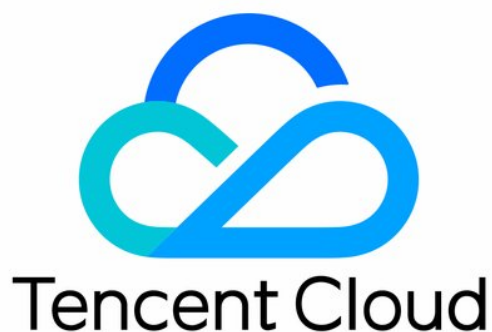


Tencent Smart Advisor-Chaotic Fault Generator

API Documentation

Product Documentation



Copyright Notice

©2013-2025 Tencent Cloud. All rights reserved.

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

Trademark Notice



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

Service Statement

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

Contents

API Documentation

History

Introduction

API Category

Making API Requests

Request Structure

Common Params

Signature v3

Signature

Responses

Task APIs

CreateTaskFromTemplate

DeleteTask

DescribeActionFieldConfigList

DescribeActionLibraryList

DescribeObjectTypeList

DescribeTask

DescribeTaskExecuteLogs

DescribeTaskList

DescribeTemplate

ExecuteTask

ExecuteTaskInstance

ModifyTaskRunStatus

TriggerPolicy

DescribeTaskPolicyTriggerLog

CreateTaskFromAction

Template Library APIs

DescribeTemplateList

Data Types

Error Codes

API Documentation

History

Last updated : 2024-09-25 10:27:42

Release 1

Release time: 2024-09-19 09:52:28

Release updates:

Improvement to existing documentation.

New APIs:

- [CreateTaskFromAction](#)
- [CreateTaskFromTemplate](#)
- [DeleteTask](#)
- [DescribeActionFieldConfigList](#)
- [DescribeActionLibraryList](#)
- [DescribeObjectTypeList](#)
- [DescribeTask](#)
- [DescribeTaskExecuteLogs](#)
- [DescribeTaskList](#)
- [DescribeTaskPolicyTriggerLog](#)
- [DescribeTemplate](#)
- [DescribeTemplateList](#)
- [ExecuteTask](#)
- [ExecuteTaskInstance](#)
- [ModifyTaskRunStatus](#)
- [TriggerPolicy](#)

New data structures:

- [ActionFieldConfigDetail](#)
- [ActionFieldConfigResult](#)
- [ActionFilter](#)
- [ActionLibraryListResult](#)
- [ApmServiceInfo](#)

- [DescribePolicy](#)
- [ObjectType](#)
- [ObjectTypeConfig](#)
- [ObjectTypeConfigFields](#)
- [ObjectTypeJsonParse](#)
- [PolicyTriggerLog](#)
- [ResourceOffline](#)
- [TagWithCreate](#)
- [TagWithDescribe](#)
- [Task](#)
- [TaskConfig](#)
- [TaskGroup](#)
- [TaskGroupAction](#)
- [TaskGroupActionConfig](#)
- [TaskGroupConfig](#)
- [TaskGroupInstance](#)
- [TaskGroupInstancesExecuteRules](#)
- [TaskListItem](#)
- [TaskMonitor](#)
- [TaskReportInfo](#)
- [Template](#)
- [TemplateGroup](#)
- [TemplateGroupAction](#)
- [TemplateListItem](#)
- [TemplateMonitor](#)
- [TemplatePolicy](#)

Introduction

Last updated : 2024-09-25 10:27:32

The Chaotic Fault Generator (CFG) is a product that follows the experimental principles of chaos engineering and combines them with Tencent Cloud's internal practices to provide highly available experiment services based on real online faults, which can help users' system improve fault tolerance and recoverability.

API Category

Last updated : 2024-09-25 10:27:33

Task APIs

API Name	Feature	Frequency Limit (maximum requests per second)
CreateTaskFromTemplate	Creates an experiment using a template	10
DeleteTask	Deletes a task	10
DescribeActionFieldConfigList	Obtains the configuration parameter list of the action field	20
DescribeActionLibraryList	Obtains the action list	100
DescribeObjectTypeList	Queries the object type list	20
DescribeTask	Queries a task	20
DescribeTaskExecuteLogs	Obtains logs generated during an experiment	20
DescribeTaskList	Queries the task list	100
DescribeTemplate	Queries a template library	20
ExecuteTask	Runs a task	10
ExecuteTaskInstance	Executes the task action on the instance	10
ModifyTaskRunStatus	Changes the task running status	10
CreateTaskFromAction	Creates an experiment with an action	20
DescribeTaskPolicyTriggerLog	Obtains guardrail triggering logs	20
TriggerPolicy	Triggers the guardrail policy	20

Template Library APIs

API Name	Feature	Frequency Limit (maximum requests per second)
DescribeTemplateList	Queries the template list	100

Making API Requests

Request Structure

Last updated : 2024-12-02 14:41:46

1. Service Address

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

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

Note: As financial availability zones and non-financial availability zones are isolated, when accessing the services in a financial availability zone (with the common parameter `Region` specifying a financial availability zone), it is necessary to specify a domain name of the financial availability zone, preferably in the same region as specified in `Region`.

Access region for financial availability zone	Domain name for financial availability zone
East China (Shanghai Finance)	cfg.ap-shanghai-fsi.tencentcloudapi.com
South China (Shenzhen Finance)	cfg.ap-shenzhen-fsi.tencentcloudapi.com

2. Communications Protocol

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

3. Request Methods

Supported HTTP request methods:

- POST (recommended)
- GET

The Content-Type types supported by POST requests:

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

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

4. Character Encoding

Only UTF-8 encoding is used.

Common Params

Last updated : 2024-12-02 14:41:46

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=AKID*****/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc96317 Here: - TC3-HMAC-SHA256: Signature method, currently fixed as this value; - Credential: Signature credential; AKID***** is the SecretId; Date is a d time, and this value must match the value of X-TC-Timestamp (a commo

			UTC time format; service is the name of the product/service, and is general name prefix. For example, a domain name <code>cvm.tencentcloudapi.com</code> refers to product and the value would be <code>cvm</code>; - 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 to obtain the temporary key and token by calling a CAM API. No token is required for a long-term key.

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

Example of an HTTP GET request structure:

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

Authorization: TC3-HMAC-SHA256 Credential=AKID*****/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=AKID*****/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=AKID*****/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
```

```
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=37ac2f4fde00b0ac9bd9eadeb459b1bbbe224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKID*****
```

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

Example of an HTTP POST request structure:

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

```
Host: cvm.tencentcloudapi.com
```

```
Content-Type: application/x-www-form-urlencoded
```

```
Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbec224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKID*****  
*****
```

Region List

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

Region	Value
North China (Beijing)	ap-beijing
Southwest China (Chengdu)	ap-chengdu
Southwest China (Chongqing)	ap-chongqing
South China (Guangzhou)	ap-guangzhou
Hong Kong/Macao/Taiwan (China) (Hong Kong (China))	ap-hongkong
Southeast Asia (Jakarta)	ap-jakarta
East China (Nanjing)	ap-nanjing
Northeast Asia (Seoul)	ap-seoul
East China (Shanghai)	ap-shanghai
East China (Shanghai Finance)	ap-shanghai-fsi
South China (Shenzhen Finance)	ap-shenzhen-fsi
Southeast Asia (Singapore)	ap-singapore
Northeast Asia (Tokyo)	ap-tokyo
Europe (Frankfurt)	eu-frankfurt

East US (Virginia)	na-ashburn
West US (Silicon Valley)	na-siliconvalley

Signature v3

Last updated : 2024-12-02 14:41:48

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

Applying for Security Credentials

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

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

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

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

Using the Resources for Developers

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

TC3-HMAC-SHA256 Signature Algorithm

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

TencentCloud API supports both GET and POST requests. For the GET method, only the Content-Type: application/x-www-form-urlencoded protocol format is supported. For the POST method, two protocol formats,

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

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

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

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

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

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

The signature calculation process is explained in detail below.

1. Concatenating the CanonicalRequest String

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

```
CanonicalRequest =
HTTPRequestMethod + '\n' +
CanonicalURI + '\n' +
```

```
CanonicalQueryString + '\n' +
CanonicalHeaders + '\n' +
SignedHeaders + '\n' +
HashedRequestPayload
```

Field Name	Explanation
HTTPRequestMethod	HTTP request method (GET or POST). This example uses <code>POST</code> .
CanonicalURI	URI parameter. Slash ("/") is used for API 3.0.
CanonicalQueryString	The query string in the URL of the originating HTTP request. This is always an empty string for POST requests, and is the string after the question mark (?) for GET requests. For example: <code>Limit=10&Offset=0</code>. 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.
CanonicalHeaders	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. Concatenation rules: - 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). The calculation result in this example is <code>content-type:application/json; charset=utf-8\nhost:cvm.tencentcloudapi.com\n</code>. 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.
SignedHeaders	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. Concatenation rules: - 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 (;). The value in this example is <code>content-type;host</code>
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 <code>TC3-HMAC-SHA256</code> .
RequestTimestamp	Request timestamp, i.e., the value of the common parameter <code>X-TC-Timestamp</code> in request header, which is the UNIX timestamp of the current time in seconds, such as <code>1551113065</code> in this example.
CredentialScope	Scope of the credential in the format of <code>Date/service/tc3_request</code> , including 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 <code>2018-05-25/cvm/tc3_request</code> .

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

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

Service	Value in the Service field in <code>Credential</code> , such as <code>cvm</code> in this example.
---------	---

2. Calculate the signature with the following pseudocode:

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

4. Concatenating the Authorization

The Authorization is concatenated as follows:

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

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

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

```
TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=ca282b0a56549857d53b2beb08b0c35871c892d42d09ae30b38d456e09ce291f
```

The following example shows a finished authorization header:

```
POST https://cvm.tencentcloudapi.com/  
Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=ca282b0a56549857d53b2beb08b0c35871c892d42d09ae30b38d456e09ce291f
```

```
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&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID*****&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGel%3D&Timestamp=1465185768&Version=2017-03-12.`

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

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

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

Java

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

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

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

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

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.tencentcloudapi.com";
        String region = "ap-guangzhou";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "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 = "AKID*****"
secret_key = "*****"

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

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

```

```

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

# ***** Step 2: Concatenate the string to sign *****
credential_scope = date + "/" + service + "/" + "tc3_request"
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).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 := "AKID*****"
    secretKey := "*****"
    host := "cvm.tencentcloudapi.com"
    algorithm := "TC3-HMAC-SHA256"
    service := "cvm"
    version := "2017-03-12"
    action := "DescribeInstances"
    region := "ap-guangzhou"
    //var timestamp int64 = time.Now().Unix()
    var timestamp int64 = 1551113065

    // step 1: build canonical request string
    httpRequestMethod := "POST"
    canonicalURI := "/"
    canonicalQueryString := ""
    canonicalHeaders := "content-type:application/json; charset=utf-8\n" + "host:" +
    host + "\n"
    signedHeaders := "content-type;host"
    payload := `{"Limit": 1, "Filters": [{"Values": ["unnamed"], "Name": "instance-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
$secretId = "AKID*****";
$secretKey = "*****";
$host = "cvm.tencentcloudapi.com";
$service = "cvm";
$version = "2017-03-12";
$action = "DescribeInstances";
$region = "ap-guangzhou";
// $timestamp = time();
$timestamp = 1551113065;
$algorithm = "TC3-HMAC-SHA256";

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

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

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

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

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

Ruby

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

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

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

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



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

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

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

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

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

NodeJS

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

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

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

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

function main() {

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

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

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

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

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

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

```

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

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

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

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

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

```

C++

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

using namespace std;

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

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

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

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

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

Last updated : 2024-12-02 14:41:49

Tencent Cloud API authenticates each access request, i.e. each request needs to include authentication information (Signature) in the common parameters to verify the identity of the requester.

The Signature is generated by the security credentials which include SecretId and SecretKey. If you don't have the security credentials yet, go to the [TencentCloud API Key](#) page to apply for them; otherwise, you cannot invoke the TencentCloud API.

1. Applying for Security Credentials

Before using the TencentCloud API for the first time, go to the [TencentCloud API Key](#) page to apply for security credentials.

Security credentials consist of SecretId and SecretKey:

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

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

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

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

2. Generating a Signature

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

Assume that the SecretId and SecretKey are:

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

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

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

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

2.1. Sorting Parameters

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

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

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

2.2. Concatenating a Request String

This step generates a request string.

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

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

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

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

2.3. Concatenating the Signature Original String

This step generates a signature original string.

The signature original string consists of the following parameters:

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

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

The concatenation result of the example is:

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

2.4. Generating a Signature String

This step generates a signature string.

First, use the HMAC-SHA1 algorithm to sign the **signature original string** obtained in the previous step, and then

encode the generated signature using Base64 to obtain the final signature.

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

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

The final signature is:

```
9FzTQAN1UZ489+BqCg1fNBQaCqw=
```

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

3. Encoding a Signature String

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

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

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

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

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

4. Signature Failure

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

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

5. Signature Demo

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

- [Python](#)
- [Java](#)
- [PHP](#)
- [Go](#)
- [NodeJS](#)
- [.NET](#)

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

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

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

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

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

Java

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

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

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

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

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



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

public static void main(String[] args) throws Exception {
    TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap 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 example: params.put ("Timestamp", System.currentTimeMillis() / 1000);
    params.put("Timestamp", 1465185768); // Common parameter
    params.put("SecretId", "AKID*****"); // Common parameter
    params.put("Action", "DescribeInstances"); // Common parameter
    params.put("Version", "2017-03-12"); // Common parameter
    params.put("Region", "ap-guangzhou"); // Common parameter
    params.put("Limit", 20); // Business parameter
    params.put("Offset", 0); // Business parameter
    params.put("InstanceIds.0", "ins-09dx96dg"); // Business parameter
    params.put("Signature", sign (getStringToSign (params), "*****", "HmacSHA1")); // Common parameter
    System.out.println(getUrl (params));
}
}
```

Python

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

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

import requests

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

```

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

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

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

```

Golang

```

package main

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

func main() {
    secretId := "AKID*****"
    secretKey := "*****"

```

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

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

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

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

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

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

PHP

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

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

ksort($param);

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

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

Ruby

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

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

method = 'GET'
endpoint = 'cvm.tencentcloudapi.com'
data = {
  'Action' => 'DescribeInstances',
  'InstanceIds.0' => 'ins-09dx96dg',
  'Limit' => 20,
```

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

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

DotNet

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

public class Application {
  public static string Sign(string signKey, string secret)
  {
    string signRet = string.Empty;
    using (HMACSHA1 mac = new HMACSHA1(Encoding.UTF8.GetBytes(signKey)))
    {
      byte[] hash = mac.ComputeHash(Encoding.UTF8.GetBytes(secret));
      signRet = Convert.ToBase64String(hash);
    }
    return signRet;
  }

  public static string MakeSignPlainText(SortedDictionary<string, string> requestParams, string requestMethod, string requestHost, string requestPath)
```

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

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

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

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

    param.Add("SecretId", SECRET_ID);
    param.Add("Region", region);
    SortedDictionary<string, string> headers = new SortedDictionary<string, string>(param, StringComparer.Ordinal);
    string sigInParam = MakeSignPlainText(headers, "GET", endpoint, "/");
    Console.WriteLine(sigInParam);
    string sigOutParam = Sign(SECRET_KEY, sigInParam);
}
```

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

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

NodeJS

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

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

function formatSignString(reqMethod, endpoint, path, strParam){
    let strSign = reqMethod + endpoint + path + "?" + strParam.slice(1);
    return strSign;
}

function sha1(secretKey, strsign){
    let signMethodMap = {'HmacSHA1': "sha1"};
    let hmac = crypto.createHmac(signMethodMap['HmacSHA1'], secretKey || "");
    return hmac.update(Buffer.from(strsign, 'utf8')).digest('base64')
}

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

```
strParam += ("&" + keys[k] + "=" + params[keys[k]]);
}
return strParam
}

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

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

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

strParam = sort_params(params)

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

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

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


Responses

Last updated : 2024-09-25 10:27:41

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.

Task APIs

CreateTaskFromTemplate

Last updated : 2024-12-02 14:41:57

1. API Description

Domain name for API request: `cfg.intl.tencentcloudapi.com`.

This API is used to create an experiment using a template.

A maximum of 10 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: <code>CreateTaskFromTemplate</code> .
Version	Yes	String	Common Params . The value used for this API: <code>2021-08-20</code> .
Region	No	String	Common Params . This parameter is not required.
TemplateId	Yes	Integer	Template ID retrieved from the template library
TaskConfig	Yes	TaskConfig	Experiment configuration parameters

3. Output Parameters

--	--	--

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

4. Example

Example1 Creating Experiments by Using Templates

This example shows you how to generate regular experiments for continuous use by using created templates.

Input Example

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

{
  "TemplateId": 689,
  "TaskConfig": {
    "TaskTitle": "Testing experiment, with instances associated and experiment name and custom action parameters of the first action in the first action group modified",
    "TaskGroupsConfig": [
      {
        "TaskGroupInstances": [
          "ins-xxxxxxx"
        ],
        "TaskGroupActionsConfig": [
          {
            "TaskGroupActionOrder": 1,
            "TaskGroupActionCustomConfiguration": "{\"timeout\":200,\"percentage\":80}"
          }
        ]
      }
    ]
  }
}
```

```
}  
}
```

Output Example

```
{  
  "Response": {  
    "RequestId": "f0aee8ac-2ed3-4a7f-a25b-f0d7d228dd30",  
    "TaskId": 50  
  }  
}
```

5. Developer Resources

SDK

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

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for Node.js](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
AuthFailure	CAM signature/authentication error.
AuthFailure.UnauthorizedOperation	Unauthorized CAM operation.

InternalError	Internal error.
InvalidParameter	Parameter error.
InvalidParameterValue	Parameter value error.
LimitExceeded	The quota limit is exceeded.
UnknownParameter	Unknown parameter.
UnsupportedOperation	The operation is not supported.

DeleteTask

Last updated : 2024-12-02 14:41:57

1. API Description

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

This API is used to delete a task.

A maximum of 10 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: DeleteTask.
Version	Yes	String	Common Params . The value used for this API: 2021-08-20.
Region	No	String	Common Params . This parameter is not required.
TaskId	Yes	Integer	Task ID

3. Output Parameters

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

4. Example

Example1 Deleting the Task

Input Example

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

{
  "TaskId": "222"
}
```

Output Example

```
{
  "Response": {
    "RequestId": "6549ed1a-911f-46dd-b6cd-2c02d5bd180f"
  }
}
```

5. Developer Resources

SDK

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

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for Node.js](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
AuthFailure	CAM signature/authentication error.
FailedOperation	Operation failed.
InternalError	Internal error.
InvalidParameter	Parameter error.
InvalidParameterValue	Parameter value error.

DescribeActionFieldConfigList

Last updated : 2024-12-02 14:41:56

1. API Description

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

This API is used to obtain the dynamic configuration parameters of the action field based on action ID, including action-specific parameters and common parameters.

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: DescribeActionFieldConfigList.
Version	Yes	String	Common Params . The value used for this API: 2021-08-20.
Region	No	String	Common Params . This parameter is not required.
ActionIds.N	Yes	Array of Integer	Action ID list
ObjectTypeId	Yes	Integer	Object type ID

3. Output Parameters

--	--	--

Parameter Name	Type	Description
Common	Array of ActionFieldConfigResult	List of general filed configuration parameters
Results	Array of ActionFieldConfigResult	List of action filed configuration parameters
ResourceOffline	Array of ResourceOffline	Information on the decommissioned resource Note: This field may return null, indicating that no valid values can be obtained.
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Example

Input Example

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

```
{
  "ObjectTypeId": "1",
  "ActionIds": [
    "1"
  ]
}
```

Output Example

```
{
  "Response": {
    "RequestId": "f3433a9a-e8fd-40b9-88e7-dd8b3f1a181f",
```

```
"Common": [
{
  "ActionId": 1,
  "ActionName": "Shut down",
  "ConfigDetail": [
    {
      "Type": "input",
      "Lable": "Action alias",
      "Field": "AliasTitle",
      "DefaultValue": "",
      "Config": "{}",
      "Required": 0,
      "Validate": "{}",
      "Visible": "{}"
    },
    {
      "Type": "number",
      "Lable": "Pre-wait time (s)",
      "Field": "PreTimeWait",
      "DefaultValue": "0",
      "Config": "{\"max\": 86400, \"min\": 0, \"tooltip\": \"Only for auto-advance mode\"}",
      "Required": 1,
      "Validate": "{}",
      "Visible": "{}"
    },
    {
      "Type": "number",
      "Lable": "Post-wait time (s)",
      "Field": "AfterTimeWait",
      "DefaultValue": "0",
      "Config": "{\"max\": 86400, \"min\": 0, \"tooltip\": \"Only for auto-advance mode\"}",
      "Required": 1,
      "Validate": "{}",
      "Visible": "{}"
    },
    {
      "Type": "number",
      "Lable": "Action timeout period (s)",
      "Field": "ActionTimeout",
      "DefaultValue": "1800",
      "Config": "{\"max\": 86400, \"min\": 0, \"tooltip\": \"The timeout period for the action\"}",
      "Required": 1,
      "Validate": "{}",
      "Visible": "{}"
    }
  ]
}
```

```
}  
]  
}  
],  
"Results": []  
}  
}
```

5. Developer Resources

SDK

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

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for Node.js](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
AuthFailure	CAM signature/authentication error.
InternalError	Internal error.
InvalidParameter	Parameter error.
InvalidParameterValue	Parameter value error.

MissingParameter	Parameters are missing
------------------	------------------------

DescribeActionLibraryList

Last updated : 2024-12-02 14:41:56

1. API Description

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

This API is used to obtain the action list of Chaotic Fault Generator.

A maximum of 100 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: DescribeActionLibraryList.
Version	Yes	String	Common Params . The value used for this API: 2021-08-20.
Region	No	String	Common Params . This parameter is not required.
Limit	Yes	Integer	0-100
Offset	Yes	Integer	Default value: 0
ObjectType	Yes	Integer	Object type ID
Filters.N	No	Array of ActionFilter	Keyword value {"action name": "a_title", "description": "a_desc", "action type": "a_type", "creation time": "a_create_time", "level-2 type": "a_resource_type"}
Attribute.N	No	Array of	Action type. 1: fault action; 2: recovery action.

		Integer	
ActionIds.N	No	Array of Integer	Filter item - action ID

3. Output Parameters

Parameter Name	Type	Description
Results	Array of ActionLibraryListResult	Query result list
Total	Integer	Number of matching records
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Querying the Data in the List of the Action Library

This example shows you how to query the data in the list of the action library.

Input Example

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

```
{
  "Limit": 10,
  "Offset": 0,
  "Filters": [
    {
      "Keyword": "a_type",
      "Values": [
```

```
"1"
]
},
"ObjectType": 1,
"Attribute": [
1
]
}
```

Output Example

```
{
  "Response": {
    "RequestId": "cZ6m0xYnnDreBNJm",
    "Results": [
      {
        "ActionName": "Shut down (testing)",
        "Desc": "Shut down the CVM.",
        "ActionType": "Platform",
        "CreateTime": "2023-07-04 11:20:43",
        "Creator": "System",
        "UpdateTime": "2023-07-04 11:20:43",
        "RiskDesc": "High risk",
        "ActionId": 1,
        "AttributeId": 1,
        "RelationActionId": 2,
        "ActionCommand": "Call the API interfaces of Tencent Cloud's corresponding products.",
        "ActionContent": "Call the cloud API StopInstances.",
        "ActionCommandType": 1,
        "ActionDetail": "<p>Call the cloud API <a href=\"https://cloud.tencent.com/document/product/213/15743\">StopInstances</a> to shut down CVMs.</p>",
        "ResourceType": "Server resources",
        "IsAllowed": true,
        "ActionBestCase": "https://cloud.tencent.com/document/product/1500/74357",
        "ObjectType": "CVM",
        "MetricIdList": [
          614,
          615
        ]
      },
      {
        "ActionName": "Start up",
        "Desc": "Start up the CVM.",
        "ActionType": "Platform",
```

```
"CreateTime": "2022-11-29 18:08:46",
"Creator": "System",
"UpdateTime": "2022-11-29 18:08:46",
"RiskDesc": "High risk",
"ActionId": 2,
"AttributeId": 2,
"RelationActionId": 1,
"ActionCommand": "Call the API interfaces of Tencent Cloud's corresponding products.",
"ActionContent": "Call the cloud API StartInstances.",
"ActionCommandType": 1,
"ActionDetail": "<p>Call the cloud API <a href=\"https://cloud.tencent.com/document/product/213/15735\">StartInstances</a> to start CVMs.</p>",
"ResourceType": "Server resources",
"IsAllowed": true,
"ActionBestCase": "",
"ObjectType": "CVM",
"MetricIdList": []
},
{
  "ActionName": "Restart",
  "Desc": "Restart",
  "ActionType": "Platform",
  "CreateTime": "2023-05-24 15:33:39",
  "Creator": "System",
  "UpdateTime": "2023-05-24 15:33:39",
  "RiskDesc": "High risk",
  "ActionId": 3,
  "AttributeId": 1,
  "RelationActionId": 0,
  "ActionCommand": "Call the API interfaces of Tencent Cloud's corresponding products.",
  "ActionContent": "Call the cloud API RebootInstances.",
  "ActionCommandType": 1,
  "ActionDetail": "<p>Call the cloud API <a href=\"https://cloud.tencent.com/document/product/213/15742\">RebootInstances</a> to restart CVMs.</p>",
  "ResourceType": "Server resources",
  "IsAllowed": true,
  "ActionBestCase": "https://tcloud4api.woa.com/document/product/1607/88863?!preview&!document=1",
  "ObjectType": "CVM",
  "MetricIdList": []
},
{
  "ActionName": "Check for high disk partition usage",
  "Desc": "Use the stress-ng stress testing tool to perform stress testing. Supported Linux distributions: Centos7.2 and later; CoreOS 1745.5.0 and later; Debian9.0
```

Page 77 of 167

```

\n then\n echo -e \"[\"`date +%Y-%m-%d %H:%M:%S\"`\"] \\c\"\n echo \"Fail to in
stall\"\n exit 1\n else\n echo -e \"[\"`date +%Y-%m-%d %H:%M:%S\"`\"] \\c\"\n e
cho \"stress-ng has already installed\"\n fi\nelse\n echo -e \"[\"`date +%Y-%m-
%d %H:%M:%S\"`\"] \\c\"\n echo \"stress-ng has already installed\"\nfi\nnecho -e
\"[\"`date +%Y-%m-%d %H:%M:%S\"`\"] \\c\"\n echo \"Starting to perform stress te
st.\"\nif [ $os_name == \"CoreOS\" ]\nthen\n docker run --rm alexeiled/stress-ng
-c 0 -l {{percentage}} --timeout {{timeout}}\nelse\n stress-ng -c 0 -l {{percenta
ge}} --timeout {{timeout}}\nfi\n\nif [[ !$? -eq 0 ]]\nthen\n echo -e \"[\"`date +
\"%Y-%m-%d %H:%M:%S\"`\"] \\c\"\n echo \"failed\"\n exit 1\nelse\n echo -e \"[\"`
date +%Y-%m-%d %H:%M:%S\"`\"] \\c\"\n echo \"completed\"\n exit 0\nfi",
"ActionCommandType": 0,
"ActionDetail": "<p>Use the tat channel to deliver the stress-ng command for stre
ss testing.</p>\n<p>Command content: stress-ng -c 0 -l {{percentage}} --timeout
{{timeout}}</p>\n<p><a href=\"https://cloud.tencent.com/document/api/1340/52676
\">tat official documentation</a></p>\n<p><a href=\"https://wiki.ubuntu.com/Kerne
l/Reference/stress-ng\">stress-ng official documentation</a></p>\n",
"ResourceType": "CPU resources",
"IsAllowed": true,
"ActionBestCase": "",
"ObjectType": "CVM",
"MetricIdList": []
},
{
"ActionName": "Check for high memory utilization",
"Desc": "Use the stress-ng stress testing tool to perform stress testing. Support
ed Linux distributions: Centos7.2 and later; CoreOS 1745.5.0 and later; Debian9.0
and later; Ubuntu 16.04.1 and later",
"ActionType": "Platform",
"CreateTime": "2022-11-29 18:08:46",
"Creator": "System",
"UpdateTime": "2022-11-29 18:08:46",
"RiskDesc": "High risk",
"ActionId": 7,
"AttributeId": 1,
"RelationActionId": 0,
"ActionCommand": "#!/bin/bash\n\nuser=$(whoami)\nif [ !$user == 'root' ]\nthen\n
sudo -i\nfi\n\nfunction command_exists(){\n if command -v $1 > /dev/null 2>&1; th
en\n return 1\n else\n return 0\n fi\n}\n\nnecho -e \"[\"`date +%Y-%m-%d %H:%M:%
S\"`\"] \\c\"\n\nos_desc=$(cat /etc/*release)\n# echo $os_desc\nndeclare -A os_d
ic\nos_dic=( [CentOS]=\"yum install stress-ng -y\" \\\n [CoreOS]=\"docker pull ale
xeiled/stress-ng\" \\\n [Debian]=\"apt-get install -y stress-ng\" \\\n [Ubuntu]=
\"apt-get install -y stress-ng\" )\n\nos_name='N/A'\n\nfor key in $(echo ${!os_dic
[*]})\ndo\n if [[ $os_desc =~ $key ]]\n then\n os_name=$key\n echo $key\n fi\ndon
e\n\nif [ $os_name == 'N/A' ]\nthen\n echo \"Unsupported Linux distributions\"\n
exit 1\nfi\n\n# os_name=$(cat /etc/*release | awk 'NR==1{print($1)}')\n\nnecho -e
\"[\"`date +%Y-%m-%d %H:%M:%S\"`\"] \\c\"\nnecho \"installing stress-ng...\"\nco
mmand_exists stress-ng\n\nif [[ $? -eq 0 ]]\nthen\n ${os_dic[$os_name]}\n echo -e

```

```

\["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\`n echo \Checking the installation st
atus...\`n command_exists stress-ng\`n if [[ $? -eq 0 && $os_name != 'CoreOS' ]]
\`n then\`n echo -e \["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\`n echo \Fail to in
stall\`n exit 1\`n else\`n echo -e \["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\`n e
cho \stress-ng has already installed\`n fi\`n else\`n echo -e \["\`date +"%Y-%m-
%d %H:%M:%S%\`"] \c\`n echo \stress-ng has already installed\`n fi\`n\`n echo -e
\["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\`n echo \Starting to perform stress te
st\`n\`n if [ $os_name == \CoreOS\` ]\`n then\`n docker run --rm alexeiled/stress-ng -
-vm-bytes $(awk '/MemAvailable/{printf "%d\\n\\n", $2 * 0.01*{percentage}};' < /
proc/meminfo)k --vm-keep -m 1 -t {{timeout}}\`n else\`n available_mem=$(awk '/^MemAv
ailable:/{printf "%d\\n\\n", $2;}' < /proc/meminfo)\`n free_mem=$(awk '/^MemFree:/
{printf "%d\\n\\n", $2;}' < /proc/meminfo)\`n buffer=$(awk '/^Buffers:/{printf "%d
\\n\\n", $2;}' < /proc/meminfo)\`n cache=$(awk '/^Cached:/{printf "%d\\n\\n", $2;}' <
/proc/meminfo)\`n if [ -n \ $available_mem\` ]\`n then\`n stress-ng --vm-bytes [$av
ailable_mem*{percentage}}/100]k --vm-keep -m 1 -t {{timeout}}\`n else\`n stress-ng
--vm-bytes [($free_mem+$buffer+$cache)*{percentage}}/100]k --vm-keep -m 1 -t
{{timeout}}\`n fi\`n fi\`n\`n if [[ ! $? -eq 0 ]]\`n then\`n echo -e \["\`date +"%Y-%m-%d
%H:%M:%S%\`"] \c\`n echo \Failed\`n exit 1\`n else\`n echo -e \["\`date +"%Y-
%m-%d %H:%M:%S%\`"] \c\`n echo \Completed\`n exit 0\`n fi",
"ActionContent": "#!/bin/bash\`n\`n user=$(whoami)\`n\`n if [ ! $user == 'root' ]\`n then\`n
sudo -i\`n fi\`n\`n function command_exists(){\`n if command -v $1 > /dev/null 2>&1; th
en\`n return 1\`n else\`n return 0\`n fi\`n}\`n\`n\`n echo -e \["\`date +"%Y-%m-%d %H:%M:%
S%\`"] \c\`n\`n nos_desc=$(cat /etc/*release)\`n\`n # echo $os_desc\`n\`n declare -A os_d
ic\`n\`n nos_dic=([CentOS]=\yum install stress-ng -y\`n \`\`n [CoreOS]=\docker pull ale
xeiled/stress-ng\`n \`\`n [Debian]=\apt-get install -y stress-ng\`n \`\`n [Ubuntu]=
\apt-get install -y stress-ng\`n)\`n\`n\`n nos_name='N/A'\`n\`n for key in $(echo ${!os_dic
[*]})\`n do\`n if [[ $os_desc =~ $key ]]\`n then\`n os_name=$key\`n echo $key\`n fi\`n don
e\`n\`n\`n if [ $os_name == 'N/A' ]\`n then\`n echo \Unsupported Linux distributions\`n
exit 1\`n fi\`n\`n\`n # os_name=$(cat /etc/*release | awk 'NR==1{print($1)}')\`n\`n echo -e
\["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\`n\`n echo \installing stress-ng...\`n\`n co
mmand_exists stress-ng\`n\`n if [[ $? -eq 0 ]]\`n then\`n ${os_dic[$os_name]}\`n echo -e
\["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\`n\`n echo \Checking the installation st
atus...\`n command_exists stress-ng\`n if [[ $? -eq 0 && $os_name != 'CoreOS' ]]
\`n then\`n echo -e \["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\`n echo \Fail to in
stall\`n exit 1\`n else\`n echo -e \["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\`n e
cho \stress-ng has already installed\`n fi\`n else\`n echo -e \["\`date +"%Y-%m-
%d %H:%M:%S%\`"] \c\`n echo \stress-ng has already installed\`n fi\`n\`n echo -e
\["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\`n echo \Starting to perform stress te
st\`n\`n if [ $os_name == \CoreOS\` ]\`n then\`n docker run --rm alexeiled/stress-ng -
-vm-bytes $(awk '/MemAvailable/{printf "%d\\n\\n", $2 * 0.01*{percentage}};' < /
proc/meminfo)k --vm-keep -m 1 -t {{timeout}}\`n else\`n available_mem=$(awk '/^MemAv
ailable:/{printf "%d\\n\\n", $2;}' < /proc/meminfo)\`n free_mem=$(awk '/^MemFree:/
{printf "%d\\n\\n", $2;}' < /proc/meminfo)\`n buffer=$(awk '/^Buffers:/{printf "%d
\\n\\n", $2;}' < /proc/meminfo)\`n cache=$(awk '/^Cached:/{printf "%d\\n\\n", $2;}' <
/proc/meminfo)\`n if [ -n \ $available_mem\` ]\`n then\`n stress-ng --vm-bytes [$av
ailable_mem*{percentage}}/100]k --vm-keep -m 1 -t {{timeout}}\`n else\`n stress-ng
--vm-bytes [($free_mem+$buffer+$cache)*{percentage}}/100]k --vm-keep -m 1 -t

```

```

{{timeout}}\n fi\nfi\n\nif [[ ! $? -eq 0 ]] \nthen\n echo -e \"[\\\"`date +%Y-%m-%d %H:%M:%S\\\"`] \\c\\\" \n echo \"Failed\\\" \n exit 1\nelse\n echo -e \"[\\\"`date +%Y-%m-%d %H:%M:%S\\\"`] \\c\\\" \n echo \"Completed\\\" \n exit 0\nfi\",
"ActionCommandType": 0,
"ActionDetail": "<p> Use the tat channel to issue the stress-ng command for stress testing</p> \n<p> Command content: stress-ng --vm-bytes $(awk '/MemAvailable/{printf \"%d\\n\\n\", $2 * 0.01*{{percentage}};}' < /proc/meminfo)k --vm-keep -m 1 -t {{timeout}}</p>\n<p><a href=\\\"https://cloud.tencent.com/document/api/1340/52676\\\">tat official documentation</a></p>\n<p><a href=\\\"https://wiki.ubuntu.com/Kernel/Reference/stress-ng\\\">stress-ng official documentation</a></p>\",
"ResourceType": "Memory resources",
"IsAllowed": true,
"ActionBestCase": "",
"ObjectType": "CVM",
"MetricIdList": []
},
{
"ActionName": "Check for high disk partition utilization",
"Desc": "Use the stress-ng stress testing tool to perform stress testing. Supported Linux distributions: Centos7.2 and later; CoreOS 1745.5.0 and later; Debian9.0 and later; Ubuntu 16.04.1 and later",
"ActionType": "Platform",
"CreateTime": "2022-11-29 18:08:46",
"Creator": "System",
"UpdateTime": "2022-11-29 18:08:46",
"RiskDesc": "High risk",
"ActionId": 8,
"AttributeId": 1,
"RelationActionId": 0,
"ActionCommand": "#!/bin/bash\n\nuser=$(whoami)\nif [[ ! $user == 'root' ]] \nthen\n sudo -i\nfi\n\nfunction command_exists()\n{\n if command -v $1 > /dev/null 2>&1; then\n\n return 1\n else\n\n return 0\n fi\n}\n\nnecho -e \"[\\\"`date +%Y-%m-%d %H:%M:%S\\\"`] \\c\\\" \n\nos_desc=$(cat /etc/*release)\n# echo $os_desc\n\ndeclare -A os_dic\nos_dic=( [CentOS]=\\\"yum install stress-ng -y\\\" \n\n [CoreOS]=\\\"docker pull alexeiled/stress-ng\\\" \n\n [Debian]=\\\"apt-get install -y stress-ng\\\" \n\n [Ubuntu]=\\\"apt-get install -y stress-ng\\\" )\n\nos_name='N/A'\n\nfor key in $(echo ${!os_dic[*]})\ndo\n if [[ $os_desc =~ $key ]]\n then\n os_name=$key\n echo $key\n fi\ndon\n\nif [[ $os_name == 'N/A' ]]\n then\n echo \"Unsupported Linux distributions\\\" \n exit 1\nfi\n\nos_name=$(cat /etc/*release | awk 'NR==1{print($1)}')\n\nnecho -e \"[\\\"`date +%Y-%m-%d %H:%M:%S\\\"`] \\c\\\" \n\nnecho \"installing stress-ng...\\\" \n\ncommand_exists stress-ng\n\nif [[ $? -eq 0 ]]\n then\n ${os_dic[$os_name]}\n echo -e \"[\\\"`date +%Y-%m-%d %H:%M:%S\\\"`] \\c\\\" \n echo \"Checking the installation status...\\\" \n\ncommand_exists stress-ng\n if [[ $? -eq 0 && $os_name != 'CoreOS' ]]\n then\n echo -e \"[\\\"`date +%Y-%m-%d %H:%M:%S\\\"`] \\c\\\" \n echo \"Fail to install\\\" \n exit 1\n else\n echo -e \"[\\\"`date +%Y-%m-%d %H:%M:%S\\\"`] \\c\\\" \n echo \"stress-ng has already installed\\\" \n fi\nelse\n echo -e \"[\\\"`date +%Y-%m-%d %H:%M:%S\\\"`] \\c\\\" \n echo \"stress-ng has already installed\\\" \nfi\n\nnecho -e

```



```

\["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\necho "Starting to perform stress test."
\nif [ $os_name == "CoreOS" ]\nthen\n cd {{dir}} && docker run --rm alexeiled/stress-ng --iomix 1 --iomix-bytes `df -k {{dir}} | awk 'NR==2{printf("%d\\n", ($3+$4)*0.01*{{percentage}}-$3)}'`k -t {{timeout}}\nelse\n cd {{dir}} && stress-ng --iomix 1 --iomix-bytes `df -k {{dir}} | awk 'NR==2{printf("%d\\n", ($3+$4)*0.01*{{percentage}}-$3)}'`k -t {{timeout}}\nfi\n\nif [[ ! $? -eq 0 ]]\nthen\n echo -e \["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\necho "Failed"\n exit 1\nelse\n echo -e \["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\necho "Completed"\n exit 0\nfi",
"ActionContent": "#!/bin/bash\n\nuser=$(whoami)\nif [ ! $user == 'root' ]\nthen\n sudo -i\nfi\n\nfunction command_exists()\n{\n if command -v $1 > /dev/null 2>&1; then\n return 1\n else\n return 0\n fi\n}\n\nnecho -e \["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\necho $os_desc\nndeclare -A os_dic\nos_dic=( [CentOS]="yum install stress-ng -y" \\\n [CoreOS]="docker pull alexeiled/stress-ng" \\\n [Debian]="apt-get install -y stress-ng" \\\n [Ubuntu]="apt-get install -y stress-ng" )\n\nos_name='N/A'\n\nfor key in $(echo ${!os_dic[*]})\ndo\n if [[ $os_desc =~ $key ]]\n then\n os_name=$key\n echo $key\n fi\ndon\n\nif [ $os_name == 'N/A' ]\nthen\n echo "Unsupported Linux distributions"\n exit 1\nfi\n\nos_name=$(cat /etc/*release | awk 'NR==1{print($1)}')\n\nnecho -e \["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\necho "installing stress-ng..." \nc\ncommand_exists stress-ng\n\nif [[ $? -eq 0 ]]\nthen\n ${os_dic[$os_name]}\n echo -e \["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\necho "Checking the installation status..." \n\ncommand_exists stress-ng\n if [[ $? -eq 0 && $os_name != 'CoreOS' ]]\n then\n echo -e \["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\necho "Fail to install"\n exit 1\n else\n echo -e \["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\necho "stress-ng has already installed"\n fi\n\nelse\n echo -e \["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\necho "stress-ng has already installed"\nfi\n\nnecho -e \["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\necho "Starting to perform stress test." \n\nif [ $os_name == "CoreOS" ]\nthen\n cd {{dir}} && docker run --rm alexeiled/stress-ng --iomix 1 --iomix-bytes `df -k {{dir}} | awk 'NR==2{printf("%d\\n", ($3+$4)*0.01*{{percentage}}-$3)}'`k -t {{timeout}}\n\nelse\n cd {{dir}} && stress-ng --iomix 1 --iomix-bytes `df -k {{dir}} | awk 'NR==2{printf("%d\\n", ($3+$4)*0.01*{{percentage}}-$3)}'`k -t {{timeout}}\nfi\n\nif [[ ! $? -eq 0 ]]\nthen\n echo -e \["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\necho "Failed"\n exit 1\nelse\n echo -e \["\`date +"%Y-%m-%d %H:%M:%S%\`"] \c\necho "Completed"\n exit 0\nfi",
"ActionCommandType": 0,
"ActionDetail": "<p>Use the tat channel to deliver the stress-ng command for stress testing</p>\n<p>Command content: stress-ng --iomix 1 --iomix-bytes `df -k {{dir}} | awk 'NR==2{printf("%d\\n", ($3+$4)*0.01*{{percentage}}-$3)}'`k -t {{timeout}}</p>\n<p><a href=\"https://cloud.tencent.com/document/api/1340/52676\">tat of ficial documentation</a></p>\n<p><a href=\"https://wiki.ubuntu.com/Kernel/Reference/stress-ng\">stress-ng official documentation</a></p>\n",
"ResourceType": "Disk resources",
"IsAllowed": true,
"ActionBestCase": "",
"ObjectType": "CVM",

```

```
"MetricIdList": []
},
{
  "ActionName": "Kernel Fault",
  "Desc": "Trigger the restart of instances",
  "ActionType": "Platform",
  "CreateTime": "2022-11-29 18:08:46",
  "Creator": "System",
  "UpdateTime": "2022-11-29 18:08:46",
  "RiskDesc": "High risk",
  "ActionId": 9,
  "AttributeId": 1,
  "RelationActionId": 24,
  "ActionCommand": "echo -e \"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\" && echo 'inject kernel error success!' && echo -e \"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\" && sleep 5 && echo -e \"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\" && echo c > /proc/sysrq-trigger &",
  "ActionContent": "echo -e \"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\" && echo 'inject kernel error success!' && echo -e \"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\" && sleep 5 && echo -e \"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\" && echo c > /proc/sysrq-trigger &",
  "ActionCommandType": 0,
  "ActionDetail": "<p>Use the tat channel to deliver kernel fault commands</p>\n<p>Command content: echo c > /proc/sysrq-trigger</p>\n<p><a href=\"https://cloud.tencent.com/document/api/1340/52676\">tat official documentation</a></p>",
  "ResourceType": "CPU resources",
  "IsAllowed": true,
  "ActionBestCase": "",
  "ObjectType": "CVM",
  "MetricIdList": []
},
{
  "ActionName": "Check disk IO load",
  "Desc": "Use the stress-ng stress testing tool to perform stress testing. Supported Linux distributions: Centos7.2 and later; CoreOS 1745.5.0 and later; Debian9.0 and later; Ubuntu 16.04.1 and later",
  "ActionType": "Platform",
  "CreateTime": "2022-11-29 18:08:46",
  "Creator": "System",
  "UpdateTime": "2022-11-29 18:08:46",
  "RiskDesc": "High risk",
  "ActionId": 10,
  "AttributeId": 1,
  "RelationActionId": 0,
  "ActionCommand": "#!/bin/bash\n\nuser=$(whoami)\nif [ ! $user == 'root' ]\nthen\n  sudo -i\nfi\n\nfunction command_exists()\n{\n  if command -v $1 > /dev/null 2>&1; then\n    return 1\n  else\n    return 0\n  fi\n}\n\nnecho -e \"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\" && echo 'inject kernel error success!' && echo -e \"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\" && sleep 5 && echo -e \"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\" && echo c > /proc/sysrq-trigger &"
```

```

S\"`\" ] \\c\"\\n\\nos_desc=$(cat /etc/*release)\\n# echo $os_desc\\n\\ndeclare -A os_d
ic\\nos_dic=([CentOS]=\"yum install stress-ng -y\" \\n [CoreOS]=\"docker pull ale
xeiled/stress-ng\" \\n [Debian]=\"apt-get install -y stress-ng\" \\n [Ubuntu]=
\"apt-get install -y stress-ng\")\\n\\nos_name='N/A'\\n\\nfor key in $(echo ${!os_dic
[*]})\\ndo\\n if [[ $os_desc =~ $key ]]\\n then\\n os_name=$key\\n echo $key\\n fi\\ndon
e\\n\\nif [ $os_name == 'N/A' ]\\nthen\\n echo \"Unsupported Linux distributions\"\\n
exit 1\\nfi\\n\\n# os_name=$(cat /etc/*release | awk 'NR==1{print($1)}')\\n\\necho -e
\"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\n\\necho \"installing stress-ng...\"\\nco
mmand_exists stress-ng\\n\\nif [[ $? -eq 0 ]]\\nthen\\n ${os_dic[$os_name]}\\n echo -e
\"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\n echo \"Checking the installation st
atus...\"\\n command_exists stress-ng\\n if [[ $? -eq 0 && $os_name != 'CoreOS' ]]
\\n then\\n echo -e \"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\n echo \"Fail to in
stall\"\\n exit 1\\n else\\n echo -e \"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\n e
cho \"stress-ng has already installed\"\\n fi\\nelse\\n echo -e \"[\"`date +\"%Y-%m-
%d %H:%M:%S\"`\" ] \\c\"\\n echo \"stress-ng has already installed\"\\nfi\\n\\necho -e
\"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\n\\necho \"Starting to perform stress te
st.\"\\nif [ $os_name == \"CoreOS\" ]\\nthen\\n docker run --rm alexeiled/stress-ng
--vm-bytes ${bytes_num}${bytes_unit} --hdd ${io_process_num} --vm-keep -t ${ti
meout}\\nelse\\n stress-ng --vm-bytes ${bytes_num}${bytes_unit} --hdd ${io_proce
ss_num} --vm-keep -t ${timeout}\\nfi\\n\\nif [[ ! $? -eq 0 ]]\\nthen\\n echo -e \"
[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\n echo \"Failed\"\\n exit 1\\nelse\\n echo
-e \"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\n echo \"Completed\"\\n exit 0\\nfi
'

\"ActionContent\": \"#!/bin/bash\\n\\nuser=$(whoami)\\nif [ ! $user == 'root' ]\\nthen\\n
sudo -i\\nfi\\n\\nfunction command_exists(){\\n if command -v $1 > /dev/null 2>&1; th
en\\n return 1\\n else\\n return 0\\n fi\\n}\\n\\n\\necho -e \"[\"`date +\"%Y-%m-%d %H:%M:%
S\"`\" ] \\c\"\\n\\nos_desc=$(cat /etc/*release)\\n# echo $os_desc\\n\\ndeclare -A os_d
ic\\nos_dic=([CentOS]=\"yum install stress-ng -y\" \\n [CoreOS]=\"docker pull ale
xeiled/stress-ng\" \\n [Debian]=\"apt-get install -y stress-ng\" \\n [Ubuntu]=
\"apt-get install -y stress-ng\")\\n\\nos_name='N/A'\\n\\nfor key in $(echo ${!os_dic
[*]})\\ndo\\n if [[ $os_desc =~ $key ]]\\n then\\n os_name=$key\\n echo $key\\n fi\\ndon
e\\n\\nif [ $os_name == 'N/A' ]\\nthen\\n echo \"Unsupported Linux distributions\"\\n
exit 1\\nfi\\n\\n# os_name=$(cat /etc/*release | awk 'NR==1{print($1)}')\\n\\necho -e
\"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\n\\necho \"installing stress-ng...\"\\nco
mmand_exists stress-ng\\n\\nif [[ $? -eq 0 ]]\\nthen\\n ${os_dic[$os_name]}\\n echo -e
\"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\n echo \"Checking the installation st
atus...\"\\n command_exists stress-ng\\n if [[ $? -eq 0 && $os_name != 'CoreOS' ]]
\\n then\\n echo -e \"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\n echo \"Fail to in
stall\"\\n exit 1\\n else\\n echo -e \"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\n e
cho \"stress-ng has already installed\"\\n fi\\nelse\\n echo -e \"[\"`date +\"%Y-%m-
%d %H:%M:%S\"`\" ] \\c\"\\n echo \"stress-ng has already installed\"\\nfi\\n\\necho -e
\"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\n\\necho \"Starting to perform stress te
st.\"\\nif [ $os_name == \"CoreOS\" ]\\nthen\\n docker run --rm alexeiled/stress-ng
--vm-bytes ${bytes_num}${bytes_unit} --hdd ${io_process_num} --vm-keep -t ${ti
meout}\\nelse\\n stress-ng --vm-bytes ${bytes_num}${bytes_unit} --hdd ${io_proce
ss_num} --vm-keep -t ${timeout}\\nfi\\n\\nif [[ ! $? -eq 0 ]]\\nthen\\n echo -e \"
[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\n echo \"Failed\"\\n exit 1\\nelse\\n echo

```

```
-e \"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\n echo \"Completed\"\\n exit 0\\nfi\"
,
\"ActionCommandType\": 0,
\"ActionDetail\": \"<p>Use the tat channel to deliver the stress-ng command for stre
ss testing.</p>\\n<p>Command content: stress-ng --vm-bytes {{bytes_num}}{{bytes_un
it}} --hdd {{io_process_num}} --vm-keep -t {{timeout}}</p>\\n<p><a href=\\\"https://
cloud.tencent.com/document/api/1340/52676\\\">tat official documentation</a></p>\",
\"ResourceType\": \"IO resources\",
\"IsAllowed\": true,
\"ActionBestCase\": \"\",
\"ObjectType\": \"CVM\",
\"MetricIdList\": []
},
{
\"ActionName\": \"Check for OOM\",
\"Desc\": \"Use the stress-ng stress testing tool to perform stress testing. Support
ed Linux distributions: Centos7.2 and later; CoreOS 1745.5.0 and later; Debian9.0
and later; Ubuntu 16.04.1 and later\",
\"ActionType\": \"Platform\",
\"CreateTime\": \"2023-07-04 11:20:43\",
\"Creator\": \"System\",
\"UpdateTime\": \"2023-07-04 11:20:43\",
\"RiskDesc\": \"High risk\",
\"ActionId\": 11,
\"AttributeId\": 1,
\"RelationActionId\": 0,
\"ActionCommand\": \"#!/bin/bash\\n\\nuser=$(whoami)\\nif [ !$user == 'root' ]\\nthen\\n
sudo -i\\nfi\\n\\nfunction command_exists(){\\n if command -v $1 > /dev/null 2>&1; th
en\\n return 1\\n else\\n return 0\\n fi\\n}\\n\\nnecho -e \"[\"`date +\"%Y-%m-%d %H:%M:%
S\"`\" ] \\c\"\\n\\nos_desc=$(cat /etc/*release)\\n# echo $os_desc\\n\\ndeclare -A os_d
ic\\nos_dic=( [CentOS]=\"yum install stress-ng -y\" \\n [CoreOS]=\"docker pull ale
xeiled/stress-ng\" \\n [Debian]=\"apt-get install -y stress-ng\" \\n [Ubuntu]=
\"apt-get install -y stress-ng\" )\\n\\nos_name='N/A'\\n\\nfor key in $(echo ${!os_dic
[*]})\\ndo\\n if [[ $os_desc =~ $key ]]\\n then\\n os_name=$key\\n echo $key\\n fi\\ndon
e\\n\\nif [ $os_name == 'N/A' ]\\nthen\\n echo \"Unsupported Linux distributions\"\\n
exit 1\\nfi\\n\\n# os_name=$(cat /etc/*release | awk 'NR==1{print($1)}')\\n\\nnecho -e
\"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\nnecho \"installing stress-ng...\"\\nco
mmand_exists stress-ng\\n\\nif [[ $? -eq 0 ]]\\nthen\\n ${os_dic[$os_name]}\\n echo -e
\"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\n echo \"Checking the installation st
atus...\"\\n command_exists stress-ng\\n if [[ $? -eq 0 && $os_name != 'CoreOS' ]]
\\n then\\n echo -e \"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\n echo \"Fail to in
stall\"\\n exit 1\\n else\\n echo -e \"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\n e
cho \"stress-ng has already installed\"\\n fi\\nelse\\n echo -e \"[\"`date +\"%Y-%m-
%d %H:%M:%S\"`\" ] \\c\"\\n echo \"stress-ng has already installed\"\\nfi\\n\\nnecho -e
\"[\"`date +\"%Y-%m-%d %H:%M:%S\"`\" ] \\c\"\\nnecho \"Starting to perform stress te
st.\"\\nif [ $os_name == \"CoreOS\" ]\\nthen\\n docker run --rm alexeiled/stress-ng
--bigheap 10 --bigheap-growth 4K -t {{timeout}}\\nelse\\n stress-ng --bigheap 10 --
```

```
"bigheap-growth 4K -t {{timeout}}\nfi\nnecho \"Completed\"\n\n",
>ActionContent": "#!/bin/bash\n\nuser=$(whoami)\nnif [ !$user == 'root' ]\nthen\nsudo -i\nfi\n\nfunction command_exists()\n{\nif command -v $1 > /dev/null 2>&1; th\nen\nreturn 1\nelse\nreturn 0\nfi\n}\n\nnecho -e \"[\"`date +%Y-%m-%d %H:%M:%S``\"] \\c\\\"\nos_desc=$(cat /etc/*release)\n# echo $os_desc\nndeclare -A os_dic\nos_dic=(CentOS=\"yum install stress-ng -y\" \\\n Debian=\"apt-get install -y stress-ng\" \\\n Ubuntu=\"apt-get install -y stress-ng\")\n\nnos_name='N/A'\n\nfor key in $(echo ${!os_dic[*]})\ndo\nif [[ $os_desc =~ $key ]]\nthen\nos_name=$key\n echo $key\nfi\ndon\n\nif [ $os_name == 'N/A' ]\nthen\n echo \"Unsupported Linux distributions\"\n\nexit 1\nfi\n\n# os_name=$(cat /etc/*release | awk 'NR==1{print($1)}')\n\nnecho -e \"[\"`date +%Y-%m-%d %H:%M:%S``\"] \\c\\\"\nnecho \"installing stress-ng...\"\ncommand_exists stress-ng\n\nif [[ $? -eq 0 ]]\nthen\n ${os_dic[$os_name]}\n echo -e \"[\"`date +%Y-%m-%d %H:%M:%S``\"] \\c\\\"\n echo \"Checking the installation status...\"\n\ncommand_exists stress-ng\nif [[ $? -eq 0 && $os_name != 'CoreOS' ]]\nthen\n echo -e \"[\"`date +%Y-%m-%d %H:%M:%S``\"] \\c\\\"\n echo \"Fail to install\"\n\nexit 1\nelse\n echo -e \"[\"`date +%Y-%m-%d %H:%M:%S``\"] \\c\\\"\n echo \"stress-ng has already installed\"\n\nfi\n\nelse\n echo -e \"[\"`date +%Y-%m-%d %H:%M:%S``\"] \\c\\\"\n echo \"stress-ng has already installed\"\n\nfi\n\nnecho -e \"[\"`date +%Y-%m-%d %H:%M:%S``\"] \\c\\\"\nnecho \"Starting to perform stress test.\"\n\nif [ $os_name == \"CoreOS\" ]\nthen\n docker run --rm alexeiled/stress-ng --bigheap 10 --bigheap-growth 4K -t {{timeout}}\n\nelse\n stress-ng --bigheap 10 --bigheap-growth 4K -t {{timeout}}\n\nfi\n\nnecho \"Completed\"\n\n\n",
>ActionCommandType": 0,
>ActionDetail": "<p>Use the tat channel to deliver the stress-ng command for stress testing</p>\n<p>Command content: stress-ng --bigheap 10 --bigheap-growth 4K -t {{timeout}}</p>\n<p><a href=\\\"https://cloud.tencent.com/document/api/1340/52676\\\">tat official documentation</a></p>",
@ResourceType": "Memory resources",
:IsAllowed": true,
>ActionBestCase": "https://cloud.tencent.com/document/product/1500/81504",
:ObjectType": "CVM",
>MetricIdList": []
},
{
>ActionName": "Empty operation",
>Desc": "Empty operation. It is used to test the process with no actual injection operation.",
>ActionType": "Platform",
>CreateTime": "2022-11-29 18:08:46",
>Creator": "System",
>UpdateTime": "2022-11-29 18:08:46",
>RiskDesc": "Low risk",
>ActionId": 12,
>AttributeId": 1,
>RelationActionId": 13,
>ActionCommand": "Call the API interfaces of Tencent Cloud's corresponding produc
```

```
ts.",
"ActionContent": "Empty operation",
"ActionCommandType": 1,
"ActionDetail": null,
"ResourceType": "Other",
"IsAllowed": true,
"ActionBestCase": "",
"ObjectType": "CVM",
"MetricIdList": []
}
],
"Total": 45
}
}
```

5. Developer Resources

SDK

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

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for Node.js](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description

InternalError	Internal error.
InvalidParameter	Parameter error.
InvalidParameterValue	Parameter value error.
MissingParameter	Parameters are missing
UnsupportedOperation	The operation is not supported.

DescribeObjectTypeInfoList

Last updated : 2024-12-02 14:41:55

1. API Description

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

This API is used to query the object type list.

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

We recommend you to use API Explorer

[Try it](#)

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params . The value used for this API: DescribeObjectTypeInfoList.
Version	Yes	String	Common Params . The value used for this API: 2021-08-20.
Region	No	String	Common Params . This parameter is not required.
SupportType	No	Integer	Supported object 0: all platform products 1: objects connected to the platform 2: some objects supported by the application

3. Output Parameters

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

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

4. Example

Example1 Request object type list

Input Example

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

```
{}
```

Output Example

```
{
  "Response": {
    "ObjectTypeList": [
      {
        "ObjectTypeId": 0,
        "ObjectTypeTitle": "abc",
        "ObjectTypeLevelOne": "abc",
        "ObjectTypeParams": {
          "Key": "abc",
          "Fields": [
            {
              "Key": "abc",
              "Header": "abc",
              "Transfer": "abc",
              "JsonParse": "abc"
            }
          ]
        }
      }
    ]
  }
}
```

```
},
"ObjectTypeJsonParse": {
  "NameSpace": "abc",
  "WorkloadName": "abc"
}
},
],
"RequestId": "abc"
}
```

5. Developer Resources

SDK

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

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for Node.js](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

DescribeTask

Last updated : 2024-12-02 14:41:55

1. API Description

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

This API is used to query a task.

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: DescribeTask.
Version	Yes	String	Common Params . The value used for this API: 2021-08-20.
Region	No	String	Common Params . This parameter is not required.
TaskId	Yes	Integer	Task ID

3. Output Parameters

Parameter Name	Type	Description
Task	Task	Task information
ReportInfo	TaskReportInfo	Experiment report information corresponding to the task. The value null

		indicates that no report is exported. Note: This field may return null, indicating that no valid values can be obtained.
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Querying Experiments

Input Example

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

```
{
  "TaskId": 0
}
```

Output Example

```
{
  "Response": {
    "ReportInfo": null,
    "RequestId": "68da5328-ff1f-4320-92c0-a67e65c8a298",
    "Task": {
      "AlarmPolicy": [],
      "ApmServiceList": [
        {
          "InstanceId": "test-ins",
          "RegionId": 1,
          "ServiceNameList": [
            "test-service"
          ]
        }
      ],
      "ApplicationId": ""
    }
  }
}
```

```
"ApplicationName": "",
"Tags": [],
"TaskCreateTime": "2023-10-09 10:55:18",
"TaskDescription": "Testing",
"TaskEndTime": null,
"TaskExpect": null,
"TaskGroups": [
{
"ObjectTypeId": 1,
"TaskGroupActions": [
{
"ActionApiType": 1,
"ActionAttribute": 1,
"ActionId": 1,
"ActionRisk": "High risk",
"ActionTitle": "Shutdown (testing)",
"ActionType": "Platform",
"IsExecuteRedo": false,
"TaskGroupActionCreateTime": "2023-10-09 10:55:18",
"TaskGroupActionCustomConfiguration": "{}",
"TaskGroupActionExecuteId": null,
"TaskGroupActionExecuteTime": null,
"TaskGroupActionGeneralConfiguration": "{\"AliasTitle\": \"\", \"PreTimeWait\": 0, \"ActionTimeout\": 1800, \"AfterTimeWait\": 0}",
"TaskGroupActionId": 13785,
"TaskGroupActionOrder": 1,
"TaskGroupActionRandomId": 457181,
"TaskGroupActionRecoverId": 355514,
"TaskGroupActionStatus": 2001,
"TaskGroupActionStatusType": 0,
"TaskGroupActionUpdateTime": "2023-10-09 10:55:18",
"TaskGroupInstances": []
},
{
"ActionApiType": 1,
"ActionAttribute": 2,
"ActionId": 2,
"ActionRisk": "High Risk",
"ActionTitle": "Start",
"ActionType": "Platform",
"IsExecuteRedo": false,
"TaskGroupActionCreateTime": "2023-10-09 10:55:18",
"TaskGroupActionCustomConfiguration": "{}",
"TaskGroupActionExecuteId": 457181,
"TaskGroupActionExecuteTime": null,
"TaskGroupActionGeneralConfiguration": "{\"PreTimeWait\": 0, \"ActionTimeout\": 1800, \"AfterTimeWait\": 0}",

```

```
"TaskGroupId": 13786,
"TaskGroupActionId": 13786,
"TaskGroupActionOrder": 2,
"TaskGroupActionRandomId": 355514,
"TaskGroupActionRecoverId": null,
"TaskGroupActionStatus": 2001,
"TaskGroupActionStatusType": 0,
"TaskGroupActionUpdateTime": "2023-10-09 10:55:18",
"TaskGroupInstances": []
},
{
"TaskGroupCreateTime": "2023-10-09 10:55:18",
"TaskGroupDescription": "1",
"TaskGroupDiscardInstanceList": [],
"TaskGroupId": 6454,
"TaskGroupInstanceList": [
"ins-knq6h3r8",
"ins-61eitwrk",
"ins-d2e45nba"
],
"TaskGroupInstancesExecuteRule": [
{
"TaskGroupInstancesExecuteMode": 3,
"TaskGroupInstancesExecuteNum": 2,
"TaskGroupInstancesExecutePercent": 100
}
],
"TaskGroupMode": 1,
"TaskGroupOrder": 1,
"TaskGroupSelectedInstanceList": [],
"TaskGroupTitle": "1",
"TaskGroupUpdateTime": "2023-10-09 10:55:18"
},
{
"TaskId": 5417,
"TaskMode": 1,
"TaskMonitors": [
{
"InstancesIds": [
"ins-knq6h3r8",
"ins-61eitwrk",
"ins-d2e45nba"
],
"MetricChineseName": "Check CPU utilization",
"MetricId": 614,
"MetricName": "CpuUsage",
"TaskMonitorId": 5850,
"TaskMonitorObjectType": 1,
```

```
"Unit": "%"  
}  
],  
"TaskOwnerUin": "700000174829",  
"TaskPauseDuration": 60,  
"TaskPlanId": null,  
"TaskPlanTitle": null,  
"TaskPolicy": null,  
"TaskProtectStrategy": null,  
"TaskRegionId": 1,  
"TaskStartTime": null,  
"TaskStatus": 1001,  
"TaskStatusType": 0,  
"TaskSummary": null,  
"TaskTag": "",  
"TaskTitle": "Testing",  
"TaskUpdateTime": "2023-10-09 10:55:18"  
}  
}  
}
```

5. Developer Resources

SDK

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

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for Node.js](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
AuthFailure	CAM signature/authentication error.
DryRunOperation	DryRun operation means the request will be successful, but the DryRun parameter is passed.
FailedOperation	Operation failed.
InternalError	Internal error.
InvalidParameter	Parameter error.
InvalidParameterValue	Parameter value error.
LimitExceeded	The quota limit is exceeded.
MissingParameter	Parameters are missing
OperationDenied	The operation is rejected.
RequestLimitExceeded	The number of requests exceeds the frequency limit.
ResourceInUse	Resources are occupied.
ResourceInsufficient	Insufficient resources.
ResourceNotFound	Resources do not exist.
ResourceUnavailable	Resources are unavailable.
ResourcesSoldOut	Resources are sold out.
UnauthorizedOperation	Unauthorized operation.
UnknownParameter	Unknown parameter.
UnsupportedOperation	The operation is not supported.

DescribeTaskExecuteLogs

Last updated : 2024-12-02 14:41:54

1. API Description

Domain name for API request: `cfg.intl.tencentcloudapi.com`.

This API is used to obtain all logs generated during an experiment.

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: DescribeTaskExecuteLogs.
Version	Yes	String	Common Params . The value used for this API: 2021-08-20.
Region	No	String	Common Params . This parameter is not required.
TaskId	Yes	Integer	Task ID
Limit	Yes	Integer	Number of returned content lines
Offset	Yes	Integer	Log start line

3. Output Parameters

--	--	--

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

4. Example

Example1 Example

Input Example

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

```
{
  "Limit": "10",
  "TaskId": "387",
  "Offset": "0"
}
```

Output Example

```
{
  "Response": {
    "RequestId": "ef03a439-2337-441b-9d53-89b618674c98",
    "LogMessage": []
  }
}
```

5. Developer Resources

SDK

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

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for Node.js](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
AuthFailure	CAM signature/authentication error.
AuthFailure.UnauthorizedOperation	Unauthorized CAM operation.
InternalError	Internal error.
InvalidParameter	Parameter error.
InvalidParameterValue	Parameter value error.
OperationDenied	The operation is rejected.

DescribeTaskList

Last updated : 2024-12-02 14:41:54

1. API Description

Domain name for API request: `cfg.intl.tencentcloudapi.com`.

This API is used to query the task list.

A maximum of 100 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: DescribeTaskList.
Version	Yes	String	Common Params . The value used for this API: 2021-08-20.
Region	No	String	Common Params . This parameter is not required.
Limit	Yes	Integer	Pagination limit
Offset	Yes	Integer	Pagination offset
TaskTitle	No	String	Experiment name
TaskTag.N	No	Array of String	Tag key
TaskStatus	No	Integer	Task status (1001: not started; 1002: in progress; 1003: paused; 1004: ended)

TaskStartTime	No	String	Start time, in fixed format: %Y-%m-%d %H:%M:%S
TaskEndTime	No	String	End time, in fixed format: %Y-%m-%d %H:%M:%S
TaskUpdateTime	No	String	Update time, in fixed format: %Y-%m-%d %H:%M:%S
Tags.N	No	Array of TagWithDescribe	Tag pair
Filters.N	No	Array of ActionFilter	Filtering criteria
TaskId.N	No	Array of Integer	Experiment ID
ApplicationId.N	No	Array of String	ID of the associated application for filtering
ApplicationName.N	No	Array of String	Associated application for filtering
TaskStatusList.N	No	Array of Integer	Task status for filtering, supporting multiple states (1001: not started; 1002: in progress; 1003: paused; 1004: ended)

3. Output Parameters

Parameter Name	Type	Description
TaskList	Array of TaskListItem	None
Total	Integer	Number of tables in the list
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Querying the Experiment List

This example shows you how to query the experiment list.

Input Example

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

```
{
  "Limit": "10",
  "TaskTag": [
    "Feiyang"
  ],
  "Offset": "0"
}
```

Output Example

```
{
  "Response": {
    "RequestId": "47e12dca-fa37-49b4-86e1-d7d3d7674640",
    "TaskList": [
      {
        "TaskId": 2,
        "TaskTitle": "Guangzhou zone 2 - shut down",
        "TaskDescription": "Guangzhou zone 1 - shut down",
        "TaskTag": "Feiyang",
        "TaskStatus": 1002,
        "TaskCreateTime": "2021-08-14 00:37:34",
        "TaskUpdateTime": "2021-09-18 19:18:28",
        "TaskPreCheckStatus": 0,
        "TaskPreCheckSuccess": false
      }
    ],
    "Total": 1
  }
}
```

Example2 Querying the Experiment List (Including Pre-check Status Information)

This example shows you how to query the experiment list (including pre-check status information).

Input Example

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

```
{
  "Limit": "2",
  "Offset": "0"
}
```

Output Example

```
{
  "Response": {
    "RequestId": "597dbef8-fcf3-46c2-9561-f87694052606",
    "TaskList": [
      {
        "TaskId": 3077,
        "TaskTitle": "Experiment pre-check test",
        "TaskDescription": "Experiment pre-check test",
        "TaskTag": "",
        "TaskStatus": 1001,
        "TaskCreateTime": "2022-09-19 12:57:22",
        "TaskUpdateTime": "2022-09-19 12:57:22",
        "TaskPreCheckStatus": 2,
        "TaskPreCheckSuccess": false
      },
      {
        "TaskId": 3076,
        "TaskTitle": "[Public cloud] Mongo node fault",
        "TaskDescription": "[Public cloud] Mongo node fault",
        "TaskTag": "",
        "TaskStatus": 1003,
        "TaskCreateTime": "2022-09-19 11:09:50",
        "TaskUpdateTime": "2022-09-19 12:17:49",
        "TaskPreCheckStatus": 0,
        "TaskPreCheckSuccess": false
      }
    ],
    "Total": 2450
  }
}
```

5. Developer Resources

SDK

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

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for Node.js](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
InternalError	Internal error.
InvalidParameter	Parameter error.
InvalidParameterValue	Parameter value error.
MissingParameter	Parameters are missing
UnknownParameter	Unknown parameter.
UnsupportedOperation	The operation is not supported.

DescribeTemplate

Last updated : 2024-12-02 14:41:53

1. API Description

Domain name for API request: `cfg.intl.tencentcloudapi.com`.

This API is used to query a template library.

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: DescribeTemplate.
Version	Yes	String	Common Params . The value used for this API: 2021-08-20.
Region	No	String	Common Params . This parameter is not required.
TemplateId	Yes	Integer	Template library ID

3. Output Parameters

Parameter Name	Type	Description
Template	Template	Template library details
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if

		the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.
--	--	--

4. Example

Example1 Querying a Template

This example shows you how to query a template.

Input Example

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

```
{
  "TemplateId": 1175
}
```

Output Example

```
{
  "Response": {
    "RequestId": "e5cf31b7-d530-4bce-8836-4c227b463b31",
    "Template": {
      "AlarmPolicy": [],
      "ApmServiceList": [],
      "Tags": [],
      "TemplateCreateTime": "2023-10-09 11:03:31",
      "TemplateDescription": "Test",
      "TemplateGroups": [
        {
          "CreateTime": "2023-10-09 11:03:31",
          "Description": "Test",
          "Mode": 1,
          "ObjectTypeId": 1,
          "Order": 1,
          "TemplateGroupActions": [
            {
              "ActionApiType": 1,
```

```
"ActionAttribute": 1,
"ActionId": 1,
"ActionTitle": "Shut down (test)",
"ActionType": "Platform",
"CreateTime": "2023-10-09 11:03:31",
"CustomConfiguration": "{}",
"ExecuteId": null,
"GeneralConfiguration": "{\"AliasTitle\": \"\", \"PreTimeWait\": 0, \"ActionTimeout\": 1800, \"AfterTimeWait\": 0}\",
"Order": 1,
"RandomId": 368448,
"RecoverId": 232618,
"TemplateGroupActionId": 3982,
"UpdateTime": "2023-10-09 11:03:31"
},
{
"ActionApiType": 1,
"ActionAttribute": 2,
"ActionId": 2,
"ActionTitle": "Start up",
"ActionType": "Platform",
"CreateTime": "2023-10-09 11:03:31",
"CustomConfiguration": "{}",
"ExecuteId": 368448,
"GeneralConfiguration": "{\"PreTimeWait\": 0, \"ActionTimeout\": 1800, \"AfterTimeWait\": 0}\",
"Order": 2,
"RandomId": 232618,
"RecoverId": null,
"TemplateGroupActionId": 3983,
"UpdateTime": "2023-10-09 11:03:31"
}
],
"TemplateGroupId": 1659,
"Title": "Test",
"UpdateTime": "2023-10-09 11:03:31"
}
],
"TemplateId": 1175,
"TemplateIsUsed": 1,
"TemplateMode": 1,
"TemplateMonitors": [
{
"MetricChineseName": "CPU utilization",
"MetricId": 614,
"MetricName": "CpuUsage",
"MonitorId": 2173,
```

```
"ObjectTypeId": 1
},
"TemplateOwnerUin": "700000174829",
"TemplatePauseDuration": 60,
"TemplatePolicy": null,
"TemplateRegionId": 1,
"TemplateSource": 0,
"TemplateTag": "",
"TemplateTitle": "Test",
"TemplateUpdateTime": "2023-10-09 11:03:31"
}
}
}
```

5. Developer Resources

SDK

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

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for Node.js](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description

AuthFailure	CAM signature/authentication error.
FailedOperation	Operation failed.
InternalError	Internal error.
InvalidParameter	Parameter error.
InvalidParameterValue	Parameter value error.
LimitExceeded	The quota limit is exceeded.
MissingParameter	Parameters are missing
OperationDenied	The operation is rejected.
ResourceInUse	Resources are occupied.
ResourceInsufficient	Insufficient resources.
UnknownParameter	Unknown parameter.
UnsupportedOperation	The operation is not supported.

ExecuteTask

Last updated : 2024-12-02 14:41:52

1. API Description

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

This API is used to run a task.

A maximum of 10 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: ExecuteTask.
Version	Yes	String	Common Params . The value used for this API: 2021-08-20.
Region	No	String	Common Params . This parameter is not required.
TaskId	Yes	Integer	ID of the task to be executed

3. Output Parameters

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

4. Example

Example1 Executing a Task

Input Example

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

{
  "TaskId": "222"
}
```

Output Example

```
{
  "Response": {
    "RequestId": "46924e75-a149-4130-aac0-853dbf0abea9"
  }
}
```

5. Developer Resources

SDK

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

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for Node.js](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
AuthFailure	CAM signature/authentication error.
FailedOperation	Operation failed.
InternalError	Internal error.
InvalidParameter	Parameter error.
UnsupportedOperation	The operation is not supported.

ExecuteTaskInstance

Last updated : 2024-12-02 14:41:51

1. API Description

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

This API is used to trigger the action of the chaos engineering experiment and perform an experiment on the instance.

A maximum of 10 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: ExecuteTaskInstance.
Version	Yes	String	Common Params . The value used for this API: 2021-08-20.
Region	No	String	Common Params . This parameter is not required.
TaskId	Yes	Integer	Task ID
TaskActionId	Yes	Integer	Task action ID
TaskInstanceIds.N	Yes	Array of Integer	Task action instance ID
IsOperateAll	Yes	Boolean	Whether to operate on the entire task
ActionType	Yes	Integer	Operation type (1: start; 2: execute; 3: skip; 5: retry)

TaskGroupId	Yes	Integer	Action group ID
-------------	-----	---------	-----------------

3. Output Parameters

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

4. Example

Example1 Example

Input Example

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

```
{
  "TaskId": "222",
  "TaskActionId": "1111",
  "TaskInstanceIds": [
    111,
    222,
    33
  ],
  "IsOperateAll": true,
  "ActionType": 1,
  "TaskGroupId": 12
}
```

Output Example

```
{
  "Response": {
    "RequestId": "6549ed1a-911f-46dd-b6cd-2c02d5bd180f"
  }
}
```

5. Developer Resources

SDK

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

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for Node.js](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
AuthFailure	CAM signature/authentication error.
AuthFailure.UnauthorizedOperation	Unauthorized CAM operation.
InternalError	Internal error.
InvalidParameter	Parameter error.
InvalidParameterValue	Parameter value error.

MissingParameter	Parameters are missing
OperationDenied	The operation is rejected.
ResourceInsufficient	Insufficient resources.
ResourceNotFound	Resources do not exist.
ResourceUnavailable	Resources are unavailable.
UnauthorizedOperation	Unauthorized operation.
UnsupportedOperation	The operation is not supported.

ModifyTaskRunStatus

Last updated : 2024-12-02 14:41:51

1. API Description

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

This API is used to change the task running status.

A maximum of 10 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: ModifyTaskRunStatus.
Version	Yes	String	Common Params . The value used for this API: 2021-08-20.
Region	No	String	Common Params . This parameter is not required.
TaskId	Yes	Integer	Task ID
Status	Yes	Integer	Task status. 1001: not started; 1002: in progress (executing); 1003: in progress (paused); 1004: execution ended.
IsExpect	No	Boolean	Whether the execution result meets expectations (This field is required when the task status is Execution Ended.)
Summary	No	String	Experiment result (This field is required when the experiment status changes to Execution Ended.)

3. Output Parameters

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

4. Example

Example1 Example 1

Input Example

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

{
  "Status": "1002",
  "TaskId": "1698"
}
```

Output Example

```
{
  "Response": {
    "RequestId": "8e9a0777-ff96-4020-8aec-6802d8103689"
  }
}
```

Example2 Terminating an Experiment

Input Example

```
POST / HTTP/1.1
Host: cfg.intl.tencentcloudapi.com
```

```
Content-Type: application/json
X-TC-Action: ModifyTaskRunStatus
<Common request parameters>
```

```
{
  "Status": "1004",
  "IsExpect": true,
  "TaskId": "222",
  "Summary": "Experiment result"
}
```

Output Example

```
{
  "Response": {
    "RequestId": "e38eca72-e4ae-4a86-9696-7df399e672bd"
  }
}
```

5. Developer Resources

SDK

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

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for Node.js](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
AuthFailure	CAM signature/authentication error.
FailedOperation	Operation failed.
InternalError	Internal error.
InvalidParameter	Parameter error.
InvalidParameterValue	Parameter value error.
MissingParameter	Parameters are missing
OperationDenied	The operation is rejected.
ResourceInUse	Resources are occupied.
UnsupportedOperation	The operation is not supported.

TriggerPolicy

Last updated : 2024-12-02 14:41:50

1. API Description

Domain name for API request: `cfg.intl.tencentcloudapi.com`.

This API is used to trigger the chaos engineering experiment guardrail (two types: trigger and recovery).

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: TriggerPolicy.
Version	Yes	String	Common Params . The value used for this API: 2021-08-20.
Region	No	String	Common Params . This parameter is not required.
TaskId	Yes	Integer	Chaos engineering experiment ID
Name	Yes	String	Name
Content	Yes	String	Triggering content
TriggerType	Yes	Integer	Triggering type. 0: trigger; 1: recovery.

3. Output Parameters

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

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

4. Example

Example1 1

1

Input Example

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

{
  "TaskId": 5491,
  "Name": "Trigger guardrail test",
  "Content": "Trigger guardrail test content",
  "TriggerType": 0
}
```

Output Example

```
{
  "Response": {
    "RequestId": "1050e4e3-ef1a-4e2f-b100-c68dfd38fc75",
    "Success": true,
    "TaskId": 5491
  }
}
```

5. Developer Resources

SDK

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

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for Node.js](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
AuthFailure	CAM signature/authentication error.
InternalError	Internal error.
InvalidParameter	Parameter error.
InvalidParameterValue	Parameter value error.
MissingParameter	Parameters are missing
RequestLimitExceeded	The number of requests exceeds the frequency limit.
UnauthorizedOperation	Unauthorized operation.
UnsupportedOperation	The operation is not supported.

DescribeTaskPolicyTriggerLog

Last updated : 2024-12-02 14:41:53

1. API Description

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

This API is used to obtain the guardrail triggering logs.

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: DescribeTaskPolicyTriggerLog.
Version	Yes	String	Common Params . The value used for this API: 2021-08-20.
Region	No	String	Common Params . This parameter is not required.
TaskId	Yes	Integer	Experiment ID
Page	Yes	Integer	Page number
PageSize	Yes	Integer	Number of entries per page

3. Output Parameters

--	--	--

Parameter Name	Type	Description
TriggerLogs	Array of PolicyTriggerLog	Triggering log Note: This field may return null, indicating that no valid values can be obtained.
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 1

1

Input Example

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

```
{
  "TaskId": 5491,
  "Page": 1,
  "PageSize": 11
}
```

Output Example

```
{
  "Response": {
    "RequestId": "1bba6839-682a-4123-9728-ec3fc141235b",
    "TriggerLogs": [
      {
        "Content": "Trigger the guardrail testing content-recovery",
        "CreatTime": "2023-11-14 12:51:33",
        "Name": "Trigger the guardrail testing-recovery",
```

```
"TaskId": 5491,
"TriggerType": 1
},
{
"Content": "Trigger the guardrail testing content",
"CreatTime": "2023-11-14 12:37:20",
"Name": "Trigger the guardrail testing",
"TaskId": 5491,
"TriggerType": 0
}
]
}
}
```

5. Developer Resources

SDK

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

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for Node.js](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
AuthFailure	CAM signature/authentication error.

AuthFailure.UnauthorizedOperation	Unauthorized CAM operation.
FailedOperation	Operation failed.
InternalError	Internal error.
InvalidParameter	Parameter error.
InvalidParameterValue	Parameter value error.
MissingParameter	Parameters are missing
RequestLimitExceeded	The number of requests exceeds the frequency limit.
UnauthorizedOperation	Unauthorized operation.
UnknownParameter	Unknown parameter.
UnsupportedOperation	The operation is not supported.

CreateTaskFromAction

Last updated : 2024-12-02 14:41:58

1. API Description

Domain name for API request: `cfg.intl.tencentcloudapi.com`.

This API is used to create an experiment with an action.

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: CreateTaskFromAction.
Version	Yes	String	Common Params . The value used for this API: 2021-08-20.
Region	No	String	Common Params . This parameter is not required.
TaskActionId	Yes	Integer	Action ID, which can be obtained by using the action list API DescribeActionLibraryList.
TaskInstances.N	Yes	Array of String	ID of the instance participating in the experiment.
TaskTitle	No	String	Experiment name. If this parameter is left blank, the action name is used by default.
TaskDescription	No	String	Experiment description. If this parameter is left

			blank, the action description is used by default.
TaskActionGeneralConfiguration	No	String	General action parameters, which need to be passed in after JSON serialization. The parameters can be obtained by using the action details API DescribeActionFieldConfigList. If this field is left blank, the default action parameters are used by default.
TaskActionCustomConfiguration	No	String	Action custom parameters need to be passed in as json serialization. They can be obtained from the action details interface DescribeActionFieldConfigList. If not filled in, the default action parameters will be used. Note: Required parameters have no default values. Be sure to pass in valid values.
TaskPauseDuration	No	Integer	Automatic experiment pause time, in minutes. If this parameter is left blank, the default value 60 is used.

3. Output Parameters

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

4. Example

Example1 Creating an Experiment with Actions

Input Example

```
POST / HTTP/1.1
Host: cfg.intl.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateTaskFromAction
```

<Common request parameters>

```
{
  "TaskActionId": 127,
  "TaskInstances": [
    "ins-87bfr32q"
  ]
}
```

Output Example

```
{
  "Response": {
    "RequestId": "f0aee8ac-2ed3-4a7f-a25b-f0d7d228dd30",
    "TaskId": 50
  }
}
```

5. Developer Resources

SDK

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

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for Node.js](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Template Library APIs

DescribeTemplateList

Last updated : 2024-12-02 14:41:50

1. API Description

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

This API is used to query the template list.

A maximum of 100 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: DescribeTemplateList.
Version	Yes	String	Common Params . The value used for this API: 2021-08-20.
Region	No	String	Common Params . This parameter is not required.
Limit	Yes	Integer	Pagination limit.Maximum value:100.
Offset	Yes	Integer	Pagination offset
Title	No	String	Experiment name
Tag.N	No	Array of String	Tag key
IsUsed	No	Integer	Status. 1: in use; 2: not in use.

Tags.N	No	Array of TagWithDescribe	Tag pair
TemplateSource	No	Integer	Template library source. 0: self-built; 1: recommended by experts.
TemplateIdList.N	No	Array of Integer	Template ID
Filters.N	No	Array of ActionFilter	Filter parameters

3. Output Parameters

Parameter Name	Type	Description
TemplateList	Array of TemplateListItem	Template library list
Total	Integer	Number of template libraries in the list
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Template List

This example shows you how to query the template list.

Input Example

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

```
{
```

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

Output Example

```
{  
  "Response": {  
    "RequestId": "0c06dc9d-1e90-4062-a038-74abf2bbd43d",  
    "TemplateList": [  
      {  
        "TemplateId": 511,  
        "TemplateTitle": "Testing template",  
        "TemplateDescription": "Test",  
        "TemplateTag": null,  
        "TemplateIsUsed": 1,  
        "TemplateCreateTime": "2021-10-12 17:28:13",  
        "TemplateUpdateTime": "2021-10-12 17:48:16",  
        "TemplateUsedNum": 0  
      },  
      {  
        "TemplateId": 509,  
        "TemplateTitle": "test",  
        "TemplateDescription": "test",  
        "TemplateTag": "test",  
        "TemplateIsUsed": 2,  
        "TemplateCreateTime": "2021-09-28 15:38:00",  
        "TemplateUpdateTime": "2021-10-11 16:26:45",  
        "TemplateUsedNum": 0  
      }  
    ],  
    "Total": 2  
  }  
}
```

Example2 Querying the Template List

This example shows you how to query the template list.

Input Example

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

```
{
  "Limit": 10,
  "Offset": 1
}
```

Output Example

```
{
  "Response": {
    "RequestId": "6775138e-92c5-40cb-ab0e-a61a5c63b5fa",
    "TemplateList": [
      {
        "TemplateCreateTime": "2023-06-06 15:05:18",
        "TemplateDescription": "3",
        "TemplateId": 914,
        "TemplateIsUsed": 1,
        "TemplateSource": 0,
        "TemplateTag": "",
        "TemplateTitle": "3",
        "TemplateUpdateTime": "2023-06-06 15:05:18",
        "TemplateUsedNum": 0
      },
      {
        "TemplateCreateTime": "2023-06-05 14:43:42",
        "TemplateDescription": "Simulate single-AZ faults at the access layer, logic layer, and data layer to verify the effectiveness of cross-AZ disaster recovery of the architecture.",
        "TemplateId": 913,
        "TemplateIsUsed": 1,
        "TemplateSource": 1,
        "TemplateTag": "",
        "TemplateTitle": "Cross-AZ disaster recovery experiment",
        "TemplateUpdateTime": "2023-06-05 14:43:42",
        "TemplateUsedNum": 0
      },
      {
        "TemplateCreateTime": "2023-06-05 14:42:31",
        "TemplateDescription": "Simulate single-AZ faults at the access layer, logic layer, and data layer to verify the effectiveness of cross-AZ disaster recovery of the architecture.",
        "TemplateId": 912,
        "TemplateIsUsed": 1,
        "TemplateSource": 1,

```

```
"TemplateTag": "",
"TemplateTitle": "Cross-AZ disaster recovery experiment",
"TemplateUpdateTime": "2023-06-05 14:42:31",
"TemplateUsedNum": 0
},
{
"TemplateCreateTime": "2023-06-05 11:48:02",
"TemplateDescription": "Simulate single-AZ faults at the access layer, logic layer, and data layer to verify the effectiveness of cross-AZ disaster recovery of the architecture.",
"TemplateId": 910,
"TemplateIsUsed": 1,
"TemplateSource": 1,
"TemplateTag": "",
"TemplateTitle": "Cross-AZ disaster recovery experiment",
"TemplateUpdateTime": "2023-06-05 11:48:02",
"TemplateUsedNum": 0
},
{
"TemplateCreateTime": "2023-06-05 11:48:02",
"TemplateDescription": "Simulate single-AZ faults at the access layer, logic layer, and data layer to verify the effectiveness of cross-AZ disaster recovery of the architecture.",
"TemplateId": 911,
"TemplateIsUsed": 1,
"TemplateSource": 1,
"TemplateTag": "",
"TemplateTitle": "Cross-AZ disaster recovery experiment",
"TemplateUpdateTime": "2023-06-05 11:48:02",
"TemplateUsedNum": 0
},
{
"TemplateCreateTime": "2023-06-02 17:38:34",
"TemplateDescription": "Simulate single-AZ faults at the access layer, logic layer, and data layer to verify the effectiveness of cross-AZ disaster recovery of the architecture.",
"TemplateId": 904,
"TemplateIsUsed": 1,
"TemplateSource": 1,
"TemplateTag": "",
"TemplateTitle": "Cross-AZ disaster recovery experiment",
"TemplateUpdateTime": "2023-06-02 17:38:34",
"TemplateUsedNum": 0
},
{
"TemplateCreateTime": "2023-06-02 17:38:34",
"TemplateDescription": "Simulate single-AZ faults at the access layer, logic layer, and data layer to verify the effectiveness of cross-AZ disaster recovery of the architecture.",
"TemplateId": 905,
"TemplateIsUsed": 1,
"TemplateSource": 1,
"TemplateTag": "",
"TemplateTitle": "Cross-AZ disaster recovery experiment",
"TemplateUpdateTime": "2023-06-02 17:38:34",
"TemplateUsedNum": 0
}
```



```
r, and data layer to verify the effectiveness of cross-AZ disaster recovery of th
e architecture.",
"TemplateId": 905,
"TemplateIsUsed": 1,
"TemplateSource": 1,
"TemplateTag": "",
"TemplateTitle": "Cross-AZ disaster recovery experiment",
"TemplateUpdateTime": "2023-06-02 17:38:34",
"TemplateUsedNum": 0
},
{
"TemplateCreateTime": "2023-06-02 17:38:34",
"TemplateDescription": "Simulate single-AZ faults at the access layer, logic laye
r, and data layer to verify the effectiveness of cross-AZ disaster recovery of th
e architecture.",
"TemplateId": 906,
"TemplateIsUsed": 1,
"TemplateSource": 1,
"TemplateTag": "",
"TemplateTitle": "Cross-AZ disaster recovery experiment",
"TemplateUpdateTime": "2023-06-02 17:38:34",
"TemplateUsedNum": 0
},
{
"TemplateCreateTime": "2023-06-02 17:38:34",
"TemplateDescription": "Simulate single-AZ faults at the access layer, logic laye
r, and data layer to verify the effectiveness of cross-AZ disaster recovery of th
e architecture.",
"TemplateId": 907,
"TemplateIsUsed": 1,
"TemplateSource": 1,
"TemplateTag": "",
"TemplateTitle": "Cross-AZ disaster recovery experiment",
"TemplateUpdateTime": "2023-06-02 17:38:34",
"TemplateUsedNum": 0
},
{
"TemplateCreateTime": "2023-06-02 17:38:34",
"TemplateDescription": "Simulate single-AZ faults at the access layer, logic laye
r, and data layer to verify the effectiveness of cross-AZ disaster recovery of th
e architecture.",
"TemplateId": 908,
"TemplateIsUsed": 2,
"TemplateSource": 1,
"TemplateTag": "",
"TemplateTitle": "Cross-AZ disaster recovery experiment",
"TemplateUpdateTime": "2023-06-13 16:30:37",
```

```
"TemplateUsedNum": 0
}
],
"Total": 43
}
}
```

5. Developer Resources

SDK

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

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for Node.js](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
InternalError	Internal error.
InvalidParameter	Parameter error.
InvalidParameterValue	Parameter value error.
MissingParameter	Parameters are missing
UnknownParameter	Unknown parameter.

UnsupportedOperation	The operation is not supported.
----------------------	---------------------------------

Data Types

Last updated : 2024-09-25 10:28:12

ActionFieldConfigDetail

Response format of dynamic action parameters

Used by actions: DescribeActionFieldConfigList.

Name	Type	Description
Type	String	Component type The options are as follows: input: text box textarea: multi-line text box number: number input box select: selector cascader: cascade selector radio: single choice time: time selection
Lable	String	Component label
Field	String	Unique identifier of the component, key when it is sent back to the backend
DefaultValue	String	Default value Note: This field may return null, indicating that no valid values can be obtained.
Config	String	The supported configuration items are as follows. You can select the configuration items as needed. If no configuration is required, set the value to {}. { placeholder: string (placeholder) tooltip: string (prompt message) reg: RegExp (regular expression for the input content format) max: number (maximum number of input characters for text boxes and upper limit of the input number for number input boxes) min: number (lower limit of the input number for number input boxes) step: number (step size for number input boxes; default value: 1)

		format: string (format for time selection, for example YYYY-MM-DD and YYYY-MM-DD HH:mm:ss) separator: string[] (separator for multi-line input boxes. If it is left blank, no separator is used, and the text string entered by the user is returned directly.) multiple: boolean (multiple-choice or not, valid for selectors and cascade selectors) options: selector options (support the following two forms) Directly provide the option array: { value: string; label: string }[] Obtain options by calling the API: { api: string (API URL), params: string[] (interface parameters, corresponding to field of the parameter configuration. The frontend uses the input values of all components corresponding to field as parameters to query data. If no value is input, the frontend directly requests data when components are loaded.) <pre>} }</pre>
Required	Integer	Whether it is required (0: no; 1: yes)
Validate	String	The compute configuration passes the verification when other fields that it depends on meet the conditions. (For example, at least one of the three form items must be filled in.) [fieldName, { config: This item is retained and will be refined based on subsequent scenes. }]
Visible	String	Whether it is visible

ActionFieldConfigResult

Action field configuration result

Used by actions: DescribeActionFieldConfigList.

Name	Type	Description
ActionId	Integer	Action ID
ActionName	String	Action name

ConfigDetail	Array of ActionFieldConfigDetail	Filed configuration details corresponding to the action
--------------	--	---

ActionFilter

Action library filtering field

Used by actions: DescribeActionLibraryList, DescribeTaskList, DescribeTemplateList.

Name	Type	Required	Description
Keyword	String	Yes	Keyword
Values	Array of String	Yes	Content for search

ActionLibraryListResult

Action library data list

Used by actions: DescribeActionLibraryList.

Name	Type	Description
ActionName	String	Action name
Desc	String	Action description
ActionType	String	Action type: ["platform" and "custom"]
CreateTime	String	Creation time
Creator	String	Creator
UpdateTime	String	Update time
RiskDesc	String	Action risk description
ActionId	Integer	Action ID
AttributeId	Integer	Action attribute (1: fault; 2: recovery)
RelationActionId	Integer	ID of the associated action
ActionCommand	String	Operation command

ActionCommandType	Integer	Action type (0: tat; 1: cloud API)
ActionContent	String	Parameters of the custom action, in JSON string format
ResourceType	String	Level-2 type Note: This field may return null, indicating that no valid values can be obtained.
ActionDetail	String	Action description Note: This field may return null, indicating that no valid values can be obtained.
IsAllowed	Boolean	Whether to allow usage by the current account Note: This field may return null, indicating that no valid values can be obtained.
ActionBestCase	String	Link to best practices Note: This field may return null, indicating that no valid values can be obtained.
ObjectType	String	Object type Note: This field may return null, indicating that no valid values can be obtained.
MetricIdList	Array of Integer	Monitoring metric ID list Note: This field may return null, indicating that no valid values can be obtained.
IsNewAction	Boolean	Whether the action is new Note: This field may return null, indicating that no valid values can be obtained.

ApmServiceInfo

Application information on Application Performance Monitoring

Used by actions: DescribeTask, DescribeTemplate.

Name	Type	Required	Description
InstanceId	String	Yes	Business ID Note: This field may return null, indicating that no valid values can be obtained.
ServiceNameList	Array of String	Yes	Application name Note: This field may return null, indicating that no valid values

			can be obtained.
RegionId	Integer	No	Region ID Note: This field may return null, indicating that no valid values can be obtained.

DescribePolicy

Protection policy

Used by actions: DescribeTask.

Name	Type	Required	Description
TaskPolicyIdList	Array of String	Yes	Protection policy ID list
TaskPolicyStatus	String	Yes	Protection policy status
TaskPolicyRule	String	Yes	Policy rule
TaskPolicyDealType	Integer	Yes	Process method when the guardrail policy takes effect. 1: sequential execution, 2: pausing. Note: This field may return null, indicating that no valid values can be obtained.

ObjectType

Object type

Used by actions: DescribeObjectTypeList.

Name	Type	Description
ObjectTypeId	Integer	Object type ID
ObjectTypeTitle	String	Object type name
ObjectTypeLevelOne	String	Level-1 object type
ObjectTypeParams	ObjectTypeConfig	Object type parameters
ObjectTypeJsonParse	ObjectTypeJsonParse	JSON parsing rule for TKE APIs. If the value is null, no

		parsing is needed. Note: This field may return null, indicating that no valid values can be obtained.
ObjectHasNewAction	Boolean	Whether new action is included Note: This field may return null, indicating that no valid values can be obtained.

ObjectTypeConfig

Object type configuration

Used by actions: DescribeObjectTypeInfoList.

Name	Type	Description
Key	String	Primary key
Fields	Array of ObjectTypeConfigFields	List of object type configuration fields

ObjectTypeConfigFields

Field information on the experiment resource object

Used by actions: DescribeObjectTypeInfoList.

Name	Type	Required	Description
Key	String	Yes	instanceId
Header	String	Yes	Instance ID
Transfer	String	No	Whether the field value needs to be translated. If not, this field returns null. Note: This field may return null, indicating that no valid values can be obtained.
JsonParse	String	No	Value returned for container Pod resources Note: This field may return null, indicating that no valid values can be obtained.

ObjectTypeJsonParse

Value returned for container Pod resources

Used by actions: DescribeObjectTypeInfoList.

Name	Type	Description
Namespace	String	Namespace Note: This field may return null, indicating that no valid values can be obtained.
WorkloadName	String	Workload name Note: This field may return null, indicating that no valid values can be obtained.
LanIP	String	Node IP address Note: This field may return null, indicating that no valid values can be obtained.
InstanceId	String	Node ID Note: This field may return null, indicating that no valid values can be obtained.

PolicyTriggerLog

Guardrail policy triggering log

Used by actions: DescribeTaskPolicyTriggerLog.

Name	Type	Description
TaskId	Integer	Experiment ID Note: This field may return null, indicating that no valid values can be obtained.
Name	String	Name Note: This field may return null, indicating that no valid values can be obtained.
TriggerType	Integer	Type. 0: trigger; 1: recovery. Note: This field may return null, indicating that no valid values can be obtained.
Content	String	Content Note: This field may return null, indicating that no valid values can be obtained.
CreateTime	String	Triggering time Note: This field may return null, indicating that no valid values can be obtained.

ResourceOffline

Decommissioned resource

Used by actions: DescribeActionFieldConfigList.

Name	Type	Required	Description
ResourceId	Integer	No	Resource ID Note: This field may return null, indicating that no valid values can be obtained.
ResourceDeleteTime	String	No	Resource decommissioning time Note: This field may return null, indicating that no valid values can be obtained.
ResourceDeleteMessage	String	No	Resource decommissioning reminder Note: This field may return null, indicating that no valid values can be obtained.

TagWithCreate

Tag information, which is used during experiment resource creation and editing.

Used by actions: CreateTaskFromTemplate.

Name	Type	Required	Description
TagKey	String	Yes	Tag key
TagValue	String	Yes	Tag value

TagWithDescribe

List of displayed tags

Used by actions: DescribeTask, DescribeTaskList, DescribeTemplate, DescribeTemplateList.

Name	Type	Required	Description
TagKey	String	Yes	Tag key
TagValue	String	Yes	Tag value

Task

Task

Used by actions: DescribeTask.

Name	Type	Required	Description
TaskId	Integer	Yes	Task ID
TaskTitle	String	Yes	Task name
TaskDescription	String	Yes	Task description
TaskTag	String	Yes	Custom tag Note: This field may return null, indicating that no valid values can be obtained.
TaskStatus	Integer	Yes	Task status. 1001: not started; 1002: in progress (executing); 1003: in progress (paused); 1004: execution ended. Note: This field may return null, indicating that no valid values can be obtained.
TaskStatusType	Integer	Yes	Task end status, indicating the status of the task when it ends. 0: not ended; 1: successful; 2: failed; 3: terminated.
TaskProtectStrategy	String	Yes	Protection policy Note: This field may return null, indicating that no valid values can be obtained.
TaskCreateTime	String	Yes	Task creation time
TaskUpdateTime	String	Yes	Task update time
TaskGroups	Array of TaskGroup	Yes	Task action group
TaskStartTime	String	Yes	Start time Note: This field may return null, indicating that no valid values can be obtained.
TaskEndTime	String	Yes	End time

			Note: This field may return null, indicating that no valid values can be obtained.
TaskExpect	Integer	Yes	Whether expectations are met. 1: expectations met; 2: expectations not met. Note: This field may return null, indicating that no valid values can be obtained.
TaskSummary	String	Yes	Experiment record Note: This field may return null, indicating that no valid values can be obtained.
TaskMode	Integer	Yes	Task mode. 1: manual execution; 2: automatic execution.
TaskPauseDuration	Integer	Yes	Automatic pause duration. Unit: minutes.
TaskOwnerUin	String	Yes	Main account that creates the experiment
TaskRegionId	Integer	Yes	Region ID
TaskMonitors	Array of TaskMonitor	Yes	Monitoring metric list Note: This field may return null, indicating that no valid values can be obtained.
TaskPolicy	DescribePolicy	Yes	Protection policy Note: This field may return null, indicating that no valid values can be obtained.
Tags	Array of TagWithDescribe	No	Tag list Note: This field may return null, indicating that no valid values can be obtained.
TaskPlanId	Integer	No	ID of the associated experiment plan Note: This field may return null, indicating that no valid values can be obtained.
TaskPlanTitle	String	No	Name of the associated experiment plan Note: This field may return null, indicating that no valid values can be obtained.
ApplicationId	String	No	ID of the associated application Note: This field may return null, indicating that no valid values can be obtained.
ApplicationName	String	No	Name of the associated application Note: This field may return null, indicating that no valid values can be obtained.

AlarmPolicy	Array of String	No	Associated alarm metrics Note: This field may return null, indicating that no valid values can be obtained.
ApmServiceList	Array of ApmServiceInfo	No	Associated APM services Note: This field may return null, indicating that no valid values can be obtained.
VerifyId	Integer	No	ID of the associated threat verification item Note: This field may return null, indicating that no valid values can be obtained.
PolicyDealType	Integer	No	Guardrail processing method. 1: roll back sequentially; 2: pause experiment. Note: This field may return null, indicating that no valid values can be obtained.

TaskConfig

Task parameters that need to be configured when an experiment is created by using a template

Used by actions: CreateTaskFromTemplate.

Name	Type	Required	Description
TaskGroupsConfig	Array of TaskGroupConfig	Yes	Action group configurations. The number of configured action groups needs to be consistent with that in the template.
TaskTitle	String	No	Experiment name after change. If this parameter is left blank, the template name is used by default.
TaskDescription	String	No	Experiment description after change. If this parameter is left blank, the template description is used by default.
TaskMode	Integer	No	Experiment execution mode. 1: manual execution; 2: automatic execution. If this parameter is left blank, the template execution mode is used by default.
TaskPauseDuration	Integer	No	Automatic pause time of the experiment, in minutes. If this parameter is left blank, the automatic pause time of the template is used by default.
Tags	Array of	No	Experiment tag. If this parameter is left blank, the

	TagWithCreate		template tag is used by default.
PolicyDealType	Integer	No	Guardrail processing method. 1: roll back sequentially; 2: pause experiment.

TaskGroup

Task group

Used by actions: DescribeTask.

Name	Type	Required	Description
TaskGroupId	Integer	Yes	Task action ID
TaskGroupTitle	String	Yes	Group name
TaskGroupDescription	String	Yes	Group description Note: This field may return null, indicating that no valid values can be obtained.
TaskGroupOrder	Integer	Yes	Task group order
ObjectTypeId	Integer	Yes	Object type ID
TaskGroupCreateTime	String	Yes	Task group creation time
TaskGroupUpdateTime	String	Yes	Task group update time
TaskGroupActions	Array of TaskGroupAction	Yes	List of actions in the group
TaskGroupInstanceList	Array of String	Yes	Instance list
TaskGroupMode	Integer	Yes	Execution mode. 1: sequential execution; 2: execution by stage.
TaskGroupDiscardInstanceList	Array of String	No	List of instances not involved in the experiment

TaskGroupSelectedInstanceList	Array of String	No	List of instances involved in the experiment Note: This field may return null, indicating that no valid values can be obtained.
TaskGroupInstancesExecuteRule	Array of TaskGroupInstancesExecuteRules	No	Machine selection rule Note: This field may return null, indicating that no valid values can be obtained.

TaskGroupAction

Task group action

Used by actions: DescribeTask.

Name	Type	Required	Description
TaskGroupActionId	Integer	Yes	Task group action ID
TaskGroupInstances	Array of TaskGroupInstance	Yes	Action instance list of the task group
ActionId	Integer	Yes	Action ID
TaskGroupActionOrder	Integer	Yes	Order of actions in the group
TaskGroupActionGeneralConfiguration	String	Yes	General configurations of actions in the group Note: This field may return null, indicating that no valid values can be obtained.
TaskGroupActionCustomConfiguration	String	Yes	Custom configurations of actions in the group Note: This field may return null, indicating that no valid values can be obtained.
TaskGroupActionStatus	Integer	Yes	Status of actions in the group

TaskGroupActionCreateTime	String	Yes	Action group creation time
TaskGroupActionUpdateTime	String	Yes	Action group update time
ActionTitle	String	Yes	Action name
TaskGroupActionStatusType	Integer	Yes	Status. 0: no status; 1: successful; 2: failed; 3: terminated; 4: skipped.
TaskGroupActionRandomId	Integer	Yes	RandomId Note: This field may return null, indicating that no valid values can be obtained.
TaskGroupActionRecoverId	Integer	Yes	RecoverId Note: This field may return null, indicating that no valid values can be obtained.
TaskGroupActionExecuteId	Integer	Yes	ExecuteId Note: This field may return null, indicating that no valid values can be obtained.
ActionApiType	Integer	No	Called API type. 0: tat; 1: cloud API. Note: This field may return null, indicating that no valid values can be obtained.
ActionAttribute	Integer	No	1: fault; 2: recovery. Note: This field may return null, indicating that no valid values can be obtained.
ActionType	String	No	Action type: platform and custom Note: This field may return null, indicating that no valid values can be obtained.
IsExecuteRedo	Boolean	No	Whether retry is allowed Note: This field may return null, indicating that no valid values can be obtained.

ActionRisk	String	No	Action risk level Note: This field may return null, indicating that no valid values can be obtained.
TaskGroupActionExecuteTime	Integer	No	Action running time Note: This field may return null, indicating that no valid values can be obtained.

TaskGroupActionConfig

Action parameters in the action group

Used by actions: CreateTaskFromTemplate.

Name	Type	Required	Description
TaskGroupActionOrder	Integer	No	Order of this action in the action group. The entire order starts from 1. If this parameter is left blank or set to an invalid value, the action whose parameters need to be modified in the template cannot be matched.
TaskGroupActionGeneralConfiguration	String	No	General action parameters, which need to be passed in after JSON serialization. The parameters can be obtained by using the template details query API. If this field is left blank, action parameters in the template are used by default.
TaskGroupActionCustomConfiguration	String	No	Custom action parameters, which need to be passed in after JSON serialization. The parameters can be obtained by using the template details query API. If this field is left blank, action parameters in the template are used by default.

TaskGroupConfig

Action group configuration item

Used by actions: CreateTaskFromTemplate.

Name	Type	Required	Description
TaskGroupInstances	Array of String	Yes	Instance object associated with the action group
TaskGroupTitle	String	No	Action group name. If this parameter is left blank, the action group name in the template is used by default.
TaskGroupDescription	String	No	Action group description. If this parameter is left blank, the action group description in the template is used by default.
TaskGroupMode	Integer	No	Action group execution mode. 1: sequential execution; 2: execution by stage. If this parameter is left blank, the action execution mode in the template is used by default.
TaskGroupActionsConfig	Array of TaskGroupActionConfig	No	Action parameters in the action group. If this field is left blank, the action parameters in the template is used by default. You only need to specify the action whose parameters are to be modified during configuration.

TaskGroupInstance

Task group action instance

Used by actions: DescribeTask.

Name	Type	Required	Description
TaskGroupInstanceId	Integer	Yes	Instance ID
TaskGroupInstanceObjectId	String	Yes	Instance ID Note: This field may return null, indicating that no valid values can be obtained.
TaskGroupInstanceStatus	Integer	Yes	Instance action execution status
TaskGroupInstanceCreateTime	String	Yes	Instance creation time

TaskGroupInstanceUpdateTime	String	Yes	Instance update time
TaskGroupInstanceStatusType	Integer	Yes	Status. 0: no status; 1: successful; 2: failed; 3: terminated; 4: skipped.
TaskGroupInstanceStartTime	String	Yes	Execution start time Note: This field may return null, indicating that no valid values can be obtained.
TaskGroupInstanceEndTime	String	Yes	Execution end time Note: This field may return null, indicating that no valid values can be obtained.
TaskGroupInstancesIsRedo	Boolean	No	Whether the instance can be retried Note: This field may return null, indicating that no valid values can be obtained.
TaskGroupInstanceExecuteTime	Integer	No	Action instance execution time Note: This field may return null, indicating that no valid values can be obtained.

TaskGroupInstancesExecuteRules

Machine selection rule

Used by actions: DescribeTask.

Name	Type	Required	Description
TaskGroupInstancesExecuteMode	Integer	No	Instance selection mode Note: This field may return null, indicating that no valid values can be obtained.
TaskGroupInstancesExecutePercent	Integer	No	Proportion of selected machines in selection by proportion mode Note: This field may return null, indicating that no valid values can be obtained.
TaskGroupInstancesExecuteNum	Integer	No	Number of selected machines in selection by quantity mode Note: This field may return null, indicating that no valid values can be obtained.

TaskListItem

Task list information

Used by actions: DescribeTaskList.

Name	Type	Description
TaskId	Integer	Task ID
TaskTitle	String	Task name
TaskDescription	String	Task description
TaskTag	String	Task tag Note: This field may return null, indicating that no valid values can be obtained.
TaskStatus	Integer	Task status (1001: not started; 1002: in progress; 1003: paused; 1004: ended)
TaskCreateTime	String	Task creation time
TaskUpdateTime	String	Task update time
TaskPreCheckStatus	Integer	0: not started; 1: in progress; 2: completed. Note: This field may return null, indicating that no valid values can be obtained.
TaskPreCheckSuccess	Boolean	Whether the environmental check is passed Note: This field may return null, indicating that no valid values can be obtained.
TaskExpect	Integer	Whether the experiment result meets expectations. 1: expectations met; 2: expectations not met. Note: This field may return null, indicating that no valid values can be obtained.
ApplicationId	String	ID of the associated application Note: This field may return null, indicating that no valid values can be obtained.
ApplicationName	String	Name of the associated application Note: This field may return null, indicating that no valid values can be obtained.
VerifyId	Integer	Verification item ID

		Note: This field may return null, indicating that no valid values can be obtained.
TaskStatusType	Integer	Status. 0: no status; 1: successful; 2: failed; 3: terminated. Note: This field may return null, indicating that no valid values can be obtained.

TaskMonitor

Monitoring metrics

Used by actions: DescribeTask.

Name	Type	Required	Description
TaskMonitorId	Integer	Yes	Experiment monitoring metric ID
MetricId	Integer	Yes	Monitoring metric ID Note: This field may return null, indicating that no valid values can be obtained.
TaskMonitorObjectTypeId	Integer	Yes	Object type ID of the monitoring metric
MetricName	String	Yes	Metric name
InstancesIds	Array of String	Yes	Instance ID list
MetricChineseName	String	Yes	Metric in Chinese Note: This field may return null, indicating that no valid values can be obtained.
Unit	String	Yes	Unit Note: This field may return null, indicating that no valid values can be obtained.

TaskReportInfo

Experiment report status information

Used by actions: DescribeTask.

Name	Type	Description

Stage	Integer	0: not started; 1: exporting; 2: export successful; 3: export failed.
CreateTime	String	Creation time
ExpirationTime	String	End time of the validity period
Expired	Boolean	Whether it is effective
CosUrl	String	Address of the COS experiment report file Note: This field may return null, indicating that no valid values can be obtained.
Log	String	Experiment report export log Note: This field may return null, indicating that no valid values can be obtained.

Template

Template library

Used by actions: DescribeTemplate.

Name	Type	Description
TemplateId	Integer	Template library ID
TemplateTitle	String	Template library name
TemplateDescription	String	Template library description
TemplateTag	String	Custom tag Note: This field may return null, indicating that no valid values can be obtained.
TemplateIsUsed	Integer	Usage status. 1: in use; 2: not in use.
TemplateCreateTime	String	Template library creation time
TemplateUpdateTime	String	Template library update time
TemplateMode	Integer	Template library mode. 1: manual execution; 2: automatic execution.
TemplatePauseDuration	Integer	Automatic pause duration. Unit: minutes.
TemplateOwnerUin	String	Main account that creates the experiment
TemplateRegionId	Integer	Region ID

TemplateGroups	Array of TemplateGroup	Action group
TemplateMonitors	Array of TemplateMonitor	Monitoring metrics
TemplatePolicy	TemplatePolicy	Guardrail monitoring Note: This field may return null, indicating that no valid values can be obtained.
Tags	Array of TagWithDescribe	Tag list Note: This field may return null, indicating that no valid values can be obtained.
TemplateSource	Integer	Template library source. 0: self-built; 1: recommended by experts. Note: This field may return null, indicating that no valid values can be obtained.
ApmServiceList	Array of ApmServiceInfo	Application information on Application Performance Monitoring Note: This field may return null, indicating that no valid values can be obtained.
AlarmPolicy	Array of String	Alarm metrics Note: This field may return null, indicating that no valid values can be obtained.
PolicyDealType	Integer	Guardrail processing method. 1: roll back sequentially; 2: pause experiment. Note: This field may return null, indicating that no valid values can be obtained.

TemplateGroup

Task group

Used by actions: DescribeTemplate.

Name	Type	Required	Description
TemplateGroupId	Integer	Yes	Template library action ID
TemplateGroupActions	Array of TemplateGroupAction	Yes	List of actions in the template library action group

Title	String	Yes	Group name
Description	String	Yes	Group description Note: This field may return null, indicating that no valid values can be obtained.
Order	Integer	Yes	Group order
Mode	Integer	Yes	Execution mode. 1: sequential execution; 2: execution by stage.
ObjectTypeId	Integer	Yes	Object type ID
CreateTime	String	Yes	Group creation time
UpdateTime	String	Yes	Group update time

TemplateGroupAction

Task group action

Used by actions: DescribeTemplate.

Name	Type	Required	Description
TemplateGroupId	Integer	Yes	Template library group action ID
ActionId	Integer	Yes	Action ID
Order	Integer	Yes	Order of actions in the group
GeneralConfiguration	String	Yes	General configurations of actions in the group Note: This field may return null, indicating that no valid values can be obtained.
CustomConfiguration	String	Yes	Custom configurations of actions in the group Note: This field may return null, indicating that no valid values can be obtained.
CreateTime	String	Yes	Action group creation time
UpdateTime	String	Yes	Action group update time
ActionTitle	String	Yes	Action name
RandomId	Integer	Yes	Random ID

			Note: This field may return null, indicating that no valid values can be obtained.
RecoverId	Integer	Yes	Recovery action ID Note: This field may return null, indicating that no valid values can be obtained.
ExecutedId	Integer	Yes	Executed action ID Note: This field may return null, indicating that no valid values can be obtained.
ActionApiType	Integer	No	Called API type. 0: tat; 1: cloud API. Note: This field may return null, indicating that no valid values can be obtained.
ActionAttribute	Integer	No	1: fault; 2: recovery. Note: This field may return null, indicating that no valid values can be obtained.
ActionType	String	No	Action type: platform and custom Note: This field may return null, indicating that no valid values can be obtained.
ActionRisk	String	No	Action risk level. 1: low-risk; 2: medium-risk; 3: high-risk. Note: This field may return null, indicating that no valid values can be obtained.

TemplateListItem

Template library list information

Used by actions: DescribeTemplateList.

Name	Type	Description
TemplateId	Integer	Template library ID
TemplateTitle	String	Template library name
TemplateDescription	String	Template library description
TemplateTag	String	Template library tag Note: This field may return null, indicating that no valid values can be obtained.
TemplateIsUsed	Integer	Template library status. 1: in use; 2: not in use.

TemplateCreateTime	String	Template library creation time
TemplateUpdateTime	String	Template library update time
TemplateUsedNum	Integer	Number of tasks associated with the template library
TemplateSource	Integer	Template library source. 0: self-built; 1: recommended by experts. Note: This field may return null, indicating that no valid values can be obtained.

TemplateMonitor

Monitoring metrics

Used by actions: DescribeTemplate.

Name	Type	Description
MonitorId	Integer	pk
MetricId	Integer	Monitoring metric ID Note: This field may return null, indicating that no valid values can be obtained.
ObjectTypeId	Integer	Object type ID of the monitoring metric
MetricName	String	Metric name
MetricChineseName	String	Metric in Chinese Note: This field may return null, indicating that no valid values can be obtained.

TemplatePolicy

Protection policy

Used by actions: DescribeTemplate.

Name	Type	Description
TemplatePolicyIdList	Array of String	Protection policy ID list

TemplatePolicyRule	String	Policy rules
TemplatePolicyDealType	Integer	Process method when the guardrail policy takes effect. 1: sequential execution, 2: pausing.

Error Codes

Last updated : 2024-09-25 10:28:12

Feature Description

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

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

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

Error Code List

Common Error Codes

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

	ensure your current local time matches the standard time.
AuthFailure.SignatureFailure	Invalid signature. Signature calculation error. Please ensure you've followed the signature calculation process described in the Signature API documentation.
AuthFailure.TokenFailure	Token error.
AuthFailure.UnauthorizedOperation	The request is not authorized. For more information, see the CAM documentation.
DryRunOperation	DryRun Operation. It means that the request would have succeeded, but the DryRun parameter was used.
FailedOperation	Operation failed.
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.
OperationDenied	The operation is rejected.
ResourcesSoldOut	Resources are sold out.