

自动化助手

API 文档

产品文档



腾讯云

【版权声明】

©2013-2024 腾讯云版权所有

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

【商标声明】

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

【服务声明】

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

文档目录

API 文档

History

Introduction

API Category

Making API Requests

Request Structure

Common Params

Signature v3

Signature

Responses

Remote Command APIs

EnableInvoker

CreateInvoker

DescribeCommands

DescribeAutomationAgentStatus

DeleteCommand

CreateCommand

DescribeInvocationTasks

CancelInvocation

DescribeInvocations

InvokeCommand

ModifyCommand

ModifyInvoker

DeleteInvoker

PreviewReplacedCommandContent

DisableInvoker

DescribeInvokers

DescribeInvokerRecords

RunCommand

Region APIs

DescribeRegions

Data Types

Error Codes

API 文档

History

最近更新时间：2023-03-14 16:55:44

Release 2

Release time: 2023-03-14 16:53:02

Release updates:

Improvement to existing documentation.

Modified data structures:

- [AutomationAgentInfo](#)
 - New members:SupportFeatures
- [CommandDocument](#)
 - New members:OutputCOSBucketUrl, OutputCOSKeyPrefix

Release 1

Release time: 2022-04-24 16:57:09

Release updates:

Improvement to existing documentation.

New APIs:

- [CancelInvocation](#)
- [CreateCommand](#)
- [CreateInvoker](#)
- [DeleteCommand](#)
- [DeleteInvoker](#)
- [DescribeAutomationAgentStatus](#)
- [DescribeCommands](#)
- [DescribeInvocationTasks](#)
- [DescribeInvocations](#)
- [DescribeInvokerRecords](#)

- [DescribeInvokers](#)
- [DescribeRegions](#)
- [DisableInvoker](#)
- [EnableInvoker](#)
- [InvokeCommand](#)
- [ModifyCommand](#)
- [ModifyInvoker](#)
- [PreviewReplacedCommandContent](#)
- [RunCommand](#)

New data structures:

- [AutomationAgentInfo](#)
- [Command](#)
- [CommandDocument](#)
- [Filter](#)
- [Invocation](#)
- [InvocationTask](#)
- [InvocationTaskBasicInfo](#)
- [Invoker](#)
- [InvokerRecord](#)
- [RegionInfo](#)
- [ScheduleSettings](#)
- [Tag](#)
- [TaskResult](#)

Introduction

最近更新时间：2023-03-14 16:55:43

Efficient and secure native Ops and deployment tool.

API Category

最近更新时间：2023-03-14 16:55:43

Remote Command APIs

API Name	Feature	Frequency Limit
CancelInvocation	Cancels the execution of a command	20
CreateCommand	Creates a command	20
CreateInvoker	Creates an invoker	20
DeleteCommand	Deletes a command	20
DeleteInvoker	Deletes an invoker	20
DescribeAutomationAgentStatus	Queries the agent status	20
DescribeCommands	Queries command details	20
DescribeInvocationTasks	Queries the execution tasks	20
DescribeInvocations	Queries the execution activities	20
DescribeInvokerRecords	Queries the execution history of an invoker	20
DescribeInvokers	Queries invokers	20
DisableInvoker	Disables an invoker	20
EnableInvoker	Enables an invoker	20
InvokeCommand	Triggers a command	20
ModifyCommand	Modifies a command	20
ModifyInvoker	Modifies an invoker	20
PreviewReplacedCommandContent	Previews a command	20
RunCommand	Executes a command	20

Region APIs

API Name	Feature	Frequency Limit
DescribeRegions	Queries regions that support TAT	20

Making API Requests

Request Structure

最近更新时间：2023-03-14 16:55:44

1. Service Address

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

South Asia (Mumbai)	tat.ap-mumbai.tencentcloudapi.com
Northeast Asia (Seoul)	tat.ap-seoul.tencentcloudapi.com
Northeast Asia (Tokyo)	tat.ap-tokyo.tencentcloudapi.com
U.S. East Coast (Virginia)	tat.na-ashburn.tencentcloudapi.com
U.S. West Coast (Silicon Valley)	tat.na-siliconvalley.tencentcloudapi.com
North America (Toronto)	tat.na-toronto.tencentcloudapi.com
Europe (Frankfurt)	tat.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

最近更新时间：2023-03-14 16:55:44

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

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

Common parameters for Signature Algorithm v3

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

Parameter Name	Type	Required	Description
X-TC-Action	String	Yes	The name of the API for the desired operation. For the specific value, see description of common parameter <code>Action</code> in the input parameters in r 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 y work with belongs. For values supported for an API, see the description c parameter <code>Region</code> in the input parameters in related API documentati parameter is not required for some APIs (which will be indicated in relatec documentation), and will not take effect even it is passed.
X-TC-Timestamp	Integer	Yes	The current UNIX timestamp that records the time when the API request for example, 1529223702. Note: If the difference between the UNIX times server time is greater than 5 minutes, a signature expiration error may oc
X-TC-Version	String	Yes	API version of the action. For the valid values, see the description of the c parameter <code>Version</code> in the API documentation. For example, the versi 2017-03-12.
Authorization	String	Yes	The HTTP authentication request header, for example: TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_requ SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc96317 Here: - TC3-HMAC-SHA256: Signature method, currently fixed as this value; - Credential: Signature credential; AKIDEXAMPLE is the SecretId; Date is UTC time, and this value must match the value of X-TC-Timestamp (a co

			parameter) in UTC time format; service is the name of the product/service generally a domain name prefix. For example, a domain name cvm.tencent refers to the CVM product and the value would be cvm; - SignedHeaders: The headers that contains the authentication information type and host are the required headers; - Signature: Signature digest.
X-TC-Token	String	No	The token used for a temporary certificate. It must be used with a 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=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
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=AKIDEXAMPLE/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
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=AKIDEXAMPLE/2018-05-30/cvm/tc3_request,
SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b1316
2d9279b0855cfba6789a8edb4c
```

```
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=37ac2f4fde00b0ac9bd9eadeb459b1bbec224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKIDEXAMPLE
```

```
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=AKIDEXAMPLE
```

Signature v3

最近更新时间：2022-05-23 15:21:24

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 `AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****` and `Gu5t9xGARNpq86cd98joQYCN3*****`, 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=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
*/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=c492e8e4
1437e97a620b728c301bb8d17e7dc0c17eeabce80c20cd70fc3a78ff" \
-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: <code>Limit=10&Offset=0</code>.</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 server will return an error indicating 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	Hash value of the request payload (i.e., the body, such as <code>{"Limit": 1, "Filter</code>

```

[{"Values": ["unnamed"], "Name": "instance-name"}]} in this example
The pseudocode for calculation is
Lowercase(HexEncode(Hash.SHA256(RequestPayload))) by SHA256 hashing the pay
of the HTTP request, performing hexadecimal encoding, and finally converting the encc
string to lowercase letters. For GET requests, RequestPayload is always an empt
string. The calculation result in this example is
99d58dfbc6745f6747f36bfca17dee5e6881dc0428a0a36f96199342bc5b4907
    
```

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

POST

/

content-type:application/json; charset=utf-8

host:cvm.tencentcloudapi.com

content-type;host

99d58dfbc6745f6747f36bfca17dee5e6881dc0428a0a36f96199342bc5b4907

2. Concatenating the String to Be Signed

The string to sign is concatenated as follows:

```

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

Field Name	Explanation
Algorithm	Signature algorithm, which is currently always TC3-HMAC-SHA256 .
RequestTimestamp	Request timestamp, i.e., the value of the common parameter X-TC-Timestamp in request header, which is the UNIX timestamp of the current time in seconds, such as 1551113065 in this example.
CredentialScope	Scope of the credential in the format of Date/service/tc3_request , including date, requested service and termination string (tc3_request). Date is a date in UTC time, whose value should match the UTC date converted by the common parameter X-TC-Timestamp ; service is the product name, which should match the domain name of the product called. The calculation result in this example is 20180525/cvm/tc3_request .

HashedCanonicalRequest	Hash value of the CanonicalRequest string concatenated in the steps above. The pseudocode for calculation is Lowercase(HexEncode(Hash.SHA256(CanonicalRequest))) The calculation result in this example is 2815843035062fffd6f2a44ea8a34818b0dc46f024b8b3786976a3ad
------------------------	---

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
2815843035062fffd6f2a44ea8a34818b0dc46f024b8b3786976a3adda7a
```

3. Calculating the Signature

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

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3*****"
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., Gu5t9xGARNpq86cd98joQYCN3*****.
Date	The Date field information in Credential, such as 2019-02-25 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>AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****</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>c492e8e41437e97a620b728c301bb8d17e7dc0c17eeabce80c20cd70fc3a78ff</code> .

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

```
TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=c492e8e41437e97a620b728c301bb8d17e7dc0c17eeabce80c20cd70fc3a78ff
```

The following example shows a finished authorization header:

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=c492e8e41437e97a620b728c301bb8d17e7dc0c17eeabce80c20cd70fc3a78ff
```

```
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=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****&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 = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*****";
    private final static String CT_JSON = "application/json; charset=utf-8";

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

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

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

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

```

```

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

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

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

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

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

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

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

```



```

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

```

Python

```

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

# Key Parameters
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3*****"

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.utcnow().fromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 1, "Filters": [{"Name": "instance-name", "Values": ["unnamed"]}]}

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

```

```

canonical_querystring + "\n" +
canonical_headers + "\n" +
signed_headers + "\n" +
hashed_request_payload)
print(canonical_request)

# ***** Step 2: Concatenate the string to sign *****
credential_scope = date + "/" + service + "/" + "tc3_request"
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).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
+ ' -H "Authorization: ' + authorization + '" '
+ ' -H "Content-Type: application/json; charset=utf-8" '
+ ' -H "Host: ' + host + '" '
+ ' -H "X-TC-Action: ' + action + '" '
+ ' -H "X-TC-Timestamp: ' + str(timestamp) + '" '
+ ' -H "X-TC-Version: ' + version + '" '
+ ' -H "X-TC-Region: ' + region + '" '
+ " -d '" + payload + "'")
    
```

Golang

```
package main

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

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

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

func main() {
    secretId := "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****"
    secretKey := "Gu5t9xGARNpq86cd98joQYCN3*****"
    host := "cvm.tencentcloudapi.com"
    algorithm := "TC3-HMAC-SHA256"
    service := "cvm"
    version := "2017-03-12"
    action := "DescribeInstances"
    region := "ap-guangzhou"
    //var timestamp int64 = time.Now().Unix()
    var timestamp int64 = 1551113065

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

```
canonicalQueryString,  
canonicalHeaders,  
signedHeaders,  
hashedRequestPayload)  
fmt.Println(canonicalRequest)  
  
// step 2: build string to sign  
date := time.Unix(timestamp, 0).UTC().Format("2006-01-02")  
credentialScope := fmt.Sprintf("%s/%s/tc3_request", date, service)  
hashedCanonicalRequest := sha256hex(canonicalRequest)  
string2sign := fmt.Sprintf("%s\n%d\n%s\n%s",  
algorithm,  
timestamp,  
credentialScope,  
hashedCanonicalRequest)  
fmt.Println(string2sign)  
  
// step 3: sign string  
secretDate := hmacsha256(date, "TC3"+secretKey)  
secretService := hmacsha256(service, secretDate)  
secretSigning := hmacsha256("tc3_request", secretService)  
signature := hex.EncodeToString([]byte(hmacsha256(string2sign, secretSigning)))  
fmt.Println(signature)  
  
// step 4: build authorization  
authorization := fmt.Sprintf("%s Credential=%s/%s, SignedHeaders=%s, Signature=%  
s",  
algorithm,  
secretId,  
credentialScope,  
signedHeaders,  
signature)  
fmt.Println(authorization)  
  
curl := fmt.Sprintf(`curl -X POST https://%s\  
-H "Authorization: %s"\  
-H "Content-Type: application/json; charset=utf-8"\  
-H "Host: %s" -H "X-TC-Action: %s"\  
-H "X-TC-Timestamp: %d"\  
-H "X-TC-Version: %s"\  
-H "X-TC-Region: %s"\  
-d '%s'`, host, authorization, host, action, timestamp, version, region, payload)  
fmt.Println(curl)  
}
```

PHP

```
<?php
$secretId = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****";
$secretKey = "Gu5t9xGARNpq86cd98joQYCN3*****";
$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 = 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****'
secret_key = 'Gu5t9xGARNpq86cd98joQYCN3*****'

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

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

```

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

```

```

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

DotNet

```

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

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

    public static byte[] HmacSHA256(byte[] key, byte[] msg)
    {
        using (HMACSHA256 mac = new HMACSHA256(key))
        {
            return mac.ComputeHash(msg);
        }
    }

    public static Dictionary<String, String> BuildHeaders(string secretid,
        string secretkey, string service, string endpoint, string region,
        string action, string version, DateTime date, string requestPayload)
    {
        string datestr = date.ToString("yyyy-MM-dd");
        DateTime startTime = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc);
        long requestTimestamp = (long)Math.Round((date - startTime).TotalMilliseconds, Mi
    
```



```

dpointRounding.AwayFromZero) / 1000;
// ***** Step 1: Concatenate the CanonicalRequest string *****
string algorithm = "TC3-HMAC-SHA256";
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string contentType = "application/json";
string canonicalHeaders = "content-type:" + contentType + "; charset=utf-8\n" +
"host:" + endpoint + "\n";
string signedHeaders = "content-type;host";
string hashedRequestPayload = SHA256Hex(requestPayload);
string canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload;
Console.WriteLine(canonicalRequest);
Console.WriteLine("-----");

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

// ***** Step 3: Calculate the signature *****
byte[] tc3SecretKey = Encoding.UTF8.GetBytes("TC3" + secretkey);
byte[] secretDate = HmacSHA256(tc3SecretKey, Encoding.UTF8.GetBytes(datestr));
byte[] secretService = HmacSHA256(secretDate, Encoding.UTF8.GetBytes(service));
byte[] secretSigning = HmacSHA256(secretService, Encoding.UTF8.GetBytes("tc3_requ
est"));
byte[] signatureBytes = HmacSHA256(secretSigning, Encoding.UTF8.GetBytes(stringTo
Sign));
string signature = BitConverter.ToString(signatureBytes).Replace("-", "").ToLower
();
Console.WriteLine(signature);
Console.WriteLine("-----");

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

```
Console.WriteLine("-----");

Dictionary<string, string> headers = new Dictionary<string, string>();
headers.Add("Authorization", authorization);
headers.Add("Host", endpoint);
headers.Add("Content-Type", contentType + "; charset=utf-8");
headers.Add("X-TC-Timestamp", requestTimestamp.ToString());
headers.Add("X-TC-Version", version);
headers.Add("X-TC-Action", action);
headers.Add("X-TC-Region", region);
return headers;
}

public static void Main(string[] args)
{
    // SecretID and SecretKey
    string SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****";
    string SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*****";

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

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

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

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

NodeJS

```

const crypto = require('crypto');

function sha256(message, secret = '', encoding) {
    const hmac = crypto.createHmac('sha256', secret)
    return hmac.update(message).digest(encoding)
}

function getHash(message, encoding = 'hex') {
    const hash = crypto.createHash('sha256')
    return hash.update(message).digest(encoding)
}

function getDate(timestamp) {
    const date = new Date(timestamp * 1000)
    const year = date.getUTCFullYear()
    const month = ('0' + (date.getUTCMonth() + 1)).slice(-2)
    const day = ('0' + date.getUTCDate()).slice(-2)
    return `${year}-${month}-${day}`
}

function main(){

const SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****"
const SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*****"

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

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

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

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

const canonicalRequest = httpRequestMethod + "\n"
    
```

```

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

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

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

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

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

C++

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

using namespace std;

string get_data(int64_t &timestamp)
{
    string utcDate;
    char buff[20] = {0};
    // time_t timenow;
    struct tm sttime;
    sttime = *gmtime(&timestamp);
    strftime(buff, sizeof(buff), "%Y-%m-%d", &sttime);
    utcDate = string(buff);
    return utcDate;
}

string int2str(int64_t n)
{
    std::stringstream ss;
    ss << n;
    return ss.str();
}

string sha256Hex(const string &str)
{
    char buf[3];
    unsigned char hash[SHA256_DIGEST_LENGTH];
    SHA256_CTX sha256;
    SHA256_Init(&sha256);
    SHA256_Update(&sha256, str.c_str(), str.size());
    SHA256_Final(hash, &sha256);
    std::string NewString = "";
    for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
    {
        sprintf(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('0');
    for (int i = 0; i < len; i++)
    {
        ss << hash[i];
    }

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

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

```

return output;
}

int main()
{
string SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****";
string SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*****";

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

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

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

// ***** Step 3: Calculate the signature *****
string kKey = "TC3" + SECRET_KEY;
string kDate = HmacSha256(kKey, date);
string kService = HmacSha256(kDate, service);
string kSigning = HmacSha256(kService, "tc3_request");
    
```

```

string signature = HexEncode(HmacSha256(kSigning, stringToSign));
cout << signature << endl;
cout << "-----" << endl;

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

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

Signature Failure

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

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

Signature

最近更新时间：2022-06-01 14:28:22

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: AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3*****

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	AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
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' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****',
  '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=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****&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=AKIDz8krbsJ5yKBZQpn74WF
kmLPx3*****&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 = 'Gu5t9xGARNpq86cd98joQYCN3*****';

```

The final signature is:

```
zmmjn35mikh6pM3V7sUEuX4wyYM=
```

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 `zmmjn35mikh6pM3V7sUEuX4wyYM=`, the final signature string request parameter (Signature) is `zmmjn35mikh6pM3V7sUEuX4wyYM%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=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****&Signature=zmmjn35mikh6pM3V7sUEuX4wyYM%3D&Timestamp=1465185768&Version=2017-03-12` .

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

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

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

Java

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Random;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;
public class TencentCloudAPIDemo {
    private final static String CHARSET = "UTF-8";
    public static String sign(String s, String key, String method) throws Exception {
        Mac mac = Mac.getInstance(method);
        SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
        mac.init(secretKeySpec);
        byte[] hash = mac.doFinal(s.getBytes(CHARSET));
        return DatatypeConverter.printBase64Binary(hash);
    }
    public static String getStringToSign(TreeMap<String, Object> params) {
        StringBuilder s2s = new StringBuilder("GETcvm.tencentcloudapi.com/?");
        // When signing, the parameters need to be sorted in lexicographical order. TreeMap is used here to guarantee the correct order.
        for (String k : params.keySet()) {
            s2s.append(k).append("=").append(params.get(k).toString()).append("&");
        }
        return s2s.toString().substring(0, s2s.length() - 1);
    }
    public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException {
        StringBuilder url = new StringBuilder("https://cvm.tencentcloudapi.com/?");
        // There is no requirement for the order of the parameters in the actual request URL.
        for (String k : params.keySet()) {
            // The request string needs to be URL encoded. As the Key is all in English letters, only the value is URL encoded here.
            url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).append("&");
        }
    }
}
```

```
}
return url.toString().substring(0, url.length() - 1);
}
public static void main(String[] args) throws Exception {
    TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap 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", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****"); // 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), "Gu5t9xGARNpq86cd98joQYCN3*
    *****", "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 = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3*****"
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 := "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****"
    secretKey := "Gu5t9xGARNpq86cd98joQYCN3*****"
    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 = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****";
$secretKey = "Gu5t9xGARNpq86cd98joQYCN3*****";
$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
    
```

```
// $param["Signature"] = $signature;
// $url = "https://cvm.tencentcloudapi.com/?".http_build_query($param);
// echo $url.PHP_EOL;
// $ch = curl_init();
// curl_setopt($ch, CURLOPT_URL, $url);
// $output = curl_exec($ch);
// curl_close($ch);
// echo json_decode($output);
```

Ruby

```
# -*- coding: UTF-8 -*-
# require ruby>=2.3.0
require 'time'
require 'openssl'
require 'base64'
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3*****"
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 = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****";
string SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*****";
string endpoint = "cvm.tencentcloudapi.com";
string region = "ap-guangzhou";
string action = "DescribeInstances";
string version = "2017-03-12";
double RequestTimestamp = 1465185768;
```

```

// long timestamp = ToTimestamp() / 1000;
// string requestTimestamp = timestamp.ToString();
Dictionary<string, string> param = new Dictionary<string, string>();
param.Add("Limit", "20");
param.Add("Offset", "0");
param.Add("InstanceIds.0", "ins-09dx96dg");
param.Add("Action", action);
param.Add("Nonce", "11886");
// param.Add("Nonce", Math.Abs(new Random().Next()).ToString());
param.Add("Timestamp", RequestTimestamp.ToString());
param.Add("Version", version);
param.Add("SecretId", SECRET_ID);
param.Add("Region", region);
SortedDictionary<string, string> headers = new SortedDictionary<string, string>(p
aram, StringComparer.Ordinal);
string sigInParam = MakeSignPlainText(headers, "GET", endpoint, "/");
Console.WriteLine(sigInParam);
string sigOutParam = Sign(SECRET_KEY, sigInParam);
Console.WriteLine("GET https://cvm.tencentcloudapi.com");
foreach (KeyValuePair<string, string> kv in headers)
{
    Console.WriteLine(kv.Key + ": " + kv.Value);
}
Console.WriteLine("Signature" + ": " + WebUtility.UrlEncode(sigOutParam));
Console.WriteLine();
string result = "https://cvm.tencentcloudapi.com/?";
foreach (KeyValuePair<string, string> kv in headers)
{
    result += WebUtility.UrlEncode(kv.Key) + "=" + WebUtility.UrlEncode(kv.Value) +
"&";
}
result += WebUtility.UrlEncode("Signature") + "=" + WebUtility.UrlEncode(sigOutPa
ram);
Console.WriteLine("GET " + result);
}
}
    
```

NodeJS

```

const crypto = require('crypto');
function get_req_url(params, endpoint){
    params['Signature'] = escape(params['Signature']);
    const url_strParam = sort_params(params)
    return "https://" + endpoint + "/" + url_strParam.slice(1);
}
function formatSignString(reqMethod, endpoint, path, strParam){
    
```

```

let strSign = reqMethod + endpoint + path + "?" + strParam.slice(1);
return strSign;
}
function sha1(secretKey, strsign){
let signMethodMap = {'HmacSHA1': "sha1"};
let hmac = crypto.createHmac(signMethodMap['HmacSHA1'], secretKey || "");
return hmac.update(Buffer.from(strsign, 'utf8')).digest('base64')
}
function sort_params(params){
let strParam = "";
let keys = Object.keys(params);
keys.sort();
for (let k in keys) {
//k = k.replace(/_/g, '.');
strParam += ("&" + keys[k] + "=" + params[keys[k]]);
}
return strParam
}
function main(){
const SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****"
const SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*****"
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

最近更新时间：2022-05-23 15:21:52

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.
<code>MissingParameter</code>	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.

Remote Command APIs

EnableInvoker

最近更新时间：2023-03-14 16:55:45

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to enable an invoker.

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: EnableInvoker.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
InvokerId	Yes	String	ID of the invoker to be enabled.

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for

locating a problem.

4. Example

Example1 Enabling an invoker

Input Example

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

{
  "InvokerId": "ivk-b7s3qa51"
}
```

Output Example

```
{
  "Response": {
    "RequestId": "64119f65-18f5-47f3-a8d2-60d9fee593bf"
  }
}
```

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 NodeJS](#)
- [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
AuthFailure	CAM signature/authentication error.
InvalidParameterValue.InvalidInvokerId	Invalid InvokerId.
ResourceNotFound	The resource does not exist.

CreateInvoker

最近更新时间：2023-03-14 16:55:47

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to create an invoker.

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: CreateInvoker.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
Name	Yes	String	Invoker name.
Type	Yes	String	Invoker type. It can only be <code>SCHEDULE</code> (recurring invokers).
CommandId	Yes	String	Remote command ID.
InstanceIds.N	Yes	Array of String	ID of the instance bound to the trigger. Up to 100 IDs are allowed.

Username	No	String	The user who executes the command.
Parameters	No	String	Custom parameters of the command.
ScheduleSettings	No	ScheduleSettings	Settings required for a recurring invoker.

3. Output Parameters

Parameter Name	Type	Description
InvokerId	String	Invoker ID.
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Creating a run-once invoker

Execute command cmd-m7uma92n on instance ins-yx05ky8j at 2021-09-01 00:00:00 UTC+8

Input Example

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

{
  "Name": "test-invoker",
  "CommandId": "cmd-m7uma92n",
  "InstanceIds": [
    "ins-yx05ky8j"
  ],
  "Type": "SCHEDULE",
  "ScheduleSettings": {
    "Policy": "ONCE",
    "InvokeTime": "2021-09-01T00:00:00+08:00"
  }
}
```

Output Example

```
{
  "Response": {
    "RequestId": "97268137-e0f7-477d-833b-766273d0ea47",
    "InvokerId": "ivk-1e1g3x2h"
  }
}
```

Example2 Creating an invoker that executes a command repeatedly

Execute command cmd-m7uma92n on instance ins-yx05ky8j at 00:00:00 UTC+8 on the first day of every month

Input Example

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

```
{
  "Name": "cron-invoker",
  "CommandId": "cmd-m7uma92n",
  "InstanceIds": [
    "ins-yx05ky8j"
  ],
  "Type": "SCHEDULE",
  "ScheduleSettings": {
    "Policy": "RECURRENCE",
    "Recurrence": "0 0 1 * *"
  }
}
```

Output Example

```
{
  "Response": {
    "RequestId": "d1d7ce29-b6ac-436d-93e0-b454f096cc50",
    "InvokerId": "ivk-n0t6rxtv"
  }
}
```

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 NodeJS](#)
- [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
AuthFailure	CAM signature/authentication error.
InternalError	Internal error.
InvalidParameter	Invalid parameter.
InvalidParameterValue.InvalidCommandId	Invalid CommandId.
InvalidParameterValue.InvalidCronExpression	Invalid crontab expression.
InvalidParameterValue.InvalidInstanceId	Invalid instance ID.
InvalidParameterValue.InvalidTimeFormat	Invalid time format.
InvalidParameterValue.InvokeTimeExpired	API invocation expired.
InvalidParameterValue.ParameterInvalidJsonFormat	The parameter is not a valid JSON string.

ResourceNotFound.CommandNotFound	The command does not exist.
ResourceNotFound.InstanceNotFound	The instance does not exist.
ResourceUnavailable.AgentNotInstalled	TAT Agent is not installed.

DescribeCommands

最近更新时间：2023-03-14 16:55:46

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to query command 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: DescribeCommands.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
CommandIds.N	No	Array of String	List of command IDs. Up to 100 IDs are allowed for each request. <code>CommandIds</code> and <code>Filters</code> cannot be specified at the same time.
Filters.N	No	Array of Filter	Filter conditions. <ul style="list-style-type: none"> <code>command-id</code> - String - Required: No - (Filter condition) Filter by the command ID. <code>command-name</code> - String - Required: No - (Filter condition) Filter by the command name.

			<ul style="list-style-type: none"> <code>command-type</code> - String - Required: No - (Filter condition) Filter by the command type. Valid values: <code>SHELL</code> or <code>POWERSHELL</code>. <code>created-by</code> - String - Required: No - (Filter condition) Filter by the creator. Valid values: <code>TAT</code> (public commands) or <code>USER</code> (custom commands). <code>tag-key</code> - String - Required: No - (Filter condition) Filter by the tag key. <code>tag-value</code> - String - Required: No - (Filter condition) Filter by the tag value. <code>tag:tag-key</code> - String - Required: No - (Filter) Filter by the tag key-value pair. The tag-key should be replaced with a specified tag key. For detailed usage, see sample 4. <p>Up to 10 <code>Filters</code> are allowed in one request. Each filter can have up to 5 <code>Filter.Values</code>. <code>CommandIds</code> and <code>Filters</code> cannot be specified at the same time.</p>
Limit	No	Integer	Number of returned results. It defaults to <code>20</code> . The maximum is 100. For more information on <code>Limit</code> , see the relevant section in the API Overview .
Offset	No	Integer	Offset. The default value is <code>0</code> . For more information on <code>Offset</code> , see the relevant section in API Introduction .

3. Output Parameters

Parameter Name	Type	Description
TotalCount	Integer	Total number of matching commands.
CommandSet	Array of Command	List of command details.
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Querying a command with the specified `CommandId`

Input Example

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

{
  "CommandIds": [
    "cmd-dvstpcyy"
  ],
  "Offset": 0,
  "Limit": 20
}
```

Output Example

```
{
  "Response": {
    "RequestId": "eb973a12-71e3-4c0c-b1d8-4b863e5f5daf",
    "TotalCount": 1,
    "CommandSet": [
      {
        "CommandId": "cmd-dvstpcyy",
        "CommandName": "run-command",
        "Description": "whoami",
        "FormattedDescription": "",
        "CreatedBy": "USER",
        "Content": "d2hvYW1p",
        "CommandType": "SHELL",
        "WorkingDirectory": "/root/",
        "Timeout": 60,
        "EnableParameter": false,
        "DefaultParameters": "",
        "Username": "root",
        "Tags": [
          {
            "Value": "test-key",
            "Key": "test-value"
          }
        ],
        "CreatedTime": "2020-11-02T02:48:11+00:00",
        "UpdatedTime": "2020-11-02T02:48:11+00:00"
      }
    ]
  }
}
```

```

}
]
}
}
    
```

Example2 Querying commands with the `command-name` filter

Input Example

```

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

```

{
  "Offset": 0,
  "Limit": 20,
  "Filters": [
    {
      "Name": "command-name",
      "Values": [
        "second-command",
        "first-command"
      ]
    }
  ]
}
    
```

Output Example

```

{
  "Response": {
    "RequestId": "6b215093-e1f6-4803-b84a-a230849e88d1",
    "TotalCount": 2,
    "CommandSet": [
      {
        "CommandId": "cmd-hb2q34lk",
        "CommandName": "second-command",
        "Description": "ps",
        "FormattedDescription": "",
        "CreatedBy": "USER",
        "Content": "cHM=",
        "CommandType": "SHELL",
        "WorkingDirectory": "/root/"
      }
    ]
  }
}
    
```

```
"Timeout": 60,
"EnableParameter": false,
"DefaultParameters": "",
"Username": "root",
"Tags": [
  {
    "Value": "test-key",
    "Key": "test-value"
  }
],
"CreatedTime": "2020-10-30T07:19:52+00:00",
"UpdatedTime": "2020-10-30T07:19:52+00:00"
},
{
  "CommandId": "cmd-63usjhmq",
  "CommandName": "first-command",
  "Description": "hello world!",
  "FormattedDescription": "",
  "CreatedBy": "USER",
  "Content": "cHM=",
  "CommandType": "SHELL",
  "WorkingDirectory": "/",
  "Timeout": 600,
  "EnableParameter": false,
  "DefaultParameters": "",
  "Username": "root",
  "Tags": [
    {
      "Value": "test-key",
      "Key": "test-value"
    }
  ],
  "CreatedTime": "2020-10-26T11:26:07+00:00",
  "UpdatedTime": "2020-11-09T08:12:45+00:00"
}
]
}
```

Example3 Querying commands with the `created-by` filter

Input Example

```
POST / HTTP/1.1
Host: tat.tencentcloudapi.com
Content-Type: application/json
```

X-TC-Action: DescribeCommands

<Common request parameters>

```
{
  "Offset": 0,
  "Limit": 1,
  "Filters": [
    {
      "Name": "created-by",
      "Values": [
        "USER"
      ]
    }
  ]
}
```

Output Example

```
{
  "Response": {
    "RequestId": "6b215093-e1f6-4803-b84a-a230849e88d1",
    "TotalCount": 2,
    "CommandSet": [
      {
        "CommandId": "cmd-hb2q341k",
        "CommandName": "second-command",
        "Description": "ps",
        "FormattedDescription": "",
        "CreatedBy": "USER",
        "Content": "cHM=",
        "CommandType": "SHELL",
        "WorkingDirectory": "/root/",
        "Timeout": 60,
        "EnableParameter": false,
        "DefaultParameters": "",
        "Username": "root",
        "Tags": [],
        "CreatedTime": "2020-10-30T07:19:52+00:00",
        "UpdatedTime": "2020-10-30T07:19:52+00:00"
      }
    ]
  }
}
```

Example4 Query commands with the tag:tag-key filter

Input Example

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

{
  "Filters": [
    {
      "Values": [
        "test-value"
      ],
      "Name": "tag:test-key"
    }
  ]
}
```

Output Example

```
{
  "Response": {
    "RequestId": "33d3d954-f73a-4a7f-869b-8792bc7a6f13",
    "TotalCount": 1,
    "CommandSet": [
      {
        "CommandId": "cmd-38ps9q4p",
        "CommandName": "tag-test-1",
        "Description": "",
        "FormattedDescription": "",
        "CreatedBy": "USER",
        "Content": "CHMK",
        "CommandType": "SHELL",
        "WorkingDirectory": "/root",
        "Timeout": 60,
        "EnableParameter": false,
        "DefaultParameters": "",
        "Username": "root",
        "Tags": [
          {
            "Key": "test-key",
            "Value": "test-value"
          }
        ],
        "CreatedTime": "2021-05-12T02:49:04Z",
```



```
"UpdateTime": "2021-05-12T02:49:04Z"  
}  
]  
}  
}
```

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 NodeJS](#)
- [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
InternalError	Internal error.
InvalidParameter	Invalid parameter.
InvalidParameter.ConflictParameter	Conflicting parameters.
InvalidParameterValue	Invalid parameter value.
InvalidParameterValue.InvalidCommandId	Invalid CommandId.

InvalidParameterValue.InvalidFilter	Invalid filter.
UnauthorizedOperation.CamAuthFailed	CAM authentication failed.
UnauthorizedOperation.InvalidToken	Invalid token.
UnauthorizedOperation.MFAExpired	The Multi-factor authentication (MFA) code expired.
UnauthorizedOperation.MFANotFound	Multi-factor authentication (MFA) does not exist.

DescribeAutomationAgentStatus

最近更新时间：2023-03-14 16:55:47

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to query the status of the TAT agent.

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: DescribeAutomationAgentStatus.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
InstanceIds.N	No	Array of String	List of instance IDs for the query.
Filters.N	No	Array of Filter	Filter conditions. <ul style="list-style-type: none"> <code>agent-status</code> - String - Required: No - (Filter condition) Filter by agent status. Valid values: <code>Online</code>, <code>Offline</code>. <code>environment</code> - String - Required: No - (Filter condition) Filter by the agent environment. Valid value: <code>Linux</code>.

			<ul style="list-style-type: none"> <code>instance-id</code> - String - Required: No - (Filter condition) Filter by the instance ID. Up to 10 <code>Filters</code> allowed in one request. For each filter, five <code>Filter.Values</code> can be specified. <code>InstanceIds</code> and <code>Filters</code> cannot be specified at the same time.
Limit	No	Integer	Number of returned results. It defaults to <code>20</code> . The maximum is 100. For more information on <code>Limit</code> , see the relevant section in the API Overview .
Offset	No	Integer	Offset. The default value is <code>0</code> . For more information on <code>Offset</code> , see the relevant section in API Introduction .

3. Output Parameters

Parameter Name	Type	Description
AutomationAgentSet	Array of AutomationAgentInfo	Agent information list.
TotalCount	Integer	Total number of matching agents.
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Querying the agent status

Input Example

```

POST / HTTP/1.1
Host: tat.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeAutomationAgentStatus
<Common request parameters>

{
  "InstanceIds": [
    "lhins-ar5wyn4x"
  ]
}
    
```

Output Example

```
{
  "Response": {
    "RequestId": "520e7b93-904d-44de-a2dd-8e4c545be4ce",
    "AutomationAgentSet": [
      {
        "InstanceId": "lhins-ar5wyn4x",
        "Version": "0.1.0",
        "LastHeartbeatTime": "2020-11-16T12:05:44+00:00",
        "AgentStatus": "Online",
        "Environment": "Linux"
      }
    ],
    "TotalCount": 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 NodeJS](#)
- [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	Invalid parameter.
InvalidParameter.ConflictParameter	Conflicting parameters.
InvalidParameterValue	Invalid parameter value.
InvalidParameterValue.InvalidFilter	Invalid filter.
InvalidParameterValue.InvalidInstanceId	Invalid instance ID.
InvalidParameterValue.LimitExceeded	Parameter limit exceeded.
UnauthorizedOperation.CamAuthFailed	CAM authentication failed.
UnauthorizedOperation.InvalidToken	Invalid token.
UnauthorizedOperation.MFAExpired	The Multi-factor authentication (MFA) code expired.
UnauthorizedOperation.MFANotFound	Multi-factor authentication (MFA) does not exist.

DeleteCommand

最近更新时间：2023-03-14 16:55:47

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to delete a command.

Commands bound to an invoker cannot be deleted.

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: DeleteCommand.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
CommandId	Yes	String	ID of the command to be deleted.

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Deleting a command

Input Example

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

{
  "CommandId": "cmd-7efujjs6"
}
```

Output Example

```
{
  "Response": {
    "RequestId": "7f79b764-bc0f-471b-89c1-ca1b93cf7e8d"
  }
}
```

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 NodeJS](#)
- [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
InternalError	Internal error.
InvalidParameterValue	Invalid parameter value.
InvalidParameterValue.InvalidCommandId	Invalid CommandId.
ResourceNotFound.CommandNotFound	The command does not exist.
ResourceUnavailable.CommandInExecuting	The command is being executed.
ResourceUnavailable.CommandInInvoker	The command has been bound to an invoker.
UnauthorizedOperation.CamAuthFailed	CAM authentication failed.
UnauthorizedOperation.InvalidToken	Invalid token.
UnauthorizedOperation.MFAExpired	The Multi-factor authentication (MFA) code expired.
UnauthorizedOperation.MFANotFound	Multi-factor authentication (MFA) does not exist.

CreateCommand

最近更新时间：2023-03-14 16:55:47

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to create a command.

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: CreateCommand.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
CommandName	Yes	String	Command name. The name can be up to 60 bytes, and contain [a-z], [A-Z], [0-9] and [_-].
Content	Yes	String	Base64-encoded command. The maximum length is 64 KB.
Description	No	String	Command description. The maximum length is 120 characters.
CommandType	No	String	Command type. <code>SHELL</code> and <code>POWERSHELL</code> are supported. The default value is <code>SHELL</code> .

WorkingDirectory	No	String	Command execution path. The default value is /root for <code>SHELL</code> commands and C:\Program Files\qcloud\tat_agent\workdir for <code>POWERSHELL</code> commands.
Timeout	No	Integer	Command timeout period. Default value: 60 seconds. Value range: [1, 86400].
EnableParameter	No	Boolean	Whether to enable the custom parameter feature. This cannot be modified once created. Default value: <code>false</code> .
DefaultParameters	No	String	The default value of the custom parameter value when it is enabled. The field type is JSON encoded string. For example, {"varA": "222"}. <code>key</code> is the name of the custom parameter and <code>value</code> is the default value. Both <code>key</code> and <code>value</code> are strings. If no parameter value is provided in the <code>InvokeCommand</code> API, the default value is used. Up to 20 custom parameters are supported. The name of the custom parameter cannot exceed 64 characters and can contain [a-z], [A-Z], [0-9] and [-_].
Tags.N	No	Array of Tag	Tags bound to the command. At most 10 tags are allowed.
Username	No	String	The username used to execute the command on the CVM or Lighthouse instance. The principle of least privilege is the best practice for permission management. We recommend you execute TAT commands as a general user. By default, the root user is used to execute commands on Linux and the System user is used on Windows.
OutputCOSBucketUrl	No	String	The COS bucket URL for uploading logs. The URL must start with <code>https</code> , such as <code>https://BucketName-123454321.cos.ap-beijing.myqcloud.com</code> .
OutputCOSKeyPrefix	No	String	The COS bucket directory where the logs are saved. Check below for the rules of the directory name. 1. It must be a combination of number, letters, and visible characters. Up to 60 characters are allowed. 2. Use a slash (/) to create a subdirectory. 3. Consecutive dots (.) and slashes (/) are not allowed. It can not start with a slash (/).

3. Output Parameters

Parameter Name	Type	Description
CommandId	String	Command ID.
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Creating a command

Input Example

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

{
  "CommandName": "hello-command",
  "Description": "hello world",
  "Content": "bHM=",
  "CommandType": "SHELL",
  "WorkingDirectory": "/",
  "Timeout": 60
}
```

Output Example

```
{
  "Response": {
    "RequestId": "9bea970a-0bab-495f-b0dd-de5eedfdf611",
    "CommandId": "cmd-7efujjs6"
  }
}
```

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 NodeJS](#)
- [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	Invalid parameter.
InvalidParameterValue	Invalid parameter value.
InvalidParameterValue.CommandContentInvalid	Invalid command content.
InvalidParameterValue.CommandNameDuplicated	Duplicate command name.
InvalidParameterValue.InvalidCommandName	Invalid command name.
InvalidParameterValue.InvalidContent	Invalid command.
InvalidParameterValue.InvalidOutputCOSBucketUrl	Invalid OutputCOSBucketUrl.
InvalidParameterValue.InvalidOutputCOSKeyPrefix	Invalid OutputCOSKeyPrefix.

InvalidParameterValue.InvalidUsername	Invalid username.
InvalidParameterValue.InvalidWorkingDirectory	Invalid command execution path.
InvalidParameterValue.ParameterDisabled	The custom parameter feature is not enabled.
InvalidParameterValue.ParameterInvalidJsonFormat	The parameter is not a valid JSON string.
InvalidParameterValue.ParameterKeyContainsInvalidChar	The parameter key contains invalid characters.
InvalidParameterValue.ParameterKeyDuplicated	Duplicate parameter keys.
InvalidParameterValue.ParameterKeyLenExceeded	The parameter key is too long.
InvalidParameterValue.ParameterNumberExceeded	Too many parameters.
InvalidParameterValue.ParameterValueNotString	The parameter value is not a string.
InvalidParameterValue.Range	The parameter value is not in the valid range.
InvalidParameterValue.SupportParametersOnlyIfEnableParameter	The custom parameter feature is not enabled.
InvalidParameterValue.TooLong	Length limit exceeded.
MissingParameter	Missing parameter.
UnauthorizedOperation.CamAuthFailed	CAM authentication failed.
UnauthorizedOperation.InvalidToken	Invalid token.
UnauthorizedOperation.MFAExpired	The Multi-factor authentication (MFA) code expired.
UnauthorizedOperation.MFANotFound	Multi-factor authentication (MFA) does not exist.

DescribeInvocationTasks

最近更新时间：2023-03-14 16:55:46

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to query execution task 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: DescribeInvocationTasks.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
InvocationTaskIds.N	No	Array of String	List of execution task IDs. Up to 100 IDs are allowed for each request. <code>InvocationTaskIds</code> and <code>Filters</code> cannot be specified at the same time.
Filters.N	No	Array of Filter	Filter conditions. <ul style="list-style-type: none"> <code>invocation-id</code> - String - Required: No - (Filter condition) Filter by the execution activity ID. <code>invocation-task-id</code> - String - Required: No - (Filter condition) Filter by the execution task ID. <code>instance-id</code> - String - Required: No - (Filter condition) Filter by the instance ID.

			<ul style="list-style-type: none"> <code>command-id</code> - String - Required: No - (Filter condition) Filter by the command ID. <p>Up to 10 <code>Filters</code> are allowed for each request. Each filter can have up to five <code>Filter.Values</code>.</p> <p><code>InvocationTaskIds</code> and <code>Filters</code> cannot be specified at the same time.</p>
Limit	No	Integer	Number of returned results. It defaults to <code>20</code> . The maximum is 100. For more information on <code>Limit</code> , see the relevant section in the API Overview .
Offset	No	Integer	Offset. The default value is <code>0</code> . For more information on <code>Offset</code> , see the relevant section in API Introduction .
HideOutput	No	Boolean	Whether to hide the output. Valid values: <ul style="list-style-type: none"> <code>True</code> (default): Hide the output <code>False</code>: Show the output

3. Output Parameters

Parameter Name	Type	Description
TotalCount	Integer	Total number of matching execution tasks.
InvocationTaskSet	Array of InvocationTask	List of execution tasks.
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Query the details of all execution tasks by execution activity ID

Input Example

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



```

{
  "Offset": 0,
  "Limit": 10,
  "HideOutput": false,
  "Filters": [
    {
      "Name": "invocation-id",
      "Values": [
        "inv-1v117hda"
      ]
    }
  ]
}

```

Output Example

```

{
  "Response": {
    "RequestId": "a1df9725-51c6-466d-a038-4c86461a8e26",
    "TotalCount": 3,
    "InvocationTaskSet": [
      {
        "CommandId": "cmd-13axrtuq",
        "CommandDocument": {
          "Content": "d2hvYW1p",
          "CommandType": "SHELL",
          "Timeout": 1,
          "Username": "root",
          "WorkingDirectory": "/root/"
        },
        "InvocationId": "inv-1v117hda",
        "InvocationTaskId": "invt-afwuqts2",
        "TaskStatus": "SUCCESS",
        "InstanceId": "ins-zj0f57ev",
        "TaskResult": {
          "ExitCode": 0,
          "Output": "cm9vdAo=",
          "Dropped": 0,
          "OutputUploadCOSErrorInfo": "",
          "OutputUrl": "",
          "ExecStartTime": "2020-11-05T07:49:58+00:00",
          "ExecEndTime": "2020-11-05T07:50:04+00:00"
        },
        "ErrorInfo": "",
        "InvocationSource": "xx",
        "StartTime": "2020-11-05T07:49:58+00:00",

```

```

"EndTime": "2020-11-05T07:50:04+00:00",
"CreatedTime": "2020-11-05T07:49:56+00:00",
"UpdatedTime": "2020-11-05T07:50:06+00:00"
},
{
"CommandId": "cmd-13axrtuq",
"CommandDocument": {
"Content": "d2hvYW1p",
"CommandType": "SHELL",
"Timeout": 1,
"Username": "root",
"WorkingDirectory": "/root/"
},
"InvocationId": "inv-1vll7hda",
"InvocationTaskId": "invt-08oe5fe2",
"TaskStatus": "SUCCESS",
"InstanceId": "ins-zj0f57ex",
"TaskResult": {
"ExitCode": 0,
"Output": "cm9vdAo=",
"Dropped": 0,
"OutputUploadCOSErrorInfo": "",
"OutputUrl": "",
"ExecStartTime": "2020-11-05T07:49:58+00:00",
"ExecEndTime": "2020-11-05T07:50:04+00:00"
},
"ErrorInfo": "",
"InvocationSource": "xx",
"StartTime": "2020-11-05T07:49:58+00:00",
"EndTime": "2020-11-05T07:50:04+00:00",
"CreatedTime": "2020-11-05T07:49:56+00:00",
"UpdatedTime": "2020-11-05T07:50:06+00:00"
},
{
"CommandId": "cmd-13axrtuq",
"CommandDocument": {
"Content": "d2hvYW1p",
"CommandType": "SHELL",
"Timeout": 1,
"Username": "root",
"WorkingDirectory": "/root/"
},
"InvocationId": "inv-1vll7hda",
"InvocationTaskId": "invt-91cpqs22",
"TaskStatus": "SUCCESS",
"InstanceId": "ins-zj0f57ew",
"TaskResult": {

```

```
"ExitCode": 0,
"Output": "cm9vdAo=",
"Dropped": 0,
"OutputUploadCOSErrorInfo": "",
"OutputUrl": "",
"ExecStartTime": "2020-11-05T07:49:58+00:00",
"ExecEndTime": "2020-11-05T07:50:04+00:00"
},
"ErrorInfo": "",
"InvocationSource": "xx",
"StartTime": "2020-11-05T07:49:58+00:00",
"EndTime": "2020-11-05T07:50:04+00:00",
"CreatedTime": "2020-11-05T07:49:56+00:00",
"UpdatedTime": "2020-11-05T07:50:06+00:00"
}
]
}
}
```

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 NodeJS](#)
- [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	Invalid parameter.
InvalidParameter.ConflictParameter	Conflicting parameters.
InvalidParameterValue	Invalid parameter value.
InvalidParameterValue.InvalidCommandId	Invalid CommandId.
InvalidParameterValue.InvalidFilter	Invalid filter.
InvalidParameterValue.InvalidInstanceId	Invalid instance ID.
InvalidParameterValue.InvalidInvocationId	Invalid execution activity ID.
InvalidParameterValue.InvalidInvocationTaskId	Invalid execution task ID.
UnauthorizedOperation.CamAuthFailed	CAM authentication failed.
UnauthorizedOperation.InvalidToken	Invalid token.
UnauthorizedOperation.MFAExpired	The Multi-factor authentication (MFA) code expired.
UnauthorizedOperation.MFANotFound	Multi-factor authentication (MFA) does not exist.

CancelInvocation

最近更新时间：2023-03-14 16:55:47

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to cancel the command executed on one or more CVM instances.

- If the command has not been delivered to the TAT agent, the task status is `PENDING` , `DELIVERING` , or `DELIVER_DELAYED` , and will be `CANCELLED` after the command is canceled.
- If the command has been delivered to the TAT agent, the task status is `RUNNING` , and will be `TERMINATED` after the command is canceled.

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: CancelInvocation.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
InvocationId	Yes	String	Execution activity ID
InstanceIds.N	No	Array of String	Instance ID list. A maximum of 100 IDs are allowed. Supported instance types: <ul style="list-style-type: none">• <code>CVM</code>

- LIGHTHOUSE

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Canceling the execution of a command

Input Example

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

{
  "InvocationId": "inv-ib21ld37",
  "InstanceIds": [
    "ins-zcewfho0"
  ]
}
```

Output Example

```
{
  "Response": {
    "RequestId": "b27421da-0ad4-4e11-b694-cda143606701"
  }
}
```

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 NodeJS](#)
- [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	Invalid parameter.
InvalidParameterValue	Invalid parameter value.
InvalidParameterValue.InstanceIsNotRelatedToInvocation	The instance with this ID has no such execution activity.
InvalidParameterValue.InvalidInvocationId	Invalid execution activity ID.
InvalidParameterValue.LimitExceeded	Parameter limit exceeded.
InvalidParameterValue.TooLong	Length limit exceeded.
LimitExceeded	Reached the quota limit.
MissingParameter	Missing parameter.
ResourceNotFound.InvocationNotFound	No execution activity found.

UnauthorizedOperation.CamAuthFailed	CAM authentication failed.
UnauthorizedOperation.InvalidToken	Invalid token.
UnauthorizedOperation.MFAExpired	The Multi-factor authentication (MFA) code expired.
UnauthorizedOperation.MFANotFound	Multi-factor authentication (MFA) does not exist.
UnknownParameter	Unknown parameter error.

DescribeInvocations

最近更新时间：2023-03-14 16:55:46

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to query execution activity 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: DescribeInvocations.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
InvocationIds.N	No	Array of String	List of execution activity IDs. Up to 100 IDs are allowed for each request. <code>InvocationIds</code> and <code>Filters</code> cannot be specified at the same time.
Filters.N	No	Array of Filter	Filter conditions. <ul style="list-style-type: none"> <code>invocation-id</code> - String - Required: No - (Filter condition) Filter by the execution activity ID. <code>command-id</code> - String - Required: No - (Filter condition) Filter by the command ID.

			<ul style="list-style-type: none"> <code>command-created-by</code> - String - Required: No - (Filter condition) Filter by the command type. Valid values: <code>TAT</code> (public commands) or <code>USER</code> (custom commands). <code>instance-kind</code> - String - Required: No - (Filter condition) Filter by the instance type. Valid values: <code>CVM</code> or <code>LIGHTHOUSE</code>. <p>Up to 10 <code>Filters</code> are allowed for each request. Each filter can have up to five <code>Filter.Values</code>. <code>InvocationIds</code> and <code>Filters</code> cannot be specified at the same time.</p>
Limit	No	Integer	Number of returned results. It defaults to <code>20</code> . The maximum is 100. For more information on <code>Limit</code> , see the relevant section in the API Overview .
Offset	No	Integer	Offset. The default value is <code>0</code> . For more information on <code>Offset</code> , see the relevant section in API Introduction .

3. Output Parameters

Parameter Name	Type	Description
TotalCount	Integer	Total number of matching execution activities.
InvocationSet	Array of Invocation	List of execution activities.
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Querying the execution activities

Input Example

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

```
{
  "Offset": 0,
  "Limit": 1,
  "InvocationIds": [
    "inv-q4zp60g0"
  ]
}
```

Output Example

```
{
  "Response": {
    "RequestId": "958a3603-d0c3-4c97-a37d-6bc24eacddac",
    "TotalCount": 1,
    "InvocationSet": [
      {
        "CommandId": "cmd-9dxzty3m",
        "CommandContent": "cHdk",
        "CommandType": "SHELL",
        "Timeout": 60,
        "InvocationSource": "USER",
        "WorkingDirectory": "/root",
        "InvocationId": "inv-q4zp60g0",
        "InstanceKind": "CVM",
        "InvocationStatus": "SUCCESS",
        "Description": "Test Invocation",
        "Parameters": "",
        "DefaultParameters": "",
        "Username": "root",
        "OutputCOSKeyPrefix": "cosprefix",
        "OutputCOSBucketUrl": "https://example-123456789.cos.ap-beijing.myqcloud.com",
        "InvocationTaskBasicInfoSet": [
          {
            "InvocationTaskId": "invt-kakne8h2",
            "TaskStatus": "SUCCESS",
            "InstanceId": "ins-zj0f57ew"
          },
          {
            "InvocationTaskId": "invt-jc2onrxm",
            "TaskStatus": "SUCCESS",
            "InstanceId": "ins-zj0f57ex"
          },
          {
            "InvocationTaskId": "invt-6xmuisyq",
            "TaskStatus": "SUCCESS",
            "InstanceId": "ins-zj0f57ev"
          }
        ]
      }
    ]
  }
}
```

```

    }
  ],
  "StartTime": "2020-11-09T07:21:59+00:00",
  "EndTime": "2020-11-09T07:22:08+00:00",
  "CreatedTime": "2020-11-09T07:21:59+00:00",
  "UpdatedTime": "2020-11-09T07:22:08+00:00"
}
]
}
}

```

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 NodeJS](#)
- [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	Invalid parameter.
InvalidParameter.ConflictParameter	Conflicting parameters.

InvalidParameterValue.InvalidCommandId	Invalid CommandId.
InvalidParameterValue.InvalidFilter	Invalid filter.
InvalidParameterValue.InvalidInvocationId	Invalid execution activity ID.
InvalidParameterValue.LimitExceeded	Parameter limit exceeded.
LimitExceeded.FilterValueExceeded	Too many <code>Filter</code> values.
UnauthorizedOperation.CamAuthFailed	CAM authentication failed.
UnauthorizedOperation.InvalidToken	Invalid token.
UnauthorizedOperation.MFAExpired	The Multi-factor authentication (MFA) code expired.
UnauthorizedOperation.MFANotFound	Multi-factor authentication (MFA) does not exist.

InvokeCommand

最近更新时间：2023-03-14 16:55:45

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to trigger a command on the specified instance and returns the execution activity ID (inv-xxxxxxx) on success. Each execution activity has one or more execution tasks (invt-xxxxxxx) and each execution task indicates an execution record on a CVM or Lighthouse instance.

- If the agent is not installed on the instance or is offline, an error is returned.
- If the command type is not supported by the agent runtime environment, an error is returned.
- The specified instance needs to be in a VPC network.
- The specified instance needs to be in the RUNNING status.
- Only one type of instances (CVM or Lighthouse) can be specified in a single request.

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: InvokeCommand.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
CommandId	Yes	String	ID of the command to be triggered.

InstanceIds.N	Yes	Array of String	IDs of instances about to execute commands. At most 100 IDs are allowed.
Parameters	No	String	<p>Custom parameters of the command. The field type is JSON encoded string. For example, {"varA": "222"}.</p> <p><code>key</code> is the name of the custom parameter and <code>value</code> is the default value. Both <code>key</code> and <code>value</code> are strings.</p> <p>If no parameter value is provided, the DefaultParameters of the command is used.</p> <p>Up to 20 custom parameters are supported.</p> <p>The name of the custom parameter cannot exceed 64 characters and can contain [a-z], [A-Z], [0-9] and [-_].</p>
Username	No	String	<p>The username used to execute the command on the CVM or Lighthouse instance.</p> <p>The principle of the least privilege is the best practice for permission management. We recommend you execute TAT commands as a general user. If this is not specified, the Username of the command is used by default.</p>
WorkingDirectory	No	String	Execution path of the command. The WorkingDirectory of the command is used by default.
Timeout	No	Integer	Command timeout period. Value range: [1, 86400]. The Timeout of the command is used by default.
OutputCOSBucketUrl	No	String	The COS bucket URL for uploading logs. The URL must start with <code>https</code> , such as <code>https://BucketName-123454321.cos.ap-beijing.myqcloud.com</code> .
OutputCOSKeyPrefix	No	String	<p>The COS bucket directory where the logs are saved. Check below for the rules of the directory name.</p> <ol style="list-style-type: none"> 1. It must be a combination of number, letters, and visible characters. Up to 60 characters are allowed. 2. Use a slash (/) to create a subdirectory. 3. "." can not be used as the folder name. It cannot start with a slash (/), and cannot contain consecutive slashes.

3. Output Parameters

Parameter Name	Type	Description

InvocationId	String	Execution activity ID.
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Triggering a command on a Lighthouse instance

Input Example

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

{
  "CommandId": "cmd-ffxdx79i",
  "InstanceIds": [
    "lhins-ar5wyn40"
  ]
}
```

Output Example

```
{
  "Response": {
    "RequestId": "41417f50-51b5-4c8d-85b7-f6c508cb228f",
    "InvocationId": "inv-8xgjrytm"
  }
}
```

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 NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
FailedOperation.CVMError	Failed to access the CVM.
FailedOperation.LighthouseError	Failed to access the Lighthouse instance.
InternalError	Internal error.
InvalidParameter	Invalid parameter.
InvalidParameter.InvalidUsername	Invalid username.
InvalidParameterValue	Invalid parameter value.
InvalidParameterValue.AgentUnsupportedCommandType	TAT Agent does not support this command type.
InvalidParameterValue.InconsistentInstance	Inconsistent instance type.
InvalidParameterValue.InvalidCommandId	Invalid CommandId.
InvalidParameterValue.InvalidInstanceId	Invalid instance ID.
InvalidParameterValue.InvalidOutputCOSBucketUrl	Invalid OutputCOSBucketUrl.
InvalidParameterValue.InvalidOutputCOSKeyPrefix	Invalid OutputCOSKeyPrefix.
InvalidParameterValue.InvalidWorkingDirectory	Invalid command execution path.

InvalidParameterValue.LackOfParameterInfo	The custom parameter feature is enabled, but custom parameters are missing.
InvalidParameterValue.LimitExceeded	Parameter limit exceeded.
InvalidParameterValue.ParameterDisabled	The custom parameter feature is not enabled.
InvalidParameterValue.ParameterInvalidJsonFormat	The parameter is not a valid JSON string.
InvalidParameterValue.ParameterKeyContainsInvalidChar	The parameter key contains invalid characters.
InvalidParameterValue.ParameterKeyDuplicated	Duplicate parameter keys.
InvalidParameterValue.ParameterKeyLenExceeded	The parameter key is too long.
InvalidParameterValue.ParameterNumberExceeded	Too many parameters.
InvalidParameterValue.ParameterValueNotString	The parameter value is not a string.
InvalidParameterValue.SupportParametersOnlyIfEnableParameter	The custom parameter feature is not enabled.
ResourceNotFound.CommandNotFound	The command does not exist.
ResourceNotFound.InstanceNotFound	The instance does not exist.
ResourceNotFound.RoleNotFound	The role does not exist.
ResourceUnavailable.AgentNotInstalled	TAT Agent is not installed.
ResourceUnavailable.AgentStatusNotOnline	TAT Agent is offline.
ResourceUnavailable.InstanceStateNotRunning	The instance is not running.
ResourceUnavailable.LighthouseUnsupportedRegion	Lighthouse is not supported in the region.
UnauthorizedOperation.CamAuthFailed	CAM authentication failed.
UnauthorizedOperation.InvalidToken	Invalid token.
UnauthorizedOperation.MFAExpired	The Multi-factor authentication (MFA) code expired.
UnauthorizedOperation.MFANotFound	Multi-factor authentication (MFA) does not exist.

ModifyCommand

最近更新时间：2023-03-14 16:55:45

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to modify a command.

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: ModifyCommand.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
CommandId	Yes	String	Command ID.
CommandName	No	String	Command name. The name can be up to 60 bytes, and contain [a-z], [A-Z], [0-9] and [_-].
Description	No	String	Command description. The maximum length is 120 characters.
Content	No	String	Base64-encoded command. The maximum length is 64 KB.
CommandType	No	String	Command type. <code>SHELL</code> and <code>POWERSHELL</code> are supported.

WorkingDirectory	No	String	Command execution path.
Timeout	No	Integer	Command timeout period. Value range: [1, 86400].
DefaultParameters	No	String	<p>The default value of the custom parameter value when it is enabled. The field type is JSON encoded string. For example, {"varA": "222"}.</p> <p>All parameters are overwritten. All default values are required for modification.</p> <p>Modification is only allowed when <code>EnableParameter</code> is <code>true</code>.</p> <p><code>key</code> is the name of the custom parameter and <code>value</code> is the default value. Both <code>key</code> and <code>value</code> are strings.</p> <p>Up to 20 custom parameters are supported.</p> <p>The name of the custom parameter cannot exceed 64 characters and can contain [a-z], [A-Z], [0-9] and [-_].</p>
Username	No	String	<p>The username used to execute the command on the CVM or Lighthouse instance.</p> <p>The principle of least privilege is the best practice for permission management. We recommend you execute TAT commands as a general user.</p>
OutputCOSBucketUrl	No	String	<p>The COS bucket URL for uploading logs. The URL must start with <code>https</code>, such as <code>https://BucketName-123454321.cos.ap-beijing.myqcloud.com</code>.</p>
OutputCOSKeyPrefix	No	String	<p>The COS bucket directory where the logs are saved. Check below for the rules of the directory name.</p> <ol style="list-style-type: none"> 1. It must be a combination of number, letters, and visible characters. Up to 60 characters are allowed. 2. Use a slash (/) to create a subdirectory. 3. "." can not be used as the folder name. It cannot start with a slash (/), and cannot contain consecutive slashes.

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Modifying a command

Input Example

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

{
  "CommandName": "second-command",
  "Description": "hello world!",
  "Content": "cHM=",
  "CommandType": "SHELL",
  "WorkingDirectory": "/",
  "Timeout": 600,
  "CommandId": "cmd-63usjhmq"
}
```

Output Example

```
{
  "Response": {
    "RequestId": "9bea970a-0bab-495f-b0dd-de5eedfdf611"
  }
}
```

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 NodeJS](#)

- [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
InternalError	Internal error.
InvalidParameter	Invalid parameter.
InvalidParameterValue	Invalid parameter value.
InvalidParameterValue.CommandContentInvalid	Invalid command content.
InvalidParameterValue.CommandNameDuplicated	Duplicate command name.
InvalidParameterValue.InvalidCommandId	Invalid CommandId.
InvalidParameterValue.InvalidCommandName	Invalid command name.
InvalidParameterValue.InvalidOutputCOSBucketUrl	Invalid OutputCOSBucketUrl.
InvalidParameterValue.InvalidOutputCOSKeyPrefix	Invalid OutputCOSKeyPrefix.
InvalidParameterValue.ParameterInvalidJsonFormat	The parameter is not a valid JSON string.
InvalidParameterValue.ParameterKeyContainsInvalidChar	The parameter key contains invalid characters.
InvalidParameterValue.ParameterKeyDuplicated	Duplicate parameter keys.
InvalidParameterValue.ParameterKeyLenExceeded	The parameter key is too long.
InvalidParameterValue.ParameterNumberExceeded	Too many parameters.
InvalidParameterValue.ParameterValueNotString	The parameter value is not a string.
InvalidParameterValue.Range	The parameter value is not in the valid

	range.
InvalidParameterValue.SupportParametersOnlyIfEnableParameter	The custom parameter feature is not enabled.
InvalidParameterValue.TooLong	Length limit exceeded.
ResourceNotFound.CommandNotFound	The command does not exist.
UnauthorizedOperation.CamAuthFailed	CAM authentication failed.
UnauthorizedOperation.InvalidToken	Invalid token.
UnauthorizedOperation.MFAExpired	The Multi-factor authentication (MFA) code expired.
UnauthorizedOperation.MFANotFound	Multi-factor authentication (MFA) does not exist.

ModifyInvoker

最近更新时间：2023-03-14 16:55:45

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to modify an invoker.

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: ModifyInvoker.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
InvokerId	Yes	String	ID of the invoker to be modified.
Name	No	String	Name of the invoker to be modified.
Type	No	String	Invoker type. It can only be <code>SCHEDULE</code> (recurring invokers).
CommandId	No	String	ID of the command to be modified.
Username	No	String	The username to be modified.

Parameters	No	String	Custom parameters to be modified.
InstanceIds.N	No	Array of String	List of instance IDs to be modified. Up to 100 IDs are allowed.
ScheduleSettings	No	ScheduleSettings	Scheduled invoker settings to be modified.

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Modifying an invoker

Input Example

```

POST / HTTP/1.1
Host: tat.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyInvoker
<Common request parameters>

{
  "InvokerId": "ivk-b7s3qa51",
  "Parameters": "{\"var\": \"1\"}"
}
    
```

Output Example

```

{
  "Response": {
    "RequestId": "7f3acde8-196d-4be4-9fa7-359f79b026bc"
  }
}
    
```

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 NodeJS](#)
- [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
AuthFailure	CAM signature/authentication error.
InternalError	Internal error.
InvalidParameterValue.InconsistentInstance	Inconsistent instance type.
InvalidParameterValue.InvalidInstanceId	Invalid instance ID.
InvalidParameterValue.InvalidInvokerId	Invalid InvokerId.
InvalidParameterValue.ParameterDisabled	The custom parameter feature is not enabled.
InvalidParameterValue.ParameterInvalidJsonFormat	The parameter is not a valid JSON string.
ResourceNotFound.CommandNotFound	The command does not exist.
ResourceNotFound.InstanceNotFound	The instance does not exist.

ResourceUnavailable.AgentNotInstalled

TAT Agent is not installed.

DeleteInvoker

最近更新时间：2023-03-14 16:55:47

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to delete an invoker.

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: DeleteInvoker.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
InvokerId	Yes	String	ID of the invoker to be deleted.

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Deleting an invoker

Input Example

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

{
  "InvokerId": "ivk-b7s3qa51"
}
```

Output Example

```
{
  "Response": {
    "RequestId": "bc16e010-7b50-4661-88e8-4d2c77d15558"
  }
}
```

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 NodeJS](#)
- [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
AuthFailure	CAM signature/authentication error.
InvalidParameterValue.InvalidInvokerId	Invalid InvokerId.
ResourceNotFound	The resource does not exist.

PreviewReplacedCommandContent

最近更新时间：2023-03-14 16:55:44

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to preview the command with custom parameters. The command is not executed.

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: PreviewReplacedCommandContent.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
Parameters	No	String	Custom parameters for the preview. The field type is JSON encoded string. For example, {"varA": "222"}. <code>key</code> is the name of the custom parameter and "value" is its specified value. Both "key" and "value" are strings. At most 20 custom parameters are supported. The name of the custom parameter cannot exceed 64 characters and can only contain [a-z], [A-Z], [0-9], [-_]. This parameter can be left empty if DefaultParameters is set for the previewed CommandId.
CommandId	No	String	The command to be previewed. If DefaultParameters is set, it is

			combined with Parameters and Parameters takes priority. CommandId or Content must be specified.
Content	No	String	Base64-encoded command to be previewed. The maximum length is 64 KB. CommandId or Content must be specified.

3. Output Parameters

Parameter Name	Type	Description
ReplacedContent	String	Base64-encoded command with custom parameters.
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Previewing the result of replacing command content

Input Example

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

{
  "Parameters": "{\"a\": \"123\"}",
  "Content": "bHMge3thfX0KZWNobyB7e2J9fSB7e2N9fQ=="
}
```

Output Example

```
{
  "Response": {
    "RequestId": "0b4c6010-42a7-45cd-b8c3-daa7de930e82",
    "ReplacedContent": "bHMgMTIzCmVjaG8ge3tifX0ge3tjfx0="
  }
}
```


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 NodeJS](#)
- [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.ConflictParameter	Conflicting parameters.
InvalidParameter.ParameterNameDuplicated	Duplicate parameter name.
InvalidParameterValue.CommandContentInvalid	Invalid command content.
InvalidParameterValue.CommandNameDuplicated	Duplicate command name.
InvalidParameterValue.InvalidCommandId	Invalid CommandId.
InvalidParameterValue.LackOfParameterInfo	The custom parameter feature is enabled, but custom parameters are missing.
InvalidParameterValue.LackOfParameters	Parameters not provided.
InvalidParameterValue.ParameterDisabled	The custom parameter feature is not enabled.

InvalidParameterValue.ParameterInvalidJsonFormat	The parameter is not a valid JSON string.
InvalidParameterValue.ParameterKeyContainsInvalidChar	The parameter key contains invalid characters.
InvalidParameterValue.ParameterKeyDuplicated	Duplicate parameter keys.
InvalidParameterValue.ParameterKeyLenExceeded	The parameter key is too long.
InvalidParameterValue.ParameterNumberExceeded	Too many parameters.
InvalidParameterValue.ParameterValueNotString	The parameter value is not a string.
InvalidParameterValue.Range	The parameter value is not in the valid range.
InvalidParameterValue.SupportParametersOnlyIfEnableParameter	The custom parameter feature is not enabled.
ResourceNotFound.CommandNotFound	The command does not exist.
UnauthorizedOperation.CamAuthFailed	CAM authentication failed.
UnauthorizedOperation.InvalidToken	Invalid token.
UnauthorizedOperation.MFAExpired	The Multi-factor authentication (MFA) code expired.
UnauthorizedOperation.MFANotFound	Multi-factor authentication (MFA) does not exist.

DisableInvoker

最近更新时间：2023-03-14 16:55:45

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to disable an invoker.

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: DisableInvoker.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
InvokerId	Yes	String	ID of the invoker to be disabled.

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Disabling an invoker

Input Example

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

{
  "InvokerId": "ivk-b7s3qa51"
}
```

Output Example

```
{
  "Response": {
    "RequestId": "a6c5b0bd-4273-47c0-8d34-c32822f3ccb0"
  }
}
```

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 NodeJS](#)
- [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
AuthFailure	CAM signature/authentication error.
InvalidParameterValue.InvalidInvokerId	Invalid InvokerId.
ResourceNotFound	The resource does not exist.

DescribeInvokers

最近更新时间：2023-03-14 16:55:46

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to query invoker 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: DescribeInvokers.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
InvokerIds.N	No	Array of String	List of invoker IDs.
Filters.N	No	Array of Filter	Filter conditions: <ul style="list-style-type: none"> <code>invoker-id</code> - String - Required: No - (Filter condition) Filter by the invoker ID. <code>command-id</code> - String - Required: No - (Filter condition) Filter by the command ID. <code>type</code> - String - Required: No - (Filter condition) Filter by the invoker type.

Limit	No	Integer	Number of returned results. Default value: 20. Maximum value: 100.
Offset	No	Integer	Offset. Default value: 0.

3. Output Parameters

Parameter Name	Type	Description
TotalCount	Integer	Number of matching invokers.
InvokerSet	Array of Invoker	Invoker information.
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Querying invokers

Input Example

```

POST / HTTP/1.1
Host: tat.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeInvokers
<Common request parameters>

{
  "Filters": [
    {
      "Name": "invoker-id",
      "Values": [
        "ivk-b7s3qa51"
      ]
    }
  ]
}
    
```

Output Example

```
{
  "Response": {
    "RequestId": "a4c828cf-31c9-42d2-a558-2ad340a76ef0",
    "TotalCount": 1,
    "InvokerSet": [
      {
        "InvokerId": "ivk-b7s3qa51",
        "Name": "test-invoker",
        "Type": "SCHEDULE",
        "CommandId": "cmd-m7uma92n",
        "Username": "root",
        "Parameters": "{\"var\": \"1\"}",
        "InstanceIds": [
          "ins-yx05ky8j"
        ],
        "Enable": false,
        "ScheduleSettings": {
          "Policy": "ONCE",
          "Recurrence": "",
          "InvokeTime": ""
        },
        "CreatedTime": "2021-08-30T06:42:02Z",
        "UpdatedTime": "2021-09-09T12:07:00Z"
      }
    ]
  }
}
```

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 NodeJS](#)
- [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
AuthFailure	CAM signature/authentication error.
InvalidParameter.ConflictParameter	Conflicting parameters.
InvalidParameterValue	Invalid parameter value.
InvalidParameterValue.InvalidFilter	Invalid filter.
InvalidParameterValue.InvalidInvokerId	Invalid InvokerId.

DescribeInvokerRecords

最近更新时间：2023-03-14 16:55:46

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to query the execution history of an invoker.

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: DescribeInvokerRecords.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
InvokerIds.N	No	Array of String	List of invoker IDs. Up to 100 IDs are allowed.
Limit	No	Integer	Number of returned results. Default value: 20. Maximum value: 100.
Offset	No	Integer	Offset. Default value: 0.

3. Output Parameters

Parameter Name	Type	Description
TotalCount	Integer	Number of matching records.
InvokerRecordSet	Array of InvokerRecord	Execution history of an invoker.
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Querying the execution history of an invoker

Input Example

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

{
  "InvokerIds": [
    "ivk-b7s3qa51"
  ],
  "Offset": 0,
  "Limit": 2
}
```

Output Example

```
{
  "Response": {
    "RequestId": "0398dcea-b3de-4ec9-8e78-976191e0a94f",
    "TotalCount": 1443,
    "InvokerRecordSet": [
      {
        "InvokerId": "ivk-b7s3qa51",
        "Reason": "start an invocation at scheduled time.",
        "InvocationId": "inv-l9l4orf1",
        "Result": "SUCCESS",
        "InvokeTime": "2021-09-06T09:33:05Z"
      },
    ],
  }
}
```

```

{
  "InvokerId": "ivk-b7s3qa51",
  "Reason": "start an invocation at scheduled time.",
  "InvocationId": "inv-7ojgezbd",
  "Result": "SUCCESS",
  "InvokeTime": "2021-09-06T09:30:05Z"
}
]
}
}
    
```

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 NodeJS](#)
- [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
AuthFailure	CAM signature/authentication error.
InvalidParameterValue	Invalid parameter value.
InvalidParameterValue.InvalidInvokerId	Invalid InvokerId.

RunCommand

最近更新时间：2023-03-14 16:55:44

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to execute a command and returns the execution activity ID (inv-xxxxxxx) on success. Each execution has one or more execution tasks (invt-xxxxxxx) and each execution task indicates an execution record on a CVM or Lighthouse instance.

- If the agent is not installed on the instance or is offline, an error is returned.
- If the command type is not supported by the agent runtime environment, an error is returned.
- The specified instance needs to be in a VPC network.
- The specified instance needs to be in the `RUNNING` status.
- Only one type of instances (CVM or Lighthouse) can be specified in a single request.

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: RunCommand.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.
Content	Yes	String	Base64-encoded command. The maximum length is 64 KB.

InstanceIds.N	Yes	Array of String	<p>IDs of instances about to execute commands. Up to 100 IDs are allowed. Supported instance types:</p> <ul style="list-style-type: none"> CVM LIGHTHOUSE
CommandName	No	String	<p>Command name. The name can be up to 60 bytes, and contain [a-z], [A-Z], [0-9] and [_-].</p>
Description	No	String	<p>Command description. The maximum length is 120 characters.</p>
CommandType	No	String	<p>Command type. SHELL and POWERSHELL are supported. The default value is SHELL .</p>
WorkingDirectory	No	String	<p>Command execution path. The default value is /root for SHELL commands and C:\Program Files\qcloud\tat_agent\workdir for POWERSHELL commands.</p>
Timeout	No	Integer	<p>Command timeout period. Default value: 60 seconds. Value range: [1, 86400].</p>
SaveCommand	No	Boolean	<p>Whether to save the command. Valid values:</p> <ul style="list-style-type: none"> True : Save False : Do not save <p>The default value is False .</p>
EnableParameter	No	Boolean	<p>Whether to enable the custom parameter feature. This cannot be modified once created. Default value: false .</p>
DefaultParameters	No	String	<p>The default value of the custom parameter value when it is enabled. The field type is JSON encoded string. For example, {"varA": "222"}.</p> <p>key is the name of the custom parameter and value is the default value. Both key and value are strings. If Parameters is not provided, the default values specified here are used.</p> <p>Up to 20 custom parameters are supported. The name of the custom parameter cannot exceed 64 characters and can contain [a-z], [A-Z], [0-9] and [_-].</p>
Parameters	No	String	<p>Custom parameters of Command . The field type is JSON encoded string. For example, {"varA": "222"}.</p> <p>key is the name of the custom parameter and value is the default value. Both key and value are strings.</p>

			<p>If no parameter value is provided, the <code>DefaultParameters</code> is used.</p> <p>Up to 20 custom parameters are supported.</p> <p>The name of the custom parameter cannot exceed 64 characters and can contain [a-z], [A-Z], [0-9] and [-_].</p>
Tags.N	No	Array of Tag	<p>The tags of the command. It is available when <code>SaveCommand</code> is <code>True</code>. A maximum of 10 tags are allowed.</p>
Username	No	String	<p>The username used to execute the command on the CVM or Lighthouse instance.</p> <p>The principle of least privilege is the best practice for permission management. We recommend you execute TAT commands as a general user. By default, the user <code>root</code> is used to execute commands on Linux and the user <code>System</code> is used on Windows.</p>
OutputCOSBucketUrl	No	String	<p>The COS bucket URL for uploading logs. The URL must start with <code>https</code>, such as <code>https://BucketName-123454321.cos.ap-beijing.myqcloud.com</code>.</p>
OutputCOSKeyPrefix	No	String	<p>The COS bucket directory where the logs are saved. Check below for the rules of the directory name.</p> <ol style="list-style-type: none"> 1. It must be a combination of number, letters, and visible characters. Up to 60 characters are allowed. 2. Use a slash (/) to create a subdirectory. 3. "." can not be used as the folder name. It cannot start with a slash (/), and cannot contain consecutive slashes.

3. Output Parameters

Parameter Name	Type	Description
CommandId	String	Command ID.
InvocationId	String	Execution activity ID.
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Executing a command on multiple CVM instances

Input Example

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

{
  "CommandName": "run-command",
  "SaveCommand": false,
  "Description": "whoami",
  "Content": "d2hvYW1p",
  "CommandType": "SHELL",
  "WorkingDirectory": "/root/",
  "Timeout": 60,
  "InstanceIds": [
    "ins-zj0f57ew",
    "ins-zj0f57ex",
    "ins-zj0f57ev"
  ]
}
```

Output Example

```
{
  "Response": {
    "RequestId": "f3e3a951-56b1-4042-af23-ba50223a8f60",
    "CommandId": "cmd-5oa8jajm",
    "InvocationId": "inv-mp6m9vmu"
  }
}
```

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 NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
FailedOperation.CVMError	Failed to access the CVM.
FailedOperation.LighthouseError	Failed to access the Lighthouse instance.
InternalError	Internal error.
InvalidParameter	Invalid parameter.
InvalidParameter.InvalidUsername	Invalid username.
InvalidParameter.ParameterNameDuplicated	Duplicate parameter name.
InvalidParameterValue	Invalid parameter value.
InvalidParameterValue.AgentUnsupportedCommandType	TAT Agent does not support this command type.
InvalidParameterValue.CommandContentInvalid	Invalid command content.
InvalidParameterValue.CommandNameDuplicated	Duplicate command name.
InvalidParameterValue.InconsistentInstance	Inconsistent instance type.
InvalidParameterValue.InvalidCommandName	Invalid command name.
InvalidParameterValue.InvalidContent	Invalid command.

InvalidParameterValue.InvalidInstanceId	Invalid instance ID.
InvalidParameterValue.InvalidOutputCOSBucketUrl	Invalid OutputCOSBucketUrl.
InvalidParameterValue.InvalidOutputCOSKeyPrefix	Invalid OutputCOSKeyPrefix.
InvalidParameterValue.InvalidUsername	Invalid username.
InvalidParameterValue.InvalidWorkingDirectory	Invalid command execution path.
InvalidParameterValue.LackOfParameterInfo	The custom parameter feature is enabled, but custom parameters are missing.
InvalidParameterValue.LimitExceeded	Parameter limit exceeded.
InvalidParameterValue.ParameterDisabled	The custom parameter feature is not enabled.
InvalidParameterValue.ParameterInvalidJsonFormat	The parameter is not a valid JSON string.
InvalidParameterValue.ParameterKeyContainsInvalidChar	The parameter key contains invalid characters.
InvalidParameterValue.ParameterKeyDuplicated	Duplicate parameter keys.
InvalidParameterValue.ParameterKeyLenExceeded	The parameter key is too long.
InvalidParameterValue.ParameterNumberExceeded	Too many parameters.
InvalidParameterValue.ParameterValueNotString	The parameter value is not a string.
InvalidParameterValue.Range	The parameter value is not in the valid range.
InvalidParameterValue.SupportParametersOnlyIfEnableParameter	The custom parameter feature is not enabled.
InvalidParameterValue.TooLong	Length limit exceeded.
MissingParameter	Missing parameter.
ResourceNotFound.InstanceNotFound	The instance does not exist.
ResourceNotFound.RoleNotFound	The role does not exist.
ResourceUnavailable.AgentNotInstalled	TAT Agent is not installed.
ResourceUnavailable.AgentStatusNotOnline	TAT Agent is offline.

ResourceUnavailable.InstanceStateNotRunning	The instance is not running.
ResourceUnavailable.LighthouseUnsupportedRegion	Lighthouse is not supported in the region.
UnauthorizedOperation.CamAuthFailed	CAM authentication failed.
UnauthorizedOperation.InvalidToken	Invalid token.
UnauthorizedOperation.MFAExpired	The Multi-factor authentication (MFA) code expired.
UnauthorizedOperation.MFANotFound	Multi-factor authentication (MFA) does not exist.

Region APIs

DescribeRegions

最近更新时间：2023-06-12 16:10:38

1. API Description

Domain name for API request: tat.tencentcloudapi.com.

This API is used to query the list of regions that supports TAT.

If the `RegionState` is `AVAILABLE`, it means that TAT is available in the region. If no value is returned, TAT is not available in the region.

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: DescribeRegions.
Version	Yes	String	Common Params . The value used for this API: 2020-10-28.
Region	No	String	Common Params . This parameter is not required for this API.

3. Output Parameters

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

Parameter Name	Type	Description
TotalCount	Integer	Number of regions
RegionSet	Array of RegionInfo	Region information list
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Querying regions that support TAT

Input Example

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

{}
```

Output Example

```
{
  "Response": {
    "TotalCount": 13,
    "RegionSet": [
      {
        "Region": "ap-guangzhou",
        "RegionName": "Guangzhou",
        "RegionState": "AVAILABLE"
      },
      {
        "Region": "ap-nanjing",
        "RegionName": "Nanjing",
        "RegionState": "AVAILABLE"
      },
      {
        "Region": "ap-shanghai",
        "RegionName": "Shanghai",
```

```
"RegionState": "AVAILABLE"
},
{
  "Region": "ap-hongkong",
  "RegionName": "Hong Kong (China)",
  "RegionState": "AVAILABLE"
},
{
  "Region": "ap-beijing",
  "RegionName": "Beijing",
  "RegionState": "AVAILABLE"
},
{
  "Region": "ap-singapore",
  "RegionName": "Singapore",
  "RegionState": "AVAILABLE"
},
{
  "Region": "na-siliconvalley",
  "RegionName": "Silicon Valley",
  "RegionState": "AVAILABLE"
},
{
  "Region": "ap-chengdu",
  "RegionName": "Chengdu",
  "RegionState": "AVAILABLE"
},
{
  "Region": "eu-frankfurt",
  "RegionName": "Frankfurt",
  "RegionState": "AVAILABLE"
},
{
  "Region": "ap-seoul",
  "RegionName": "Seoul",
  "RegionState": "AVAILABLE"
},
{
  "Region": "ap-chongqing",
  "RegionName": "Chongqing",
  "RegionState": "AVAILABLE"
},
{
  "Region": "ap-mumbai",
  "RegionName": "Mumbai",
  "RegionState": "AVAILABLE"
},
},
```

```
],  
  "RequestId": "6fb7f9db-b7da-4cb8-a912-3a3b1690f3a6"  
}
```

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 NodeJS](#)
- [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
InternalError	Internal error.
UnauthorizedOperation.CamAuthFailed	CAM authentication failed.
UnauthorizedOperation.InvalidToken	Invalid token.
UnauthorizedOperation.MFAExpired	The Multi-factor authentication (MFA) code expired.
UnauthorizedOperation.MFANotFound	Multi-factor authentication (MFA) does not exist.

Data Types

最近更新时间：2023-03-14 16:55:48

AutomationAgentInfo

TAT agent information

Used by actions: DescribeAutomationAgentStatus.

Name	Type	Description
InstanceId	String	Instance ID.
Version	String	Agent version.
LastHeartbeatTime	Timestamp ISO8601	Last heartbeat time
AgentStatus	String	Agent status. Valid values: <ul style="list-style-type: none"> Online Offline
Environment	String	Agent runtime environment. Valid values: <ul style="list-style-type: none"> Linux : Linux instance Windows : Windows instance
SupportFeatures	Array of String	Features supported by the TAT agent.

Command

Command details.

Used by actions: DescribeCommands.

Name	Type	Description
CommandId	String	Command ID.
CommandName	String	Command name.
Description	String	Command description.
Content	String	Base64-encoded command.

CommandType	String	Command type.
WorkingDirectory	String	Command execution path.
Timeout	Integer	Command timeout period.
CreatedTime	Timestamp ISO8601	Command creation time.
UpdatedTime	Timestamp ISO8601	Command update time.
EnableParameter	Boolean	Whether to enable the custom parameter feature.
DefaultParameters	String	Default custom parameter value.
FormattedDescription	String	Formatted description of the command. This parameter is an empty string for user commands and contains values for public commands.
CreatedBy	String	Command creator. <code>TAT</code> indicates a public command and <code>USER</code> indicates a personal command.
Tags	Array of Tag	The list of tags bound to the command.
Username	String	The user who executes the command on the instance.
OutputCOSBucketUrl	String	The COS bucket URL for uploading logs.
OutputCOSKeyPrefix	String	The COS bucket directory where the logs are saved.

CommandDocument

Command execution details.

Used by actions: DescribeInvocationTasks.

Name	Type	Description
Content	String	Base64-encoded command.
CommandType	String	Command type.
Timeout	Integer	Timeout period.
WorkingDirectory	String	Execution path.

Username	String	The user who executes the command.
OutputCOSBucketUrl	String	URL of the COS bucket to store the output
OutputCOSKeyPrefix	String	Prefix of the output file name

Filter

Key-value pair filters for conditional filtering queries, such as filtering ID, name, and status.

- If there are multiple `Filter` parameters, the relationship among them is the logical `AND`.
- If there are multiple `Values` for the same `Filter`, the relationship among the `Values` for the same `Filter` is the logical `OR`.

Take the [DescribeInstances](#) API as an example. You can use the following filters to query the instances whose availability zone (`zone`) is Guangzhou 1 **and** billing method (`instance-charge-type`) is prepaid **or** pay-as-you-go:

```
Filters.0.Name=zone
&Filters.0.Values.0=ap-guangzhou-1
&Filters.1.Name=instance-charge-type
&Filters.1.Values.0=PREPAID
&Filters.1.Values.1=POSTPAID_BY_HOUR
```

Used by actions: DescribeAutomationAgentStatus, DescribeCommands, DescribeInvocationTasks, DescribeInvocations, DescribeInvokers.

Name	Type	Required	Description
Name	String	Yes	Field to be filtered.
Values	Array of String	Yes	Filter values of the field.

Invocation

Execution activity details.

Used by actions: DescribeInvocations.

--	--	--

Name	Type	Description
InvocationId	String	Execution activity ID.
CommandId	String	Command ID.
InvocationStatus	String	Execution task status. Valid values: <ul style="list-style-type: none"> PENDING: Pending RUNNING: Running SUCCESS: Success FAILED: Failed TIMEOUT: Command timed out PARTIAL_FAILED: Partial failure
InvocationTaskBasicInfoSet	Array of InvocationTaskBasicInfo	Execution task information list.
Description	String	Execution activity description.
StartTime	Timestamp ISO8601	Start time of the execution activity.
EndTime	Timestamp ISO8601	End time of the execution activity.
CreatedTime	Timestamp ISO8601	Time when the execution activity is created.
UpdatedTime	Timestamp ISO8601	Time when the execution activity is updated.
Parameters	String	Values of custom parameters.
DefaultParameters	String	Default custom parameter value.
InstanceKind	String	Type of the instance executing the command. Valid values: <code>CVM</code> , <code>LIGHTHOUSE</code> .
Username	String	The user who executes the command on the instance.
InvocationSource	String	Invocation source.
CommandContent	String	Base64-encoded command
CommandType	String	Command type
Timeout	Integer	Command timeout period, in seconds.
WorkingDirectory	String	Working directory for executing the command.
OutputCOSBucketUrl	String	The COS bucket URL for uploading logs.

OutputCOSKeyPrefix	String	The COS bucket directory where the logs are saved.
--------------------	--------	--

InvocationTask

Execution task.

Used by actions: DescribeInvocationTasks.

Name	Type	Description
InvocationId	String	Execution activity ID.
InvocationTaskId	String	Execution task ID.
CommandId	String	Command ID.
TaskStatus	String	Execution task status. Valid values: <ul style="list-style-type: none"> PENDING: Pending DELIVERING: Delivering DELIVER_DELAYED: Delivery delayed DELIVER_FAILED: Delivery failed START_FAILED: Failed to start the command RUNNING: Running SUCCESS: Success FAILED: Failed to execute the command. The exit code is not 0 after execution. TIMEOUT: Command timed out TASK_TIMEOUT: Task timed out CANCELLING: Canceling CANCELLED: Canceled (canceled before execution) TERMINATED: Terminated (canceled during execution)
InstanceId	String	Instance ID.
TaskResult	TaskResult	Execution result.
StartTime	Timestamp ISO8601	Start time of the execution task.
EndTime	Timestamp ISO8601	End time of the execution task.
CreatedTime	Timestamp ISO8601	Creation time.

UpdatedTime	Timestamp ISO8601	Update time.
CommandDocument	CommandDocument	Command details of the execution task.
ErrorInfo	String	Error message displayed when the execution task fails.
InvocationSource	String	Invocation source.

InvocationTaskBasicInfo

Execution task description.

Used by actions: DescribeInvocations.

Name	Type	Description
InvocationTaskId	String	Execution task ID.
TaskStatus	String	Execution task status. Valid values: <ul style="list-style-type: none"> PENDING: Pending DELIVERING: Delivering DELIVER_DELAYED: Delivery delayed DELIVER_FAILED: Delivery failed START_FAILED: Failed to start the command RUNNING: Running SUCCESS: Success FAILED: Failed to execute the command. The exit code is not 0 after execution. TIMEOUT: Command timed out TASK_TIMEOUT: Task timed out CANCELLING: Canceling CANCELLED: Canceled (canceled before execution) TERMINATED: Terminated (canceled during execution)
InstanceId	String	Instance ID.

Invoker

Invoker information.

Used by actions: DescribeInvokers.

--	--	--

Name	Type	Description
InvokerId	String	Invoker ID.
Name	String	Invoker name.
Type	String	Invoker type.
CommandId	String	Command ID.
Username	String	Username.
Parameters	String	Custom parameters.
InstanceIds	Array of String	Instance ID list.
Enable	Boolean	Whether to enable the invoker.
ScheduleSettings	ScheduleSettings	Execution schedule of the invoker. This field is returned for recurring invokers. Note: This field may return <code>null</code> , indicating that no valid values can be obtained.
CreatedTime	Timestamp ISO8601	Creation time.
UpdatedTime	Timestamp ISO8601	Modification time.

InvokerRecord

Execution history of the invoker.

Used by actions: DescribeInvokerRecords.

Name	Type	Description
InvokerId	String	Invoker ID.
InvokeTime	Timestamp ISO8601	Execution time.
Reason	String	Execution reason.
InvocationId	String	Command execution ID.
Result	String	Trigger result.

RegionInfo

Information of a region.

Used by actions: DescribeRegions.

Name	Type	Description
Region	String	Region name, such as <code>ap-guangzhou</code>
RegionName	String	Region description, such as <code>Guangzhou</code>
RegionState	String	Region status. <code>AVAILABLE</code> indicates the region is available.

ScheduleSettings

Settings of a scheduled invoker

Used by actions: CreateInvoker, DescribeInvokers, ModifyInvoker.

Name	Type	Required	Description
Policy	String	Yes	Execution policy: <ul style="list-style-type: none"> <code>ONCE</code> : Execute once <code>RECURRENCE</code> : Execute repeatedly
Recurrence	String	No	Trigger the crontab expression. This field is required if <code>Policy</code> is <code>RECURRENCE</code> . The crontab expression is parsed in UTC+8.
InvokeTime	Timestamp ISO8601	No	The next execution time of the invoker. This field is required if <code>Policy</code> is <code>ONCE</code> .

Tag

Information on tags

Used by actions: CreateCommand, DescribeCommands, RunCommand.

Name	Type	Required	Description

Key	String	Yes	Tag key.
Value	String	Yes	Tag value.

TaskResult

Task result.

Used by actions: DescribeInvocationTasks.

Name	Type	Description
ExitCode	Integer	ExitCode of the execution.
Output	String	Base64-encoded command output. The maximum length is 24 KB.
ExecStartTime	Timestamp ISO8601	Time when the execution is started.
ExecEndTime	Timestamp ISO8601	Time when the execution is ended.
Dropped	Integer	Dropped bytes of the command output.
OutputUrl	String	COS URL of the logs.
OutputUploadCOSErrorInfo	String	Error message for uploading logs to COS.

Error Codes

最近更新时间：2023-03-14 16:55:48

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.
InternalServerError	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
AuthFailure	CAM signature/authentication error.
FailedOperation.CVMError	Failed to access the CVM.
FailedOperation.LighthouseError	Failed to access the Lighthouse instance.
InvalidParameter.ConflictParameter	Conflicting parameters.
InvalidParameter.InvalidUsername	Invalid username.
InvalidParameter.ParameterNameDuplicated	Duplicate parameter name.
InvalidParameterValue.AgentUnsupportedCommandType	TAT Agent does not support this command type.
InvalidParameterValue.CommandContentInvalid	Invalid command content.
InvalidParameterValue.CommandNameDuplicated	Duplicate command name.

InvalidParameterValue.InconsistentInstance	Inconsistent instance type.
InvalidParameterValue.InstanceIsNotRelatedToInvocation	The instance with this ID has no such execution activity.
InvalidParameterValue.InvalidCommandId	Invalid CommandId.
InvalidParameterValue.InvalidCommandName	Invalid command name.
InvalidParameterValue.InvalidContent	Invalid command.
InvalidParameterValue.InvalidCronExpression	Invalid crontab expression.
InvalidParameterValue.InvalidFilter	Invalid filter.
InvalidParameterValue.InvalidInstanceId	Invalid instance ID.
InvalidParameterValue.InvalidInvocationId	Invalid execution activity ID.
InvalidParameterValue.InvalidInvocationTaskId	Invalid execution task ID.
InvalidParameterValue.InvalidInvokerId	Invalid InvokerId.
InvalidParameterValue.InvalidOutputCOSBucketUrl	Invalid OutputCOSBucketUrl.
InvalidParameterValue.InvalidOutputCOSKeyPrefix	Invalid OutputCOSKeyPrefix.
InvalidParameterValue.InvalidTimeFormat	Invalid time format.
InvalidParameterValue.InvalidUsername	Invalid username.
InvalidParameterValue.InvalidWorkingDirectory	Invalid command execution path.
InvalidParameterValue.InvokeTimeExpired	API invocation expired.
InvalidParameterValue.LackOfParameterInfo	The custom parameter feature is enabled, but custom parameters are missing.
InvalidParameterValue.LackOfParameters	Parameters not provided.
InvalidParameterValue.LimitExceeded	Parameter limit exceeded.
InvalidParameterValue.ParameterDisabled	The custom parameter feature is not enabled.
InvalidParameterValue.ParameterInvalidJsonFormat	The parameter is not a valid JSON string.
InvalidParameterValue.ParameterKeyContainsInvalidChar	The parameter key contains invalid characters.

InvalidParameterValue.ParameterKeyDuplicated	Duplicate parameter keys.
InvalidParameterValue.ParameterKeyLenExceeded	The parameter key is too long.
InvalidParameterValue.ParameterNumberExceeded	Too many parameters.
InvalidParameterValue.ParameterValueNotString	The parameter value is not a string.
InvalidParameterValue.Range	The parameter value is not in the valid range.
InvalidParameterValue.SupportParametersOnlyIfEnableParameter	The custom parameter feature is not enabled.
InvalidParameterValue.ToolLong	Length limit exceeded.
LimitExceeded.FilterValueExceeded	Too many <code>Filter</code> values.
ResourceNotFound.CommandNotFound	The command does not exist.
ResourceNotFound.InstanceNotFound	The instance does not exist.
ResourceNotFound.InvocationNotFound	No execution activity found.
ResourceNotFound.RoleNotFound	The role does not exist.
ResourceUnavailable.AgentNotInstalled	TAT Agent is not installed.
ResourceUnavailable.AgentStatusNotOnline	TAT Agent is offline.
ResourceUnavailable.CommandInExecuting	The command is being executed.
ResourceUnavailable.CommandInInvoker	The command has been bound to an invoker.
ResourceUnavailable.InstanceStateNotRunning	The instance is not running.
ResourceUnavailable.LighthouseUnsupportedRegion	Lighthouse is not supported in the region.
UnauthorizedOperation.CamAuthFailed	CAM authentication failed.
UnauthorizedOperation.InvalidToken	Invalid token.
UnauthorizedOperation.MFAExpired	The Multi-factor authentication (MFA) code expired.
UnauthorizedOperation.MFANotFound	Multi-factor authentication (MFA) does not exist.