Face Recognition Service

API Reference

Issue 05
Date 2018-05-23
Copyright © Huawei Technologies Co., Ltd. 2020. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions

and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd. All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://www.huawei.com
Email: support@huawei.com
# Contents

1 Before You Start .................................................................................................................. 1  
1.1 Overview .......................................................................................................................... 1  
1.2 API Calling ...................................................................................................................... 1  
1.3 Endpoints ......................................................................................................................... 1  
1.4 Restrictions and Limitations .......................................................................................... 2  
1.5 Concepts ......................................................................................................................... 3  
  
2 API Overview ...................................................................................................................... 5  
  
3 Calling APIs ....................................................................................................................... 7  
3.1 Applying for Face Recognition ....................................................................................... 7  
3.2 Making an API Request .................................................................................................. 8  
3.3 Authentication ............................................................................................................... 12  
3.4 Response ....................................................................................................................... 13  
  
4 Face Recognition APIs (V1) .............................................................................................. 16  
4.1 Querying Service Status ............................................................................................... 16  
4.2 Face Detection ............................................................................................................... 18  
4.3 Face Verification .......................................................................................................... 27  
4.4 Face LiveDetect .......................................................................................................... 31  
4.5 Face Retrieval .............................................................................................................. 38  
4.6 Facial Image Library Management .............................................................................. 43  
4.6.1 Creating a Facial Image Library .............................................................................. 43  
4.6.2 Querying All Facial Image Libraries ...................................................................... 46  
4.6.3 Querying a Facial Image Library ........................................................................... 48  
4.6.4 Deleting a Facial Image Library ............................................................................ 50  
4.7 Facial Resource Management ...................................................................................... 51  
4.7.1 Adding a Face ......................................................................................................... 51  
4.7.2 Querying a Face ...................................................................................................... 56  
4.7.3 Updating a Face ...................................................................................................... 58  
4.7.4 Deleting a Face ...................................................................................................... 61  
4.7.5 Batch Deleting Faces ............................................................................................ 63  
  
5 Face Recognition APIs (V2) .............................................................................................. 66  
5.1 Face Detection .............................................................................................................. 66  
5.2 Face Verification ............................................................................................................ 70
1 Before You Start

1.1 Overview

Face Recognition quickly detects faces in images, analyzes key facial points, obtains face attributes, and implements precise face comparison and search. This service is perfect for identity authentication, e-attendance, trace tracking, and passenger flow analysis.

Face Recognition provides open Application Programming Interfaces (APIs) for users. You can use these APIs as instructed by this document.

If you plan to access Face Recognition through an API, ensure that you are familiar with Face Recognition concepts. For details, see Service Overview.

1.2 API Calling

Face Recognition supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details about API calling, see Calling APIs.

Additionally, Face Recognition offers software development kits (SDKs) for multiple programming languages. For details about how to use SDKs, see Face Recognition SDK Reference.

1.3 Endpoints

An endpoint is the request address for calling an API. Endpoints vary depending on services and regions. For the endpoints of all services, see Regions and Endpoints.

Table 1-1 lists Face Recognition endpoints. Select a desired one based on service requirements.
Table 1-1 Face Recognition endpoints

<table>
<thead>
<tr>
<th>Region</th>
<th>Endpoint Region</th>
<th>Endpoint</th>
<th>Protocol</th>
<th>Deployed Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN North-Beijing1</td>
<td>cn-north-1</td>
<td>face.cn-north-1.myhuaweicloud.com</td>
<td>HTTPS</td>
<td>Face Recognition APIs (V1)</td>
</tr>
<tr>
<td>CN North-Beijing4</td>
<td>cn-north-4</td>
<td>face.cn-north-4.myhuaweicloud.com</td>
<td>HTTPS</td>
<td>Face Recognition APIs (V1) and Face Recognition APIs (V2)</td>
</tr>
<tr>
<td>CN East-Shanghai1</td>
<td>cn-east-3</td>
<td>face.cn-east-3.myhuaweicloud.com</td>
<td>HTTPS</td>
<td>Face Recognition APIs (V2)</td>
</tr>
<tr>
<td>CN South-Guangzhou</td>
<td>cn-south-1</td>
<td>face.cn-south-1.myhuaweicloud.com</td>
<td>HTTPS</td>
<td>Face Recognition APIs (V1) and Face Recognition APIs (V2)</td>
</tr>
</tbody>
</table>

1.4 Restrictions and Limitations

Due to various factors such as technology and cost, Face Recognition has some restrictions. The system-wide restrictions affect all sub-services. In addition to system-wide restrictions, sub-services have their independent restrictions.

System-Wide Restrictions

- Only images in JPG, PNG, JPEG, or BMP format can be recognized.
- Each user can use 10 facial image libraries free of charge. Each library contains a maximum of 100,000 facial features. If you want to expand the scale of a facial image library, contact HUAWEI CLOUD customer service to confirm the expansion scale and price.
- Use standard JSON format in the body of the `application/json` request.
- Do not use carriage return characters in Base64 code.
- The system does not save images or videos of users.

Face Detection/Verification/Retrieval (V1/V2)

- The total size of the two images input for face verification is less than 8 MB.
- The image size must be **less than 8 MB**. If the image is too large, the latency is long and the image information volume is small. It is recommended that the image size be **less than 1 MB**.
The image resolution must be less than 4,096 x 2,160. The face resolution in an image must be greater than 80 x 80. It is recommended that the face resolution be greater than 120 x 120.

To ensure the recognition effect, facial images need to meet the following requirements:

a. The illumination should be greater than 200 lux and there is no light reflection or shadow caused by strong light.

b. The overall image is clear without obvious motion blur and the face in it is not blocked.

c. The side face angle does not exceed 30°, and the tilt angle and horizontal angle do not exceed 15°. The face in an image must be a vertically placed front face.

Face LiveDetect

- Currently, only video files and Base64-encoded videos can be detected. User clients need to obtain the video streams, save them as files, and then call the LiveDetect API.
- The size of a video file cannot exceed 8 MB. It is recommended that the video file be compressed to 200 KB to 2 MB on the client.
- The video duration must be 1 to 15 seconds.
- The recommended frame rate is 10 fps to 30 fps.
- The encapsulation format can be MP4, AVI, FLV, WEBM, ASF, or MOV.
- The video encoding format can be H.261, H.263, H.264, HEVC, VC-1, VP8, VP9, or WMV3.

1.5 Concepts

- Account
  An account is created upon successful registration with HUAWEI CLOUD. The account has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity and should not be used directly to perform routine management. For security purposes, create IAM users and grant them permissions for routine management.

- User
  An IAM user is created using an account to use cloud services. Each IAM user has its own identity credentials (password and access keys).
  An IAM user can view the account ID and user ID on the My Credentials page of the console. The account name, username, and password will be required for API authentication.

- Region
  Regions are divided based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified into universal regions and dedicated regions. A
universal region provides universal cloud services for common tenants. A dedicated region provides specific services for specific tenants. For details, see Region and AZ.

- **Availability Zone (AZ)**
  
  An AZ contains one or more physical data centers. Each AZ has independent cooling, fire extinguishing, moisture-proof, and electricity facilities. Within an AZ, computing, network, storage, and other resources are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to support cross-AZ high-availability systems.

- **Project**
  
  Projects group and isolate resources (including compute, storage, and network resources) across physical regions. A default project is provided for each HUAWEI CLOUD region, and subprojects can be created under each default project. Users can be granted permissions to access all resources in a specific project. For more refined access control, create subprojects under a project and purchase resources in the subprojects. Users can then be assigned permissions to access only specific resources in the subprojects.

![Figure 1-1 Project isolating model](image)
APIs provided by Face Recognition comply with RESTful API design specifications. Table 2-1 lists the APIs.

**Table 2-1 Face Recognition APIs**

<table>
<thead>
<tr>
<th>Type</th>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Query</td>
<td><strong>Querying Service Status</strong></td>
<td>This API is used to query the service enabling status.</td>
</tr>
<tr>
<td>Face Detection</td>
<td><strong>Face Detection</strong></td>
<td>This API is used to detect, locate, and analyze the face in an input image, and output the key facial points and attributes.</td>
</tr>
<tr>
<td>Face Verification</td>
<td><strong>Face Verification</strong></td>
<td>This API is used to compare two faces to verify whether they belong to the same person and return the confidence level. If each input image contains multiple faces, the API selects the largest face for comparison.</td>
</tr>
<tr>
<td>Face LiveDetect</td>
<td><strong>Face LiveDetect</strong></td>
<td>Face LiveDetect determines whether a person in a video is alive by checking whether the person's actions in the video are consistent with those in the input action list. If multiple faces appear, the largest face is selected.</td>
</tr>
<tr>
<td>Face Retrieval</td>
<td><strong>Face Retrieval</strong></td>
<td>This API is used to search an existing facial image library for one or more faces similar to the input face, and return corresponding confidence levels.</td>
</tr>
<tr>
<td>Type</td>
<td>API</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Facial Image Library Management</td>
<td>Creating a Facial Image Library</td>
<td>This API is used to create a facial image library for storing facial features. You can create a maximum of 10 facial image libraries. Each library can contain a maximum of 100,000 facial features. If you need higher specifications, contact the customer service personnel.</td>
</tr>
<tr>
<td></td>
<td>Querying All Facial Image Libraries</td>
<td>This API is used to query the statuses of all facial image libraries of the current user.</td>
</tr>
<tr>
<td></td>
<td>Querying a Facial Image Library</td>
<td>This API is used to query the status of a facial image library.</td>
</tr>
<tr>
<td></td>
<td>Deleting a Facial Image Library</td>
<td>This API is used to delete a facial image library and all faces in the library.</td>
</tr>
<tr>
<td></td>
<td>Adding a Face</td>
<td>This API is used to add faces to a facial image library. All detected faces will be added to the library.</td>
</tr>
<tr>
<td></td>
<td>Querying a Face</td>
<td>This API is used to query the face information in a specified facial image library.</td>
</tr>
<tr>
<td></td>
<td>Updating a Face</td>
<td>This API is used to update a single face based on its face ID (face_id).</td>
</tr>
<tr>
<td></td>
<td>Deleting a Face</td>
<td>This API is used to delete a face from the facial image library based on the specified field.</td>
</tr>
<tr>
<td></td>
<td>Batch Deleting Faces</td>
<td>This API is used to batch delete multiple faces that meet specified criteria you customize.</td>
</tr>
</tbody>
</table>
3.1 Applying for Face Recognition

Before using the APIs, you must apply for Face Recognition as follows:

**NOTE**

Before applying for Face Recognition, register with HUAWEI CLOUD and complete real-name authentication.

**Procedure**

1. Log in to the Face Recognition Management Console on HUAWEI CLOUD.
2. Select the sub-service to be enabled based on service requirements, for example, Face Detection, and click Enable Service on the right.
3. Click Service Authorization in the upper right corner to complete OBS authorization. Figure 3-1 shows the page.

**NOTE**

- You also need to authorize Face Recognition to access data stored on the Object Storage Service (OBS) to ensure highest performance.
- During OBS authorization, if the system shows that the number of agencies reaches the upper limit, log in to the Identity and Access Management service console to delete or create agencies.
- When the service status is Enabled, you can call the APIs.
3.2 Making an API Request

This section describes the structure of a REST API request, and uses the IAM API for obtaining a user token as an example to demonstrate how to call an API. The obtained token can then be used to authenticate the calling of other APIs.

Request URI

A request URI is in the following format:

{URI-scheme} :// {Endpoint} / {resource-path} ? {query-string}

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI-scheme</td>
<td>Protocol used to transmit requests. All APIs use HTTPS.</td>
</tr>
<tr>
<td>Endpoint</td>
<td>Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from Endpoints. For example, the endpoint of IAM in the CN North-Beijing4 region is iam.cn-north-4.myhuaweicloud.com.</td>
</tr>
<tr>
<td>resource-path</td>
<td>Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the resource-path of the API used to obtain a user token is /v3/auth/tokens.</td>
</tr>
<tr>
<td>query-string</td>
<td>Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of &quot;Parameter name=Parameter value&quot;. For example, ? limit=10 indicates that a maximum of 10 data records will be displayed.</td>
</tr>
</tbody>
</table>

For example, to obtain an IAM token in CN North-Beijing4, obtain the endpoint of IAM (iam.cn-north-4.myhuaweicloud.com) for this region and the resource-
path (/v3/auth/tokens) in the URI of the API used to obtain a user token. Then, construct the URI as follows:


**NOTE**

To simplify the URI display in this document, each API is provided only with a resource-path and a request method. The URI-scheme of all APIs is HTTPS, and the endpoints of all APIs in the same region are identical.

**Request Method**

The HTTP protocol defines the following request methods that can be used to send a request to the server:

**Table 3-2 HTTP methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>Requests the server to return specified resources.</td>
</tr>
<tr>
<td>PUT</td>
<td>Requests the server to update specified resources.</td>
</tr>
<tr>
<td>POST</td>
<td>Requests the server to add resources or perform special operations.</td>
</tr>
<tr>
<td>DELETE</td>
<td>Requests the server to delete specified resources, for example, an object.</td>
</tr>
<tr>
<td>HEAD</td>
<td>Same as GET except that the server must return only the response header.</td>
</tr>
<tr>
<td>PATCH</td>
<td>Requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.</td>
</tr>
</tbody>
</table>

For example, in the case of the API used to obtain a user token, the request method is POST. The request is as follows:


**Request Header**

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

**Table 3-3** describes common request header fields.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Mandatory</th>
<th>Example Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-type</td>
<td>Request body type or format. Its default value is application/json.</td>
<td>Yes</td>
<td>application/json</td>
</tr>
<tr>
<td>Content-Length</td>
<td>Length of the request body. The unit is byte.</td>
<td>This parameter is mandatory for POST and PUT requests, but must be left blank for GET requests.</td>
<td>3495</td>
</tr>
<tr>
<td>X-Project-Id</td>
<td>Project ID. This parameter is used to obtain the token for each project.</td>
<td>No</td>
<td>e9993fc787d94b6c886cbaa340f9c0f4</td>
</tr>
<tr>
<td>X-Auth-Token</td>
<td>User token, a response to the API used to obtain a user token.</td>
<td>No</td>
<td>-</td>
</tr>
<tr>
<td>X.Sdk-Date</td>
<td>Time when the request is sent. The time is in YYYYMMDD'T'HHMMSS'Z' format.</td>
<td>No</td>
<td>20190307T101459Z</td>
</tr>
<tr>
<td></td>
<td>The value is the current Greenwich Mean Time (GMT) of the system.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Mandatory</th>
<th>Example Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization</td>
<td>Signature authentication information. The value is obtained from the request signature result and is required when the AK and SK are used to encrypt the signature. Type: string Default value: none</td>
<td>No This parameter is mandatory for AK/SK authentication.</td>
<td>SDK-HMAC-SHA256 Credential=ZIRRKMT WPTQFQ1WKNKB/20150907//ec2/sdk_request, SignedHeaders=content-type;host;x-sdk-date, Signature=55741b610f3c9fa3ae40b5a8021ebf7ebc2a28a603fc62d25cb3bfe6608e1994</td>
</tr>
<tr>
<td>Host</td>
<td>Information about the requested server. The value can be obtained from the URL of the service API. The value is hostname[:port]. If the port number is not specified, the default port is used. The default port number for HTTPS is 443.</td>
<td>No This parameter is mandatory for AK/SK authentication.</td>
<td>code.test.com or code.test.com:443</td>
</tr>
</tbody>
</table>

**NOTE**

In addition to supporting token-based authentication, public cloud APIs also support authentication using access key ID/secret access key (AK/SK). During AK/SK-based authentication, an SDK is used to sign the request, and the **Authorization** (signature information) and **X-Sdk-Date** (time when the request is sent) header fields are automatically added to the request. For more details, see [API Request Signing Guide](#). For example, the API used to obtain a user token does not require authentication. Therefore, only the **Content-Type** field needs to be added to requests for calling the API. An example of such requests is as follows:

```
Content-Type: application/json
```

#### Request Body

The body of a request is often sent in a structured format as specified in the **Content-Type** header field. The request body transfers content except the request header. If the request body contains Chinese characters, these characters must be coded in UTF-8.

The request body varies between APIs. Some APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

In the case of the API used to obtain a user token, the request parameters and parameter description can be obtained from the API request. The following

---

*Face Recognition Service*

*API Reference* 3 Calling APIs

**Issue 05 (2018-05-23)**

Copyright © Huawei Technologies Co., Ltd.
provides an example request with a body included. Replace `username`, `domainname`, and `********` (login password) with the actual values. `cn-north-4` indicates the project name. To learn how to obtain your username, account name, and project ID, see [Obtaining the Username, Account Name, and Project Name](#).

---

## 3.3 Authentication

Requests for calling an API can be authenticated using either of the following methods:

- **Token-based authentication**: Requests are authenticated using a token.
- **AK/SK-based authentication**: Requests are authenticated by encrypting the request body using an AK/SK pair.

### Token-based Authentication

---

**NOTE**

The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.
A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API.

In Making an API Request, the process of calling the API used to obtain a user token is described. After a token is obtained, the X-Auth-Token header field must be added to requests to specify the token when calling other APIs. For example, if the token is ABCDEFJ...., X-Auth-Token: ABCDEFJ.... can be added to a request as follows:

Content-Type: application/json
X-Auth-Token: ABCDEFJ....

AK/SK-based Authentication

**NOTE**

AK/SK-based authentication supports API requests with a body not larger than 12 MB. For API requests with a larger body, token-based authentication is recommended.

In AK/SK-based authentication, AK/SK is used to sign requests and the signature is then added to the requests for authentication.

- AK: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.
- SK: secret access key used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

In AK/SK-based authentication, you can use an AK/SK to sign requests based on the signature algorithm or use the signing SDK to sign requests. For details about how to sign requests and use the signing SDK, see API Request Signing Guide.

**NOTICE**

The signing SDK is only used for signing requests and is different from the SDKs provided by services.

For details about how to obtain the AK/SK, see Obtaining an AK/SK.

3.4 Response

After sending a request, you will receive a response, including a status code, response header, and response body.

**Status Code**

A status code is a group of digits, ranging from 1xx to 5xx. It indicates the status of a request. For more information, see Status Codes.

For example, if status code 201 is returned for the calling of the API used to obtain a user token, the request is successful.
Response Header

Similar to a request, a response also has a header, for example, **Content-Type**.

**Table 3-4 Common response headers**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Type</td>
<td>Media type of the message body sent to a receiver&lt;br&gt;Type: string&lt;br&gt;Default value: application/json; charset=UTF-8</td>
<td>Yes</td>
</tr>
<tr>
<td>X-request-id</td>
<td>This field carries the request ID for task tracing.&lt;br&gt;Type: string.&lt;br&gt;request_id-timestamp-hostname&lt;br&gt;(The request_id is the UUID generated on the server.&lt;br&gt;timestamp indicates the current timestamp, and hostname is the name of the server that processes the current API.)&lt;br&gt;Default value: none</td>
<td>No</td>
</tr>
<tr>
<td>X-ratelimit</td>
<td>This field carries the total number of flow control requests.&lt;br&gt;Type: integer&lt;br&gt;Default value: none</td>
<td>No</td>
</tr>
<tr>
<td>X-ratelimit-used</td>
<td>This field carries the number of remaining requests.&lt;br&gt;Type: integer&lt;br&gt;Default value: none</td>
<td>No</td>
</tr>
<tr>
<td>X-ratelimit-window</td>
<td>This field carries the flow control unit.&lt;br&gt;Type: string. The unit is minute, hour, or day.&lt;br&gt;Default value: hour</td>
<td>No</td>
</tr>
</tbody>
</table>

**Figure 1** shows the response header fields for the API used to obtain a user token.

The **x-subject-token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.
Response Body

The body of a response is often returned in structured format as specified in the Content-Type header field. The response body transfers content except the response header.

The following is part of the response body for the API used to obtain a user token.

```json
{
    "token": {
        "expires_at": "2019-02-13T06:52:13.855000Z",
        "methods": [
            "password"
        ],
        "catalog": [
            {
                "endpoints": [
                    {
                        "region_id": "cn-north-4",
                        ........
                    }
                ],
                ........
            }
        ],
        ........
    }
}
```

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

```json
{
    "error_msg": "The format of message is error",
    "error_code": "AS.0001"
}
```

In the response body, error_code is an error code, and error_msg provides information about the error. For details, see Error Code.
4.1 Querying Service Status

Function

This API is used to query the service enabling status.

URI

- URI format
  GET /v1/{project_id}/subscribe
- Parameter description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>Yes</td>
<td>String</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
</tbody>
</table>

Request Message

N/A
### Response Message

**Table 4-1 Response parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_face_set_number</td>
<td>Integer</td>
<td>Maximum number of facial image libraries. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>detect_service</td>
<td>Object</td>
<td>Sub-service information. For details about the ServiceInfo structure, see ServiceInfo. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>compare_service</td>
<td>Object</td>
<td>Sub-service information. For details about the ServiceInfo structure, see ServiceInfo. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>live_detect_service</td>
<td>Object</td>
<td>Sub-service information. For details about the ServiceInfo structure, see ServiceInfo. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>search_service</td>
<td>Object</td>
<td>Sub-service information. For details about the ServiceInfo structure, see ServiceInfo. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see Error Code. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>
Samples

- Sample request
  GET https://{endpoint}/v1/{project_id}/subscribe
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYJKoZIhvcNAQcColINODCCDTQCAQExDT...

- Sample response
  ```json
  {
  "max_face_set_number": 10,
  "detect_service": {
    "subscribe_status": true,
    "create_time": "2018-09-03 08:16:48"
  },
  "compare_service": {
    "subscribe_status": true,
    "create_time": "2018-09-03 08:16:48"
  },
  "search_service": {
    "subscribe_status": true,
    "create_time": "2018-09-03 08:16:48"
  },
  "live_detect_service": {
    "subscribe_status": true,
    "create_time": "2018-09-03 08:16:48"
  }
  }
  ```

- Failed sample response
  ```json
  {
  "error_code": "FRS.0019",
  "error_msg": "The service has not been subscribed."
  }
  ```

Status Code

For details about the status code, see Status Codes.

Error Code

For details about the error code, see Error Code.

4.2 Face Detection

Function

This API is used to detect, locate, and analyze the face in an input image, and output the key facial points and attributes.

Restrictions:
- Only images in JPG, PNG, JPEG, or BMP format can be recognized.
- Use standard JSON format in the body of the application/json request.
- Do not use carriage return characters in Base64 code.
- The system does not save images of users.
- The image size must be **less than 8 MB**. If the image is too large, the latency is long and the image information volume is small. It is recommended that the image size be **less than 1 MB**.

- The image resolution must be **less than 4,096 x 2,160**. The face resolution in an image must be **greater than 80 x 80**. It is recommended that the face resolution be **greater than 120 x 120**.

- To ensure the recognition effect, facial images need to meet the following requirements:
  a. The illumination should be greater than 200 lux and there is no light reflection or shadow caused by strong light.
  b. The overall image is clear without obvious motion blur and the face in it is not blocked.
  c. The side face angle does not exceed 30°, and the tilt angle and horizontal angle do not exceed 15°. The face in an image must be a vertically placed front face.

- For details about other restrictions, see Restrictions and Limitations.

**Suggestions:**
- A larger image does not significantly improve the recognition algorithm precision but will cause a long latency. Therefore, you are advised to upload an image smaller than **1 MB**. Generally, **500 KB** is sufficient.
- It is recommended that the size of an image stored on OBS be less than **1 MB**.
- It is recommended that the face resolution in an image be greater than **120 x 120**.

**URI**

- **URI format**
  
  POST /v1/{project_id}/face-detect

- **Parameter description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>Yes</td>
<td>String</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
</tbody>
</table>
## Request Message

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>image_url</code></td>
<td>Either <code>image_url</code>, <code>image_file</code>, or <code>image_base64</code> is mandatory.</td>
<td>String</td>
<td>Image URL. Currently, only the URL of an OBS bucket on HUAWEI CLOUD is supported and Face Recognition must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see Service Authorization.</td>
</tr>
<tr>
<td><code>image_file</code></td>
<td>Either <code>image_file</code>, <code>image_url</code>, or <code>image_base64</code> is mandatory.</td>
<td>File</td>
<td>Local image file, whose size cannot exceed 8 MB. It is recommended that the image size be less than 1 MB. The request format is Multipart when a file is being loaded.</td>
</tr>
<tr>
<td><code>image_base64</code></td>
<td>Either <code>image_base64</code>, <code>image_file</code>, or <code>image_url</code> is mandatory.</td>
<td>String</td>
<td>Image data (Base64-encoded). Its requirements are as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● The image size after Base64 encoding cannot exceed 8 MB. It is recommended that the image size be less than 1 MB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● The image is in JPG, JPEG, BMP, or PNG format.</td>
</tr>
<tr>
<td><code>attributes</code></td>
<td>No</td>
<td>String</td>
<td>Whether to return the facial attribute list. Multiple attributes are separated by commas (,). Currently, the following attributes are supported:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● 0: Face posture</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● 1: Gender</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● 2: Age</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● 3: Key facial point</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● 4: Dress (hat, glasses)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● 5: Smiling face</td>
</tr>
</tbody>
</table>
**Response Message**

**Table 4-3** Response parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>faces</td>
<td>List</td>
<td>Detected face. For details about the DetectFace structure, see <a href="#">DetectFace</a>. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see <a href="#">Error Code</a>. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

**Samples**

- **Sample request (Method 1: Use a Base64-encoded image.)**
  
  POST https://{endpoint}/v1/{project_id}/face-detect
  
  Request Header:
  
  Content-Type: application/json
  
  X-Auth-Token: MIINRwYJKoZIhvNAQcColINODCCDTQCAQExDT...
  
  Request Body:
  
  ```json
  {
  "image_base64": "/9j/4AAQSkZJRgABAgEASABIAAD",
  "attributes": "0,1,2,3,4,5"
  }
  ```

- **Sample request (Method 2: Use an image file.)**
  
  POST https://{endpoint}/v1/{project_id}/face-detect
  
  Request Header:
  
  X-Auth-Token: MIINRwYJKoZIhvNAQcColINODCCDTQCAQExDT...
  
  Request Body:
  
  ```json
  {
  "image_file": File (image file)
  "attributes": 0,1,2,3,4,5
  }
  ```

- **Sample request (Method 3: Use the image URL.)**
  
  POST https://{endpoint}/v1/{project_id}/face-detect
  
  Request Header:
Request Body:
{
    "image_url": "/BucketName/ObjectName",
    "attributes": ["0,1,2,3,4,5"
}]

- Sample response
{
    "faces": [
        {
            "bounding_box": {
                "width": 174,
                "top_left_y": 37,
                "top_left_x": 22,
                "height": 174
            },
            "attributes": {
                "age": 35,
                "smile": "yes",
                "gender": "male",
                "dress": {
                    "glasses": "none",
                    "hat": "none"
                },
                "headpose": {
                    "pitch_angle": -3.8639126,
                    "roll_angle": -3.988193,
                    "yaw_angle": -1.0292832
                }
            },
            "landmark": {
                "eyebrow_contour": {
                    "point": [
                        {
                            "x": 158.78517,
                            "y": 90.02418
                        },
                        {
                            "x": 150.86162,
                            "y": 82.432076
                        },
                        {
                            "x": 131.3591,
                            "y": 80.78191
                        },
                        {
                            "x": 121.39423,
                            "y": 87.62841
                        },
                        {
                            "x": 132.25839,
                            "y": 88.44499
                        },
                        {
                            "x": 149.63844,
                            "y": 89.3718
                        },
                        {
                            "x": 59.65225,
                            "y": 99.64757
                        },
                        {
                            "x": 65.96038,
                            "y": 88.92635
                        },
                        {
                            "x": 85.72037,
                            "y": 88.92635
                        }
                    ]
                }
            }
        }
    ]
}
"eyes_contour": {
  "point": [
    {"x": 69.22003, "y": 105.52183},
    {"x": 81.599174, "y": 98.70017},
    {"x": 94.22879, "y": 103.18641},
    {"x": 82.32345, "y": 107.02255},
    {"x": 150.97876, "y": 99.29232},
    {"x": 137.91818, "y": 93.49849},
    {"x": 125.78866, "y": 100.8499},
    {"x": 138.81555, "y": 102.29028},
    {"x": 75.16096, "y": 100.01521},
    {"x": 75.6541, "y": 107.08471},
    {"x": 89.18221, "y": 106.11414},
    {"x": 88.87559, "y": 98.87617},
    {"x": 130.7761, "y": 95.49036}
  ]
}
"mouth_contour": {  
"point": [  
{  
"x": 82.56091,  
"y": 160.6078  
},  
{  
"x": 91.546906,  
"y": 154.97893  
},  
{  
"x": 101.45366,  
"y": 152.51962  
},  
{  
"x": 111.45223,  
"y": 153.75433  
},  
{  
"x": 122.52581,  
"y": 152.07089  
},  
{  
"x": 133.3687,  
"y": 152.31891  
},  
{  
"x": 142.48122,  
"y": 154.72552  
},  
{  
"x": 136.62144,  
"y": 167.81747  
},  
{  
"x": 126.79148,  
"y": 175.79927  
},  
{  
"x": 113.959526,  
"y": 179.32193  
},  
{  
"x": 99.90567,  
"y": 178.14986  
},  
{  
"x": 88.95302,  
"y": 172.11215  
},  
{  
"x": 95.30528,  
"y": 168.83522  
},  
{  
"x": 112.094025,  
"y": 170.73698  
}  
]  
}
4.3 Face Verification

Function

This API is used to compare two faces to verify whether they belong to the same person and return the confidence level. If each input image contains multiple faces, the API selects the largest face for comparison.

Restrictions:

- The total size of the two images input for face verification is less than 8 MB.
- Only images in JPG, PNG, JPEG, or BMP format can be recognized.
- Use standard JSON format in the body of the application/json request.
- Do not use carriage return characters in Base64 code.
- The system does not save images of users.
- The image size must be less than 8 MB. If the image is too large, the latency is long and the image information volume is small. It is recommended that the image size be less than 1 MB.
- The image resolution must be less than 4,096 x 2,160. The face resolution in an image must be greater than 80 x 80. It is recommended that the face resolution be greater than 120 x 120.
- To ensure the recognition effect, facial images need to meet the following requirements:
  a. The illumination should be greater than 200 lux and there is no light reflection or shadow caused by strong light.
  b. The overall image is clear without obvious motion blur and the face in it is not blocked.
  c. The side face angle does not exceed 30°, and the tilt angle and horizontal angle do not exceed 15°. The face in an image must be a vertically placed front face.

Status Code

For details about the status code, see Status Codes.

Error Code

For details about the error code, see Error Code.
For details about other restrictions, see Restrictions and Limitations.

Suggestions:
- A larger image does not significantly improve the recognition algorithm precision but will cause a long latency. Therefore, you are advised to upload an image smaller than 1 MB. Generally, 500 KB is sufficient.
- It is recommended that the size of an image stored on OBS be less than 1 MB.
- It is recommended that the face resolution in an image be greater than 120 x 120.

URI
- URI format
  POST /v1/{project_id}/face-compare
- Parameter description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
</tbody>
</table>

Request Message

Table 4-4 Request parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>image1_url</td>
<td>String</td>
<td>Either</td>
<td>Image URL. Currently, only the URL of an OBS bucket on HUAWEI CLOUD is supported and Face Recognition must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see Service Authorization.</td>
</tr>
<tr>
<td>image1_url</td>
<td>String</td>
<td>Either</td>
<td>Image URL. Currently, only the URL of an OBS bucket on HUAWEI CLOUD is supported and Face Recognition must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see Service Authorization.</td>
</tr>
<tr>
<td>image1_file</td>
<td>File</td>
<td>Either</td>
<td>Local image file, whose size cannot exceed 8 MB. It is recommended that the image size be less than 1 MB. The request format is Multipart when a file is being loaded.</td>
</tr>
<tr>
<td>image1_url</td>
<td>String</td>
<td>Either</td>
<td>Image URL. Currently, only the URL of an OBS bucket on HUAWEI CLOUD is supported and Face Recognition must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see Service Authorization.</td>
</tr>
<tr>
<td>image1_url</td>
<td>String</td>
<td>Either</td>
<td>Image URL. Currently, only the URL of an OBS bucket on HUAWEI CLOUD is supported and Face Recognition must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see Service Authorization.</td>
</tr>
<tr>
<td>image1_file</td>
<td>File</td>
<td>Either</td>
<td>Local image file, whose size cannot exceed 8 MB. It is recommended that the image size be less than 1 MB. The request format is Multipart when a file is being loaded.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Type</td>
<td>Mandatory</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------</td>
<td>----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>image1_base64</td>
<td>String</td>
<td>Either</td>
<td>Image data (Base64-encoded). Its requirements are as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>image1_base64, image1_url,</td>
<td>• The image size after Base64 encoding cannot exceed 8 MB. It is recommended that the image size be <strong>less than 1 MB</strong>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>image1_file is mandatory.</td>
<td>• The image is in JPG, JPEG, BMP, or PNG format.</td>
</tr>
<tr>
<td>image2_url</td>
<td>String</td>
<td>Either</td>
<td>Image URL. Currently, only the URL of an OBS bucket on HUAWEI CLOUD is supported and Face Recognition must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see <strong>Service Authorization</strong>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>image2_url, image2_file,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>or image2_base64 is</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>mandatory.</td>
<td></td>
</tr>
<tr>
<td>image2_file</td>
<td>File</td>
<td>Either</td>
<td>Local image file, whose size cannot exceed 8 MB. It is recommended that the image size be <strong>less than 1 MB</strong>. The request format is Multipart when a file is being loaded.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>image2_file, image2_url,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>or image2_base64 is</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>mandatory.</td>
<td></td>
</tr>
<tr>
<td>image2_base64</td>
<td>String</td>
<td>Either</td>
<td>Image data (Base64-encoded). Its requirements are as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>image2_base64, image2_url,</td>
<td>• The image size after Base64 encoding cannot exceed 8 MB. It is recommended that the image size be <strong>less than 1 MB</strong>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>image2_file or</td>
<td>• The image is in JPG, JPEG, BMP, or PNG format.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>image2_base64 is mandatory.</td>
<td></td>
</tr>
</tbody>
</table>
Response Message

Table 4-5 Response parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>image1_face</td>
<td>Object</td>
<td>Face detected in the first image. For details about the DetectFace structure, see DetectFace. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>image2_face</td>
<td>Object</td>
<td>Face detected in the second image. For details about the DetectFace structure, see DetectFace. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>similarity</td>
<td>Double</td>
<td>Face similarity. The value ranges from 0 to 1. A larger value indicates a higher similarity degree. Generally, if the value is greater than 0.93, the faces in two images belong to one person. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see Error Code. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

Samples

- Sample request (Method 1: Use a Base64-encoded image.)
  POST https://(endpoint)/v1/(project_id)/face-compare
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...
  Request Body:
  {
    "image1_base64": "i9j/4AAQSkZJRgABAgEASABIAAD",
    "image2_base64": "i9j/4AAQSkZJRgABAgEASABIAAD"
  }

- Sample request (Method 2: Use an image file.)
  POST https://(endpoint)/v1/(project_id)/face-compare
  Request Header:
  X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...
  Request Body:
Sample request (Method 3: Use the image URL.)

POST https://{endpoint}/v1/{project_id}/face-compare
Request Header:
Content-Type: application/json
X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

Request Body:
{
  "image1_url": "BucketName/ObjectName",
  "image2_url": "BucketName/ObjectName"
}

Sample response
{
  "image1_face": {
    "bounding_box": {
      "width": 174,
      "top_left_y": 37,
      "top_left_x": 22,
      "height": 174
    },
    "similarity": 0.4078676104545593,
  },
  "image2_face": {
    "bounding_box": {
      "width": 118,
      "top_left_y": 28,
      "top_left_x": 94,
      "height": 118
    }
  }
}

Failed sample response
{
  "error_code": "FRS.0501",
  "error_msg": "Detect no face, check out your picture."
}

Status Code
For details about the status code, see Status Codes.

Error Code
For details about the error code, see Error Code.

4.4 Face LiveDetect

Function
Face LiveDetect determines whether a person in a video is alive by checking whether the person's actions in the video are consistent with those in the input action list. If multiple faces appear, the largest face is selected.

Restrictions:
- Currently, only video files and Base64-encoded videos can be detected. User clients need to obtain the video streams, save them as files, and then call the LiveDetect API.
• The size of a video file **cannot exceed 8 MB**. It is recommended that the video file be compressed to **200 KB to 2 MB** on the client.
• Use standard JSON format in the body of the `application/json` request.
• Do not use carriage return characters in Base64 code.
• The system does not save videos of users.
• For details about other restrictions, see Restrictions and Limitations.

Suggestions:
• The recommended frame rate is **10 fps to 30 fps**.
• It is recommended that the video file be compressed to **200 KB to 2 MB** on the client.

**URI**

• **URI format**

```plaintext
POST /v1/{project_id}/live-detect
```

• **Parameter description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
</tbody>
</table>
## Request Message

### Table 4-6 Request parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| video_url       | Either **video_url**, **video_file**, or **video_base64** is mandatory.   | String | Video URL. Currently, only the URL of an OBS bucket on HUAWEI CLOUD is supported and Face Recognition must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see Service Authorization. The video requirements are as follows:  
  - The video size after Base64 encoding cannot exceed 8 MB.  
  - The video duration must be 1 to 15 seconds.  
  - The recommended frame rate is 10 fps to 30 fps.  
  - The encapsulation format can be MP4, AVI, FLV, WEBM, ASF, or MOV.  
  - The video encoding format can be H.261, H.263, H.264, HEVC, VC-1, VP8, VP9, or WMV3. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>video_file</td>
<td>Either <code>video_file</code>, <code>video_url</code>, or <code>video_base64</code> is mandatory.</td>
<td>File</td>
<td>Local video file. The request format is Multipart. The video requirements are as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● The size of a video file cannot exceed 8 MB. It is recommended that the video file be compressed to <strong>200 KB to 2 MB</strong> on the client.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● The video duration must be 1 to 15 seconds.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● The recommended frame rate is 10 fps to 30 fps.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● The encapsulation format can be MP4, AVI, FLV, WEBM, ASF, or MOV.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● The video encoding format can be H.261, H.263, H.264, HEVC, VC-1, VP8, VP9, or WMV3.</td>
</tr>
</tbody>
</table>
### Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| video_base64| Either video_base64, video_url, or video_file | String | Video data (Base64-encoded). Its requirements are as follows:  
- The video size after Base64 encoding cannot exceed 8 MB. It is recommended that the video file be compressed to **200 KB to 2 MB** on the client.  
- The video duration must be 1 to 15 seconds.  
- The recommended frame rate is 10 fps to 30 fps.  
- The encapsulation format can be MP4, AVI, FLV, WEBM, ASF, or MOV.  
- The video encoding format can be H.261, H.263, H.264, HEVC, VC-1, VP8, VP9, or WMV3. |
| actions     | Yes       | String | Action code sequence list. Actions are separated by commas (.). Currently, the following actions are supported:  
- 1: Shake the head to the left.  
- 2: Shake the head to the right.  
- 3: Nod the head.  
- 4: Mouth movement |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action_time</td>
<td>No</td>
<td>String</td>
<td>String of the action time array. The length of the array is the same as the number of actions. Each item contains the start time and end time of the action in the corresponding sequence. The unit is the milliseconds from the video start time.</td>
</tr>
</tbody>
</table>

**Response Message**

**Table 4-7 Response parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>video-result</td>
<td>Object</td>
<td>LiveDetect result. For details about the VideoDetectResult structure, see <a href="#">VideoDetectResult</a>. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>warning-list</td>
<td>Object</td>
<td>Warning information list. For details about the WarningList structure, see <a href="#">WarningList</a>. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see <a href="#">Error Code</a>. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

## Samples

- **Sample request (Method 1: Use a Base64-encoded video.)**
  
  POST https://{endpoint}/v1/{project_id}/live-detect
  
  Request Header:
  
  Content-Type: application/json
  
  X-Auth-Token: MIINRwYJKoZl/hvcNAQcCOLIwDCTQCAQExDT...
  
  Request Body:
  
  ```
  {
      "video_base64":"/9j/4AAQSkZJRgABAgEASABIAAD",
      "actions":"1,3,2",
      "action_time":"1000-3000,4000-7000,9000-12000"
  }
  ```

- **Sample request (Method 2: Use a video file.)**
  
  POST https://{endpoint}/v1/{project_id}/live-detect
  
  Request Header:
  
  X-Auth-Token: MIINRwYJKoZl/hvcNAQcCOLIwDCTQCAQExDT...
  
  Request Body:
  
  ```
  video_file: File (Video file)
  actions: 1,3,2
  action_time: 1000-3000,4000-7000,9000-12000
  ```

- **Sample request (Method 3: Use the video URL.)**
  
  POST https://{endpoint}/v1/{project_id}/live-detect
  
  Request Header:
  
  Content-Type: application/json
  
  X-Auth-Token: MIINRwYJKoZl/hvcNAQcCOLIwDCTQCAQExDT...
  
  Request Body:
  
  ```
  {
      "video_url":"/BucketName/ObjectName",
      "actions":"1,3,2",
      "action_time":"1000-3000,4000-7000,9000-12000"
  }
  ```

- **Sample response**
  
  ```
  "video-result": {
    "alive": true,
    "actions": [{
      "action": 1,
      "confidence": 0.823
    },
    {
      "action": 3,
      "confidence": 0.823
    },
    {
      "action": 2,
      "confidence": 0.823
    }],
    "picture": "/9j/4AAQSkZJRgABAQABAQABAQAD/2w..."
  },
  "warning-list": []
  ```
4.5 Face Retrieval

Function

This API is used to search an existing facial image library for one or more faces similar to the input face, and return corresponding confidence levels.

You can input an image or face ID to retrieve faces. If multiple facial images are input, the largest face detected in the images is used for retrieval.

Restrictions:

- Only images in JPG, PNG, JPEG, or BMP format can be recognized.
- Use standard JSON format in the body of the `application/json` request.
- Do not use carriage return characters in Base64 code.
- The system does not save images of users.
- The image size must be less than **8 MB**. If the image is too large, the latency is long and the image information volume is small. It is recommended that the image size be less than **1 MB**.
- The image resolution must be less than **4,096 x 2,160**. The face resolution in an image must be greater than **80 x 80**. It is recommended that the face resolution be greater than **120 x 120**.
- To ensure the recognition effect, facial images need to meet the following requirements:
  a. The illumination should be greater than 200 lux and there is no light reflection or shadow caused by strong light.
  b. The overall image is clear without obvious motion blur and the face in it is not blocked.
  c. The side face angle does not exceed 30°, and the tilt angle and horizontal angle do not exceed 15°. The face in an image must be a vertically placed front face.
- For details about other restrictions, see Restrictions and Limitations.

Suggestions:

- A larger image does not significantly improve the recognition algorithm precision but will cause a long latency. Therefore, you are advised to upload an image smaller than **1 MB**. Generally, **500 KB** is sufficient.
It is recommended that the size of an image stored on OBS be less than 1 MB.
It is recommended that the face resolution in an image be greater than 120 x 120.

**URI**

- **URI format**
  ```
  POST /v1/{project_id}/face-sets/{face_set_name}/search
  ```
- **Parameter description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the facial image library</td>
</tr>
</tbody>
</table>

**Request Message**

**Table 4-8 Request parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>image_url</td>
<td>String</td>
<td>Either</td>
<td>Image URL. Currently, only the URL of an OBS bucket on HUAWEI CLOUD is supported and Face Recognition must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see Service Authorization.</td>
</tr>
<tr>
<td>face_id</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>image_file</td>
<td>File</td>
<td>Either</td>
<td>Local image file, whose size cannot exceed 8 MB. It is recommended that the image size be less than 1 MB. The request format is Multipart when a file is being loaded.</td>
</tr>
<tr>
<td>face_id</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Type</td>
<td>Mandatory</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>----------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>image_base64</td>
<td>String</td>
<td>Either</td>
<td>Image data (Base64-encoded). Its requirements are as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>image_base</td>
<td>● The image size after Base64 encoding cannot exceed 8 MB. It is recommended that the image size be <strong>less than 1 MB</strong>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>64, image_url, image_file, face_id</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>is mandatory.</td>
<td></td>
</tr>
<tr>
<td>face_id</td>
<td>String</td>
<td>Either</td>
<td>Face ID returned by the system after a face is imported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>face_id, image_url, image_file, or image_base64 is mandatory.</td>
<td></td>
</tr>
<tr>
<td>top_n</td>
<td>Integer</td>
<td>No</td>
<td>N faces returned that are most similar to the input one. The default value of N is <strong>10</strong>.</td>
</tr>
<tr>
<td>threshold</td>
<td>Double</td>
<td>No</td>
<td>Face similarity threshold. If the similarity degree of a face is lower than the threshold, the face is not returned. The value ranges from 0 to 1. The recommended value is <strong>0.93</strong>. The default value is 0.</td>
</tr>
<tr>
<td>sort</td>
<td>JsonArray</td>
<td>No</td>
<td>Field sorting. For details, see <strong>Sort Syntax</strong>.</td>
</tr>
<tr>
<td>filter</td>
<td>String</td>
<td>No</td>
<td>Filtering criteria. For details, see <strong>Filter Syntax</strong>.</td>
</tr>
<tr>
<td>return_fields</td>
<td>String</td>
<td>No</td>
<td>Returned customized field.</td>
</tr>
</tbody>
</table>
Response Message

Table 4-9 Response parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>faces</td>
<td>List</td>
<td>Face set to be retrieved. For details, see SearchFace. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see Error Code. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

Samples

- **Sample request (Method 1: Use a Base64-encoded image.)**
  
  POST `https://(endpoint)/v1/(project_id)/face-sets/showFaceSet/search`

  Request Header:
  Content-Type: `application/json`
  X-Auth-Token: MIINRwYJKoZIhvNAQcColINODCCDTQCAQExDT...

  Request Body:
  ```
  {
    "image_base64": "/9j/4AAQSkZJRgABAgEASABIAAD",
    "sort": [
      {
        "timestamp": "asc"
      }
    ],
    "return_fields": ["timestamp", "id"],
    "filter": "timestamp:12"
  }
  ```

- **Sample request (Method 2: Use an image file.)**
  
  POST `https://(endpoint)/v1/(project_id)/face-sets/showFaceSet/search`

  Request Header:
  X-Auth-Token: MIINRwYJKoZIhvNAQcColINODCCDTQCAQExDT...

  Request Body:
  ```
  image_file: File (image file)
  return_fields: ["timestamp","id"]
  filter: timestamp:12
  ```
• Sample request (Method 3: Use the image URL.)

POST https://{endpoint}/v1/{project_id}/face-sets/showFaceSet/search
Request Header:
Content-Type: application/json
X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQEExDT...

Request Body:
{
"image_url":"/BucketName/ObjectName",
"sort" : [

  { "timestamp" : "asc"
  }],
"return_fields" : ["timestamp", "id"],
"filter" : "timestamp:12"
}

• Sample request (Method 4: Use the face ID.)

POST https://{endpoint}/v1/{project_id}/face-sets/showFaceSet/search
Request Header:
Content-Type: application/json
X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQEExDT...

Request Body:
{
  "face_id":"6KLB1Ktu",
  "sort" : [

  { "timestamp" : "asc"
  }],
  "return_fields" : ["timestamp", "id"],
  "filter" : "timestamp:12"
}

• Sample response

{
  "faces": [

  { "bounding_box": {
    "width" : 170,
    "top_left_y" : 37,
    "top_left_x" : 20,
    "height" : 170
  },
  "similarity" : 0.996146,
  "external_image_id" : "123",
  "external_fields" : {
    "id" : "home",
    "timestamp" : 12
  },
  "face_id" : "6KLB1Ktu"
  },

  { "bounding_box": {
    "width" : 170,
    "top_left_y" : 37,
    "top_left_x" : 20,
    "height" : 170
  },
  "similarity" : 0.996146,
  "external_image_id" : "12",
  "external_fields" : {
    "id" : "home1",
    "timestamp" : 12
  },
  "face_id" : "PexOpqRj"
  }
}
4.6 Facial Image Library Management

4.6.1 Creating a Facial Image Library

Function

This API is used to create a facial image library for storing facial features. You can create a maximum of 10 facial image libraries. Each library can contain a maximum of 100,000 facial features. If you need higher specifications, contact the customer service personnel.

**NOTE**

Use standard JSON format in the body of the `application/json` request.

URI

- **URI format**
  
  `POST /v1/{project_id}/face-sets`

- **Parameter description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
</tbody>
</table>
Request Message

Table 4-10 Request parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the facial image library. Do not start the name of a facial image library with an underscore (_). Otherwise, the Cloud Eye service cannot collect the number of faces.</td>
</tr>
<tr>
<td>face_set_capacity</td>
<td>Integer</td>
<td>No</td>
<td>Maximum capacity of a facial image library. The value is an integer multiple of 10,000, for example, 30,000. The default value is 100,000, and the maximum value is 100,000.</td>
</tr>
<tr>
<td>external_fields</td>
<td>Json</td>
<td>No</td>
<td>User-defined data. User-defined fields cannot be named in the following system fields: vector, bounding_box, external_image_id, face_id, create_time, _id, _all, and _source. Duplication of the JSON character string is not verified. The key value of a user-defined field contains 1 to 36 characters, and the value of the string type contains 1 to 256 characters. For details, see Customized Fields.</td>
</tr>
</tbody>
</table>

Response Message

- Parameter description
### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_info</td>
<td>Object</td>
<td>Facial image library information. For details, see <a href="#">FaceSetInfo</a>. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see <a href="#">Error Code</a>. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

### Samples

- **Sample request**
  
  ```
  POST https://{endpoint}/v1/{project_id}/face-sets
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQEExDT...
  
  Request Body:
  ```
  ```json
  {
    "face_set_name": "test",
    "face_set_capacity": 100000,
    "external_fields": {
      "timestamp": {
        "type": "long"
      },
      "id": {
        "type": "string"
      },
      "number": {
        "type": "integer"
      }
    }
  }
  ```

- **Sample response**
  ```json
  {
    "face_set_info": {
      "face_number": 0,
      "face_set_id": "WYYOFIGb",
      "face_set_name": "test",
      "create_date": "2018-05-28 02:19:00",
      "face_set_capacity": 10000,
      "external_fields": {
        "timestamp": 1527232660000,
        "id": "00000000-0000-0000-0000-000000000000",
        "number": 0
      }
    }
  }
  ```
4.6.2 Querying All Facial Image Libraries

Function

This API is used to query the statuses of all facial image libraries of the current user.

URI

- URI format
  
  GET /v1/{project_id}/face-sets

- Parameter description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
</tbody>
</table>

Request Message

NA
### Response Message

#### Table 4-11 Response parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_sets_info</td>
<td>List</td>
<td>Information about all facial image libraries. For details, see FaceSetInfo. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see Error Code. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

### Samples

- **Sample request**
  
  Get `https://{endpoint}/v1/{project_id}/face-sets`

  Request Header:
  
  Content-Type: application/json
  
  X-Auth-Token: MIINRwJyKzIhvCNQcCoIINODCCDTQCAQEaDT...

- **Sample response**
  
  ```json
  {
    "face_sets_info": [
      {
        "face_number": 0,
        "face_set_id": "y0XMMZTO",
        "face_set_name": "test",
        "create_date": "2018-05-11 07:49:40",
        "face_set_capacity": 10000,
        "external_fields": {
          "number": {
            "type": "integer"
          },
          "id": {
            "type": "string"
          },
          "timestamp": {
            "type": "long"
          }
        }
      }
    ]
  }
  ```
4.6.3 Querying a Facial Image Library

**Function**

This API is used to query the status of a facial image library.

**URI**

- **URI format**
  
  GET /v1/{project_id}/face-sets/{face_set_name}

- **Parameter description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the facial image library.</td>
</tr>
</tbody>
</table>

**Request Message**

NA
## Response Message

### Table 4-12 Response parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_info</td>
<td>Object</td>
<td>Facial image library information. For details, see <a href="#">FaceSetInfo</a>. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see <a href="#">Error Code</a>. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

### Samples

- **Sample request**
  
  Get https://{endpoint}/v1/{project_id}/face-sets/showFaceSet
  
  Request Header:
  
  Content-Type: application/json
  
  X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

- **Sample response**

  ```json
  
  {  
    "face_set_info": {  
      "face_number": 94,  
      "face_set_id": "T785tx1N",  
      "face_set_name": "showFaceSet",  
      "create_date": "2018-05-10 01:44:39",  
      "face_set_capacity": 10000,  
      "external_fields": {  
        "number": {  
          "type": "integer"  
        },  
        "id": {  
          "type": "string"  
        },  
        "timestamp": {  
          "type": "long"  
        }  
      }  
    }  
  }
  ```
4.6.4 Deleting a Facial Image Library

Function

This API is used to delete a facial image library and all faces in the library.

URI

- URI format

  DELETE /v1/{project_id}/face-sets/{face_set_name}

- Parameter description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the facial image library.</td>
</tr>
</tbody>
</table>

Request Message

NA

Response Message

Table 4-13 Response parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Name of the facial image library. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see Error Code. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

### Samples

- **Sample request**
  ```
  DELETE https://{endpoint}/v1/{project_id}/face-sets/showFaceSet
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...
  ```

- **Sample response**
  ```
  {  
  "face_set_name": "showFaceSet"
  }
  ```

- **Failed sample response**
  ```
  {  
  "error_code": "FRS.0002",
  "error_msg": "The authentication token is abnormal."
  }
  ```

### Status Code

For details about the status code, see [Status Codes](#).

### Error Code

For details about the error code, see [Error Code](#).

### 4.7 Facial Resource Management

#### 4.7.1 Adding a Face

**Function**

This API is used to add faces to a facial image library. All detected faces will be added to the library.

**Restrictions:**
- Only images in JPG, PNG, JPEG, or BMP format can be recognized.
- Use standard JSON format in the body of the `application/json` request.
- Do not use carriage return characters in Base64 code.
- The system does not save images of users.
- The image size must be **less than 8 MB**. If the image is too large, the latency is long and the image information volume is small. It is recommended that the image size be **less than 1 MB**.
- The image resolution must be **less than 4,096 x 2,160**. The face resolution in an image must be **greater than 80 x 80**. It is recommended that the face resolution be **greater than 120 x 120**.
- To ensure the recognition effect, facial images need to meet the following requirements:
  a. The illumination should be greater than 200 lux and there is no light reflection or shadow caused by strong light.
  b. The overall image is clear without obvious motion blur and the face in it is not blocked.
  c. The side face angle does not exceed 30°, and the tilt angle and horizontal angle do not exceed 15°. The face in an image must be a vertically placed front face.
- For details about other restrictions, see Restrictions and Limitations.

Suggestions:
- A larger image does not significantly improve the recognition algorithm precision but will cause a long latency. Therefore, you are advised to upload an image smaller than **1 MB**. Generally, **500 KB** is sufficient.
- It is recommended that the size of an image stored on OBS be less than **1 MB**.
- It is recommended that the face resolution in an image be greater than **120 x 120**.

**URI**

- **URI format**
  ```
  POST /v1/{project_id}/face-sets/{face_set_name}/faces
  ```
  
- **Parameter description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the facial image library.</td>
</tr>
</tbody>
</table>
### Request Message

#### Table 4-14 Request parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>image_url</td>
<td>String</td>
<td>Either</td>
<td>Image URL. Currently, only the URL of an OBS bucket on HUAWEI CLOUD is supported and Face Recognition must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see Service Authorization.</td>
</tr>
<tr>
<td>image_file</td>
<td>File</td>
<td>Either</td>
<td>Local image file, whose size cannot exceed 8 MB. It is recommended that the image size be less than 1 MB. The request format is Multipart when a file is being loaded.</td>
</tr>
<tr>
<td>image_base64</td>
<td>String</td>
<td>Either</td>
<td>Image data (Base64-encoded). Its requirements are as follows: - The image size after Base64 encoding cannot exceed 8 MB. It is recommended that the image size be less than 1 MB. - The image is in JPG, JPEG, BMP, or PNG format.</td>
</tr>
<tr>
<td>external_image_id</td>
<td>String</td>
<td>No</td>
<td>External image ID specified by the user. It is bound to the current image. If the user does not provide one, it is generated by the system. The ID contains 1 to 36 characters, including letters, digits, hyphens (-), and underscores (_). Other special characters are not allowed.</td>
</tr>
<tr>
<td>external_fields</td>
<td>Json</td>
<td>No</td>
<td>Enters a value based on the customized data type. This field is defined when a face set is created. Duplication of the JSON character string is not verified. For details, see Customized Fields.</td>
</tr>
</tbody>
</table>
## Response Message

### Table 4-15 Response parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Name of the facial image library. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>face_set_id</td>
<td>String</td>
<td>ID of the facial image library. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>faces</td>
<td>List</td>
<td>Face structure in the facial image library. For details, see FaceSetFace. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see Error Code. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

### Samples

- Sample request (Method 1: Use a Base64-encoded image.)

```plaintext
POST https://(endpoint)/v1/(project_id)/face-sets/showFaceSet/faces
Request Header:
Content-Type: application/json
X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

Request Body:
{
  "image_base64": "/9j/4AAQSkZJRgABAgEASABIAAD",
  "external_image_id": "imageID",
  "external_fields": {
    "timestamp": 12,
  }
}```
● Sample request (Method 2: Use an image file.)
POST https://{endpoint}/v1/{project_id}/face-sets/showFaceSet/faces
Request Header:
X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

Request Body:
image_file: File (image file)
external_image_id: imageID
external_fields: 
  "timestamp": 12,"id": "home"

● Sample request (Method 3: Use the image URL.)
POST https://{endpoint}/v1/{project_id}/face-sets/showFaceSet/faces
Request Header:
Content-Type: application/json
X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

Request Body:
{
  "image_url":"/BucketName/ObjectName",
  "external_image_id":imageID,
  "external_fields": {
    "timestamp": 12,
    "id": "home"
  }
}

● Sample response
{
  "face_set_id": "T785tx1N",
  "face_set_name": "showFaceSet",
  "faces": [
    {
      "bounding_box": {
        "width": 63,
        "top_left_y": 100,
        "top_left_x": 221,
        "height": 63
      },
      "external_image_id": "Xr0phyap",
      "external_fields": {
        "timestamp": 12,
        "id": "home"
      },
      "face_id": "JLa9hYLI"
    }
  ]
}

● Failed sample response
{
  "error_code": "FRS.0404",
  "error_msg": "Detect no face, can not add it to face set."
}

Status Code

For details about the status code, see Error Code.

Error Code

For details about the error code, see Error Code.
4.7.2 Querying a Face

**Function**

This API is used to query the face information in a specified facial image library.

**URI**

- **URI format**
  
  GET /v1/{project_id}/face-sets/{face_set_name}/faces?offset=xxx&limit=xxx
  
  or
  
  GET /v1/{project_id}/face-sets/{face_set_name}/faces?face_id={face_id}

- **Parameter description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see <a href="#">Obtaining the Project ID, Account Name, and AK/SK</a>.</td>
</tr>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the facial image library</td>
</tr>
<tr>
<td>offset</td>
<td>Integer</td>
<td>Either</td>
<td>Start number of the data records to be read. The default value is 0.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>offset or face_id is mandatory.</td>
<td></td>
</tr>
<tr>
<td>limit</td>
<td>Integer</td>
<td>Either</td>
<td>Number of records to be read. The default value is 5.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>limit or face_id is mandatory.</td>
<td></td>
</tr>
<tr>
<td>face_id</td>
<td>String</td>
<td>Either</td>
<td>Face ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>face_id or offset is mandatory.</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE**

The sum of values of offset and limit cannot exceed 10,000.

**Request Message**

NA
Response Message

Table 4-16 Response parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Name of the facial image library. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>face_set_id</td>
<td>String</td>
<td>ID of the facial image library. It is a randomly generated string containing eight characters. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>faces</td>
<td>List</td>
<td>Returned face information. For details, see FaceSetFace. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see Error Code. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

Samples

- Sample request
  Get https://(endpoint)/v1/(project_id)/face-sets/showFaceSet/faces?offset=0&limit=1
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYJKoZIhvcNAQcColINODCCDTQCAQEoDT...

- Sample response
  
  ```json
  {
    "face_set_id": "T785tx1N",
  }
  ```
"face_set_name": "showFaceSet",
"faces": [
{
"bounding_box": {
"width": 63,
"top_left_y": 100,
"top_left_x": 221,
"height": 63
},
"external_image_id": "aIzRAa58",
"face_id": "cFydu4d2",
"external_fields": {
"number": 122,
"id": "home",
"timestamp": 12
}
}
]

- Failed sample response
{
"error_code": "FRS.0002",
"error_msg": "The authentication token is abnormal."
}

**Status Code**

For details about the status code, see [Status Codes](#).

**Error Code**

For details about the error code, see [Error Code](#).

### 4.7.3 Updating a Face

**Function**

This API is used to update a single face based on its face ID (face_id).

**NOTE**

Use standard JSON format in the body of the `application/json` request.

**URI**

- **URI format**
  
  PUT `/v1/{project_id}/face-sets/{face_set_name}/faces`

- **Parameter description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see <a href="#">Obtaining the Project ID, Account Name, and AK/SK</a></td>
</tr>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the facial image library</td>
</tr>
</tbody>
</table>
Request Message

**Table 4-17 Request parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_id</td>
<td>String</td>
<td>Yes</td>
<td>Face ID, the unique ID generated by the system</td>
</tr>
<tr>
<td>external_image_id</td>
<td>String</td>
<td>No</td>
<td>External image ID specified by the user. It is bound to the current image. If the user does not provide one, it is generated by the system. The ID contains 1 to 36 characters, including letters, digits, hyphens (-), and underscores (_). Other special characters are not allowed. Either parameter <code>external_image_id</code> or <code>external_fields</code> is modified.</td>
</tr>
<tr>
<td>external_fields</td>
<td>Json</td>
<td>No</td>
<td>Duplication of the JSON character string is not verified. The key value of a user-defined field contains 1 to 36 characters, and the value of the string type contains 1 to 256 characters. For details, see Customized Fields. Either parameter <code>external_image_id</code> or <code>external_fields</code> is modified.</td>
</tr>
</tbody>
</table>

Response Message

**Table 4-18 Response parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Name of the facial image library</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>face_set_id</td>
<td>String</td>
<td>ID of the facial image library.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>face_number</td>
<td>Integer</td>
<td>Number of updated faces. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see Error Code. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

**Samples**

- **Sample request**
  
  PUT https://{endpoint}/v1/{project_id}/face-sets/showFaceSet/faces
  
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

  Request Body:
  
  ```
  {
    "face_id": "iexEBb6t",
    "external_image_id": "imageID",
    "external_fields": {
      "timestamp": 12,
      "id": "300018629384756"
    }
  }
  ```

- **Sample response**

  ```
  {
    "face_number": 1,
    "face_set_id": "T78Stx1N",
    "face_set_name": "showFaceSet"
  }
  ```

- **Failed sample response**

  ```
  {
    "error_code": "FRS.0303",
    "error_msg": "The face id is not exist, checkout your input."
  }
  ```

**Status Code**

For details about the status code, see Status Codes.
Error Code

For details about the error code, see Error Code.

4.7.4 Deleting a Face

Function

This API is used to delete a face from the facial image library based on the specified field.

URI

- **URI format**
  
  DELETE /v1/{project_id}/face-sets/{face_set_name}/faces?field_name=field_value

- **Parameter description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the facial image library</td>
</tr>
<tr>
<td>field_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the field deleted based on conditions. Fixed fields (external_image_id and face_id) and customized fields (excluding null strings and values) are supported. For more information about customized fields, see Customized Fields.</td>
</tr>
</tbody>
</table>

Request Message

N/A

Response Message

Table 4-19 Response parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Name of the facial image library</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>face_set_id</td>
<td>String</td>
<td>ID of the facial image library. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>face_number</td>
<td>Integer</td>
<td>Number of deleted faces. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see Error Code. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

**Samples**

- **Sample request (Method 1: Use external_image_id to delete a facial image.)**
  
  ```
  DELETE https://{endpoint}/v1/{project_id}/face-sets/showFaceSet/faces?external_image_id=imageID
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYIKoZlhvcNAQcColINODCDTQCAQExDT...
  ```

- **Sample request (Method 2: Use face_id to delete a facial image.)**
  
  ```
  DELETE https://{endpoint}/v1/{project_id}/face-sets/showFaceSet/faces?face_id=faceID
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYIKoZlhvcNAQcColINODCDTQCAQExDT...
  ```

- **Sample request (Method 3: Use a customized field to delete a facial image.)**
  
  ```
  DELETE https://{endpoint}/v1/{project_id}/face-sets/showFaceSet/faces?id=home
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYIKoZlhvcNAQcColINODCDTQCAQExDT...
  ```

- **Sample response**
  ```
  { "face_number": 1,
    "face_set_id": "T785tx1N",
    "face_set_name": "showFaceSet"
  }
  ```

- **Failed sample response**
  ```
  { "error_code": "FRS.0402",
    "error_msg": "External id is not exist, can not delete face"
  }
  ```
4.7.5 Batch Deleting Faces

Function

This API is used to batch delete multiple faces that meet specified criteria you customize.

NOTE

Use standard JSON format in the body of the application/json request.

URI

- URI format
  DELETE /v1/{project_id}/face-sets/{face_set_name}/faces/batch
- Parameter description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the facial image library</td>
</tr>
</tbody>
</table>

Request Message

Table 4-20 Request parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>String</td>
<td>Yes</td>
<td>Filtering criteria. For details, see Filter Syntax.</td>
</tr>
</tbody>
</table>

Response Message

Table 4-21 Response parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Name of the facial image library This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>face_set_id</td>
<td>String</td>
<td>ID of the facial image library. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>face_number</td>
<td>Integer</td>
<td>Number of deleted faces. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see Error Code.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

### Samples

- **Sample request (Use filter to batch delete faces.)**
  
  DELETE https://{endpoint}/v1/{project_id}/face-sets/showFaceSet/faces/batch

  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQEExDT...

  Request Body:
  ```
  {
    "filter": "age:[20 TO 30]"
  }
  ```

- **Sample response**
  ```
  {
    "face_number": 1,
    "face_set_id": "T785tx1N",
    "face_set_name": "showFaceSet"
  }
  ```

- **Failed sample response**
  ```
  {
    "error_code": "FRS.0407",
    "error_msg": "All the data not suitable, no data to be deleted."
  }
  ```

### Status Code

For details about the status code, see [Status Codes](#).
Error Code

For details about the error code, see Error Code.
5.1 Face Detection

Function

This API is used to detect, locate, and analyze the face in an input image, and output the key facial points and attributes.

Restrictions:

- Only images in JPG, PNG, JPEG, or BMP format can be recognized.
- Use standard JSON format in the body of the `application/json` request.
- Do not use carriage return characters in Base64 code.
- The system does not save images of users.
- The image size must be **less than 8 MB**. If the image is too large, the latency is long and the image information volume is small. It is recommended that the image size be **less than 1 MB**.
- The image resolution must be **less than 4,096 x 2,160**. The face resolution in an image must be **greater than 80 x 80**. It is recommended that the face resolution be **greater than 120 x 120**.
- To ensure the recognition effect, facial images need to meet the following requirements:
  a. The illumination should be greater than 200 lux and there is no light reflection or shadow caused by strong light.
  b. The overall image is clear without obvious motion blur and the face in it is not blocked.
  c. The side face angle does not exceed 30°, and the tilt angle and horizontal angle do not exceed 15°. The face in an image must be a vertically placed front face.
- For details about other restrictions, see Restrictions and Limitations.

Suggestions:

- A larger image does not significantly improve the recognition algorithm precision but will cause a long latency. Therefore, you are advised to upload an image smaller than **1 MB**. Generally, **500 KB** is sufficient.
● It is recommended that the size of an image stored on OBS be less than 1 MB.
● It is recommended that the face resolution in an image be greater than 120 x 120.

**URI**

- **URI format**
  POST /v2/{project_id}/face-detect

- **Parameter description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>Yes</td>
<td>String</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
</tbody>
</table>

**Request Message**

*Table 5-1 Request parameters*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>image_url</td>
<td>Either image_url, image_file, or image_base64 is mandatory</td>
<td>String</td>
<td>Image URL. Currently, only the URL of an OBS bucket on HUAWEI CLOUD is supported and Face Recognition must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see Service Authorization.</td>
</tr>
<tr>
<td>image_file</td>
<td>Either image_file, image_url, or image_base64 is mandatory</td>
<td>File</td>
<td>Local image file, whose size cannot exceed 8 MB. It is recommended that the image size be less than 1 MB. The request format is Multipart.</td>
</tr>
</tbody>
</table>
### Response Message

#### Table 5-2 Response parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>faces</td>
<td>List</td>
<td>Detected face. For details about the DetectFace structure, see [DetectFace]. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see [Error Code]. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td><strong>Parameter</strong></td>
<td><strong>Type</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>---------------</td>
<td>----------</td>
<td>-----------------</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

**Samples**

- **Sample request (Method 1: Use a Base64-encoded image.)**
  POST https://(endpoint)/v2/(project_id)/face-detect
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...
  Request Body:
  ```json
  {
    "image_base64": "/9j/4AAQSkZJRgABAgEASABIAAD",
    "attributes": "1,2"
  }
  ```

- **Sample request (Method 2: Use an image file.)**
  POST https://(endpoint)/v2/(project_id)/face-detect
  Request Header:
  X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...
  Request Body:
  ```json
  image_file: File (image file)
  attributes: "1,2"
  ```

- **Sample request (Method 3: Use the image URL.)**
  POST https://(endpoint)/v2/(project_id)/face-detect
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...
  Request Body:
  ```json
  {
    "image_url": "/BucketName/ObjectName",
    "attributes": "1,2"
  }
  ```

- **Sample response**
  ```json
  {
    "faces": [
      {
        "bounding_box": {
          "width": 174,
          "top_left_y": 37,
          "top_left_x": 22,
          "height": 174
        },
        "attributes": {
          "age": 35,
          "gender": "male"
        }
      }
    ]
  }
  ```

- **Failed sample response**
  ```json
  {
    "error_code": "FRS.0019",
  }
  ```
5.2 Face Verification

Function

This API is used to compare two faces to verify whether they belong to the same person and return the confidence level. If each input image contains multiple faces, the API selects the largest face for comparison.

Restrictions:

- The total size of the two images input for face verification is less than 8 MB.
- Only images in JPG, PNG, JPEG, or BMP format can be recognized.
- Use standard JSON format in the body of the application/json request.
- Do not use carriage return characters in Base64 code.
- The system does not save images of users.
- The image size must be **less than 8 MB**. If the image is too large, the latency is long and the image information volume is small. It is recommended that the image size be **less than 1 MB**.
- The image resolution must be **less than 4,096 x 2,160**. The face resolution in an image must be **greater than 80 x 80**. It is recommended that the face resolution be greater than 120 x 120.
- To ensure the recognition effect, facial images need to meet the following requirements:
  a. The illumination should be greater than 200 lux and there is no light reflection or shadow caused by strong light.
  b. The overall image is clear without obvious motion blur and the face in it is not blocked.
  c. The side face angle does not exceed 30°, and the tilt angle and horizontal angle do not exceed 15°. The face in an image must be a vertically placed front face.
- For details about other restrictions, see Restrictions and Limitations.

Suggestions:

- A larger image does not significantly improve the recognition algorithm precision but will cause a long latency. Therefore, you are advised to upload an image smaller than **1 MB**. Generally, **500 KB** is sufficient.
- It is recommended that the size of an image stored on OBS be less than **1 MB**.
- It is recommended that the face resolution in an image be greater than **120 x 120**.

**URI**

- URI format
  
  ```
  POST /v2/{project_id}/face-compare
  ```

- Parameter description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
</tbody>
</table>

**Request Message**

Table 5-3 Request parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>image1_url</td>
<td>String</td>
<td></td>
<td>Image URL. Currently, only the URL of an OBS bucket on HUAWEI CLOUD is supported and Face Recognition must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see Service Authorization.</td>
</tr>
<tr>
<td>image1_file</td>
<td>File</td>
<td></td>
<td>Local image file, whose size cannot exceed 8 MB. It is recommended that the image size be <strong>less than 1 MB</strong>. The request format is Multipart.</td>
</tr>
</tbody>
</table>
| image1_base64| String   |            | Image data (Base64-encoded). Its requirements are as follows:

  - The image size after Base64 encoding cannot exceed 8 MB. It is recommended that the image size be **less than 1 MB**.
  - The image is in JPG, JPEG, BMP, or PNG format. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>image2_url</td>
<td>String</td>
<td>Either image2_url, image2_file, or image2_base64 is mandatory.</td>
<td>Image URL. Currently, only the URL of an OBS bucket on HUAWEI CLOUD is supported and Face Recognition must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see Service Authorization.</td>
</tr>
<tr>
<td>image2_file</td>
<td>File</td>
<td>Either image2_file, image2_url, or image2_base64 is mandatory.</td>
<td>Local image file, whose size cannot exceed 8 MB. It is recommended that the image size be less than 1 MB. The request format is Multipart.</td>
</tr>
</tbody>
</table>
| image2_base64 | String | Either image2_base64, image2_url, or image2_file is mandatory. | Image data (Base64-encoded). Its requirements are as follows:  
  - The image size after Base64 encoding cannot exceed 8 MB. It is recommended that the image size be less than 1 MB.  
  - The image is in JPG, JPEG, BMP, or PNG format.                                                                                             |

**Response Message**

*Table 5-4 Response parameters*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>image1_face</td>
<td>Object</td>
<td>Face detected in the first image. For details about the DetectFace structure, see DetectFace. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>image2_face</td>
<td>Object</td>
<td>Face detected in the second image. For details about the DetectFace structure, see DetectFace. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>--------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>similarity</td>
<td>Double</td>
<td>Face similarity. The value ranges from 0 to 1. A larger value indicates a higher similarity degree. Generally, if the value is greater than 0.93, the faces in two images belong to one person. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see Error Code. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

**Samples**

- **Sample request (Method 1: Use a Base64-encoded image.)**
  
  POST https://{endpoint}/v2/{project_id}/face-compare
  Request Header: 
  Content-Type: application/json
  X-Auth-Token: MIINRwYJKoZIvctNAQcCollNODCCDTQCAQExDT...

  Request Body:
  ```json
  {
    "image1_base64":/9j/4AAQSkZJRgABAgEASABIAAD,
    "image2_base64":/9j/4AAQSkZJRgABAgEASABIAAD
  }
  ```

- **Sample request (Method 2: Use an image file.)**
  
  POST https://{endpoint}/v2/{project_id}/face-compare
  Request Header: 
  X-Auth-Token: MIINRwYJKoZIvctNAQcCollNODCCDTQCAQExDT...

  Request Body:
  ```json
  image1_file: File (image file)
  image2_file: File (image file)
  ```

- **Sample request (Method 3: Use the image URL.)**
  
  POST https://{endpoint}/v2/{project_id}/face-compare
  Request Header: 
  Content-Type: application/json
  X-Auth-Token: MIINRwYJKoZIvctNAQcCollNODCCDTQCAQExDT...

  Request Body:
  ```json
  {
    "image1_url":"BucketName/ObjectName",
    "image2_url":"BucketName/ObjectName"
  }
  ```

- **Sample response**
  ```json
  {
    "image1_face": {
      "bounding_box": {
        "width": 174,
        ...}
  ```
5.3 Face Retrieval

Function

This API is used to search an existing facial image library for one or more faces similar to the input face, and return corresponding confidence levels.

You can input an image or face ID to retrieve faces. If multiple facial images are input, the largest face detected in the images is used for retrieval.

Restrictions:

- Only images in JPG, PNG, JPEG, or BMP format can be recognized.
- Use standard JSON format in the body of the application/json request.
- Do not use carriage return characters in Base64 code.
- The system does not save images of users.
- The image size must be less than 8 MB. If the image is too large, the latency is long and the image information volume is small. It is recommended that the image size be less than 1 MB.
- The image resolution must be less than 4,096 x 2,160. The face resolution in an image must be greater than 80 x 80. It is recommended that the face resolution be greater than 120 x 120.
- To ensure the recognition effect, facial images need to meet the following requirements:
  a. The illumination should be greater than 200 lux and there is no light reflection or shadow caused by strong light.
b. The overall image is clear without obvious motion blur and the face in it is not blocked.

c. The side face angle does not exceed $30^\circ$, and the tilt angle and horizontal angle do not exceed $15^\circ$. The face in an image must be a vertically placed front face.

- For details about other restrictions, see **Restrictions and Limitations**.

**Suggestions:**

- A larger image does not significantly improve the recognition algorithm precision but will cause a long latency. Therefore, you are advised to upload an image smaller than 1 MB. Generally, 500 KB is sufficient.

- It is recommended that the size of an image stored on OBS be less than 1 MB.

- It is recommended that the face resolution in an image be greater than $120 \times 120$.

**URI**

- **URI format**

  POST /v2/{project_id}/face-sets/{face_set_name}/search

- **Parameter description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see <strong>Obtaining the Project ID, Account Name, and AK/SK</strong>.</td>
</tr>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the facial image library</td>
</tr>
</tbody>
</table>

**Request Message**

**Table 5-5 Request parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>image_url</td>
<td>String</td>
<td>Either</td>
<td>Image URL. Currently, only the URL of an OBS bucket on HUAWEI CLOUD is supported and Face Recognition must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see <strong>Service Authorization</strong>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>image_url, image_file, image_base 64, or face_id is mandatory.</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Type</td>
<td>Mandatory</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>image_file</td>
<td>File</td>
<td>Either</td>
<td>Local image file, whose size cannot exceed 8 MB. It is recommended that the image size be <strong>less than 1 MB</strong>. The request format is Multipart.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>image_file, image_url, image_base64, or face_id is mandatory.</td>
<td></td>
</tr>
<tr>
<td>image_base64</td>
<td>String</td>
<td>Either</td>
<td>Image data (Base64-encoded). Its requirements are as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>image_base64, image_url, image_file, or face_id is mandatory.</td>
<td></td>
</tr>
<tr>
<td>face_id</td>
<td>String</td>
<td>Either</td>
<td>Face ID returned by the system after a face is imported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>face_id, image_url, image_file, or image_base64 is mandatory.</td>
<td></td>
</tr>
<tr>
<td>top_n</td>
<td>Integer</td>
<td>No</td>
<td><strong>N</strong> faces returned that are most similar to the input one. The default value of <strong>N</strong> is <strong>10</strong>.</td>
</tr>
<tr>
<td>threshold</td>
<td>Double</td>
<td>No</td>
<td>Face similarity threshold. If the similarity degree of a face is lower than the threshold, the face is not returned. The value ranges from <strong>0</strong> to <strong>1</strong>. The recommended value is <strong>0.93</strong>. The default value is <strong>0</strong>.</td>
</tr>
<tr>
<td>sort</td>
<td>JsonArray</td>
<td>No</td>
<td>Field sorting. For details, see <strong>Sort Syntax</strong>.</td>
</tr>
<tr>
<td>filter</td>
<td>String</td>
<td>No</td>
<td>Filtering criteria. For details, see <strong>Filter Syntax</strong>.</td>
</tr>
<tr>
<td>return_fields</td>
<td>String</td>
<td>No</td>
<td>Returned customized field.</td>
</tr>
</tbody>
</table>
Response Message

Table 5-6 Response parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>faces</td>
<td>List</td>
<td>Face set to be retrieved. For details, see SearchFace. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see Error Code. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

Samples

- **Sample request (Method 1: Use a Base64-encoded image.)**
  POST https://{endpoint}/v2/{project_id}/face-sets/showFaceSet/search
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYIKoZIhvNAQcCoIINODCCDTQCAQExDT...
  Request Body:
  ```
  {
    "image_base64": "/9j/4AAQSkZJRgABAgEASABIAAD",
    "sort": [
      {
        "timestamp": "asc"
      }
    ],
    "return_fields": ["timestamp", "id"],
    "filter": "timestamp:12"
  }
  ```

- **Sample request (Method 2: Use an image file.)**
  POST https://{endpoint}/v2/{project_id}/face-sets/showFaceSet/search
  Request Header:
  X-Auth-Token: MIINRwYIKoZIhvNAQcCoIINODCCDTQCAQExDT...
  Request Body:
  ```
  image_file: File (image file)
  return_fields: ["timestamp","id"]
  filter: timestamp:12
  ```

- **Sample request (Method 3: Use the image URL.)**
  POST https://{endpoint}/v2/{project_id}/face-sets/showFaceSet/search
  Request Header:
  Content-Type: application/json
X-Auth-Token: MIINRwYJKoZlhlcvNAQcColINODCCDTQCAQEExDT...

Request Body:
{
   "image_url": "/BucketName/ObjectName",
   "sort": [
      {
         "timestamp": "asc"
      }
   ],
   "return_fields": ["timestamp", "id"],
   "filter": "timestamp:12"
}

● Sample request (Method 4: Use the face ID.)
POST https://{endpoint}/v2/{project_id}/face-sets/showFaceSet/search
Request Header:
Content-Type: application/json
X-Auth-Token: MIINRwYJKoZlhlcvNAQcColINODCCDTQCAQEExDT...

Request Body:
{
   "face_id": "6KLB1Ktu",
   "sort": [
      {
         "timestamp": "asc"
      }
   ],
   "return_fields": ["timestamp", "id"],
   "filter": "timestamp:12"
}

● Sample response
{
   "faces": [
      {
         "bounding_box": {
            "width": 170,
            "top_left_y": 37,
            "top_left_x": 20,
            "height": 170
         },
         "similarity": 0.996146,
         "external_image_id": "123",
         "external_fields": {
            "id": "home",
            "timestamp": 12
         },
         "face_id": "6KLB1Ktu"
      },
      {
         "bounding_box": {
            "width": 170,
            "top_left_y": 37,
            "top_left_x": 20,
            "height": 170
         },
         "similarity": 0.996146,
         "external_image_id": "12",
         "external_fields": {
            "id": "home1",
            "timestamp": 12
         },
         "face_id": "PexOpqRj"
      }
   ]
}

● Failed sample response
{
   "error_code": "FRS.0018",
}
5.4 Facial Image Library Management

5.4.1 Creating a Facial Image Library

Function

This API is used to create a facial image library for storing facial features. You can create a maximum of 10 facial image libraries. Each library can contain a maximum of 100,000 facial features. If you need higher specifications, contact the customer service personnel.

NOTE

Use standard JSON format in the body of the application/json request.

URI

- URI format
  POST /v2/{project_id}/face-sets

- Parameter description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
</tbody>
</table>
## Request Message

### Table 5-7 Request parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
</table>
| face_set_name | String  | Yes       | Name of the facial image library  
Do not start the name of a facial image library with an underscore (_). Otherwise, the Cloud Eye service cannot collect the number of faces. |
| face_set_capacity | Integer | No       | Maximum capacity of a facial image library. The value is an integer multiple of 10,000, for example, 30,000.  
The default value is 100,000, and the maximum value is 100,000. |
| external_fields | Json    | No       | User-defined data. User-defined fields cannot be named in the following system fields: `vector`, `bounding_box`, `external_image_id`, `face_id`, `create_time`, `_id`, `_all`, and `_source`.  
Duplication of the JSON character string is not verified. The key value of a user-defined field contains 1 to 36 characters, and the value of the string type contains 1 to 256 characters.  
For details, see [Customized Fields](#). |

## Response Message

- Parameter description
### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_info</td>
<td>Object</td>
<td>Facial image library information. For details, see <a href="#">FaceSetInfo</a>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see <a href="#">Error Code</a>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

### Samples

- **Sample request**
  
  POST https://(endpoint)/v2/(project_id)/face-sets
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYJKoZlvcNAQcCoIINODCCDTQCAQEExDT...
  
  Request Body:
  
  ```json
  {  
      "face_set_name": "test",  
      "face_set_capacity": 100000,  
      "external_fields": {  
          "timestamp": {  
              "type": "long"  
          },  
          "id": {  
              "type": "string"  
          },  
          "number": {  
              "type": "integer"  
          }  
      }  
  }
  ```

- **Sample response**
  
  ```json
  {  
      "face_set_info": {  
          "face_number": 0,  
          "face_set_id": "WYYOFIGb",  
          "face_set_name": "test",  
          "create_date": "2018-05-28 02:19:00",  
          "face_set_capacity": 10000,  
          "external_fields": {  
              "timestamp": {  
                  "type": "long"  
              }  
          }  
      }  
  }
  ```
5.4.2 Querying All Facial Image Libraries

Function

This API is used to query the statuses of all facial image libraries of the current user.

URI

- URI format
  
  GET /v2/{project_id}/face-sets

- Parameter description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
</tbody>
</table>

Request Message

NA
Response Message

Table 5-8 Response parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_sets_info</td>
<td>List</td>
<td>Information about all facial image libraries. For details, see <code>FaceSetInfo</code>. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see <code>Error Code</code>. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

Samples

- **Sample request**
  
  Get https://{endpoint}/v2/{project_id}/face-sets  
  Request Header:  
  Content-Type: application/json  
  X-Auth-Token: MIINRwYJkoZlhcNAQcCoIINODCCDTQCAQE<DT...  

- **Sample response**
  
  
  ```json
  {  
    "face_sets_info": [  
      {  
        "face_number": 0,  
        "face_set_id": "y0XMMZTO",  
        "face_set_name": "test",  
        "create_date": "2018-05-11 07:49:40",  
        "face_set_capacity": 10000,  
        "external_fields": {  
          "number": {  
            "type": "integer"  
          },  
          "id": {  
            "type": "string"  
          },  
          "timestamp": {  
            "type": "long"  
          }  
        }  
      }  
    ]  
  }  
  ```
5.4.3 Querying a Facial Image Library

Function

This API is used to query the status of a facial image library.

URI

- URI format
  
  GET /v2/{project_id}/face-sets/{face_set_name}

- Parameter description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the facial image library</td>
</tr>
</tbody>
</table>

Request Message

NA
### Response Message

#### Table 5-9 Response parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_info</td>
<td>Object</td>
<td>Facial image library information. For details, see <a href="#">FaceSetInfo</a>. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see <a href="#">Error Code</a>. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

### Samples

- **Sample request**
  
  Get https://{endpoint}/v2/{project_id}/face-sets/showFaceSet  
  Request Header:  
  Content-Type: application/json  
  X-Auth-Token: MIINRwYlKoZlhcNAQcCoIINODCCDTQCAQExDT...

- **Sample response**
  
  ```json
  {
    "face_set_info": {
      "face_number": 94,
      "face_set_id": "T785tx1N",
      "face_set_name": "showFaceSet",
      "create_date": "2018-05-10 01:44:39",
      "face_set_capacity": 10000,
      "external_fields": {
        "number": {
          "type": "integer"
        },
        "id": {
          "type": "string"
        },
        "timestamp": {
          "type": "long"
        }
      }
    }
  }
  ```
Failed sample response
{
  "error_code": "FRS.0202",
  "error_msg": "The service has been freeze."
}

Status Code
For details about the status code, see Status Codes.

Error Code
For details about the error code, see Error Code.

5.4.4 Deleting a Facial Image Library

Function
This API is used to delete a facial image library and all faces in the library.

URI

- **URI format**
  DELETE /v2/{project_id}/face-sets/{face_set_name}

- **Parameter description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the facial image library</td>
</tr>
</tbody>
</table>

Request Message
N/A

Response Message

**Table 5-10 Response parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Name of the facial image library. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see Error Code. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

**Samples**

- **Sample request**

  DELETE https://{endpoint}/v2/{project_id}/face-sets/showFaceSet
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYJKoZIhvcNAQcColINODCCDTQCAQExDT...

- **Sample response**

  ```
  {
    "face_set_name": "showFaceSet"
  }
  ```

- **Failed sample response**

  ```
  {
    "error_code": "FRS.0002",
    "error_msg": "The authentication token is abnormal."
  }
  ```

**Status Code**

For details about the status code, see Status Codes.

**Error Code**

For details about the error code, see Error Code.

### 5.5 Facial Resource Management

#### 5.5.1 Adding a Face

**Function**

This API is used to add faces to a facial image library. The largest face detected in an image will be added to the library.

**Restrictions:**
- Only images in JPG, PNG, JPEG, or BMP format can be recognized.
- Use standard JSON format in the body of the `application/json` request.
- Do not use carriage return characters in Base64 code.
- The system does not save images of users.
- The image size must be **less than 8 MB**. If the image is too large, the latency is long and the image information volume is small. It is recommended that the image size be **less than 1 MB**.
- The image resolution must be **less than 4,096 x 2,160**. The face resolution in an image must be **greater than 80 x 80**. It is recommended that the face resolution be **greater than 120 x 120**.
- To ensure the recognition effect, facial images need to meet the following requirements:
  a. The illumination should be greater than 200 lux and there is no light reflection or shadow caused by strong light.
  b. The overall image is clear without obvious motion blur and the face in it is not blocked.
  c. The side face angle does not exceed 30°, and the tilt angle and horizontal angle do not exceed 15°. The face in an image must be a vertically placed front face.
- For details about other restrictions, see [Restrictions and Limitations](#).

**Suggestions:**
- A larger image does not significantly improve the recognition algorithm precision but will cause a long latency. Therefore, you are advised to upload an image smaller than **1 MB**. Generally, **500 KB** is sufficient.
- It is recommended that the size of an image stored on OBS be less than **1 MB**.

**URI**

- **URI format**
  ```
  POST /v2/{project_id}/face-sets/{face_set_name}/faces
  ```
- **Parameter description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see <a href="#">Obtaining the Project ID, Account Name, and AK/SK</a>.</td>
</tr>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the facial image library</td>
</tr>
</tbody>
</table>
## Request Message

### Table 5-11 Request parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>image_url</td>
<td>String</td>
<td>Either</td>
<td>Image URL. Currently, only the URL of an OBS bucket on HUAWEI CLOUD is supported and Face Recognition must have the permission to read data in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 values</td>
<td>the OBS bucket. For details about how to enable the read permission, see Service Authorization.</td>
</tr>
<tr>
<td>image_file</td>
<td>File</td>
<td>Either</td>
<td>Local image file, whose size cannot exceed 8 MB. It is recommended that the image size be less than 1 MB. The request format is Multipart.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 values</td>
<td></td>
</tr>
<tr>
<td>image_base64</td>
<td>String</td>
<td>Either</td>
<td>Image data (Base64-encoded). Its requirements are as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 values</td>
<td>• The image size after Base64 encoding cannot exceed 8 MB. It is recommended that the image size be less than 1 MB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or</td>
<td>• The image is in JPG, JPEG, BMP, or PNG format.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mandatory.</td>
<td></td>
</tr>
<tr>
<td>external_image_id</td>
<td>String</td>
<td>No</td>
<td>External image ID specified by the user. It is bound to the current image. If the user does not provide one, it is generated by the system.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The ID contains 1 to 36 characters, including letters, digits, hyphens (-), and underscores (_). Other special characters are not allowed.</td>
</tr>
<tr>
<td>external_fields</td>
<td>Json</td>
<td>No</td>
<td>Enters a value based on the customized data type.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This field is defined when a face set is created. Duplication of the JSON character string is not verified. For details, see Customized Fields.</td>
</tr>
</tbody>
</table>
Response Message

Table 5-12 Response parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Name of the facial image library</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>face_set_id</td>
<td>String</td>
<td>ID of the facial image library</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>faces</td>
<td>List</td>
<td>Face structure in the facial image library.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For details, see <a href="#">FaceSetFace</a>. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For details, see <a href="#">Error Code</a>. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

Samples

- Sample request (Method 1: Use a Base64-encoded image.)

  POST https://{endpoint}/v2/{project_id}/face-sets/showFaceSet/faces

  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYJKoZIhvNAQcCoIINODCCDTQCAQExDT...

  Request Body:
  {
    "image_base64": "/9j/4AAQSkZJRgABAgEASABIAAD",
    "external_image_id": "imageID",
    "external_fields": {
      "timestamp": 12,
      "id": "home"  }"
### Sample request (Method 2: Use an image file.)

**POST** `https://{endpoint}/v2/{project_id}/face-sets/showFaceSet/faces`

**Request Header:**
- `X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...`

**Request Body:**
- `image_file`: File (image file)
- `external_image_id`: imageID
- `external_fields`: `{ "timestamp": 12, "id": "home" }`

### Sample request (Method 3: Use the image URL.)

**POST** `https://{endpoint}/v2/{project_id}/face-sets/showFaceSet/faces`

**Request Header:**
- `Content-Type: application/json`
- `X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...`

**Request Body:**
- `{  
  "image_url": "/BucketName/ObjectName",
  "external_image_id": "imageID",
  "external_fields": { 
    "timestamp": 12,
    "id": "home"
  }
} `

### Sample response

```json
{
  "face_set_id": "T785tx1N",
  "face_set_name": "showFaceSet",
  "faces": [
    {
      "bounding_box": {
        "width": 63,
        "top_left_y": 100,
        "top_left_x": 221,
        "height": 63
      },
      "external_image_id": "Xr0phyap",
      "external_fields": { 
        "timestamp": 12,
        "id": "home"
      },
      "face_id": "JLa9hYLI"
    }
  ]
}
```

### Failed sample response

```json
{
  "error_code": "FRS.0404",
  "error_msg": "Detect no face, can not add it to face set."
}
```

## Status Code

For details about the status code, see [Status Codes](#).

## Error Code

For details about the error code, see [Error Code](#).
5.5.2 Querying a Face

**Function**

This API is used to query the face information in a specified facial image library.

**URI**

- **URI format**
  
  GET /v2/{project_id}/face-sets/{face_set_name}/faces?offset=xxx&limit=xxx
  
  or
  
  GET /v2/{project_id}/face-sets/{face_set_name}/faces?face_id={face_id}

- **Parameter description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the facial image library</td>
</tr>
<tr>
<td>offset</td>
<td>Integer</td>
<td>Either offset or face_id is mandatory.</td>
<td>Start number of data records to be read. The default value is 0.</td>
</tr>
<tr>
<td>limit</td>
<td>Integer</td>
<td>Either limit or face_id is mandatory.</td>
<td>Number of records to be read. The default value is 5.</td>
</tr>
<tr>
<td>face_id</td>
<td>String</td>
<td>Either face_id or offset is mandatory.</td>
<td>Face ID</td>
</tr>
</tbody>
</table>

**NOTE**

The sum of values of offset and limit cannot exceed 10,000.

**Request Message**

N/A
Response Message

Table 5-13 Response parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Name of the facial image library. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>face_set_id</td>
<td>String</td>
<td>ID of the facial image library. It is a randomly generated string containing eight characters. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>faces</td>
<td>List</td>
<td>Returned face information. For details, see FaceSetFace. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see ErrorCode. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

Samples

- **Sample request**
  Get https://{endpoint}/v2/{project_id}/face-sets/showFaceSet/faces?offset=0&limit=1
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

- **Sample response**
  ```json
  {
    "face_set_id": "T785tx1N",
    "face_set_name": "showFaceSet",
    "faces": [
      {
        "bounding_box": {
          "width": 63,
          "top_left_y": 100,
          "top_left_x": 221,
          "height": 63
        },
        "external_image_id": "aIzRAa58",
        "face_id": "cFydu4d2",
        "external_fields": {
          "number": 122,
          "id": "home",
          ...]```
● Failed sample response
{
  "error_code": "FRS.0002",
  "error_msg": "The authentication token is abnormal."
}

**Status Code**

For details about the status code, see [Status Codes](#).

**Error Code**

For details about the error code, see [Error Code](#).

### 5.5.3 Updating a Face

**Function**

This API is used to update a single face based on its face ID (face_id).

**NOTE**

Use standard JSON format in the body of the application/json request.

**URI**

- **URI format**
  
  ```
  PUT   /v2/{project_id}/face-sets/{face_set_name}/faces
  ```

- **Parameter description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see <a href="#">Obtaining the Project ID, Account Name, and AK/SK</a>.</td>
</tr>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the facial image library</td>
</tr>
</tbody>
</table>

**Request Message**

<table>
<thead>
<tr>
<th>Table 5-14 Request parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>face_id</td>
</tr>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>--------------------</td>
</tr>
<tr>
<td>external_image_id</td>
</tr>
<tr>
<td>external_fields</td>
</tr>
</tbody>
</table>

**Response Message**

**Table 5-15 Response parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Name of the facial image library. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>face_set_id</td>
<td>String</td>
<td>ID of the facial image library. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>face_number</td>
<td>Integer</td>
<td>Number of updated faces. This parameter is not included when the API fails to be called.</td>
</tr>
</tbody>
</table>
### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see Error Code. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

### Samples

- **Sample request**
  
  PUT https://(endpoint)/v2/(project_id)/face-sets/showFaceSet/faces
  
  Request Header:
  
  Content-Type: application/json
  X-Auth-Token: MIINRwYJKoZlhcNAQcCoIINODCCDTQCAQEExDT...
  
  Request Body:
  
  ```json
  {
  "face_id": "iexEBb6l",
  "external_image_id": "imageID",
  "external_fields": {
  "timestamp": 12,
  "id": "300018629384756"
  } }
  ```

- **Sample response**
  
  ```json
  {
  "face_number": 1,
  "face_set_id": "T785tx1N",
  "face_set_name": "showFaceSet"
  }
  ```

- **Failed sample response**
  
  ```json
  {
  "error_code": "FRS.0303",
  "error_msg": "The face id is not exist, checkout your input."}
  ```

### Status Code

For details about the status code, see Status Codes.

### Error Code

For details about the error code, see Error Code.
5.5.4 Deleting a Face

Function

This API is used to delete a face from the facial image library based on the specified field.

URI

- URI format

  
  DELETE /v2/{project_id}/face-sets/{face_set_name}/faces?field_name=field_value

- Parameter description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see Obtaining the Project ID, Account Name, and AK/SK.</td>
</tr>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the facial image library</td>
</tr>
<tr>
<td>field_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the field deleted based on conditions. Fixed fields (external_image_id and face_id) and customized fields (excluding null strings and values) are supported. For more information about customized fields, see Customized Fields.</td>
</tr>
</tbody>
</table>

Request Message

N/A

Response Message

Table 5-16 Response parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Name of the facial image library This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>face_set_id</td>
<td>String</td>
<td>ID of the facial image library This parameter is not included when the API fails to be called.</td>
</tr>
</tbody>
</table>
## Face Recognition APIs (V2)

### 5.5.5 Batch Deleting Faces

#### Function

This API is used to batch delete multiple faces that meet specified criteria you customize.

**NOTE**

Use standard JSON format in the body of the `application/json` request.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_number</td>
<td>Integer</td>
<td>Number of deleted faces. This parameter is not included when the API fails to be called.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see Error Code. This parameter is not included when the API is successfully called.</td>
</tr>
<tr>
<td>error_msg</td>
<td>String</td>
<td>Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.</td>
</tr>
</tbody>
</table>

### Samples

- **Sample request (Method 1: Use `external_image_id` to delete a facial image.)**
  
  ```
  DELETE https://{endpoint}/v2/{project_id}/face-sets/showFaceSet/faces?external_image_id=imageID
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYIkoZIhvNAQcColINODCCDTQCAQExDT...
  ```

- **Sample request (Method 2: Use `face_id` to delete a facial image.)**
  
  ```
  DELETE https://{endpoint}/v2/{project_id}/face-sets/showFaceSet/faces?face_id=faceID
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYIkoZIhvNAQcColINODCCDTQCAQExDT...
  ```

- **Sample request (Method 3: Use a customized field to delete a facial image.)**
  
  ```
  DELETE https://{endpoint}/v2/{project_id}/face-sets/showFaceSet/faces?id=home
  Request Header:
  Content-Type: application/json
  X-Auth-Token: MIINRwYIkoZIhvNAQcColINODCCDTQCAQExDT...
  ```

- **Sample response**
  
  ```
  {"face_number": 1,
   "face_set_id": "T785tx1N",
   "face_set_name": "showFaceSet"
  }
  ```

- **Failed sample response**
  
  ```
  {"error_code": "FRS.0402",
   "error_msg": "External id is not exist, can not delete face"
  }
URI

- **URI format**
  
  `DELETE /v2/{project_id}/face-sets/{face_set_name}/faces/batch`

- **Parameter description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Yes</td>
<td>Project ID. For details about how to obtain the ID, see <a href="#">Obtaining the Project ID, Account Name, and AK/SK</a>.</td>
</tr>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Yes</td>
<td>Name of the facial image library.</td>
</tr>
</tbody>
</table>

**Request Message**

**Table 5-17 Request parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>String</td>
<td>Yes</td>
<td>Filtering criteria. For details, see <a href="#">Filter Syntax</a>.</td>
</tr>
</tbody>
</table>

**Response Message**

**Table 5-18 Response parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Name of the facial image library.</td>
</tr>
<tr>
<td>face_set_id</td>
<td>String</td>
<td>ID of the facial image library.</td>
</tr>
<tr>
<td>face_number</td>
<td>Integer</td>
<td>Number of deleted faces.</td>
</tr>
<tr>
<td>error_code</td>
<td>String</td>
<td>Error code returned after the API fails to be called. For details, see <a href="#">Error Code</a>.</td>
</tr>
</tbody>
</table>

This parameter is not included when the API fails to be called.
### Samples

- **Sample request (Use filter to batch delete faces.)**
  
  DELETE https://{endpoint}/v2/{project_id}/face-sets/showFaceSet/faces/batch
  
  Request Header:
  
  Content-Type: application/json
  X-Auth-Token: MIINRWYJKoZIhvcNAQcColINODCCDTQCAQExDT...

  Request Body:

  ```json
  {
    "filter": "age:[20 TO 30]"
  }
  ```

- **Sample response**

  ```json
  {
    "face_number": 1,
    "face_set_id": "T785tx1N",
    "face_set_name": "showFaceSet"
  }
  ```

- **Failed sample response**

  ```json
  {
    "error_code": "FRS.0407",
    "error_msg": "All the data not suitable, no data to be deleted."
  }
  ```

### Status Code

For details about the status code, see [Status Codes](#).

### Error Code

For details about the error code, see [Error Code](#).
6 Public Data Structures

6.1 Common Message Headers

Common Request Headers

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Mandatory</th>
<th>Example Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-type</td>
<td>Request body type or format. Its default value is application/json.</td>
<td>Yes</td>
<td>application/json</td>
</tr>
<tr>
<td>Content-Length</td>
<td>Length of the request body. The unit is byte.</td>
<td></td>
<td>3495</td>
</tr>
<tr>
<td>X-Project-Id</td>
<td>Project ID. This parameter is used to obtain the token for each project.</td>
<td>No</td>
<td>e9993fc787d94b6c886 cbaa340f9c0f4</td>
</tr>
<tr>
<td>X-Auth-Token</td>
<td>User token, a response to the API used to obtain a user token. This API is the only one that does not require authentication.</td>
<td>No</td>
<td>-</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Mandatory</td>
<td>Example Value</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>X-Sdk-Date</td>
<td>Time when the request is sent. The time is in <code>YYYYMMDD'T'HHMMS'S'Z'</code> format. The value is the current Greenwich Mean Time (GMT) of the system.</td>
<td>No</td>
<td>20190307T101459Z</td>
</tr>
<tr>
<td>Authorization</td>
<td>Signature authentication information. The value is obtained from the request signature result and is required when the AK and SK are used to encrypt the signature. Type: string Default value: none</td>
<td>No</td>
<td>SDK-HMAC-SHA256 Credential=ZIRRKMT WPTQFQ11WKNKB/20150907//ec2/sdk_request, SignedHeaders=content-type;host;x-sdk-date, Signature=55741b610f3c9fa3ae40b5a8021ebf7ebc2a28a603fc62d25cb3bfe6608e1994</td>
</tr>
<tr>
<td>Host</td>
<td>Information about the requested server. The value can be obtained from the URL of the service API. The value is <code>hostname[:port]</code>. If the port number is not specified, the default port is used. The default port number for <code>https</code> is 443.</td>
<td>No</td>
<td>code.test.com or code.test.com:443</td>
</tr>
</tbody>
</table>

**NOTE**

For details about other parameters in the header, see the HTTPS protocol documentation.
Common Response Headers

Table 6-2 Common response headers

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Type</td>
<td>Media type of the message body sent to a receiver. Type: string. Default value: <code>application/json; charset=UTF-8</code></td>
<td>Yes</td>
</tr>
<tr>
<td>X-request-id</td>
<td>This field carries the request ID for task tracing. Type: string. Default value: none.</td>
<td>No</td>
</tr>
<tr>
<td>X-ratelimit</td>
<td>This field carries the total number of flow control requests. Type: integer. Default value: none</td>
<td>No</td>
</tr>
<tr>
<td>X-ratelimit-used</td>
<td>This field carries the number of remaining requests. Type: integer. Default value: none.</td>
<td>No</td>
</tr>
<tr>
<td>X-ratelimit-window</td>
<td>This field carries the flow control unit. Type: string. The unit is minute, hour, or day. Default value: hour</td>
<td>No</td>
</tr>
</tbody>
</table>

6.2 Message Object Structures

6.2.1 AllParam

Function

This topic describes all parameter types involved in Face Recognition.
## Table 6-3 Structure format description

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_id</td>
<td>String</td>
<td>Project ID. For details about how to obtain the ID, see <a href="#">Obtaining the Project ID, Account Name, and AK/SK</a>.</td>
</tr>
<tr>
<td>image_url</td>
<td>String</td>
<td>Image URL. Currently, only the URL of an OBS bucket on HUAWEI CLOUD is supported and Face Recognition must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see <a href="#">Service Authorization</a>.</td>
</tr>
<tr>
<td>image_file</td>
<td>File</td>
<td>Local image file, whose size cannot exceed 8 MB. It is recommended that the image size be <strong>less than 1 MB</strong>. The request format is Multipart.</td>
</tr>
</tbody>
</table>
| image_base64    | String   | Image data (Base64-encoded). Its requirements are as follows:  
  - The image size after Base64 encoding cannot exceed 8 MB. It is recommended that the image size be **less than 1 MB**.  
  - The image is in JPG, JPEG, BMP, or PNG format. |
<p>| similarity      | Double   | Face similarity. The value ranges from 0 to 1. A larger value indicates a higher similarity degree. Generally, if the value is greater than <strong>0.93</strong>, the faces in two images belong to one person. |
| face_set_name   | String   | Name of the facial image library. The value contains 1 to 64 characters, including letters, digits, hyphens (-), and underscores (_). Other special characters are not allowed. |
| face_set_capacity | Integer  | Maximum capacity of a facial image library. The value is an integer multiple of 10000, for example, 30000. The default value is <strong>100000</strong>, and the maximum value is <strong>1000000</strong>. |
| face_id         | String   | Face ID returned by the system after a face is imported. The value is a string of eight uppercase and lowercase letters that are randomly generated. |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>external_image_id</td>
<td>String</td>
<td>External image ID specified by the user. It is bound to the current image. If the user does not provide one, it is generated by the system. The ID contains 1 to 36 characters, including letters, digits, hyphens (-), and underscores (_). Other special characters are not allowed.</td>
</tr>
<tr>
<td>external_fields</td>
<td>Json</td>
<td>Enters a value based on the customized data type. This field is defined when you create the facial image library. Duplication of the JSON character string is not verified. For details, see section Customized Fields.</td>
</tr>
<tr>
<td>top_n</td>
<td>Integer</td>
<td>$N$ faces returned that are most similar to the input one. The default value of $N$ is 10. If the first five faces are returned, the value of variable $N$ is 5. Value range: 1-1000</td>
</tr>
<tr>
<td>threshold</td>
<td>Double</td>
<td>Face similarity threshold. If the similarity degree of a face is lower than the threshold, the face is not returned. The value ranges from 0 to 1. The recommended value is 0.93. The default value is 0.</td>
</tr>
<tr>
<td>offset</td>
<td>Integer</td>
<td>Start number of data records to be read. The default value is 0.</td>
</tr>
<tr>
<td>limit</td>
<td>Integer</td>
<td>Number of records to be read. The default value is 5.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| video_url  | String | Video URL. Currently, only the URL of an OBS bucket on HUAWEI CLOUD is supported and Face Recognition must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see Service Authorization. The video requirements are as follows:  
  - The video size after Base64 encoding cannot exceed 8 MB.  
  - The video duration must be 1 to 15 seconds.  
  - The recommended frame rate is 10 fps to 30 fps.  
  - The encapsulation format can be MP4, AVI, FLV, WEBM, ASF, or MOV.  
  - The video encoding format can be H.261, H.263, H.264, HEVC, VC-1, VP8, VP9, or WMV3. |
| video_file | File   | Local video file. The request format is Multipart. The video requirements are as follows:  
  - The size of a video file cannot exceed 8 MB. It is recommended that the video file be compressed to 200 KB to 2 MB on the client.  
  - The video duration must be 1 to 15 seconds.  
  - The recommended frame rate is 10 fps to 30 fps.  
  - The encapsulation format can be MP4, AVI, FLV, WEBM, ASF, or MOV.  
  - The video encoding format can be H.261, H.263, H.264, HEVC, VC-1, VP8, VP9, or WMV3. |
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| video_base64| String | Video data (Base64-encoded). Its requirements are as follows:  
  ● The video size after Base64 encoding cannot exceed 8 MB. It is recommended that the video file be compressed to **200 KB to 2 MB** on the client.  
  ● The video duration must be 1 to 15 seconds.  
  ● The recommended frame rate is 10 fps to 30 fps.  
  ● The encapsulation format can be MP4, AVI, FLV, WEBM, ASF, or MOV.  
  ● The video encoding format can be H.261, H.263, H.264, HEVC, VC-1, VP8, VP9, or WMV3. |
| actions     | String | Action code sequence list. Actions are separated by commas (,). Currently, the following actions are supported:  
  ● 1: Shake the head to the left.  
  ● 2: Shake the head to the right.  
  ● 3: Nod the head.  
  ● 4: Mouth movement |
<p>| action_time | String | String of the action time array. The length of the array is the same as the number of actions. Each item contains the start time and end time of the action in the corresponding sequence. The unit is the milliseconds from the video start time. |
| error_code  | String | Error code returned after the API fails to be called. |
| error_msg   | String | Error message returned after the API fails to be called. |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| attributes | String  | Whether to return the facial attribute list. Multiple attributes are separated by commas (,). Currently, the following attributes are supported:  
- 0: Face posture  
- 1: Gender  
- 2: Age  
- 3: Key facial point  
- 4: Dress (hat, glasses)  
- 5: Smiling face |

6.2.2 DetectFace

Function

This topic describes the face structures returned by Face Detection and Face Verification.

Parameter Description

<table>
<thead>
<tr>
<th>Table 6-4 Structure format description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>bounding_box</td>
</tr>
<tr>
<td>landmark</td>
</tr>
<tr>
<td>attributes</td>
</tr>
</tbody>
</table>
6.2.3 Landmark

Function

This topic describes structures of key facial points.

Parameter Description

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nose_contour</td>
<td>List [Point]</td>
<td>Nose contour. Point is the coordinate value of the contour.</td>
</tr>
<tr>
<td>mouth_contour</td>
<td>List [Point]</td>
<td>Mouth contour. Point is the coordinate value of the contour.</td>
</tr>
<tr>
<td>eyebrow_contour</td>
<td>List [Point]</td>
<td>Eyebrow contour. Point is the coordinate value of the contour.</td>
</tr>
<tr>
<td>eyes_contour</td>
<td>List [Point]</td>
<td>Eyes contour. Point is the coordinate value of the contour.</td>
</tr>
<tr>
<td>face_contour</td>
<td>List [Point]</td>
<td>Face contour. Point is the coordinate value of the contour.</td>
</tr>
</tbody>
</table>

6.2.4 Attributes

Function

This topic describes facial attribute structures.

Parameter Description

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>Integer</td>
<td>Age</td>
</tr>
<tr>
<td>smile</td>
<td>String</td>
<td>A smiling face</td>
</tr>
<tr>
<td>gender</td>
<td>String</td>
<td>Gender. Possible values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• male</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• female</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• unknown</td>
</tr>
<tr>
<td>dress</td>
<td>List of strings</td>
<td>Contains the glass and hat attributes.</td>
</tr>
</tbody>
</table>
### Table 6-7 Structure format description

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bounding_box</td>
<td>BoundingBox object</td>
<td>Position of a face in an image</td>
</tr>
<tr>
<td>face_id</td>
<td>String</td>
<td>Face ID, the unique ID generated by the system</td>
</tr>
<tr>
<td>external_image_id</td>
<td>String</td>
<td>ID of the external image to which a face belongs</td>
</tr>
</tbody>
</table>

**6.2.5 FaceSetFace**

**Function**

This topic describes the face structures in the facial image library.

**Parameter Description**
6.2.6 SearchFace

Function

This topic describes face structures returned by Face Retrieval.

Parameter Description

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bounding_box</td>
<td>BoundingBox object</td>
<td>Position of a face in an image</td>
</tr>
<tr>
<td>face_id</td>
<td>String</td>
<td>Face ID, the unique ID generated by the system</td>
</tr>
<tr>
<td>external_image_id</td>
<td>String</td>
<td>ID of the external image to which a face belongs</td>
</tr>
<tr>
<td>similarity</td>
<td>Double</td>
<td>Similarity degree in Face Retrieval</td>
</tr>
<tr>
<td>external_fields</td>
<td>Json</td>
<td>Additional field a user customizes</td>
</tr>
</tbody>
</table>

6.2.7 FaceSetInfo

Function

This topic describes the basic information about a facial image library.

Parameter Description

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>face_set_name</td>
<td>String</td>
<td>Name of the facial image library</td>
</tr>
</tbody>
</table>
### 6.2.8 BoundingBox

**Function**

This topic describes the position of a face in an image. The origin of coordinates (0, 0) is in the upper left corner.

**Parameter Description**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>top_left_x</td>
<td>Integer</td>
<td>Horizontal coordinate of the upper-left corner of a rectangle</td>
</tr>
<tr>
<td>top_left_y</td>
<td>Integer</td>
<td>Vertical coordinate of the upper-left corner of a rectangle</td>
</tr>
<tr>
<td>width</td>
<td>Integer</td>
<td>Width of a rectangle</td>
</tr>
<tr>
<td>height</td>
<td>Integer</td>
<td>Height of a rectangle</td>
</tr>
</tbody>
</table>

### 6.2.9 VideoDetectResult

**Function**

This topic describes the result structures of Face LiveDetect.
**Parameter Description**

<table>
<thead>
<tr>
<th>Table 6-11 Structure format description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>alive</td>
</tr>
<tr>
<td>picture</td>
</tr>
<tr>
<td>actions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 6-12 actions field description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
</tbody>
</table>
| action | Integer | Action ID. Possible values are:  
  ● 1: Shake the head to the left.  
  ● 2: Shake the head to the right.  
  ● 3: Nod the head.  
  ● 4: Mouth movement |
| confidence | Double | Confidence level. The value ranges from 0 to 1. |

**6.2.10 ServiceInfo**

**Function**

This topic describes structures that record sub-service information.

**Parameter Description**

<table>
<thead>
<tr>
<th>Table 6-13 Structure format description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>subscribe_status</td>
</tr>
<tr>
<td>create_time</td>
</tr>
</tbody>
</table>
6.2.11 WarningList

Function

This topic describes warning information about video LiveDetect.

Parameter Description

Table 6-14 Structure and Format Description

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>warningCode</td>
<td>Integer</td>
<td>Warning ID</td>
</tr>
<tr>
<td>warningMsg</td>
<td>String</td>
<td>Warning information</td>
</tr>
</tbody>
</table>

Table 6-15 Errors

<table>
<thead>
<tr>
<th>warningCode</th>
<th>warningMsg</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The face is not facing forward.</td>
</tr>
<tr>
<td>4</td>
<td>Two faces are detected.</td>
</tr>
<tr>
<td>5</td>
<td>No face is detected.</td>
</tr>
<tr>
<td>6</td>
<td>The range of motion is too small.</td>
</tr>
<tr>
<td>7</td>
<td>The video quality is poor or the person in the video is not a real person.</td>
</tr>
<tr>
<td>8</td>
<td>Failed to select a preferred image.</td>
</tr>
</tbody>
</table>

6.3 Customized Fields

Data Type

The String, Integer, Float, Double, Boolean, and Long types are supported.

NOTE

1. external_image_id, bounding_box, similarity, face_id, create_time, vector, _id, _all, and _source are built-in fields that cannot be customized.
2. The number of customized fields cannot exceed 10. The key value contains 1 to 36 characters, including digits, letters, underscores ( _ ), and hyphens (-).
3. The value of the String type contains 1 to 256 characters, including digits, letters, underscores ( _ ), and hyphens (-).
4. Duplicate fields will be overwritten.
5. You are not allowed to add a suffix to a value of the numeral type. For example, 1.0f, 100L, and 1.0d are incorrect.
**Syntax Logic**

*external_fields* is defined in JSON format and the name corresponds to the data type.

Request body:

```json
{
  "face_set_name": "test",
  "face_set_capacity": "100000",
  "external_fields": {
    "location": {
      "type": "long"
    },
    "timestamp": {
      "type": "integer"
    },
    "male": {
      "type": "boolean"
    },
    "title": {
      "type": "string"
    },
    "weight": {
      "type": "double"
    },
    "score": {
      "type": "float"
    }
  }
}
```

---

**6.4 Sort Syntax**

The Sort syntax is in the JSON array format. Only the numeral type is supported. Repeated fields will be overwritten.

- **Example 1:** Sorting of a single field
  ```json
  "sort": [
    {
      "location": "desc"
    }
  ]
  ```

- **Example 2:** Sorting of multiple fields
  ```json
  "sort": [
    {
      "timestamp": "desc"
    },
    {
      "rowkey": "asc"
    }
  ]
  ```

---

**6.5 Filter Syntax**

**Data Type**

1. In value type ranges, [] indicates that the two end values are included while {} indicates that the two end values are not included.

   - rowkey:[1 TO *) Equal to or greater than 1
   - rowkey:[* TO 1] Equal to or less than 1
   - rowkey:[1 TO 10] 1 to 10
2. Single value
rowkey:1 The value is 1.

3. If multiple conditions need to be met, use () to differentiate priorities.
rowkey:[1 TO *] && externalImageID:1
(rowkey:[1 TO *] && externalImageID:1) || timestamp:1000
rowkey:[1 TO *] && (externalImageID:1 || timestamp:1000)

4. Non-statement. You need to add brackets before and after a non-statement.
   externalImageID:1 && (rowkey:2)

---

**String Type**

1. Single value
title:quick

2. Multiple conditions
title:quick && color:brown

3. Non-statement. You need to add brackets before and after a non-statement.
   (!color:brown)

---

**Boolean Type**

Single value
male:true

---

**NOTE**

1. If there are too many logical conditions, use () to differentiate priorities.
2. Null string and value search is not supported.
## 7.1 Status Codes

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Continue</td>
<td>The client should continue with its request. This provisional response informs the client that part of the request has been received and has not yet been rejected by the server.</td>
</tr>
<tr>
<td>101</td>
<td>Switching Protocols</td>
<td>The protocol should be switched. The protocol can only be switched to a newer protocol. For example, the current HTTP protocol is switched to a later version of HTTP.</td>
</tr>
<tr>
<td>201</td>
<td>Created</td>
<td>The request for creating a class has been successfully processed.</td>
</tr>
<tr>
<td>202</td>
<td>Accepted</td>
<td>The request has been accepted, but the processing has not been completed.</td>
</tr>
<tr>
<td>203</td>
<td>Non-Authoritative Information</td>
<td>Unauthorized information. The request is successful.</td>
</tr>
<tr>
<td>204</td>
<td>NoContent</td>
<td>The request has been fulfilled, but the HTTP response does not contain a response body. The status code is returned in response to an HTTP OPTIONS request.</td>
</tr>
<tr>
<td>205</td>
<td>Reset Content</td>
<td>The server has successfully processed the request, but does not return any content.</td>
</tr>
<tr>
<td>206</td>
<td>Partial Content</td>
<td>The server has successfully processed a part of the GET request.</td>
</tr>
<tr>
<td>Status Code</td>
<td>Message</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>300</td>
<td>Multiple Choices</td>
<td>There are multiple options for the location of the requested resource. The response contains a list of resource characteristics and addresses from which the user or user agent (such as a browser) can choose the most appropriate one.</td>
</tr>
<tr>
<td>301</td>
<td>Moved Permanently</td>
<td>This and all future requests should be permanently directed to the given URI indicated in this response.</td>
</tr>
<tr>
<td>302</td>
<td>Found</td>
<td>The requested resource resides temporarily under a different URI.</td>
</tr>
<tr>
<td>303</td>
<td>See Other</td>
<td>The response to the request can be found under a different URI.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The response to the request can be found under a different URI, and should be retrieved using a GET or POST method.</td>
</tr>
<tr>
<td>304</td>
<td>Not Modified</td>
<td>The requested resource has not been modified. In such a case, there is no need to retransmit the resource since the client still has a previously-downloaded copy.</td>
</tr>
<tr>
<td>305</td>
<td>Use Proxy</td>
<td>The requested resource must be accessed through a proxy.</td>
</tr>
<tr>
<td>306</td>
<td>Unused</td>
<td>This HTTP status code is no longer used.</td>
</tr>
<tr>
<td>400</td>
<td>BadRequest</td>
<td>Specifies invalid requests. Do not retry the request before modification.</td>
</tr>
<tr>
<td>401</td>
<td>Unauthorized</td>
<td>The authentication information provided by the client is incorrect or invalid.</td>
</tr>
<tr>
<td>402</td>
<td>Payment Required</td>
<td>This status code is reserved for future use.</td>
</tr>
<tr>
<td>403</td>
<td>Forbidden</td>
<td>The server has received the request and understood it, but the server is refusing to respond to it. The client should modify the request instead of re-initiating it.</td>
</tr>
<tr>
<td>404</td>
<td>NotFound</td>
<td>The requested resource cannot be found. Do not retry the request before modification.</td>
</tr>
<tr>
<td>405</td>
<td>MethodNotAllowed</td>
<td>The request contains one or more methods not supported for the resource. Do not retry the request before modification.</td>
</tr>
<tr>
<td>Status Code</td>
<td>Message</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>406</td>
<td>Not Acceptable</td>
<td>The server could not fulfill the request according to the content characteristics of the request.</td>
</tr>
<tr>
<td>407</td>
<td>Proxy Authentication Required</td>
<td>This code is similar to 401, but indicates that the client must first authenticate itself with the proxy.</td>
</tr>
<tr>
<td>408</td>
<td>Request Time-out</td>
<td>The server timed out waiting for the request. The client may repeat the request without modifications at any time later.</td>
</tr>
<tr>
<td>409</td>
<td>Conflict</td>
<td>The request could not be processed due to a conflict with the current state of the resource. This status code indicates that the resource that the client is attempting to create already exists, or that the request has failed to be processed because of the update of the conflict request.</td>
</tr>
<tr>
<td>410</td>
<td>Gone</td>
<td>The requested resource cannot be found. The status code indicates that the requested resource has been deleted permanently.</td>
</tr>
<tr>
<td>411</td>
<td>Length Required</td>
<td>The server refuses to process the request without a defined Content-Length.</td>
</tr>
<tr>
<td>412</td>
<td>Precondition Failed</td>
<td>The server does not meet one of the preconditions that the requester puts on the request.</td>
</tr>
<tr>
<td>413</td>
<td>Request Entity Too Large</td>
<td>The server is refusing to process a request because the request entity is tool large for the server to process. The server may disable the connection to prevent the client from sending requests consecutively. If the server cannot process the request temporarily, the response will contain a Retry-After header field.</td>
</tr>
<tr>
<td>414</td>
<td>Request-URI Too Large</td>
<td>The Request-URI is too long for the server to process.</td>
</tr>
<tr>
<td>415</td>
<td>Unsupported Media Type</td>
<td>The server does not support the media type in the request.</td>
</tr>
<tr>
<td>416</td>
<td>Requested range not satisfiable</td>
<td>The requested range is invalid.</td>
</tr>
<tr>
<td>417</td>
<td>Expectation Failed</td>
<td>The server fails to meet the requirements of the Expect request-header field.</td>
</tr>
<tr>
<td>422</td>
<td>UnprocessableEntity</td>
<td>The request was well-formed but was unable to be followed due to semantic errors.</td>
</tr>
</tbody>
</table>
### 7.2 Error Code

No data is returned if an API fails to be called. You can locate the cause of an error according to the error code of each API. When the API calling fails, HTTP status code 4xx or 5xx is returned. The returned message body contains the specific error code and error information. If you fail to locate the cause of an error, contact HUAWEI CLOUD service support and provide the error code so that we can help you solve the problem as soon as possible.

- **Sample exception response**
  ```json
  {
    "error_code": "FRS.0202",
    "error_msg": "The service has been freeze."
  }
  ```

- **Parameter description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>error_code</td>
<td>No</td>
<td>String</td>
<td>Error code</td>
</tr>
</tbody>
</table>
### Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>error_msg</td>
<td>No</td>
<td>String</td>
<td>Error message</td>
</tr>
</tbody>
</table>

### Error code description

<table>
<thead>
<tr>
<th>Type</th>
<th>Status Code</th>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common service errors</td>
<td>403</td>
<td>FRS.0002</td>
<td>Token authentication failed.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0010</td>
<td>The request header is missing or empty.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0011</td>
<td>A parameter is missing.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0012</td>
<td>Request parameters are in incorrect formats.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0014</td>
<td>The input is not in JSON format.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0015</td>
<td>An error occurred when parsing the Base64 image.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0016</td>
<td>The format of the uploaded file is not supported.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0017</td>
<td>The uploaded body exceeds the allowed range.</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>FRS.0018</td>
<td>Internal service error.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0019</td>
<td>The service has not been enabled.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0020</td>
<td>The sub-service has not been enabled.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0021</td>
<td>Invalid OBS URL.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0022</td>
<td>Failed to obtain the OBS file.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0023</td>
<td>The size of the file on OBS exceeds the upper limit.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0024</td>
<td>The file does not exist on OBS.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0027</td>
<td>OBS unauthorized.</td>
</tr>
<tr>
<td></td>
<td>403</td>
<td>FRS.0028</td>
<td>The project ID does not match the token.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0029</td>
<td>The format of filter criteria is incorrect.</td>
</tr>
<tr>
<td>Type</td>
<td>Status Code</td>
<td>Error Code</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0030</td>
<td>The returned field is not defined.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0031</td>
<td>The sorting field is not contained in the returned fields.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0032</td>
<td>The sorting field and type are not supported. Only numeral type is supported.</td>
</tr>
<tr>
<td>Service management errors (error code range: 201-300)</td>
<td>400</td>
<td>FRS.0201</td>
<td>The user has been registered.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0202</td>
<td>The service has been frozen.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0203</td>
<td>Failed to update user information.</td>
</tr>
<tr>
<td>Facial image library resource errors (error code range: 301-400)</td>
<td>400</td>
<td>FRS.0301</td>
<td>Failed to create the existing facial image library.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0302</td>
<td>The facial image library does not exist.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0304</td>
<td>No face is detected. Failed to search the face.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0305</td>
<td>A new facial image library cannot be created because the number of facial image libraries exceeds the upper limit.</td>
</tr>
<tr>
<td></td>
<td>403</td>
<td>FRS.0306</td>
<td>New faces cannot be added because the number of faces exceeds the upper limit of a facial image library.</td>
</tr>
<tr>
<td>Face resource errors (error code range: 401-500)</td>
<td>400</td>
<td>FRS.0401</td>
<td>The value of the corresponding type cannot be found and cannot be deleted.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0402</td>
<td>The entered field does not exist.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0403</td>
<td>The face ID does not exist.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0404</td>
<td>No face is detected. Failed to add the face.</td>
</tr>
<tr>
<td>Type</td>
<td>Status Code</td>
<td>Error Code</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0405</td>
<td>The customized field cannot be added because it is not defined.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0406</td>
<td>The imported data type does not match the definition.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0407</td>
<td>No matched data is found during batch deletion.</td>
</tr>
<tr>
<td>Face verification errors</td>
<td>400</td>
<td>FRS.0501</td>
<td>No face is detected in the image.</td>
</tr>
<tr>
<td>(error code range: 501-600)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Face LiveDetect errors</td>
<td>400</td>
<td>FRS.0701</td>
<td>Video parsing error.</td>
</tr>
<tr>
<td>(error code range: 701-800)</td>
<td>400</td>
<td>FRS.0702</td>
<td>The action is not supported.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0703</td>
<td>The action time is invalid.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0704</td>
<td>The number of imported actions exceeds 10.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0705</td>
<td>The matching mode is not supported.</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>FRS.0706</td>
<td>The video duration is not allowed. The duration must be 1 to 15 seconds.</td>
</tr>
<tr>
<td>Face label errors</td>
<td>400</td>
<td>FRS.0100</td>
<td>Check whether any parameter is invalid (for example, empty).</td>
</tr>
<tr>
<td>(error code range: 100-200)</td>
<td>403</td>
<td>FRS.0102</td>
<td>Check whether the input date format is correct.</td>
</tr>
<tr>
<td></td>
<td>403</td>
<td>FRS.0103</td>
<td>Check whether the quality of the input facial image meets the requirements.</td>
</tr>
<tr>
<td></td>
<td>403</td>
<td>FRS.0104</td>
<td>Check whether the input facial image is duplicated.</td>
</tr>
<tr>
<td></td>
<td>404</td>
<td>FRS.0105</td>
<td>Check whether the face_label_id parameter is correctly set.</td>
</tr>
<tr>
<td>Type</td>
<td>Status Code</td>
<td>Error Code</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>403</td>
<td>FRS.0106</td>
<td></td>
<td>The number of front faces in a face set has exceeded the upper limit. Please check the number of members. By default, a face set contains a maximum of 10 facial images. If you need a higher scale, contact the customer service personnel.</td>
</tr>
<tr>
<td>403</td>
<td>FRS.0107</td>
<td></td>
<td>The number of auxiliary faces in a face set has exceeded the upper limit. Please check the number of auxiliary faces. By default, a face set contains a maximum of 50 auxiliary facial images. If you need a higher scale, contact the customer service personnel.</td>
</tr>
<tr>
<td>400</td>
<td>FRS.0108</td>
<td></td>
<td>Check whether the entered OBS bucket name and parameters are correct and whether the agency service is enabled.</td>
</tr>
<tr>
<td>400</td>
<td>FRS.0109</td>
<td></td>
<td>Check value length of the <code>face_set_name</code> parameter. It cannot exceed 24 bytes.</td>
</tr>
<tr>
<td>400</td>
<td>FRS.0110</td>
<td></td>
<td>Check whether the entered value of <code>face_set_name</code> contains invalid characters. The value can contain only letters, digits, underscores (_), and hyphens (-).</td>
</tr>
<tr>
<td>400</td>
<td>FRS.0111</td>
<td></td>
<td>Invalid OBS path. Check whether the entered value of <code>face_obskey</code> is valid. Do not start the path with a <code>/</code> and the path must contain a level-1 directory, for example, <code>pictures/face001.jpg</code>.</td>
</tr>
<tr>
<td>Type</td>
<td>Status Code</td>
<td>Error Code</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>400</td>
<td>FRS.0112</td>
<td></td>
<td>Check the format of images marked by <code>face_obskey</code>. Currently, only JPG, PNG, JPEG, and BMP images are supported.</td>
</tr>
<tr>
<td>400</td>
<td>FRS.0113</td>
<td></td>
<td>Check whether the <code>face_obskey</code> path complies with the face set requirements. If not, you can add the <code>face_set_name</code> parameter to the API for reporting facial images at terminal side.</td>
</tr>
<tr>
<td>400</td>
<td>FRS.0114</td>
<td></td>
<td>Check whether the value of <code>face_date</code> is the current date.</td>
</tr>
<tr>
<td>400</td>
<td>FRS.0115</td>
<td></td>
<td>Check whether the value of <code>query_date</code> is later than the current date.</td>
</tr>
<tr>
<td>400</td>
<td>FRS.0116</td>
<td></td>
<td>Check whether the year of <code>query_date</code> is the same as the year of the current day.</td>
</tr>
<tr>
<td>400</td>
<td>FRS.0117</td>
<td></td>
<td>Check whether the value of <code>query_date</code> meets the format requirements.</td>
</tr>
<tr>
<td>403</td>
<td>FRS.0118</td>
<td></td>
<td>Check whether the <code>query_date</code> exceeds the 30-day limitation.</td>
</tr>
<tr>
<td>403</td>
<td>FRS.0119</td>
<td></td>
<td>Check whether the size of the image exceeds 8 MB.</td>
</tr>
<tr>
<td>403</td>
<td>FRS.0120</td>
<td></td>
<td>The system is busy. Check whether APIs are frequently called. If service requests increase, contact Huawei technical support for service expansion.</td>
</tr>
<tr>
<td>403</td>
<td>FRS.0121</td>
<td></td>
<td>Check whether the face posture and angle of the input facial image meet the requirements.</td>
</tr>
</tbody>
</table>
### Gateway error code

<table>
<thead>
<tr>
<th>Type</th>
<th>Status Code</th>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway errors</td>
<td>404</td>
<td>APIG.0101</td>
<td>The API does not exist.</td>
</tr>
<tr>
<td>(error code range:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-400)</td>
<td>413/504</td>
<td>APIG.0201</td>
<td>The request body exceeds the allowed range or the backend times out.</td>
</tr>
<tr>
<td></td>
<td>401</td>
<td>APIG.0301</td>
<td>Authentication failed.</td>
</tr>
</tbody>
</table>

### 7.3 Obtaining the Project ID, Account Name, and AK/SK

- In **Authentication**, you need to enter the account name, username, or AK/SK.
- A project ID is required for some URLs when an API is called.

To obtain a project ID (example), perform the following operations:

**Obtaining a Project ID from the Console**

1. Log in to the [Face Recognition Management Console](#).
2. Move the cursor over your username in the upper right corner and click **My Credentials** from the drop-down list.
3. On the displayed **My Credentials** page, view the username, account name, or project ID.
4. Click **Access Keys** to view the added AK.
   - If no AK/SK pair is available, click **Create Access Key** to add one.
   - If you have generated the AK/SK, find the AK/SK file you downloaded. Generally, the file name is **credentials.csv**.

### Obtaining a Project ID by Calling an API

A project ID can also be obtained by calling a specific API. For details, see [Querying Project Information Based on the Specified Criteria](#).

The API for obtaining a project ID is **GET https://{Endpoint}/v3/projects/**. `{Endpoint}` indicates the endpoint of IAM, which can be obtained from [Regions and Endpoints](#). For details about API authentication, see [Authentication](#).

The following is an example response. The value of `id` is the project ID.

```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180d14e684b",
      "is_domain": false,
      "parent_id": "65382450e8f64ac0870cd180d14e684b",
      "name": "project_name",
      "description": "",
      "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/a4a5d4098fa4474fa22cd05f897d6b99"
      },
      "id": "a4a5d4098fa4474fa22cd05f897d6b99",
      "enabled": true
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "https://www.example.com/v3/projects"
  }
}
<table>
<thead>
<tr>
<th>Release Date</th>
<th>Description</th>
</tr>
</thead>
</table>
| 2019-10-16   | This is the sixth official release. Added the following topics:  
  ● **Face Recognition APIs (V2)** |
| 2018-05-23   | This is the fifth official release.  
  ● Adjusted the document content structure.  
  ● Added restrictions on each API.  
  ● Added the following topics:  
    – **Endpoints**  
    – **Restrictions and Limitations**  
    – **Concepts**  
    – **Status Codes** |
| 2018-12-18   | This issue is the fourth official release and incorporates the following change:  
  Added the following APIs and related error codes:  
  ● **Updating a Face**  
  ● **Batch Deleting Faces** |
<table>
<thead>
<tr>
<th>Release Date</th>
<th>Description</th>
</tr>
</thead>
</table>
| 2018-12-12   | This issue is the third official release and incorporates the following change: Added the Face Label APIs and the following topics:  
  ● 7.2.1-Adding Facial Images of Members  
  ● 7.2.2-Querying Facial Images of Members  
  ● 7.2.3-Deleting Facial Images of Members  
  ● 7.2.4-Reporting Faces Captured at Terminal Side  
  ● 7.2.5-Querying Label ID List of Member Faces  
  ● 7.2.6-Querying Label ID List of Unfamiliar Faces  
  ● 7.2.7-Querying Face Label Content  
  ● 7.2.8-Querying Human Figure Sets |
| 2018-09-30   | This issue is the second official release and incorporates the following change:  
  ● Added the following topics:  
    - API Overview  
    - Face LiveDetect  
    - Landmark  
    - Attributes  
    - VideoDetectResult  
    - ServiceInfo  
  ● Deleted the following topics:  
    - APIs for Enabling Services  
    - APIs for Disabling Services  
    - Status Code  
  ● Updated some API parameters and examples and adjusted the document structure. |
| 2018-04-30   | The issue is the first official release. |