Cloud Message Queue

API Documentation

Product Documentation

Tencent Cloud

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

API Documentation History

Last updated：2024-11-27 16:09:13

Existing Release

Release time: 2024-11-27 16:08:40

Existing APIs/data structures are as follows:

Improvement to existing documentation.

Existing APIs:

- ClearQueue
- ClearSubscriptionFilterTags
- CreateQueue
- CreateSubscribe
- CreateTopic
- DeleteQueue
- DeleteSubscribe
- DeleteTopic
- DescribeDeadLetterSourceQueues
- DescribeQueueDetail
- DescribeSubscriptionDetail
- DescribeTopicDetail
- ModifyQueueAttribute
- ModifySubscriptionAttribute
- ModifyTopicAttribute
- RewindQueue
- UnbindDeadLetter

Existing data structures:

- DeadLetterPolicy
- DeadLetterSource
- Filter
- QueueSet

- Subscription
- Tag
- TopicSet
- TransactionPolicy

API Category

Last updated：2024-11-27 16:09:05

Subscription APIs

API Name	Feature	Frequency Limit (maximum requests per second)
ClearSubscriptionFilterTags	Clears the message tags of subscriber	20
CreateSubscribe	Creates subscription	20
DeleteSubscribe	Deletes subscription	20
DescribeSubscriptionDetail	Queries subscription details	20
ModifySubscriptionAttribute	Modifies subscription attributes	20

Topic APIs

API Name	Feature	Frequency Limit (maximum requests per second)
CreateTopic	Creates topic	20
DeleteTopic	Deletes topic	20
DescribeTopicDetail	Queries topic details	20
ModifyTopicAttribute	Modifies topic attributes	20

Queue APIs

API Name	Feature	Frequency Limit (maximum requests per second)
ClearQueue	Clears all messages in queue	20
CreateQueue	Creates queue	20

DeleteQueue	Deletes queue	20
DescribeDeadLetterSourceQueues	Enumerates the source queues of dead letter queue	20
DescribeQueueDetail	Enumerates queues	20
ModifyQueueAttribute	Modifies queue attributes	20
RewindQueue	Rewinds queue	20
UnbindDeadLetter	Unbinds dead letter queue	20

Making API Requests

Request Structure

Last updated：2024-11-27 16:09:05

1. Service Address

The API supports access from either a nearby region (at cmq.intl.tencentcloudapi.com) or a specified region (at cmq.ap-guangzhou.tencentcloudapi.com for Guangzhou, for example).

We recommend using the domain name to access the nearest server. When you call an API, the request is automatically resolved to a server in the region **nearest** to the location where the API is initiated. For example, when you initiate an API request in Guangzhou, this domain name is automatically resolved to a Guangzhou server, the result is the same as that of specifying the region in the domain like "cmq.ap-guangzhou.tencentcloudapi.com".

Note: For latency-sensitive businesses, we recommend that you specify the region in the domain name.

Tencent Cloud currently supports the following regions:

Hosted region	Domain name
Local access region (recommended, only for non-financial availability zones)	cmq.intl.tencentcloudapi.com
South China (Guangzhou)	cmq.ap-guangzhou.tencentcloudapi.com
East China (Shanghai)	cmq.ap-shanghai.tencentcloudapi.com
North China (Beijing)	cmq.ap-beijing.tencentcloudapi.com
Southwest China (Chengdu)	cmq.ap-chengdu.tencentcloudapi.com
Southwest China (Chongqing)	cmq.ap-chongqing.tencentcloudapi.com
Hong Kong, Macao, Taiwan (Hong Kong, China)	cmq.ap-hongkong.tencentcloudapi.com
Southeast Asia (Singapore)	cmq.ap-singapore.tencentcloudapi.com
Southeast Asia (Bangkok)	cmq.ap-bangkok.tencentcloudapi.com

South Asia (Mumbai)	cmq.ap-mumbai.tencentcloudapi.com
Northeast Asia (Seoul)	cmq.ap-seoul.tencentcloudapi.com
Northeast Asia (Tokyo)	cmq.ap-tokyo.tencentcloudapi.com
U.S. East Coast (Virginia)	cmq.na-ashburn.tencentcloudapi.com
U.S. West Coast (Silicon Valley)	cmq.na-siliconvalley.tencentcloudapi.com
Europe (Frankfurt)	cmq.eu-frankfurt.tencentcloudapi.com

2. Communications Protocol

All the Tencent Cloud APIs communicate via HTTPS, providing highly secure communication tunnels.

3. Request Methods

Supported HTTP request methods:

- POST (recommended)
- GET

The Content-Type types supported by POST requests:

- application/json (recommended). The TC3-HMAC-SHA256 signature algorithm must be used.
- application/x-www-form-urlencoded. The HmacSHA1 or HmacSHA256 signature algorithm must be used.
- multipart/form-data (only supported by certain APIs). You must use TC3-HMAC-SHA256 to calculate the signature.

The size of a GET request packet is up to 32 KB. The size of a POST request is up to 1 MB when the HmacSHA1 or HmacSHA256 signature algorithm is used, and up to 10 MB when TC3-HMAC-SHA256 is used.

4. Character Encoding

Only UTF-8 encoding is used.

Common Params

Last updated：2024-11-27 16:09:07

Common parameters are used for all APIs authenticating requestors. Common parameters must be included in all API requests, and they will not be described in individual API documents.

The exact contents of the common parameters will vary depending on the version of the signature method you use.

Common parameters for Signature Algorithm v3

When the TC3-HMAC-SHA256 algorithm is used, the common parameters should be uniformly placed in the HTTP request header, as shown below:

Parameter Name	Type	Required	Description
X-TC-Action	String	Yes	The name of the API for the desired operation. For the specific value, see description of common parameter `Action` in the input parameters in r documentation. For example, the API for querying the CVM instance list i `DescribeInstances` .
X-TC-Region	String	Yes	Region parameter, which is used to identify the region to which the data y work with belongs. For values supported for an API, see the description c parameter `Region` in the input parameters in related API documentati parameter is not required for some APIs (which will be indicated in relate documentation), and will not take effect even it is passed.
X-TC-Timestamp	Integer	Yes	The current UNIX timestamp that records the time when the API request for example, 1529223702. Note: If the difference between the UNIX times server time is greater than 5 minutes, a signature expiration error may occ
X-TC-Version	String	Yes	API version of the action. For the valid values, see the description of the c parameter `Version` in the API documentation. For example, the versi 2017-03-12.
Authorization	String	Yes	The HTTP authentication request header, for example: TC3-HMAC-SHA256 Credential=AKID*******/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc96317 Here: - TC3-HMAC-SHA256: Signature method, currently fixed as this value; - Credential: Signature credential; AKID******* is the SecretId; Date is a d time, and this value must match the value of X-TC-Timestamp (a commo

			UTC time format; service is the name of the product/service, and is gener name prefix. For example, a domain name cvm.tencentcloudapi.com refe product and the value would be cvm; - SignedHeaders: The headers that contains the authentication informatic type and host are the required headers; - Signature: Signature digest.
X-TC-Token	String	No	The token used for a temporary certificate. It must be used with a tempora can obtain the temporary key and token by calling a CAM API. No token i a long-term key.

Assuming you want to query the list of Cloud Virtual Machine instances in the Guangzhou region, the request structure in the form of request URL, request header and request body may be as follows:

Example of an HTTP GET request structure:

```
https://cvm.tencentcloudapi.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKID********************************/20
18-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993
f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: ap-guangzhou
```

The following example shows you how to structure an HTTP POST (application/json) request:

```
https://cvm.tencentcloudapi.com/

Authorization: TC3-HMAC-SHA256 Credential=AKID********************************/20
18-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5
924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: application/json
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou

{"Offset":0,"Limit":10}
```

Example of an HTTP POST (multipart/form-data) request structure (only supported by specific APIs):

```
https://cvm.tencentcloudapi.com/

Authorization: TC3-HMAC-SHA256 Credential=AKID********************************/20
18-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5
924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: multipart/form-data; boundary=58731222010402
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou

--58731222010402
Content-Disposition: form-data; name="Offset"

0
--58731222010402
Content-Disposition: form-data; name="Limit"

10
--58731222010402--
```

Common parameters for Signature Algorithm v1

To adopt the HmacSHA1 and HmacSHA256 signature methods, common parameters must be put into the request string, as shown below:

Parameter Name	Type	Required	Description
Action	String	Yes	The name of the API for the desired operation. For the specific value, see the description of common parameter `Action` in the input parameters in related API documentation. For example, the API for querying the CVM instance list is `DescribeInstances`.
Region	String	Yes	Region parameter, which is used to identify the region to which the data you want to work with belongs. For values supported for an API, see the description of common parameter `Region` in the input parameters in related API documentation. Note: This parameter is not required for some APIs (which will be indicated in related API documentation), and will not take effect even if it is passed.

Timestamp	Integer	Yes	The current UNIX timestamp that records the time when the API request was initiated, for example, 1529223702. If the difference between the value and the current system time is too large, a signature expiration error may occur.
Nonce	Integer	Yes	A random positive integer used along with `Timestamp` to prevent replay attacks.
SecretId	String	Yes	The identifying SecretId obtained on the Cloud API Key page. A SecretId corresponds to a unique SecretKey which is used to generate the request signature (Signature).
Signature	String	Yes	Request signature used to verify the validity of this request. This is calculated based on the actual input parameters. For more information about how this is calculated, see the API authentication documentation.
Version	String	Yes	API version of the action. For the valid values, see the description of the common input parameter `Version` in the API documentation. For example, the version of CVM is 2017-03-12.
SignatureMethod	String	No	Signature method. Currently, only HmacSHA256 and HmacSHA1 are supported. The HmacSHA256 algorithm is used to verify the signature only when this parameter is specified as HmacSHA256. In other cases, the signature is verified with HmacSHA1.
Token	String	No	The token used for a temporary certificate. It must be used with a temporary key. You can obtain the temporary key and token by calling a CAM API. No token is required for a long-term key.

Assuming you want to query the list of Cloud Virtual Machine instances in the Guangzhou region, the request structure in the form of request URL, request header and request body may be as follows:

Example of an HTTP GET request structure:

```
https://cvm.tencentcloudapi.com/?Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKID********************************

Host: cvm.tencentcloudapi.com
Content-Type: application/x-www-form-urlencoded
```

Example of an HTTP POST request structure:

```
https://cvm.tencentcloudapi.com/

Host: cvm.tencentcloudapi.com
Content-Type: application/x-www-form-urlencoded

Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=
1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d18
1d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKID****************************
****
```

Region List

The supported Region field values for all APIs in this product are listed as below. For any API that does not support any of the following regions, this field will be described additionally in the relevant API document.

Region	Value
North China (Beijing)	ap-beijing
Southwest China (Chongqing)	ap-chongqing
South China (Guangzhou)	ap-guangzhou
East China (Shanghai)	ap-shanghai

Signature v3

Last updated：2024-11-27 16:09:09

TencentCloud API authenticates every single request, i.e., the request must be signed using the security credentials in the designated steps. Each request has to contain the signature information (Signature) in the common request parameters and be sent in the specified way and format.

Applying for Security Credentials

The security credential used in this document is a key, which includes a SecretId and a SecretKey. Each user can have up to two pairs of keys.

- SecretId: Used to identify the API caller, which is just like a username.
- SecretKey: Used to authenticate the API caller, which is just like a password.
- **You must keep your security credentials private and avoid disclosure; otherwise, your assets may be compromised. If they are disclosed, please disable them as soon as possible.**

You can apply for the security credentials through the following steps:

1. Log in to the Tencent Cloud Console.
2. Go to the TencentCloud API Key console page.
3. On the TencentCloud API Key page, click **Create** to create a SecretId/SecretKey pair.

Using the Resources for Developers

TencentCloud API comes with SDKs for seven commonly used programming languages, including Python, Java, PHP, Go, NodeJS and .NET. In addition, it provides API Explorer which enables online call, signature verification, and SDK code generation. If you have any troubles calculating a signature, consult these resources.

TC3-HMAC-SHA256 Signature Algorithm

Compatible with the previous HmacSHA1 and HmacSHA256 signature algorithms, the TC3-HMAC-SHA256 signature algorithm is more secure and supports larger requests and JSON format with better performance. We recommend using TC3-HMAC-SHA256 to calculate the signature.

TencentCloud API supports both GET and POST requests. For the GET method, only the Content-Type: application/x-www-form-urlencoded protocol format is supported. For the POST method, two protocol formats, Content-Type: application/json and Content-Type: multipart/form-data, are supported. The JSON format is supported by default for all business APIs, and the multipart format is supported only for specific business APIs. In this case, the API cannot be called in JSON format. See the specific business API documentation for more information. The POST method is recommended, as there is no difference in the results of both the methods, but the GET method only supports request packets up to 32 KB.

The following uses querying the list of CVM instances in the Guangzhou region as an example to describe the steps of signature splicing. We chose this API because:

1. CVM is activated by default, and this API is often used;
2. It is read-only and does not change the status of existing resources;
3. It covers many types of parameters, which allows it to be used to demonstrate how to use arrays containing data structures.

In the example, we try to choose common parameters and API parameters that are prone to mistakes. When you actually call an API, please use parameters based on the actual conditions. The parameters vary by API. Do not copy the parameters and values in this example.

Assuming that your SecretId and SecretKey are `AKID********************************` and `********************************` , respectively, if you want to view the status of the instance in the Guangzhou region whose CVM instance name is "unnamed" and have only one data entry returned, then the request may be:

```
curl -X POST https://cvm.tencentcloudapi.com \
-H "Authorization: TC3-HMAC-SHA256 Credential=AKID*******************************
*/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=ca282b0a
56549857d53b2beb08b0c35871c892d42d09ae30b38d456e09ce291f" \
-H "Content-Type: application/json; charset=utf-8" \
-H "Host: cvm.tencentcloudapi.com" \
-H "X-TC-Action: DescribeInstances" \
-H "X-TC-Timestamp: 1551113065" \
-H "X-TC-Version: 2017-03-12" \
-H "X-TC-Region: ap-guangzhou" \
-d '{"Limit": 1, "Filters": [{"Values": ["unnamed"], "Name": "instance-name"}]}'
```

The signature calculation process is explained in detail below.

1. Concatenating the CanonicalRequest String

Concatenate the canonical request string (CanonicalRequest) in the following pseudocode format:

```
CanonicalRequest =
HTTPRequestMethod + '\n' +
CanonicalURI + '\n' +
CanonicalQueryString + '\n' +
CanonicalHeaders + '\n' +
SignedHeaders + '\n' +
HashedRequestPayload
```

Field Name	Explanation
HTTPRequestMethod	HTTP request method (GET or POST). This example uses `POST` .
CanonicalURI	URI parameter. Slash ("/") is used for API 3.0.
CanonicalQueryString	The query string in the URL of the originating HTTP request. This is always an empty st “” for POST requests, and is the string after the question mark (?) for GET requests. Fo example: Limit=10&Offset=0。 Note: `CanonicalQueryString` must be URL-encoded, referencing RFC3986, the UTF8 character set. We recommend using the programming language library. All speci characters must be encoded and capitalized.
CanonicalHeaders	Header information for signature calculation, including at least two headers of `host` `content-type` . Custom headers can be added to participate in the signature proce improve the uniqueness and security of the request. Concatenation rules: 1. Both the key and value of the header should be converted to lowercase with the lead and trailing spaces removed, so they are concatenated in the format of key:value\n format; 2. If there are multiple headers, they should be sorted in ASCII ascending order by the header keys (lowercase). The calculation result in this example is `content-type:application/json; charset=utf-8\nhost:cvm.tencentcloudapi.com\n` . Note: `content-type` must match the actually sent content. In some programming languages, a charset value would be added even if it is not specified. In this case, the request sent is different from the one signed, and the sever will return an error indicating signature verification failed.
SignedHeaders	Header information for signature calculation, indicating which headers of the request participate in the signature process (they must each individually correspond to the head in CanonicalHeaders). `Content-type` and `host` are required headers. Concatenation rules: 1. Both the key and value of the header should be converted to lowercase; 2. If there are multiple headers, they should be sorted in ASCII ascending order by the header keys (lowercase) and separated by semicolons (;). The value in this example is `content-type;host`
HashedRequestPayload	Hash value of the request payload (i.e., the body, such as `{"Limit": 1, "Filter`

	`[{"Values": ["unnamed"], "Name": "instance-name"}]}` in this exampl The pseudocode for calculation is Lowercase(HexEncode(Hash.SHA256(RequestPayload))) by SHA256 hashing the pay of the HTTP request, performing hexadecimal encoding, and finally converting the enco string to lowercase letters. For GET requests, `RequestPayload` is always an empt string. The calculation result in this example is `99d58dfbc6745f6747f36bfca17dee5e6881dc0428a0a36f96199342bc5b4`

According to the rules above, the `CanonicalRequest` string obtained in the example is as follows:

```
POST
/

content-type:application/json; charset=utf-8
host:cvm.tencentcloudapi.com

content-type;host
99d58dfbc6745f6747f36bfca17dee5e6881dc0428a0a36f96199342bc5b4907
```

2. Concatenating the String to Be Signed

The string to sign is concatenated as follows:

```
StringToSign =
Algorithm + \n +
RequestTimestamp + \n +
CredentialScope + \n +
HashedCanonicalRequest
```

Field Name	Explanation
Algorithm	Signature algorithm, which is currently always `TC3-HMAC-SHA256` .
RequestTimestamp	Request timestamp, i.e., the value of the common parameter `X-TC-Timestamp` ir request header, which is the UNIX timestamp of the current time in seconds, such as `1551113065` in this example.
CredentialScope	Scope of the credential in the format of `Date/service/tc3_request` , includin date, requested service and termination string (tc3_request). **`Date` is a date in U time, whose value should match the UTC date converted by the common parameter `X-TC-Timestamp` ;** `service` is the product name, which should m the domain name of the product called. The calculation result in this example is `201 25/cvm/tc3_request` .

HashedCanonicalRequest	Hash value of the CanonicalRequest string concatenated in the steps above. The pseudocode for calculation is Lowercase(HexEncode(Hash.SHA256(CanonicalRequ The calculation result in this example is `2815843035062fffda5fd6f2a44ea8a34818b0dc46f024b8b3786976a3ad`

Note:

1. Date has to be calculated from the timestamp "X-TC-Timestamp" and the time zone is UTC+0. If you add the system's local time zone information (such as UTC+8), calls can succeed both day and night but will definitely fail at 00:00. For example, if the timestamp is 1551113065 and the time in UTC+8 is 2019-02-26 00:44:25, the UTC+0 date in the calculated Date value should be 2019-02-25 instead of 2019-02-26.
2. Timestamp must be the same as your current system time, and your system time and standard time must be synced; if the difference between Timestamp and your current system time is larger than five minutes, the request will fail. If your system time is out of sync with the standard time for a while, the request will fail and return a signature expiration error.

According to the preceding rules, the string to be signed obtained in the example is as follows:

```
TC3-HMAC-SHA256
1551113065
2019-02-25/cvm/tc3_request
2815843035062fffda5fd6f2a44ea8a34818b0dc46f024b8b3786976a3adda7a
```

3. Calculating the Signature

1. Calculate the derived signature key with the following pseudocode:

```
SecretKey = "********************************"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

Field Name	Explanation
SecretKey	The original SecretKey, i.e., `********************************` .
Date	The Date field information in `Credential` , such as `2019-02-25` in this example.

Service	Value in the Service field in `Credential` , such as `cvm` in this example.

2. Calculate the signature with the following pseudocode:

```
Signature = HexEncode(HMAC_SHA256(SecretSigning, StringToSign))
```

4. Concatenating the Authorization

The Authorization is concatenated as follows:

```
Authorization =
Algorithm + ' ' +
'Credential=' + SecretId + '/' + CredentialScope + ', ' +
'SignedHeaders=' + SignedHeaders + ', ' +
'Signature=' + Signature
```

Field Name	Explanation
Algorithm	Signature algorithm, which is always `TC3-HMAC-SHA256` .
SecretId	The SecretId in the key pair, i.e., `AKID********************************` .
CredentialScope	Credential scope (see above). The calculation result in this example is `2019-02-25/cvm/tc3_request` .
SignedHeaders	Header information for signature calculation (see above), such as `content-type;host` in this example.
Signature	Signature value. The calculation result in this example is `ca282b0a56549857d53b2beb08b0c35871c892d42d09ae30b38d456e09ce291f` .

According to the rules above, the value obtained in the example is:

```
TC3-HMAC-SHA256 Credential=AKID********************************/2019-02-25/cvm/tc
3_request, SignedHeaders=content-type;host, Signature=ca282b0a56549857d53b2beb08b
0c35871c892d42d09ae30b38d456e09ce291f
```

The following example shows a finished authorization header:

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKID********************************/20
19-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=ca282b0a5654
9857d53b2beb08b0c35871c892d42d09ae30b38d456e09ce291f
```

```
Content-Type: application/json; charset=utf-8
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1551113065
X-TC-Region: ap-guangzhou

{"Limit": 1, "Filters": [{"Values": ["unnamed"], "Name": "instance-name"}]}
```

5. Signature Demo

When calling API 3.0, you are recommended to use the corresponding Tencent Cloud SDK 3.0 which encapsulates the signature process, enabling you to focus on only the specific APIs provided by the product when developing. See SDK Center for more information. Currently, the following programming languages are supported:

- Python
- Java
- PHP
- Go
- NodeJS
- .NET

To further explain the signing process, we will use a programming language to implement the process described above. The request domain name, API and parameter values in the sample are used here. This goal of this example is only to provide additional clarification for the signature process, please see the SDK for actual usage.

The final output URL might be: https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID********************************&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12.

Note: The key in the example is fictitious, and the timestamp is not the current time of the system, so if this URL is opened in the browser or called using commands such as curl, an authentication error will be returned: Signature expired. In order to get a URL that can work properly, you need to replace the SecretId and SecretKey in the example with your real credentials and use the current time of the system as the Timestamp.

Note: In the example below, even if you use the same programming language, the order of the parameters in the URL may be different for each execution. However, the order does not matter, as long as all the parameters are included in the URL and the signature is calculated correctly.

Note: The following code is only applicable to API 3.0. It cannot be directly used in other signature processes. Even with an older API, signature calculation errors may occur due to the differences in details. Please refer to the corresponding documentation.

Java

```
import java.nio.charset.Charset;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TencentCloudAPITC3Demo {
private final static Charset UTF8 = StandardCharsets.UTF_8;
private final static String SECRET_ID = "AKID********************************";
private final static String SECRET_KEY = "********************************";
private final static String CT_JSON = "application/json; charset=utf-8";

public static byte[] hmac256(byte[] key, String msg) throws Exception {
Mac mac = Mac.getInstance("HmacSHA256");
SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
mac.init(secretKeySpec);
return mac.doFinal(msg.getBytes(UTF8));
}

public static String sha256Hex(String s) throws Exception {
MessageDigest md = MessageDigest.getInstance("SHA-256");
byte[] d = md.digest(s.getBytes(UTF8));
return DatatypeConverter.printHexBinary(d).toLowerCase();
}

public static void main(String[] args) throws Exception {
String service = "cvm";
String host = "cvm.tencentcloudapi.com";
String region = "ap-guangzhou";
String action = "DescribeInstances";
String version = "2017-03-12";
String algorithm = "TC3-HMAC-SHA256";
String timestamp = "1551113065";
//String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
// Pay attention to the time zone; otherwise, errors may occur
sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

// ************* Step 1: Concatenate the CanonicalRequest string *************
```

```java
String httpRequestMethod = "POST";
String canonicalUri = "/";
String canonicalQueryString = "";
String canonicalHeaders = "content-type:application/json; charset=utf-8\n" + "host:" + host + "\n";
String signedHeaders = "content-type;host";

String payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"unnamed\"], \"Name\": \"instance-name\"}]}";
String hashedRequestPayload = sha256Hex(payload);
String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryString + "\n"
+ canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
System.out.println(canonicalRequest);

// ************* Step 2: Concatenate the string to sign *************
String credentialScope = date + "/" + service + "/" + "tc3_request";
String hashedCanonicalRequest = sha256Hex(canonicalRequest);
String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope +
"\n" + hashedCanonicalRequest;
System.out.println(stringToSign);

// ************* Step 3: Calculate the signature *************
byte[] secretDate = hmac256(("TC3" + SECRET_KEY).getBytes(UTF8), date);
byte[] secretService = hmac256(secretDate, service);
byte[] secretSigning = hmac256(secretService, "tc3_request");
String signature = DatatypeConverter.printHexBinary(hmac256(secretSigning, stringToSign)).toLowerCase();
System.out.println(signature);

// ************* Step 4: Concatenate the Authorization *************
String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
System.out.println(authorization);

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("Content-Type", CT_JSON);
headers.put("Host", host);
headers.put("X-TC-Action", action);
headers.put("X-TC-Timestamp", timestamp);
headers.put("X-TC-Version", version);
headers.put("X-TC-Region", region);

StringBuilder sb = new StringBuilder();
sb.append("curl -X POST https://").append(host)
```

```
.append(" -H \"Authorization: ").append(authorization).append("\"")
.append(" -H \"Content-Type: application/json; charset=utf-8\"")
.append(" -H \"Host: ").append(host).append("\"")
.append(" -H \"X-TC-Action: ").append(action).append("\"")
.append(" -H \"X-TC-Timestamp: ").append(timestamp).append("\"")
.append(" -H \"X-TC-Version: ").append(version).append("\"")
.append(" -H \"X-TC-Region: ").append(region).append("\"")
.append(" -d '").append(payload).append("'");
System.out.println(sb.toString());
}
}
```

Python

```
# -*- coding: utf-8 -*-
import hashlib, hmac, json, os, sys, time
from datetime import datetime

# Key Parameters
secret_id = "AKID********************************"
secret_key = "********************************"

service = "cvm"
host = "cvm.tencentcloudapi.com"
endpoint = "https://" + host
region = "ap-guangzhou"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
#timestamp = int(time.time())
timestamp = 1551113065
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 1, "Filters": [{"Name": "instance-name", "Values": ["unnamed"
]}]}

# ************* Step 1: Concatenate the CanonicalRequest string *************
http_request_method = "POST"
canonical_uri = "/"
canonical_querystring = ""
ct = "application/json; charset=utf-8"
payload = json.dumps(params)
canonical_headers = "content-type:%s\nhost:%s\n" % (ct, host)
signed_headers = "content-type;host"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "\n" +
canonical_uri + "\n" +
canonical_querystring + "\n" +
canonical_headers + "\n" +
signed_headers + "\n" +
hashed_request_payload)
print(canonical_request)

# ************* Step 2: Concatenate the string to sign *************
credential_scope = date + "/" + service + "/" + "tc3_request"
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexd
igest()
string_to_sign = (algorithm + "\n" +
str(timestamp) + "\n" +
credential_scope + "\n" +
hashed_canonical_request)
print(string_to_sign)


# ************* Step 3: Calculate the Signature *************
# Function for computing signature digest
def sign(key, msg):
return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()
secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
secret_service = sign(secret_date, service)
secret_signing = sign(secret_service, "tc3_request")
signature = hmac.new(secret_signing, string_to_sign.encode("utf-8"), hashlib.sha2
56).hexdigest()
print(signature)

# ************* Step 4: Concatenate the Authorization *************
authorization = (algorithm + " " +
"Credential=" + secret_id + "/" + credential_scope + ", " +
"SignedHeaders=" + signed_headers + ", " +
"Signature=" + signature)
print(authorization)

print('curl -X POST ' + endpoint
+ ' -H "Authorization: ' + authorization + '"'
+ ' -H "Content-Type: application/json; charset=utf-8"'
+ ' -H "Host: ' + host + '"'
+ ' -H "X-TC-Action: ' + action + '"'
+ ' -H "X-TC-Timestamp: ' + str(timestamp) + '"'
+ ' -H "X-TC-Version: ' + version + '"'
+ ' -H "X-TC-Region: ' + region + '"'
+ " -d '" + payload + "'")
```

Golang

```
package main

import (
"crypto/hmac"
"crypto/sha256"
"encoding/hex"
"fmt"
"time"
)

func sha256hex(s string) string {
b := sha256.Sum256([]byte(s))
return hex.EncodeToString(b[:])
}

func hmacsha256(s, key string) string {
hashed := hmac.New(sha256.New, []byte(key))
hashed.Write([]byte(s))
return string(hashed.Sum(nil))
}

func main() {
secretId := "AKID********************************"
secretKey := "********************************"
host := "cvm.tencentcloudapi.com"
algorithm := "TC3-HMAC-SHA256"
service := "cvm"
version := "2017-03-12"
action := "DescribeInstances"
region := "ap-guangzhou"
//var timestamp int64 = time.Now().Unix()
var timestamp int64 = 1551113065

// step 1: build canonical request string
httpRequestMethod := "POST"
canonicalURI := "/"
canonicalQueryString := ""
canonicalHeaders := "content-type:application/json; charset=utf-8\n" + "host:" + host + "\n"
signedHeaders := "content-type;host"
payload := `{"Limit": 1, "Filters": [{"Values": ["unnamed"], "Name": "instance-name"}]}`
hashedRequestPayload := sha256hex(payload)
canonicalRequest := fmt.Sprintf("%s\n%s\n%s\n%s\n%s\n%s",
httpRequestMethod,
canonicalURI,
canonicalQueryString,
canonicalHeaders,
signedHeaders,
hashedRequestPayload)
fmt.Println(canonicalRequest)

// step 2: build string to sign
date := time.Unix(timestamp, 0).UTC().Format("2006-01-02")
credentialScope := fmt.Sprintf("%s/%s/tc3_request", date, service)
hashedCanonicalRequest := sha256hex(canonicalRequest)
string2sign := fmt.Sprintf("%s\n%d\n%s\n%s",
algorithm,
timestamp,
credentialScope,
hashedCanonicalRequest)
fmt.Println(string2sign)

// step 3: sign string
secretDate := hmacsha256(date, "TC3"+secretKey)
secretService := hmacsha256(service, secretDate)
secretSigning := hmacsha256("tc3_request", secretService)
signature := hex.EncodeToString([]byte(hmacsha256(string2sign, secretSigning)))
fmt.Println(signature)

// step 4: build authorization
authorization := fmt.Sprintf("%s Credential=%s/%s, SignedHeaders=%s, Signature=%
s",
algorithm,
secretId,
credentialScope,
signedHeaders,
signature)
fmt.Println(authorization)

curl := fmt.Sprintf(`curl -X POST https://%s\
-H "Authorization: %s"\
-H "Content-Type: application/json; charset=utf-8"\
-H "Host: %s" -H "X-TC-Action: %s"\
-H "X-TC-Timestamp: %d"\
-H "X-TC-Version: %s"\
-H "X-TC-Region: %s"\
-d '%s'`, host, authorization, host, action, timestamp, version, region, payload)
fmt.Println(curl)
}
```

PHP

```php
<?php
$secretId = "AKID********************************";
$secretKey = "********************************";
$host = "cvm.tencentcloudapi.com";
$service = "cvm";
$version = "2017-03-12";
$action = "DescribeInstances";
$region = "ap-guangzhou";
// $timestamp = time();
$timestamp = 1551113065;
$algorithm = "TC3-HMAC-SHA256";

// step 1: build canonical request string
$httpRequestMethod = "POST";
$canonicalUri = "/";
$canonicalQueryString = "";
$canonicalHeaders = "content-type:application/json; charset=utf-8\n"."host:".$host."\n";
$signedHeaders = "content-type;host";
$payload = '{"Limit": 1, "Filters": [{"Values": ["unnamed"], "Name": "instance-name"}]}';
$hashedRequestPayload = hash("SHA256", $payload);
$canonicalRequest = $httpRequestMethod."\n"
.$canonicalUri."\n"
.$canonicalQueryString."\n"
.$canonicalHeaders."\n"
.$signedHeaders."\n"
.$hashedRequestPayload;
echo $canonicalRequest.PHP_EOL;

// step 2: build string to sign
$date = gmdate("Y-m-d", $timestamp);
$credentialScope = $date."/".$service."/tc3_request";
$hashedCanonicalRequest = hash("SHA256", $canonicalRequest);
$stringToSign = $algorithm."\n"
.$timestamp."\n"
.$credentialScope."\n"
.$hashedCanonicalRequest;
echo $stringToSign.PHP_EOL;

// step 3: sign string
$secretDate = hash_hmac("SHA256", $date, "TC3".$secretKey, true);
$secretService = hash_hmac("SHA256", $service, $secretDate, true);
$secretSigning = hash_hmac("SHA256", "tc3_request", $secretService, true);
$signature = hash_hmac("SHA256", $stringToSign, $secretSigning);
echo $signature.PHP_EOL;
```

```
// step 4: build authorization
$authorization = $algorithm
." Credential=".$secretId."/".$credentialScope
.", SignedHeaders=content-type;host, Signature=".$signature;
echo $authorization.PHP_EOL;

$curl = "curl -X POST https://".$host
.' -H "Authorization: '.$authorization.'"'
.' -H "Content-Type: application/json; charset=utf-8"'
.' -H "Host: '.$host.'"'
.' -H "X-TC-Action: '.$action.'"'
.' -H "X-TC-Timestamp: '.$timestamp.'"'
.' -H "X-TC-Version: '.$version.'"'
.' -H "X-TC-Region: '.$region.'"'
." -d '".$payload."'";
echo $curl.PHP_EOL;
```

Ruby

```
# -*- coding: UTF-8 -*-
# require ruby>=2.3.0
require 'digest'
require 'json'
require 'time'
require 'openssl'

# Key Parameters
secret_id = 'AKID********************************'
secret_key = '********************************'

service = 'cvm'
host = 'cvm.tencentcloudapi.com'
endpoint = 'https://' + host
region = 'ap-guangzhou'
action = 'DescribeInstances'
version = '2017-03-12'
algorithm = 'TC3-HMAC-SHA256'
# timestamp = Time.now.to_i
timestamp = 1551113065
date = Time.at(timestamp).utc.strftime('%Y-%m-%d')

# ************* Step 1: Concatenate the CanonicalRequest string *************
http_request_method = 'POST'
canonical_uri = '/'
canonical_querystring = ''
```

```
canonical_headers = "content-type:application/json; charset=utf-8\nhost:#{host}\n"
signed_headers = 'content-type;host'
# params = { 'Limit' => 1, 'Filters' => [{ 'Name' => 'instance-name', 'Values' => ['unnamed'] }] }
# payload = JSON.generate(params, { 'ascii_only' => true, 'space' => ' ' })
# json will generate in random order, to get specified result in example, we hard-code it here.
payload = '{"Limit": 1, "Filters": [{"Values": ["unnamed"], "Name": "instance-name"}]}'
hashed_request_payload = Digest::SHA256.hexdigest(payload)
canonical_request = [
http_request_method,
canonical_uri,
canonical_querystring,
canonical_headers,
signed_headers,
hashed_request_payload,
].join("\n")

puts canonical_request

# ************* Step 2: Concatenate the string to sign *************
credential_scope = date + '/' + service + '/' + 'tc3_request'
hashed_request_payload = Digest::SHA256.hexdigest(canonical_request)
string_to_sign = [
algorithm,
timestamp.to_s,
credential_scope,
hashed_request_payload,
].join("\n")
puts string_to_sign

# ************* Step 3: Calculate the Signature *************
digest = OpenSSL::Digest.new('sha256')
secret_date = OpenSSL::HMAC.digest(digest, 'TC3' + secret_key, date)
secret_service = OpenSSL::HMAC.digest(digest, secret_date, service)
secret_signing = OpenSSL::HMAC.digest(digest, secret_service, 'tc3_request')
signature = OpenSSL::HMAC.hexdigest(digest, secret_signing, string_to_sign)
puts signature

# ************* Step 4: Concatenate the Authorization *************
authorization = "#{algorithm} Credential=#{secret_id}/#{credential_scope}, SignedHeaders=#{signed_headers}, Signature=#{signature}"
puts authorization

puts 'curl -X POST ' + endpoint \
+ ' -H "Authorization: ' + authorization + '"' \
+ ' -H "Content-Type: application/json; charset=utf-8"' \
+ ' -H "Host: ' + host + '"' \
+ ' -H "X-TC-Action: ' + action + '"' \
+ ' -H "X-TC-Timestamp: ' + timestamp.to_s + '"' \
+ ' -H "X-TC-Version: ' + version + '"' \
+ ' -H "X-TC-Region: ' + region + '"' \
+ " -d '" + payload + "'"
```

DotNet

```
using System;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Text;

public class Application
{
public static string SHA256Hex(string s)
{
using (SHA256 algo = SHA256.Create())
{
byte[] hashbytes = algo.ComputeHash(Encoding.UTF8.GetBytes(s));
StringBuilder builder = new StringBuilder();
for (int i = 0; i < hashbytes.Length; ++i)
{
builder.Append(hashbytes[i].ToString("x2"));
}
return builder.ToString();
}
}
public static byte[] HmacSHA256(byte[] key, byte[] msg)
{
using (HMACSHA256 mac = new HMACSHA256(key))
{
return mac.ComputeHash(msg);
}
}

public static Dictionary<String, String> BuildHeaders(string secretid,
string secretkey, string service, string endpoint, string region,
string action, string version, DateTime date, string requestPayload)
{
string datestr = date.ToString("yyyy-MM-dd");
DateTime startTime = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc);
long requestTimestamp = (long)Math.Round((date - startTime).TotalMilliseconds, Mi
dpointRounding.AwayFromZero) / 1000;
// ************* Step 1: Concatenate the CanonicalRequest string *************
string algorithm = "TC3-HMAC-SHA256";
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string contentType = "application/json";
string canonicalHeaders = "content-type:" + contentType + "; charset=utf-8\n" + "host:" + endpoint + "\n";
string signedHeaders = "content-type;host";
string hashedRequestPayload = SHA256Hex(requestPayload);
string canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload;
Console.WriteLine(canonicalRequest);
Console.WriteLine("----------------------------------");

// ************ Step 2: Concatenate the string to sign *************
string credentialScope = datestr + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = SHA256Hex(canonicalRequest);
string stringToSign = algorithm + "\n" + requestTimestamp.ToString() + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
Console.WriteLine(stringToSign);
Console.WriteLine("----------------------------------");

// ************* Step 3: Calculate the signature *************
byte[] tc3SecretKey = Encoding.UTF8.GetBytes("TC3" + secretkey);
byte[] secretDate = HmacSHA256(tc3SecretKey, Encoding.UTF8.GetBytes(datestr));
byte[] secretService = HmacSHA256(secretDate, Encoding.UTF8.GetBytes(service));
byte[] secretSigning = HmacSHA256(secretService, Encoding.UTF8.GetBytes("tc3_request"));
byte[] signatureBytes = HmacSHA256(secretSigning, Encoding.UTF8.GetBytes(stringToSign));
string signature = BitConverter.ToString(signatureBytes).Replace("-", "").ToLower();
Console.WriteLine(signature);
Console.WriteLine("----------------------------------");

// ************* Step 4: Concatenate the Authorization *************
string authorization = algorithm + " "
+ "Credential=" + secretid + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", "
+ "Signature=" + signature;
Console.WriteLine(authorization);
Console.WriteLine("----------------------------------");

Dictionary<string, string> headers = new Dictionary<string, string>();
headers.Add("Authorization", authorization);
headers.Add("Host", endpoint);
headers.Add("Content-Type", contentType + "; charset=utf-8");
headers.Add("X-TC-Timestamp", requestTimestamp.ToString());
headers.Add("X-TC-Version", version);
headers.Add("X-TC-Action", action);
headers.Add("X-TC-Region", region);
return headers;
}
public static void Main(string[] args)
{
// SecretID and SecretKey
string SECRET_ID = "AKID********************************";
string SECRET_KEY = "********************************";

string service = "cvm";
string endpoint = "cvm.tencentcloudapi.com";
string region = "ap-guangzhou";
string action = "DescribeInstances";
string version = "2017-03-12";

// The timestamp `2019-02-26 00:44:25` used here is only for reference. In a project, use the following parameter:
// DateTime date = DateTime.UtcNow;
// Enter the correct time zone. We recommend using UTC timestamp to avoid errors.
DateTime date = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc).AddSeconds(1551113065);
string requestPayload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}";

Dictionary<string, string> headers = BuildHeaders(SECRET_ID, SECRET_KEY, service, endpoint, region, action, version, date, requestPayload);

Console.WriteLine("POST https://cvm.tencentcloudapi.com");
foreach (KeyValuePair<string, string> kv in headers)
{
Console.WriteLine(kv.Key + ": " + kv.Value);
}
Console.WriteLine();
Console.WriteLine(requestPayload);
}
}
```

NodeJS

```
const crypto = require('crypto');

function sha256(message, secret = '', encoding) {
const hmac = crypto.createHmac('sha256', secret)
return hmac.update(message).digest(encoding)
}
function getHash(message, encoding = 'hex') {
const hash = crypto.createHash('sha256')
return hash.update(message).digest(encoding)
}
function getDate(timestamp) {
const date = new Date(timestamp * 1000)
const year = date.getUTCFullYear()
const month = ('0' + (date.getUTCMonth() + 1)).slice(-2)
const day = ('0' + date.getUTCDate()).slice(-2)
return `${year}-${month}-${day}`
}
function main(){

const SECRET_ID = "AKID********************************"
const SECRET_KEY = "********************************"

const endpoint = "cvm.tencentcloudapi.com"
const service = "cvm"
const region = "ap-guangzhou"
const action = "DescribeInstances"
const version = "2017-03-12"
//const timestamp = getTime()
const timestamp = 1551113065
const date = getDate(timestamp)

// ************* Step 1: Concatenate the CanonicalRequest string *************
const signedHeaders = "content-type;host"

const payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"unnamed\"], \"Name\": \"instance-name\"}]}"

const hashedRequestPayload = getHash(payload);
const httpRequestMethod = "POST"
const canonicalUri = "/"
const canonicalQueryString = ""
const canonicalHeaders = "content-type:application/json; charset=utf-8\n" + "host:" + endpoint + "\n"

const canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload
console.log(canonicalRequest)
console.log("----------------------------")

// ************* Step 2: Concatenate the string to sign *************
const algorithm = "TC3-HMAC-SHA256"
const hashedCanonicalRequest = getHash(canonicalRequest);
const credentialScope = date + "/" + service + "/" + "tc3_request"
const stringToSign = algorithm + "\n" +
timestamp + "\n" +
credentialScope + "\n" +
hashedCanonicalRequest
console.log(stringToSign)
console.log("----------------------------")

// ************* Step 3: Calculate the signature *************
const kDate = sha256(date, 'TC3' + SECRET_KEY)
const kService = sha256(service, kDate)
const kSigning = sha256('tc3_request', kService)
const signature = sha256(stringToSign, kSigning, 'hex')
console.log(signature)
console.log("----------------------------")

// ************* Step 4: Concatenate the Authorization *************
const authorization = algorithm + " " +
"Credential=" + SECRET_ID + "/" + credentialScope + ", " +
"SignedHeaders=" + signedHeaders + ", " +
"Signature=" + signature
console.log(authorization)
console.log("----------------------------")

const Call_Information = 'curl -X POST ' + "https://" + endpoint
+ ' -H "Authorization: ' + authorization + '"'
+ ' -H "Content-Type: application/json; charset=utf-8"'
+ ' -H "Host: ' + endpoint + '"'
+ ' -H "X-TC-Action: ' + action + '"'
+ ' -H "X-TC-Timestamp: ' + timestamp.toString() + '"'
+ ' -H "X-TC-Version: ' + version + '"'
+ ' -H "X-TC-Region: ' + region + '"'
+ " -d '" + payload + "'"
console.log(Call_Information)
}
main()
```

C++

```
#include <iostream>
#include <iomanip>
#include <sstream>
#include <string>
#include <stdio.h>
#include <time.h>
#include <openssl/sha.h>
#include <openssl/hmac.h>

using namespace std;

string get_data(int64_t &timestamp)
{
string utcDate;
char buff[20] = {0};
// time_t timenow;
struct tm sttime;
sttime = *gmtime(&timestamp);
strftime(buff, sizeof(buff), "%Y-%m-%d", &sttime);
utcDate = string(buff);
return utcDate;
}
string int2str(int64_t n)
{
std::stringstream ss;
ss << n;
return ss.str();
}
string sha256Hex(const string &str)
{
char buf[3];
unsigned char hash[SHA256_DIGEST_LENGTH];
SHA256_CTX sha256;
SHA256_Init(&sha256);
SHA256_Update(&sha256, str.c_str(), str.size());
SHA256_Final(hash, &sha256);
std::string NewString = "";
for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
{
snprintf(buf, sizeof(buf), "%02x", hash[i]);
NewString = NewString + buf;
}
return NewString;
```

```
}
string HmacSha256(const string &key, const string &input)
{
unsigned char hash[32];

HMAC_CTX *h;
#if OPENSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX hmac;
HMAC_CTX_init(&hmac);
h = &hmac;
#else
h = HMAC_CTX_new();
#endif

HMAC_Init_ex(h, &key[0], key.length(), EVP_sha256(), NULL);
HMAC_Update(h, ( unsigned char* )&input[0], input.length());
unsigned int len = 32;
HMAC_Final(h, hash, &len);

#if OPENSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX_cleanup(h);
#else
HMAC_CTX_free(h);
#endif

std::stringstream ss;
ss << std::setfill('0');
for (int i = 0; i < len; i++)
{
ss << hash[i];
}

return (ss.str());
}
string HexEncode(const string &input)
{
static const char* const lut = "0123456789abcdef";
size_t len = input.length();

string output;
output.reserve(2 * len);
for (size_t i = 0; i < len; ++i)
{
const unsigned char c = input[i];
output.push_back(lut[c >> 4]);
output.push_back(lut[c & 15]);
}
```

```
return output;
}

int main()
{
string SECRET_ID = "AKID********************************";
string SECRET_KEY = "********************************";

string service = "cvm";
string host = "cvm.tencentcloudapi.com";
string region = "ap-guangzhou";
string action = "DescribeInstances";
string version = "2017-03-12";
int64_t timestamp = 1551113065;
string date = get_data(timestamp);

// ************* Step 1: Concatenate the CanonicalRequest string *************
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string canonicalHeaders = "content-type:application/json; charset=utf-8\nhost:" +
host + "\n";
string signedHeaders = "content-type;host";
string payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"unnamed\"], \"Name
\": \"instance-name\"}]}";
string hashedRequestPayload = sha256Hex(payload);
string canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canoni
calQueryString + "\n"
+ canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
cout << canonicalRequest << endl;
cout << "-----------------------" << endl;

// ************* Step 2: Concatenate the string to sign *************
string algorithm = "TC3-HMAC-SHA256";
string RequestTimestamp = int2str(timestamp);
string credentialScope = date + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = sha256Hex(canonicalRequest);
string stringToSign = algorithm + "\n" + RequestTimestamp + "\n" + credentialScop
e + "\n" + hashedCanonicalRequest;
cout << stringToSign << endl;
cout << "-----------------------" << endl;

// ************* Step 3: Calculate the signature ***************
string kKey = "TC3" + SECRET_KEY;
string kDate = HmacSha256(kKey, date);
string kService = HmacSha256(kDate, service);
string kSigning = HmacSha256(kService, "tc3_request");
string signature = HexEncode(HmacSha256(kSigning, stringToSign));
cout << signature << endl;
cout << "-----------------------" << endl;

// ************* Step 4: Concatenate the Authorization *************
string authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
cout << authorization << endl;
cout << "------------------------" << endl;

string headers = "curl -X POST https://" + host + "\n"
+ " -H \"Authorization: " + authorization + "\n"
+ " -H \"Content-Type: application/json; charset=utf-8\"" + "\n"
+ " -H \"Host: " + host + "\n"
+ " -H \"X-TC-Action: " + action + "\n"
+ " -H \"X-TC-Timestamp: " + RequestTimestamp + "\n"
+ " -H \"X-TC-Version: " + version + "\n"
+ " -H \"X-TC-Region: " + region + "\n"
+ " -d '" + payload;
cout << headers << endl;
return 0;
};
```

Signature Failure

The following situational error codes for signature failure may occur. Please resolve the errors accordingly.

Error Code	Description
AuthFailure.SignatureExpire	Signature expired. Timestamp and server time cannot differ by more than five minutes.
AuthFailure.SecretIdNotFound	The key does not exist. Please go to the console to check whether it is disabled or you copied fewer or more characters.
AuthFailure.SignatureFailure	Signature error. It is possible that the signature was calculated incorrectly, the signature does not match the content actually sent, or the SecretKey is incorrect.
AuthFailure.TokenFailure	Temporary certificate token error.
AuthFailure.InvalidSecretId	Invalid key (not a TencentCloud API key type).

Signature

Last updated：2024-11-27 16:09:10

Tencent Cloud API authenticates each access request, i.e. each request needs to include authentication information (Signature) in the common parameters to verify the identity of the requester.
The Signature is generated by the security credentials which include SecretId and SecretKey. If you don't have the security credentials yet, go to the TencentCloud API Key page to apply for them; otherwise, you cannot invoke the TencentCloud API.

1. Applying for Security Credentials

Before using the TencentCloud API for the first time, go to the TencentCloud API Key page to apply for security credentials.
Security credentials consist of SecretId and SecretKey:

- SecretId is used to identify the API requester.
- SecretKey is used to encrypt the signature string and verify it on the server.
- **You must keep your security credentials private and avoid disclosure.**

You can apply for the security credentials through the following steps:

1. Log in to the Tencent Cloud Console.
2. Go to the TencentCloud API Key page.
3. On the API Key Management page, click **Create Key** to create a SecretId/SecretKey pair.

Note: Each account can have up to two pairs of SecretId/SecretKey.

2. Generating a Signature

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

Assume that the SecretId and SecretKey are:

- SecretId: `AKID********************************`
- SecretKey: `********************************`

Note: This is just an example. For actual operations, please use your own SecretId and SecretKey.

Take the Cloud Virtual Machine's request to view the instance list (DescribeInstances) as an example. When you invoke this API, the request parameters may be as follows:

Parameter name	Description	Parameter value
Action	Method name	DescribeInstances
SecretId	Key ID	`AKID********************************`
Timestamp	Current timestamp	1465185768
Nonce	Random positive integer	11886
Region	Region where the instance is located	ap-guangzhou
InstanceIds.0	ID of the instance to query	ins-09dx96dg
Offset	Offset	0
Limit	Allowed maximum output	20
Version	API version number	2017-03-12

2.1. Sorting Parameters

First, sort all the request parameters in an ascending lexicographical order (ASCII code) by their names. Notes: (1) Parameters are sorted by their names instead of their values; (2) The parameters are sorted based on ASCII code, not in an alphabetical order or by values. For example, InstanceIds.2 should be arranged after InstanceIds.12. You can complete the sorting process using a sorting function in a programming language, such as the ksort function in PHP. The parameters in the example are sorted as follows:

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

When developing in another programming language, you can sort these sample parameters and it will work as long as you obtain the same results.

2.2. Concatenating a Request String

This step generates a request string.

Format the request parameters sorted in the previous step into the form of "parameter name"="parameter value". For example, for the Action parameter, its parameter name is "Action" and its parameter value is "DescribeInstances", so it will become Action=DescribeInstances after formatted.

Note: The "parameter value" is the original value but not the value after URL encoding.

Then, concatenate the formatted parameters with "&". The resulting request string is as follows:

```
Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0
&Region=ap-guangzhou&SecretId=AKID********************************&Timestamp=1465
185768&Version=2017-03-12
```

2.3. Concatenating the Signature Original String

This step generates a signature original string.

The signature original string consists of the following parameters:

1. HTTP method: POST and GET modes are supported, and GET is used here for the request. Please note that the method name should be in all capital letters.
2. Request server: the domain name of the request to view the list of instances (DescribeInstances) is cvm.tencentcloudapi.com. The actual request domain name varies by the module to which the API belongs. For more information, see the instructions of the specific API.
3. Request path: The request path in the current version of TencentCloud API is fixed to /.
4. Request string: the request string generated in the previous step.

The concatenation rule of the signature original string is: Request method + request host + request path + ? + request string

The concatenation result of the example is:

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&L
imit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID*******************
*************&Timestamp=1465185768&Version=2017-03-12
```

2.4. Generating a Signature String

This step generates a signature string.

First, use the HMAC-SHA1 algorithm to sign the **signature original string** obtained in the previous step, and then encode the generated signature using Base64 to obtain the final signature.

The specific code is as follows with the PHP language being used as an example:

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

The final signature is:

```
9FzTQAN1UZ489+BqCg1fNBQaCqw=
```

When developing in another programming language, you can sign and verify the original in the example above and it works as long as you get the same results.

3. Encoding a Signature String

The generated signature string cannot be directly used as a request parameter and must be URL encoded.

For example, if the signature string generated in the previous step is 9FzTQAN1UZ489+BqCg1fNBQaCqw=, the final signature string request parameter (Signature) is 9FzTQAN1UZ489%2BBqCg1fNBQaCqw%3D, which will be used to generate the final request URL.

Note: If your request method is GET, or the request method is POST and the Content-Type is application/x-www-form-urlencoded, then all the request parameter values need to be URL encoded (except the parameter key and the symbol of =) when sending the request. Non-ASCII characters need to be encoded with UTF-8 before URL encoding.

Note: The network libraries of some programming languages automatically URL encode all parameters, in which case there is no need to URL encode the signature string; otherwise, two rounds of URL encoding will cause the signature to fail.

Note: Other parameter values also need to be encoded using RFC 3986. Use %XY in percent-encoding for special characters such as Chinese characters, where "X" and "Y" are hexadecimal characters (0-9 and uppercase A-F), and using lowercase will cause an error.

4. Signature Failure

The following situational error codes for signature failure may occur. Please resolve the errors accordingly.

Error code	Error description
AuthFailure.SignatureExpire	The signature is expired
AuthFailure.SecretIdNotFound	The key does not exist
AuthFailure.SignatureFailure	Signature error
AuthFailure.TokenFailure	Token error
AuthFailure.InvalidSecretId	Invalid key (not a TencentCloud API key type)

5. Signature Demo

When calling API 3.0, you are recommended to use the corresponding Tencent Cloud SDK 3.0 which encapsulates the signature process, enabling you to focus on only the specific APIs provided by the product when developing. See SDK Center for more information. Currently, the following programming languages are supported:

- Python
- Java
- PHP
- Go
- NodeJS
- .NET

To further explain the signing process, we will use a programming language to implement the process described above. The request domain name, API and parameter values in the sample are used here. This goal of this example is only to provide additional clarification for the signature process, please see the SDK for actual usage.

The final output URL might be: `https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID********************************&Signature=9FzTQAN1UZ489%2BBqCg1fNBQaCqw%3D&Timestamp=1465185768&Version=2017-03-12` .

Note: The key in the example is fictitious, and the timestamp is not the current time of the system, so if this URL is opened in the browser or called using commands such as curl, an authentication error will be returned: Signature expired. In order to get a URL that can work properly, you need to replace the SecretId and SecretKey in the example with your real credentials and use the current time of the system as the Timestamp.

Note: In the example below, even if you use the same programming language, the order of the parameters in the URL may be different for each execution. However, the order does not matter, as long as all the parameters are included in the URL and the signature is calculated correctly.

Note: The following code is only applicable to API 3.0. It cannot be directly used in other signature processes. Even with an older API, signature calculation errors may occur due to the differences in details. Please refer to the corresponding documentation.

Java

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

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

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

public static String getStringToSign(TreeMap<String, Object> params) {
StringBuilder s2s = new StringBuilder("GETcvm.tencentcloudapi.com/?");
// When signing, the parameters need to be sorted in lexicographical order. TreeMap is used here to guarantee the correct order.
for (String k : params.keySet()) {
s2s.append(k).append("=").append(params.get(k).toString()).append("&");
}
return s2s.toString().substring(0, s2s.length() - 1);
}

public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException {
StringBuilder url = new StringBuilder("https://cvm.tencentcloudapi.com/?");
// There is no requirement for the order of the parameters in the actual request URL.
for (String k : params.keySet()) {
```

```
// The request string needs to be URL encoded. As the Key is all in English letters, only the value is URL encoded here.
url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).append("&");
}
return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap enables automatic sorting
// A random number should be used when actually calling, for example: params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
params.put("Nonce", 11886); // Common parameter
// The current time of the system should be used when actually calling, for example: params.put("Timestamp", System.currentTimeMillis() / 1000);
params.put("Timestamp", 1465185768); // Common parameter
params.put("SecretId", "AKID********************************"); // Common parameter
params.put("Action", "DescribeInstances"); // Common parameter
params.put("Version", "2017-03-12"); // Common parameter
params.put("Region", "ap-guangzhou"); // Common parameter
params.put("Limit", 20); // Business parameter
params.put("Offset", 0); // Business parameter
params.put("InstanceIds.0", "ins-09dx96dg"); // Business parameter
params.put("Signature", sign(getStringToSign(params), "********************************", "HmacSHA1")); // Common parameter
System.out.println(getUrl(params));
}
}
```

Python

Note: If running in a Python 2 environment, the following requests dependency package must be installed first: `pip install requests`.

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

import requests

secret_id = "AKID********************************"
secret_key = "********************************"
```

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

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

if __name__ == '__main__':
endpoint = "cvm.tencentcloudapi.com"
data = {
'Action' : 'DescribeInstances',
'InstanceIds.0' : 'ins-09dx96dg',
'Limit' : 20,
'Nonce' : 11886,
'Offset' : 0,
'Region' : 'ap-guangzhou',
'SecretId' : secret_id,
'Timestamp' : 1465185768, # int(time.time())
'Version': '2017-03-12'
}
s = get_string_to_sign("GET", endpoint, data)
data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
print(data["Signature"])
# An actual invocation would occur here, which may incur fees after success
# resp = requests.get("https://" + endpoint, params=data)
# print(resp.url)
```

Golang

```
package main

import (
"bytes"
"crypto/hmac"
"crypto/sha1"
"encoding/base64"
"fmt"
"sort"
)

func main() {
secretId := "AKID********************************"
secretKey := "********************************"
params := map[string]string{
"Nonce": "11886",
"Timestamp": "1465185768",
"Region": "ap-guangzhou",
"SecretId": secretId,
"Version": "2017-03-12",
"Action": "DescribeInstances",
"InstanceIds.0": "ins-09dx96dg",
"Limit": "20",
"Offset": "0",
}

var buf bytes.Buffer
buf.WriteString("GET")
buf.WriteString("cvm.tencentcloudapi.com")
buf.WriteString("/")
buf.WriteString("?")

// sort keys by ascii asc order
keys := make([]string, 0, len(params))
for k, _ := range params {
keys = append(keys, k)
}
sort.Strings(keys)

for i := range keys {
k := keys[i]
buf.WriteString(k)
buf.WriteString("=")
buf.WriteString(params[k])
buf.WriteString("&")
}
buf.Truncate(buf.Len() - 1)

hashed := hmac.New(sha1.New, []byte(secretKey))
hashed.Write(buf.Bytes())

fmt.Println(base64.StdEncoding.EncodeToString(hashed.Sum(nil)))
}
```

PHP

```
<?php
$secretId = "AKID********************************";
$secretKey = "********************************";
$param["Nonce"] = 11886;//rand();
```

```
$param["Timestamp"] = 1465185768;//time();
$param["Region"] = "ap-guangzhou";
$param["SecretId"] = $secretId;
$param["Version"] = "2017-03-12";
$param["Action"] = "DescribeInstances";
$param["InstanceIds.0"] = "ins-09dx96dg";
$param["Limit"] = 20;
$param["Offset"] = 0;

ksort($param);

$signStr = "GETcvm.tencentcloudapi.com/?";
foreach ( $param as $key => $value ) {
$signStr = $signStr . $key . "=" . $value . "&";
}
$signStr = substr($signStr, 0, -1);

$signature = base64_encode(hash_hmac("sha1", $signStr, $secretKey, true));
echo $signature.PHP_EOL;
// need to install and enable curl extension in php.ini
// $param["Signature"] = $signature;
// $url = "https://cvm.tencentcloudapi.com/?".http_build_query($param);
// echo $url.PHP_EOL;
// $ch = curl_init();
// curl_setopt($ch, CURLOPT_URL, $url);
// $output = curl_exec($ch);
// curl_close($ch);
// echo json_decode($output);
```

Ruby

```
# -*- coding: UTF-8 -*-
# require ruby>=2.3.0
require 'time'
require 'openssl'
require 'base64'

secret_id = "AKID********************************"
secret_key = "********************************"

method = 'GET'
endpoint = 'cvm.tencentcloudapi.com'
data = {
'Action' => 'DescribeInstances',
'InstanceIds.0' => 'ins-09dx96dg',
'Limit' => 20,
'Nonce' => 11886,
'Offset' => 0,
'Region' => 'ap-guangzhou',
'SecretId' => secret_id,
'Timestamp' => 1465185768, # Time.now.to_i
'Version' => '2017-03-12',
}
sign = method + endpoint + '/?'
params = []
data.sort.each do |item|
params << "#{item[0]}=#{item[1]}"
end
sign += params.join('&')
digest = OpenSSL::Digest.new('sha1')
data['Signature'] = Base64.encode64(OpenSSL::HMAC.digest(digest, secret_key, sig
n))
puts data['Signature']

# require 'net/http'
# uri = URI('https://' + endpoint)
# uri.query = URI.encode_www_form(data)
# p uri
# res = Net::HTTP.get_response(uri)
# puts res.body
```

DotNet

```
using System;
using System.Collections.Generic;
using System.Net;
using System.Security.Cryptography;
using System.Text;

public class Application {
public static string Sign(string signKey, string secret)
{
string signRet = string.Empty;
using (HMACSHA1 mac = new HMACSHA1(Encoding.UTF8.GetBytes(signKey)))
{
byte[] hash = mac.ComputeHash(Encoding.UTF8.GetBytes(secret));
signRet = Convert.ToBase64String(hash);
}
return signRet;
}
public static string MakeSignPlainText(SortedDictionary<string, string> requestPa
rams, string requestMethod, string requestHost, string requestPath)
```

```
{
string retStr = "";
retStr += requestMethod;
retStr += requestHost;
retStr += requestPath;
retStr += "?";
string v = "";
foreach (string key in requestParams.Keys)
{
v += string.Format("{0}={1}&", key, requestParams[key]);
}
retStr += v.TrimEnd('&');
return retStr;
}

public static void Main(string[] args)
{
string SECRET_ID = "AKID********************************";
string SECRET_KEY = "********************************";

string endpoint = "cvm.tencentcloudapi.com";
string region = "ap-guangzhou";
string action = "DescribeInstances";
string version = "2017-03-12";
double RequestTimestamp = 1465185768;
// long timestamp = ToTimestamp() / 1000;
// string requestTimestamp = timestamp.ToString();
Dictionary<string, string> param = new Dictionary<string, string>();
param.Add("Limit", "20");
param.Add("Offset", "0");
param.Add("InstanceIds.0", "ins-09dx96dg");
param.Add("Action", action);
param.Add("Nonce", "11886");
// param.Add("Nonce", Math.Abs(new Random().Next()).ToString());

param.Add("Timestamp", RequestTimestamp.ToString());
param.Add("Version", version);

param.Add("SecretId", SECRET_ID);
param.Add("Region", region);
SortedDictionary<string, string> headers = new SortedDictionary<string, string>(p
aram, StringComparer.Ordinal);
string sigInParam = MakeSignPlainText(headers, "GET", endpoint, "/");
Console.WriteLine(sigInParam);
string sigOutParam = Sign(SECRET_KEY, sigInParam);
Console.WriteLine("GET https://cvm.tencentcloudapi.com");
foreach (KeyValuePair<string, string> kv in headers)
{
Console.WriteLine(kv.Key + ": " + kv.Value);
}
Console.WriteLine("Signature" + ": " + WebUtility.UrlEncode(sigOutParam));
Console.WriteLine();

string result = "https://cvm.tencentcloudapi.com/?";
foreach (KeyValuePair<string, string> kv in headers)
{
result += WebUtility.UrlEncode(kv.Key) + "=" + WebUtility.UrlEncode(kv.Value) +
"&";
}
result += WebUtility.UrlEncode("Signature") + "=" + WebUtility.UrlEncode(sigOutPa
ram);
Console.WriteLine("GET " + result);
}
}
```

NodeJS

```
const crypto = require('crypto');

function get_req_url(params, endpoint){
params['Signature'] = escape(params['Signature']);
const url_strParam = sort_params(params)
return "https://" + endpoint + "/?" + url_strParam.slice(1);
}

function formatSignString(reqMethod, endpoint, path, strParam){
let strSign = reqMethod + endpoint + path + "?" + strParam.slice(1);
return strSign;
}
function sha1(secretKey, strsign){
let signMethodMap = {'HmacSHA1': "sha1"};
let hmac = crypto.createHmac(signMethodMap['HmacSHA1'], secretKey || "");
return hmac.update(Buffer.from(strsign, 'utf8')).digest('base64')
}

function sort_params(params){
let strParam = "";
let keys = Object.keys(params);
keys.sort();
for (let k in keys) {
//k = k.replace(/_/g, '.');
strParam += ("&" + keys[k] + "=" + params[keys[k]]);
}
return strParam
}

function main(){
const SECRET_ID = "AKID********************************"
const SECRET_KEY = "********************************"

const endpoint = "cvm.tencentcloudapi.com"
const Region = "ap-guangzhou"
const Version = "2017-03-12"
const Action = "DescribeInstances"
const Timestamp = 1465185768
// const Timestamp = Math.round(Date.now() / 1000)
const Nonce = 11886
//const nonce = Math.round(Math.random() * 65535)

let params = {};
params['Action'] = Action;
params['InstanceIds.0'] = 'ins-09dx96dg';
params['Limit'] = 20;
params['Offset'] = 0;
params['Nonce'] = Nonce;
params['Region'] = Region;
params['SecretId'] = SECRET_ID;
params['Timestamp'] = Timestamp;
params['Version'] = Version;

strParam = sort_params(params)

const reqMethod = "GET";
const path = "/";
strSign = formatSignString(reqMethod, endpoint, path, strParam)
console.log(strSign)
console.log("-----------------------")

params['Signature'] = sha1(SECRET_KEY, strSign)
console.log(params['Signature'])
console.log("-----------------------")

const req_url = get_req_url(params, endpoint)
console.log(params['Signature'])
console.log("-----------------------")
console.log(req_url)
}
main()
```

Responses

Last updated：2024-11-27 16:09:11

Response for Successful Requests

For example, when calling CAM API (version: 2017-03-12) to view the status of instances (DescribeInstancesStatus), if the request has succeeded, you may see the response as shown below:

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

- The API will return `Response` , which contains `RequestId` , as long as it processes the request. It does not matter if the request is successful or not.
- RequestId is the unique ID of an API request. Contact us with this ID when an exception occurs.
- Except for the fixed fields, all fields are action-specified. For the definitions of action-specified fields, see the corresponding API documentation. In this example, `TotalCount` and `InstanceStatusSet` are the fields specified by the API `DescribeInstancesStatus` . 0 `TotalCount` means that the requester owns 0 CVM instance so the `InstanceStatusSet` is empty.

Response for Failed Requests

If the request has failed, you may see the response as shown below:

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

- The presence of the `Error` field indicates that the request has failed. A response for a failed request will include `Error` , `Code` and `Message` fields.
- `Code` is the code of the error that helps you identify the cause and solution. There are two types of error codes so you may find the code in either common error codes or API-specified error codes.
- Message explains the cause of the error. Note that the returned messages are subject to service updates. The information the messages provide may not be up-to-date and should not be the only source of reference.
- RequestId is the unique ID of an API request. Contact us with this ID when an exception occurs.

Common Error Codes

If there is an Error field in the response, it means that the API call failed. The Code field in Error indicates the error code. The following table lists the common error codes that all actions can return.

Error Code	Description
AuthFailure.InvalidSecretId	Invalid key (not a TencentCloud API key type).
AuthFailure.MFAFailure	MFA failed.
AuthFailure.SecretIdNotFound	The key does not exist.
AuthFailure.SignatureExpire	Signature expired.
AuthFailure.SignatureFailure	Signature error.
AuthFailure.TokenFailure	Token error.
AuthFailure.UnauthorizedOperation	The request does not have CAM authorization.
DryRunOperation	DryRun Operation. It means that the request would have succeeded, but the DryRun parameter was used.
FailedOperation	Operation failed.
InternalError	Internal error.
InvalidAction	The API does not exist.
InvalidParameter	Incorrect parameter.
InvalidParameterValue	Invalid parameter value.
LimitExceeded	Quota limit exceeded.
MissingParameter	A parameter is missing.

NoSuchVersion	The API version does not exist.
RequestLimitExceeded	The number of requests exceeds the frequency limit.
ResourceInUse	Resource is in use.
ResourceInsufficient	Insufficient resource.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	Resource is unavailable.
UnauthorizedOperation	Unauthorized operation.
UnknownParameter	Unknown parameter.
UnsupportedOperation	Unsupported operation.
UnsupportedProtocol	HTTPS request method error. Only GET and POST requests are supported.
UnsupportedRegion	API does not support the requested region.

Topic APIs

ModifyTopicAttribute

Last updated：2024-11-27 16:09:13

1. API Description

Domain name for API request: cmq.intl.tencentcloudapi.com.

This API is used to modify topic attributes.

A maximum of 20 requests can be initiated per second for this API.

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: ModifyTopicAttribute.
Version	Yes	String	Common Params. The value used for this API: 2019-03-04.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product.
TopicName	Yes	String	Topic name, which is unique under the same account in an individual region. It is a string of up to 64 characters, which must begin with a letter and can contain letters, digits, and dashes (-).
MaxMsgSize	No	Integer	Maximum message length. Value range: 1024-65536 bytes (i.e., 1-64 KB). Default value: 65536.
MsgRetentionSeconds	No	Integer	Message retention period. Value range: 60-86400 seconds

			(i.e., 1 minute-1 day). Default value: 86400.
Trace	No	Boolean	Whether to enable message trace. true: yes, false: no. If this field is left empty, the feature will not be enabled.

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Modifying topic attributes

Input Example

```
https://cmq.intl.tencentcloudapi.com/?Action=ModifyTopicAttribute
&TopicName=ConnTopic&Qps=20000
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "8a04c6b1-dec5-4979-b3b2-34ab3b3402b2"
}
}
```

5. Developer Resources

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation	Operation failed.
FailedOperation.TryLater	The operation is in progress. Please try again later.
InvalidParameter	Invalid parameter.
LimitExceeded	The quota limit is exceeded.
ResourceInUse	The resource is in use.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	The resource is unavailable.
ResourcesSoldOut	The resources have been sold out.
UnauthorizedOperation	Unauthorized operation.

DescribeTopicDetail

Last updated：2024-11-27 16:09:14

1. API Description

Domain name for API request: cmq.intl.tencentcloudapi.com.

This API is used to query topic details.

A maximum of 20 requests can be initiated per second for this API.

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: DescribeTopicDetail.
Version	Yes	String	Common Params. The value used for this API: 2019-03-04.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product.
Offset	No	Integer	Starting position of queue list to be returned on the current page in case of paginated return. If a value is entered, `limit` is required. If this parameter is left empty, 0 will be used by default.
Limit	No	Integer	Number of queues to be returned per page in case of paginated return. If this parameter is not passed in, 20 will be used by default. Maximum value: 50.
Filters.N	No	Array of Filter	Currently, only filtering by `TopicName` is supported, and only one filter value can be entered

TagKey	No	String	Tag match
TopicName	No	String	Exact match by `TopicName`

3. Output Parameters

Parameter Name	Type	Description
TotalCount	Integer	TotalCount
TopicSet	Array of TopicSet	TopicSet
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Querying topic details

Input Example

```
https://cmq.intl.tencentcloudapi.com/?Action=DescribeTopicDetail
&Offset=0&Limit=2
&<Common request parameters>
```

Output Example

```
{
"Response": {
"TotalCount": 2,
"TopicSet": [
{
"Tags": [],
"TopicId": "topic-rga4l1o4",
"TopicName": "ConnTopic",
"CreateUin": 20548499,
"MsgRetentionSeconds": 10000,
"MaxMsgSize": 20000,
"Qps": 10000,
"FilterType": 1,
"CreateTime": 1581516588,
"LastModifyTime": 1581563581,
"MsgCount": 0
},
{
"Tags": [],
"TopicId": "topic-388k6x98",
"TopicName": "test123",
"CreateUin": 20548499,
"MsgRetentionSeconds": 86400,
"MaxMsgSize": 65536,
"Qps": 5000,
"FilterType": 1,
"CreateTime": 1581493669,
"LastModifyTime": 1581495310,
"MsgCount": 0
}
],
"RequestId": "68ebb46b-5eac-467a-9942-1a5da83a65ca"
}
}
```

5. Developer Resources

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation	Operation failed.
FailedOperation.TryLater	The operation is in progress. Please try again later.
InvalidParameter	Invalid parameter.
InvalidParameterValue.NoTaskId	The task ID does not exist.
LimitExceeded	The quota limit is exceeded.
ResourceInUse	The resource is in use.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	The resource is unavailable.
ResourcesSoldOut	The resources have been sold out.
UnauthorizedOperation	Unauthorized operation.

DeleteTopic

Last updated：2024-11-27 16:09:15

1. API Description

Domain name for API request: cmq.intl.tencentcloudapi.com.

This API is used to delete a topic.

A maximum of 20 requests can be initiated per second for this API.

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: DeleteTopic.
Version	Yes	String	Common Params. The value used for this API: 2019-03-04.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product.
TopicName	Yes	String	Topic name, which is unique under the same account in an individual region. It is a string of up to 64 characters, which must begin with a letter and can contain letters, digits, and dashes (-).

3. Output Parameters

Parameter Name	Type	Description

RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Deleting topic

Input Example

```
https://cmq.intl.tencentcloudapi.com/?Action=DeleteTopic
&TopicName=ConnTopic
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "96b28b30-393a-48aa-9f00-07dbe6d86229"
}
}
```

5. Developer Resources

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation	Operation failed.
FailedOperation.TryLater	The operation is in progress. Please try again later.
InvalidParameter	Invalid parameter.
LimitExceeded	The quota limit is exceeded.
ResourceInUse	The resource is in use.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	The resource is unavailable.
ResourcesSoldOut	The resources have been sold out.
UnauthorizedOperation	Unauthorized operation.

CreateTopic

Last updated：2024-11-27 16:09:16

1. API Description

Domain name for API request: cmq.intl.tencentcloudapi.com.

This API is used to create a topic.

A maximum of 20 requests can be initiated per second for this API.

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: CreateTopic.
Version	Yes	String	Common Params. The value used for this API: 2019-03-04.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product.
TopicName	Yes	String	Topic name, which is unique under the same account in an individual region. It is a string of up to 64 characters, which must begin with a letter and can contain letters, digits, and dashes (-).
MaxMsgSize	No	Integer	Maximum message length. Value range: 1024-65536 bytes (i.e., 1-64 KB). Default value: 65536.
FilterType	No	Integer	Message match policy for a specified topic.
MsgRetentionSeconds	No	Integer	Message retention period. Value range: 60-86400 seconds

			(i.e., 1 minute-1 day). Default value: 86400.
Trace	No	Boolean	Whether to enable message trace. true: yes, false: no. If this field is left empty, the feature will not be enabled.

3. Output Parameters

Parameter Name	Type	Description
TopicId	String	TopicName
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Creating topic

Input Example

```
https://cmq.intl.tencentcloudapi.com/?Action=CreateTopic
&TopicName=ConnTopic
&<Common request parameters>
```

Output Example

```
{
"Response": {
"TopicId": "topic-gzz05csc",
"RequestId": "3e0dff9d-9ed5-47c3-beb2-a42c1d69e1cc"
}
}
```

5. Developer Resources

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation	Operation failed.
FailedOperation.TryLater	The operation is in progress. Please try again later.
InvalidParameter	Invalid parameter.
LimitExceeded	The quota limit is exceeded.
ResourceInUse	The resource is in use.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	The resource is unavailable.
ResourcesSoldOut	The resources have been sold out.
UnauthorizedOperation	Unauthorized operation.

Subscription APIs

ModifySubscriptionAttribute

Last updated：2024-11-27 16:09:17

1. API Description

Domain name for API request: cmq.intl.tencentcloudapi.com.

This API is used to modify subscription attributes.

A maximum of 20 requests can be initiated per second for this API.

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: ModifySubscriptionAttribute.
Version	Yes	String	Common Params. The value used for this API: 2019-03-04.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product.
TopicName	Yes	String	Topic name, which is unique under the same account in an individual region. It is a string of up to 64 characters, which must begin with a letter and can contain letters, digits, and dashes (-).
SubscriptionName	Yes	String	Subscription name, which is unique in the same topic under the same account in an individual region. It is a string of up to 64

			characters, which must begin with a letter and can contain letters, digits, and dashes (`-`).
NotifyStrategy	No	String	CMQ push server retry policy in case an error occurs while pushing a message to `Endpoint` . Valid values: 1. BACKOFF_RETRY: backoff retry, which is to retry at a fixed interval, discard the message after a certain number of retries, and continue to push the next message. 2. EXPONENTIAL_DECAY_RETRY: exponential decay retry, which is to retry at an exponentially increasing interval, such as 1s, 2s, 4s, 8s, and so on. As a message can be retained in a topic for one day, failed messages will be discarded at most after one day of retry. Default value: EXPONENTIAL_DECAY_RETRY.
NotifyContentFormat	No	String	Push content format. Valid values: 1. JSON, 2. SIMPLIFIED, i.e., the raw format. If `Protocol` is `queue` , this value must be `SIMPLIFIED` . If `Protocol` is `HTTP` , both options are acceptable, and the default value is `JSON` .
FilterTags.N	No	Array of String	Message body tag (used for message filtering). The number of tags cannot exceed 5, and each tag can contain up to 16 characters. It is used in conjunction with the `MsgTag` parameter of `(Batch)PublishMessage` . Rules: 1. If `FilterTag` is not set, no matter whether `MsgTag` is set, the subscription will receive all messages published to the topic; 2. If the `FilterTag` array has a value, only when at least one of the values in the array also exists in the `MsgTag` array (i.e., `FilterTag` and `MsgTag` have an intersection) can the subscription receive messages published to the topic; 3. If the `FilterTag` array has a value, but `MsgTag` is not set, then no message published to the topic will be received, which can be considered as a special case of rule 2 as `FilterTag` and `MsgTag` do not intersect in this case. The overall design idea of rules is based on the intention of the subscriber.
BindingKey.N	No	Array of String	The number of `BindingKey` cannot exceed 5, and the length of each `BindingKey` cannot exceed 64 bytes. This field indicates the filtering policy for subscribing to and receiving messages. Each `BindingKey` can contain up to 15 `.` , i.e., up to 16 phrases.

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Modifying subscription attributes

This example shows you how to modify subscription attributes.

Input Example

```
https://cmq.intl.tencentcloudapi.com/?Action=ModifySubscriptionAttribute
&TopicName=test&SubscriptionName=test&BindingKey.0=test
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "3e0dff9d-9ed5-47c3-beb2-a42c1d69e1cc"
}
}
```

5. Developer Resources

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go

- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation	Operation failed.
FailedOperation.TryLater	The operation is in progress. Please try again later.
InvalidParameter	Invalid parameter.
InvalidParameterValue.NoTaskId	The task ID does not exist.
LimitExceeded	The quota limit is exceeded.
ResourceInUse	The resource is in use.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	The resource is unavailable.
ResourcesSoldOut	The resources have been sold out.
UnauthorizedOperation	Unauthorized operation.

DescribeSubscriptionDetail

Last updated：2024-11-27 16:09:19

1. API Description

Domain name for API request: cmq.intl.tencentcloudapi.com.

This API is used to query subscription details.

A maximum of 20 requests can be initiated per second for this API.

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: DescribeSubscriptionDetail.
Version	Yes	String	Common Params. The value used for this API: 2019-03-04.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product.
TopicName	Yes	String	Topic name, which is unique under the same account in an individual region. It is a string of up to 64 characters, which must begin with a letter and can contain letters, digits, and dashes (`-`).
Offset	No	Integer	Starting position of topic list to be returned on the current page in case of paginated return. If a value is entered, `limit` is required. If this parameter is left empty, 0 will be used by default
Limit	No	Integer	Number of topics to be returned per page in case of paginated return. If this parameter is not passed in, 20 will be used by default. Maximum

			value: 50.
Filters.N	No	Array of Filter	Filter parameter. Currently, only filtering by `SubscriptionName` is supported, and only one keyword is allowed.

3. Output Parameters

Parameter Name	Type	Description
TotalCount	Integer	Total number
SubscriptionSet	Array of Subscription	Subscription attribute set Note: this field may return null, indicating that no valid values can be obtained.
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Querying subscription details

Input Example

```
https://cmq.intl.tencentcloudapi.com/?Action=DescribeSubscriptionDetail
&TopicName=ConnTopic
&<Common request parameters>
```

Output Example

```
{
"Response": {
"TotalCount": 0,
"SubscriptionSet": [],
"RequestId": "c67d0d56-2406-49b2-8e82-320b9beab46c"
}
}
```

5. Developer Resources

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation	Operation failed.
FailedOperation.TryLater	The operation is in progress. Please try again later.
InvalidParameter	Invalid parameter.
InvalidParameterValue.NoTaskId	The task ID does not exist.
LimitExceeded	The quota limit is exceeded.
ResourceInUse	The resource is in use.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	The resource is unavailable.
ResourcesSoldOut	The resources have been sold out.

UnauthorizedOperation	Unauthorized operation.

DeleteSubscribe

Last updated：2024-11-27 16:09:20

1. API Description

Domain name for API request: cmq.intl.tencentcloudapi.com.

This API is used to delete a subscription.

A maximum of 20 requests can be initiated per second for this API.

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: DeleteSubscribe.
Version	Yes	String	Common Params. The value used for this API: 2019-03-04.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product.
TopicName	Yes	String	Topic name, which is unique under the same account in an individual region. It is a string of up to 64 characters, which must begin with a letter and can contain letters, digits, and dashes (-).
SubscriptionName	Yes	String	Subscription name, which is unique in the same topic under the same account in an individual region. It is a string of up to 64 characters, which must begin with a letter and can contain letters, digits, and dashes (-).

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Deleting subscription

Input Example

```
https://cmq.intl.tencentcloudapi.com/?Action=DeleteSubscribe
&TopicName=ConnTopic&SubscriptionName=york
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "f8efab10-0ac6-44cf-8d09-67bb228deaf4"
}
}
```

5. Developer Resources

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js

- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation	Operation failed.
FailedOperation.TryLater	The operation is in progress. Please try again later.
InvalidParameter	Invalid parameter.
LimitExceeded	The quota limit is exceeded.
ResourceInUse	The resource is in use.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	The resource is unavailable.
ResourcesSoldOut	The resources have been sold out.
UnauthorizedOperation	Unauthorized operation.

CreateSubscribe

Last updated：2024-11-27 16:09:21

1. API Description

Domain name for API request: cmq.intl.tencentcloudapi.com.

This API is used to create a subscription.

A maximum of 20 requests can be initiated per second for this API.

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: CreateSubscribe.
Version	Yes	String	Common Params. The value used for this API: 2019-03-04.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product.
TopicName	Yes	String	Topic name, which is unique under the same account in an individual region. It is a string of up to 64 characters, which must begin with a letter and can contain letters, digits, and dashes (-).
SubscriptionName	Yes	String	Subscription name, which is unique in the same topic under the same account in an individual region. It is a string of up to 64 characters, which must begin with a letter and can contain letters, digits, and dashes (-).
Protocol	Yes	String	Subscription protocol. Currently, two protocols are supported:

			http and queue. To use the `http` protocol, you need to build your own web server to receive messages. With the `queue` protocol, messages are automatically pushed to a CMQ queue and you can pull them concurrently.
Endpoint	Yes	String	`Endpoint` for notification receipt, which is distinguished by `Protocol` . For `http` , `Endpoint` must begin with `http://` and `host` can be a domain name or IP. For `Queue` , enter `QueueName` . Please note that currently the push service cannot push messages to a VPC; therefore, if a VPC domain name or address is entered for `Endpoint` , pushed messages will not be received. Currently, messages can be pushed only to the public network and basic network.
NotifyStrategy	No	String	CMQ push server retry policy in case an error occurs while pushing a message to `Endpoint` . Valid values: 1. BACKOFF_RETRY: backoff retry, which is to retry at a fixed interval, discard the message after a certain number of retries, and continue to push the next message; 2. EXPONENTIAL_DECAY_RETRY: exponential decay retry, which is to retry at an exponentially increasing interval, such as 1s, 2s, 4s, 8s, and so on. As a message can be retained in a topic for one day, failed messages will be discarded at most after one day of retry. Default value: EXPONENTIAL_DECAY_RETRY.
FilterTag.N	No	Array of String	Message body tag (used for message filtering). The number of tags cannot exceed 5, and each tag can contain up to 16 characters. It is used in conjunction with the `MsgTag` parameter of `(Batch)PublishMessage` . Rules: 1. If `FilterTag` is not set, no matter whether `MsgTag` is set, the subscription will receive all messages published to the topic; 2. If the `FilterTag` array has a value, only when at least one of the values in the array also exists in the `MsgTag` array (i.e., `FilterTag` and `MsgTag` have an intersection) can the subscription receive messages published to the topic; 3. If the `FilterTag` array has a value, but `MsgTag` is not set, then no message published to the topic will be received, which can be considered as a special case of rule 2 as `FilterTag` and `MsgTag` do not intersect in this case. The overall design idea of rules is based on the intention of the subscriber.
BindingKey.N	No	Array of	The number of `BindingKey` cannot exceed 5, and the length of each `BindingKey` cannot exceed 64 bytes. This

		String	field indicates the filtering policy for subscribing to and receiving messages. Each `BindingKey` can contain up to 15 `.` , i.e., up to 16 phrases.
NotifyContentFormat	No	String	Push content format. Valid values: 1. JSON, 2. SIMPLIFIED, i.e., the raw format. If `Protocol` is `queue` , this value must be `SIMPLIFIED` . If `Protocol` is `http` , both options are acceptable, and the default value is `JSON` .

3. Output Parameters

Parameter Name	Type	Description
SubscriptionId	String	SubscriptionId
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Creating subscription

Input Example

```
https://cmq.intl.tencentcloudapi.com/?Action=CreateSubscribe
&TopicName=ConnTopic&SubscriptionName=ConnSubQueue&Protocol=queue&Endpoint=queue_
sub
&<Common request parameters>
```

Output Example

```
{
"Response": {
"SubscriptionId": "subsc-39gyuuhd",
"RequestId": "1620b635-6071-47c7-ac1e-975afe5104a7"
}
}
```

5. Developer Resources

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation	Operation failed.
FailedOperation.TryLater	The operation is in progress. Please try again later.
InvalidParameter	Invalid parameter.
InvalidParameterValue.NoTaskId	The task ID does not exist.
LimitExceeded	The quota limit is exceeded.
ResourceInUse	The resource is in use.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	The resource is unavailable.
ResourcesSoldOut	The resources have been sold out.

UnauthorizedOperation	Unauthorized operation.

ClearSubscriptionFilterTags

Last updated：2024-11-27 16:09:22

1. API Description

Domain name for API request: cmq.intl.tencentcloudapi.com.

This API is used to clear the message tags of a subscriber.

A maximum of 20 requests can be initiated per second for this API.

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: ClearSubscriptionFilterTags.
Version	Yes	String	Common Params. The value used for this API: 2019-03-04.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product.
TopicName	Yes	String	Topic name, which is unique under the same account in an individual region. It is a string of up to 64 characters, which must begin with a letter and can contain letters, digits, and dashes (-).
SubscriptionName	Yes	String	Subscription name, which is unique in the same topic under the same account in an individual region. It is a string of up to 64 characters, which must begin with a letter and can contain letters, digits, and dashes (-).

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Clearing the message tags of subscriber

Input Example

```
https://cmq.intl.tencentcloudapi.com/?Action=ClearSubscriptionFilterTags
&TopicName=ConnTopic&SubscriptionName=Queue
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "3e0dff9d-9ed5-47c3-beb2-a42c1d69e1cc"
}
}
```

5. Developer Resources

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js

- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

There is no error code related to the API business logic. For other error codes, please see Common Error Codes.

Queue APIs

UnbindDeadLetter

Last updated：2024-11-27 16:09:23

1. API Description

Domain name for API request: cmq.intl.tencentcloudapi.com.

This API is used to unbind a dead letter queue.

A maximum of 20 requests can be initiated per second for this API.

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: UnbindDeadLetter.
Version	Yes	String	Common Params. The value used for this API: 2019-03-04.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product.
QueueName	Yes	String	Source queue name of dead letter policy. Calling this API will clear the dead letter queue policy of this queue.

3. Output Parameters

Parameter	Type	Description

Name		
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Unbinding dead letter queue

This example shows you how to unbind a dead letter queue.

Input Example

```
https://cmq.intl.tencentcloudapi.com/?Action=UnbindDeadLetter
&QueueName=test
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "3e0dff9d-9ed5-47c3-beb2-a42c1d69e1cc"
}
}
```

5. Developer Resources

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation	Operation failed.
FailedOperation.TryLater	The operation is in progress. Please try again later.
InvalidParameter	Invalid parameter.
InvalidParameterValue.NoTaskId	The task ID does not exist.
LimitExceeded	The quota limit is exceeded.
ResourceInUse	The resource is in use.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	The resource is unavailable.
ResourcesSoldOut	The resources have been sold out.
UnauthorizedOperation	Unauthorized operation.

RewindQueue

Last updated：2024-11-27 16:09:24

1. API Description

Domain name for API request: cmq.intl.tencentcloudapi.com.

This API is used to rewind a queue.

A maximum of 20 requests can be initiated per second for this API.

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: RewindQueue.
Version	Yes	String	Common Params. The value used for this API: 2019-03-04.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product.
QueueName	Yes	String	Queue name, which is unique under the same account in an individual region. It is a string of up to 64 characters, which must begin with a letter and can contain letters, digits, and dashes (`-`).
StartConsumeTime	Yes	Integer	After this time is set, the `(Batch)receiveMessage` API will consume the messages received after this timestamp in the order in which they are produced.

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Rewinding consumption position in queue

This example shows you how to rewind the consumption position in a queue.

Input Example

```
https://cmq.intl.tencentcloudapi.com/?Action=RewindQueue
&QueueName=test&StartConsumeTime=1582108595
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "3e0dff9d-9ed5-47c3-beb2-a42c1d69e1cc"
}
}
```

5. Developer Resources

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET

- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

There is no error code related to the API business logic. For other error codes, please see Common Error Codes.

ModifyQueueAttribute

Last updated：2024-11-27 16:09:25

1. API Description

Domain name for API request: cmq.intl.tencentcloudapi.com.

This API is used to modify queue attributes.

A maximum of 20 requests can be initiated per second for this API.

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: ModifyQueueAttribute.
Version	Yes	String	Common Params. The value used for this API: 2019-03-04.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product.
QueueName	Yes	String	Queue name, which is unique under the same account in an individual region. It is a string of up to 64 characters, which must begin with a letter and can contain letters, digits, and dashes (-).
MaxMsgHeapNum	No	Integer	Maximum number of heaped messages. The value range is 1,000,000-10,000,000 during the beta test and can be 1,000,000-1,000,000,000 after the product is officially released. The default value is 10,000,000 during the beta

			test and will be 100,000,000 after the product is officially released.
PollingWaitSeconds	No	Integer	Long polling wait time for message reception. Value range: 0-30 seconds. Default value: 0.
VisibilityTimeout	No	Integer	Message visibility timeout period. Value range: 1-43200 seconds (i.e., 12 hours). Default value: 30.
MaxMsgSize	No	Integer	Maximum message length. Value range: 1024-65536 bytes (i.e., 1-64 KB). Default value: 65536.
MsgRetentionSeconds	No	Integer	Message retention period. Value range: 60-1296000 seconds (i.e., 1 minute-15 days). Default value: 345600 (i.e., 4 days).
RewindSeconds	No	Integer	Maximum message rewindable period. Value range: 0-msgRetentionSeconds (maximum message retention period of a queue). 0 means not to enable message rewinding.
FirstQueryInterval	No	Integer	First query time
MaxQueryCount	No	Integer	Maximum number of queries
DeadLetterQueueName	No	String	Dead letter queue name
MaxTimeToLive	No	Integer	Maximum period in seconds before an unconsumed message expires, which is required if `MaxTimeToLivepolicy` is 1. Value range: 300-43200. This value should be smaller than `MsgRetentionSeconds` (maximum message retention period)
MaxReceiveCount	No	Integer	Maximum number of receipts
Policy	No	Integer	Dead letter queue policy
Trace	No	Boolean	Whether to enable message trace. true: yes, false: no. If this field is left empty, the feature will not be enabled.

3. Output Parameters

Parameter Name	Type	Description

RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Modifying queue attributes

This example shows you how to modify queue attributes.

Input Example

```
https://cmq.intl.tencentcloudapi.com/?Action=ModifyQueueAttribute
&QueueName=test&MaxMsgSize=1024
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "3e0dff9d-9ed5-47c3-beb2-a42c1d69e1cc"
}
}
```

5. Developer Resources

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation	Operation failed.
FailedOperation.TryLater	The operation is in progress. Please try again later.
InvalidParameter	Invalid parameter.
InvalidParameterValue.NoTaskId	The task ID does not exist.
LimitExceeded	The quota limit is exceeded.
ResourceInUse	The resource is in use.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	The resource is unavailable.
ResourcesSoldOut	The resources have been sold out.
UnauthorizedOperation	Unauthorized operation.

DescribeQueueDetail

Last updated：2024-11-27 16:09:26

1. API Description

Domain name for API request: cmq.intl.tencentcloudapi.com.

This API is used to enumerate queues.

A maximum of 20 requests can be initiated per second for this API.

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: DescribeQueueDetail.
Version	Yes	String	Common Params. The value used for this API: 2019-03-04.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product.
Offset	No	Integer	Starting position of queue list to be returned on the current page in case of paginated return. If a value is entered, `limit` is required. If this parameter is left empty, 0 will be used by default
Limit	No	Integer	Number of queues to be returned per page in case of paginated return. If this parameter is not passed in, 20 will be used by default. Maximum value: 50.
Filters.N	No	Array of Filter	Filter parameter. Currently, filtering by `QueueName` is supported, and only one keyword is allowed

TagKey	No	String	Tag search
QueueName	No	String	Exact match by `QueueName`

3. Output Parameters

Parameter Name	Type	Description
TotalCount	Integer	Total number of queues
QueueSet	Array of QueueSet	Queue list
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Getting queue list

Input Example

```
https://cmq.intl.tencentcloudapi.com/?Action=DescribeQueueDetail
&<Common request parameters>
```

Output Example

```
{
"Response": {
"TotalCount": 4,
"QueueSet": [
{
"Tags": [],
"QueueId": "queue-kc7m75to",
"QueueName": "test",
"CreateUin": 20548499,
"Qps": 5000,
"Bps": 52428800,
"MaxDelaySeconds": 3600,
"MaxMsgHeapNum": 100000000,
"PollingWaitSeconds": 0,
"MsgRetentionSeconds": 345600,
"VisibilityTimeout": 30,
"MaxMsgSize": 65536,
"RewindSeconds": 0,
"CreateTime": 1581471003,
"LastModifyTime": 1581471003,
"Transaction": null,
"DeadLetterSource": [],
"DeadLetterPolicy": {
"DeadLetterQueue": "queue-0v0y40j4",
"DeadLetterQueueName": "test123",
"Policy": 0,
"MaxReceiveCount": 1,
"MaxTimeToLive": null
},
"TransactionPolicy": null,
"ActiveMsgNum": 0,
"InactiveMsgNum": 0,
"DelayMsgNum": 0,
"RewindMsgNum": 0,
"MinMsgTime": 1582015467
},
{
"Tags": [],
"QueueId": "queue-0v0y40j4",
"QueueName": "test123",
"CreateUin": 20548499,
"Qps": 5000,
"Bps": 52428800,
"MaxDelaySeconds": 3600,
"MaxMsgHeapNum": 23232323,
"PollingWaitSeconds": 3,
"MsgRetentionSeconds": 232323,
"VisibilityTimeout": 12,
"MaxMsgSize": 121223,
"RewindSeconds": 0,
"CreateTime": 1581412802,
"LastModifyTime": 1581471003,
"Transaction": null,
"DeadLetterSource": [
{
"QueueId": "queue-kc7m75to",
"QueueName": "test"
}
],
"DeadLetterPolicy": null,
"TransactionPolicy": null,
"ActiveMsgNum": 0,
"InactiveMsgNum": 0,
"DelayMsgNum": 0,
"RewindMsgNum": 0,
"MinMsgTime": 1582015467
},
{
"Tags": [],
"QueueId": "queue-dvukxexc",
"QueueName": "dead",
"CreateUin": 20548499,
"Qps": 5000,
"Bps": 52428800,
"MaxDelaySeconds": 3600,
"MaxMsgHeapNum": 100000000,
"PollingWaitSeconds": 0,
"MsgRetentionSeconds": 345600,
"VisibilityTimeout": 30,
"MaxMsgSize": 65536,
"RewindSeconds": 0,
"CreateTime": 1581418590,
"LastModifyTime": 1581418590,
"Transaction": null,
"DeadLetterSource": [],
"DeadLetterPolicy": null,
"TransactionPolicy": null,
"ActiveMsgNum": 0,
"InactiveMsgNum": 0,
"DelayMsgNum": 0,
"RewindMsgNum": 0,
"MinMsgTime": 1582015467
},
{
"Tags": [],
"QueueId": "queue-cdb91yt6",
"QueueName": "test21",
"CreateUin": 20548499,
"Qps": 5000,
"Bps": 52428800,
"MaxDelaySeconds": 3600,
"MaxMsgHeapNum": 100000000,
"PollingWaitSeconds": 0,
"MsgRetentionSeconds": 345600,
"VisibilityTimeout": 30,
"MaxMsgSize": 65536,
"RewindSeconds": 0,
"CreateTime": 1581413384,
"LastModifyTime": 1581413384,
"Transaction": null,
"DeadLetterSource": [],
"DeadLetterPolicy": null,
"TransactionPolicy": null,
"ActiveMsgNum": 0,
"InactiveMsgNum": 0,
"DelayMsgNum": 0,
"RewindMsgNum": 0,
"MinMsgTime": 1582015467
}
],
"RequestId": "3f9a8b31-ac0d-4abd-8af2-e8564ceac30a"
}
}
```

5. Developer Resources

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation	Operation failed.
FailedOperation.TryLater	The operation is in progress. Please try again later.
InvalidParameter	Invalid parameter.
LimitExceeded	The quota limit is exceeded.
ResourceInUse	The resource is in use.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	The resource is unavailable.
ResourcesSoldOut	The resources have been sold out.
UnauthorizedOperation	Unauthorized operation.

DescribeDeadLetterSourceQueues

Last updated：2024-11-27 16:09:28

1. API Description

Domain name for API request: cmq.intl.tencentcloudapi.com.

This API is used to enumerate the source queues of a dead letter queue.

A maximum of 20 requests can be initiated per second for this API.

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: DescribeDeadLetterSourceQueues.
Version	Yes	String	Common Params. The value used for this API: 2019-03-04.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product.
DeadLetterQueueName	Yes	String	Dead letter queue name
Limit	No	Integer	Starting position of topic list to be returned on the current page in case of paginated return. If a value is entered, `limit` is required. If this parameter is left empty, 0 will be used by default.
Offset	No	Integer	Number of topics to be returned per page in case of paginated return. If this parameter is not passed in, 20 will be used by default. Maximum value: 50.

Filters.N	No	Array of Filter	Filters source queue name of dead letter queue. Currently, only filtering by `SourceQueueName` is supported

3. Output Parameters

Parameter Name	Type	Description
TotalCount	Integer	Number of eligible queues
QueueSet	Array of DeadLetterSource	Source queues of dead letter queue
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Enumerating the source queues of dead letter queue

This example shows you how to enumerate the source queues of a dead letter queue.

Input Example

```
https://cmq.intl.tencentcloudapi.com/?Action=DescribeDeadLetterSourceQueues
&DeadLetterQueueName=test123
&<Common request parameters>
```

Output Example

```
{
"Response": {
"QueueSet": [
{
"QueueName": "test",
"QueueId": "queue-kc7m75to"
}
],
"TotalCount": 1,
"RequestId": "83fce81a-7305-49b4-a154-983dd76204da"
}
}
```

5. Developer Resources

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

There is no error code related to the API business logic. For other error codes, please see Common Error Codes.

DeleteQueue

Last updated：2024-11-27 16:09:29

1. API Description

Domain name for API request: cmq.intl.tencentcloudapi.com.

This API is used to delete a queue.

A maximum of 20 requests can be initiated per second for this API.

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: DeleteQueue.
Version	Yes	String	Common Params. The value used for this API: 2019-03-04.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product.
QueueName	Yes	String	Queue name, which is unique under the same account in an individual region. It is a string of up to 64 characters, which must begin with a letter and can contain letters, digits, and dashes (`-`).

3. Output Parameters

Parameter Name	Type	Description

RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Deleting queue

This example shows you how to delete a queue.

Input Example

```
https://cmq.intl.tencentcloudapi.com/?Action=DeleteQueue
&QueueName=test
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "3e0dff9d-9ed5-47c3-beb2-a42c1d69e1cc"
}
}
```

5. Developer Resources

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation	Operation failed.
InvalidParameter	Invalid parameter.
ResourceInUse	The resource is in use.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	The resource is unavailable.
UnauthorizedOperation	Unauthorized operation.

CreateQueue

Last updated：2024-11-27 16:09:30

1. API Description

Domain name for API request: cmq.intl.tencentcloudapi.com.

This API is used to create a queue.

A maximum of 20 requests can be initiated per second for this API.

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: CreateQueue.
Version	Yes	String	Common Params. The value used for this API: 2019-03-04.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product.
QueueName	Yes	String	Queue name, which is unique under the same account in an individual region. It is a string of up to 64 characters, which must begin with a letter and can contain letters, digits, and dashes (-).
MaxMsgHeapNum	No	Integer	Maximum number of heaped messages. The value range is 1,000,000-10,000,000 during the beta test and can be 1,000,000-1,000,000,000 after the product is officially released. The default value is 10,000,000 during the beta

			test and will be 100,000,000 after the product is officially released.
PollingWaitSeconds	No	Integer	Long polling wait time for message reception. Value range: 0-30 seconds. Default value: 0.
VisibilityTimeout	No	Integer	Message visibility timeout period. Value range: 1-43200 seconds (i.e., 12 hours). Default value: 30.
MaxMsgSize	No	Integer	Maximum message length. Value range: 1024-65536 bytes (i.e., 1-64 KB). Default value: 65536.
MsgRetentionSeconds	No	Integer	Message retention period. Value range: 60-1296000 seconds (i.e., 1 minute-15 days). Default value: 345600 (i.e., 4 days).
RewindSeconds	No	Integer	Whether to enable the message rewinding feature for a queue. Value range: 0-msgRetentionSeconds, where 0 means not to enable this feature, while `msgRetentionSeconds` indicates that the maximum rewindable period is the message retention period of the queue.
Transaction	No	Integer	1: transaction queue, 0: general queue
FirstQueryInterval	No	Integer	First lookback interval
MaxQueryCount	No	Integer	Maximum number of lookbacks
DeadLetterQueueName	No	String	Dead letter queue name
Policy	No	Integer	Dead letter policy. 0: message has been consumed multiple times but not deleted, 1: `Time-To-Live` has elapsed
MaxReceiveCount	No	Integer	Maximum receipt times. Value range: 1-1000
MaxTimeToLive	No	Integer	Maximum period in seconds before an unconsumed message expires, which is required if `policy` is 1. Value range: 300-43200. This value should be smaller than `msgRetentionSeconds` (maximum message retention period)
Trace	No	Boolean	Whether to enable message trace. true: yes, false: no. If this field is not set, the feature will not be enabled

3. Output Parameters

Parameter Name	Type	Description
QueueId	String	"queueId" of a successfully created queue
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Creating queue

This example shows you how to create a queue.

Input Example

```
https://cmq.intl.tencentcloudapi.com/?Action=CreateQueue
&QueueName=test
&<Common request parameters>
```

Output Example

```
{
"Response": {
"QueueId": "queue-ges05csc",
"RequestId": "3e0dff9d-9ed5-47c3-beb2-a42c1d69e1cc"
}
}
```

5. Developer Resources

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java

- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation	Operation failed.
FailedOperation.TryLater	The operation is in progress. Please try again later.
InvalidParameter	Invalid parameter.
LimitExceeded	The quota limit is exceeded.
ResourcesSoldOut	The resources have been sold out.
UnauthorizedOperation	Unauthorized operation.

ClearQueue

Last updated：2024-11-27 16:09:31

1. API Description

Domain name for API request: cmq.intl.tencentcloudapi.com.

This API is used to clear all messages in a queue.

A maximum of 20 requests can be initiated per second for this API.

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: ClearQueue.
Version	Yes	String	Common Params. The value used for this API: 2019-03-04.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product.
QueueName	Yes	String	Queue name, which is unique under the same account in an individual region. It is a string of up to 64 characters, which must begin with a letter and can contain letters, digits, and dashes (-).

3. Output Parameters

Parameter Name	Type	Description

RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Clearing heaped messages in queue

This example shows you how to clear heaped messages in a queue.

Input Example

```
https://cmq.intl.tencentcloudapi.com/?Action=ClearQueue
&QueueName=test
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "3e0dff9d-9ed5-47c3-beb2-a42c1d69e1cc"
}
}
```

5. Developer Resources

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation	Operation failed.
FailedOperation.TryLater	The operation is in progress. Please try again later.
InvalidParameter	Invalid parameter.
InvalidParameterValue.NoTaskId	The task ID does not exist.
LimitExceeded	The quota limit is exceeded.
ResourceInUse	The resource is in use.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	The resource is unavailable.
ResourcesSoldOut	The resources have been sold out.
UnauthorizedOperation	Unauthorized operation.

Introduction

Last updated：2020-06-19 15:17:54

Welcome to Cloud Message Queue (CMQ).

Tencent Cloud CMQ is a distributed message queuing service used for storing messages transmitted between processes. It is designed to provide reliable message-oriented asynchronous communications service between distributed applications or between different components of the same application. Messages are stored in highly reliable and available queues, which allows multiple processes to read and write simultaneously without interacting with each other. Tencent Cloud CMQ allows messages to be transmitted, with no data loss, between distributed components of the applications that run different tasks. Components are not required to always remain available.

A queue acts as a buffer between a data sender and a data receiver, eliminating issues caused by circumstances in which the data sender works faster than the data receiver or the data sender or receiver only connects to the network intermittently.

In the traditional model of process communications, a client requests services from a server and waits for the server's responses. However, this model has many disadvantages. For example, when network conditions are poor, the client's request may be lost; or when the server takes too long to process a request, the request might fail due to timeout.

To resolve these issues, Tencent Cloud launched the CMQ service for message distribution and management. With Tencent Cloud CMQ, you can decouple components of an application. Each component can run independently, and communications between the components can be simplified. Any component of a distributed application can store messages in queues, and Tencent Cloud CMQ ensures that each message is transmitted at least once and can be read and written many times. A single queue can be simultaneously used by multiple components of a distributed application, which are not required to collaborate with each other. All components can programmatically retrieve and operate messages using CMQ APIs.

⚠ **Note：**

All CMQ APIs in this section have been upgraded to API 3.0. Future CMQfeatures will also be added here. We recommend using API 3.0.
Legacy APIs remain available, but will not be updated. For more information, please see CMQ API Overview (legacy).

Data Types

Last updated：2024-11-27 16:09:32

DeadLetterPolicy

DeadLetterPolicy

Used by actions: DescribeQueueDetail.

Name	Type	Description
DeadLetterQueueName	String	DeadLetterQueueName Note: this field may return null, indicating that no valid values can be obtained.
DeadLetterQueue	String	DeadLetterQueue Note: this field may return null, indicating that no valid values can be obtained.
Policy	Integer	Policy Note: this field may return null, indicating that no valid values can be obtained.
MaxTimeToLive	Integer	MaxTimeToLive Note: this field may return null, indicating that no valid values can be obtained.
MaxReceiveCount	Integer	MaxReceiveCount Note: this field may return null, indicating that no valid values can be obtained.

DeadLetterSource

DeadLetterSource

Used by actions: DescribeDeadLetterSourceQueues, DescribeQueueDetail.

Name	Type	Description
QueueId	String	QueueId Note: this field may return null, indicating that no valid values can be obtained.

QueueName	String	QueueName Note: this field may return null, indicating that no valid values can be obtained.

Filter

Filter parameter

Used by actions: DescribeDeadLetterSourceQueues, DescribeQueueDetail, DescribeSubscriptionDetail, DescribeTopicDetail.

Name	Type	Required	Description
Name	String	No	Filter parameter name
Values	Array of String	No	Value

QueueSet

Batch queue attribute information

Used by actions: DescribeQueueDetail.

Name	Type	Description
QueueId	String	QueueId
QueueName	String	QueueName
Qps	Integer	Qps Note: this field may return null, indicating that no valid values can be obtained.
Bps	Integer	Bps Note: this field may return null, indicating that no valid values can be obtained.
MaxDelaySeconds	Integer	MaxDelaySeconds Note: this field may return null, indicating that no valid values can be obtained.
MaxMsgHeapNum	Integer	MaxMsgHeapNum Note: this field may return null, indicating that no valid values can be obtained.

PollingWaitSeconds	Integer	PollingWaitSeconds Note: this field may return null, indicating that no valid values can be obtained.
MsgRetentionSeconds	Integer	MsgRetentionSeconds Note: this field may return null, indicating that no valid values can be obtained.
VisibilityTimeout	Integer	VisibilityTimeout Note: this field may return null, indicating that no valid values can be obtained.
MaxMsgSize	Integer	MaxMsgSize Note: this field may return null, indicating that no valid values can be obtained.
RewindSeconds	Integer	RewindSeconds Note: this field may return null, indicating that no valid values can be obtained.
CreateTime	Integer	CreateTime Note: this field may return null, indicating that no valid values can be obtained.
LastModifyTime	Integer	LastModifyTime Note: this field may return null, indicating that no valid values can be obtained.
ActiveMsgNum	Integer	ActiveMsgNum Note: this field may return null, indicating that no valid values can be obtained.
InactiveMsgNum	Integer	InactiveMsgNum Note: this field may return null, indicating that no valid values can be obtained.
DelayMsgNum	Integer	DelayMsgNum Note: this field may return null, indicating that no valid values can be obtained.
RewindMsgNum	Integer	RewindMsgNum Note: this field may return null, indicating that no valid values can be obtained.
MinMsgTime	Integer	MinMsgTime Note: this field may return null, indicating that no valid values can be obtained.

Transaction	Boolean	Transaction Note: this field may return null, indicating that no valid values can be obtained.
DeadLetterSource	Array of DeadLetterSource	DeadLetterSource Note: this field may return null, indicating that no valid values can be obtained.
DeadLetterPolicy	DeadLetterPolicy	DeadLetterPolicy Note: this field may return null, indicating that no valid values can be obtained.
TransactionPolicy	TransactionPolicy	TransactionPolicy Note: this field may return null, indicating that no valid values can be obtained.
CreateUin	Integer	Creator `uin` Note: this field may return null, indicating that no valid values can be obtained.
Tags	Array of Tag	Tag Note: this field may return null, indicating that no valid values can be obtained.
Trace	Boolean	Message trace flag. true: enabled, false: not enabled Note: this field may return null, indicating that no valid values can be obtained.

Subscription

Subscription response parameter

Used by actions: DescribeSubscriptionDetail.

Name	Type	Description
SubscriptionName	String	SubscriptionName Note: this field may return null, indicating that no valid values can be obtained.
SubscriptionId	String	SubscriptionId Note: this field may return null, indicating that no valid values can be obtained.
TopicOwner	Integer	TopicOwner

		Note: this field may return null, indicating that no valid values can be obtained.
MsgCount	Integer	MsgCount Note: this field may return null, indicating that no valid values can be obtained.
LastModifyTime	Integer	LastModifyTime Note: this field may return null, indicating that no valid values can be obtained.
CreateTime	Integer	CreateTime Note: this field may return null, indicating that no valid values can be obtained.
BindingKey	Array of String	BindingKey Note: this field may return null, indicating that no valid values can be obtained.
Endpoint	String	Endpoint Note: this field may return null, indicating that no valid values can be obtained.
FilterTags	Array of String	FilterTags Note: this field may return null, indicating that no valid values can be obtained.
Protocol	String	Protocol Note: this field may return null, indicating that no valid values can be obtained.
NotifyStrategy	String	NotifyStrategy Note: this field may return null, indicating that no valid values can be obtained.
NotifyContentFormat	String	NotifyContentFormat Note: this field may return null, indicating that no valid values can be obtained.

Tag

Tag

Used by actions: DescribeQueueDetail, DescribeTopicDetail.

Name	Type	Description
TagKey	String	Tag key Note: this field may return null, indicating that no valid values can be obtained.
TagValue	String	Tag value Note: this field may return null, indicating that no valid values can be obtained.

TopicSet

Field for displaying returned topic information

Used by actions: DescribeTopicDetail.

Name	Type	Description
TopicId	String	TopicId Note: this field may return null, indicating that no valid values can be obtained.
TopicName	String	TopicName Note: this field may return null, indicating that no valid values can be obtained.
MsgRetentionSeconds	Integer	MsgRetentionSeconds Note: this field may return null, indicating that no valid values can be obtained.
MaxMsgSize	Integer	MaxMsgSize Note: this field may return null, indicating that no valid values can be obtained.
Qps	Integer	Qps Note: this field may return null, indicating that no valid values can be obtained.
FilterType	Integer	FilterType Note: this field may return null, indicating that no valid values can be obtained.
CreateTime	Integer	CreateTime Note: this field may return null, indicating that no valid values can be obtained.
LastModifyTime	Integer	LastModifyTime Note: this field may return null, indicating that no valid values can be

		obtained.
MsgCount	Integer	MsgCount Note: this field may return null, indicating that no valid values can be obtained.
CreateUin	Integer	CreateUin Note: this field may return null, indicating that no valid values can be obtained.
Tags	Array of Tag	Tags Note: this field may return null, indicating that no valid values can be obtained.
Trace	Boolean	Whether to enable message trace for a topic. true: yes, false: no Note: this field may return null, indicating that no valid values can be obtained.

TransactionPolicy

TransactionPolicy

Used by actions: DescribeQueueDetail.

Name	Type	Description
FirstQueryInterval	Integer	FirstQueryInterval Note: this field may return null, indicating that no valid values can be obtained.
MaxQueryCount	Integer	MaxQueryCount Note: this field may return null, indicating that no valid values can be obtained.

Error Codes

Last updated：2024-11-27 16:09:32

Feature Description

If there is an Error field in the response, it means that the API call failed. For example:

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

Code in Error indicates the error code, and Message indicates the specific information of the error.

Error Code List

Common Error Codes

Error Code	Description
ActionOffline	This API has been deprecated.
AuthFailure.InvalidAuthorization	`Authorization` in the request header is invalid.
AuthFailure.InvalidSecretId	Invalid key (not a TencentCloud API key type).
AuthFailure.MFAFailure	MFA failed.
AuthFailure.SecretIdNotFound	Key does not exist. Check if the key has been deleted or disabled in the console, and if not, check if the key is correctly entered. Note that whitespaces should not exist before or after the key.
AuthFailure.SignatureExpire	Signature expired. Timestamp and server time cannot differ by more than five minutes. Please

	ensure your current local time matches the standard time.
AuthFailure.SignatureFailure	Invalid signature. Signature calculation error. Please ensure you’ve followed the signature calculation process described in the Signature API documentation.
AuthFailure.TokenFailure	Token error.
AuthFailure.UnauthorizedOperation	The request is not authorized. For more information, see the CAM documentation.
DryRunOperation	DryRun Operation. It means that the request would have succeeded, but the DryRun parameter was used.
FailedOperation	Operation failed.
InternalError	Internal error.
InvalidAction	The API does not exist.
InvalidParameter	Incorrect parameter.
InvalidParameterValue	Invalid parameter value.
InvalidRequest	The multipart format of the request body is incorrect.
IpInBlacklist	Your IP is in uin IP blacklist.
IpNotInWhitelist	Your IP is not in uin IP whitelist.
LimitExceeded	Quota limit exceeded.
MissingParameter	A parameter is missing.
NoSuchProduct	The product does not exist.
NoSuchVersion	The API version does not exist.
RequestLimitExceeded	The number of requests exceeds the frequency limit.
RequestLimitExceeded.GlobalRegionUinLimitExceeded	Uin exceeds the frequency limit.
RequestLimitExceeded.IPLimitExceeded	The number of ip requests exceeds the frequency limit.
RequestLimitExceeded.UinLimitExceeded	The number of uin requests exceeds the frequency

	limit.
RequestSizeLimitExceeded	The request size exceeds the upper limit.
ResourceInUse	Resource is in use.
ResourceInsufficient	Insufficient resource.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	Resource is unavailable.
ResponseSizeLimitExceeded	The response size exceeds the upper limit.
ServiceUnavailable	Service is unavailable now.
UnauthorizedOperation	Unauthorized operation.
UnknownParameter	Unknown parameter.
UnsupportedOperation	Unsupported operation.
UnsupportedProtocol	HTTP(S) request protocol error; only GET and POST requests are supported.
UnsupportedRegion	API does not support the requested region.

Service Error Codes

Error Code	Description
FailedOperation.TryLater	The operation is in progress. Please try again later.
InvalidParameterValue.NoTaskId	The task ID does not exist.
ResourcesSoldOut	The resources have been sold out.

API 2017

Note

Last updated：2024-12-18 17:34:59

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Welcome to Tencent Cloud message queue CMQ (Cloud Message Queue).

Tencent Cloud message queuing (Cloud Message Queue, (hereinafter referred to as CMQ) is a distributed message queuing service that stores messages transmitted between processes and provides reliable message-based asynchronous communication services between different distributed applications or between different components of an application. Messages are stored in highly reliable and highly available message queues, and multiple processes can read and write at the same time without interfering with each other. With Tencent Cloud CMQ, users can pass information between distributed components executing different job's applications without losing messages or requiring that each component is always available.

The queue acts as a buffer between the data sender and the data receiver. Queues can solve the problems when the working speed of the data sender is faster than that of the data receiver, or when the data sender or data receiver is only intermittently connected to the network.

In the traditional process communication mode, the client requests the service of the server and waits for the response of the server. However, this model has many disadvantages: for example, when the network condition is bad, the client's request may be lost; when the server-side processing time is too long, it may cause the client to wait for a long time and the request timed out and failed.

To this end, Tencent Cloud introduced message queuing service for message distribution and management. With Tencent Cloud CMQ, you can separate the components of an application so that they can run independently, while simplifying message management between components. Any component of a distributed application can store messages in queues, and Tencent Cloud CMQ ensures that each message is delivered at least once and supports multiple reads and writes. A single queue can be used by multiple distributed application components at the same time without the need for collaboration between these components. All components can use CMQ API to retrieve and manipulate messages programmatically.

For all supported operations, see API Overview Page .

Please make sure that you have a good understanding of these interfaces before using them. CMQ product description .

> The helper account cannot be used for CMQ operations at this time.

Queue model

Glossary of terms

Below is the list of common terms used in this document:

Terminology	Full name	English	Description
CMQ	Cloud Message Queue	Cloud message queue	Tencent Cloud's message service includes queue model, topic model, high-performance message service and so on.
Queue	Queue	Queue	Queue is a first-in-first-out data model. Producers can add data to the end of the queue by Concurrence, and consumers can pull (pull) data from the head of the queue by Concurrence.
Active	Active	Can be consumed (visible) state	When the message is in Active state, it can be consumed by all consumers, but only one consumer can successfully consume it. At this time, the message becomes Inactive and is not visible to other consumers.
Inactive	Inactive	Non-consumable (invisible) state	When a message is in the Inactive state, it indicates that the message is being consumed by a consumer, and other consumers cannot consume the message.
(Batch) ReceiveMessage	(Batch) Receive Message	Consumption message	The ReceiveMessage operation simply sets the message from the Active state to the Inactive state so that the message cannot be consumed by other consumers. However, after consumption, Explicit has to call the (Batch) DeleteMessage operation to delete the message from the queue, otherwise, after the time specified by visibilityTimeout, the message becomes Active again and can be consumed by other consumers, resulting in unexpected duplicate messages.
MaxMsgHeapNum	Maximum Message Heap Number	Maximum number of Message Retention	In order to prevent the message from being lost because the message is not consumed in time, the message service has the function of heap message. Heap has an upper limit on the number of messages. Producers can no longer add messages to the queue after the limit is exceeded. After the message is consumed and deleted, the producer can add the message to the queue.

PollingWaitSeconds	Polling Wait Seconds	Long-polling Waiting Time for Message Receipt	When consumers want to pull messages from the queue for consumption, the queue may have no data for the time being. Consumers may not want to return immediately (similar to non-blocking mode) and want to wait for a period of time to see if a message will come. (similar to blocking mode), pollingWaitSeconds is similar to the timeout in blocking mode. After this time, it will return regardless of whether there is a message or not. If you want to consume messages nonblocking, set the value to 0.
MsgRetentionSeconds	Message Retention Seconds	Message Lifecycle	Heap's message in the queue has a retention time, after which it has not been consumed by consumers, it will be deleted by the queue, and consumers can no longer consume it.
ReceiptHandle	Receipt Handle	Message receipt handle	This handle is returned when the message is consumed, and only the handle that is currently consuming the message can delete the message. If the consumption time is too long and exceeds the visibilityTimeout, the message will be consumed by other consumers, and the previous consumer's handle to get the message will become invalid and cannot be used to delete the message.
Qps throttling	QPS Throttling	QPS restriction	The term originally means a limit on the number of requests per second, but it is actually a limit on the number of messages per second. The number of requests per second for a single interface (SendMessage, ReceiveMessage, DeleteMessage),). But for batch interfaces (BatchSendMessage, BatchReceiveMessage, BatchDeleteMessage),), it is the sum of all batch values within 1 second. The purpose of this restriction is to provide users with a more stable and fair product. If you need a larger message limit, you can Submit a ticket Apply to us.

Definition of input parameters and return parameters

Limit and offset

Limit and offset are parameters used to control paging. When the corresponding result is in the form of a list, if the number exceeds the value limited by limit, only limit values are returned. Users can control paging through two parameters: limit and offset: limit is the maximum number of entries returned at a time, and offset is the offset.

For example, the parameter offset=0&limit=20 returns items 0 to 20, offset=20&limit=20 returns items 20 to 40, offset=40&limit=20 returns items 40 to 60, and so on.

Id.n

Id.n is a format in which multiple parameters are entered at the same time. When a format like this is encountered, multiple input parameters can be passed at the same time.

For example, id.0= "10.12.243.21" & id.1= "10.12.243.21" & id.2= "10.12.243.21" & id.3= "10.12.243.21". And so on (starting with 0 below).

API Getting Started

Users can use the CMQ SDK (it is recommended that multiple language versions are available), or call Cloud API directly (it is more troublesome, it is recommended for users other than SDK language) to use CMQ services:

1. Select region and private and public network. Unlike other Tencent Cloud services, the request domain name of message queuing API varies with different regions. You need to select the domain name corresponding to the region. The composition rules of the request domain name are as follows `cmq-queue-region.api.qcloud.com/v2/index.php` The region field needs to replace: gz (Guangzhou), sh (Shanghai) and bj (Beijing) with a specific region. If the machine used by the user is a Tencent Cloud server, the private network domain name should be preferred, otherwise the public network domain name should be selected.
2. Create a queue . SDK (or cloud API) calls the CreateQueue API.
3. Message sending . SDK (or cloud API) calls the SendMessage (or BatchSendMessage) API.
4. Consumption message . SDK (or cloud API) calls the ReceiveMessage (or BatchReceiveMessage) API.
5. Delete message . SDK (or cloud API) calls DeleteMessage (or BatchDeleteMessage).
6. Delete queue . SDK (or cloud API) calls the DeleteQueue API.

Topic model

Glossary of terms

Below is the list of common terms used in this document:

Terminology	English	Description
Subscriber	Subscriber	Refers to the subscriber of the service in CMQ-Topic mode.
Production	Produce	Refers to the operation of the producer to write a message to the Topic.
Shipping	Subscription	Refers to the process of Topic's message to subscriber shipping.
Message receiving	Message-receiving model (PUSH)	The Topic model of CMQ, which already supports the PUSH mode of active push.

mode (PUSH)		
Retry policy	NotifyStrategy	The NotifyStrategy property of the subscription, which is the retry policy when an error occurs in pushing the message to the receiver. This policy is enabled by default. There are two options, one of which must be checked. a. Backoff retry: retry 3 times with a random value between 10 and 20 seconds. After more than 3 times, the message is discarded for the subscriber and will not be retried. b. Recession index retry (default check): retry 176 times, the total retry time is 1 day, the interval is: 2 ^0,2^ 1,. , 512, 512,. , 512 seconds.
Message Lifecycle	MsgRetentionSeconds	The longest survival time of a message in TOPIC. After the time specified by this parameter from sending to the queue, the message will be deleted regardless of whether the message has been fetched or not. The default value is 86400s (1 day), and modification is not allowed.
Max Message Length	MaxMsgSize	Limits the maximum length of the message body allowed to be sent to the queue; in units, byte, valid values range from 1024 to 65536byte, that is, 1K to 64K.
Message Retention	MessageRetentionPeriod	On by default. There is a message from the producer, which has not yet triggered shipping to the subscriber, or the subscriber failed to receive the message. For the time being, heap went to the Topic and tried again. The project cannot be configured. The maximum heap time is 1 day.
Retry verification	Status code	After Topic shipping arrives at the subscriber, if the HTTPS return code is 200, it will be considered successful.
Add subscriber tags	FilterTag	When you add a subscriber, you can add FilterTag,. After adding FilterTag, the subscriber can only receive messages with that FilterTag. A single Tag is a string of no more than 16 characters, and a single subscriber can add up to 5 Tag. As long as one of the Tag, matches the filter tag of Topic, you can receive the message from Topic shipping. If the message does not have any tags, the subscriber cannot receive this type of message.
Add message filter tags	Messagetag	That is, message label and message type, which are used to distinguish the message classification under the Topic of a certain CMQ. MQ allows consumers to filter messages by Tag, ensuring that consumers end up consuming only the types of messages they care about. This feature is not enabled by default. When it is not enabled, all messages are sent to all subscribers. When the subscriber sets Tag, the subscriber cannot receive the message

		because of a mismatch. The message filtering tag describes the label of message filtering in the subscription (only messages with consistent labels are pushed). A string with no more than 16 characters in a single Tag, and a single Message can add up to 5 Tag.
Open log track	LoggingEnabled	Whether to enable the log management feature. True: enable, False: disable. When enabled, the original Log of CMQ is written to Cloud Object Storage COS. And users can do Log aggregation query through the CMQ console to avoid the tedious need to build their own analysis system.

Definition of input parameters and return parameters

Limit and offset

Parameters used by limit and offset to control paging. When the corresponding result is in the form of a list, if the number exceeds the value limited by limit, only limit values are returned. Users can control paging through two parameters: limit and offset: limit is the maximum number of entries returned at a time, and offset is the offset.

For example, the parameter offset=0&limit=20 returns items 0 to 20, offset=20&limit=20 returns items 20 to 40, offset=40&limit=20 returns items 40 to 60, and so on.

Id.n

Id.n is a format in which multiple parameters are entered at the same time. When a format like this is encountered, multiple input parameters can be passed at the same time.

For example, id.0= "10.12.243.21" & id.1= "10.12.243.21" & id.2= "10.12.243.21" & id.3= "10.12.243.21". And so on (starting with 0 below).

API Getting Started

Users can use the CMQ SDK (it is recommended that multiple language versions are available), or call Cloud API directly (it is more troublesome, it is recommended for users other than SDK language) to use CMQ services:

1. Select region and private and public network. Unlike other Tencent Cloud services, the request domain name of message queuing API varies with different regions. You need to select the domain name corresponding to the region. The composition rules of the request domain name are as follows `cmq-topic-region.api.qcloud.com/v2/index.php` The region field needs to replace: gz (Guangzhou), sh (Shanghai) and bj (Beijing) with a specific region. If the machine used by the user is a Tencent Cloud server, the private network domain name should be preferred, otherwise the public network domain name should be selected.
2. Create topic . SDK (or cloud API) calls the CreateTopic API.
3. Modify Topic Attributes . SDK (or cloud API) calls the SetTopicAttributes API.
4. Get topic list . SDK (or cloud API) calls the ListTopic API.
5. Get topic attribute . SDK (or cloud API) calls the GetTopicAttributes API.
6. Delete topic . SDK (or cloud API) calls the DeleteTopic API.

7. Publish news . SDK (or cloud API) calls the PublishMessage API.
8. Batch publish messages . SDK (or cloud API) calls the BatchPublishMessage API.
9. Shipping news .
10. Create a subscription . SDK (or cloud API) calls the Subscribe API.
11. Get subscription list . SDK (or cloud API) calls the ListSubscriptionByTopic API.
12. Modify subscription properties . SDK (or cloud API) calls the SetSubscriptionAttributes API
13. Get subscription properties . SDK (or cloud API) calls the GetSubscriptionAttributes API.
14. Delete Subscription . SDK (or cloud API) calls the Unsubscribe API.

API Overview

Last updated：2024-12-18 17:34:59

Queue Model

Queue APIs

API Feature	Action ID	Description
Creating queue	CreateQueue	Creates queue under your account.
Getting queue list	ListQueue	Lists queues under your account. Data can be obtained by page.
Getting queue attribute	GetQueueAttributes	Gets the attribute of created queue.
Modifying queue attribute	SetQueueAttributes	Modifies the attribute of message queue.
Deleting queue	DeleteQueue	Deletes created queue.

Message APIs

API Feature	Action ID	Description
Sending message	SendMessage	Sends message to specified queue.
Sending messages in batches	BatchSendMessage	Sends messages in batches to specified queue.
Consuming message	ReceiveMessage	Consumes message in queue.
Consuming messages in batches	BatchReceiveMessage	Consumes multiple messages in queue.
Deleting message	DeleteMessage	Deletes consumed message.
Deleting messages in batches	BatchDeleteMessage	Deletes consumed messages in batches.

Topic Model

Topic APIs

API Feature	Action ID	Description

Creating topic	CreateTopic	Creates topic under your account.
Modifying topic attribute	SetTopicAttributes	Modifies the attribute of created topic.
Getting topic list	ListTopic	Lists topics under your account. Data can be obtained by page.
Getting topic attribute	GetTopicAttributes	Gets the attribute of created topic.
Deleting topic	DeleteTopic	Deletes created topic.

Message APIs

API Feature	Action ID	Description
Publishing message	PublishMessage	Publishes message to specified topic.
Publishing messages in batches	BatchPublishMessage	Publishes messages to specified topic in batches.

Subscription APIs

API Feature	Action ID	Description
Creating subscription	Subscribe	Creates subscription under your account.
Getting subscription list	ListSubscriptionByTopic	Lists subscriptions under topic. Data can be obtained by page.
Modifying subscription attribute	SetSubscriptionAttributes	Sets the attribute of created subscription.
Getting subscription attributes	GetSubscriptionAttributes	Gets the attribute of created subscription.
Deleting subscription	Unsubscribe	Deletes created subscription.

Usage Examples

Last updated：2024-12-18 17:35:00

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

In order to help users quickly use the cloud messaging service (CMQ) API, a Use Case is given here, which can guide users to create queues, send messages, consume messages, delete messages and delete queues using API.

> The parameters in the following example are for reference only and are actually based on the values returned by the system. For ease of illustration, operations on messages take a single operation as an example, batch interfaces such as Send messages in batch 、 Bulk delete messages Etc.) is also available.

Please refer to the following instructions to replace {$region} in the domain name with the corresponding region:

Domain name requested by public network API: `https://cmq-queue-{$region}.api.qcloud.com`

Domain name requested by private network API: `http://cmq-queue-{$region}.api.tencentyun.com`

> {$region} need to replace: gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Finance), szjr (Shenzhen Finance), hk (Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore with a specific region. The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Queue model

Take private network as an example (when operating in a public network, you need to replace the tencentyun in the domain name with the composition rules of the domain name requested by qcloud), as shown in `cmq-queue-{$region}.api.tencentyun.com/v2/index.php` The {$region} field needs to be replaced by a specific region:: gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Finance), szjr (Shenzhen Finance), hk (Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore). Common Parameters The region value in should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to

the region specified by the domain name region based on the region value of the domain name.
For more information on the generation rules of Signature parameters, please see Signature Method .

Create a queue

Before you create a queue, please refer to the Create a queue interface Description: adjust the attribute value of the queue according to your own service.
If we create a queue in Guangzhou and expect business messages to be faster than production, the specific request parameters required are shown in the following table:

Parameter name	Description	Sample value
QueueName	The queue name must be unique under the same account in a single region. The queue name is a string of no more than 64 characters, must begin with a letter, and the rest can contain letters, numbers, and dashes (-).	Test-queue-1
PollingWaitSeconds	Long-polling Waiting Time for Message Receipt . The value range is 0-30 seconds, and the default value is 0.	thirty

To sum up, combined with common request parameters and API request parameters, the final request form is as follows:

```
http://cmq-queue-gz.api.tencentyun.com/v2/index.php?
Action=CreateQueue
&Region=gz
&Timestamp=1465750149
&Nonce=46364
&SecretId=AKIDxxxxugEY
&Signature=5umi9gUWpTTyk18V2g%2FYi56hqls%3D
&queueName=test-queue-1
&pollingWaitSeconds=30
```

Returned result:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555",
"queueId":"queue-ajksdfasdowe"
}
```

At this point, we have created a queue that can manage queues and send consumer messages.

Message sending

Please refer to the Message sending interface Description.

If we create the example above `test-queue-1` Send a message in the queue. The specific request parameters required are shown in the following table:

Parameter name	Description	Sample value
QueueName	The queue name must be unique under the same account in a single region. The queue name is a string of no more than 64 characters, must begin with a letter, and the rest can contain letters, numbers, and dashes (-).	Test-queue-1
MsgBody	Message body. The size is at least 1Byte, and the maximum length is limited by the set queue message maximum length attribute.	After This'is test message (url coding, it becomes `This%27is+test+message`)

Request:

```
http://cmq-queue-gz.api.tencentyun.com/v2/index.php?
Action=SendMessage
&Region=gz
&Timestamp=1465750149
&Nonce=46365
&SecretId=AKIDxxxxugEY
&Signature=5umi9gUWagTTyk18V2g%2FYi56hqls%3D
&queueName=test-queue-1
&msgBody=This%27is+test+message
```

Returned result:

```
{
"code" : 0,
"message" : "",
"requestId":"145346456555",
"msgId":"123345346"
}
```

Consumption message

Please refer to the message carefully before consuming the message. Consumer message interface Description. According to the characteristics of the business, you can choose to specify the value of pollingWaitSeconds.

If we create from the example above `test-queue-1` Messages are consumed in the queue. The specific request parameters required are shown in the following table:

Parameter name	Description	Sample value
QueueName	The queue name must be unique under the same account in a single region. The queue name is a string of no more than 64 characters, must begin with a letter, and the rest can contain letters, numbers, and dashes (-).	Test-queue-1
PollingWaitSeconds	The long polling waiting time for this request. The value range is 0-30 seconds, and the default value is 0.	10

Request:

```
http://cmq-queue-gz.api.tencentyun.com/v2/index.php?
Action=ReceiveMessage
&Region=gz
&Timestamp=1465750150
&Nonce=46368
&SecretId=AKIDxxxxugEY
&Signature=5umi9gUaagTTyk18V2g%2FYi56hqls%3D
&queueName=test-queue-1
&pollingWaitSeconds=10
```

Returned result:

```
{
"code" : 0,
"message" : "",
"requestId":"145346635355",
"msgBody":"This is test message",
"msgId":"123345346",
"receiptHandle": "283748239349283",
"enqueueTime": 1462351990,
"firstDequeueTime": 1462352990,
"nextVisibleTime": 1462352999,
"dequeueCount": 1
}
```

Delete message

In general, messages should be deleted once consumed, unless the business has a need for repeated consumption. Please read carefully before deleting the message Delete message interface Description.

For example, if we spend `test-queue-1` A message in the queue and delete it after consumption. **Be sure to delete before the time of nextVisibleTime** Otherwise, the receiptHandle will become invalid, causing the deletion to fail. The specific request parameters required are shown in the following table:

Parameter	Description	Sample value

name		
QueueName	The queue name must be unique under the same account in a single region. The queue name is a string of no more than 64 characters, must begin with a letter, and the rest can contain letters, numbers, and dashes (-).	Test-queue-1
ReceiptHandle	Each consumption returns a unique Message Handler, which is used to delete messages. The handle generated when and only when the message was last consumed can be used to delete this message.	"283748239349283" (receiptHandle in the example above)

Request:

```
http://cmq-queue-gz.api.tencentyun.com/v2/index.php?
Action=DeleteMessage
&Region=gz
&Timestamp=1465750151
&Nonce=46369
&SecretId=AKIDxxxxugEY
&Signature=5umi9gUaagTasdfk18V2g%2FYi56hqls%3D
&queueName=test-queue-1
&receiptHandle=283748239349283
```

Returned result:

```
{
"code" : 0,
"message" : "",
"requestId":"14534454555"
}
```

Delete queue

Before deleting the queue, please refer to the Delete queue interface Description. When the queue is no longer in use, it needs to be deleted. The specific request parameters required are shown in the following table:

Parameter name	Description	Value
QueueName	The queue name must be unique under the same account in a single region. The queue name is a string of no more than 64 characters, the first character must be a letter, and the rest can contain letters, numbers, and dashes (-).	Test-queue-1

Request:

```
http://cmq-queue-gz.api.tencentyun.com/v2/index.php?
Action=DeleteQueue
&Region=gz
&Timestamp=1465750152
&Nonce=46370
&SecretId=AKIDxxxxugEY
&Signature=5umi9gUaagTasasdl18V2g%2FYi56hqls%3D
&queueName=test-queue-1
```

Returned result:

```
{
"code" : 0,
"message" : "",
"requestId":"14534454555"
}
```

Request Domain Description

Last updated：2025-03-26 16:29:16

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Fees will be incurred whenever public network downstream traffic is generated from using a public network domain name, including beta testing. We strongly recommend that users with services on Tencent Cloud use private network domain names because doing so will not incur traffic fees.

Request Domain Name

Queue Model

Please refer to the notes below to replace {$region} in the domain name with the corresponding region name:

Domain name for public network API request: `https://cmq-queue-{$region}.api.qcloud.com`

Domain name for private network API request: `http://cmq-queue-{$region}.api.tencentyun.com`

Topic Model

Please refer to the notes below to replace {$region} in the domain name with the corresponding region name:

Domain name for public network API request: `https://cmq-topic-{$region}.api.qcloud.com`

Domain name for private network API request: `http://cmq-topic-{$region}.api.tencentyun.com`

Notes

{$region} must be replaced with a specific region, such as gz (Guangzhou), sh (Shanghai), bj (Beijing), hk (Hong Kong), cd (Chengdu), ca (North America), usw (West US), use (East US), th (Thailand), and sg (Singapore). The value of "region" in the common parameters should be consistent with that in the domain name. If there is any inconsistency, the request will be sent to the region specified in the domain name.

The domain name for public network request only supports HTTPS, and the domain name for private network request only supports HTTP.

Some of the input parameters are optional. Default value is used for those not specified.

All the output parameters are returned to the user when the request is successful; otherwise, at least code, message, and requestId are returned.

Region

The "region" in the API request domain name must be replaced with a specific region. The region is subject to the "region" value of the domain name instead of that in the common parameters. Currently, the regions launched in CMQ are shown in the following table:

Queue Model

Region	Replacement Value	Domain Name for Public Network Request	Domain Name for Private Network Request
Beijing	bj	`https://cmq-queue-bj.api.qcloud.com`	`http://cmq-queue-bj.api.tencentyun.com`
Shanghai	sh	`https://cmq-queue-sh.api.qcloud.com`	`http://cmq-queue-sh.api.tencentyun.com`
Guangzhou	gz	`https://cmq-queue-gz.api.qcloud.com`	`http://cmq-queue-gz.api.tencentyun.com`
Hong Kong	hk	`https://cmq-queue-hk.api.qcloud.com`	`http://cmq-queue-hk.api.tencentyun.com`
Chengdu	cd	`https://cmq-queue-cd.api.qcloud.com`	`http://cmq-queue-cd.api.tencentyun.com`
North America	ca	`https://cmq-queue-ca.api.qcloud.com`	`http://cmq-queue-ca.api.tencentyun.com`
West US	usw	`https://cmq-queue-usw.api.qcloud.com`	`http://cmq-queue-usw.api.tencentyun.com`
East US	use	`https://cmq-queue-use.api.qcloud.com`	`http://cmq-queue-use.api.tencentyun.com`
Thailand	th	`https://cmq-queue-th.api.qcloud.com`	`http://cmq-queue-th.api.tencentyun.com`
Singapore	sg	`https://cmq-queue-sg.api.qcloud.com`	`http://cmq-queue-sg.api.tencentyun.com`

Topic Model

Region	Replacement Value	Domain Name for Public Network Request	Domain Name for Private Network Request

Beijing	bj	`https://cmq-topic-bj.api.qcloud.com`	`http://cmq-topic-bj.api.tencentyun.com`
Shanghai	sh	`https://cmq-topic-sh.api.qcloud.com`	`http://cmq-topic-sh.api.tencentyun.com`
Guangzhou	gz	`cmq-topic-gz.api.qcloud.com`	`cmq-topic-gz.api.tencentyun.com`
Hong Kong	hk	`https://cmq-topic-hk.api.qcloud.com`	`http://cmq-topic-hk.api.tencentyun.com`
North America	ca	`https://cmq-topic-ca.api.qcloud.com`	`http://cmq-topic-ca.api.tencentyun.com`
Chengdu	cd	`https://cmq-topic-cd.api.qcloud.com`	`http://cmq-topic-cd.api.tencentyun.com`
West US	usw	`https://cmq-topic-usw.api.qcloud.com`	`http://cmq-topic-usw.api.tencentyun.com`
East US	use	`https://cmq-topic-use.api.qcloud.com`	`http://cmq-topic-use.api.tencentyun.com`
Thailand	th	`https://cmq-topic-th.api.qcloud.com`	`http://cmq-topic-th.api.tencentyun.com`
Singapore	sg	`https://cmq-topic-sg.api.qcloud.com`	`http://cmq-topic-sg.api.tencentyun.com`

Update History

Last updated：2024-12-18 17:34:59

Date	Update
August 8, 2016	A document for queue model creation is added.
December 12, 2016	A document for topic model creation is added.

Calling Methods

Request Structure

Request Structure Overview

Last updated：2024-12-18 17:35:00

The call to a TencentCloud API is done by sending a request to the server address of the API and adding the corresponding request parameters to the request based on the API description. The structure of a TencentCloud API request consists of service address, communication protocol, request method, request parameters, and character encoding as detailed below:

Service Address

The service access address of TencentCloud API depends on the specific module. For more information, please see the description of each API.

Communication Protocol

Most TencentCloud APIs communicate over HTTPS, providing highly secure communication tunnels.

Request Method

TencentCloud API supports both POST and GET requests.

Note:

1. POST and GET requests cannot be mixed. If GET is used, parameters will be taken from the query string. If POST is used, parameters will be taken from the request body, and parameters in the query string will be ignored. The parameter format rules of the two request methods are identical. GET requests are generally used. If the parameter string is too long, POST is recommended.
2. Parameters sent in GET requests have to be URL-encoded. This is not needed for POST requests.
3. The maximum length of a GET request varies by browser and server settings. For example, the limit is 2 KB in Internet Explorer and 8 KB in Firefox. For long API requests with a lot of parameters, we recommend you use the POST method so as to avoid request failure due to overlong string.

4. For POST requests, the input parameters should be in the format of `x-www-form-urlencoded` , because TencentCloud API gets request parameters from `$_POST` .

Request Parameters

Two types of request parameters are required for each TencentCloud API request: common ones and API ones. Common request parameters are required for every API (please see Common Request Parameters), while API request parameters are specific to each API (please see "Request Parameters" in each API document).

Character Encoding

Both the request and returned result of TencentCloud API are encoded with the UTF-8 character set.

Common Request Parameters

Last updated：2024-12-18 17:34:59

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

A complete Tencent Cloud API request requires two types of request parameters: common request parameters and API request parameters. This article will introduce six common request parameters that Tencent Cloud needs to use in API request. For more information on API request parameters, please see API Request Parameters Chapter.

Common request parameters are the request parameters that every API needs to use. Developers need to bring these common request parameters every time they use Tencent Cloud API to send requests, otherwise the request will fail. Also, you need to capitalize the first letter in each common request parameter so that it can be differentiated from a API request parameter.

The following lists the specific common request parameters:

Note:

The interface examples in this article take Tencent Cloud CVM as an example. Please correspond to the actual products for the specific use of Tencent Cloud products.

Parameter name	Description	Type	Required
Action	The name of the instruction API for the specific operation, for example, Tencent Cloud CVM user calls Querying the List of Instances Interface, the Action parameter is DescribeInstances.	String	Yes
Region	The region parameter, which is used to identify the instance of the region you want to operate on. For more information, see Regions and AZs List, or use the Query a list of regions API check. 1. Normally, this parameter is required, and if it does not need to be passed, it will be described in the appropriate API. 2. Some areas are under internal testing and are currently only open to some users.	String	No
Timestamp	Timestamp, the current UNIX, can record the time when the API request was initiated.	UInt	Yes
Nonce	Users can customize random positive integers, combined with Timestamp, to prevent replay attacks.	UInt	Yes
SecretID	In Cloud API Key A SecretId that identifies the identity applied on the SecretId, corresponds to a unique SecretKey, and the SecretKey is used to generate the request signature Signature. For details, please refer to Signature Method Chapter.	String	Yes

Signature	The request signature, which is used to verify the validity of the request, needs to be calculated by the user according to the actual input parameters. The calculation method can be used for reference. Signature Method Chapter.	String	Yes
SignatureMethod	Signature method. HmacSHA256 and HmacSHA1 are supported. The HmacSHA256 method is used to verify signatures only when the parameter is specified as HmacSHA256. Otherwise, HmacSHA1 is used. Please refer to the detailed signature calculation method. Signature Method Chapter.	String	No
Token	The Token, used by temporary certificate needs to be used in conjunction with a temporary key. A token is not required when a long-term key is being used.	String	No

Usage Examples

The following examples show how a common request parameter looks like in a TencentCloud API request. For example, if you want to query the list of Tencent Cloud CVM instances in Guangzhou region, the desired request format is:

```
Https://cvm.api.qcloud.com/v2/index.php?Action=DescribeInstances&SecretId=xxxxxxx&Region=ap-guangzhou&Timestamp=1465055529&Nonce=59485&Signature=mysignature&SignatureMethod=HmacSHA256&<API Request Parameters>
```

API Request Parameters

Last updated：2024-12-18 17:34:59

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

A complete Tencent Cloud API request requires two types of request parameters: common request parameters and API request parameters. This article will introduce the API request parameters that Tencent Cloud needs to use for API requests. For more information on common request parameters, please see Common Request Parameters Chapter.

API request parameters vary with different APIs. The initial letter of each API request parameter is in lowercase so that it can be differentiated from a common request parameter.

Note:

The parameters in this article take Tencent Cloud CVM as an example. The actual parameters of each Tencent Cloud product should be explained according to the API parameters of the actual product.

The following parameter list is based on Tencent Cloud CVM Querying the List of Instances Take the API (DescribeInstances) as an example. The API request parameters supported by the API are as follows:

Parameter name	Description	Type	Required
InstanceIds.n	The ID array of CVM instances to be queried, with the array subscript starting at 0. You can use either instanceId or unInstanceId. The unified resource ID unInstanceId is recommended.	String	No
LanIps.n	The Private IP array of Cloud Virtual Machine to be queried.	String	No
SearchWord	The host alias set by the user.	String	No
Offset	Offset, default is 0.	Int	No
Limit	The maximum number of servers that can be queried at a time, the default is 20, and the maximum is 100.	Int	No
Status	The status of the host to be queried.	Int	No
Simplify	Get non-real-time data, and get non-real-time data when passing parameters to add simplify=1.	Int	No
ZoneId	If ID, is not passed in the availability zone, all CVM instances in the availability zone will be queried. To specify an availability zone, you can call the Query availability zone This API ((DescribeAvailabilityZones)) is used to query.	Int	No

The elements of each parameter are described as follows:

Parameter name: The name of the request parameter supported by the interface, which users can use as the interface request parameter when using this interface. If the parameter name is `“.n”` At the end, it indicates that this parameter is an array, and you need to pass in the array parameters in turn when using it.

Required: Indicates whether this parameter is required. If it is "Yes", it must be passed in to call the interface; if it is "No", it can not be passed in.

Type: The data type of this interface parameter.

Description: Briefly describes the contents of the request parameters for this API.

Usage Examples

The following example shows how API request parameters look in an API request link for a Tencent Cloud product. If, for example, you want to query the list of scaling groups for a Tencent Cloud CVM, the request link should look like this:

`Https://cvm.api.qcloud.com/v2/index.php?&& instanceIds.0=ins-0hm4gvho&instanceIds.1=ins-8oby8q00&offset=0&limit=20&status=2&zoneId=100003`

Final request form

Last updated：2024-12-18 17:34:59

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Splicing rule

A Tencent Cloud API request URL is constructed as follows:

Note:

Https:// + request domain name + request path +? + final request parameter string

The elements of each URL are described as follows:

Request domain name: The request domain name is determined by the product to which the API belongs or the module of the product to which the API belongs. The request domain name for different products or modules of different products may be different. For example, the request domain name for querying instance list (DescribeInstances) of Tencent Cloud CVM is as follows: `cvm.api.qcloud.com` . For more information about the request domain names for different products, see the relevant API documents.

Request path: Tencent Cloud API corresponds to the request path of a product. Generally speaking, a product corresponds to a fixed path (for example, Tencent Cloud CVM request path is fixed as `/v2/index.php`).

Final request parameter string: The request parameter string of the interface includes common request parameters and interface request parameters.

Usage Examples

The final request URL for a TencentCloud API is as follows:

With Tencent Cloud CVM Querying the List of Instances Take the API (DescribeInstances) as an example. The first six parameters are common request parameters, and the last six parameters are API request parameters.

```
https://cvm.api.qcloud.com/v2/index.php?
Action=DescribeInstances
&SecretId=xxxxxxx
&Region=gz
&Timestamp=1465055529
&Nonce=59485
&Signature=mysignature //Common request parameters
&instanceIds.0=ins-0hm4gvho
&instanceIds.1=ins-8oby8q00
&offset=0
&limit=20
&status=2
&zoneId=100003 //API Request Parameters
```

Request Result

Successful Response

Last updated：2024-12-18 17:34:59

If an API call succeeds, the `code` field in the final returned result will be 0, the `message` field will be empty, and the returned result data will be displayed.

Below is a sample:

```
{
    "code": "0",
    "message": ""
    <Returned result data>
}
```

Error Response

Last updated：2024-12-18 17:34:59

If an API call fails, the `code` field in the final returned result will not be 0, and a detailed error message will be displayed in the `message` field. You can go to Error Codes and use `code` and `message` to check the error information.

Below is a sample error:

```
{
    "code": "5100",
    "message": "(100004) incorrect projectId",
}
```

Signature

Last updated：2024-12-18 17:34:59

TencentCloud API authenticates each access request, that is, each request must include signing information (Signature) in the common request parameters to verify the identity of the user. The signature is generated by the security credentials which consist of a `SecretId` and a `SecretKey` . If you do not have the security credentials yet, you can apply for them at Tencent Cloud's official website; otherwise, you cannot call TencentCloud API.

Signature Algorithm Description

CMQ allows clients to use two signature algorithms: SHA1 and SHA256, which can be specified in the `SignatureMethod` parameter. If the parameter value is `HmacSHA256` , SHA256 will be used for signature calculation; if this parameter is not specified or its value is not `HmacSHA256` , SHA1 will be used.

1. Apply for security credentials

Before using TencentCloud API for the first time, you need to apply for security credentials in the Tencent Cloud Console, which consists of `SecretId` and `SecretKey` . `SecretId` is used to identify the API requester, while `SecretKey` is a key used for signature string encryption and authentication by the server. Please keep your `SecretKey` private and do not disclose it to others.

You can apply for security credentials as follows:

1.1. Log in to the Tencent Cloud Console.

1.2. Click **Tencent Cloud Services** and select **Access Key** under **Management and Audit** to go to the TencentCloud API key management page.

1.3. On the TencentCloud API Access Key Management page, click **Create** to create a pair of `SecretId/SecretKey` . Each account can have up to two `SecretId/SecretKey` pairs.

2. Generate a signature string

With the `SecretId` and `SecretKey` , a signature string can be generated as described below:

Suppose the `SecretId` and `SecretKey` are:

SecretId: `AKIDPcY*****CVYLn3zT`

SecretKey: `pPgfLip*****aU7UbQyFFx`

Note:

This is just an example. To perform actual operations, use your own `SecretId` and `SecretKey` .

For example, if you call the `SendMessage` API to send a message, the request parameters may be as follows:

Parameter Name	Description	Value
Action	Method name	SendMessage
SecretId	Key ID	`AKIDPcY*****CVYLn3zT`
Timestamp	Current timestamp	1534154812
SignatureMethod	Signature algorithm	HmacSHA1
Nonce	Random positive integer	2889712707386595659
queueName	Name of the queue sending message	test1
RequestClient	Client version	SDK_Python_1.3
clientRequestId	Unique custom ID of client	123***1231
delaySeconds	Delay time	0
msgBody	Message content to be sent	msg

As shown above, the request has only five common request parameters (`Action` , `SecretId` , `Timestamp` , `Nonce` , and `SignatureMethod`) instead of the six ones described in "common request parameters". Actually, the sixth parameter `Signature` (signature string) is generated by other parameters (including signaling request parameters) together in the following steps:

2.1. Sort parameters

First, sort all request parameters by their names in ascending lexicographical order, just like sorting words in a dictionary in ascending alphabetical or numerical order. That is to say, sort the parameters by their first letters, and then sort the parameters with the same first letter by their second letters and so on. You can do this with the aid of relevant sorting functions in the programming language, such as the `ksort` function in PHP. The sorting results of the above sample parameters are as follows:

```
Action=SendMessage
Nonce=2889712707386595659
RequestClient=SDK_Python_1.3
SecretId=AKIDPcY*****CVYLn3zT
SignatureMethod=HmacSHA1
Timestamp=1534154812
clientRequestId=123***1231
delaySeconds=0
```

```
msgBody=msg
queueName=test1
```

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

2.2. Concatenate a request string

This step generates a request string.

Format the request parameters sorted in the previous step into the form of `parameter=value` . For example, for the `Action` parameter, its parameter is `Action` and its value is `SendMessages` ; therefore, the parameter will be formatted into `Action=SendMessage` .

Note:

The `value` is the original value instead of the URL-encoded value.

If an input parameter contains an underscore, the underscore needs to be replaced with a ".".

Then, concatenate the formatted parameters with `&` . The generated request string will be as follows:

```
Action=SendMessage&Nonce=2889712707386595659&RequestClient=SDK_Python_1.3&Secre
tId=AKIDPcY*****CVYLn3zT&SignatureMethod=HmacSHA1&Timestamp=1534154812&clientRe
questId=123***1231&delaySeconds=0&msgBody=msg&queueName=test1
```

2.3. Generate an original signature string

This step generates the original signature string.

The original signature string consists of the following parameters:

Request method: POST and GET methods are supported. GET is used here for the request. Please note that the method name should be in all capital letters.

Request domain name: here, suppose the private domain name of the CMQ service in the Guangzhou region `cmq-queue-gz.api.tencentyun.com` is requested.

Request path: the request path of TencentCloud API is always `/v2/index.php` .

Request string: the request string generated in the previous step.

The rule for concatenating the original string of the signature is `request method + request server + request path + ? + request string` .

The concatenation result in the example is as follows:

```
POSTcmq-queue-gz.api.tencentyun.com/v2/index.php?
Action=SendMessage&Nonce=2889712707386595659&RequestClient=SDK_Python_1.3&Secre
tId=AKIDPcY*****CVYLn3zT&SignatureMethod=HmacSHA1&Timestamp=1534154812&clientRe
questId=123***1231&delaySeconds=0&msgBody=msg&queueName=test1
```

2.4. Generate a signature string

This step generates a signature string.

Use the HMAC-SHA1 algorithm to sign the **original signature string** obtained in the previous step, and then Base64-encode the generated signature to get the final signature.

The specific code when PHP is used is as follows:

```
$secretKey = 'pPgfLipfEXZ7VcRzhAMIyPaU7UbQyFFx';
$srcStr = 'POSTcmq-queue-gz.api.tencentyun.com/v2/index.php?
Action=SendMessage&Nonce=2889712707386595659&RequestClient=SDK_Python_1.3&Secre
tId=AKIDPcY*****CVYLn3zT&SignatureMethod=HmacSHA1&Timestamp=1534154812&clientRe
questId=123***1231&delaySeconds=0&msgBody=msg&queueName=test1';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

The obtained signature string is as follows:

```
C16WEtEXsD5v5tnaUMLAbZewXhI=
```

When any other programming language is used for development and the original signature in the example is verified, the same result as described above should be obtained.

3. Encode a signature string

Note:

The generated signature string cannot be directly used as a request parameter and needs to be URL-encoded.

Parameters sent in GET requests have to be URL-encoded.

For example, if the signature string generated in the previous step is `C16WEtEXsD5v5tnaUMLAbZewXhI=` , it will be encoded to `C16WEtEXsD5v5tnaUMLAbZewXhI%3d` , and the final value of the `Signature` request parameter will be `C16WEtEXsD5v5tnaUMLAbZewXhI%3d` , which will be used to generate the final request URL.

The final request string is:

```
clientRequestId=1231231231&Nonce=2889712707386595659&Timestamp=1534154812&msgBo
dy=msg&Action=SendMessage&SignatureMethod=HmacSHA1&RequestClient=SDK_Python_1.3
&Signature=C16WEtEXsD5v5tnaUMLAbZewXhI%3D&delaySeconds=0&SecretId=AKIDPcY*****C
VYLn3zT&queueName=test1
```

Queue Model

Queue APIs

Create a queue

Last updated：2024-12-18 17:45:19

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (CreateQueue) is used to create a new queue under a user account.
Domain name requested by public network API: `https://cmq-queue-{$region}.api.qcloud.com`
Domain name requested by private network API: `http://cmq-queue-{$region}.api.tencentyun.com`
The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Finance), szjr (Shenzhen Finance), hk (Hong Kong, China), cd (Chengdu), ca (North America), usw (American West), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Note:

At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, it is strongly recommended that users of the service on Tencent Cloud use **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
QueueName	Yes	String	Queue name, which is unique under the same account in a single region. The queue name is a string of no more than 64 characters, must begin with a letter, and the rest can contain letters, numbers, and dashes (-).

MaxMsgHeapNum	No	Int	Maximum number of heap messages. The range of values during the public test period is `1,000,000 - 10,000,000` The range can be reached after the official Activate `1000,000-1000,000,000` . The default value during public testing is `10,000,000` After Activate, it is officially called `100,000,000` .
PollingWaitSeconds	No	Int	Long-polling Waiting Time for Message Receipt. The value range is 200ms-30s, and the default is 200ms.
VisibilityTimeout	No	Int	Message visibility timed out. Value range: 1-43200 seconds (that is, within 12 hours). The default value is 30.
MaxMsgSize	No	Int	Max Message Length. Value range: 1024-1048576 Byte (that is, 1-1024K), with a default value of 65536.
MsgRetentionSeconds	No	Int	Message Lifecycle. Value range: 60-1296000 seconds (1min-15 days). Default is 345600 (4 days).
RewindSeconds	No	Int	The maximum rewind time of the message in rewind queue, which ranges from 0 to 43200 seconds. 0 means the message rewind is not enabled.

Output Parameter

Parameter name	Type	Description
Code	Int	0: indicates success. Others: error. For more information, please see Common error code .
Message	String	Error message.
RequestId	String	Request ID generated by the server. When an internal error occurs on the server, the user can submit this ID to Backend Background to locate the problem.
QueueId	String	The unique identification ID of the queue.

Error Codes

Please check Common error code .

Example

Enter:

```
Https://domain/v2/index.php?Action=CreateQueue & queueName=test-queue-123 & <Common
```

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555",
"queueId":"queue-ajksdfasdowe"
}
```

Get queue list

Last updated：2024-12-18 17:45:19

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (ListQueue) is used to list the queues under a user account, and the data can be obtained by page.
Domain name requested by public network API: `https://cmq-queue-{$region}.api.qcloud.com`
Domain name requested by private network API: `http://cmq-queue-{$region}.api.tencentyun.com`
The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Finance), szjr (Shenzhen Finance), hk (Hong Kong, China), cd (Chengdu), ca (North America), usw (American West), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.
Note:
At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, it is strongly recommended that users of the service on Tencent Cloud use **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters Document.

Parameter name	Required	Type	Description
SearchWord	No	String	Used to filter the queue list, Backend Background uses fuzzy matching to return the list of queues that meet the criteria. If this parameter is left empty, all queues under the account will be returned by default.
Offset	No	Int	This page gets the starting position of the queue list when paging. If Enter reaches this value, Enter limit must also be required. When this value defaults, Backend Background takes the default value of 0

Limit	No	Int	This page gets the number of queues when paging. If this parameter is not passed, the parameter defaults to 20 and the maximum value is 50.

Output Parameter

Parameter name	Type	Description
Code	Int	Please see Error Codes
Message	String	Error message.
RequestId	String	Request ID generated by the server. When an internal error occurs on the server, the user can submit this ID to Backend Background to locate the problem.
TotalCount	Int	The total number of queues returned by this request under the user account, rather than the number of queues obtained on this page after paging
QueueList	Array	Queue list information, each element represents the information of a queue

QueueList is defined as follows:

Parameter name	Type	Description
QueueId	String	Unique identification ID of the queue
QueueName	String	Queue name, which is unique under the same account in a single region. The queue name is a string of no more than 64 characters, must begin with a letter, and the rest can contain letters, numbers, and dashes `-` ## Example

Enter:

```
Https://domain/v2/index.php?Action=ListQueue & searchWord=test & <Common Request Pa
```

Output:

```
{
    "code": 0,
    "message": "",
    "requestId": "14534664555",
    "totalCount": 10,
    "queueList": [{
            "queueId": "queue-asdfo",
            "queueName": "test-queue1"
        },
        {
            "queueId": "queue-asdsafo",
            "queueName": "queue-test1"
        }
    ]
}
```

Get queue properties

Last updated：2024-12-18 17:45:20

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (GetQueueAttributes) is used to obtain the attributes of a created queue. In addition to the settable properties set when the queue is created, you can also get the queue creation time, the last time the queue property was modified, and the Statistics (approximate value) of the message in the queue.

Domain name requested by public network API: `https://cmq-queue-{$region}.api.qcloud.com`

Domain name requested by private network API: `http://cmq-queue-{$region}.api.tencentyun.com`

The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Finance), szjr (Shenzhen Finance), hk (Hong Kong, China), cd (Chengdu), ca (North America), usw (American West), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Note:

At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
QueueName	Yes	String	Queue name, which is unique under the same account in a single region. The queue name is a string of no more than 64 characters, must begin with a letter, and the rest can contain letters, numbers, and dashes (-).

Output Parameter

Parameter name	Type	Description
Code	Int	0: successful. 4440: queue does not exist. The meaning of other return values can be found in Error Codes .
Message	String	Error message.
RequestId	String	Request ID generated by the server. When an internal error occurs on the server, the user can submit this ID to Backend Background to locate the problem.
MaxMsgHeapNum	Int	Maximum number of heap messages. The range of values during the public test period is `1,000,000 - 10,000,000` The range can be reached after the official Activate `1000,000-1000,000,000` . The default value during public testing is `10,000,000` After Activate, it is officially called `100,000,000` .
PollingWaitSeconds	Int	Long-polling Waiting Time for Message Receipt. The value range is 0-30 seconds, and the default value is 0.
VisibilityTimeout	Int	Message visibility timed out. Value range: 1-43200 seconds (that is, within 12 hours). The default value is 30.
MaxMsgSize	Int	Max Message Length. Value range: 1024-1048576 Byte (that is, 1K-1024K), with a default value of 65536.
MsgRetentionSeconds	Int	Message retention cycle. The value range is 60 mi 1296000 seconds (1min-15 days), and the default value is 345600 seconds (4 days).
CreateTime	Int	The time when the queue was created. Returns Unix timestamp, accurate to seconds.
LastModifyTime	Int	When the queue property was last modified. Returns Unix timestamp, accurate to seconds.
ActiveMsgNum	Int	The total number of messages in the queue in the Active state (not in the consumed state), which is an approximate value.
InactiveMsgNum	Int	The total number of messages in the queue in the Inactive state (being consumed), which is an approximate value.
RewindSeconds	Int	The maximum rewind time of the message in rewind queue, which ranges from 0 to 43200 seconds. 0 means the message rewind is not enabled.
RewindmsgNum	Int	The number of messages that have been deleted by calling the API

		DelMsg, but are still in rewind's retention time.
MinMsgTime	Int	The minimum unconsumed time of the message, in seconds.
QueueName	String	The name of the message queue.
QueueId	String	Message queuing ID.
CreateUin	Int	Created by Uin.
Bps	Int	Bandwidth limit.
QPS	Int	The limit on the number of messages produced per second, and the size of consumption messages is 1.1 times this value.
Tags	Array	Associate's label.

Example

Enter:

```
Https://domain/v2/index.php?Action=GetQueueAttributes & queueName=test-queue-123 &
```

Output:

```
{
    "code": 0,
    "message": "",
    "codeDesc": "Success",
    "requestId": "1661914201",
    "maxMsgHeapNum": 100000000,
    "pollingWaitSeconds": 3,
    "visibilityTimeout": 43200,
    "maxMsgSize": 65536,
    "msgRetentionSeconds": 86400,
    "rewindSeconds": 86400,
    "delayMsgNum": 0,
    "minMsgTime": 1564626851,
    "rewindMsgNum": 0,
    "inactiveMsgNum": 0,
    "activeMsgNum": 2,
    "lastModifyTime": 1563877026,
    "createTime": 1563877026,
    "queueName": "dns",
    "queueId": "queue-aiav4lys",
    "createUin": 100010439978,
    "Bps": 52428800,
    "qps": 5000,
    "tags": []
}
```

Modify queue properties

Last updated：2024-12-18 17:45:20

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (SetQueueAttributes) is used to modify the attributes of message queues.
Domain name requested by public network API: `https://cmq-queue-{$region}.api.qcloud.com`
Domain name requested by private network API: `http://cmq-queue-{$region}.api.tencentyun.com`
The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Finance), szjr (Shenzhen Finance), hk (Hong Kong, China), cd (Chengdu), ca (North America), usw (American West), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.
The queue name cannot be modified.
At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
QueueName	Yes	String	Queue name, which is unique under the same account in a single region. The queue name is a string of no more than 64 characters, must begin with a letter, and the rest can contain letters, numbers, and dashes (-).
MaxMsgHeapNum	No	Int	Maximum number of heap messages. The range of values during the public test period is `1,000,000 - 10,000,000` The range can be reached after the official Activate `1000,000-1000,000,000` . The default value

			during public testing is `10,000,000` After Activate, it is officially called `100,000,000` .
PollingWaitSeconds	No	Int	Long-polling Waiting Time for Message Receipt. The value range is 0-30 seconds, and the default value is 0.
VisibilityTimeout	No	Int	Message visibility timed out. Value range: 1-43200 seconds (that is, within 12 hours). The default value is 30.
MaxMsgSize	No	Int	Max Message Length. Value range is 1024-1048576Byte (that is, 1K-1024K), with a default value of 65536.
MsgRetentionSeconds	No	Int	Message retention cycle. Value range: 60-1296000 seconds (1min-15 days), with a default value of 345600 (4 days).
RewindSeconds	No	Int	The longest rewind time of the message. The maximum value range is 0-msgRetentionSeconds, message. The maximum rewind of the message is the preservation period of the message in the queue. 0 means the message rewind is not enabled.

Output Parameter

Parameter name	Type	Description
Code	Int	0: successful. 4440: queue does not exist. The meaning of other return values can be found in Error Codes .
Message	String	Error message.
RequestId	String	Request ID generated by the server. When an internal error occurs on the server, the user can submit this ID to Backend Background to locate the problem.
MaxMsgHeapNum	Int	The maximum number of heap messages after the change.
PollingWaitSeconds	Int	The changed message receives a long polling wait time.
VisibilityTimeout	Int	The visibility of the changed message timed out.
MaxMsgSize	Int	The maximum length of the changed message.
MsgRetentionSeconds	Int	The changed message life cycle.
RewindSeconds	Int	The longest news after the change rewind time.

Example

Enter:

```
Https://domain/v2/index.php?Action=SetQueueAttributes & queueName=test-queue-123 &
```

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555",
"queueId":"queue-ajksdfasdowe",
"maxMsgHeapNum":10000000,
"pollingWaitSeconds":20,
"visibilityTimeout":0,
"maxMsgSize":65536,
"msgRetentionSeconds":345600
}
```

Delete queue

Last updated：2024-12-18 17:45:19

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (DeleteQueue) is used to delete a created queue.
Domain name requested by public network API: `https://cmq-queue-{$region}.api.qcloud.com`
Domain name requested by private network API: `http://cmq-queue-{$region}.api.tencentyun.com`
The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Financial), szjr (Shenzhen Financial), hk (China Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Note:

At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
QueueName	Yes	String	Queue name, which is unique under the same account in a single region. The queue name is a string of no more than 64 characters, must begin with a letter, and the rest can contain letters, numbers, and dashes (-).

Output Parameter

Parameter name	Type	Description
Code	Int	0: successful. 4440: queue does not exist. The meaning of other return values can be found in Error Codes .
Message	String	Error message.
RequestId	String	Request ID generated by the server. When an internal error occurs on the server, the user can submit this ID to Backend Background to locate the problem.

Example

Enter:

```
Https://domain/v2/index.php?Action=DeleteQueue & queueName=test-queue-123 & <Common
```

<pre>Https://domain/v2/index.php?Action=DeleteQueue & queueName=test-queue-123 & <<a href="https://www.tencentcloud.com/document/product/213/31574">Common Request Parameters</a>></pre>

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555"
}
```

Rewind queue

Last updated：2024-12-18 17:45:19

1. API Description

This API (RewindQueue) is used for message rewind. For more information, please see Message Rewind.
Domain for public network API request: cmq-queue-region.api.qcloud.com
Domain for private network API request: cmq-queue-region.api.tencentyun.com

Note:

Whenever (including during alpha test) any public network downstream traffic is generated from the use of a public network domain, a fee will be charged. It is strongly recommended that users whose services are on the Tencent Cloud use **private network** domains, because no fee will be charged for the traffic consumed in the private network.
region should be replaced with a specific region: gz (Guangzhou), sh (Shanghai), or bj (Beijing). The region value in the common parameters should be consistent with the region value of the domain. If there is an inconsistency, the request will be sent to the region specified by the domain.
Public network domain requests both support http and https. Private network requests only support http.
Some of the input parameters are optional. If not specified, the default value will be taken.
All the output parameters will be returned to the user when the request is successful; otherwise, at least code, message, and requestId will be returned.

2. Input Parameters

The following request parameter list only provides API request parameters. For other parameters, refer to Common Request Parameters.

Parameter Name	Required	Type	Description
queueName	Yes	String	Queue name. This is unique under the same account in one region. The queue name is a string of no more than 64 characters, which can contain letters, numbers, and hyphens (-) and must begin with a letter.
startConsumeTime	Yes	Unix timestamp	If this time is set, API (Batch) receiveMessage will consume messages after this timestamp based on the order of production messages.

3. Output Parameters

Parameter Name	Type	Description
code	Int	0: Succeeded, others: Error. See the table below for specific errors.
message	String	Error message.
requestId	String	Request ID generated by server. When there is an internal error on the server, users can submit this ID to the backend to locate the problem.

Error Code	Module Error Code	Error Message	Description
6050	10670	your queue cannot be rewinded	Queue cannot be rewound.
4000	10680	invalid rewind time, you can get the vaild range of that via GetQueueAttributes	Invalid rewind time. You can view the queue attributes to get the correct rewind time.
4000	10700	rewindSeconds cannot be greater than msgRetentionSeconds	The rewind time cannot be greater than the message retention period.

Note: The error codes listed in the above table are specific to this API. If the error code you are looking for is not in the table, you may find it in the Common Error Codes.

4. Example

Input:

```
https://domain/v2/index.php?Action=RewindQueue &queueName=test-queue-123&startConsu
```

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555"
}
```

Message APIs

Message sending

Last updated：2024-12-18 17:45:19

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (SendMessage) is used to send a message to the specified queue.
Domain name requested by public network API: `https://cmq-queue-{$region}.api.qcloud.com`
Domain name requested by private network API: `http://cmq-queue-{$region}.api.tencentyun.com`
The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Finance), szjr (Shenzhen Finance), hk (Hong Kong, China), cd (Chengdu), ca (North America), usw (American West), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Note:

At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
QueueName	Yes	String	Queue name, which is unique under the same account in a single region. The queue name is a string of no more than 64 characters, must begin with a letter, and the rest can contain letters, numbers, and dashes (-).
MsgBody	Yes	String	Message body. At least 1Byte, the maximum length is limited by the

			set queue message maximum length attribute.
DelaySeconds	No	Int	In seconds, indicates how long it takes to delay the message before the user can see the message after it is sent to the queue.

Output Parameter

Parameter name	Type	Description
Code	Int	0: indicates success. Others: error. For more information, please see Common error code .
Message	String	Error message.
RequestId	String	When an internal server error occurs in the request ID, generated by the server, the user can submit this ID to Backend Background to locate the problem.
MsgId	String	The server generates a unique identity ID for the message.

Error Codes

Please check Common error code .

Example

Enter:

```
Https://domain/v2/index.php?Action=SendMessage & queueName=test-queue-123 & msgBody
```

`Https://domain/v2/index.php?Action=SendMessage & queueName=test-queue-123 & msgBody=helloworld &` Common Request Parameters

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555",
"msgId":"123345346"
}
```

Send messages in batch

Last updated：2024-12-18 17:45:19

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (BatchSendMessage) is used to send batches (up to 16 messages) to a specified queue.
Domain name requested by public network API: `https://cmq-queue-{$region}.api.qcloud.com`
Domain name requested by private network API: `http://cmq-queue-{$region}.api.tencentyun.com`
The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Financial), szjr (Shenzhen Financial), hk (China Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.
Note:
At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
QueueName	Yes	String	Queue name, which is unique under the same account in a single region. The queue name is a string of no more than 64 characters, must begin with a letter, and the rest can contain letters, numbers, and dashes (-).
MsgBody.n	Yes	String	Message body. Represents a message in this batch. Currently, the number of messages in batch cannot exceed 16.

			For the convenience of users, n can start with 0 or 1, but it must be continuous, such as sending two messages, which can be (msgBody.0, msgBody.1) or (msgBody.1, msgBody.2). Note: since the total size of all messages (excluding headers and other parameters, only msgBody) is limited to 64k, it is recommended to plan the number of batches to be sent in advance.
DelaySeconds	No	Int	It is measured in seconds, indicating how long it takes to delay the message before the user can see it after it is sent to the queue. (this delay is valid for a batch of messages and does not support many-to-many mapping)

Output Parameter

Parameter name	Type	Description
Code	Int	0: indicates success. Others: error. For more information, please see Common error code .
Message	String	Error message.
RequestId	String	When an internal server error occurs in the request ID, generated by the server, the user can submit this ID to Backend Background to locate the problem.
MsgList	Array	The server generates a unique identification ID list of messages, and each element is the information of a message.

MsgList is defined as follows:

Parameter name	Type	Description
MsgId	String	The server generates a unique identity ID for the message.

Error Codes

Please check Common error code .

Example

Enter:

```
Https://domain/v2/index.php?Action=BatchSendMessage & queueName=test-queue-123 & ms
```

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555",
"msgList":
  [
    {
      "msgId":"123345346"
    },
    {
       "msgId":"456436346"
    }
  ]
}
```

Consuming Messages

Last updated：2024-12-18 17:45:19

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

API Description

This API is used to consume a message in the queue.
Calling the `ReceiveMessage` API will turn the status of the obtained message to `inactive` , and the period of being `inactive` is specified by the `visibilityTimeout` queue attribute (for more information, please see CreateQueue API). After successful consumption within the period of `visibilityTimeout` , you are recommended to call the `(batch) DeleteMessage` API to delete the message; otherwise, it will return to the `active` status and may be consumed again. This ensures that the message is consumed at least once, but cannot guarantee the idempotency. Therefore, deduplication logic is required at the business side.
API public domain name: `https://cmq-queue-{$region}.api.qcloud.com`
API private domain name: `http://cmq-queue-{$region}.api.tencentyun.com`
In the above domain names, **`{$region}` needs to be replaced with a specific region**: `gz` (Guangzhou), `sh` (Shanghai), `bj` (Beijing), `hk` (Hong Kong (China)), `cd` (Chengdu), `ca` (North America), `usw` (West US), or `sg` (Singapore). The value of `region` in the common parameters should be the same as that of `region` in the domain name; otherwise, the latter will prevail, i.e., requests will be sent to the region as specified by `region` in the domain name.
Note:
Whenever downstream traffic is generated from using the public domain name, traffic fees will be charged (even during the beta test period). Therefore, you are strongly recommended to use the **private** domain name as traffic fees will not be incurred there.

Input Parameters

The list below contains only the API request parameters. Other parameters can be found in Common Request Parameters.

Parameter Name	Required	Type	Description
queueName	Yes	String	Queue name, which is unique under the same account in an

			individual region. It is a string of up to 64 characters, which must begin with a letter and can contain letters, digits, and dashes (`-`).
pollingWaitSeconds	No	Int	Long polling wait time in seconds. Value range: 0–30. If this parameter is not set, the value of `pollingWaitSeconds` in queue attributes will be used by default.

Output Parameters

Parameter Name	Type	Description
code	Int	0: success; others: error. For more information on errors, please see Common Error Codes.
message	String	Error message.
requestId	String	Request ID generated by the server, which can be submitted to the backend for troubleshooting when an internal server error occurs.
msgBody	String	Body of the consumed message.
msgId	String	Unique ID of the consumed message.
receiptHandle	String	Unique message handler returned from each consumption, which is used to delete the message. Only a message handler generated from the last consumption can be used to delete the message. After successful consumption within the period of `visibilityTimeout` , you can use the handler to call the `(batch) DeleteMessage` API to delete the message. If `visibilityTimeout` has elapsed, the handler will become invalid.
enqueueTime	Int	Time when consumption is created and enters the queue. A Unix timestamp will be returned which is accurate down to the second.
nextVisibleTime	Int	Time when a message is visible next time (i.e., consumable again). A Unix timestamp will be returned which is accurate down to the second.
dequeueCount	Int	Reserved field.
firstDequeueTime	Int	Reserved field.

Error Codes

Please see Common Error Codes.

Samples

Input:

```
https://domain/v2/index.php?Action=ReceiveMessage &queueName=test-queue-123 &<Commo
```

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555",
"msgBody":"helloworld1",
"msgId":"123345346",
"receiptHandle": "283748239349283",
"enqueueTime": 1462351990,
"firstDequeueTime": 1462352990,
"nextVisibleTime": 1462352999,
"dequeueCount": 2
}
```

Bulk consumption of messages

Last updated：2024-12-18 17:45:19

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (BatchReceiveMessage) is used to consume multiple messages (up to 16 messages at present) in the queue.

The length of time that the BatchReceiveMessage operation will change the status of the obtained message to inactive,inactive is specified by the Queue property visibilityTimeout see CreateQueue interface). Consumers need to call API (batch) DeleteMessage to delete the message after successful consumption within visibilityTimeout time, otherwise the message will become active again and the message can be consumed again by consumers.

Domain name requested by public network API: `https://cmq-queue-{$region}.api.qcloud.com`

Domain name requested by private network API: `http://cmq-queue-{$region}.api.tencentyun.com`

The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Financial), szjr (Shenzhen Financial), hk (China Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Note:

At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
QueueName	Yes	String	Queue name, which is unique under the same account in a single region. The queue name is a string of no more than 64

			characters, must begin with a letter, and the rest can contain letters, numbers, and dashes (-).
NumOfMsg	Yes	Int	The number of messages consumed this time. The range of values is 1-16.
PollingWaitSeconds	No	Int	The long polling waiting time for this request. The value range is 0-30 seconds. If this parameter is not set, the pollingWaitSeconds value in the queue attribute is used by default.

Output Parameter

Parameter name	Type	Description
Code	Int	0: indicates success. Others: error. For more information, please see Common error code .
Message	String	Error message.
RequestId	String	Request ID generated by the server. When an internal error occurs on the server, the user can submit this ID to Backend Background to locate the problem.
MsgInfoList	Array	A list of message information, with each element being the specific information of a message.

MsgInfoList is defined as follows:

Parameter name	Type	Description
MsgBody	String	The message body of consumption.
MsgId	String	The consumed message uniquely identifies the ID.
ReceiptHandle	String	Each consumption returns a unique Message Handler, which is used to delete consumption. Only the handle generated by the last consumption of the message can be used to delete the message. And the validity period is visibilityTimeout, that is, the period of time for which hidden is taken out. After that time, the handle becomes invalid.
EnqueueTime	Int	The time when consumption is produced and enters the queue. Returns Unix timestamp, accurate to seconds.
FirstDequeueTime	Int	Reserved field.

NextVisibleTime	Int	The next time the message is visible (can be consumed again). Returns Unix timestamp, accurate to seconds.
DequeueCount	Int	Reserved field.

Error Codes

Please check Common error code .

Example

Enter:

```
Https://domain/v2/index.php?Action=BatchReceiveMessage & queueName=test-queue-123 &
```

Output:

```
{
    "code": 0,
    "message": "",
    "requestId": "14534664555",
    "msgInfoList":
    [
        {
            "msgBody": "helloworld1",
            "msgId": "123345346",
            "receiptHandle": "283748239349283",
            "enqueueTime": 1462351990,
            "firstDequeueTime": 1462352990,
            "nextVisibleTime": 1462352999,
            "dequeueCount": 2
        },
        {
            "msgBody": "helloworld2",
            "msgId": "1233453456",
            "receiptHandle": "28374345763283",
            "enqueueTime": 1462351990,
            "firstDequeueTime": 1462352990,
            "nextVisibleTime": 1462352999,
            "dequeueCount": 2
        }
    ]
}
```

Delete message

Last updated：2024-12-18 17:45:19

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (DeleteMessage) is used to delete messages that have been consumed. Consumers need to use the ReceiptHandle obtained after the last consumption as a parameter to locate the message to be deleted.

This operation can only be successful before the NextVisibleTime; if the NextVisibleTime, message changes back to the Active state, the ReceiptHandle will become invalid, the deletion will fail, and the new ReceiptHandle needs to be re-consumed. When Concurrence consumes, if the message is deleted by one of the consumers, the other consumers will no longer be able to get the deleted message.

Domain name requested by public network API: `https://cmq-queue-{$region}.api.qcloud.com`

Domain name requested by private network API: `http://cmq-queue-{$region}.api.tencentyun.com`

The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Financial), szjr (Shenzhen Financial), hk (China Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Note:

At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
QueueName	Yes	String	Queue name, which is unique under the same account in a single region. The queue name is a string of no more than 64 characters,

			must begin with a letter, and the rest can contain letters, numbers, and dashes (-).
ReceiptHandle	Yes	String	The last consumption returned a unique Message Handler, which is used to delete messages.

Output Parameter

Parameter name	Type	Description
Code	Int	0: indicates success. Others: error. For more information, please see Common error code .
Message	String	Error message.
RequestId	String	Request ID generated by the server. When an internal error occurs on the server, the user can submit this ID to Backend Background to locate the problem.

Error Codes

Please check Common error code .

Example

Enter:

```
Https://domain/v2/index.php?Action=DeleteMessage & queueName=test-queue-123 & recei
```

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555"
}
```

Bulk delete messages

Last updated：2024-12-18 17:45:19

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (BatchDeleteMessage) is used to delete messages that have been consumed in batches (a maximum of 16 messages can be deleted at a time). Consumers need to use the ReceiptHandle obtained after the last consumption as a parameter to locate the messages to be deleted.

This operation can only be successful before the NextVisibleTime; if the NextVisibleTime, message changes back to the Active state, the ReceiptHandle will become invalid, the deletion will fail, and the new ReceiptHandle needs to be re-consumed. When Concurrence consumes, if the message is deleted by one of the consumers, the other consumers will no longer be able to get the deleted message.

Domain name requested by public network API: `https://cmq-queue-{$region}.api.qcloud.com`

Domain name requested by private network API: `http://cmq-queue-{$region}.api.tencentyun.com`

The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Financial), szjr (Shenzhen Financial), hk (China Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Note:

At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
QueueName	Yes	String	Queue name, which is unique under the same account in a single region. The queue name is a string of no more than 64 characters,

			must begin with a letter, and the rest can contain letters, numbers, and dashes (-).
ReceiptHandle.n	Yes	String	Message Handler returned when the message was last consumed. For the convenience of users, n can start with 0 or 1, but it must be continuous, such as deleting two messages, which can be (receiptHandle.0,receiptHandle.1) or (receiptHandle.1, receiptHandle.2).

Output Parameter

Parameter name	Type	Description
Code	Int	0: successful, 4420: maximum QPS limit reached, 4440: queue does not exist, 6010: delete message part failed, 6020: delete message all failed. For more information on the meaning of other return values, please see Error Codes .
Message	String	Error message.
RequestId	String	Request ID generated by the server. When an internal error occurs on the server, the user can submit this ID to Backend Background to locate the problem.
ErrorList	Array	Error list that could not be deleted successfully. Each element lists the error and reason why the message could not be successfully deleted.

ErrorList is defined as follows:

Parameter name	Type	Description
Code	Int	0: indicates success. Others: error. For more information, please see Common error code .
Message	String	Error message.
ReceiptHandle	String	Corresponding to the failed Message Handler deletion.

Error Codes

Please check Common error code .

Example

Enter:

```
Https://domain/v2/index.php?Action=BatchDeleteMessage & queueName=test-queue-123 &
```

Output (when all are successful):

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555"
}
```

Output (in case of partial failure):

```
{
"code" : 6010,
"message" : "delete message partially failed",
"requestId":"14534664555",
"errorList":
[
{
"code" : 4430,
"message" : "invalid receiptHandle",
"receiptHandle":"4364564575"
}
]
}
```

Topic Model

Topic APIs

Creating a Topic

Last updated：2024-12-18 17:45:19

1. API Description

This API (CreateTopic) is used to create a new topic under the user's account.
Domain for public network API request: `cmq-topic-region.api.qcloud.com`
Domain for private network API request: `cmq-topic-region.api.tencentyun.com`

Note:

Whenever (including during alpha test) any public network downstream traffic is generated from the use of a public network domain, a fee will be charged. It is strongly recommended that users whose services are on the Tencent Cloud use **private network** domains, because no fee will be charged for the traffic consumed in the private network.
region should be replaced with a specific region: gz (Guangzhou), sh (Shanghai), or bj (Beijing). The region value in the common parameters should be consistent with the region value of the domain. If there is an inconsistency, the request will be sent to the region specified by the domain.
Public network domain requests support both http and https. Private network requests only support http.
Some of the input parameters are optional. If not specified, the default value will be taken.
All the output parameters will be returned to the user when the request is successful; otherwise, at least code, message, and requestId will be returned.

2. Input Parameters

The following request parameter list only provides API request parameters. For other parameters, refer to Common Request Parameters.

Parameter Name	Required	Type	Description
topicName	Yes	String	Topic name. It is unique under the same account in one region. The topic name is a string of no more than 64 characters, which can contain letters, numbers, and hyphens (-) and must begin with a letter.
maxMsgSize	No	Int	Maximum message length. Value range is 1024-65536 Bytes (1-64 K). Default is 65536.

filterType	No	Int	It is used to specify a message matching policy for a topic: filterType = 1 or null indicates that filterTag is used to filter messages for all the subscriptions under the topic; filterType = 2 indicates that bindingKey is used for filtering. Note: This parameter cannot be changed once configured.

3. Output Parameters

Parameter Name	Type	Description
code	Int	0: Succeeded, others: Error. See the table below for specific errors.
message	String	Error message.
requestId	String	Request ID generated by server. When there is an internal error on the server, users can submit this ID to the backend to locate the problem.
topicId	String	Unique ID of a topic. Please note that Cloud APIs are called through name instead of ID.

4. Error Codes

Error Code	Module Error Code	Error Message	Description
4460	10550	topic is already existed	A topic with the same name exists under the same account.
4000	10590	topic name format error	The format of the topic name is incorrect.
4450	10610	number of topics has reached the limit	The number of topics has reached the limit. Currently, the limit is 1,000.
6040	10660	It will take some time to release resources of previous topic before you can create a new topic with the same name. Please try later.	Failed to create a topic with the same name. It will take some time to release resources of the previous topic with the same name. Currently, to ensure data consistency of CMQ, it is not allowed to create a new topic with the same name within ten seconds after a topic is deleted.

Note: The error codes listed in the above table are specific to this API.

5. Example

Input:

```
https://domain/v2/index.php?Action=CreateTopic
&topicName=test-topic-123
&<Common Request Parameters>
```

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555",
"topicId":"topic-ajksdfasdowe"
}
```

Modify Topic Attributes

Last updated：2024-12-18 17:45:19

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (SetTopicAttributes) is used to modify the attributes of message topic.
Domain name requested by public network API: `https://cmq-topic-{$region}.api.qcloud.com`
Domain name requested by private network API: `http://cmq-topic-{$region}.api.tencentyun.com`
The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Financial), szjr (Shenzhen Financial), hk (China Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Note:

At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
TopicName	Yes	String	Topic's name is unique under the same account in a single region. The name of topic is a string of no more than 64 characters, which must begin with a letter, and the rest can contain letters, numbers and underscores (-).
MaxMsgSize	No	Int	Max Message Length. Value range: 1024-65536 Byte (that is, 1-64K). The default value is 65536.

Output Parameter

Parameter name	Type	Description
Code	Int	0: indicates success. 4440: topic does not exist. The meaning of other return values can be referenced. Error Codes .
Message	String	Error message.
RequestId	String	Request ID generated by the server. When an internal error occurs on the server, the user can submit this ID to Backend Background to locate the problem.

Example

Enter:

```
https://domain/v2/index.php?Action=SetTopicAttributes
&topicName=test-Topic-123
&maxMsgSize=1024
&<Common request parameters>
```

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555"
}
```

Get topic list

Last updated：2024-12-18 17:45:19

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (ListTopic) is used to list the list of topic under a user's account, and the data can be obtained by page.
Domain name requested by public network API: `https://cmq-topic-{$region}.api.qcloud.com`
Domain name requested by private network API: `http://cmq-topic-{$region}.api.tencentyun.com`
The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Financial), szjr (Shenzhen Financial), hk (China Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Note:

At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
SearchWord	No	String	Used to filter topic list, Backend Background uses fuzzy matching to return eligible topic list. If this parameter is left empty, all topic under the account will be returned by default.
Offset	No	Int	When paging, this page gets the starting position of topic's list. If Enter reaches this value, Enter limit must also be required. When this value defaults, Backend Background takes the default value of 0

Limit	No	Int	This page gets the number of topic when paging. If this parameter is not passed, the parameter defaults to 20, and the maximum value is 50.

The meanings of offset and limit are the same as those of offset and limit of SQL.

Output Parameter

Parameter name	Type	Description
Code	Int	Please see Error Codes Document
Message	String	Error message
RequestId	String	Request ID generated by the server. When an internal error occurs on the server, the user can submit this ID to Backend Background to locate the problem.
TotalCount	Int	The total number of topic returned by this request under the user account, rather than the number of topic obtained on this page after paging.
TopicList	Array	Topic list information. Each element represents a topic's information.

TopicList is defined as follows:

Parameter name	Type	Description
TopicId	String	Topic's unique logo ID
TopicName	String	Topic's name is unique under the same account in a single region. Topic's name is a string of no more than 64 characters, which must begin with a letter, and the rest can contain letters, numbers and underscores. `–` ## Example

Enter:

```
https://domain/v2/index.php?Action=ListTopic
&searchWord=test
&<Common request parameters>
```

Output:

```
{
    "code": 0,
    "message": "",
    "requestId": "14534664555",
    "totalCount": 10,
    "topicList": [{
            "topicId": "topic-asdfo",
            "topicName": "test-topic1"
        },
        {
            "topicId": "topic-asdsafo",
            "topicName": "topic-test1"
        }
    ]
}
```

Get topic attribute

Last updated：2024-12-18 17:45:20

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (GetTopicAttributes) is used to obtain the attributes of a created topic. In addition to the settable property set when topic was created, you can also get the creation time of topic, the time of the last modification of topic attribute, and the Statistics (approximate value) of the message in topic.

Domain name requested by public network API: `https://cmq-topic-{$region}.api.qcloud.com`

Domain name requested by private network API: `http://cmq-topic-{$region}.api.tencentyun.com`

The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Financial), szjr (Shenzhen Financial), hk (China Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Note:

At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters Document.

Parameter name	Required	Type	Description
TopicName	Yes	String	Topic's name is unique under the same account in a single region. Topic's name is a string of no more than 64 characters, which must begin with a letter, and the rest can contain letters, numbers and underscores. `-` ## Output Parameter

Parameter name	Type	Description
Code	Int	0: indicates success. 4440: topic does not exist. The meaning of other return values can be referenced. Error Codes Document
Message	String	Error message
RequestId	String	Request ID generated by the server. When an internal error occurs on the server, the user can submit this ID to Backend Background to locate the problem.
MsgCount	Int	Current number of messages in this topic (number of Message Retention)
MaxMsgSize	Int	Max Message Length. Value range: 1024-1048576Byte (that is, 1-1024K). The default value is 65536.
MsgRetentionSeconds	Int	The longest survival time of a message in topic will be deleted in seconds after the time specified by this parameter has elapsed since it was sent to topic, regardless of whether the message was successfully pushed to the user. Fixed to one day (86400 seconds), this property cannot be modified
CreateTime	Int	The founding time of topic. Return Unix timestamp, accurate to seconds
LastModifyTime	Int	The time when the topic attribute was last modified. Returns Unix timestamp, accurate to seconds.
FilterType	Int	Describes the filtering policy that the user chooses when creating a subscription: FilterType = 1 indicates that the user uses filterTag tag filtering FilterType = 2 indicates that the user uses bindingKey filtering.
CreateUin	Int	The creator Uin,CAM authentication resource is composed of this field.
QPS	Int	The number of publish messages per second.
TopicId	String	Topic's ID.

Example

Enter:

```
https://domain/v2/index.php?Action=GetTopicAttributes
&topicName=test-topic-123
&<Common request parameters>
```

Output:

```
{
    "code": 0,
    "message": "",
    "codeDesc": "Success",
    "requestId": "2074225773",
    "msgCount": 0,
    "maxMsgSize": 65536,
    "msgRetentionSeconds": 86400,
    "createTime": 1563368086,
    "lastModifyTime": 1564576253,
    "createUin": 100008950635,
    "qps": 5000,
    "topicId": "topic-kalspsqc",
    "filterType": 1,
    "tags": []
}
```

Delete topic

Last updated：2024-12-18 17:45:19

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (DeleteTopic) is used to delete a created topic.
Domain name requested by public network API: `https://cmq-topic-{$region}.api.qcloud.com`
Domain name requested by private network API: `http://cmq-topic-{$region}.api.tencentyun.com`
The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Financial), szjr (Shenzhen Financial), hk (China Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Note:

At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
TopicName	Yes	String	Topic's name is unique under the same account in a single region. The name of topic is a string of no more than 64 characters, which must begin with a letter, and the rest can contain letters, numbers and underscores (-).

Output Parameter

Parameter name	Type	Description
Code	Int	0: indicates success. Others: error. For more information, please see Common error code .
Message	String	Error message.
RequestId	String	Request ID generated by the server. When an internal error occurs on the server, the user can submit this ID to Backend Background to locate the problem.

Error Codes

Please check Common error code .

Example

Enter:

```
https://domain/v2/index.php?Action=DeleteTopic
&TopicName=test-Topic-123
&<Common request parameters>
```

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555"
}
```

Message APIs

Publish news

Last updated：2024-12-18 17:45:20

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (PublishMessage) is used to send a message from publish to the specified topic.

Domain name requested by public network API: `https://cmq-topic-{$region}.api.qcloud.com`

Domain name requested by private network API: `http://cmq-topic-{$region}.api.tencentyun.com`

The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Financial), szjr (Shenzhen Financial), hk (China Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Note:

At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
TopicName	Yes	String	Topic's name is unique under the same account in a single region. The name of topic is a string of no more than 64 characters, which must begin with a letter, and the rest can contain letters, numbers and underscores (-).
MsgBody	Yes	String	Message body. At least 1 byte, the maximum length is limited by the

			set topic message maximum length attribute.
MsgTag.n	No	String	Message filter tag. Message label (for message filtering). The number of tags cannot exceed 5, and each label cannot exceed 16 characters. and Subscribe The filterTag parameter of the API is used together. Rules: (1) if filterTag is not set, subscriptions receive all messages from publish to Topic, regardless of whether msgTag is set or not. (2) if the filterTag array has a value, the subscription will receive the message from publish to Topic only if at least one value in the array also exists in the msgTag array (that is, there is an intersection between filterTag and msgTag). (3) if the filterTag array has a value, but msgTag is not set, then no messages from publish to Topic will be received, which can be regarded as a special case of (2). In this case, filterTag and msgTag do not intersect. The overall design idea of the rules is based on the wishes of the subscribers.
RoutingKey	No	String	Length < = 64 bytes, this field is used to indicate the routing path to send the message, with a maximum of 15 ".", that is, a maximum of 16 phrases. RoutingKey cannot be arbitrarily specified when a message is sent to a exchange of type topic. To meet the above format requirements, a message with routingKey specified by the subscriber will be pushed to all consumers with which the BindingKey can match. This matching situation has two relationships: 1 * (asterisk), which can replace a word (a continuous string of letters); 2 # (pound sign): can match one or more characters.

Output Parameter

Parameter name	Type	Description
Code	Int	0: indicates success. Others: error. For more information, please see Common error code .
Message	String	Error message.
RequestId	String	When an internal server error occurs in the request ID, generated by the server, the user can submit this ID to Backend Background to locate the problem.
MsgId	String	The server generates a unique identity ID for the message.

Error Codes

Please check Common error code .

Example

Enter:

```
https://domain/v2/index.php?Action=PublishMessage
&topicName=test-topic-123
&msgBody=helloworld
&<Common request parameters>
```

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555",
"msgId":"123345346"
}
```

Batch publish messages

Last updated：2025-02-12 10:58:54

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (BatchPublishMessage) is used to send publish batch messages (up to 16 messages) to the specified topic.
Domain name requested by public network API: `https://cmq-topic-{$region}.api.qcloud.com`
Domain name requested by private network API: `http://cmq-topic-{$region}.api.tencentyun.com`
The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Financial), szjr (Shenzhen Financial), hk (China Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Note:

At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
TopicName	Yes	String	Topic's name is unique under the same account in a single region. The name of topic is a string of no more than 64 characters, which must begin with a letter, and the rest can contain letters, numbers and underscores (-).
MsgBody.n	Yes	String	Message body. Represents a message in this batch. Currently, the number of messages in batch cannot exceed 16.

			For the convenience of users, n can start with 0 or 1, but it must be continuous, such as sending two messages, which can be (msgBody.0, msgBody.1) or (msgBody.1, msgBody.2). Note: since the total size of all messages (excluding headers and other parameters, only msgBody) is limited to 64k, it is recommended to plan the number of batches to be sent in advance.
MsgTag.n	No	String	Message filter tag. Message label (for message filtering). The number of tags cannot exceed 5, and each label cannot exceed 16 characters. and Subscribe The filterTag parameter of the API is used together. Rules: (1) if filterTag is not set, subscriptions receive all messages from publish to Topic regardless of whether msgTag is set or not. (2) if the filterTag array has a value, the subscription will receive the message from publish to Topic only if at least one value in the array also exists in the msgTag array (that is, there is an intersection between filterTag and msgTag). (3) if the filterTag array has a value, but msgTag is not set, then no messages from publish to Topic will be received, which can be regarded as a special case of (2). In this case, filterTag and msgTag do not intersect. The overall design idea of the rules is based on the wishes of the subscribers.
RoutingKey	No	String	Length < = 64 bytes, this field is used to indicate the routing path to send the message, with a maximum of 15 ".", that is, a maximum of 16 phrases. When a message is sent to a exchange of type topic, it is not optional to specify that the routingKey, needs to meet the above format requirements. A message with routingKey specified by the subscriber will be pushed to all consumers with which the BindingKey can match, which is related in two ways: 1 * (asterisk): can replace a word (a continuous string of letters); 2 # (pound sign): can match one or more characters.

Output Parameter

Parameter name	Type	Description
Code	Int	0: indicates success. Others: error. For more information, please see Common error code .
Message	String	Error message.

RequestId	String	When an internal server error occurs in the request ID, generated by the server, the user can submit this ID to Backend Background to locate the problem.
MsgList	Array	The server generates a unique identification ID list of messages, and each element is the information of a message.

MsgList is defined as follows:

Parameter name	Type	Description
MsgId	String	The server generates a unique identity ID for the message.

Error Codes

Please check Common error code .

Example

Enter:

```
https://domain/v2/index.php?Action=BatchPublishMessage
&topicName=test-topic-123
&msgBody.1=helloworld1
&msgBody.2=helloworld2
&<Common request parameters>
```

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555",
"msgList":
  [
    {
      "msgId":"123345346"
    },
    {
       "msgId":"456436346"
    }
  ]
}
```

Delivering Message

Last updated：2024-12-18 17:45:20

Queue Endpoint Subscription

CMQ can publish the message text of a topic and deliver it to the subscribed queue, so consumers can read it from the queue.

HTTP Endpoint Subscription

Delivery description

CMQ can push topic messages to the subscribed HTTP endpoint by sending POST requests. The message can be in JSON or SIMPLIFIED format.
JSON format: the body of the pushed HTTP request contains the message body and attribute information. **The `Content-type` is `text/plain` .**
SIMPLIFIED format: the body of the pushed HTTP request is the message body, while information such as `msgId` will be delivered to the subscriber in the HTTP request header.
If the HTTP service of the subscriber returns a standard 2xx response code (such as 200), the delivery succeeded; otherwise, the delivery failed, and the delivery retry policy has been triggered. If no response is returned after the timeout period elapses, CMQ will also consider the request as a failure and trigger the delivery retry policy. The detection timeout period is about 15 seconds.

Header of HTTP delivery request

Parameter Name	Description
x-cmq-request-id	`requestId` of the current push message
x-cmq-message-id	`msgId` of the current push message
x-cmq-message-tag	Tag of the current push message

Body of HTTP delivery request

The body of an HTTP request in JSON format contains the message body and attribute information.

Parameter Name	Type	Description
TopicOwner	String	`APPID` of the owner of the subscribed topic

topicName	String	Topic name
subscriptionName	String	Subscription name
msgId	String	Message ID
msgBody	String	Message body
publishTime	Int	Message published time

The body of an HTTP request in SIMPLIFIED format is the message body published by the publisher.

Response of HTTP delivery request

If a 2xx response code is returned, the HTTP service of the subscriber has normally processed the delivered message; if another response code is returned or no response is returned after the timeout period elapses, an error will be reported, and the delivery retry policy will be triggered.

Sample request

In this sample, the subscribed HTTP endpoint is `http://test.com/cgi` .

JSON format:

```
POST /cgi HTTP/1.1
Host: test.com
Content-Length: 761
Content-Type: text/plain
User-Agent: Qcloud Notification Service Agent
x-cmq-request-id: 2394928734
x-cmq-message-id: 6942316962
x-cmq-message-tag: a, b

{"TopicOwner":100015036,"topicName":"MyTopic","subscriptionName":"mysubscription","msgId":"6942316962","msgBody":"test message","publishTime":11203432}
```

SIMPLIFIED format:

```
POST /cgi HTTP/1.1
Host: test.com
Content-Length: 123
Content-Type: text/plain
User-Agent: Qcloud Notification Service Agent
x-cmq-request-id: 2394928734
x-cmq-message-id: 6942316962
x-cmq-message-tag: a, b
```

```
test message
```

Sample message subscription

The following is a message subscription demo, where all messages are delivered in the POST method; therefore, you only need to rewrite the `do_POST` method.This sample will print the received HTTP POST request content and deserialize `post data json` to traverse the printed request data.

```
#!/usr/bin/python
from BaseHTTPServer import HTTPServer, BaseHTTPRequestHandler
import json
class TestHTTPHandle(BaseHTTPRequestHandler):
    def do_POST(self):
        content_len = int(self.headers.getheader('content-length',0))
        post_body = self.rfile.read(content_len)
        print "receive cmq topic publisher request:"
        print self.headers
        print post_body
        post_data = json.loads(post_body)
        for k,v in post_data.iteritems():
                print "key:%s value:%s" % (k,v)
        #response http status 200
        self.send_response(200)
        self.end_headers()
        self.wfile.write('ok')
def start_server(port):
        http_server = HTTPServer(('0.0.0.0', int(port)),TestHTTPHandle)
        http_server.serve_forever()
if __name__ == '__main__':
        start_server(80)
```

Subscription APIs

Create a subscription

Last updated：2024-12-18 17:47:44

1. API Description

This API (Subscribe) is used to create a new subscription under one of a user's topics.

Domain for public network API request: `cmq-topic-region.api.qcloud.com`

Domain for private network API request: `cmq-topic-region.api.tencentyun.com`

Note:

At any time (including alpha test), any downstream traffic generated when using public network domain will incur traffic fee. It is strongly recommended that users on Tencent Cloud use **private network** domain, as it will not incur any traffic fee.

2. Input Parameters

The following request parameter list only provides API request parameters. For other parameters, refer to Common Request Parameters.

Parameter Name	Required	Type	Description
topicName	Yes	String	Topic name. It is unique under the same account in a single region. Topic name is a string with no more than 64 characters. It must start with letter, and the rest may contain letters, numbers and dashes (-).
subscriptionName	Yes	String	Subscription name. It is unique for the same topic under the same account in a single region. Subscription name is a string with no more than 64 characters. It must start with letter, and the rest may contain letters, numbers and dashes (-).
protocol	Yes	String	Subscription protocol. Currently, two types of protocols are supported: HTTP, queue. When using HTTP protocol, users need to build web server to receive messages on their own. When using queue, messages will be automatically pushed to the CMQ queue and users can pull the messages concurrently.
notifyStrategy	No	String	Retry policy for the CMQ push server in case of errors when pushing messages to the endpoint. Available values are: 1)

			BACKOFF_RETRY (backoff retry). Re-push the message at regular intervals. Discard the message after a certain number of reties have been committed, then continue pushing the next message; 2) EXPONENTIAL_DECAY_RETRY (exponential decay retry). The interval between retries is increased exponentially, for example, retry in 1s first, then 2s, 4s, 8s...Since the cycle of a Topic message is one day, the retry operations will last for at most one day before the message is discarded. Default is EXPONENTIAL_DECAY_RETRY.
notifyContentFormat	No	String	Notification content format. Available values are: 1) JSON; 2) SIMPLIFIED, i.e. raw format. If the protocol is queue, the value must be SIMPLIFIED. If the protocol is HTTP, both values are available. Default is JSON.
filterTag.n	No	String	Message body. Message tag (used to filter message). There can be up to 5 tags, each containing no more than 16 characters. It is used in combination with the msgTag parameter of (Batch)PublishMessage. Rules: 1) If filterTag is not configured, the subscribers will receive all the messages published on Topic no matter whether msgTag is configured or not; 2) If filterTag array is configured with values, the subscribers will receive messages published on Topic only if at lease one of the values in the array exists in the msgTag array as well (that is, filterTag and msgTag intersect with each other); 3) If filterTag array is configured with values but msgTag is not configured, the subscribers will not receive any messages published on Topic. This can be considered a special case of 2), when there is no intersection between filterTag and msgTag. The design concept of the rules is based on the wills of the subscribers.
BindingKey	No	String	This field indicates the filtering policy for subscribing to and receiving messages. There can be up to 5 `BindingKey` , and each of them can contain up to 64 bytes, which can have up to 15 `.` , i.e., up to 16 phrases.

3. Output Parameters

Parameter Name	Type	Description
code	Int	0: Succeeded, others: Error. For detailed errors, please refer to the table below.

message	String	Error message.
requestId	String	ID of the request generated by server. When there is an internal error on the server, users can submit this ID to backend to locate the problem.
codeDesc	String	Error message description.

Error Code	Module Error Code	Error Message	Description
4490	10470	subscription is already existed	A subscription with the same name already exists under the same Topic of the same account.
4500	10480	number of subscription has reached the limit	The number of subscriptions under the same Topic has exceeded the limit. Currently, the limit is 100.
4000	10490	number of filterTag exceed limit	The number of filterTags exceeded the limit. Currently, the limit is 5.
4000	10500	endpoint format error	Incorrect endpoint format. Possible errors include: 1) URL contains space(s); 2) URL does not start with "http://" for http protocol; 3) Invalid URL; 4) protocol does not match with endpoint.
4000	10510	undefined protocol	Undefined protocol. Please check for spelling errors.
4000	10520	undefined notify retry stragety	Undefined push notification retry policy. Please check for spelling errors.
4000	10530	undefined notify content format	Undefined push notification format. Please check for spelling errors.
4510	10570	url connot contain any blank characters	URL cannot contain any blank characters.
4000	10580	subscription name format error	Incorrect subscription name format.
4000	10620	subscription name format error	Incorrect subscription name format.
4000	10630	illegal endpoint	Invalid endpoint.
4000	10640	notifyContentFormat	If the protocol field is "queue", notifyContentFormat must

		of protocol queue must be SIMPLIFIED	be SIMPLIFIED.
6050	10740	too many filterTag or bindingKey	There are too many filterTag or bindingKey, please check the parameter configuration.
4000	10710	parameters lack of bindingKey	bindingKey is missing.
4000	10670	too many filterTag	There are too many filterTag, please check the number of parameters.
4000	10680	too many bindingKey	There are too many bindingKey, please check the number of parameters.

Note: The error codes listed in the above table are specific to the API. If the error code you are looking for is not here, you may find it in the Common Error Codes.

4. Example

Input:

```
https://domain/v2/index.php?Action=Subscribe
&topicName=test-topic-123
&subscriptionName=test-subscription-123
&protocol=http
&endpoint=http://your_host/your_path
&<Common request parameters>
```

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555"
}
```

Get subscription list

Last updated：2024-12-18 17:47:44

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (ListSubscriptionByTopic) is used to list subscriptions under a certain topic of a user, and the data can be obtained by page.

Domain name requested by public network API: `https://cmq-topic-{$region}.api.qcloud.com`

Domain name requested by private network API: `http://cmq-topic-{$region}.api.tencentyun.com`

The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Financial), szjr (Shenzhen Financial), hk (China Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Note:

At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
TopicName	Yes	String	Topic's name is unique under the same account in a single region. The name of topic is a string of no more than 64 characters, which must begin with a letter, and the rest can contain letters, numbers and underscores (-).
SearchWord	No	String	Used to filter the subscription list, Backend Background uses fuzzy matching to return the subscription list that meets the criteria. If this

			parameter is left empty, all subscriptions under the account will be returned by default.
Offset	No	Int	This page gets the starting position of the subscription list when paging. If Enter reaches this value, Enter limit must also be required. When this value defaults, Backend Background takes the default value of 0. Value range: 0-1000.
Limit	No	Int	This page gets the number of subscriptions when paging. The value range of this parameter is 0-100. If the parameter is not passed, it defaults to 20.

The meanings of offset and limit are the same as those of offset and limit of SQL.

Output Parameter

Parameter name	Type	Description
Code	Int	Please see Error Codes
RequestId	String	Request ID generated by the server. When an internal error occurs on the server, the user can submit this ID to Backend Background to locate the problem.
TotalCount	Int	The total number of topic returned by this request under the user account, rather than the number of topic obtained on this page after paging.
SubscriptionList	Array	Topic list information, each element represents a topic information.

SubscriptionList is defined as follows:

Parameter name	Type	Description
SubscriptionId	String	Subscribe to ID. Subscription ID is used when pulling monitoring data.
SubscriptionName	String	The subscription name is unique under the same topic of the same account in a single region. The subscription name is a string of no more than 64 characters, must begin with a letter, and the rest can contain letters, numbers, and dashes (-).
Protocol	String	Protocol, who subscribes to, currently supports two kinds of Protocol: HTTP and queue. With HTTP Protocol, users need to build their own Web Server to

		accept messages. Using queue, messages will be automatically pushed to CMQ queue, users can pull messages Concurrence.
Endpoint	String	The endpoint, that receives the notification is distinguished according to Protocol protocol: for HTTP,endpoint, you must use `http://` At the beginning, host can be a domain name or IP.

Example

Enter:

```
https://domain/v2/index.php?Action=ListSubscriptionByTopic
&searchWord=abc
&<Common request parameters>
```

Output:

```
{
    "code":0,
    "message":"",
    "requestId":"14534664555",
    "totalCount":2,
    "subscriptionList":[
        {
            "subscriptionId":"subsc-sdkfl",
            "subscriptionName":"test-sub2",
            "protocol":"http",
            "endpoint":"http://testhost/testpath"
        },
        {
            "subscriptionId":"subsc-34lasdk",
            "subscriptionName":"test-sub1",
            "protocol":"queue",
            "endpoint":"test-queue-name"
        }
    ]
}
```

Modify subscription properties

Last updated：2024-12-18 17:47:44

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (SetSubscriptionAttributes) is used to modify the attributes of message subscriptions.
Domain name requested by public network API: `https://cmq-topic-{$region}.api.qcloud.com`
Domain name requested by private network API: `http://cmq-topic-{$region}.api.tencentyun.com`
The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Financial), szjr (Shenzhen Financial), hk (China Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Note:

At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
TopicName	Yes	String	Topic's name is unique under the same account in a single region. The name of topic is a string of no more than 64 characters, which must begin with a letter, and the rest can contain letters, numbers and underscores (-).
SubscriptionName	Yes	String	The subscription name is unique under the same topic of the same account in a single region. The subscription name is a string of no more than 64 characters, must begin with a letter, and the rest can contain letters, numbers, and dashes (-).

NotifyStrategy	No	String	The retry policy of the endpoint push server when an error occurs in the push message to CMQ. The values are as follows: (1) BACKOFF_RETRY, Backoff retry. Try again at regular intervals, and after retrying a certain number of times, discard the message and continue to push the next message. (2) EXPONENTIAL_DECAY_RETRY, index decline retry. The interval of each retry increases exponentially, for example, at the beginning of 1s, followed by 2sfocus 4s, 8s because the period of the Topic message is one day, so the message will be discarded for up to one day. The default value is EXPONENTIAL_DECAY_RETRY.
NotifyContentFormat	No	String	The format of the push content. Value: (1) JSON; (2) SIMPLIFIED, is raw format. If protocol is queue, the value must be SIMPLIFIED. If protocol is HTTP, both values can be used, and the default value is JSON.
FilterTag.n	No	String	Message body. Message label (for message filtering). The number of tags cannot exceed 5, and each label cannot exceed 16 characters. and (Batch) PublishMessage The msgTag parameter is used in conjunction with the following rules: (1) if filterTag is not set, subscriptions receive all messages from publish to Topic, regardless of whether msgTag is set or not; (2) if the filterTag array has a value, the subscription will receive the message from publish to Topic only if at least one value in the array also exists in the msgTag array (that is, there is an intersection between filterTag and msgTag). (3) if the filterTag array has a value, but msgTag is not set, then no messages from publish to Topic will be received, which can be regarded as a special case of (2). In this case, filterTag and msgTag do not intersect. The overall design idea of the rules is based on the wishes of the subscribers.
BindingKey.n	Yes	String array	The number of bindingKey does not exceed 5, and each bindingKey is no more than 64 bytes long. This field represents the filtering policy for subscribing to receive messages. Each bindingKey contains a maximum of 15 ".", that is, a maximum of 16 phrases.

Output Parameter

Parameter	Type	Description

name		
Code	Int	0: indicates success. Others: error. For more information, please see Common error code .
Message	String	Error message.
RequestId	String	Request ID generated by the server. When an internal error occurs on the server, the user can submit this ID to Backend Background to locate the problem.

Error Codes

Please check Common error code .

Example

Enter:

```
https://domain/v2/index.php?Action=SetSubscriptionAttributes
&topicName=test-Topic-123
&subscriptionName=test-subscription-123
&filterTag.0=football
&<Common request parameters>
```

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555"
}
```

Delete Subscription

Last updated：2024-12-18 17:47:44

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (Unsubscribe) is used to delete a created subscription.
Domain name requested by public network API: `https://cmq-topic-{$region}.api.qcloud.com`
Domain name requested by private network API: `http://cmq-topic-{$region}.api.tencentyun.com`
The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Financial), szjr (Shenzhen Financial), hk (China Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Note

At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
TopicName	Yes	String	Topic's name is unique under the same account in a single region. The name of topic is a string of no more than 64 characters, which must begin with a letter, and the rest can contain letters, numbers and underscores (-).
SubscriptionName	Yes	String	The subscription name is unique under the same topic of the same account in a single region. The subscription name is a string of no more than 64 characters, must begin with a letter, and the rest can contain letters, numbers, and dashes (-).

Output Parameter

Parameter name	Type	Description
Code	Int	0: indicates success. 4440: topic does not exist. The meaning of other return values can be referenced. Error Codes .
Message	String	Error message.
RequestId	String	Request ID generated by the server. When an internal error occurs on the server, the user can submit this ID to Backend Background to locate the problem.

Example

Enter:

```
https://domain/v2/index.php?Action=Unsubscribe
&topicName=test-Topic-123
&subscriptionName=test-subscription-123
&<Common request parameters>
```

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555"
}
```

Get subscription properties

Last updated：2024-12-18 17:47:44

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (GetSubscriptionAttributes) is used to obtain the attributes of a created subscription.
Domain name requested by public network API: `https://cmq-topic-{$region}.api.qcloud.com`
Domain name requested by private network API: `http://cmq-topic-{$region}.api.tencentyun.com`
The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Financial), szjr (Shenzhen Financial), hk (China Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.

Note

At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
TopicName	Yes	String	Topic's name is unique under the same account in a single region. The name of topic is a string of no more than 64 characters, which must begin with a letter, and the rest can contain letters, numbers and underscores (-).
SubscriptionName	Yes	String	The subscription name is unique under the same topic of the same account in a single region. The subscription name is a string of no more than 64 characters, must begin with a letter, and the rest can contain letters, numbers, and dashes (-).

Output Parameter

Parameter name	Type	Description
Code	Int	0: indicates success. For more information on the meaning of other return values, please see Error Codes .
Message	String	Error message.
RequestId	String	Request ID generated by the server. When an internal error occurs on the server, the user can submit this ID to Backend Background to locate the problem.
TopicOwner	String	The APPID of the subscription owner.
MsgCount	Int	The number of messages waiting for shipping to be subscribed.
Protocol	String	Protocol, who subscribes to, currently supports two kinds of Protocol: HTTP and queue. With HTTP Protocol, users need to build their own Web Server to accept messages. Using queue, messages will be automatically pushed to CMQ queue, users can pull messages Concurrence.
Endpoint	String	The endpoint, that receives the notification is distinguished according to Protocol's protocol: for HTTP,endpoint, it must start with "http://", host can be a domain name or IP;, enter queueName for queue,.
NotifyStrategy	String	The retry policy of the endpoint push server when an error occurs in the push message to CMQ. The values are: (1) BACKOFF_RETRY, Backoff retry. Try again at regular intervals, and after retrying for a certain number of times, discard the message and continue to push the next message. (2) EXPONENTIAL_DECAY_RETRY, index decline retry. The interval of each retry increases exponentially, for example, at the beginning of 1s, followed by 2s, 4s, 8s. Because the cycle of Topic messages is one day, at most one day of retry will discard the message. The default value is EXPONENTIAL_DECAY_RETRY.
NotifyContentFormat	String	The format of the push content. Value: (1) JSON; (2) SIMPLIFIED, is raw format. If protocol is queue, the value must be SIMPLIFIED. If protocol is HTTP, both values can be used, and the default value is JSON.
CreateTime	Int	The time when the subscription was created. Returns Unix timestamp, accurate to seconds.
LastModifyTime	Int	When the subscription property was last modified. Returns Unix timestamp, accurate to seconds.

BindingKey	String array	Represents the filtering policy for subscribing to receive messages.

Example

Enter:

```
https://domain/v2/index.php?Action=GetSubscriptionAttributes
&topicName=test-topic-123
&subscriptionName=test-subscription-123
&<Common request parameters>
```

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555",
"topicOwner":"1231884",
"msgCount":234,
"protocol":"http",
"endpoint":"http://testhost/testpath",
"notifyStrategy":"EXPONENTIAL_DECAY_RETRY",
"notifyContentFormat":"SIMPLIFIED",
"createTime":1462268960,
"lastModifyTime": 1462269960
}
```

Clear the subscription label

Last updated：2024-12-18 17:47:44

Note:

This is a legacy API which has been hidden and will no longer be updated. We recommend using the new CMQ API 3.0 which is standardized and faster.

Interface description

This API (ClearSubscriptionFilterTags) is used to clear the tags of subscriptions.
Domain name requested by public network API: `https://cmq-topic-{$region}.api.qcloud.com`
Domain name requested by private network API: `http://cmq-topic-{$region}.api.tencentyun.com`
The above **{$region} in the domain name needs to be replaced with a specific region** : gz (Guangzhou), sh (Shanghai), bj (Beijing), shjr (Shanghai Financial), szjr (Shenzhen Financial), hk (China Hong Kong), cd (Chengdu), ca (North American), usw (Maxi), sg (Singapore). The region value in the common parameters should be consistent with the region value of the domain name. If there is any inconsistency, the request will be sent to the region specified by the domain name region based on the region value of the domain name.
Note:
At any time (including during internal testing), if the public network downstream Traffic is generated by using a public network domain name, Traffic and cost will be charged. Therefore, users of the service on Tencent Cloud are strongly recommended to use it. **Private network** Domain name, private network will not produce Traffic cost.

Input Parameter

The following request parameter list only lists the API request parameters. For other parameters, please see Common Request Parameters page.

Parameter name	Required	Type	Description
TopicName	Yes	String	Topic's name is unique under the same account in a single region. The name of topic is a string of no more than 64 characters, which must begin with a letter, and the rest can contain letters, numbers and underscores (-).
SubscriptionName	Yes	String	The subscription name is unique under the same topic of the same account in a single region. The subscription name is a string of no more than 64 characters, must begin with a letter, and the rest can contain letters, numbers, and dashes (-).

Output Parameter

Parameter name	Type	Description
Code	Int	0: indicates success, others: error. The detailed error is shown in the following table.
Message	String	Error message.
RequestId	String	Request ID generated by the server. When an internal error occurs on the server, the user can submit this ID to Backend Background to locate the problem.

Example

Enter:

```
https://domain/v2/index.php?Action=ClearSubscriptionFilterTags
&topicName=test-Topic-123
&subscriptionName=test-subscription-123
&<Common request parameters>
```

Output:

```
{
"code" : 0,
"message" : "",
"requestId":"14534664555"
}
```

Error Codes

Last updated：2024-12-18 17:45:20

Function description

If the API call fails, the error code code in the final returned result is not 0. The error message field will display the detailed error message. Users can query specific error messages in the error code list according to code and message. Example:

```
{
   "code": 5100,
   "message": "(100004) incorrect projectId"
}
```

Error code list

Common error code

Error Codes	Module error code	English description	Description	Treatment measures
4000	10000	Invalid request parameters	Illegal request parameter	Please check the input parameters according to the requirements of the document.
4000	10010	Lacked of required parameters	Parameter missing	Please follow the necessary parameters of Enter in accordance with the requirements of the document.
4000	10110	Request parameters error	Incorrect parameter	Please check whether the value of the parameter is Valid according to the requirements of the document.
4000	10280	Action is not existed	The requested operation does not exist	The requested operation does not exist. Please make sure that the action is called correctly.
4000	10310	Error: parameter s	Error: parameter% s is malformed	The format of the parameter is incorrect. Please check the format of

		key format error		the input parameter according to the requirements of the document.
4000	10320	No such parameter s	No parameter s	The parameter is missing. Please confirm the validity of the parameter.
4000	10330	Parameter s is NOT a repeatable parameter	Parameter s is not repeatable	This parameter cannot be repeated. Please confirm whether the parameter is passed correctly.
4000	10350	Parameter s value or length is out of range	Parameter s value or length out of range	The parameter is out of the specified range. Please confirm the validity of the parameter.
4000	10360	Parameter s error type	Parameter s is of the wrong type	The parameter type is incorrect. Please refer to the documentation to confirm the parameter.
4000	10370	% s parameter batch size is out of range	Parameter s batch number exceeds limit	The parameters of the array exceed the limit. Please control the number of input parameters.
4000	10380	% s parameter is not consequent	Parameter s is not continuous	When parameters are passed as an array, make sure that the subscript is continuous
4000	10390	Lacked of required parameter s	Missing parameter s	The specified parameter is missing, please make sure to enter the parameter
4000	10400	Cannot find parameter s in uri	Parameter s not found	The specified parameter is missing, please make sure to enter the parameter
4000	10410	Unexpected http s only GET or POST is supported	The unexpected parameter s currently supports only GET or POST	Currently, the request method only supports GET or POST.
4000	10420	Cannot parse s, or request size is more than 1MB	Cannot process s, or the request size exceeds 1MB	The size of the request exceeds 1 MB. Please control the size of the message.

4000	10430	Action name s is not existed	Operation s does not exist	The operation you entered does not exist. Please confirm the action.
4000	10440	Account illegal, it may be an assistant account	The account is illegal. It may be a helper account.	Please check the validity of the account, or determine whether the account is a sub-account without Permission.
4000	10461	No cam authentication	No CAM Permission	There is no Permission at present. Please confirm the range of Permission.
4100	10030	Authentication failed	Authentication failed	Authentication failed. Please follow the Signature Document authentication
4100	10080	Secret id status error	SecretId status error	The key secretId status is incorrect. Please check the validity of secretId.
4100	10270	Secret id is not existed	SecretId does not exist	The secretId you specified does not exist. Please check the validity of secretId
4300	10040	Charge overdue	The account in arrears	The account has in arrears. If you need any service, please Top up.
4420	10250	Qps throttling	The frequency of API calls exceeds the limit	The frequency of calling the API exceeds the limit. Please control the calling frequency.
4480	10460	Exceed interface frequency limit, please slow down	Interface frequency limit exceeded	The frequency of calling the API exceeds the limit. Please control the calling frequency.
6000	10050	Server internal error	Service internal error	Internal error, please Submit a ticket Contact the technician to view the log

Queue common error code

Error Codes	Module error code	English description	Description	Treatment measures
4000	10450	Secret id dosen't begin with AKID	SecretId must start with AKID	SecretId must start with AKID. Please

				confirm the validity of secretId.
4000	4440 (10100)	Queue is not existed, or deleted	The queue does not exist or has been deleted	The message queue you specified does not exist. Please verify that the queue name is correct.
4000	10692	delay seconds is out of range	delay message is out of range	confirm the legal delay range and adjust the delay parameters
6040	10660	It will take some time to release resources of previous queue before you create a new one with the same name, please try later	Failed to create queue with the same name. Because you need to release resources to delete a queue with the same name just now, in order to ensure data consistency, CMQ cannot create a queue with the same name within 30 seconds after deleting the queue.	Try again later
6050	10670	Your queue cannot be rewinded	The queue cannot rewind	The queue cannot rewind

Topic common error code

Error Codes	Module error code	English description	Description	Treatment measures
4000	10490	Number of filterTag exceed limit filterTag	The quantity exceeds the limit. At present, the maximum number is five.	Please control the number of filter tags
4000	10500	Endpoint format error	Endpoint format error	Please check the format of endpoint. Possible errors are: (1) url contains spaces; (2) url of HTTP does not start with "http://"; (3) illegal

				url; (4) protocol does not correspond to endpoint.
4000	10510	Undefined protocol	Undefined Protocol	Please check to see if the spelling is wrong
4000	10540	There exists subscriptions under this topic, please unsubscribe all of them before DeleteTopic	Before deleting Topic, you must make sure that there are no subscriptions under Topic. This is to prevent erroneous deletion	Please delete all subscriptions and try again
4000	10590	(1) topic name format error (2) subscription name format error	(1) incorrect format of Topic name (2) incorrect format of subscription name	(1) Please check the Topic format (2) incorrect subscription name format
4000	10630	Illegal endpoint	Illegal endpoint	Please have the correct endpoint of Enter.
4000	10640	NotifyContentFormat of protocol queue must be SIMPLIFIED	When the protocol field is queue, notifyContentFormat must be SIMPLIFIED	Please pay attention to the value
4000	10670	Too many filterTag filterTag	The quantity is too large, please check the number of parameters	Please check the number of filter tags
4000	10710	Parameters lack of bindingKey	Missing bindingKey	Please enter bindingKey
4000	4440 (10600)	Topic is not existed, or deleted	Topic does not exist.	Topic does not exist. Please check the legitimacy of topic.
4450	10610	Number of topics has reached the limit Topic	The number has reached the limit. At present, the maximum number is 1000.	The number of topic reached the upper limit.
4490	10470	Subscription is already existed	Under the same Topic of the same account,	Please select the subscription you

			the subscription with the same name already exists	created
4500	10480	Number of subscription has reached the limit	The number of subscriptions under the same Topic exceeds the limit. At present, the maximum number of subscriptions is 500.	Please control the number of subscriptions
4510	10570	Url connot contain any blank characters	Url cannot contain white space characters	Incorrect url format
6040	10660	It will take some time to release resources of previous topic before you can create a new topic with the same name. Please try later.	Failed to create topic with the same name. Because you just deleted topic with the same name, you need to release resources. Currently, in order to ensure data consistency, CMQ cannot create topic with the same name within 30 seconds after deleting topic.	Please try again later.

Module error code

Error Codes	Module error code	English description	Description	Treatment measures
4000	4460	Queue is already existed,case insensitive	Queue already exists	The queue with this name already exists. Please verify the correctness of the name.
4000	10020	Queue name format error	Incorrect queue name format	The queue name is in an incorrect format. For more information on the rules, please see the QueueName Field description

4000	10120	Message body can't be empty	Message content can not be empty	The content of the message sent cannot be empty. Please set the content of the message.
4000	10470	ReceiptHandle error	ReceiptHandle error.	ReceiptHandle is a string
4000	10520	Undefined notify retry stragety	Undefined message push retry policy	Please check to see if the spelling is wrong
4000	10530	Undefined notify content format	Undefined message push format	Please check to see if the spelling is wrong
4000	10680	Too many bindingKey	BindingKey exceeds limit	BindingKey exceeds the limit. Please control the number of bindingKey.
4000	10691	Too many delimeters	Too many separators	There are too many delimiters for tags or bindingKey. Please refer to the relevant documentation for restrictions.
4000	10700	Parameters lack of routingKey	Missing routingKey	Please add routingKey
4000	10720	Too many msgTag	Message label exceeds limit	The number of message tags exceeds the maximum limit. Please control the number of tags.
4100	10031	Connection is not authenticated	The connection has not been authenticated	The connection has not been authenticated yet. Please authenticate it first.
4100	10032	Connection is already authenticated	The connection has been authenticated	The connection has been authenticated. Do not conduct secondary authentication.
4400	10230	Exceed maximum	The message size has	The message size you

		message size	exceeded the limit	set is out of range, please refer to Description The document sets the message correctly
4410	10240	Reach maximum retention number of message	The number of messages has reached the maximum number of reserved messages in the queue.	Your message queue has reached the maximum number of heap. Please consume the messages in the queue or upgrade the number of heap messages in the queue.
4430	10260	Receipt handle is invalid	Illegal handle	Illegal handle, please confirm the validity of the handle
4450	10220	Number of queues has reached the limit	The current number of queues has reached the maximum	The number of queues you have created has reached the quota Activate. If you have any special needs, please. Submit a ticket Consult
4470	10300	Total message size exceed 1MB	Message size exceeds 1MB	The size of the message you sent is more than 1MB. Please resize the message.
4490	10770	Message id is invalid	Illegal message ID	The message ID you entered is incorrect. Please check the validity of the message ID.
4490	10780	Message operation is not allowed	Disallowed message state transition	The status of the transaction message conflicts with the state you set
4490	10790	Transaction message is not supported	Queues do not support transaction messages	Queues do not support transaction messages

6000	10090	Send message failed	Failed to send message	Internal error, please Submit a ticket Contact the technician to view the log
6000	10130	Recieve message failed	Failed to receive message	Internal error, please Submit a ticket Contact the technician to view the log
6000	10140	Delete message failed	Failed to delete message	Internal error, please Submit a ticket Contact the technician to view the log
6010	10150	Delete message partially failed	Failed to delete some messages	Internal error, please Submit a ticket Contact the technician to view the log
6000	10160	Get queue attributes failed	Failed to get queue properties	Internal error, please Submit a ticket Contact the technician to view the log
6000	10170	Set queue attributes failed	Failed to set queue properties	Internal error, please Submit a ticket Contact the technician to view the log
6000	10180	Delete queue failed	Failed to delete queue	Internal error, please Submit a ticket Contact the technician to view the log
6000	10190	List queue failed	Failed to get queue list	Internal error, please Submit a ticket Contact the technician to view the log
6020	10290	Batch delete message failed	Batch delete operation failed	Internal error, please Submit a ticket Contact the technician to view the log
6030	10650	Topic has no	Publish's message	Please add a

		subscription, please create a subscription before publishing message	failed. Topic does not have a subscription at present.	subscription for the current topic
6030	10730	No bindingKey or filterTag matches the routingKey or msgTag	The message cannot shipping: the message tag cannot match the subscriber routing keyword or filter tag	Please check whether the BindingKey of topic subscriber or the RoutingKey of production message is correct.
6040	10750	Transaction confirmation failed	Failed to confirm transaction message	Internal error, please Submit a ticket Contact the technician to view the log
6040	10760	Transaction confirmation partially failed	Failed to confirm part of the transaction message	Internal error, please Submit a ticket Contact the technician to view the log
6050	10740	Too many filterTag or bindingKey	The number of filter tags or routing keywords exceeds the limit	The filter label you set
6070	10690	Too many unacked (inactive) messages or delayed messages	There are too many messages in the current queue that have not been deleted or delayed.	Please delete the message that the consumption is completed.
7000	10200	No message	No message	Currently, there is no message. Please make sure that there are messages in the queue before consuming.