Direct Connect

API Document

Product Introduction

Contents

API Document
DC API 2017
Introduction

Last updated：2018-06-25 14:47:58

Welcome to Tencent Cloud Direct Connect (DC). You can call the APIs discussed in this document to work with direct connect gateways and direct connect tunnels. Before using these APIs, please make sure that you have thorough understanding of the DC product as well as its usages and billing methods.

In case of any conflict between the value or available range of any parameter provided in the "API Description" of this document and that provided on the Tencent Cloud official website, **the latter shall prevail**.

API Overview

Last updated：2018-06-25 14:26:17

Feature	Action ID
Query the list of physical direct connects	DescribeDirectConnects
Create a direct connect tunnel	CreateDirectConnectTunnel
Query the list of direct connect tunnels	DescribeDirectConnectTunnels

Physical Direct Connect
Querying Direct Connect List

Last updated：2018-06-25 17:52:55

Description

This API (DescribeDirectConnects) is used to query the list of direct connects.
Domain name for API request: dc.api.qcloud.com

Request

Syntax:

```
GET https://dc.api.qcloud.com/v2/index.php?Action=DescribeDirectConnects
&<Common request parameters>
&directConnectId=dc-kd7d06of
```

Request parameters

The following request parameter list only provides API request parameters. Common request parameters are required when the API is called. For more information, please see Common Request Parameters page. The Action field for this API is DescribeDirectConnects.

Parameter Name	Required	Type	Description
directConnectId	No	String	Direct connect ID, for example: dc-kd7d06of. All direct connects created by the developer are returned if this parameter is not specified.

Response

Response example:

```
{
"code": 0,
```

```
"message": "",
"data": [
{
}
]
}
```

Response parameters

Parameter Name	Type	Description
code	Int	Error code. 0: Successful; other values: Failed.
message	String	Error message.
data.n	Array	Returned array.
data.n.directConnectId	String	Direct connect ID assigned by the system, for example: dc-kd7d06of.
data.n.directConnectName	String	Direct connect name.
data.n.status	String	Direct connect status. 0: Running; 1: Expired; 2: Deleting; 3: Deleted; 11: Requesting; 12: Rejected; 13: To be paid; 14: Paid; 15: Building; 16: Building stopped.
data.n.provider	String	Direct connect provider.
data.n.portType	int	Interface type. 1: 100Base-T 100 MB electric interface; 2: 1000Base-T 1 GB electric interface; 3: 1000Base-LX 1 GB single-mode optical port (10 km); 4: 10GBase-LR 1 TB single-mode optical port (10 km)
data.n.accessPoints	String	Access point.
data.n.bandwidth	String	Direct connect bandwidth (in Mbps).
data.n.loaIssueTime	String	Direct Connect expiration time.

Response error codes

The following error codes only include the business logic error codes for this API. For additional common error codes, please see VPC Error Codes.

| Error Code | Description |

Example

Request

GET https://dc.api.qcloud.com/v2/index.php?Action=DescribeDirectConnects
&<common request parameters>
&directConnectId=dc-kd7d06of

Response


```
{
"code": 0,
"message": "",
"data": [
{
"directConnectId": "dc-3cavza1z",
"directConnectName": "Li Hao Test Direct Connect @123112",
"status": 15,
"provider": "China Telecom",
"portType": 8,
"accessPoints": "1",
"bandwidth": 200,
"loaIssueTime": "2017-09-02 15:41:00"
}
]
}
```

Direct Connect Channel
Creating Direct Connect Channel

Last updated：2018-06-25 14:27:12

API Description

This API (CreateDirectConnectTunnel) is used to create a direct connect tunnel.
Domain Name: `dc.api.qcloud.com`

Request Parameters

The following request parameter list only provides API request parameters. Common request parameters are required when the API is called. For more information, please see Common Request Parameters. The Action field for this API is `CreateDirectConnectTunnel` .

Parameter	Required	Type	Description
directConnectId	Yes	String	Direct connect ID, for example: dc-kd7d06of
directConnectTunnelName	Yes	String	Direct connect tunnel name
ownerAccount	No	String	Physical direct connect owner. Default is the current customer (physical direct connect owner). To share the direct connect, please enter the account ID of the developer who shares the direct connect.
networkType	No	Int	Network type. Default is 1. 1: VPC 0: BM network
region	Yes	String	Network region
vpcId	Yes	String	Unified ID of VPC/BM network
directConnectGatewayId	Yes	String	Direct connect gateway ID, for example: dcg-d545ddf

Parameter	Required	Type	Description
bandwidth	No	Int	Direct connect bandwidth (in Mbps) 0: Unlimited
routeMode	No	Int	0: BGP routing 1: Static routing Default is BGP routing.
bgpPeers.asn	No	String	BGP asn
bgpPeers.authKey	No	String	BGP key
routeFilterPrefixes.n.cidr	No	String	Peer IP address range
vlanId	Yes	Int	vlanId. Value range: 0-3000 0: Disable sub-API
localGatewayIp	No	String	localGatewayIp, Tencent's IP
peerGatewayIp	No	String	peerGatewayIp, user's IP
peeringSubnetMask	No	String	The mask of interconnection IP, which must be defined in the same subnet. It supports **24-30** bits, and is represented in dotted decimal notation, for example: 255.255.255.252
remark	No	String	Notes

Response Parameters

Parameter	Type	Description
code	Int	Error code 0: Successful Other values: Failed
message	String	Error message
directConnectTunnelId	String	Direct connect tunnel ID

Sample Code

Request example

```
GET https://dc.api.qcloud.com/v2/index.php?Action=CreateDirectConnectTunnel
&<<a href="https://cloud.tencent.com/doc/api/229/6976">common request parameters</a>>
&directConnectId=dc-kd7d06of
&directConnectTunnelName=baytest
&region=gz
&vpcId=vpc-abcdefg
&directConnectGatewayId=dcg-abcdefg
&vlanId=400
&routeMode=1
&routeFilterPrefixes.0.cidr=172.256.12.0/24
&routeFilterPrefixes.1.cidr=172.256.13.0/24
&localGatewayIp=169.254.64.1
&peerGatewayIp=169.254.64.2
&peeringSubnetMask=255.255.255.252
&remark=create
```

Response example

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

Modify Direct Connect Tunnel

Last updated：2018-06-25 15:04:51

Description

This API (ModifyDirectConnectTunnel) is used to modify the parameters of a direct connect tunnel.
Domain name: dc.api.qcloud.com

Request Parameters

Parameter	Type	Required	Description
directConnectTunnelId	String	Yes	Direct connect tunnel ID For example: dcx-abcdefgh
directConnectTunnelName	String	No	Direct connect tunnel name
peerAsn	Int	No	Customer's BGP asn
authKey	String	No	Customer's BGP key
routeFilterPrefixes.n.cidr	String	No	Customer's peer IP address range For example: 169.254.0.0/28
localGatewayIp	String	No	localGatewayIp, Tencent's interconnection IP
peerGatewayIp	String	No	peerGatewayIp, customer's interconnection IP
peeringSubnetMask	String	No	The mask of interconnection IP, which must be defined in the same subnet. It supports **24-30** bits, and is represented in dotted decimal notation, for example: 255.255.255.252

Response Parameters

Parameter	Type	Description

Parameter	Type	Description
code	Int	Error code 0: Successful Other values: Failed
message	String	Error message

Sample Code

Request example

```
GET https://dc.api.qcloud.com/v2/index.php?Action=DeleteDirectConnectTunnel
&<<a href="https://cloud.tencent.com/doc/api/229/6976">common request parameters</a>>
&directConnectTunnelId=dcx-abcdefgh
&bandwidth=10
&routeFilterPrefixes.0.cidr=172.256.12.0/24
&routeFilterPrefixes.1.cidr=172.256.13.0/24
&localGatewayIp=169.254.64.1
&peerGatewayIp=169.254.64.2
&peeringSubnetMask=255.255.255.252
```

Response example

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

Delete Direct Connect Tunnel

Last updated：2018-06-25 15:05:39

Description

This API (DeleteDirectConnectTunnel) is used to delete a direct connect tunnel.
Domain name: dc.api.qcloud.com

Request Parameters

Parameter	Type	Required	Description
directConnectTunnelId	String	Yes	Direct connect tunnel ID For example: dcx-abcdefgh

Response Parameters

Parameter	Type	Description
code	Int	Error code 0: Successful Other values: Failed
message	String	Error message

Sample Code

Request example

```
GET https://dc.api.qcloud.com/v2/index.php?Action=DeleteDirectConnectTunnel
&<<a href="https://cloud.tencent.com/doc/api/229/6976">common request parameters</a>>
&directConnectTunnelId=dcx-abcdefgh
```

Response example

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

Querying Direct Connect Channel List

Last updated：2018-06-28 15:22:36

Description

This API (DescribeDirectConnectTunnels) is used to query the list of direct connect tunnels.
Domain name: `dc.api.qcloud.com`

Request Parameters

The following request parameter list only provides the API request parameters. Common request parameters are required when the API is called. For more information, please see the Common Request Parameters page. The Action field for this API is `DescribeDirectConnectTunnels`.

Parameter	Required	Type	Description
directConnectId	No	String	Direct connect ID For example: dc-kd7d06of
directConnectTunnelId	No	String	Direct connect tunnel ID For example: dcx-kd7d0125

Response Parameters

Parameter	Type	Description
code	Int	Error code 0: Successful Other values: Failed
message	String	Error message
data.n	Array	Returned array
data.n.directConnectTunnelId	String	Direct connect tunnel ID assigned by the system For example: dcx-kd7d0125
data.n.directConnectTunnelName	String	Direct connect tunnel name

Parameter	Type	Description
data.n.directConnectId	String	Direct connect ID assigned by the system For example: dc-kd7d06of
data.n.ownerAccount	String	Developer account ID of the direct connect
data.n.networkType	Int	Network type. Default is 0. 0: VPC 1: BM network
data.n.region	String	Network region
data.n.vpcId	String	Unified ID of VPC/BM network
data.n.directConnectGatewayId	String	Direct connect gateway ID, for example: dcg-d545ddf
data.n.bandwidth	Int	Direct connect bandwidth (in Mbps)
data.n.routeMode	Int	0: BGP routing 1: Static routing Default is BGP routing.
data.n.bgpPeers.asn	string	BGP ASN
data.n.bgpPeers.authKey	String	BGP key
data.n.routeFilterPrefixes.n.cidr	String	Peer IP address range
data.n.status	Int	Status of the direct connect tunnel 0: Connected 1: Requesting 2: Configuring 6: Configured 20: Waiting for connection 21: Rejected
data.n.vlan	Int	vlan Id
data.n.localGatewayIp	String	Tencent's IP
data.n.peerGatewayIp	String	Customer's IP
data.n.peeringSubnetMask	String	The mask of interconnection IP For example: 255.255.255.252
data.n.remark	String	Notes

Sample Code

Request example

GET https://dc.api.qcloud.com/v2/index.php?Action=DescribeDirectConnectTunnels
&<common request parameters>
&directConnectId=dc-kd7d06of

Response example


```
{
"code": 0,
"message": "",
"data": [
{
"directConnectTunnelId": "dcx-2nakhj58",
"directConnectTunnelName": "barrytest2",
"directConnectId": "dc-5e8ak079",
"networkType": 1,
"ownerAccount": "",
"region": "gz",
"vpcId": "vpc-kx49lmyv",
"bandwidth": 0,
"routeMode": 0,
"bgpPeers": {
"asn": "10",
"authKey": "124545d"
},
"routeFilterPrefixes": [
{
"cidr": ""
}
],
"status": 1,
"vlan": 0,
"localGatewayIp":"169.254.64.1",
"peerGatewayIp":"169.254.64.2",
"peeringSubnetMask":"255.255.255.252",
"remark": ""
}
]
}
```

Accept Direct Connect Tunnel Application

Last updated：2018-06-25 15:07:05

Description

This API (AcceptDirectConnectTunnel) is used to accept the application for sharing the direct connect tunnel.
Domain name: dc.api.qcloud.com

Request Parameters

Parameter	Type	Required	Description
directConnectTunnelId	String	Yes	Direct connect tunnel ID For example: dcx-abcdefgh

Response Parameters

Parameter	Type	Description
code	Int	Error code 0: Successful Other values: Failed
message	String	Error message

Sample Code

Request example

```
GET https://dc.api.qcloud.com/v2/index.php?Action=AcceptDirectConnectTunnel
&<<a href="https://cloud.tencent.com/doc/api/229/6976">common request parameters</a>>
&directConnectTunnelId=dcx-abcdefgh
```

Response example

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

Delete Direct Connect Tunnel

Last updated：2018-06-25 17:52:18

Description

This API (RefuseDirectConnectTunnel) is used to reject the application for sharing a direct connect tunnel.
Domain name: dc.api.qcloud.com

Request Parameters

Parameter	Type	Required	Description
directConnectTunnelId	String	Yes	Direct connect tunnel ID For example: dcx-abcdefgh

Response Parameters

Parameter	Type	Description
code	Int	Error code 0: Successful Other values: Failed
message	String	Error message

Sample Code

Request example

```
GET https://dc.api.qcloud.com/v2/index.php?Action=RefuseDirectConnectTunnel
&<<a href="https://cloud.tencent.com/doc/api/229/6976">common request parameters</a>>
&directConnectTunnelId=dcx-abcdefgh
```

Response example

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

Calling Methods
Request Structure

Last updated：2018-07-03 11:53:35

A Tencent Cloud API call is completed by sending a request towards Tencent Cloud API server address while adding corresponding request parameters into the request according to API description. A Tencent Cloud API request is composed of server address, communication protocol, request method, request parameters and character encoding. Details are described below:

Service Address

The service connection address of Tencent Cloud APIs depends on the modules. For more information, please see the descriptions of each API.

Communication Protocol

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

Request Method

Tencent Cloud APIs support both POST and GET requests.

Note:

1. POST and GET requests cannot be used together. If you use GET method, parameters are obtained from Querystring.If you use POST method, parameters are obtained from Request Body, in which case parameters in Querystring will be ignored. The parameter formats in these request methods are the same. We use GET requests generally. But it is recommended to use POST if the parameter strings are too long.
2. If GET method is used, all request parameters need to be encoded with URL encoding. This is not required if POST method is used.
3. The maximum length for GET requests depends on browsers and different server configurations. For example, the length limit is 2K for traditional IE browser, 8K for Firefox. For long API requests

with a large number of parameters, it is recommended to use POST method to avoid request failure due to exceeded string length.
4. For POST requests, you need to pass parameters in the form of `x-www-form-urlencoded`, since the API on the cloud obtains request parameters from $_POST.

Request Parameters

Two types of parameters are required for each Tencent Cloud API request: common request parameters and API request parameters. Common request parameters are required for all APIs (for more information, please see Common Request Parameters section), while API request parameters are specific to each API. For more information, please see the "Request Parameters" description of each API.

Character Encoding

All requests for Tencent Cloud APIs and their returned results are encoded using UTF-8 character set.

Request Signature

Last updated：2018-07-06 17:07:44

Tencent Cloud API will verify every access request. Every request needs to contain a signature in common request parameters for identity verification. The signature is generated by your security credential, which include SecretID and SecretKey. If you do not have a security credential, you need to apply for it on Tencent Cloud official website. Otherwise, you cannot call Tencent Cloud APIs.

1. Applying for a Security Credential

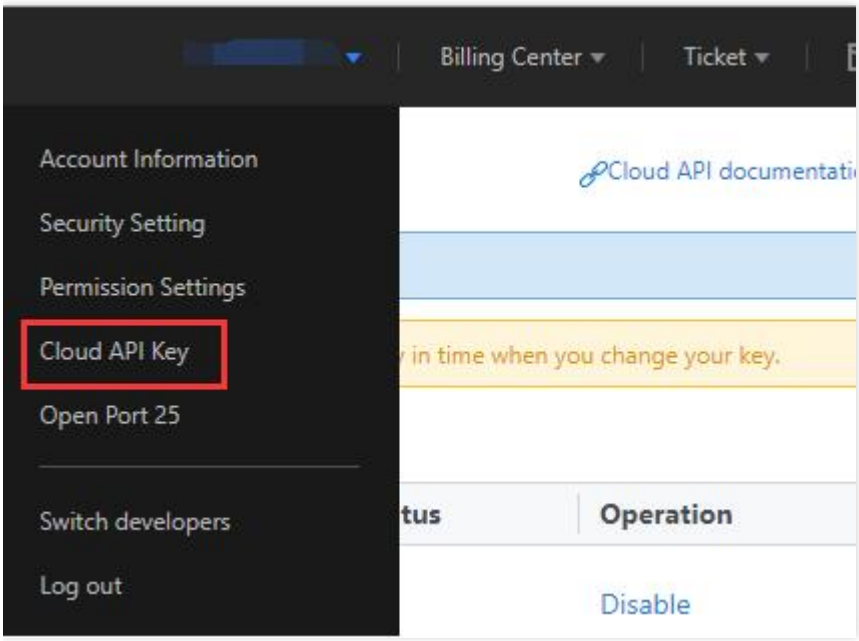
Before the first-time use of the cloud API, you need to apply for a security credential on the Tencent Cloud CVM console.
The security credential includes SecretId and SecretKey, where:

> SecretId: Used to indicate the API caller identity.
>
> SecretKey: The key used to encrypt the signed string and verify it on the server.
>
> Note: The API key is an important credential to build Tencent Cloud API requests. Any one with is key can get access to your Tencent Cloud resources. For the security of your property and services, please properly save and regularly change the key. Delete the old key in time when you change your key.

Do the followings to apply for a security credential:

1) Log in to the Tencent Cloud Console.

2) Select account name in the top right corner on the navigation bar, and choose "Cloud API Key" in the drop-down box to access the Cloud API key management page.



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

A developer account can have up to two SecretId/SecretKey pairs.
A QQ account that has been added by the developer as a sub-user can apply for different security credentials on different developer consoles.
The sub-user security credential currently can only call some cloud APIs.

2. Generating a signature

You can generate a signed string after obtaining the SecretId and SecretKey. The following details the procedure for generating a signed string.

Suppose the user's SecretId and SecretKey are:

SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA

SecretKey: Gu5t9xGARNpq86cd98joQYCN3Cozk1qA

Note: This is only an example. Please continue the subsequent operations with your own SecretId and SecretKey!

Here we use the DescribeInstances request as an example. When you call this API, the request parameters may be as follows:

Parameter	Description	Value
Action	Method name	DescribeInstances
SecretId	Key ID	AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA
Timestamp	Current timestamp	1465185768
Nonce	Random positive integer	11886
Region	Instance region	gz
instanceIds.0	Instance ID to be queried	ins-09dx96dg
offset	Offset value	0
limit	Maximum allowable output	20

According to the above table, only five public request parameters are listed here: Action, SecretId, Timestamp, Nonce and Region, rather than six parameters. In fact, the sixth parameter Signature is generated by other parameters (including the instruction request parameter). The specific steps are as follows:

2.1. Sorting parameters

First of all, sort all request parameters in ascending lexicographic order, which means intuitively they are in the same order as words in the dictionary, according to the order of the alphabet or numeration table: The first letter is considered first, then the second letter, and so on. You can use the relevant sorting function in the programming language, for example, the PHP ksort function, to achieve this feature. The results of sorting the parameters in the above example are as follows:

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

```
'offset' : 0,
}
```

When using other programming languages in development, you can also sort the parameters in the above example, as long as the same results are obtained.

2.2. Combining a request string

This step generates a request string.
Format the request parameters sorted in previous step to the form of "Parameter"="Value". For example, for the Action parameter, the name is "Action" and the value "DescribeInstances"; the formatted form is Action=DescribeInstances.
Note: 1. "Value" here is the original value rather than the URL-encoded value. 2. If the input parameter contains an underscore, you need to convert it to ".".

Then combine these parameters in formatted form with the "&" symbol and the resulting request string is:

```
Action=DescribeInstances&Nonce=11886&Region=gz&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA&Timestamp=1465185768&instanceIds.0=ins-09dx96dg&limit=20&offset=0
```

2.3. Combining a signed source string

This step generates the original string of a signature.
The signed source string consists of the following parameters:

1) Request mode: POST and GET modes are supported and the GET mode is used here. Note that all letters are uppercase.
2) Request host: The domain name for the DescribeInstances request is cvm.api.qcloud.com. The actual request domain name varies with the module to which the API belongs. For details, refer to the description of each API.
3) Request path: The request path for the cloud API is fixed at /v2/index.php.
4) Request string: It is the request string generated in the previous step.

The combination rule for signed source strings is:

```
Request method + Request server + Request path + ? + Request string
```

The combination results in this example are:

```
GETcvm.api.qcloud.com/v2/index.php?Action=DescribeInstances&Nonce=11886&Region=gz&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA&Timestamp=1465185768&instanceIds.0=ins-09dx9
```

```
6dg&limit=20&offset=0
```

2.4. Generating a signature

This step generates a signature.
First, the HMAC-SHA1 algorithm is used to complete signature with the **signed source string** obtained in the previous step, and then the generated signed string is encoded using Base64 to obtain the final signed string.

Specific codes are as follows (the PHP language as an example):

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3Cozk1qA';
$srcStr = 'GETcvm.api.qcloud.com/v2/index.php?Action=DescribeInstances&Nonce=11886&Region=gz&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA&Timestamp=1465185768&instanceIds.0=ins-09dx96dg&limit=20&offset=0';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

The resulting signed string is:

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

When using other programming languages in development, you can also verify the signed source string in the above example, as long as the same signed string is obtained.

3. Signature encoding

The generated signature cannot be used directly as a request parameter, and URL encoding is required. If the signature generated in the previous step is `NSI3UqqD99b/UJb4tbG/xZpRW64=` , then the URL-encoded string is `NSI3UqqD99b/UJb4tbG/xZpRW64=` . Therefore, the resulting signature request parameter is: `NSI3UqqD99b/UJb4tbG/xZpRW64=` , which will be used to generate the final request URL.

Note: If you use the GET request mode, then URL encoding is required for all request parameter values. Some language libraries will automatically encode the URL. Duplicate encoding will cause signature verification failure.

4. Authentication failure

When the authentication fails, possible errors are as follows:

Error Code	Type	Description
4100	Verification failed	Signature verification failed, Please ensure that the signature in your request parameters is calculated correctly. Or maybe the key state is incorrect. Make sure that the API key is valid and not disabled.
4101	Not authorized by the developer to access this API	The sub-user is not authorized to call this API. Please contact the developer for authorization. For details, refer to Authorization policies.
4102	Not authorized by the developer to access the resources operated by this API	The resource parameters that you request contain resources whose access is not authorized by the developer. Please check in the message field for the resource IDs that you do not have permissions to view. Please contact the developer for authorization. For details, refer to Authorization Policies.
4103	Only the developer SecretId can call this API currently.	The sub-user SecretID cannot call this API, and only the developer SecretID can.

Common Request Parameters

Last updated：2018-08-14 18:08:40

A complete Tencent Cloud API request requires two types of request parameters: common request parameters and API request parameters. This document describes 6 common request parameters used in Tencent Cloud API requests. For more information about API request parameters, please see API Request Parameters.

Common request parameters are required in every API. When developers use Tencent Cloud APIs to send requests, they should make sure that the requests carry these common request parameters. Otherwise, the requests will fail. The initial letter of each common request parameter is in uppercase so that it can be differentiated from API request parameters.

Common request parameters are as follows:

Note:
This document illustrates APIs specific to Tencent Cloud CVMs. For APIs specific to other Tencent Cloud products, please see the relevant documents.

Parameter Name	Description	Type	Required
Action	The name of the API for the desired operation. For example, when a Tencent Cloud CVM user calls the API Query Instance List, the Action parameter is DescribeInstances.	String	Yes
Region	Region parameter, which is used to identify the region to which the instance you want to work with belongs. For more information, please see Regions and Availability Zones, or use the API Query Region List. **Note:** 1. Unless otherwise specified in the API document, this parameter is required generally. 2. Some of the regions are under internal trial and only available to certain users.	String	No
Timestamp	The current UNIX timestamp that records the time at which the API request was initiated.	UInt	Yes
Nonce	A random positive integer that is used in conjunction with Timestamp to prevent replay attacks.	UInt	Yes

Parameter Name	Description	Type	Required
SecretId	The SecretId that is applied for under Cloud API Key to identify identity. A SecretId corresponds to a unique SecretKey, which is used to generate the request Signature. For more information, please see Signature Method.	String	Yes
Signature	Request signature, which is used to verify the validity of the request. The signature must be computed based on input parameters. For more information, please see Signature Method.	String	Yes
SignatureMethod	Signature method. Supported methods include HmacSHA256 and HmacSHA1. The HmacSHA256 method is used to verify signatures only when the parameter is specified as HmacSHA256. Otherwise, HmacSHA1 is used. For more information, please see Signature Method.	String	No
Token	The token used for the temporary certificate, which must be used together with a temporary key. No token is required for a long-term key.	String	No

Use Case

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

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

Response

Response Structure

Last updated：2018-06-25 14:51:42

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

Name	Type	Description	Required
code	Int	Error code on the result. 0: Successful; other values: Failed. For more information, please seeError Codes page.	
message	String	Request result	

For example:

Example requests that use common parameters:

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

Possible returned result is as follows:

```
{
"code":0,
"message": "success",
"instanceSet":
[{
"instanceId":"qcvm1234",
"cpu":1,
"mem":2,
"disk":20,
"bandwidth":65535,
"os":"centos_62_64",
"lanIp":"10.207.248.186",
"wanIp":null,
"status":0
}]
}
```

Error Codes

Last updated：2018-06-25 14:52:26

The error code in the response body summarizes the result of the calling and execution of a Tencent Cloud API.
Any error code other than 0 indicates the request is not properly executed. An error message describes the error in details. You can get the API execution result based on the error code.
On some terminals, such as browsers, `message` in Chinese is displayed in Unicode and needs to be decoded.

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

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

5200	Failed to purchase the resource	The resource purchase failed. This may be caused by unsupported instance configuration or insufficient resource.
5300	Failed to purchase the resource	The resource purchase failed because of insufficient balance.
5400	Part of operations performed successfully	Part of the batch operations have been performed successfully. For more information, please see the returned value of method.
5500	User failed to pass identity verification	Resource purchase failed because the user failed to pass identity verification.
6000	Internal error on the server	An internal error occurred on the server. Try again later or contact customer service for help.
6100	Not supported by the version	The API is not supported by this version or is under maintenance. Note: When this error occurs, first check whether the domain name of the API is correct. Different modules may have different domain names.
6200	API is temporarily unavailable	The API is under maintenance and is unavailable. Try again later.

Async Task API Response

Last updated：2018-06-25 14:50:47

1. Format of Returned Results for Ordinary Asynchronous Task APIs

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

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

2. Format of Returned Results for Batch Asynchronous Task APIs

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

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

For example:

```
{
"code":0,
"message": "success",
"detail":
```

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


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

Sample Code

Last updated：2018-08-27 17:56:45

Download SDK code

- **Sample Code (PHP)**
  - Go to GitHub for PHP SDK >>
  - Download PHP SDK >>
- **Sample Code (JAVA)**
  - Go to GitHub for JAVA >>
  - Download JAVA SDK >>
- **Sample Code (Python)**
  - Go to GitHub for Python >>
  - Download Python SDK >>
- **Sample Code (.NET)**
  - Go to GitHub for .NET SDK >>
  - Download .NET SDK >>

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

Sample Code (PHP)

```
<?php


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

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

/*Unless otherwise specified, all APIs other than MultipartUploadVodFile support GET and POST meth
```

```
ods.*/
$HttpMethod="GET";

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

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


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

/*The following two parameters are the ones specific to the API DescribeInstances and are used to query specific VM list.*/
$PRIVATE_PARAMS = array(
'instanceIds.0'=> 'qcvm00001',
'instanceIds.1'=> 'qcvm00002',
);


/******************************************************************************/


CreateRequest($HttpUrl,$HttpMethod,$COMMON_PARAMS,$secretKey, $PRIVATE_PARAMS, $isHttps);


function CreateRequest($HttpUrl,$HttpMethod,$COMMON_PARAMS,$secretKey, $PRIVATE_PARAMS, $isHttps)
{
$FullHttpUrl = $HttpUrl."/v2/index.php";

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

```
/**********************************Generate the original signature text.*********************************
* Combine the request method, URL and sorted request parameters into the following format to gene
rate the original signature text. In this example, the original signature text is as follows:
* GETcvm.api.qcloud.com/v2/index.php?Action=DescribeInstances&Nonce=345122&Region=gz
* &SecretId=AKIDz8krbsJ5yKBZQ·1pn74WFkmLPx3gnPhESA&Timestamp=1408704141
* &instanceIds.0=qcvm12345&instanceIds.1=qcvm56789
* ********************************************************************************/
$SigTxt = $HttpMethod.$FullHttpUrl."?";

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

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

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

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


/***************Combine the request strings together. The request parameters and the signature string
need to be encoded using urlencode.*******************/
$Req = "Signature=".urlencode($Signature);
foreach ($ReqParaArray as $key => $value)
{
$Req=$Req."&".$key."=".urlencode($value);
}

/*********************************Send the request*********************************/
if($HttpMethod === 'GET')
{
```

```
if($isHttps === true)
{
$Req="https://".$FullHttpUrl."?".$Req;
}
else
{
$Req="http://".$FullHttpUrl."?".$Req;
}

$Rsp = file_get_contents($Req);

}
else
{
if($isHttps === true)
{
$Rsp= SendPost("https://".$FullHttpUrl,$Req,$isHttps);
}
else
{
$Rsp= SendPost("http://".$FullHttpUrl,$Req,$isHttps);
}
}

var_export(json_decode($Rsp,true));
}

function SendPost($FullHttpUrl,$Req,$isHttps)
{

$ch = curl_init();
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $Req);

curl_setopt($ch, CURLOPT_URL, $FullHttpUrl);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
if ($isHttps === true) {
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);
}

$result = curl_exec($ch);

return $result;
}
```