Face Recognition Service

API Reference

Issue	01
Date	2023-04-25

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

HUAWEI 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.

Contents

1 Before You Start

1.1 Overview

Face Recognition Service (FRS) quickly detects faces in images, obtains face attributes, and implements face comparison and search.

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

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

1.2 API Calling

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

Additionally, FRS provides a Java SDK. For details about how to use the SDK, see **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 FRS endpoints

Region	Endpoint Region	Endpoint	Protocol	Deployed Service
CN-Hong Kong	ap-southeast-1	face.ap-southeast-1.myhuaweicloud.com	HTTPS	FRS APIs
AP-Bangkok	ap-southeast-2	face.ap-southeast-2.myhuaweicloud.com	HTTPS	FRS APIs

1.4 Restrictions and Limitations

Due to various factors such as technology and cost, FRS 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. Each library contains a maximum of 100,000 facial features.
- 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

- 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 size is too large, the image transmission will take a long time. It is recommended that the image size be **less than 1 MB**.
- The image resolution must be **less than 4096 x 2160**. 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 and 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. Do not directly use an account for routine management, but create users and assign them permissions for that.
- User

  A user is created on in IAM using an account to use cloud services. Each user has its own identity credentials (password and access keys).

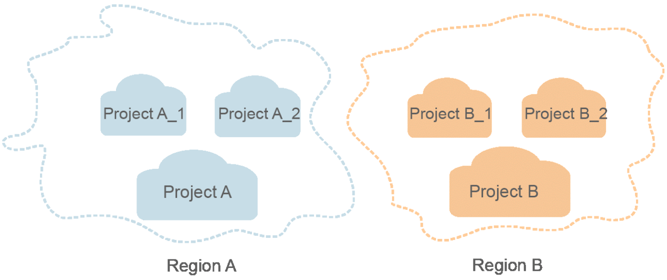
  You can check the account ID and user ID on the **My Credentials** page of the console. The account name, username, and password will be required for API authentication.
- Region

  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.
- 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



2 API Overview

APIs provided by FRS comply with RESTful API design specifications. **Table 2-1** lists the APIs.

Table 2-1 FRS APIs

Type	API	Description
Face Detection	Face detection	This API detects and analyzes faces in an input image and outputs the positions of the faces in the image. If an image contains multiple faces, all facial features that meet the search criteria are returned.
Face Verification	Face verification	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.
Face LiveDetect	Face live detecting	This API is used to determine 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.
Face Retrieval	Face retrieval	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.

Type	API	Description
Facial Image Library Management	Creating a facial image library	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.
	Querying all facial image libraries	This API is used to query the statuses of all facial image libraries of the current user.
	Querying a facial image library	This API is used to query the status of a facial image library.
	Deleting a facial image library	This API is used to delete a facial image library and all faces in the library.
Facial Resource Management	Adding a face	This API is used to add faces to a facial image library. All detected faces in the input facial image will be added to the library.
	Querying a face	This API is used to query the face information in a specified facial image library.
	Updating a face	This API is used to update a single face based on its face ID (**face_id**).
	Deleting a face	This API is used to delete a face from a facial image library based on a specified field.
	Deleting faces in batches	This API is used to batch delete multiple faces that meet specified criteria you customize.

3 Calling APIs

3.1 Applying for FRS

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

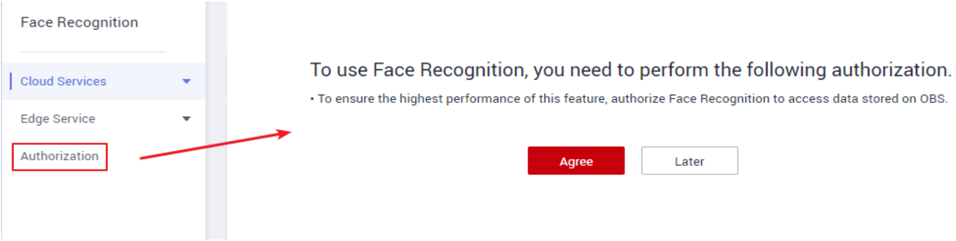
Procedure

1. To subscribe to and use FRS APIs for the first time, **contact sales** to get assistance.
2. Select the sub-service to be enabled based on service requirements, for example, **Face Detection**, and click **Enable Service** on the right.

NOTE

- Face Recognition may need to use the data stored on OBS, so you need to authorize Face Recognition to access data stored on OBS. In the left navigation pane, choose **Authorization**, and click **Agree**. The **Agree** button changes to **Authorized**.
- 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.

Figure 3-1 Service authorization



To use Face Recognition, you need to perform the following authorization.

• To ensure the highest performance of this feature, authorize Face Recognition to access data stored on OBS.

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}

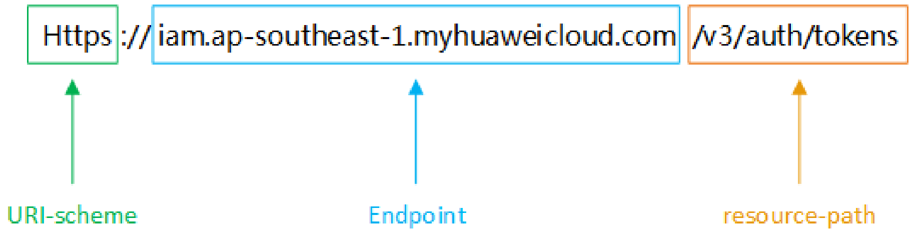
Table 3-1 Request URI

Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	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-Hong Kong** region is **iam.ap-southeast-1.myhuaweicloud.com**.
resource-path	Access path of an API for performing a specified operation. You can obtain the value of this parameter from the URI of the specific API. For example, the **resource-path** of the API used to obtain a user token is **/v3/auth/tokens**.
query-string	Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of *"Parameter name=Parameter value"*. For example, **? limit=10** indicates that a maximum of 10 data records will be displayed.

For example, to obtain an IAM token in the **CN-Hong Kong** region, obtain the endpoint of IAM (**iam.ap-southeast-1.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:

```
https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
```

Figure 3-2 Example URI



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

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

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

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
```

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 3-3 Common request header fields

Parameter	Description	Mandatory	Example Value
Content-type	Request body type or format. Its default value is **application/json**.	Yes	application/json

Parameter	Description	Mandatory	Example Value
X-Project-Id	Project ID, which is used to obtain the token of a project. For how to obtain the project ID, see **Obtaining the Project ID/Account Name/AK/SK**.	No	e9993fc787d94b6c886cbaa340f9c0f4
X-Auth-Token	User token, a response to the API used to **obtain a user token**. This API is the only one that does not require authentication.	This parameter is mandatory for token authentication.	-
X-Sdk-Date	Time when the request is sent. The time is in **YYYYMMDD'T'HHMMSS'Z'** format. The value is the current Greenwich Mean Time (GMT) of the system.	This parameter is mandatory for AK/SK-based authentication, but not required for PKI token-based authentication	20190307T101459Z
Authorization	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	This parameter is mandatory for AK/SK-based authentication. For details about AK/SK-based authentication, see **Authentication**.	SDK-HMAC-SHA256 Credential=ZIRRKMTWPTQFQI1WKNKB/20150907//ec2/sdk_request, SignedHeaders=content-type;host;x-sdk-date, Signature=55741b610f3c9fa3ae40b5a8021ebf7ebc2a28a603fc62d25cb3bfe6608e1994

Parameter	Description	Mandatory	Example Value
Host	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**.	This parameter is mandatory for AK/SK-based authentication.	code.test.com or code.test.com:443

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:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
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 provides an example request with a body included. Replace *username*, *domainname*, ******** (login password), and *xxxxxxxx* (project name) with the actual values and obtain the token specified for the project. For example, if the project name is set to ap-southeast-1, the obtained token applies to the services in ap-southeast-1. For details about how to obtain a username, domainname, and project name, see **Obtaining Account, IAM User, Group, Project, Region, and Agency Information**.

NOTE

- The **scope** parameter specifies where a token takes effect. Its value can be **project** or **domain**. In the preceding example, the value of **scope** is **project**, indicating that the obtained token can access only resources of the specified project. If the value of **scope** is **domainname**, the obtained token can access all resources of the specified account. For details, see **Obtaining a User Token**.
- **project name**: On the **My Credentials** page, the value in the **Project** column is the project name. For details about how to obtain the project name, see **Obtaining Account, IAM User, Group, Project, Region, and Agency Information**.

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
{
  "auth": {
    "identity": {
      "methods": ["password"],
      "password": {
        "user": {
          "name": "username",
          "password": "**********",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "ap-southeast-1"
      }
    }
  }
}
```

If all data required for the API request is available, you can send the request to call the API through curl, Postman, or coding. If you choose to use Postman, version 7.24.0 is recommended. In the response to the API used to **obtain a user token**, **x-subject-token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

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.

When calling the API to **obtain a user token**, you must set **auth.scope** in the request body to **project**.

```
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",
                    "password": "********",
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "xxxxxxxx"
            }
        }
    }
}
```

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:

```

GET https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/projects
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

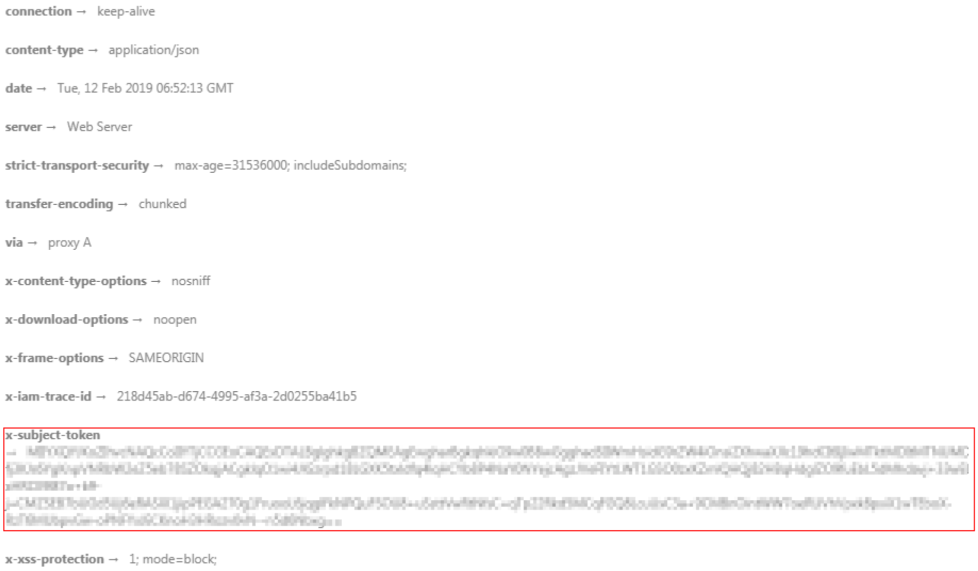
Parameter	Description	Mandatory
Content-Type	Media type of the message body sent to a receiver Type: string Default value: **application/json; charset=UTF-8**	Yes
X-request-id	This field carries the request ID for task tracing. Type: string. **request_id-timestamp-hostname** (The **request_id** is the UUID generated on the server. **timestamp** indicates the current timestamp, and **hostname** is the name of the server that processes the current API.) Default value: none	No
X-ratelimit	This field carries the total number of flow control requests. Type: integer Default value: none	No

Parameter	Description	Mandatory
X-ratelimit-used	This field carries the number of remaining requests. Type: integer Default value: none	No
X-ratelimit-window	This field carries the flow control unit. Type: string. The unit is minute, hour, or day. Default value: hour	No

Figure Header fields of the response to the request for obtaining a user token 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.

Figure 3-3 Header fields of the response to the request for obtaining a user token



```
connection → keep-alive

content-type → application/json

date → Tue, 12 Feb 2019 06:52:13 GMT

server → Web Server

strict-transport-security → max-age=31536000; includeSubdomains;

transfer-encoding → chunked

via → proxy A

x-content-type-options → nosniff

x-download-options → noopen

x-frame-options → SAMEORIGIN

x-iam-trace-id → 218d45ab-d674-4995-af3a-2d0255ba41b5

x-subject-token
→ [redacted]

x-xss-protection → 1; mode=block;
```

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**.

```
{
    "token": {
        "expires_at": "2019-02-13T06:52:13.855000Z",
        "methods": [
            "password"
        ],
        "catalog": [
            {
                "endpoints": [
                    {
                        "region_id": "ap-southeast-1",
......
```

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

```
{
    "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 Codes**.

4 APIs

4.1 Face Detection

Function

This API detects and analyzes faces in an input image and outputs the positions of the faces in the image. If an image contains multiple faces, all facial features that meet the search criteria are returned.

Prerequisites:

Ensure that you have enabled **FRS**. For detailed operations, see **Applying for FRS**.

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 enough.
- 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

POST /v2/{project_id}/face-detect

Table 4-1 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the ID, see **Obtaining the Project ID/ Account Name/AK/SK**.

Request Parameters

Table 4-2 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. During API authentication using a token, the token is added to requests to obtain permissions for calling the API. The value of **X-Subject-Token** in the response header is the obtained token.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

Table 4-3 Request body parameters

Parameter	Mandatory	Type	Description
image_url	Either **image_url**, **image_file**, or **image_base64** is mandatory.	String	Image URL. Currently, only the URL of an OBS bucket on Huawei Cloud is supported and FRS must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see **Service Authorization**.
image_file	Either **image_file**, **image_url**, or **image_base64** is mandatory.	File	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.
image_base64	Either **image_base64**, **image_file**, or **image_url** is mandatory.	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.

Response Parameters

Status code: 200

Table 4-4 Response body parameter

Parameter	Type	Description
faces	Array of **DetectFace** objects	Detected face. This parameter is not included when the API fails to be called.

Table 4-5 DetectFace

Parameter	Type	Description
bounding_box	**BoundingBox** object	Position of a face in an image

Table 4-6 BoundingBox

Parameter	Type	Description
width	Integer	Width of a rectangle
top_left_y	Integer	Vertical coordinate of the upper-left corner of a rectangle
top_left_x	Integer	Horizontal coordinate of the upper-left corner of a rectangle
height	Integer	Height of a rectangle

Status code: 400

Table 4-7 Response body parameters

Parameter	Type	Description
error_code	String	Error code when calling the API failed. This parameter is not included when the API is successfully called.
error_msg	String	Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.

Example Requests

For details about how to obtain the value of **X-Auth-Token**, see **Authentication**.

- Example 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:
{
    "image_base64": "/9j/4AAQSkZJRgABAgEASABIAAD"

}
```

- Example request (Method 2: Use an image file.)

```
POST https://{endpoint}/v2/{project_id}/face-detect
Request Header:
```

```
X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

Request Body:
  image_file: File (image file)
```

- Example 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:
{
    "image_url": "/BucketName/ObjectName"

}
```

Example Responses

Status code: 200

Response example (successful request)

```
{
  "faces": [
    {
      "bounding_box": {
        "width": 174,
        "top_left_y": 37,
        "top_left_x": 22,
        "height": 174
      }
    }
  ]
}
```

Status code: 400

Example response (failed request)

```
{
  "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 Codes**.

4.2 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.

Prerequisites:

Ensure that you have enabled **FRS**. For detailed operations, see **Applying for FRS**.

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 enough.
- 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

POST /v2/{project_id}/face-compare

Table 4-8 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the ID, see **Obtaining the Project ID/ Account Name/AK/SK**.

Request Parameters

Table 4-9 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. During API authentication using a token, the token is added to requests to obtain permissions for calling the API. The value of **X-Subject-Token** in the response header is the obtained token.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

Table 4-10 Request body parameters

Parameter	Type	Mandatory	Description
image1_url	String	Either **image1_url**, **image1_file**, or **image1_base64** is mandatory.	Image URL. Currently, only the URL of an OBS bucket on Huawei Cloud is supported and FRS must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see **Service Authorization**.
image1_file	File	Either **image1_file**, **image1_url**, or **image1_base64** is mandatory.	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.

Parameter	Type	Mandatory	Description
image1_base64	String	Either **image1_base64**, **image1_url**, or **image1_file** is mandatory.	Image data (Base64-encoded). The value must meet the following requirements: ● 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.
image2_url	String	Either **image2_url**, **image2_file**, or **image2_base64** is mandatory.	Image URL. Currently, only the URL of an OBS bucket on Huawei Cloud is supported and FRS must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see **Service Authorization**.
image2_file	File	Either **image2_file**, **image2_url**, or **image2_base64** is mandatory.	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.
image2_base64	String	Either **image2_base64**, **image2_url**, or **image2_file** is mandatory.	Image data (Base64-encoded). The value must meet the following requirements: ● 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 Parameters

Status code: 200

Table 4-11 Response body parameters

Parameter	Type	Description
image1_face	**CompareFace** object	Face detected in the first image. For details, see **DetectFace**. This parameter is not included when the API fails to be called.
image2_face	**CompareFace** object	Face detected in the second image. For details, see **DetectFace**.
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 **0.93**, the faces in two images belong to one person. This parameter is not included when the API fails to be called.

Table 4-12 CompareFace

Parameter	Type	Description
bounding_box	**BoundingBox** object	Position of a face in an image

Table 4-13 BoundingBox

Parameter	Type	Description
width	Integer	Width of a rectangle
top_left_y	Integer	Vertical coordinate of the upper-left corner of a rectangle
top_left_x	Integer	Horizontal coordinate of the upper-left corner of a rectangle
height	Integer	Height of a rectangle

Status code: 400

Table 4-14 Response body parameters

Parameter	Type	Description
error_code	String	Error code when calling the API failed. This parameter is not included when the API is successfully called.

Parameter	Type	Description
error_msg	String	Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.

Example Requests

For details about how to obtain the value of **X-Auth-Token**, see **Authentication**.

- Example 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: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

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

- Example request (Method 2: Use an image file.)

```
POST https://{endpoint}/v2/{project_id}/face-compare
Request Header:
X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

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

- Example request (Method 3: Use the image URL.)

```
POST https://{endpoint}/v2/{project_id}/face-compare
Request Header:
Content-Type: application/json
X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

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

Example Responses

Status code: 200

Response example (successful request)

```
{
  "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
    }
  }
}
```

Status code: 400

Example response (failed request)

```
{
  "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 Codes**.

4.3 Face LiveDetect

Function

This API is used to determine 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.

Prerequisites:

Ensure that you have enabled **FRS**. For detailed operations, see **Applying for FRS**.

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

POST /v2/{project_id}/live-detect

Table 4-15 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the ID, see **Obtaining the Project ID/ Account Name/AK/SK**.

Request Parameters

Table 4-16 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. During API authentication using a token, the token is added to requests to obtain permissions for calling the API. The value of **X-Subject-Token** in the response header is the obtained token.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

Table 4-17 Request body parameters

Parameter	Mandatory	Type	Description
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 FRS 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.

Parameter	Mandatory	Type	Description
video_file	Either **video_file**, **video_url**, or **video_base64** is mandatory.	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.

Parameter	Mandatory	Type	Description
video_base64	Either **video_base64**, **video_url**, or **video_file** is mandatory.	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**: Turning the face to the left ● **2**: Turning the face to the right ● **3**: Head nodding ● **4**: Mouth movement

Parameter	Mandatory	Type	Description
action_time	No	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.
nod_threshold	No	Double	Threshold for determining the amplitude of a nod action, in degrees. The value ranges from 1 to 90. The default value is **10**. A larger value indicates more difficulties in determining whether an action is a nod. This parameter is not supported for Hong Kong sites.

Response Parameters

Status code: 200

Table 4-18 Response body parameters

Parameter	Type	Description
video-result	**video-result** object	LiveDetect result. For details, see **VideoDetectResult**.
warning-list	Array of **WarningList** objects	Warning information list. For details, see **WarningList**.

Table 4-19 video-result

Parameter	Type	Description
alive	Boolean	Whether a living figure is detected

Parameter	Type	Description
actions	Array of **ActionsList** objects	Action list
picture	String	Base64 code of the image in which the maximum face is detected

Table 4-20 ActionsList

Parameter	Type	Description
confidence	Double	Confidence level. The value ranges from 0 to 1.
action	Integer	Action ID. Possible values are as follows: **1**: turning face to the left; **2**: turning face to the right; **3**: head nodding; **4**: mouth movement.

Table 4-21 WarningList

Parameter	Type	Description
warningCode	Integer	Warning ID
warningMsg	String	Warning message

Status code: 400

Table 4-22 Response body parameters

Parameter	Type	Description
error_code	String	Error code when calling the API failed. This parameter is not included when the API is successfully called.
error_msg	String	Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.

Example Requests

- Example request (Method 1: Use the Base64 encoded string of a video.)

```
POST https://{endpoint}/v2/{project_id}/live-detect
 Request Header:
 Content-Type: application/json
 X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...
 Request Body:
 {
    "video_base64":"/9j/4AAQSkZJRgABAgEASABIAAD",
    "actions":"1,3,2",
    "action_time":"1000-3000,4000-7000,9000-12000"
 }
```

- Example request (Method 2: Use a video file.)

```
POST https://{endpoint}/v2/{project_id}/live-detect
 Request Header:
 X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

 Request Body:
 video_file: File (Video file)
 actions: 1,3,2
 action_time: 1000-3000,4000-7000,9000-12000
```

- Example request (Method 3: Use the video URL.)

```
POST https://{endpoint}/v2/{project_id}/live-detect
 Request Header:
 Content-Type: application/json
 X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

 Request Body:
 {
    "video_url":"/BucketName/ObjectName",
    "actions":"1,3,2",
    "action_time":"1000-3000,4000-7000,9000-12000"
 }
```

Example Responses

Status code: 200

Example response for a successful request

```
{
  "video-result": {
    "alive": true,
    "actions": [{
        "action": 1,
        "confidence": 0.823
      },{
        "action": 3,
        "confidence": 0.823
      },{
        "action": 2,
        "confidence": 0.823
      }],
    "picture": "/9j/4AAQSkZJRgABAQEAYABgAAD/2w..."
  },
  "warning-list": []
}
```

Status code: 400

Example response for a failed request

```
{
   "error_code": "FRS.0701",
   "error_msg": "Parse video data failed."
 }
```

Status Codes

See **Status Codes**.

Error Codes

See **Error Codes**.

4.4 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.

Prerequisites:

Ensure that you have enabled **FRS**. For detailed operations, see **Applying for FRS**.

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 enough.
- 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

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

Table 4-23 Path parameters

Parameter	Type	Mandatory	Description
project_id	String	Yes	Project ID. For details about how to obtain the ID, see **Obtaining the Project ID/ Account Name/AK/SK**.
face_set_name	String	Yes	Name of a facial image library.

Request Parameters

Table 4-24 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. During API authentication using a token, the token is added to requests to obtain permissions for calling the API. The value of **X-Subject-Token** in the response header is the obtained token.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

Table 4-25 Request body parameters

Parameter	Type	Mandatory	Description
image_url	String	Either **image_url**, **image_file**, **image_base64**, or **face_id** is mandatory.	Image URL. Currently, only the URL of an OBS bucket on Huawei Cloud is supported and FRS must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see **Service Authorization**.

Parameter	Type	Mandatory	Description
image_file	File	Either **image_file**, **image_url**, **image_base64**, or **face_id** is mandatory.	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.
image_base64	String	Either **image_base64**, **image_url**, **image_file**, or **face_id** is mandatory.	Image data (Base64-encoded). The value must meet the following requirements: ● 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.
face_id	String	Either **face_id**, **image_url**, **image_file**, or **image_base64** is mandatory.	Face ID returned by the system after a face is imported
top_n	Integer	No	*N* faces returned that are most similar to the input one. The default value of **N** is **10**. The returned faces are sorted by confidence in descending order.
threshold	Double	No	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**.
sort	JSONArray	No	Field sorting. For details, see **Sort Syntax**.
filter	String	No	Filtering criteria. For details, see **Filter Syntax**.
return_fields	JsonArray	No	Returned customized field.

Response Parameters

Status code: 200

Table 4-26 Response body parameter

Parameter	Type	Description
faces	Array of **SearchFace** objects	Face set to be retrieved. For details, see **SearchFace**.

Table 4-27 SearchFace

Parameter	Type	Description
bounding_box	**BoundingBox** object	Position of a face in an image
similarity	Double	Similarity degree in Face Retrieval
external_fields	Object	Additional field a user customizes
external_image_id	String	ID of the external image to which a face belongs
face_id	String	Face ID, which is a unique ID generated by the system

Table 4-28 BoundingBox

Parameter	Type	Description
width	Integer	Width of a rectangle
top_left_y	Integer	Vertical coordinate of the upper-left corner of a rectangle
top_left_x	Integer	Horizontal coordinate of the upper-left corner of a rectangle
height	Integer	Height of a rectangle

Status code: 400

Table 4-29 Response body parameters

Parameter	Type	Description
error_code	String	Error code when calling the API failed. This parameter is not included when the API is successfully called.
error_msg	String	Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.

Example Requests

For details about how to obtain the value of **X-Auth-Token**, see **Authentication**.

- Example 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: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

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

- Example request (Method 2: Use an image file.)

```
POST https://{endpoint}/v2/{project_id}/face-sets/showFaceSet/search
Request Header:
X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

Request Body:
image_file: File (image file)
return_fields: ["timestamp","id"]
filter: timestamp:12
```

- Example 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: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

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

- Example 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: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

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

Example Responses

Status code: 200

Response example (successful request)

```
{
  "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"
    }
  ]
}
```

Status code: 400

Example response (failed request)

```
{
  "error_code": "FRS.0018",
  "error_msg": "The service inner error."
}
```

Status Code

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

Error Code

For details about the error code, see **Error Codes**.

4.5 Facial Image Library Management

4.5.1 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.

Prerequisites:

Ensure that you have enabled **FRS**. For detailed operations, see **Applying for FRS**.

NOTE

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

URI

POST /v2/{project_id}/face-sets

Table 4-30 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the ID, see **Obtaining the Project ID/ Account Name/AK/SK**.

Request Parameters

Table 4-31 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. During API authentication using a token, the token is added to requests to obtain permissions for calling the API. The value of **X-Subject-Token** in the response header is the obtained token.

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

Table 4-32 Request body parameters

Parameter	Type	Mandatory	Description
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 **100000**, and the maximum value is **100000**. Each user can use 10 facial image libraries. Each library contains 100,000 facial features.
external_fields	Map<String,**TypeInfo**>	No	Custom data. Custom fields cannot be named **vector**, **bounding_box**, **external_image_id**, **face_id**, **create_time**, **_id**, **_all** or **_source**, which are built-in fields. The value of a custom field can contain 1 to 36 characters, while the value of a string field can contain 1 to 256 characters. For details, see **Customized Fields**. If you want to use custom fields to add faces to the facial image library in subsequent operations, you need to define this field when creating the facial image library.

Table 4-33 TypeInfo

Parameter	Mandatory	Type	Description
type	No	String	Data type. The value can be **string**, **integer**, **double**, or **long**.

Response Parameters

Status code: 200

Table 4-34 Response body parameter

Parameter	Type	Description
face_set_info	**FaceSetInfo** object	Facial image library information. For details, see **FaceSetInfo**. This parameter is not included when the API fails to be called.

Table 4-35 FaceSetInfo

Parameter	Type	Description
face_number	Integer	Number of existing facial features in a facial image library
external_fields	Object	Additional field a user customizes
face_set_id	String	ID of the facial image library. It is a randomly generated string containing eight characters.
face_set_name	String	Name of the facial image library
create_date	String	Creation time
face_set_capacity	Integer	Maximum capacity of a facial image library. If **face_set_capacity** is not specified when a face image library is created, the face image library can contain a maximum of 100,000 facial features by default.

Status code: 400

Table 4-36 Response body parameters

Parameter	Type	Description
error_code	String	Error code when calling the API failed. This parameter is not included when the API is successfully called.
error_msg	String	Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.

Example Requests

For details about how to obtain the value of **X-Auth-Token**, see **Authentication**.

- Example request

```
POST https://{endpoint}/v2/{project_id}/face-sets
Request Header:
Content-Type: application/json
X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

Request Body:
{
  "face_set_name": "test",
  "face_set_capacity": 100000,
  "external_fields" : {
    "timestamp" : {
      "type" : "long"
    },
    "id" : {
      "type" : "string"
    },
    "number" : {
      "type" : "integer"
    }
  }

}
```

Example Responses

Status code: 200

Response example (successful request)

```
{
  "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": 100000,
    "external_fields": {
      "timestamp" : {
        "type" : "long"
      },
      "id" : {
        "type" : "string"
      },
      "number" : {
        "type" : "integer"
      }
  }
 }

}
```

Status code: 400

Example response (failed request)

```
{
 "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 Codes**.

4.5.2 Querying All Facial Image Libraries

Function

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

Prerequisites:

Ensure that you have enabled **FRS**. For detailed operations, see **Applying for FRS**.

URI

GET /v2/{project_id}/face-sets

Table 4-37 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the ID, see **Obtaining the Project ID/ Account Name/AK/SK**.

Request Parameters

Table 4-38 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. During API authentication using a token, the token is added to requests to obtain permissions for calling the API. The value of **X-Subject-Token** in the response header is the obtained token.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

Response Parameters

Status code: 200

Table 4-39 Response body parameter

Parameter	Type	Description
face_sets_info	Array of **FaceSetInfo** objects	Facial image library information. For details, see **FaceSetInfo**. This parameter is not included when the API fails to be called.

Table 4-40 FaceSetInfo

Parameter	Type	Description
face_number	Integer	Number of existing facial features in a facial image library
external_fields	Object	Additional field a user customizes
face_set_id	String	ID of the facial image library. It is a randomly generated string containing eight characters.
face_set_name	String	Name of the facial image library
create_date	String	Creation time

Parameter	Type	Description
face_set_capacity	Integer	Maximum capacity of a facial image library. If **face_set_capacity** is not specified when a face image library is created, the face image library can contain a maximum of 100,000 facial features by default.

Status code: 400

Table 4-41 Response body parameters

Parameter	Type	Description
error_code	String	Error code when calling the API failed. This parameter is not included when the API is successfully called.
error_msg	String	Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.

Example Requests

For details about how to obtain the value of **X-Auth-Token**, see **Authentication**.

- Example request

```
GET https://{endpoint}/v2/{project_id}/face-sets
Request Header:
Content-Type: application/json
X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...
```

Example Responses

Status code: 200

Response example (successful request)

```
{
  "face_sets_info": [
    {
      "face_number": 0,
      "face_set_id": "ylXMMZTO",
      "face_set_name": "test",
      "create_date": "2018-05-11 07:49:40",
      "face_set_capacity": 100000,
      "external_fields": {
        "number": {
          "type": "integer"
        },
        "id": {
          "type": "string"
        },
        "timestamp": {
          "type": "long"
        }
      }
    }
  ]

}
```

Status code: 400

Example response (failed request)

```
{
  "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 Codes**.

4.5.3 Querying a Facial Image Library

Function

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

Prerequisites:

Ensure that you have enabled **FRS**. For detailed operations, see **Applying for FRS**.

URI

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

Table 4-42 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the ID, see **Obtaining the Project ID/ Account Name/AK/SK**.
face_set_name	Yes	String	Name of a facial image library.

Request Parameters

Table 4-43 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. During API authentication using a token, the token is added to requests to obtain permissions for calling the API. The value of **X-Subject-Token** in the response header is the obtained token.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

Response Parameters

Status code: 200

Table 4-44 Response body parameter

Parameter	Type	Description
face_set_info	**FaceSetInfo** object	Facial image library information. For details, see **FaceSetInfo**. This parameter is not included when the API fails to be called.

Table 4-45 FaceSetInfo

Parameter	Type	Description
face_number	Integer	Number of faces in a facial image library
external_fields	Object	Additional field a user customizes
face_set_id	String	ID of the facial image library. It is a randomly generated string containing eight characters.
face_set_name	String	Name of the facial image library
create_date	String	Creation time
face_set_capacity	Integer	Maximum capacity of a facial image library

Status code: 400

Table 4-46 Response body parameters

Parameter	Type	Description
error_code	String	Error code when calling the API failed. This parameter is not included when the API is successfully called.
error_msg	String	Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.

Example Requests

For details about how to obtain the value of **X-Auth-Token**, see **Authentication**.

- Example request
```
GET https://{endpoint}/v2/{project_id}/face-sets/showFaceSet
Request Header:
Content-Type: application/json
X-Auth-Token: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...
```

Example Responses

Status code: 200

Response example (successful request)

```
{
  "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": 100000,
    "external_fields": {
      "number": {
        "type": "integer"
      },
      "id": {
        "type": "string"
      },
      "timestamp": {
        "type": "long"
      }
    }
  }
}
```

Status code: 400

Example response (failed request)

```
{
  "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 Codes**.

4.5.4 Deleting a Facial Image Library

Function

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

Prerequisites:

Ensure that you have enabled **FRS**. For detailed operations, see **Applying for FRS**.

URI

DELETE /v2/{project_id}/face-sets/{face_set_name}

Table 4-47 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the ID, see **Obtaining the Project ID/Account Name/AK/SK**.
face_set_name	Yes	String	Name of a facial image library.

Request Parameters

Table 4-48 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. During API authentication using a token, the token is added to requests to obtain permissions for calling the API. The value of **X-Subject-Token** in the response header is the obtained token.

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

Response Parameters

Status code: 200

Table 4-49 Response body parameter

Parameter	Type	Description
face_set_name	String	Name of the facial image library. This parameter is not included when the API fails to be called.

Status code: 400

Table 4-50 Response body parameters

Parameter	Type	Description
error_code	String	Error code when calling the API failed. This parameter is not included when the API is successfully called.
error_msg	String	Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.

Example Requests

For details about how to obtain the value of **X-Auth-Token**, see **Authentication**.

- Example request

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

Example Responses

Status code: 200

Response example (successful request)

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

Status code: 400

Example response (failed request)

```
{
  "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 Codes**.

4.6 Facial Resource Management

4.6.1 Adding a Face

Function

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

Prerequisites:

Ensure that you have enabled **FRS**. For detailed operations, see **Applying for FRS**.

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 enough.
- 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

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

Table 4-51 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the ID, see **Obtaining the Project ID/ Account Name/AK/SK**.
face_set_name	Yes	String	Name of a facial image library.

Request Parameters

Table 4-52 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. During API authentication using a token, the token is added to requests to obtain permissions for calling the API. The value of **X-Subject-Token** in the response header is the obtained token.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

Table 4-53 Request body parameters

Parameter	Type	Mandatory	Description
image_url	String	Either **image_url**, **image_file**, or **image_base64** is mandatory.	Image URL. Currently, only the URL of an OBS bucket on Huawei Cloud is supported and FRS must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see **Service Authorization**.
image_file	File	Either **image_file**, **image_url**, or **image_base64** is mandatory.	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.
image_base64	String	Either **image_base64**, **image_file**, or **image_url** is mandatory.	Image data (Base64-encoded). The value must meet the following requirements: ● 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.
external_image_id	String	No	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.
external_fields	Object	No	Custom data You need to define this field when creating a facial image library so that you can use custom fields to add faces. For details, see **Customized Fields**.

Parameter	Type	Mandatory	Description
single	boolean	No	Whether to add the largest face in the image to the library. The options are as follows: ● **true**: Only the largest face in the facial image is added to the library. ● **false** (default value): **true**: All faces in the image are added to the library.

Response Parameters

Status code: 200

Table 4-54 Response body parameters

Parameter	Type	Description
face_set_id	String	ID of the facial image library. This parameter is not included when the API fails to be called.
face_set_name	String	Name of the facial image library. This parameter is not included when the API fails to be called.
faces	Array of **FaceSetFace** objects	Face structure in the facial image library. For details, see **FaceSetFace**. This parameter is not included when the API fails to be called.

Table 4-55 FaceSetFace

Parameter	Type	Description
bounding_box	**BoundingBox** object	Position of a face in an image. For details about the BoundingBox structure, see **BoundingBox**.
external_fields	Object	Additional field a user customizes
external_image_id	String	ID of the external image to which a face belongs
face_id	String	Face ID, which is a unique ID generated by the system

Table 4-56 BoundingBox

Parameter	Type	Description
width	Integer	Width of a rectangle
top_left_y	Integer	Vertical coordinate of the upper-left corner of a rectangle
top_left_x	Integer	Horizontal coordinate of the upper-left corner of a rectangle
height	Integer	Height of a rectangle

Status code: 400

Table 4-57 Response body parameters

Parameter	Type	Description
error_code	String	Error code when calling the API failed. This parameter is not included when the API is successfully called.
error_msg	String	Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.

Example Requests

For details about how to obtain the value of **X-Auth-Token**, see **Authentication**.

- Example 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: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

Request Body:
{
  "image_base64": "/9j/4AAQSkZJRgABAgEASABIAAD",
  "external_image_id": "imageID",
  "external_fields": {
     "timestamp": 12,
     "id": "home"
  }
}
```

- Example 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"}
```

- Example 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"
  }
}
```

Example Responses

Status code: 200

Response example (successful request)

```
{
  "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": "JLa9hYLl"
    }
  ]
}
```

Status code: 400

Example response (failed request)

```
{
  "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 Codes**.

4.6.2 Querying a Face

Function

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

Prerequisites:

Ensure that you have enabled **FRS**. For detailed operations, see **Applying for FRS**.

URI

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}

Table 4-58 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the ID, see **Obtaining the Project ID/ Account Name/AK/SK**.
face_set_name	Yes	Yes	Name of a facial image library.

Table 4-59 Query parameters

Parameter	Mandatory	Type	Description
offset	No. This parameter is alternative to **face_id**.	Integer	Start number of data records to be read. The default value is **0**.
limit	No. This parameter is alternative to **face_id**.	Integer	Number of data records to be read. The default value is **5**.
face_id	No. This parameter is alternative to **offset**.	String	Face ID.

NOTE

The sum of values of **offset** and **limit** cannot exceed 10,000. The face recognition system is not a storage system and does not support data traversal. You can save the **face_id** and query each imported face based on the **face_id**.

Request Parameters

Table 4-60 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. During API authentication using a token, the token is added to requests to obtain permissions for calling the API. The value of **X-Subject-Token** in the response header is the obtained token.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

Response Parameters

Status code: 200

Table 4-61 Response body parameters

Parameter	Type	Description
face_set_id	String	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.
face_set_name	String	Name of the facial image library. This parameter is not included when the API fails to be called.
faces	Array of **FaceSetFace** objects	Face structure in the facial image library. For details, see **FaceSetFace**. This parameter is not included when the API fails to be called.

Table 4-62 FaceSetFace

Parameter	Type	Description
bounding_box	**BoundingBox** object	Position of a face in an image. For details about the BoundingBox structure, see **BoundingBox**.
external_fields	Object	Additional field a user customizes

Parameter	Type	Description
external_image_id	String	ID of the external image to which a face belongs
face_id	String	Face ID, which is a unique ID generated by the system

Table 4-63 BoundingBox

Parameter	Type	Description
width	Integer	Width of a rectangle
top_left_y	Integer	Vertical coordinate of the upper-left corner of a rectangle
top_left_x	Integer	Horizontal coordinate of the upper-left corner of a rectangle
height	Integer	Height of a rectangle

Status code: 400

Table 4-64 Response body parameters

Parameter	Type	Description
error_code	String	Error code when calling the API failed. This parameter is not included when the API is successfully called.
error_msg	String	Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.

Example Requests

For details about how to obtain the value of **X-Auth-Token**, see **Authentication**.

- Example 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...
```

Example Responses

Status code: 200

Response example (successful request)

```
{
  "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
      }
    }
  ]
}
```

Status code: 400

Example response (failed request)

```
{
  "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 Codes**.

4.6.3 Updating a Face

Function

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

Prerequisites:

Ensure that you have enabled **FRS**. For detailed operations, see **Applying for FRS**.

NOTE

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

URI

PUT /v2/{project_id}/face-sets/{face_set_name}/faces

Table 4-65 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the ID, see **Obtaining the Project ID/ Account Name/AK/SK**.
face_set_name	String	Yes	Name of a facial image library.

Request Parameters

Table 4-66 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. During API authentication using a token, the token is added to requests to obtain permissions for calling the API. The value of **X-Subject-Token** in the response header is the obtained token.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

Table 4-67 Request body parameters

Parameter	Type	Mandatory	Description
face_id	String	Yes	Face ID, which is a unique ID generated by the system
external_image_id	String	No	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 **external_image_id** or **external_fields** is modified.

Parameter	Type	Mandatory	Description
external_fields	Object	No	The value of a custom field can contain 1 to 36 characters, while the value of a string field can contain 1 to 256 characters. For details, see **Customized Fields**. Either parameter **external_image_id** or **external_fields** is modified.

Response Parameters

Status code: 200

Table 4-68 Response body parameters

Parameter	Type	Description
face_number	Integer	Number of updated faces. This parameter is not included when the API fails to be called.
face_set_id	String	ID of the facial image library. This parameter is not included when the API fails to be called.
face_set_name	String	Name of the facial image library. This parameter is not included when the API fails to be called.

Status code: 400

Table 4-69 Response body parameters

Parameter	Type	Description
error_code	String	Error code when calling the API failed. This parameter is not included when the API is successfully called.
error_msg	String	Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.

Example Requests

For details about how to obtain the value of **X-Auth-Token**, see **Authentication**.

- Example request

```
PUT https://{endpoint}/v2/{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"
  }
}
```

Example Responses

Status code: 200

Response example (successful request)

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

Status code: 400

Example response (failed request)

```
{
  "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 Codes**.

4.6.4 Deleting a Face

Function

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

Prerequisites:

Ensure that you have enabled **FRS**. For detailed operations, see **Applying for FRS**.

URI

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

Table 4-70 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the ID, see **Obtaining the Project ID/ Account Name/AK/SK**.
face_set_name	String	Yes	Name of the facial image library.
field_name	String	Yes	Condition based on which the face is to be deleted. Fixed fields (**external_image_id** and **face_id**) and **customized fields** (excluding empty strings and null) are supported.

Request Parameters

Table 4-71 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. During API authentication using a token, the token is added to requests to obtain permissions for calling the API. The value of **X-Subject-Token** in the response header is the obtained token.
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

Response Parameters

Status code: 200

Table 4-72 Response body parameters

Parameter	Type	Description
face_number	Integer	Number of deleted faces. This parameter is not included when the API fails to be called.

Parameter	Type	Description
face_set_id	String	ID of the facial image library. This parameter is not included when the API fails to be called.
face_set_name	String	Name of the facial image library. This parameter is not included when the API fails to be called.

Status code: 400

Table 4-73 Response body parameters

Parameter	Type	Description
error_code	String	Error code when calling the API failed. This parameter is not included when the API is successfully called.
error_msg	String	Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.

Example Requests

For details about how to obtain the value of **X-Auth-Token**, see **Authentication**.

- Example 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: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...
```

- Example 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: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...
```

- Example 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: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...
```

Example Responses

Status code: 200

Response example (successful request)

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

Status code: 400

Example response (failed request)

```
{
  "error_code": "FRS.0402",
  "error_msg": "External id is not exist, can not delete face"
}
```

4.6.5 Deleting Faces in Batches

Function

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

Prerequisites:

Ensure that you have enabled **FRS**. For detailed operations, see **Applying for FRS**.

NOTE

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

URI

DELETE /v2/{project_id}/face-sets/{face_set_name}/faces/batch

Table 4-74 Path parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the ID, see **Obtaining the Project ID/ Account Name/AK/SK**.
face_set_name	String	Yes	Name of a facial image library.

Request Parameters

Table 4-75 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. During API authentication using a token, the token is added to requests to obtain permissions for calling the API. The value of **X-Subject-Token** in the response header is the obtained token.

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	MIME type of the request body. The value is **application/json**.

Table 4-76 Request body parameters

Parameter	Type	Mandatory	Description
filter	String	Yes	Filtering criteria. For details, see **Filter Syntax**.

Response Parameters

Status code: 200

Table 4-77 Response body parameters

Parameter	Type	Description
face_number	Integer	Number of deleted faces. This parameter is not included when the API fails to be called.
face_set_id	String	ID of the facial image library. This parameter is not included when the API fails to be called.
face_set_name	String	Name of the facial image library. This parameter is not included when the API fails to be called.

Status code: 400

Table 4-78 Response body parameters

Parameter	Type	Description
error_code	String	Error code when calling the API failed. This parameter is not included when the API is successfully called.
error_msg	String	Error message returned after the API fails to be called. This parameter is not included when the API is successfully called.

Example Requests

For details about how to obtain the value of **X-Auth-Token**, see **Authentication**.

- Example 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: MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDT...

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

Example Responses

Status code: 200

Response example (successful request)

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

Status code: 400

Example response (failed request)

```
{
  "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 Codes**.

5 Public Data Structures

5.1 Common Message Headers

Common Request Headers

Table 5-1 Common request header fields

Parameter	Description	Mandatory	Example Value
Content-type	Request body type or format. Its default value is **application/json**.	Yes	application/json
X-Project-Id	Project ID, which is used to obtain the token of a project. For how to obtain the project ID, see **Obtaining the Project ID/Account Name/AK/SK**.	No	e9993fc787d94b6c886cbaa340f9c0f4
X-Auth-Token	User token, a response to the API used to **obtain a user token**. This API is the only one that does not require authentication.	This parameter is mandatory for token authentication.	-

Parameter	Description	Mandatory	Example Value
X-Sdk-Date	Time when the request is sent. The time is in **YYYYMMDD'T'HHMMSS'Z'** format. The value is the current Greenwich Mean Time (GMT) of the system.	This parameter is mandatory for AK/SK-based authentication, but not required for PKI token-based authentication	20190307T101459Z
Authorization	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	This parameter is mandatory for AK/SK-based authentication. For details about AK/SK-based authentication, see **Authentication**.	SDK-HMAC-SHA256 Credential=ZIRRKMTWPTQFQI1WKNKB/20150907//ec2/sdk_request, SignedHeaders=content-type;host;x-sdk-date, Signature=55741b610f3c9fa3ae40b5a8021ebf7ebc2a28a603fc62d25cb3bfe6608e1994
Host	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**.	This parameter is mandatory for AK/SK-based authentication.	code.test.com or code.test.com:443

NOTE

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

Common Response Headers

Table 5-2 Common response headers

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

5.2 Message Object Structures

5.2.1 AllParam

Function

This topic describes all parameter types involved in FRS.

Parameter

Table 5-3 Structure format description

Parameter	Type	Description
project_id	String	Project ID. For details about how to obtain the ID, see **Obtaining the Project ID/ Account Name/AK/SK**.
image_url	String	Image URL. Currently, only the URL of an OBS bucket on Huawei Cloud is supported and FRS must have the permission to read data in the OBS bucket. For details about how to enable the read permission, see **Service Authorization**.
image_file	File	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.
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.
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 **0.93**, 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 **100000**, and the maximum value is **100000**. You can create facial image libraries as you need.
face_id	String	Face ID returned by the system after a face is imported. The value consists of eight uppercase and lowercase letters that are randomly generated.

Parameter	Type	Description
external_image_id	String	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.
external_fields	JSON	Enters a value based on the customized data type. This field is defined when you create the facial image library. For details, see **Customized Fields**.
top_n	Integer	*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
threshold	Double	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**.
offset	Integer	Start number of data records to be read. The default value is **0**.
limit	Integer	Number of records to be read. The default value is **5**.
video_url	String	Video URL. Currently, only the URL of an OBS bucket on Huawei Cloud is supported and FRS 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.

Parameter	Type	Description
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.
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
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.

Parameter	Type	Description
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.

5.2.2 DetectFace

Function

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

Parameter Description

Table 5-4 Structure format description

Parameter	Type	Description
bounding_box	**BoundingBox** object	Position of a face in an image. For details about the BoundingBox structure, see **BoundingBox**.

5.2.3 Landmark

Function

This topic describes structures of key facial points.

Parameter Description

Table 5-5 Structure format description

Parameter	Type	Description
nose_contour	List [Point]	Nose contour. Point is the coordinate value of the contour.
mouth_contour	List [Point]	Mouth contour. Point is the coordinate value of the contour.
eyebrow_contour	List [Point]	Eyebrow contour. Point is the coordinate value of the contour.
eyes_contour	List [Point]	Eyes contour. Point is the coordinate value of the contour.

Parameter	Type	Description
face_contour	List [Point]	Face contour. Point is the coordinate value of the contour.

5.2.4 FaceSetFace

Function

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

Parameter Description

Table 5-6 Structure format description

Parameter	Type	Description
bounding_box	**BoundingBox** object	Position of a face in an image For details about the BoundingBox structure, see **BoundingBox**.
face_id	String	Face ID, the unique ID generated by the system
external_image_id	String	ID of the external image to which a face belongs
external_fields	JSON	Additional field a user customizes

5.2.5 SearchFace

Function

This topic describes face structures returned by Face Retrieval.

Parameter Description

Table 5-7 Structure format description

Parameter	Type	Description
bounding_box	**BoundingBox** object	Position of a face in an image For details about the BoundingBox structure, see **BoundingBox**.

Parameter	Type	Description
face_id	String	Face ID, the unique ID generated by the system
external_image_id	String	ID of the external image to which a face belongs
similarity	Double	Similarity degree in Face Retrieval
external_fields	JSON	Additional field a user customizes

5.2.6 FaceSetInfo

Function

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

Parameter Description

Table 5-8 Structure format description

Parameter	Type	Description
face_set_name	String	Name of the facial image library
face_set_id	String	ID of the facial image library. It is a randomly generated string containing eight characters.
create_date	String	Creation time
face_set_capacity	Integer	Maximum capacity of a facial image library
face_number	Integer	Number of faces in a facial image library
external_fields	JSON	Additional field a user customizes

5.2.7 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 5-9 Structure format description

Parameter	Type	Description
top_left_x	Integer	Horizontal coordinate of the upper-left corner of a rectangle
top_left_y	Integer	Vertical coordinate of the upper-left corner of a rectangle
width	Integer	Width of a rectangle
height	Integer	Height of a rectangle

5.2.8 VideoDetectResult

Function

This topic describes the result structures of Face LiveDetect.

Parameter Description

Table 5-10 Structure format description

Parameter	Type	Description
alive	Boolean	Whether a living figure is detected
picture	String	Base64 coding of the image in which the maximum face is detected
actions	List	Action list For details about the parameters, see **Table actions field description**.

Table 5-11 actions field description

Parameter	Type	Description
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.

5.2.9 ServiceInfo

Function

This topic describes structures that record sub-service information.

Parameter Description

Table 5-12 Structure format description

Parameter	Type	Description
subscribe_status	Boolean	Whether to enable the sub-service
create_time	String	Time when the sub-service is enabled

5.2.10 WarningList

Function

This topic describes warning information about video LiveDetect.

Parameter Description

Table 5-13 Structure and Format Description

Parameter	Type	Description
warningCode	Integer	Warning ID
warningMsg	String	Warning information

Table 5-14 Errors

warningCode	warningMsg
1	The face is not facing forward.
2	The video does not exceed 1 second.
3	The video lasts for more than 15 seconds.
4	Two faces are detected.

warningCode	warningMsg
5	No face is detected.
6	The range of motion is too small.
7	The video quality is poor or the person in the video is not a real person.
8	Failed to select a preferred image.
101	The overall quality of the facial image is low.
102	The face is blurred.
103	The face angle is too large.
104	The face is blocked.
105	The image is dark.
106	The image contains multiple faces.

5.3 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:
{
    "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"
        }
    }
}
```

5.4 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

```
"sort":[
          {
              "location": "desc"
          }
]
```

- Example 2: Sorting of multiple fields

```
"sort": [
            {
             "timestamp": "desc"
             },
             {
                "rowkey": "asc"
             }
]
```

5.5 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.

6 Appendix

6.1 Status Codes

Table 6-1 Status code

Status Code	Message	Description
100	Continue	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.
101	Switching Protocols	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.
200	OK	The request has been fulfilled.
201	Created	The request for creating a class has been successfully processed.
202	Accepted	The request has been accepted, but the processing has not been completed.
203	Non-Authoritative Information	Unauthorized information. The request is successful.
204	NoContent	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.
205	Reset Content	The server has successfully processed the request, but does not return any content.

Status Code	Message	Description
206	Partial Content	The server has successfully processed a part of the GET request.
300	Multiple Choices	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.
301	Moved Permanently	This and all future requests should be permanently directed to the given URI indicated in this response.
302	Found	The requested resource resides temporarily under a different URI.
303	See Other	The response to the request can be found under a different URI, The response to the request can be found under a different URI, and should be retrieved using a GET or POST method.
304	Not Modified	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.
305	Use Proxy	The requested resource must be accessed through a proxy.
306	Unused	This HTTP status code is no longer used.
400	BadRequest	Specifies invalid requests. Do not retry the request before modification.
401	Unauthorized	The authentication information provided by the client is incorrect or invalid.
402	Payment Required	This status code is reserved for future use.
403	Forbidden	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.
404	NotFound	The requested resource cannot be found. Do not retry the request before modification.
405	MethodNotAllowed	The request contains one or more methods not supported for the resource. Do not retry the request before modification.

Status Code	Message	Description
406	Not Acceptable	The server could not fulfill the request according to the content characteristics of the request.
407	Proxy Authentication Required	This code is similar to 401, but indicates that the client must first authenticate itself with the proxy.
408	Request Time-out	The server timed out waiting for the request. The client may repeat the request without modifications at any time later.
409	Conflict	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.
410	Gone	The requested resource cannot be found. The status code indicates that the requested resource has been deleted permanently.
411	Length Required	The server refuses to process the request without a defined **Content-Length**.
412	Precondition Failed	The server does not meet one of the preconditions that the requester puts on the request.
413	Request Entity Too Large	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.
414	Request-URI Too Large	The Request-URI is too long for the server to process.
415	Unsupported Media Type	The server does not support the media type in the request.
416	Requested range not satisfiable	The requested range is invalid.
417	Expectation Failed	The server fails to meet the requirements of the **Expect** request-header field.
422	UnprocessableEntity	The request was well-formed but was unable to be followed due to semantic errors.

Status Code	Message	Description
429	TooManyRequests	The client has sent excessive number of requests to the server within a given time (exceeding the limit on the access frequency of the client), or the server has received an excessive number of requests within a given time (beyond its processing capability). In this case, the client should repeat requests after the time specified in the **Retry-After** header of the response expires.
500	InternalServerError	The server is able to receive the request but unable to understand it.
501	Not Implemented	The server does not support the requested function.
502	Bad Gateway	The server acting as a gateway or proxy has received an invalid request from a remote server.
503	ServiceUnavailable	The requested service is invalid. Do not retry the request before modification.
504	ServerTimeout	The request cannot be fulfilled within a given time. This status code is returned to the client only when the **Timeout** parameter is specified in the request.
505	HTTP Version not supported	The server does not support the HTTP protocol version used in the request.

6.2 Error Codes

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 4*xx* or 5*xx* 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.

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to the instructions provided in **API Gateway Error Codes**.

- Example exception response

```
{
  "error_code": "FRS.0202",
  "error_msg": " The service has been freeze."
 }
```

- Parameter description

Parameter	Mandatory	Type	Description
error_code	No	String	Error code
error_msg	No	String	Error message

-

Type	Status Code	Error Code	Description	Measure
Common service errors (error code range: 1–99)	403	FRS.0002	Token authentication failed. The user token is incorrect or has expired.	Obtain your token again and use the token to call the FRS APIs. For details, see **Authentication**.
	400	FRS.0010	The request header is missing or empty.	Check whether the common message header is configured. For details, see **Making an API Request**.
	400	FRS.0011	A parameter is missing.	Check whether the entered parameters are complete.
	400	FRS.0012	Request parameters are in incorrect formats.	Check whether the format of parameters is valid.
	400	FRS.0013	The face image size is too large or too small.	Upload a facial image that meets the requirements. 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.

Type	Status Code	Error Code	Description	Measure
	400	FRS.0014	The input is not in JSON format.	Check whether the JSON format of the body is valid.
	400	FRS.0015	An error occurred when parsing the Base64 image.	Re-encode the image and enter the correct Base64 code in the body. Ensure that the code format is the same as that of the example.
	400	FRS.0016	The format of the uploaded file is not supported.	Upload a file in a format that can be recognized. For details, see **Restrictions and Limitations**.
	400	FRS.0017	The uploaded body exceeds the allowed range.	Check the size of the body parameters. For details, see **Restrictions and Limitations**.
	500	FRS.0018	Internal service error.	Contact the customer service for assistance.
	400	FRS.0019	The service has not been enabled.	Log in to the **FRS management console** and enable the sub-service you need.
	400	FRS.0020	The sub-service has not been enabled.	Log in to the **FRS management console** and enable the sub-service you need.
	400	FRS.0021	Invalid OBS URL.	Check whether the OBS URL is correct by referring to the request example.

Type	Status Code	Error Code	Description	Measure
	400	FRS.0022	Failed to obtain the OBS file.	Check whether the file exists on OBS and whether the access permission on the file is correct.
	400	FRS.0023	The size of the file on OBS exceeds the upper limit.	Check whether the size of the file on OBS meets the API requirements. For details, see **Restrictions and Limitations**.
	400	FRS.0024	The file does not exist on OBS.	Check whether the file exists on OBS.
	400	FRS.0027	OBS unauthorized.	Log in to the **FRS management console** and click **Service Authorization** in the upper right corner to complete OBS authorization.
	403	FRS.0028	The project ID does not match the token.	Check whether the project ID for obtaining the token is the same as that in the request URL.
	400	FRS.0029	The format of filter criteria is incorrect.	Check whether the input format is valid. For details, see **Request Parameters** of Face Retrieval.

Type	Status Code	Error Code	Description	Measure
	400	FRS.0030	**return_fields** is not defined.	Check whether **return_fields** is defined. For details, see **Request Parameters** of Face Retrieval.
	400	FRS.0031	The **sort** field is not contained in the returned fields.	Check whether the **sort** field is defined. For details, see **Request Parameters** of Face Retrieval.
	400	FRS.0032	The **sort** field and type are not supported. Only numeral type is supported.	The **sort** field and type are not supported. Only numeral type is supported.
	400	FRS.0033	The number of requests has exceeded the traffic control threshold.	You are advised to control the request policy or retry. You can expand the capacity to increase the QPS quota.
	400	FRS.0035	The URL is invalid.	Check whether the OBS URL is correct by referring to the request example.
	400	FRS.0036	Failed to obtain the file from the URL.	Use the URL provided by OBS to upload the image and check whether the URL is correct.
Service management errors (error code range: 201–300)	400	FRS.0201	The user has been registered.	The user has been registered.
	400	FRS.0202	The service has been frozen.	Contact the customer service for assistance.

Type	Status Code	Error Code	Description	Measure
	400	FRS.0203	Failed to update user information.	Check the entered user information.
Facial image library resource errors (error code range: 301–400)	400	FRS.0301	You cannot create an existing facial image library.	Check the name of the facial image library to be created. If the name already exists, change another name and create it again.
	400	FRS.0302	The facial image library does not exist.	Check whether the input facial image library exists.
	400	FRS.0304	No face is detected. Failed to search the face.	Upload a facial image that meets the requirements. For details, see **Restrictions and Limitations**.
	400	FRS.0305	A new facial image library cannot be created because the number of facial image libraries exceeds the upper limit.	Check whether the number of facial image libraries exceeds the upper limit or contact the customer service.
	403	FRS.0306	New faces cannot be added because the number of faces exceeds the upper limit of a facial image library.	Add a new facial image library and add faces to it.
Face resource errors (error code range: 401–500)	400	FRS.0401	The value of the corresponding type cannot be found and cannot be deleted.	For details, see the descriptions of the facial resource management APIs.

Type	Status Code	Error Code	Description	Measure
	400	FRS.0402	The entered field does not exist.	For details, see the descriptions of the facial resource management APIs.
	400	FRS.0403	The face ID does not exist.	Check whether the face ID exists.
	400	FRS.0404	No face is detected. Failed to add the face.	Upload a facial image that meets the requirements. For details, see **Restrictions and Limitations**.
	400	FRS.0405	The customized field cannot be added because it is not defined.	Check whether the customized field to be added exists. If the customized field does not exist, create it and then add it.
	400	FRS.0406	The imported data type does not match the defined data type.	Check whether the imported data type is the same as the defined data type.
	400	FRS.0407	No matched data is found during batch deletion.	Check whether the data to be deleted exists.
Face verification errors (error code range: 501–600)	400	FRS.0501	No face is detected in the image.	Upload a facial image that meets the requirements. For details, see **Restrictions and Limitations**.

Type	Status Code	Error Code	Description	Measure
Face LiveDetect errors (error code range: 701–800)	400	FRS.0701	Video parsing error.	Check whether the video is damaged.
	400	FRS.0702	The action is not supported.	Check whether the entered action exists. For details, see **Face LiveDetect**.
	400	FRS.0703	The action time is invalid.	Check whether the input is valid. For details, see **Face LiveDetect**.
	400	FRS.0704	The number of imported actions exceeds 10.	The number of imported actions exceeds 10.
	400	FRS.0705	The matching mode is not supported.	Check whether the input is valid. For details, see **Restrictions and Limitations**.
	400	FRS.0706	The video duration is not allowed. The duration must be 1 to 15 seconds.	The video duration is not within 1 to 15 seconds. Use a video that meets the duration requirement.
	400	FRS.0707	No face is detected in the image.	Check whether there is a face in the image or whether the face meets the restrictions. For details, see **Restrictions and Limitations**.
	400	FRS.0708	Check whether the quality of the input facial image meets the requirements.	Check whether the quality of the input facial image meets the requirements. For details, see **Restrictions and Limitations**.

Type	Status Code	Error Code	Description	Measure
	400	FRS.0709	Check whether the input facial image contains multiple faces.	Check whether the input facial image contains multiple faces.
	400	FRS.0751	Failed to parse video data.	Check whether the quality of the input facial video meets the requirements. For details, see **Restrictions and Limitations**.

- Gateway error code

Type	Status Code	Error Code	Description	Measure
Gateway errors (error code range: 1–400)	404	APIG.0101	The API does not exist.	The request address or URL used for calling the API does not exist or has not been released. Check whether the request address and URL are correct.
	413/504	APIG.0201	The request body exceeds the allowed range or the backend times out.	Check whether the request size is valid. Contact the customer service to check whether the backend service is running properly.
	401	APIG.0301	Authentication failed.	Perform authentication. For details, see **Authentication**.

Type	Status Code	Error Code	Description	Measure
	401	APIG.0307	The token needs to be updated.	- The validity period of the token is 24 hours. Obtain the token again to call the API. - Check whether the endpoint in the API URL is correct. Services deployed in different regions cannot be called across regions. If APIs in different regions are called, the token is invalid and error code APIG.0307 is displayed.

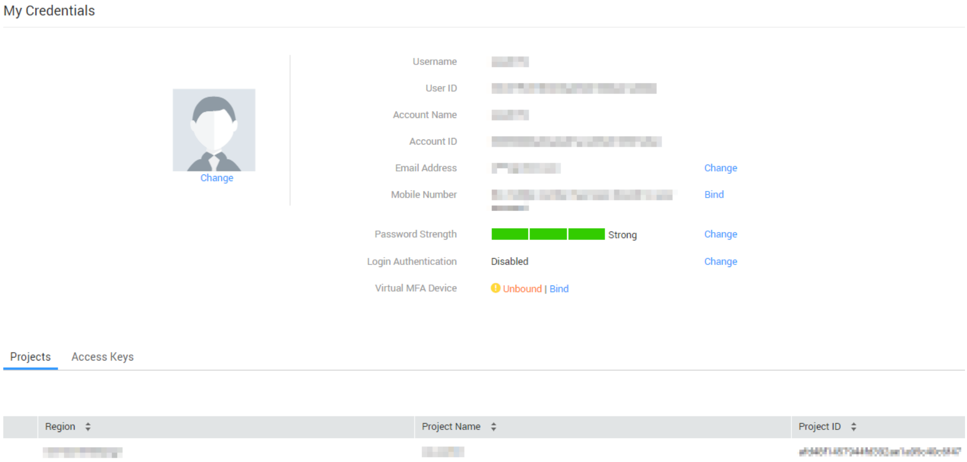
6.3 Obtaining the Project ID/Account Name/AK/SK

6.3.1 Obtaining the Project ID/Account Name

Obtaining the Project ID/Account Name from the Console

1. Log in to the **management console**.
2. Hover over the username and select **My Credentials** from the drop-down list.
3. On the **My Credentials** page, check the username, account name, and the project ID.

Figure 6-1 Checking the project ID



My Credentials

Change

Username	[redacted]	
User ID	[redacted]	
Account Name	[redacted]	
Account ID	[redacted]	
Email Address	[redacted]	Change
Mobile Number	[redacted]	Bind
Password Strength	Strong	Change
Login Authentication	Disabled	Change
Virtual MFA Device	Unbound | Bind	

Projects Access Keys

Region	Project Name	Project ID
[redacted]	[redacted]	[redacted]

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**. Select **Headers**, add **KEY** as **X-Auth-Token**, and set **VALUE** to the obtained token. For details about API authentication, see **Authentication**.

The following is an example response. For example, if FRS is deployed in the **ap-southeast-1** region, the value of **name** in the response body is **ap-southeast-1**. The value of **id** in **projects** is the project ID.

```
{
    "projects": [
        {
            "domain_id": "65382450e8f64ac0870cd180d14e684b",
            "is_domain": false,
            "parent_id": "65382450e8f64ac0870cd180d14e684b",
            "name": "ap-southeast-1",
            "description": "",
            "links": {
                "next": null,
                "previous": null,
                "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
            },
            "id": "a4a5d4098fb4474fa22cd05f897d6b99",
            "enabled": true
        }
    ],
    "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects"
    }
}
```

6.3.2 Obtaining the Account ID

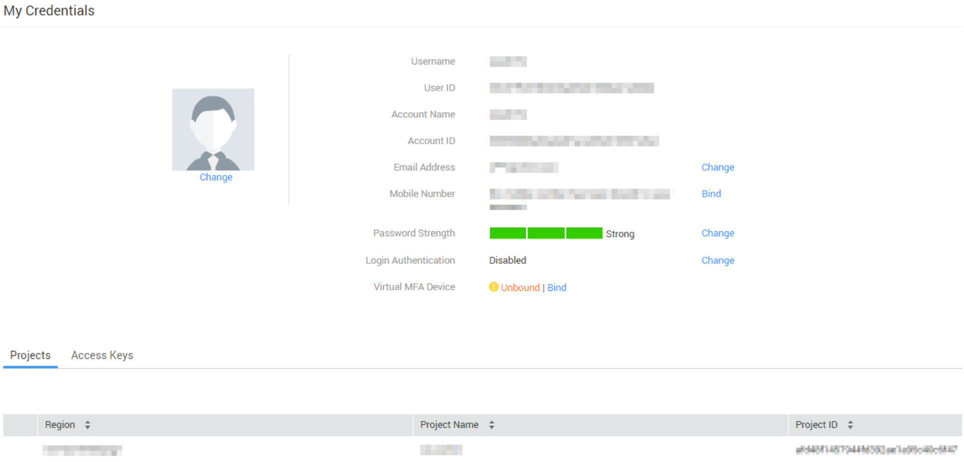
An account ID (**domain-id**) is required for some URLs when an API is called. To obtain an account ID, perform the following steps:

1. Log in to the management console.

2. Hover over the username and choose **My Credentials** from the drop-down list.

   On the **My Credentials** page, check **Account ID**.

Figure 6-2 Checking the account ID



My Credentials

Username [redacted]

User ID [redacted]

Account Name [redacted]

Account ID [redacted]

Email Address [redacted]

Mobile Number [redacted]

Password Strength Strong

Login Authentication Disabled

Virtual MFA Device Unbound | Bind

Region	Project Name	Project ID
[redacted]	[redacted]	[redacted]

6.3.3 Obtaining an Access Key ID/Secret Access Key (AK/SK)

1. Log in to the **FRS management console**.
2. In the upper right corner of the page, click the username and choose **My Credentials** from the drop-down list.

   The **My Credentials** page is displayed.
3. Choose **Access Keys** > **Create Access Key**.

   The **Create Access Key** dialog box is displayed.
4. Enter the login password. If a mobile phone or email address is associated, obtain the verification code for authentication.

   After the authentication is successful, the **Download Access Key** dialog box is displayed.
5. Click **OK** to download and save the access key.
6. If you have generated the AK/SK, find the AK/SK file you downloaded. Generally, the file name is **credentials.csv**.