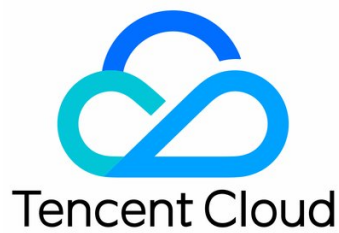


Optical Character Recognition

API Documentation

Product Documentation



Copyright Notice

©2013-2019 Tencent Cloud. All rights reserved.

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

Trademark Notice

 Tencent Cloud

All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

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

Contents

API Documentation

- History

- API Category

- Making API Requests

 - Request Structure

 - Common Params

 - Signature v3

 - Signature

 - Responses

- General OCR APIs

 - GeneralBasicOCR

 - GeneralAccurateOCR

- Card OCR APIs

 - BankCardOCR

 - MLIDCardOCR

 - MLIDPassportOCR

 - HKIDCardOCR

- Industry-specific Document Recognition APIs

 - TableOCR

- Data Types

- Error Codes

API Documentation

History

Last updated : 2021-07-07 11:27:47

Release 5

Release time: 2021-07-07 11:25:32

Release updates:

Improvement to existing documentation.

Modified APIs:

- [BankCardOCR](#)
 - New input parameters:EnableQualityValue
 - New output parameters:QualityValue
- [GeneralAccurateOCR](#)
 - New input parameters:IsWords
- [GeneralBasicOCR](#)
 - New input parameters:IsWords

New data structures:

- [DetectedWordCoordPoint](#)
- [DetectedWords](#)

Modified data structures:

- [TextDetection](#)
 - New members:Words, WordCoordPoint

Release 4

Release time: 2021-01-27 15:26:16

Release updates:

Improvement to existing documentation.

Modified APIs:

- [BankCardOCR](#)
 - New output parameters:CardType, CardName, BorderCutImage, CardNoImage, WarningCode

Release 3

Release time: 2021-01-18 15:18:26

Release updates:

Improvement to existing documentation.

Modified APIs:

- [BankCardOCR](#)
 - New input parameters:RetBorderCutImage, RetCardNoImage, EnableCopyCheck, EnableReshootCheck, EnableBorderCheck

Release 2

Release time: 2020-09-10 17:01:40

Release updates:

Improvement to existing documentation.

New APIs:

- [GeneralAccurateOCR](#)
- [TableOCR](#)

Modified APIs:

- [GeneralBasicOCR](#)
 - New input parameters: IsPdf, PdfPageNumber
 - New output parameters: PdfPageSize
- [MLIDCardOCR](#)
 - New output parameters: Birthday

New data structures:

- [TextTable](#)

Existing Release

Release time: 2020-08-06 19:48:59

Existing APIs/data structures are as follows:

Improvement to existing documentation.

Existing APIs:

- [BankCardOCR](#)
- [GeneralBasicOCR](#)
- [HKIDCardOCR](#)
- [MLIDCardOCR](#)
- [MLIDPassportOCR](#)

Existing data structures:

- [Coord](#)
- [ItemCoord](#)
- [TextDetection](#)

API Category

Last updated : 2021-01-18 15:19:06

General OCR APIs

API Name	Feature
GeneralBasicOCR	Recognizes general print.
GeneralAccurateOCR	Recognizes general print (high-precision)

Card OCR APIs

API Name	Feature
BankCardOCR	Recognizes a bank card
HKIDCardOCR	Recognizes Hong Kong (China) identity card
MLIDCardOCR	Recognizes a Malaysian identity card
MLIDPassportOCR	Recognizes a non-Mainland China passport

Industry-specific Document Recognition APIs

API Name	Feature
TableOCR	Form OCR (v1)

Making API Requests

Request Structure

Last updated : 2020-09-11 10:10:08

1. Service Address

The API supports access from either a nearby region (at ocr.tencentcloudapi.com) or a specified region (at ocr.ap-guangzhou.tencentcloudapi.com for Guangzhou, for example).

We recommend using the domain name to access the nearest server. When you call an API, the request is automatically resolved to a server in the region **nearest** to the location where the API is initiated. For example, when you initiate an API request in Guangzhou, this domain name is automatically resolved to a Guangzhou server, the result is the same as that of specifying the region in the domain like "ocr.ap-guangzhou.tencentcloudapi.com".

**Note: For latency-sensitive businesses, we recommend that you specify the region in the domain name. **

Tencent Cloud currently supports the following regions:

Hosted region	Domain name
Local access region (recommended, only for non-financial availability zones)	ocr.tencentcloudapi.com
South China (Guangzhou)	ocr.ap-guangzhou.tencentcloudapi.com
East China (Shanghai)	ocr.ap-shanghai.tencentcloudapi.com
North China (Beijing)	ocr.ap-beijing.tencentcloudapi.com
Southwest China (Chengdu)	ocr.ap-chengdu.tencentcloudapi.com
Southwest China (Chongqing)	ocr.ap-chongqing.tencentcloudapi.com
Hong Kong, Macao, Taiwan (Hong Kong, China)	ocr.ap-hongkong.tencentcloudapi.com
Southeast Asia (Singapore)	ocr.ap-singapore.tencentcloudapi.com
Southeast Asia (Bangkok)	ocr.ap-bangkok.tencentcloudapi.com
South Asia (Mumbai)	ocr.ap-mumbai.tencentcloudapi.com
Northeast Asia (Seoul)	ocr.ap-seoul.tencentcloudapi.com
Northeast Asia (Tokyo)	ocr.ap-tokyo.tencentcloudapi.com
U.S. East Coast (Virginia)	ocr.na-ashburn.tencentcloudapi.com
U.S. West Coast (Silicon Valley)	ocr.na-siliconvalley.tencentcloudapi.com
North America (Toronto)	ocr.na-toronto.tencentcloudapi.com
Europe (Frankfurt)	ocr.eu-frankfurt.tencentcloudapi.com
Europe (Moscow)	ocr.eu-moscow.tencentcloudapi.com

2. Communications Protocol

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

3. Request Methods

Supported HTTP request methods:

- POST (recommended)

- GET

The Content-Type types supported by POST requests:

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

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

4. Character Encoding

Only UTF-8 encoding is used.

Common Params

Last updated : 2020-05-22 10:41:31

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

Signature Algorithm v3

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

Parameter Name	Type	Required	Description
X-TC-Action	String	Yes	The name of the API for the desired operation. For the specific value, see the description of common parameter <code>Action</code> in the input parameters in related API documentation. For example, the API for querying the CVM instance list is <code>DescribeInstances</code> .
X-TC-Region	String	Yes	Region parameter, which is used to identify the region to which the data you want to work with belongs. For values supported for an API, see the description of common parameter <code>Region</code> in the input parameters in related API documentation. Note: This parameter is not required for some APIs (which will be indicated in related API documentation), and will not take effect even it is passed.
X-TC-Timestamp	Integer	Yes	The current UNIX timestamp that records the time when the API request was initiated, for example, 1529223702. Note: If the difference between the UNIX timestamp and the server time is greater than 5 minutes, a signature expiration error may occur.
X-TC-Version	String	Yes	API version of the action. For the valid values, see the description of the common input parameter <code>Version</code> in the API documentation. For example, the version of CVM is 2017-03-12.
Authorization	String	Yes	The HTTP authentication request header, 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: 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, a domain name <code>cvm.tencentcloudapi.com</code> refers to the CVM product and the value would be <code>cvm</code> ; - 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 a temporary certificate. It must be used with a temporary key. You can obtain the temporary key and token by calling a CAM API. No token is required for a long-term key.

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

Example of an HTTP GET request structure:

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

```
Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmlPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

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

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

```
X-TC-Action: DescribeInstances
```

```
X-TC-Version: 2017-03-12
```

```
X-TC-Timestamp: 1539084154
```

```
X-TC-Region: ap-guangzhou
```

The following example shows you how to structure an HTTP POST (application/json) request:

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

```
Authorization: TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: application/json
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou
```

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

Example of an HTTP POST (multipart/form-data) request structure (only supported by specific APIs):

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

```
Authorization: TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: multipart/form-data; boundary=58731222010402
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou
```

```
--58731222010402
```

```
Content-Disposition: form-data; name="Offset"
```

```
0
```

```
--58731222010402
```

```
Content-Disposition: form-data; name="Limit"
```

```
10
```

```
--58731222010402--
```

Signature Algorithm v1

To adopt the HmacSHA1 and HmacSHA256 signature methods, common parameters must be put into the request string, as shown below:

Parameter Name	Type	Required	Description
Action	String	Yes	The name of the API for the desired operation. For the specific value, see the description of common parameter <code>Action</code> in the input parameters in related API documentation. For example, the API for querying the CVM instance list is <code>DescribeInstances</code> .
Region	String	Yes	Region parameter, which is used to identify the region to which the data you want to work with belongs. For values supported for an API, see the description of common parameter <code>Region</code> in the input parameters in related API documentation. Note: This parameter is not required for some APIs (which will be indicated in related API documentation), and will not take effect even if it is passed.
Timestamp	Integer	Yes	The current UNIX timestamp that records the time when the API request was initiated, for example, 1529223702. If the difference between the value and the current system time is too large, a signature expiration error may occur.
Nonce	Integer	Yes	A random positive integer used along with <code>Timestamp</code> to prevent replay attacks.
SecretId	String	Yes	The identifying SecretId obtained on the Cloud API Key page. A SecretId corresponds to a unique SecretKey which is used to generate the request signature (Signature).
Signature	String	Yes	Request signature used to verify the validity of this request. This is calculated based on the actual input parameters. For more information about how this is calculated, see the API authentication documentation.
Version	String	Yes	API version of the action. For the valid values, see the description of the common input parameter <code>Version</code> in the API documentation. For example, the version of CVM is 2017-03-12.

Parameter Name	Type	Required	Description
SignatureMethod	String	No	Signature method. Currently, only HmacSHA256 and HmacSHA1 are supported. The HmacSHA256 algorithm is used to verify the signature only when this parameter is specified as HmacSHA256. In other cases, the signature is verified with HmacSHA1.
Token	String	No	The token used for a temporary certificate. It must be used with a temporary key. You can obtain the temporary key and token by calling a CAM API. No token is required for a long-term key.

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

Example of an HTTP GET request structure:

```
https://cvm.tencentcloudapi.com/?Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKIDEXAMPLE
```

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

Example of an HTTP POST request structure:

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

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

```
Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKIDEXAMPLE
```

Region List

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

Region	Value
North China (Beijing)	ap-beijing
South China (Guangzhou)	ap-guangzhou
Hong Kong/Macao/Taiwan (Hong Kong, China)	ap-hongkong
Northeast Asia Pacific (Seoul)	ap-seoul
East China (Shanghai)	ap-shanghai
Southeast Asia Pacific (Singapore)	ap-singapore
North America (Toronto)	na-toronto

Signature v3

Last updated : 2020-09-11 10:10:08

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

Applying for Security Credentials

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

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

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

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

Using the Resources for Developers

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

TC3-HMAC-SHA256 Signature Algorithm

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

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

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

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

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

Assuming that your SecretId and SecretKey are `AKIDz8krbsJ5yKBZQpn74WFkMLPx3*****` and `Gu5t9xGARNpq86cd98joQYCN3*****`, respectively, if you want to view the status of the instance in the Guangzhou region whose CVM instance name is "unnamed" and have only one data entry returned, then the request may be:

```
curl -X POST https://cvm.tencentcloudapi.com ¥
-H "Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkMLPx3*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, S
ignature=c492e8e41437e97a620b728c301bb8d17e7dc0c17eeabce80c20cd70fc3a78ff" ¥
-H "Content-Type: application/json; charset=utf-8" ¥
-H "Host: cvm.tencentcloudapi.com" ¥
-H "X-TC-Action: DescribeInstances" ¥
```

```
-H "X-TC-Timestamp: 1551113065" ¥
-H "X-TC-Version: 2017-03-12" ¥
-H "X-TC-Region: ap-guangzhou" ¥
-d '{"Limit": 1, "Filters": [{"Values": ["unnamed"], "Name": "instance-name"}]}'
```

The signature calculation process is explained in detail below.

1. Concatenating the CanonicalRequest String

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

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

Field Name	Explanation
HTTPRequestMethod	HTTP request method (GET or POST). This example uses <code>POST</code> .
CanonicalURI	URI parameter. Slash ("/") is used for API 3.0.
CanonicalQueryString	The query string in the URL of the originating HTTP request. This is always an empty string "" for POST requests, and is the string after the question mark (?) for GET requests. For example: <code>Limit=10&Offset=0</code> . Note: <code>CanonicalQueryString</code> must be URL-encoded, referencing RFC3986 , the UTF8 character set. We recommend using the programming language library. All special characters must be encoded and capitalized.
CanonicalHeaders	Header information for signature calculation, including at least two headers of <code>host</code> and <code>content-type</code> . Custom headers can be added to participate in the signature process to improve the uniqueness and security of the request. Concatenation rules: 1. Both the key and value of the header should be converted to lowercase with the leading and trailing spaces removed, so they are concatenated in the format of <code>key:value¥n</code> format; 2. If there are multiple headers, they should be sorted in ASCII ascending order by the header keys (lowercase). The calculation result in this example is <code>content-type:application/json; charset=utf-8¥nhost:cvm.tencentcloudapi.com¥n</code> . Note: <code>content-type</code> must match the actually sent content. In some programming languages, a charset value would be added even if it is not specified. In this case, the request sent is different from the one signed, and the sever will return an error indicating that signature verification failed.
SignedHeaders	Header information for signature calculation, indicating which headers of the request participate in the signature process (they must each individually correspond to the headers in CanonicalHeaders). <code>Content-type</code> and <code>host</code> are required headers. Concatenation rules: 1. Both the key and value of the header should be converted to lowercase; 2. If there are multiple headers, they should be sorted in ASCII ascending order by the header keys (lowercase) and separated by semicolons (;). The value in this example is <code>content-type;host</code>
HashedRequestPayload	Hash value of the request payload (i.e., the body, such as <code>{"Limit": 1, "Filters": [{"Values": ["unnamed"], "Name": "instance-name"}]}</code> in this example). The pseudocode for calculation is <code>Lowercase(HexEncode(Hash.SHA256(RequestPayload)))</code> by SHA256 hashing the payload of the HTTP request, performing hexadecimal encoding, and finally converting the encoded string to lowercase letters. For GET requests, <code>RequestPayload</code> is always an empty string. The calculation result in this example is <code>99d58dfbc6745f6747f36bfca17dee5e6881dc0428a0a36f96199342bc5b4907</code> .

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

```
POST
/
content-type:application/json; charset=utf-8
host:cvm.tencentcloudapi.com
```

```
content-type;host
99d58dfbc6745f6747f36bfca17dee5e6881dc0428a0a36f96199342bc5b4907
```

2. Concatenating the String to Be Signed

The string to sign is concatenated as follows:

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

Field Name	Explanation
Algorithm	Signature algorithm, which is currently always TC3-HMAC-SHA256 .
RequestTimestamp	Request timestamp, i.e., the value of the common parameter X-TC-Timestamp in the request header, which is the UNIX timestamp of the current time in seconds, such as 1551113065 in this example.
CredentialScope	Scope of the credential in the format of Date/service/tc3_request , including the date, requested service and termination string (tc3_request). Date is a date in UTC time, whose value should match the UTC date converted by the common parameter X-TC-Timestamp ; service is the product name, which should match the domain name of the product called. The calculation result in this example is 2019-02-25/cvm/tc3_request .
HashedCanonicalRequest	Hash value of the CanonicalRequest string concatenated in the steps above. The pseudocode for calculation is Lowercase(HexEncode(Hash.SHA256(CanonicalRequest))). The calculation result in this example is 2815843035062ffda5fd6f2a44ea8a34818b0dc46f024b8b3786976a3adda7a .

Note:

1. Date has to be calculated from the timestamp "X-TC-Timestamp" and the time zone is UTC+0. If you add the system's local time zone information (such as UTC+8), calls can succeed both day and night but will definitely fail at 00:00. For example, if the timestamp is 1551113065 and the time in UTC+8 is 2019-02-26 00:44:25, the UTC+0 date in the calculated Date value should be 2019-02-25 instead of 2019-02-26.
2. Timestamp must be the same as your current system time, and your system time and standard time must be synced; if the difference between Timestamp and your current system time is larger than five minutes, the request will fail. If your system time is out of sync with the standard time for a while, the request will fail and return a signature expiration error.

According to the preceding rules, the string to be signed obtained in the example is as follows:

```
TC3-HMAC-SHA256
1551113065
2019-02-25/cvm/tc3_request
2815843035062ffda5fd6f2a44ea8a34818b0dc46f024b8b3786976a3adda7a
```

3. Calculating the Signature

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

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3*****"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

Field Name	Explanation
SecretKey	The original SecretKey, i.e., Gu5t9xGARNpq86cd98joQYCN3***** .
Date	The Date field information in Credential , such as 2019-02-25 in this example.

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

2) Calculate the signature with the following pseudocode:

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

4. Concatenating the Authorization

The Authorization is concatenated as follows:

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

Field Name	Explanation
Algorithm	Signature algorithm, which is always <code>TC3-HMAC-SHA256</code> .
SecretId	The SecretId in the key pair, i.e., <code>AKIDz8krbsJ5yKBZQpn74WFkmlPx3*****</code> .
CredentialScope	Credential scope (see above). The calculation result in this example is <code>2019-02-25/cvm/tc3_request</code> .
SignedHeaders	Header information for signature calculation (see above), such as <code>content-type;host</code> in this example.
Signature	Signature value. The calculation result in this example is <code>c492e8e41437e97a620b728c301bb8d17e7dc0c17eeabce80c20cd70fc3a78ff</code> .

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

```
TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmlPx3*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=c492e8e41437e97a620b728c301bb8d17e7dc0c17eeabce80c20cd70fc3a78ff
```

The following example shows a finished authorization header:

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmlPx3*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=c492e8e41437e97a620b728c301bb8d17e7dc0c17eeabce80c20cd70fc3a78ff
Content-Type: application/json; charset=utf-8
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1551113065
X-TC-Region: ap-guangzhou

{"Limit": 1, "Filters": [{"Values": ["unnamed"], "Name": "instance-name"}]}
```

5. Signature Demo

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

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

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

The final output URL might be: `https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId=0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****&Signature=ElIP9YW3pW28FpsEdkXt%2F%2BWcGel%3D&Timestamp=1465185768&Version=2017-03-12.`

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

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

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

Java

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

public class TencentCloudAPITC3Demo {
    private final static Charset UTF8 = StandardCharsets.UTF_8;
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*****";
    private final static String CT_JSON = "application/json; charset=utf-8";

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

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

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

        // ***** Step 1: Concatenate the CanonicalRequest string *****
        String httpRequestMethod = "POST";
        String canonicalUri = "/";
        String canonicalQueryString = "";
        String canonicalHeaders = "content-type:application/json; charset=utf-8" + "host:" + host + "\n";
        String signedHeaders = "content-type;host";

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



```

String hashedRequestPayload = sha256Hex(payload);
String canonicalRequest = httpRequestMethod + "%n" + canonicalUri + "%n" + canonicalQueryString + "%n"
+ canonicalHeaders + "%n" + signedHeaders + "%n" + hashedRequestPayload;
System.out.println(canonicalRequest);

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

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

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

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

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

```

Python

```

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

# Key Parameters
secret_id = "AKIDz8krbsJ5yKBZQpn74WfkmLPx3*****"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3*****"

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

# ***** Step 1: Concatenate the CanonicalRequest string *****

```

```

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

# ***** Step 2: Concatenate the string to sign *****
credential_scope = date + "/" + service + "/" + "tc3_request"
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()
string_to_sign = (algorithm + "\n" +
str(timestamp) + "\n" +
credential_scope + "\n" +
hashed_canonical_request)
print(string_to_sign)

# ***** Step 3: Calculate the Signature *****
# Function for computing signature digest
def sign(key, msg):
return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()
secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
secret_service = sign(secret_date, service)
secret_signing = sign(secret_service, "tc3_request")
signature = hmac.new(secret_signing, string_to_sign.encode("utf-8"), hashlib.sha256).hexdigest()
print(signature)

# ***** Step 4: Concatenate the Authorization *****
authorization = (algorithm + " " +
"Credential=" + secret_id + "/" + credential_scope + ", " +
"SignedHeaders=" + signed_headers + ", " +
"Signature=" + signature)
print(authorization)

print('curl -X POST ' + endpoint
+ ' -H "Authorization: ' + authorization + '" '
+ ' -H "Content-Type: application/json; charset=utf-8"'
+ ' -H "Host: ' + host + '" '
+ ' -H "X-TC-Action: ' + action + '" '
+ ' -H "X-TC-Timestamp: ' + str(timestamp) + '" '
+ ' -H "X-TC-Version: ' + version + '" '
+ ' -H "X-TC-Region: ' + region + '" '
+ " -d '" + payload + "'")

```

Golang

```

package main

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

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

```

```

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

func main() {
    secretId := "AKIDz8krbsJ5yKBZQpn74WFkmlPx3*****"
    secretKey := "Gu5t9xGARNpq86cd98joQYCN3*****"
    host := "cvm.tencentcloudapi.com"
    algorithm := "TC3-HMAC-SHA256"
    service := "cvm"
    version := "2017-03-12"
    action := "DescribeInstances"
    region := "ap-guangzhou"
    //var timestamp int64 = time.Now().Unix()
    var timestamp int64 = 1551113065

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

    // step 2: build string to sign
    date := time.Unix(timestamp, 0).UTC().Format("2006-01-02")
    credentialScope := fmt.Sprintf("%s/%s/tc3_request", date, service)
    hashedCanonicalRequest := sha256hex(canonicalRequest)
    string2sign := fmt.Sprintf("%s\r\n%d\r\n%s\r\n",
    algorithm,
    timestamp,
    credentialScope,
    hashedCanonicalRequest)
    fmt.Println(string2sign)

    // step 3: sign string
    secretDate := hmacsha256(date, "TC3"+secretKey)
    secretService := hmacsha256(service, secretDate)
    secretSigning := hmacsha256("tc3_request", secretService)
    signature := hex.EncodeToString([]byte(hmacsha256(string2sign, secretSigning)))
    fmt.Println(signature)

    // step 4: build authorization
    authorization := fmt.Sprintf("%s Credential=%s/%s, SignedHeaders=%s, Signature=%s",
    algorithm,
    secretId,
    credentialScope,
    signedHeaders,
    signature)
    fmt.Println(authorization)

    curl := fmt.Sprintf(`curl -X POST https://%s%
-H "Authorization: %s"%
-H "Content-Type: application/json; charset=utf-8"%
-H "Host: %s" -H "X-TC-Action: %s"%
-H "X-TC-Timestamp: %d"%
-H "X-TC-Version: %s"%
-H "X-TC-Region: %s"%

```

```
-d '%s`', host, authorization, host, action, timestamp, version, region, payload)
fmt.Println(curl)
}
```

PHP

```
<?php
$secretId = "AKIDz8krbsJ5yKBZQpn74WFkMLPx3*****";
$secretKey = "Gu5t9xGARNpq86cd98joQYCN3*****";
$host = "cvm.tencentcloudapi.com";
$service = "cvm";
$version = "2017-03-12";
$action = "DescribeInstances";
$region = "ap-guangzhou";
// $timestamp = time();
$timestamp = 1551113065;
$algorithm = "TC3-HMAC-SHA256";

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

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

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

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

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

Ruby

```

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

# Key Parameters
secret_id = 'AKIDz8krbsJ5yKBZQpn74WfkmLPx3*****'
secret_key = 'Gu5t9xGARNpq86cd98joQYCN3*****'

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

# ***** Step 1: Concatenate the CanonicalRequest string *****
http_request_method = 'POST'
canonical_uri = '/'
canonical_querystring = ''
canonical_headers = "content-type:application/json; charset=utf-8\nhost:#{host}\n"
signed_headers = 'content-type;host'
# params = { 'Limit' => 1, 'Filters' => [{ 'Name' => 'instance-name', 'Values' => ['unnamed'] }] }
# payload = JSON.generate(params, { 'ascii_only' => true, 'space' => ' ' })
# json will generate in random order, to get specified result in example, we hard-code it here.
payload = '{"Limit": 1, "Filters": [{"Values": ["unnamed"], "Name": "instance-name"}]}'
hashed_request_payload = Digest::SHA256.hexdigest(payload)
canonical_request = [
  http_request_method,
  canonical_uri,
  canonical_querystring,
  canonical_headers,
  signed_headers,
  hashed_request_payload,
].join("\n")

puts canonical_request

# ***** Step 2: Concatenate the string to sign *****
credential_scope = date + '/' + service + '/' + 'tc3_request'
hashed_request_payload = Digest::SHA256.hexdigest(canonical_request)
string_to_sign = [
  algorithm,
  timestamp.to_s,
  credential_scope,
  hashed_request_payload,
].join("\n")
puts string_to_sign

# ***** Step 3: Calculate the Signature *****
digest = OpenSSL::Digest.new('sha256')
secret_date = OpenSSL::HMAC.digest(digest, 'TC3' + secret_key, date)
secret_service = OpenSSL::HMAC.digest(digest, secret_date, service)
secret_signing = OpenSSL::HMAC.digest(digest, secret_service, 'tc3_request')
signature = OpenSSL::HMAC.hexdigest(digest, secret_signing, string_to_sign)
puts signature

# ***** Step 4: Concatenate the Authorization *****
authorization = "#{algorithm} Credential=#{secret_id}/#{credential_scope}, SignedHeaders=#{signed_headers}, Signature=#{signature}"
puts authorization

puts 'curl -X POST ' + endpoint %
+ ' -H "Authorization: ' + authorization + '" %

```

```

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

```

DotNet

```

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

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

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

    public static Dictionary<String, String> BuildHeaders(string secretid,
        string secretkey, string service, string endpoint, string region,
        string action, string version, DateTime date, string requestPayload)
    {
        string datestr = date.ToString("yyyy-MM-dd");
        DateTime startTime = new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc);
        long requestTimestamp = (long)Math.Round((date - startTime).TotalMilliseconds, MidpointRounding.AwayFromZero) / 1000;
        // ***** Step 1: Concatenate the CanonicalRequest string *****
        string algorithm = "TC3-HMAC-SHA256";
        string httpRequestMethod = "POST";
        string canonicalUri = "/";
        string canonicalQueryString = "";
        string contentType = "application/json";
        string canonicalHeaders = "content-type:" + contentType + "; charset=utf-8%n" + "host:" + endpoint + "%n";
        string signedHeaders = "content-type;host";
        string hashedRequestPayload = SHA256Hex(requestPayload);
        string canonicalRequest = httpRequestMethod + "%n"
            + canonicalUri + "%n"
            + canonicalQueryString + "%n"
            + canonicalHeaders + "%n"
            + signedHeaders + "%n"
            + hashedRequestPayload;
        Console.WriteLine(canonicalRequest);
        Console.WriteLine("-----");

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

```

```
// ***** Step 3: Calculate the signature *****
byte[] tc3SecretKey = Encoding.UTF8.GetBytes("TC3" + secretkey);
byte[] secretDate = HmacSHA256(tc3SecretKey, Encoding.UTF8.GetBytes(datestr));
byte[] secretService = HmacSHA256(secretDate, Encoding.UTF8.GetBytes(service));
byte[] secretSigning = HmacSHA256(secretService, Encoding.UTF8.GetBytes("tc3_request"));
byte[] signatureBytes = HmacSHA256(secretSigning, Encoding.UTF8.GetBytes(stringToSign));
string signature = BitConverter.ToString(signatureBytes).Replace("-", "").ToLower();
Console.WriteLine(signature);
Console.WriteLine("-----");

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

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

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

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

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

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

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

NodeJS

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

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

function getHash(message, encoding = 'hex') {
```

```

const hash = crypto.createHash('sha256')
return hash.update(message).digest(encoding)
}

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

function main(){

const SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkLPx3*****"
const SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*****"

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

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

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

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

const canonicalRequest = httpRequestMethod + "n"
+ canonicalUri + "n"
+ canonicalQueryString + "n"
+ canonicalHeaders + "n"
+ signedHeaders + "n"
+ hashedRequestPayload
console.log(canonicalRequest)
console.log("-----")

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

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

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

```



```

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

```

C++

```

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

using namespace std;

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

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

string sha256Hex(const string &str)
{
    char buf[3];
    unsigned char hash[SHA256_DIGEST_LENGTH];
    SHA256_CTX sha256;
    SHA256_Init(&sha256);
    SHA256_Update(&sha256, str.c_str(), str.size());
    SHA256_Final(hash, &sha256);
    std::string NewString = "";
    for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
    {
        snprintf(buf, sizeof(buf), "%02x", hash[i]);
        NewString = NewString + buf;
    }
    return NewString;
}

string HmacSha256(const string &key, const string &input)
{
    unsigned char hash[32];

    HMAC_CTX *h;
    #if OPENSSL_VERSION_NUMBER < 0x10100000L
    HMAC_CTX hmac;
    HMAC_CTX_init(&hmac);
    h = &hmac;

```

```
#else
h = HMAC_CTX_new();
#endif

HMAC_Init_ex(h, &key[0], key.length(), EVP_sha256(), NULL);
HMAC_Update(h, ( unsigned char* )&input[0], input.length());
unsigned int len = 32;
HMAC_Final(h, hash, &len);

#if OPENSSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX_cleanup(h);
#else
HMAC_CTX_free(h);
#endif

std::stringstream ss;
ss << std::setfill('0');
for (int i = 0; i < len; i++)
{
ss << hash[i];
}

return (ss.str());
}

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

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

int main()
{
string SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmlPx3*****";
string SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*****";

string service = "cvm";
string host = "cvm.tencentcloudapi.com";
string region = "ap-guangzhou";
string action = "DescribeInstances";
string version = "2017-03-12";
int64_t timestamp = 155113065;
string date = get_data(timestamp);

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

// ***** Step 2: Concatenate the string to sign *****
string algorithm = "TC3-HMAC-SHA256";
string RequestTimestamp = int2str(timestamp);
string credentialScope = date + "/" + service + "/" + "tc3_request";
```

```

string hashedCanonicalRequest = sha256Hex(canonicalRequest);
string stringToSign = algorithm + "\n" + RequestTimestamp + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
cout << stringToSign << endl;
cout << "-----" << endl;

// ***** Step 3: Calculate the signature *****
string kKey = "TC3" + SECRET_KEY;
string kDate = HmacSha256(kKey, date);
string kService = HmacSha256(kDate, service);
string kSigning = HmacSha256(kService, "tc3_request");
string signature = HexEncode(HmacSha256(kSigning, stringToSign));
cout << signature << endl;
cout << "-----" << endl;

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

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

```

Signature Failure

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

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

Signature

Last updated : 2020-09-11 10:10:08

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

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

1. Applying for Security Credentials

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

Security credentials consist of SecretId and SecretKey:

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

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

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

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

2. Generating a Signature

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

Assume that the SecretId and SecretKey are:

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

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

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

Parameter name	Description	Parameter value
Action	Method name	DescribeInstances
SecretId	Key ID	AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
Timestamp	Current timestamp	1465185768
Nonce	Random positive integer	11886
Region	Region where the instance is located	ap-guangzhou
InstanceId.0	ID of the instance to query	ins-09dx96dg
Offset	Offset	0
Limit	Allowed maximum output	20
Version	API version number	2017-03-12

2.1. Sorting Parameters

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

```
{
  'Action' : 'DescribeInstances',
  'InstanceIds.0' : 'ins-09dx96dg',
  'Limit' : 20,
  'Nonce' : 11886,
  'Offset' : 0,
  'Region' : 'ap-guangzhou',
  'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmlPx3*****',
  'Timestamp' : 1465185768,
  'Version' : '2017-03-12',
}
```

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

2.2. Concatenating a Request String

This step generates a request string.

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

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

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

```
Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmlPx3*****
**&Timestamp=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 server: the domain name of the request to view the list of instances (DescribeInstances) is cvm.tencentcloudapi.com. The actual request domain name varies by the module to which the API belongs. For more information, see the instructions of the specific API.
3. Request path: The request path in the current version of TencentCloud API is fixed to /.
4. Request string: the request string generated in the previous step.

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

The concatenation result of the example is:

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmlPx3*****&Timestamp=1465185768&Version=2017-03-12
```

2.4. Generating a Signature String

This step generates a signature string.

First, use the HMAC-SHA1 algorithm to sign the **signature original string** obtained in the previous step, and then encode the generated signature using Base64 to obtain the final signature.

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

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3*****';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmlPx3*****&Timestamp=1465185768&Version=2017-03-12';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

The final signature is:

```
zmmjn35mik6pM3V7sUEuX4wyYM=
```

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 `zmmjn35mik6pM3V7sUEuX4wyYM=`, the final signature string request parameter (Signature) is `zmmjn35mik6pM3V7sUEuX4wyYM%3D`, which will be used to generate the final request URL.

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

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

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

4. Signature Failure

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

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

5. Signature Demo

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

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

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

The final output URL might be: `https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&offset=0&Region=ap-`

guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WfkmLPx3*****&Signature=zmmjn35mikh6pM3V7sUEuX4wyYM%3D&Timestamp=1465185768&Version=2017-03-12 .

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

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

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

Java

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

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

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

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

    public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException {
        StringBuilder url = new StringBuilder("https://cvm.tencentcloudapi.com/?");
        // There is no requirement for the order of the parameters in the actual request URL.
        for (String k : params.keySet()) {
            // The request string needs to be URL encoded. As the key is all in English letters, only the value is URL encoded here.
            url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).append("&");
        }
        return url.toString().substring(0, url.length() - 1);
    }

    public static void main(String[] args) throws Exception {
        TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap 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", "AKIDz8krbsJ5yKBZQpn74WfkmLPx3*****"); // Common parameter
        params.put("Action", "DescribeInstances"); // Common parameter
        params.put("Version", "2017-03-12"); // Common parameter
        params.put("Region", "ap-guangzhou"); // Common parameter
        params.put("Limit", 20); // Business parameter
        params.put("Offset", 0); // Business parameter
        params.put("InstanceIds.0", "ins-09dx96dg"); // Business parameter
        params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3*****", "HmacSHA1")); // Common parameter
        System.out.println(getUrl(params));
    }
}
```

Python

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

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

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3*****"

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

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

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

Golang

```
package main

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

func main() {
    secretId := "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****"
    secretKey := "Gu5t9xGARNpq86cd98joQYCN3*****"
    params := map[string]string{
        "Nonce": "11886",
        "Timestamp": "1465185768",
        "Region": "ap-guangzhou",
        "SecretId": secretId,
        "Version": "2017-03-12",
        "Action": "DescribeInstances",
        "InstanceIds.0": "ins-09dx96dg",
        "Limit": "20",
    }
```



```

"Offset": "0",
}

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

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

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

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

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

```

PHP

```

<?php
$secretId = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****";
$secretKey = "Gu5t9xGARNpq86cd98joQYCN3*****";
$params["Nonce"] = 11886;//rand();
$params["Timestamp"] = 1465185768;//time();
$params["Region"] = "ap-guangzhou";
$params["SecretId"] = $secretId;
$params["Version"] = "2017-03-12";
$params["Action"] = "DescribeInstances";
$params["InstanceIds.0"] = "ins-09dx96dg";
$params["Limit"] = 20;
$params["Offset"] = 0;

ksort($param);

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

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

```

Ruby

```

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

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmlPx3*****"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3*****"

method = 'GET'
endpoint = 'cvm.tencentcloudapi.com'
data = {
  'Action' => 'DescribeInstances',
  'InstanceIds.0' => 'ins-09dx96dg',
  'Limit' => 20,
  'Nonce' => 11886,
  'Offset' => 0,
  'Region' => 'ap-guangzhou',
  'SecretId' => secret_id,
  'Timestamp' => 1465185768, # Time.now.to_i
  'Version' => '2017-03-12',
}
sign = method + endpoint + '/?'
params = []
data.sort.each do |item|
  params << "#{item[0]}=#{item[1]}"
end
sign += params.join('&')
digest = OpenSSL::Digest.new('sha1')
data['Signature'] = Base64.encode64(OpenSSL::HMAC.digest(digest, secret_key, sign))
puts data['Signature']

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

```

DotNet

```

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

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

    public static string MakeSignPlainText(SortedDictionary<string, string> requestParams, string requestMethod, string requestHost, string requestPath)
    {
        string retStr = "";
        retStr += requestMethod;
        retStr += requestHost;
        retStr += requestPath;
        retStr += "?";
        string v = "";
        foreach (string key in requestParams.Keys)

```

```

{
v += string.Format("{0}={1}&", key, requestParams[key]);
}
retStr += v.TrimEnd('&');
return retStr;
}

public static void Main(string[] args)
{
string SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkMLPx3*****";
string SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*****";

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

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

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

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

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

```

NodeJS

```

const crypto = require('crypto');

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

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

```

```
}  
function sha1(secretKey, strsign){  
  let signMethodMap = {'HmacSHA1': 'sha1'};  
  let hmac = crypto.createHmac(signMethodMap['HmacSHA1'], secretKey || '');  
  return hmac.update(Buffer.from(strsign, 'utf8')).digest('base64')  
}  
  
function sort_params(params){  
  let strParam = "";  
  let keys = Object.keys(params);  
  keys.sort();  
  for (let k in keys) {  
    //k = k.replace(/_/g, '.');  
    strParam += ("&" + keys[k] + "=" + params[keys[k]]);  
  }  
  return strParam  
}  
  
function main(){  
  const SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmlPx3*****"  
  const SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*****"  
  
  const endpoint = "cvm.tencentcloudapi.com"  
  const Region = "ap-guangzhou"  
  const Version = "2017-03-12"  
  const Action = "DescribeInstances"  
  const Timestamp = 1465185768  
  // const Timestamp = Math.round(Date.now() / 1000)  
  const Nonce = 11886  
  //const nonce = Math.round(Math.random() * 65535)  
  
  let params = {};  
  params['Action'] = Action;  
  params['InstanceIds.0'] = 'ins-09dx96dg';  
  params['Limit'] = 20;  
  params['Offset'] = 0;  
  params['Nonce'] = Nonce;  
  params['Region'] = Region;  
  params['SecretId'] = SECRET_ID;  
  params['Timestamp'] = Timestamp;  
  params['Version'] = Version;  
  
  strParam = sort_params(params)  
  
  const reqMethod = "GET";  
  const path = "/";  
  strSign = formatSignString(reqMethod, endpoint, path, strParam)  
  console.log(strSign)  
  console.log("-----")  
  
  params['Signature'] = sha1(SECRET_KEY, strSign)  
  console.log(params['Signature'])  
  console.log("-----")  
  
  const req_url = get_req_url(params, endpoint)  
  console.log(params['Signature'])  
  console.log("-----")  
  console.log(req_url)  
}  
main()
```

Responses

Last updated : 2020-03-02 12:05:01

Response for Successful Requests

For example, when calling CAM API (version: 2017-03-12) to view the status of instances (DescribeInstancesStatus), if the request has succeeded, you may see the response as shown below:

```
{
  "Response": {
    "TotalCount": 0,
    "InstanceStatusSet": [],
    "RequestId": "b5b41468-520d-4192-b42f-595cc34b6c1c"
  }
}
```

- The API will return `Response`, which contains `RequestId`, as long as it processes the request. It does not matter if the request is successful or not.
- `RequestId` is the unique ID of an API request. Contact us with this ID when an exception occurs.
- Except for the fixed fields, all fields are action-specified. For the definitions of action-specified fields, see the corresponding API documentation. In this example, `TotalCount` and `InstanceStatusSet` are the fields specified by the API `DescribeInstancesStatus`. `0` `TotalCount` means that the requester owns 0 CVM instance so the `InstanceStatusSet` is empty.

Response for Failed Requests

If the request has failed, you may see the response as shown below:

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

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

Common Error Codes

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

Error Code	Description
AuthFailure.InvalidSecretId	Invalid key (not a TencentCloud API key type).
AuthFailure.MFAFailure	MFA failed.
AuthFailure.SecretIdNotFound	The key does not exist.
AuthFailure.SignatureExpire	Signature expired.

Error Code	Description
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.
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.

General OCR APIs

GeneralBasicOCR

Last updated : 2021-07-08 14:31:51

1. API Description

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

This API is used to detect and recognize characters in an image in the following 20 languages: Chinese, English, Japanese, Korean, Spanish, French, German, Portuguese, Vietnamese, Malay, Russian, Italian, Dutch, Swedish, Finnish, Danish, Norwegian, Hungarian, Thai, and Arabic. Mixed characters in English and each supported language can be recognized together.

It can recognize printed text in paper documents, online images, ads, signboards, menus, video titles, profile photos, etc.

Strengths: it can automatically recognize the text language, return the text box coordinate information, and automatically rotate tilted text to the upright direction.

This API is not fully available for the time being. For more information, please contact your [Tencent Cloud sales rep.](#)

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

We recommend you to use API Explorer

[Try it](#)

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: GeneralBasicOCR.
Version	Yes	String	Common parameter. The value used for this API: 2018-11-19.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product. This API only supports: ap-beijing, ap-guangzhou, ap-hongkong, ap-seoul, ap-shanghai, na-toronto
ImageBase64	No	String	Base64-encoded value of image/PDF. The image/PDF cannot exceed 7 MB after being Base64-encoded. A resolution above 600x800 is recommended. PNG, JPG, JPEG, BMP, and PDF formats are supported.
ImageUrl	No	String	URL address of image/PDF. (This field is not supported outside Chinese mainland) The image/PDF cannot exceed 7 MB after being Base64-encoded. A resolution above 600x800 is recommended. PNG, JPG, JPEG, BMP, and PDF formats are supported. We recommend you store the image in Tencent Cloud, as a Tencent Cloud URL can guarantee higher download speed and stability. The download speed and stability of non-Tencent Cloud URLs may be low.
Scene	No	String	Reserved field.

Parameter Name	Required	Type	Description
LanguageType	No	String	Language to be recognized. The language can be automatically recognized or manually specified. Chinese-English mix (zh) is selected by default. Mixed characters in English and each supported language can be recognized together. Valid values: zh\auto\jap\kor spa\fre\ger\por vie\may\rus\ita hol\swe\fin\dan nor\hun\tha\lat\ara Value meanings: Chinese-English mix, automatic recognition, Japanese, Korean, Spanish, French, German, Portuguese, Vietnamese, Malay, Russian, Italian, Dutch, Swedish, Finnish, Danish, Norwegian, Hungarian, Thai, Latin, Arabic.
IsPdf	No	Boolean	Whether to enable PDF recognition. Default value: false. After this feature is enabled, both images and PDF files can be recognized at the same time.
PdfPageNumber	No	Integer	Page number of the PDF page that needs to be recognized. Only one single PDF page can be recognized. This parameter is valid if the uploaded file is a PDF and the value of the IsPdf parameter is true . Default value: 1.
IsWords	No	Boolean	Whether to return the character information. Default value: false

3. Output Parameters

Parameter Name	Type	Description
TextDetections	Array of TextDetection	Information of recognized text, including the text line content, confidence, text line coordinates, and text line coordinates after rotation correction. For more information, please click the link on the left.
Language	String	Detected language. For more information on the supported languages, please see the description of the LanguageType input parameter.
Angel	Float	Image rotation angle in degrees. 0° indicates horizontal text, a positive value indicates clockwise rotation, and a negative value indicates anticlockwise rotation.
PdfPageSize	Integer	Total number of PDF pages to be returned if the image is a PDF. Default value: 0.
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Recognizing general print ([debugging tool](#))

Input Example

```
https://ocr.tencentcloudapi.com/?Action=GeneralBasicOCR
&ImageUrl=https://xx/a.jpg
&<Common request parameters>
```

Output Example

```
{
  "Response": {
    "Angel": 6.5,
    "TextDetections": [
      {
        "AdvancedInfo": "xx",
        "Polygon": [
          {
```



```
"y": 211,
"x": 450
},
{
  "y": 223,
  "x": 560
},
{
  "y": 244,
  "x": 558
},
{
  "y": 232,
  "x": 448
}
],
"Confidence": 99,
"ItemPolygon": {
  "y": 273,
  "x": 473,
  "Height": 22,
  "Width": 112
},
"WordCoordPoint": [
  {
    "WordCoordinate": [
      {
        "y": 0,
        "x": 0
      }
    ]
  }
],
"DetectedText": "xx",
"Words": [
  {
    "Confidence": 0,
    "Character": "xx"
  }
],
"AdvancedInfo": "xx",
"Polygon": [
  {
    "y": 233,
    "x": 370
  },
  {
    "y": 265,
    "x": 649
  },
  {
    "y": 331,
    "x": 642
  },
  {
    "y": 299,
    "x": 362
  }
],
"Confidence": 99,
"ItemPolygon": {
  "y": 304,
  "x": 396,
  "Width": 282,
  "Height": 68
},
"WordCoordPoint": [
  {
```

```
"WordCoordinate": [
{
  "Y": 0,
  "X": 0
}
],
"DetectedText": "xx",
"Words": [
{
  "Confidence": 0,
  "Character": "xx"
}
],
{
  "AdvancedInfo": "xx",
  "Polygon": [
{
  "Y": 318,
  "X": 402
},
{
  "Y": 339,
  "X": 587
},
{
  "Y": 370,
  "X": 584
},
{
  "Y": 349,
  "X": 398
}
],
  "Confidence": 99,
  "ItemPolygon": {
    "Y": 385,
    "X": 437,
    "Width": 188,
    "Height": 32
  },
  "WordCoordPoint": [
{
  "WordCoordinate": [
{
  "Y": 0,
  "X": 0
}
]
}
],
"DetectedText": "xx",
"Words": [
{
  "Confidence": 0,
  "Character": "xx"
}
],
{
  "AdvancedInfo": "xx",
  "Polygon": [
{
  "Y": 366,
  "X": 386
},
{
  "Y": 390,
```

```
"x": 591
},
{
  "y": 423,
  "x": 587
},
{
  "y": 399,
  "x": 382
}
],
"Confidence": 99,
"ItemPolygon": {
  "y": 435,
  "x": 427,
  "Width": 207,
  "Height": 34
},
"WordCoordPoint": [
  {
    "WordCoordinate": [
      {
        "y": 0,
        "x": 0
      }
    ]
  }
],
"DetectedText": "xx",
"Words": [
  {
    "Confidence": 0,
    "Character": "xx"
  }
],
"RequestId": "xx",
"Language": "xx",
"PdfPageSize": 0
}
```

5. Developer Resources

SDK

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

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

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
FailedOperation.DownloadError	File download failed.
FailedOperation.EmptyImageError	The image is empty.
FailedOperation.EngineRecognizeTimeout	Recognition by the engine timed out.
FailedOperation.ImageDecodeFailed	Image decoding failed.
FailedOperation.ImageNoText	No text is detected in the image.
FailedOperation.LanguageNotSupport	The input language is not supported.
FailedOperation.OcrFailed	OCR failed.
FailedOperation.UnknowError	Unknown error.
FailedOperation.UnOpenError	The service is not activated.
InvalidParameterValue.InvalidParameterValueLimit	Incorrect parameter value.
LimitExceeded.TooLargeFileError	The file is too large.
ResourcesSoldOut.ChargeStatusException	Exceptional billing status.

GeneralAccurateOCR

Last updated : 2021-07-09 15:59:27

1. API Description

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

This API is used to detect and recognize characters in an image. It can recognize Chinese, English, Chinese-English, digits, and special symbols and return the text box positions and characters.

It is suitable for scenarios with a lot of characters in complex layouts and requiring high recognition accuracy, such as examination papers, online images, signboards, and legal documents.

Strengths: compared with general print recognition, it provides higher-precision character recognition services. Its accuracy and recall rate are higher in difficult scenarios such as a large number of characters, long strings of digits, small characters, blurry characters, and tilted text.

This API is not fully available for the time being. For more information, please contact your [Tencent Cloud sales rep.](#)

A maximum of 10 requests can be initiated per second for this API.

We recommend you to use API Explorer

[Try it](#)

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: GeneralAccurateOCR.
Version	Yes	String	Common parameter. The value used for this API: 2018-11-19.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product. This API only supports: ap-beijing, ap-guangzhou, ap-hongkong, ap-shanghai, na-toronto
ImageBase64	No	String	Base64-encoded value of image. The image cannot exceed 7 MB in size after being Base64-encoded. A resolution above 600x800 is recommended. PNG, JPG, JPEG, and BMP formats are supported. Either <code>ImageUrl</code> or <code>ImageBase64</code> of the image must be provided; if both are provided, only <code>ImageUrl</code> will be used.
ImageUrl	No	String	URL address of image. (This field is not supported outside Chinese mainland) The image cannot exceed 7 MB after being Base64-encoded. A resolution above 600x800 is recommended. PNG, JPG, JPEG, and BMP formats are supported. We recommend you store the image in Tencent Cloud, as a Tencent Cloud URL can guarantee higher download speed and stability. The download speed and stability of non-Tencent Cloud URLs may be low.
IsWords	No	Boolean	Whether to return the character information. Default value: <code>false</code>

3. Output Parameters

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

Parameter Name	Type	Description
TextDetections	Array of TextDetection	Information on recognized text, including the text line content, confidence, text line coordinates, and text line coordinates after rotation correction. For more information, please click the link on the left.
Angel	Float	Image rotation angle in degrees. 0° indicates horizontal text. A positive value indicates clockwise rotation. A negative value indicates anticlockwise rotation.
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Recognizing general print (high-precision) ([debugging tool](#))

Input Example

```
https://ocr.tencentcloudapi.com/?Action=GeneralAccurateOCR
&ImageUrl=https://xx/a.jpg
&<Common request parameters>
```

Output Example

```
{
  "Response": {
    "TextDetections": [
      {
        "AdvancedInfo": "xx",
        "Polygon": [
          {
            "Y": 231,
            "X": 449
          },
          {
            "Y": 211,
            "X": 452
          },
          {
            "Y": 227,
            "X": 563
          },
          {
            "Y": 247,
            "X": 560
          }
        ],
        "Confidence": 99,
        "ItemPolygon": {
          "Y": 211,
          "X": 449,
          "Height": 37,
          "Width": 115
        },
        "WordCoordPoint": [
          {
            "WordCoordinate": [
              {
                "Y": 0,
                "X": 0
              }
            ]
          }
        ],
        "DetectedText": "xx",
        "Words": [
          {
            "Confidence": 0,

```

```
"Character": "xx"
}
]
},
{
"AdvancedInfo": "xx",
"Polygon": [
{
"Y": 308,
"X": 361
},
{
"Y": 242,
"X": 368
},
{
"Y": 271,
"X": 648
},
{
"Y": 338,
"X": 641
}
],
"Confidence": 99,
"ItemPolygon": {
"Y": 242,
"X": 361,
"Width": 288,
"Height": 97
},
"WordCoordPoint": [
{
"WordCoordinate": [
{
"Y": 0,
"X": 0
}
]
}
],
"DetectedText": "xx",
"Words": [
{
"Confidence": 0,
"Character": "xx"
}
],
{
"AdvancedInfo": "xx",
"Polygon": [
{
"Y": 349,
"X": 394
},
{
"Y": 316,
"X": 398
},
{
"Y": 337,
"X": 593
},
{
"Y": 371,
"X": 589
}
],
"Confidence": 99,
```

```
"ItemPolygon": {
  "Y": 316,
  "X": 394,
  "Width": 200,
  "Height": 56
},
"WordCoordPoint": [
  {
    "WordCoordinate": [
      {
        "Y": 0,
        "X": 0
      }
    ]
  }
],
"DetectedText": "xx",
"Words": [
  {
    "Confidence": 0,
    "Character": "xx"
  }
],
{
  "AdvancedInfo": "xx",
  "Polygon": [
    {
      "Y": 401,
      "X": 381
    },
    {
      "Y": 368,
      "X": 385
    },
    {
      "Y": 389,
      "X": 594
    },
    {
      "Y": 422,
      "X": 590
    }
  ],
  "Confidence": 98,
  "ItemPolygon": {
    "Y": 368,
    "X": 381,
    "Width": 214,
    "Height": 55
  },
  "WordCoordPoint": [
    {
      "WordCoordinate": [
        {
          "Y": 0,
          "X": 0
        }
      ]
    }
  ],
  "DetectedText": "xx",
  "Words": [
    {
      "Confidence": 0,
      "Character": "xx"
    }
  ]
},
```



```
"RequestId": "xx",
"Angle": 0.0
}
}
```

5. Developer Resources

SDK

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

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

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
FailedOperation.DownloadError	File download failed.
FailedOperation.EmptyImageError	The image is empty.
FailedOperation.EngineRecognizeTimeout	Recognition by the engine timed out.
FailedOperation.ImageBlur	The image is blurry.
FailedOperation.ImageDecodeFailed	Image decoding failed.
FailedOperation.ImageNoText	No text is detected in the image.
FailedOperation.ImageSizeTooLarge	The image is too large. Please see the description of image size limit in the output parameters.
FailedOperation.OcrFailed	OCR failed.
FailedOperation.UnKnowError	Unknown error.
FailedOperation.UnOpenError	The service is not activated.
InvalidParameter.EngineImageDecodeFailed	Image decoding failed.
InvalidParameterValue.InvalidParameterValueLimit	Incorrect parameter value.
LimitExceeded.TooLargeFileError	The file is too large.
ResourcesSoldOut.ChargeStatusException	Exceptional billing status.

Card OCR APIs

BankCardOCR

Last updated : 2021-07-07 11:29:44

1. API Description

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

This API is used to detect and recognize key fields such as the card number, bank information, and expiration date on mainstream bank cards in Mainland China.

This API is not fully available for the time being. For more information, please contact your [Tencent Cloud sales rep.](#)

A maximum of 10 requests can be initiated per second for this API.

We recommend you to use API Explorer

[Try it](#)

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: BankCardOCR.
Version	Yes	String	Common parameter. The value used for this API: 2018-11-19.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product. This API only supports: ap-beijing, ap-guangzhou, ap-hongkong, ap-seoul, ap-shanghai, na-toronto
ImageBase64	No	String	Base64-encoded value of the image. The image cannot exceed 7 MB after being Base64-encoded. A resolution above 500 x 800 is recommended. PNG, JPG, JPEG, and BMP formats are supported. It is recommended that the card part occupy more than 2/3 area of the image. Either the <code>ImageUrl</code> or <code>ImageBase64</code> of the image must be provided. If both are provided, only <code>ImageUrl</code> will be used.
ImageUrl	No	String	URL address of image. (This field is not supported outside Chinese mainland) Supported image formats: PNG, JPG, JPEG. GIF is currently not supported. Supported image size: the downloaded image cannot exceed 7 MB after being Base64-encoded. The download time of the image cannot exceed 3 seconds. We recommend you store the image in Tencent Cloud, as a Tencent Cloud URL can guarantee higher download speed and stability. The download speed and stability of non-Tencent Cloud URLs may be low.
RetBorderCutImage	No	Boolean	Whether to return the bank card image data after preprocessing (precise cropping and alignment). Default value: <code>false</code>
RetCardNoImage	No	Boolean	Whether to return the card number image data after slicing. Default value: <code>false</code>
EnableCopyCheck	No	Boolean	Whether to enable photocopy check. If the input image is a bank card photocopy, an alarm will be returned. Default value: <code>false</code>
EnableReshootCheck	No	Boolean	Whether to enable photograph check. If the input image is a bank card photograph, an alarm will be returned. Default value: <code>false</code>
EnableBorderCheck	No	Boolean	Whether to enable obscured border check. If the input image is a bank card with obscured border, an alarm will be returned. Default value: <code>false</code>

Parameter Name	Required	Type	Description
EnableQualityValue	No	Boolean	Whether to return the image quality value, which measures how clear an image is. Default value: <code>false</code>

3. Output Parameters

Parameter Name	Type	Description
CardNo	String	Card number
BankInfo	String	Bank information
ValidDate	String	Expiration date. Format: 07/2023
CardType	String	Card type
CardName	String	Card name
BorderCutImage	String	Sliced image data Note: this field may return <code>null</code> , indicating that no valid values can be obtained.
CardNoImage	String	Card number image data Note: this field may return <code>null</code> , indicating that no valid values can be obtained.
WarningCode	Array of Integer	Warning code: -9110: the bank card date is invalid. -9111: the bank card border is incomplete. -9112: the bank card image is reflective. -9113: the bank card image is a photocopy. -9114: the bank card image is a photograph. Multiple warning codes may be returned at a time. Note: this field may return <code>null</code> , indicating that no valid values can be obtained.
QualityValue	Integer	Image quality value, which is returned when <code>EnableQualityValue</code> is set to <code>true</code> . The smaller the value, the less clear the image is. Value range: 0–100 (a threshold greater than or equal to 50 is recommended.) Note: this field may return <code>null</code> , indicating that no valid value is obtained.
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Recognizing a bank card

Input Example

```
https://ocr.tencentcloudapi.com/?Action=BankCardOCR
&ImageUrl=https://xx/a.jpg
&<Common request parameters>
```

Output Example

```
{
  "Response": {
    "BankInfo": "China Merchants Bank (03080000)",
    "ValidDate": "07/2023",
    "CardType": "Credit card",
    "CardName": "China Merchants Bank credit card",
    "BorderCutImage": null,
    "CardNoImage": null,
    "QualityValue": 80,
    "WarningCode": [
      -9110
    ]
  }
}
```

```
],
"CardNo": "6225768888888888",
"RequestId": "68f8cbf-6004-0111a-ac18-6f1a9308fs100mab"
}
```

5. Developer Resources

SDK

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

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

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
FailedOperation.DownloadError	File download failed.
FailedOperation.ImageDecodeFailed	Image decoding failed.
FailedOperation.OcrFailed	OCR failed.
FailedOperation.UnknowError	Unknown error.
FailedOperation.UnOpenError	The service is not activated.
InvalidParameter.EngineImageDecodeFailed	Image decoding failed.
InvalidParameterValue.InvalidParameterValueLimit	Incorrect parameter value.
LimitExceeded.TooLargeFileError	The file is too large.
ResourcesSoldOut.ChargeStatusException	Exceptional billing status.

MLIDCardOCR

Last updated : 2021-05-25 11:00:55

1. API Description

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

This API is used to recognize a Malaysian identity card. Recognizable fields include identity card number, name, gender, and address. It has the features of cropping identity photos and alarming for photographed or photocopied documents.

This API is not fully available for the time being. For more information, please contact your [Tencent Cloud sales rep.](#)

A maximum of 5 requests can be initiated per second for this API.

We recommend you to use API Explorer

[Try it](#)

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: MLIDCardOCR.
Version	Yes	String	Common parameter. The value used for this API: 2018-11-19.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product. This API only supports: ap-guangzhou, ap-singapore
ImageBase64	No	String	Base64-encoded value of an image. Supported image formats: PNG, JPG, JPEG. GIF is currently not supported. Supported image size: the downloaded image cannot exceed 7 MB after being Base64-encoded. The download time of the image cannot exceed 3 seconds.
ImageUrl	No	String	URL of an image. (This field is not supported outside the Chinese mainland) Supported image formats: PNG, JPG, JPEG. GIF is currently not supported. Supported image size: the downloaded image cannot exceed 7 MB after being Base64-encoded. The download time of the image cannot exceed 3 seconds. We recommend storing the image in Tencent Cloud, as a Tencent Cloud URL can guarantee higher download speed and stability. The download speed and stability of non-Tencent Cloud URLs may be low.
RetImage	No	Boolean	Whether to return an image

3. Output Parameters

Parameter Name	Type	Description
ID	String	Identity card number
Name	String	Name
Address	String	Address
Sex	String	Gender
Warn	Array of Integer	Alarm code

		-9103 Alarm for photographed document -9102 Alarm for photocopied document -9106 Alarm for covered card
Image	String	Identity photo
AdvancedInfo	String	Extended field: { ID:{ Confidence:0.9999 }, Name:{ Confidence:0.9996 } }
Type	String	Certificate type MyKad ID card MyPR Permanent resident card MyTentera Military identity card MyKAS Temporary ID card POLIS Police card IKAD Work permit MyKid Kid card
Birthday	String	Date of birth (currently, this field is only supported for IKAD).
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Recognizing a Malaysian identity card

Input Example

```
https://ocr.tencentcloudapi.com/?Action=MLIDCardOCR
&ImageUrl=https://xx/a.jpg
&<Common request parameters>
```

Output Example

```
{
  "Response": {
    "Name": "KAVIN ONG KHI MN",
    "ID": "710716-08-6085",
    "Address": "NO 11 PERSIARN PERAJRIT 4 TAMA PERAK 31400 IPOH ERAK",
    "Sex": "LEAKI",
    "Birthday": "",
    "Warn": [],
    "Image": "",
    "AdvancedInfo": "{ \"ID\": { \"Confidence\": \"1.0000\" }, \"Name\": { \"Confidence\": \"0.9996\" }, \"Address\": { \"Confidence\": \"0.9997\" }, \"Sex\": { \"Confidence\": \"0.9999\" } }",
    "Type": "MyKad",
    "RequestId": "c969da05-54e3-4d0a-a55d-b3ef90d4ebf5"
  }
}
```

5. Developer Resources

SDK

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

- [Tencent Cloud SDK 3.0 for Python](#)

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

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
FailedOperation.DownloadError	File download failed.
FailedOperation.ImageDecodeFailed	Image decoding failed.
FailedOperation.OcrFailed	OCR failed.
FailedOperation.UnKnowError	Unknown error.
FailedOperation.UnOpenError	The service is not activated.
InvalidParameterValue.InvalidParameterValueLimit	Incorrect parameter value.
LimitExceeded.TooLargeFileError	The file is too large.

MLIDPassportOCR

Last updated : 2021-07-07 11:27:48

1. API Description

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

This API is used to recognize a passport issued in Hong Kong/Macao/Taiwan (China) or other countries/regions. Recognizable fields include passport ID, name, date of birth, gender, expiration date, issuing country/region, and nationality. It has the features of cropping identity photos and alarming for photographed or photocopied documents.

This API is not fully available for the time being. For more information, please contact your [Tencent Cloud sales rep.](#)

A maximum of 5 requests can be initiated per second for this API.

We recommend you to use API Explorer

[Try it](#)

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: MLIDPassportOCR.
Version	Yes	String	Common parameter. The value used for this API: 2018-11-19.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product. This API only supports: ap-guangzhou, ap-singapore
ImageBase64	Yes	String	Base64-encoded value of image. The image cannot exceed 7 MB in size after being Base64-encoded. A resolution above 500x800 is recommended. PNG, JPG, JPEG, and BMP formats are supported. It is recommended that the card part occupies more than 2/3 area of the image.
RetImage	No	Boolean	Whether to return an image. Default value: false.

3. Output Parameters

Parameter Name	Type	Description
ID	String	Passport ID
Name	String	Name
DateOfBirth	String	Date of birth
Sex	String	Gender (F: female, M: male)
DateOfExpiration	String	Expiration date
IssuingCountry	String	Issuing country
Nationality	String	Nationality
Warn	Array of Integer	Alarm code -9103 Alarm for spoofed card

		-9102 Alarm for photocopied card -9106 Alarm for covered card
Image	String	Identity photo
AdvancedInfo	String	Extended field: { ID:{ Confidence:0.9999 }, Name:{ Confidence:0.9996 } }
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

debugging-tool)">

Example1 Recognizing a non-mainland China passport ([debugging tool](#))

Input Example

```
https://ocr.tencentcloudapi.com/?Action=MLIDPassportOCR
&ImageBase64=Base64 encoding
&<Common request parameters>
```

Output Example

```
{
  "Response": {
    "ID": "E6918C",
    "Name": "LIM HEG CHUN STEE",
    "IssuingCountry": "SGP",
    "Nationality": "SGP",
    "DateOfBirth": "",
    "Sex": "M",
    "DateOfExpiration": "230414",
    "Warn": [],
    "Image": "",
    "AdvancedInfo": "{¥IssuingCountry¥: {¥Confidence¥: ¥0.9500¥}, ¥Name¥: {¥Confidence¥: ¥0.9500¥}, ¥ID¥: {¥Confidence¥: ¥0.9500¥}, ¥Nationality¥: {¥Confidence¥: ¥0.9500¥}, ¥Sex¥: {¥Confidence¥: ¥0.9500¥}, ¥DateOfExpiration¥: {¥Confidence¥: ¥0.9500¥}",
    "RequestId": "0ee989d3-d064-45ec-bccb-63f5064247b4"
  }
}
```

5. Developer Resources

SDK

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

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

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
FailedOperation.DownloadError	File download failed.
FailedOperation.ImageDecodeFailed	Image decoding failed.
FailedOperation.NoPassport	Not a passport.
FailedOperation.OcrFailed	OCR failed.
FailedOperation.UnKnowError	Unknown error.
FailedOperation.UnOpenError	The service is not activated.
InvalidParameterValue.InvalidParameterValueLimit	Incorrect parameter value.
LimitExceeded.TooLargeFileError	The file is too large.
ResourcesSoldOut.ChargeStatusException	Exceptional billing status.

HKIDCardOCR

Last updated : 2021-04-12 14:14:14

1. API Description

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

This API is used to recognize key fields on the photo side of a Hong Kong (China) identity card, including name in Chinese, name in English, telecode for name, date of birth, gender, document symbol, date of the first issue, date of the last receipt, identity card number, and permanent residency attribute. It can check for card authenticity and crop the identity photo.

This API is not fully available for the time being. For more information, please contact your [Tencent Cloud sales rep](#).

A maximum of 5 requests can be initiated per second for this API.

We recommend you to use API Explorer

[Try it](#)

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: HKIDCardOCR.
Version	Yes	String	Common parameter. The value used for this API: 2018-11-19.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product. This API only supports: ap-beijing, ap-guangzhou, ap-hongkong, ap-seoul, ap-shanghai, na-toronto
DetectFake	Yes	Boolean	Whether to check for authenticity.
ReturnHeadImage	Yes	Boolean	Whether to return identity photo.
ImageBase64	No	String	Base64-encoded value of image. Supported image formats: PNG, JPG, JPEG. GIF is currently not supported. Supported image size: the downloaded image cannot exceed 3 MB in size after being Base64-encoded. The download time of the image cannot exceed 3 seconds.
ImageUrl	No	String	URL address of image. (This field is not supported outside Chinese mainland) Supported image formats: PNG, JPG, JPEG. GIF is currently not supported. Supported image size: the downloaded image cannot exceed 3 MB after being Base64-encoded. The download time of the image cannot exceed 3 seconds. We recommend you store the image in Tencent Cloud, as a Tencent Cloud URL can guarantee higher download speed and stability. The download speed and stability of non-Tencent Cloud URLs may be low.

3. Output Parameters

Parameter Name	Type	Description
CnName	String	Name in Chinese
EnName	String	Name in English
TelexCode	String	Telecode for the name in Chinese

Sex	String	Gender. Valid values: Male, Female
Birthday	String	Date of birth
Permanent	Integer	Permanent identity card. 0: non-permanent; 1: permanent; -1: unknown.
IdNum	String	Identity card number
Symbol	String	Document symbol, i.e., the symbol under the date of birth, such as "***AZ"
FirstIssueDate	String	First issue date
CurrentIssueDate	String	Last receipt date
FakeDetectResult	Integer	Authenticity check. 0: unable to judge (because the image is blurred, incomplete, reflective, too dark, etc.); 1: forged; 2: authentic. Note: this field may return null, indicating that no valid values can be obtained.
HeadImage	String	Base64-encoded identity photo Note: this field may return null, indicating that no valid values can be obtained.
WarningCode	Array of Integer	Multiple alarm codes. If the ID card is spoofed, photocopied, or doctored, the corresponding alarm code will be returned. -9102: alarm for photocopied document -9103: alarm for spoofed document -9104: alarm for doctored document -9105: alarm for forged document
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Recognizing a Hong Kong (China) identity card

Input Example

```
https://ocr.tencentcloudapi.com/?Action=HKIDCardOCR
&ImageUrl=https://xx/a.jpg
&DetectFake=true
&ReturnHeadImage=false
&<Common request parameters>
```

Output Example

```
{
  "Response": {
    "CnName": "SAN, Nan",
    "EnName": "SAN, Nan",
    "TelexCode": "30000000000",
    "Sex": "Female",
    "Birthday": "01-01-2001",
    "Permanent": 1,
    "IdNum": "C00000(E)",
    "Symbol": "***AZ",
    "FirstIssueDate": "(09-99)",
    "CurrentIssueDate": "23-09-10",
    "FakeDetectResult": 1,
    "WarningCode": [
      -9102,
      -9103
    ],
    "HeadImage": "xx",
  }
}
```

```
"RequestId": "fba1c9ad-aeb3-4418-9ecf-80ab1b5fc875"  
}  
}
```

5. Developer Resources

SDK

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

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

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
FailedOperation.DownloadError	File download failed.
FailedOperation.ImageDecodeFailed	Image decoding failed.
FailedOperation.NoHKIDCard	Not a Hong Kong identity card.
FailedOperation.OcrFailed	OCR failed.
FailedOperation.UnKnowError	Unknown error.
FailedOperation.UnOpenError	The service is not activated.
InvalidParameterValue.InvalidParameterValueLimit	Incorrect parameter value.
LimitExceeded.TooLargeFileError	The file is too large.
ResourcesSoldOut.ChargeStatusException	Exceptional billing status.

Industry-specific Document Recognition APIs

TableOCR

Last updated : 2021-04-12 14:18:28

1. API Description

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

This API is used to detect and recognize Chinese and English forms in images. It can return the text content of each cell and save the recognition result as Excel.

This API is not fully available for the time being. For more information, please contact your [Tencent Cloud sales rep.](#)

A maximum of 10 requests can be initiated per second for this API.

We recommend you to use API Explorer

[Try it](#)

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: TableOCR.
Version	Yes	String	Common parameter. The value used for this API: 2018-11-19.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product. This API only supports: ap-beijing, ap-guangzhou, ap-hongkong, ap-shanghai, na-toronto
ImageBase64	No	String	Base64-encoded value of image. Supported image formats: PNG, JPG, JPEG. GIF is not supported at present. Supported image size: the downloaded image cannot exceed 3 MB in size after being Base64-encoded. The download time of the image cannot exceed 3 seconds. Either <code>ImageUrl</code> or <code>ImageBase64</code> of the image must be provided; if both are provided, only <code>ImageUrl</code> will be used.
ImageUrl	No	String	URL address of image. (This field is not supported outside Chinese mainland) Supported image formats: PNG, JPG, JPEG. GIF is currently not supported. Supported image size: the downloaded image cannot exceed 3 MB after being Base64-encoded. The download time of the image cannot exceed 3 seconds. We recommend you store the image in Tencent Cloud, as a Tencent Cloud URL can guarantee higher download speed and stability. The download speed and stability of non-Tencent Cloud URLs may be low.

3. Output Parameters

Parameter Name	Type	Description
TextDetections	Array of TextTable	Recognized text. For more information, please click the link on the left
Data	String	Base64-encoded Excel data.

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

4. Example

Example1 Recognizing a form

Input Example

```
https://ocr.tencentcloudapi.com/?Action=TableOCR
&ImageUrl=http%3A%2F%2Fimg3.redocn.com%2Ftupian%2F20180301%2Fkongbaiqiuzhijianlibiaogesheji_9230633.jpg
&<Common request parameters>
```

Output Example

```
{
  "Response": {
    "TextDetections": [
      {
        "ColTL": -1,
        "RowTL": -1,
        "ColBR": -1,
        "RowBR": -1,
        "Text": "Item number",
        "Type": "header",
        "Confidence": 99,
        "Polygon": [
          {
            "X": 3378,
            "Y": 363
          },
          {
            "X": 3383,
            "Y": 646
          },
          {
            "X": 3304,
            "Y": 648
          },
          {
            "X": 3299,
            "Y": 365
          }
        ],
        "AdvancedInfo": "{}"
      },
      {
        "ColTL": -1,
        "RowTL": -1,
        "ColBR": -1,
        "RowBR": -1,
        "Text": "Dish/:11181 1 19 Medical 101",
        "Type": "header",
        "Confidence": 53,
        "Polygon": [
          {
            "X": 3470,
            "Y": 1285
          },
          {
            "X": 3501,
            "Y": 3092
          },
          {
            "X": 3361,
            "Y": 3094
          }
        ]
      }
    ]
  }
}
```

```

{
  "X": 3330,
  "Y": 1287
},
],
"AdvancedInfo": "{}"
},
{
  "ColTL": -1,
  "RowTL": -1,
  "ColBr": -1,
  "RowBr": -1,
  "Text": "Export declaration form",
  "Type": "header",
  "Confidence": 99,
  "Polygon": [
    {
      "X": 3201,
      "Y": 2046
    },
    {
      "X": 3211,
      "Y": 2618
    },
    {
      "X": 3169,
      "Y": 2618
    },
    {
      "X": 3159,
      "Y": 2046
    }
  ],
  "AdvancedInfo": "{}"
},
{
  "ColTL": -1,
  "RowTL": -1,
  "ColBr": -1,
  "RowBr": -1,
  "Text": "Production and sales company name (required)",
  "Type": "header",
  "Confidence": 98,
  "Polygon": [
    {
      "X": 2525,
      "Y": 184
    },
    {
      "X": 2532,
      "Y": 594
    },
    {
      "X": 2491,
      "Y": 595
    },
    {
      "X": 2484,
      "Y": 185
    }
  ],
  "AdvancedInfo": "{}"
}
],
"Data": "bWxQSwECFAAUAAAACAAAAAAcXSLdcBAAD9BQAADQAAAAAAAAAAAAAAAAAAcLAAAEgwvc3R5bGVzLnhtbFBLAQIUABQAAAAIAAAAAAC8TIYcSAYAAP4kAAATAAAAAAAAAAAAAAAAAAAK8uAAB4bC90aGVtZS90aGVtZTEueG1sUEsBAhQAFAAAAAAGAAAAANG5P9oECwAAU0AABgAAAAAAAAAAAAAAAAAAKDUAAHsL3dvcmtzaGVldHMvc2hLZXQyLnhtbFBLAQIUABQAAAAIAAAAAADjsY7PqAMAAE8PAAAYAAAAAAAAAAAAAAAAAGJAAAB4bC93b3Jrc2hLZXRzL3NoZWV0MS54bWxQSwUGAAAAAAAwADAALAwAAQEQAAAA",
"RequestId": "98f8fcfb-933a-4e95-ac48-6f1a9308fs53"
}

```


5. Developer Resources

SDK

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

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

Command Line Interface

- [Tencent Cloud CLI 3.0](#)

6. Error Code

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

Error Code	Description
FailedOperation.DownloadError	File download failed.
FailedOperation.EmptyImageError	The image is empty.
FailedOperation.ImageDecodeFailed	Image decoding failed.
FailedOperation.ImageNoText	No text is detected in the image.
FailedOperation.LanguageNotSupport	The input language is not supported.
FailedOperation.OcrFailed	OCR failed.
FailedOperation.UnKnowError	Unknown error.
FailedOperation.UnOpenError	The service is not activated.
InvalidParameterValue.InvalidParameterValueLimit	Incorrect parameter value.
LimitExceeded.TooLargeFileError	The file is too large.
ResourcesSoldOut.ChargeStatusException	Exceptional billing status.

Data Types

Last updated : 2021-07-07 11:27:49

Coord

Coordinates

Used by actions: GeneralAccurateOCR, GeneralBasicOCR, TableOCR.

Name	Type	Description
X	Integer	Horizontal coordinate
Y	Integer	Vertical coordinate

DetectedWordCoordPoint

Coordinates of a word's four corners in a clockwise order on the input image, starting from the upper-left corner

Used by actions: GeneralAccurateOCR, GeneralBasicOCR.

Name	Type	Description
WordCoordinate	Array of Coord	Coordinates of a word's four corners in a clockwise order on the input image, starting from the upper-left corner

DetectedWords

Information about a character detected, including the character itself and its confidence

Used by actions: GeneralAccurateOCR, GeneralBasicOCR.

Name	Type	Description
Confidence	Integer	Confidence. Value range: 0-100
Character	String	A possible character

ItemCoord

Pixel coordinates of the text line in the image after rotation correction, which is in the format of (X-coordinate of top-left point, Y-coordinate of top-left point, width, height) .

Used by actions: GeneralAccurateOCR, GeneralBasicOCR.

Name	Type	Description
X	Integer	X-coordinate of top-left point.
Y	Integer	Y-coordinate of top-left point.
Width	Integer	Width
Height	Integer	Height

TextDetection

OCR result.

Used by actions: GeneralAccurateOCR, GeneralBasicOCR.

Name	Type	Description
DetectedText	String	Recognized text line content.
Confidence	Integer	Confidence. Value range: 0-100.
Polygon	Array of Coord	Text line coordinates, which are represented as 4 vertex coordinates. Note: this field may return null, indicating that no valid values can be obtained.
AdvancedInfo	String	Extended field. The paragraph information <code>Parag</code> returned by the <code>GeneralBasicOcr</code> API contains <code>ParagNo</code> .
ItemPolygon	ItemCoord	Pixel coordinates of the text line in the image after rotation correction, which is in the format of (X-coordinate of top-left point, Y-coordinate of top-left point, width, height).
Words	Array of DetectedWords	Information about a character, including the character itself and its confidence. Supported APIs: <code>GeneralBasicOCR</code> , <code>GeneralAccurateOCR</code>
WordCoordPoint	Array of DetectedWordCoordPoint	Coordinates of a word's four corners on the input image. Supported APIs: <code>GeneralBasicOCR</code> , <code>GeneralAccurateOCR</code>

TextTable

Form recognition result.

Used by actions: TableOCR.

Name	Type	Description
ColTl	Integer	Column index of the top-left corner of the cell.
RowTl	Integer	Row index of the top-left corner of the cell.
ColBr	Integer	Column index of the bottom-right corner of the cell.
RowBr	Integer	Row index of the bottom-right corner of the cell.
Text	String	Cell text
Type	String	Cell type. Valid values: body, header, footer
Confidence	Integer	Confidence. Value range: 0-100
Polygon	Array of Coord	Text line coordinates, which are represented as 4 vertex coordinates.
AdvancedInfo	String	Extended field

Error Codes

Last updated : 2021-07-07 11:27:49

Feature Description

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

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

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

Error Code List

Common Error Codes

Error Code	Description
ActionOffline	This API has been deprecated.
AuthFailure.InvalidAuthorization	Authorization in the request header is invalid.
AuthFailure.InvalidSecretId	Invalid key (not a TencentCloud API key type).
AuthFailure.MFAFailure	MFA failed.
AuthFailure.SecretIdNotFound	Key does not exist. Check if the key has been deleted or disabled in the console, and if not, check if the key is correctly entered. Note that whitespaces should not exist before or after the key.
AuthFailure.SignatureExpire	Signature expired. Timestamp and server time cannot differ by more than five minutes. Please ensure your current local time matches the standard time.
AuthFailure.SignatureFailure	Invalid signature. Signature calculation error. Please ensure you've followed the signature calculation process described in the Signature API documentation.
AuthFailure.TokenFailure	Token error.
AuthFailure.UnauthorizedOperation	The request is not authorized. For more information, see the CAM documentation.
DryRunOperation	DryRun Operation. It means that the request would have succeeded, but the DryRun parameter was used.
FailedOperation	Operation failed.
InternalError	Internal error.
InvalidAction	The API does not exist.
InvalidParameter	Incorrect parameter.
InvalidParameterValue	Invalid parameter value.
InvalidRequest	The multipart format of the request body is incorrect.
LimitExceeded	Quota limit exceeded.
MissingParameter	A parameter is missing.
NoSuchProduct	The product does not exist.

NoSuchVersion	The API version does not exist.
RequestLimitExceeded	The number of requests exceeds the frequency limit.
RequestSizeLimitExceeded	The request size exceeds the upper limit.
ResourceInUse	Resource is in use.
ResourceInsufficient	Insufficient resource.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	Resource is unavailable.
ServiceUnavailable	Service is unavailable now.
UnauthorizedOperation	Unauthorized operation.
UnknownParameter	Unknown parameter.
UnsupportedOperation	Unsupported operation.
UnsupportedProtocol	HTTP(S) request protocol error; only GET and POST requests are supported.
UnsupportedRegion	API does not support the requested region.

Service Error Codes

Error Code	Description
FailedOperation.DownloadError	File download failed.
FailedOperation.EmptyImageError	The image is empty.
FailedOperation.EngineRecognizeTimeout	Recognition by the engine timed out.
FailedOperation.ImageBlur	The image is blurry.
FailedOperation.ImageDecodeFailed	Image decoding failed.
FailedOperation.ImageNoText	No text is detected in the image.
FailedOperation.ImageSizeTooLarge	The image is too large. Please see the description of image size limit in the output parameters.
FailedOperation.LanguageNotSupport	The input language is not supported.
FailedOperation.NoHKIDCard	Not a Hong Kong identity card.
FailedOperation.NoPassport	Not a passport.
FailedOperation.OcrFailed	OCR failed.
FailedOperation.UnKnowError	Unknown error.
FailedOperation.UnOpenError	The service is not activated.
InvalidParameter.EngineImageDecodeFailed	Image decoding failed.
InvalidParameterValue.InvalidParameterValueLimit	Incorrect parameter value.
LimitExceeded.TooLargeFileError	The file is too large.
ResourcesSoldOut.ChargeStatusException	Exceptional billing status.