Jovision Technology Co., Ltd.

AI System Development Protocol

V1.3.2

2020-04-08

© Copyright 2000-2020: Jovision Technology Co., Ltd. All Rights Reserved

**Statement**

Thank you for choosing Jovision. This protocol list is intended for users of Face Recognition System. It may contain several technical incorrect places or printing errors, and the content is subject to change without notice. All information in this booklet has been thoroughly reviewed.

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Content** | **Revised by** |
| 2019.05.10 | V1.0 | First Release:   1. Added push interface of face capture results 2. Added push interface of face comparison results 3. Add interface to modify black and white list | Pang Xin |
| 2019.05.29 | V1.1 | 1. Added Face snapshot interface adds face image name and face background image name | Shang Jing |
| 2019.07.17 | V1.2 | 1. Modified list import, delete, query interface | Wei Xianzhi |
| 2019.11.01 | V1.2.1 | 1. Modified the coordinate format of the captured face image 2. Modified method name in json data | Pang Xin |
| 2019.11.26 | V1.3 | 1. Modified the comparison upload data format 2. Modified the equipment issued instructions. Improve the relevant interface 3. Added TCP transmission 4. Added token and data transmission security | Pang Xin |
| 2020.03.12 | V1.3.1 | 1. Added temperature attribute to face report interface | Pang Xin |
| 2020.04.08 | V1.3.2 | 1. Increase temperature display for face comparison 2. Modify heartbeat response and synchronous data response 3. Modify dataLength field description | Wei Xianzhi |

[1 Introduction 6](#_Toc38030768)

[1.1 Overview 6](#_Toc38030769)

[1.2 HTTP Protocol Information Category 7](#_Toc38030770)

[Face recognition information 7](#_Toc38030771)

[Face comparison information 7](#_Toc38030772)

[Heartbeat packet information 7](#_Toc38030773)

[1.3 Applicability 7](#_Toc38030774)

[2 Message Format 7](#_Toc38030775)

[2.1 Message Header 7](#_Toc38030776)

[2.2 Message Body 8](#_Toc38030777)

[3 Key Acquisition 9](#_Toc38030778)

[4 Heartbeat Push 10](#_Toc38030779)

[5 Device push data to server 11](#_Toc38030780)

[5.1 Upload face capture data (Temporarily unused) 12](#_Toc38030781)

[5.2 Upload face comparison data 13](#_Toc38030782)

[5.3 HTTP Push face capture data (Reserved) (Temporarily unused) 17](#_Toc38030783)

[6 Device processing receives processing server information 21](#_Toc38030784)

[6.1 Get a picture from the device list library (Temporarily unused) 21](#_Toc38030785)

[6.2 Add a picture to the device 23](#_Toc38030786)

[6.3 Delete a picture in the list library 24](#_Toc38030787)

[6.4 Modify a picture in the list library 26](#_Toc38030788)

[6.5 Add face pictures to the list library in batches 28](#_Toc38030789)

[6.6 Batch delete face pictures in the list library 29](#_Toc38030790)

[6.7 Get a list of device list libraries (Not used temporarily) 30](#_Toc38030791)

[6.8 The device actively synchronizes server data 33](#_Toc38030792)

[7 HTTP protocol error returns JSON format 34](#_Toc38030793)

[Error code 34](#_Toc38030794)

# Introduction

## Overview

This document specifies the API of Jovision’s Face recognition device and Face recognition with integrated Cloud platform.

The snapshot machine or comparison machine pushes the face recognition or comparison result information through the TCP / HTTP protocol. The user needs to log in to the device to enable the push function and set the server information. The operation is as follows:

Login camera with IE. Go to **Intelligent Analysis-> Face Detection-> Server Configuration-> Set server information.**

For example, fill the push address at http://www.sunywo.com/faceinfo

and http://www.sunywo.com/keepalive:

Where http://www.sunywo.com/faceinfo receives face recognition/face comparison push information. http://www.sunywo.com/keepalive Receive heartbeat information.

* **Grab face and upload HTTP**: check / uncheck (enable HTTP push function or not)
* **HTTP address**: http://www.sunywo.com/faceinfo (address to receive face recognition / face comparison information)
* **HTTP port**: http server port
* **Upload path**: http server path
* **Heartbeat**: checked / unchecked (enable heartbeat push)
* **Heartbeat server address: http://www.sunywo.com/keepalive (address to receive heartbeat information)**
* **Heartbeat interval**: (Unit: Second. Value greater than 0. Default 30 seconds)

After filling in the information, save the server information. When the IPC generates the identification information, it will send the identification information to the designated address。

## HTTP Protocol Information Category

### Face recognition information

[Face Recognition Camera] Push [Face capture information]: HTTP request header (POST)/TCP and face recognition information (JSON)

### Face comparison information

[Comparison] Push [Face comparison information]: TCP and face comparison information (JSON)

### Heartbeat packet information

[Face Recognition Camera] and [Comparison] Send [Heartbeat Information]: HTTP Request header (POST) TCP + Heartbeat Packet Information (JSON)

## Applicability

* This API only applicable to support Jovision’s Face capture device, Integrated face recognition device, Access control, Attendance device, Ai NVR, Cloud platform, App, CMS etc.
* This API is platform language independent.

# Message Format

The message of this protocol is in HTTP format. The HTTP header should be added to the message. After the HTTP header, two carriage return line breaks should be added, namely "\r \n \r \n". The message body is JSON format data, and the communication messages are encoded in UTF-8.

## Message Header

**Request message HTTP header**:

POST / HTTP/1.0

Connection: keep-alive

Content-Type: application/json;charset=utf-8

Content-Length: 172

Host: 192.168.5.146:8127

**Response message HTTP header**:

HTTP/1.1 200 OK

Content-Length: 347

Content-Type: application/json;charset=utf-8

Connection: close

## Message Body

The body of the message is a string in JSON format.

Equipment type: Complete the whole system, and the type of communication equipment between nodes is the key information as:

1. Cloud platform add devices can be categorized according to this field
2. After adding face database to cloud platform, import information to comparison machine.
3. After modifying the face database for the comparison machine through the App, the changed information needs to be pushed to the opened server on time such as cloud platforms.

typedef enum

{

DEV\_IPC\_CAPTURE = 0x10, // Face Recognition Camera

DEV\_IPC\_COMPARE = 0x12, // Comparison Machine

DEV\_IPC\_BINOCULAR\_GATE = 0x14, // Binocular Access Control / Attendance

DEV\_IPC\_MONOCULAR\_GATE = 0x16, // Monocular Access Control / Attendance

DEV\_NVR\_SMART = 0x18, //Ai NVR

DEV\_AI\_CLOUDE= 0x1A, //Cloud Platform

DEV\_C\_APP = 0x1C, //App

DEV\_C\_WEIXIN = 0x1E,//WeChat

DEV\_C\_CMS = 0x20,//CMS

DEV\_C\_OTHERS, // Other sub control tools such as attendance client

DEV\_MAX

}SW\_DEV\_TYPE;

# Key Acquisition

The key is generated by UUID + device key (the key of each type of device is the same) + random number, and the corresponding token is valid if there is a long connection at present

Access path http://attend.jovisionai.com:8070/cloud\_kq/api/getAccessToken

**Push Parameters**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| method | Fixed "accesstoken" | 20 | string |  |
| uuid | Device UUID | 32 | string |  |
| device\_key | Key Same for each type of device | 32 | string |  |
| device\_type | Device type | SW\_DEV\_TYPE | integer |  |
| mac | Device MAC Address | 64 | string |  |
| timestamp | Unix time stamp |  | integer |  |
| account | The account password of the device itself, which is generated during control digest |  |  |  |
| --user |  |  |  |  |
| --passwd |  |  |  |  |

**Return Parameter**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| code | 0 got token successfully  -1001 system exception, failed to get token  -1004 device not registered, unable to get token |  |  |  |
| retMsg | Return to prompt |  | string |  |
|  | dataObject |  | Object type (contains the following data) |  |
| method | Fixed "accesstoken" | 20 | string |  |
| uuid | Device UUID | 32 | string |  |
| expires\_in | Token expiration date (Unix timestamp) |  | integer |  |
| token | String | 64 |  |  |
| tcp\_port | Upload face data interface |  |  |  |

# Heartbeat Push

Domain name: attend.jovisionai.com Port：18567

**Push parameters**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| method | Fixed "keepalive" | 20 | String |  |
| uuid | Device uuid | 32 | String |  |
| mac | Device mac address | 64 | string |  |
| device\_type | Device type | SW\_DEV\_TYPE | integer |  |
| timestamp | Unix Timestamp |  | string |  |
| soft\_version | Main program version | 128 | string |  |
| token |  |  | string |  |

**Return Parameter**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| dataLength | Data length, the entire json data length, including the dataLength field itself |  | string |  |
| data |  |  |  |  |
| --method | Fixed method of obtaining information | keepalive | String |  |
| --code | 0 heartbeat push succeeded  -1002 device not in server list, invalid heartbeat  -1003 token is invalid |  |  |  |
| retMsg | Return to prompt |  |  |  |

# Device push data to server

The details specified in this section only contains the message body in JSON format. Please add HTTP header in actual use.

The parameters in the request and response messages are examples only, please refer to the actual parameters when used.

## Upload face capture data (Temporarily unused)

The facial recognition result data is automatically uploaded to the server. The format of the uploaded JSON data is as follows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| token | Key | 64 | string |  |
| method | Fixed "sw\_face\_capture\_push" | 20 | string |  |
| uuid | Device uuid | 32 | string |  |
| mac | Device mac address | 64 | string |  |
| host | Device local IP |  | string |  |
| product\_name | Product name | 64 | string |  |
| device\_type | Device type | SW\_DEV\_TYPE | integer |  |
| data | json | 64 | string |  |
| --channel | Channel number | 0-3 | integer | Keep |
| --position | Face coordinates  x:  y:  w:  h: | Relative to the original frame | integer |  |
| --capture\_time | Capture time, e.g.,  2018-04-17 14:53:23 |  | string |  |
| --send\_time | Send time, e.g.,  2018-04-17 14:53:23 |  | string |  |
| --quality | Face quality score | 0 ~ 99999 | integer |  |
| --track\_id | Tracking ID |  | integer |  |
| --attribute | Face Recognition Properties:  Age  Gender  Glasses  Mask  Beard  Race  Emotion | Age [0 ~ 100];  Gender [1: Male 2: Female];  Glasses [0: No glasses 1: Wear glasses 2: Wear sunglasses]  Mask/beard [0: No 1: Yes];  Race [1: Yellow 2: Black 3: White];  Emotion [1: Angry 2: Calm 3: Happy 4: Sad 5: Surprised]; | integer |  |
| --img\_data | Face image data (base64 coding) |  | string |  |
| --img\_name | Face image name |  | string |  |
| --img\_fr | Face background image data (base64 coding)  Note: In "Face Recognition. Parameter Settings", "Image upload format" select "Upload original image" or "Upload face and original image" to have original image data | Current main stream resolution | string |  |
| --img\_fr\_name | Face background picture name |  | string |  |
| --face\_feature | Face 128~512bit feature value (data format:  feature1, feature2, ..., feature127, feature128) featureX is a float value converted to a string) |  | string |  |
| --temperature | realtime\_t : Current temperature  warning\_t : alarm temperature  warning\_type : human body, over temperature, high temperature, linkage, fast |  | string |  |

## Upload face comparison data

Path: http://attend.jovisionai.com:8070/cloud\_kq/api/face/push

The face comparison result data is automatically uploaded to the server, and the format of the uploaded JSON data is as follows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| token | Key | 64 | string |  |
| method | Fixed "sw\_face\_compare\_push" | 20 | string |  |
| uuid | Device uuid | 32 | string |  |
| mac | Device mac address | 64 | string |  |
| host | Device local IP |  | string |  |
| product\_name | Device name | 64 | string |  |
| device\_type | Device type | SW\_DEV\_TYPE | integer |  |
| data | json | 64 | string |  |
| --channel | Channel number | 0-3 | Integer | Keep |
| --index\_id | Unique ID for the entire system, from the server |  | Long integer |  |
| --pic\_sort | Category list:  1: Blacklist  2: White list  3: VIP  4: Custom  It can be distributed by the server later |  | integer |  |
| --pic\_name | Face database picture name (.jpg) | 64 | string |  |
| --match\_score | Comparison score | 0 ~ 1 | float |  |
| --position | Face coordinates  x:  y:  w:  h: | Relative to the original frame | integer |  |
| --capture\_time | Capture time, relative to 1970 seconds |  | string |  |
| --send\_time | Send time, relative to 1970 seconds |  | string |  |
| --quality | Face quality score | 0 ~ 100 | integer |  |
| --visit\_count | Cumulative visits | 1 ~ 2147483647 | integer |  |
| --attribute | Face Recognition Properties:  Age  Gender  Glasses  Mask  Beard  Race  Emotion | age[0 ~ 100];  Gender[1:Male 2:Female];  Glasses[0:No glasses 1: Wear glasses 2: Wear sunglasses]  Mask/beard[0:No 1:Yes];  Race[1: Yellow 2: Black 3: White];  Emotion[1: Angry 2: Calm 3: Happy 4: Sad 5: Surprised]; | integer |  |
| --img\_data | Face image data (base64 coding) |  | string |  |
| --img\_name | Face image name |  | string |  |
| --img\_fr | Face background image data (base64 coding)  Note: In "Face Recognition. Parameter Settings", "Image upload format" select "Upload original image" or "Upload face and original image" to have original image data | Current main stream resolution | string |  |
| --img\_fr\_name | Face background picture name |  | string |  |
| --face\_feature | Face 128~512bit feature value (data format:  feature1, feature2, ..., feature127, feature128) featureX is a float value converted to a string) |  | string |  |

**Return parameter**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| code | 0 Data push successful  -1001 System Exception  -1003 token invalid |  |  |  |
| retMsg | Return to prompt |  |  |  |

Example:

{

"token": "xxx",

"method": "sw\_face\_compare\_push",

"uuid": "xxx",

"mac": "xxx",

"host": "xxx",

"product\_name": "xxx",

"device\_type": "xxx",

"data":

{

"channel": "12221SSS13Q4",

"index\_id": "12",

"pic\_sort": "xxx",

"pic\_name": xxx.jpg,

"position": 0,

"capture\_time": "xxx",

"send\_time": "xxx",

"quality": 2,

"visit\_count": 0,

"attribute":

{

"age": 32,

"gender":1,

"glasses": 1,

"mask": 1,

"race": 1,

"beard": 1,

"emotion":1

},

"img\_data": " ",

"img\_name": "FACE\_10006612\_20181114T220926120\_2.jpg",

"img\_fr": " ",

"img\_fr\_name": "FACE\_10006612\_20181114T220926120\_2\_YT.jpg",

"face\_feature": ""

}

}

Server response message:

{

code:0,

retMsg: “Data uploaded successfully.”

}

## HTTP Push face capture data (Reserved) (Temporarily unused)

The face recognition result data is automatically uploaded to the HTTP server, and the uploaded JSON data format is as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| UserName | Server username | 32 | string |  |
| Password | Server password MD5 check value | 32 | string |  |
| Capture field | Fixed "sw\_cap\_face\_report" | 20 | string |  |
| DeviceID | Device ID (CloudSEE ID) | 64 | string |  |
| DeviceUUID | Device UUID | 32 | string |  |
| DeviceName | Device name | 64 | string |  |
| DeviceType | Device type | SW\_DEV\_TYPE | integer |  |
| Channel | Channel number | 0-3 | integer | Keep |
| HostIP | Current local IP of the device |  | string |  |
| DeviceMac | Device MACaddress | 64 | string |  |
| FaceID | Face ID | 0~9999999 | integer |  |
| FaceQuality | Face quality score | 0~65536 | integer |  |
| Position | Face position  X1  Y1  X2  Y2 | Relative to the original frame | integer |  |
| CaptureTime | Capture time, for example:  2018-04-17 14:53:23 |  | string |  |
| Img | Face image data (base64 coding) |  | string |  |
| ImgName | Face image name |  | string |  |
| ImgFr | Face background image data (base64 encoding)  Note: In "Face Recognition. Parameter Setting", "Picture Upload Format" select "Upload Original Image" or "Upload Face and Original Image" to have original image data | Current main stream resolution | string |  |
| ImgFrName | Face background picture name |  | string |  |
| FaceFeature | Face 128bit feature value (data format: feature1, feature2, ..., feature127, feature128) featureX is a float value converted to string) |  | string | Keep |
| Attribute | Face Recognition Properties:  Age  Gender  Glasses  Mask  Beard  Race  Emotion | Age [0 ~ 100];  Gender [1: Male 2: Female];  Glasses [0: No glasses 1: Wear glasses 2: Wear sunglasses]  Mask/beard [0: No 1: Yes];  Race [1: Yellow 2: Black 3: White];  Emotion [1: Angry 2: Calm 3: Happy 4: Sad 5: Surprised]; | Integer | Keep |

Example:

{

"UserName": "admin",

"Password": "123456789abcdef123456789abcdef12",

"sw\_cap\_face\_report":

{

"DeviceID": "12221SSS13Q4",

"DeviceUUID": "",

"DeviceName": "TEST",

"DeviceType": 0x10,

"Channel": 0,

"HostIP": "192.168.1.100",

"DeviceMac": "12-22-1S-SS-13-Q4",

"Info":

{

"FaceID": 1,

"FaceQuality": 1,

"Position":

{

"x1": 0,

"y1": 0,

"x2": 0,

"y2": 0

},

"CaptureTime": "2018-04-23 09:04:00",

"Img": "base64 ",

"ImgName": "FACE\_12221SSS13Q4\_20190529T071814496\_142.jpg",

"Imgfr": "base64 ",

"ImgfrName": "FACE\_12221SSS13Q4\_20190529T071814496\_142\_YT.jpg",

"FaceFeature":"128bit float",

"Attribute":

{

"Age": 32,

"Gender":1,

"Glasses": 1,

"Mask": 1,

"Race": 1,

"Beard": 1,

"Emotion":1

},

}

}

}

Server response message:

{

"sw\_cap\_face\_report":

{

"result": 0,

"info": "ok"

}

}

# Device processing receives processing server information

## Get a picture from the device list library (Temporarily unused)

Request parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| method | Fixed method of obtaining information | sw\_server\_get\_pic | string |  |
| user | name: for username, digest: for MD5 encrypted password |  | string |  |
| param |  |  | string |  |
| --index\_id | Unique ID of the entire system, from the server |  | Long integer |  |
| --pic\_name | Face database picture name (.jpg) | 64 | string |  |

Response parameter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| method | Fixed method of obtaining information | sw\_server\_get\_pic | string |  |
| user | name: for username, digest: for MD5 encrypted password |  | string |  |
| result |  |  |  |  |
| --index\_id | Unique ID of the entire system, from the server |  | long integer |  |
| --pic\_name | Face database picture name (.jpg) | 64 | string |  |
| --img\_data | Face image data (base64 coding) |  | string |  |

Request message:

{

"sentcnt":1234,

"user":{"name":"admin","digest":"3ed07a078ad9d9cec0480d2e9736e14e"},

"method": "ivp\_face\_get\_one\_pic",

"param":

{

"SortMode": 0, // Black List and White List type

"TimeStamp":"", //Time stamp

"LibIndex" : 0,

"FlieIndex" : 0,

"PicName": "chenzhen\_12354.jpg"

"DeviceType": 0x10,

}

}

Response message:

{

"sentcnt":1234,

"user":{"name":"admin","digest":"3ed07a078ad9d9cec0480d2e9736e14e"},

"method": "ivp\_face\_get\_one\_pic",

"result":

{

"SortMode": 0, // Black List and White List type

"LibIndex": 0,

"FlieIndex": 0,

"PicName": "chenzhen\_12354.jpg",

"Img": " ...(Picture data base64 encoding, omitted)"

"DeviceType": 0x10,

"error":

{

"errorcode": 0 // The face list is obtained successfully;

}

}

}

## Add a picture to the device

Request parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| dataLength | Data length. The entire json data length, including the dataLength field itself |  | string |  |
| data |  |  |  |  |
| method | Fixed method of obtaining information | sw\_server\_add\_pic | string |  |
| param |  |  | string |  |
| --index\_id | Unique ID of the entire system, from the server |  | long integer |  |
| --attribute | Face attributes:  name: name  gender: gender (1 male, 0 female)  card\_id: ID | Basic parameter information json format (to be extended) | integer |  |
| --img\_data | Face image data (base64 coding) |  | string |  |
| --jpegBase64MD5 | MD5 value of Base64 |  | string |  |

Response parameter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| method | Fixed method of obtaining information | sw\_server\_add\_pic | string |  |
| uuid | Device uuid | 32 | string |  |
| result |  |  |  |  |
| --errorcode | 0 Success  -1002 // Device not registered on server platform  -1003 // token invalid  -1004 // Device not registered on server platform  -2001 //cjson prase err  -2002 //exceed max database num  //get feature failed  -2004 //invaild jpg format  -2005 //sdcard is not available  -2006 //model sdk not init |  |  |  |

Request

{"dataLength":27788,"data":{"method":"sw\_server\_add\_pic","param":{"attribute":{"name":"34","gender":1,"card\_id":"34"},"index\_id":478,"img\_data":"" ,"jpegBase64MD5":""}}}

Response

{"result":{"errorcode":0},"method":"sw\_server\_add\_pic","index\_id":475,"uuid":"12721SSQSQCA"}

## Delete a picture in the list library

Request parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| dataLength | Data length. The entire json data length, including the dataLength field itself |  | string |  |
| data |  |  |  |  |
| method | Fixed method of obtaining information | sw\_server\_del\_pic | string |  |
| param | Pass a parameter. |  | string |  |
| --index\_id | Unique ID of the entire system, from the server. No ID 0 |  | long integer |  |

Response parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| method | Fixed method of obtaining information | sw\_server\_del\_pic | string |  |
| uuid | Device uuid | 32 | string |  |
| Index\_id | Unique ID of the entire system, from the server. No ID 0 |  | long integer |  |
| result |  |  |  |  |
| --errorcode | 0 Success  -1002 // Device not registered on server platform  -1003 // token invalid  -1004 // Device not registered on server platform  -2001 //cjson prase err  -2002 //exceed max database num  //get feature failed  -2004 //invaild jpg format  -2005 //sdcard is not available  -2006 //model sdk not init |  |  |  |

Request:

{"dataLength":80,"data":{"method":"sw\_server\_del\_pic","param":{"index\_id":480}}}

Response:

{"result":{"errorcode":0},"method":"sw\_server\_del\_pic","index\_id":480,"uuid":"12721SSQSQCA"}

## Modify a picture in the list library

Request parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| dataLength | Data length. The entire json data length, including the dataLength field itself |  | string |  |
| data |  |  |  |  |
| method | Fixed method of obtaining information | sw\_server\_update\_pic | string |  |
| param |  |  | string |  |
| --index\_id | Unique ID of the entire system, from the server. |  | long integer |  |
| --attribute | Face attributes:  name: name  gender: gender (1 male, 0 female)  card\_id: ID | Basic parameter information json format (to be extended) | integer |  |
| --img\_data | Face image data (base64 coding) |  | string |  |
| --jpegBase64MD5 | MD5 value of Base64 |  | string |  |

Response parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| method | Fixed method of obtaining information | sw\_server\_add\_pic | string |  |
| uuid | Device uuid | 32 | string |  |
| Index\_id | Unique ID of the entire system, from the server. No ID 0 |  | long integer |  |
| result |  |  |  |  |
| --errorcode | 0 Success  -1002 // Device not registered on server platform  -1003 // token invalid  -1004 // Device not registered on server platform  -2001 //cjson prase err  -2002 //exceed max database num  //get feature failed  -2004 //invaild jpg format  -2005 //sdcard is not available  -2006 //model sdk not init |  |  |  |

Request:

{"dataLength":136,"data":{"method":"sw\_server\_update\_pic","param":{"attribute":{"name":"34","gender":1,"card\_id":"34"},"index\_id":478"img\_data":"" ,"jpegBase64MD5":""}}}

Response:

{"result":{"errorcode":-2},"method":"sw\_server\_update\_pic","index\_id":478,"uuid":"12721SSQSQCA"}

## Add face pictures to the list library in batches

Request parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| dataLength | Data length. The entire json data length, including the dataLength field itself |  | string |  |
| data |  |  |  |  |
| method | Fixed method of obtaining information | sw\_server\_add\_multiple\_pic | string |  |
| param | [], [] Multiple independent |  | string |  |
| --index\_id | Unique ID of the entire system, from the server. |  | long integer |  |
| --attribute | Face attributes:  name: name  gender: gender (1 male, 0 female)  card\_id: ID | Basic parameter information json format (to be extended) | integer |  |
| --img\_data | Face image data (base64 coding) |  | string |  |
| --jpegBase64MD5 | MD5 value of Base64 |  | string |  |

Response parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| method | Fixed method of obtaining information | sw\_server\_add\_multiple\_pic | string |  |
| uuid | Device uuid | 32 | string |  |
| result | [],[]Multiple independence |  |  |  |
| --index\_id | Unique ID of the entire system, from the server. | Only return add errors | long integer |  |
| --errorcode | 0 Success  -1002 // Device not registered on server platform  -1003 // token invalid  -1004 // Device not registered on server platform  -2001 //cjson prase err  -2002 //exceed max database num  //get feature failed  -2004 //invaild jpg format  -2005 //sdcard is not available  -2006 //model sdk not init |  |  |  |

## Batch delete face pictures in the list library

Request parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| dataLength | Data length. The entire json data length, including the dataLength field itself |  | string |  |
| data |  |  |  |  |
| method | Fixed method of obtaining information | sw\_server\_del\_multiple\_pic | string |  |
| param | [id],[id],[id] Just pass a parameter |  | string |  |
| --index\_id | Unique ID of the entire system, from the server. |  | long integer |  |

Response parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| method | Fixed method of obtaining information | sw\_server\_del\_multiple\_pic | string |  |
| uuid | Device uuid | 32 | string |  |
| result | [],[]Multiple independent |  |  |  |
| --index\_id | Unique ID of the entire system, from the server. | Only return failed | Long integer |  |
| --errorcode | 0 Success  -1002 // Device not registered on server platform  -1003 // token invalid  -1004 // Device not registered on server platform  -2001 //cjson prase err  -2002 //exceed max database num  //get feature failed  -2004 //invaild jpg format  -2005 //sdcard is not available  -2006 //model sdk not init |  |  |  |

Request:

{"dataLength":92"，data":{"method":"sw\_server\_del\_multiple\_pic","param":[{"index\_id":424},{"index\_id":423}]}}

Response:

{"method":"sw\_server\_del\_multiple\_pic","uuid":"12721SSQSQCA","result":[{"index\_id":424,"errorcode":0},{"index\_id":423,"errorcode":0}]}

## Get a list of device list libraries (Not used temporarily)

Request parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| method | Fixed method of obtaining information | sw\_server\_get\_sort | string |  |
| user | name: for username, digest: for MD5 encrypted password |  | string |  |
| uuid | Device uuid | 32 | string |  |
| param |  |  |  |  |
| --pic\_sort | Category list:  1: Blacklist  2: White list  3: VIP  4: Custom  It can be distributed by the server later |  | integer |  |

Response parameter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| method | Fixed method of obtaining information | sw\_server\_get\_sort | string |  |
| uuid | Device uuid | 32 | string |  |
| result | [index\_id],[index\_id],[index\_id] |  |  |  |
| --pic\_sort | Category list:  1: Blacklist  2: White list  3: VIP  4: Custom  It can be distributed by the server later |  | integer |  |
| --index\_id | Unique ID of the entire system, from the server. |  | long integer |  |

Request message:

{

"sentcnt":1234,

"user":{"name":"admin","digest":"3ed07a078ad9d9cec0480d2e9736e14e"},

"method": "ivp\_facercg\_get\_face\_rcg\_list",

"param":

{

"ChannelID":0,

"faceRcgPageNum":1,

"faceRcgPageSize":12,

"faceRcgLibType":0

}

}

Response message:

{

"sentcnt":1234,

"user":{"name":"admin","digest":"3ed07a078ad9d9cec0480d2e9736e14e"},

"method": "ivp\_facercg\_get\_face\_rcg\_list",

"result":

{

faceRcgCount":17 // Total list

"faceRcgList":[

{

"faceRcgID":1,

"faceID":"15090",

"faceRcgName":"Alex"

"gender":1,"

age":39,"

jpegURL":"sdcard/face\_20190717/102441\_403.jpg"

},

{

"faceRcgID":2,

"faceID":"15092",

"faceRcgName":"Alex"

"gender":1,"

age":39,"

jpegURL":"sdcard/face\_20190717/102441\_403.jpg"

},

]

},

"error":

{

"errorcode": 0 // The face list is obtained successfully;

}

}

## The device actively synchronizes server data

TCP Request parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| token | Key | 64 | string |  |
| method | Fixed method of obtaining information | sw\_device\_sync\_data | string |  |
| uuid | Device uuid |  | string |  |

TCP Response parameter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| dataLength | Data length. The entire json data length, including the dataLength field itself |  | string |  |
| data |  |  |  |  |
| --method | Fixed method of obtaining information | sw\_device\_sync\_data | string |  |
| --retMsg | Return to prompt |  |  |  |
| --code | Error code |  | integer |  |

# HTTP protocol error returns JSON format

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter Name** | **Description** | **Maximum length / range** | **Type** | **Remark** |
| code | Error code |  | integer |  |
| retMsg | Return description |  | string |  |
| dataObj |  |  |  |  |

**json format**

{

"code":"error",

" retMsg": 0，

" dataObj": {}

}

### Error code

|  |  |
| --- | --- |
| **Error Code** | **Return Description** |
| 100001 | Other |
| 100002 | Failed |
| 100003 | Failed to extract feature value |
| 100004 | File name number duplicate |
| 100005 | The number of faces in the database has exceeded the maximum number. |
| 100006 | Add timeout |
| 100007 | Parameter error |
| 100008 | File size too big |
| 100009 | Low storage space |
| 100010 | File open failed |
| 100011 | No database |
| 100012 | Failed to read picture |
| 100013 | Database file is damaged |
| 100014 | Poor picture quality |
| 100015 | Wrong picture size, width and height cannot be odd |
| 100016 | Failed to detect faces (no faces detected or multiple faces detected) |
| 100017 | Picture format is wrong |
| 100018 | Face area error |
| 100019 | Algorithm creates handle error |
| 100020 | The device is busy |