Approval Management

Product Documentation

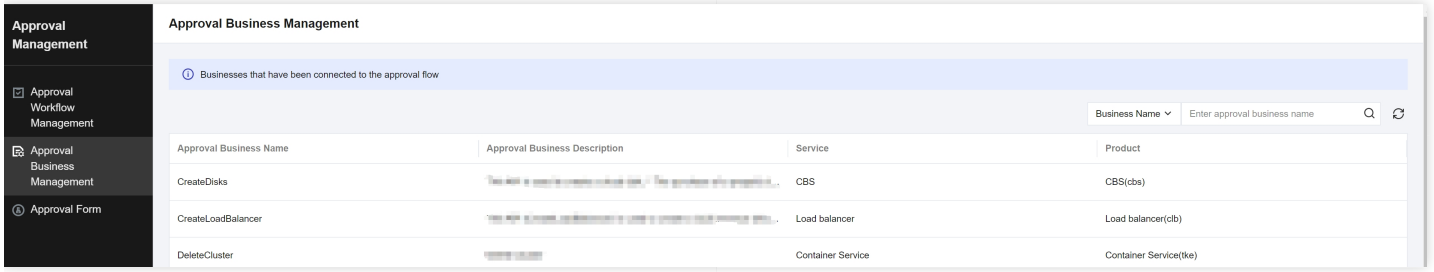
Contents

Approval Management 3
- Operation Guide 3
  - Managing Approval Businesses 3
  - Approval Process Management 4
- API Documentation 8
  - Approval flow service in the tenant console.(tapproval) 8
    - version(2022-05-18) 8
      - API overview 8
      - CallingMethod 9
        - signaturev1 9
        - signaturev3 17
        - request 26
        - result 27
        - PublicParameter 30
      - Review service APIs 32
        - BatchPerformApproval 32
        - BatchSubmitApproval 34
        - DeleteApprovalFlow 36
        - ModifyApprovalFlow 38
        - OperateFlowStatus 40
        - QueryApprovalFlowStatus 42
        - QueryCurrApprovalDetail 43
        - QueryCustomerApprovalDetail 45
        - QueryPendingApprovalDoc 47
        - SendApprovalReminder 49
        - SendApprovalVerifyCode 51
        - VerifyApprovalSmsCode 53
        - WithdrawApplication 55
      - Data structure 57
      - ErrorCode 63

Operation Guide
Managing Approval Businesses

Managing Approval Businesses

The approval business list displays the names of all businesses integrated with Approval Management.



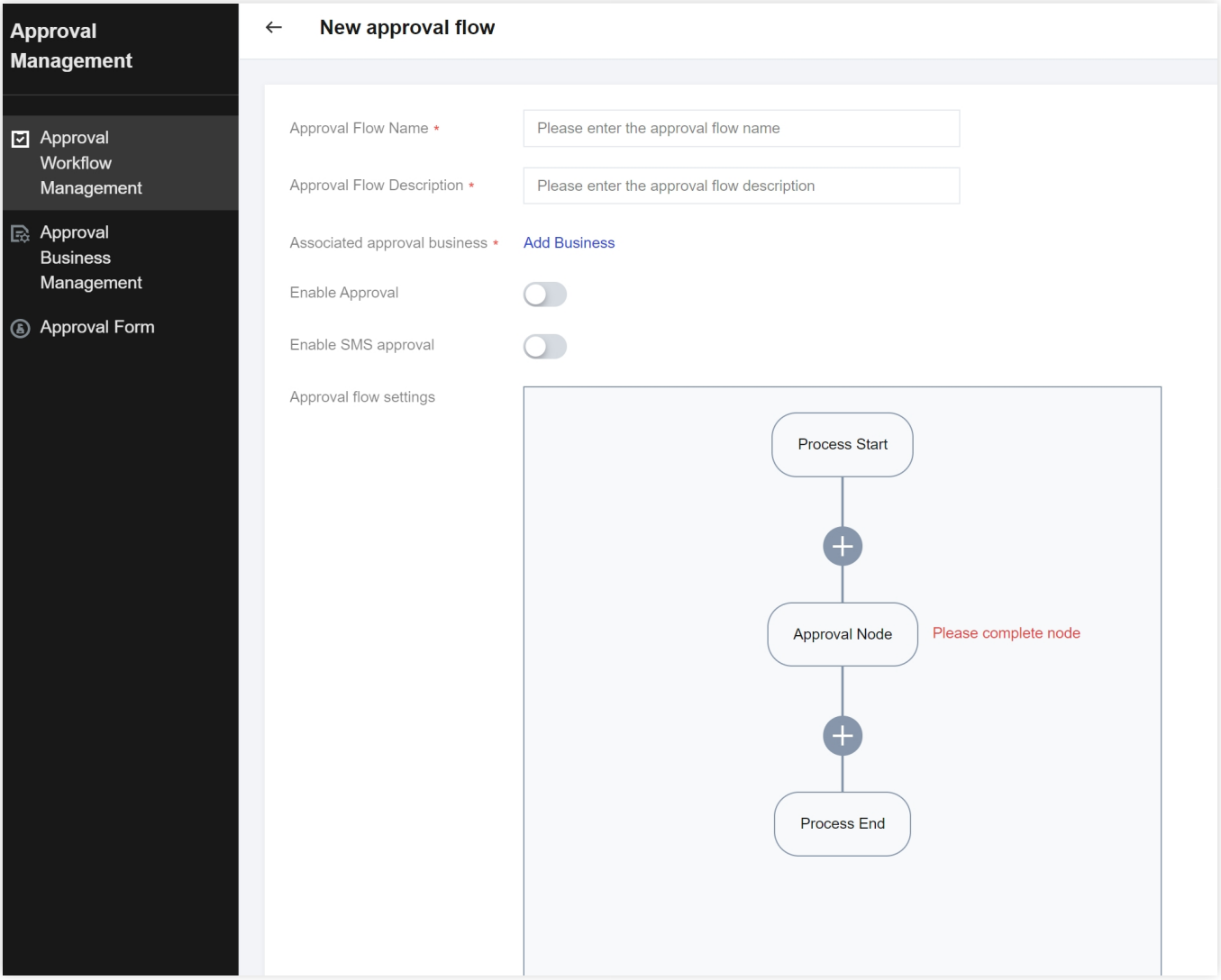
Approval Process Management

Managing Approval Processes

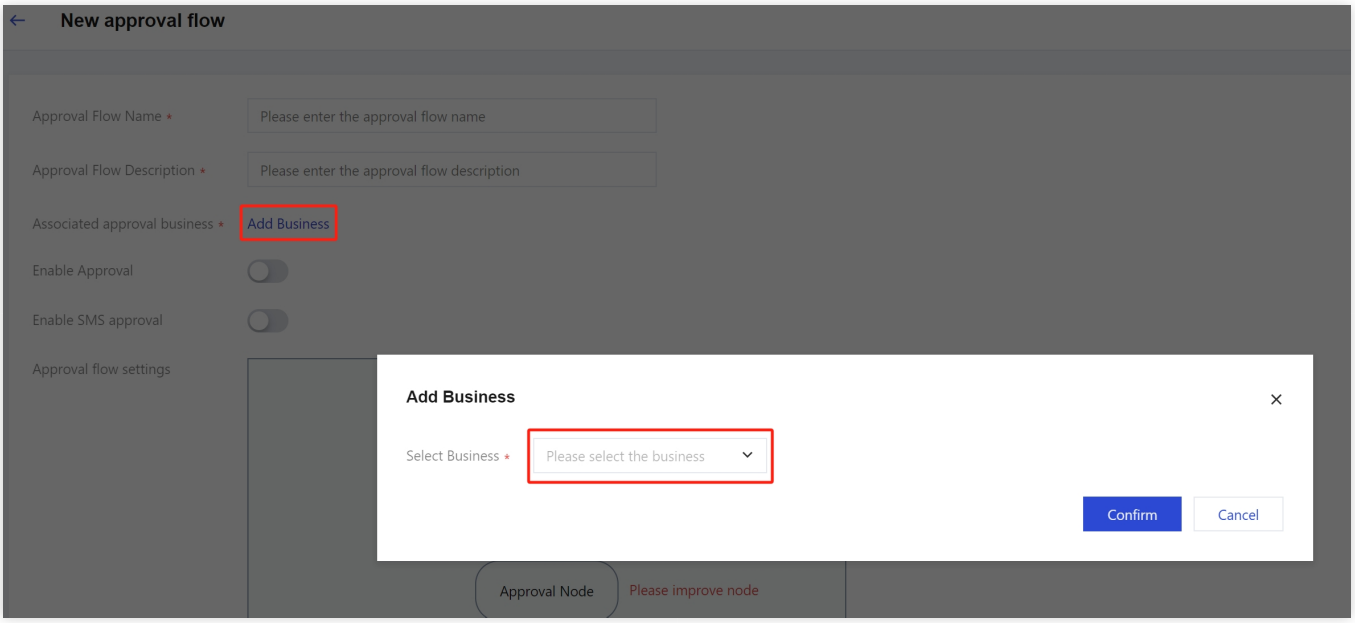
An approval process is defined based on approval businesses. It defines which approvers are required when users trigger specific businesses on pages.

Creating Approval Processes

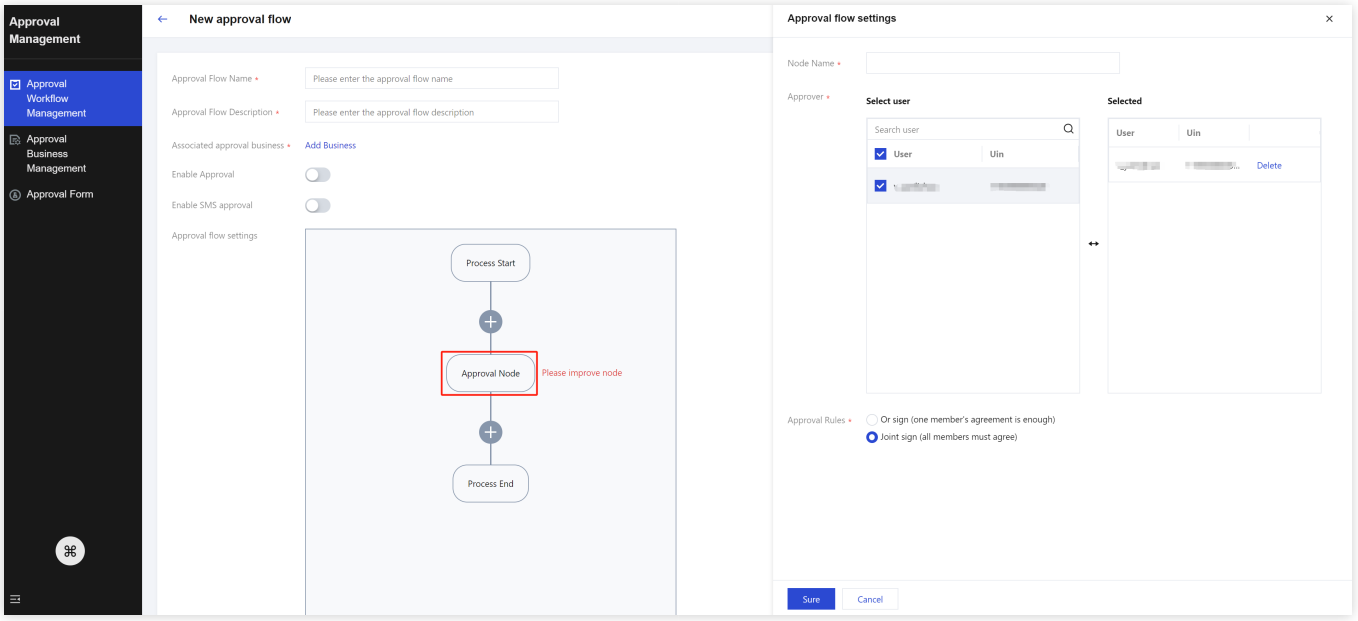
- Enter a name and description for the approval process.



- Add the business.
  Select a business to be approved. The approval processes created by a tenant only take effect for this tenant but not for other tenants.



- Set the approval process.
  You can set multiple approval nodes. Currently, a maximum of 5 approval nodes can be set. Multiple approvers can be set for each node, and the approver list displays users under the current tenant.
  Suppose that the operation-side administrator defines an approval process for a business, and the applicable scope of the approval process includes tenant A, while tenant A also defines and enables an approval process for the same business on the tenant side. When users under tenant A trigger the business on the page, the actual approval process is formed by combining the approval process for the business under tenant A and the one on the operation side. When tenant A triggers the business on the page, the business is first approved by the corresponding approvers set under tenant A and then approved by the corresponding approvers set on the operation side.
  For example, an approval process for "CVM instance creation" is defined and enabled on the operation side. Its applicable scope includes tenant A, and the approver is the sub-user user01 on the operation side. Tenant A also defines and enables an approval process for "CVM instance creation" on the tenant side, with the approver being the sub-user user02 of tenant A. At this point, when a sub-user under tenant A creates a CVM instance, an approval form is generated is first sent to the sub-user user02 of tenant A. If user02 rejects the approval form, the approval form status becomes Rejected, and the CVM instance cannot be created. If the sub-user user02 of tenant A approves the approval form, the form is sent to the sub-user user01 on the operation side. If user01 approves the form, the approval form status becomes Approved, and the CVM instance will be created.

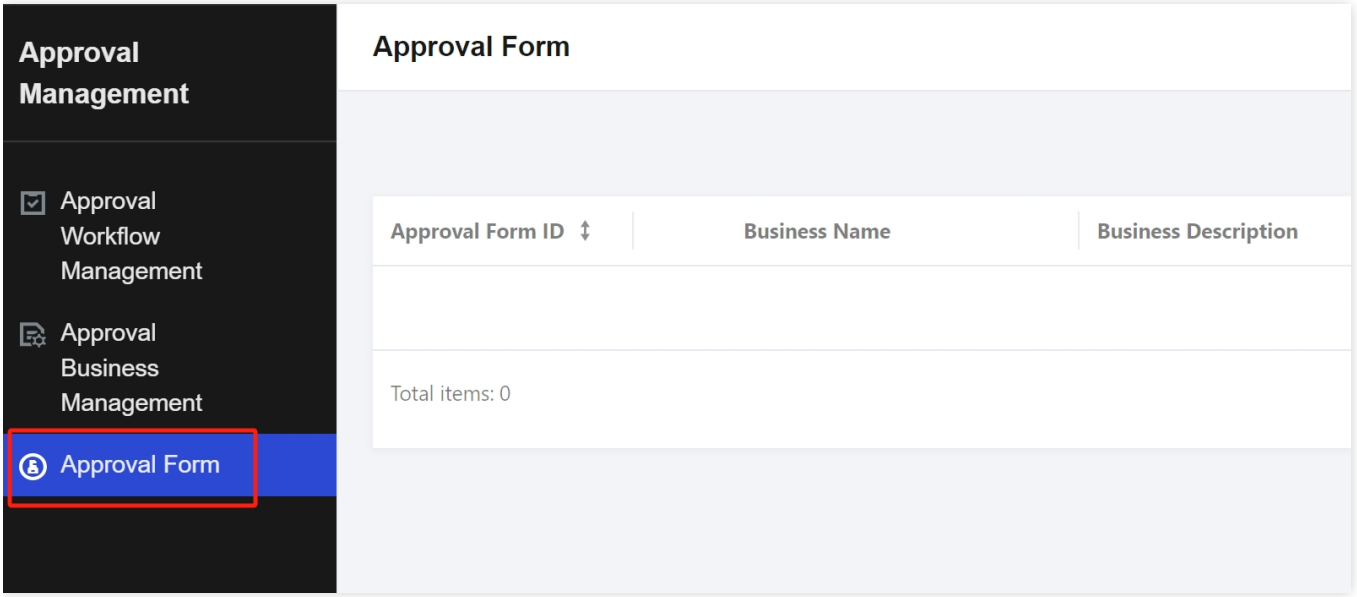


Enabling/Disabling Approval Processes

You can enable/disable the approval process in the approval process list. If the process is enabled, an approval form will be generated to trigger the approval process when users within the scope of the approval process trigger the corresponding business on the page. If the process is disabled, the approval process will become invalid.

Viewing All Approval Forms

If you assign the permission ApprovalFullAccess to a sub-user, the sub-user can view all approval forms of the current tenant on the Approval Management > Approval Form page.



Processing Approval Forms

Each user can click My Approval to view approval forms that are pending processing and have been processed by the user. Users can choose to approve or reject the forms.

Viewing the Application List

Each user can click My Application to view approval forms that are pending application and have been applied for. Users can also view the details and status of approval forms.

API Documentation
Approval flow service in the tenant console.(tapproval) version(2022-05-18)
API overview

API version

V3

Review service APIs

Interface name	Interface function
BatchPerformApproval	BatchPerformApproval
BatchSubmitApproval	BatchSubmitApproval
DeleteApprovalFlow	DeleteApprovalFlow
ModifyApprovalFlow	ModifyApprovalFlow
OperateFlowStatus	OperateFlowStatus
QueryApprovalFlowStatus	QueryApprovalFlowStatus
QueryCurrApprovalDetail	QueryCurrApprovalDetail
QueryCustomerApprovalDetail	QueryCustomerApprovalDetail
QueryPendingApprovalDoc	QueryPendingApprovalDoc
SendApprovalReminder	SendApprovalReminder
SendApprovalVerifyCode	SendApprovalVerifyCode
VerifyApprovalSmsCode	VerifyApprovalSmsCode
WithdrawApplication	WithdrawApplication

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 thesignature original stringobtained 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, Swhich 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 valu
e is URL encoded here.
            url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).app
end("&");
        }
        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 automati
c 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
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

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-urlencode
d";
    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_URLENCODED);
        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

PThe 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 cor
rect."
        },
        "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 Action in the interface documentation. For example, for the cloud server's query instance list interface, the value is DescribeInstances .
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 Region 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 Timestamp is used to prevent replay attacks.
SecretId	String	Yes	The SecretId is an identifier applied for the cloud API key to identify the user's identity. A SecretId corresponds to a unique SecretKey , and the SecretKey is used to generate the request signature Signature .
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 Version 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.

Review service APIs BatchPerformApproval

1. Interface Description

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

Batch Approval

Default interface request rate limit: 20 times per second.

Interface update time: 2023-12-15 04:27:58.

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: BatchPerformApproval
Version	Yes	No	String	Public parameter, The value of this interface: 2022-05-18
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
PaperIDs	Yes	No	Array of Uint64	Pending Approval ID List Example: 1
Operate	Yes	No	Uint64	Enumerated Values. 12: Reject, 14: Agree Example: 14
Opinion	No	No	String	Review comment. Example: Approved

3. Output Parameters

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

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

BatchSubmitApproval

1. Interface Description

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

Batch Submit Approval Applications

Default interface request rate limit: 20 times per second.

Interface update time: 2023-12-15 04:25:16.

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: BatchSubmitApproval
Version	Yes	No	String	Public parameter, The value of this interface: 2022-05-18
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
Reason	Yes	No	String	Application Reasons Example: Reason
PaperIDs	Yes	No	Array of Uint64	Approval Form ID List Example: 1

3. Output Parameters

Parameter name	Type	Description

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

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

DeleteApprovalFlow

1. Interface Description

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

Deletes a review process.

Default interface request rate limit: 20 times per second.

Interface update time: 2023-12-15 04:21:07.

The interface is both signature verification and authentication.

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: DeleteApprovalFlow
Version	Yes	No	String	Public parameter, The value of this interface: 2022-05-18
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
FlowID	Yes	No	Uint64	Review process ID. Example: 1

3. Output Parameters

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

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

ModifyApprovalFlow

1. Interface Description

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

Modifies a review process.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-04-26 21:40:23.

The interface is both signature verification and authentication.

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: ModifyApprovalFlow
Version	Yes	No	String	Public parameter, The value of this interface: 2022-05-18
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
Name	Yes	No	String	Review process name. Example: Name
Description	Yes	No	String	Description of the approval flow. Example: Description
SchemaProps	Yes	No	String	Action schema that can be modified. Example: SchemaProps
Activated	Yes	No	Bool	Whether to activate. Example: true
ActionIDs	Yes	No	Array of Uint64	List of the associated business IDs. Example: [1,2,3]
Stages	Yes	No	Array of StageParam	Node List Example:View
Scopes	Yes	No	Array of Scope	Authorization scope. Example:View
FlowID	Yes	No	Uint64	Review process ID. Example: 1

Parameter name	Required	Allow NULL	Type	Description
AllowSms	No	No	Bool	Allow Approval Form to Use Mobile Verification Code for Approval Example: false
AutoReject	No	No	Bool	Whether to automatically reject approval. Example: false
Remark	No	No	String	Remarks Example: remark

3. Output Parameters

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.Approval	Internal error

OperateFlowStatus

1. Interface Description

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

Operates on the status of the approval flow.

Default interface request rate limit: 20 times per second.

Interface update time: 2023-12-15 04:21:07.

The interface is both signature verification and authentication.

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: OperateFlowStatus
Version	Yes	No	String	Public parameter, The value of this interface: 2022-05-18
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
FlowID	Yes	No	Uint64	Review process ID. Example: 1
Activated	Yes	No	Bool	true: activated. false: approving. Example: true

3. Output Parameters

Parameter name	Type	Description

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

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

QueryApprovalFlowStatus

1. Interface Description

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

Queries the status of the tenant-associated approval flow.

Default interface request rate limit: 20 times per second.

Interface update time: 2024-11-06 21:52:54.

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: QueryApprovalFlowStatus
Version	Yes	No	String	Public parameter, The value of this interface: 2022-05-18
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
ApprovalApiInfo	Yes	No	ApiInfo	Parameters used for the v3 API for querying the status of the approval flow. Example:View

3. Output Parameters

Parameter name	Type	Description
Data	IsOpenAttr	data 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.

QueryCurrApprovalDetail

1. Interface Description

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

My Approvals - My Approvals/My Applications: View Approval Details Interface

Default interface request rate limit: 20 times per second.

Interface update time: 2025-04-26 22:46:29.

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: QueryCurrApprovalDetail
Version	Yes	No	String	Public parameter, The value of this interface: 2022-05-18
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
ID	Yes	No	Uint64	Approval Form ID Example: 1

3. Output Parameters

Parameter name	Type	Description
Data	Array of ApprovalPaperAttr	Data Example:View

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

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

QueryCustomerApprovalDetail

1. Interface Description

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

Queries the details of the approval flow order in the tenant console.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-04-26 22:30:20.

The interface is both signature verification and authentication.

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: QueryCustomerApprovalDetail
Version	Yes	No	String	Public parameter, The value of this interface: 2022-05-18
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
ID	Yes	No	Uint64	Approval ticket ID. Example: 1

3. Output Parameters

Parameter name	Type	Description
Data	Array of ApprovalPaperAttr	Data Example:View

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

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

QueryPendingApprovalDoc

1. Interface Description

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

Query Pending Application List

Default interface request rate limit: 20 times per second.

Interface update time: 2025-01-24 22:43:58.

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: QueryPendingApprovalDoc
Version	Yes	No	String	Public parameter, The value of this interface: 2022-05-18
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
ActionName	No	No	String	Business Name Example: Name
ApplicantUin	No	No	String	Applicant UIN. Example: 130000000281
Applicant	No	No	String	Applicant Example: Lucy
OwnerAccount	No	No	String	Root Account Example: Tom
Reason	No	No	String	Application Reasons Example: Reason

Parameter name	Required	Allow NULL	Type	Description
Limit	Yes	No	Uint64	Number of the page to return. Example: 1
Offset	Yes	No	Uint64	Offset. Example: 10
ID	No	No	Uint64	Approval ticket ID. Example: 1
Status	No	No	Uint64	Approval ticket state. Example: 1
Sort	No	No	Order	Information about field used to sort data. Example:View

3. Output Parameters

Parameter name	Type	Description
Data	ApprovalPaperSet	Return data 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.

SendApprovalReminder

1. Interface Description

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

Send Approval Reminder

Default interface request rate limit: 20 times per second.

Interface update time: 2023-12-15 04:31:06.

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: SendApprovalReminder
Version	Yes	No	String	Public parameter, The value of this interface: 2022-05-18
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
PaperID	Yes	No	Uint64	Approval Form ID Example: 1

3. Output Parameters

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

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

SendApprovalVerifyCode

1. Interface Description

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

Send Approval Verification Code

Default interface request rate limit: 20 times per second.

Interface update time: 2023-12-15 04:59:22.

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: SendApprovalVerifyCode
Version	Yes	No	String	Public parameter, The value of this interface: 2022-05-18
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
ApproverUin	No	No	String	Approver UIN Example: 130000000281
PaperID	No	No	Uint64	Approval Form ID Example: 1
StageSerialNum	No	No	Uint64	Current Approval Form Node Sequence Number Example: 1

3. Output Parameters

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

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

VerifyApprovalSmsCode

1. Interface Description

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

Verify Approval SMS Code

Default interface request rate limit: 20 times per second.

Interface update time: 2023-12-15 05:01:08.

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: VerifyApprovalSmsCode
Version	Yes	No	String	Public parameter, The value of this interface: 2022-05-18
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
ApproverUin	No	No	String	Approver UIN Example: 130000000281
PaperID	No	No	Uint64	Approval Form ID Example: 1
StageSerialNum	No	No	Uint64	Current Node Sequence Number of Approval Form Example: 1
Code	No	No	String	Approval SMS Code Example: 1234

3. Output Parameters

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

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

WithdrawApplication

1. Interface Description

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

If a tenant cancels a submitted approval form, the tenant will be notified of the cancellation result. The current approver does not need to process this approval form.

Default interface request rate limit: 20 times per second.

Interface update time: 2024-04-14 21:18:23.

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: WithdrawApplication
Version	Yes	No	String	Public parameter, The value of this interface: 2022-05-18
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
PaperID	No	No	Uint64	Approval ticket ID. Example: 1
Reason	No	No	String	Reasons for Canceling Approval Form Example: Reason

3. Output Parameters

Parameter name	Type	Description

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

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

Data structure

ApproverInfoParam

ApproverInfoParam

Referenced by the following interfaces: CreateCustomerFlow, GetApprovedSet, GetFlowDetail, ModifyApprovalFlow, QueryApplicationForm, QueryCurrApprovalDetail, QueryCurrApprovalDoc, QueryCustomerApprovalDetail, QueryPendingApprovalDoc

name	required	NULL allowed	type	description
ApproverUin	no	no	String	ApproverUin Example: 130000000281
ApproverUserName	no	no	String	ApproverUserName Example: Lucy
ApproverStatus	no	no	String	ApproverStatus Example: 14

SealParam

SealParam for user approve record

Referenced by the following interfaces: CreateCustomerFlow, GetApprovedSet, GetFlowDetail, ModifyApprovalFlow, QueryApplicationForm, QueryCurrApprovalDetail, QueryCurrApprovalDoc, QueryCustomerApprovalDetail, QueryPendingApprovalDoc

name	required	NULL allowed	type	description
ID	yes	no	Int64	ID Example: 1
PaperID	yes	no	Int64	Approval Form ID Example: 1
StageSerialNum	yes	no	Int64	Approval phase sequence. Example: 1
OpUin	yes	no	String	OpUin Example: 130000000281
ApproveTime	yes	no	Datetime_iso	ApproveTime Example: 2024-08-03T12:00:00+08:00
Operate	yes	no	Int64	Operate Example: 14
Opinion	yes	no	String	Opinion Example: Approved
Request	no	yes	String	Request Example: json RawRequest

Scope

Validity user scope.

Referenced by the following interfaces: CreateCustomerFlow, GetFlowDetail, ModifyApprovalFlow

name	required	NULL allowed	type	description
ActionID	yes	no	Uint64	ID of the API operation performed on the service. Example: 1
Users	yes	no	Array of UserScope	List of the users by which the operation is performed. Example:View

FlowBaseAttr

Approval process basic attribute.

Referenced by the following interfaces: DescribeFlowSet, QueryCustomerActionScope

name	required	NULL allowed	type	description
FlowID	no	no	Int64	Approval process ID. Example: 18
Platform	no	no	String	Platform Example: tcloud
Name	no	no	String	Approval flow name Example: Name
Description	no	no	String	Description of the approval flow. Example: Description
SchemaProps	no	no	String	SchemaProps Example: SchemaProps
Activated	no	no	Bool	Enable/Disable Example: true
OwnerUin	no	no	String	Owner UIN Example: 130000000013
CreateUin	no	no	String	Creator UIN. Example: 130000000013
CTime	no	no	Datetime_iso	Creation date Example: 2025-04-25T17:12:03+08:00
UpTime	no	no	Datetime_iso	Modification Date Example: 2025-04-25T17:12:03+08:00
Version	no	no	Int64	Version. Example: 12

name	required	NULL allowed	type	description
CreateUser	no	no	String	Create User Example: Lucy
ActionIDs	no	no	Array of Int64	Approval business ID. Example: [1,2,3]
AllowSms	no	no	Bool	Whether to allow approval via SMS. Example: false
AutoReject	no	no	Bool	Whether to automatically reject. Example: false
Remark	no	no	String	Remarks Example: Remark

QueryActionParma

Queries actions.

Referenced by the following interfaces: QueryActionSet

name	required	NULL allowed	type	description
ActionIDs	no	no	Array of Uint64	ID list. Example: [1,2,3]
Action	no	no	String	Action name. Example: CreateFlow
Module	no	no	String	Associated module. Example: Module
Limit	yes	no	Uint64	Number of the page to return. Example: 1
Offset	yes	no	Uint64	Offset. Example: 10
ActionName	no	no	String	Business name. Example: Name
ModuleName	no	no	String	The name of the module corresponding to the API. Example: approval
YunProductName	no	no	String	The name of the product corresponding to the API. Example: BMPPro
Sort	no	no	Order	Sort field. Example:View
Filters	no	no	Array of FilterAttr	Filter Condition List Example:View

ActionAttr

Approval business.

Referenced by the following interfaces: QueryActionSet

name	required	NULL allowed	type	description
Action	no	no	String	Approval business API name. Example: AddMachine
Module	no	no	String	Approval business module. Example: cvm
Version	no	no	String	Approval business version. Example: 2022-02-02
ActionID	no	no	Int64	Approval business ID. Example: 1
Description	no	no	String	Approval business description Example: AddMachine
Schema	no	no	String	Schema Example: Schema
ActionName	no	no	String	Approval business API name. Example: AddMachine
ModuleName	no	no	String	Approval business module name. Example: cvm
YunProductName	no	no	String	Approval business cloud product name. Example: cvm
Category	no	no	String	Type Example: 6
Creator	no	no	String	Creator Example: admin
UpdateUser	no	no	String	Modifier Example: Lucy
CreateUserUin	no	no	String	Creator UIN. Example: 130000000281
UpdateUserUin	no	no	String	Modifier UIN. Example: 130000000281
CreateTime	no	no	Datetime_iso	Creation date Example: 2024-08-03T12:00:00+08:00
UpdateTime	no	no	Datetime_iso	Modification Date Example: 2024-08-03T12:00:00+08:00

name	required	NULL allowed	type	description
Origin	no	no	Int64	Source Example: 1
Type	no	no	Int64	Type Example: 6
JsonPath	no	no	String	JsonPath Example: JsonPath

ActionRelatedInfo

Information about approval flow associated with approval business

Referenced by the following interfaces: QueryCustomerActionScope

name	required	NULL allowed	type	description
Users	no	no	Array of UserScope	Associated user information (-1: All tenants) Example:View
Flow	no	no	Array of FlowBaseAttr	Associated approval flow information Example:View
ActionID	no	no	Int64	Approval Service ID Example: 1

IsOpenAttr

Whether to enable approval workflow.

Referenced by the following interfaces: QueryApprovalFlowStatus

name	required	NULL allowed	type	description
IsOpen	yes	no	Bool	Whether there is an active approval workflow. True: Yes, False: No Example: false

FlowDetail

Approval process details.

Referenced by the following interfaces: GetFlowDetail

name	required	NULL allowed	type	description
FlowID	no	no	Int64	Approval process ID. Example: 18
Platform	no	no	String	Platform Example: tcloud
Name	no	no	String	Approval flow name Example: Name
Description	no	no	String	Description of the approval flow. Example: Description
SchemaProps	no	no	String	SchemaProps Example: SchemaProps
Activated	no	no	Bool	Enable/Disable Example: true
OwnerUin	no	no	String	Owner UIN Example: 130000000013
CreateUin	no	no	String	Creator UIN. Example: 130000000013
CTime	no	no	Datetime_iso	Creation date Example: 2025-04-25T17:12:03+08:00
UpTime	no	no	Datetime_iso	Modification Date Example: 2025-04-25T17:12:03+08:00
Version	no	no	Int64	Version. Example: 12
CreateUser	no	no	String	Create User Example: Lucy
ActionIDs	no	yes	Array of Int64	Approval business ID. Example: [1,2,3]
AllowSms	no	yes	Bool	Whether to allow approval via SMS. Example: false
AutoReject	no	yes	Bool	Whether to automatically reject. Example: false
Remark	no	yes	String	Remarks Example: Remark
Stages	no	yes	Array of StageParam	Approval node list. Example:View
Scopes	no	yes	Array of Scope	Approval scope list. Example:View

FlowsData

Approval Flow List

Referenced by the following interfaces: DescribeFlowSet

name	required	NULL allowed	type	description
Flows	no	no	Array of FlowBaseAttr	Approval Flow List Example:View
Total	no	no	Int64	Total number Example: 100

ApprovalPaperAttr

Attributes of Approval Form

Referenced by the following interfaces: GetApprovedSet, QueryApplicationForm, QueryCurrApprovalDetail, QueryCurrApprovalDoc, QueryCustomerApprovalDetail, QueryPendingApprovalDoc

name	required	NULL allowed	type	description
Action	yes	no	String	Corresponding Cloud API Interface Example: CreateFlow
ActionDescription	yes	no	String	Approval Service Description Information Example: Description
ActionID	yes	no	Int64	Approval Service ID Example: 1
ActionName	yes	no	String	Name of Approval Service Example: Name
Applicant	yes	no	String	Applicant Example: Lucy
ApplicantUin	yes	no	String	Applicant UIN Example: 130000000281
CTime	yes	no	String	Application time Example: 2024-08-03T12:00:00+08:00
CallbackStatus	yes	no	Int64	Callback status after the approval Is approved. 0: Not Started, 100: Callback Succeeded, 101: Callback Failed Example: 100
CurrStageNum	yes	no	Int64	Current Approval Node Example: 1
FlowDescription	no	no	String	Associated Approval Workflow Description Information Example: Description
FlowID	yes	no	Int64	Approval Workflow ID Example: 1
ModuleName	yes	no	String	Name of Cloud API Module Corresponding to Approval Service Example: approval
OwnerAccount	no	no	String	Root account name Example: Tom
OwnerUin	yes	no	String	Primary Account UIN Example: 130000000281
PaperID	yes	no	Int64	Approval Form ID Example: 1
ParentID	no	no	Int64	Parent Approval Form ID Example: 1
ProductName	yes	no	String	Cloud Product Name Example: Name
Reason	no	no	String	Approval Reasons Example: Reason
Schema	no	no	String	schema Example: Schema
RequestBody	no	yes	String	Request Parameters, unable to comply with Cloud API 3 Specification here Example: RequestBody
Seals	no	yes	SealParam	Approved Information Example:View
Status	no	no	Int64	Approval form status. 0: Approval Form Initialized, 1: Approval in Progress, 11: Application Canceled, 12: Approval Rejected, 14: Approval Approved, 15: Automatically Approved Example: 1
Stages	no	yes	Array of StageParam	Corresponding Approval Node Information Example:View
CallBackResult	no	no	String	CallBackResult Example: CallBackResult
FinalStatus	no	no	Uint64	Final Approval Form Status Example: 14
RelatedPapers	no	no	Array of ApprovalPaperAttr	Associated Approval Form Information Example:View
Category	no	no	Uint64	Approval Form Type. 1: Triggered by Page, 2: Merged Application Form Example: Category
LastRemindTime	no	yes	String	Last Reminder Time for Approver Example: 2024-08-03T12:00:00+08:00
AllowSms	no	no	Bool	Whether to allow approval via SMS. Example: false

ApiInfo

Cloud API. The V3 port is used. Trituple settings must be configured.

Referenced by the following interfaces: QueryApprovalFlowStatus, QueryCustomerApprovalDoc

name	required	NULL allowed	type	description

name	required	NULL allowed	type	description
Action	yes	no	String	API name. Example: CreateFlow
Module	yes	no	String	Component Name Example: Module
Version	yes	no	String	Version. Example: 2024-01-01

ApprovalPaperSet

Approval Form List Information

Referenced by the following interfaces: GetApprovedSet, QueryApplicationForm, QueryCurrApprovalDoc, QueryPendingApprovalDoc

name	required	NULL allowed	type	description
PaperSet	yes	yes	Array of ApprovalPaperAttr	Approval Form List Example:View
Total	yes	no	Int64	Total number Example: 100

QueryActionsOutput

QueryActionsOutput

Referenced by the following interfaces: QueryActionSet

name	required	NULL allowed	type	description
Total	no	no	Int64	Total number Example: 100
Actions	no	yes	Array of ActionAttr	Approval business list. Example:View

UserScope

Validity user scope.

Referenced by the following interfaces: CreateCustomerFlow, GetFlowDetail, ModifyApprovalFlow, QueryCustomerActionScope

name	required	NULL allowed	type	description
Uin	no	no	String	--1: all tenants are covered. Other values: UINs of specific tenants. Example: 130000000281
IsSubAccount	no	no	Bool	Specifies whether the account is a sub-account. Example: true
UserName	no	no	String	Username Example: Lucy

Order

Information about field used to sort data.

Referenced by the following interfaces: DescribeFlowSet, GetApprovedSet, QueryActionSet, QueryApplicationForm, QueryCurrApprovalDoc, QueryCustomerApprovalDoc, QueryPendingApprovalDoc

name	required	NULL allowed	type	description
Field	yes	no	String	Name of a field used to sort data. Example: ActionName
IsDesc	yes	no	Bool	Specifies whether to sort data in descending order. true: ascending order. false: descending order. Example: true

FilterAttr

Filter Parameters

Referenced by the following interfaces: DescribeFlowSet, QueryActionSet

name	required	NULL allowed	type	description
Name	yes	no	String	Parameter Name Example: Name
Operator	yes	no	String	Operators, Supporting equal and like Example: in
Value	yes	no	String	Parameter Value Example: dcdb

StageParam

Stage parameters.

Referenced by the following interfaces: CreateCustomerFlow, GetApprovedSet, GetFlowDetail, ModifyApprovalFlow, QueryApplicationForm, QueryCurrApprovalDetail, QueryCurrApprovalDoc, QueryCustomerApprovalDetail, QueryPendingApprovalDoc

name	required	NULL allowed	type	description
StageID	no	no	Uint64	stage id Example: 1
Name	yes	no	String	Node name. Example: Name
SingleSeal	yes	no	Bool	false: signs by all approvers. true: signs by one approver. Example: true

name	required	NULL allowed	type	description
Approvers	yes	yes	Array of String	List of approver UINs in the tenant console. Example: [130000000281]
SerialNumber	yes	no	Uint64	Serial number of the node. Example: 1
Seals	no	yes	Array of SealParam	Seals Example:View
StageStatus	no	no	Int64	StageStatus Example: 14
ApproverInfo	no	yes	Array of ApproverInfoParam	ApproverInfo Example:View

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 cor
rect."
        },
        "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.Approval	Internal error

Error codes	Description
InternalError.DatabaseError	Database error