

Config

API Documentation

Product Documentation



Copyright Notice

©2013–2026 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by the Tencent corporate group, including its parent, subsidiaries and affiliated companies, as the case may be. Trademarks of third parties referred to in this document are owned by their respective proprietors.

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

- Introduction

- API Category

- Making API Requests

 - Request Structure

 - Common Params

 - Signature v3

 - Signature

 - Responses

- Rule APIs

 - PutEvaluations

 - ListConfigRules

 - ListAggregateConfigRules

- Resource APIs

 - ListDiscoveredResources

 - DescribeDiscoveredResource

 - ListAggregateDiscoveredResources

- Data Types

- Error Codes

API Documentation

History

Last updated: 2026-04-16 16:49:02

Release 2

Release time: 2025-01-15 16:55:45

Release updates:

Improvement to existing documentation.

New APIs:

- [ListAggregateDiscoveredResources](#)

New data structures:

- [AggregateResourceInfo](#)

Modified data structures:

- [ConfigRule](#)
 - **Modified members:** AccountGroupId, AccountGroupName, RuleOwnerId, ManageTriggerType

Release 1

Release time: 2024-12-02 15:05:48

Release updates:

Improvement to existing documentation.

New APIs:

- [DescribeDiscoveredResource](#)
- [ListAggregateConfigRules](#)
- [ListConfigRules](#)
- [ListDiscoveredResources](#)
- [PutEvaluations](#)

New data structures:

- [Annotation](#)
- [ConfigRule](#)
- [Evaluation](#)
- [Filter](#)
- [InputParameter](#)
- [InputParameterForManage](#)
- [ResourceListInfo](#)
- [SourceConditionForManage](#)
- [Tag](#)
- [TriggerType](#)

Introduction

Last updated: 2026-04-16 16:48:57

CloudConfig (Config) is a cloud resource auditing and governance platform. It continuously records and evaluates the configuration information and change operations of different cloud resources in different regions under multiple customer accounts. CloudConfig (Config) supports Tencent Cloud best practices, providing customers with managed compliance rules and various scenario compliance packages, and also supports custom rules. It helps users continuously evaluate the compliance of resource configurations.

API Category

Last updated: 2026-04-16 16:48:57

Resource APIs

API Name	Feature	Frequency Limit (maximum requests per second)
DescribeDiscoveredResource	Resource details	20
ListAggregateDiscoveredResources	Account Group accesses the resource list	20
ListDiscoveredResources	This API is used to get the resource list.	20

Rule APIs

API Name	Feature	Frequency Limit (maximum requests per second)
ListAggregateConfigRules	Gets the account group rule list	20
ListConfigRules	Gets the rule list	20
PutEvaluations	Reports custom rule evaluation results	20

Making API Requests

Request Structure

Last updated: 2026-04-16 16:48:58

1. Service Address

The API supports access from either a nearby region (at `config.intl.tencentcloudapi.com`) or a specified region (at `config.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 "`config.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)	<code>config.intl.tencentcloudapi.com</code>
South China (Guangzhou)	<code>config.ap-guangzhou.tencentcloudapi.com</code>
East China (Shanghai)	<code>config.ap-shanghai.tencentcloudapi.com</code>
East China (Nanjing)	<code>config.ap-nanjing.tencentcloudapi.com</code>
North China (Beijing)	<code>config.ap-beijing.tencentcloudapi.com</code>
Southwest China (Chengdu)	<code>config.ap-chengdu.tencentcloudapi.com</code>
Southwest China (Chongqing)	<code>config.ap-chongqing.tencentcloudapi.com</code>
Hong Kong, Macao, Taiwan (Hong Kong, China)	<code>config.ap-hongkong.tencentcloudapi.com</code>
Southeast Asia (Singapore)	<code>config.ap-</code>

	singapore.tencentcloudapi.com
Southeast Asia (Jakarta)	config.ap-jakarta.tencentcloudapi.com
Southeast Asia (Bangkok)	config.ap-bangkok.tencentcloudapi.com
Northeast Asia (Seoul)	config.ap-seoul.tencentcloudapi.com
Northeast Asia (Tokyo)	config.ap-tokyo.tencentcloudapi.com
U.S. East Coast (Virginia)	config.na-ashburn.tencentcloudapi.com
U.S. West Coast (Silicon Valley)	config.na-siliconvalley.tencentcloudapi.com
South America (São Paulo)	config.sa-saopaulo.tencentcloudapi.com
Europe (Frankfurt)	config.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: 2026-04-16 16:48:58

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 the description of common parameter <code>Action</code> in the input parameters in related API documentation. For example, the API for querying the CVM instance list is <code>DescribeInstances</code> .
X-TC-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 <code>Region</code> 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 it is passed.
X-TC-Timestamp	Integer	Yes	The current UNIX timestamp that records the time when the API request was initiated, for example, 1529223702. Note: If the difference between the UNIX timestamp and the server time is greater than 5 minutes, a signature expiration error may occur.
X-TC-Version	String	Yes	API version of the action. For the valid values, see the description of the common input parameter <code>Version</code> in the API documentation. For example, the version of CVM is 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=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 Here: - TC3-HMAC-SHA256: Signature method, currently fixed as this value; - Credential: Signature credential; AKID*** is the SecretId; Date is a date in UTC time, and this value must match the value of X-TC-Timestamp (a common parameter) in UTC time format; service is the name of the product/service, and is generally a domain name prefix. For example, a domain name <code>cvm.tencentcloudapi.com</code> refers to the CVM product and the value would be <code>cvm</code> ; - SignedHeaders: The headers that contains the authentication information; <code>content-type</code> and <code>host</code> 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 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/?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 <code>Action</code> in the input parameters in related API documentation. For example, the API for querying the CVM instance list is <code>DescribeInstances</code> .
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 <code>Region</code> 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 <code>Timestamp</code> 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 <code>Version</code> 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=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&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
Hong Kong/Macao/Taiwan (China) (Hong Kong (China))	ap-hongkong
Southeast Asia (Singapore)	ap-singapore

Signature v3

Last updated: 2026-04-16 17:12:27

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=a7b85514
48762bd123d6f79e81815e31a92013640a6cef36a08ad4b292a4d2f2" \
-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 <code>POST</code> .
CanonicalURI	URI parameter. Slash ("/") is used for API 3.0.
CanonicalQueryString	<p>The query string in the URL of the originating HTTP request. This is always an empty string "" for POST requests, and is the string after the question mark (?) for GET requests. For example: Limit=10&Offset=0.</p> <p>Note: <code>CanonicalQueryString</code> must be URL-encoded, referencing RFC3986, the UTF8 character set. We recommend using the programming language library. All special characters must be encoded and capitalized.</p>
CanonicalHeaders	<p>Header information for signature calculation, including at least two headers of <code>host</code> and <code>content-type</code> . Custom headers can be added to participate in the signature process to improve the uniqueness and security of the request.</p> <p>Concatenation rules:</p> <ol style="list-style-type: none"> Both the key and value of the header should be converted to lowercase with the leading and trailing spaces removed, so they are concatenated in the format of <code>key:value\n</code> format; If there are multiple headers, they should be sorted in ASCII ascending order by the header keys (lowercase). <p>The calculation result in this example is <code>content-type:application/json; charset=utf-8\nhost:cvm.tencentcloudapi.com\n</code> .</p> <p>Note: <code>content-type</code> 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 that signature verification failed.</p>
SignedHeaders	<p>Header information for signature calculation, indicating which headers of the request participate in the signature process (they must each individually correspond to the headers in CanonicalHeaders). <code>Content-type</code> and <code>host</code> are required headers.</p> <p>Concatenation rules:</p> <ol style="list-style-type: none"> Both the key and value of the header should be converted to lowercase; If there are multiple headers, they should be sorted in ASCII ascending order by the header keys (lowercase) and separated by semicolons (;). <p>The value in this example is <code>content-type;host</code></p>
HashedRequestPayload	<p>Hash value of the request payload (i.e., the body, such as <code>{"Limit": 1, "Filters": [{"Values": ["unnamed"], "Name": "instance-name"}]}</code> in this example). The pseudocode for calculation is <code>Lowercase(HexEncode(Hash.SHA256(RequestPayload)))</code> by SHA256 hashing the payload of the HTTP request, performing hexadecimal encoding, and finally converting the encoded string to lowercase letters. For GET requests, <code>RequestPayload</code> is always an empty string. The calculation result in this example is <code>99d58dfbc6745f6747f36bfca17dee5e6881dc0428a0a36f96199342bc5b4907</code> .</p>

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 <code>TC3-HMAC-SHA256</code> .
RequestTimestamp	Request timestamp, i.e., the value of the common parameter <code>X-TC-Timestamp</code> in the request header, which is the UNIX timestamp of the current time in seconds, such as <code>1551113065</code> in this example.
CredentialScope	Scope of the credential in the format of <code>Date/service/tc3_request</code> , including the date, requested service and termination string (<code>tc3_request</code>). Date is a date in UTC time, whose value should match the UTC date converted by the common parameter <code>X-TC-Timestamp</code> ; <code>service</code> is the product name, which should match the domain name of the product called. The calculation result in this example is <code>2019-02-25/cvm/tc3_request</code> .
HashedCanonicalRequest	Hash value of the CanonicalRequest string concatenated in the steps above. The pseudocode for calculation is <code>Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))</code> . The calculation result in this example is <code>2815843035062ffda5fd6f2a44ea8a34818b0dc46f024b8b3786976a3adda7a</code> .

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
2815843035062ffffda5fd6f2a44ea8a34818b0dc46f024b8b3786976a3adda7a
```

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., <code>*****</code> .
Date	The Date field information in <code>Credential</code> , such as <code>2019-02-25</code> in this example.
Service	Value in the Service field in <code>Credential</code> , such as <code>cvm</code> 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 <code>TC3-HMAC-SHA256</code> .
SecretId	The SecretId in the key pair, i.e., <code>AKID*****</code> .
CredentialScope	Credential scope (see above). The calculation result in this example is <code>2019-02-25/cvm/tc3_request</code> .
SignedHeaders	Header information for signature calculation (see above), such as <code>content-type;host</code> in this example.
Signature	Signature value. The calculation result in this example is <code>a7b8551448762bd123d6f79e81815e31a92013640a6cef36a08ad4b292a4d2f2</code> .

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

```
TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=a7b8551448762bd123d6f79e81815e31a92013640a6cef36a08ad4b292a4d2f2
```

The following example shows a finished authorization header:

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=a7b8551448762bd123d6f79e81815e31a92013640a6cef36a08ad4b292a4d2f2
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%2BWcGel%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 =

```

```

g">&quot;1551113065&quot;</span>;
<span class="hljs-comment">//String timestamp = String.valueOf(System.current
TimeMillis() / 1000);</span>
SimpleDateFormat sdf = <span class="hljs-keyword">new</span> SimpleDateFormat
(<span class="hljs-string">&quot;yyyy-MM-dd&quot;</span>);
<span class="hljs-comment">// Pay attention to the time zone; otherwise, erro
rs may occur</span>
sdf.setTimeZone(TimeZone.getTimeZone(<span class="hljs-string">&quot;UTC&quo
t;</span>));
<span class="hljs-keyword">String</span> date = sdf.format(<span class="hljs-keyword">new</span> Date(Long.valueOf(timestamp + <span class="hljs-string">&quot;
000&quot;</span>));

<span class="hljs-comment">// ***** Step 1: Concatenate the Canonical
Request string *****</span>
<span class="hljs-keyword">String</span> httpRequestMethod = <span class="hljs-string">&quot;POST&quot;</span>;
<span class="hljs-keyword">String</span> canonicalUri = <span class="hljs-string">&quot;/&quot;</span>;
<span class="hljs-keyword">String</span> canonicalQueryString = <span class="hljs-string">&quot;&quot;</span>;
<span class="hljs-keyword">String</span> canonicalHeaders = <span class="hljs-string">&quot;content-type:application/json; charset=utf-8&quot;</span> +
<span class="hljs-string">&quot;host:&quot;</span> + host + <span class="hljs-string">&quot;\n&quot;</span>;
<span class="hljs-keyword">String</span> signedHeaders = <span class="hljs-string">&quot;content-type;host&quot;</span>;

<span class="hljs-keyword">String</span> payload = <span class="hljs-string">&quot;{\&quot;Limit\&quot;: 1, \&quot;Filters\&quot;: [{\&quot;Values\&quot;:
[\&quot;unnamed\&quot;], \&quot;Name\&quot;: \&quot;instance-name\&quot;}]}&quot;</span>;
<span class="hljs-keyword">String</span> hashedRequestPayload = sha256Hex(payload);
<span class="hljs-keyword">String</span> canonicalRequest = httpRequestMethod
+ <span class="hljs-string">&quot;\n&quot;</span> + canonicalUri + <span class="hljs-string">&quot;\n&quot;</span> + canonicalQueryString + <span class="hljs-string">&quot;\n&quot;</span>
+ canonicalHeaders + <span class="hljs-string">&quot;\n&quot;</span> + signed
Headers + <span class="hljs-string">&quot;\n&quot;</span> + hashedRequestPayl
oad;
System.out.<span class="hljs-built_in">println</span>(canonicalRequest);

<span class="hljs-comment">// ***** Step 2: Concatenate the string to
sign *****</span>
<span class="hljs-keyword">String</span> credentialScope = date + <span class="hljs-string">&quot;/&quot;</span> + service + <span class="hljs-string">&quot;&quot;</span>

```

```

ot;/&quot;</span> + <span class="hljs-string">&quot;tc3_request&quot;</span>;
<span class="hljs-keyword">String</span> hashedCanonicalRequest = sha256Hex(c
anonicalRequest);
<span class="hljs-keyword">String</span> stringToSign = algorithm + <span cla
ss="hljs-string">&quot;\n&quot;</span> + timestamp + <span class="hljs-strin
g">&quot;\n&quot;</span> + credentialScope + <span class="hljs-string">&quo
t;\n&quot;</span> + hashedCanonicalRequest;
System.out.<span class="hljs-built_in">println</span>(stringToSign);

<span class="hljs-comment">// ***** Step 3: Calculate the signature *
*****</span>
<span class="hljs-built_in">byte</span>[] secretDate = hmac256((<span class
="hljs-string">&quot;TC3&quot;</span> + SECRET_KEY).getBytes(UTF8), date);
<span class="hljs-built_in">byte</span>[] secretService = hmac256(secretDate,
service);
<span class="hljs-built_in">byte</span>[] secretSigning = hmac256(secretServi
ce, <span class="hljs-string">&quot;tc3_request&quot;</span>);
<span class="hljs-keyword">String</span> signature = DatatypeConverter.printH
exBinary(hmac256(secretSigning, stringToSign)).toLowerCase();
System.out.<span class="hljs-built_in">println</span>(signature);

<span class="hljs-comment">// ***** Step 4: Concatenate the Authoriza
tion *****</span>
<span class="hljs-keyword">String</span> authorization = algorithm + <span cl
ass="hljs-string">&quot; &quot;</span> + <span class="hljs-string">&quot;Cred
ential=&quot;</span> + SECRET_ID + <span class="hljs-string">&quot;/&quot;</s
pan> + credentialScope + <span class="hljs-string">&quot;; &quot;</span>
+ <span class="hljs-string">&quot;SignedHeaders=&quot;</span> + signedHeaders
+ <span class="hljs-string">&quot;; &quot;</span> + <span class="hljs-strin
g">&quot;Signature=&quot;</span> + signature;
System.out.<span class="hljs-built_in">println</span>(authorization);

TreeMap<<span class="hljs-keyword">String</span>, <span class="hljs-keywor
d">String</span>> headers = <span class="hljs-keyword">new</span> TreeMap<<
<span class="hljs-keyword">String</span>, <span class="hljs-keyword">String</
span>>();
headers.put(<span class="hljs-string">&quot;Authorization&quot;</span>, autho
rization);
headers.put(<span class="hljs-string">&quot;Content-Type&quot;</span>, CT_JS
ON);
headers.put(<span class="hljs-string">&quot;Host&quot;</span>, host);
headers.put(<span class="hljs-string">&quot;X-TC-Action&quot;</span>, actio
n);
headers.put(<span class="hljs-string">&quot;X-TC-Timestamp&quot;</span>, time
stamp);
headers.put(<span class="hljs-string">&quot;X-TC-Version&quot;</span>, versio
n);

```

```

headers.put (<span class="hljs-string">&quot;X-TC-Region&quot;</span>, regio
n);

StringBuilder sb = <span class="hljs-keyword">new</span> StringBuilder();
sb.<span class="hljs-built_in">append</span> (<span class="hljs-string">&quot;
curl -X POST https://&quot;</span>).<span class="hljs-built_in">append</span>
(host)
.<span class="hljs-built_in">append</span> (<span class="hljs-string">&quot; -
H \&quot;Authorization: &quot;</span>).<span class="hljs-built_in">append</sp
an>(authorization).<span class="hljs-built_in">append</span> (<span class="hlj
s-string">&quot; \&quot;&quot;&quot;</span>)
.<span class="hljs-built_in">append</span> (<span class="hljs-string">&quot; -
H \&quot;Content-Type: application/json; charset=utf-8&quot;&quot;</span>)
.<span class="hljs-built_in">append</span> (<span class="hljs-string">&quot; -
H \&quot;Host: &quot;</span>).<span class="hljs-built_in">append</span> (hos
t).<span class="hljs-built_in">append</span> (<span class="hljs-string">&quot;
t; \&quot;&quot;&quot;</span>)
.<span class="hljs-built_in">append</span> (<span class="hljs-string">&quot; -
H \&quot;X-TC-Action: &quot;</span>).<span class="hljs-built_in">append</span>
(action).<span class="hljs-built_in">append</span> (<span class="hljs-strin
g">&quot; \&quot;&quot;&quot;</span>)
.<span class="hljs-built_in">append</span> (<span class="hljs-string">&quot; -
H \&quot;X-TC-Timestamp: &quot;</span>).<span class="hljs-built_in">append</s
pan>(timestamp).<span class="hljs-built_in">append</span> (<span class="hljs-s
tring">&quot; \&quot;&quot;&quot;</span>)
.<span class="hljs-built_in">append</span> (<span class="hljs-string">&quot; -
H \&quot;X-TC-Version: &quot;</span>).<span class="hljs-built_in">append</spa
n>(version).<span class="hljs-built_in">append</span> (<span class="hljs-strin
g">&quot; \&quot;&quot;&quot;</span>)
.<span class="hljs-built_in">append</span> (<span class="hljs-string">&quot; -
H \&quot;X-TC-Region: &quot;</span>).<span class="hljs-built_in">append</span>
(region).<span class="hljs-built_in">append</span> (<span class="hljs-strin
g">&quot; \&quot;&quot;&quot;</span>)
.<span class="hljs-built_in">append</span> (<span class="hljs-string">&quot; -
d &#x27;&quot;</span>).<span class="hljs-built_in">append</span> (payload).<sp
an class="hljs-built_in">append</span> (<span class="hljs-string">&quot;
&#x27;
&quot;</span>);
System.out.<span class="hljs-built_in">println</span> (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": [{"Values": ["unnamed"], "Name": "instance-name"}]}
```

* Step 1: Concatenate the Canonical Request 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")).hexdigest()
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.sha256).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

+ <span class="hljs-string">#x27; -H &quot;Authorization: #x27;</span> + au
thorization <span class="hljs-comment">+</span> <span class="hljs-comment">#
x27;&quot;#x27;</span>
+ <span class="hljs-string">#x27; -H &quot;Content-Type: application/json; c
harset=utf-8&quot;#x27;</span>
+ <span class="hljs-string">#x27; -H &quot;Host: #x27;</span> + host <span
class="hljs-comment">+</span> <span class="hljs-comment">#x27;&quot;#x27;</
span>
+ <span class="hljs-string">#x27; -H &quot;X-TC-Action: #x27;</span> + acti
on <span class="hljs-comment">+</span> <span class="hljs-comment">#x27;&qu
ot;#x27;</span>
+ <span class="hljs-string">#x27; -H &quot;X-TC-Timestamp: #x27;</span> + s
tr(timestamp) <span class="hljs-comment">+</span> <span class="hljs-comment">
#x27;&quot;#x27;</span>
+ <span class="hljs-string">#x27; -H &quot;X-TC-Version: #x27;</span> + ver
sion <span class="hljs-comment">+</span> <span class="hljs-comment">#x27;&qu
ot;#x27;</span>
+ <span class="hljs-string">#x27; -H &quot;X-TC-Region: #x27;</span> + regi
on <span class="hljs-comment">+</span> <span class="hljs-comment">#x27;&qu
ot;#x27;</span>
+ <span class="hljs-string">#x27; -d #x27;&quot;#x27;</span> + payload <span cla
ss="hljs-comment">+</span> <span class="hljs-comment">#x27;&quot;#x27;</spa
n><span class="hljs-comment">)</span>

```

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

    <span class="hljs-comment">// step 1: build canonical request string</span>
    httpRequestMethod := <span class="hljs-string">"POST"</span>
    canonicalURI := <span class="hljs-string">"/</span>
    canonicalQueryString := <span class="hljs-string">""</span>
    canonicalHeaders := <span class="hljs-string">"content-type:application/
    json; charset=utf-8</span> + <span class="hljs-string">"host:<qu
    ot;</span> + host + <span class="hljs-string">"</span>
    signedHeaders := <span class="hljs-string">"content-type;host</spa
    n>
    payload := <span class="hljs-string">`{<span class="hljs-string">"Limit"</span>: 1, <span class="hljs-string">"Filters"</span>
```

```

ot;: [{"&quot;Values&quot;: [{"&quot;unnamed&quot;}, {"&quot;Name&quot;: &quot;instance-name&quot;}]}]`</span>
hashedRequestPayload := sha256hex(payload)
canonicalRequest := fmt.Sprintf(<span class="hljs-string">&quot;%s\n%s\n%s\n%s\n%s\n%s&quot;</span>,
httpRequestMethod,
canonicalURI,
canonicalQueryString,
canonicalHeaders,
signedHeaders,
hashedRequestPayload)
fmt.Println(canonicalRequest)

```

```

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

```

```

<span class="hljs-comment">// step 3: sign string</span>
secretDate := hmacsha256(date, <span class="hljs-string">&quot;TC3&quot;</span>+secretKey)
secretService := hmacsha256(service, secretDate)
secretSigning := hmacsha256(<span class="hljs-string">&quot;tc3_request&quot;</span>, secretService)
signature := hex.EncodeToString([]<span class="hljs-keyword">byte</span>(hmacsha256(string2sign, secretSigning)))
fmt.Println(signature)

```

```

<span class="hljs-comment">// step 4: build authorization</span>
authorization := fmt.Sprintf(<span class="hljs-string">&quot;%s Credential=%s/%s, SignedHeaders=%s, Signature=%s&quot;</span>,
algorithm,
secretId,
credentialScope,
signedHeaders,
signature)
fmt.Println(authorization)

```

```

curl := fmt.Sprintf(<span class="hljs-string">`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
$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 Canonical Request 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}, Signed
Headers=#{signed_headers}, Signature=#{signature}"
puts authorization

puts 'curl -X POST ' + endpoint
```

```

+ <span class="hljs-string">#x27; -H &quot;Authorization: #x27;</span> + au
thorization <span class="hljs-comment">+</span> <span class="hljs-comment">#
x27;&quot;#x27;</span> <span class="hljs-comment">\</span>
+ <span class="hljs-string">#x27; -H &quot;Content-Type: application/json; c
harset=utf-8&quot;#x27;</span> \
+ <span class="hljs-string">#x27; -H &quot;Host: #x27;</span> + host <span
class="hljs-comment">+</span> <span class="hljs-comment">#x27;&quot;#x27;</
span> <span class="hljs-comment">\</span>
+ <span class="hljs-string">#x27; -H &quot;X-TC-Action: #x27;</span> + acti
on <span class="hljs-comment">+</span> <span class="hljs-comment">#x27;&quo
t;#x27;</span> <span class="hljs-comment">\</span>
+ <span class="hljs-string">#x27; -H &quot;X-TC-Timestamp: #x27;</span> + t
imestamp.to_s <span class="hljs-comment">+</span> <span class="hljs-comment">
#x27;&quot;#x27;</span> <span class="hljs-comment">\</span>
+ <span class="hljs-string">#x27; -H &quot;X-TC-Version: #x27;</span> + ver
sion <span class="hljs-comment">+</span> <span class="hljs-comment">#x27;&qu
ot;#x27;</span> <span class="hljs-comment">\</span>
+ <span class="hljs-string">#x27; -H &quot;X-TC-Region: #x27;</span> + regi
on <span class="hljs-comment">+</span> <span class="hljs-comment">#x27;&quo
t;#x27;</span> <span class="hljs-comment">\</span>
+ <span class="hljs-string">&quot; -d #x27;&quot;</span> + payload <span cla
ss="hljs-comment">+</span> <span class="hljs-comment">&quot;#x27;&quot;</spa
n>

```

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);
}
}
}

```

```

<span class="hljs-function"><span class="hljs-keyword">public</span> <span class="hljs-keyword">static</span> Dictionary<String, String> <span class="hljs-title">BuildHeaders</span>( <span class="hljs-params"><span class="hljs-built_in">string</span> secretid,
<span class="hljs-built_in">string</span> secretkey, <span class="hljs-built_in">string</span> service, <span class="hljs-built_in">string</span> endpoint, <span class="hljs-built_in">string</span> region,
<span class="hljs-built_in">string</span> action, <span class="hljs-built_in">string</span> version, DateTime date, <span class="hljs-built_in">string</span> requestPayload</span> ) </span>
{
<span class="hljs-built_in">string</span> datestr = date.ToString( <span class="hljs-string">&quot;yyyy-MM-dd&quot;</span> );
DateTime startTime = <span class="hljs-keyword">new</span> DateTime( <span class="hljs-number">1970</span>, <span class="hljs-number">1</span>, <span class="hljs-number">1</span>, <span class="hljs-number">0</span>, <span class="hljs-number">0</span>, <span class="hljs-number">0</span>, <span class="hljs-number">0</span>,
DateTimeKind.Utc);
<span class="hljs-built_in">long</span> requestTimestamp = ( <span class="hljs-built_in">long</span> ) Math.Round( ( date - startTime ). TotalMilliseconds, MidpointRounding.AwayFromZero) / <span class="hljs-number">1000</span>;
<span class="hljs-comment">// ***** Step 1: Concatenate the Canonical Request string *****</span>
<span class="hljs-built_in">string</span> algorithm = <span class="hljs-string">&quot;TC3-HMAC-SHA256&quot;</span>;
<span class="hljs-built_in">string</span> httpRequestMethod = <span class="hljs-string">&quot;POST&quot;</span>;
<span class="hljs-built_in">string</span> canonicalUri = <span class="hljs-string">&quot;/&quot;</span>;
<span class="hljs-built_in">string</span> canonicalQueryString = <span class="hljs-string">&quot;&quot;</span>;
<span class="hljs-built_in">string</span> contentType = <span class="hljs-string">&quot;application/json&quot;</span>;
<span class="hljs-built_in">string</span> canonicalHeaders = <span class="hljs-string">&quot;content-type:&quot;</span> + contentType + <span class="hljs-string">&quot;; charset=utf-8\n&quot;</span> + <span class="hljs-string">&quot;

```

```

t;host:&quot;;</span> + endpoint + <span class="hljs-string">&quot;;\n&quot;;</span>
span>;
<span class="hljs-built_in">string</span> signedHeaders = <span class="hljs-string">&quot;;content-type;host&quot;;</span>;
<span class="hljs-built_in">string</span> hashedRequestPayload = SHA256Hex(requestPayload);
<span class="hljs-built_in">string</span> canonicalRequest = httpRequestMethod + <span class="hljs-string">&quot;;\n&quot;;</span>
+ canonicalUri + <span class="hljs-string">&quot;;\n&quot;;</span>
+ canonicalQueryString + <span class="hljs-string">&quot;;\n&quot;;</span>
+ canonicalHeaders + <span class="hljs-string">&quot;;\n&quot;;</span>
+ signedHeaders + <span class="hljs-string">&quot;;\n&quot;;</span>
+ hashedRequestPayload;
Console.WriteLine(canonicalRequest);
Console.WriteLine(<span class="hljs-string">&quot;;-----
-----&quot;;</span>);

<span class="hljs-comment">// ***** Step 2: Concatenate the string to
sign *****</span>
<span class="hljs-built_in">string</span> credentialScope = datestr + <span class="hljs-string">&quot;;/&quot;;</span> + service + <span class="hljs-string">&quot;;/&quot;;</span> + <span class="hljs-string">&quot;;tc3_request&quot;;</span>
span>;
<span class="hljs-built_in">string</span> hashedCanonicalRequest = SHA256Hex(
canonicalRequest);
<span class="hljs-built_in">string</span> stringToSign = algorithm + <span class="hljs-string">&quot;;\n&quot;;</span> + requestTimestamp.ToString() + <span class="hljs-string">&quot;;\n&quot;;</span> + credentialScope + <span class="hljs-string">&quot;;\n&quot;;</span> + hashedCanonicalRequest;
Console.WriteLine(stringToSign);
Console.WriteLine(<span class="hljs-string">&quot;;-----
-----&quot;;</span>);

<span class="hljs-comment">// ***** Step 3: Calculate the signature *
*****</span>
<span class="hljs-built_in">byte</span>[] tc3SecretKey = Encoding.UTF8.GetBytes(<span class="hljs-string">&quot;;TC3&quot;;</span> + secretkey);
<span class="hljs-built_in">byte</span>[] secretDate = HmacSHA256(tc3SecretKey, Encoding.UTF8.GetBytes(datestr));
<span class="hljs-built_in">byte</span>[] secretService = HmacSHA256(secretDate, Encoding.UTF8.GetBytes(service));
<span class="hljs-built_in">byte</span>[] secretSigning = HmacSHA256(secretService, Encoding.UTF8.GetBytes(<span class="hljs-string">&quot;;tc3_request&quot;;</span>));
<span class="hljs-built_in">byte</span>[] signatureBytes = HmacSHA256(secretSigning, Encoding.UTF8.GetBytes(stringToSign));
<span class="hljs-built_in">string</span> signature = BitConverter.ToString(s

```

```

signatureBytes).Replace("<span class='hljs-string'>&quot;;-&quot;</span>", "<span
class='hljs-string'>&quot;&quot;</span>").ToLower();
Console.WriteLine(signature);
Console.WriteLine("<span class='hljs-string'>&quot;;-----
-----&quot;</span>");

<span class='hljs-comment'>// ***** Step 4: Concatenate the Authoriza
tion *****</span>
<span class='hljs-built_in'>string</span> authorization = algorithm + <span c
lass='hljs-string'>&quot;; &quot;</span>
+ <span class='hljs-string'>&quot;;Credential=&quot;</span> + secretid + <span
class='hljs-string'>&quot;;/&quot;</span> + credentialScope + <span class='hlj
s-string'>&quot;;, &quot;</span>
+ <span class='hljs-string'>&quot;;SignedHeaders=&quot;</span> + signedHeaders
+ <span class='hljs-string'>&quot;;, &quot;</span>
+ <span class='hljs-string'>&quot;;Signature=&quot;</span> + signature;
Console.WriteLine(authorization);
Console.WriteLine("<span class='hljs-string'>&quot;;-----
-----&quot;</span>");

Dictionary<<span class='hljs-built_in'>string</span>, <span class='hljs-bu
ilt_in'>string</span>> headers = <span class='hljs-keyword'>new</span> Dictio
nary<<span class='hljs-built_in'>string</span>, <span class='hljs-built_i
n'>string</span>>();
headers.Add("<span class='hljs-string'>&quot;;Authorization&quot;</span>", autho
rization);
headers.Add("<span class='hljs-string'>&quot;;Host&quot;</span>", endpoint);
headers.Add("<span class='hljs-string'>&quot;;Content-Type&quot;</span>", conten
tType + <span class='hljs-string'>&quot;;; charset=utf-8&quot;</span>);
headers.Add("<span class='hljs-string'>&quot;;X-TC-Timestamp&quot;</span>", requ
estTimestamp.ToString());
headers.Add("<span class='hljs-string'>&quot;;X-TC-Version&quot;</span>", versio
n);
headers.Add("<span class='hljs-string'>&quot;;X-TC-Action&quot;</span>", actio
n);
headers.Add("<span class='hljs-string'>&quot;;X-TC-Region&quot;</span>", regio
n);
<span class='hljs-keyword'>return</span> headers;
}

<span class='hljs-function'><span class='hljs-keyword'>public</span> <span cl
ass='hljs-keyword'>static</span> <span class='hljs-keyword'>void</span> <span
class='hljs-title'>Main</span>(<span class='hljs-params'><span class='hljs-bu
ilt_in'>string</span>[] args</span></span>
{
<span class='hljs-comment'>// SecretID and SecretKey</span>
<span class='hljs-built_in'>string</span> SECRET_ID = <span class='hljs-strin
g'>&quot;;AKID*****&quot;</span>;

```

```

<span class="hljs-built_in">string</span> SECRET_KEY = <span class="hljs-string">&quot;*****&quot;</span>;

<span class="hljs-built_in">string</span> service = <span class="hljs-string">&quot;cvm&quot;</span>;
<span class="hljs-built_in">string</span> endpoint = <span class="hljs-string">&quot;cvm.tencentcloudapi.com&quot;</span>;
<span class="hljs-built_in">string</span> region = <span class="hljs-string">&quot;ap-guangzhou&quot;</span>;
<span class="hljs-built_in">string</span> action = <span class="hljs-string">&quot;DescribeInstances&quot;</span>;
<span class="hljs-built_in">string</span> version = <span class="hljs-string">&quot;2017-03-12&quot;</span>;

<span class="hljs-comment">// The timestamp `2019-02-26 00:44:25` used here is only for reference. In a project, use the following parameter:</span>
<span class="hljs-comment">// DateTime date = DateTime.UtcNow;</span>
<span class="hljs-comment">// Enter the correct time zone. We recommend using UTC timestamp to avoid errors.</span>
DateTime date = <span class="hljs-keyword">new</span> DateTime(<span class="hljs-number">1970</span>, <span class="hljs-number">1</span>, <span class="hljs-number">1</span>, <span class="hljs-number">0</span>, <span class="hljs-number">0</span>, <span class="hljs-number">0</span>, <span class="hljs-number">0</span>, <span class="hljs-number">0</span>, <span class="hljs-number">1551113065</span>);
<span class="hljs-built_in">string</span> requestPayload = <span class="hljs-string">&quot;{\&quot;Limit\&quot;: 1, \&quot;Filters\&quot;: [{\&quot;Values\&quot;: [\&quot;unnamed\&quot;], \&quot;Name\&quot;: \&quot;instance-name\&quot;}]}&quot;</span>;

Dictionary<&lt;<span class="hljs-built_in">string</span>, <span class="hljs-built_in">string</span>>> headers = BuildHeaders(SECRET_ID, SECRET_KEY, service, endpoint, region, action, version, date, requestPayload);

Console.WriteLine(<span class="hljs-string">&quot;POST https://cvm.tencentcloudapi.com&quot;</span>);
<span class="hljs-keyword">foreach</span> (<span class="hljs-built_in">KeyValuePair<&lt;<span class="hljs-built_in">string</span>, <span class="hljs-built_in">string</span>>> kv <span class="hljs-keyword">in</span> headers)
{
    Console.WriteLine(kv.Key + <span class="hljs-string">&quot;: &quot;</span> + 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 <span class="hljs-subst">${year}</span>-<span class="hljs-subst">${month}</span>-<span class="hljs-subst">${day}</span>
}

function main() {

  <span class="hljs-keyword">const</span> SECRET_ID = <span class="hljs-string">'&quot;AKID*****&quot;</span>;
  <span class="hljs-keyword">const</span> SECRET_KEY = <span class="hljs-string">'&quot;*****&quot;</span>;

  <span class="hljs-keyword">const</span> endpoint = <span class="hljs-string">'&quot;cvm.tencentcloudapi.com&quot;</span>
  <span class="hljs-keyword">const</span> service = <span class="hljs-string">'&quot;cvm&quot;</span>
  <span class="hljs-keyword">const</span> region = <span class="hljs-string">'&quot;ap-guangzhou&quot;</span>
  <span class="hljs-keyword">const</span> action = <span class="hljs-string">'&quot;DescribeInstances&quot;</span>
  <span class="hljs-keyword">const</span> version = <span class="hljs-string">'&quot;2017-03-12&quot;</span>
  <span class="hljs-comment">//const timestamp = getTime()</span>
  <span class="hljs-keyword">const</span> timestamp = <span class="hljs-number">1551113065</span>
  <span class="hljs-keyword">const</span> <span class="hljs-built_in">date</span>

```

```

n> = getDate(timestamp)

<span class="hljs-comment">// ***** Step 1: Concatenate the Canonical
Request string *****</span>
<span class="hljs-keyword">const</span> signedHeaders = <span class="hljs-str
ing">&quot;;content-type;host&quot;;</span>

<span class="hljs-keyword">const</span> payload = <span class="hljs-string">&
quot;{\&quot;Limit&quot;;: 1, \&quot;Filters&quot;;: [{\&quot;Values&quot;;:
[\&quot;unnamed&quot;], \&quot;Name&quot;;: \&quot;instance-name&quot;}}&
quot;;</span>

<span class="hljs-keyword">const</span> hashedRequestPayload = getHash(payload
d);
<span class="hljs-keyword">const</span> httpRequestMethod = <span class="hljs-str
ing">&quot;;POST&quot;;</span>
<span class="hljs-keyword">const</span> canonicalUri = <span class="hljs-str
ing">&quot;;/&quot;;</span>
<span class="hljs-keyword">const</span> canonicalQueryString = <span class="h
ljs-string">&quot;;&quot;;</span>
<span class="hljs-keyword">const</span> canonicalHeaders = <span class="hljs-str
ing">&quot;;content-type:application/json; charset=utf-8\n&quot;;</span> + <
span class="hljs-string">&quot;;host:&quot;;</span> + endpoint + <span class="h
ljs-string">&quot;;\n&quot;;</span>

<span class="hljs-keyword">const</span> canonicalRequest = httpRequestMethod
+ <span class="hljs-string">&quot;;\n&quot;;</span>
+ canonicalUri + <span class="hljs-string">&quot;;\n&quot;;</span>
+ canonicalQueryString + <span class="hljs-string">&quot;;\n&quot;;</span>
+ canonicalHeaders + <span class="hljs-string">&quot;;\n&quot;;</span>
+ signedHeaders + <span class="hljs-string">&quot;;\n&quot;;</span>
+ hashedRequestPayload
<span class="hljs-built_in">console</span>.log(canonicalRequest)
<span class="hljs-built_in">console</span>.log(<span class="hljs-string">&quo
t;;-----&quot;;</span>)

<span class="hljs-comment">// ***** Step 2: Concatenate the string to
sign *****</span>
<span class="hljs-keyword">const</span> algorithm = <span class="hljs-strin
g">&quot;;TC3-HMAC-SHA256&quot;;</span>
<span class="hljs-keyword">const</span> hashedCanonicalRequest = getHash(cano
nicalRequest);
<span class="hljs-keyword">const</span> credentialScope = <span class="hljs-b
uilt_in">date</span> + <span class="hljs-string">&quot;;/&quot;;</span> + servi
ce + <span class="hljs-string">&quot;;/&quot;;</span> + <span class="hljs-strin
g">&quot;;tc3_request&quot;;</span>
<span class="hljs-keyword">const</span> stringToSign = algorithm + <span clas

```

```

s="hljs-string">&quot;\n&quot;</span> +
timestamp + <span class="hljs-string">&quot;\n&quot;</span> +
credentialScope + <span class="hljs-string">&quot;\n&quot;</span> +
hashedCanonicalRequest
<span class="hljs-built_in">console</span>.log(stringToSign)
<span class="hljs-built_in">console</span>.log(<span class="hljs-string">&quot;
t;-----&quot;</span>)

<span class="hljs-comment">// ***** Step 3: Calculate the signature *
*****</span>
<span class="hljs-keyword">const</span> kDate = sha256(<span class="hljs-built_in">date</span>, <span class="hljs-string">&#x27;TC3&#x27;</span> + SECRET_
KEY)
<span class="hljs-keyword">const</span> kService = sha256(service, kDate)
<span class="hljs-keyword">const</span> kSigning = sha256(<span class="hljs-string">&#x27;tc3_request&#x27;</span>, kService)
<span class="hljs-keyword">const</span> signature = sha256(stringToSign, kSig
ning, <span class="hljs-string">&#x27;hex&#x27;</span>)
<span class="hljs-built_in">console</span>.log(signature)
<span class="hljs-built_in">console</span>.log(<span class="hljs-string">&quot;
t;-----&quot;</span>)

<span class="hljs-comment">// ***** Step 4: Concatenate the Authoriza
tion *****</span>
<span class="hljs-keyword">const</span> authorization = algorithm + <span cla
ss="hljs-string">&quot; &quot;</span> +
<span class="hljs-string">&quot;Credential=&quot;</span> + SECRET_ID + <span
class="hljs-string">&quot;/&quot;</span> + credentialScope + <span class="hlj
s-string">&quot;, &quot;</span> +
<span class="hljs-string">&quot;SignedHeaders=&quot;</span> + signedHeaders +
<span class="hljs-string">&quot;, &quot;</span> +
<span class="hljs-string">&quot;Signature=&quot;</span> + signature
<span class="hljs-built_in">console</span>.log(authorization)
<span class="hljs-built_in">console</span>.log(<span class="hljs-string">&quot;
t;-----&quot;</span>)

<span class="hljs-keyword">const</span> Call_Information = <span class="hljs-s
tring">&#x27;curl -X POST &#x27;</span> + <span class="hljs-string">&quot;ht
tps://&quot;</span> + endpoint
+ <span class="hljs-string">&#x27; -H &quot;Authorization: &#x27;</span> + au
thorization + <span class="hljs-string">&#x27;&quot;&#x27;</span>
+ <span class="hljs-string">&#x27; -H &quot;Content-Type: application/json; c
harset=utf-8&quot;&#x27;</span>
+ <span class="hljs-string">&#x27; -H &quot;Host: &#x27;</span> + endpoint +
<span class="hljs-string">&#x27;&quot;&#x27;</span>
+ <span class="hljs-string">&#x27; -H &quot;X-TC-Action: &#x27;</span> + acti
on + <span class="hljs-string">&#x27;&quot;&#x27;</span>

```

```

+ <span class="hljs-string">#x27; -H &quot;X-TC-Timestamp: &#x27;</span> + t
imestamp.toString() + <span class="hljs-string">#x27;&quot;&#x27;</span>
+ <span class="hljs-string">#x27; -H &quot;X-TC-Version: &#x27;</span> + ver
sion + <span class="hljs-string">#x27;&quot;&#x27;</span>
+ <span class="hljs-string">#x27; -H &quot;X-TC-Region: &#x27;</span> + regi
on + <span class="hljs-string">#x27;&quot;&#x27;</span>
+ <span class="hljs-string">&quot;-d &#x27;&quot;</span> + payload + <span c
lass="hljs-string">&quot;&#x27;&quot;</span>
<span class="hljs-built_in">console</span>.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 OPENSSSL_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 OPENSSSL_VERSION_NUMBER < 0x10100000L
    HMAC_CTX_cleanup(h);
#else
    HMAC_CTX_free(h);
#endif
}

```

```

std::stringstream ss;
ss &&& std::setfill<>(<span class="hljs-built_in">0</span>);
<span class="hljs-keyword">for</span> (<span class="hljs-keyword">int</span>
i = <span class="hljs-number">0</span>; i <&lt; len; i++)
{
ss &&& hash[i];
}

<span class="hljs-keyword">return</span> (ss.<span class="hljs-built_in">str</span>());
}
string HexEncode(const string &input)
{
static const char* lut = "0123456789abcdef";
size_t len = input.length();

string output;
output.<span class="hljs-built_in">reserve</span>(<span class="hljs-number">2</span> * len);
<span class="hljs-keyword">for</span> (<span class="hljs-keyword">size_t</span>
i = <span class="hljs-number">0</span>; i <&lt; len; ++i)
{
<span class="hljs-keyword">const</span> <span class="hljs-keyword">unsigned</span>
<span class="hljs-keyword">char</span> c = input[i];
output.<span class="hljs-built_in">push_back</span>(lut[c >> <span class="hljs-number">4</span>]);
output.<span class="hljs-built_in">push_back</span>(lut[c & <span class="hljs-number">15</span>]);
}
<span class="hljs-keyword">return</span> output;
}

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

string service = <span class="hljs-string">"cvm"</span>;
string host = <span class="hljs-string">"cvm.tencentcloudapi.com"</span>;
string>;

```

```

string region = <span class="hljs-string">&quot;ap-guangzhou&quot;</span>;
string action = <span class="hljs-string">&quot;DescribeInstances&quot;</span>
>;
string version = <span class="hljs-string">&quot;2017-03-12&quot;</span>;
<span class="hljs-keyword">int64_t</span> timestamp = <span class="hljs-number">1551113065</span>;
string date = <span class="hljs-built_in">get_data</span>(timestamp);

<span class="hljs-comment">// ***** Step 1: Concatenate the Canonical
Request string *****</span>
string httpRequestMethod = <span class="hljs-string">&quot;POST&quot;</span>;
string canonicalUri = <span class="hljs-string">&quot;/&quot;</span>;
string canonicalQueryString = <span class="hljs-string">&quot;&quot;</span>;
string canonicalHeaders = <span class="hljs-string">&quot;content-type:applic
ation/json; charset=utf-8\nhost:&quot;</span> + host + <span class="hljs-strin
g">&quot;\n&quot;</span>;
string signedHeaders = <span class="hljs-string">&quot;content-type;host&quo
t;</span>;
string payload = <span class="hljs-string">&quot;{\&quot;Limit\&quot;: 1, \&
quot;Filters\&quot;: [{\&quot;Values\&quot;: [\&quot;unnamed\&quot;], \&quot;N
ame\&quot;: \&quot;instance-name\&quot;}]}&quot;</span>;
string hashedRequestPayload = <span class="hljs-built_in">sha256Hex</span>(pa
yload);
string canonicalRequest = httpRequestMethod + <span class="hljs-string">&quo
t;\n&quot;</span> + canonicalUri + <span class="hljs-string">&quot;\n&quot;</
span> + canonicalQueryString + <span class="hljs-string">&quot;\n&quot;</span>
+ canonicalHeaders + <span class="hljs-string">&quot;\n&quot;</span> + signed
Headers + <span class="hljs-string">&quot;\n&quot;</span> + hashedRequestPayl
oad;
cout &lt;&lt; canonicalRequest &lt;&lt; endl;
cout &lt;&lt; <span class="hljs-string">&quot;-----&quot;</
span> &lt;&lt; endl;

<span class="hljs-comment">// ***** Step 2: Concatenate the string to
sign *****</span>
string algorithm = <span class="hljs-string">&quot;TC3-HMAC-SHA256&quot;</spa
n>;
string RequestTimestamp = <span class="hljs-built_in">int2str</span>(timestam
p);
string credentialScope = date + <span class="hljs-string">&quot;/&quot;</span>
+ service + <span class="hljs-string">&quot;/&quot;</span> + <span class="hlj
s-string">&quot;tc3_request&quot;</span>;
string hashedCanonicalRequest = <span class="hljs-built_in">sha256Hex</span>
(canonicalRequest);
string stringToSign = algorithm + <span class="hljs-string">&quot;\n&quot;</s
pan> + RequestTimestamp + <span class="hljs-string">&quot;\n&quot;</span> + c

```

```

redentialScope + <span class="hljs-string">&quot;\n&quot;</span> + hashedCano
nicalRequest;
cout &lt;&lt; stringToSign &lt;&lt; endl;
cout &lt;&lt; <span class="hljs-string">&quot;-----&quot;</
span> &lt;&lt; endl;

<span class="hljs-comment">// ***** Step 3: Calculate the signature *
*****</span>
string kKey = <span class="hljs-string">&quot;TC3&quot;</span> + SECRET_KEY;
string kDate = <span class="hljs-built_in">HmacSha256</span>(kKey, date);
string kService = <span class="hljs-built_in">HmacSha256</span>(kDate, servic
e);
string kSigning = <span class="hljs-built_in">HmacSha256</span>(kService, <sp
an class="hljs-string">&quot;tc3_request&quot;</span>);
string signature = <span class="hljs-built_in">HexEncode</span>(<span class
="hljs-built_in">HmacSha256</span>(kSigning, stringToSign));
cout &lt;&lt; signature &lt;&lt; endl;
cout &lt;&lt; <span class="hljs-string">&quot;-----&quot;</
span> &lt;&lt; endl;

<span class="hljs-comment">// ***** Step 4: Concatenate the Authoriza
tion *****</span>
string authorization = algorithm + <span class="hljs-string">&quot; &quot;</s
pan> + <span class="hljs-string">&quot;Credential=&quot;</span> + SECRET_ID +
<span class="hljs-string">&quot;/&quot;</span> + credentialScope + <span clas
s="hljs-string">&quot;, &quot;</span>
+ <span class="hljs-string">&quot;SignedHeaders=&quot;</span> + signedHeaders
+ <span class="hljs-string">&quot;, &quot;</span> + <span class="hljs-strin
g">&quot;Signature=&quot;</span> + signature;
cout &lt;&lt; authorization &lt;&lt; endl;
cout &lt;&lt; <span class="hljs-string">&quot;-----&quot;
</span> &lt;&lt; endl;

string headers = <span class="hljs-string">&quot;curl -X POST https://&quot;
</span> + host + <span class="hljs-string">&quot;\n&quot;</span>
+ <span class="hljs-string">&quot;-H \&quot;Authorization: &quot;</span> + a
uthorization + <span class="hljs-string">&quot;\n&quot;</span>
+ <span class="hljs-string">&quot;-H \&quot;Content-Type: application/json;
charset=utf-8&quot;&quot;</span> + <span class="hljs-string">&quot;\n&quot;
</span>
+ <span class="hljs-string">&quot;-H \&quot;Host: &quot;</span> + host + <sp
an class="hljs-string">&quot;\n&quot;</span>
+ <span class="hljs-string">&quot;-H \&quot;X-TC-Action: &quot;</span> + act
ion + <span class="hljs-string">&quot;\n&quot;</span>
+ <span class="hljs-string">&quot;-H \&quot;X-TC-Timestamp: &quot;</span> +
RequestTimestamp + <span class="hljs-string">&quot;\n&quot;</span>
+ <span class="hljs-string">&quot;-H \&quot;X-TC-Version: &quot;</span> + ve

```

```
rsion + <span class="hljs-string">&quot;\n&quot;</span>
+ <span class="hljs-string">&quot; -H \&quot;X-TC-Region: &quot;</span> + reg
ion + <span class="hljs-string">&quot;\n&quot;</span>
+ <span class="hljs-string">&quot; -d &#x27;&quot;</span> + payload;
cout &lt;&lt; headers &lt;&lt; endl;
<span class="hljs-keyword">return</span> <span class="hljs-number">0</span>;
};
```

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: 2026-04-16 16:49:01

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:

```
7RAM2xfNMO9EiVTNmPg06MRnCvQ=
```

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 7RAM2xfNMO9EiVTNmPg06MRnCvQ=, the final signature string request parameter (Signature) is 7RAM2xfNMO9EiVTNmPg06MRnCvQ%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=7RAM2xfNMO9EiVTNmPg06MRnCvQ%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 lette
rs, only the value is URL encoded here.
url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHAR
SET)).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 enable
s 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 examp
le: params.put("Timestamp", System.currentTimeMillis() / 1000);
params.put("Timestamp", 1465185768); // Common parameter
params.put("SecretId", "AKID*****"); // Common paramet
er
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 = "*****";
$params["Nonce"] = 11886;//rand();
$params["Timestamp"] = 1465185768;//time();
$params["Region"] = "ap-guangzhou";
$params["SecretId"] = $secretId;
$params["Version"] = "2017-03-12";
$params["Action"] = "DescribeInstances";
$params["InstanceIds.0"] = "ins-09dx96dg";
$params["Limit"] = 20;
$params["Offset"] = 0;

ksort($params);

$signStr = "GETcvm.tencentcloudapi.com/?";
foreach ($params 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
// $params["Signature"] = $signature;
// $url = "https://cvm.tencentcloudapi.com/?".http_build_query($params);
// 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, sign))
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> requestParams, 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>(param, 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: 2026-04-16 16:49:01

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
<code>AuthFailure.InvalidSecretId</code>	Invalid key (not a TencentCloud API key type).
<code>AuthFailure.MFAFailure</code>	MFA failed.
<code>AuthFailure.SecretIdNotFound</code>	The key does not exist.
<code>AuthFailure.SignatureExpire</code>	Signature expired.
<code>AuthFailure.SignatureFailure</code>	Signature error.
<code>AuthFailure.TokenFailure</code>	Token error.
<code>AuthFailure.UnauthorizedOperation</code>	The request does not have CAM authorization.
<code>DryRunOperation</code>	DryRun Operation. It means that the request would have succeeded, but the <code>DryRun</code> parameter was used.
<code>FailedOperation</code>	Operation failed.
<code>InternalError</code>	Internal error.
<code>InvalidAction</code>	The API does not exist.
<code>InvalidParameter</code>	Incorrect parameter.
<code>InvalidParameterValue</code>	Invalid parameter value.
<code>LimitExceeded</code>	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.

Rule APIs

PutEvaluations

Last updated: 2026-04-16 17:09:17

1. API Description

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

This API is used to report custom rule evaluation results.

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: PutEvaluations.
Version	Yes	String	Common Params . The value used for this API: 2022-08-02.
Region	Yes	String	Common Params . For more information, please see the list of regions supported by the product.
ResultToken	Yes	String	Callback token. Obtained from the ResultToken value in the Context of the selected Serverless Cloud Function (SCF) for the custom rule.

Evaluations.N	Yes	Array of Evaluation	Custom rule evaluation result information.
---------------	-----	-------------------------------------	--

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 Reporting Custom Rule Evaluation Results

This example shows you how to report custom rule evaluation results.

Input Example

```
POST / HTTP/1.1
Host: config.intl.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: PutEvaluations
<Common request parameters>
```

```
{
  "ResultToken": "Wm9yZlY3WmlKa3cxaW1oQ1u-H3WA6JZnH46cUAn2DWGJ9dp3GwJyhMyXsJyeXRcqa8sCimZKH6hy-7RGW_DEtva2Bjd36ryyDg0tbNOFYpf3r6lJB_gHRUCPRrr8HQbEvCoeoGw-KH7LqNUsoa4GzyrvKx2Ak1vycWzfAGaiTISWoTQ2mYr_BYSSvc00771U1Y4hieGJyolSxxUf1V9fDoIXFQikxW6AmU4cTUUpEJ-OF4Mvbq_7quPYZQOjDuo9cAxxUX-D8w==",
  "Evaluations": [
    {
      "ComplianceResourceId": "disk-26itbqha",
```

```
"ComplianceResourceType": "QCS::CBS::Disk",
"ComplianceRegion": "ap-guangzhou",
"ComplianceType": "NON_COMPLIANT",
"Annotation": {
  "Configuration": "1",
  "DesiredValue": "2",
  "Operator": "equal",
  "Property": "age"
}
}
]
}
```

Output Example

```
{
  "Response": {
    "RequestId": "d947eba9-f908-4d2e-9b3d-63bde43abd1a"
  }
}
```

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
InvalidParameter	Parameter error.
ResourceNotFound.AccountGroupsNotExist	Account group does not exist.
ResourceNotFound.ResourceNotExist	The resource does not exist.
ResourceNotFound.RulesNotExist	The rule does not exist.

ListConfigRules

Last updated: 2026-04-16 16:49:03

1. API Description

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

This API is used to get the rule list.

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: ListConfigRules.
Version	Yes	String	Common Params . The value used for this API: 2022-08-02.
Region	Yes	String	Common Params . For more information, please see the list of regions supported by the product.
Limit	Yes	Integer	Page limit
Offset	Yes	Integer	Offset.
OrderType	No	String	Sort type. Descending: desc, Ascending: asc.
RiskLevel.N	No	Array of Integer	Risk level 1: High risk.

			2: Medium risk. 3: Low risk.
State	No	String	Rule status
ComplianceResult.N	No	Array of String	Evaluation result
RuleName	No	String	Name of the rule

3. Output Parameters

Parameter Name	Type	Description
Total	Integer	Total number
Items	Array of ConfigRule	Details
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 Rule List

Input Example

```
POST / HTTP/1.1
Host: config.intl.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ListConfigRules
<Common request parameters>

{
  "Offset": 0,
  "Limit": 10,
  "OrderType": "",
  "RiskLevel": [],
  "State": "ACTIVE",
```

```
"ComplianceResult": [],  
"RuleName": ""  
}
```

Output Example

```
{  
  "Response": {  
    "RequestId": "149e116a-90ef-45f3-9d5d-3d77fd1c9eb3",  
    "Items": [  
      {  
        "Annotation": null,  
        "CompliancePackId": "cp-xzfv0vu007feuhwi8auv",  
        "CompliancePackName": "Compliance 1",  
        "ComplianceResult": "NON_COMPLIANT",  
        "ConfigRuleId": "cr-13vkg9c31dixgabkepxe",  
        "ConfigRuleInvokedTime": null,  
        "CreateTime": "2022-11-16 14:25:01",  
        "Description": "The user must be associated with at least one user group in CAM a  
ccess management to comply with the rule.",  
        "Identifier": "cam-user-group-bound",  
        "IdentifierType": "SYSTEM",  
        "InputParameter": [],  
        "Labels": [],  
        "ManageInputParameter": [],  
        "ResourceType": [  
          "QCS::CAM::User"  
        ],  
        "RiskLevel": 3,  
        "RuleName": "Associate CAM sub-user with user group"  
        "ServiceFunction": null,  
        "SourceCondition": [  
          {  
            "DesiredValue": "1",  
            "EmptyAs": "COMPLIANT",  
            "Operator": "GreaterOrEquals",  
            "Required": false,  
            "SelectPath": "$User.GroupBindNum"  
          }  
        ],  
        "Status": "ACTIVE",  
        "TriggerType": [  
          {  
            "MaximumExecutionFrequency": "TwentyFour_Hours",  
            "MessageType": "ScheduledNotification"  
          }  
        ]  
      }  
    ]  
  }  
}
```

```
]
},
{
  "Annotation": null,
  "CompliancePackId": "",
  "CompliancePackName": null,
  "ComplianceResult": "NON_COMPLIANT",
  "ConfigRuleId": "cr-bdunf5kx3aywn0ac5bkk",
  "ConfigRuleInvokedTime": null,
  "CreateTime": "2022-11-16 14:22:59",
  "Description": "In CAM, a user must be associated with at least one user group to comply with the rule."
  "Identifier": "cam-user-group-bound",
  "IdentifierType": "SYSTEM",
  "InputParameter": [],
  "Labels": [],
  "ManageInputParameter": [],
  "ResourceType": [
    "QCS::CAM::User"
  ],
  "RiskLevel": 3,
  "RuleName": "Associate CAM sub-user with user group"
  "ServiceFunction": null,
  "SourceCondition": [
    {
      "DesiredValue": "1",
      "EmptyAs": "COMPLIANT",
      "Operator": "GreaterOrEquals",
      "Required": false,
      "SelectPath": "$User.GroupBindNum"
    }
  ],
  "Status": "ACTIVE",
  "TriggerType": [
    {
      "MaximumExecutionFrequency": "TwentyFour_Hours",
      "MessageType": "ScheduledNotification"
    }
  ]
},
{
  "Annotation": null,
  "CompliancePackId": "",
  "CompliancePackName": null,
  "ComplianceResult": "NON_COMPLIANT",
  "ConfigRuleId": "cr-2d3brhnyvazqb9j1e16o",
  "ConfigRuleInvokedTime": null,
```

```
"CreateTime": "2022-11-16 11:36:45",
"Description": "In CAM, a user must be associated with at least one user group to
comply with the rule."
"Identifier": "cam-user-group-bound",
"IdentifierType": "SYSTEM",
"InputParameter": [],
"Labels": [],
"ManageInputParameter": [],
"ResourceType": [
"QCS::CAM::User"
],
"RiskLevel": 3,
"RuleName": "Associate CAM sub-user with user group"
"ServiceFunction": null,
"SourceCondition": [
{
"DesiredValue": "1",
"EmptyAs": "COMPLIANT",
"Operator": "GreaterOrEquals",
"Required": false,
"SelectPath": "$User.GroupBindNum"
}
],
"Status": "ACTIVE",
"TriggerType": [
{
"MaximumExecutionFrequency": "TwentyFour_Hours",
"MessageType": "ScheduledNotification"
}
]
],
"Total": 3
}
}
```

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
InvalidParameter	Parameter error.

ListAggregateConfigRules

Last updated: 2026-04-16 16:49:03

1. API Description

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

This API is used to get the account group rule list.

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: ListAggregateConfigRules.
Version	Yes	String	Common Params . The value used for this API: 2022-08-02.
Region	Yes	String	Common Params . For more information, please see the list of regions supported by the product.
Limit	Yes	Integer	Specifies the limit per page.
Offset	Yes	Integer	Offset.
AccountGroupId	Yes	String	Account group ID
OrderType	No	String	Sort type, descending: desc, ascending: asc.

RiskLevel.N	No	Array of Integer	Risk level 1: High risk. 2: Medium risk. 3: Low risk.
State	No	String	Rule status
ComplianceResult.N	No	Array of String	Evaluation result
RuleName	No	String	Name of the rule
RuleOwnerId	No	Integer	Rule ownership account ID

3. Output Parameters

Parameter Name	Type	Description
Total	Integer	Total number
Items	Array of ConfigRule	Details
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 Rule List of Account Group

This example shows you how to get the rule list of an account group.

Input Example

```
POST / HTTP/1.1
Host: config.intl.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ListAggregateConfigRules
<Common request parameters>
```

```
{
  "Offset": 0,
  "Limit": 10,
  "OrderType": "",
  "RiskLevel": [],
  "State": "ACTIVE",
  "ComplianceResult": [
    "abc"
  ],
  "RuleName": "",
  "RuleOwnerId": 1,
  "AccountGroupId": "ca-sdfs7734h24h3"
}
```

Output Example

```
{
  "Response": {
    "RequestId": "149e116a-90ef-45f3-9d5d-3d77fd1c9eb3",
    "Items": [
      {
        "RegionsScope": [
          "ap-shanghai"
        ],
        "TagsScope": [
          {
            "TagKey": "tag1",
            "TagValue": "tag2"
          }
        ],
        "ExcludeResourceIdsScope": [
          "ins-asdasd"
        ],
        "Annotation": null,
        "CompliancePackId": "cp-xzfv0vu007feuhwi8auv",
        "CompliancePackName": "Compliance 1",
        "ComplianceResult": "NON_COMPLIANT",
        "ConfigRuleId": "cr-13vkg9c31dixgabkepxe",
        "ConfigRuleInvokedTime": null,
        "CreateTime": "2022-11-16 14:25:01",
        "Description": "The rule is met if the user in account access management is associated with at least one user group.",
        "Identifier": "cam-user-group-bound",

```

```
"IdentifierType": "SYSTEM",
"InputParameter": [],
"Labels": [],
"ManageInputParameter": [],
"ResourceType": [
  "QCS::CAM::User"
],
"RiskLevel": 3,
"RuleName": "Associate CAM Sub-user with User Group"
"ServiceFunction": null,
"SourceCondition": [
  {
    "DesiredValue": "1",
    "EmptyAs": "COMPLIANT",
    "Operator": "GreaterOrEquals",
    "Required": false,
    "SelectPath": "$User.GroupBindNum"
  }
],
"Status": "ACTIVE",
"TriggerType": [
  {
    "MaximumExecutionFrequency": "TwentyFour_Hours",
    "MessageType": "ScheduledNotification"
  }
]
}
],
"Total": 1
}
}
```

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
InvalidParameter	Parameter error.
ResourceNotFound.AccountGroupIsNotExist	Account group does not exist.

Resource APIs

ListDiscoveredResources

Last updated: 2026-04-16 16:49:04

1. API Description

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

This API is used to get the resource list.

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: ListDiscoveredResources.
Version	Yes	String	Common Params . The value used for this API: 2022-08-02.
Region	Yes	String	Common Params . For more information, please see the list of regions supported by the product.
MaxResults	Yes	Integer	Items per Page
Filters.N	No	Array of Filter	resourceName: Resource name resourceId: Resource ID

Tags.N	No	Array of Tag	Tag
NextToken	No	String	Next page token.
OrderType	No	String	Sorting method asc, desc

3. Output Parameters

Parameter Name	Type	Description
Items	Array of ResourceListInfo	Details
NextToken	String	Next page Note: This field may return null, indicating that no valid value is found.
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 resource list

Input Example

```
POST / HTTP/1.1
Host: config.intl.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ListDiscoveredResources
<Common request parameters>

{
  "OrderType": "xx",
  "NextToken": "xx",
  "MaxResults": 1,
  "Filters": [
    {
```

```
"Values": [  
  "xx"  
],  
"Name": "xx"  
}  
],  
"Tags": [  
  {  
    "TagKey": "xx",  
    "TagValue": "xx"  
  }  
]  
}
```

Output Example

```
{  
  "Response": {  
    "Items": [  
      {  
        "ResourceStatus": "xx",  
        "Tags": [  
          {  
            "TagKey": "xx",  
            "TagValue": "xx"  
          }  
        ],  
        "ResourceType": "xx",  
        "ResourceId": "xx",  
        "ResourceCreateTime": 1,  
        "ResourceRegion": "xx",  
        "ResourceName": "xx",  
        "ResourceZone": "xx",  
        "ResourceDelete": 1,  
        "ComplianceResult": "xx"  
      }  
    ],  
    "NextToken": "xx",  
    "RequestId": "xx"  
  }  
}
```

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
InvalidParameter	Parameter error.

DescribeDiscoveredResource

Last updated: 2026-04-16 16:49:05

1. API Description

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

Resource 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: DescribeDiscoveredResource.
Version	Yes	String	Common Params . The value used for this API: 2022-08-02.
Region	Yes	String	Common Params . For more information, please see the list of regions supported by the product.
ResourceId	Yes	String	Resource ID
ResourceType	Yes	String	Resource type
ResourceRegion	Yes	String	Resource region

3. Output Parameters

Parameter Name	Type	Description
ResourceId	String	Resource ID Note: This field may return null, indicating that no valid value is found.
ResourceType	String	Resource type Note: This field may return null, indicating that no valid value is found.
ResourceName	String	Resource Name Note: This field may return null, indicating that no valid value is found.
ResourceRegion	String	Resource region Note: This field may return null, indicating that no valid value is found.
ResourceZone	String	Resource availability zone Note: This field may return null, indicating that no valid value is found.
Configuration	String	Resource configuration Note: This field may return null, indicating that no valid value is found.
ResourceCreateTime	String	Resource creation time Note: This field may return null, indicating that no valid value is found.
Tags	Array of Tag	Resource tag Note: This field may return null, indicating that no valid value is found.
UpdateTime	String	Resource update time Note: This field may return null, indicating that no valid value is found.
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 Resource Details

This example shows you how to get resource details.

Input Example

```
POST / HTTP/1.1
Host: config.intl.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeDiscoveredResource
<Common request parameters>

{
  "ResourceId": "ins-2av11cxx",
  "ResourceType": "QCS::CVM::Instance",
  "ResourceRegion": "ap-guangzhou"
}
```

Output Example

```
{
  "Response": {
    "RequestId": "2ea2d804-d278-4631-b04d-70bd40e1a478",
    "Configuration": "{\"InstanceState\":\"RUNNING\", \"InstanceName\":\"Unnamed\", \"InstanceId\":\"ins-111\", \"InstanceType\":\"S5.SMALL2\", \"VirtualPrivateCloud\": {\"VpcId\":\"vpc-333\", \"SubnetId\":\"subnet-qxupkefw\", \"AsVpcGateway\": false, \"PrivateIpAddresses\": null, \"Ipv6AddressCount\": 0}, \"PrivateIpAddresses\": [\"111\"], \"PublicIpAddresses\": null, \"OsName\":\"TencentOS Server 4 for x86_64\", \"Memory\": 2, \"CPU\": 1, \"InternetAccessible\": {\"InternetChargeType\": \"\", \"InternetMaxBandwidthOut\": 0, \"PublicIpAssigned\": false, \"BandwidthPackageId\": \"\"}, \"ImageId\": \"img-333\", \"InstanceChargeType\": \"PREPAID\", \"SecurityGroupIds\": [\"sg-222\"], \"DataDisks\": null, \"SystemDisk\": {\"DiskType\": \"CLOUD_PREMIUM\", \"DiskId\": \"disk-auh4557w\"}, \"ExpiredTime\": \"2024-12-28T08:07:12Z\", \"RenewFlag\": \"NOTIFY_AND_MANUAL_RENEW\", \"LatestOperation\": \"\", \"Placement\": {\"ProjectId\": 0}, \"LatestOperationState\": \"\", \"CamRoleName\": \"\"}"
    "ResourceCreateTime": "2024-11-28 16:07:12",
    "ResourceId": "ins-2av11cxx",
    "ResourceName": "Unnamed"
    "ResourceRegion": "ap-guangzhou",
```

```
"ResourceType": "QCS::CVM::Instance",
"ResourceZone": "",
"Tags": [],
"UpdateTime": "2024-11-28 16:08:36"
}
}
```

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
InvalidParameter	Parameter error.
ResourceNotFound.ResourceNotExist	The resource does not exist.

ListAggregateDiscoveredResources

Last updated: 2026-04-16 16:49:04

1. API Description

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

Account Group access the list of resources.

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: ListAggregateDiscoveredResources.
Version	Yes	String	Common Params . The value used for this API: 2022-08-02.
Region	Yes	String	Common Params . For more information, please see the list of regions supported by the product.
MaxResults	Yes	Integer	Items per Page
AccountGroupid	Yes	String	Account group ID
Filters.N	No	Array of Filter	resourceName: Resource name; resourceid: Resource ID; resourceType: Resource type

Tags.N	No	Array of Tag	
NextToken	No	String	Next page token.
OrderType	No	String	Sorting method asc, desc

3. Output Parameters

Parameter Name	Type	Description
Items	Array of AggregateResourceInfo	Details.
NextToken	String	next page Note: This field may return null, indicating that no valid value is found.
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 resource list

This API is used to get the resource list.

Input Example

```
POST / HTTP/1.1
Host: config.intl.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ListAggregateDiscoveredResources
<Common request parameters>

{
  "AccountGroupId": "ca-sdfsdfsdf",
  "OrderType": "DESC",
  "Tags": [
```

```
{
  "TagKey": "TAG1",
  "TagValue": "TAG3"
},
"MaxResults": 1,
"Filters": [
  {
    "Values": [
      "CVM"
    ],
    "Name": "resourceType"
  }
],
"NextToken": "sdfsdfsdf456457rsf"
}
```

Output Example

```
{
  "Response": {
    "Items": [
      {
        "ResourceOwnerName": "nickname",
        "ResourceStatus": "1",
        "ResourceOwnerId": 1,
        "Tags": [
          {
            "TagKey": "TAG2",
            "TagValue": "33ATG5"
          }
        ],
        "ResourceType": "cvm",
        "ResourceId": "ins-324234",
        "ResourceCreateTime": "234234234234",
        "ResourceRegion": "ap-hangzhou",
        "ResourceName": "server"
        "ResourceZone": "ap-guangzhou",
        "ResourceDelete": 1,
        "ComplianceResult": "COMPLIANCE"
      }
    ],
    "NextToken": "0f6ac54682ee49d5b0",
    "RequestId": "3d105d8-5820-434f-9fac-76f92"
  }
}
```

```
}  
}
```

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
InternalServerError	Internal error.
InvalidParameter	Parameter error.
ResourceNotFound.AccountGroupsNotExist	Account group does not exist.

Data Types

Last updated: 2026-04-16 16:49:05

AggregateResourceInfo

Resource list information response parameters structure

Used by actions: ListAggregateDiscoveredResources.

Name	Type	Description
ResourceType	String	Resource type
ResourceName	String	Resource name
ResourceId	String	Resource ID
ResourceRegion	String	Region Note: This field may return null, indicating that no valid value is found.
ResourceStatus	String	Resource Status Note: This field may return null, indicating that no valid value is found.
ResourceDelete	Integer	Whether to delete. 1: Deleted; 0: Not deleted. Note: This field may return null, indicating that no valid value is found.
ResourceCreateTime	String	Resource creation time Note: This field may return null, indicating that no valid value is found.
Tags	Array of Tag	Tag information Note: This field may return null, indicating that no valid value is found.
ResourceZone	String	Availability zone

		Note: This field may return null, indicating that no valid value is found.
ComplianceResult	String	Compliance status Note: This field may return null, indicating that no valid value is found.
ResourceOwnerId	Integer	Resource owner uid
ResourceOwnerName	String	User nickname Note: This field may return null, indicating that no valid value is found.

Annotation

Compliance details

Used by actions: ListAggregateConfigRules, ListConfigRules, PutEvaluations.

Name	Type	Required	Description
Configuration	String	Yes	Current actual configuration of the resource. It can contain 0 to 256 characters, which is the non-compliant configuration of the resource. Note: This field may return null, indicating that no valid value is found.
DesiredValue	String	Yes	Desired configuration of the resource. It can contain 0 to 256 characters, which is the compliant configuration of the resource. Note: This field may return null, indicating that no valid value is found.
Operator	String	No	Comparison operator between current and desired configuration of the resource. Length is 0–16 characters. This field may be empty when custom rule reporting evaluation result.
Property	String	No	JSON path of current configuration in resource attribute structure. Length is 0–256 characters. This field may be empty when custom rule reporting evaluation result.

ConfigRule

Rule details

Used by actions: ListAggregateConfigRules, ListConfigRules.

Name	Type	Description
Identifier	String	Rule identifier Note: This field may return null, indicating that no valid value is found.
RuleName	String	Name of the rule Note: This field may return null, indicating that no valid value is found.
InputParameter	Array of InputParameter	Rule parameters Note: This field may return null, indicating that no valid value is found.
SourceCondition	Array of SourceConditionForManage	Rule trigger condition. Note: This field may return null, indicating that no valid value is found.
ResourceType	Array of String	Resource types supported by rule. The rule only applies to specified resource types. Note: This field may return null, indicating that no valid value is found.
Labels	Array of String	Rule ownership tag Note: This field may return null, indicating that no valid value is found.
RiskLevel	Integer	Rule risk level 1: Low risk 2: Medium risk 3: High risk Note: This field may return null, indicating that no valid value is found.
ServiceFunction	String	Function corresponding to rule Note: This field may return null, indicating that no valid value is found.
CreateTime	String	Creation time

		Format: YYYY-MM-DD h:i:s Note: This field may return null, indicating that no valid value is found.
Description	String	Rule description Note: This field may return null, indicating that no valid value is found.
Status	String	ACTIVE: Enabled NO_ACTIVE: Disabled Note: This field may return null, indicating that no valid value is found.
ComplianceResult	String	Compliance: 'COMPLIANT' 'NON_COMPLIANT' 'NOT_APPLICABLE' Note: This field may return null, indicating that no valid value is found.
Annotation	Annotation	["", ""] Note: This field may return null, indicating that no valid value is found.
ConfigRuleInvokedTime	String	Rule evaluation time Format: YYYY-MM-DD h:i:s Note: This field may return null, indicating that no valid value is found.
ConfigRuleId	String	Rule ID Note: This field may return null, indicating that no valid value is found.
IdentifierType	String	CUSTOMIZE Managed rule Note: This field may return null, indicating that no valid value is found.
CompliancePackId	String	Compliance package ID Note: This field may return null, indicating that no valid value is found.
TriggerType	Array of TriggerType	Trigger Type Scheduled trigger

		Triggered by configuration change Note: This field may return null, indicating that no valid value is found.
ManageInputParameter	Array of InputParameterForManage	Parameter details Note: This field may return null, indicating that no valid value is found.
CompliancePacName	String	Rule name Note: This field may return null, indicating that no valid value is found.
RegionsScope	Array of String	Associated region Note: This field may return null, indicating that no valid value is found.
TagsScope	Array of Tag	Associate Tag Note: This field may return null, indicating that no valid value is found.
ExcludeResourceIdsScope	Array of String	The rule is invalid for the specified resource ID, meaning it does not evaluate the resource. Note: This field may return null, indicating that no valid value is found.
AccountGroupID	String	Account group ID Note: This field may return null, indicating that no valid value is found.
AccountGroupName	String	Account group name Note: This field may return null, indicating that no valid value is found.
RuleOwnerID	Integer	Rule owner user ID Note: This field may return null, indicating that no valid value is found.
ManageTriggerType	Array of String	Trigger methods supported by preset rules Scheduled trigger Triggered by configuration change

Evaluation

Custom rule evaluation result

Used by actions: PutEvaluations.

Name	Type	Required	Description
ComplianceResourceid	String	Yes	Evaluated resource id. It can contain 0 to 256 characters.
ComplianceResourceType	String	Yes	Evaluated resource type. Supported: QCS::CVM::Instance、 QCS::CBS::Disk、 QCS::VPC::Vpc、 QCS::VPC::Subnet、 QCS::VPC::SecurityGroup、 QCS::CAM::User、 QCS::CAM::Group、 QCS::CAM::Policy、 QCS::CAM::Role、 QCS::COS::Bucket
ComplianceRegion	String	Yes	Evaluated resource region. It can contain 0 to 32 characters.
ComplianceType	String	Yes	Compliance type. Valid values: COMPLIANT: Compliant, NON_COMPLIANT: Non-compliant
Annotation	Annotation	No	Supplementary information for non-compliant resources.

Filter

Resource list filter

Used by actions: ListAggregateDiscoveredResources, ListDiscoveredResources.

Name	Type	Required	Description
Name	String	No	Query field name Resource name: resourceName Resource ID: resourceid Resource type: resourceType Resource region: resourceRegion Deletion status: resourceDelete 0 not deleted, 1 deleted resourceregionandzone region/az
Values	Array of	No	Value of the field to query

String

InputParameter

Parameter value

Used by actions: ListAggregateConfigRules, ListConfigRules.

Name	Type	Required	Description
ParameterKey	String	Yes	Parameter name
Type	String	No	Parameter type. Required type: Require, optional type: Optional.
Value	String	No	Parameter value Note: This field may return null, indicating that no valid value is found.

InputParameterForManage

Rule input parameters

Used by actions: ListAggregateConfigRules, ListConfigRules.

Name	Type	Required	Description
ValueType	String	No	Value type. Integer: Integer, String: String. Note: This field may return null, indicating that no valid value is found.
ParameterKey	String	No	Parameter key Note: This field may return null, indicating that no valid value is found.
Type	String	No	Parameter type. Required type: Required, Optional type: Optional. Note: This field may return null, indicating that no valid value is found.
DefaultValue	String	No	Default value

			Note: This field may return null, indicating that no valid value is found.
Description	String	No	Description Note: This field may return null, indicating that no valid value is found.

ResourceListInfo

Resource list information response parameters structure

Used by actions: ListDiscoveredResources.

Name	Type	Description
ResourceType	String	Resource type
ResourceName	String	Resource name
ResourceId	String	Resource ID
ResourceRegion	String	Region Note: This field may return null, indicating that no valid value is found.
ResourceStatus	String	Resource Status Note: This field may return null, indicating that no valid value is found.
ResourceDelete	Integer	1: Deleted. 2: Not deleted. Note: This field may return null, indicating that no valid value is found.
ResourceCreateTime	String	Resource creation time Note: This field may return null, indicating that no valid value is found.
Tags	Array of Tag	Tag information Note: This field may return null, indicating that no valid value is found.

ResourceZone	String	Availability zone Note: This field may return null, indicating that no valid value is found.
ComplianceResult	String	Compliance status. Note: This field may return null, indicating that no valid value is found.

SourceConditionForManage

Managing end rule conditions

Used by actions: ListAggregateConfigRules, ListConfigRules.

Name	Type	Required	Description
EmptyAs	String	No	Condition is empty, Compliant: COMPLIANT, Non-compliant: NON_COMPLIANT, Not applicable: NOT_APPLICABLE. Note: This field may return null, indicating that no valid value is found.
SelectPath	String	No	Configuration path Note: This field may return null, indicating that no valid value is found.
Operator	String	No	Operators Note: This field may return null, indicating that no valid value is found.
Required	Boolean	No	Required or not. Note: This field may return null, indicating that no valid value is found.
DesiredValue	String	No	Expected value Note: This field may return null, indicating that no valid value is found.

Tag

Tag

Used by actions: DescribeDiscoveredResource, ListAggregateConfigRules, ListAggregateDiscoveredResources, ListConfigRules, ListDiscoveredResources.

Name	Type	Required	Description
TagKey	String	No	Tag key Note: This field may return null, indicating that no valid value is found.
TagValue	String	No	Tag value Note: This field may return null, indicating that no valid value is found.

TriggerType

Rule supports trigger type

Used by actions: ListAggregateConfigRules, ListConfigRules.

Name	Type	Required	Description
MessageType	String	Yes	Trigger Type
MaximumExecutionFrequency	String	No	Trigger time period Note: This field may return null, indicating that no valid value is found.

Error Codes

Last updated: 2026-04-16 16:49:06

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	<code>Authorization</code> 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
ResourceNotFound.AccountGroupsNotExist	Account group does not exist.
ResourceNotFound.ResourceNotExist	The resource does not exist.
ResourceNotFound.RulesNotExist	The rule does not exist.