人脸核身

API 文档

产品文档

腾讯云

文档目录

API 文档

History

最近更新时间：2024-05-31 11:01:14

Release 28

Release time: 2024-05-20 23:03:02

Release updates:

Improvement to existing documentation.

New data structures:

- MainlandIDCard

Modified data structures:

- NormalCardInfo
  - New members:MainlandIDCard

Release 27

Release time: 2024-05-20 22:44:26

Release updates:

Improvement to existing documentation.

Deleted data structures:

- MainlandIDCard

Modified data structures:

- NormalCardInfo
  - **Deleted members:** MainlandIDCard

Release 26

Release time: 2024-05-20 22:42:09

Release updates:

Improvement to existing documentation.

New data structures:

- MainlandIDCard

Modified data structures:

- NormalCardInfo
  - New members:MainlandIDCard

Release 25

Release time: 2024-05-20 22:17:09

Release updates:

Improvement to existing documentation.

Deleted data structures:

- MainlandIDCard

Modified data structures:

- NormalCardInfo
  - **Deleted members:** MainlandIDCard

Release 24

Release time: 2024-05-16 19:56:10

Release updates:

Improvement to existing documentation.

New data structures:

- MainlandIDCard

Modified data structures:

- NormalCardInfo
  - New members:MainlandIDCard

Release 23

Release time: 2024-05-16 19:40:33

Release updates:

Improvement to existing documentation.

New APIs:

- ApplyCardVerification
- ApplyLivenessToken
- ApplyWebVerificationBizTokenIntl
- ApplyWebVerificationToken
- CompareFaceLiveness
- CreateUploadUrl
- DetectReflectLivenessAndCompare
- GenerateReflectSequence
- GetCardVerificationResult
- GetFaceIdResultIntl
- GetFaceIdTokenIntl
- GetLivenessResult
- GetWebVerificationResult
- GetWebVerificationResultIntl
- LivenessCompare
- VideoLivenessCompare

New data structures:

- OCRResult
- VerificationDetail
- WebVerificationConfigIntl

Release 22

Release time: 2024-05-16 15:19:54

Release updates:

Improvement to existing documentation.

New data structures:

- MacaoIDCard
- NormalCardInfo
- NormalHKIDCard
- NormalIndonesiaIDCard
- NormalMLIDCard
- NormalThailandIDCard

Modified data structures:

- CardInfo
  - New members:MacaoIDCard
- CardVerifyResult
  - New members:IsEdit, NormalCardInfo, WarnCardInfos
  - **Deprecate members:** CardInfo

Release 21

Release time: 2024-05-16 15:09:34

Release updates:

Improvement to existing documentation.

New APIs:

- ApplySdkVerificationToken

Release 20

Release time: 2024-05-16 14:17:28

Release updates:

Improvement to existing documentation.

Deleted APIs:

- ApplyCardVerification

- ApplyLivenessToken
- ApplySdkVerificationToken
- ApplyWebVerificationBizTokenIntl
- ApplyWebVerificationToken
- CompareFaceLiveness
- CreateUploadUrl
- DetectReflectLivenessAndCompare
- GenerateReflectSequence
- GetCardVerificationResult
- GetFaceIdResultIntl
- GetFaceIdTokenIntl
- GetLivenessResult
- GetWebVerificationResult
- GetWebVerificationResultIntl
- LivenessCompare
- VideoLivenessCompare

Deleted data structures:

- OCRResult
- VerificationDetail
- WebVerificationConfigIntl

Release 19

Release time: 2024-05-15 11:43:57

Release updates:

Improvement to existing documentation.

Modified data structures:

- WebVerificationConfigIntl
  - New members:AutoDowngrade

Release 18

Release time: 2024-04-03 14:53:37

Release updates:

Improvement to existing documentation.

Modified data structures:

- OCRResult
  - New members:CardCutImageBase64, CardBackCutImageBase64

Release 17

Release time: 2024-04-03 11:45:09

Release updates:

Improvement to existing documentation.

Modified data structures:

- WebVerificationConfigIntl
  - New members:SecurityLevel, SkipPrivacyPolicy, IdCardCutReturn, ThemeColor, Language

Release 16

Release time: 2024-02-23 16:25:37

Release updates:

Improvement to existing documentation.

New APIs:

- ApplyCardVerification
- GetCardVerificationResult

Release 15

Release time: 2024-02-22 17:14:27

Release updates:

Improvement to existing documentation.

New data structures:

- Address

Modified data structures:

- GeneralCard
  - New members:Address
- WebVerificationConfigIntl
  - New members:AutoSkipStartPage, DisableCheckOcrWarnings

Release 14

Release time: 2024-02-21 17:14:48

Release updates:

Improvement to existing documentation.

Modified data structures:

- CardVerifyResult
  - New members:CardInfo

Release 13

Release time: 2023-12-21 11:10:27

Release updates:

Improvement to existing documentation.

Modified APIs:

- ApplyWebVerificationBizTokenIntl
  - **Modified input parameters:** CompareImageBase64

Release 12

Release time: 2023-12-14 10:31:11

Release updates:

Improvement to existing documentation.

Modified APIs:

- GetWebVerificationResultIntl
  - New output parameters:OCRResult, Extra

New data structures:

- CardInfo
- GeneralCard
- HKIDCard
- IndonesiaDrivingLicense
- IndonesiaIDCard
- InternationalIDPassport
- MLIDCard
- OCRResult
- PhilippinesDrivingLicense
- PhilippinesSSSID
- PhilippinesTinID
- PhilippinesUMID
- PhilippinesVoteID
- SingaporeIDCard
- ThailandIDCard

Modified data structures:

- WebVerificationConfigIntl
  - New members:CheckMode, IDCardType

Release 11

Release time: 2023-10-25 18:08:10

Release updates:

Improvement to existing documentation.

New APIs:

- CompareFaceLiveness

Release 10

Release time: 2023-09-14 19:16:04

Release updates:

Improvement to existing documentation.

Modified APIs:

- ApplyWebVerificationBizTokenIntl
  - New input parameters:Config
  - New output parameters:VerificationURL
  - **Deprecate output parameters:** VerificationUrl

New data structures:

- WebVerificationConfigIntl

Release 9

Release time: 2023-08-09 15:01:28

Release updates:

Improvement to existing documentation.

New APIs:

- ApplyWebVerificationBizTokenIntl
- GetWebVerificationResultIntl

Release 8

Release time: 2023-06-21 14:27:26

Release updates:

Improvement to existing documentation.

Modified APIs:

- ApplySdkVerificationToken
  - New input parameters:CheckMode, SecurityLevel, CompareImage

Release 7

Release time: 2023-04-17 17:05:09

Release updates:

Improvement to existing documentation.

New APIs:

- GetFaceIdResultIntl
- GetFaceIdTokenIntl

Release 6

Release time: 2022-09-05 16:36:53

Release updates:

Improvement to existing documentation.

New APIs:

- ApplyLivenessToken
- ApplySdkVerificationToken
- ApplyWebVerificationToken
- GetLivenessResult
- GetSdkVerificationResult
- GetWebVerificationResult

New data structures:

- CardVerifyResult
- CompareResult
- VerificationDetail

Release 5

Release time: 2022-08-12 14:32:32

Release updates:

Improvement to existing documentation.

New APIs:

- VideoLivenessCompare

New data structures:

- FileInfo

Release 4

Release time: 2022-06-09 15:12:17

Release updates:

Improvement to existing documentation.

New APIs:

- CreateUploadUrl
- GenerateReflectSequence

Release 3

Release time: 2022-01-04 10:32:07

Release updates:

Improvement to existing documentation.

New APIs:

- DetectReflectLivenessAndCompare

Release 2

Release time: 2021-12-09 16:08:47

Release updates:

Improvement to existing documentation.

Modified APIs:

- LivenessCompare

- New input parameters:ImageUrl, VideoUrl
- **Modified input parameters:** ImageBase64, VideoBase64

Existing Release

Release time: 2020-12-09 15:17:18

Existing APIs/data structures are as follows:

Improvement to existing documentation.

Existing APIs:

- LivenessCompare

Introduction

最近更新时间：2024-04-03 11:45:22

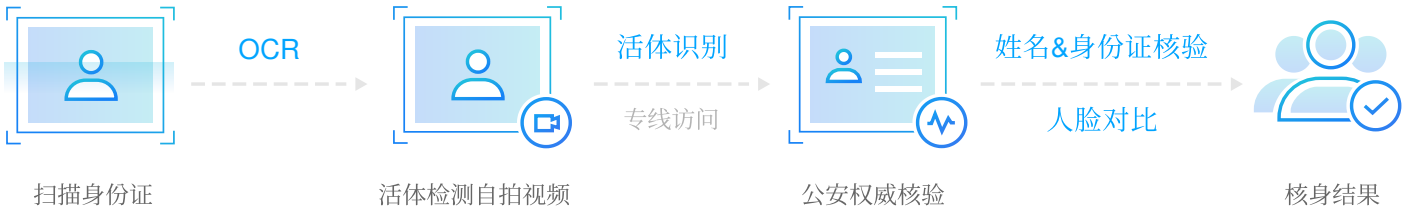
产品概述

腾讯云慧眼人脸核身（原金融级身份认证身份）方案，是依托于证件OCR识别、活体检测、1：1人脸比对等AI技术，实现自然人真实身份核验的产品。秒级确认用户身份，帮助提升业务办理效率，降低人力成本。

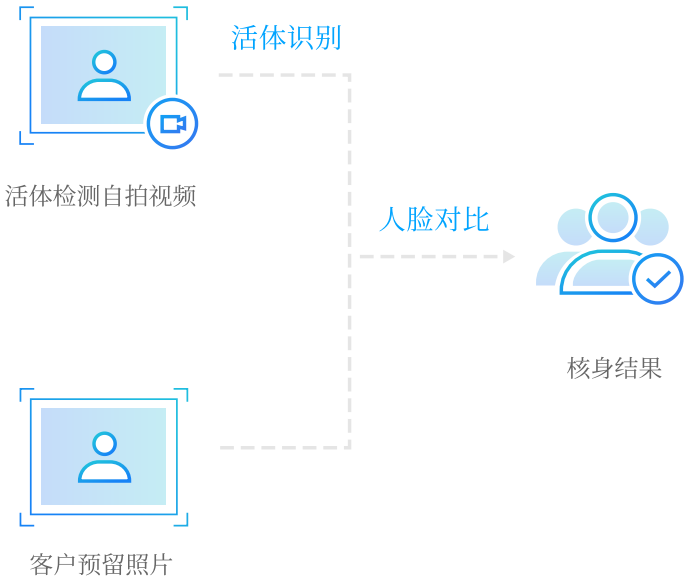
功能介绍

人脸核身为您提供了OCR、活体检测、1:1人脸比对的配套服务，可根据您的需求灵活组合。

1.与权威数据库比对



2.与上传照片比对



身份证 OCR识别

支持识别身份证正反面，一次扫描即可返回身份证号、姓名、有效时间等所有字段，数字识别准确率达到 99.9% 以上，在多个字段上处于领先水平。处理倾斜、暗光、曝光、阴影等异常情况稳定性好，自适应判别纠正技术大大提高识别准确率。

活体检测

活体检测， 通过对一段实时录制的自拍视频进行检测，从而确认当前用户为真人，主要针对高安全性要求的人脸核身验证而研发的一种技术， 防止照片、视频、静态3D建模等各种不同类型的攻击； 目前支持唇语读数、动作、静默等多种活体模式，适用于多类验证场景。

人脸比对

人脸比对，根据面部特征，计算两张人脸的相似度，自动进行身份鉴别。立足于腾讯社交数据大平台收集的海量人脸训练集，结合高维 LBP、PCA、LDA 联合贝叶斯、度量学习、迁移学习、深度神经网络，2017年LFW 测评准确度为 99.80%。并支持多场景下的识别验证，对年龄、姿态及光强均有较好的支持度。

API Category

最近更新时间：2024-05-16 19:40:39

Other APIs

API Name	Feature	Frequency Limit (maximum requests per second)
CreateUploadUrl	Generates upload URLs	20
ApplyLivenessToken	Applies for tokens for liveness detection	20
GetLivenessResult	Gets results of liveness detection	20
GenerateReflectSequence	Obtains light sequences	5
DetectReflectLivenessAndCompare	Compares liveness based on reflection	5

Liveness Detection and Face Comparison (Pure API) APIs

API Name	Feature	Frequency Limit (maximum requests per second)
CompareFaceLiveness	CompareFaceLiveness	20
VideoLivenessCompare	Detects liveness and compares face based on video	20

PaaS APIs of FaceID

API Name	Feature	Frequency Limit (maximum requests per second)
LivenessCompare	Detects liveness and compares faces	100

ApplyCardVerification	ApplyCardVerification	-
GetCardVerificationResult	GetCardVerificationResult	-

Liveness Detection and Face Comparison (App SDK) APIs

API Name	Feature	Frequency Limit (maximum requests per second)
GetFaceIdTokenIntl	Gets the SDK token for liveness detection and face comparison	20
GetFaceIdResultIntl	Gets the verification result of liveness detection and face comparison	20

eKYC and Liveness Detection and Face Comparison (Mobile HTML5) APIs

API Name	Feature	Frequency Limit (maximum requests per second)
ApplyWebVerificationToken	Applies for tokens for web-based verification	20
GetWebVerificationResult	Gets results of web-based verification	20
ApplyWebVerificationBizTokenIntl	Applies for a BizToken for the web-based verification service	20
GetWebVerificationResultIntl	Gets the result of a u200dweb-based verification process	20

eKYC (App SDK) APIs

API Name	Feature	Frequency Limit (maximum requests per second)
ApplySdkVerificationToken	Applies for tokens for SDK-based verification	20

GetSdkVerificationResult	Get results of SDK-based verification	20

Making API Requests
Request Structure

最近更新时间：2024-04-03 11:45:22

1. Service Address

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

Southeast Asia (Bangkok)	faceid.ap-bangkok.tencentcloudapi.com
South Asia (Mumbai)	faceid.ap-mumbai.tencentcloudapi.com
Northeast Asia (Seoul)	faceid.ap-seoul.tencentcloudapi.com
Northeast Asia (Tokyo)	faceid.ap-tokyo.tencentcloudapi.com
U.S. East Coast (Virginia)	faceid.na-ashburn.tencentcloudapi.com
U.S. West Coast (Silicon Valley)	faceid.na-siliconvalley.tencentcloudapi.com
North America (Toronto)	faceid.na-toronto.tencentcloudapi.com
Europe (Frankfurt)	faceid.eu-frankfurt.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

最近更新时间：2024-05-16 19:40:39

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

The exact contents of the common parameters will vary depending on the version of the signature method you use.

Common parameters for Signature Algorithm v3

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

Parameter Name	Type	Required	Description
X-TC-Action	String	Yes	The name of the API for the desired operation. For the specific value, see description of common parameter `Action` in the input parameters in r documentation. For example, the API for querying the CVM instance list i `DescribeInstances` .
X-TC-Region	String	Yes	Region parameter, which is used to identify the region to which the data y work with belongs. For values supported for an API, see the description c parameter `Region` in the input parameters in related API documentati parameter is not required for some APIs (which will be indicated in relate documentation), and will not take effect even it is passed.
X-TC-Timestamp	Integer	Yes	The current UNIX timestamp that records the time when the API request for example, 1529223702. Note: If the difference between the UNIX times server time is greater than 5 minutes, a signature expiration error may oc
X-TC-Version	String	Yes	API version of the action. For the valid values, see the description of the c parameter `Version` in the API documentation. For example, the versi 2017-03-12.
Authorization	String	Yes	The HTTP authentication request header, for example: TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_requ SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc96317 Here: - TC3-HMAC-SHA256: Signature method, currently fixed as this value; - Credential: Signature credential; AKIDEXAMPLE is the SecretId; Date is UTC time, and this value must match the value of X-TC-Timestamp (a co

			parameter) in UTC time format; service is the name of the product/service generally a domain name prefix. For example, a domain name cvm.tencen refers to the CVM product and the value would be cvm; - SignedHeaders: The headers that contains the authentication informatic 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 tempor can obtain the temporary key and token by calling a CAM API. No token i 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/20
18-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993
f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
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=582c400e06b5924a6f2b5d7d672d79c15b1316
2d9279b0855cfba6789a8edb4c
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=582c400e06b5924a6f2b5d7d672d79c15b1316
2d9279b0855cfba6789a8edb4c
Content-Type: multipart/form-data; boundary=58731222010402
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou

--58731222010402
Content-Disposition: form-data; name="Offset"

0
--58731222010402
Content-Disposition: form-data; name="Limit"

10
--58731222010402--
```

Common parameters for Signature Algorithm v1

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

Parameter Name	Type	Required	Description
Action	String	Yes	The name of the API for the desired operation. For the specific value, see the description of common parameter `Action` in the input parameters in related API documentation. For example, the API for querying the CVM instance list is `DescribeInstances`.
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 `Region` 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 `Timestamp` 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 `Version` in the API documentation. For example, the version of CVM is 2017-03-12.
SignatureMethod	String	No	Signature method. Currently, only HmacSHA256 and HmacSHA1 are supported. The HmacSHA256 algorithm is used to verify the signature only when this parameter is specified as HmacSHA256. In other cases, the signature is verified with HmacSHA1.
Token	String	No	The token used for a temporary certificate. It must be used with a temporary key. You can obtain the temporary key and token by calling a CAM API. No token is required for a long-term key.

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

Example of an HTTP GET request structure:

```
https://cvm.tencentcloudapi.com/?Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=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=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d18
1d02&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
Southeast Asia (Bangkok)	ap-bangkok
Hong Kong/Macao/Taiwan (China) (Hong Kong (China))	ap-hongkong
Southeast Asia (Jakarta)	ap-jakarta
Southeast Asia (Singapore)	ap-singapore

Signature v3

最近更新时间：2024-04-03 11:45:25

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, Signature=c492e8e4
1437e97a620b728c301bb8d17e7dc0c17eeabce80c20cd70fc3a78ff" \
-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 `POST` .
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 st “” for POST requests, and is the string after the question mark (?) for GET requests. Fo example: Limit=10&Offset=0。 Note: `CanonicalQueryString` must be URL-encoded, referencing RFC3986, the UTF8 character set. We recommend using the programming language library. All speci characters must be encoded and capitalized.
CanonicalHeaders	Header information for signature calculation, including at least two headers of `host` `content-type` . Custom headers can be added to participate in the signature proce 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 lead and trailing spaces removed, so they are concatenated in the format of key:value\n 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 `content-type:application/json; charset=utf-8\nhost:cvm.tencentcloudapi.com\n` . Note: `content-type` 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 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 head in CanonicalHeaders). `Content-type` and `host` 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 `content-type;host`
HashedRequestPayload	Hash value of the request payload (i.e., the body, such as `{"Limit": 1, "Filter`

	`[{"Values": ["unnamed"], "Name": "instance-name"}]` in this exampl The pseudocode for calculation is Lowercase(HexEncode(Hash.SHA256(RequestPayload))) by SHA256 hashing the pay of the HTTP request, performing hexadecimal encoding, and finally converting the enco string to lowercase letters. For GET requests, `RequestPayload` is always an empt string. The calculation result in this example is `99d58dfbc6745f6747f36bfca17dee5e6881dc0428a0a36f96199342bc5b49`

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 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 date, requested service and termination string (tc3_request). **`Date` is a date in UT time, whose value should match the UTC date converted by the common parameter `X-TC-Timestamp`**; `service` is the product name, which should m the domain name of the product called. The calculation result in this example is `2019-25/cvm/tc3_request`.

HashedCanonicalRequest	Hash value of the CanonicalRequest string concatenated in the steps above. The pseudocode for calculation is Lowercase(HexEncode(Hash.SHA256(CanonicalRequ The calculation result in this example is `2815843035062fffda5fd6f2a44ea8a34818b0dc46f024b8b3786976a3ad`

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
2815843035062fffda5fd6f2a44ea8a34818b0dc46f024b8b3786976a3adda7a
```

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 `Credential`, such as `cvm` 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 `TC3-HMAC-SHA256`.
SecretId	The SecretId in the key pair, i.e., `AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******`.
CredentialScope	Credential scope (see above). The calculation result in this example is `2019-02-25/cvm/tc3_request`.
SignedHeaders	Header information for signature calculation (see above), such as `content-type;host` in this example.
Signature	Signature value. The calculation result in this example is `c492e8e41437e97a620b728c301bb8d17e7dc0c17eeabce80c20cd70fc3a78ff`.

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&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12.

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

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

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

Java

```
import java.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\n" + "hos
t:" + 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" + canoni
calQueryString + "\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, string
ToSign)).toLowerCase();
System.out.println(signature);

// ************* Step 4: Concatenate the Authorization *************
String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + creden
tialScope + ", "
+ "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")).hexd
igest()
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.sha2
56).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\n" + "host:" +
host + "\n"
signedHeaders := "content-type;host"
payload := `{"Limit": 1, "Filters": [{"Values": ["unnamed"], "Name": "instance-na
me"}]}`
hashedRequestPayload := sha256hex(payload)
canonicalRequest := fmt.Sprintf("%s\n%s\n%s\n%s\n%s\n%s",
httpRequestMethod,
canonicalURI,
```

```
canonicalQueryString,
canonicalHeaders,
signedHeaders,
hashedRequestPayload)
fmt.Println(canonicalRequest)

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

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

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

curl := fmt.Sprintf(`curl -X POST https://%s\
-H "Authorization: %s"\
-H "Content-Type: application/json; charset=utf-8"\
-H "Host: %s" -H "X-TC-Action: %s"\
-H "X-TC-Timestamp: %d"\
-H "X-TC-Version: %s"\
-H "X-TC-Region: %s"\
-d '%s'`, host, authorization, host, action, timestamp, version, region, payload)
fmt.Println(curl)
}
```

PHP

```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 = "";
$canonicalHeaders = "content-type:application/json; charset=utf-8\n"."host:".$hos
t."\n";
$signedHeaders = "content-type;host";
$payload = '{"Limit": 1, "Filters": [{"Values": ["unnamed"], "Name": "instance-na
me"}]}';
$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;
```

```php
// 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

```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-nam
e"}]}'
hashed_request_payload = Digest::SHA256.hexdigest(payload)
canonical_request = [
http_request_method,
canonical_uri,
canonical_querystring,
canonical_headers,
signed_headers,
hashed_request_payload,
].join("\n")

puts canonical_request

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

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

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

puts 'curl -X POST ' + endpoint \
```

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

DotNet

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

public class Application
{
public static string SHA256Hex(string s)
{
using (SHA256 algo = SHA256.Create())
{
byte[] hashbytes = algo.ComputeHash(Encoding.UTF8.GetBytes(s));
StringBuilder builder = new StringBuilder();
for (int i = 0; i < hashbytes.Length; ++i)
{
builder.Append(hashbytes[i].ToString("x2"));
}
return builder.ToString();
}
}
public static byte[] HmacSHA256(byte[] key, byte[] msg)
{
using (HMACSHA256 mac = new HMACSHA256(key))
{
return mac.ComputeHash(msg);
}
}

public static Dictionary<String, String> BuildHeaders(string secretid,
string secretkey, string service, string endpoint, string region,
string action, string version, DateTime date, string requestPayload)
{
string datestr = date.ToString("yyyy-MM-dd");
DateTime startTime = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc);
long requestTimestamp = (long)Math.Round((date - startTime).TotalMilliseconds, Mi
```

```
dpointRounding.AwayFromZero) / 1000;
// ************* Step 1: Concatenate the CanonicalRequest string *************
string algorithm = "TC3-HMAC-SHA256";
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string contentType = "application/json";
string canonicalHeaders = "content-type:" + contentType + "; charset=utf-8\n" +
"host:" + endpoint + "\n";
string signedHeaders = "content-type;host";
string hashedRequestPayload = SHA256Hex(requestPayload);
string canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload;
Console.WriteLine(canonicalRequest);
Console.WriteLine("-----------------------------------");

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

// ************* Step 3: Calculate the signature *************
byte[] tc3SecretKey = Encoding.UTF8.GetBytes("TC3" + secretkey);
byte[] secretDate = HmacSHA256(tc3SecretKey, Encoding.UTF8.GetBytes(datestr));
byte[] secretService = HmacSHA256(secretDate, Encoding.UTF8.GetBytes(service));
byte[] secretSigning = HmacSHA256(secretService, Encoding.UTF8.GetBytes("tc3_requ
est"));
byte[] signatureBytes = HmacSHA256(secretSigning, Encoding.UTF8.GetBytes(stringTo
Sign));
string signature = BitConverter.ToString(signatureBytes).Replace("-", "").ToLower
();
Console.WriteLine(signature);
Console.WriteLine("-----------------------------------");

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

```
Console.WriteLine("-----------------------------------");

Dictionary<string, string> headers = new Dictionary<string, string>();
headers.Add("Authorization", authorization);
headers.Add("Host", endpoint);
headers.Add("Content-Type", contentType + "; charset=utf-8");
headers.Add("X-TC-Timestamp", requestTimestamp.ToString());
headers.Add("X-TC-Version", version);
headers.Add("X-TC-Action", action);
headers.Add("X-TC-Region", region);
return headers;
}
public static void Main(string[] args)
{
// SecretID and SecretKey
string SECRET_ID = "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 proj
ect, 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\\u5
47d\\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 = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"
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\n" + "hos
t:" + 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 OPENSSL_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 = 1551113065;
string date = get_data(timestamp);

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

// ************* Step 3: Calculate the signature ***************
string kKey = "TC3" + SECRET_KEY;
string kDate = HmacSha256(kKey, date);
string kService = HmacSha256(kDate, service);
string kSigning = HmacSha256(kService, "tc3_request");
```

```
string signature = HexEncode(HmacSha256(kSigning, stringToSign));
cout << signature << endl;
cout << "-----------------------" << endl;

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

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

Signature Failure

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

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

Signature

最近更新时间：2024-04-03 11:45:25

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
InstanceIds.0	ID of the instance to query	ins-09dx96dg
Offset	Offset	0
Limit	Allowed maximum output	20
Version	API version number	2017-03-12

2.1. Sorting Parameters

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

```
{
'Action' : 'DescribeInstances',
'InstanceIds.0' : 'ins-09dx96dg',
'Limit' : 20,
'Nonce' : 11886,
'Offset' : 0,
'Region' : 'ap-guangzhou',
'SecretId' : '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=1465
185768&Version=2017-03-12
```

2.3. Concatenating the Signature Original String

This step generates a signature original string.

The signature original string consists of the following parameters:

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

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

The concatenation result of the example is:

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&L
imit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WF
kmLPx3*******&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:

```
zmmjn35mikh6pM3V7sUEuX4wyYM=
```

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 zmmjn35mikh6pM3V7sUEuX4wyYM=, the final signature string request parameter (Signature) is zmmjn35mikh6pM3V7sUEuX4wyYM%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.DatatypeConverter;

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

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

public static String getStringToSign(TreeMap<String, Object> params) {
StringBuilder s2s = new StringBuilder("GETcvm.tencentcloudapi.com/?");
// When signing, the parameters need to be sorted in lexicographical order. TreeM
ap 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 UnsupportedEnc
odingException {
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 lette
rs, only the value is URL encoded here.
url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHAR
SET)).append("&");
}
return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap enable
s automatic sorting
// A random number should be used when actually calling, for example: params.put
("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
params.put("Nonce", 11886); // Common parameter
// The current time of the system should be used when actually calling, for examp
le: params.put("Timestamp", System.currentTimeMillis() / 1000);
params.put("Timestamp", 1465185768); // Common parameter
params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"); // Common paramet
er
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*******";
$param["Nonce"] = 11886;//rand();
```

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

ksort($param);

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

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

Ruby

```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, sig
n))
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> requestPa
rams, 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>(p
aram, StringComparer.Ordinal);
string sigInParam = MakeSignPlainText(headers, "GET", endpoint, "/");
Console.WriteLine(sigInParam);
string sigOutParam = Sign(SECRET_KEY, sigInParam);
```

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

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

NodeJS

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

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

function formatSignString(reqMethod, endpoint, path, strParam){
let strSign = reqMethod + endpoint + path + "?" + strParam.slice(1);
return strSign;
}
function sha1(secretKey, strsign){
let signMethodMap = {'HmacSHA1': "sha1"};
let hmac = crypto.createHmac(signMethodMap['HmacSHA1'], secretKey || "");
return hmac.update(Buffer.from(strsign, 'utf8')).digest('base64')
}

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

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

function main(){
const SECRET_ID = "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

最近更新时间：2024-04-03 11:45:26

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 s
ignature 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.
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.

Liveness Detection and Face Comparison (Pure API) APIs CompareFaceLiveness

最近更新时间：2024-04-03 11:45:38

1. API Description

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

This interface supports judgment of real person and photo comparison to verify the user's identity online. By passing the video and photo into the interface, it will first judge whether the person in the video is real. If yes, it judges whether the person in the video is the same one as the uploaded photo and returns authentication result.

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

We recommend you to use API Explorer

Try it

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: CompareFaceLiveness.
Version	Yes	String	Common Params. The value used for this API: 2018-03-01.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product. This API only supports: ap-bangkok, ap-hongkong, ap-singapore.
ImageBase64	Yes	String	Base64 value of photos used for face comparison. The size of image data encoded by Base64 shall not exceed 3M, only

			jpg and png are supported. Please use standard Base64 encoding (use = for padding). Refer to RFC4648 for encoding specifications. Example values: "/9j/4AAQSk... (total length:61944)KiiK//2Q=="
VideoBase64	Yes	String	Base64 value of photos used for face comparison. The size of image data encoded by Base64 shall not exceed 3M, only jpg and png are supported. Please use standard Base64 encoding (use = for padding). Refer to RFC4648 for encoding specifications. Example values: "/9j/4AAQSk... (total length:61944)KiiK//2Q=="
LivenessType	Yes	String	The liveness detection type. Valid values: `LIP` , `ACTION` , and `SILENT` . `LIP` : Numeric mode; `ACTION` : Motion mode; `SILENT` : silent mode. Select one of them. Example value: "SILENT"
ValidateData	No	String	When the “LivenessType” parameter is “ACTION”, it must be specified. It is used to control the action sequence. Action types: 1 (open mouth) 2 (blink) 3 (nod) 4 (shake head). Select one or two from the four actions. Example of passing single action parameter: "1". Example of passing multiple action parameters: "4,2". When the “LivenessType” parameter value is “SILENT”, it shall be unspecified. Example value: ""

3. Output Parameters

Parameter Name	Type	Description
Result	String	Service error code. When the return value is "Success", it indicates that the liveness detection and face comparison succeeded. It is determined that they are the same person. When the return value is "FailedOperation.CompareLowSimilarity", it indicates that the liveness detection succeeded, and the face comparison similarity is lower than 70 points. It is determined that they are not the same person. For other error cases, please refer to Liveness Face Comparison (Pure API) Error Code

		(https://intl.cloud.tencent.com/document/product/1061/55390). Example Value: "Success".
Description	String	Description of business results. Example value: "Success"
Sim	Float	This value is valid when the “Result” parameter is "Success" or "FailedOperation.CompareLowSimilarity." This value indicates the similarity of face comparison. Value range: [0.00, 100.00]. The false pass rate for threshold 70 is 1 in 1,000, and the false pass rate for threshold 80 is 1 in 1,000. Example value: 80.00
BestFrameBase64	String	The optimal screenshot of the video after verification is the value encoded by BASE64, jpg format. Note: This field may return “null”, indicating that no valid value can be obtained. Example values: "/9j/4AAQSk... (total length:142036)s97n//2Q=="
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Liveness detection failed, the specified action was not detected.

Liveness detection failed, the specified action was not detected.

Input Example

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

{
"ImageBase64": "iVBORw0KGg...(total length:121036)s97n//2Q==",
"VideoBase64": "AAAAGGZ0eX...(total length:1651021)AAwAAAAEecg=",
"LivenessType": "ACTION",
"ValidateData": "4,2"
}
```

Output Example

```
{
"Response": {
"Result": "FailedOperation.ActionFirstAction",
"Description": "The first motion is not detected.",
"Sim": 0,
"BestFrameBase64": "/9j/4AAQSk...(total length:161021)W/M7/M/9k=",
"RequestId": "df5afd82-6469-4a4a-bd62-debf8c2ef94f"
}
}
```

Example2 Liveness detection and face comparison passed. Determined to be the same person.

Liveness detection and face comparison passed. Determined to be the same person.

Input Example

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

{
"LivenessType": "SILENT",
"ImageBase64": "iVBORw0KGg...(total length:121036)s97n//2Q==",
"VideoBase64": "AAAAGGZ0eX...(total length:1651021)AAwAAAAEecg=",
"ValidateData": ""
}
```

Output Example

```
{
"Response": {
"Result": "Success",
"Description": "Success",
"Sim": 100,
"BestFrameBase64": "/9j/4AAQSk...(total length:142036)s97n//2Q==",
"RequestId": "f89097ac-4003-4d73-acb3-696d4057b9eb"
}
}
```

Example3 Liveness detection and face comparison failed. Determined not to be the same person.

Liveness detection and face comparison failed. Determined not to be the same person.

Input Example

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

{
"ImageBase64": "iVBORw0KGg...(total length:121036)s97n//2Q==",
"VideoBase64": "AAAAGGZ0eX...(total length:1651021)AAwAAAAEecg=",
"LivenessType": "ACTION",
"ValidateData": "1"
}
```

Output Example

```
{
"Response": {
"Result": "FailedOperation.CompareLowSimilarity",
"Description": "The comparison similarity did not reach the passing standard.",
"Sim": 9.21,
"BestFrameBase64": "/9j/4AAQSk...(total length:138021)8ASrH/2Q==",
"RequestId": "6176fad1-f078-445b-8a4d-c8a903528b5a"
}
}
```

5. Developer Resources

SDK

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

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET

- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

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

Error Code	Description
FailedOperation.UnKnown	Unknown internal error.
InvalidParameter	Invalid parameter.
InvalidParameterValue	Incorrect parameter value.
UnauthorizedOperation.Activating	Activating the service.
UnauthorizedOperation.Arrears	The account is in arrears.
UnauthorizedOperation.NonAuthorize	Identity verification has not been completed for the account.
UnauthorizedOperation.Nonactivated	The service has not been activated.

Liveness Detection and Face Comparison (App SDK) APIs GetFaceIdTokenIntl

最近更新时间：2024-04-03 11:45:38

1. API Description

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

This API is used to apply for an SDK token before calling the liveness detection and face comparison SDK each time. The SDK token is used throughout the identity verification process and to get the verification result after the verification is completed. A token is valid for one identity verification process only.

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

We recommend you to use API Explorer

Try it

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: GetFaceIdTokenIntl.
Version	Yes	String	Common Params. The value used for this API: 2018-03-01.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product. This API only supports: ap-hongkong, ap-jakarta, ap-singapore.
CheckMode	No	String	The detection mode. Valid values: `liveness` : Liveness detection only.

			`compare` : Liveness detection and face comparison. Default value: `liveness` .
SecureLevel	No	String	The verification security level. Valid values: `1` : Video-based liveness detection. `2` : Motion-based liveness detection. `3` : Reflection-based liveness detection. `4` : Motion- and reflection-based liveness detection. Default value: `4` .
Image	No	String	The photo (in Base64) to compare. This parameter is required when the value of `CheckMode` is `compare` .
Extra	No	String	The pass-through parameter, which can be omitted if there are no special requirements.

3. Output Parameters

Parameter Name	Type	Description
SdkToken	String	The SDK token, which is used throughout the verification process and to get the verification result.
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Applying for a token

This example shows you how to apply for an SDK-based verification token.

Input Example

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

```
{
"CheckMode": "liveness",
"SecureLevel": "4",
"Extra": "idxxxx"
}
```

Output Example

```
{
"Response": {
"RequestId": "c27f8a53-1766-4d62-84fc-c400843e9e21",
"SdkToken": "91BF5AD0-C5C9-41CC-9562-DB35BBA2712D"
}
}
```

5. Developer Resources

SDK

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

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

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

Error Code	Description

FailedOperation.ImageSizeTooLarge	The image is too large.
InternalError	Internal error.
InvalidParameter	Invalid parameter.

GetFaceIdResultIntl

最近更新时间：2024-04-03 11:45:39

1. API Description

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

This API is used to get the verification result with the corresponding SDK token after the identity verification process is completed. The SDK token is valid for two hours (2*3,600s) after generation and can be called multiple times.

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

We recommend you to use API Explorer

Try it

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: GetFaceIdResultIntl.
Version	Yes	String	Common Params. The value used for this API: 2018-03-01.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product. This API only supports: ap-hongkong, ap-jakarta, ap-singapore.
SdkToken	Yes	String	The ID of the SDK-based liveness detection and face comparison process, which is generated when the `GetFaceIdTokenIntl` API is called.

3. Output Parameters

Parameter Name	Type	Description
Result	String	The return code of the verification result. 0: Succeeded. 1001: System error. 1004: Liveness detection and face comparison failed. 2004: The image passed in is too large or too small. 2012: Several faces were detected. 2013: No face was detected, or the face detected was incomplete. 2014: The image resolution is too low or the quality does not meet the requirements. 2015: Face comparison failed. 2016: The similarity did not reach the standard passing threshold. -999: The verification process wasn't finished.
Description	String	The description of the verification result.
BestFrame	String	The best frame screenshot (in Base64) obtained during the verification.
Video	String	The video file (Base64) for verification.
Similarity	Float	The similarity, with a value range of 0-100. A greater value indicates higher similarity. This parameter is returned only in the `compare` (liveness detection and face comparison) mode. Note: This field may return `null`, indicating that no valid values can be obtained.
Extra	String	The pass-through parameter. Note: This field may return null, indicating that no valid values can be obtained.
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Getting the verification result

This example shows you how to get the verification result.

Input Example

```
POST / HTTP/1.1
Host: faceid.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: GetFaceIdResultIntl
```

```
<Common request parameters>

{
"SdkToken": "aeee6d62-b2b2-4634-9662-919a5ac729ab"
}
```

Output Example

```
{
"Response": {
"Result": "0",
"Description": "Success",
"BestFrame": "AAAAHGZ0eXBtcDQyAAAAAWlzb2...",
"Video": "/9j/4AAQSkZJRgABAQAASABIAAD/4QBMR...",
"Similarity": 98.8,
"Extra": "abc",
"RequestId": "aea12d62-b2b2-4634-9662-919a5ac729ab"
}
}
```

5. Developer Resources

SDK

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

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

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

Error Code	Description
InternalError	Internal error.
InvalidParameter	Invalid parameter.
InvalidParameterValue.BizTokenExpired	BizToken expired.
InvalidParameterValue.BizTokenIllegal	Invalid BizToken.
OperationDenied	Operation denied.

eKYC and Liveness Detection and Face Comparison (Mobile HTML5) APIs GetWebVerificationResultIntl

最近更新时间：2024-04-03 11:45:27

1. API Description

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

This API is used to get the verification result with the corresponding BizToken after the u200dweb-based verification is completed. The token is valid for three days (259,200s) after issuance and can be called multiple times.

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

We recommend you to use API Explorer

Try it

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: GetWebVerificationResultIntl.
Version	Yes	String	Common Params. The value used for this API: 2018-03-01.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product. This API only supports: ap-jakarta, ap-singapore.
BizToken	Yes	String	The token for the web-based verification, which is generated using the `ApplyWebVerificationBizTokenIntl` API.

3. Output Parameters

Parameter Name	Type	Description
ErrorCode	Integer	The final result of this verification. `0` indicates that the person is the same as that in the photo. For other error codes, see Liveness Detection and Face Comparison (Mobile HTML5) Error Codes Note: u200dThis field may return null, indicating that no valid values can be obtained.
ErrorMsg	String	The description of the final verification result. Note: u200dThis field may return null, indicating that no valid values can be obtained.
VerificationDetailList	Array of VerificationDetail	The detailed verification result list of this process. Retries are allowed, so a verification process may have several entries of results. Note: u200dThis field may return null, indicating that no valid values can be obtained.
VideoBase64	String	The Base64-encoded string of the video collected from the video stream. Retries are allowed, and this field returns only the data collected in the last verification. If no video is collected, null is returned. Note: u200dThis field may return null, indicating that no valid values can be obtained.
BestFrameBase64	String	The Base64-encoded string of the best face screenshot u200dcollected from the video stream. Retries are allowed, and this field returns only the data collected in the last verification. If no best face screenshot is collected, null is returned. Note: u200dThis field may return null, indicating that no valid values can be obtained.
OCRResult	Array of OCRResult	Card recognize result. Note: This field may return null, indicating that no valid values can be obtained.
Extra	String	The passthrough parameter of the business, max 1,000 characters, which will be returned in GetWebVerificationResultIntl.
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Getting the result of a web-based verification process

Input Example

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

{
"BizToken": "EE13636D-1985-42CA-BD61-73F4C8B687E6"
}
```

Output Example

```
{
"Response": {
"ErrorCode": 0,
"ErrorMsg": "abc",
"VerificationDetailList": [
{
"ErrorCode": 0,
"ErrorMsg": "abc",
"LivenessErrorCode": 0,
"LivenessErrorMsg": "abc",
"CompareErrorCode": 0,
"CompareErrorMsg": "abc",
"ReqTimestamp": 1,
"Similarity": 0,
"Seq": "abc"
}
],
"VideoBase64": "abc",
"BestFrameBase64": "abc",
"OCRResult": [
{
"IsPass": true,
"CardImageBase64": "abc",
"CardInfo": {
"HKIDCard": {
```

```
"CnName": "abc",
"EnName": "abc",
"IdNum": "abc",
"Birthday": "abc",
"Sex": "abc"
},
"MLIDCard": {
"Name": "abc",
"ID": "abc",
"Sex": "abc",
"Address": "abc",
"Type": "abc",
"Birthday": "abc"
},
"PhilippinesVoteID": {
"VIN": "abc",
"FirstName": "abc",
"LastName": "abc",
"Birthday": "abc",
"CivilStatus": "abc",
"Citizenship": "abc",
"Address": "abc",
"PrecinctNo": "abc"
},
"IndonesiaIDCard": {
"NIK": "abc",
"Nama": "abc",
"TempatTglLahir": "abc",
"JenisKelamin": "abc",
"GolDarah": "abc",
"Alamat": "abc",
"RTRW": "abc",
"KelDesa": "abc",
"Kecamatan": "abc",
"Agama": "abc",
"StatusPerkawinan": "abc",
"Perkerjaan": "abc",
"KewargaNegaraan": "abc",
"BerlakuHingga": "abc",
"IssuedDate": "abc",
"Provinsi": "abc",
"Kota": "abc"
},
"PhilippinesDrivingLicense": {
"Name": "abc",
"LastName": "abc",
"FirstName": "abc",
```

```
"MiddleName": "abc",
"Nationality": "abc",
"Sex": "abc",
"Address": "abc",
"LicenseNo": "abc",
"ExpiresDate": "abc",
"AgencyCode": "abc",
"Birthday": "abc"
},
"PhilippinesTinID": {
"LicenseNumber": "abc",
"FullName": "abc",
"Address": "abc",
"Birthday": "abc",
"IssueDate": "abc"
},
"PhilippinesSSSID": {
"LicenseNumber": "abc",
"FullName": "abc",
"Birthday": "abc"
},
"PhilippinesUMID": {
"Surname": "abc",
"MiddleName": "abc",
"GivenName": "abc",
"Sex": "abc",
"Birthday": "abc",
"Address": "abc",
"CRN": "abc"
}
},
"RequestId": "abc"
}
],
"RequestId": "abc"
}
}
```


5. Developer Resources

SDK

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

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

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

Error Code	Description
InternalError	Internal error.
InvalidParameter	Invalid parameter.
InvalidParameterValue	Incorrect parameter value.
InvalidParameterValue.BizTokenExpired	BizToken expired.
InvalidParameterValue.BizTokenIllegal	Invalid BizToken.
OperationDenied	Operation denied.

ApplyWebVerificationBizTokenIntl

最近更新时间：2024-04-03 11:45:29

1. API Description

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

This API is used to apply for a BizToken before calling the web-based verification service each time. This token is required for initiating a verification process and getting the result after the verification is completed.

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

We recommend you to use API Explorer

Try it

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this AP ApplyWebVerificationBizTokenIntl.
Version	Yes	String	Common Params. The value used for this AP 2018-03-01.
Region	Yes	String	Common Params. For more information, plea the list of regions supported by the product. T only supports: ap-jakarta, ap-singapore.
RedirectURL	Yes	String	The web callback URL to redirect to after the verification is completed, including the protoc hostname, and path. After the verification process is completed, th BizToken of this process will be spliced to th callback URL in the format of https://intl.cloud.tencent.com/products/faceid

			token={BizToken} before redirect. Example: https://intl.cloud.tencent.com/products/faceic
CompareImageBase64	No	String	The Base64-encoded string (max 8 MB in si the photo to be compared.The Data URI sch header needs to be removed from the encod string Example: xhBQAAACBjSFJNAAB6****AAAASUVORI
Extra	No	String	The passthrough parameter of the business, 1,000 characters, which will be returned in GetWebVerificationResultIntl.
Config	No	WebVerificationConfigIntl	The parameter control the page configuratio Example: {"AutoSkip": true,"CheckMode": 1,"IdCardType": "HKIDCard"}

3. Output Parameters

Parameter Name	Type	Description
BizToken	String	The token for the web-based verification, which is generated using the ApplyWebVerificationBizTokenIntl API. Example: 81EEF678-28EE-4759-A82E-6CBBBE6BC442
VerificationURL	String	The verification URL to be opened with a browser to start the verification process. Example: https://intl.faceid.qq.com/reflect/?token=81EEF678-28EE-4759-A82E-6CBBBE6BC442
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Apply Web Verification BizToken

Apply Web Verification BizToken

Input Example

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

{
"CompareImageBase64": "xhBQAAACBjSFJNAAB6****AAAASUVORK5CYII=",
"RedirectURL": "https://intl.cloud.tencent.com/products/faceid",
"Extra": "ExtraString"
}
```

Output Example

```
{
"Response": {
"VerificationURL": "https://intl.faceid.qq.com/reflect/?token=81EEF678-28EE-4759-A82E-6CBBBE6BC442",
"BizToken": "81EEF678-28EE-4759-A82E-6CBBBE6BC442",
"RequestId": "b16194cd-8f52-4e66-882a-eb6bf15c016d"
}
}
```

5. Developer Resources

SDK

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

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

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

Error Code	Description
InternalError	Internal error.
InvalidParameter	Invalid parameter.
InvalidParameterValue	Incorrect parameter value.
UnauthorizedOperation.ActivateError	Service activation exception.
UnauthorizedOperation.Activating	Activating the service.
UnauthorizedOperation.Arrears	The account is in arrears.
UnauthorizedOperation.Nonactivated	The service has not been activated.

eKYC (App SDK) APIs
ApplySdkVerificationToken

最近更新时间：2024-05-31 11:01:15

1. API Description

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

This API is used to apply for a token before calling the Identity Verification SDK service each time. This token is required for initiating the verification process and getting the result after the verification is completed.

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

We recommend you to use API Explorer

Try it

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: ApplySdkVerificationToken.
Version	Yes	String	Common Params. The value used for this API: 2018-03-01.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product. This API only supports: ap-hongkong, ap-jakarta, ap-singapore.
CheckMode	No	Integer	The verification mode. Valid values: 1: OCR + liveness detection + face comparison 2: Liveness detection + face comparison 3: Liveness detection Default value: 1

SecurityLevel	No	Integer	The security level of the verification. Valid values: 1: Video-based liveness detection 2: Motion-based liveness detection 3: Reflection-based liveness detection 4: Motion- and reflection-based liveness detection Default value: 4
IdCardType	No	String	The identity document type. Valid values: 1. HK (default): Identity card of Hong Kong (China) 2. ML: Malaysian identity card 3. IndonesiaIDCard: Indonesian identity card 4. PhilippinesVoteID: Philippine voters ID card 5. PhilippinesDrivingLicense: Philippine driver's license 6. PhilippinesTinID: Philippine TIN ID card 7. PhilippinesSSSID: Philippine SSS ID card 8. PhilippinesUMID: Philippine UMID card 9. MLIDPassport: Passport issued in Hong Kong/Macao/Taiwan (China) or other countries/regions 10..MacaoIDCard: Macao ID Card 11.ThailandIDCard: Thailand ID Card 12.MainlandIDCard: Mainland ID Card 13.SingaporeIDCard: Singapore ID Card 14.JapanIDCard: Japan ID Card 15.MLDrivingLicense: Malaysian Driving License 16.IndonesiaDrivingLicense: Indonesia Driving License 17.ThailandDrivingLicense: Thailand Driving License 18.SingaporeDrivingLicense: Singapore Driving License 19.JapanDrivingLicense: Japan Driving License
CompareImage	No	String	The Base64-encoded value of the photo to compare, which is required only when `CheckMode` is set to `2`.
DisableChangeOcrResult	No	Boolean	Whether to forbid the modification of the OCR result by users. Default value: `false` (modification allowed). (Currently, this parameter is not applied.)
DisableCheckOcrWarnings	No	Boolean	Whether to disable the OCR warnings. Default value: `false` (not disable), where OCR warnings are enabled and the OCR result will not be returned if there is a warning.

			This feature applies only to Hong Kong (China) identity cards, Malaysian identity cards, and passports.
Extra	No	String	A passthrough field, which is returned together with the verification result and can contain up to 1,024 bits.

3. Output Parameters

Parameter Name	Type	Description
SdkToken	String	The token used to identify an SDK-based verification process. It is valid for 7,200s and can be used to get the verification result after the process is completed.
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Applying for a token

This example shows you how to apply for a verification token.

Input Example

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

{
"CheckMode": 1,
"SecurityLevel": 4,
"NeedVerifyIdCard": true,
"Extra": "fewfewf",
"IdCardType": "HK"
}
```

Output Example

```
{
"Response": {
"SdkToken": "A561B769-C347-4724-A69A-6C3B3483E107",
"RequestId": "d73c0c05-f7ff-419c-84cb-0756303b1925"
}
}
```


5. Developer Resources

SDK

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

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

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

GetSdkVerificationResult

最近更新时间：2024-05-16 19:40:41

1. API Description

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

This API is used to get the verification result with the corresponding token after the SDK-based verification is completed. The token is valid for three days after issuance and can be called multiple times.

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

We recommend you to use API Explorer

Try it

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: GetSdkVerificationResult.
Version	Yes	String	Common Params. The value used for this API: 2018-03-01.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product. This API only supports: ap-hongkong, ap-jakarta, ap-singapore.
SdkToken	Yes	String	The token used to identify an SDK-based verification process.

3. Output Parameters

Parameter Name	Type	Description

Result	String	The result code of the verification result.
Description	String	The verification result description.
ChargeCount	Integer	The charge count.
CardVerifyResults	Array of CardVerifyResult	The results of multiple OCR processes (in order). The result of the final process is used as the valid result.
CompareResults	Array of CompareResult	The results of multiple liveness detection processes (in order). The result of the final process is used as the valid result.
Extra	String	Data passed through in the process of getting the token.
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Getting the verification result

This example shows you how to get the verification result.

Input Example

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

{
"SdkToken": "D2B55F0C-FB5D-4FB6-8765-3E931EBBFC79"
}
```

Output Example

```
{
"Response": {
"CardVerifyResults": [
{
"IsPass": true,
"CardVideo": {
```

```
"Url": "https://intl-reflect-h5-1257237511.cos.ap-guangzhou.myqcloud.com",
"MD5": "682e24b207acf1825286c1fceef5631c",
"Size": 9430792
},
"CardImage": {
"Url": "https://intl-reflect-h5-1257237511.cos.ap-guangzhou.myqcloud.com",
"MD5": "667c2448b10b09ee9ec14ab2b0d36608",
"Size": 232267
},
"CardInfoOcrJson": {
"Url": "https://intl-reflect-h5-1257237511.cos.ap-guangzhou.myqcloud.com",
"MD5": "0ae50935bb50cd70e6e34f81ff2e3fbd",
"Size": 224
},
"RequestId": "8e510d65-c26e-4de7-991d-e07ef0ad953d"
}
],
"CompareResults": [
{
"ErrorCode": "1001",
"ErrorMsg": "Failed to call the liveness engine",
"LiveData": {
"Url": "https://intl-reflect-h5-1257237511.cos.ap-guangzhou.myqcloud.com",
"MD5": "f624d26bfb45e149b293097037819feb",
"Size": 719585
},
"LiveVideo": {
"Url": "https://intl-reflect-h5-1257237511.cos.ap-guangzhou.myqcloud.com",
"MD5": "e87070d8eb95f64bc01b12e03cc8f533",
"Size": 887224
},
"LiveErrorCode": "1001",
"LiveErrorMsg": "Failed to call the liveness engine",
"CompareErrorCode": "",
"CompareErrorMsg": "",
"BestFrame": {
"Url": "https://intl-reflect-h5-1257237511.cos.ap-guangzhou.myqcloud.com",
"MD5": "c4f217871aaeb0180e40152f61658835",
"Size": 122984
},
"ProfileImage": {
"Url": "https://intl-reflect-h5-1257237511.cos.ap-guangzhou.myqcloud.com",
"MD5": "d51df99b25d87785ea5e2dfb0d6e920f",
"Size": 23091
},
"Sim": 0,
"IsNeedCharge": true,
```

```
"CardInfoInputJson": {
"Url": "https://intl-reflect-h5-1257237511.cos.ap-guangzhou.myqcloud.com",
"MD5": "02497f445dc72a330b468fa24529315b",
"Size": 219
},
"RequestId": "6e498069-4d51-4032-82a8-9adb53cda85b"
}
],
"ChargeCount": 4,
"Description": "Failed to call the liveness engine",
"Extra": "",
"RequestId": "b8cb2269-08b2-426c-8be8-c7142c7e64e4",
"Result": "1001"
}
}
```


5. Developer Resources

SDK

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

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

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

ID Verification (Pure API) APIs
GetCardVerificationResult

最近更新时间：2024-04-03 11:45:32

1. API Description

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

The interface supports obtaining the certificate authentication result based on the token.

We recommend you to use API Explorer

Try it

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: GetCardVerificationResult.
Version	Yes	String	Common Params. The value used for this API: 2018-03-01.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product. This API only supports: ap-hongkong, ap-jakarta, ap-singapore.
CardVerificationToken	Yes	String	The token used to identify an verification process. It can be used to get the verification result after the process is completed.

3. Output Parameters

Parameter Name	Type	Description

Status	String	Pass status. When Warning and Rejected are returned, please check the specific reasons in the WarnInfo structure return. Example values are as follows: Passed Warning Rejected
WarnInfo	Array of String	Warning information returned by document verification.
Nationality	String	Nationality code. Complies with standard ISO 3166-1 alpha-3. Example value: IDN
CardType	String	Card Type. The supported options are: ID_CARD PASSPORT DRIVING_LICENSE AUTO Example value: ID_CARD
CardSubType	String	Subtype of the ID document.
CardInfo	CardInfo	Recognition results of the ID document.
IDVerificationToken	String	The token used to identify an verification process. It can be used to get the verification result after the process is completed.
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Sample response

Input Example

```
POST / HTTP/1.1
Host: faceid.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: GetCardVerificationResult
```

```
<common parameters>

{
"CardVerificationToken": "abc"
}
```

Output Example

```
{
"Response": {
"Status": "abc",
"WarnInfo": [
"abc"
],
"Nationality": "abc",
"CardType": "abc",
"CardSubType": "abc",
"CardInfo": {
"HKIDCard": {
"CnName": "abc",
"EnName": "abc",
"TelexCode": "abc",
"Sex": "abc",
"Birthday": "abc",
"Permanent": "abc",
"IdNum": "abc",
"Symbol": "abc",
"FirstIssueDate": "abc",
"CurrentIssueDate": "abc"
},
"MLIDCard": {
"Name": "abc",
"ID": "abc",
"Sex": "abc",
"Address": "abc",
"Type": "abc",
"Birthday": "abc"
},
"PhilippinesVoteID": {
"VIN": "abc",
"FirstName": "abc",
"LastName": "abc",
"Birthday": "abc",
"CivilStatus": "abc",
"Citizenship": "abc",
"Address": "abc",
```

```
"PrecinctNo": "abc"
},
"IndonesiaIDCard": {
"NIK": "abc",
"Nama": "abc",
"TempatTglLahir": "abc",
"JenisKelamin": "abc",
"GolDarah": "abc",
"Alamat": "abc",
"RTRW": "abc",
"KelDesa": "abc",
"Kecamatan": "abc",
"Agama": "abc",
"StatusPerkawinan": "abc",
"Perkerjaan": "abc",
"KewargaNegaraan": "abc",
"BerlakuHingga": "abc",
"IssuedDate": "abc",
"Provinsi": "abc",
"Kota": "abc"
},
"PhilippinesDrivingLicense": {
"Name": "abc",
"LastName": "abc",
"FirstName": "abc",
"MiddleName": "abc",
"Nationality": "abc",
"Sex": "abc",
"Address": "abc",
"LicenseNo": "abc",
"ExpiresDate": "abc",
"AgencyCode": "abc",
"Birthday": "abc"
},
"PhilippinesTinID": {
"LicenseNumber": "abc",
"FullName": "abc",
"Address": "abc",
"Birthday": "abc",
"IssueDate": "abc"
},
"PhilippinesSSSID": {
"LicenseNumber": "abc",
"FullName": "abc",
"Birthday": "abc"
},
"PhilippinesUMID": {
```

```
"Surname": "abc",
"MiddleName": "abc",
"GivenName": "abc",
"Sex": "abc",
"Birthday": "abc",
"Address": "abc",
"CRN": "abc"
},
"InternationalIDPassport": {
"LicenseNumber": "abc",
"FullName": "abc",
"Surname": "abc",
"GivenName": "abc",
"Birthday": "abc",
"Sex": "abc",
"DateOfExpiration": "abc",
"IssuingCountry": "abc",
"NationalityCode": "abc",
"PassportCodeFirst": "abc",
"PassportCodeSecond": "abc"
},
"GeneralCard": {
"LicenseNumber": "abc",
"PersonalNumber": "abc",
"PassportCodeFirst": "abc",
"PassportCodeSecond": "abc",
"ExpirationDate": "abc",
"DueDate": "abc",
"IssuedDate": "abc",
"IssuedAuthority": "abc",
"IssuedCountry": "abc",
"FullName": "abc",
"FirstName": "abc",
"LastName": "abc",
"Sex": "abc",
"Age": "abc",
"Birthday": "abc",
"BirthPlace": "abc",
"Nationality": "abc",
"RegistrationNumber": "abc",
"Address": {
"Country": "abc",
"PostalCode": "abc",
"Subdivision": "abc",
"City": "abc",
"FormattedAddress": "abc",
"LineOne": "abc",
```

```
"LineTwo": "abc",
"LineThree": "abc",
"LineFour": "abc",
"LineFive": "abc"
}
},
"IndonesiaDrivingLicense": {
"LastName": "abc",
"FirstName": "abc",
"LicenseNumber": "abc",
"Birthday": "abc",
"Address": "abc",
"ExpirationDate": "abc",
"IssuedDate": "abc",
"IssuedCountry": "abc"
},
"ThailandIDCard": {
"LastName": "abc",
"FirstName": "abc",
"LicenseNumber": "abc",
"DateOfBirth": "abc",
"DateOfExpiry": "abc",
"DateOfIssue": "abc",
"IssuedCountry": "abc"
},
"SingaporeIDCard": {
"ChName": "abc",
"EnName": "abc",
"ID": "abc",
"Sex": "abc",
"CountryOfBirth": "abc",
"Birthday": "abc",
"Address": "abc",
"Race": "abc",
"NRICCode": "abc",
"PostCode": "abc",
"DateOfExpiration": "abc",
"DateOfIssue": "abc"
}
},
"IDVerificationToken": "abc",
"RequestId": "abc"
}
}
```

5. Developer Resources

SDK

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

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

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.UnKnowError	Unknown error.
FailedOperation.UnOpenError	The service is not activated.
InvalidParameterValue.InvalidParameterValueLimit	Parameter value is wrong.
ResourceUnavailable.InArrears	The account is in arrears.
ResourceUnavailable.ResourcePackageRunOut	The account has exhausted the resource package for this service.

ApplyCardVerification

最近更新时间：2024-04-03 11:45:33

1. API Description

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

The types of national cards supported by the API and whether instructions on the back of the card are required are as follows:

Nationality	CardType	Back side required
Indonesia	ID card	No
Indonesia	Drving license	No
Hongkong	ID card	Yes
Thailand	ID card	No
Thailand	Drving license	Yes
Malaysia	ID card	Yes
Malaysia	Drving license	Yes
Singapore	ID card	Yes
Singapore	Drving license	Yes
Philippine	ID card	Yes
Philippine	Drving license	No
Japan	ID card	Yes
Japan	Drving license	No

We recommend you to use API Explorer

Try it

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: ApplyCardVerification.
Version	Yes	String	Common Params. The value used for this API: 2018-03-01.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product. This API only supports: ap-hongkong, ap-jakarta, ap-singapore.
Nationality	Yes	String	Please select the country code of ID document. IDN: Indonesia HKG: Hong Kong THA: Thailand MYS: Malaysia SGP: Singapore JPN: Japan AUTO: Automatic Identification
CardType	Yes	String	Please select the type of ID document. The supported options are: ID_CARD PASSPORT DRIVING_LICENSE AUTO
ImageBase64Front	No	String	Base64 value for the front of the document. Supported image formats: PNG, JPEG, GIF format is not supported yet. Supported image size: The downloaded image cannot exceed 5M after Base64 encoding. The image download takes no more than 3 seconds. Supported image resolution: 8000*8000. One of ImageUrlFront and ImageBase64 Front of the image must be provided. If both are provided, only ImageUrlFront will be used.
ImageBase64Back	No	String	Base64 value of the reverse side of the document. Supported image formats: PNG, JPEG, GIF format is not supported yet. Supported image size: The downloaded image cannot exceed 5M after Base64 encoding. The image download takes no more than 3 seconds. Maximum supported image resolution:

			8000*8000. For some certificates, one of ImageUrlBack and ImageBase64Back must be provided. If both are provided, only ImageUrlBack will be used.
ImageUrlFront	No	String	The URL value on the back of the certificate. Supported image formats: PNG, JPEG, GIF format is not supported yet. Supported image size: The downloaded image cannot exceed 5M after Base64 encoding. The image download takes no more than 3 seconds. Maximum supported image resolution: 8000*8000. One of ImageUrlFront and ImageBase64Front of the image must be provided. If both are provided, only ImageUrlFront will be used.
ImageUrlBack	No	String	The URL value on the back of the certificate. Supported image formats: PNG, JPEG, GIF format is not supported yet. Supported image size: The downloaded image cannot exceed 5M after Base64 encoding. The image download takes no more than 3 seconds. Maximum supported image resolution: 8000*8000. For some certificates, one of ImageUrlBack and ImageBase64Back must be provided. If both are provided, only ImageUrlBack will be used.

3. Output Parameters

Parameter Name	Type	Description
CardVerificationToken	String	The token used to identify an verification process. It can be used to get the verification result after the process is completed.
AsyncCardVerificationMaxPollingTimes	Integer	The maximum number of polls for calling the pull result interface polling.
AsyncCardVerificationPollingWaitTime	Integer	The interval for polling when calling the pull result interface (in seconds).
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Sample response

Input Example

```
POST / HTTP/1.1
Host: ocr.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ApplyIDVerification
<common parameters>

{
"ImageBase64Front": "abc",
"ImageBase64Back": "abc",
"ImageUrlFront": "abc",
"ImageUrlBack": "abc",
"Nationality": "abc",
"CardType": "abc"
}
```

Output Example

```
{
"Response": {
"CardVerificationToken": "abc",
"RequestId": "abc"
}
}
```

5. Developer Resources

SDK

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

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

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.ImageSizeTooLarge	The image is too large.
FailedOperation.UnKnowError	Unknown error.
FailedOperation.UnOpenError	The service is not activated.
InvalidParameter.EngineImageDecodeFailed	Image decoding failed.
InvalidParameterValue.InvalidFileContentSize	The image file content size is abnormal.
InvalidParameterValue.InvalidParameterValueLimit	Parameter value is wrong.
ResourceUnavailable.ImageDownloadError	Image file download failed.
ResourceUnavailable.InArrears	The account is in arrears.
ResourceUnavailable.ResourcePackageRunOut	The account has exhausted the resource package for this service.

Other APIs

ApplyLivenessToken

最近更新时间：2024-04-03 11:45:36

1. API Description

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

This API is used to apply for a token before calling the liveness detection service each time. This token is required for initiating the verification process and getting the result after the verification is completed.

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

We recommend you to use API Explorer

Try it

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: ApplyLivenessToken.
Version	Yes	String	Common Params. The value used for this API: 2018-03-01.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product. This API only supports: ap-hongkong, ap-jakarta, ap-singapore.
SecureLevel	No	String	Enumerated value. Valid values: `1`, `2`, `3`, and `4`. Their meanings are as follows: 1 - silent 2 - blinking 3 - light 4 - blinking + light (default)

3. Output Parameters

Parameter Name	Type	Description
SdkToken	String	The token used to identify an SDK-based verification process. It is valid for 10 minutes and can be used to get the verification result after the process is completed.
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Applying for a token

Input Example

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

{
"SecureLevel": "1"
}
```

Output Example

```
{
"Response": {
"SdkToken": "A561B769-C347-4724-A69A-6C3B3483E107",
"RequestId": "d73c0c05-f7ff-419c-84cb-0756303b1925"
}
}
```

5. Developer Resources

SDK

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

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

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

Error Code	Description
InternalError	Internal error.
InternalError.UnKnown	Unknown internal error.
InvalidParameter	Invalid parameter.
InvalidParameterValue	Incorrect parameter value.

GetLivenessResult

最近更新时间：2024-04-03 11:45:33

1. API Description

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

This API is used to get the verification result with the corresponding token (SdkToken) after the liveness detection is completed. The token is valid for two hours after issuance and can be called multiple times.

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

We recommend you to use API Explorer

Try it

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

2. Input Parameters

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

Parameter Name	Required	Type	Description
Action	Yes	String	Common Params. The value used for this API: GetLivenessResult.
Version	Yes	String	Common Params. The value used for this API: 2018-03-01.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product. This API only supports: ap-hongkong, ap-jakarta, ap-singapore.
SdkToken	Yes	String	The token used to identify an SDK-based verification process.

3. Output Parameters

Parameter Name	Type	Description

Result	String	The final verification result.
Description	String	The description of the final verification result.
BestFrame	FileInfo	The face screenshot.
Video	FileInfo	The video for the detection.
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Getting the verification result

Input Example

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

{
"SdkToken": "D2B55F0C-FB5D-4FB6-8765-3E931EBBFC79"
}
```

Output Example

```
{
"Response": {
"Video": {
"Url": "https://intl-reflect-h5-1257237511.cos.ap-guangzhou.myqcloud.com",
"MD5": "682e24b207acf1825286c1fceef5631c",
"Size": 9430792
},
"BestFrame": {
"Url": "https://intl-reflect-h5-1257237511.cos.ap-guangzhou.myqcloud.com",
"MD5": "667c2448b10b09ee9ec14ab2b0d36608",
"Size": 232267
},
"Description": "Failed to call the liveness engine",
"RequestId": "b8cb2269-08b2-426c-8be8-c7142c7e64e4",
```

```
"Result": "1001"
}
}
```

5. Developer Resources

SDK

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

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

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

Error Code	Description
InternalError	Internal error.
InvalidParameter	Invalid parameter.
InvalidParameterValue.BizTokenExpired	BizToken expired.
InvalidParameterValue.BizTokenIllegal	Invalid BizToken.
OperationDenied	Operation denied.

GenerateReflectSequence

最近更新时间：2024-04-03 11:45:34

1. API Description

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

This API is used to generate an appropriate light sequence based on the information collected by the liveness comparison (reflection-based) SDK and pass the light sequence into the SDK to start the identity verification process. The data generated with the SDK must be stored in COS, and the region of the COS bucket must be same as that of requests made with this API. We recommend that you pass resources with upload link APIs.

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 Params. The value used for this API: GenerateReflectSequence.
Version	Yes	String	Common Params. The value used for this API: 2018-03-01.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product. This API only supports: ap-bangkok, ap-hongkong, ap-singapore.
DeviceDataUrl	Yes	String	The resource URL of the data package generated by the SDK.
DeviceDataMd5	Yes	String	The MD5 hash value of the data package generated by the SDK.
SecurityLevel	No	String	1 - silent

			2 - blinking 3 - light 4 - blinking + light (default)

3. Output Parameters

Parameter Name	Type	Description
ReflectSequenceUrl	String	The resource URL of the light sequence, which needs to be downloaded and passed through to the SDK to start the identity verification process.
ReflectSequenceMd5	String	The MD5 hash value of the light sequence, which is used to check whether the light sequence is altered.
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Obtaining a light sequence

Input Example

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

{
"DeviceDataUrl": "https://faceid-resource-sg-1254418846.cos.ap-singapore.myqcloud.com/faceid%2FApplyWebVerificationToken%2F1300268875%2F20b11b59-572d-406d-8d94-e6e05782134c",
"DeviceDataMd5": "d41d8cd98f00b204e9800998ecf8427e"
}
```

Output Example

```
{
"Response": {
```

```
"ReflectSequenceUrl": "https://faceid-resource-sg-1254418846.cos.ap-singapore.myqcloud.com",
"ReflectSequenceMd5": "d41d8cd98f00b204e9800998ecf8427e",
"RequestId": "32-323233-323"
}
}
```


5. Developer Resources

SDK

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

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

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

Error Code	Description
FailedOperation.DownLoadError	File download failed.
FailedOperation.DownLoadTimeoutError	File download timed out.

DetectReflectLivenessAndCompare

最近更新时间：2024-04-03 11:45:35

1. API Description

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

This API is used to detect liveness with the package generated by the liveness comparison (reflection-based) SDK, and to compare the person detected with that in the image passed in.
The image and the data generated with the SDK must be stored in COS, and the region of the COS bucket must be same as that of requests made with this API. We recommend that you pass resources with upload link APIs.

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 Params. The value used for this API: DetectReflectLivenessAndCompare.
Version	Yes	String	Common Params. The value used for this API: 2018-03-01.
Region	Yes	String	Common Params. For more information, please see the list of regions supported by the product. This API only supports: ap-bangkok, ap-hongkong, ap-singapore.
LiveDataUrl	Yes	String	URL of the liveness detection data package generated by the SDK
LiveDataMd5	Yes	String	MD5 hash value (32-bit) of the liveness detection data package generated by the SDK, which is used to verify the LiveData consistency.

ImageUrl	Yes	String	URL of the target image for comparison
ImageMd5	Yes	String	MD5 hash value (32-bit) of the target image for comparison, which is used to verify the `Image` consistency.

3. Output Parameters

Parameter Name	Type	Description
BestFrameUrl	String	Temporary URL of the best screenshot (.jpg) of the video after successful verification. Both the screenshot and the URL are valid for two hours only, so you need to download the screenshot within this period.
BestFrameMd5	String	MD5 hash value (32-bit) of the best screenshot of the video after successful verification, which is used to verify the `BestFrame` consistency.
Result	String	Service error code. `Success` will be returned for success. For error information, see the `FailedOperation` section in the error code list below.
Description	String	Service result description
Sim	Float	Similarity. Value range: [0.00, 100.00]. As a recommendation, when the similarity is greater than or equal to 70, it can be determined that the two faces are of the same person. You can adjust the threshold according to your specific scenario (the FAR at the threshold of 70 is 0.1%, and FAR at the threshold of 80 is 0.01%).
RequestId	String	The unique request ID, generated by the server, will be returned for every request (if the request fails to reach the server for other reasons, the request will not obtain a RequestId). RequestId is required for locating a problem.

4. Example

Example1 Compared successfully

Input Example

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

```
{
"ImageMd5": "d41d8cd98f00b204e9800998ecf8427e",
"LiveDataUrl": "https://faceid-resource-sg-1254418846.cos.ap-singapore.myqcloud.com/faceid%2FApplyWebVerificationToken%2F1300268875%2F20b11b59-572d-406d-8d94-e6e05782134c",
"ImageUrl": "https://faceid-resource-sg-1254418846.cos.ap-singapore.myqcloud.com/faceid%2FApplyWebVerificationToken%2F1300268875%2F20b11b59-572d-406d-8d94-e6e057821 34c",
"LiveDataMd5": "d41d8cd98f00b204e9800998ecf8427e"
}
```


Output Example


```
{
"Response": {
"BestFrameUrl": "https://faceid-resource-sg-1254418846.cos.ap-singapore.myqcloud.com/faceid%2FApplyWebVerificationToken%2F1300268875%2F20b11b59-572d-406d-8d94-e6e05782134c",
"Description": "Success",
"BestFrameMd5": "d41d8cd98f00b204e9800998ecf8427e",
"RequestId": "00577fa0-9d11-459e-a455-fc202ecd65bc",
"Sim": 96.3,
"Result": "Success"
}
}
```


5. Developer Resources

SDK

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

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for Node.js
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

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

Error Code	Description
FailedOperation.ActionLightDark	The lighting is too dim.
FailedOperation.ActionLightStrong	The lighting is too strong.
FailedOperation.DownLoadError	File download failed.
FailedOperation.DownLoadTimeoutError	File download timed out.
FailedOperation.ImageBlur	The image is blurry.
FailedOperation.ImageDecodeFailed	Image decoding failed.
FailedOperation.ImageSizeTooLarge	The image is too large.
FailedOperation.LifePhotoDetectFake	Real person comparison failed.
FailedOperation.LifePhotoSizeError	The image passed in is too large or too small.
FailedOperation.LivessDetectFail	Liveness detection failed.
FailedOperation.LivessDetectFake	Suspected spoofed recording.
FailedOperation.LivessSystemError	Error calling the liveness engine API.
FailedOperation.LivessUnknownError	Video-based real person detection failed.
FailedOperation.SilentDetectFail	Real person detection failed.
FailedOperation.SilentFaceDetectFail	No face is detected in the video.
FailedOperation.SilentFaceQualityFail	Low face quality.
FailedOperation.SilentMultiFaceFail	Multiple faces are detected in the video.
FailedOperation.SilentPictureLiveFail	The video might be spoofed.

FailedOperation.SilentThreshold	Real person detection did not reach the passing standard.
InternalError.ActionLightDark	The lighting is too dim.
InternalError.ActionLightStrong	The lighting is too strong.
InternalError.ActionNodetectFace	Failed to detect a full face.
InternalError.CompareLowSimilarity	The similarity did not reach the passing standard.
InternalError.LifePhotoPoorQuality	The resolution of the image passed in is too low. Please upload a new one.
InternalError.LifePhotoSizeError	The image passed in is too large or too small.

Data Types

最近更新时间：2024-05-20 22:58:05

Address

Used by actions: GetCardVerificationResult, GetSdkVerificationResult, GetWebVerificationResultIntl.

Name	Type	Description
Country	String	Nationality.
PostalCode	String	Post code.
Subdivision	String	Subregion.
City	String	City.
FormattedAddress	String	Complete address.
LineOne	String	The first line of address.
LineTwo	String	The second line of address.
LineThree	String	The third line of address.
LineFour	String	The fourth line of address.
LineFive	String	The fifth line of address.

CardInfo

License OCR result

Used by actions: GetCardVerificationResult, GetWebVerificationResultIntl.

Name	Type	Description
HKIDCard	HKIDCard	Hong Kong ID Card Note: This field may return null, indicating that no valid values can be obtained.
MLIDCard	MLIDCard	Malaysia ID Card Note: This field may return null, indicating that no

		valid values can be obtained.
PhilippinesVoteID	PhilippinesVoteID	Philippines VoteID Card Note: This field may return null, indicating that no valid values can be obtained.
IndonesiaIDCard	IndonesiaIDCard	Indonesia ID Card Note: This field may return null, indicating that no valid values can be obtained.
PhilippinesDrivingLicense	PhilippinesDrivingLicense	Philippines Driving License Note: This field may return null, indicating that no valid values can be obtained.
PhilippinesTinID	PhilippinesTinID	Philippines TinID Note: This field may return null, indicating that no valid values can be obtained.
PhilippinesSSSID	PhilippinesSSSID	Philippines SSSID Note: This field may return null, indicating that no valid values can be obtained.
PhilippinesUMID	PhilippinesUMID	Philippines UMID Note: This field may return null, indicating that no valid values can be obtained.
InternationalIDPassport	InternationalIDPassport	ID Cards of Hong Kong, Macao and Taiwan (China), and International Passport Note: This field may return null, indicating that no valid values can be obtained.
GeneralCard	GeneralCard	General license information Note: This field may return null, indicating that no valid values can be obtained.
IndonesiaDrivingLicense	IndonesiaDrivingLicense	Indonesia Driving License Note: This field may return null, indicating that no valid values can be obtained.
ThailandIDCard	ThailandIDCard	Thailand ID Card Note: This field may return null, indicating that no valid values can be obtained.
SingaporeIDCard	SingaporeIDCard	Singapore ID Card Note: This field may return null, indicating that no valid values can be obtained.
MacaoIDCard	MacaoIDCard	Macao ID Card

		Note: This field may return null, indicating that no valid values can be obtained.

CardVerifyResult

The OCR result of a user's identity document during the eKYC verification process.

Used by actions: GetSdkVerificationResult.

Name	Type	Description
IsPass	Boolean	Whether the authentication or OCR process is successful.
IsEdit	Boolean	Whether the user modified the card recognition result
CardVideo	FileInfo	The download URL of the video used for identity document verification, which is valid for 10 minutes. This parameter is returned only if video-based identity document verification is enabled. Note: This field may return null, indicating that no valid value can be obtained.
CardImage	FileInfo	The download URL of the identity document image, which is valid for 10 minutes. Note: This field may return null, indicating that no valid value can be obtained.
CardInfoOcrJson	FileInfo	The OCR result (in JSON) of the identity document image. If verification or OCR fails, this parameter is left empty. The URL is valid for 10 minutes. (1) Hong Kong (China) identity card When the value of `IdCardType` is `HK`: - CnName (string): Name in Chinese. - EnName (string): Name in English. - TelexCode (string): The code corresponding to the name in Chinese. - Sex (string): Gender. Valid values: `M` (male) and `F` (female). - Birthday (string): Date of birth. - Permanent (int): Whether it is a permanent residence identity card. Valid values: `0` (non-permanent), `1` (permanent), and `-1` (unknown). - IdNum (string): Identity card number. - Symbol (string): The ID symbol below the date of birth, such as "***AZ". - FirstIssueDate (string): Month and year of first registration. - CurrentIssueDate (string): The date of latest issuance.

(2) Malaysian identity card
When the value of `IdCardType` is `ML`:
- Sex (string): Gender. Valid values: `LELAKI` (male) and `PEREMPUAN` (female).
- Birthday (string): Date of birth.
- ID (string): Identity card number.
- Name (string): Name.
- Address (string): Address.
- Type (string): Identity document type.

(3) Philippine identity document
When the value of `IdCardType` is `PhilippinesVoteID`:
- Birthday (string): Date of birth.
- Address (string): Address.
- LastName (string): Last name.
- FirstName (string): First name.
- VIN (string): Voter's identification number (VIN).
- CivilStatus (string): Civil status.
- Citizenship (string): Citizenship.
- PrecinctNo (string): Precinct.

When the value of `IdCardType` is `PhilippinesDrivingLicense`:
- Sex (string): Gender.
- Birthday (string): Date of birth.
- Name (string): Name.
- Address (string): Address.
- LastName (string): Last name.
- FirstName (string): First name.
- MiddleName (string): Middle name.
- Nationality (string): Nationality.
- LicenseNo (string): License number.
- ExpiresDate (string): Expiration date.
- AgencyCode (string): Agency code.

When the value of `IdCardType` is `PhilippinesTinID`:
- LicenseNumber (string): Tax identification number (TIN).
- FullName (string): Full name.
- Address (string): Address.
- Birthday (string): Date of birth.
- IssueDate (string): Issue date.

When the value of `IdCardType` is `PhilippinesSSSID`:
- LicenseNumber (string): Common reference number (CRN).

		- FullName (string): Full name. - Birthday (string): Date of birth. When the value of `IdCardType` is `PhilippinesUMID`: - Surname (string): Surname. - MiddleName (string):Middle name. - GivenName (string): Given name. - Sex (string): Gender. - Birthday (string): Date of birth. - Address (string): Address. - CRN (string): Common reference number (CRN). (4) Indonesian identity card When the value of `IdCardType` is `IndonesiaIDCard`: - NIK (string): Single Identity Number. - Nama (string): Full name. - TempatTglLahir (string): Place and date of birth. - JenisKelamin (string): Gender. - GolDarah (string): Blood type. - Alamat (string): Address. - RTRW (string): Street. - KelDesa (string): Village. - Kecamatan (string): Region. - Agama (string): Religion. - StatusPerkawinan (string): Marital status. - Perkerjaan (string): Occupation. - KewargaNegaraan (string): Nationality. - BerlakuHingga (string): Expiry date. - IssuedDate (string): Issue date. (5) A passport issued in Hong Kong/Macao/Taiwan (China) or other countries/regions When the value of `IdCardType` is `MLIDPassport`: - FullName (string): Full name. - Surname (string): Surname. - GivenName (string): Given name. - Birthday (string): Date of birth. - Sex (string): Gender. Valid values: `F` (female) and `M` (male). - DateOfExpiration (string): Expiration date. - IssuingCountry (string): Issuing country. - NationalityCode (string): Country/region code. Note: This field may return null, indicating that no valid values can be obtained.
RequestId	String	The request ID of a single process.

NormalCardInfo	NormalCardInfo	License OCR result
WarnCardInfos	Array of Integer	Card warning information -9101 Alarm for covered certificate, -9102 Alarm for photocopied certificate, -9103 Alarm for photographed certificate, -9107 Alarm for reflective certificate, -9108 Alarm for blurry image, -9109 This capability is not enabled.

CompareResult

The description of a single comparison result.

Used by actions: GetSdkVerificationResult.

Name	Type	Description
ErrorCode	String	The final verification result code. 0: Success. 1001: Failed to call the liveness detection engine. 1004: Face detection failed. 2004: The uploaded face image is too large or too small. 2012: The face is not fully exposed. 2013: No face is detected. 2014: The resolution of the uploaded image is too low . Please upload a new one. 2015: Face comparison failed. 2016: The similarity did not reach the passing standard.
ErrorMsg	String	The description of the final verification result.
LiveData	FileInfo	The liveness algorithm package generated during this SDK-based liveness detection.
LiveVideo	FileInfo	The download URL of the video used for verification, which is valid for 10 minutes.
LiveErrorCode	String	The liveness detection result code. 0: Success. 1001: Failed to call the liveness detection engine. 1004: Face detection failed.
LiveErrorMsg	String	The description of the liveness detection result.

BestFrame	FileInfo	The download URL of the face screenshot during verification, which is valid for 10 minutes. Note: This field may return null, indicating that no valid value can be obtained.
ProfileImage	FileInfo	The download URL of the profile photo screenshot from the identity document, which is valid for 10 minutes.
CompareErrorCode	String	The face comparison result code. 0: Success. 2004: The uploaded face image is too large or too small. 2012: The face is not fully exposed. 2013: No face is detected. 2014: The resolution of the uploaded image is too low . Please upload a new one. 2015: Face comparison failed. 2016: The similarity did not reach the passing standard. Note: This field may return null, indicating that no valid value can be obtained.
CompareErrorMsg	String	The description of the face comparison result. Note: This field may return null, indicating that no valid values can be obtained.
Sim	Float	The similarity score of face comparison. Note: This field may return null, indicating that no valid values can be obtained.
IsNeedCharge	Boolean	This parameter is disused.
CardInfoInputJson	FileInfo	The identity document photo info edited by the user. Currently, this parameter is not applied. Note: This field may return null, indicating that no valid values can be obtained.
RequestId	String	The request ID of this verification process.

FileInfo

The description of a file, including a download URL and the MD5 checksum and size of the file.

Used by actions: GetLivenessResult, GetSdkVerificationResult, VideoLivenessCompare.

Name	Type	Required	Description

Url	String	Yes	The URL for downloading the file
MD5	String	Yes	The 32-bit MD5 checksum of the file
Size	Integer	Yes	The file size

GeneralCard

General liscense information.

Used by actions: GetCardVerificationResult, GetSdkVerificationResult, GetWebVerificationResultIntl.

Name	Type	Description
LicenseNumber	String	License number Note: This field may return null, indicating that no valid values can be obtained.
PersonalNumber	String	Personal number, which is returned when it is a passport Note: This field may return null, indicating that no valid values can be obtained.
PassportCodeFirst	String	The first line of passport machine reading code Note: This field may return null, indicating that no valid values can be obtained.
PassportCodeSecond	String	The first line of passport machine reading code Note: This field may return null, indicating that no valid values can be obtained.
ExpirationDate	String	Date of expiry in the format of YYYY-MM-DD Note: This field may return null, indicating that no valid values can be obtained.
DueDate	String	Valid date in the format of YYYY-MM-DD Note: This field may return null, indicating that no valid values can be obtained.
IssuedDate	String	Date of issue in the format of YYYY-MM-DD Note: This field may return null, indicating that no valid values can be obtained.
IssuedAuthority	String	Issuing authority Note: This field may return null, indicating that no valid values can be obtained.

IssuedCountry	String	Issuing country, which is returned following the ISO 3166 country coding specification Note: This field may return null, indicating that no valid values can be obtained. Example: MYS
FullName	String	Full Name Note: This field may return null, indicating that no valid values can be obtained.
FirstName	String	First name Note: This field may return null, indicating that no valid values can be obtained.
LastName	String	Last name Note: This field may return null, indicating that no valid values can be obtained.
Sex	String	Gender on the license - M: male - F: female - X: other gender Note: This field may return null, indicating that no valid values can be obtained. Example: M
Age	String	Age. 0 indicates that no valid information is obtained. Example: 0
Birthday	String	Birthday Note: This field may return null, indicating that no valid values can be obtained.
BirthPlace	String	Birth place Note: This field may return null, indicating that no valid values can be obtained.
Nationality	String	Nationality, which is returned following the ISO 3166 country coding specification Note: This field may return null, indicating that no valid values can be obtained. Example: IND
RegistrationNumber	String	Registration number Note: This field may return null, indicating that no valid values can be obtained.

Address	Address	Address Note: This field may return null, indicating that no valid values can be obtained.

HKIDCard

Hong Kong ID card.

Used by actions: GetCardVerificationResult, GetWebVerificationResultIntl.

Name	Type	Description
CnName	String	
EnName	String	English name Note: This field may return null, indicating that no valid values can be obtained. Example: SAN, Nan
TelexCode	String	Telex code correspondint to the Chinese name Note: This field may return null, indicating that no valid values can be obtained.
Sex	String	Gender: "Male-M" or "Female-F" Note: This field may return null, indicating that no valid values can be obtained.
Birthday	String	Birthday Note: This field may return null, indicating that no valid values can be obtained. Example: 01-01-2001
Permanent	String	Permanent resident ID card: 0-non-permanent; 1-permanent; -1-unknown Note: This field may return null, indicating that no valid values can be obtained.
IdNum	String	ID card number Note: This field may return null, indicating that no valid values can be obtained. Example: C000000(E)
Symbol	String	Lisence symbol, which is the symbol below Birthday. Example: "***AZ" Note: This field may return null, indicating that no valid values can be obtained.
FirstIssueDate	String	The first date of issue Note: This field may return null, indicating that no valid values can be obtained.
CurrentIssueDate	String	The current date of issue Note: This field may return null, indicating that no valid values can be obtained.

IndonesiaDrivingLicense

Indonesia driving license.

Used by actions: GetCardVerificationResult, GetSdkVerificationResult, GetWebVerificationResultIntl.

Name	Type	Description
LastName	String	Last name Note: This field may return null, indicating that no valid values can be obtained.
FirstName	String	First name Note: This field may return null, indicating that no valid values can be obtained.
LicenseNumber	String	License number Note: This field may return null, indicating that no valid values can be obtained.
Birthday	String	Birthday Note: This field may return null, indicating that no valid values can be obtained.
Address	String	Address Note: This field may return null, indicating that no valid values can be obtained.
ExpirationDate	String	Expiration date Note: This field may return null, indicating that no valid values can be obtained.
IssuedDate	String	Date of issue Note: This field may return null, indicating that no valid values can be obtained.
IssuedCountry	String	Issuing country Note: This field may return null, indicating that no valid values can be obtained.

IndonesiaIDCard

Indonesia ID card.

Used by actions: GetCardVerificationResult, GetWebVerificationResultIntl.

Name	Type	Description
NIK	String	License number Note: This field may return null, indicating that no valid values can be obtained.
Nama	String	Name Note: This field may return null, indicating that no valid values can be obtained.

TempatTglLahir	String	Birth place/Birthday Note: This field may return null, indicating that no valid values can be obtained.
JenisKelamin	String	Gender Note: This field may return null, indicating that no valid values can be obtained.
GolDarah	String	Blood type Note: This field may return null, indicating that no valid values can be obtained.
Alamat	String	Address Note: This field may return null, indicating that no valid values can be obtained.
RTRW	String	Street Note: This field may return null, indicating that no valid values can be obtained.
KelDesa	String	Village Note: This field may return null, indicating that no valid values can be obtained.
Kecamatan	String	Region Note: This field may return null, indicating that no valid values can be obtained.
Agama	String	Religious beliefs Note: This field may return null, indicating that no valid values can be obtained.
StatusPerkawinan	String	Marital status Note: This field may return null, indicating that no valid values can be obtained.
Perkerjaan	String	Job Note: This field may return null, indicating that no valid values can be obtained.
KewargaNegaraan	String	Nationality Note: This field may return null, indicating that no valid values can be obtained.
BerlakuHingga	String	ID card validity period Note: This field may return null, indicating that no valid values can be obtained.
IssuedDate	String	Date of issue Note: This field may return null, indicating that no valid values can be obtained.
Provinsi	String	Province Note: This field may return null, indicating that no valid values can be obtained.
Kota	String	City Note: This field may return null, indicating that no valid values can be obtained.

InternationalIDPassport

ID cards of Hong Kong, Macao and Taiwan (China), and international passport.

Used by actions: GetCardVerificationResult, GetSdkVerificationResult, GetWebVerificationResultIntl.

Name	Type	Description
LicenseNumber	String	Passport ID Note: This field may return null, indicating that no valid values can be obtained.
FullName	String	Full name Note: This field may return null, indicating that no valid values can be obtained.
Surname	String	Last name Note: This field may return null, indicating that no valid values can be obtained.
GivenName	String	First name Note: This field may return null, indicating that no valid values can be obtained.
Birthday	String	Birthday Note: This field may return null, indicating that no valid values can be obtained.
Sex	String	Gender (F-Female, M-Male) Note: This field may return null, indicating that no valid values can be obtained.
DateOfExpiration	String	Expiration date Note: This field may return null, indicating that no valid values can be obtained.
IssuingCountry	String	Issuing country Note: This field may return null, indicating that no valid values can be obtained.
NationalityCode	String	Nationality code Note: This field may return null, indicating that no valid values can be obtained.
PassportCodeFirst	String	The first line at the bottom, the MRZ Code sequence Note: This field may return null, indicating that no valid values can be obtained.
PassportCodeSecond	String	The second line at the bottom, the MRZ Code sequence Note: This field may return null, indicating that no valid values can be

		obtained.

MLIDCard

Malaysia ID card.

Used by actions: GetCardVerificationResult, GetWebVerificationResultIntl.

Name	Type	Description
Name	String	Full Name Note: This field may return null, indicating that no valid values can be obtained.
ID	String	License number Note: This field may return null, indicating that no valid values can be obtained.
Sex	String	Gender Note: This field may return null, indicating that no valid values can be obtained.
Address	String	Address Note: This field may return null, indicating that no valid values can be obtained.
Type	String	Lisence type MyKad ID card MyPR Permanent resident ID card MyTentera Military ID card MyKAS Temporary ID card POLIS Police ID card IKAD Labor ID card MyKid Juvenile ID card Example: MyKad
Birthday	String	Birthday (Currently, this filed only supports IKAD labor ID card and MyKad ID card) Note: This field may return null, indicating that no valid values can be obtained.

MacaoIDCard

Macao ID Card

Used by actions: GetCardVerificationResult, GetSdkVerificationResult, GetWebVerificationResultIntl.

Name	Type	Description

FirstName	String	First name Note: This field may return null, indicating that no valid values can be obtained.
LastName	String	Last name Note: This field may return null, indicating that no valid values can be obtained.
Birthday	String	Birthday Note: This field may return null, indicating that no valid values can be obtained.
ExpirationDate	String	Expiration date Note: This field may return null, indicating that no valid values can be obtained.
LicenseNumber	String	License number Note: This field may return null, indicating that no valid values can be obtained.
Sex	String	Sex Note: This field may return null, indicating that no valid values can be obtained.
Age	String	Age Note: This field may return null, indicating that no valid values can be obtained.
IssuedCountry	String	Issued country Note: This field may return null, indicating that no valid values can be obtained.
Field1	String	MRZ1 on card Note: This field may return null, indicating that no valid values can be obtained.
Field2	String	MRZ2 on card Note: This field may return null, indicating that no valid values can be obtained.

MainlandIDCard

Mainland ID Card OCR

Used by actions: GetSdkVerificationResult.

Name	Type	Required	Description
FullName	String	No	Chinese name Note: This field may return null, indicating that no valid values can be obtained.
Sex	String	No	Sex Note: This field may return null, indicating that no valid values can be obtained.

Nation	String	No	Nation Note: This field may return null, indicating that no valid values can be obtained.
Birthday	String	No	Birthday Note: This field may return null, indicating that no valid values can be obtained.
LicenseNumber	String	No	License number Note: This field may return null, indicating that no valid values can be obtained.
FormattedAddress	String	No	Address Note: This field may return null, indicating that no valid values can be obtained.

NormalCardInfo

License OCR result

Used by actions: GetSdkVerificationResult.

Name	Type	Description
HKIDCard	NormalHKIDCard	Hong Kong ID Card Note: This field may return null, indicating that no valid values can be obtained.
MLIDCard	NormalMLIDCard	Malaysia ID Card Note: This field may return null, indicating that no valid values can be obtained.
PhilippinesVoteID	PhilippinesVoteID	Philippines VoteID Card Note: This field may return null, indicating that no valid values can be obtained.
IndonesiaIDCard	NormalIndonesiaIDCard	Indonesia ID Card Note: This field may return null, indicating that no valid values can be obtained.
PhilippinesDrivingLicense	PhilippinesDrivingLicense	Philippines Driving License Note: This field may return null, indicating that no valid values can be obtained.
PhilippinesTinID	PhilippinesTinID	Philippines TinID

		Note: This field may return null, indicating that no valid values can be obtained.
PhilippinesSSSID	PhilippinesSSSID	Philippines SSSID Note: This field may return null, indicating that no valid values can be obtained.
PhilippinesUMID	PhilippinesUMID	Philippines UMID Note: This field may return null, indicating that no valid values can be obtained.
InternationalIDPassport	InternationalIDPassport	ID Cards of Hong Kong, Macao and Taiwan (China), and International Passport Note: This field may return null, indicating that no valid values can be obtained.
GeneralCard	GeneralCard	General license information Note: This field may return null, indicating that no valid values can be obtained.
IndonesiaDrivingLicense	IndonesiaDrivingLicense	Indonesia Driving License Note: This field may return null, indicating that no valid values can be obtained.
ThailandIDCard	NormalThailandIDCard	Thailand ID Card Note: This field may return null, indicating that no valid values can be obtained.
SingaporeIDCard	SingaporeIDCard	Singapore ID Card Note: This field may return null, indicating that no valid values can be obtained.
MacaoIDCard	MacaoIDCard	Macao ID Card Note: This field may return null, indicating that no valid values can be obtained.
MainlandIDCard	MainlandIDCard	Mainland ID Card Note: This field may return null, indicating that no valid values can be obtained.

NormalHKIDCard

Hong Kong ID card.

Used by actions: GetSdkVerificationResult.

Name	Type	Description
ChineseName	String	Chinese name Note: This field may return null, indicating that no valid values can be obtained.
FullName	String	English name Note: This field may return null, indicating that no valid values can be obtained. Example: SAN, Nan
RegistrationNumber	String	Telex code correspondint to the Chinese name Note: This field may return null, indicating that no valid values can be obtained.
Sex	String	Gender: "Male-M" or "Female-F" Note: This field may return null, indicating that no valid values can be obtained.
Birthday	String	Birthday Note: This field may return null, indicating that no valid values can be obtained. Example: 01-01-2001
Permanent	String	Permanent resident ID card: 0-non-permanent; 1-permanent; -1-unknown Note: This field may return null, indicating that no valid values can be obtained.
LicenseNumber	String	ID card number Note: This field may return null, indicating that no valid values can be obtained. Example: C000000(E)
Symbol	String	Lisence symbol, which is the symbol below Birthday. Example: "***AZ" Note: This field may return null, indicating that no valid values can be obtained.
IssuedDate	String	The first date of issue Note: This field may return null, indicating that no valid values can be obtained.
CurrentIssueDate	String	The current date of issue Note: This field may return null, indicating that no valid values can be obtained.

NormalIndonesiaIDCard

Indonesia ID card.

Used by actions: GetSdkVerificationResult.

Name	Type	Description
LicenseNumber	String	License number Note: This field may return null, indicating that no valid values can be obtained.
FullName	String	Name Note: This field may return null, indicating that no valid values can be obtained.
Birthday	String	Birth place/Birthday Note: This field may return null, indicating that no valid values can be obtained.
Sex	String	Gender Note: This field may return null, indicating that no valid values can be obtained.
BloodType	String	Blood type Note: This field may return null, indicating that no valid values can be obtained.
FormattedAddress	String	Address Note: This field may return null, indicating that no valid values can be obtained.
Street	String	Street Note: This field may return null, indicating that no valid values can be obtained.
Village	String	Village Note: This field may return null, indicating that no valid values can be obtained.
Area	String	Region Note: This field may return null, indicating that no valid values can be obtained.
Religion	String	Religious beliefs Note: This field may return null, indicating that no valid values can be obtained.
MaritalStatus	String	Marital status Note: This field may return null, indicating that no valid values can be obtained.
Occupation	String	Job Note: This field may return null, indicating that no valid values can be obtained.
Nationality	String	Nationality Note: This field may return null, indicating that no valid values can be obtained.
DueDate	String	ID card validity period

		Note: This field may return null, indicating that no valid values can be obtained.
IssuedDate	String	Date of issue Note: This field may return null, indicating that no valid values can be obtained.
Province	String	Province Note: This field may return null, indicating that no valid values can be obtained.
City	String	City Note: This field may return null, indicating that no valid values can be obtained.

NormalMLIDCard

Malaysia ID card.

Used by actions: GetSdkVerificationResult.

Name	Type	Description
FullName	String	Full Name Note: This field may return null, indicating that no valid values can be obtained.
LicenseNumber	String	License number Note: This field may return null, indicating that no valid values can be obtained.
Sex	String	Gender Note: This field may return null, indicating that no valid values can be obtained.
FormattedAddress	String	Address Note: This field may return null, indicating that no valid values can be obtained.
Type	String	Lisence type MyKad ID card MyPR Permanent resident ID card MyTentera Military ID card MyKAS Temporary ID card POLIS Police ID card IKAD Labor ID card MyKid Juvenile ID card Example: MyKad
Birthday	String	Birthday (Currently, this filed only supports IKAD labor ID card and MyKad ID card) Note: This field may return null, indicating that no valid values can be obtained.

NormalThailandIDCard

Thailand ID Card

Used by actions: GetSdkVerificationResult.

Name	Type	Description
LicenseNumber	String	LicenseNumber Note: This field may return null, indicating that no valid values can be obtained.
FullName	String	Thailand name Note: This field may return null, indicating that no valid values can be obtained.
LastName	String	Last name Note: This field may return null, indicating that no valid values can be obtained.
FirstName	String	First name Note: This field may return null, indicating that no valid values can be obtained.
Birthday	String	Birthday Note: This field may return null, indicating that no valid values can be obtained.
FormattedAddress	String	Address Note: This field may return null, indicating that no valid values can be obtained.
ExpirationDate	String	Expiration date Note: This field may return null, indicating that no valid values can be obtained.
IssuedDate	String	Issued date Note: This field may return null, indicating that no valid values can be obtained.
RegistrationNumber	String	Registration number Note: This field may return null, indicating that no valid values can be obtained.
Religion	String	Religion Note: This field may return null, indicating that no valid values can be obtained.

ThaiBirthday	String	Birthday in Thai Note: This field may return null, indicating that no valid values can be obtained.
ThaiExpirationDate	String	Expiration date in Thai Note: This field may return null, indicating that no valid values can be obtained.
ThaiIssueDate	String	Issued date in Thai Note: This field may return null, indicating that no valid values can be obtained.

OCRResult

The content of a single license in the license information.

Used by actions: GetWebVerificationResultIntl.

Name	Type	Description
IsPass	Boolean	Is the indentity verification or OCR process passed
CardImageBase64	String	The Base64 of ID card image Note: This field may return null, indicating that no valid values can be obtained.
CardInfo	CardInfo	OCR result of the ID card.
RequestId	String	The request id
CardCutImageBase64	String	Base64 of cropped image of ID card
CardBackCutImageBase64	String	Base64 of the cropped image on the reverse side of the ID card

PhilippinesDrivingLicense

Philippines driving license

Used by actions: GetCardVerificationResult, GetSdkVerificationResult, GetWebVerificationResultIntl.

Name	Type	Description
Name	String	Full Name

		Note: This field may return null, indicating that no valid values can be obtained.
LastName	String	Last name Note: This field may return null, indicating that no valid values can be obtained.
FirstName	String	First name Note: This field may return null, indicating that no valid values can be obtained.
MiddleName	String	Middle name Note: This field may return null, indicating that no valid values can be obtained.
Nationality	String	Nationality Note: This field may return null, indicating that no valid values can be obtained.
Sex	String	Gender Note: This field may return null, indicating that no valid values can be obtained.
Address	String	Address Note: This field may return null, indicating that no valid values can be obtained.
LicenseNo	String	License number Note: This field may return null, indicating that no valid values can be obtained.
ExpiresDate	String	Date of expiry Note: This field may return null, indicating that no valid values can be obtained.
AgencyCode	String	Agency code Note: This field may return null, indicating that no valid values can be obtained.
Birthday	String	Birthday Note: This field may return null, indicating that no valid values can be obtained.

PhilippinesSSSID

Philippines SSSID Card

Used by actions: GetCardVerificationResult, GetSdkVerificationResult, GetWebVerificationResultIntl.

Name	Type	Description
LicenseNumber	String	License number Note: This field may return null, indicating that no valid values can be obtained.
FullName	String	Full name Note: This field may return null, indicating that no valid values can be obtained.

Birthday	String	Birthday Note: This field may return null, indicating that no valid values can be obtained.

PhilippinesTinID

Philippines TinID Card

Used by actions: GetCardVerificationResult, GetSdkVerificationResult, GetWebVerificationResultIntl.

Name	Type	Description
LicenseNumber	String	License number Note: This field may return null, indicating that no valid values can be obtained.
FullName	String	Full name Note: This field may return null, indicating that no valid values can be obtained.
Address	String	Address Note: This field may return null, indicating that no valid values can be obtained.
Birthday	String	Birthday Note: This field may return null, indicating that no valid values can be obtained.
IssueDate	String	Date of issue Note: This field may return null, indicating that no valid values can be obtained.

PhilippinesUMID

Philippines UMID Card

Used by actions: GetCardVerificationResult, GetSdkVerificationResult, GetWebVerificationResultIntl.

Name	Type	Description
Surname	String	Surname Note: This field may return null, indicating that no valid values can be obtained.
MiddleName	String	Middle Name Note: This field may return null, indicating that no valid values can be obtained.
GivenName	String	First name Note: This field may return null, indicating that no valid values can be obtained.

Sex	String	Gender Note: This field may return null, indicating that no valid values can be obtained.
Birthday	String	Birthday Note: This field may return null, indicating that no valid values can be obtained.
Address	String	Address Note: This field may return null, indicating that no valid values can be obtained.
CRN	String	CRN code Note: This field may return null, indicating that no valid values can be obtained.

PhilippinesVoteID

Philippines VoteID Card

Used by actions: GetCardVerificationResult, GetSdkVerificationResult, GetWebVerificationResultIntl.

Name	Type	Description
VIN	String	VIN of Philippines VoteID Note: This field may return null, indicating that no valid values can be obtained.
FirstName	String	First name Note: This field may return null, indicating that no valid values can be obtained.
LastName	String	Last name Note: This field may return null, indicating that no valid values can be obtained.
Birthday	String	Birthday Note: This field may return null, indicating that no valid values can be obtained.
CivilStatus	String	Civil status Note: This field may return null, indicating that no valid values can be obtained.
Citizenship	String	Nationality Note: This field may return null, indicating that no valid values can be obtained.
Address	String	Address Note: This field may return null, indicating that no valid values can be obtained.
PrecinctNo	String	Region Note: This field may return null, indicating that no valid values can be obtained.

SingaporeIDCard

Singapore ID Card

Used by actions: GetCardVerificationResult, GetSdkVerificationResult, GetWebVerificationResultIntl.

Name	Type	Description
ChName	String	Chinese name Note: This field may return null, indicating that no valid values can be obtained.
EnName	String	English name Note: This field may return null, indicating that no valid values can be obtained.
ID	String	License number Note: This field may return null, indicating that no valid values can be obtained.
Sex	String	Gender Note: This field may return null, indicating that no valid values can be obtained.
CountryOfBirth	String	Country of birth Note: This field may return null, indicating that no valid values can be obtained.
Birthday	String	Birthday Note: This field may return null, indicating that no valid values can be obtained.
Address	String	Address (on the back) Note: This field may return null, indicating that no valid values can be obtained.
Race	String	Nationality (on the back) Note: This field may return null, indicating that no valid values can be obtained.
NRICCode	String	NRIC number (on the back) Note: This field may return null, indicating that no valid values can be obtained.
PostCode	String	Post number (on the front) Note: This field may return null, indicating that no valid values can be obtained.
DateOfExpiration	String	Date of expiry (on the back) Note: This field may return null, indicating that no valid values can be obtained.
DateOfIssue	String	Date of issue (on the back) Note: This field may return null, indicating that no valid values can be obtained.

ThailandIDCard

Thailand ID Card

Used by actions: GetCardVerificationResult, GetWebVerificationResultIntl.

Name	Type	Description
LastName	String	Last name Note: This field may return null, indicating that no valid values can be obtained.
FirstName	String	First name Note: This field may return null, indicating that no valid values can be obtained.
LicenseNumber	String	License number Note: This field may return null, indicating that no valid values can be obtained.
DateOfBirth	String	Birthday Note: This field may return null, indicating that no valid values can be obtained.
DateOfExpiry	String	Date of expiry Note: This field may return null, indicating that no valid values can be obtained.
DateOfIssue	String	Date of issue Note: This field may return null, indicating that no valid values can be obtained.
IssuedCountry	String	Issuing country Note: This field may return null, indicating that no valid values can be obtained.

VerificationDetail

The details of the verification process.

Used by actions: GetWebVerificationResult, GetWebVerificationResultIntl.

Name	Type	Description
ErrorCode	Integer	The final result of this verification. `0` indicates that the person is the same as that in the photo. Note: u200dThis field may return null, indicating that no valid values can be obtained.
ErrorMsg	String	The description of the final verification result. Note: u200dThis field may return null, indicating that no valid values can be obtained.
LivenessErrorCode	Integer	The result of this liveness detection process. `0` indicates success. Note: u200dThis field may return null, indicating that no valid values can be

		obtained.
LivenessErrorMsg	String	The result description of this liveness detection process. Note: This field may return null, indicating that no valid values can be obtained.
CompareErrorCode	Integer	The result of this comparison process. `0` indicates that the person in the best face screenshot collected from the video stream is the same as that in the uploaded image for comparison. Note: u200dThis field may return null, indicating that no valid values can be obtained.
CompareErrorMsg	String	The result description of this comparison process. Note: u200dThis field may return null, indicating that no valid values can be obtained.
ReqTimestamp	Integer	The timestamp (ms) of this verification process. Note: This field may return null, indicating that no valid values can be obtained.
Similarity	Float	The similarity of the best face screenshot collected from the video stream and the uploaded image for comparison in this verification process. Value range: [0.00, 100.00]. By default, the person in the screenshot is determined to be the same person in the image if the similarity is greater than or equal to 70. Note: u200dThis field may return null, indicating that no valid values can be obtained.
Seq	String	Unique ID of this verification process. Note: This field may return null, indicating that no valid values can be obtained.

WebVerificationConfigIntl

eKYC Web related configuration

Used by actions: ApplyWebVerificationBizTokenIntl.

Name	Type	Required	Description
AutoSkipStartPage	Boolean	No	When starting verification, whether to skip the starting verification page. If true, enter the verification process directly. The default is false. This configuration will not take effect if the downgrade policy is triggered.
AutoSkip	Boolean	No	When the verification passed, whether to skip the

			result page and automatically jump to RedirectURL. The default value is false. Example value: false
CheckMode	Integer	No	Detection mode, parameter values are as follows: 1: OCR+living detection & face comparison; 2: Living detection & face comparison; 3: Living detection; The default value is 2. Example value: 3
IDCardType	String	No	The type of lisence used for verification. The following types are supported. 1.HKIDCard: Hong Kong (China) ID card 2.MLIDCard: Malaysia ID card 3.IndonesiaIDCard: Indonesia ID card 4.PhilippinesVoteID: Philippines VoteID card 5.PhilippinesDrivingLicense: Philippines driving license 6.PhilippinesTinID: Philippines TinID card 7.PhilippinesSSSID: Philippines SSSID card 8.PhilippinesUMID: Philippines UMID card 9.InternationalIDPassport: ID cards of Hong Kong, Macao and Taiwan (China), and international passport. Example: HKIDCard
DisableCheckOcrWarnings	Boolean	No	Whether to turn off document alarms, the default is false (the alarm detection function is turned on). When enabled, the identity authentication process will be intercepted based on the alarm status of the certificate. If you need to use the document authentication function, please contact us.
SecurityLevel	Integer	No	Liveness security level: 1:Silent mode;2:Action mode;3:Lighting mode;4:Action+Lighting mode;default value is 3
SkipPrivacyPolicy	Boolean	No	Whether to skip the agreement page, the default is false. When SkipPrivacyPolicy is false, the agreement page will be displayed and the privacy agreement needs to be checked; when SkipPrivacyPolicy is true, the agreement page will be skipped and the liveness process will be entered directly without checking the privacy agreement page.
IdCardCutReturn	Boolean	No	The default value is false. If it is false, the original ID

			image will be displayed. If it is true, the cut ID image will be displayed.
ThemeColor	String	No	Front-end theme color, in the format of RGB hexadecimal color code. The default value is "#2d72+1". If the format is incorrect, the default value color will be used.
Language	String	No	International language, the default value is en (English). Currently supported: th: Thai; en: English;
AutoDowngrade	Integer	No	Automatic downgrade mode, with the following parameter values: 1: Downgrade to silent live mode; 2: Disable downgrade mode. The default value is 1.

Error Codes

最近更新时间：2024-05-16 19:40:45

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.
IpInBlacklist	Your IP is in uin IP blacklist.
IpNotInWhitelist	Your IP is not in uin IP whitelist.
LimitExceeded	Quota limit exceeded.
MissingParameter	A parameter is missing.
NoSuchProduct	The product does not exist.
NoSuchVersion	The API version does not exist.
RequestLimitExceeded	The number of requests exceeds the frequency limit.
RequestLimitExceeded.GlobalRegionUinLimitExceeded	Uin exceeds the frequency limit.
RequestLimitExceeded.IPLimitExceeded	The number of ip requests exceeds the frequency limit.
RequestLimitExceeded.UinLimitExceeded	The number of uin requests exceeds the frequency

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

Service Error Codes

Error Code	Description
FailedOperation.ActionCloseEye	No motions of eye closing are detected.
FailedOperation.ActionFaceClose	The face is too close to the screen.
FailedOperation.ActionFaceFar	The face is too far from the screen.
FailedOperation.ActionFaceLeft	The face is too far left from the screen.
FailedOperation.ActionFaceRight	The face is too far right from the screen.
FailedOperation.ActionFirstAction	No movement is detected.
FailedOperation.ActionLightDark	The lighting is too dim.
FailedOperation.ActionLightStrong	The lighting is too strong.
FailedOperation.ActionNodetectFace	Failed to detect a full face.

FailedOperation.ActionOpenMouth	No motions of mouth opening are detected.
FailedOperation.CompareFail	Comparison failed.
FailedOperation.CompareLowSimilarity	The comparison similarity did not reach the passing standard.
FailedOperation.CompareSystemError	Error calling the comparison engine API.
FailedOperation.DownLoadError	File download failed.
FailedOperation.DownLoadTimeoutError	File download timed out.
FailedOperation.ImageBlur	The image is blurry.
FailedOperation.ImageDecodeFailed	Image decoding failed.
FailedOperation.ImageSizeTooLarge	The image is too large.
FailedOperation.LifePhotoDetectFaces	Multiple faces are detected.
FailedOperation.LifePhotoDetectFake	Real person comparison failed.
FailedOperation.LifePhotoDetectNoFaces	Failed to detect a full face.
FailedOperation.LifePhotoPoorQuality	The resolution of the image passed in is too low. Please upload a new one.
FailedOperation.LifePhotoSizeError	The image passed in is too large or too small.
FailedOperation.LipFaceIncomplete	The face is not fully exposed.
FailedOperation.LipMoveSmall	The lip movement range is too small.
FailedOperation.LipNetFailed	Failed to pull the video. Please try again.
FailedOperation.LipSizeError	The video is empty or its size is inappropriate. The recording duration should be about 6 seconds.
FailedOperation.LipVideoInvalid	The video format is incorrect.
FailedOperation.LipVideoQuaility	The video definition is too low.
FailedOperation.LipVoiceDetect	No sound is detected.
FailedOperation.LipVoiceLow	The volume of the video is too low.
FailedOperation.LipVoiceRecognize	Speech recognition failed.
FailedOperation.LivessBestFrameError	Face detection failed. Unable to extract the photo for

	comparison.
FailedOperation.LivessDetectFail	Liveness detection failed.
FailedOperation.LivessDetectFake	Suspected spoofed recording.
FailedOperation.LivessSystemError	Error calling the liveness engine API.
FailedOperation.LivessUnknownError	Video-based real person detection failed.
FailedOperation.SilentDetectFail	Real person detection failed.
FailedOperation.SilentEyeLiveFail	Eye detection failed.
FailedOperation.SilentFaceDetectFail	No face is detected in the video.
FailedOperation.SilentFaceQualityFail	Low face quality.
FailedOperation.SilentFaceWithMaskFail	A face mask is detected.
FailedOperation.SilentMouthLiveFail	Mouth detection failed.
FailedOperation.SilentMultiFaceFail	Multiple faces are detected in the video.
FailedOperation.SilentPictureLiveFail	The video might be spoofed.
FailedOperation.SilentThreshold	Real person detection did not reach the passing standard.
FailedOperation.SilentTooShort	The video is too short. Please capture a video longer than 2 seconds.
FailedOperation.UnKnowError	Unknown error.
FailedOperation.UnKnown	Unknown internal error.
FailedOperation.UnOpenError	The service is not activated.
InternalError.ActionLightDark	The lighting is too dim.
InternalError.ActionLightStrong	The lighting is too strong.
InternalError.ActionNodetectFace	Failed to detect a full face.
InternalError.CompareLowSimilarity	The similarity did not reach the passing standard.
InternalError.LifePhotoPoorQuality	The resolution of the image passed in is too low. Please upload a new one.
InternalError.LifePhotoSizeError	The image passed in is too large or too small.

InternalError.UnKnown	Unknown internal error.
InvalidParameter.EngineImageDecodeFailed	Image decoding failed.
InvalidParameterValue.BizTokenExpired	BizToken expired.
InvalidParameterValue.BizTokenIllegal	Invalid BizToken.
InvalidParameterValue.InvalidFileContentSize	The image file content size is abnormal.
InvalidParameterValue.InvalidParameterValueLimit	Parameter value is wrong.
OperationDenied	Operation denied.
ResourceUnavailable.ImageDownloadError	Image file download failed.
ResourceUnavailable.InArrears	The account is in arrears.
ResourceUnavailable.ResourcePackageRunOut	The account has exhausted the resource package for this service.
UnauthorizedOperation.ActivateError	Service activation exception.
UnauthorizedOperation.Activating	Activating the service.
UnauthorizedOperation.Arrears	The account is in arrears.
UnauthorizedOperation.NonAuthorize	Identity verification has not been completed for the account.
UnauthorizedOperation.Nonactivated	The service has not been activated.