



Dynamic Face Protocol Description (Cloud Server Edition)

1. 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

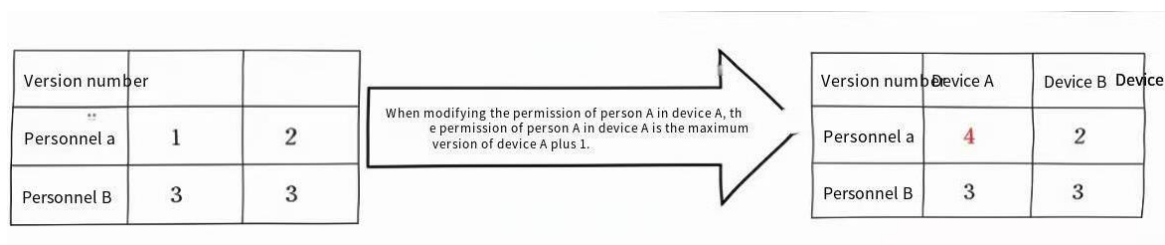
The maximum version number of the enterprise parameters, equipment parameters, time period, time group, permission and other functions will be recorded in the current equipment respectively.

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 with permissions a and b in the existing device A. The current permission version number of device A is 3, and the maximum version number on the software side is 3. At this time, the heartbeat parameter powerParamVer=3 is uploaded, and the software replies with powerParamVer=3. The device will not request



permission agreements

2) When software modifier A gains permission on device A, the software update permission table personnel update the version number to 4 on device A. At this time, the heartbeat parameter powerParamVer=3 is uploaded, and the software replies with powerParamVer=4. At this time, the device will request the protocol to "obtain user permissions ("/MJApi/user_power ") " parameter value paramVer=3. The software needs to reply with permission data for version 4 with a version number greater than 3 to the device. When the device receives the data, whether it is successful is judged according to the following two situations:

If the device fails to save data with paramVer=4, it will request the [Result Feedback Interface ("/MJApi/codeResult")] to notify the software of the failure

Regardless of whether the save fails or succeeds, the device will request paramVer=4, and the software will reply with data greater than paramVer=4,

Until the software replies with result=2 (indicating no new data for permissions), the device will no longer request permissions in this round of heartbeat;

Note: If it is a non authorized interface and the device fails to save, it will repeatedly request the previous package software to reply with the version number. Check if there are any errors in the protocol content (the software can compare the version number according to requirements to determine whether to save the failed version information and skip this version to update the next version)



("/MJApi/user_power")

Example 1: Device version 3, software version 1,3
—no updates available

	Device Request	Software reply
1	paramVer=3	result=2

Received result=2.
The device is no longer requesting permission in this round of heartbeat.
The next heartbeat of the device will be powerParamVer=1

Example of Personnel
Permission
Synchronization

("/MJApi/user_power")

Example 2: Device version 3, software versions
available are 1, 3, 4, 5—updates required for 4 and 5

	Device Request	Software reply
1	paramVer=3	result=1,paramVer=4
2	paramVer=4	result=1,paramVer=5
3	paramVer=5	result=2

the next heartbeat of the device will be powerParamVer=5

("/MJApi/user_power")

Example 3: The device version number is 3, and the
software version numbers are 1, 3, 4, and 5, with
version 4 containing data errors

	Device Request	Software reply
1	paramVer=3	result=1,paramVer=4
3	paramVer=4	result=1,paramVer=5
4	paramVer=5	result=2

the next heartbeat of the device will be powerParamVer=5

("/MJApi/codeResult")

	Device Request	Software reply
2	accNo=Personnel Number, code>Error encoding	result=1

("/MJApi/comp_param")

Example 1: Device version number 1,
software version numbers 1, 3, 4, 5

	Device Request	Software reply
1	paramVer=1	result=1,paramVer=3
2	paramVer=3	result=1,paramVer=4

The device received the second packet
paramVer=4
and failed to save the data

3	paramVer=3	result=1,paramVer=4
4	paramVer=3	result=1,paramVer=4
n	paramVer=3	result=1,paramVer=4

the next heartbeat of the device will be powerParamVer=3

Example explanation of
synchronization for non personnel
permission agreements

he platform keeps replying with incorrect data, and the
device will repeatedly request the previous version
number in the next request,
If you want to skip this version and continue to the
next version, you can reply with paramVer=5 for the
correct data, or reply with result=2 to temporarily stop
the synchronization of this interface in this round

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

date	revise content	Firmware version number
01/08/2026		2025112500

2. 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 (255.255.255.255,14440))	
required parameter	
{ "Pass": "", // Udp communication password, 6 digits, factory password is "" "appCode": 1001 // instruction identification }	
Response parameters	



```
{
"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
    }
}
```

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

parameter declaration

Instruction: 1002 Description:UDP sends broadcast data (255.255.255.255,14440))
required parameter



```
{
  "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
}
```

Response parameters

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

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

parameter declaration

Instruction: 1003 Description: UDP sends broadcast data (255.255.255.255,14440))
required parameter



```
{  
  "pass": "",  
  "deviceKey": "FFFFFFFFFFFFFFFF",  
  "appCode": 1003,  
  "companyId": 1000100001,  
  "signKey": "FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF", // Enterprise check-in key 16  
  bytes (16-wise string)  
}
```

Response parameters

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

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



```
{
"head": {
"companyId": 1000100000, // 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
}
```

Response parameters

no-reply

2.6. Initialize Device (1020)

parameter declaration

Order: 1020 Description: UDP sends device data (device IP, 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": "21082705204272-1000100000-5024668358A3BF1C-02-1022-8044" // serial number, software customization, equipment reply back }</pre>
Response parameters
<pre>{"result":true,"message":"","data":{"reqSeqNo":"21082705204272-1000100000-5024668358A3BF1C-02-102 0-8044"}}</pre>



2.7. Restart the device (1021)

parameter declaration

Instructions: 1021 Description: UDP sends device data
required parameter
<pre>{ "pass": "", "deviceKey": "FFFFFFFFFFFFFFFF", "appCode": 1021 , "reqSeqNo": "21082705204272-1000100000-5024668358A3BF1C-02-1022-8044" // serial number, software customization, equipment reply back }</pre>
Response parameters
<pre>{"result":true,"message":"","data":{"reqSeqNo\":"21082705204272-1000100000-5024668358A3BF1C-02-1022-8044\"}}</pre>

2.8. Remote door opening (1022)

parameter declaration

Instructions: 1022 Description: UDP sends device data
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": "21082705204272-1000100000-5024668358A3BF1C-02-1022-8044" }</pre>
Response parameters
<pre>{"result":true,"message":"","data":{"reqSeqNo\":"21082705204272-1000100000-5024668358A3BF1C-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": "21082705204272-1000100000-5024668358A3BF1C-02-102 3-8044" }</pre>
Response parameters
<pre>{"result":true,"message":"","data":{"reqSeqNo\":\"21082705204272-1000100000-5024668358A3BF1C-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": "21082705204272-1000100000-5024668358A3BF1C-02-1022-8044" }</pre>
Response parameters
<pre>{"result":true,"message":"","data":{"reqSeqNo\":\"21082705204272-1000100000-5024668358A3BF1C-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": "FFFEFD FCFBFAF9F8F7F6F5F4F3F2F1F0" // 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": 1014, // 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)

Parameter details

Order: 1026 Description: UDP sends device data (device IP, required parameter
<pre>{ "pass": "", "deviceKey": "FFFFFFFFFFFFFFFF", "appCode": 1026, "accNo": 0, // Personnel account "regType": 0, // Type of registration, 0: fingerprint 1: palm vein "reqSeqNo": "21082705204272-1000100000-5024668358A3BF1C-02-1025-8044" // Flow number, software customization, device reply back }</pre>
Response parameters(regType=0)
<pre>{"result": true, "message": "", "data": { "reqSeqNo": "21082705204272-1000100000-5024668358A3BF1C-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": "21082705204272-1000100000-5024668358A3BF1C-02-1026-8044", "palmId": 1, // Whether there is a palm vein(0: No 1: Yes) "palmInfo": "", // base64 formatted string }</pre>



3.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 /MJAapi/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,"protocolVersion":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
Device sends a service request parameter <pre>{ "head": { "companyId": 1000100000, // 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> 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)".
Service response parameters
Response to general <pre>public class ResultInfo<T> { 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> For example: successful check-in reply <pre>{"result": 1, "message": "", "data": "", "serverTime": "2021-06-24"}</pre>

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

Metho	URL:			
regist	/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)"
lableSN	Label serial	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

Metho	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
reply				
param	descri	type	Mus	Additio
paramVer	Version number of	Json	Y	
heartInterval	Heartbeat time	Int	Y	second

instance



equipment requirement:

```
{"head":{"companyId":1000100000,"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. et 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

Metho	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
reply				



param	descri	type	Mus	Additio
		Json	Y	Up to 2 groups
partId	Time period	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
partEnd	terminal time	Int		1800 for 18:00
partVWay	verification mode	Int		Time-period validation type

instance

equipment requirement:

```
{"head":{"companyId":1000100000,"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,"partSection":[{"partBegin":0,"partEnd":2359,"partVWay":10},{partBegin":0,"partEnd":0,"partVWay":0},{partBegin":0,"partEnd":0,"partVWay":0},{partBegin":0,"partEnd":0,"partVWay":0},{partBegin":0,"partEnd":0,"partVWay":0},{partBegin":0,"partEnd":0,"partVWay":0}]}]}
```

3.4. Get the time group ("/ MJApi / time_group")

explain:

Parameter details

Metho	URL:			
Time	/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
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,
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

instance

equipment requirement:

```
{
  "head": {
    "companyId": 1000100000,
    "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
    },
    {
      "groupId": 2,
      "paramVer": 2,
      "opType": 1,
      "tWeekSun": 0,
      "tWeekMon": 0,
      "tWeekTue": 0,
      "tWeekWed": 0,
      "tWeekThu": 0,
      "tWeekFri": 0,
      "tWeekSat": 0
    },
    {
      "groupId": 3,
      "paramVer": 3,
      "opType": 1,
      "tWeekSun": 0,
      "tWeekMon": 0,
      "tWeekTue": 0,
      "tWeekWed": 0,
      "tWeekThu": 0,
      "tWeekFri": 0,
      "tWeekSat": 0
    },
    {
      "groupId": 4,
      "paramVer": 4,
      "opType": 1,
      "tWeekSun": 0,
      "tWeekMon": 0,
      "tWeekTue": 0,
      "tWeekWed": 0,
      "tWeekThu": 0,
      "tWeekFri": 0,
      "tWeekSat": 0
    },
    {
      "groupId": 5,
      "paramVer": 5,
      "opType": 1,
      "tWeekSun": 0,
      "tWeekMon": 0,
      "tWeekTue": 0,
      "tWeekWed": 0,
      "tWeekThu": 0,
      "tWeekFri": 0,
      "tWeekSat": 0
    }
  ]
}
```

3.5. Get the device parameters ("/ MJApi / dev_param")

explain:



Parameter details

Method	URL:			
	/MJApi/dev_param			
equipment requirement				
parame	descri	type	Mus	Additional
head	Directive head	Json	Y	Refer to Common Parameters
paramVer	version number	Int	Y	Maximum version number of the current
reply				
parame	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	Int	Y	After setting, the time group is normally open, and outside the



Silicon Wireless Systems Pvt. Ltd.

	group			segment, it is online, Enable: doorWorkWay=1,timeGroup>0 Cancel: doorWorkWay=0
openPwd	Dynamic password (remote password)	String	N	6-digit number, no empty string set for distribution
entrAndExitType	Setting of entrance and exit directions	Int	N	0: Entering 1: Leaving
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 3:Italy 4:Russian 5:Indonesian 6:Spanish 7:Portuguese 8:Vietnamese 9:Arabic 10:Thai (Multi language versions, device specific)
faceRepeatEnable	Continuous recognition enable switch	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

instance



equipment requirement:

```
{"head":{"companyId":1000100000,"deviceKey":"DF60A894035E4832","doorType":2,"termId":1,"termType":70,"protocolVer":1,"signString":"F9901E9D99E4F75EA1DD819085265AB4"},"paramVer":0}
```

Software reply:

```
{"result":1,"message":"success","data":{"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}], "doorParam":[{"doorWorkWay":0,"appKeyButton":1,"timeGroup":0,"closeDelay":5}]}}
```

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				
parameter	descri	type	Mus	Additio
head	Directive head	Json	Y	Refer to Common Parameters
paramVer	version number	Int	Y	Maximum version number of the
reply				
paramet	descri	type	Mus	Additiona
paramVer	version number	Int	Y	
accNo	Personnel account	Int	Y	
name	Name of	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	Format: 'yyyy MM dd HH: mm: ss' is valid for minutes
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	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	The hexadecimal string converted from the Byte array can be obtained through the Udp protocol by "obtaining registration data based on account number (1026)"
fp2	Second fingerprint data	String	N	The hexadecimal string converted from Byte array
palmId	Is there a palm vein	Int	Y	None: 0, Yes: 1, Permission must include palm vein data
palmInfo	palmar vein data	Base64	N	You can obtain registration data based on account number (1026) through the Udp protocol

instance

equipment requirement:

```
{"head":{"companyId":1000100000,"deviceKey":"DF60A894035E4832","doorType":2,"termId":1,"termType":70,"protocolVer":1,"signString":"F9901E9D99E4F75EA1DD819085265AB4"},"paramVer":0}
```

Software reply:

```
{"result":1,"message":"success","data":{"paramVer":18,"accNo":109,"empNo":"109","name":"109","opType":1,"doorId":1,"cardSN":"147275","phoneNo":"","password":"147275","beginExpDate":"2000-01-01 00:00:00","endExpDate":"2099-01-01 00:00:00","timeGroup":0,"faceId1":1,"faceImage1":null,"isManager":0,"fpId1":1,"fpId2":1,"fp1":"...","fp2":"..."}}
```

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

Metho	URL:
	/MJApi/up_log



Silicon Wireless Systems Pvt. Ltd.

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
reqSeqNo	Record serial	String	N	
logType	record type	Int	N	(0: Normal record, 1: record illegal
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 Way of passage
alarmCode	Alarm code	Int	N	consult Alarm code
passStatus	Pass state	Int	N	0: Pass 1: No pass
doorId	Door number	Int	N	Face default 1
readId		Int	N	Face default 1
temperature	thermometric	Int	N	Temperature measuring temperature, 4 digits, display the temperature needs to be divided by
capJpgBase64	Capture photos	Base64	N	Device upload does not have this field. Please ignore it
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": "166763541351012600",
    "logType": 1,
    "accNo": 13614991,
    "passTime": "2022-11-05 16:03:33",
    "operation": 0,
    "alarmCode": 91,
    "passStatus": 1,
    "doorId": 1,
    "readId": 1,
    "temperature": 0
  },
  "body": {
    "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
    }
  }
}
```

Software reply:

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



3.8. Heartbeat (" / MJAapi / 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

Metho	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
devConfigVer	Device version	Int	Y	The device currently has the
timePartVer	Time period	Int	Y	The device currently has the
timeGroupVer	Time group	Int	Y	The device currently has the
powerParamVer	Permission	Int	Y	The device currently has the
reply				
param	descri	type	Mus	Additio
compParamVer	Enterprise version	Int	Y	Cloud current largest version
devConfigVer	Device version	Int	Y	Cloud current largest version
timePartVer	Time period	Int	Y	Cloud current largest version
timeGroupVer	Time group	Int	Y	Cloud current largest version
powerParamVer	Permission	Int	Y	Cloud current largest version

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,"powerParamVer":0}}
```

----The version number of the reply is the same as the device request, and the device is waiting for the next heartbeat of the request

软件回复:

```
{"result":1,"message":"success","data":{"compParamVer":8,"devConfigVer":1,"timePartVer":15,"timeGroupVer":13,"powerParamVer":0}}
```

--The version number replied by compParamVer is larger than the device request. The device will start requesting the enterprise parameter ("/MJApi/comp_param") interface in sequence and wait for the duration of the request heartbeat until there is no update

3.9. Obtain scheduled door opening 2("/MJApi/time_timing")

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

Parameter details

Metho		URL:			
		/MJApi/time_timing			
equipment requirement					
param		descri	type	Mus	Additio
head		Directive head	Json	Y	Refer to Common Parameters
reply					
param		descri	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	Int	Y	For example, 8:00 is represented by
	endDate1	End time of time	Int	Y	For example, 14:00 is represented
	timingMode1	Door control	Int	Y	0: Online 1: Normally open 2:
	beginDate2	Start time of time	Int	Y	For example, 8:00 is represented by
	endDate2	End time of time	String	Y	For example, 14:00 is represented
	timingMode2	Door control	Int	Y	0: Online 1: Normally open 2:
	beginDate3	Start time of time	Int	Y	For example, 8:00 is represented by



endDate3	End time of time	String	Y	For example, 14:00 is represented
timingMode3	Door control	Int	Y	0: Online 1: Normally open 2:

instance

equipment requirement:

```
{ "head": { "companyId": "1000121628", "deviceKey": "FFFF8ED7D2590054", "doorType": 1, "termId": 0, "termType": 35, "protocolVer": 2, "signString": "14ba8c590a93f9fa67280e91fc1245" }}
```

Software reply:

```
{ "result": 1, "message": "success", "data": { "timing": [ { "beginDate1": 800, "endDate1": 1200, "timingMode1": 0, "beginDate2": 0, "endDate2": 0, "timingMode2": 0, "beginDate3": 0, "endDate3": 0, "timingMode3": 0 }, { "beginDate1": 800, "endDate1": 1200, "timingMode1": 1, "beginDate2": 0, "endDate2": 0, "timingMode2": 0, "beginDate3": 0, "endDate3": 0, "timingMode3": 0 }, { "beginDate1": 800, "endDate1": 1200, "timingMode1": 1, "beginDate2": 0, "endDate2": 0, "timingMode2": 0, "beginDate3": 0, "endDate3": 0, "timingMode3": 0 }, { "beginDate1": 0, "endDate1": 0, "timingMode1": 0, "beginDate2": 0, "endDate2": 0, "timingMode2": 0, "beginDate3": 0, "endDate3": 0, "timingMode3": 0 }, { "beginDate1": 0, "endDate1": 0, "timingMode1": 0, "beginDate2": 0, "endDate2": 0, "timingMode2": 0, "beginDate3": 0, "endDate3": 0, "timingMode3": 0 }, { "beginDate1": 0, "endDate1": 0, "timingMode1": 0, "beginDate2": 0, "endDate2": 0, "timingMode2": 0, "beginDate3": 0, "endDate3": 0, "timingMode3": 0 }, { "beginDate1": 0, "endDate1": 0, "timingMode1": 0, "beginDate2": 0, "endDate2": 0, "timingMode2": 0, "beginDate3": 0, "endDate3": 0, "timingMode3": 0 }, { "beginDate1": 0, "endDate1": 0, "timingMode1": 0, "beginDate2": 0, "endDate2": 0, "timingMode2": 0, "beginDate3": 0, "endDate3": 0, "timingMode3": 0 }, { "beginDate1": 0, "endDate1": 0, "timingMode1": 0, "beginDate2": 0, "endDate2": 0, "timingMode2": 0, "beginDate3": 0, "endDate3": 0, "timingMode3": 0 }, { "beginDate1": 0, "endDate1": 0, "timingMode1": 0, "beginDate2": 0, "endDate2": 0, "timingMode2": 0, "beginDate3": 0, "endDate3": 0, "timingMode3": 0 } ] }
```

4.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
20	All certified or

4.2. Record alarm code

Value (decimal)	meaning
0	normal



80	Illegal card
81	The door timed out and did not close
83	fire alarm
86	Illegal opening and closing of doors
90	Expired
91	stranger
95	Alarm deactivated
113	Incorrect 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



Silicon Wireless Systems Pvt. Ltd.

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