ID value, batch operation supported. See examples for how to use |

## 3. Output Parameters

| Parameter Name | Type | Description |
|---|---|---|
| code | Int | Error code. 0: Successful; other values: Failed |
| message | String | Error message |

## 4. Example

Input 1

```
https://domain/v2/index.php?Action=StartLVBChannel&channelIds.1=96171715553394807&Common request parame
```

Output 1

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

# 5. Standard Parameter Definitions

Channel status definitions:

| Value | Status |
|---|---|
| 0 | No input stream |
| 1 | LVB in progress |
| 2 | Exception |
| 3 | Disable |

Receiver type definitions:

| Type | Description |
|---|---|
| 1 | RTMP output |
| 2 | HLS output |
| 3 | RTMP and HLS output |

LVB stream protocol definitions:

| Type | Description |
|---|---|
| 1 | RTMP push |
| 2 | RTMP pull |
| 3 | HLS pull |

# Disable Live Channel

Last updated：2018-07-10 17:14:16

## 1. API Description

Domain name: live.api.qcloud.com

This API (StopLVBChannel) is used to enter the ID of an LVB channel to stop the channel (channels can be stopped in batch).

## 2. Input Parameters

| Parameter Name | Required | Type | Description |
|---|---|---|---|
| channelIds | Yes | Array | Channel ID value, batch operation supported. See examples for how to use |

## 3. Output Parameters

| Parameter Name | Type | Description |
|---|---|---|
| code | Int | Error code. 0: Successful; other values: Failed |
| message | String | Error message |

## 4. Example

Input 1

```
https://domain/v2/index.php?Action=StopLVBChannel&channelIds.1=96171715553394810&Common request paramet
```

Output 1

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

## 5. Standard Parameter Definitions

Channel status definitions:

| Value | Status |
|---|---|
| 0 | No input stream |
| 1 | LVB in progress |
| 2 | Exception |
| 3 | Disable |

Receiver type definitions:

| Type | Description |
|---|---|
| 1 | RTMP output |
| 2 | HLS output |
| 3 | RTMP and HLS output |

LVB stream protocol definitions:

| Type | Description |
|---|---|
| 1 | RTMP push |
| 2 | RTMP pull |
| 3 | HLS pull |

# Query Live Channel List

Last updated：2018-07-10 17:18:43

## 1. API Description

Domain name: live.api.qcloud.com

This API (DescribeLVBChannelList) is used to obtain all LVB channel information of a user, including channel ID, current status, name, and creation time.

## 2. Input Parameters

| Parameter Name | Required | Type | Description |
|---|---|---|---|
| channelStatus | No | Unsigned integer | Channel status, used to filter channel list (0: without input stream; 1: with input stream; 2: exception; 3: closed; 4: incomplete configuration) |
| ascDesc | No | Unsigned integer | Order the results are in. By default, the results are sorted by creation time. 0: by time in an ascending order; 1: by time in a descending order. |
| pageNo | No | Unsigned integer | Page number. For example, to view the list in page 3, set the value to 3. |
| pageSize | No | Unsigned integer | The number of channels displayed on each page. |
| orderBy | No | String | By default, the results are sorted by the channel creation time. |

## 3. Output Parameters

| Parameter Name | Type | Description |
|---|---|---|
| code | Int | Error code. 0: Successful; other values: Failed |
| message | String | Error message |
| all_count | int | Total number of channels |

| | | |
|---|---|---|
| channelSet | Array | Channel list |

## 4. Example

Input 1

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

Output 1

```
{
"code": 0,
"message": "",
"all_count": "2",
"channelSet": [
{
"channel_id": "961717155533394811",
"channel_name": "World Table Tennis Championships 3",
"channel_status": "0",
"create_time": "2015-07-23 20:05:52"
},
{
"channel_id": "961717155533394810",
"channel_name": "World Table Tennis Championships 2",
"channel_status": "0",
"create_time": "2015-07-23 19:54:05"
}
]
}
```

## 5. Standard Parameter Definitions

Channel status definitions:

| Value | Status |
|---|---|
| 0 | No input stream |
| 1 | LVB in progress |
| 2 | Exception |

| | |
|---|---|
| 3 | Disable |

Receiver type definitions:

| Type | Description |
|---|---|
| 1 | RTMP output |
| 2 | HLS output |
| 3 | RTMP and HLS output |

LVB stream protocol definitions:

| Type | Description |
|---|---|
| 1 | RTMP push |
| 2 | RTMP pull |
| 3 | HLS pull |

# Query Live Channel Details

Last updated：2018-07-10 17:19:43

## 1. API Description

Domain name: live.api.qcloud.com

This API (DescribeLVBChannel) is used to enter the ID number of a channel to be queried to get its current status, name, description, LVB source, and output source information.

## 2. Input Parameters

| Parameter Name | Required | Type | Description |
|---|---|---|---|
| channelId | Yes | Unsigned integer | Channel ID |

## 3. Output Parameters

| Parameter Name | Type | Description |
|---|---|---|
| code | Int | Error code. 0: Successful; other values: Failed |
| message | String | Error message |
| channelInfo | Array | Channel information, such as its current status, name, description, LVB source, and output source information. |

## 4. Example

Input 1

```
https://domain/v2/index.php?Action=DescribeLVBChannel&channelId=96171715553394807&Common request parame
```

Output 1

```
{
"code": 0,
"message": "",
"channelInfo": [
{
"channel_id": "XXX",
"channel_name": "XXX",
"channel_describe": "XXX",
"channel_status": "1",
"upstream_list": [
{
"sourceName": "RTMP",
"sourceID": "YYYYYYYYYYY",
"sourceType": "1",
"sourceAddress": "rtmp://2000.livepush.myqcloud.com/live/YYYYYYYYYYYYYYYYYY?bizid2000"
}
],
"hls_downstream_address": "",
"rtmp_downstream_address": "rtmp://2000.liveplay.myqcloud.com/live/XXX",
"player_id": "226",
"resolution": null,
"password": null
}
]
}
```

## 5. Standard Parameter Definitions

Channel status definitions:

| Value | Status |
|---|---|
| 0 | No input stream |
| 1 | LVB in progress |
| 2 | Exception |
| 3 | Disable |

Receiver type definitions:

| Type | Description |
|---|---|
| 1 | RTMP output |
| 2 | HLS output |

| | |
|---|---|
| 3 | RTMP and HLS output |

LVB stream protocol definitions:

| Type | Description |
|---|---|
| 1 | RTMP push |
| 2 | RTMP pull |
| 3 | HLS pull |

# Create Recording Task

Last updated：2018-07-18 17:08:48

## 1. API Description

Domain name: live.api.qcloud.com

Note: **This API (CreateRecord) is used to store recorded files on the VOD platform. If you want to use the recording feature, you need to activate VOD Service first**. When recording files are stored, the charges (including charges for storage and downstream playback traffic) are calculated with VOD billing method. For more information, please see relevant document. Note: API calling timeout should be greater than 3 seconds, because retries within 3 seconds and frequent calls may lead to duplicate recording tasks.

Recording rules
(1) The recording task starts at the preset time and ends when the LVB ends
(2) The recording is fragmented by time, and the longest fragment time is 0.5 hour
(3) The recording is stopped and a fragment is generated in case of a stream interruption
A new fragment recording task is started after the stream is resumed to repeat the preceding steps until the entire recording is completed
Fragment recording request URL format: `http://(VOD bizid).vod.myqcloud.com/(vid).f0.flv`
Or view the fragment on the VOD page

## 2. Input Parameters

| Parameter Name | Required | Type | Description |
|---|---|---|---|
| channelId | Yes | String | Channel ID |
| startTime | Yes | String | Start time (China Standard Time), such as 2017-01-01 10:10:01 |
| endTime | Yes | String | End time (China Standard Time), such as 2017-01-01 10:10:01 |
| tapeType | Yes | Int | Whether to enable real-time recording. 1 - Enable; 0 - Disable; 1 is recommended.<br>a. Creating real-time recording tasks requires that VJs push streams actively. The video recording starts at the moment when this API is successfully called. In this case, the parameter of task start time is ignored.<br>b. The real-time recording task supports a maximum length of 30 minutes. If the time difference between the passed task end time |

| | | | |
|---|---|---|---|
| | | | and the current time is longer than 30 minutes, only a length of 30 minutes is recorded. The recommended recording length is within 5 minutes.<br>c. If real-time recording is disabled, the parameter of task start time is required, and the time difference between the end time and the start time should not be longer than 1 day. |

## 3. Output Parameters

| Parameter Name | Type | Description |
|---|---|---|
| code | Int | Error code. 0: Successful; other values: Failed |
| message | String | Error message |
| task_id | Int | Task ID. 64-bit unsigned integer |

## 4. Example

Input 1

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

Output 1

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

}
```

# End Recording Task

Last updated：2018-07-10 17:20:33

## 1. API Description

Domain name: live.api.qcloud.com

Note: This API (StopRecord) is used to store recorded files on the VOD platform. If you want to use the recording feature, you need to activate a VOD account first and ensure that the account is available. When recording files are stored, the charges (including charges for storage and downstream playback traffic) are calculated with VOD billing method. For more information, please see relevant document.

## 2. Input Parameters

| Parameter Name | Required | Type | Description |
|---|---|---|---|
| channelId | Yes | String | Channel ID |
| taskId | Yes | Int | Task ID |

## 3. Output Parameters

| Parameter Name | Type | Description |
|---|---|---|
| code | Int | Error code. 0: Successful; other values: Failed |
| message | String | Error message |

## 4. Example

Input 1

```
http://domain/v2/index.php?Action=StopRecord&channelId=123&taskId=1&Common request parameters
```

Output 1

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

# Delete Recording Task

Last updated：2018-07-10 17:11:48

## 1. API Description

Domain name: live.api.qcloud.com

## 2. Input Parameters

| Parameter Name | Required | Type | Description |
|---|---|---|---|
| channelId | Yes | String | Channel ID |
| taskId | Yes | Int | Task ID |

## 3. Output Parameters

| Parameter Name | Type | Description |
|---|---|---|
| code | Int | Error code. 0: Successful; other values: Failed |
| message | String | Error message |

## 4. Example

Input 1

```
http://domain/v2/index.php?Action=DeleteRecord&channelId=123&taskId=1&Common request parameters
```

Output 1

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

# Query Recording Task List

Last updated：2018-07-10 17:16:54

## 1. API Description

Domain name: live.api.qcloud.com

Note: **This API (DescribeRecordList) is used to query files recorded in the last two months.**

## 2. Input Parameters

| Parameter Name | Required | Type | Description |
|---|---|---|---|
| channelId | Yes | String | Channel ID |
| pageNo | No | Int | Page number. Default is 1 |
| pageSize | No | Int | Page size. Default is 10 |

## 3. Output Parameters

| Parameter Name | Type | Description |
|---|---|---|
| code | Int | Error code. 0: Successful; other values: Failed |
| message | String | Error message |
| totalCount | Int | Total number |
| taskSet | Array | Task result set |

The file information is composed as follows:

| Parameter Name | Type | Description |
|---|---|---|
| id | Int | Task ID |
| startTime | String | Time when the fragmentation process starts |
| endTime | String | Time when the fragmentation process ends |

| | | |
|---|---|---|
| status | Int | Task status. 0 - Not started, 1 - Starting, 2 - Completed, 3 - Exception |
| count | Int | Number of generated fragments |

## 4. Example

Input 1

```
http://domain/v2/index.php?Action=DescribeRecordList&channelId=16093104850681751595&pageSize=10&pageNo=
```

Output 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"
}
]

}
```

# Query Recording File (Recommended)

Last updated：2018-07-10 17:17:34

## 1. API Description

Domain name: live.api.qcloud.com

Notes:

1. This API (GetVodRecordFiles) is used to query the result of the recorded file of channel, including file name, task ID, file ID, and recording message.
2. **You can query files recorded in the last two months.**

## 2. Input Parameters

| Parameter Name | Required | Type | Description |
|---|---|---|---|
| channelId | Yes | String | Channel ID |
| startTime | Yes | String | Enter the value encoded from YYYY-MM-DD HH:MM:SS For example: 2016-01-01 00:00:00 (encoded from 2016-01-01 00:00:00) |
| endTime | No | String | Enter YYYY-MM-DD HH:MM:SS (encoded value). There is no limit on the end time by default |
| pageNum | No | Int | Page number. Default is 1 |
| pageSize | No | Int | Number of records in a page. Default is 20 |

## 3. Output Parameters

| Parameter Name | Type | Description |
|---|---|---|
| code | Int | Error code. 0: Successful; other values: Failed |
| message | String | Error message |
| total_count | int | Total number of returned files |

| | | |
|---|---|---|
| filesInfo | Array | [ { "fileId": "9896125784085567721", "fileName": "LVB -aaa-20160112-1455-20160112-1457", "reportMessage": null, "taskId": "57", "startTime": "7", "endTime": "128" }] |

## 4. Example

Input 1

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

Output 1

```
{
"code": 0,
"message": "",
"codeDesc": "Success",
"filesInfo": [
{
"fileId": "9896125784085567721",
"fileName": "LVB -aaa-20160112-1455-20160112-1457",
"reportMessage": null,
"taskId": "57",
"startTime": "7",
"endTime": "128"
},
{
"fileId": "9896125784085567722",
"fileName": "LVB -aaa-20160112-1457-20160112-1459",
"reportMessage": null,
"taskId": "57",
"startTime": "128",
"endTime": "250"
},
{
"fileId": "9896125784085567723",
"fileName": "LVB -aaa-20160112-1459-20160112-1459",
"reportMessage": null,
"taskId": "57",
"startTime": "250",
"endTime": "299"
},
{
"fileId": "9896125784085567727",
"fileName": "LVB -aaa-20160112-1510-20160112-1512",
"reportMessage": null,
"taskId": "58",
"startTime": "7",
"endTime": "132"
},
{
"fileId": "9896125784085567736",
"fileName": "LVB -aaa-20160112-1512-20160112-1514",
"reportMessage": null,
"taskId": "58",
"startTime": "132",
"endTime": "258"
},
{
"fileId": "9896125784085567739",
"fileName": "LVB -aaa-20160112-1514-20160112-1515",
"reportMessage": null,
"taskId": "58",
"startTime": "258",
"endTime": "301"
},
{
"fileId": "9896125784085567756",
"fileName": "LVB -aaa-20160112-1525-20160112-1526",
"reportMessage": null,
"taskId": "61",
"startTime": "7",
"endTime": "60"
}
],
"totalCount": 7
}
```

# Query Recorded Shard List

Last updated：2018-07-10 17:16:28

## 1. API Description

Domain name: live.api.qcloud.com

Note: This API (DescribeRecord) is used to store recorded files on the VOD platform. If you want to use the recording feature, you need to activate a VOD account first and ensure that the account is available. When recording files are stored, the charges (including charges for storage and downstream playback traffic) are calculated with VOD billing method. For more information, please see relevant document.

## 2. Input Parameters

| Parameter Name | Required | Type | Description |
|---|---|---|---|
| channelId | Yes | String | Channel ID |
| taskId | Yes | Int | Task ID |
| pageNo | No | Int | Page number. Default is 1 |
| pageSize | No | Int | Page size. Default is 10 |

## 3. Output Parameters

| Parameter Name | Type | Description |
|---|---|---|
| code | Int | Error code. 0: Successful; other values: Failed |
| message | String | Error message |
| totalCount | Int | Total number |
| fileSet | Array | File result set |

The file information is composed as follows:

| Parameter Name | Type | Description |
|---|---|---|

| | | |
|---|---|---|
| fileId | String | File ID |
| startTime | String | Time when the fragmentation process starts |
| endTime | String | Time when the fragmentation process ends |
| fileName | String | File name |

## 4. Example

Input 1

```
http://domain/v2/index.php?Action=DescribeRecord&channelId=16093104850681751595&taskId=10&pageSize=10&p
```

Output 1

```
{

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

}
```

After the recording, the VOD platform will generate a file, for example:

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

Complete information on access methods can be found via field and API VOD API

# Error Code List

Last updated：2018-07-10 17:27:24

## Categorization of Error Codes

The error codes listed here can be categorized into **Access Layer Error Codes** and **Video Cloud Error Codes**. The access layer error codes are uniformly defined across the Tencent Cloud access layer, and the video cloud error codes are only needed in LVB and VOD services.

For more information on how to distinguish between the two types of error codes and how they work, please see Dual Error Codes.

## Access Layer Error Codes

| Error Code | Message | Description |
|---|---|---|
| 4000 | Invalid request parameter | Required parameters are missing, or the format of parameter values is incorrect. For error message, please see the "message" field in error description. |
| 4100 | Authentication failed | Signature authentication failed. For more information, please see the "Authentication" section in the document. |
| 4200 | Request expired | The request has expired. For more information, please see the "Request Validity" section in the document. |
| 4300 | Access denied | The account is blocked or not within the user range of the API. |
| 4400 | Quota exceeded | The number of requests exceeded the quota limit. For more information, please see the "Request Quota" section in the document. |
| 4500 | Replay attack | The Nonce and Timestamp parameters ensure that each request is executed only once on the server. Therefore, the Nonce value cannot be the same as the last one, and the difference between Timestamp and Tencent server time cannot be greater than 2 hours. |
| 4600 | Unsupported protocol | The protocol is not supported. For more information, please see the relevant document. |
| 5000 | Resource does not exist | The instance corresponding to the resource ID does not exist, or the instance has been returned, or another user's resource is accessed. |

| Error Code | Message | Description |
|---|---|---|
| 5100 | Resource operation failed | A failure message is returned from the function layer. For error message, please see the "message" field in error description. |
| 5200 | Failed to purchase the resource | The resource purchase failed. This may be caused unsupported instance configuration or insufficient resources. |
| 5300 | Failed to purchase the resource | The resource purchase failed because of insufficient balance. |
| 5400 | Part of operations performed successfully | The batch operation was successful on some resources. For more information, please see the returned value of the method. |
| 5500 | User failed to pass identity verification | The resource purchase failed because the user failed to pass identity verification. |
| 6000 | Internal error with the server | An internal error occurred with the server. Try again later or contact customer service. |
| 6100 | Not supported in the version | This API is not supported in this version or is under maintenance. Note: When this error occurs, check whether the domain name for the API is correct. Domain name may vary with different modules. |
| 6200 | API is unavailable | The API is under maintenance and is unavailable. Try again later. |

## Video Cloud Error Codes

| Error Code | Message |
|---|---|
| 1000 | Incorrect input parameter |
| 1001 | An error occurred while obtaining the user account |
| 1002 | The user does not exist |
| 1003 | Incorrect user account ID |

| Error Code | Message |
|---|---|
| 3000 | Network error |
| 4001 | Incorrect JSON format |
| 20100 | An error occurred while creating a channel |
| 20101 | Number of channels exceeds the limit |
| 20102 | An error occurred while obtaining the channel information |
| 20103 | An error occurred while modifying the channel information |
| 20104 | An error occurred while obtaining the channel list |
| 20105 | An error occurred while deleting a channel |
| 20106 | An error occurred while disabling a channel |
| 20107 | An error occurred while enabling a channel |
| 20108 | Channel is not in an editable status |
| 20109 | Channel is not in disabled status |
| 20111 | No LVB source is found for the channel |
| 20112 | Incorrect LVB source protocol |
| 20113 | Incorrect receiver protocol |
| 20114 | Failed to activate the service |
| 20200 | Pull URL is empty |
| 20201 | An error occurred while obtaining the LVB source information |
| 20202 | An error occurred while creating a LVB source |
| 20203 | An error occurred while deleting a LVB source |
| 20204 | Pull URL is empty |
| 20250 | Blacklist and whitelist are empty |
| 20251 | Blacklist and whitelist do not exist |
| 20252 | An error occurred while obtaining the blacklist and whitelist |
| 20253 | Number of name lists exceeds the limit |

| Error Code | Message |
|---|---|
| 20300 | An error occurred while modifying channel status |
| 20301 | Channel does not exist |
| 20320 | Failed to create recording task |
| 20321 | Failed to obtain the list of recording tasks |
| 20322 | Time conflict between recording tasks |
| 20323 | The start time and end time of the recording task are invalid |
| 20326 | Failed to modify recording task |
| 20327 | The recording task does not exist |
| 20328 | Screencap task does not exist |
| 20500 | Watermark does not exist |