Face Capture Http API

Protocol version: 0.37
Last update date: 2019/04/14

Revision history

version	Updated	Update staff	update content
0.3	2017/12/26	Chen Chao	1. Added the acquisition and setting of application parameters; 2. Modify the camera_no field in the captured data push content to device_no;
0.4	2018/01/09	Chen Chao	1. The external network control service control is added to the application parameters; 2. Added application parameter reset; 3. Added stream (camera) parameter acquisition/setting/resetting;
0.5	2018/01/29	Chen Chao	Add FTP upload method control for data upload;
0.6	2018/03/21	Gan Miao	1. Add, delete, modify the list to add template image, Wiegand card number, validity period support, and response content to add personnel ID information; 2. Increase age, gender detection enable switch; 3. Add feature data and template (add) image output control switch; 4. Snapshot output increases age and gender;
0.7	2018/03/22	Chen Chao	1. Increase the missing age and gender detection enable switch (optional); 2. Modify the feature data and template (add) image output control switch as optional;
0.8	2018/03/23	Chen Chao	1. Stream (camera) parameters increase brightness setting;
0.9	2018/04/13	Gan Miao	1. Add support for matching query by condition; 2. Add and modify the normalized image field in the portrait data protocol to support the feature extraction by the camera;
0.10	2018/05/23	Gan Miao	1. Add a snapshot/screenshot
0.11	2018/05/30	Chen Chao	Add/modify/view portrait comparison data protocol to increase the initial validity period
0.12	2018/06/05	Chen Chao	Add work mode control
0.13	2018/07/16	Gan Miao	Add gpio/Wiegand device control protocol
0.14	2018/08/13	Gan Miao	Add capture data ACK protocol
0.15	2018/09/25	Gan Miao	Snapshot data to push the content, add matching personnel registration source and personnel list validity

			period attributes
0.16	2018/10/8	Chen Chao	Increase the control attribute of the comparison mode
0.17	2018/11/8	Gan Miao	Increase user authentication code reading and writing
0.18	2018/11/23	Chen Chao	1. The heartbeat packet adds the device serial number and IP address, and expands the heartbeat response packet; 2. Increase equipment time acquisition and modification functions; 3. Increase the restart system function; 4. Add the function of restarting the capture application;
0.19	2019/01/25	Chen Chao	Increase ID card information output
0.20	2019/01/28	Gan Miao	1. Add camera serial number to data upload message 2. Add upload heartbeat switch configuration
0.21	2019/02/18	Xiao He	Reorganize the documents in the client/server mode and data/control functions of the device
0.21	2019/03/15	Chen Chao	Increase the scheduling timetable query and modification.
0.22	2019/03/20	Gan Miao	Increase Wiegand 66 field
0.22	2019/04/01	Gan Miao	Add remote system upgrade and related error codes
0.23	2019/05/07	Gan Miao	Push content to increase portrait quality field
0.24	2019/05/23	Gan Miao	Push content to increase face rotation angle
0.25	2019/07/23	Gan Miao	1. Added Wiegand card number field in the push content 2. Add other comparison modes for application parameters
0.26	2019/08/12	Gan Miao	1. Registrants add user-defined fields
0.27	2019/08/21	Gan Miao	1. Add audio control related protocols
0.28	2019/10/09	Gan Miao	1. Add QR code data to captured data
0.29	2019/11/05	Gan Miao	1. Add configuration and upload of lcd screensaver pictures 2. Add file transfer protocol
0.30	2019/12/18	Xiao He	Modify the main body of the screen saver to be "media", in addition to pictures, it also supports video
0.30	2020/01/09	Xiao He	dv300cv500 supports original image registration
0.30	2020/1/19	Xiao He	av200 supports original image registration
0.31	2020/2/3	Xiao He	http data link supports playback of the returned text
0.32	2020/2/9	Xiao He	Snapshot data into body temperature
0.33	2020/2/14	Xiao He	Change comet description Open beta function: https In data uploading and external network penetration, the ip/domain name starting with https:// means using https (no prefix means http); temporarily

			only dv300cv500 models are supported
0.34	2020/2/19	You Fang	1. Add name password aes encryption 2. Add the captured data to the mask 3. Apply parameters to add body temperature, mask-related thresholds and switches 4. Add gps information
0.35	2020/3/11	Xiao He	Support domain name support during http configuration parameters (only the http protocol part in upload and penetration; only support cv500dv300)
0.36	2020/3/13	Xiao He	1. The original image can be modified by field, only for cv500dv300 2. Ev300 supports http configuration domain name
0.37	2020/4/14	Xiao He	Add a description of the reason for the comparison failure
0.38	2020/4/28	You Fang	1. Add mask detection configuration parameters 2. Increase the screen display picture 3. Configure all camera parameters as optional
0.39	2020/6/29	Xiao He	1. Add an interface for incrementally adding scheduling rules 2. Automatic cleaning switch for expired personnel 3. Name privacy mode switch 4. Minimum face detection size
0.40	2020/8/20	Xiao He	1. Join history query 2. Supports screen display text when responding to snapshot data
0.41	2020/8/21	Xiao He	Add new personnel to deliver the interface. Used to integrate the registration and modification of all previous original images and normalized images

1 1. Overview

1.1 Working mode

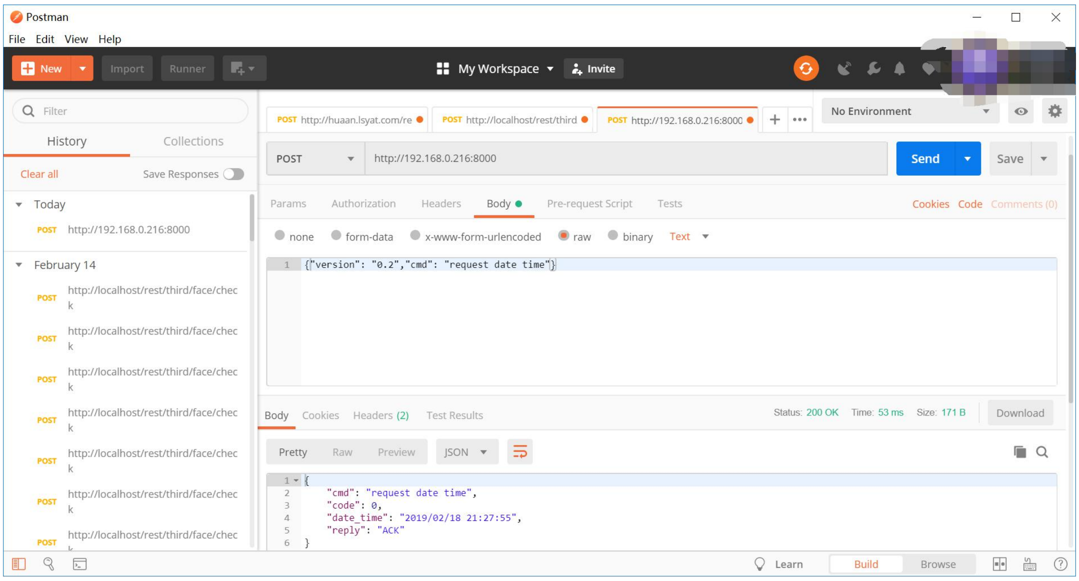
1.1.1 Device as server

Service port: 8000

HTTP POST method

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

This function is suitable for building an http client to access the device when the device IP is reachable in the local area network. The following figure is an example of using Postman to access the device to request its time



POST http://192.168.0.216:8000

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary Text

```
{"version": "0.2","cmd": "request date time"}
```

Body Cookies Headers (2) Test Results

Status: 200 OK Time: 53 ms Size: 171 B

JSON

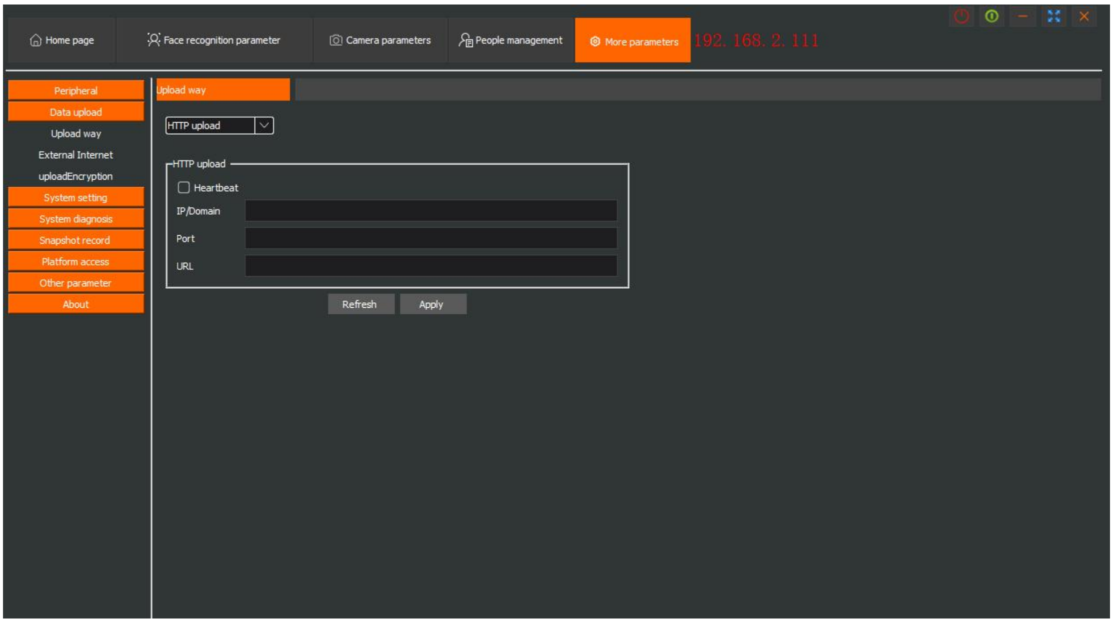
```
{
    "cmd": "request date time",
    "code": 0,
    "date_time": "2019/02/18 21:27:55",
    "reply": "ACK"
}
```

History:

- POST http://192.168.0.216:8000
- POST http://localhost/rest/third/face/check
- POST http://localhost/rest/third/face/check
- POST http://localhost/rest/third/face/check
- POST http://localhost/rest/third/face/check
- POST http://localhost/rest/third/face/check
- POST http://localhost/rest/third/face/check
- POST http://localhost/rest/third/face/check

1.1.2 Device as data client

The data push address can be set through the facedemo software that comes with the device firmware or through the protocol in 1.1.1. The following figure shows the push address setting and data reception display



Upload way

HTTP upload

HTTP upload

☐ Heartbeat

IP/Domain

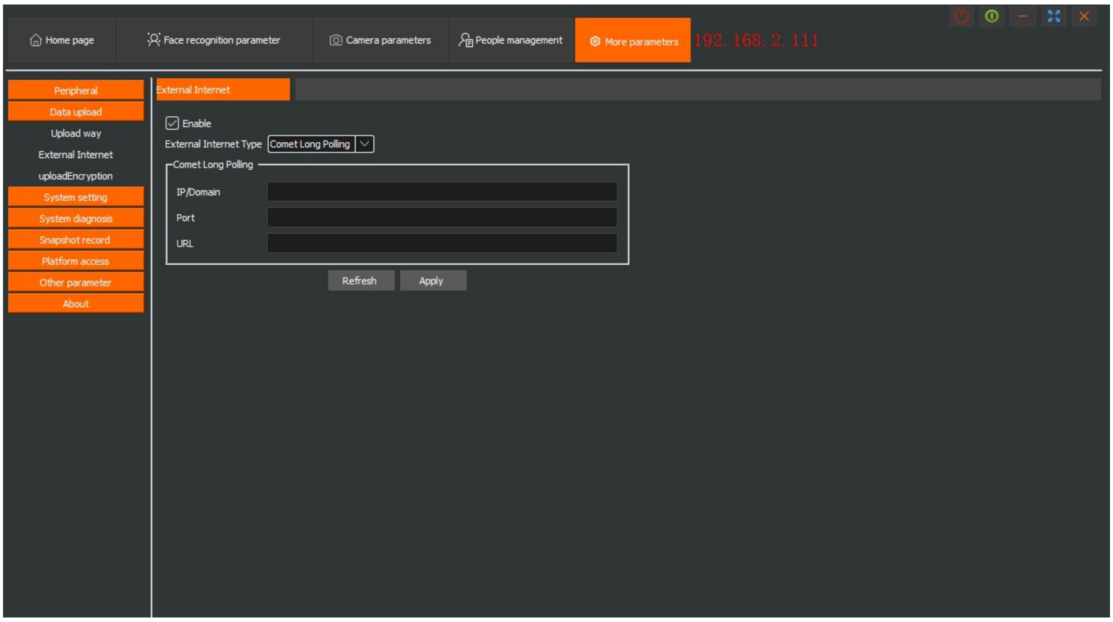
Port

URL

In this mode, you need to use any language (such as java/php/c#) to implement a web server and listen to a request path, and then configure the server's ip port and path to the device, and wait for the device to capture the face. The data will be pushed to the pre-configured address. The content of the pushed data is in 2.2

1.1.3 Device as configuration client

When the device is configured as a client, the parameters are also configured through facedemo.exe, as shown in the figure below



☑ Enable

External Internet Type: Comet Long Polling

Comet Long Polling

IP/Domain

Port

URL

In this mode, the device will actively connect to the http path you set and send a heartbeat. After receiving the heartbeat, your listener can respond to a business command, such as adding a face. If there is no business to deal with, nothing will be returned. , The device will request again after 15 seconds; when the device has processed the possible service commands, in the next question and answer, its request will not be a heartbeat but will be the result of processing the last service command.

In this mode, the server can add a command_id field to the business operation to identify the task number of the current operation, and the device will return this data after processing is completed, so that your application can quickly retrieve the processing of the operation in multithreading and asynchronous result.

PS: For this 15-second interval, you don’t need to think it’s too long, because the device will only wait 15 seconds if it does not receive any response. If it receives any response in http protocol format (including reply heartbeat and business instructions), it will Process immediately and return the result.

Note: If there is no instruction issued, you must not close the connection (call response::close), because the device will reconnect immediately, resulting in an endless loop for both parties; it also cannot return the heartbeat immediately, which will also cause an endless loop for both parties. It should be no Respond or sleep for a few seconds before responding. If you feel that you may need to send data within 15 seconds, it doesn't matter, the data device within 15 seconds can also be received and processed.

Note: If the data upload and external network penetration use the same processor address (the URL is exactly the same), the heartbeat during data upload cannot be turned on!

1.2 Document description

1.2.1 Document Convention

1. The commented line is marked with'#' in the description;
2. All strings are encoded in UTF-8 format;
3. All positions and dimensions are in pixels;
4. json content is case sensitive;
5. All fields without special instructions are required;
6. All lengths specified when referring to string length restrictions do not contain'\0';
7. All JSON objects may expand fields without prior notice, so they cannot be automatically converted to OOP objects using the AOP framework. For example, after a certain version of the captured data, the "match_type" field is added to indicate the channel through which the person passes (such as swiping face, swiping card). If the jackson automatic assembly of the JAVA platform is used, an exception will occur because of the previous object Without this field definition, Jackson will think that you have missed something and dare not handle it privately.
8. You cannot use transfer-encoding: chunked mode, you must use content-length (using ngnix, basically will have this problem!)

1.2.2 General definition

1) Role definition

0: ordinary personnel
1: Whitelisted people
2: Blacklisted personnel
-1: All personnel

1.3 Genera request format

```
{
 # Requester protocol version
 "version": "0.2",

 # Request command (actually the message ID)
 "cmd": "...",

 # Other content
}
```

1.4 General response format

```
{
 # Reply mark
 "reply": "ACK",
  # Response command (specify which command is the response, if it is empty, it means that cmd could not be resolved)
  "cmd": "request persons",

  # Reply code
  "code": 0,

  # devise serial number
  "device_sn":"0123xx-xxxx-xxxxEE"
}
```

Note: When the response code is 1 (the protocol version does not match), other content should include the protocol version information of the responding party ("version": "0.2").

1.5 Data definition

```
/// Personnel update items.
enum ParsonDataFlags {
/*! Unknown. */
  DATA_FLAG_NULL = 0x0,

/*! Update name. */
  DATA_FLAG_NAME = 0x1,

/*! Update the character. */
  DATA_FLAG_ROLE = 0x1 << 1,

/*! Update Wiegand card number. */
  DATA_FLAG_WG = 0x1 << 2,

/*! Update the effective deadline. */
  DATA_FLAG_EFFET = 0x1 << 3,

/*! Update the effective start time. */
  DATA_FLAG_EFFECTSTART = 0x1 << 4,

/*! Update the scheduling mode. */
  DATA_FLAG_SCHEDULE = 0x1 << 5,

/*! Update user-defined fields. */
  DATA_FLAG_USERPARAM = 0x1 << 6,

/*! Update the normalized picture. */
```

```
  DATA_FLAG_NORM_IMAGE = 0x1 << 7,

/*! Update the thumbnail image. */
  DATA_FLAG_REG_IMAGES = 0x1 << 8,

};

// lcd display item
enum LcdDisplayItem {
        Title = 0x1, // Title
        Time = 0x1<<1, // time
        Ip = 0x1<<2, // IP
        RegNum=0x1<<3, // Number of registered persons
        SN=0x1<<4, // serial number
        Wiegand=0x1<<5, // Wiegand card number
        FourG=0x1<<6, // 4G signal
        RecoNum=0x1<<7// Temperature measurement count (the total number of persons
whose temperature is measured)
};
```

1.6 Error code

0: The request has been successfully processed
1: Protocol version does not match
2: The server does not contain the service corresponding to the request
3: The request packet contains illegal fields
4: Authentication failed
5: The system is busy
6: Insufficient resources
7: System function authorization failed
8: System function is authorized
9: The upgrade package does not match
10: File download failed
11: File integrity verification failed
12~19: reserved
20: Data entry reaches the upper limit
21: Record already exists
22: Record does not exist
23: Failed to write data
24: Failed to read data
25: Failed to extract feature value
26: Portrait quality does not meet registration requirements
30: Wiegand card number does not support fuzzy matching

35: Image decoding failed
36: The image is too large, and the jpg image used to extract features cannot exceed 10M
37: Normalized image failed
38: Face size is too small
39: Portrait quality is too poor
40: The number of faces in the image is not 1
41: The face is incomplete in the image

2 2. Detailed agreement definition

2.1 Portrait comparison data management

/ Instructions for the use of feature data and image data**/**

1) What are feature data, registered images, and normalized images?

Feature data: Feature data is the data extracted by the algorithm to mark the characteristics of the portrait by processing the original image of the portrait

It consists of 513 floating-point numbers.

Registered image: The registered image is a thumbnail image of the original portrait image after zooming, which is used for display on the camera LCD and software.

Normalized image: The normalized image is the feature extracted by the algorithm by processing the original image of the portrait to mark the features of the portrait

Image, it is a 150x150 BGR image.

2) How to extract feature data and feature images?

Extract feature data: The feature data has been discarded, so SDK for feature data extraction is currently not provided.

Extract registered images and normalized images: C++ uses the HA_GetJpgFeatureImage interface in libhasdk. C# and Java use HaCamera.twistImage interface in serversdk_java and HttpDemoC# respectively.

3) How to use feature data and feature images?

There are currently two ways to add portraits, one is to use feature data, and the other is to use feature images. In non-special circumstances, feature image addition is currently used. When using the feature image to add, the number of face features is filled with a value of 0, and the face feature data is filled in the array. Fill in 1 for the number of registered images, and the registered image data is the data extracted by the sdk after base64 encoding. The number of normalized images is filled with 1, and the normalized image data is filled with the data extracted by sdk and the data after base64 encoding.

The registration of original jpg images will be gradually opened from 2020, please refer to section 2.1.5

2.1.1 Browse portrait comparison data

2.1.1.1 Request

```
{
 # Requester protocol version
 "version": "0.2",

 # Command
 "cmd": "request persons",

 # Role (0: ordinary personnel. 1: whitelisted personnel. 2: blacklisted personnel. -1: all personnel.)
 "role": -1,

 # Page number (page number, counting from 1. -1: request all data.)
 "page_no": 1,

 # Page size (the number of persons contained in a single page. The maximum does not exceed 20.)
 "page_size": 10,

 # Feature data mark (whether feature data is required. 0: not required, non-zero: required.)
 "feature_flag": 1,

 # Image data mark (whether image data is needed. 0: not required, non-zero: required.)
 "image_flag": 0,

   # Matching mode (0: precise query, non-zero: fuzzy query.)-optional (default is 0)
 "query_mode": 0,

 # Query condition combination, the presence or absence of the corresponding field indicates whether the condition is matched--optional
 "condition": {
        # Personnel ID--optional
 "person_id": "0001",

          # Personnel name--optional
 "person_name": "XiaoMing",

          # 32bit Wiegand access control card number, fill in this field when querying 32bit Wiegand card number, choose one with long_card_no
          #This field does not support fuzzy matching--optional
 "wg_card_no": 8,
        # 64bit Wiegand access card number, fill in this field when querying the 64bit Wiegand card number, choose one of wg_card_no
        #This field does not support fuzzy matching--optional
  "long_card_no": 18446744073709551615,

  # Validity period start time interval starting point, and time_start_end exist at the same time -optional
  # useless means that the field is not included in the query conditions
  "term_start_start":"2017/01/01 00:00:00",

  # The end of the valid period start time interval, which exists at the same time as term_start_start--optional
  # useless means that the field is not included in the query conditions
  "term_start_end": "2020/01/01 00:00:00",

        # The beginning of the expiration time of the validity period, which exists at the same time as time_end--optional
        #never Means permanent invalidation, forever means permanent validity
  "term_start":"2017/01/01 00:00:00",

        # The end of the expiration date of the validity period, which exists at the same time as term_start--optional
        #never Means permanent invalidation, forever means permanent validity
  "term ": "2020/01/01 00:00:00"
     }
}
```

2.1.1.2 Response

```
{
  # Reply mark
  "reply": "ACK",

  # Response command (specify which command is the response)
  "cmd": "request persons",

  # Reply code
  "code": 0,

  # The following fields exist when the response code is 0

  # Total number of records (number of records in the portrait comparison database)
  "total": 2000,
  # Page number (page number, counting from 1. -1: request all data.)
  "page_no": 1

  # Actual number of records (less than or equal to the page size when requested)
  "count": 3

  # Portrait comparison data
  "persons": [
  {
  # Person ID
  "id": "0001",

  # Personnel name
  "name": "XiaoMing",

  # Personnel role
  "role": 1,

 # Personnel classification. Value range: 0~5. 0 means no classification, others are used for time-sharing scheduling.
  "kind": 0,

  # Face feature data unit size
  "feature_unit_size": 128,

  # The number of facial features (maximum is 5)
  "feature_num": 1,

  # Face feature data (the length of the array is the product of the unit size of the face feature data and the number of face features)
  "feature_data": [
  1.0,
           ...
  ],

    # Number of images (total number of sample images, currently only supports 1 image)
    "image_num": 1,

    # Register image data
  "reg_images ": [
      {
      # Image format
      "format": "jpg",
        # Image data (binary data corresponds to base64)
        "image_data":"..."
        },
        …
 ],

   # 32bit Wiegand protocol access card number
 "wg_card_id": 8,

      # 64bit Wiegand Protocol access control card number, when the wg_card_id field does not
exist, use this field as the access control card number
 "long_card_id": 18446744073709551615,

 # Validity period start time (useless means not to use this field)-optional
 "term_start": "2017/11/10 12:00:00",

   # Validity deadline (forever means permanent validity, never means permanent invalid)
      "term": "2017/12/18 16:45:30",

      # User-defined text content, which can be used for camera TTS playback, LCD display, or
      as a user platform customization # attribute use
      "customer_text":"uuid",

      # Scheduling category code. Value range: 1~15.
 "kind": 1
 },
     ...
 ]
}
```

2.1.1.3 Applicable working mode

Device as server

The device is configured as a client (in this case, the device needs to "ask" first to initiate a heartbeat, please add the cammand_id field when responding; the processing result of the device will be given to you in the next "ask")

2.1.2 Upload portrait data

That is to "deliver" personnel to the device

2.1.2.1 request

```
{
  "version": "0.2",

  "cmd": "upload person",

  # Personnel ID (the length does not exceed 19 bytes)
  "id": "0001",

# The following attributes are not required
# When adding, it is not necessary to pass any pictures, then it may be used in the mode of just swiping card or swiping (identity)
# When adding, you can even pass only the id and omit even the card number. At this time, it can still work normally in the (identity) only mode
# When modifying, replace the passed attributes with the old ones, and ignore the ones not passed

  # Person's name (the length does not exceed 63 bytes,Note that it is not a character)
  # Optional when modifying
  "name": "XiaoMing",

# Personnel role. (1: Whitelist 2: Blacklist)
  # When adding, do not pass the default is 1
  "role": 1,

# Personnel classification. Value range: 0-15. 0 means no classification, others are used for time-sharing scheduling.
  # When adding, do not pass the default value is 0
  "kind": 0,

  # Register image (base64)
# Support jpg and png. The width and height of png should not be too large. In theory, the resolution of jpg is not required. Keep the file smaller than 4M, but if it is too large, it will be very slow.
  # It is recommended that the resolution be controlled within 200W
  "reg_image": "/9j/4AAQSkZJRgABAQAAAQABAAD/...",

  # 32bit Wiegand protocol access card number
  # Note: For comparison when connecting Wiegand 26 34 reader
  "wg_card_id": 8,

  # 64bit access control card number, when the wg_card_id field does not exist, use this field as the access control card number
  # Note: For comparison when connecting Wiegand 37 66 card reader
  "long_card_id": 18446744073709551615,

  # Validity period start time (useless means not to use this field)
  # When adding, do not pass the default is useless
  "term_start": "2017/11/10 12:00:00",

  # Validity deadline (forever means permanent validity, never means permanent invalid)
  # When adding, do not pass the default forever
  "term": "2017/12/18 16:45:30",

  # User-defined text content (no more than 68 bytes)
  # Use as follows:
  # ①The platform user id is UUID, and the device can only store a 19-byte number due to historical problems, you can use this field to correspond
  # ②The voice broadcast or screen display after successful personnel comparison can use this field as part of the output
  "customer_text":"",

# The following content is advanced usage

  # Normalized image data
  # Choose one from reg_image, if both are passed, normal_image takes precedence
  # Normalized map is an advanced usage, generally not used
# Simply put, the picture needs to be processed when it reaches the device. This data is a processed result
  # Therefore, passing this value to add or modify personnel will be very fast (that is, no operations, only IO)
  # There are two sources of this data, one is taken from one device, and it can be synchronized to other devices at this time
  # The second is to use the underlying SDK (function exported by dll/so) to generate from the original picture
  "normal_image": "",

  # Staff update mode (0: automatic 1: add 2: modify)
  # The default is automatic, that is, overwrite if there is, add if not
  # If manually specified as a new addition, if it exists, an error will be reported without performing the entry operation
  # Similarly, if it is specified as a modification, if it does not exist, an error will be reported and the operation of updating the given field will not be performed
  # This field is useful in some cases, such as the need to passively determine whether a person has been deleted from the device (due to front-end misoperation)
  # When this happens, I originally wanted to update the field, but it became a new operation, which caused data confusion
  "upload_mode":1,

  # Thumbnail (base64 in jpg format. The avatar is displayed on the screen after the card is successfully clocked in)
```

```
 # Used in conjunction with normal_image, in non-advanced mode, this value will be automatically generated from reg_image
 "thumb_image": "/9j/4AAQSkZJRgABAQAAAQABAAD/..."
}
```

2.1.2.2 Response

```
{
 "reply": "ACK",

 "cmd": "upload person",

 # Status code, possible values:
 # 0: Success, personnel have been added or modified
 #2: The device has not yet implemented this function (version earlier than August 21, 2020)
 #3: The given data is incorrect (if this response code appears, there will be desc to describe the specific error in text)
#6: Insufficient resources. Insufficient device memory
# 20: Data entry reaches the upper limit. Equipment staff is full
#21: The record already exists. The upload_mode in the request message body is 1, and the device already has a user with the corresponding id
#22: The record does not exist. The upload_mode in the request message body is 2, and there is no user with the corresponding id in the device
# 25: Failed to extract the face. No face in the image
#35: Image decoding failed. The given image is not jpg or png
#36: The image is too large. JPG file cannot be larger than 10M
#37: Normalization failed. Only appears in normal_image data that gives the error
 #38: Face size is too small
 #39: Face quality is too poor
 #40: There are multiple faces in the image
 #41: The face is incomplete in the image
 "code": 0,

 # Id passed in the request
 "id": "0001"
}
```

2.1.3 Add portrait comparison data

2.1.3.1 request

```
{
 # Requester protocol version
 "version": "0.2",
  # Command
  "cmd": "add person",

  # Personnel ID (the length does not exceed 19 bytes)
  "id": "0001",

  # Person's name (the length does not exceed 15 bytes)
  "name": "XiaoMing",

# Personnel role. The value is not less than 0.
  "role": 1,

  # Whether to overwrite the existing ID after opening the registration ID to overwrite the registration again
  "cover": false,

# Personnel classification. Value range: 0-15. 0 means no classification, others are used for time-sharing scheduling.
  "kind": 0,

  # Face feature data unit size
  # No longer used, fixed delivery 0
  "feature_unit_size": 0,

  # Number of face features
  # No longer used, fixed delivery 0
  "feature_num": 0,

  # Face feature data (the length of the array is the product of the unit size of the face feature data and the number of face features)
  # No longer used, always pass empty array
  "feature_data": [],

   # Number of registered images (total number of sample images, currently only supports 1 image)
  "image_num": 1,

   # Register image (template image) data
  "reg_images": [
      {
       # Image format
       "format": "jpg",

       # Image data (binary data corresponds to base64)
       "image_data":"..."
```

```
        },
        …
  ],

    # Number of normalized images (support 1~5)
    # Please send 1
  "norm_image_num": 1,

    # Normalized image data
  "norm_images": [
        {
  # Image width
  "width": 150,

            # Image width
  "height": 150,

  # Image data (binary data corresponds to base64)
  "image_data":"..."
        },
        ...
  ],

    # 32bit Wiegand protocol access card number
  "wg_card_id": 8,

    # 64bit access control card number, when the wg_card_id field does not exist, use this field as
the access control card number
  "long_card_id": 18446744073709551615,

  # Validity period start time (useless means not to use this field)-optional
  "term_start": "2017/11/10 12:00:00",

    # Validity deadline (forever means permanent validity, never means permanent invalid)
    "term": "2017/12/18 16:45:30",

    # User-defined text content, which can be used for camera TTS playback, LCD display, or as a
    user platform custom attribute# Use
    "customer_text":"uuid"
}
```

2.1.3.2 Response

```
{
 # Reply mark
 "reply": "ACK",

 # Response command (specify which command is the response)
 "cmd": "add person",

 # Reply code
 "code": 0,

   # Added person ID
 "id": "0001"
}
```

2.1.3.3 Applicable working mode

Device as server

The device is configured as a client (at this time, the device needs to "ask" to initiate a heartbeat, please add the command_id field when responding; the processing result of the device will be given to you in the next "ask")

2.1.4 Modify portrait comparison data

2.1.4.1 request

```
 {
 # Requester protocol version
 "version": "0.2",

 # Command
 "cmd": "update person",

 # Personnel ID (the length does not exceed 19 bytes)
 "id": "0001",

 # Update flag, processed by bit, when the corresponding bit is 1, it means that an item needs to
    be modified
   # For the meaning of the corresponding bits, see chapter 1.5 enum ParsonDataFlags. Note: If
    this field is not available, all updates are made.
   "update_flags": 2,

 # Person's name (the length does not exceed 15 bytes)
 "name": "XiaoMing",

# Personnel role. The value is not less than 0.
 "role": 1,
# Personnel classification. Value range: 0-15. 0 means no classification, others are used for time-sharing scheduling.
 "kind": 0,

 # Face feature data unit size
 # No longer used, always pass 0
 "feature_unit_size": 0,

 # Number of face features
 # No longer used, always pass 0
 "feature_num": 0,

 # Face feature data (the length of the array is the product of the unit size of the face feature data and the number of face features)
 # No longer used, fixed pass empty array
 "feature_data": [],

  # Number of registered images (total number of sample images, currently only supports 1 image)
 "image_num": 1,

  # Register image (template image) data
 "reg_images": [
     {
      # Image format
      "format": "jpg",

      # Image data (binary data corresponds to base64)
      "image_data":"..."
     },
     …
 ],

  # Number of normalized images (support 1~5)
  # Please send 1
 "norm_image_num": 1,

  # Normalized image data
 "norm_images": [
     {
 # Image width
 "width": 150,

       # Image width
 "height": 150,

 # Image data (binary data corresponds to base64)
 "image_data":"..."
      },
      ...
 ],

   # 32bit Wiegand protocol access card number
 "wg_card_id": 8,

   # 64bit access control card number, when the wg_card_id field does not exist, use this field as
the access control card number
 "long_card_id": 18446744073709551615,

 # Validity period start time (useless means not to use this field)-optional
 "term_start": "2017/11/10 12:00:00",

   # Validity deadline (forever means permanent validity, never means permanent invalid)
   "term": "2017/12/18 16:45:30",

   # User-defined text content, which can be used for camera TTS playback, LCD display, or as a
   user platform custom attribute# Use
"customer_text":"uuid"

}
```

2.1.4.2 Response

```
{
 # Reply mark
 "reply": "ACK",

 # Response command (specify which command is the response)
 "cmd": "update person",

 # Reply code
 "code": 0,

   # Modified personnel ID
 "id": "0001"
}
```

2.1.4.3 Example

```
 {
 # Requester protocol version
 "version": "0.2",

 # Command
 "cmd": "update person",

 # Personnel ID)
 "id": "0001",

 # Update logo update name and role
 "update_flags": 3,

 # Need to update id
 "id": "123",

 # Person's name (the length does not exceed 15 bytes)
 "name": "XiaoMing",

# Personnel role. The value is not less than 0.
 "role": 1
}
```

2.1.4.4 Applicable working mode

Device as server

The device is configured as a client (at this time, the device needs to "ask" to initiate a heartbeat, please add the command_id field when responding; the processing result of the device will be given to you in the next "ask")

2.1.5 Delete portrait comparison data

2.1.5.1 request

```
{
 # Requester protocol version
 "version": "0.2",

 # Command
 "cmd": "delete person(s)",

 # Mark
 # -1: Delete by ID
 # 0: Delete all ordinary personnel
 #1: Delete all whitelisted people
 #2: Delete all blacklisted persons
 #3: Delete all personnel records
 "flag": 1,

# Person ID. When deleting by ID, the value should not be an empty string.
 "id": "0001"
}
```

2.1.5.2 Response

```
{
 # Reply mark
 "reply": "ACK",

 # Response command (specify which command is the response)
 "cmd": "delete person(s)",

 # Reply code
 "code": 0,

   # Deleted person ID (when deleting by role, this field is an empty string)
 "id": "0001"
}
```

2.1.5.3 Applicable working mode

Device as server

The device is configured as a client (at this time, the device needs to "ask" to initiate a heartbeat, please add the command_id field when responding; the processing result of the device will be given to you in the next "ask")

2.1.6 Add portrait comparison data

2.1.6.1 request

```
{
 # Requester protocol version
 "version": "0.2",
  # Command
  "cmd": "add person jpg",

  # Personnel ID (the length does not exceed 19 bytes)
  "id": "0001",

  # Person's name (the length does not exceed 15 bytes)
  "name": "XiaoMing",

# Personnel role. The value is not less than 0.
  "role": 1,

# Personnel classification. Value range: 0-15. 0 means no classification, others are used for time-
sharing scheduling.
  "kind": 0,

  # Whether to overwrite the existing ID after opening the registration ID to overwrite the
registration again
  "cover": false,

   # Number of registered images (total number of sample images, currently only supports 1 image)
  "image_num": 1,

   # Register image (template image) data
  "reg_images": [
      {
       # Image format
       # Only supports jpg, the caller needs to ensure the file format, pay attention, the suffix is
      jpg but the content does not match, it will not work
       "format": "jpg",

       # Image data (binary data corresponds to base64)
       "image_data":"..."
      },
      …
  ],

   # 32bit Wiegand protocol access card number
  "wg_card_id": 8,

   # 64bit access control card number, when the wg_card_id field does not exist, use this field as
the access control card number
  "long_card_id": 18446744073709551615,
 # Validity period start time (useless means not to use this field)-optional
 "term_start": "2017/11/10 12:00:00",

  # Validity deadline (forever means permanent validity, never means permanent invalid)
  "term": "2017/12/18 16:45:30",

  # User-defined text content, which can be used for camera TTS playback, LCD display, or as a
  user platform custom attribute# Use
  "customer_text":"uuid",
}
```

2.1.6.2 Response

```
{
 # Reply mark
 "reply": "ACK",

 # Response command (specify which command is the response)
 "cmd": "add person jpg",

 # Reply code
 "code": 0,

  # Added person ID
 "id": "0001"
}
```

2.1.6.3 Applicable working mode

Device as server

The device is configured as a client (at this time, the device needs to "ask" to initiate a heartbeat, please add the command_id field when responding; the processing result of the device will be given to you in the next "ask")

2.1.7 Modify portrait comparison data

2.1.7.1 request

```
 {
 # Requester protocol version
 "version": "0.2",

 # Command
 "cmd": "update person jpg",
 # Personnel ID (the length does not exceed 19 bytes)
  "id": "0001",

 # Person's name (the length does not exceed 15 bytes)
 "name": "XiaoMing",

  # Update flag, processed by bit, when the corresponding bit is 1, it means that an item needs to
be modified
  # For the meaning of the corresponding bits, see chapter 1.5 enum ParsonDataFlags. Note: If
    this field is not available, all updates are made.
  # Different from 2.1.3, because this request is modified from the original image, so when
    update_flags contains DATA_FLAG_NORM_IMAGE or DATA_FLAG_REG_IMAGES, the
    image will be modified
  "update_flags": 1,

# Personnel role. The value is not less than 0.
 "role": 1,

# Personnel classification. Value range: 0-15. 0 means no classification, others are used for time-sharing scheduling.
 "kind": 0,

  # Number of registered images (total number of sample images, currently only supports 1 image)
 "image_num": 1,

  # Register image (template image) data
 "reg_images": [
     {
      # Image format
      # Only supports jpg, the caller needs to ensure the file format, pay attention, the suffix is
     jpg but the content does not match, it will not work
      "format": "jpg",

      # Image data (binary data corresponds to base64)
      "image_data":"..."
     },
     …
 ],

  # 32bit Wiegand protocol access card number
 "wg_card_id": 8,

  # 64bit access control card number, when the wg_card_id field does not exist, use this field as
```

```
the access control card number
 "long_card_id": 18446744073709551615,

 # Validity period start time (useless means not to use this field)-optional
 "term_start": "2017/11/10 12:00:00",

  # Validity deadline (forever means permanent validity, never means permanent invalid)
  "term": "2017/12/18 16:45:30",

  # User-defined text content, which can be used for camera TTS playback, LCD display, or as a
  user platform custom attribute# Use
"customer_text":"uuid"

}
```

2.1.7.2 Response

```
{
 # Reply mark
 "reply": "ACK",

 # Response command (specify which command is the response)
 "cmd": "update person",

 # Reply code
 "code": 0,

  # Modified personnel ID
 "id": "0001"
}
```

2.1.7.3 Example

```
 {
 # Requester protocol version
 "version": "0.2",

 # Command
 "cmd": "update person",

 # Personnel ID)
 "id": "0001",
 # Update logo update name and role
 "update_flags": 3,

 # Need to update id
 "id": "123",

 # Person's name (the length does not exceed 15 bytes)
 "name": "XiaoMing",

# Personnel role. The value is not less than 0.
 "role": 1
}
```

2.1.7.4 Applicable working mode

Device as server

The device is configured as a client (at this time, the device needs to "ask" to initiate a heartbeat, please add the command_id field when responding; the processing result of the device will be given to you in the next "ask")

2.2 Snapshot data

2.2.1 Snapshot data push

2.2.1.1 Push content

```
{
 # Pusher protocol version
 "version": "0.2",

 # Command
 "cmd", "face",

 # Serial number
 "sequence_no": 1,

 # Device (camera) number
 "device_no": "0001",
  # Device (camera) serial number
 "device_sn": "000001-000002-0000BC",

 # Point number
 "addr_no": "0001",

 # Point name
 "addr_name": "XX Road",

 # Capture time
  "cap_time": "2017/12/18 16:45:30.003",

 # Real-time capture logo
 "is_realtime": 1,

 # Reasons for inaccessibility
 # This field is available when the personnel are valid but not passable
 # "Comparison failed image" must be turned on to send out
 # Note: After the failed comparison image is turned on, this field must be used as the basis for
judgment to avoid the situation that the gate is opened even when the person is not in the
scheduling time.
 # Restricted personnel: -2, personnel expired: -3, not in scheduling time: -4, holidays: -5,
temperature is too high: -6, mask: -7, without helmet: -8, card number is not registered:- 9,
Personal ID does not match: -10, Unauthorized: -11
 "match_failed_reson":0,

 # Comparison result (0: no comparison. -1: comparison failed. Value greater than 0: the
confidence score when the comparison is successful (100 points system).)
 "match_result": 80,

 # Match personnel information
 "match":
  {
      # Whether to encrypt (the name and ID fields after encryption are AES encrypted binary
data to base64, and the remaining fields are not entered        Line encryption)
     "is_encryption":false,

      # Person ID
     "person_id": "0001",

     # Personnel name
     "person_name": "XiaoMing",
        # Personnel role
        "person_role": 1,

        # Registered image (template image) format
        "format": "jpg",

        # Register image (template image) data (binary data corresponds to base64)
        "image": "…",

         # Person registration source none: No such information. software: Registered software
registration.
         # cluster: Non-sense input registration. sync: Cloud synchronization registration. unknown:
unknown
        "origin ": "software",

         # Person list valid attribute none: no such information permanent_list: permanent valid list
         # temporary_list: temporary valid list invalid_list: permanent invalid list unknown:
unknown
        "person_attr": "permanent_list",

     # User-defined text content, which can be used for camera TTS playback, LCD display, or as a
         user platform customization # attribute use
         "customer_text":"uuid"
   },

  # Panoramic image mark (whether it contains a panoramic image)
  "overall_pic_flag": true,

  # Panoramic image (the panoramic image flag is true)
  "overall_pic":
  {
  # Image format
  "format": "jpg",

  # Image data (binary data corresponds to base64)
  "data": "...",

  # The face position is at the X coordinate of the panorama
  "face_x": 150,

  # The face position is at the Y coordinate of the panorama
  "face_y": 150,

  # The width of the face in the panorama
```

```
"face_width": 200,

# Face height in panorama
"face_height": 200
},

# Close-up image flag (whether it contains a close-up image)
"closeup_pic_flag": true,

# Close-up image (close-up image flag is true)
"closeup_pic":
{
# Image format
"format": "jpg",

# Image data (binary data corresponds to base64)
"data": "...",

# The face position is at the X coordinate of the close-up image
"face_x": 0,

# The face position is at the Y coordinate of the close-up image
"face_y": 0,

# The width of the face in the close-up image
"face_width": 200,

# Face height in close-up
"face_height": 200
},

# Video logo (whether video is included)
"video_flag": true,

# Video (Video flag is true)
"video":
{
# Video start time
"start_time": "2017/12/18 16:45:30.003",

# Video end time (seconds)
"end_time": "2017/12/18 16:45:32.003",

# Video format (avi/mp4)
```

```
 "format": "mp4",

 # Video data (binary data corresponds to base64)
 "data": "..."
 },

 # Personnel attribute information
 "person":
   {
 # Gender (male means male, female means female, empty string means no such information)
 "sex": "male",

     # Age (0 means no such information)
     "age": 24,

     # Whether to wear a safety helmet. none: No helmet is worn. white: White helmet. blue: blue
helmet.
     # orange: Orange safety helmet. red: red helmet. yellow: yellow safety helmet.
     "hat": "none",

     # Body temperature
 "temperatur": 36.9,

     # Whether to wear a mask
 "has_mask": true,

     # Portrait quality. (0~100)
     "face_quality": 70,

     # Torsion angle. The head rotation plane is not in the same plane as the picture. The value
range is -90~90.
     "turn_angle": 0,

     #Plane rotation angle. The head rotation plane is in the same plane as the image. The value
range is -90~90.
     "rotate_angle": 0,

     # 32bit Wiegand protocol access card number
     "wg_card_id": 8,

     # 64bit Wiegand Protocol access control card number, when the wg_card_id field does not
exist, use this field as the access control card number
 "long_card_id": 18446744073709551615
   },
# Face feature data
"feature_data": [
1.0,
...
],

# ID information
"id_card": {
# Numbering
"number": "511xxx",

    # Name
    "name": "Zhang San",

    # date of birth
    "birthday": "19801001",

    # Gender (male: male female: female)
    "sex": "male",

    # Nation
    "national": "Chinese",

    # address
    "residence_address": "Sichuan Province xxx",

    # issuing authority
    "organ_issue": "xxxpublic security bureau",

    # Validity start time
    "valid_date_start": "20051001",

    # Expiration date
    "valid_date_end": "20151001"
},

# QR code data, this field is present when the QR code is detected, otherwise there is no such field
"qr_code": {
# QR code type
"qr_type": "QR-Code",

    # QR code data
```

```
     "qr_data": "admin"
    },

    # location information
    # This information is only available when an external gps device is connected
    "gps":{
        # Longitude (string)
          "n":"...",
          # Latitude (string)
          "e":"...",
          # Reference number of satellites
          "strength":0
    }
}
```

2.2.1.2 Push content response

Note: The server must reply as follows. This is not for the success of your server logic, as long as you receive it, you must return. Previously, after the client received the data, it was judged that the person had been deleted from the server, he would not return or the return code was not 0. In this way, the device will be stuck on this record and keep sending.

As long as you receive it, you must return as follows

```
{
  "reply": "ACK",

  "cmd": "face ",

  # Reply code
  "code": 0,

  # Serial number (need to be consistent with the one pushed)
  "sequence_no": 1,

   # Capture time (need to be the same as the one pushed,Don’t reformat, no punctuation can be wrong)
  "cap_time": "2017/12/18 16:45:30.003",

  # Gate control (optional), used in online mode
  "gateway_ctrl":{
      # Device type, gpio: io device wiegand: Wiegand device
  "device_type": "gpio",

      # When the device type is gpio, this field indicates the io number
      # When the device type is wiegand, this field indicates the Wiegand card number,
    "device_no": 1,

    # Wiegand protocol access control card number (long integer), this field is used in wg36,
wg44, wg66 protocols
    # When using this field, device_no must be assigned to 0
 "long_card_id": 9223372036854775807,


 # Control mode, test: test mode force: forced mode scene: scene mode
 # Test mode: used for testing, no record of opening information is generated;
    # Forced mode: the output signal will be forced to the device;
    # Scene mode: Whether to output a signal to the device depends on whether the device is
enabled;
 "ctrl_mode": "force",

 # Personnel id, only used in online mode, used to generate gate opening records
 "person_id": "0001"
  },
  # The voice to be played (optional)
  "tts":{
      # Voice text (required,UTF8 encoding), the length does not exceed 64 bytes
      "text":"Welcome home, Zhang San"
  },
  # The text that needs to be displayed on the screen (optional)
  "Text_display":{
    # Text display position
 "position":{
 "x":0,
 "y":500
 },
    # Text existence time unit ms
 "alive_time": 1000,
 # font size
 "font_size": 100,
    # Text interval
 "font_spacing": 1,
 # Text color format ARGB
 "font_color": "0xffffffff",
 # Text content
 "text":"Hello"
   }
}
```

2.2.1.3 Applicable working mode

Device configuration data client

2.2.2 Historical data query

2.2.2.1 request

```
{
	"version":"0.29",
	"cmd":"request records",
	# Page number (starting from 1)
	"page_no":2,
	# Page size (up to 20)
	"page_size":20,
	# Whether to retrieve the close-up picture (0: No 1: Yes)
	"face_image_flag":1,
	# Whether to retrieve the registration image (0: No 1: Yes)
	"reg_image_flag":0,

#The fields below are optional

	# Record generation time, unix timestamp, the following example means 2020/8/20 1:1:1
	# It is not necessary to pass the time interval
	#	But if you need to query the time interval, you must also give time_end
	"time_start":1597856461,
	"time_end":1597885261,
	# Same as time interval query, here is a given score interval for query
	#	Same as above, if you need to follow this condition, you need to pass two fields
	"score_start":85,
	"score_end":92,
	# Query by upload status (1: successfully uploaded to the server 0: not successfully uploaded
to the server)
	"upload_state":0

	# For the id and name in the query condition, whether to use fuzzy matching (0: No 1: Yes)
	# Do not pass the default value is 0, it is invalid when the id and/or name query is not passed,
and it is not necessary to provide
	"fuzzy_flag":0,
	# Query by id
	"id":"linxing",
	# Query by name
	"name":"Lin Xing"
}
```

2.2.2.2 response

```
{
 "cmd": "request records",
     "code": 0,
     # Data volume on this page
 "count": 20,
     "device_sn": "01231B-87182C-C1C6EE",
     # Page number
 "page_no": 1,
     "reply": "ACK",
     # Eligible total data
     "total": 4379,
     # Data on this page
     # Data examples mainly look at the first two
     # One is a successful comparison, the other is a stranger
     # Some fields that failed to compare are absent
     # Note: Even if you pass the acquisition picture, there may not be pictures, including the registration picture, because the actual situation may be very complicated
 "records": [
 {
 "body_temp": 36.303585052490234,
 "customer_id": "",
 "id": "YunZhongShuangYue",
 "match_failed_reson": 0,
 "match_type": 1,
 "name": "肖何",
 "person_name_ext": "",
                "reg_image": "/9j/4AAQSkZJRgABAQAAAQABAAD/...",
 "role": 1,
 "score": 94,
 "sequence": 34894,
 "time": "2020/06/05 11:51:18.454",
 "upload_state": 1
 },
 {
 "body_temp": 36.335811614990234,
 "match_failed_reson": -11,
 "match_type": 0,
 "score": -1,
 "sequence": 34903,
 "time": "2020/07/01 17:14:18.116",
 "upload_state": 0
 },
...
 ]
}
```

2.2.2.3 Applicable working mode

Device as server
Device configuration client

2.3 Application configuration

2.3.1 Get application parameters

2.3.1.1 request

```
{
 # Requester protocol version
 "version": "0.2",

 # Command
 "cmd": "request app params"
}
```

2.3.1.2 Response

```
{
 # Reply mark
 "reply": "ACK",

 # Response command (specify which command is the response)
 "cmd": "request app params",

 # Reply code
 "code": 0,

 "device_info": {
# Device (camera) number. The length is less than 32.
 "device_no": "0001",

# Point number. The length is less than 32.
 "addr_no": "0001",

# Point name. The length is less than 96.
 "addr_name": "XX Road"
 },

 # External network control service
 "extranet": {
 # Control switch
     "enable": true,

     # Control mode (optional field http/tcp, if there is no such field, the default is tcp). Only exists when the control switch is true.
 "mode": "http",

# Server IP address. Only exists when the control switch is true.
     "server_ip": "192.168.0.23",

     # Server domain name (only one exists with server_ip; only exists in http mode)
     # This item can be provided separately; if server_ip is also provided, server_ip will be given priority
     "server_domain": "abcom",

# Server port. Only exists when the control switch is true.
     "server_port": 2001,

     # Server URL. Only exists when the control switch is true and the control mode is http.
 "url": ""
 },

 # Data upload control
 "upload_info": {
 # Transmission method (http/tcp/ftp/ webservice /none)
 "method": "http",

 # Server IP (transmission mode is not none exists)
     "server_ip": "192.168.0.172",

     # Server domain name (only one exists with server_ip; only exists in http mode)
     # This item can be provided separately; if server_ip is also provided, server_ip will be given priority
     "server_domain": "abcom",

 # Server port number (transmission mode is not none exists)
 "server_port": 2000,

# The length of the service URL is less than 102. (Http and webservice transmission methods exist, no longer include IP and port)
```

```
  "service_url": "face.php",

  # Authorized user name (exists only in ftp transmission mode)
  "user": "user",

  # Authorization password (exists only in ftp transmission mode)
  "password": "123",

# Upload path, support specified directory or full path, the length is less than 70. (Only exists in ftp transmission mode)
  # Directory: End with'/', the file name will be automatically generated by the snapshot system.
  # Full path: All configurations that do not end with'/' will be treated as full path mode (note whether the same name exists, if it exists, there will be an overwrite).
  # Support the following variables (variables are composed of pairs of'%') to form a directory or full path. Unrecognized variable names will not be translated.
# %ADDR_NO% Point number. Apply to all pictures.
# "%ADDR_NAME%" Point name. Apply to all pictures.
# "%DEVICE_NO%" The device number. Apply to all pictures.
# "%SEQUENCE%" Snapshot data sequence number (6 digits). Applied to a group of pictures.
# "%RAND%" Random number (6 digits). Applied to a group of pictures.
# "%PID%" Personnel number, XXXXXX when there is no comparison information. Applied to a group of pictures.
# "%PNAME%" Person's name, XXXXXX when unmatched information. Applied to a group of pictures.
# "%PROLE%" Personnel role, X for unmatched information. (0: ordinary person 1: white list 2: black list) applied to a group of pictures.
# "%Y%" The year of the capture (4 digits). Apply to a single picture.
# "%M%" The month of the capture (2 digits). Apply to a single picture.
# "%D%" Snapshot time (2 digits). Apply to a single picture.
# "%H%" The hour of the capture (2 digits). Apply to a single picture.
# "%M%" Snapshot time and minute (2 digits). Apply to a single picture.
# "%S%" Snapshot time and seconds (2 digits). Apply to a single picture.
# "%MS%" Milliseconds (3 digits) when capturing. Apply to a single picture.
# "%INDEX%" The index (1 bit) located in a group of captured pictures, the index value starts from 1. Apply to a single picture.
      "pattern": "/"
  },

  # Face capture control
  "face": {
      # Minimum detection face size
      "min_face_size":120,
      # Name privacy mode
      "name_privacy":true,
    # Compare switch. true: open false: off.
    "enable_match": true,

# Be sure of the score. Value range: 70~100. (Exists when the comparison switch is true)
    "match_score": 70,

    # Remove duplicate switches. true: open false: off.
    "enable_dereplication": true,

    # Repeat timeout. Value range: 3~60. (Exists when the remove duplicate switch is true)
    "derep_timeout ": 3,

    # Interval unit of car audio flipping on and off: minutes
    "audio_flip_interval": 5,

    # Image output format. overall: panorama closeup: close-up both: panorama + close-up none:
no picture.
    "pic_form": "closeup",
    # Register image (template image) output switch
    "enable_output_reg": false,

  # Characteristic data output switch
    "enable_output_feature": false,

    # Age detection switch
  "enable_age": false,

    # Gender detection switch
  "enable_sex": false,

  # Living body detection switch
    "enable_alive": false,

    "body_temperature": {

        # Body temperature detection switch
        "enable":true

        # Opening the body temperature limit value (above this body temperature will not be
    opened)
        "limit":38.0,

        # Outdoor mode
        "outdoor":true,
        # Multi-frame confirmation
        "multi_frame_confir":false,

        # Temperature measurement range minimum
        "range_min":36.0,

        # Maximum temperature measurement range
        "range_max":42.0,

        # Temperature correction value
        "correction_value":0.25,

        # Use Fahrenheit
        "fahrenheit_unit":false
    },

    "mask":{
        # Mask detection switch
        "enable":true,

        # No access switch without a mask
        "must_wear_mask":true,

        # Without mask alarm switch
        "no_mask_alarm":true
    },
    # Remote upgrade configuration
    "remote_upgrader":{
        # Remote upgrade switch
        "enable":true
    }
},
  # Gate control
  "gate_control":{

    # Output mode gpio: relay output wiegand: Wiegand output
    "output_mode": "gpio"

    # Open gate mode
    #0: Not initialized
    #     1: Correspondence-face-swiping function
    #     2: Correspondence-swipe ID card to open the gate function
    #     3: Correspondence---Swipe face + ID card opening function
    #     4: Correspondence---swiping card opening function
    #     5: Correspondence-swipe face + swipe card opening function
    #     6: Correspondence-pass-by opening function
    #     7: Correspondence---swipe face + white list gate opening function or swipe card + white list gate opening function
    #     8: Correspondence-swipe face or ID card
    #     9: Correspondence-swipe card and take screenshot
    #     10: Correspondence-brush face or (brush face + ID card)
    #     11: Correspondence-swiping the ID card (whitelist) only needs the ID number in the whitelist
    #     12: Correspondence-swipe ID card to open the gate (no need to compare)
    #     13: Correspondence-swipe the face or swipe the ID card to open the gate (the ID card does not need to be compared)
    #     20: Correspondence-face swiping or RFID panoramic snapshot
    "gate_mode": 1,

    # Comparison mode 1: automatic mode 2: online mode 3: offline mode
    "match_mode": 1,

    # gpio working status 0: normally closed 1: normally open
    "gpio_state":0,

    # gpio closed hold time 500~5000ms
    "gpio_retention_time":1000,

    # gpio After the relay linkage is turned on, the comparison is successful and all the relays are output
    "gpio_linked":false,

    # wiegand Wiegand output and relay output after Wiegand linkage is turned on
    "wg_linked":false,

    # wiegand Wiegand Protocol wg26: Wiegand 26 Protocol wg34: Wiegand 34 Protocol
    "wg_protocol":"wg26",

    # wiegand Access public Wiegand card number maximum 4294967295
    "wg_id_public":8888,

    # wiegand automatically generate Wiegand card number id starting id maximum 4294967295
    "wg_create_id_begin":1111,

    # wiegand automatically generate Wiegand card number id end id max 4294967295
    "wg_create_id_end":9999999
  },
  # Fill light control
  "led_control":{
      # 1: Normally open 2: Automatic control 3: Normally closed
      "led_mode": 1,

      # led light brightness 1~100
      "led_brightness": 80,

      # led sensitivity low: low mid: medium high: high (only for automatic mode)
      "led_sensitivity": "mid"
   },

  # Screen display control control
  "lcd_control":{
      # Title display (utf8 encoding)
      "lcd_title": "Face Recognition",

      # led display item see enum LcdDisplayItem multiple selection display or operation
      # eg: display title+time+ip 0b 0001 | 0b 0010 | 0b0100 = 0b0111 = 7
      "lcd_display_item": 7,

      # lcd mirror switch
      "led_mirror": true
   },

  # QR code recognition control
  "qr_code_control":{
      # QR code recognition switch
      "enable":true,

      # Repeat scan interval unit: second 0~255
      "repeat_interval": 10
   },

   "name_list":{
        // Automatically clean up expired personnel
        "auto_clean":true
   }
}
```

2.3.1.3 Applicable working mode

Device as server

The device is configured as a client (at this time, the device needs to "ask" to initiate a heartbeat, please add the command_id field when responding; the processing result of the device will be given to you in the next "ask")

2.3.2 Set application parameters

Note: The parameters not specified are optional

2.3.2.1 request

```
{
 # Requester protocol version
 "version": "0.2",

 # Command
 "cmd": "update app params",

 # device info
 "device_info": {
 # device number, length < 32
 "device_no": "0001",

 # position number, length < 32
 "addr_no": "0001",

 # position name, length < 96
 "addr_name": "xx road"
 },

 # platform integration
 "extranet": {
 # global switch
     "enable": true,

     # the following section exists only when “enable” = true
     # working mode: http or tcp (default)
 "mode": "http",

 # server ip
     "server_ip": "192.168.0.23",

     # server domain name, mutually exclusive with server_ip, http mode only
     # server_ip has a higher priority over server_domainis if both exist
     "server_domain": "a.b.com",

 # server port number
```

```
    "server_port": 2001,

    # server url, http mode only
 "url": ""
 },

 # upload configure
 "upload_info": {
 # working mode: http, tcp, ftp, webservice, none
 "method": "http",

 #the following section doesn’t exists when method = “none”
 # server ip
    "server_ip": "192.168.0.172",

    # server domain name, mutually exclusive with server_ip, http mode only
    # server_ip has a higher priority over server_domainis if both exist
    "server_domain": "a.b.com",

 # server port number
 "server_port": 2000,

 # server url, < 102 character, http and webservice only
 "service_url": "face.php",

 # user name, ftp mode only
 "user": "user",

 # password, ftp mode only
 "password": "123",

 # upload destination directory, < 70 characters, ftp only
 # if ends with ‘/’, the file name part will be generated automatically
 # if not ends with ‘/’, the file name part is provided. Be careful with the naming to avoid overwrite.
 # following variables are supported.
 # %ADDR_NO%: position number
 # "%ADDR_NAME%": position name
 # "%DEVICE_NO%": device number
 # "%SEQUENCE%": sequence number
 # "%RAND%": random number
 # "%PID%": person id
 # "%PNAME%": person name
 # "%PROLE%": person category,  0：normal,  1：whitelisted  2：blacklisted
```

```
# "%Y%": capture year
# "%M%": capture month (2 characters)
# "%D%": capture day (2 characters）
# "%H%": capture hour (2 characters)
# "%M%": capture minute (2 characters)
# "%S%": capture seconds (2 character)
# "%MS%": capture milliseconds (3 characters)
# "%INDEX%": 1-based index of the captured image
    "pattern": "/"
},

# face capture control
"face": {
    # minimal face size
    "min_face_size":120,
    # enable name masking or not
    "name_privacy":true,
    # enable face matching or not. true：enable, false：disable
    "enable_match": true,

# the confidence score (70-100), exists if enable_match = true
    "match_score": 70,

    # duplicated face detection filter. true：enable, false：disable
    "enable_dereplication": true,

    # duplication interval in seconds (3-60), exists if enable_match = true
    "derep_timeout ": 3,

    # audio flip interval in minutes
    "audio_flip_interval": 5,

    # image output. “overall”: full image, “closeup”：closeup, “both”: both full and closeup, “none”: none
    "pic_form": "closeup",
    # whether include registration image in output
    "enable_output_reg": false,

# whether include feature data in output
    "enable_output_feature": false,

    # whether enable age detection
"enable_age": false,
    # whether enable gender detection
"enable_sex": false,

# whether enable liveness detection
    "enable_alive": false,

    "body_temperature": {

        # whether enable body temperature detection
        "enable":true

        # threshold temperature for gate access control (gate won’t open if ones temperature is
    higher than this value)
        "limit":38.0,

        # outdoor mode
        "outdoor":true,

        # multi-frame confirmation
        "multi_frame_confir":false,

        # lower bound of body temperature range
        "range_min":36.0,

        # upper bound of body temperature range
        "range_max":42.0,

        # temperature correction value
        "correction_value":0.25,

        # use Fahrenheit degree
        "fahrenheit_unit":false,
        # fast temperature measurement mode
        "fast_mode": false
    },

    "mask":{
        # mask detection enable or not
        "enable":true,

        # no passing without mask
        "must_wear_mask":true,

        # alarm without mask
        "no_mask_alarm":true
    }

},
# gate access control
"gate_control":{

    # output mode. gpio: electric relay, wiegand: Wiegand
    "output_mode": "gpio"

    # gate access control mode
    # 0: n/a
    # 1: by face recognition
    # 2: by id card
    # 3: by face recognition + id card
    # 4: by Wiegand card
    # 5: by face recognition + Wiegand card
    # 6: by face detection (any face detected)
    # 7: by face recognition + whitelisted or id card + whitelisted
    # 8: by face recognition + id card
    # 9: by Wiegand card, meanwhilte take a picture
    # 10: by face recognition or face recognition + id card
    # 11: by id card (the id card number must be in the whitelist)
    # 12: by id card (no matching is performed)
    # 13: by face recognition or id card (no matching is performed)
    # 20: by face recognition or RFID full image
    "gate_mode": 1,

    # working mode. 1: auto, 2: online, 3: offline
    "match_mode": 1,

    # gpio working mode. 0: always closed, 1: always open
    "gpio_state":0,

    # gpio close holding time interval in milliseconds (500-5000)
    "gpio_retention_time":1000,

    # gpio electric relay coordinated action, output on all electric relay if matching succeed
    "gpio_linked":false,

    # wiegand Wiegand coordinated action, output on Wiegand device if matching succeed
    "wg_linked":false,

    # wiegand protocol. wg26: 26-bit Wiegand, wg34: 34-bit Wiegand
```

```
      "wg_protocol":"wg26",

      # public Wiegand card number
      "wg_id_public":8888,

      # starting value of auto generated Wiegand card number, max 4294967295
      "wg_create_id_begin":1111,

      # ending value of auto generated Wiegand card number, max 4294967295
      "wg_create_id_end":9999999
   },
  # led control
  "led_control":{
      # working mode.   1: always on, 2: auto, 3: always off
      "led_mode": 1,

      # led brightness (1-100)
      "led_brightness": 80,

      # led sensitivity. options: “low”, “mid”, “high” (auto mode only)
      "led_sensitivity": "mid"
   },

  # lcd control
  "lcd_control":{
      # title (utf8 encoding)
      "lcd_title": "Face recognition",

      # led display item. See enum LcdDisplayItem, bitwise OR operator is supported
      # e.g., to display title + time + ip, set 0b 0001 | 0b 0010 | 0b0100 = 0b0111 = 7
      "lcd_display_item": 7,

      # whether mirror the lcd display
      "led_mirror": true
   },

  # qr code control
  "qr_code_control":{
      # whether enable qr code
      "enable":true,

      # scan interval in seconds (0-255)
      "repeat_interval": 10
   },
 # remote upgrade
 "remote_upgrader":{
     # whether enable remote upgrade
     "enable":true
   },

   "name_list":{
         // auto-clean expired registration
         "auto_clean":true
   }

}
```

2.3.2.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "update app params",

 # error code
 "code": 0
}
```

2.3.2.3 Applicable working mode

Device as server
Device as configuration client

2.3.3 Reset application configuration

2.3.3.1 Request

```
{
 # protocol version
 "version": "0.2",

 # the command
 "cmd": "reset app params"
}
```

2.3.3.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command the response is for
 "cmd": "reset app params",

 # error code
 "code": 0
}
```

2.3.3.3 Applicable Working Mode

Device as server
Device as configure client

2.3.4 Get Streaming parameter

2.3.4.1 Request

```
{
 # protocol version
 "version": "0.2",

 # the command
 "cmd": "request stream params"
}
```

2.3.4.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "request stream params",

 # error code
 "code": 0,

 # imaging
 "image": {
```

```
 # max exposure (300-9900us)
 "max_exposure": 4000,

 # max gain (0-40DB)
 "max_gain": 20,

 # contrast (0-128)
 "contrast": 60,

 # saturation (1-255)
    "saturation": 110,

 # brightness (1-100)
 "brightness": 46
 },

 # video parameter
 "video": {
 # bitrate (1024-4096kbps)
 "bitrate": 2048
 }
}
```

2.3.4.3 Applicable Working Mode

Device as server
Device as configuration client

2.3.5 Set streaming parameter

2.3.5.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "update stream params",

 # imaging
 "image": {
 # max exposure (300-9900us, optional)
 "max_exposure": 4000,

 # max gain (0-40DB, optional)
 "max_gain": 20,

 # contrast (0-128, optional)
 "contrast": 60,

 # saturation (1-255, optional)
     "saturation": 110,

 # brightness (1-100, optional)
 "brightness": 46
 },

 # video parameter
 "video": {
 # bitrate (1024-4096kbps)
 "bitrate": 2048
 }
}
```

2.3.5.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "update stream params",

 # error code
 "code": 0
}
```

2.3.5.3 Applicable Working Mode

Device as server
Device as configuration client

2.3.6 Reset streaming parameter

2.3.6.1 Request

```
{
 # protocol version
 "version": "0.2",
  # command
  "cmd": "reset stream params"
}
```

2.3.6.2 Response

```
{
  # ack flag
  "reply": "ACK",

  # the command
  "cmd": "reset stream params",

  # error code
  "code": 0
}
```

2.3.6.3 Applicable Working Mode

Device as server
Device as configuration client

2.3.7 Get gate access control rule

2.3.7.1 Overview of the rule

1. There can be up to 15 rules. The rule can be configured to work on every day or on specific weekday.

2.3.7.2 Request

```
{
  # protocol version
  "version": "0.2",

  # command
  "cmd": "request schedule params"
}
```

2.3.7.3 Response

```
{
```

```
# ack flag
"reply": "ACK",

# the command
"cmd": "request schedule params",

# device SN
"device_sn": "012311-35A0DF-78FBEE",

# the rules array. Max 15 rules
"kinds": [
# a single rule
{
# name
"name": "a rule name",

# rule number (1-15)
"kind": 1,

# mode. Available options: “daily”, “weekly”
"mode": "daily",

# configure for each day
"days": [
# configure for a single day
    {
"sections": [
          # time frames
          {
# start time (inclusive)
"start": {
# hour
"hour": 8,
# minute
"minute": 0
},

# end time (exclusive)
"end": {
# hour
"hour": 12,
# minute
"minute": 0
}
```

```
 }
 ]
         }
 ]
 }
 ]
}
```

2.3.8 Set gate access control rule

2.3.8.1 Request

```
{
 # protocol version
 "version": "0.21",

 # command
 "cmd" : "update schedule params",

 # rules array, max 15 rules
 "kinds" : [
 # a single rule
 {
 # rule name
 "name" : "rule name",

 # rule number (1-15)
 "kind" : 1,

 # mode. Available options: “daily”, “weekly”
 "mode" : "daily",

 # configure for each day
 "days" : [
 {
 # configure for a single day
 "sections" : [
 {
 # start time (inclusive)
 "start" : {
 # hour
 "hour" : 8,
 # minute
 "minute" : 10
 },

 # end time (exclusive)
 "end" : {
 # hour
 "hour" : 18,
 # minute
 "minute" : 10
 }
 }
 ]
 }
 ]
 }
 ]
}
```

2.3.8.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "update schedule params",

 # error code (0: success)
 "code": 0,

 # device SN
 "device_sn": "012311-35A0DF-78FBEE",

 # error message, “OK” for success, otherwise the error message
 "desc": "OK"
}
```

2.3.9 Add gate access control rule

Same as 2.3.8 , but add to the end of the rules array, the kinds array size can be 1 to 15.
cmd is “add schedule kind”

2.4 Snapshot

2.4.1.1 Request

```
{
 # protocol
 "version": "0.2",

 # command
 "cmd": "snapshot"
}
```

2.4.1.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "snapshot",

 # error code (0: success)
 "code": 0,

   "image ":
      {
       # the format of image
       "format": "jpg",

       # the image data in base64 format
       "image_data":"…",

       # time
       "time_stamp": "2020/01/01 00:00:00"
      }
}
```

2.4.1.3 Applicable Working Mode

Device as server
Device as configuration client

2.5 Device Control

2.5.1.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "gpio control",

 # gpio port number
 "port": 1,

 # action. “on”: close, “off”: open
 # only “on” (close) is supported, off is done automatically by device
 "ctrl_type": "on",

 # action mode.
   # “test”: test only, no record of gate opening is generated
   # “force”: force output of the signal
   # “scene”: output of signal is controlled by whether the device is enabled
"ctrl_mode": "force",

 # person id. Online mode only, used to generate gate opening record
 "person_id": "0001"
}
```

2.5.1.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": " gpio control",

 # error code (0: success)
 "code": 0
}
```

2.5.1.3 Applicable Working Mode

Device as server

Device as configuration client

2.6 Wiegand device control

2.6.1.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "wiegand control",

 # short Wiegand card number, if you use the long Wiegand card number, set this field to 0
 "wiegand_card_no": 0,

   # long Wiegand card number, for wg36、wg44、wg66
 "long_card_id": 9223372036854775807,

   # action mode.
   # “test”: test only, no record of gate opening is generated
   # “force”: force output of the signal
   # “scene”: output of signal is controlled by whether the device is enabled
 "ctrl_mode": "force",

 # person id. Online mode only, used to generate gate opening record
 "person_id": "0001"
}
```

2.6.1.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": " wiegand control",

 # error code (0: success)
 "code": 0
}
```

2.6.1.3 Applicable Working Mode

Device as server
Device as configuration client

2.7 Heartbeat

2.7.1.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "heart beat",

 # device ip. Can be empty if the initiator of the request is not camera
 "ip": "192.168.1.88",

  # device SN (maximum 32 characters). Can be empty if the initiator of the request is not camera
 "device_sn": "000001-000002-0000BC",

  # device number (maximum 32 characters). Can be empty if the initiator of the request is not camera
 "device_no": "0001",

 # position number (maximum 32 characters). Can be empty if the initiator of the request is not camera
 "addr_no": "0001",

 # position name (maximum 96 character). Can be empty if the initiator of the request is not camera
 "addr_name": "some road name",

   # gps info (gps capable device only)
   "gps":{
      # latitude
       "n":"...",
       # longitude
       "e":"...",
       # number of GPS satellite
       "strength":0
    }
}
```

2.7.1.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "heart beat",

 # error code (0: success)
 "code": 0
}
```

2.7.1.3 Applicable Working Mode

Device as data client (no gps info will be included, since it’s included in the capture record)
Device as configuration client

2.8 User defined authentication code

2.8.1 Read authentication code

2.8.1.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "request customer auth code"
}
```

2.8.1.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "request customer auth code",

 # error code (0: success)
 "code": 0,
 # the authentication code (base64)
 "customer_auth_code": "AAA"
}
```

2.8.1.3 Applicable Working Mode

Device as server
Device as configuration client

2.8.2 Write authentication code (not supported)

2.8.2.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "update customer auth code",

 # the authentication code (base64)
 "customer_auth_code": "AAA"
}
```

2.8.2.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "update customer auth code",

 # error code (0: success)
 "code": 0
}
```

2.8.2.3 Applicable Working Mode

Device as server
Device as configuration client

2.9 System control

2.9.1 Get system time

2.9.1.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "request date time"
}
```

2.9.1.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "request date time",

 # error code (0: success)
 "code": 0,

 # the time
 "date_time": "2018/11/23 16:00:00"
}
```

2.9.1.3 Applicable Working Mode

Device as server
Device as configuration client

2.9.2 System Time

2.9.2.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "update date time",
 # system time
 "date_time": "2018/11/23 16:00:00"
}
```

2.9.2.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "update date time",

 # error code (0: success)
 "code": 0,
}
```

2.9.2.3 Applicable Working Mode

Device as server
Device as configuration client

2.9.3 Reboot Device

2.9.3.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "reboot"
}
```

2.9.3.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "reboot",

 # error code (0: success)
 "code": 0
}
```

2.9.3.3 Applicable Working Mode

Device as server
Device as configuration client

2.9.4 Restart Application

2.9.4.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "restart application"
}
```

2.9.4.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "restart application",

 # error code (0: success)
 "code": 0
}
```

2.9.4.3 Applicable Working Mode

Device as server
Device as configuration client

2.10 Control

2.10.1 Get screensaver configure

2.10.1.1 Request

```
{
 # protocol version
```

```
  "version": "0.2",

  # command
  "cmd": "request lcd screensaver"
}
```

2.10.1.2 Response

```
{
  # ack flag
  "reply": "ACK",

  # the command
  "cmd": "request lcd screensaver",

  # error code (0: success)
  "code": 0,

  # mode
    # “none”: no screensaver
    # “extinguish”: turn off display
    # “media”：play image or video
  "screensaver_mode":"media"

    # description of error message
  "desc": "OK"
}
```

2.10.2 Configure screensaver

2.10.2.1 Request

```
{
  # protocol version
  "version": "0.2",

  # command
  "cmd": "update lcd screensaver",

# mode
    # “none”: no screensaver
    # “extinguish”: turn off display
    # “media”：play image or video
  "screensaver_mode":"media"
}
```

2.10.2.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "update lcd screensaver",

 # error code (0: success)
 "code": 0,

   # error message
 "desc": "OK"
}
```

2.10.3 Get screensaver media configure

2.10.3.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "request lcd media"
}
```

2.10.3.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "request lcd media",

 # error code (0: success)
 "code": 0,

 # screensaver media files (maximum 10)
 "screensaver_media_list":[
   {
    # 1-based index of the file
    "sequence":1,

    # file name
    "media_name":”news1.png”,

    # path to the file，see  2.14.1 Download from device
    "path": "/data_fs/screensaver/NO1.news1.png"
   },
   {
   # 1-based index of the file
    "sequence":2,

    # file name
    "media_name":”news2.png”,

    # path to the file，see  2.14.1 Download from device
    "path": "/data_fs/screensaver/NO2.news2.png"
   }
  ],

  # error message
 "desc": "OK"
}
```

2.10.4 Update media file

2.10.4.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "update lcd media",

 # updated media files (maximum 10)
 "screensaver_media_list":[
   {
    # 1-based index of the file
    "sequence":1,
    # file name
    "media_name":"news1.png"
   }
  ]
}
```

2.10.4.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "update lcd media",

 # error code (0: success)
 "code": 0,

  # updated media files (maximum 10)
 "screensaver_media_list": [
   {
    # 1-based index of the file
    "sequence":1,

    # file name
    "media_name":”news1.png”,

    # path to the file, see   2.14.1 Download from device
    "path": "/data_fs/screensaver/NO1.news1.png"
   }
  ],

  # error message
 "desc": "OK"
}
```

2.10.5 Delete media file

2.10.5.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "delete lcd media",

 # the media files to be deleted (max. 10)
 "screensaver_media_list": [
    {
     # 1-based screensaver media index
     "sequence":1,

     # media name
     "media_name":”news1.png”
    },
    {
     # 1-based screensaver media index
     "sequence":2,

     # media name
     "media_name":”news2.png”
    }
  ]
}
```

2.10.5.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "delete lcd media",

 # error code (0: success)
 "code": 0,

   # deleted media files (max. 10)
 "screensaver_media_list": [
    {
     # 1-based media index
     "sequence":1,

     # media name
     "media_name":”news1.png”
   },
   {
    # 1-based media index
    "sequence":2,

    # media name
    "media_name":”news2.png”
   }
  ],

  # error message
 "desc": "OK"
}
```

2.10.6 Add OSD text

2.10.6.1 Request

```
{
 "version": "0.2",
 # command
 "cmd": "text display",
  # encoding. “utf8” (default) and “gb2312” are supported
 "coding_type":"utf8",
 "text_list":[
 {
    # the position of the text
 "position":{
 "x":0,
 "y":500
 },
    # text display duration in milliseconds
 "alive_time": 1000,
 # font size
 "font_size": 100,
    # font spacing
 "font_spacing": 1,
 # color in ARGB
 "font_color": "0xffffffff",
 # the text content
 "text":"some text 1"
 },
 {
 "position":{
```

```
 "x":0,
 "y":610
 },
 "alive_time": 1000,
 "font_size": 100,
 "font_spacing": 1,
 "font_color": "0xff00ff00",
 "text":"some text 2"
 }
 ]
}
```

2.10.6.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": " text display ",

 # error code (0: success)
 "code": 0,

   # error message
 "desc": "OK"
}
```

2.10.7 Add OSD image

2.10.7.1 Request

```
{
 "version": "0.2",
 # command
 "cmd": "image display",
   # the image
```

```
  "position":{
 "x":0,
 "y":500
 },
 # image display duration in milliseconds. 0: forever
 "alive_time": 1000,
    # image format, only support: png, jpg
    "format": "jpg",
    # image data, base64 encoded
    "image_data":"…"
}
```

2.10.7.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": " image display ",

 # error code (0: success)
 "code": 0,

  # error message
 "desc": "OK"
}
```

2.11 Remote upgrade

2.11.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "ask_upgrade",

  # device SN
 "device_sn": "000001-000002-0000BC",

  # device number (max. 32 characters)
 "device_no": "0001",

   # last upgrade code (used to diagnose last upgrade fail)
 "last_upgrade_code": 0,

 # current firmware version
 "firmware": {
 # system type
     "system": "FaceGate",

     # firmware version
 "appversion": "0.10.2beta",

 # algorithm version
     "algversion": "HV6.0",

 # chip model
     "platform": "HI3516AV200",

 # sensor model
     "sensor": "IMX327D",

 # motherboard model
     "board": "FSAN_02",

     # 镜头电机型号
     "motor": "RSA",

 # lcd model
     "lcd": "LCD_720_576",

 # firmware compile time
     "timestamp": "2019-02-28 18:22:01",

   }
}
```

2.11.1.1 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "ask_upgrade",
 # error code (0: success)
 "code": 0,

 # upgrade type. “none”： no upgrade, “normal”: normal upgrade, “force”: force upgrade (no version check)
"upgrade_type": "none",

 # upgrade file, optional when upgrade_type = “none”
 "upgrade_file": {
 # the path to download upgrade file
     "url": "http://120.88.44.47/upgrade",

     # md5 of the upgrade file
     "md5sum": "621b929a45c388c7e5ac4e260ea3dc99"
  },

  # new firmware info
 "firmware": {
 # system type
     "system": "FaceGate",

     # firmware version
 "appversion": "0.10.2beta",

 # algorithm version
     "algversion": "HV6.0",

 # chip model
     "platform": "HI3516AV200",

 # sensor model
     "sensor": "IMX327D",

 # motherboard model
     "board": "FSAN_02",

     # motor model
     "motor": "RSA",

 # lcd model
     "lcd": "LCD_720_576",

 # time of firmware compilation
```

```
      "timestamp": "2019-02-28 18:22:01"
    }
}
```

2.11.1.2 Applicable Working Mode

Device as client

2.12 Audio

2.12.1 Play TTS

2.12.1.1 Request

```
{
  # protocol version
  "version": "0.2",

  # command
  "cmd": "tts play",

  # the text to play (only Chinese is supported)
  "text":"认证成功"
}
```

2.12.1.2 Response

```
{
  # ack flag
  "reply": "ACK",

  # the command
  "cmd": " tts play ",

  # error code (0: success)
  "code": 0,

    # error message
  "desc": "OK"
}
```

2.12.2 Play audio online

2.12.2.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "play audio stream",

 # the audio to play
 "audio":{
 # format (only wav format is supported)
      "format": "wav",

      # the audio data in base64 format
      "data": "+uqcOrrsSrssetuMmxvM60AAAA"
    }
}
```

2.12.2.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "play audio stream",

 # error code (0: success)
 "code": 0,

   # error message
 "desc": "OK"
}
```

2.12.3 Get built-in audios

2.12.3.1 Request

```
{
 # protocol version
 "version": "0.2",
 # command
 "cmd": "browse audio list"
}
```

2.12.3.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "browse audio list",

 # error code (0: success)
 "code": 0,

 "audio_list": [
        {
        "audio_no": 0,

        "audio_name": "NO0.pass.wav"
        } ,
        …
    ],

   # error message
 "desc": "OK"
}
```

2.12.4 Test playing built-in audio

2.12.4.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "play audio",

 # audio number
 "audio_no": 0,
   # audio name
   "audio_name": "NO0.pass.wav"
}
```

2.12.4.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "play audio",

 # error code (0: success)
 "code": 0,

   # error message
 "desc": "OK"
}
```

2.13 Hardware info

2.13.1 Get hardware info

2.13.1.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "get hardware info"
}
```

2.13.1.2 Response

```
{
 "reply": "ACK",
     # the command
 "cmd": "get hardware info",
 # error code (0: success)
 "code": 0,
     # device SN
     "device_sn": "012353-1FF506-A573EE",
    # motherboard info
"Board": {
"manu": "FSAN",
"model": "FSAN_01"
    },
    # processor info
"Chip": {
"manu": "HISI",
"model": "HI3516AV200"
    },
    # Flash info
"Flash": {
"manu": "",
"model": "",
"size": 0
    },
    # 4G info
    "FourG": {
        "moduler_4g": "NO" # “YES” “NO”
    },
    # LCD info
"Lcd": {
"manu": "",
"model": "LCD_WY_720_576",
"support": [
"LCD_WY_720_576",
"LCD_WY_1280x800",
"LCD_WY_1024x600"
]
    },
    # motor info
"Motor": {
"manu": "",
"model": "",
"type": "RSA"
    },
    # sensor info
"Sensor": {
"manu": "SONY",
"model": "IMX185"
    },
    # wifi info
"WIFI": {
"manu": "",
 "model": ""
 }
}
```

2.13.2 Set hardware info

2.13.2.1 Request

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd": "set hardware info",
   # LCD info
 "Lcd": {
      "manu": "",
 "model": "LCD_WY_720_576"
      },
      # 4G info
      "FourG": {
 "moduler_4g": "NO" # “YES” “NO”
      }
}
```

2.13.2.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "set hardware info",

 # error code (0: success)
 "code": 0
}
```

2.14 File transfer

2.14.1 Download from device

2.14.1.1 Request

```
{
  # protocol version
  "version": "0.2",

  # command
  "cmd": "download file",

  #   path to file
  "path": "/data_fs/screensaver/NO1.news.png",

  # the offset from the start of the file in bytes
  "offset": 0,

  # the length of the download (max. 10240 bytes)
  "size": 10240
}
```

2.14.1.2 Response

```
{
    "reply": "ACK",

    # the command
    "cmd": "download file",

  # error code (0: success)
  "code": 0,

    # path to the file
  "path": "/data_fs/screensaver/NO1.news.png",

  # the total length of the file in bytes
  "total_size": 1024000,

  # current offset (together with “size” field, can be used to test if it’s last packet)
"offset": 0,

  # current download size (together with “offset” field, can be used to test if it’s last packet)
"size": 204800,

  # current data chunk (base64)
  "data": "…",

  # md5sum of current data chunk
  "md5sum": "621b929a45c388c7e5ac4e260ea3dc99",

    # error message
  "desc": "OK"
}
```

2.14.2 Upload to device

2.14.2.1 Request

```
{
  # protocol version
  "version": "0.2",

  # command
  "cmd": "upload file",

    # destination path
  "path": "/data_fs/screensaver/NO1.news.png",

  # total length of the file
  "total_size": 1024000,

  # current offset
"offset": 0,

  # size of current chunk in bytes (max. 200kB)
"size": 204800,

    # current chunk of data（base64）
  "data": "…",

  # md5sum of   current chunk
  "md5sum": "621b929a45c388c7e5ac4e260ea3dc99"
}
```

2.14.2.2 Response

```
{
 # ack flag
 "reply": "ACK",

 # the command
 "cmd": "set hardware info",

 # error code (0: success)
 "code": 0

   # destination path
 "path": "/data_fs/screensaver/NO1.news.png",

 # current offset
 "offset": 0,

 # size of current chunk
 "size": 10240,

   # error message
 "desc": "OK"
}
```

2.15 Data upload encryption

2.15.1 Configure

2.15.1.1 Request

```
{
   "version": "0.21"
 # command
   "cmd": "updata aes",
   # original key in base64 format (max. 16 bytes when converted back to binary data)
 "ori_key": "MTIzNDUAAAAAAAAAAAAAAA==",
   # enable or disable encryption (optional, if omitted, keep current setting)
 "enable": true,
   # the new key in base64 format (max. 16 bytes when converted back to binary data, optional)
"new_key": "MTIzNDU=",
}
```

2.15.1.2 Response

```
{
 # ack flag
 "reply": "ACK",
 # the command
 "cmd": " updata aes ",
 # error code (0: success)
 "code": 0,
   # error message
 "desc": "OK"
}
```

2.15.2 Get configuration

2.15.2.1 Request

```
{
     "version": "0.21"
     # command
     "cmd": "request aes",
}
```

2.15.2.2 Response

```
{
     "version": "0.21",
 # the command
 "cmd": "request aes",
 # error code
     "code":0,
     # error message
     "desc":"OK",
     # whether is enabled
 "enable":true
}
```

2.16 SIP

2.16.1 Set SIP account

2.16.1.1 Request

```
{
    "version": "0.21"

  # command
    "cmd": "sip register",

    # domain name or ip
  "domain": "192.168.0.12",

    # server port number
    "port": 7121,

    # user name
  "username": "0123",

    # password (optional)
  "password": "123456"
}
```

2.16.1.2 Response

```
{
  # ack flag
  "reply": "ACK",
  # the command
  "cmd": "sip register",

  # error code (0: success)
  "code": 0,

    # error message
  "desc": "OK"
}
```

2.16.2 Make sip call

2.16.2.1 Request

```
{
    "version": "0.21",

    # command
    "cmd": "sip call",

    # destination sip account
    # format: sip:username@domain:port
    "url": "sip:someone@192.168.0.188:5060"
}
```

2.16.2.2 Response

```
{
    "version": "0.21",

 # the command
    "cmd": "sip call",

 # error code
    "code":0,

    # error message
    "desc":"OK"
}
```

2.16.3 Hang up a sip call

2.16.3.1 Request

```
{
    "version": "0.21"

    # command
    "cmd": "sip hangup"
}
```

2.16.3.2 Response

```
{
    "version": "0.21",

 # the command
    "cmd": "sip hangup",

 # error code
    "code":0,

    # error message
    "desc":"OK"
}
```

2.16.4 SIP event

No request, only push notification

2.16.4.1 Push notification

```
{
 # protocol version
 "version": "0.2",

 # command
 "cmd", "sip event",


 # device number
 "device_no": "0001",

   # device SN
 "device_sn": "000001-000002-0000BC",

 # position number
 "addr_no": "0001",

 # position name
 "addr_name": "some road",

   # event. 0: call ended, 1: call started, 2: caller ringing, 3: caller rejected, 4: caller connected, 5:called, 6: called connected
 "sip_event": 1,
}
```

2.17 4G info

2.17.1 Get

2.17.1.1 Request

```
{
   "version": "0.21"
  # command
   "cmd": "request 4g infor"
}
```

2.17.1.2 Response

```
{
      # the command
      "cmd": "request 4g infor",

  "reply": "ACK",

      # error code
      "code": 0,

      # device number
      "device_sn": "0123FE-738B3A-70E3EE",

      # 4G module ready
      "module_ready": true,

      # SIM card ready
      "sim_ready": true,

      # SIM ICCID
      "ICCID": "",

      # SIM IMSI
      "IMSI": "",
        # operator
        "operator": "china mobile",

        # 4G RSSI
  "signal_strength": 0,

        # GPS info
  "gps": {
  #longitude
  "e": "",
#latitude
  "n": "",
  "strength": 0
  }
}
```

2.17.2 4G data usage query

2.17.2.1 Request

```
{

"version": "0.21",
  # command
    "cmd": "traffic statistics",
    # month
    "month": 6,
    # day (optional, if omitted, data usage of whole month will be returned)
    # format: start day – end day (optionl)
    # e.g.: "1-9": from day one to day nine
    # “1-”: from day one to current day
    #“1”: day one only
    "day": "1-9",
}
```

2.17.2.2 Response

```
{
        # the command
        "cmd": "traffic statistics",

  "reply": "ACK",

        # error code
```

```
    "code": 0,
    "statistics":[
        {

            # date of data usage.
            "date":"6-1"，#June 1st

            # data usage in bytes
              # bytes received
              "bytes_rx":1024,

              # bytes transmitted
              "bytes_tx":1024
            }，
            {
               # date of data usage, June 2nd
              "date":"6-2"，

              # bytes received
              "bytes_rx":1024,

              # bytes transmitted
              "bytes_tx":1024
            }
    ]

}
```

2.18 Volume

2.18.1 Configure volume

2.18.1.1 Request

```
{
    "version": "0.21",
    // command
    "cmd": "camera volume",
    // action
    "method": "SET",
    // the volume
    "volume": 50
}
```

2.18.1.2 Response

```
{
    "version": "0.21",
    # the command
    "cmd": "camera volume",
    # error code
    "code":0,
    # error message
    "desc":"OK"
}
```

2.18.2 Get volume

2.18.2.1 Request

```
{
    "version": "0.21",
    # command
    "cmd": "camera volume",
    # action
    "method": "GET"
}
```

2.18.2.2 Response

```
{
    "version": "0.21",

    # the command
    "cmd": "camera volume",
    # error code
    "code":0,
    # error message
    "desc":"OK",
    # the volume
    "volume": 50
}
```