

# Dynamic Face Protocol Description (Cloud Server Edition)

## 一、 protocol specification

### 1.1. Content description

This protocol is an interface for data communication between Internet cloud and online face devices.

The agreement contains two kinds of protocols.

One is the UDP protocol to achieve the search device (broadcast), configure the device IP (broadcast), initialization, restart;

Note: The search device, configuration device IP, configuration of enterprise information requires LAN operation, initialize device, restart device, remote door opening, emergency door opening can be LAN operation or cloud operation

The other is implemented by http, such as issuing permissions, upload records and other functions. The device is the version number of each protocol of the client requesting the software server, and the device synchronizes the data through the version number maintained by the software.

### 1.2. operating process:

1. UDP protocol configuration device parameters (IP and port number in the LAN, enterprise information)

[search equipment](#)=====>> Configure the equipment parameters ([Set the server IP and the port number](#), [Configure the enterprise information and the enterprise keys](#))=====>>[device initialize](#) (The main function is to empty the device version number, the device will request from version number 0)

2. http protocol (after the device binds the IP port of the server, it will actively request the version number information, and the software will reply the data to the device according to the protocol to synchronize the data)

The order of equipment request is as follows (subject to the equipment)

[Equipment check-in](#)=====>>[Issue enterprise parameters](#)=====>>[Issue equipment](#)

parameters=====>>Time period of issuance=====>>Issue time group=====>>Issue authority

explain:

1) Introduce the concept of the next version number

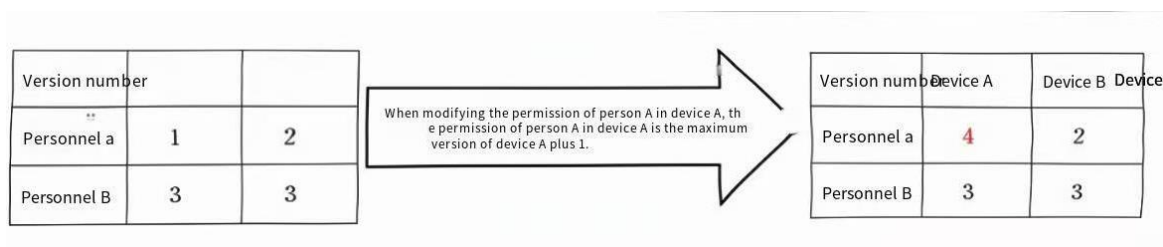
The maximum version number of the enterprise parameters, equipment parameters, time period, time group, permission and other functions will be recorded in the current equipment respectively.

Software will also record the software version number (such as modified the enterprise parameters of heartbeat time field, software enterprise parameter version number will add 1 save), when the heartbeat ("/ MJApi / heartbeat") protocol, the device will send the largest version of the device to the software, the software version number, if the software version number is greater than the version of the device, will request the corresponding function protocol. The synchronization does not stopped until the version number of the requested device is the same as the software version result=2. Sync one protocol until the next instruction.

2) Device restart, initialize the device or device IP these instructions will follow the device signature request from the process, and then synchronize the data and then maintain the version number through the heartbeat.

Example of the synchronous permission version:

**2.1 The following figure shows the data update description of the software permission Table**



**2.2 Synchronize the version process description through the heartbeat**

1) There are two personnel rights a and b in the existing device A. The current permission version number of device A is 3, and the maximum version number of the software end is 3. At this time, the upload heartbeat parameter powerParamVer=3, and the software replies powerParamVer=3, and the device will not request the permission agreement

2) when the software modifier A after the authority of device A, the software

update permission table update version 4 in device A, upload the heartbeat parameter powerParamVer=3, powerParamVer the software reply powerParamVer=4, then the device will request the agreement "get user permission (" / MJApi / user\_power ") " parameter value paramVer=3, the software needs to reply the permission data of version 4 greater than 3 to the device, when the device received data, whether the success according to the following two situations:

If the device again requests the permission paramVer=4, indicating that version 4 is successfully issued, and the software replies result=2 (indicating that the permission has no new data), the heartbeat of the device will no longer request the permission;

If the device request continues to request paramVer=3, it means that the distribution of version 4 fails and check the protocol for errors.

**2.3 The credit card swiping record need not be maintained through the version number, and the device will be uploaded directly.**

If there is a card swiping record, the device will be sent to the server immediately (except when the current device has other processing functions). If the server is not connected during the upload, continue to wait for the service to receive and resume, and then upload the next record until the server receives the reply

**2.4 Instructions such as remote door opening and emergency door opening are implemented through Udp protocol, and the WAN IP port number information of the device can be obtained through the heartbeat packet of UDP sent by the device**

### 1.3. Modify the log

[illegible]

## 二、UDP protocol

### 2.1. Common interface return description:

```
public class ResultInfo<T> {  
    private Boolean result; / Whether the operation is successful, success is true, failure is false  
    private T data; // The service data returned by the interface can be type  
    private String message for numerical value, string or set; // The information  
    returned by the interface is usually the cause information of the error type  
    code  
}
```

For the example of the interface return involved in the document, the return data of individual interfaces will be slightly adjusted, and the true return results shall prevail.

## 2.2. Search Device (UDP Broadcast) (1001)

## parameter declaration

Instruction: 1001	Description: UDP sends broadcast data
<pre>(0xFF 0xFF 0xFF 0xFF 1001 01)</pre>	
required parameter	
<pre>{     "Pass": "", // Udp communication password, 6 digits, factory password is ""     "appCode": 1001 // instruction identification }</pre>	
Response parameters	
<pre>{     "result": true,     "message": "",     "data": {         "deviceKey": "5024668358A3BF1C", //         "deviceType": 70, // Device type, ref         "firmWareVer": "2021082501", // Version number, number, or letter         "doorType": 2, // door type, 1: single door 2: double door 4: four door         "termId": 1, // Device number         "onlineWay": 0, // network connection mode         "enableDHCP": 0, // whether the dynamic IP acquisition is enabled         "localIp": "192.168.1.49", // ip address         "subAdd": "255.255.255.0", // subnet mask         "defaGate": "192.168.1.1", // Gateway         "serverIp": "192.168.1.160", // server IP         "serverPort": 1444 1, // server port         "lableSN": 5000004, // Tab serial number         "deviceModel": "JH 51" // model     } }</pre>	

### 2.3. Configuration of IP (UDP Broadcast) (1002)

#### parameter declaration

Instruction: 1002 Description:UDP sends broadcast data (255.255.255.255,14440))
required parameter
<pre>{ "Pass": "", // UDP communication password, udp protocol requires this parameter   "deviceKey":"FFFFFFFFFFFFFFFF", "appCode": 1002, "newPass": "123456", // Udp communication password, password 6 digits, do not change the old password, here only pass the latest password "enableDHCP": 0, "localIp": "192.168.1.200", "subAdd": "255.255.255.0", "defaGate": "192.168.1.1", "serverIp": "192.168.1.111", // IP of udp service and http server, IP of the device active message   "serverPort": 1444 1, // udp service and http server port, custom   "termId": 1 // Device number }</pre>
Response parameters
<pre>{"result":true,"message":"","data":""}</pre>

## 2.4. Configuring Enterprise Information (UDP Broadcast) (1003)

### parameter declaration

Instruction: 1003    Description: UDP    sends    broadcast    data (255.255.255.255,14440))
required parameter
<pre>{ "pass": "", "deviceKey": "FFFFFFFFFFFFFFFF", "appCode": 1003, "companyId": 1000100001, "signKey": "FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF", // Enterprise check-in key 16 bytes (16-wise string) }</pre>
Response parameters
<pre>{"result":true,"message":"","data":""}</pre>

## 2.5. Server receiving device Udp heartbeat (2001)

### parameter declaration

1. Remote door opening, emergency door opening, initialization, restart, and LAN operation can support cloud operation.

If you need to operate in the cloud, you need to write the UDP server, which will receive 14440 by default. The server IP and port can be set in the [Configure IP (UDP Broadcast) (1002)] protocol

#### 2. Cloud operation process

1) The UDP server receives the heartbeat sent by the device and records the device IP and port

2) When operating the remote door opening, the software sends instructions to the UDP server and forwards them to the device through the saved device address

3) Whether the device replies to the successful message returned by the server

Instructions: 2001    Description: The Device sends the UDP service
required parameter

<pre>{ "head": { "companyId": 1000182636, // Enterprise number "deviceKey": "5024668358A3BF1C", // device serial number "doorType": 2, // door type "termId": 1, // Device number "termType": 70, // Device type "protocolVer": 1, "signString": "530BF8E880D503E1B81DB25699D31D0E" }, "appCode": 2001 }</pre>
Response parameters
no-reply

## 2.6. Initialize Device (1020)

### parameter declaration

Order: 1020 Description: UDP sends device data (device IP, 14440))
required parameter
<pre>{ "pass": "", "deviceKey": "FFFFFFFFFFFFFFFF",   "appCode": 1020,   "initType": 0, // Do not issue this field and settings 0: restore factory Settings 1: empty permissions and records   "reqSeqNo": "21082705204272-1000182636-5024668358A3BF1C-02-1022-8044" // serial number, software customization, equipment reply back }</pre>
Response parameters
<pre>{"result": true, "message": "", "data": "{ \"reqSeqNo\": \"21082705204272-1000182636-5024668358A3BF1C-02-102 0-8044\" }"}</pre>

## 2.7. Restart the device (1021)

### parameter declaration

Instructions: 1021 Description: UDP sends device data (device IP,
required parameter
<pre>{ "pass": "", "deviceKey": "FFFFFFFFFFFFFFFF", "appCode": 1021 ,   "reqSeqNo": "21082705204272-1000182636-5024668358A3BF1C-02-1022-8044" // serial number, software customization, equipment reply back }</pre>
Response parameters
<pre>{"result":true,"message":"","data":{"reqSeqNo\":"21082705204272- 1000182636-5024668358A3BF1C-02-1022-8044\"}}</pre>

## 2.8. Remote door opening (1022)

### parameter declaration

Instructions: 1022 Description: UDP sends device data (device IP,
required parameter
<pre>{ "pass": "", "deviceKey": "5024668358A3BF1C", // Serial number "appCode": 1022, // instruction identification "doorId": 2, // door number, 1:1 door 2:2 door 3:3 door 4:4 door "reqSeqNo": "21082705204272-1000182636-5024668358A3BF1C-02-1022-8044" }</pre>
Response parameters
<pre>{"result":true,"message":"","data":{"reqSeqNo\":"21082705204272- 1000182636-5024668358A3BF1C-02-1022-8044\"}}</pre>

## 2.9. User remote door opening (1023)

### parameter declaration

Instructions: 1022 Description: UDP sends device data (device IP, 14440))
required parameter
<pre>{ "pass": "", "deviceKey": "5024668358A3BF1C", // Serial number "appCode": 102 3, // instruction identification   "doorId": 2, // door number, 1:1 door 2:2 door 3:3 door 4:4 door   "AccNo": 2. Open the door remotely "reqSeqNo": "21082705204272-1000182636-5024668358A3BF1C-02-102 3-8044" }</pre>
Response parameters
<pre>{"result":true,"message":"","data":{"\"reqSeqNo\": \"21082705204272-1000182636-5024668358A3BF1C-02-1022-8044\"}}</pre>

## 2.10. Emergency opening and closing of the door (1024)

### parameter declaration

Order: 1024 Description: UDP sends device data (device IP, 14440))
required parameter
<pre>{ "pass": "", "deviceKey": "5024668358A3BF1C", // Serial number "appCode": 1024, // instruction identification   "doorId": 2, // door number, 1:1 door 2:2 door 3:3 door 4:4 door   "controlWay": 0, // Control type, 0: online 1: open 2: closed "reqSeqNo": "21082705204272-1000182636-5024668358A3BF1C-02-1022-8044" }</pre>
Response parameters
<pre>{"result":true,"message":"","data":{"\"reqSeqNo\": \"21082705204272-1000182636-5024668358A3BF1C-02-1022-8044\"}}</pre>

## 2.11. Configuration of OEM (UDP Broadcast) (1010)

### parameter declaration

Order: 1010 Description: UDP sends device data (255.255.255.255,14440)
required parameter
<pre>{ "pass": "",   "deviceKey": "5024668358A3BF1C", // Serial number "appCode": 1010, // instruction identification "oemPassword": "FFFEFDFCFBFAF9F8F7F6F5F4F3F2F1F0" // OEM password }</pre>
Response parameters
<pre>{"result":true,"message":"","data":""}</pre>

## 2.12. Configuration Equipment Model (UDP Broadcast) (1014)

### parameter declaration

Order: 1014 Description: UDP sends device data (255.255.255.255,14440)
required parameter
<pre>{ "pass": "", "deviceKey": "5024668358A3BF1C", // Serial number "appCode": 10 14, // instruction identification   "device Model": "JH 50" // Set the model such as JH 50 }</pre>
Response parameters
<pre>{"result":true,"message":"","data":""}</pre>

## 2.13. Obtain registration data based on account number(1026)

### 参数说明

Order: 1026 Description: UDP sends device data (device IP, 14440))
required parameter
<pre>{   "pass": "",   "deviceKey": "FFFFFFFFFFFFFFFF",   "appCode": 1026,   "accNo": 0, //Personnel account   "regType": 0, //Type of registration, 0: fingerprint 1: palm vein   "reqSeqNo": "21082705204272-1000182636-5024668358A3BF1C-02-1025-8044"//Flow number, software customization, device reply back</pre>
Response parameters(regType=0)
<pre>{"result": true, "message": "", "data": "{   "reqSeqNo": "21082705204272-1000182636-5024668358A3BF1C-02-1026-8044",   "fpId1": 1, //Whether there's a first fingerprint(0: No 1: Yes)   "fpId2": 1, //Whether there's a second fingerprint(0: No 1: Yes)   "fp1": "", //The first fingerprint is a string of hexadecimal numbers consisting of 800 bytes   "fp2": "", //The second fingerprint is a string of hexadecimal numbers consisting of 800 bytes</pre>
Response parameters(regType=1)
<pre>{"result": true, "message": "", "data": "{   "reqSeqNo": "21082705204272-1000182636-5024668358A3BF1C-02-1026-8044",   "palmId": 1, //Whether there is a palm vein(0: No 1: Yes)   "palmInfo": "", //base64 formatted string }</pre>

### 三、 http Interface Call (Method: POST, utf-8)

#### 1) Http interface specification

The Http interface protocol, the server protocol of the cloud, is designed to receive messages actively sent by the device to the service.

Device request post; Content-Type: application/json; charset=utf-8 .

Software reply to Content-Type: application / json; charset = utf-8; Content-Length:

##### Example of the device sending a request message

```
POST /MJApi/sign HTTP/1.0
```

```
Host: 192.168.1.165
```

```
User-Agent:Mozilla/5.0 (lwip;9250) qu
```

```
Content-Length: 299
```

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

```
{ "head": { "companyId": "1000121628", "deviceKey": "DE618492975F7B39", "doorType": 4, "termId": 1, "termType": 70, "protocolVer": 1, "signString": "3B7BABC2F50070014FAEAD06719F3A3"}, "firmWareVer": "2024031501", "oem": "00000000000000000000000000000000", "lableSN": "5402855", "deviceModel": "5040" }
```

##### Software reply:

```
HTTP/1.1 200 Http Server OK
```

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

```
Content-Length: 121
```

```
Connection: close
```

```
{ "result": 1, "message": "success", "data": { "serverTime": "2024-04-23 17:06:02", "mainKey": "4534CC4615BB4A8A719215BC381CE2B5" } }
```

**Note: First of all, the version number of the software maintenance equipment information, the content of a function of the same device is added, modify and delete operations need to increase the version number, After receiving the heartbeat command, the latest local version number to the device (the specific protocol), the device will request the corresponding data according to the version number, the software received the device request the version number is the largest version number of the current device, The version number that the software replies to the device is data in the local database with one version number larger than the requested version number**

illustrate:

The process of obtaining the version number through the heartbeat is divided into two situations. Take the permission as an example, the current version number of the device is 1, the current request version number of the device in the heartbeat is 1,

First case: If the service replies that the version number of the permission in the heartbeat is 1, the device will not request the permission.

The second case: if the service reply permission version number in the heartbeat is 11, greater than the device request (hardware request version number is the largest version number of the current device), the service to sort from small to large, get more than 1 minimum version number data, such as 3, then reply to the device version

number is 3 permission, the device will request version number 3, so loop until the device request 11, software reply {"result": 2, "data": "", "message": ""}, hardware will not request permissions

## 2) Protocol Common Parameters

Instruction header: The header parameter of the server is consistent
Device sends a service request parameter
<pre>{ "head": { "companyId": 1000182636, // Enterprise number "deviceKey": "DF60A894035E4832", // device serial number "doorType": 2, // Number: 1: single door; 2: double door; 4: four doors "termId": 1, // Device number "termType": 50, // Device type "protocolVer": 1, "signString": "86186F3AA8EE9D22BE1BC661F3A28C36" // Enterprise key }, "firmWareVer": 2021062201, //   "oem": "FFFEFDFCFBFAF9F8F7F6F5F4F3F2F1F0" }</pre> <p>Note: signString: 16 decimal number of 16 bytes, the check-in key when the equipment check-in agreement, the others are communication keys, Check-in key: access through the Udp protocol "1.4. Configuration Enterprise Information (UDP Broadcast) (1003)".</p>
Service response parameters
<p>Response to general</p> <pre>public class ResultInfo&lt;T&gt; {     private int result; // Whether the operation is successful, success is 1, failure is 0, no new data update is 2,3 verification failure to return to the device check-in     private T data; // The service data returned by the interface may be type     private String message such as numerical value, string or set; // The information returned by the interface is usually the cause information of error type code, character or number, and may not reply to Chinese }</pre> <p>For example: successful check-in reply</p>

Example of interface return involved in the document. The return data of individual interfaces will be slightly adjusted, and the true return results shall prevail.

### 3.1. Device check-in ("/ MJApi / sign")

Note: After the device service IP is configured, the device will actively send the "device check-in" request to the server

#### Parameter details

Method	URL:			
register	/MJApi/sign			
equipment requirement				
param	descri	type	Mus	Additio
head	Directive head	Json	Y	Refer to Common Parameters
firmWareVer	Firmware version	string	Y	
oem	Oem password	string	Y	Field value is the original data for the Udp protocol "configuration OEM (UDP Broadcast) (1010)" configuration
lableSN	Label serial number	Int	Y	
deviceModel	model	string	Y	
reply				
param	descri	type	Mus	Additio
serverTime	Server time	Json	Y	Format: (yyyy-MM-dd HH: mm: ss)
mainKey	Enterprise communication password	Int	Y	After checking in, the device sends the command key (16 decimal splicing string)

#### instance

ask:

```
{"head":{"companyId":1000121628,"deviceKey":"DE6958470F874035","doorType":2,"termId":1,"termType":70,"protocolVer":1,"signString":"7BE74253BA1603498CDBE15F05CAEA2F"},"firmWareVer":"2022080101","oem":"00000000000000000000000000000000","lableSN":5200000,"deviceModel":" JH01"}
```

Software reply:

```
{"result": 1,"message": "", "data": {"serverTime":"2021-06-24 15:57:43","mainKey":"FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"}}
```

### 3.2. Get enterprise parameters ("/ MJApi / comp \_ param")

Note: "Device check-in" protocol, the server returns, the device will automatically request "enterprise parameters"

#### Parameter details

Method	URL:			
Enterprise	/MJApi/comp_param			
equipment requirement				
param	descri	type	Mus	Additio
head	Directive head	Json	Y	Refer to Common Parameters
paramVer	version number	Int	Y	Maximum version number of the current
reply				
param	descri	type	Mus	Additio
paramVer	Version number of the	Json	Y	
heartInterval	Heartbeat time	Int	Y	second

#### instance

```
equipment requirement:
{"head":{"companyId":1000182636,"deviceKey":"DF60A894035E4832","doorType":2,"termId":1,"termType":50,"protocolVer":1,"signString":"86186F3AA8EE9D22BE1BC661F3A28C36"},"paramVer":0}

Software reply:
{"result": 1,"message": "", "data": {"paramVer":1,"heartInterval":60}}
```

### 3.3. Get the time period ("/ MJApi / time\_part")

explain:

1. "heartbeat" protocol, the version number of the device is smaller than the version number of the local database, the device will automatically request the protocol

2. A maximum transmission of 2 time periods, when the version number update is completed, send "result": 2 result code. In addition, it should be noted that the data of the version number must be arranged in order, which is the last data version, which must be the largest in this package

3. Release time period

Example: the current hardware version number 0, the software version number 6

It needs to be sent three times

The first time: hardware send paramVer =0, software back to 1 to 5 version number data

Second time: hardware send paramVer =5; data of version number of software return 6

Third: hardware post paramVer =6; software back {"result": 2, "message": "no new data", "data": null}

4. The following protocol is roughly the same as the time period process

### Parameter details

Method		URL:			
time		/MJApi/time_part			
equipment requirement					
param		descri	type	Mus	Additio
head		Directive head	Json	Y	Refer to Common Parameters
paramVer		version number	Int	Y	Maximum version number of the current
reply					
param		descri	type	Mus	Additio
			Json	Y	Up to 2 groups
partId		Heartbeat time	Int	Y	Up to 200 time periods
paramVer		version number			
opType		Is enabled			Enabled, 0: No,1 enabled
partSection					Each group has 6 time periods
	partBegin	start time	Int		If 8:00,800, not enabled start and end are 0
	partEnd	terminal time	Int		1800 for 18:00
	partVWay	verification mode	Int		<u>Time-period validation</u>

## instance

equipment requirement:

```
{"head":{"companyId":1000182636,"deviceKey":"DF60A894035E4832","doorType":2,"termId":1,"termType":70,"protocolVer":1,"signString":"F9901E9D99E4F75EA1DD819085265AB4"},"paramVer":0}
```

Software reply:

```
{"result":1,"message":"","data":[{"partId":1,"paramVer":1,"opType":1,"outControl":0,"partSection":[{"partBegin":0,"partEnd":2359,"partVWay":10,"inControl":0},{partBegin":0,"partEnd":0,"partVWay":0,"inControl":0},{partBegin":0,"partEnd":0,"partVWay":0,"inControl":0},{partBegin":0,"partEnd":0,"partVWay":0,"inControl":0},{partBegin":0,"partEnd":0,"partVWay":0,"inControl":0},{partBegin":0,"partEnd":0,"partVWay":0,"inControl":0}]]}}
```

### 3.4. Get the time group ("/ MJApi / time\_group")

explain:

#### Parameter details

Method		URL:			
Time group		/MJApi/time_group			
equipment requirement					
param		descri	type	Mus	Additio
head		Directive head	Json	Y	Refer to Common Parameters
paramVer		version number	Int	Y	Maximum version number of the current
reply					
param		descri	type	Mus	Additio
			Json	Y	Up to 2 groups
groupId		Time group number	Int	Y	Max. 200 custom time groups 1 to 200, Built-in equipment 0: Traffic allowed 255: No traffic
paramVer		version number	Int		
opType		Is enabled	Int		Enabled, 0: No,1 enabled
tWeekSun		weekday	Int		time quantum ID
tWeekMon		Monday	Int		time quantum ID
tWeekTue		Tuesday	Int		time quantum ID
tWeekWed		Wednesday	Int		time quantum ID
tWeekThu		Thursday	Int		time quantum ID
tWeekFri		Fri	Int		time quantum ID
tWeekSat		Sat	Int		time quantum ID

tlsAppHoliday	Whether the time group is higher than the holidays	Int		1: Time group is higher than holidays, and time group is valid 0: Effective during holidays

## instance

equipment requirement:

```
{
  "head": {
    "companyId": 1000182636,
    "deviceKey": "DF60A894035E4832",
    "doorType": 2,
    "termId": 1,
    "termType": 70,
    "protocolVer": 1,
    "signString": "F9901E9D99E4F75EA1DD819085265AB4",
    "paramVer": 0
  }
}
```

Software reply:

```
{
  "result": 1,
  "message": "",
  "data": [
    {
      "groupId": 1,
      "paramVer": 1,
      "opType": 1,
      "tWeekSun": 1,
      "tWeekMon": 1,
      "tWeekTue": 1,
      "tWeekWed": 1,
      "tWeekThu": 1,
      "tWeekFri": 1,
      "tWeekSat": 1,
      "tIsAppHoliday": 0
    },
    {
      "groupId": 2,
      "paramVer": 2,
      "opType": 1,
      "tWeekSun": 0,
      "tWeekMon": 0,
      "tWeekTue": 0,
      "tWeekWed": 0,
      "tWeekThu": 0,
      "tWeekFri": 0,
      "tWeekSat": 0,
      "tIsAppHoliday": 0
    },
    {
      "groupId": 3,
      "paramVer": 3,
      "opType": 1,
      "tWeekSun": 0,
      "tWeekMon": 0,
      "tWeekTue": 0,
      "tWeekWed": 0,
      "tWeekThu": 0,
      "tWeekFri": 0,
      "tWeekSat": 0,
      "tIsAppHoliday": 0
    },
    {
      "groupId": 4,
      "paramVer": 4,
      "opType": 1,
      "tWeekSun": 0,
      "tWeekMon": 0,
      "tWeekTue": 0,
      "tWeekWed": 0,
      "tWeekThu": 0,
      "tWeekFri": 0,
      "tWeekSat": 0,
      "tIsAppHoliday": 0
    },
    {
      "groupId": 5,
      "paramVer": 5,
      "opType": 1,
      "tWeekSun": 0,
      "tWeekMon": 0,
      "tWeekTue": 0,
      "tWeekWed": 0,
      "tWeekThu": 0,
      "tWeekFri": 0,
      "tWeekSat": 0,
      "tIsAppHoliday": 0
    }
  ]
}
```

## 3.5. Get the device parameters ("/ MJApi / dev\_param")

explain:

### Parameter details

Method	URL:			
	/MJApi/dev_param			
equipment requirement				
paramet	descri	type	Mus	Additional
head	Directive head	Json	Y	Refer to Common Parameters
paramVer	version number	Int	Y	Maximum version number of the current device
reply				
paramet	descri	type	Mus	Additional
paramVer	version number	Int	Y	

wgOutputModel	WG input/output mode	Int	Y	0 "WG26",1: "WG34" Default: 1
wgOutputEnable	Do you want to enable output WG	Int	Y	0: No, 1: Yes Default: 1
wgOutputFormat	WG output format	Int	Y	0: Card number, 1: Employee number Default: 1

faceThreshold		recognition threshold	Int	Y	Default 50 Integer with valid values from 10 to 60 (subject to device adjustment)
lightEnable		Do you want to turn on the fill light	Int	Y	0 off 1 automatic fill light (detected face lights on, no face lights off after one minute), 2 time periods, default: 1
lightTime		Supplemental lighting period	Int	Y	Up to 3 paragraphs
	beginTime	Time slot start time	Int	Y	8: 00 is represented by 800
	endTime	End time of time period	Int	Y	18: 00 is represented by 1800
doorParam		Door parameter settings	Json	Y	
	appKeyButton	Enable door opening button	Int	Y	0: Not enabled, 1: Enable Default: 1
	closeDelay	Door opening delay (seconds)	Int	Y	Unit seconds, (0~255), default 3 seconds
	doorWorkWay	Door working mode	Int	Y	0 Normal 1 Timer, 3 Multi Card Factory Default 0
	timeGroup	Timed door opening time group	Int	Y	After setting, the time group is normally open, and outside the segment, it is online, Enable: doorWorkWay=1,timeGroup>0 Cancel: doorWorkWay=0
displayStandbyTime		Off screen time	int	Y	Set the screen hold time, 0 does not sleep (unit: seconds), (0-600s)
Language		language	int	Y	1: English 2:Persian
faceRepeatEnable		Continuous recognition enable switch	Int	N	0: Not enabled, 1: Enable default: 0
isCheckMask		Enable masks	int	N	0: Not enabled, 1: Enable default: 0
faceLiving		Live threshold	Int	N	0: Do not enable (1~80)
faceDistance		Recognition distance	Int	N	0: Far 1: Medium 2: Near default 0 Near: 0.3-0.5m Medium: 0.3-1m Distance: 0.3-2.5m
Volume		sound	Int	N	0 - 100
repeatOpenTime		Repeat confirmation time	int	N	0-60 minutes default 1
ioModeType		Attendance status type	int	N	0: Do not enable 1: Manual 2: Automatic
ioModeParam		Attendance Status Collection	json	N	There are a total of 10 states, and 10 will be issued in order at once
	ioModeName	Attendance status name	string	N	
	ioModeTime	Opening time	int	N	8: 00 is represented by 800

dstEnable	Enable daylight saving time	string	Y	0: Not enabled, 1: Enable default: 0
dstStart	The start time of daylight saving time	string	N	Date format "MM/dd HH:mm"
dstEnd	The end time of daylight saving time	string	N	Date format "MM/dd HH:mm"

## instance

```
equipment requirement:
{"head":{"companyId":1000182636,"deviceKey":"DF60A894035E4832","doorType":2,"termId":1,"termType":70,"protocolVer":1,"signString":"F9901E9D99E4F75EA1DD819085265AB4"},"paramVer":0}
Software reply:
{"result":1,"message":"success","data":{"isCheckedMask":0,"wgOutputEnable":1,"wgOutputModel":1,"wgOutputFormat":0,"repeatOpenTime":1,"faceRepeatEnable":1,"faceThreshold":50,"faceLiving":10,"faceDistance":0,"Volume":8,"Language":1,"displayStandbyTime":60,"lightEnable":1,"lightTime":[{"beginTime":1730,"endTime":730}], "dstEnable":1,"dstStart":"05/22 17:35","dstEnd":"05/22 18:01","doorParam":[{"doorWorkWay":0,"appKeyButton":1,"timeGroup":0,"closeDelay":5}], "ioModeType":0,"ioModeParam":[{"ioModeName":"On Duty","ioModeTime":0}, {"ioModeName":"Off Duty","ioModeTime":0}, {"ioModeName":"Lunch In","ioModeTime":0}, {"ioModeName":"Lunch Out","ioModeTime":0}, {"ioModeName":"OT In","ioModeTime":0}, {"ioModeName":"OT Out","ioModeTime":0}, {"ioModeName":"Tea In","ioModeTime":0}, {"ioModeName":"Tea Out","ioModeTime":0}, {"ioModeName":"Pray In","ioModeTime":0}, {"ioModeName":"Pray Out","ioModeTime":0}]
}
```

### 3.6. Gets user permission ("/ MJApi / user\_power")

Description: Issue 1 version number at a time

## Parameter details

Method	URL:			
	/MJApi/user_power			
equipment requirement				
param	descri	type	Mus	Additio
head	Directive head	Json	Y	Refer to Common Parameters
paramVer	version number	Int	Y	Maximum version number of the current
reply				
param	descri	type	Mus	Additio
paramVer	version number	Int	Y	

accNo	Personnel account	Int	Y	
name	Name of personnel	String	Y	Supports up to 32 bytes
opType	operation type	Int	Y	1: Upload, 0 to delete
cardSN	card number	String	Y	0: No card number
password	password	String	Y	Six digits
beginExpDate	Start Date	String	Y	For example: 2099-12-31 00:00:00
endExpDate	cut-off time	String	Y	For example: 2099-12-31 00:00:00
timeGroup	Time group	Int	Y	Time group number
faceId1	Is there a photo 1	Int	Y	No: -1, yes: 0
faceImage1	Photo 1	Base64	Y	Unlabeled Base64, no null,
isManager	Is it an administrator	Int	Y	0: Ordinary user 1: administrator

fpId1	Is there a first fingerprint	Int	Y	None: 0, Yes: 1
fpId2	Is there a second fingerprint	Int	Y	None: 0, Yes: 1
fp1	The first fingerprint data	String	N	
fp2	Second fingerprint data	String	N	
palmId	Is there a palm vein	Int	Y	None: 0, Yes: 1
palmInfo	palmar vein data	Base64	N	

--	--	--	--	--

## instance

```
equipment requirement:
{"head":{"companyId":1000182636,"deviceKey":"DF60A894035E4832","doorType":2,"termId":1,"termType":70,"protocolVer":1,"signString":"F9901E9D99E4F75EA1DD819085265AB4"},"paramVer":0}

Software reply:
{"result":1, "message": "success", "data":{"paramVer":1, "accNo":10, "name": "zero", "opType":1, "doorId":1, "cardSN": "4714111", "password": "112233", "beginExpDate":null, "endExpDate": "2099-01-01 00:00:00", "timeGroup":0, "faceId1":0, "faceImage1":"/9j/. Omit the... H // 2 Q =="}}}
```

### 3.7. Upload the record ("/MJApi/up\_log")

Note: havLog=0 upload door magnetic record, havLog=1 upload record, the device uploaded record to return a successful prompt, otherwise the device will be sent

## Parameter details

Method	URL:
	/MJApi/up_log
equipment requirement	

param	descri	type	Mus	Additio
head	Directive head	Json	Y	Refer to Common Parameters
havLog	Is it a record	Int	Y	1: Record, 0: Submit records when the door
reqSeqNo	Record serial number	String	N	
logType	record type	Int	N	(0: Normal record, 1: record illegal user
accNo	Personnel account	Int	N	
passTime	Credit card time	String	N	Such as 2021-08-30 14:42:46
operation	Way of passage	Int	N	consult <a href="#">Way of passage</a>
alarmCode	Alarm code	Int	N	consult <a href="#">Alarm code</a>
passStatus	Pass state	Int	N	0: Pass 1: No pass
doorId	Door number	Int	N	Face default 1
readId	<a href="#">Attendance status</a>	Int	N	<a href="#">Attendance status (1~10)</a>
temperature	thermometric	Int	N	Temperature measuring temperature, 4 digits, display the temperature needs to be divided by 100, no temperature measuring
lockStatus	Door magnetic state		Y	Arrays in door order, 1 door, 2 doors...
	doorId	Door number	int	
	status	Door magnetic state	int	Door magnetic state, 0: off (closed) 1: on
reply				
param	descri	type	Mus	Additio

## instance

equipment requirement:

1. When recorded

```
{
  "head": {
    "companyId": "1000121628",
    "deviceKey": "FFFF864298358F6E",
    "doorType": 1,
    "termId": 0,
    "termType": 33,
    "protocolVer": 2,
    "signString": "C29C0C245D7B70E2EE6DF633A92DA0A2"
  },
  "havLog": 1,
  "reqSeqNo": "166763604851012604",
  "logType": 0,
  "accNo": 10,
  "passTime": "2022-11-05 16:14:08",
  "operation": 2,
  "alarmCode": 0,
  "passStatus": 0,
  "doorId": 1,
  "readId": 1,
  "temperature": 0,
  "lockStatus": [
    {
      "doorId": 1,
      "status": 0
    }
  ]
}
```

2. Upload the door magnetic records

```
{
  "head": {
    "companyId": "1000182636",
    "deviceKey": "5024668358A3BF1C",
    "doorType": 2,
    "termId": 1,
    "termType": 70,
    "protocolVer": 1,
    "signString": "530BF8E880D503E1B81DB25699D31D0E"
  },
  "havLog": 0,
  "lockStatus": [
    {
      "doorId": 1,
      "status": 1
    },
    {
      "doorId": 2,
      "status": 0
    }
  ]
}
```

Software reply:

```
{
  "result": 1,
  "message": "",
  "data": ""
}
```

### 3.8. Heartbeat (" / MJApi / heartbeat ")

Description: Send it once a minute. The maximum version number of the cloud service obtained by the device, such as the version number greater than the device, will actively request to update the device

#### Parameter details

Method	URL:			
	/MJApi/heartbeat			
equipment requirement				
param	descri	type	Mus	Additio
head	Directive head	Json	Y	Refer to Common Parameters
compParamVer	Enterprise version	Int	Y	The device currently has the maximum
devConfigVer	Device version	Int	Y	The device currently has the maximum
timePartVer	Time period version	Int	Y	The device currently has the maximum
timeGroupVer	Time group version	Int	Y	The device currently has the maximum
powerParamVer	Permission version	Int	Y	The device currently has the maximum
reply				
param	descri	type	Mus	Additio
compParamVer	Enterprise version	Int	Y	Cloud current largest version number
devConfigVer	Device version	Int	Y	Cloud current largest version number
timePartVer	Time period version	Int	Y	Cloud current largest version number
timeGroupVer	Time group version	Int	Y	Cloud current largest version number
powerParamVer	Permission version	Int	Y	Cloud current largest version number

#### instance

```
equipment requirement:
{"head":{"companyId":1000121628,"deviceKey":"FFFF864298358F6E","doorType":1,"termId":0,"termType":33,"protocolVer":2,"signString":"C29C0C245D7B70E2EE6DF633A92DA0A2"},"compParamVer":7,"devConfigVer":1,"timePartVer":15,"timeGroupVer":13,"powerParamVer":0}
Software reply:
{"result":1,"message":"success","data":{"compParamVer":7,"devConfigVer":1,"timePartVer":15,"timeGroupVer":13,"holidayVer":0,"powerParamVer":0}}
```

### 3.9. Obtain scheduled door opening 2("/MJApi/time\_timing")

Note: Valid when doorWorkWay=0 in the device parameters ("/MJApi/dev\_param")

## 参数详情

Method		URL:			
		/MJApi/time_timing			
equipment requirement					
param		descrip	type	Mus	Additio
head		Directive head	Json	Y	Refer to Common Parameters
reply					
param		descrip	type	Mus	Additio
timing		An array of time slots within a week		Y	The array represents three time slots for each day, from Monday to Sunday
	beginDate1	Start time of time slot 1	Int	Y	For example, 8:00 is represented by 800
	endDate1	End time of time slot 1	Int	Y	For example, 14:00 is represented by 1400
	timingModel1	Door control method	Int	Y	0: Online 1: Normally open 2: Normally
	beginDate2	Start time of time slot 2	Int	Y	For example, 8:00 is represented by 800
	endDate2	End time of time slot 2	String	Y	For example, 14:00 is represented by 1400
	timingMode2	Door control method	Int	Y	0: Online 1: Normally open 2: Normally
	beginDate3	Start time of time slot 2	Int	Y	For example, 8:00 is represented by 800
	endDate3	End time of time slot 2	String	Y	For example, 14:00 is represented by 1400
	timingMode3	Door control method	Int	Y	0: Online 1: Normally open 2: Normally

## 示例

设备请求:

```
{
  "head": {
    "companyId": 1000121628,
    "deviceKey": "FFFF8ED7D2590054",
    "doorType": 1,
    "termId": 0,
    "termType": 35,
    "protocolVer": 2,
    "signString": "14ba8c590a93fbe9fa67280e91fc1245"
  }
}
```

软件回复：

[illegible]

## 四、 type definition

### 4.1. Validation Type-Time-period validation type

Value (decimal)	meaning
15	Face+ fingerprint
16	password + Face/FP/palm
17	Card + Face/FP/palm
2 0	All certified or

### 4.2. Record alarm code

Value (decimal)	meaning
0	normal
80	Illegal card
81	The door timeout is not closed
83	fire alarm
86	Illegal opening and closing
90	Has expired
91	stranger
95	Alarm
13	wrong password

### 4.3. Record the mode of passage

Value (decimal)	meaning
0	Card
1	fingerprint
2	Face
3	Remote user unlocks the door (mobile terminal)
4	Temporary password to open the door
6	Personal password

9	QR code to open the door
10	Card+personal password
11	fingerprint + password
12	Face+personal password
13	fingerprint + card
14	Face+card
15	face + fingerprint
21	palm vein
22	palm + card
23	palm + FP
24	palm + Face
25	palm + password
43	Not open during the designated time period
46	Holiday period
100	Remote door opening
101	Button to open the door
102	Timed door opening
104	EMERGENCY OPEN
105	Emergency door closing
108	Restore Online
200	Multi card door opening (card)
201	Multi card door opening (fingerprint)
202	Multi card door opening (facial)
206	Multi card door opening (personal password)
209	Multi card door opening (QR code door opening)
210	Multi card door opening (card+password)
211	Multi card door opening (fingerprint+password)
212	Multi card door opening (face+password)
213	Multi card door opening (fingerprint+card)
214	Multi card door opening

	(face+card)
215	Multi card door opening (face+fingerprint)
221	Multi card door opening (palm)
222	Multi card door opening (palm+card)
223	Multi card door opening (palm+FP)
224	Multi card door opening (palm+face)
225	Multi card door opening (palm+password)

#### 4. 4. device type

Value (decimal)	meaning
33	Face-cloud version
35	Face-cloud version