

Face ID

Getting Started

Product Documentation



Copyright Notice

©2013-2022 Tencent Cloud. All rights reserved.

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

Trademark Notice

 Tencent Cloud

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

Service Statement

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

Contents

Getting Started

Process Guide

Quick API Run

Error Codes

Getting Started

Process Guide

Last updated : 2022-06-09 11:32:54

This document describes relevant operations and provides documentation links to help you better use Tencent Cloud FaceID.

Access Process

Step 1. Sign up and log in

Log in to [Tencent Cloud](#). If you do not have an account yet, sign up as instructed in [Signing Up](#).

Step 2. Apply for the service

Log in to the [FaceID console](#) to activate the service.

Step 3. Use the service

We recommend using the [API 3.0 Explorer](#) for online call, signature verification, SDK code generation, quick API search and more, which significantly reduces the difficulty in using TencentCloud API 3.0. You can use [API 3.0 Explorer](#) to generate **server SDKs** and use them with the compiled Tencent Cloud SDKs to quickly call the FaceID service. The server SDKs are available in multiple programming languages, such as Python, Java, PHP, Go, Node.js, .NET, etc. You can download the corresponding SDK in the documentation of each service or get it from the [SDK Center](#).

FaceID also provides a "Quick API Run" tutorial to help you debug and test the service.

Step 4. View calls

For more information on FaceID billing and payment modes, see [Billing Overview](#).

You can log in to the [FaceID console](#) to view call details.

Quick API Run

Last updated : 2022-06-14 16:12:44

Overview

This document describes how to call Tencent Cloud FaceID APIs through API 3.0 Explorer and integrate SDKs in the corresponding programming language into your project after you purchase the FaceID service. You can access FaceID APIs quickly in the following steps.

Prerequisites

Enter [API 3.0 Explorer](#) to call APIs in the following steps.


Directions

Step 1

Select the CompareFace API on the left sidebar.


Step 2

Enter your private key information and required parameters.

Private Key [View Key](#) 

SecretId

SecretKey

[More Options](#) 

Input Parameters View Only Required Parameters

Region

- Region: region information in the domain name that determines the access point. For example, `faceid.ap-shanghai.tencentcloudapi.com` indicates that Shanghai is the access point. The common parameter `Region` specifies the region of business resources to be accessed; for example, `Region=ap-beijing` indicates resources in the Beijing region will be accessed. If no region is specified in the domain name, a nearby region will be accessed by default, which may cause problems. If an IP cannot be resolved, Guangzhou region will be used by default. The region for the domain name and the common parameter `Region` can be different, but this may increase access latency. We recommend using the same region for the domain name and the common parameter `Region`, such as South China (Guangzhou)/ap-guangzhou.
- RuleId: Used to specify use cases. After your apply to activate the service, you can create RuleId in the [FaceID console](#) and call it after your application is approved. If you have any questions, contact the FaceID WeChat assistant (account: faceid001).

Step 3

Select the programming language to generate codes.

Enter the parameters on the left to generate codes. Part of the field information in the generated code is subject to the entered content. To adjust an input parameter, modify its value on the left and generate the code again.

Step 4

Integrate the SDK into the project.

Integrate the SDK into the project as instructed in the usage guide on the upper right-hand corner. You can call the corresponding API using the code generated in [Step 3](#).

The screenshot displays the 'Private Key' configuration page on the left and the 'JAVA SDK Usage Guide' on the right. In the configuration page, the 'SecretId' and 'SecretKey' fields are filled with 'secretid' and 'secretkey' respectively. The 'Region' dropdown is set to 'ap-tokyo'. The SDK code on the right shows the initialization of the FaceID client with these credentials and the region.

```

import com.tencentcloudapi.common.Credential;
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;
import com.tencentcloudapi.common.exception.TencentCloudSDKException;

import com.tencentcloudapi.faceid.v20180301.FaceidClient;

import com.tencentcloudapi.faceid.v20180301.models.LivenessCompareRequest;
import com.tencentcloudapi.faceid.v20180301.models.LivenessCompareResponse;

public class LivenessCompare
{
    public static void main(String [] args) {
        try{
            Credential cred = new Credential("secretid", "secretkey");

            HttpProfile httpProfile = new HttpProfile();
            httpProfile.setEndpoint("faceid.tencentcloudapi.com");

            ClientProfile clientProfile = new ClientProfile();
            clientProfile.setHttpProfile(httpProfile);

            FaceidClient client = new FaceidClient(cred, "ap-tokyo", clientProfile);

            String params = "{}";
            LivenessCompareRequest req = LivenessCompareRequest.fromJsonString(params, LivenessCompareRequest.class);

```

Notes

- You only need to focus on the `Region` field in common parameters when using SDKs to make calls. We recommend using `ap-guangzhou` for both the domain name and `Region`.
- Address for generating `SecretId` / `SecretKey` : `https://console.tencentcloud.com/cam/capi`.
- For base64-encoded images or videos, remove the `data:image/jpeg;base64,` prefix and the `\n` line break.
- If the request result is as shown below, you need to manually configure the signature type:

```

[TencentCloudSDKException]message:AuthFailure.SignatureFailure-The provided credentials
could not be validated because of exceeding request size limit, you need to use
new signature
method `TC3-HMAC-SHA256`. requestId:719970d4-5814-4dd9-9757-a3f11ecc9b20

```

Configure the signature type:

```
clientProfile.setSignMethod("TC3-HMAC-SHA256"); // Specify the signature algorithm (default value: HmacSHA256)
```

- If the API request exceeds 1 MB, only v3 authentication (TC3-HMAC-SHA256) can be used. API 3.0 SDK supports Node.js, Python, Java, PHP, and Go, but not .NET and C#. For unsupported languages, you need to implement API authentication v3 to call APIs, and we recommend using the signature generation tool in [API 3.0 Explorer](#) to verify the signature.

Code Generating Online Call **Signature generation** Parameter Description Feedback

Signature generation

Select the signature version:

API 3.0 Signature V3
API 3.0 Signature V1
API 2.0 Signature

For the API 3.0 signature, please click the "Generate Signature" button below. The system will generate a signature as an example to show you the signing process step by step. Finally, you will be provided with the signature data generated by POST. [View signature document](#) (When the parameter changes, you need to click the button to regenerate the signature process data)

[Generate signature](#)

Error Codes

Last updated : 2022-12-01 12:21:12

PaaS error codes

Error Code	Description	Billable
Success	Detection succeeded.	Yes
FailedOperation.CompareLowSimilarity	The similarity did not reach the passing standard.	Yes
InvalidParameter	Invalid parameter.	No
InvalidParameterValue	Invalid parameter value.	No
FailedOperation.DownLoadError	Download error.	No
FailedOperation.UnKnown	Unknown error.	No
FailedOperation.IdPhotoPoorQuality	The identity photo was in a too low resolution. Please upload a new one.	No
FailedOperation.LifePhotoDetectFaces	Several faces were detected.	No
FailedOperation.LifePhotoDetectFake	Real person comparison failed.	No
FailedOperation.LifePhotoDetectNoFaces	No full face was detected.	No
FailedOperation.CompareFail	Comparison failed.	No
FailedOperation.LivessSystemError	An error occurred while calling the liveness detection engine API.	No

SaaS error codes

Error Code	Description
0	Succeeded.
1	Invalid request. Some parameters are missing.

Error Code	Description
2	Invalid request. Parameters are incorrect.
3	Invalid service request.
4	System error.
8	Failed to store the file. Please try again later.
9	Invalid file format. Please try again later.
10	File system error. Please try again later.
11	File system timeout. Please try again later.
14	The verification was finished.
15	The token expired. Please try again.
16	Too many attempts.
17	The process wasn't finished.
202	Failed to obtain the video.
1001	An error occurred while calling the liveness detection engine API.
1002	Suspected spoofed recording.
1003	Video-based real person detection failed.
1004	Face detection failed. It was unable to extract the photo for comparison.
1005	Liveness detection failed.
1101	No voice was detected.
1102	The face was not fully exposed.
1103	Voice recognition failed.
1104	Invalid video format.
1105	Failed to pull the video. Please try again.
1106	The volume of the video is too low.
1107	The video is empty or its size is inappropriate. The video duration should be about 6s.

Error Code	Description
1108	The video resolution is too low.
1109	The lip movement range is too small.
1201	The lighting is too dim.
1202	The lighting is too strong.
1203	The face is too close to the screen.
1204	The face is too far right from the screen.
1205	The face is too far from the screen.
1206	The face is too far left from the screen.
1207	No eye closing motions were detected.
1208	The first motion was not detected.
1209	No mouth opening motions were detected.
1210	No full face was detected.
1301	Real person detection failed.
1302	Real person detection did not reach the passing standard.
1303	The video is too short. Please capture a video longer than 2s.
1401	Reflection-based liveness detection failed.
2001	An error occurred while calling the comparison engine API.
2004	The image passed in is too large or too small.
2009	The image passed in is in a too low resolution. Please upload a new one.
2010	Face detection failed. It was unable to extract the photo for comparison.
2011	Real person comparison failed.
2012	Several faces were detected.
2013	No full face was detected.
2014	The image passed in is in a too low resolution. Please upload a new one.

Error Code	Description
2015	Comparison failed.
2016	The similarity did not reach the passing standard.