

Elevator controller protocol description

1. Version introduction	5	4
2. Agreement description	6	5
2.1. HTC protocol, Post mode, encoded as UTF-8, device port: 14460	6	5
2.2. General return description of the interface:	6	5
2.3. Operation process:	7	6
2.4., Precautions	7	6
3. Hhttp Interface Call (Method: POST-utf-8)	8	7
3.1. Set the Equipment Communication Password (setPassword)	8	7
3.1.1. Request for data	8	7
3.1.2.postman instance	8	7
3.1.3. instance	8	7
3.2. Obtain equipment Serial number (getDeviceKey).....	9	8
3.2.1. Request for data	9	8
3.2.2.postman instance	9	8
3.2.3. instance	9	8
3.3. Obtain the equipment OEM (getOEM).....	10	9
3.3.1. Request for data	10	9
3.3.2.postman instance	10	9
3.3.3. instance	10	9
3.4. Acquisition acquisition Time (getDeviceTime).....	11	10
3.4.1. Request for data	11	10
3.4.2.postman instance	11	10
3.4.3. instance	11	10
3.5. Set time (setTime).....	12	11
3.5.1. Request for data	12	11
3.5.2.postman instance	12	11
3.5.3. instance	12	11
3.6. Obtaining equipment information (getDeviceInfo)	13	12
3.6.1. Request for data	13	12
3.6.2.postman instance	13	12
3.6.3. instance	13	12
3.7. Set the equipment parameters (setConfig) - (controller)	14	13
3.7.1. Request for data	14	13
3.7.2.postman instance	16	15
3.7.3. instance	16	15
3.8. Obtain equipment parameters (getConfig) - (controller)	17	16
3.8.1. Request for data	17	16
3.8.2.postman instance	17	16
3.8.3. instance	17	16
3.9. Set equipment parameters (setConfig) - (face ladder control).....	18	17
3.9.1. Request for data	18	17
3.9.2.postman instance	20	19
3.9.3. instance	21	20
3.10. Obtain Equipment Parameters (getConfig) - (face ladder control).....	21	20
3.10.1. Request for data	21	20
3.10.2. postman Example.....	21	20
3.10.3. instance	21	20
3.11. Restart the equipment (restartDevice)	22	21

3.11.1. Request for data	22	21
3.11.2. postman Example.....	23	22
3.11.3. instance	23	22
3.12. Initialization Equipment (resetDevice).....	23	22
3.12.1. Request for data	23	22
3.12.2. postman Example.....	24	23
3.12.3. instance	24	23
3.13. Remote door opening (openDoorControl).....	24	23
3.13.1. Request for data	24	23
3.13.2. postman Example.....	25	24
3.13.3. instance	25	24
3.14. Issue period (up to 5 groups at a time) (setTimePart).....	25	24
3.14.1. Request data	25	24
3.14.2. postman Example.....	26	25
3.14.3. instance	27	26
3.15. Issue time group (up to 5 groups at a time) (setTimeGroup).....	27	26
3.15.1. Request for data	27	26
3.15.2. postman Example.....	28	28
3.15.3. instance	29	28
3.16. Issue holidays (up to 20 groups at a time) (setTimeHoliday).....	29	29
3.16.1. Request for data	29	29
3.16.2. postman Example.....	30	29
3.16.3. instance	30	30
3.17. Issue authority (setUserPower) - (controller)	30	30
3.17.1. Request for data	31	30
3.17.2. postman Example.....	31	31
3.17.3. instance	32	31
3.18. Get permission (getUserPower) - (controller)	33	32
3.18.1. Request for data	33	32
3.18.2. postman Example.....	33	33
3.18.3. instance	34	33
3.19. Issue authority (setUserPower) - (face ladder control).....	34	34
3.19.1. Request for data	34	34
3.19.2. postman Example.....	35	35
3.19.3. instance	36	35
3.20. Obtain permission (getUserPower) - (face ladder control).....	37	36
3.20.1. Request data	37	36
3.20.2. postman Example.....	37	37
3.20.3. instance	37	37
3.20.4. postman Example.....	38	38
3.20.5. instance	39	38
3.21. Empty authority (deleteAllUserPower)	39	38
3.21.1. Request data	39	38
3.21.2. postman Example.....	40	39
3.21.3. instance	40	39
3.22. Number of acquired records (getLogCount).....	40	39
3.22.1. Request data	40	39
3.22.2. postman Example.....	41	40
3.22.3. instance	41	40
3.23. Obtain record Content (getLogInfo)	41	41

3.23.1. Request data	41	41
3.23.2. postman Example.....	42	41
3.23.3. instance	42	41
3.24. Delete the latest record (deleteNewLog)	43	42
3.24.1. Request data	43	42
3.24.2. postman Example.....	43	43
3.24.3. instance	43	43
3.25. Empty Equipment Record (deleteAllLog)	44	43
3.25.1. Request data	44	43
3.25.2. postman Example.....	44	44
3.25.3. instance	44	44
3.26. Set up the equipment active upload record (setLogCallBack).....	45	44
3.26.1. Request data	45	45
3.26.2. postman Example.....	46	46
3.26.3. instance	46	46
3.27. Reset Administrator Rights (deleteAllUserManager) - (face ladder control).....	47	47
3.27.1. Request data	47	47
3.27.2. postman Example.....	48	48
3.27.3. instance	48	48
4. Advanced features.....	48	48
4.1. Set the timing door opening (setTimingControl).....	48	48
4.1.1. Request for data	49	49
4.1.2. postman instance	49	49
4.1.3. instance	50	50
5. The UDP protocol	51	51
5.1. General return description of the interface	51	51
5.2. Search Equipment (UDP Broadcast) (1001).....	51	51
5.2.1. Parameter description.....	51	51
5.3. Configuration of IP (UDP Broadcast) (1002).....	52	52
5.3.1. Parameter description.....	52	52
5.4. Configuration of OEM (UDP Broadcast) (1010).....	53	53
5.4.1. Parameter description.....	53	53
6. Type definition.....	54	54
6.1. Validation Type-Time period validation type.....	54	54
6.2. Record the alarm code	54	54
6.3. Record the traffic modes.....	55	55
6.4., Device type	56	56
6.5., Control gate type.....	56	56
6.6. Door control mode	56	56

1. Version introduction

[illegible]

2. protocol specification

2.1. http protocol, Post mode, coded as UTF-8, device port: 14460

Interface root Address: http: // Device ip Address: 14460 /

Interface form: provide external service through HTTP request.

Interface security: the initial call to the interface needs to set the device password (pass) first, and then call any later Interfaces require the incoming device password (pass) as the interface security verification secret key.

Interface example illustration: All examples in this chapter use postman,

A) POST, with the request parameters in the body

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

2.2. Common interface return description:

```
public class ResultInfo<T> {  
private Boolean result; // Whether the operation is successful, the success is  
true, and the failure is false  
Private String data; // Business data returned by the interface  
can be numeric, string or collection, etc  
Private String message; // Information returned by the interface,  
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.

2.3. operating process:

Equipment for the first time: start the device into the main interface. The device has no initial password.

2.4. matters need attention

- When calling the device interface, do not call the interface of the same device on other client servers.
- If the call interface is returned for parameter exception, check in these steps:
 - 1) Check whether the parameter name is misspelled and whether there is a space or return in the parameter name
 - 2) Check whether the parameter values meet the specification, such as the Int value is not defined, the id contains illegal characters other than the number letters, etc
 - 3) Whether Json contains the space (using the Json string returned directly by the interface needs to remove the space), whether there is more or less? " Please carefully check the postman example of each interface
- If the call interface returns empty, indicating an url error (IP error, spelling error, field omission, etc.)

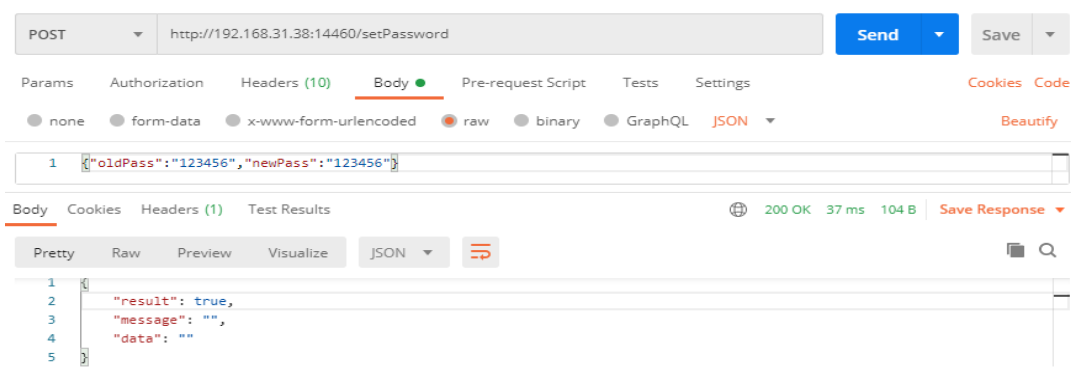
3. Http Interface Call (Method: POST-utf-8)

3.1.Set the device communication password (setPassword)

3.1.1. Request data

Method	URL			
POST	The http: // Device IP: 14460 / setPassword			
parameter name	description	type	required	explain
oldPass	The old password	String	Y	New device or reset (restore initialization) device, oldPass "", change the password, respectively into the new and old password. This interface call does not require an incoming pass Password length: 6 bits Format: number
newPass	New password	String	Y	

3.1.2. postman instance



3.1.3. instance

- Request example

```
{  
  "pass": ""  
}
```

- Return an example

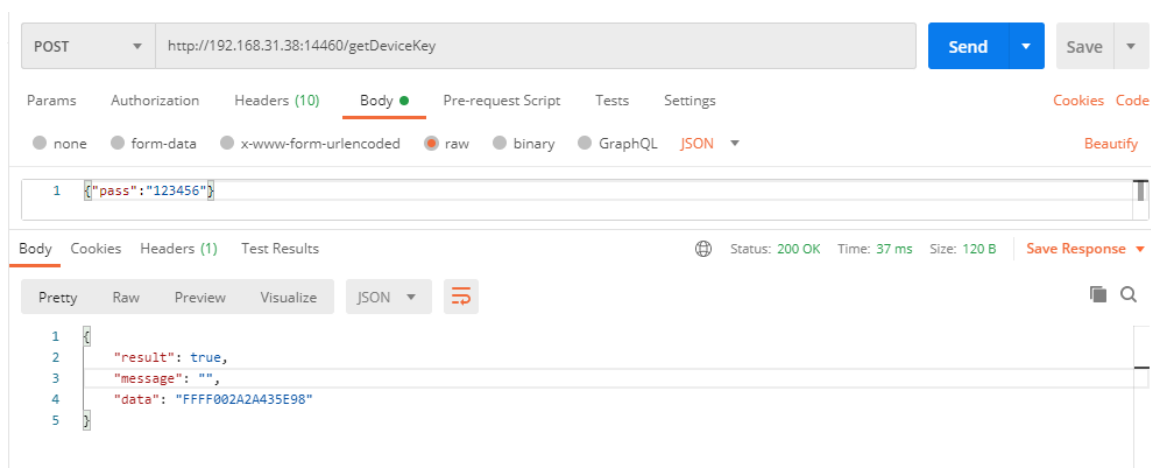
```
{  
  "result": true,  
  "message": "",  
  "data": "5024668308708C1C"  
}
```


3.2. Get the device serial number (getDeviceKey)

3.2.1. Request data

Method	URL			
POST	The http: // Device IP: 14460 / getDeviceKey			
parameter name	description	type	Must	Additional instructions
pass	password	String	N	This parameter does not pass

3.2.2. postman instance



3.2.3. instance

● Request example

```
{
  "pass": ""
}
```

● Return an example

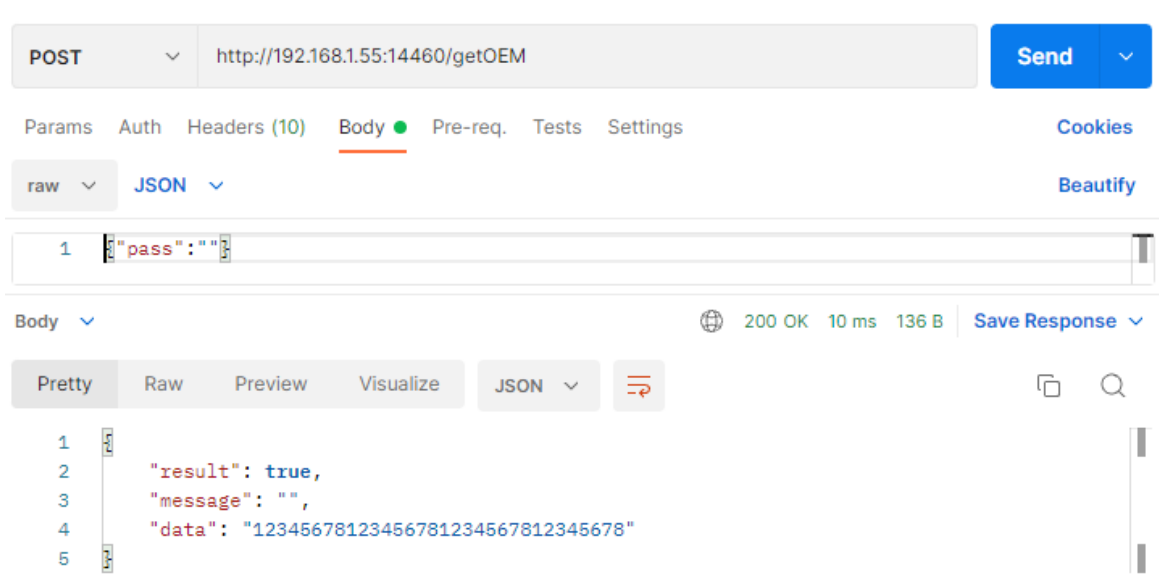
```
{
  "result": true,
  "message": "",
  "data": "5024668308708C1C"
}
```

3.3. Obtain the device OEM (getOEM)

3.3.1. Request data

Method	URL			
POST	The http: // Device IP: 14460 / getOEM			
parameter name	description	type	Must pass	Additional instructions
pass	pass word	String	Y	

3.3.2. postman instance



3.3.3. instance

● Request example

```
{  
  "pass": ""  
}
```

● Return an example

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

```

"message": "",
"data": "12345678123456781234567812345678"
}

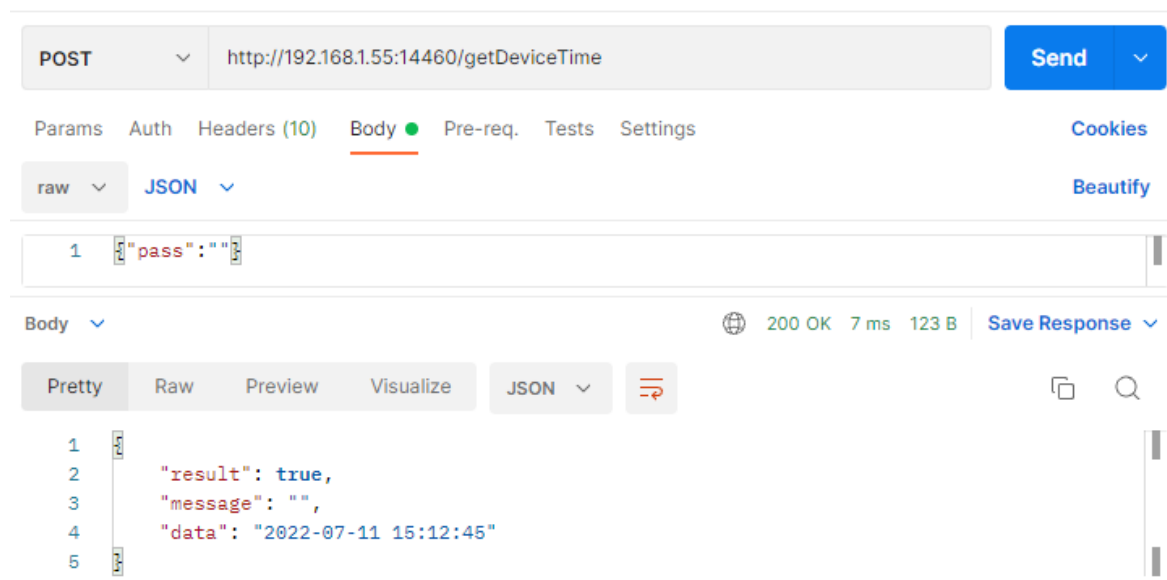
```

3.4. Acquisition of device time (getDeviceTime)

3.4.1. Request data

Method	URL			
POST	The http: // Device IP: 14460 / getDeviceTime			
parameter name	description	type	Must pass	Additional instructions
pass	pass word	String	Y	

3.4.2. postman instance



3.4.3. instance

● Request example

```

{
  "pass": ""
}

```

```
}
```

● Return an example

```
{
  "result": true,
  "message": "",
  "data": "2022-07-11 15:12:45"
}
```

3.5. Set Time (setTime)

3.5.1. Request data

Method	URL			
POST	The http: // Device IP: 14460 / setTime			
parameter name	description	type	Must pass	Additional instructions
pass	password	String	Y	
serverTime	password	String	Y	Time format: yyyy-MM-ddHH:mm:ss

3.5.2. postman instance

POST http://192.168.1.55:14460/setTime

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies

raw JSON Beautify

1 {\"pass\": \"\", \"serverTime\": \"2020-07-11 15:18:15\"}

Body 200 OK 31 ms 104 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": true,
3   "message": "",
4   "data": ""
5 }
```

3.5.3. instance

● Request example

```
{  
  "pass": "",  
  "serverTime": "2020-07-11 15:18:15"  
}
```

● Return an example

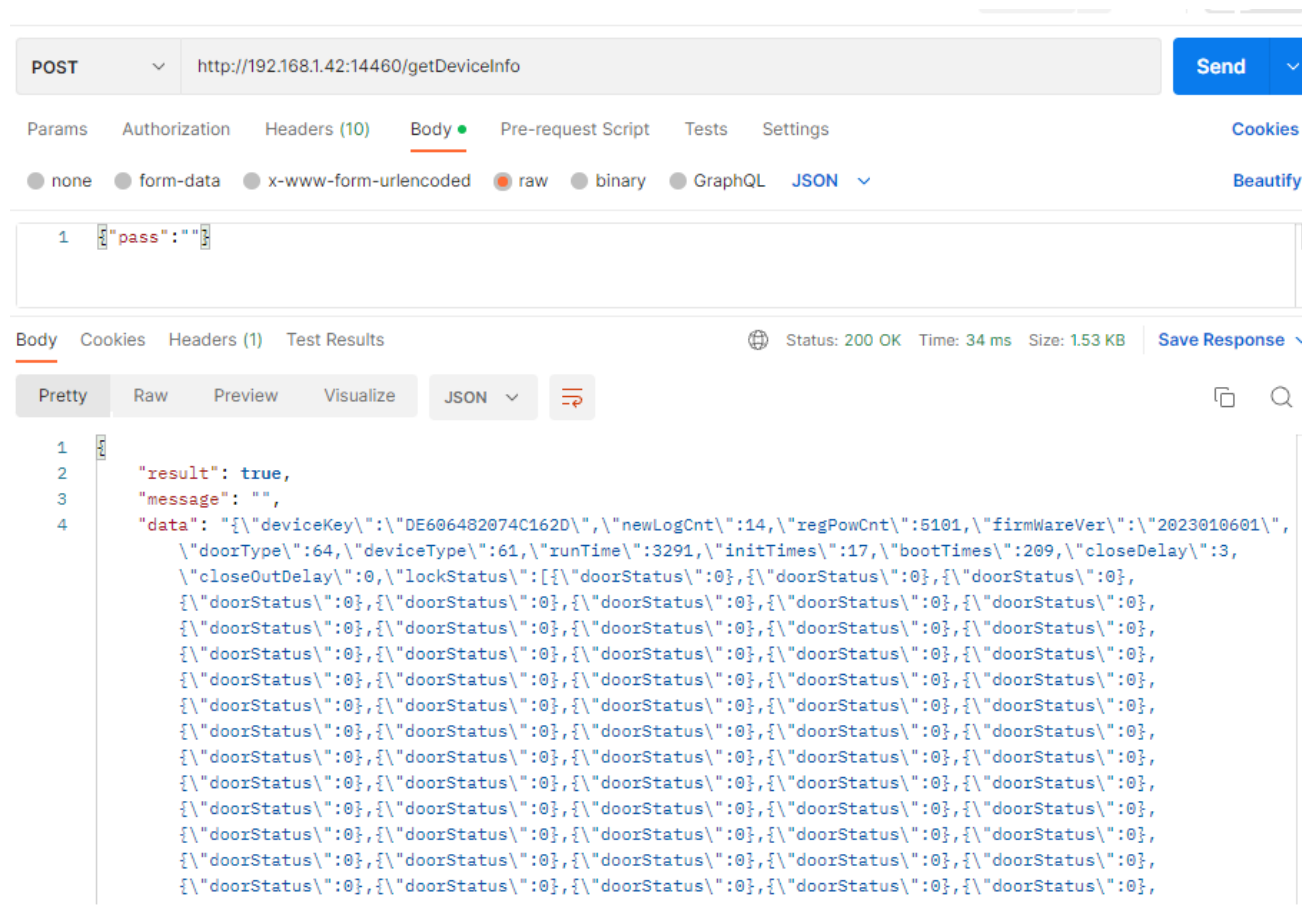
```
{  
  "result": true,  
  "message": "",  
  "data": ""  
}
```

3.6. Obtain equipment information (getDeviceInfo) - (controller)

3.6.1. Request data

Method	URL			
POST	The http: // Device IP: 14460 / getDeviceInfo			
parameter name	description	type	Must pass	Additional instructions
pass	password	String	Y	

3.6.2. postman instance



3.6.3. instance

- **Request example**

```
{
  "pass": ""
}
```

- **Return an example**

[illegible]

[illegible]

explain:

newLogCnt: Number of records

regPowCnt: Number of permissions

firmWareVer: 2021082501 Version No

doorId: Door number

closeDelay: Door opening delay (seconds)

closeOutDelay: Closing delay (seconds)

lockStatus: Door magnetic state status: 0: off (short circuit) 1: on (off)

doorStatus: Door control method 0: online 1: open 2: closed

3.7. Obtain equipment information (getDeviceInfo)- (Face ladder control)

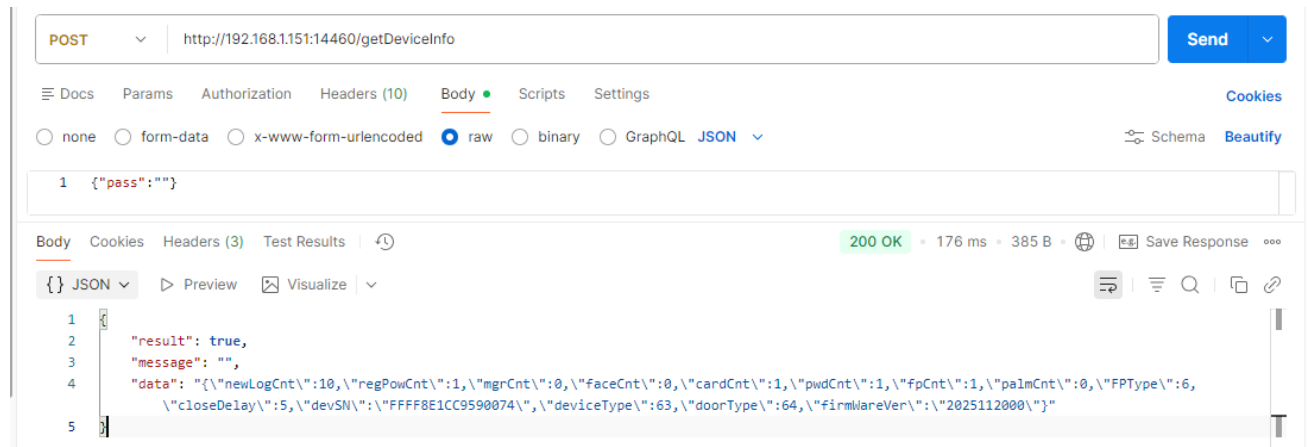
3.7.1. Request data

Method	URL			
POST	The http: // Device IP: 14460 / getDeviceInfo			
parameter name	description	type	Must pass	Additional instructions
pass	password	String	Y	
Reply parameter name	description	type	Must pass	Additional instructions
newLogCnt	Number of new records	Int	Y	
regPowCnt	Number of registrations	Int	Y	

mgrCnt	Number of administrators	Int	Y	
faceCnt	Number of facial recognition devices	Int	Y	
cardCnt	Number of cards	Int	Y	
pwdCnt	Number of permissions	Int	Y	
fpCnt	Number of fingerprints	Int	Y	
palmCnt	Number of palmar veins	Int	Y	
FPTYPE	Number of fingerprints	Int	Y	
devSN	Device serial number	Int	Y	
deviceType	Device Type	Int	Y	
doorType	Number of floors	Int	Y	Default 64
firmWareVer	version	Int	Y	

	number			

3.7.2. postman instance



3.7.3. instance

● Request example

```
{
  "pass": ""
}
```

● Return an example

```
{
  "result": true,
  "message": "",
  "data":
    "{ \"newLogCnt\":10, \"regPowCnt\":1, \"mgrCnt\":0, \"faceCnt\":0, \"cardCnt\":1, \"pwdCnt\":1, \"fpCnt\":1, \"palmCnt\":0, \"FPTType\":6, \"closeDelay\":5, \"devSN\": \"FFFF8E1CC9590074\", \"deviceType\":63, \"doorType\":64, \"firmWareVer\": \"2025112000\" }"
}
```

3.8. Set the Equipment parameters (setConfig) - (controller)

3.8.1. Request data

Method		URL			
POST		The http: // Device IP: 14460 / setConfig			
parameter name	description	type	Must pass	Additional instructions	
pass	password	String	Y		
logSaveType	Storage method	Int	Y	When the record storage is full, 0: stop, 1: override	
reVerify	swipe card interval	Int	Y	Default 0	
manual_control	Open the manual ladder (button)		Y	Open manual opening ladder (button) open 1 by default	
appFirstFloor	Ignoring the first layer		Y	Whether to ignore the first floor (when the single floor is direct, send the first floor and the direct floor, can ignore the first floor) default open 1	
fire_alarm_control	Fire control mode	Int	Y	1: often open 2: often close	
closeDelay	Close time delay seconds	Int	Y	Decimal number, in seconds, range 0 to 255	
doorParam	Set set of gate parameters	Json	Y	Send the door number in order from small to large	
doorWorkWay	work	Int	Y	0 Ordinary 1 timing, factory	

● Return an example

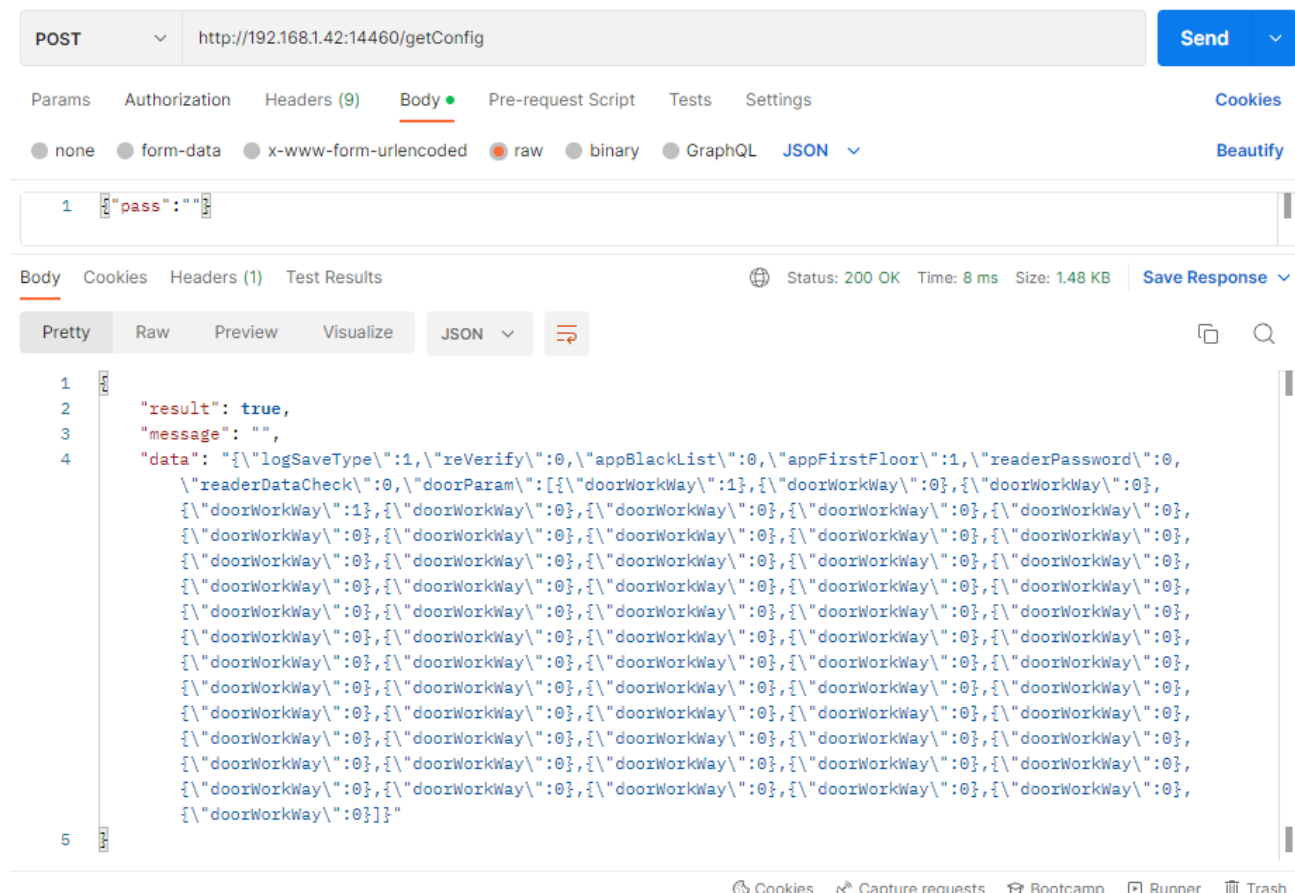
```
{  
  "result": true,  
  "message": "",  
  "data": ""  
}
```

3.9. Obtain Equipment parameters (getConfig) - (Controller)

3.9.1. Request data

Method	URL			
POST	The http: / / Device IP: 14460 / getConfig			
parameter name	description	type	Must pass	Additional instructions
pass	password	String	Y	

3.9.2. postman instance



3.9.3. instance

- **Request example**

```
{
  "pass": ""
}
```

- **Return an example**

[illegible]

[illegible]

Note: Parameters are set in the same way as the equipment parameters

3.10.Set Equipment Parameters (setConfig) - (Face ladder control)

3.10.1. Request data

Method	URL			
POST	The http: // Device IP: 14460 / setConfig			
parameter name	description	type	Must pass	Additional instructions
pass	password	String	Y	
sumFloor	Total floor number	Int	Y	Default value is 16, and maximum is 64
firstFloor	Start floor	Int	Y	Default value: 1; range-5~63
secFloorTime	The selected layer page displays the time	Int	Y	After the device enters the layer selection page, the exit time when there is no operation, the unit second (5-60 seconds) default: 15 seconds
faceLiving	Vivian valve value	Int	Y	Default value 10 0: Do not enable liveness detection

					Integer with valid values ranging from 10 to 99 (subject to device adjustments)
faceThreshold		Vivian valve value	Int	Y	Default 50 Integer with valid values from 0 to 99 (subject to device adjustment)
faceDistance		Identify 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
lightEnable		Fill light switch	Int	N	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	N	Support one sections
	beginTime	Time slot start time	N	N	8: 00 is represented by 800
	endTime	End time of time period	N	N	18: 00 is represented by 1800
displayStandbyTime		Screen-off time	Int	N	Set the screen hold time, 0 does not sleep (unit: seconds), (0-600s)
Language		language	Int	N	0: Chinese 1: English
screenType		Screen saver type	Int	N	0: Black Screen 1: Screen saver
faceRepeatEnable		Continuous	Int	N	0: Not enabled, 1: Enable

	recognition enable switch			default: 0
Volume	sound	Int	N	0 - 100
repeatOpenTime	Repeat confirmation time	Int	N	0-60 minutes, interval for saving facial recognition records
wgOutputEnable	Do you want to enable output WG	Int	Y	0: No, 1: YesDefault: 1
wgOutputFormat	WG input and output format	Int	Y	0: Card number, 1: Employee numberDefault: 1
wgOutputModel	WG input/output mode	Int	Y	0: "WG26",1: "WG34"Default: 1

3.10.2. postman instance

3.10.3. instance

● Request example

```
{
  "pass": "",
  "wgOutputEnable": 1,
  "wgOutputModel": 0,
  "wgOutputFormat": 1,
  "repeatOpenTime": 1,
  "faceRepeatEnable": 1,
  "faceThreshold": 50,
  "faceLiving": 45,
```



```

    "faceDistance": 0,
    "Volume": 8,
    "Language": 1,
    "screenType": 0,
    "displayStandbyTime": 60,
    "lightEnable": 1,
    "lightTime": [
        {
            "beginTime": 1730,
            "endTime": 730
        }
    ],
    "firstFloor": 1,
    "sumFloor": 16,
    "secFloorTime": 5
}

```

● Return an example

```

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

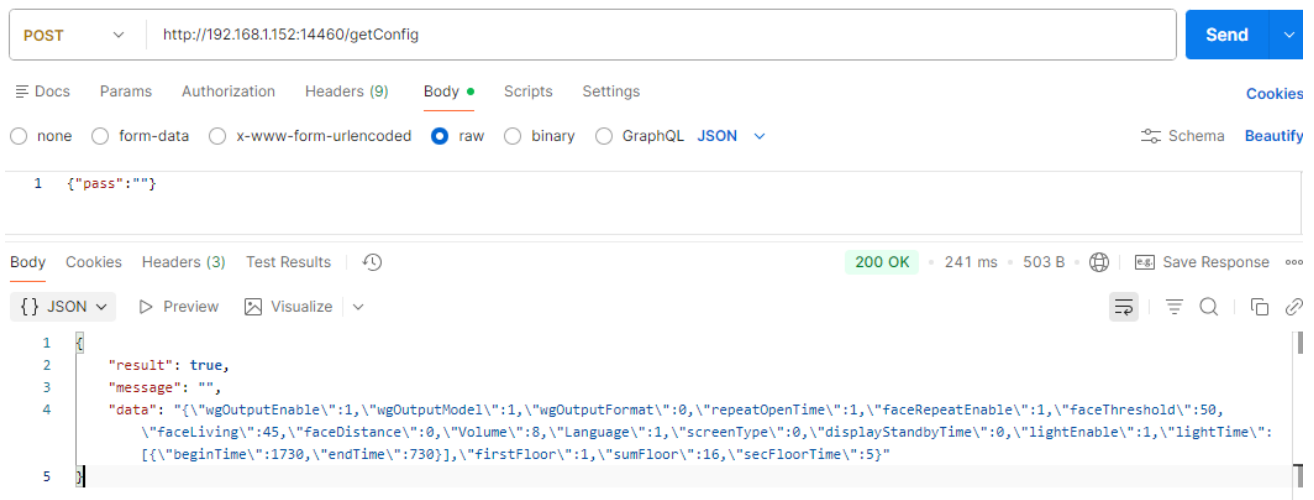
```

3.11.Obtain Equipment Parameters (getConfig) - (face ladder control)

3.11.1. Request data

Method	URL			
POST	The http: // Device IP: 14460 / getConfig			
parameter name	description	type	Must pass	Additional instructions
pass	password	String	Y	

3.11.2. postman instance



3.11.3. instance

● Request example

```
{
  "pass": ""
}
```

● Return an example

```
{
  "result": true,
  "message": "",
  "data":
  "{\"wgOutputEnable\\\":1,\"wgOutputModel\\\":1,\"wgOutputFormat\\\":0,\"repeatOpenTime\\\":1,\"faceRepeatE
nable\\\":1,\"faceThreshold\\\":50,\"faceLiving\\\":45,\"faceDistance\\\":0,\"Volume\\\":8,\"Language\\\":1,\"
screenType\\\":0,\"displayStandbyTime\\\":0,\"lightEnable\\\":1,\"lightTime\\\":[{\\\"beginTime\\\":1730,\"end
Time\\\":730}],\\\"firstFloor\\\":1,\"sumFloor\\\":16,\"secFloorTime\\\":5}\""
}
```

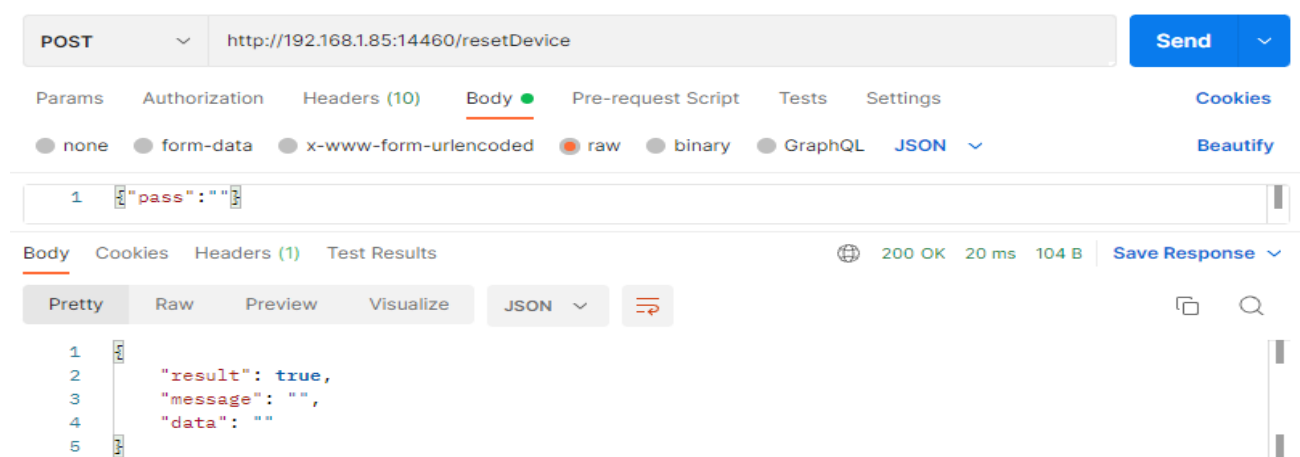
Note: Parameters are set in the same way as the equipment parameters

3.12.Restart device (restartDevice)

3.12.1. Request data

Method	URL			
POST	The http: // Device IP: 14460 / restartDevice			
parameter name	description	type	Must pass	Additional instructions
pass	password	String	Y	

3.12.2. postman instance



3.12.3. instance

● Request example

```
{
  "pass": ""
}
```

● Return an example

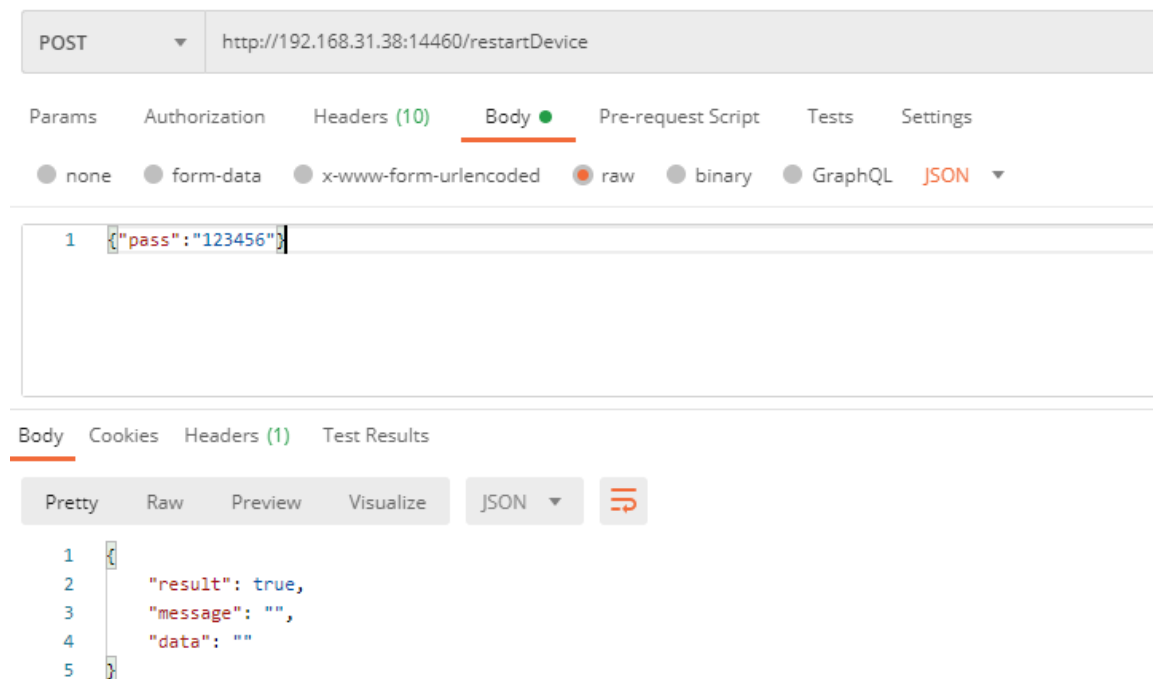
```
{
  "result": true,
  "message": "",
  "data": ""
}
```

3.13.Initialization Device (resetDevice)

3.13.1. Request data

Method	URL			
POST	The http: / / Device IP: 14460 / resetDevice			
parameter name	description	type	Must	Additional instructions
pass	password	String	Y	

3.13.2. postman instance



3.13.3. instance

● Request example

```
{  
  "pass": ""  
}
```

● Return an example

```
{
```

```

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

```

3.14.Remote door opening (openDoorControl)

3.14.1. Request data

Method	URL			
POST	The http: // Device IP: 14460 / openDoorControl			
parameter name	description	type	Must	Additional instructions
pass	password	String	Y	
doorId	Floor number	Int	Y	Starting from 1 (1~64)

3.14.2. postman instance

The screenshot shows a Postman interface for a POST request. The URL is `http://192.168.1.55:14460/openDoorControl`. The request body is a JSON object: `{ "pass": "", "doorId": 1 }`. The response is a 200 OK status with a response time of 79 ms and a body size of 104 B. The response body is displayed in JSON format: `{ "result": true, "message": "", "data": "" }`.

3.14.3. instance

● Request example

```

{
"pass": "",
"doorId": 1
}

```

```
}
```

● Return an example

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

3.15.Issue period (up to 5 groups at a time) (setTimePart)

Note: Up to 200 custom groups, numbered from 1 to 200

3.15.1. Request data

Method		URL			
POST		The http: // Device IP: 14460 / setTimePart			
parameter name		description	type	Must pass	Additional instructions
pass		password	String	Y	
timePart		Time period collection	Json	Y	
	partId	Time period number	Int	Y	1~200
	outControl	Extra-section control mode	Int	Y	0: Online 1: often open 2: often closed
	partSection	Time period collection	Json	Y	Set sets 6 small segments
	partBegin	Start time of time	Int	Y	If 8:00,800, not enabled start and end are 0
	partEnd	Time end time	Int	Y	1800 for 18:00
	partVWay	Validation type	Int	Y	The reference type defines the validation type
	inControl	Control mode in	Int	Y	0: Online 1: often open 2:


```
0,"inControl":0}}},{ "partId":5,"outControl":0,"partSection":[{"partBegin":1100,"partEnd":2000,"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},{ "partBegin":0,"partEnd":0,"partVWay":0,"inControl":0},{ "partBegin":0,"partEnd":0,"partVWay":0,"inControl":0}]]}}
```

Return an example

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

3.16.Issue time group (up to 5 groups at one time) (setTimeGroup)

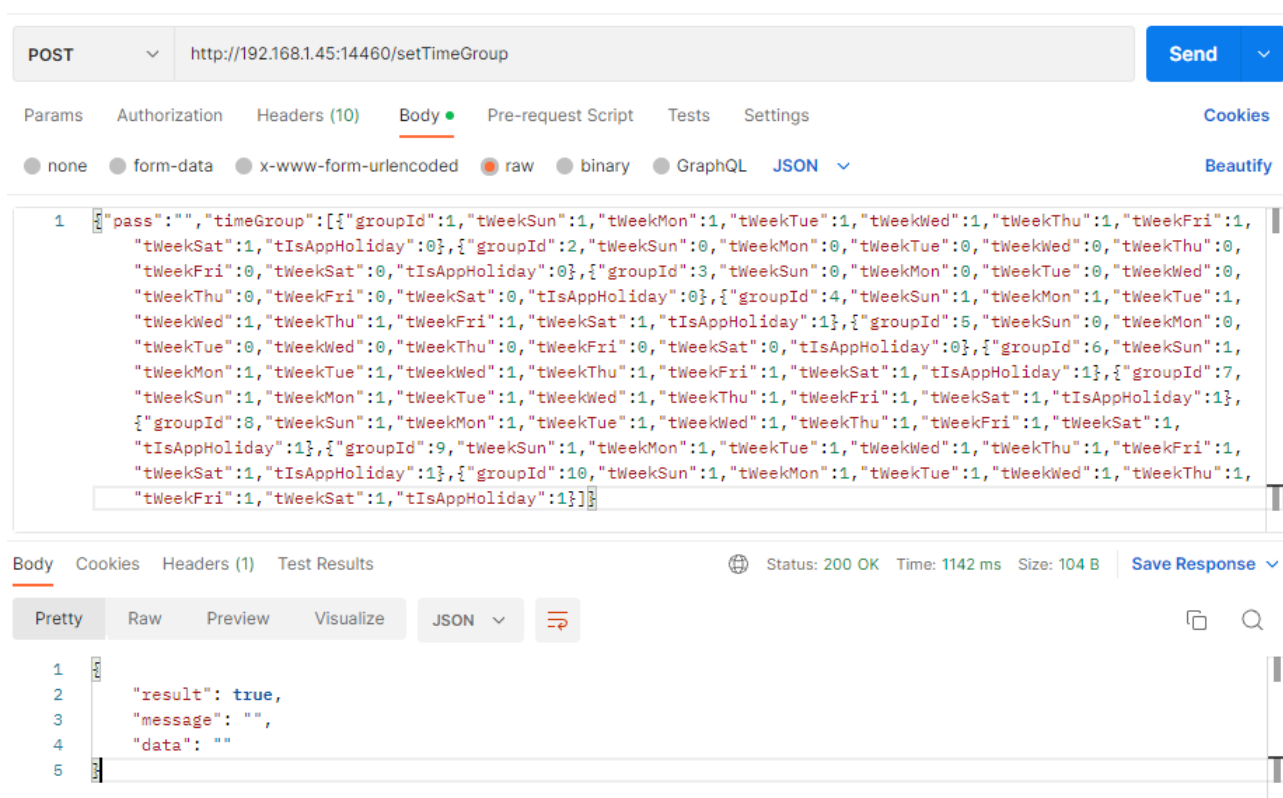
Note: Up to 200 groups can be customized, numbered from 1 to 200, default 0 and 255 groups do not need to be set separately, 0: allowed, 255 means no traffic. The time group of issuing authority does not support the setting of segment and segment control mode

3.16.1. Request data

Method		URL			
POST		The http: // Device IP: 14460 / setTimeGroup			
parameter name		description	type	Must pass	Additional instructions
pass		password	String	Y	
timeGroup		Time group collection			Issue 5 groups at a time
	groupId	Time group number	Int	Y	1~200
	tWeekSun	Select the time period on Sunday	Int	Y	Select the time period number
	tWeekMon	Select the period on Monday	Int	Y	Select the time period number
	tWeekTue	Select time	Int	Y	Select the time period

	periods on Tuesday			number
tWeekWed	Select time periods on Wednesday	Int	Y	Select the time period number
tWeekThu	Select time periods on Thursday	Int	Y	Select the time period number
tWeekFri	Select time periods on Friday	Int	Y	Select the time period number
tWeekSat	Select the time period on Saturday	Int	Y	Select the time period number
tIsAppHoliday	Whether the time group is higher than the holidays	Int	Y	1: Time group is higher than holidays, and time group is valid 0: Effective during holidays

3.16.2. postman instance



3.16.3. instance

● Request example

```
{
  "pass": "",
  "timeGroup": [
    {
      "groupId": 1,
      "tWeekSun": 1,
      "tWeekMon": 1,
      "tWeekTue": 1,
      "tWeekWed": 1,
      "tWeekThu": 1,
      "tWeekFri": 1,
      "tWeekSat": 1,
      "tIsAppHoliday": 0
    },
    {
      "groupId": 2,
      "tWeekSun": 0,
      "tWeekMon": 0,
      "tWeekTue": 0,
      "tWeekWed": 0,
      "tWeekThu": 0,
      "tWeekFri": 0,
      "tWeekSat": 0,
      "tIsAppHoliday": 0
    },
    {
      "groupId": 3,
      "tWeekSun": 0,
      "tWeekMon": 0,
      "tWeekTue": 0,
      "tWeekWed": 0,
      "tWeekThu": 0,
      "tWeekFri": 0,
      "tWeekSat": 0,
      "tIsAppHoliday": 0
    },
    {
      "groupId": 4,
      "tWeekSun": 1,
      "tWeekMon": 1,
      "tWeekTue": 1,
      "tWeekWed": 1,
      "tWeekThu": 1,
      "tWeekFri": 1,
      "tWeekSat": 1,
      "tIsAppHoliday": 1
    },
    {
      "groupId": 5,
      "tWeekSun": 0,
      "tWeekMon": 0,
      "tWeekTue": 0,
      "tWeekWed": 0,
      "tWeekThu": 0,
      "tWeekFri": 0,
      "tWeekSat": 0,
      "tIsAppHoliday": 0
    },
    {
      "groupId": 6,
      "tWeekSun": 1,
      "tWeekMon": 1,
      "tWeekTue": 1,
      "tWeekWed": 1,
      "tWeekThu": 1,
      "tWeekFri": 1,
      "tWeekSat": 1,
      "tIsAppHoliday": 1
    },
    {
      "groupId": 7,
      "tWeekSun": 1,
      "tWeekMon": 1,
      "tWeekTue": 1,
      "tWeekWed": 1,
      "tWeekThu": 1,
      "tWeekFri": 1,
      "tWeekSat": 1,
      "tIsAppHoliday": 1
    },
    {
      "groupId": 8,
      "tWeekSun": 1,
      "tWeekMon": 1,
      "tWeekTue": 1,
      "tWeekWed": 1,
      "tWeekThu": 1,
      "tWeekFri": 1,
      "tWeekSat": 1,
      "tIsAppHoliday": 1
    },
    {
      "groupId": 9,
      "tWeekSun": 1,
      "tWeekMon": 1,
      "tWeekTue": 1,
      "tWeekWed": 1,
      "tWeekThu": 1,
      "tWeekFri": 1,
      "tWeekSat": 1,
      "tIsAppHoliday": 1
    },
    {
      "groupId": 10,
      "tWeekSun": 1,
      "tWeekMon": 1,
      "tWeekTue": 1,
      "tWeekWed": 1,
      "tWeekThu": 1,
      "tWeekFri": 1,
      "tWeekSat": 1,
      "tIsAppHoliday": 1
    }
  ]
}
```

● Return an example

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

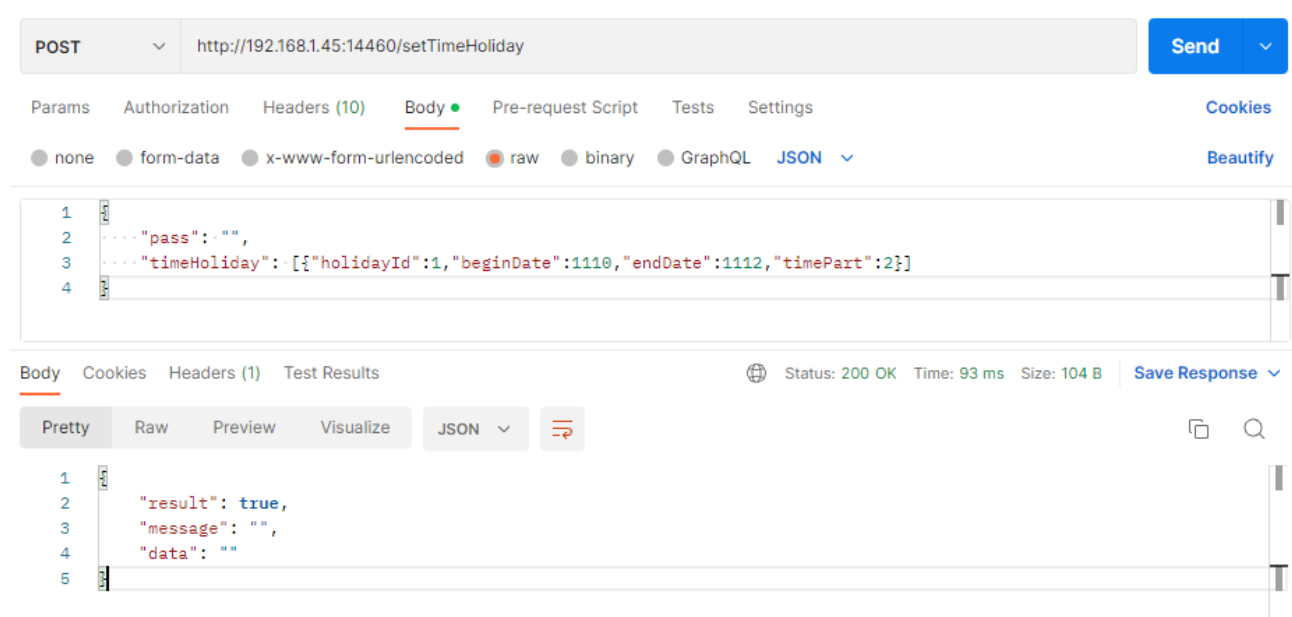
3.17.Issue holidays (up to 20 groups at a time) (setTimeHoliday)- (controller)

Note: up to 50 groups can be customized, 20 groups at a time, can be distributed in 3 times,

3.17.1. Request data

Method		URL			
POST		The http: // Device IP: 14460 / setTimeHoliday			
parameter name		description	type	Must pass	Additional instructions
pass		password	String	Y	
timeHoliday		Holiday collection			A maximum of 20 groups at a time
	holidayId	Holiday number	Int	Y	1~50, the unique holiday number
	beginDate	date commenced	Int	Y	On May 1, the format is: 501
	endDate	deadline	Int	Y	Month day
	timePart	Time period number	Int	Y	Select the time period number

3.17.2. postman instance



3.17.3. instance

● Request example

```
{
  "pass": "",
  "timeHoliday": [{"holidayId": 1, "beginDate": 1110, "endDate": 1112, "timePart": 2}]
}
```

● Return an example

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

3.18. Issue authority (setUserPower) - (controller)

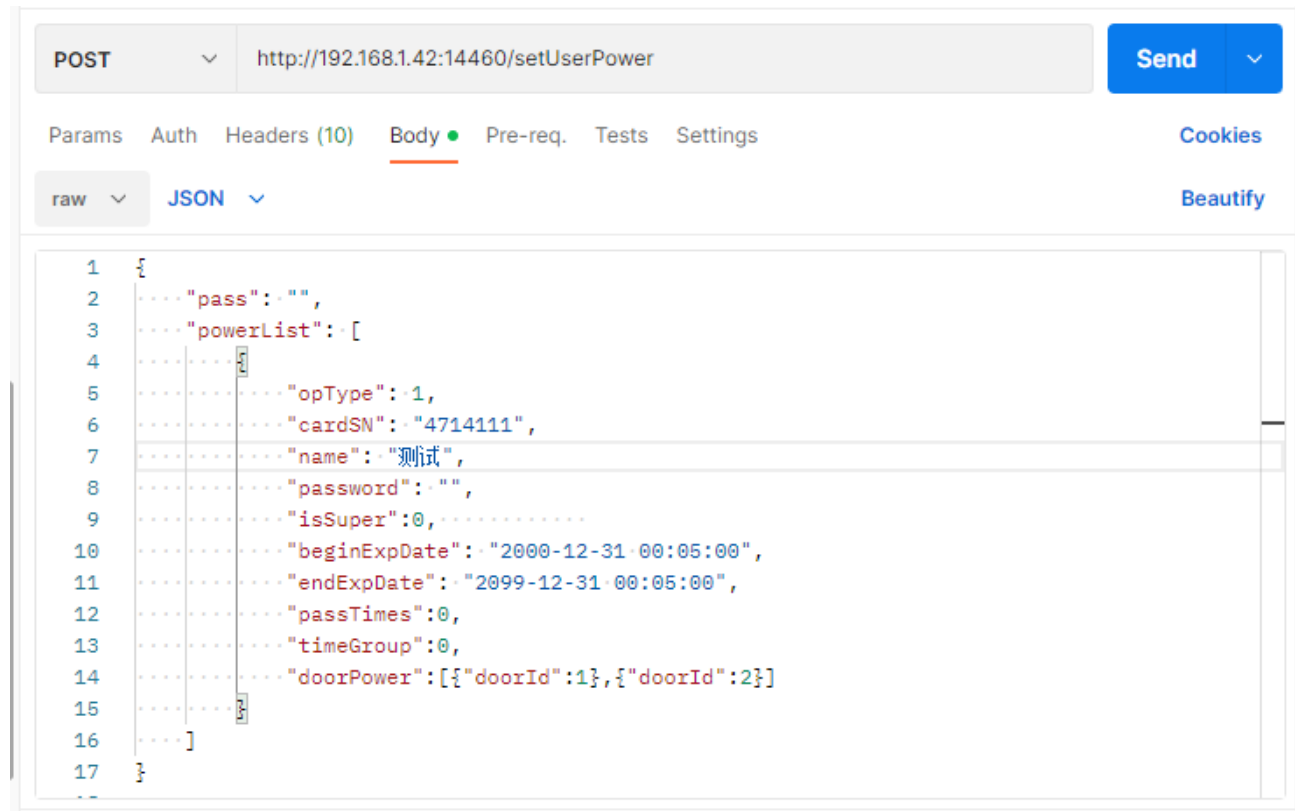
Note: The permission list collection can upload the permission of one person at a time, and the permission is unique with the "card number (cardSN)".

3.18.1. Request data

Method		URL			
POST		The http: // Device IP: 14460 / setUserPower			
parameter name	description	type	Must pass	Additional instructions	
pass	password	String	Y		
powerList	Collection of permissions lists			You can upload one person's permission at a time	
	opType	operation type	Int	Y	1: Upload, 0 to delete
	cardSN	card number	String	Y	
	password	password	String	Y	6 digits, no password, and an empty string
	isSuper	Card type	Int	Y	0: Normal card 1: Super card
	endExpDate	cut-off time	String	Y	For example: 2099-12-31 18:30:00
	beginExpDate	date commenced	String	Y	Year month date time point, the validity period is accurate to the point
	passTimes	The number of credit card	Int	Y	0: An unlimited number of times
	timeGroup	Time group number	Int	Y	
	doorPower	Gate permissions group collection			

	doorId	Door number	Int		The number starts at 1

3.18.2. postman instance



3.18.3. instance

● Request example

```

{
  "pass": "",
  "powerList": [
    {
      "opType": 1,
      "cardSN": "4714111",
      "Name": "Test",
      "password": "",
      "isSuper": 0,
      "beginExpDate": "2000-12-31 00:05:00",

```

```

"endExpDate": "2099-12-31 00:05:00",
"passTimes":0,
"timeGroup":0,
"doorPower":[{"doorId":1},{ "doorId":2}]
}
]
}

```

● Return an example

```

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

```

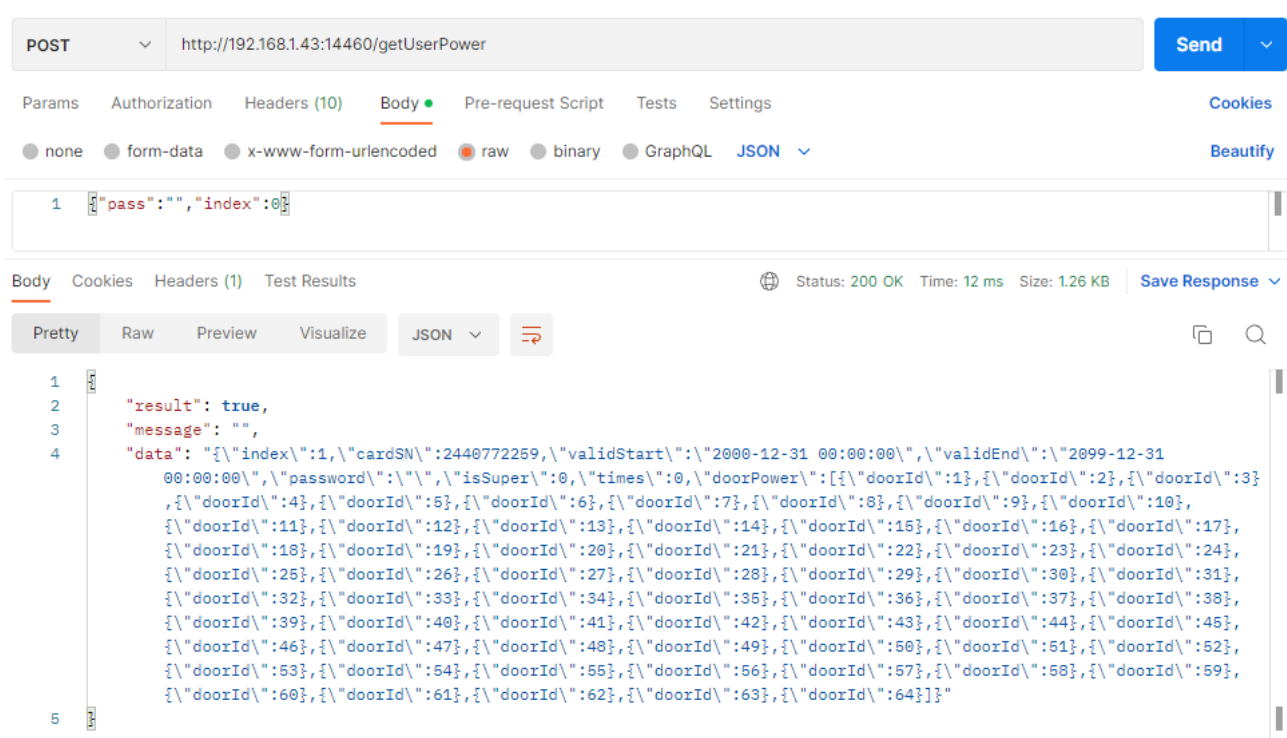
3.19.Get permission (getUserPower) - (controller)

Description: The index starts from 0, and for every successful acquisition, the device will return the new index and return the index to the device until the device appears, complete, and the acquisition is complete

3.19.1. Request data

Method	URL			
POST	The http: // Device IP: 14460 / getUserPower			
parameter name	description	type	Must pass	Additional instructions
pass	password	String	Y	
index	Permission index	Int	Y	The index starts at 0

3.19.2. postman instance



3.19.3. instance

● Request example 1

```
{
  \"pass\": \"\",
  \"index\": 0
}
```

● Return sample 1

```
{
  \"result\": true,
  \"message\": \"\",
  \"data\": [{\"index\": 1, \"cardSN\": 2112217589, \"validStart\": \"2000-00-00 00:00:00\", \"validEnd\": \"2070-04-
03 00:00:00\", \"password\": \"\", \"isSuper\": 0, \"times\": 0, \"timeGroup\": 0, \"doorPower\": [{\"doorId\": 1}, {\"doorId\": 2}, {\"doorId\": 3}
]}]
}
```

validStart: Start time

validEnd: End time

isSuper: 0 normal card

● Request example 2

```
{
  \"pass\": \"\",
```



```
"index": 1
}
```

● Return sample 2

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

3.20.Issue authority (setUserPower) - (face ladder control)

Note: The permission list collection can upload one person at a time, and the permission is unique to "card number (Acc No)".

3.20.1. Request data

Method		URL			
POST		The http: // Device IP: 14460 / setUserPower			
parameter name		description	type	Must pass	Additional instructions
pass		password	String	Y	
powerList		Collection of permissions lists			You can upload one person's permission at a time
	opType	operation type	Int	Y	1: Upload, 0 to delete
	accNo	job number	Int	Y	
	cardSN	card number	String	Y	
	endExpDate	cut-off time	String	Y	For example: 2099-12-31 18:30:00
	beginExpDate	date commenced	String	Y	Year month date time point, the validity period is accurate to the point
	passTimes	The number	Int	Y	0: An unlimited number of

	of credit card			times
timeGroup	Time group number	Int	Y	
faceId1	Do you have photo 1	Int	Y	None: -1, Yes: 0, Permission must include photo
faceImage1	Photo 1	Base64	Y	If there is no, then null. Base64 without labels,
name	Name	String	Y	Within 100k, the facial photo should not be too blurry, with a resolution of 1080 * 720 or below and 240 * 360 or above
password	Personal password	String	Y	Supports 4 Chinese characters
dept	Department Number	Int		6-digit number, pass empty string without password
isManager	Is it an administrator	Int	Y	Department ID (1-50)
fpId1	Do you have fingerprint 1	Int	Y	0: Regular User 1: Administrator (menu accessible)
fpId2	Do you have fingerprints 2	Int	Y	0: No fingerprints 1: There are fingerprints
fp1	Fingerprint 1 data	String	N	0: No fingerprints 1: There are fingerprints
fp2	Fingerprint 2 data	String	N	810 byte Byte array converted to hexadecimal string, obtained by protocol getUserPower
palmId	Is there a palmar vein present	Int	Y	810 byte Byte array converted to hexadecimal string, obtained by protocol getUserPower
palmInfo	Palm vein	Base64	N	None: 0, Yes: 1, Permission

		data			must include palm vein data
	doorPower	Gate permissions group collection			
	doorId	Door number	Int		The number starts at 1

3.20.2. postman instance

POST http://{{IP}}:{{port}}/setUserPower Send

Docs Params Authorization Headers (10) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Schema Beautify

```

1 {
2   "powerList": [
3     {
4       "doorPower": [
5         {
6           "doorId": 1
7         },
8         {
9           "doorId": 2
10        }
11      ],
12      "opType": 1,
13      "cardSN": "4714111",
14      "name": "zero",
15      "beginExpDate": "2000-01-01 00:00:00",
16      "endExpDate": "2026-10-14 23:59:00",
17      "password": "123654",
18      "timeGroup": 0,
19      "accNo": 10,
20      "faceId1": -1,
21      "faceImage1": null,
22      "isManager": 0
23    }
24  ],
25  "pass": ""
26 }

```

Body Cookies Headers (3) Test Results 200 OK • 389 ms • 145 B Save Response

{ JSON Preview Visualize

```

1 {
2   "result": true,
3   "message": "",
4   "data": ""
5 }

```

3.20.3. instance

● Request example

```
{
  "powerList": [
    {
      "doorPower": [
        {
          "doorId": 1
        },
        {
          "doorId": 2
        }
      ],
      "opType": 1,
      "cardSN": "4714111",
      "name": "zero",
      "beginExpDate": "2000-01-01 00:00:00",
      "endExpDate": "2026-10-14 23:59:00",
      "password": "123654",
      "timeGroup": 0,
      "accNo": 10,
      "faceId1": -1,
      "faceImage1": null,
      "isManager": 0
    }
  ],
  "pass": ""
}
```

● Return an example

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

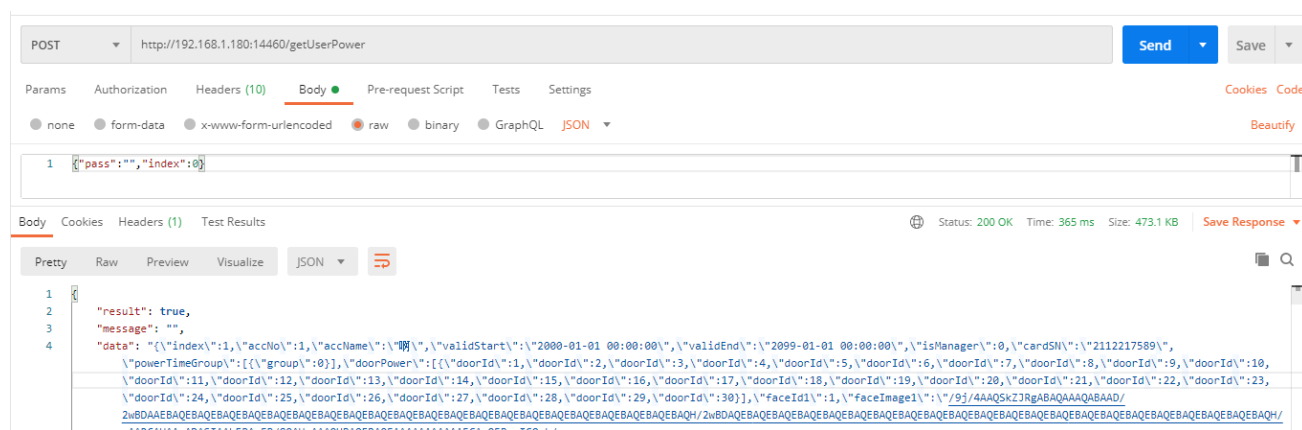
3.21.Access permission (getUserPower) - (face ladder control)

Description: The index starts from 0, and for every successful acquisition, the device will return the new index and return the index to the device until the device appears, complete, and the acquisition is complete

3.21.1. Request data

Method	URL			
POST	The http: // Device IP: 14460 / getUserPower			
parameter name	description	type	Must pass	Additional instructions
pass	password	String	Y	
index	Permission index	Int	Y	The index starts at 0

3.21.2. postman instance



3.21.3. instance

- **Request example 1**

```
{
  "pass": "",
  "index": 0
}
```

- **Return sample 1**

```
{
  "result": true,
  "message": "",
  "data": "{\"index\":1,\"accNo\":1,\"accName\":\"11\",\"validStart\":\"2000-01-01 00:00:00\",\"validEnd\":\"2099-01-01 00:00:00\",\"isManager\":0,\"cardSN\":\"2112217589\",\"powerTimeGroup\":[{\"group\":0}],\"doorPower\":[{\"doorId\":1,\"doorId\":2,\"doorId\":3,\"doorId\":4,\"doorId\":5,\"doorId\":6,\"doorId\":7,\"doorId\":8,\"doorId\":9,\"doorId\":10,\"doorId\":11,\"doorId\":12,\"doorId\":13,\"doorId\":14,
```

```
\\"doorId\\":15,\\"doorId\\":16,\\"doorId\\":17,\\"doorId\\":18,\\"doorId\\":19,\\"doorId\\":20,\\"doorId\\":21,
\\"doorId\\":22,\\"doorId\\":23,\\"doorId\\":24,\\"doorId\\":25,\\"doorId\\":26,\\"doorId\\":27,\\"doorId\\":28,
\\"doorId\\":29,\\"doorId\\":30}],\\"faceId1\\":1,\\"faceImage1\\":\\"/9j/省略...2Q==\\""} }
```

accName: Name

validStart: Start time

validEnd: End time

group: Time group

isManager: Administrator

FaceId1: Does it contain a face? 0: None: Yes

FpId1: Does it contain fingerprints? 0: None: Yes

PalmID: Does it contain palmar vein 0: None 1: Yes

● Request example 2

```
{
  "pass": "",
  "index": 1
}
```

● Return sample 2

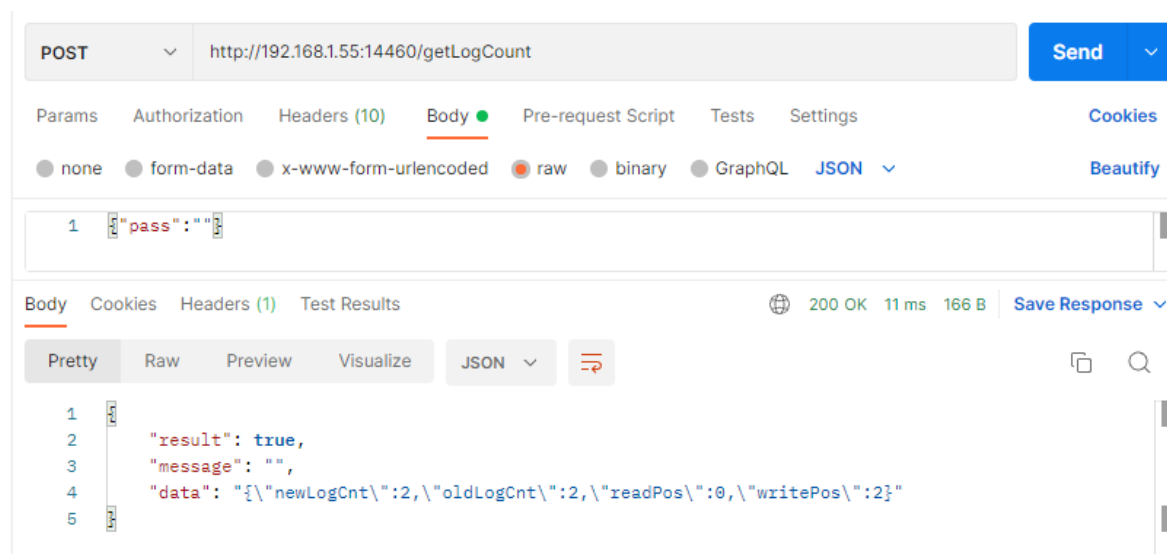
```
{
  "result": true,
  "message": "complete"
}
```

3.22.Number of acquired records (getLogCount)

3.22.1. Request data

Metho	URL			
POST	The http: // Device IP: 14460 / getLogCount			
parameter name	description	type	Must	Additional instructions
pass	password	String	Y	

3.22.2. postman instance



3.22.3. instance

● Request example

```
{  
  "pass": ""  
}
```

● Return an example

```
{  
  "result": true,  
  "message": "",  
  "data": { "newLogCnt": 2, "oldLogCnt": 2, "readPos": 0, "writePos": 2 }  
}
```

pour:

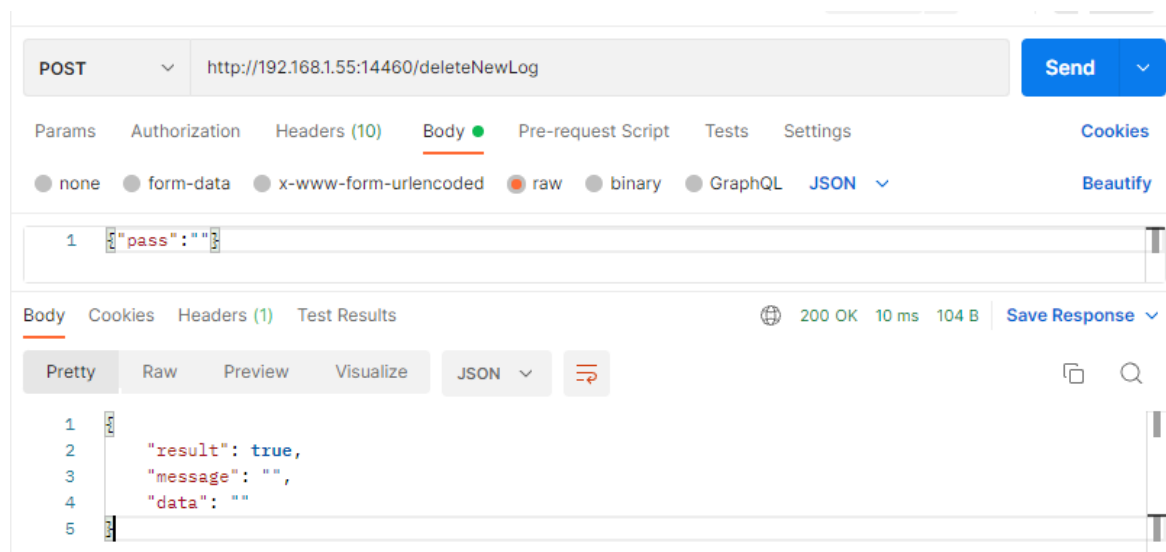
newLogCnt: // Number of new records
oldLogCnt And // Number of historical records

3.23. Get Record Content (getLogInfo)

3.23.1. Request data

Method	URL			
POST	The http: // Device IP: 14460 / getLogInfo			
parameter name	description	type	Must	Additional instructions

3.24.2. postman instance



3.24.3. instance

● Request example

```
{"pass": ""}
```

● Return an example

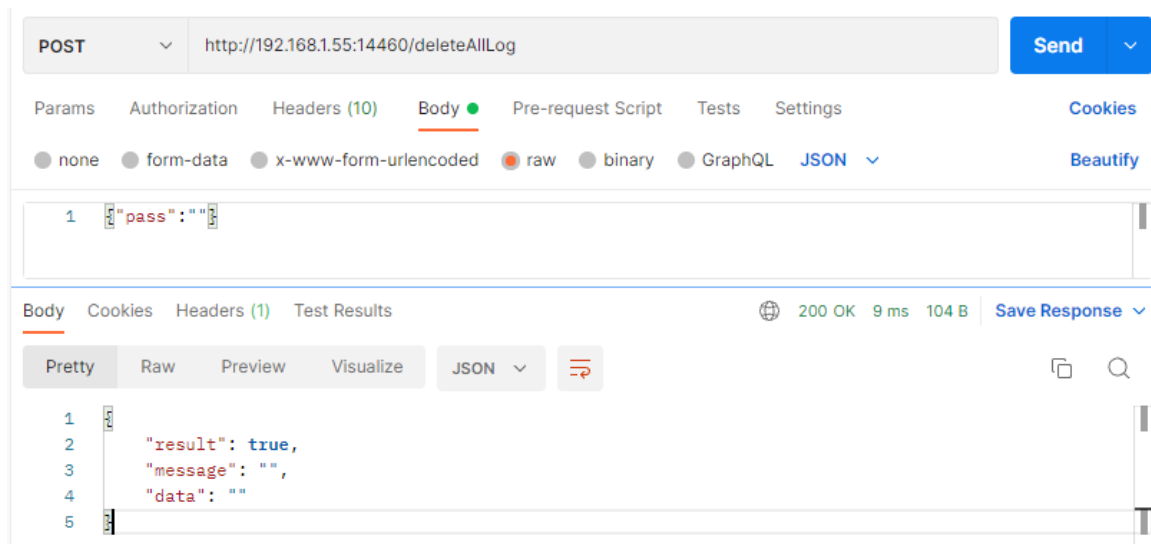
```
{
  "result": true,
  "message": "",
  "data": ""
}
```

3.25.Empty Equipment Record (deleteAllLog)

3.25.1. Request data

Method	URL			
POST	The http: // Device IP: 14460 / deleteAllLog			
parameter name	description	type	Must	Additional instructions
pass	password	String	Y	

3.25.2. postman instance



3.25.3. instance

● Request example

```
{  
  "pass": ""  
}
```

● Return an example

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

3.26. Set up the device active upload record (setLogCallback)

pour:

1. After the callback is set up successfully, you need to monitor the callback server ip, the device gets the upload record in real time, and the software replies to the device json string '{" result ": 1}', and then the device will pass the next one. (All data should be replied to)

2. Examples of recorded messages

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
```

```
{"deviceKey":"DF6224041B5C1C21","havLog":0,"lockStatus":[{"doorId":1,"status":1,"doorStatus":0}]}
```

Software reply:

```
HTTP/1.1 200 Http Server OK
Content-Type: application/json; charset=utf-8
Content-Length: 121
Connection: close
```

```
{"result":1}
```

Part of the code for the c # software reply can refer to:

```
THeader[] headers =
{
// Please refer to the http protocol here
new THeader() { Name= "Content-Type", Value = "application/json; charset=utf-8" },
new THeader() { Name= "Content-Length", Value = result.Length.ToString() },
};

IDictionary<string, int> parameters = new Dictionary<string, int>();
parameters.Add("result", 1);
string postData = JsonConvert.SerializeObject(parameters);
result = System.Text.Encoding.UTF8.GetBytes(postData);

server.SendResponse(connId, HttpStatusCode.Ok, "Http Server OK", headers, result, result.Length);
server.Release(connId);
```

3. The uploaded record is the record after the callback is successfully set. The record before the callback setting needs to be collected from the records

3.26.1. Request data

Method	URL			
POST	And http: // Device IP: 14460 / setLogCallBack			
parameter name	description	type	Must pass	Additional instructions
pass	password	String	Y	
logCallBackURL	Service IP where the device proactively	String	Y	E. g.: "http: // 192.168.31.179:13333 / expog" Set "" cancels the real-time upload records


```
"doorStatus":0}, {"doorId":2, "doorStatus":0}, {"doorId":3, "doorStatus":0}, {"doorId":4, "doorStatus":0}, {"doorId":5,
"doorStatus":0}, {"doorId":6, "doorStatus":0}, {"doorId":7, "doorStatus":0}, {"doorId":8, "doorStatus":0}, {"doorId":9,
"doorStatus":0}, {"doorId":10, "doorStatus":0}, {"doorId":11, "doorStatus":0}, {"doorId":12, "doorStatus":0}, {"doorId":13,
"doorStatus":0}, {"doorId":14, "doorStatus":0}, {"doorId":15, "doorStatus":0}, {"doorId":16, "doorStatus":0}, {"doorId":17,
"doorStatus":0}, {"doorId":18, "doorStatus":0}, {"doorId":19, "doorStatus":0}, {"doorId":20, "doorStatus":0}, {"doorId":21,
"doorStatus":0}, {"doorId":22, "doorStatus":0}, {"doorId":23, "doorStatus":0},
{"doorId":24, "doorStatus":0}, {"doorId":25, "doorStatus":0}, {"doorId":26, "doorStatus":0}, {"doorId":27, "doorStatus":0}, {"d
oorId":28, "doorStatus":0}, {"doorId":29, "doorStatus":0}, {"doorId":30, "doorStatus":0}, {"doorId":31, "doorStatus":0}, {"doorI
d":32, "doorStatus":0}, {"doorId":33, "doorStatus":0}, {"doorId":34, "doorStatus":0}, {"doorId":35, "doorStatus":0}, {"doorId":3
6, "doorStatus":0}, {"doorId":37, "doorStatus":0}, {"doorId":38, "doorStatus":0}, {"doorId":39, "doorStatus":0}, {"doorId":40, "d
oorStatus":0}, {"doorId":41, "doorStatus":0}, {"doorId":42, "doorStatus":0}, {"doorId":43, "doorStatus":0}, {"doorId":44, "door
Status":0}, {"doorId":45, "doorStatus":0}, {"doorId":46, "doorStatus":0}, {"doorId":47, "doorStatus":0}, {"doorId":48, "doorStat
us":0}, {"doorId":49, "doorStatus":0}, {"doorId":50, "doorStatus":0}, {"doorId":51, "doorStatus":0}, {"doorId":52, "doorStatus":
0}, {"doorId":53, "doorStatus":0}, {"doorId":54, "doorStatus":0}, {"doorId":55, "doorStatus":0}, {"doorId":56, "doorStatus":0}, {"
doorId":57, "doorStatus":0}, {"doorId":58, "doorStatus":0}, {"doorId":5
```

2. Face ladder control device heartbeat 30s once

```
{"deviceKey":"FFFF9AD90718BAF9","firmWareVer":2024072901,"regPowCnt":5}
```

3. Face ladder control equipment record

```
{"deviceKey":"FFFF9AD90718BAF9","havLog":1,"reqSeqNo":"172223233362072900","logType":0,"accNo":0,"visitorId"
:0,"passTime":"2024-07-29
13:52:13","operation":100,"alarmCode":0,"passStatus":0,"doorId":1,"readId":1,"confidence":0,"temperature":0,"lockStatus":
[{"doorId":1,"status":0}]}
```

4. AF

4.1. Set the timing door opening (setTimingControl)- (controller)

Note: To enable this function, set the doorWorkWay parameter in the setConfig interface, but not reset the doorWorkWay parameter in the setConfig interface

4.1.1. Request data

Method	URL
--------	-----

POST		The http: // Device IP: 14460 / setTimingControl			
parameter name		description	type	Must pass	Additional instructions
pass		password	String	Y	
timingInfo					
	doorId	Door number	Int	Y	
	timeGroup	Time group number	Int	Y	

4.1.2. postman instance

POST

http://192.168.1.58:14460/setTimingControl

Send

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Cookies

Beautiful

```

1 {
2   "pass": "",
3   "timingInfo": [
4     {
5       "doorId": 1,
6       "timeGroup": 1
7     }
8   ]
9 }

```

Body

Cookies

Headers (1)

Test Results

Status: 200 OK

Time: 81 ms

Size: 104 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```

1 {
2   "result": true,
3   "message": "",
4   "data": ""
5 }

```


5. UDP protocol

5.1. Interface universal 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 privateString 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.

5.2. Search Device (UDP Broadcast) (1001)

Note: The search did not return, you can check whether there are multiple network cards or whether there is a closed firewall

5.2.1. parameter declaration

Instruction: 1001 Description: UDP sends broadcast data (255.255.255.255,14440)	
required parameter	
{ "Pass": "", / / Udp communication password, 6 digits, the factory password is "" "appCode": 1001 / / instruction identification }	
Response parameters	

```

{
"result": true,
"message": "",
"data": {
    "deviceKey": "5024668358A3BF1C", // Serial number
    "deviceType": 61, // Device type, reference
    "firmWareVer": "2021082501", // version number, and the characters include
letters or numbers
    "doorType": 64, // floors
    "termId": 1, // equipment number
    "onlineWay": 0, // Network connection mode
    "enableDHCP": 0, // Whether the Dynamic Acquisition IP is enabled
    "localIp": "192.168.1.49", // ip address
    "subAdd": "255.255.255.0", // The subnet mask
    "defaGate": "192.168.1.1", // Gateway
    "serverIp": "192.168.1.160", // server IP
    "serverPort": 14440 // Server side port
    "lableSN": 5000004, // Label serial number
    "deviceModel": "JH 51", // model number
    "Oem": "FFFEFD FCFBFAF9F8F7F6F5F4F3F2F1F0" // oem, by
configuring the OEM settings
}
}

```

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

5.3.1. 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, // Auto get IP 0 is not enabled, 1 enabled "localIp": "192.168.1.200", "subAdd": "255.255.255.0", "defaGate": "192.168.1.1" "termId": 1 // equipment number } </pre>
Response parameters
<pre>{"result":true,"message":"","data":""}</pre>

5.4. Configuration of OEM (UDP Broadcast) (1010)

5.4.1. parameter declaration

Instruction: 1010 Description: UDP sends broadcast data (255.255.255.255,14440))
required parameter
<pre> { "pass": "", "deviceKey": "5024668358A3BF1C", // Serial number "appCode": 1010, // instruction identification "oemPassword": "FFFEFD FCFBFAF9F8F7F6F5F4F3F2F1F0" // OEM password, hex number, 32 characters } </pre>
Response parameters
<pre>{"result":true,"message":"","data":""}</pre>

6. type definition

6.1. Validation Type-Time-period validation type

Value (decimal)	meaning
0	block
20	All certified or

Value (decimal)	Implications (face ladder control)
15	Face+fingerprint
16	Password+face/fingerprint/palm print
17	Card+Face/Fingerprint/Palmprint
20	All certifications or

6.2. Record alarm code

Value (decimal)	meaning
0	normal
80	Illegal card
83	fire alarm
89	Blacklist alarm
90	Has expired
92	Has passed the effective number
93	Read header data check error
95	Alarm

Value (decimal)	Implications (face ladder control)
0	normal

80	Illegal card
83	fire alarm
90	Has expired
95	Alarm
91	stranger

6.3. Record the mode of passage

Value (decimal)	Implications (face ladder control)
0	card
1	fingerprint
2	face
3	Remote user opening the door (mobile end)
4	Temporary password to open the door
6	Personal password
8	Super Card Open Door
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 print+card
23	Palm print+fingerprint
24	Palmprint+Face
25	Palm print+password
41	Insufficient card swiping interval
42	The door is always closed and cannot pass through
43	Not open during the designated time period

46	Holiday period
97	Open the ladder manually
100	Remote door opening
101	Button to open the door
102	Timed door opening
104	Emergency door opening
105	Emergency door closing
108	Restore online

6.4. device type

Value (decimal)	meaning
61	electric lift controller
63	Face ladder control

6.5. Control door type

Value (decimal)	meaning
64	number of floor levels

6.6. Door control mode

Value (decimal)	meaning
0	online
1	normally open
2	normal close