

TCenter

Product Documentation





Contents

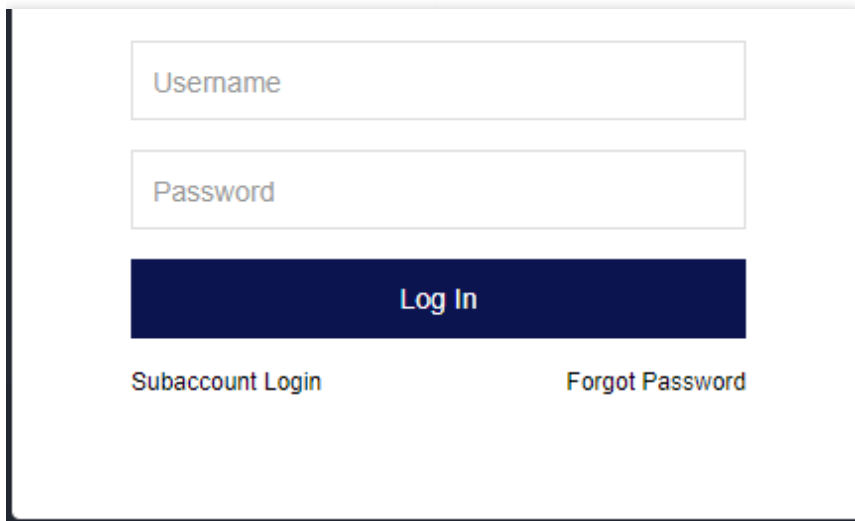
TCenter 3

- Operation Guide 3
 - Log-in 3
 - Account Information 5
 - Account Cancellation 7
 - Cloud Workload Protection Score 8
 - Security Settings 9
 - Resource Management 11
 - Filing Management 14
- FAQs 15
 - Common Log-in Issues 15
- API Documentation 16
 - Region Info(region) 16
 - version(2022-06-27) 16
 - API overview 16
 - CallingMethod 17
 - signaturev1 17
 - signaturev3 25
 - request 34
 - result 35
 - PublicParameter 38
 - Public Cloud Region APIs 40
 - DescribeRegions 40
 - DescribeRegionsAndZones 42
 - DescribeZones 44
 - Data structure 46
 - ErrorCode 51
 - Project directory management.(org) 53
 - version(2021-10-01) 53
 - API overview 53
 - CallingMethod 54
 - signaturev1 54
 - signaturev3 62
 - request 71
 - result 72
 - PublicParameter 75
 - Organization management APIs 77
 - AddOrganization 77
 - AddOrganizationMemberPolicy 79
 - DeleteOrganization 81
 - DeleteOrganizationMembers 83
 - DescribeOrganizationMembers 85
 - DescribeOrganizationNonMembers 87
 - DescribeOrganizationPoliciesTemplate 89
 - DescribeOrganizationProjects 91
 - DescribeOrganizationQuotas 93
 - DescribeOrganizationResources 95
 - DescribeOrganizations 97
 - ModifyOrganization 99
 - ModifyOrganizationMemberPolicy 101
 - ModifyOrganizationProjects 103
 - Data structure 105
 - ErrorCode 110

Operation Guide

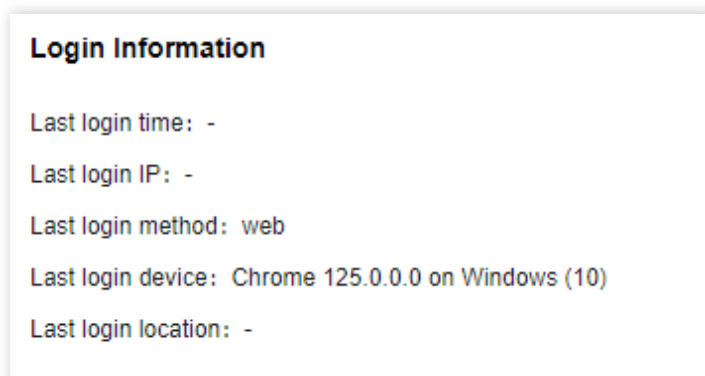
Log-in

Log-in



The login form is contained within a light gray box with a dark border. It features two input fields: 'Username' and 'Password', both with light gray placeholder text. Below these fields is a dark blue 'Log In' button with white text. At the bottom of the form, there are two links: 'Subaccount Login' and 'Forgot Password'.

- When a root account is created on the operator, the password needs to be reset upon the first login.
- During log-in, the system will detect the login location. If it's not a frequently used location, a mobile captcha is required for identity authentication.
- After log-in, the overview page can display the last login time, IP, method, and login district.



The 'Login Information' box is a light gray rectangle with a dark border. It contains the following text:

Login Information

Last login time: -

Last login IP: -

Last login method: web

Last login device: Chrome 125.0.0.0 on Windows (10)

Last login location: -

- After log-in, support configuring the session timeout for sub-users logging in at the tenant side. The default timeout for tenants is set to the time configured on the operator.

Note:

- After configuration, the session timeout for all tenants will be modified
- Tenants can still modify their respective timeout times later. The operator only has the function of resetting
- The operator's convergecloud_session_timeout value needs to be limited to between 30 and 1440, and must be an integer

Registration

When registering, the email address serves as the login account name.

Account Information

Overview of Account Information

Account information provides access to basic details of the login account, including account nickname, account ID, APP ID, and account permissions.

Feature Introduction

View Account Information

You can view the account name, account ID, and App ID of the login account.

Account Nickname: The account nickname can be edited and is not unique.

Account ID: The account ID is the unique identifier of the account.

App ID: App ID identifies the ownership of resources, and the App ID is the same for both the root account and sub-accounts.

Account Permissions

You can check whether the account is a root account or a sub-account. The root account has full permissions, while the permissions of sub-accounts are configured by the root account.

Qualification Approval

The Qualification Approval function restricts product purchases based on user verification status, and the verification status of different users does not affect each other. Only users who have passed the qualification approval have permission to purchase cloud services.

If a user's qualification verification status is not yet verified, verification is required first.

Specific Operations:

- Click Submit Qualification Certification and choose either personal or enterprise application.



- For personal application, enter real name, contact number, address, and relevant ID card information. For enterprise application, enter full company name, company contact person, company address, company contact number, and business license information.
- Click submit. After approval, the account will be eligible to purchase cloud resources.

Account Cancellation

Operation Scenarios

This document will walk you through the process of deregistering your cloud platform account.

Notes

Once an account is cancelled, it cannot be restored. To ensure the safety of your account assets, cloud resources, and account information, please backup all data associated with your account before submitting the account deregistration request.



Cloud Workload Protection Score

Log in to the tenant side management console. On the Overview page, check the Cloud Workload Protection Score card on the right to view Cloud Workload Protection Platform (CWP) information. For more details, see [CWP](#).

Security Settings

Overview of Security Settings

Security settings can manage all account-related security information under the login account, including passwords, phone numbers, emails, password rules, and log-in protection.

Feature Introduction

View General Information

Under the General Features, you can modify passwords, phone numbers, and emails.

- **Modify Password:** To modify the password, you must enter the old password first. The new password should be entered twice and must be the same. Password rules can be set by the root account in the user settings. Password rules may include: a combination of uppercase and lowercase letters, digits, and certain English punctuation marks (such as . / \ _ , excluding spaces), and should not contain the username. You can also set the minimum length of the password, the password expiration time, and the quantities of recently used passwords that cannot be repeated. Modifying the password through the console is suitable for scenarios where you remember the old password, the password is about to expire, or the password has been compromised. Modification restrictions: Requires the old password and the ability to log in to the console.
- **Modify Phone Number:** To modify the phone number, you only need to enter the new phone number and the verification code. Scene for modifying the phone number: The old phone number is no longer in use and needs to be replaced with a new one. Modification restrictions: Must be able to log in to the console.
- **Modify Email:** Modifying the email is similar to modifying the phone number, you only need to enter the new email and the phone number. Scene for modifying the email: The email is no longer in use and needs to be replaced with a new one. Modification restrictions: Must be able to log in to the console.

Sensitive Operation

Sensitive operations are used to set account protection policies and further enhance account security.

When performing sensitive operations, a SMS verification code is required.

Enable conditions: Already integrated with an internal enterprise SMS gateway.

Enable Log-in Protection: When enabled, users will be prompted for identity verification before performing sensitive operations to ensure the security of your account.

Disable Log-in Protection: After disabling, login does not require mobile verification.

Suitable scenes for enabling sensitive operations: log-in protection recommended for industries with high confidentiality requirements such as government, finance, and banks.

Log-in Protection

Users can set the type of log-in protection in the security settings, which includes three types of identity authentication: phone number verification code, virtual MFA device, and third-party MFA device.

Virtual MFA Device: Implemented based on the Google Authenticator algorithm. Users need to download the Google Authenticator app as a client to obtain a token. Enter the token during log-in to complete the identity authentication.

Third-party MFA Device: MFA authentication tool provided by a third-party vendor. Users need to bind their account to the third-party system and download the client app. Enter the token from the client app during log-in to complete the identity authentication.



Resource Management

Choose Cloud Services > Management & Audit > Resource Management to go to the resource management console.

Resource Management

Creating Projects

1. Choose Project Management > Create to create a project.
2. In the pop-up window, enter the project name and click Confirm.
3. After confirming the creation of the project, you will see the basic information on the current page, indicating that the project has been successfully created.
4. Click Edit on the right of the project name to modify the project name in the pop-up dialog box.

Project List

Add another project according to the steps for creating a project, and you can see the information of all added projects on the current page.

Projects			
Projects can help you resolve the complexity of resource grouping and authorization management within a single Cloud account.			
Create		Search by name or ID.	
Project ID/Name	Created By	Created At	Action
pr-e6410ea9			Quotas Delete
Total items: 1		20 / page	1 / 1 page

Deleting Projects

On the Project List page, click Delete on the row of the project to be deleted. In the pop-up box, click Confirm to pop up another dialog box indicating that the deletion is successful. Click Confirm again. The project is no longer in the project list, indicating that it has been deleted.

Resource Management

1. On the page of the added project, click the ID under Project ID/Name to manage resources for that project.
2. Transfer other project resources or tenant resources into the current project.
3. Click Transfer in Resources, and a pop-up window will appear where you can choose to transfer other project resources or tenant resources into the current project according to your needs. After verification, click Confirm to obtain the information of the transferred project. Here, transferring tenant resources is used as an example.
4. The newly added resources can be managed via Manage based on the actual situation.
5. To transfer the resources of the current project to another project, select the resources to be transferred and click Transfer Resources. In the pop-up window, select the target project and click Confirm. A success message will appear once the transfer is complete.

Member Management

1. Choose Cloud Access Management > User Management > Users to create a user.
2. Choose Cloud Services > Management & Audit > Resource Management > Project Management, and click the ID under Project ID/Name to manage members for that project.

3. Add members and assign permissions.

Step 1: Select a user — choose an existing user from the list to add to the current project.

Step 2: Grant permissions through policies — configure different permissions according to the required roles.

Step 3: Confirm completion.

4. Remove.

After adding members and assigning permissions, you can remove authorized members if you wish to revoke their access.

Quota Management

1. On the Project Management page, perform Quota Management operations on the added project.
2. On this page, you can perform operations such as Add Quota Item, Query, and Reset. You can

download a quota template, fill in the relevant data, and then import it for quota configuration.

3. Choose Cloud Services > Management & Audit > Resource Management > Project Management, and click the ID under Project ID/Name to perform quota management, query, and reset operations on the project.

Project Directory Management

Creating Project Directories

1. Choose Cloud Services > Management & Audit > Resource Management to go to the resource management console.
2. Choose Project Directory Management > Create to create a project directory.
3. In the pop-up window, select what the project directory is affiliated to, enter the project directory name, and click Confirm.
4. Click Edit on the right of the project directory name to modify the project directory name in the pop-up dialog box.

Resource Management

1. Click Manage on the right of the basic information area of the Directory page to separately view Project View, Resource View, and Project Quota View.



2. Click in the upper right corner to download the view list.

Filing Management

Financial Zone group cloud users need to perform domain name registration through the public cloud filing system. You can copy the filing code token from the Financial Zone and verify it in the public cloud system. This document describes the detailed operation for generating filing code in the Financial Zone.

limitation factor

This operation is only applicable to Financial Zone users.

Generate Filing Code

1. Log in to the Financial Zone console.
2. In the upper right corner of the page, select Account > Filing Management to enter the Filing Management page.
3. On the Filing Management page, click Generate Authorization Code.
4. Select Cloud Resource Type, Authorized Account, and set Remark information.
5. Click OK.
6. After the filing code is generated, click Copy in the operation column. Once copied, you can proceed to the public cloud for verification.

Delete Filing Code

Select the specified filing code, click Delete in the operation column, and confirm the deletion of the filing code.

FAQs

Common Log-in Issues

Common Log-in Issues

1. How to retrieve the password if I forget it?

- On the login page, click on the Forgot Password link.
- Retrieve the root account password.

2. In the log-in protection scene, the SMS verification code was not received

After logging in, if you encounter the following interface, it means that the administrator has enabled log-in protection.

After clicking Get Verification code, if you did not receive the code, it means that the phone number associated with your sub-account is incorrect. Please contact the administrator to modify the associated phone number for you.



API Documentation

Region Info(region)

version(2022-06-27)

API overview

API version

V3

Public Cloud Region APIs

Interface name	Interface function
DescribeRegions	DescribeRegions
DescribeRegionsAndZones	DescribeRegionsAndZones
DescribeZones	DescribeZones

CallingMethod signaturev1

Converge Cloud 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 API Key page to apply for them; otherwise, you cannot invoke the API.

1. Applying for Security Credentials

Before using the API for the first time, go to the 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 Converge Cloud Console.
2. Go to the Cloud 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: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

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	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	Current timestamp	1465185768
Nonce	Random positive integer	11886
Region	Region where the instance is located	luzon
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' : 'luzon',
  'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
  '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=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&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 host: the domain name of the request to view the list of instances (DescribeInstances) is cvm.convergecloud.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 Cloud 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.convergecloud.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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 = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';  
$srcStr = 'GETcvm.convergecloud.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit  
=20&Nonce=11886&Offset=0&Region=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPL  
E&Timestamp=1465185768&Version=2017-03-12';  
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));  
echo $signStr;
```

The final signature is:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

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 EliP9YW3pW28FpsEdkXt/+WcGeI= , the final signature string request parameter (Signature) is EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%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



Error code	Error description
AuthFailure.SecretIdNotFound	The key does not exist
AuthFailure.SignatureFailure	Signature error
AuthFailure.TokenFailure	Token error
AuthFailure.InvalidSecretId	Invalid key (not a Cloud API key type)

5. Signature Demo

When calling API 3.0, you are recommended to use the corresponding Converge 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
- Node

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.convergecloud.com/?](https://cvm.convergecloud.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=Elip9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12)

[Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=Elip9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12](https://cvm.convergecloud.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=Elip9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12)

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

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

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

Java

```
import java.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 CloudAPIDemo {
    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.convergecloud.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.convergecloud.com/?");
        // There is no requirement for the order of the parameters in the actual request URL.
        for (String k : params.keySet()) {
            // The request string needs to be URL encoded. As the Key is all in English letters, only the value
            // is URL encoded here.
            url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).append("&");
        }
        return url.toString().substring(0, url.length() - 1);
    }
}
```

```
}

public static void main(String[] args) throws Exception {
    TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap enables automatic sorting
    // A random number should be used when actually calling, for example: params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
    params.put("Nonce", 11886); // Common parameter
    // The current time of the system should be used when actually calling, for example: params.put("Timestamp", System.currentTimeMillis() / 1000);
    params.put("Timestamp", 1465185768); // Common parameter
    params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // Common parameter
    params.put("Action", "DescribeInstances"); // Common parameter
    params.put("Version", "2017-03-12"); // Common parameter
    params.put("Region", "luzon"); // 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), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE", "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 = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

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.convergecloud.com"
    data = {
        'Action': 'DescribeInstances',
        'InstanceIds.0': 'ins-09dx96dg',
        'Limit': 20,
        'Nonce': 11886,
        'Offset': 0,
        'Region': 'luzon',
        '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"])
    # n actual invocation would occur here, which may incur fees after success
    # resp = requests.get("https://" + endpoint, params=data)
    # print(resp.url)
```

signaturev3

Converge Cloud The API will authenticate each access request, meaning that each request must include signature information (Signature) in the public request parameters to verify the requester's identity. The signature information is generated by the security credentials, which include SecretId and SecretKey . If the user does not have security credentials, please apply for them on the Cloud API Key page, otherwise they cannot call the Cloud API interfaces.

1. Apply for security credentials.

Before using the Cloud API for the first time, please go to the Cloud API Key page to apply for a security credential.

Security credentials include SecretId and SecretKey:

- SecretId The security credentials are used to identify the identity of the API caller.
- SecretKey The secret key is used to encrypt and sign the string for server-side verification of the signature.
- **Users must strictly protect their security credentials and avoid disclosure them.**

The specific steps to apply for security credentials are as follows:

1. Log in to Converge Cloud Administration Center Console
2. Go to the console page for cloud API keys
3. On the Cloud API Key page, click New to create a pair of SecretId/SecretKey

Note: Developer accounts can have up to two pairs SecretId / SecretKey.

2. TC3-HMAC-SHA256 signature method

Note: For GET method, only Content-Type: application/x-www-form-urlencoded protocol format is supported. For POST method, Content-Type: application/json and Content-Type: multipart/form-data are currently supported. By default, all business interfaces are supported in json format. Only specific business interfaces are supported in multipart format. In this case, the interface cannot be called in json format. Refer to the specific business interface document description.

The following takes Cloud Virtual Machine querying Region-1 instance list as an example to introduce the signature calculation process step by step. We only use two parameters for the query instance list: Limit and Offset, called using the GET method.

Assume that the SecretId and SecretKey of the user are AKIDz 8krbsJ 5yKBZQpn 74WFkmLPx 3EXAMPLE and Gu 5t9xGARNpq 86cd 98joQYCN3EXAMPLE, respectively.

2.1. splice specification request string

Concatenate CanonicalRequest strings in the following format:

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

- HTTPRequestMethod: HTTP request methods (GET, POST), GET in this example;
- CanonicalURI: URI parameter, API 3.0 fixed to forward slash (/);
- CanonicalQueryString: Query string in the URL that initiates the HTTP request. For POST requests, it is fixed as an empty string. For GET requests, it is the question mark (?) in the URL. The following string content, this example value is: Limit=10&Offset=0. Note: CanonicalQueryString needs to be URL encoded.
- CanonicalHeaders: The header information of participating signature, at least including host and content-type, can also be added with customized header to improve the uniqueness and security of its own request. Splicing rules: 1) head key and value are uniformly converted to lowercase, and the first and last spaces are removed, and spliced according to the format of key:value\n;2) multiple heads are spliced according to the dictionary sorting of head key (lowercase). In this example:
content-type: application/x-www-form-urlencoded\nhost: cvm.convergecloud.com\n
- SignedHeaders: The header information involved in the signature indicates which headers participate in the signature of this request, and the header content contained in CanonicalHeaders is one-to-one corresponding. content-type and host are required headers. constructed as: 1) The head key is uniformly converted to lowercase; 2) Multiple header keys (lowercase) are concatenated according to dictionary sorting and separated by semicolons (;). In this case: content-type;host
- HashedRequestPayload: The hash value of the request body, calculated as Lowercase(HexEncode(Hash.SHA256(RequestPayload))), SHA256 hash the entire payload of the HTTP request body, then hexadecimal encoding, and finally converting the encoding string to lowercase letters. Note: For GET requests, RequestPayload is fixed as an empty string. For POST requests, RequestPayload is the payload of the HTTP request body.

According to the above rules, the resulting canonical request string in the example is as follows (for clarity, \nnewlines are replaced by a new line printed separately):

GET

```
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.convergecloud.com

content-type;host
e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. Concatenate the string to be signed

Concatenate the string to be signed in the following format:

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

- Algorithm: Signature algorithm, currently fixed as TC3-HMAC-SHA256;
- RequestTimestamp: Request timestamp, i.e. the X-TC-Timestamp value of the request header. The request in the example above is 1539084154;
- CredentialScope: Credential range in the format Date/service/tc3_request, containing date, requested service and termination string (tc3_request). **Date is the date of UTC standard time, the value needs to be consistent with UTC standard time converted by common parameter X-TC-Timestamp **;service is the product name, which must be consistent with the product domain name called, such as cvm. The above example request, the value is 2018-10-09/cvm/tc3_request;
- HashedCanonicalRequest: The hash value of the canonical request string obtained by concatenating the above steps is calculated by Lowercase(HexEncode(Hash.SHA256(CanonicalRequest))).

note:

1. Date must be calculated from timestamp X-TC-Timestamp and time zone UTC+0. Adding local time zone information to the system, such as East Zone 8, will cause calls to succeed during the day and night, but calls to fail in the early hours of the morning. Assuming the timestamp is 1551113065, the time in the East Eight District is 2019-02-26 00:44:25, but the calculated Date with UTC+0 should be 2019-02-25 instead of 2019-02-26.
2. Timestamp must be the current system time and ensure that the system time and standard time are synchronized. If the difference exceeds five minutes, it will fail. If it is not synchronized with the standard time for a long time, it may cause the request to fail after running for a period of time (return signature expiration error).

According to the above rules, the string to be signed in the example is as follows (for clarity,\nnewlines are

replaced by new lines printed):

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. calculate the signature

1. Compute the derived signature key, pseudocode as follows

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey: Original SecretKey;
- Date: That is, the Date field information in Credential, as shown in the example above, is 2018-10-09;
- Service: That is, the Service field information in Credential, as shown in the example above, is cvm;

2. Calculate the signature, pseudocode as follows

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

- SecretSigning: That is, the derivative signature key calculated above;
- StringToSign: That is, the character string to be signed calculated in step 2;

2.4. Splicing Authorization

Splice as follows Authorization:

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

- Algorithm: Signature method, fixed as TC3-HMAC-SHA256;
- SecretId: SecretId in key pair;
- CredentialScope: See above, Scope of Voucher;
- SignedHeaders: See above, header information for participating signatures;
- Signature: signature value



According to the above rules, the values obtained in the example are:

TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474

The final complete invocation information is as follows:

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

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474

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

Host: cvm.convergecloud.com

X-TC-Action: DescribeInstances

X-TC-Version: 2017-03-12

X-TC-Timestamp: 1539084154

X-TC-Region: luzon

3. signing failed

According to the actual situation, there are the following signature failure error codes, please handle according to the actual situation

error code	error description
AuthFailure.SignatureExpire	signature expiration
AuthFailure.SecretIdNotFound	key is not present
AuthFailure.SignatureFailure	signature error
AuthFailure.TokenFailure	token error
AuthFailure.InvalidSecretId	Illegal key (not cloud API key type)

4. Signature Demo

Java

```
import java.io.BufferedReader;  
import java.io.InputStream;
```

```
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.convergecloud.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencoded";
    private final static String CT_JSON = "application/json";
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(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(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.convergecloud.com";
        String region = "luzon";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        // Pay attention to the time zone, otherwise it is easy to make mistakes.
        sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
        String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

        // ***** Step 1: Assemble the specification request string *****
    }
}
```

```
String httpRequestMethod = "GET";
String canonicalUri = "/";
String canonicalQueryString = "Limit=10&Offset=0";
String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
String signedHeaders = "content-type;host";
String hashedRequestPayload = DigestUtils.sha256Hex("");
String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\n"
    + canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
System.out.println(canonicalRequest);

// ***** Step 2: Stitching the string to be signed *****
String credentialScope = date + "/" + service + "/" + "tc3_request";
String hashedCanonicalRequest = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
System.out.println(stringToSign);

// ***** Step 3: Calculate the signature *****
byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
byte[] secretService = sign256(secretDate, service);
byte[] secretSigning = sign256(secretService, "tc3_request");
String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
System.out.println(signature);

// ***** Step 4: Splicing 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("Host", host);
headers.put("Content-Type", CT_X_WWW_FORM_URL_ENCODED);
headers.put("X-TC-Action", action);
headers.put("X-TC-Timestamp", timestamp);
headers.put("X-TC-Version", version);
headers.put("X-TC-Region", region);
}
}
```

Python

```
# -*- coding: utf-8 -*-
```

```
import hashlib, hmac, json, os, sys, time
from datetime import datetime

# key parameters
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.convergecloud.com"
endpoint = "https://" + host
region = "luzon"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ***** Step 1: Assemble the specification request string *****
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
payload = ""
if http_request_method == "POST":
    canonical_querystring = ""
    ct = "json"
    payload = json.dumps(params)
canonical_headers = "content-type:application/%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: Stitching the string to be signed *****
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 *****
# Compute signature digest function
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: Splicing Authorization *****
authorization = (algorithm + " " +
                "Credential=" + secret_id + "/" + credential_scope + ", " +
                "SignedHeaders=" + signed_headers + ", " +
                "Signature=" + signature)
print(authorization)

# Common parameters added to request header
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

request

1. Service Address

Region refers to the geographical area of a physical data center. Converge Cloud ensures complete isolation between different regions, providing maximum stability and fault tolerance. To reduce access latency and improve download speed, it is recommended to choose the region closest to your customers.

You can view the complete region list by using the API interface "DescribeRegions".

2. Communications Protocol

All the Converge 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.

result

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 check your signature is correct."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

- The presence of the Error field indicates that the request has failed. A response for a failed request will include Error, Code and Message fields.

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

Common Error Codes

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

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



Error Code	Description
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	hHTTPS request method error. Only GET and POST requests are supported.
UnsupportedRegion	API does not support the requested region.



PublicParameter

Common parameters are parameters used to identify users and interface authentication purposes. If not necessary, these parameters are not described in the separate interface document of each interface, but these parameters need to be carried in each request before the request can be normally initiated.

signature method v3

When using TC3-HMAC-SHA256 signature method, the common parameters need to be uniformly placed in the HTTP Header request header, as follows:

parameter name	type	required	described
X-TC-Action	String	Yes	The name of the operation interface. Refer to the description of the public parameter Action in the input parameters of the interface documentation for the value. For example, the value for the query instance list interface of the cloud server is DescribeInstances.
X-TC-Region	String	Yes	region parameters, Data used to identify which region you wish to operate on. The region value accepted by the interface refers to the description of the input parameter common parameter Region in the interface document. Note: Some interfaces do not need to pass this parameter, which is specifically stated in the interface documentation. In this case, even passing this parameter will not take effect.
X-TC-Timestamp	Integer	Yes	The current UNIX timestamp, which records the time when the API request was initiated. for example 1529223702. Note: If the time difference from the server exceeds 5 minutes, it will cause a signature expiration error.
X-TC-Version	String	Yes	Version of API for operation. Refer to the description of the public parameter Version in the interface document for the value. For example, the version of the cloud server. 2017-03-12.
Authorization	String	Yes	HTTP The standard authentication header field is "Authorization". For example: TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 Here, - TC3-HMAC-SHA256: Signature method, currently fixed as this value, - Credential: Credential: Signature credential; AKIDEXAMPLE is the SecretId; Date is a date in UTC time, and this value must match the value of X-TC-Timestamp (a common parameter) in UTC time format; service is the name of the product/service, and is generally a domain name prefix. For example,cvm; - SignedHeaders: The headers that contains the authentication information; content-type and host are the required headers; - Signature: Signature digest.
X-TC-Token	String	No	The token used for temporary certificates needs to be used in conjunction with a temporary key. Temporary keys and tokens need to be obtained by calling the access management service interface. Long-term keys do not require tokens.

signature method v1



When using HmacSHA1 and HmacSHA256 signature methods, common parameters need to be placed uniformly in the request string, as follows

parameter name	type	required	described
Action	String	Yes	The name of the interface to be operated. The value refers to the explanation of the public parameter <code>Action</code> in the interface documentation. For example, for the cloud server's query instance list interface, the value is <code>DescribeInstances</code> .
Region	String	Yes	The region parameter is used to identify which region's data you want to operate on. The region value accepted by the interface is specified in the input parameter of the public parameter <code>Region</code> in the interface documentation. Note: Some interfaces do not require this parameter to be passed, which will be specifically mentioned in the interface documentation. Even if this parameter is passed, it will not take effect in this case.
Timestamp	Integer	Yes	The current UNIX timestamp, which can record the time when the API request is initiated. For example, 1529223702. If the difference between the current time is too large, it will cause a signature expiration error.
Nonce	Integer	Yes	A random positive integer combined with <code>Timestamp</code> is used to prevent replay attacks.
SecretId	String	Yes	The <code>SecretId</code> is an identifier applied for the cloud API key to identify the user's identity. A <code>SecretId</code> corresponds to a unique <code>SecretKey</code> , and the <code>SecretKey</code> is used to generate the request signature <code>Signature</code> .
Signature	String	Yes	Request signature is used to verify the legitimacy of this request. Users need to calculate it based on the actual input parameters. The specific calculation method can be found in the interface authentication documentation.
Version	String	Yes	The version of the API being operated on. The value refers to the description of the public parameter <code>Version</code> in the interface documentation. For example, the version of the cloud server. 2017-03-12.
SignatureMethod	String	No	Signature method, currently supports HmacSHA256 and HmacSHA1. HmacSHA256 algorithm is used to verify signatures only if this parameter is specified as HmacSHA 256, otherwise HmacSHA1 is used to verify signatures.
Token	String	No	Token used for temporary certificate needs to be used together with temporary key. The temporary key and Token need to be obtained from the Access Management service invocation interface. Long-term keys do not require tokens.

zone list

A region (Region) refers to the geographical area of a physical data center. Converge Cloud Translation: To deliver verification that completely isolates different regions, ensuring maximum stability and fault tolerance between regions. In order to reduce access latency and improve download speed, it is recommended that you choose the region closest to your customers.

You can view the complete region list by using the API interface Query Region List.



Public Cloud Region APIs

DescribeRegions

1. Interface Description

Interface request domain name: region.api3.convergecloud.com.

DescribeRegions

Default interface request rate limit: 20 times per second.

Interface update time: 2025-02-27 21:08:23.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see [public request parameters](#).

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeRegions
Version	Yes	No	String	Public parameter, The value of this interface: 2022-06-27
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
Product	No	No	String	Product Example: "VPC"
Scene	No	No	Int64	Scene Example: ""

3. Output Parameters

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



name		
TotalCount	Int64	TotalCount Example: 10
RegionSet	Array of RegionInfo	RegionSet Example: View
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

There is no error code related to business logic for this interface, For other error codes, see [public ErrorCode](#).



DescribeRegionsAndZones

1. Interface Description

Interface request domain name: region.api3.convergecloud.com.

DescribeRegionsAndZones

Default interface request rate limit: 20 times per second.

Interface update time: 2025-02-27 21:05:58.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see [public request parameters](#).

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeRegionsAndZones
Version	Yes	No	String	Public parameter, The value of this interface: 2022-06-27
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
Product	No	No	String	TCE product code. Example: "VPC"
Scene	No	No	Int64	Scene Example: ""
IsIgnoreWhiteList	No	No	Bool	IsIgnoreWhiteList (extension field). Example: false
ExcludeDisabledRegions	No	No	Bool	ExcludeDisabledRegions (extension field). Example: false



3. Output Parameters

Parameter name	Type	Description
TotalCount	Int64	TotalCount Example: 3
ResourceRegionSet	Array of ResourceRegionInfo	RegionSet Example: View
ProductType	String	ProductType Example: ""
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

There is no error code related to business logic for this interface, For other error codes, see [public ErrorCode](#).



DescribeZones

1. Interface Description

Interface request domain name: region.api3.convergecloud.com.

DescribeZones

Default interface request rate limit: 20 times per second.

Interface update time: 2025-02-27 21:10:17.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see [public request parameters](#).

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeZones
Version	Yes	No	String	Public parameter, The value of this interface: 2022-06-27
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
Product	No	No	String	Product Example: "VPC"
Scene	No	No	Int64	Scene Example: ""

3. Output Parameters

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



Parameter name	Type	Description
TotalCount	Int64	TotalCount Example: 10
ZoneSet	Array of ZoneInfo	ZoneSet Example: View
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

There is no error code related to business logic for this interface, For other error codes, see [public ErrorCode](#).



Data structure

RegionInfo

RegionInfo

Referenced by the following interfaces: DescribeRegions

name	required	NULL allowed	type	description
Region	no	no	String	Region Example: "hqtest-ops"
RegionName	no	no	String	RegionName Example: "Small-scale dual-AZ integrated test environment"
RegionState	no	no	String	RegionState Example: "AVAILABLE"
RegionTypeMC	no	no	String	RegionTypeMC Example: ""
LocationMC	no	no	String	LocationMC Example: "South"
RegionNameMC	no	no	String	RegionNameMC Example: "Small-scale dual-AZ integrated test environment"
RegionIdMC	no	no	String	RegionIdMC Example: "60000052"

ResourceRegionInfo

ResourceRegionInfo

Referenced by the following interfaces: DescribeRegionsAndZones

name	required	NULL allowed	type	description
Region	no	yes	String	Region



name	required	NULL allowed	type	description
				Example: "Region1"
RegionId	no	yes	String	RegionId Example: "60000052"
RegionName	no	yes	String	RegionName Example: "Region1"
RegionArea	no	yes	String	RegionArea Example: "Region1"
RegionState	no	yes	String	RegionState Example: "UNAVAILABLE"
Location	no	yes	String	Location Example: "South"
LocationEn	no	yes	String	LocationEn Example: ""
LocationJp	no	yes	String	LocationJp Example: ""
LocationKo	no	yes	String	LocationKo Example: ""
RegionNameEn	no	yes	String	RegionNameEn Example: "hqtest-ops"
RegionNameJp	no	yes	String	RegionNameJp Example: ""
RegionNameKo	no	yes	String	RegionNameKo Example: "Small-scale dual-AZ integrated test environment"
RegionShortName	no	yes	String	RegionShortName Example: "Small-scale dual-AZ integrated test environment"
InnerDomainName	no	yes	String	InnerDomainName Example: ""
OuterDomainName	no	yes	String	OuterDomainName Example: ""



name	required	NULL allowed	type	description
RegionType	no	yes	String	RegionType Example: ""
Weight	no	yes	Int64	Weight Example: 0
OnlineState	no	yes	String	OnlineState Example: ""
ProductWhiteList	no	yes	String	ProductWhiteList Example: ""
RegionTypeMC	no	yes	Int64	RegionTypeMC Example: 0
ResourceZoneSet	no	yes	Array of ResourceZoneInfo	ResourceZoneSet Example: View

ResourceZoneInfo

ResourceZoneInfo

Referenced by the following interfaces: DescribeRegionsAndZones

name	required	NULL allowed	type	description
Zone	no	yes	String	Zone Example: "hqtest-ops-1"
ZoneName	no	yes	String	ZoneName Example: "Zone 1 of small-scale dual-AZ integrated test environment"
ZoneState	no	yes	String	ZoneState Example: "UNAVAILABLE"
ZoneId	no	yes	String	ZoneId Example: "60520001"
ZoneNameEn	no	yes	String	ZoneNameEn Example: hqtest-ops-1



name	required	NULL allowed	type	description
ZoneNameJp	no	yes	String	ZoneNameJp Example: ""
ZoneNameKo	no	yes	String	ZoneNameKo Example: ""
WhiteList	no	yes	String	WhiteList Example: ""
ProductWhiteList	no	yes	String	ProductWhiteList Example: ""
Weight	no	yes	Int64	Weight Example: 0
SaleType	no	yes	String	SaleType Example: ""
MachineRoomType	no	yes	String	MachineRoomType Example: ""
LifeState	no	yes	String	LifeState Example: ""
OnlineState	no	yes	String	OnlineState Example: ""
ParentZone	no	yes	String	ParentZone Example: ""
ParentZoneId	no	yes	String	ParentZoneId Example: ""
ParentZoneName	no	yes	String	ParentZoneName Example: ""
ZoneType	no	yes	String	ZoneType Example: ""

ZoneInfo

ZoneInfo

Referenced by the following interfaces: DescribeZones



name	required	NULL allowed	type	description
Zone	no	yes	String	Zone Example: "hqtest-ops-1"
ZoneName	no	yes	String	ZoneName Example: "Zone 1 of small-scale dual-AZ integrated test environment"
ZoneState	no	yes	String	ZoneState Example: "AVAILABLE"
ZoneId	no	yes	String	ZoneId Example: "60520001"
ZoneIdMC	no	yes	String	ZoneIdMC Example: "60520001"
ParentZone	no	yes	String	ParentZone Example: ""
ParentZoneId	no	yes	String	ParentZoneId Example: ""
ParentZoneName	no	yes	String	ParentZoneName Example: ""
ZoneType	no	yes	String	ZoneType Example: ""
MachineRoomType	no	yes	String	MachineRoomType Example: ""



ErrorCode

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 codes	Description
AuthFailure.InvalidSecretId	Invalid key (not a Cloud 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.



Error codes	Description
AuthFailure.UnauthorizedOperation	The request does not have CAM authorization.
DryRunOperation	DryRun Operation. It means that the request would have succeeded, but the DryRun parameter was used.
FailedOperation	Operation failed.
InternalError	Internal error.
InvalidAction	The API does not exist.
InvalidParameter	Incorrect parameter.
InvalidParameterValue	Invalid parameter value.
LimitExceeded	Quota limit exceeded.
MissingParameter	A parameter is missing.
NoSuchVersion	The API version does not exist.
RequestLimitExceeded	The number of requests exceeds the frequency limit.
ResourceInUse	Resource is in use.
ResourceInsufficient	Insufficient resource.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	Resource is unavailable.
UnauthorizedOperation	Unauthorized operation.
UnknownParameter	Unknown parameter.
UnsupportedOperation	Unsupported operation.
UnsupportedProtocol	HTTPS request method error. Only GET and POST requests are supported.
UnsupportedRegion	API does not support the requested region.

Service Error Codes



Project directory management.(org)

version(2021-10-01)

API overview

API version

V3

Organization management APIs

Interface name	Interface function
AddOrganization	AddOrganization
AddOrganizationMemberPolicy	AddOrganizationMemberPolicy
DeleteOrganization	DeleteOrganization
DeleteOrganizationMembers	DeleteOrganizationMembers
DescribeOrganizationMembers	DescribeOrganizationMembers
DescribeOrganizationNonMembers	DescribeOrganizationNonMembers
DescribeOrganizationPoliciesTemplate	DescribeOrganizationPoliciesTemplate
DescribeOrganizationProjects	DescribeOrganizationProjects
DescribeOrganizationQuotas	DescribeOrganizationQuotas
DescribeOrganizationResources	DescribeOrganizationResources
DescribeOrganizations	DescribeOrganizations
ModifyOrganization	ModifyOrganization
ModifyOrganizationMemberPolicy	ModifyOrganizationMemberPolicy
ModifyOrganizationProjects	ModifyOrganizationProjects

CallingMethod signaturev1

Converge Cloud 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 API Key page to apply for them; otherwise, you cannot invoke the API.

1. Applying for Security Credentials

Before using the API for the first time, go to the 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 Converge Cloud Console.
2. Go to the Cloud 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: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

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	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	Current timestamp	1465185768
Nonce	Random positive integer	11886
Region	Region where the instance is located	luzon
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' : 'luzon',
  'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
  '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=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&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 host: the domain name of the request to view the list of instances (DescribeInstances) is cvm.convergecloud.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 Cloud 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.convergecloud.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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 = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';  
$srcStr = 'GETcvm.convergecloud.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit  
=20&Nonce=11886&Offset=0&Region=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPL  
E&Timestamp=1465185768&Version=2017-03-12';  
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));  
echo $signStr;
```

The final signature is:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

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 EliP9YW3pW28FpsEdkXt/+WcGeI= , the final signature string request parameter (Signature) is EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%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



Error code	Error description
AuthFailure.SecretIdNotFound	The key does not exist
AuthFailure.SignatureFailure	Signature error
AuthFailure.TokenFailure	Token error
AuthFailure.InvalidSecretId	Invalid key (not a Cloud API key type)

5. Signature Demo

When calling API 3.0, you are recommended to use the corresponding Converge 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
- Node

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.convergecloud.com/?`

`Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=Elip9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12`

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

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

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

Java

```
import java.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 CloudAPIDemo {
    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.convergecloud.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.convergecloud.com/?");
        // There is no requirement for the order of the parameters in the actual request URL.
        for (String k : params.keySet()) {
            // The request string needs to be URL encoded. As the Key is all in English letters, only the value
            // is URL encoded here.
            url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).append("&");
        }
        return url.toString().substring(0, url.length() - 1);
    }
}
```

```
}

public static void main(String[] args) throws Exception {
    TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap enables automatic sorting
    // A random number should be used when actually calling, for example: params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
    params.put("Nonce", 11886); // Common parameter
    // The current time of the system should be used when actually calling, for example: params.put("Timestamp", System.currentTimeMillis() / 1000);
    params.put("Timestamp", 1465185768); // Common parameter
    params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // Common parameter
    params.put("Action", "DescribeInstances"); // Common parameter
    params.put("Version", "2017-03-12"); // Common parameter
    params.put("Region", "luzon"); // 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), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE", "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 = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

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.convergecloud.com"
    data = {
        'Action': 'DescribeInstances',
        'InstanceIds.0': 'ins-09dx96dg',
        'Limit': 20,
        'Nonce': 11886,
        'Offset': 0,
        'Region': 'luzon',
        '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"])
    # n actual invocation would occur here, which may incur fees after success
    # resp = requests.get("https://" + endpoint, params=data)
    # print(resp.url)
```

signaturev3

Converge Cloud The API will authenticate each access request, meaning that each request must include signature information (Signature) in the public request parameters to verify the requester's identity. The signature information is generated by the security credentials, which include SecretId and SecretKey . If the user does not have security credentials, please apply for them on the Cloud API Key page, otherwise they cannot call the Cloud API interfaces.

1. Apply for security credentials.

Before using the Cloud API for the first time, please go to the Cloud API Key page to apply for a security credential.

Security credentials include SecretId and SecretKey:

- SecretId The security credentials are used to identify the identity of the API caller.
- SecretKey The secret key is used to encrypt and sign the string for server-side verification of the signature.
- **Users must strictly protect their security credentials and avoid disclosure them.**

The specific steps to apply for security credentials are as follows:

1. Log in to Converge Cloud Administration Center Console
2. Go to the console page for cloud API keys
3. On the Cloud API Key page, click New to create a pair of SecretId/SecretKey

Note: Developer accounts can have up to two pairs SecretId / SecretKey.

2. TC3-HMAC-SHA256 signature method

Note: For GET method, only Content-Type: application/x-www-form-urlencoded protocol format is supported. For POST method, Content-Type: application/json and Content-Type: multipart/form-data are currently supported. By default, all business interfaces are supported in json format. Only specific business interfaces are supported in multipart format. In this case, the interface cannot be called in json format. Refer to the specific business interface document description.

The following takes Cloud Virtual Machine querying Region-1 instance list as an example to introduce the signature calculation process step by step. We only use two parameters for the query instance list: Limit and Offset, called using the GET method.

Assume that the SecretId and SecretKey of the user are AKIDz 8krbsJ 5yKBZQpn 74WFkmLPx 3EXAMPLE and Gu 5t9xGARNpq 86cd 98joQYCN3EXAMPLE, respectively.

2.1. splice specification request string

Concatenate CanonicalRequest strings in the following format:

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

- HTTPRequestMethod: HTTP request methods (GET, POST), GET in this example;
- CanonicalURI: URI parameter, API 3.0 fixed to forward slash (/);
- CanonicalQueryString: Query string in the URL that initiates the HTTP request. For POST requests, it is fixed as an empty string. For GET requests, it is the question mark (?) in the URL. The following string content, this example value is: Limit=10&Offset=0. Note: CanonicalQueryString needs to be URL encoded.
- CanonicalHeaders: The header information of participating signature, at least including host and content-type, can also be added with customized header to improve the uniqueness and security of its own request. Splicing rules: 1) head key and value are uniformly converted to lowercase, and the first and last spaces are removed, and spliced according to the format of key:value\n;2) multiple heads are spliced according to the dictionary sorting of head key (lowercase). In this example:
content-type: application/x-www-form-urlencoded\nhost: cvm.convergecloud.com\n
- SignedHeaders: The header information involved in the signature indicates which headers participate in the signature of this request, and the header content contained in CanonicalHeaders is one-to-one corresponding. content-type and host are required headers. constructed as: 1) The head key is uniformly converted to lowercase; 2) Multiple header keys (lowercase) are concatenated according to dictionary sorting and separated by semicolons (;). In this case: content-type;host
- HashedRequestPayload: The hash value of the request body, calculated as Lowercase(HexEncode(Hash.SHA256(RequestPayload))), SHA256 hash the entire payload of the HTTP request body, then hexadecimal encoding, and finally converting the encoding string to lowercase letters. Note: For GET requests, RequestPayload is fixed as an empty string. For POST requests, RequestPayload is the payload of the HTTP request body.

According to the above rules, the resulting canonical request string in the example is as follows (for clarity, \nnewlines are replaced by a new line printed separately):

GET

```
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.convergecloud.com

content-type;host
e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. Concatenate the string to be signed

Concatenate the string to be signed in the following format:

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

- Algorithm: Signature algorithm, currently fixed as TC3-HMAC-SHA256;
- RequestTimestamp: Request timestamp, i.e. the X-TC-Timestamp value of the request header. The request in the example above is 1539084154;
- CredentialScope: Credential range in the format Date/service/tc3_request, containing date, requested service and termination string (tc3_request). **Date is the date of UTC standard time, the value needs to be consistent with UTC standard time converted by common parameter X-TC-Timestamp **;service is the product name, which must be consistent with the product domain name called, such as cvm. The above example request, the value is 2018-10-09/cvm/tc3_request;
- HashedCanonicalRequest: The hash value of the canonical request string obtained by concatenating the above steps is calculated by Lowercase(HexEncode(Hash.SHA256(CanonicalRequest))).

note:

1. Date must be calculated from timestamp X-TC-Timestamp and time zone UTC+0. Adding local time zone information to the system, such as East Zone 8, will cause calls to succeed during the day and night, but calls to fail in the early hours of the morning. Assuming the timestamp is 1551113065, the time in the East Eight District is 2019-02-26 00:44:25, but the calculated Date with UTC+0 should be 2019-02-25 instead of 2019-02-26.
2. Timestamp must be the current system time and ensure that the system time and standard time are synchronized. If the difference exceeds five minutes, it will fail. If it is not synchronized with the standard time for a long time, it may cause the request to fail after running for a period of time (return signature expiration error).

According to the above rules, the string to be signed in the example is as follows (for clarity,\nnewlines are

replaced by new lines printed):

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. calculate the signature

1. Compute the derived signature key, pseudocode as follows

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey: Original SecretKey;
- Date: That is, the Date field information in Credential, as shown in the example above, is 2018-10-09;
- Service: That is, the Service field information in Credential, as shown in the example above, is cvm;

2. Calculate the signature, pseudocode as follows

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

- SecretSigning: That is, the derivative signature key calculated above;
- StringToSign: That is, the character string to be signed calculated in step 2;

2.4. Splicing Authorization

Splice as follows Authorization:

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

- Algorithm: Signature method, fixed as TC3-HMAC-SHA256;
- SecretId: SecretId in key pair;
- CredentialScope: See above, Scope of Voucher;
- SignedHeaders: See above, header information for participating signatures;
- Signature: signature value



According to the above rules, the values obtained in the example are:

TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474

The final complete invocation information is as follows:

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

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474

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

Host: cvm.convergecloud.com

X-TC-Action: DescribeInstances

X-TC-Version: 2017-03-12

X-TC-Timestamp: 1539084154

X-TC-Region: luzon

3. signing failed

According to the actual situation, there are the following signature failure error codes, please handle according to the actual situation

error code	error description
AuthFailure.SignatureExpire	signature expiration
AuthFailure.SecretIdNotFound	key is not present
AuthFailure.SignatureFailure	signature error
AuthFailure.TokenFailure	token error
AuthFailure.InvalidSecretId	Illegal key (not cloud API key type)

4. Signature Demo

Java

```
import java.io.BufferedReader;  
import java.io.InputStream;
```

```
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.convergecloud.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencoded";
    private final static String CT_JSON = "application/json";
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(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(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.convergecloud.com";
        String region = "luzon";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        // Pay attention to the time zone, otherwise it is easy to make mistakes.
        sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
        String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

        // ***** Step 1: Assemble the specification request string *****
    }
}
```

```
String httpRequestMethod = "GET";
String canonicalUri = "/";
String canonicalQueryString = "Limit=10&Offset=0";
String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
String signedHeaders = "content-type;host";
String hashedRequestPayload = DigestUtils.sha256Hex("");
String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\n"
    + canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
System.out.println(canonicalRequest);

// ***** Step 2: Stitching the string to be signed *****
String credentialScope = date + "/" + service + "/" + "tc3_request";
String hashedCanonicalRequest = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
System.out.println(stringToSign);

// ***** Step 3: Calculate the signature *****
byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
byte[] secretService = sign256(secretDate, service);
byte[] secretSigning = sign256(secretService, "tc3_request");
String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
System.out.println(signature);

// ***** Step 4: Splicing 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("Host", host);
headers.put("Content-Type", CT_X_WWW_FORM_URL_ENCODED);
headers.put("X-TC-Action", action);
headers.put("X-TC-Timestamp", timestamp);
headers.put("X-TC-Version", version);
headers.put("X-TC-Region", region);
}
}
```

Python

```
# -*- coding: utf-8 -*-
```



```
import hashlib, hmac, json, os, sys, time
from datetime import datetime

# key parameters
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.convergecloud.com"
endpoint = "https://" + host
region = "luzon"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ***** Step 1: Assemble the specification request string *****
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
payload = ""
if http_request_method == "POST":
    canonical_querystring = ""
    ct = "json"
    payload = json.dumps(params)
canonical_headers = "content-type:application/%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: Stitching the string to be signed *****
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 *****
# Compute signature digest function
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: Splicing Authorization *****
authorization = (algorithm + " " +
                "Credential=" + secret_id + "/" + credential_scope + ", " +
                "SignedHeaders=" + signed_headers + ", " +
                "Signature=" + signature)
print(authorization)

# Common parameters added to request header
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

request

1. Service Address

Region refers to the geographical area of a physical data center. Converge Cloud ensures complete isolation between different regions, providing maximum stability and fault tolerance. To reduce access latency and improve download speed, it is recommended to choose the region closest to your customers.

You can view the complete region list by using the API interface "DescribeRegions".

2. Communications Protocol

All the Converge 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.

result

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 check your signature is correct."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

- The presence of the Error field indicates that the request has failed. A response for a failed request will include Error, Code and Message fields.

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

Common Error Codes

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

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



Error Code	Description
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	hHTTPS request method error. Only GET and POST requests are supported.
UnsupportedRegion	API does not support the requested region.



PublicParameter

Common parameters are parameters used to identify users and interface authentication purposes. If not necessary, these parameters are not described in the separate interface document of each interface, but these parameters need to be carried in each request before the request can be normally initiated.

signature method v3

When using TC3-HMAC-SHA256 signature method, the common parameters need to be uniformly placed in the HTTP Header request header, as follows:

parameter name	type	required	described
X-TC-Action	String	Yes	The name of the operation interface. Refer to the description of the public parameter Action in the input parameters of the interface documentation for the value. For example, the value for the query instance list interface of the cloud server is DescribeInstances.
X-TC-Region	String	Yes	region parameters, Data used to identify which region you wish to operate on. The region value accepted by the interface refers to the description of the input parameter common parameter Region in the interface document. Note: Some interfaces do not need to pass this parameter, which is specifically stated in the interface documentation. In this case, even passing this parameter will not take effect.
X-TC-Timestamp	Integer	Yes	The current UNIX timestamp, which records the time when the API request was initiated. for example 1529223702. Note: If the time difference from the server exceeds 5 minutes, it will cause a signature expiration error.
X-TC-Version	String	Yes	Version of API for operation. Refer to the description of the public parameter Version in the interface document for the value. For example, the version of the cloud server. 2017-03-12.
Authorization	String	Yes	HTTP The standard authentication header field is "Authorization". For example: TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 Here, - TC3-HMAC-SHA256: Signature method, currently fixed as this value, - Credential: Credential: Signature credential; AKIDEXAMPLE is the SecretId; Date is a date in UTC time, and this value must match the value of X-TC-Timestamp (a common parameter) in UTC time format; service is the name of the product/service, and is generally a domain name prefix. For example,cvm; - SignedHeaders: The headers that contains the authentication information; content-type and host are the required headers; - Signature: Signature digest.
X-TC-Token	String	No	The token used for temporary certificates needs to be used in conjunction with a temporary key. Temporary keys and tokens need to be obtained by calling the access management service interface. Long-term keys do not require tokens.

signature method v1



When using HmacSHA1 and HmacSHA256 signature methods, common parameters need to be placed uniformly in the request string, as follows

parameter name	type	required	described
Action	String	Yes	The name of the interface to be operated. The value refers to the explanation of the public parameter <code>Action</code> in the interface documentation. For example, for the cloud server's query instance list interface, the value is <code>DescribeInstances</code> .
Region	String	Yes	The region parameter is used to identify which region's data you want to operate on. The region value accepted by the interface is specified in the input parameter of the public parameter <code>Region</code> in the interface documentation. Note: Some interfaces do not require this parameter to be passed, which will be specifically mentioned in the interface documentation. Even if this parameter is passed, it will not take effect in this case.
Timestamp	Integer	Yes	The current UNIX timestamp, which can record the time when the API request is initiated. For example, 1529223702. If the difference between the current time is too large, it will cause a signature expiration error.
Nonce	Integer	Yes	A random positive integer combined with <code>Timestamp</code> is used to prevent replay attacks.
SecretId	String	Yes	The <code>SecretId</code> is an identifier applied for the cloud API key to identify the user's identity. A <code>SecretId</code> corresponds to a unique <code>SecretKey</code> , and the <code>SecretKey</code> is used to generate the request signature <code>Signature</code> .
Signature	String	Yes	Request signature is used to verify the legitimacy of this request. Users need to calculate it based on the actual input parameters. The specific calculation method can be found in the interface authentication documentation.
Version	String	Yes	The version of the API being operated on. The value refers to the description of the public parameter <code>Version</code> in the interface documentation. For example, the version of the cloud server. 2017-03-12.
SignatureMethod	String	No	Signature method, currently supports HmacSHA256 and HmacSHA1. HmacSHA256 algorithm is used to verify signatures only if this parameter is specified as HmacSHA 256, otherwise HmacSHA1 is used to verify signatures.
Token	String	No	Token used for temporary certificate needs to be used together with temporary key. The temporary key and Token need to be obtained from the Access Management service invocation interface. Long-term keys do not require tokens.

zone list

A region (Region) refers to the geographical area of a physical data center. Converge Cloud Translation: To deliver verification that completely isolates different regions, ensuring maximum stability and fault tolerance between regions. In order to reduce access latency and improve download speed, it is recommended that you choose the region closest to your customers.

You can view the complete region list by using the API interface Query Region List.

Organization management APIs

AddOrganization

1. Interface Description

Interface request domain name: org.api3.convergecloud.com.

API for adding an organization

Default interface request rate limit: 20 times per second.

Interface update time: 2022-12-09 16:59:51.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see [public request parameters](#).

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: AddOrganization
Version	Yes	No	String	Public parameter, The value of this interface: 2021-10-01
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
ParentId	Yes	No	String	Parent organization ID. Example: root
OrgName	Yes	No	String	Organization name. The maximum length of the value is 64 characters. Example: org1

3. Output Parameters

Parameter name	Type	Description
OrgId	String	Organization ID. For example, you can set the value to org-7737f12a. Example: org-9842cb32
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.



4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see [public ErrorCode](#).

Error Code	Description
InternalError.ThirdPartServiceBadRequest	Failed to call the service.
InvalidParameter.EmptyParameter	There is an empty parameter.
InvalidParameter.OrganizationNameTooLong	Name too long.
UnauthorizedOperation.AuthFailedError	Authentication failed
InternalError.DatabaseError	DB error.



AddOrganizationMemberPolicy

1. Interface Description

Interface request domain name: org.api3.convergecloud.com.

API for adding a user to an organization and granting permissions to the user

Default interface request rate limit: 20 times per second.

Interface update time: 2021-11-11 15:10:44.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see [public request parameters](#).

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: AddOrganizationMemberPolicy
Version	Yes	No	String	Public parameter, The value of this interface: 2021-10-01
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
OrgId	Yes	No	String	Organization ID. Example: org-9842cb32
Uins	Yes	No	Array of Uint64	List of user UINs. Example: [111,112]
PolicyNames	Yes	No	Array of String	List of policy names. Example: ["policygen-20141112201913","policygen-20141112201914"]

3. Output Parameters

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



Parameter name	Type	Description
FailedUins	Array of AddOrgMemberPolicyRes	List of users that the system fails to add. Example: View
SuccessfulUins	Array of AddOrgMemberPolicyRes	List of users that the system successfully add. Example: View
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see [public ErrorCode](#).

Error Code	Description
InternalError.ThirdPartServiceBadRequest	Failed to call the service.

DeleteOrganization

1. Interface Description

Interface request domain name: org.api3.convergecloud.com.

API for deleting an organization. You cannot delete an organization if the organization has projects associated with the subordinate organizations.

Default interface request rate limit: 20 times per second.

Interface update time: 2022-12-09 17:00:47.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see [public request parameters](#).

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DeleteOrganization
Version	Yes	No	String	Public parameter, The value of this interface: 2021-10-01
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
OrgId	Yes	No	String	Organization ID. Example: org-9842cb32

3. Output Parameters

Parameter name	Type	Description
OrgId	String	Organization ID. Example: org-9842cb32



Parameter name	Type	Description
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see [public ErrorCode](#).

Error Code	Description
InternalError.ThirdPartServiceBadRequest	Failed to call the service.
InvalidParameter.EmptyParameter	There is an empty parameter.
UnauthorizedOperation.AuthFailedError	Authentication failed
InternalError.DatabaseError	DB error.
FailedOperation.OrganizationProjectNotEmpty	Error of still associated project.

DeleteOrganizationMembers

1. Interface Description

Interface request domain name: org.api3.convergecloud.com.

API for deleting an organization member

Default interface request rate limit: 20 times per second.

Interface update time: 2021-11-11 15:10:44.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see [public request parameters](#).

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DeleteOrganizationMembers
Version	Yes	No	String	Public parameter, The value of this interface: 2021-10-01
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
OrgId	Yes	No	String	Organization ID. Example: org-9842cb32
Uins	Yes	No	Array of Uint64	List of UINs. Example: [111,112]

3. Output Parameters

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



Parameter name	Type	Description
Uins	Array of UInt64	List of UINs. Example: [111,112]
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see [public ErrorCode](#).

Error Code	Description
InternalError.ThirdPartServiceBadRequest	Failed to call the service.



DescribeOrganizationMembers

1. Interface Description

Interface request domain name: org.api3.convergecloud.com.

API for querying an organization member

Default interface request rate limit: 20 times per second.

Interface update time: 2024-10-14 19:37:38.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see [public request parameters](#).

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeOrganizationMembers
Version	Yes	No	String	Public parameter, The value of this interface: 2021-10-01
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
PageNumber	No	No	UInt64	Number of the page to return. Example: 100
PageSize	No	No	UInt64	Items per page. Example: 100
Filter	No	No	Filter	Filter Example: View
OrgId	Yes	No	String	Organization ID. Example: org-9842cb32



3. Output Parameters

Parameter name	Type	Description
MemberSet	Array of OrganizationMember	Organization member. Example: View
TotalCount	UInt64	Total. Example: 100
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see [public ErrorCode](#).

Error Code	Description
InternalError.ThirdPartServiceBadRequest	Failed to call the service.



DescribeOrganizationNonMembers

1. Interface Description

Interface request domain name: org.api3.convergecloud.com.

API for querying non-organization members

Default interface request rate limit: 20 times per second.

Interface update time: 2024-10-14 19:32:17.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see [public request parameters](#).

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeOrganizationNonMembers
Version	Yes	No	String	Public parameter, The value of this interface: 2021-10-01
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
OrgId	Yes	No	String	Organization ID. Example: org-9842cb32
PageNumber	No	No	Uint64	Number of the page to return. Example: 100
PageSize	No	No	Uint64	Items per page. Example: 100
Filter	No	No	Filter	Filter Example: View

3. Output Parameters

Parameter name	Type	Description
MemberSet	Array of OrganizationNonMember	Non-organization member. Example: View
TotalCount	UInt64	Total. Example: 100
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see [public ErrorCode](#).

Error Code	Description
InternalError.ThirdPartServiceBadRequest	Failed to call the service.



DescribeOrganizationPoliciesTemplate

1. Interface Description

Interface request domain name: org.api3.convergecloud.com.

API for querying a general policy template of an organization

Default interface request rate limit: 20 times per second.

Interface update time: 2024-10-14 19:32:05.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see [public request parameters](#).

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeOrganizationPoliciesTemplate
Version	Yes	No	String	Public parameter, The value of this interface: 2021-10-01
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
PageNumber	No	No	Uint64	Number of the page to return. Example: 100
PageSize	No	No	Uint64	Items per page. Example: 100

3. Output Parameters

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



Parameter name	Type	Description
PolicySet	Array of Policy	Policy list Example: View
TotalCount	UInt64	Total. Example: 100
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see [public ErrorCode](#).

Error Code	Description
InternalError.ThirdPartServiceBadRequest	Failed to call the service.



DescribeOrganizationProjects

1. Interface Description

Interface request domain name: org.api3.convergecloud.com.

API for querying the project list of an organization

Default interface request rate limit: 20 times per second.

Interface update time: 2022-12-09 16:57:10.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see [public request parameters](#).

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeOrganizationProjects
Version	Yes	No	String	Public parameter, The value of this interface: 2021-10-01
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
PageNumber	No	No	Uint64	Number of pages. You can set the value to 1. Example: 100
PageSize	No	No	Uint64	Items per page. Example: 100
Filter	No	No	Filter	Filter Example: View
OrgId	Yes	No	String	Organization ID. Example: org-9842cb32



3. Output Parameters

Parameter name	Type	Description
ProjectSet	Array of Project	List of projects. Example: View
TotalCount	UInt64	Total. Example: 100
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see [public ErrorCode](#).

Error Code	Description
InternalError.ThirdPartServiceBadRequest	Failed to call the service.
UnauthorizedOperation.AuthFailedError	Authentication failed
InternalError.DatabaseError	DB error.
InvalidParameter.EmptyParameter	There is an empty parameter.

DescribeOrganizationQuotas

1. Interface Description

Interface request domain name: org.api3.convergecloud.com.

API for querying the project quota of an organization

Default interface request rate limit: 20 times per second.

Interface update time: 2023-03-07 14:51:47.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see [public request parameters](#).

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeOrganizationQuotas
Version	Yes	No	String	Public parameter, The value of this interface: 2021-10-01
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
OrgId	Yes	No	String	Organization ID. Example: org-9842cb32
PageNumber	No	No	UInt64	Number of the page to return. Example: 100
PageSize	No	No	UInt64	Items per page. Example: 100
Filter	No	No	ProjectQuotaFilter	Filter Example: View

3. Output Parameters

Parameter name	Type	Description
QuotaSet	Array of ProjectQuota	List of quotas. Example: View
TotalCount	UInt64	Total. Example: 100
ProductSet	Array of ProductInfo	Product Information List Example: View
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see [public ErrorCode](#).

Error Code	Description
InternalError.ThirdPartServiceBadRequest	Failed to call the service.
InvalidParameter.EmptyParameter	There is an empty parameter.
UnauthorizedOperation.AuthFailedError	Authentication failed
InternalError.DatabaseError	DB error.



DescribeOrganizationResources

1. Interface Description

Interface request domain name: org.api3.convergecloud.com.

API for querying the project resource list of an organization

Default interface request rate limit: 20 times per second.

Interface update time: 2024-10-14 19:31:52.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see [public request parameters](#).

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeOrganizationResources
Version	Yes	No	String	Public parameter, The value of this interface: 2021-10-01
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
OrgId	Yes	No	String	Organization ID. Example: org-9842cb32
PageNumber	Yes	No	Uint64	Number of the page to return. Example: 100
PageSize	Yes	No	Uint64	Entries per page. Example: 100
Filter	Yes	No	ProjectResourceFilter	Filter Example: View

3. Output Parameters

Parameter name	Type	Description
ResourceSet	Array of Resource	Resource list Example: View
TotalCount	UInt64	Total. Example: 100
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see [public ErrorCode](#).

Error Code	Description
InternalError.ThirdPartServiceBadRequest	Failed to call the service.
UnauthorizedOperation.AuthFailedError	Authentication failed
InternalError.DatabaseError	DB error.
InvalidParameter.EmptyParameter	There is an empty parameter.

DescribeOrganizations

1. Interface Description

Interface request domain name: org.api3.convergecloud.com.

API for querying an organization

Default interface request rate limit: 20 times per second.

Interface update time: 2022-12-09 17:00:25.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see [public request parameters](#).

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeOrganizations
Version	Yes	No	String	Public parameter, The value of this interface: 2021-10-01
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
Filter	No	No	DescribeOrganizationsFilter	Filter Example: View
MemberUin	No	No	String	Member UIN. Example: 100

3. Output Parameters

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



name		
OrgSet	Array of Organization	List of organizations. Example: View
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see [public ErrorCode](#).

Error Code	Description
InternalError.ThirdPartServiceBadRequest	Failed to call the service.
InvalidParameter.EmptyParameter	There is an empty parameter.
UnauthorizedOperation.AuthFailedError	Authentication failed
InternalError.DatabaseError	DB error.

ModifyOrganization

1. Interface Description

Interface request domain name: org.api3.convergecloud.com.

API for modifying an organization

Default interface request rate limit: 20 times per second.

Interface update time: 2022-12-09 17:00:10.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see [public request parameters](#).

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: ModifyOrganization
Version	Yes	No	String	Public parameter, The value of this interface: 2021-10-01
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
OrgId	Yes	No	String	Organization ID. Example: org-9842cb32
OrgName	Yes	No	String	Organization Name Example: org1

3. Output Parameters

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



Parameter name	Type	Description
OrgId	String	Organization ID. Example: org-9842cb32
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see [public ErrorCode](#).

Error Code	Description
InternalError.ThirdPartServiceBadRequest	Failed to call the service.
InvalidParameter.EmptyParameter	There is an empty parameter.
InvalidParameter.OrganizationNameTooLong	Name too long.
UnauthorizedOperation.AuthFailedError	Authentication failed
InternalError.DatabaseError	DB error.



ModifyOrganizationMemberPolicy

1. Interface Description

Interface request domain name: org.api3.convergecloud.com.

API for modifying a policy on organization members

Default interface request rate limit: 20 times per second.

Interface update time: 2021-11-11 15:10:44.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see [public request parameters](#).

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: ModifyOrganizationMemberPolicy
Version	Yes	No	String	Public parameter, The value of this interface: 2021-10-01
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
OrgId	Yes	No	String	Organization ID. Example: org-9842cb32
AccountUin	Yes	No	Uint64	Account UIN. Example: 100
PolicyNames	Yes	No	Array of String	Required policy list. Example: ["policygen-20141112201913","policygen-20141112201914"]

3. Output Parameters

Parameter name	Type	Description
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to



Parameter name	Type	Description
		be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see [public ErrorCode](#).

Error Code	Description
InternalError.ThirdPartServiceBadRequest	Failed to call the service.

ModifyOrganizationProjects

1. Interface Description

Interface request domain name: org.api3.convergecloud.com.

API for adding or removing organization projects.

Default interface request rate limit: 20 times per second.

Interface update time: 2021-11-11 15:10:44.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see [public request parameters](#).

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: ModifyOrganizationProjects
Version	Yes	No	String	Public parameter, The value of this interface: 2021-10-01
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
OrgId	Yes	No	String	Organization ID. Example: org-9842cb32
Operate	Yes	No	String	Operation type. Valid values: Add and Move. Example: Add
Projects	Yes	No	Array of String	Project that you want to add or remove. Example: ["pr-dcd34c11","pr-dcd34c12"]

3. Output Parameters

Parameter name	Type	Description
FailedProjects	Array of String	List of projects that the system fails to add or remove. Example: ["pr-dcd34c11","pr-dcd34c12"]
SuccessfulProjects	Array of String	List of projects that the system successfully add or remove. Example: ["pr-dcd34c11","pr-dcd34c12"]
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see [public ErrorCode](#).

Error Code	Description
InternalError.ThirdPartServiceBadRequest	Failed to call the service.



Data structure

Organization

Organization

Referenced by the following interfaces: DescribeOrganizations

name	required	NULL allowed	type	description
Id	no	no	UInt64	Id Example: 100
OrgId	no	no	String	Organization ID. Example: org-9842cb32
OrgName	no	no	String	Organization Name Example: org-9842cb32
CreatorUin	no	no	String	UIN of the creator. Example: 100
Creator	no	no	String	Created By Example: admin
CreateTime	no	no	String	Creation time. Example: 2014-08-03 12:00:00
Children	no	no	Array of Organization	Subordinate organization. Example: View

OwnedPolicies

Associated policy.

Referenced by the following interfaces: DescribeOrganizationMembers

name	required	NULL allowed	type	description
PolicyId	yes	no	UInt64	Policy ID. Example: 100
PolicyName	yes	no	String	Policy name Example: policygen-1
Description	yes	no	String	Policy description. Example: policydesc

DescribeOrganizationsFilter

Queries conditions for querying organizations.

Referenced by the following interfaces: DescribeOrganizations

name	required	NULL allowed	type	description
Level	no	no	UInt64	Level. Default value: 3. The value 3 specifies that the system searches organizations that meet the search conditions in level 1,2, and 3. Example: 3
Keyword	no	no	String	Search conditions Example: 100
OrgId	no	no	String	Organization ID. Example: org-9842cb32

Filter

Filter

Referenced by the following interfaces: DescribeOrganizationMembers, DescribeOrganizationNonMembers, DescribeOrganizationProjects

name	required	NULL allowed	type	description
Keyword	no	no	String	Filter Example: 100



name	required	NULL allowed	type	description
OrgIds	no	no	Array of String	List of organization IDs. Example: ["org-9842cb32","org-9842cb33"]

OrganizationMember

Organization member.

Referenced by the following interfaces: DescribeOrganizationMembers

name	required	NULL allowed	type	description
Uin	yes	no	Uint64	User UIN. Example: 100
Name	yes	no	String	User Name Example: admin
OwnedPolicies	yes	no	Array of OwnedPolicies	Associated policy. Example: View
JoinTime	yes	no	Datetime	Time when a user joins an organization. Example: 2014-08-03 12:00:00

ProjectQuotaFilter

Quota query conditions.

Referenced by the following interfaces: DescribeOrganizationQuotas

name	required	NULL allowed	type	description
ProductCode	no	no	String	Level-1 product. Example: p_cvm
OrgIds	no	no	Array of String	List of organization IDs. Example: ["org-9842cb32","org-9842cb33"]

OrganizationNonMember

Non-organization member.

Referenced by the following interfaces: DescribeOrganizationNonMembers

name	required	NULL allowed	type	description
Uin	yes	no	Uint64	User UIN. Example: 100
Name	yes	no	String	User Name Example: admin

Project

Project

Referenced by the following interfaces: DescribeOrganizationProjects

name	required	NULL allowed	type	description
ProjectId	yes	no	String	Project ID Example: pr-9842cb32
ProjectName	yes	no	String	Project Name Example: pr1
Creator	yes	no	String	Created By Example: admin
CreateTime	yes	no	Datetime	Creation time. Example: 2014-08-03 12:00:00
CreatorUin	yes	no	Uint64	Creator UIN. Example: 100



name	required	NULL allowed	type	description
OrgId	yes	no	String	Organization ID — non-empty value indicates the user is already in another organization. Example: org-9842cb32
OrgName	yes	no	String	Organization Name Example: org1
OrgOperator	yes	no	String	Operator for transferring a project to an organization. Example: admin
OrgOperationTime	yes	no	Datetime	Time when a project is transferred to an organization. Example: 2014-08-03 12:00:00

AddOrgMemberPolicyRes

Result of user addition and authorization within the organization.

Referenced by the following interfaces: AddOrganizationMemberPolicy

name	required	NULL allowed	type	description
Uin	yes	no	UInt64	User UIN. Example: 100
PolicyName	yes	no	String	Policy name Example: policy1

Policy

Organization policy.

Referenced by the following interfaces: DescribeOrganizationPoliciesTemplate

name	required	NULL allowed	type	description
PolicyName	yes	no	String	Policy name Example: policygen-1
Description	yes	no	String	Policy description. Example: policydesc

Resource

Resource

Referenced by the following interfaces: DescribeOrganizationResources

name	required	NULL allowed	type	description
ProductCode	yes	no	String	Product Code Example: p_cvm
ProductGroupName	yes	no	String	Product category name. Example: cvm
ProductName	yes	no	String	Product name. Example: cvm
ProjectId	yes	no	String	Project ID. Example: pr-dcd34c11
ProjectName	yes	no	String	Project Name Example: pr1
RegionId	yes	no	UInt64	Region ID Example: 5000001
RegionName	yes	no	String	Region name. Example: region1
RegionEnName	yes	no	String	English name of the region. Example: region1
ResourceId	yes	no	String	Resource ID. Example: ins-asd223
ResourceName	yes	no	String	Resource Name Example: ins1



name	required	NULL allowed	type	description
ResourceType	yes	no	String	Resource type. Example: cvm
ServiceType	yes	no	String	ServiceType Example: cvm

ProjectResourceFilter

Filter conditions for project resources.

Referenced by the following interfaces: DescribeOrganizationResources

name	required	NULL allowed	type	description
ProductCode	no	no	String	Product Code Example: p_cvm
OrgIds	no	no	Array of String	List of organization IDs. Example: ["org-9842cb32","org-9842cb33"]
Product	no	no	String	Product name. Example: cvm

ProductInfo

Product Information

Referenced by the following interfaces: DescribeOrganizationQuotas

name	required	NULL allowed	type	description
ProductCode	no	no	String	Product Code Example: p_cvm
ProductName	no	no	String	Product name. Example: cvm

ProjectQuota

Project quota.

Referenced by the following interfaces: DescribeOrganizationQuotas

name	required	NULL allowed	type	description
ProjectId	yes	no	String	Project ID. Example: pr-9842cb32
ProjectName	yes	no	String	Project Name Example: pr1
ProductName	yes	no	String	Product name. Example: cvm
ProductCode	yes	no	String	Product Code Example: p_cvm
SubProductCode	yes	yes	String	Level-2 product. Example: sp_cvm_vself2
BillingItemCode	yes	yes	String	Level-3 product. Example: v_cvm_cpu
SubBillingItemCode	yes	yes	String	Level-4 product. Example: sv_cvm_mem_s3
QuotaKey	yes	no	String	Quota key. Example: p_cvm###
QuotaName	yes	yes	String	Quota Item Name Example: u4e91u786cu76d8CBS-u5e26u5bbdu6027u80fd
QuotaValue	yes	no	String	Total Quota Example: 100
QuotaLeft	yes	no	Uint64	The remaining quota. Example: 100



name	required	NULL allowed	type	description
QuotaUsed	yes	no	Uint64	Used Quota Example: 100
CreateTime	yes	no	String	Creation time. Example: 2014-08-03 12:00:00
UpdateTime	yes	no	String	Update time. Example: 2014-08-03 12:00:00
SubProductName	yes	no	String	Subproduct Name Example: eip_hour
Unit	yes	no	String	Unit Example: 100
BillingItemName	yes	no	String	Product item Example: CPU
SubBillingItemName	yes	no	String	Product details Example: TDSQL for MySQL - I/O Edition - CPU - Single



ErrorCode

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 codes	Description
AuthFailure.InvalidSecretId	Invalid key (not a Cloud 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.

Error codes	Description
AuthFailure.UnauthorizedOperation	The request does not have CAM authorization.
DryRunOperation	DryRun Operation. It means that the request would have succeeded, but the DryRun parameter was used.
FailedOperation	Operation failed.
InternalError	Internal error.
InvalidAction	The API does not exist.
InvalidParameter	Incorrect parameter.
InvalidParameterValue	Invalid parameter value.
LimitExceeded	Quota limit exceeded.
MissingParameter	A parameter is missing.
NoSuchVersion	The API version does not exist.
RequestLimitExceeded	The number of requests exceeds the frequency limit.
ResourceInUse	Resource is in use.
ResourceInsufficient	Insufficient resource.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	Resource is unavailable.
UnauthorizedOperation	Unauthorized operation.
UnknownParameter	Unknown parameter.
UnsupportedOperation	Unsupported operation.
UnsupportedProtocol	HTTPS request method error. Only GET and POST requests are supported.
UnsupportedRegion	API does not support the requested region.

Service Error Codes

Error codes	Description
InternalError.ThirdPartServiceBadRequest	Failed to call the service.



Error codes	Description
InternalError.DatabaseError	DB error.
InvalidParameter.EmptyParameter	There is an empty parameter.
InvalidParameter.OrganizationNameTooLong	Name too long.
UnauthorizedOperation.AuthFailedError	Authentication failed
FailedOperation.OrganizationProjectNotEmpty	Error of still associated project.