

Catalogue

1. Version introduction	4
2. Protocol description	4
2.1. Http protocol, Post mode, UTF-8 encoding, device port: 14460.	4
2.2. General return description of Interface:	5
2.3. Operation process:	5
2.4. Matters needing attention.	5
3. Http interface call (method: POST-utf-8)	6
3.1. Set the device communication password (setPassword)	6
3.1.1. Request data	6
3.1.2. Postman example	7
3.1.3. Example	7
3.2. Get device serial number (getDeviceKey)	8
3.2.1. Request data	8
3.2.2. Postman example	8
3.2.3. Example	8
3.3. Get device OEM (getOEM)	9
3.3.1. Request data	9
3.3.2. Postman example	9
3.3.3. Example	9
3.4. Get device time (getDeviceTime)	10
3.4.1. Request data	10
3.4.2. Postman example	10
3.4.3. Example	10
3.5. Set Time (setTime)	11
3.5.1. Request data	11
3.5.2. Postman example	11
3.5.3. Example	11
3.6. Get device information (getDeviceInfo)	12
3.6.1. Request data	12
3.6.2. Postman example	12
3.6.3. Example	12
3.7. Set device parameters (setConfig)	13
3.7.1. Request data	13
3.7.2. Postman example	16
3.7.3. Example	16
3.8. Get device parameters (getConfig)	17
3.8.1. Request data	17
3.8.2. Postman example	18
3.8.3. Example	18
3.9. Restart Device (restartDevice)	19

3.9.1. Request data	19
3.9.2. Postman example	19
3.9.3. Example	19
3.10. Reset Device (resetDevice)	19
3.10.1. Request data	19
3.10.2. Postman example	20
3.10.3. Example	20
3.11. Remote door opening (openDoorControl).....	20
3.11.1. Request data	20
3.11.2. Postman example	21
3.11.3. Example	21
3.12. Emergency switch/locked door (setEmergencyControl).....	21
3.12.1. Request data	21
3.12.2. Postman example	22
3.12.3. Example	22
3.13. Set time part (at most 7 groups at a time) (setTimePart).....	23
3.13.1. Request data	23
3.13.2. Postman example	24
3.13.3. Example	24
3.14. Get time part (get 5 groups at a time) (getTimePart).....	24
3.14.1. Request data	24
3.14.2. Postman example	25
3.14.3. Example	25
3.15. Set time group (at most 9 groups at a time) (setTimeGroup)	26
3.15.1. Request data	26
3.15.2. Postman example	27
3.15.3. Example	28
3.16. Get time groups (get 9 groups at a time) (getTimeGroup)	30
3.16.1. Request data	30
3.16.2. Postman example	31
3.16.3. Example	31
3.17. Set user power (setUserPower)	32
3.17.1. Request data	32
3.17.2. Postman example	34
3.17.3. Example	34
3.18. Get power (getUserPower)	34
3.18.1. Request data	35
3.18.2. Postman example	35
3.18.3. Example	35
3.19. Clear power (deleteAllUserPower)	36
3.19.1. Request data	36
3.19.2. Postman example	37
3.19.3. Example	37
3.20. Get Number of logs (getLogCount).....	37

3.20.1. Request data	37
3.20.2. Postman example	38
3.20.3. Example	38
3.21. Get log information (getLogInfo)	38
3.21.1. Request data	38
3.21.2. Postman example	39
3.21.3. Example	39
3.22. Delete the latest log (deleteNewLog)	40
3.22.1. Request data	40
3.22.2. Postman example	40
3.22.3. Example	40
3.23. Clear the device log (deleteAllLog).....	41
3.23.1. Request data	41
3.23.2. Postman example	41
3.23.3. Example	41
3.24. Set the device active uploading log (setLogCallBack)	42
3.24.1. Request data	42
3.24.2. Postman example	42
3.24.3. Example	43
3.25. Get the active upload log server address (getLogCallBack)	43
3.25.1. Request data	43
3.25.2. Postman example	44
3.25.3. Example	44
3.26. Issuing department (setDepartment).....	44
3.26.1. Request data.....	44
3.26.2. Postman example.....	45
3.26.3. Example.....	45
3.27. Obtain department (getDepartment).....	46
3.27.1. Request data.....	46
3.27.2. Postman example.....	46
3.27.3. Example.....	47
4. UDP protocol	47
4.1. General return description of Interface	47
4.2. Search device (UDP broadcast) (1001)	47
4.2.1. Parameter description	47
4.3. Configure IP (UDP broadcast) (1002).....	48
4.3.1. Parameter description	48
4.4. Configure OEM (UDP broadcast) (1010)	49
4.4.1. Parameter description	49
4.5. Configure device model (UDP broadcast) (1014).....	50
4.5.1. Parameter description	50
5. Type definition	50
5.1. Verification type - time part verification type	50
5.2. log alarm code	50

5.3. log the way of passage.....	51
5.4. Device type	53
5.5. Door control type.....	53
5.6. Door control mode.....	53

1. Version introduction

Date	Modify content	Firmware Version
12/17/2024	First edition	
08/22/2025	1. Modify device parameter protocol 2. Modify the readId: attendance status value in the record	2025082200

2. Protocol description

2.1. Http protocol, Post mode, UTF-8 encoding, device port: 14460

Root address of interface: http://device ip address: 14460/

Interface form: Provide external services through HTTP requests.

Interface security: The device password (pass) needs to be set for the first call of the interface, and the device password (pass) needs to be passed in as the interface security verification key for the subsequent call of any interface.

Example description of interface: All examples in this chapter use postman,

a) Put POST request parameters in body

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

2.2. General return description of Interface:

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

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

2.3. Operation process:

Use the device for the first time: Enter the main identification interface after the device is powered on. **The device has no initial password.**

2.4. Matters needing attention

- When calling the device interface, do not call the interface of the same device on other client servers at the same time.
- If the calling interface returns the prompt "Parameter exception", please check as follows:
 - 1) Check whether the parameter name is misspelled and whether there are spaces or ENTER in the parameter name
 - 2) Check whether the parameter value conforms to the specification, such as the Int value is undefined, the id contains illegal characters other than alphanumeric characters, etc

- 3) Whether there are spaces in Json (the Json string returned by the interface needs to be removed), and whether there are more or less ". Please carefully check the postman example of each interface
- If the calling interface returns null, it means that the url is incorrect (IP error, spelling error, field omission, etc.)

3. Http interface call (method: POST-utf-8)

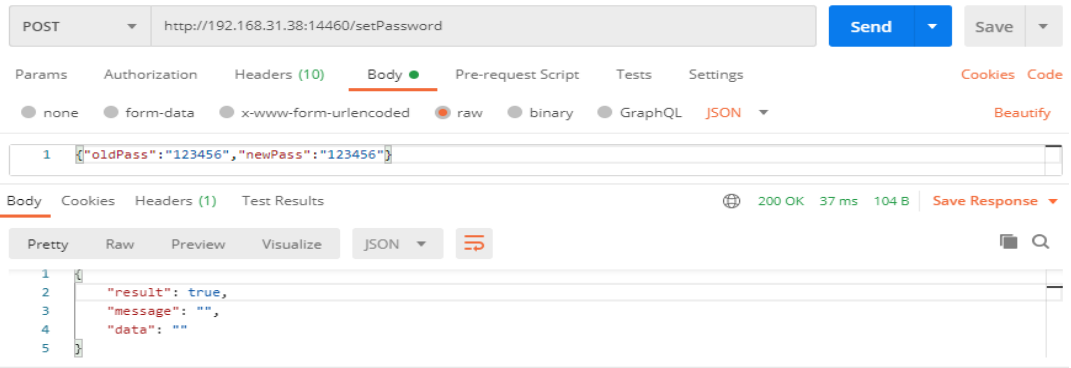
3.1. Set the device communication password (setPassword)

3.1.1. Request data

Method	URL			
POST	http://Device IP:14460/setPassword			
Parameter name	Describe	Type	Mandatory	Description
oldPass	Old password	String	Y	The new device or the device after reset (restore initialization), oldPass ". When changing the password, the new and old passwords are passed in
newPass	New password	String	Y	

				respectively. This interface call does not need to pass in Password length: 8 digits or less Format: letters or numbers
--	--	--	--	---

3.1.2. Postman example



3.1.3. Example

● Request example

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

● Return to example

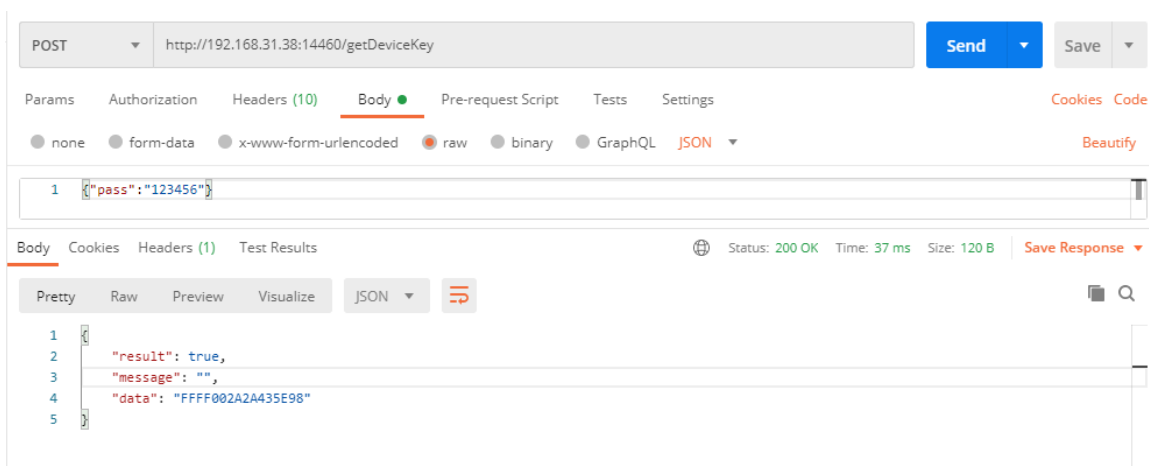
```
{  
  "result": true,  
  "message": "",  
  "data": "5024668308708C1C"  
}
```

3.2. Get device serial number (getDeviceKey)

3.2.1. Request data

Method	URL			
POST	http://Device IP:14460/getDeviceKey			
Parameter name	Describe	Type	Must	Additional notes
pass	Password	String	N	This parameter can not be passed

3.2.2. Postman example



3.2.3. Example

● Request example

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

● Return to example

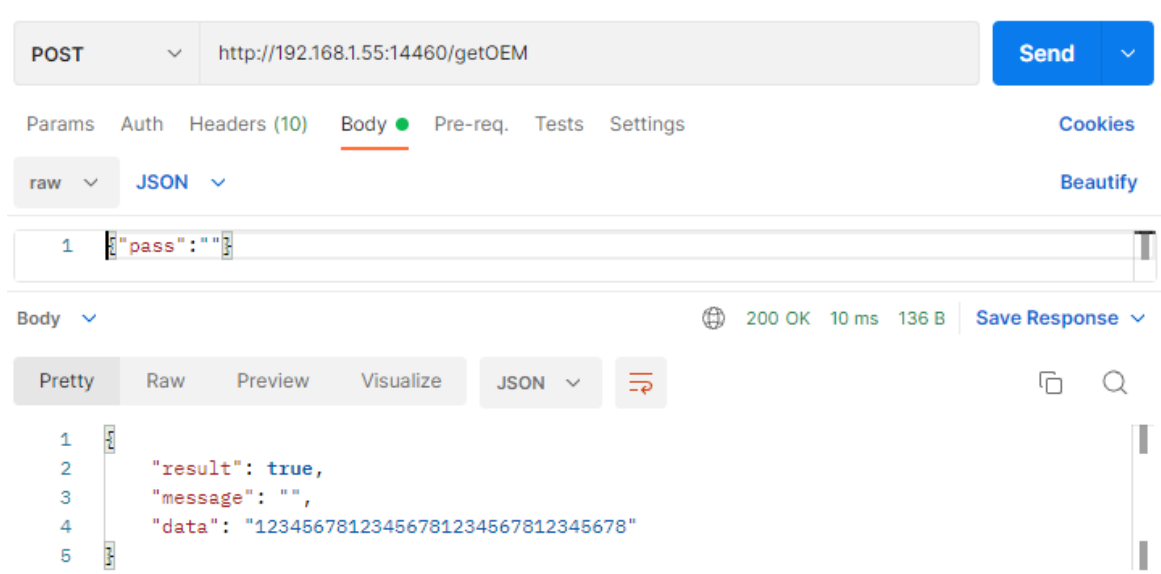
```
{  
  "result": true,  
  "message": "",  
  "data": "5024668308708C1C"  
}
```


3.3. Get device OEM (getOEM)

3.3.1. Request data

Method	URL			
POST	http://Device IP:14460/getOEM			
Parameter name	Describe	Type	Must send	Additional notes
pass	Pass word	String	Y	

3.3.2. Postman example



3.3.3. Example

● Request example

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

● Return to example

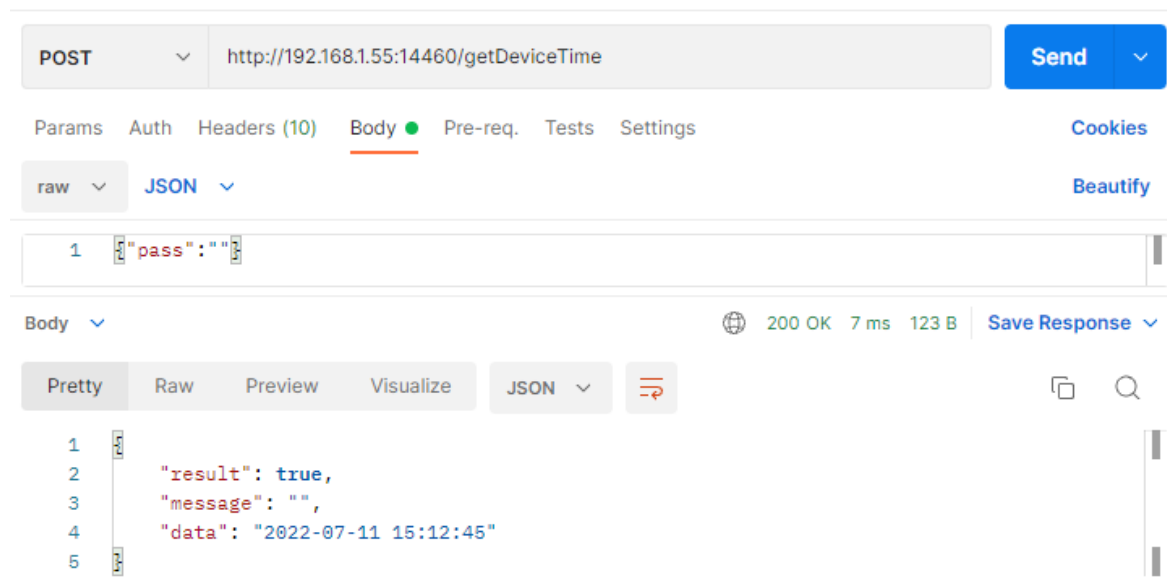
```
{
  "result": true,
  "message": "",
  "data": "12345678123456781234567812345678"
}
```

3.4. Get device time (getDeviceTime)

3.4.1. Request data

Method	URL			
POST	http://Device IP:14460/getDeviceTime			
Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	

3.4.2. Postman example



3.4.3. Example

● Request example

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

● Return to example

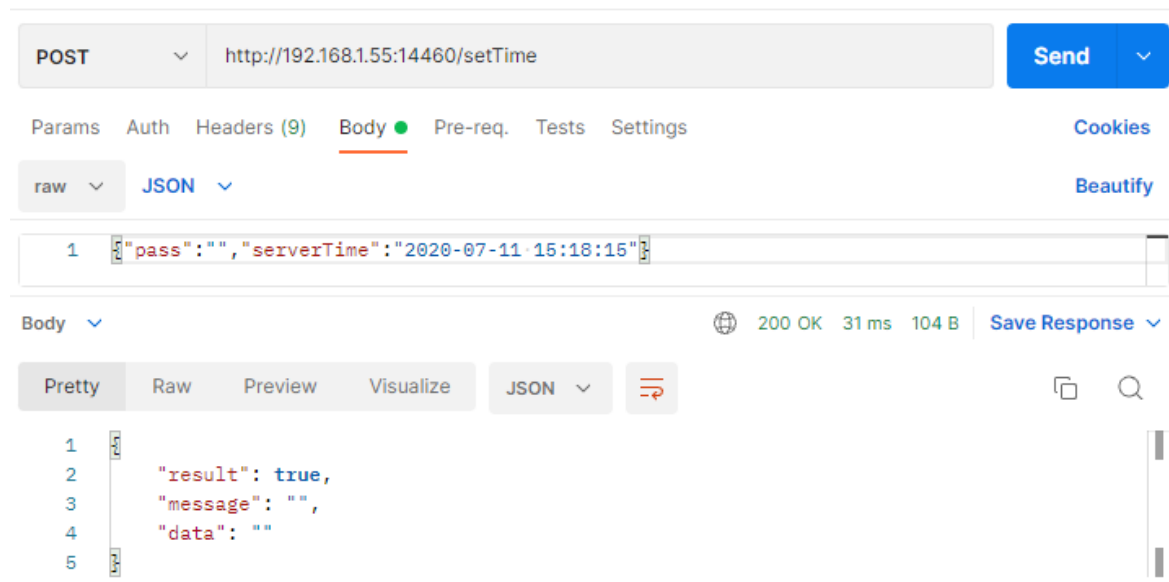
```
{
  "result": true,
  "message": "",
  "data": "2022-07-11 15:12:45"
}
```

3.5. Set Time (setTime)

3.5.1. Request data

Method	URL			
POST	http://Device IP:14460/setTime			
Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	
serverTime	Password	String	Y	Time format: yyyy-MM-dd HH: mm: ss

3.5.2. Postman example



3.5.3. Example

● Request example

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

● Return to example

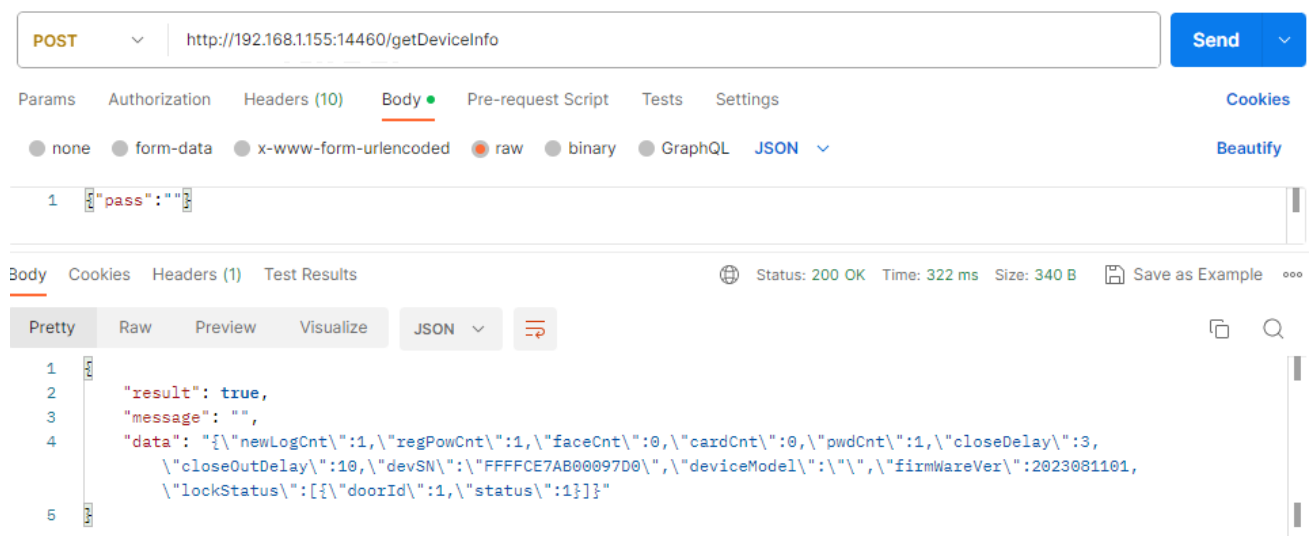
```
{
  "result": true,
  "message": "",
  "data": ""
}
```

3.6. Get device information (getDeviceInfo)

3.6.1. Request data

Method	URL			
POST	http://Device IP:14460/getDeviceInfo			
Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	

3.6.2. Postman example



3.6.3.Example

● Request example

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

● Return to example

```
{
  "result": true,
  "message": "",
  "data":
  "{ \"newLogCnt\": 0, \"regPowCnt\": 1, \"mgrCnt\": 0, \"faceCnt\": 0, \"cardCnt\": 1, \"pwdCnt\": 1, \"fpCnt\": 2, \"palmCnt\": 1, \"FPTType\": 6, \"closeDelay\": 5, \"devSN\": \"FFFF8E1CC9590074\", \"firmWareVer\": \"2025081900\" }"
}
```

Description:

```

{"newLogCnt\:0,//records
"regPowCnt\:1,// permissions
"mgrCnt\:0,// Number of administrators
"faceCnt\:0,// face
"cardCnt\:1,//Cards
"pwdCnt\:1,//Number of passwords
"fpCnt\:2,//Number of fingerprints
"palmCnt\:1,//Number of palmar veins
"FPTYPE\:6,//Fingerprint type
"closeDelay\:5,//Unit seconds
"devSN\":"FFFF8E1CC9590074\","//SN
"firmWareVer\":"2025081900\","//version number
}

```

3.7. Set device parameters (setConfig)

3.7.1. Request data

Method	URL			
POST	http://Device IP:14460/setConfig			
Parameter name	Describe	Type	Must send	Additional notes
pass	password	String	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
isCheckMask	Enable masks	int	N	0: Not enabled, 1: Enable default: 0
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
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"

3.7.2. Postman example

The screenshot shows a Postman interface for a POST request to `http://{{IP}}:{{port}}/setConfig`. The request body is a JSON object containing various configuration parameters. The response is a 200 OK status with a JSON body indicating success.

Request:

```
POST http://{{IP}}:{{port}}/setConfig
```

Body (JSON):

```
{
  "pass": "",
  "alarmDelay": 1,
  "firealarmSwitch": 0,
  "isCheckMask": 0,
  "wgOutputEnable": 1,
  "wgOutputModel": 1,
  "wgOutputFormat": 0,
  "repeatOpenTime": 1,
  "faceRepeatEnable": 1,
  "faceThreshold": 50,
  "faceLiving": 45,
  "faceDistance": 0,
  "Volume": 8,
  "Language": 1,
  "displayStandbyTime": 60,
  "lightEnable": 1,
  "lightTime": [
    {
      "beginTime": 1730,
      "endTime": 730
    }
  ],
  "doorParam": [
    {
      "doorWorkday": 0,
      "appKeyButton": 1,
      "timeGroup": 0,
      "doorMultiUsers": 1,
      "closeDelay": 5
    }
  ],
  "ioModeType": 1,
  "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
    }
  ],
  "dstEnable": 1,
  "dstStart": "05/22 08:01",
  "dstEnd": "06/22 08:01"
}
```

Response:

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

3.7.3. Example

● Request example


```
{
  "pass": "",
  "alarmDelay": 1,
  "firealarmSwitch": 0,
  "isCheckedMask": 0,
  "wgOutputEnable": 1,
  "wgOutputModel": 1,
  "wgOutputFormat": 0,
  "repeatOpenTime": 1,
  "faceRepeatEnable": 1,
  "faceThreshold": 50,
  "faceLiving": 45,
  "faceDistance": 0,
  "Volume": 8,
  "Language": 1,
  "displayStandbyTime": 60,
  "lightEnable": 1,
  "lightTime": [
    {
      "beginTime": 1730,
      "endTime": 730
    }
  ],
  "doorParam": [
    {
      "doorWorkWay": 0,
      "appKeyButton": 1,
      "timeGroup": 0,
      "doorMultiUsers": 1,
      "closeDelay": 5
    }
  ],
  "ioModeType": 1,
  "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
    }
  ],
  "dstEnable": 1,
  "dstStart": "05/22 08:01",
  "dstEnd": "06/22 08:01"
}
```

● Return to example

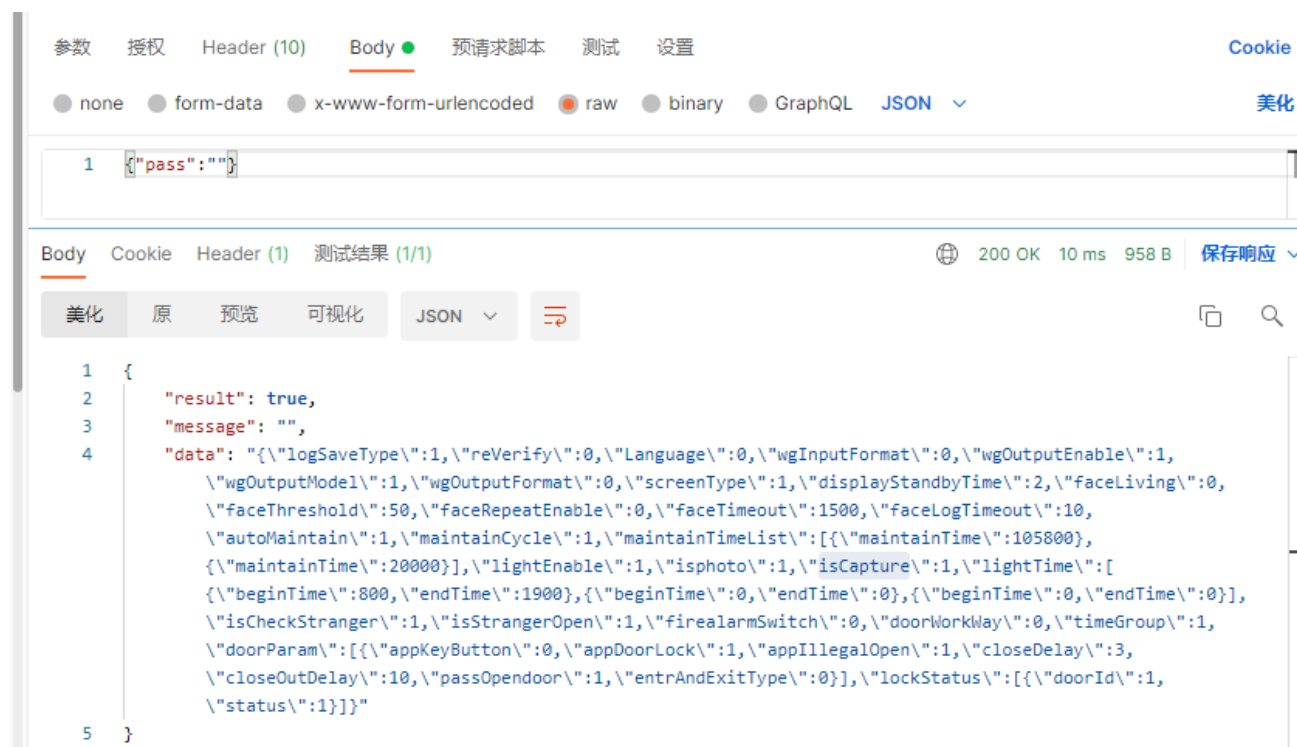
```
{
  "result": true,
  "message": "",
  "data": ""
}
```

3.8. Get device parameters (getConfig)

3.8.1. Request data

Method	URL			
POST	http://Device IP:14460/getConfig			
Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	

3.8.2. Postman example



3.8.3. Example

● Request example

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

● Return to example

```
{
  "result": true,
  "message": "",
  "data": "{\n\"logSaveType\":1,\n\"reVerify\":0,\n\"Language\":0,\n\"wgInputFormat\":0,\n\"wgOutputEnable\n\n\":1,\n\"wgOutputModel\":1,\n\"wgOutputFormat\":0,\n\"screenType\":1,\n\"displayStandbyTime\":2,\n\"faceLivi\n\nng\":0,\n\"faceThreshold\":50,\n\"faceRepeatEnable\":0,\n\"faceTimeout\":1500,\n\"faceLogTimeout\":10,\n\"au\n\n toMaintain\":1,\n\"maintainCycle\":1,\n\"maintainTimeList\":[\n\n    {\n\n      \n\"maintainTime\":105800},\n\n    {\n\n      \n\"maintainTime\n\n \":20000}],\n\n  \n\"lightEnable\":1,\n\"isphoto\":1,\n\"isCapture\":1,\n\"lightTime\":[\n\n    {\n\n      \n\"beginTime\":800,\n\"end\n\n Time\":1900},\n\n    {\n\n      \n\"beginTime\":0,\n\"endTime\":0},\n\n    {\n\n      \n\"beginTime\":0,\n\"endTime\":0}],\n\n  \n\"isCheckStranger\":\n\n 1,\n\"isStrangerOpen\":1,\n\"firealarmSwitch\":0,\n\"doorWorkWay\":0,\n\"timeGroup\":1,\n\"doorParam\":[\n\n    {\n\n      \n\"a\n\n ppKeyButton\":0,\n\"appDoorLock\":1,\n\"appIllegalOpen\":1,\n\"closeDelay\":3,\n\"closeOutDelay\":10,\n\"pas\n\n sOpendoor\":1,\n\"entrAndExitType\":0}],\n\n  \n\"lockStatus\":[\n\n    {\n\n      \n\"doorId\":1,\n\"status\":1}]]}"
```

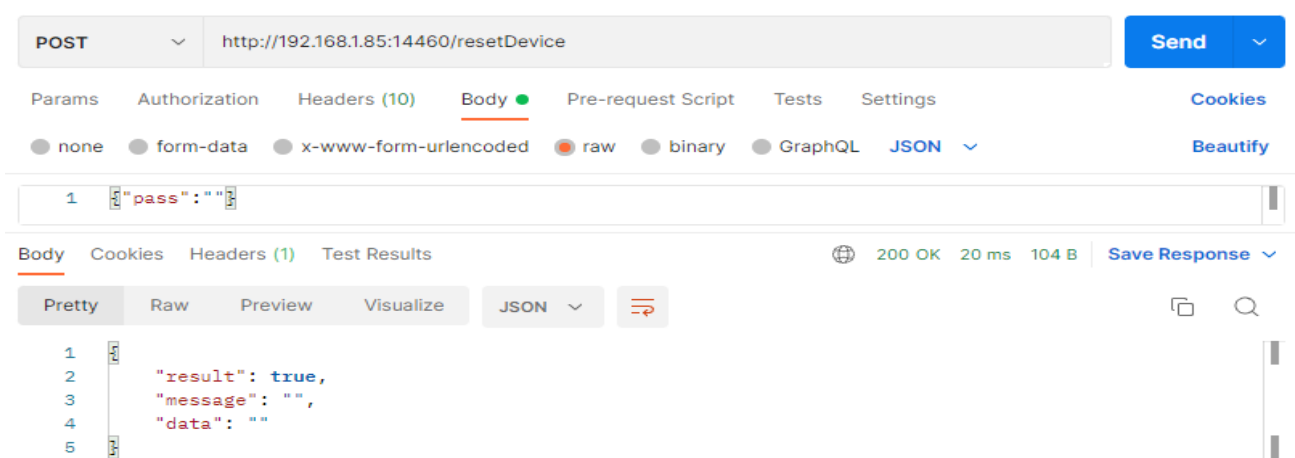
Note: The parameters are the same as setting Device parameters

3.9. Restart Device (restartDevice)

3.9.1. Request data

Method	URL			
POST	http://Device IP:14460/restartDevice			
Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	

3.9.2. Postman example



3.9.3.Example

● Request example

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

● Return to example

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

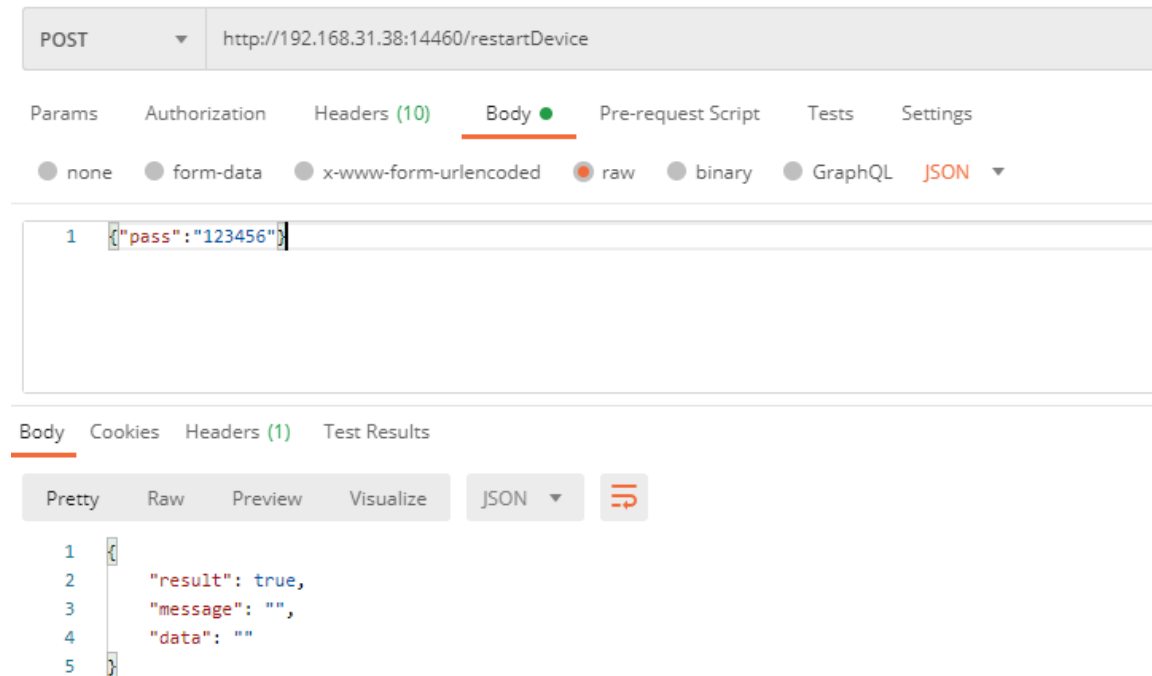
3.10.Reset Device (resetDevice)

3.10.1. Request data

Method	URL
POST	http://Device IP:14460/resetDevice

Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	

3.10.2. Postman example



3.10.3. Example

● Request example

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

● Return to example

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

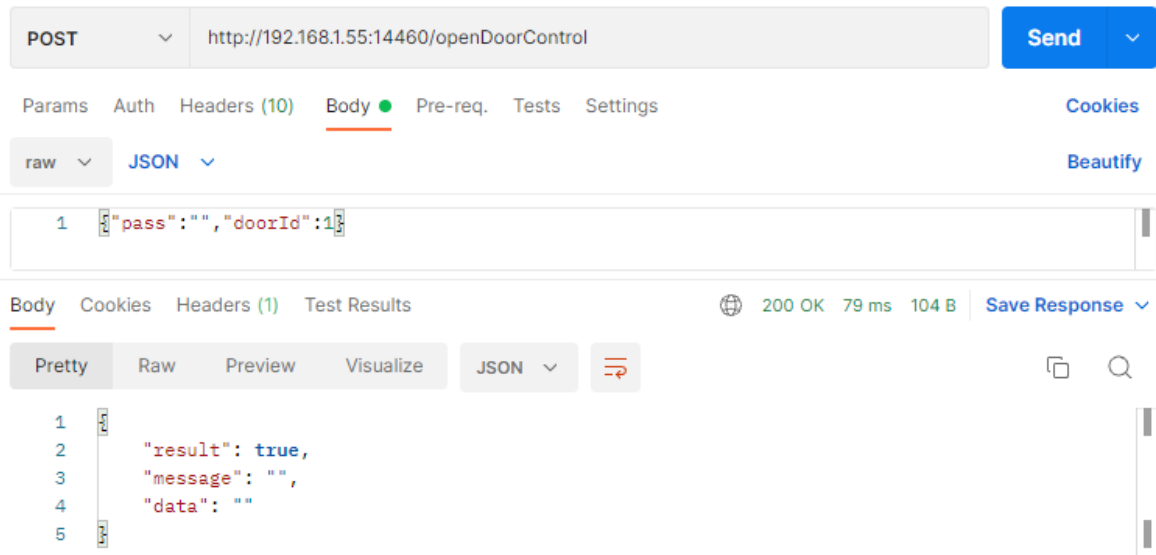
3.11.Remote door opening (openDoorControl)

3.11.1. Request data

Method	URL
POST	<code>http://Device IP:14460/openDoorControl</code>

Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	
doorId	Door number	Int	Y	

3.11.2. Postman example



3.11.3. Example

● Request example

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

● Return to example

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

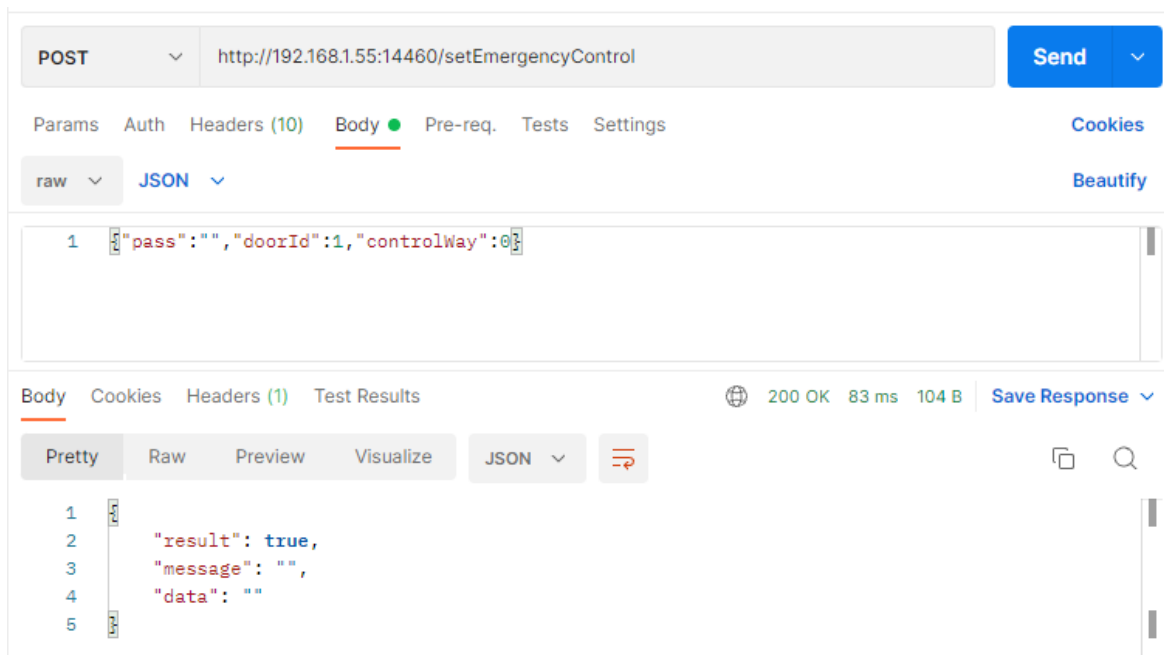
3.12. Emergency switch/locked door (setEmergencyControl)

3.12.1. Request data

Method	URL			
POST	http://Device IP:14460/setEmergencyControl			
Parameter name	Describe	Type	Must send	Additional notes

pass	Password	String	Y	
controlWay	Emergency door opening and closing mode	Int	Y	0:online ,1: normally open, 2: normally closed, 3. locked

3.12.2. Postman example



3.12.3. Example

● Request example

```
{
  "pass": "",
  "doorId": 1,
  "controlWay": 0
}
```

● Return to example

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

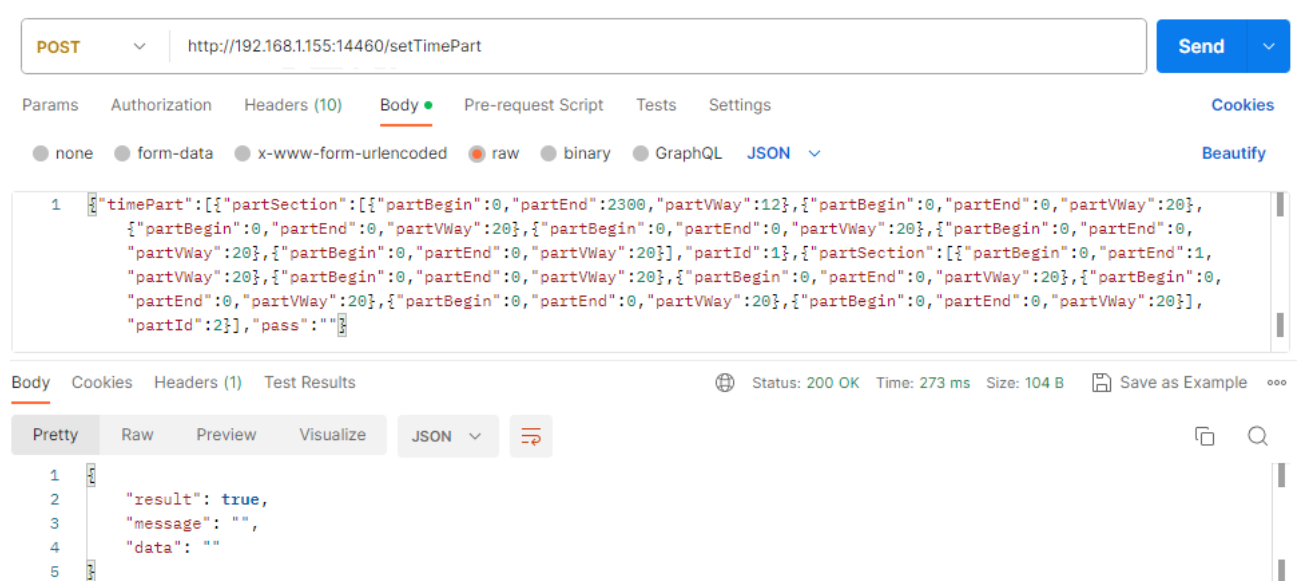
3.13.Set time part (at most 7 groups at a time) (setTimePart)

Note: You can set up to 200 custom groups with numbers from 1 to 200

3.13.1. Request data

Method		URL			
POST		http://Device IP:14460/setTimePart			
Parameter name		Describe	Type	Must send	Additional notes
pass		Password	String	Y	
timePart		Time part collection	Json	Y	
	partId	Time part number	Int	Y	1~200
	partSection	Time part collection	Json	Y	Set 6 segments
	partBegin	Start time of the period	Int	Y	Eg. 8:00 is 800. If disabled, the start and end are both 0
	partEnd	End time of the period	Int	Y	Eg. 18:00 is 1800
	partVWay	Verification type	Int	Y	Definition of reference type and verification typ

3.13.2. Postman example



3.13.3. Example

Request example

```
{
  "timePart": [
    {
      "partSection": [
        {
          "partBegin": 0,
          "partEnd": 2300,
          "partVWay": 20
        },
        {
          "partBegin": 0,
          "partEnd": 0,
          "partVWay": 20
        },
        {
          "partBegin": 0,
          "partEnd": 0,
          "partVWay": 20
        },
        {
          "partBegin": 0,
          "partEnd": 0,
          "partVWay": 20
        },
        {
          "partBegin": 0,
          "partEnd": 0,
          "partVWay": 20
        }
      ],
      "partId": 1
    },
    {
      "partSection": [
        {
          "partBegin": 0,
          "partEnd": 1,
          "partVWay": 20
        },
        {
          "partBegin": 0,
          "partEnd": 0,
          "partVWay": 20
        },
        {
          "partBegin": 0,
          "partEnd": 0,
          "partVWay": 20
        },
        {
          "partBegin": 0,
          "partEnd": 0,
          "partVWay": 20
        },
        {
          "partBegin": 0,
          "partEnd": 0,
          "partVWay": 20
        },
        {
          "partBegin": 0,
          "partEnd": 0,
          "partVWay": 20
        },
        {
          "partBegin": 0,
          "partEnd": 0,
          "partVWay": 20
        }
      ],
      "partId": 2
    }
  ],
  "pass": ""
}
```

Return to example

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

3.14. Get time part (get 5 groups at a time) (getTimePart)

Description: The index starts from 0. The first packet index=0. The second packet needs to pass the maximum partId in the return array to the device as the index until the device appears. Complete means the completion of acquisition

3.14.1. Request data

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

POST		http://Device IP:14460/getTimePart		
Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	
index	Time part index	Int	Y	Index starts at 0

3.14.2. Postman example

The screenshot shows a Postman interface for a POST request to `http://192.168.1.5:14460/getTimePart`. The request body is a JSON object: `{ "pass": "", "index": 0 }`. The response is a detailed JSON object with a `result` of `true`, an empty `message`, and a large `data` array containing time part information for five different sections.

```

{
  "result": true,
  "message": "",
  "data": [
    {
      "timePart": [
        {
          "partId": 1,
          "partSection": [
            {
              "partBegin": 1000,
              "partEnd": 1900,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0
            }
          ]
        }
      ]
    },
    {
      "timePart": [
        {
          "partId": 2,
          "partSection": [
            {
              "partBegin": 200,
              "partEnd": 300,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0
            }
          ]
        }
      ]
    },
    {
      "timePart": [
        {
          "partId": 3,
          "partSection": [
            {
              "partBegin": 300,
              "partEnd": 400,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0
            }
          ]
        }
      ]
    },
    {
      "timePart": [
        {
          "partId": 4,
          "partSection": [
            {
              "partBegin": 400,
              "partEnd": 500,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0
            }
          ]
        }
      ]
    },
    {
      "timePart": [
        {
          "partId": 5,
          "partSection": [
            {
              "partBegin": 500,
              "partEnd": 600,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0
            }
          ]
        }
      ]
    }
  ]
}

```

3.14.3. Example

● Request example 1

```

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

```

● Return to example 1

```

{
  "result": true,
  "message": "",
  "data": [
    {
      "timePart": [
        {
          "partId": 1,
          "partSection": [
            {
              "partBegin": 1000,
              "partEnd": 1900,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0
            }
          ]
        }
      ]
    },
    {
      "timePart": [
        {
          "partId": 2,
          "partSection": [
            {
              "partBegin": 200,
              "partEnd": 300,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0
            }
          ]
        }
      ]
    },
    {
      "timePart": [
        {
          "partId": 3,
          "partSection": [
            {
              "partBegin": 300,
              "partEnd": 400,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0
            }
          ]
        }
      ]
    },
    {
      "timePart": [
        {
          "partId": 4,
          "partSection": [
            {
              "partBegin": 400,
              "partEnd": 500,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0
            }
          ]
        }
      ]
    },
    {
      "timePart": [
        {
          "partId": 5,
          "partSection": [
            {
              "partBegin": 500,
              "partEnd": 600,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0,
              "partBegin": 0,
              "partEnd": 0,
              "partVWay": 0
            }
          ]
        }
      ]
    }
  ]
}

```

```
rtVWay\":0},{\"partBegin\":0,\"partEnd\":0,\"partVWay\":0},{\"partBegin\":0,\"partEnd\":0,\"partVW
ay\":0},{\"partBegin\":0,\"partEnd\":0,\"partVWay\":0}}]]}"
}
```

● Request example 2

```
{
  "pass": "",
  "index": 5
}
```

● Return to example 2

```
{
  "result": true,
  "message": "",
  "data": "{\"timePart\": [{\"partId\": 7, \"partSection\": [{\"partBegin\": 700, \"partEnd\": 800, \"pa
rtVWay\": 0}, {\"partBegin\": 0, \"partEnd\": 0, \"partVWay\": 0}, {\"partBegin\": 0, \"partEnd\": 0, \"partVW
ay\": 0}, {\"partBegin\": 0, \"partEnd\": 0, \"partVWay\": 0}, {\"partBegin\": 0, \"partEnd\": 0, \"partVWay\"
: 0}, {\"partBegin\": 0, \"partEnd\": 0, \"partVWay\": 0}], {\"partId\": 199, \"partSection\": [{\"partBegin
\": 900, \"partEnd\": 1000, \"partVWay\": 0}, {\"partBegin\": 0, \"partEnd\": 0, \"partVWay\": 0}, {\"partBegi
n\": 0, \"partEnd\": 0, \"partVWay\": 0}, {\"partBegin\": 0, \"partEnd\": 0, \"partVWay\": 0}, {\"partBegin\":
0, \"partEnd\": 0, \"partVWay\": 0}, {\"partBegin\": 0, \"partEnd\": 0, \"partVWay\": 0}]]}]}"
}
```

● Request example 3

```
{
  "pass": "",
  "index": 199
}
```

● Return to example 3

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

3.15. Set time group (at most 9 groups at a time) (setTimeGroup)

Note: You can set up to 200 user-defined groups with numbers ranging from 1 to 200. The default 0 and 255 groups do not need to be set separately, 0: pass allowed, 255 means pass prohibited

3.15.1. Request data

Method	URL			
POST	http://Device IP:14460/setTimeGroup			
Parameter name	Describe	Type	Must send	Additional notes

pass	Password	String	Y	
timeGroup	Time group collection			Distribute 9 groups at a time
groupId	Time group number	Int	Y	1~200
tWeekSun	Select time part on Sunday	Int	Y	Select time part number
tWeekMon	Select time part on Monday	Int	Y	Select time part number
tWeekTue	Select time part on Tuesday	Int	Y	Select time part number
tWeekWed	Select time part on Wednesday	Int	Y	Select time part number
tWeekThu	Select time part on Thursday	Int	Y	Select time part number
tWeekFri	Select time part on Friday	Int	Y	Select time part number
tWeekSat	Select time part on Saturday	Int	Y	Select time part number

3.15.2. Postman example

The screenshot shows a Postman interface for a POST request to `http://192.168.1.55:14460/setTimeGroup`. The request body is a JSON object with the following structure:

```

{
  "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
    }
  ]
}

```

The response is a 200 OK status with the following JSON body:

```

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

```

3.15.3. Example

● 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
    }
]
}

```

● Return to example

```

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

```

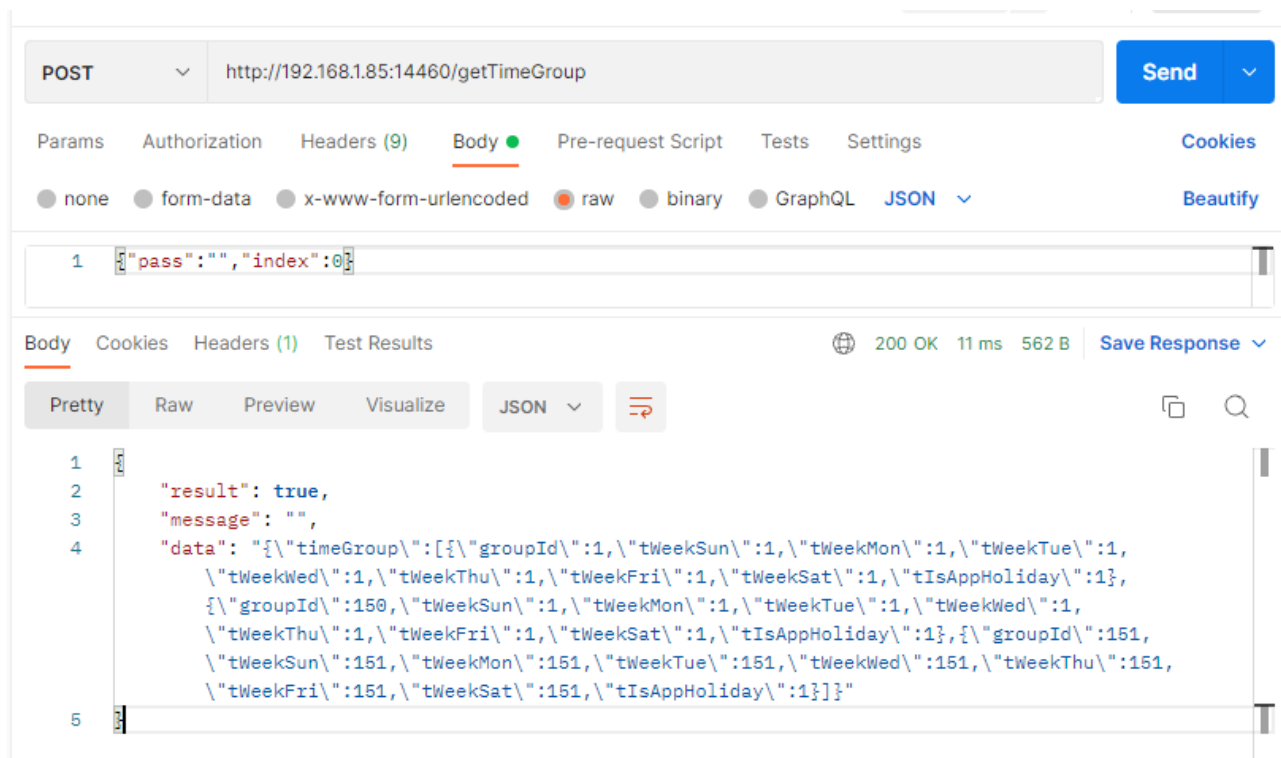
3.16. Get time groups (get 9 groups at a time) (getTimeGroup)

Description: The index starts from 0. The first packet index=0. The second packet needs to pass the maximum groupId in the returned array to the device as the index until the device appears. Complete means the acquisition is complete

3.16.1. Request data

Method	URL			
POST	http://Device IP:14460/setTimeGroup			
Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	
index	Time group index	Int	Y	Index starts at 0

3.16.2. Postman example



3.16.3. Example

● Request example 1

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

● Return to example 1

```
{
  "result": true,
  "message": "",
  "data": "{\n\"timeGroup\": [{\n\"groupId\":1,\n\"tWeekSun\":1,\n\"tWeekMon\":1,\n\"tWeekTue\":1,\n\"tWeekWed\":1,\n\"tWeekThu\":1,\n\"tWeekFri\":1,\n\"tWeekSat\":1,\n\"tIsAppHoliday\":1}, {\n\"groupId\":150,\n\"tWeekSun\":1,\n\"tWeekMon\":1,\n\"tWeekTue\":1,\n\"tWeekWed\":1,\n\"tWeekThu\":1,\n\"tWeekFri\":1,\n\"tWeekSat\":1,\n\"tIsAppHoliday\":1}, {\n\"groupId\":151,\n\"tWeekSun\":151,\n\"tWeekMon\":151,\n\"tWeekTue\":151,\n\"tWeekWed\":151,\n\"tWeekThu\":151,\n\"tWeekFri\":151,\n\"tWeekSat\":151,\n\"tIsAppHoliday\":1}]]}"
}
```

● Request example 2

```
{
  "pass": "",
  "index": 151
}
```

● Return to example 2

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

3.17.Set user power (setUserPower)

Note: The power list collection can upload 5 power at a time, and the power are unique with the card number

The door power group set, powerTimeGroup, is the power of all the doors of the current device. If four-door device needs to cover the door 1, 2, 3, the door power group only covers door 1, 2, 3, without four-door power. If you delete the power of all the doors for this person, you can issue the parameter opType 0

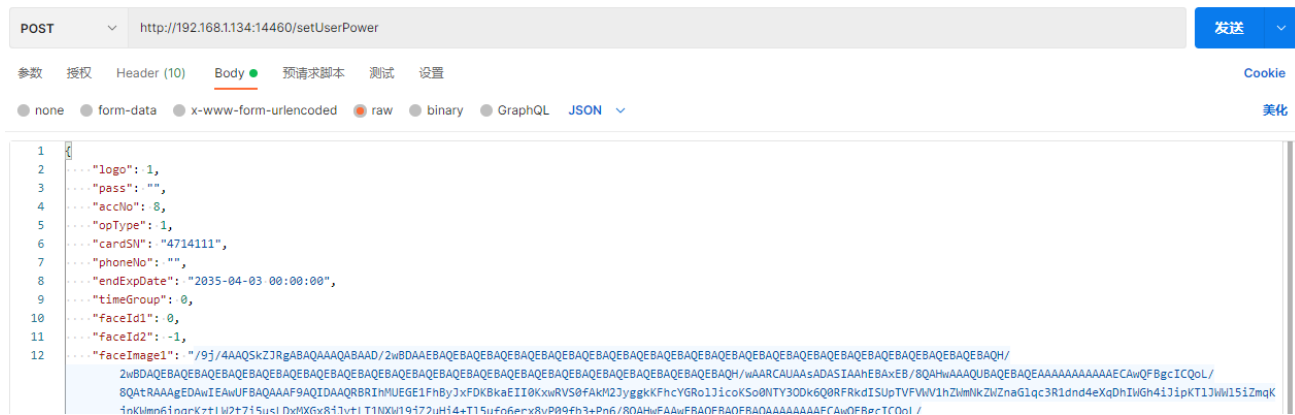
3.17.1. Request data

Method	URL			
POST	http://设备 IP:14460/setUserPower			
Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	
accNo	work number	Int	Y	unique identification
opType	Operation type	Int	Y	1: Upload, 0 delete
cardSN	Card No	String	Y	Take the card number with the card number, but not the last 9 digits of the mobile phone number
endExpDate	deadline	String	Y	For example: 2099-12-31 18:30:00
beginExpDate	Start Date	String	Y	Date format "yyyy-MM-dd HH:mm"
timeGroup	Time group number	Int	Y	
faceId1	Do you have any photos	Int	Y	None: -1, Yes: 0, Permission must include photo
faceImage1	photo	Base64	Y	If there is no, then null.

				Base64 without labels, Within 100k, the facial photo should not be too blurry, with a resolution of 1080 * 720 or below and 240 * 360 or above
name	Employee name	String	Y	Supports up to 32 bytes
password	Password	String	Y	8 and less letters and numbers, no password empty string
dept	Department Number	Int		Department ID (1-50)
isManager	Is it an administrator	Int	Y	0: Regular User 1: Administrator (menu accessible)
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	Obtain from the protocol getUserPower
fp2	Second fingerprint data	String	N	Obtain from the protocol getUserPower
palmlId	Is there a palm vein	Int	Y	None: 0, Yes: 1

palmlInfo	palmar vein data	Base64	N	Obtain from the protocol getUserPower
-----------	------------------	--------	---	---------------------------------------

3.17.2. Postman example



3.17.3. Example

● Request example

```
{
  "pass": "",
  "accNo": 3,
  "opType": 1,
  "isManager": 0,
  "password": "",
  "cardSN": "0000000002",
  "endExpDate": "2029-03-18:00:00",
  "timeGroup": 0,
  "name": "N0000002",
  "faceId1": 0,
  "dept": 1,
  "faceImage1": "/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAAGGBQgHBwJCQgKDBQNDAsLDBkSEw8UHRofHh0aHBwgJC4nICIsIxwKDcpLDAXNDQ0Hyc5PTgyPC4zNDL/2wBDAQKJCQwLDBgNDRgyIRwhMjIyMjIyMjI.../9k=",
  "fpId1": 1,
  "fpId2": 1,
  "fp1": "80AE25033A371F39...",
  "fp2": "80AE25033A371F39...",
  "palmId": 1,
  "palmInfo": "zAAAADEBAABJpVPs7DVROIxnW...省略"
}
```

● Return to example

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

3.18. Get power (getUserPower)

Description: The index starts from 0. Every time the acquisition is successful, the device will return a new index. This index will be returned to the device until the

device appears. Complete means the completion of acquisition

3.18.1. Request data

Method	URL			
POST	http://Device IP:14460/getUserPower			
Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	
index	power index	Int	Y	Index starts at 0

3.18.2. Postman example

[illegible]

3.18.3. Example

- **Request example 1**

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

- **Return to example 1**

```
{
  "result": true,
  "message": "",
  "data": [{"index":1,"accNo":3,"accName":"N0000002","dept":1,"password":"123123\
","validStart":"2000-01-01 00:00:00","validEnd":"2029-03-
23 18:00:00","isManager":0,"cardSN":"2","powerTimeGroup":[{"doorId":1,"group":0}],{"f
```

```
aceId1\":1,\"faceImage1\":\"/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UHR  
ofHh0aHBwgJC4nICIsIxwcKDcpLDAxNDQ0Hyc5PTgyPC4zNDL/...  
.../9k=\\\"}\"
```

Parameter Description:

accNo:work number

accName:name

cardSN:Card

password:password

validStart:Validity period start time

validEnd:Expiration date

isManager:Is it an administrator

dept:department number

group:time group

faceId1:Is there photo data? 0: No 1: Yes

faceImage1:Face photo in base64 format

fpId1:Is there a first fingerprint 0: No 1: Yes

fpId2:Is there a second fingerprint 0: No 1: Yes

fp1:The first fingerprint data,The returned information is consistent with the issued one

fp2:Second fingerprint data,The returned information is consistent with the issued one

palmId:Is there a palmar vein, 0: No 1: Yes

palmInfo:palmar vein,A string in base64 format

● Request example 2

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

● Return to example 2

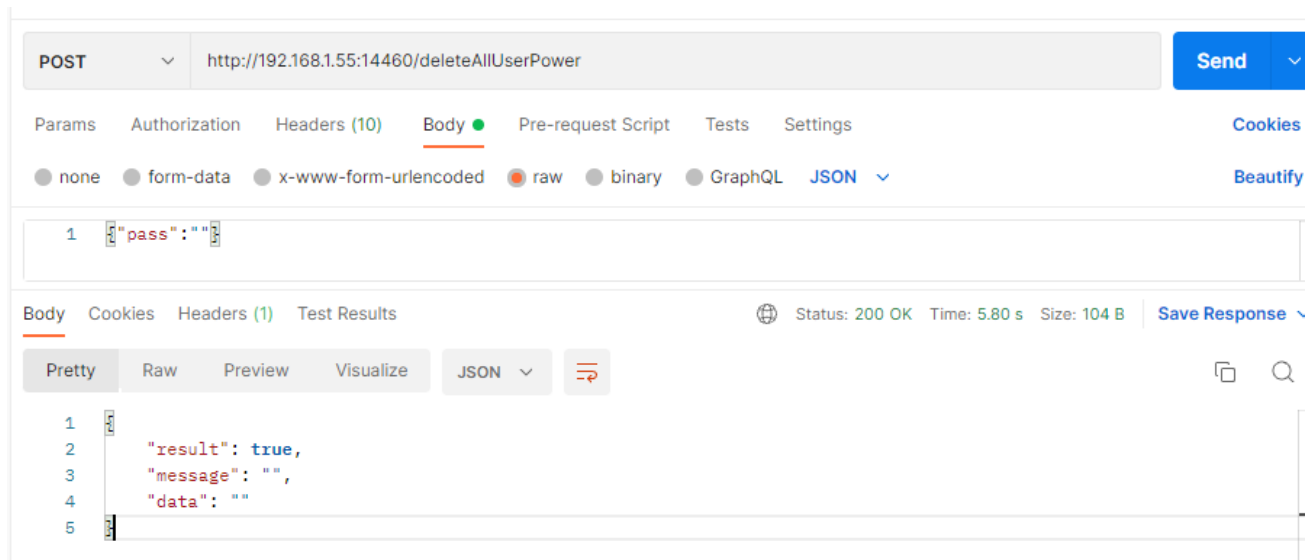
```
{  
  "result": true,  
  "message": "complete"  
}
```

3.19.Clear power (deleteAllUserPower)

3.19.1. Request data

Method	URL			
POST	http://Device IP:14460/deleteAllUserPower			
Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	

3.19.2. Postman example



3.19.3. Example

● Request example

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

● Return to example

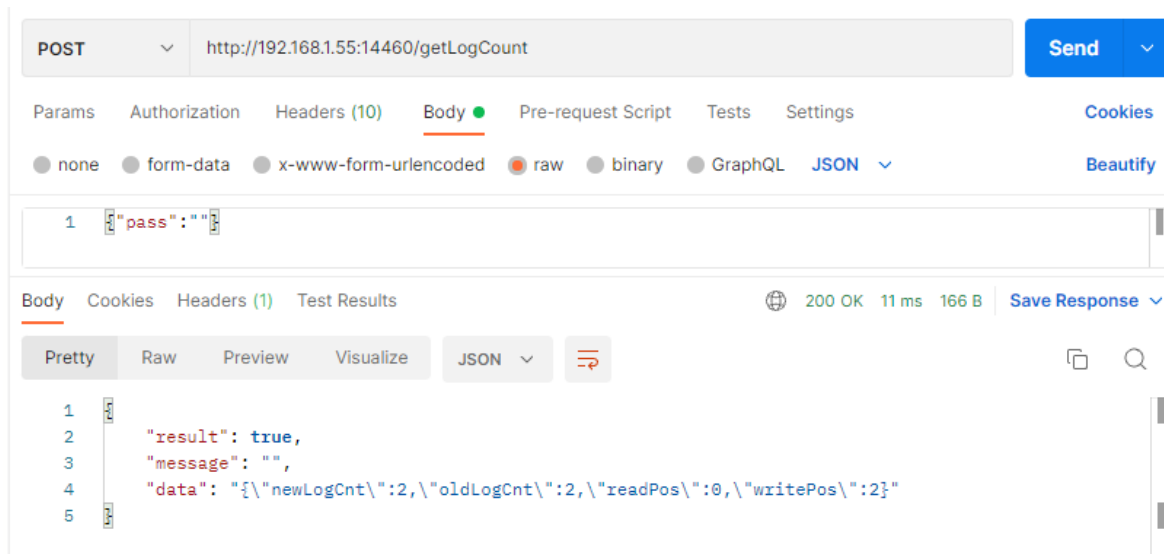
```
{  
  "result": true,  
  "message": "",  
  "data": ""  
}
```

3.20. Get Number of logs (getLogCount)

3.20.1. Request data

Metho	URL			
POST	http://Device IP:14460/getLogCount			
Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	

3.20.2. Postman example



3.20.3. Example

● Request example

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

● Return to example

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

Note:

NewLogCnt//Number of new logs

OldLogCnt//Number of history logs

3.21.Get log information (getLogInfo)

3.21.1. Request data

Method	URL			
POST	http://Device IP:14460/getLogInfo			
Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	
flag	Sign	Int	Y	0 :history, 1 :new log

length	Number of acquisition	Int	Y	Number of logs retrieved at one time
index	Indexes	Int	Y	The index number starts from 0. For example, if there are 100 logs and 8 logs are collected at a time, the first request is 0 and the second request is 8
i sphoto	Whether to collect photos	Int	N	0:Not collecting 1:collect

3.21.2. Postman example

3.21.3. Example

● Request example

```
{"pass":"","flag":1,"length":10,"index":0}
```

● Return to example

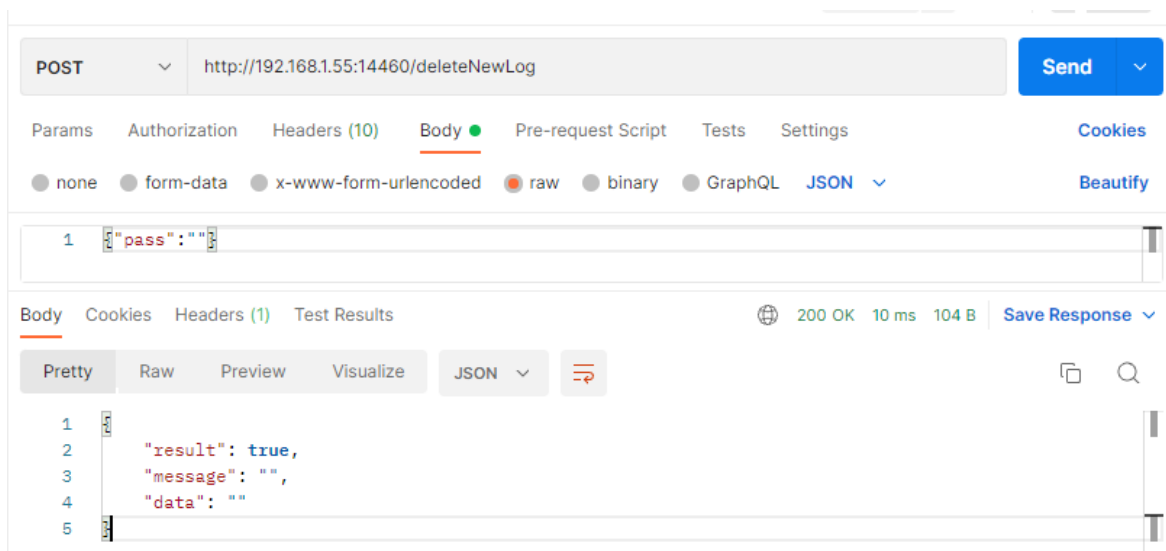
```
{
  "result": true,
  "message": "",
  "data": "[
    {\\"deviceKey\\":\\"FFFAAB69B83FC3D\\",\\"device serial number
    \\"reqSeqNo\\":\\"168629945364978903\\",\\"log serial number, integer data
    \\"logType\\":0,\\"log type (0: normal log, 1: illegal user log, 2: system and alarm log)
    \\"accNo\\":3,\\"When the account number or alarmCode=91 and operation=0, this field is the card
    number
    \\"passTime\\":\\"2023-06-09 16:30:53\\",\\"logging time: Time format: yyyy-MM-dd HH: mm: ss
    \\"operation\\":2,\\" Pass mode , reference attribute, log the pass line way
    \\"alarmCode\\":0,\\"Alarm status , Reference attribute, log alarm code
    \\"passStatus\\":0,\\"Pass status, 0 passed 1, not passed
    \\"doorId\\":1,\\"door number
    \\"readId\\":1,\\"Attendance status (1~10)
    \\"temperature\\":0,\\"This field is meaningless, ignore it
  ]"
}
```

3.22.Delete the latest log (deleteNewLog)

3.22.1. Request data

Method	URL			
POST	http://Device IP:14460/deleteNewLog			
Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	

3.22.2. Postman example



3.22.3. Example

● Request example

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

● Return to example

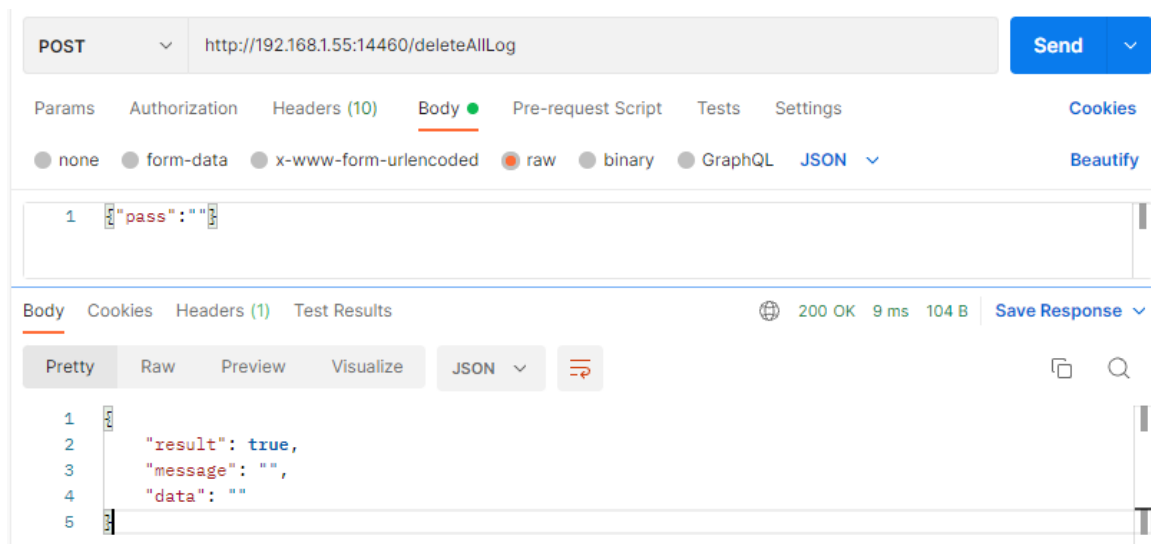
```
{  
  "result": true,  
  "message": "",  
  "data": ""  
}
```


3.23. Clear the device log (deleteAllLog)

3.23.1. Request data

Method	URL			
POST	http://Device IP:14460/deleteAllLog			
Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	

3.23.2. Postman example



3.23.3. Example

● Request example

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

● Return to example

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

3.24.Set the device active uploading log (setLogCallBack)

"Note: After the callback setting is successful, you need to listen to and log the callback server IP,

The uploaded log is the log after the callback setting is successful. The log before the callback setting needs to be collected

3.24.1. Request data

Method	URL			
POST	http://Device IP:14460/setLogCallBack			
Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	
logCallBackURL	The device actively uploads the log service IP	String	Y	For example:" http://192.168.31.179:13333/uplog " Set "" to cancel real-time log uploading
readCard	Card reading mode or not	Int		1: card reading, 0 real-time monitoring

3.24.2. Postman example

The screenshot shows a Postman interface for a POST request. The URL is `http://192.168.1.55:14460/setLogCallBack`. The request body is a JSON object with the following fields:

```
1 {
2   "pass": "",
3   "logCallBackURL": "http://192.168.1.103:13333/uplog",
4   "readCard": 0
5 }
```

The response is a 200 OK status with a response time of 81 ms and a body size of 104 B. The response body is a JSON object:

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

3.24.3. Example

● Request example

```
{  
  "pass": "",  
  "logCallBackURL": "http://192.168.1.103:13333/uplog",  
  "readCard": 0  
}
```

● Return to example

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

Real-time monitoring log:

1. When returning the card swiping log, refer to "Get LogInfo" for the content

Example:

```
{"deviceKey": "FFFFAAB69B83FC3D", "havLog": 1, "reqSeqNo": "168630277464978901", "logType": 0, "accNo": 3, "visitorId": 0, "passTime": "2023-06-09 17:26:14", "operation": 2, "alarmCode": 0, "passStatus": 0, "doorId": 1, "readId": 1, "temperature": 0, "lockStatus": [{"doorId": 1, "status": 0}]}
```

2.

Example:

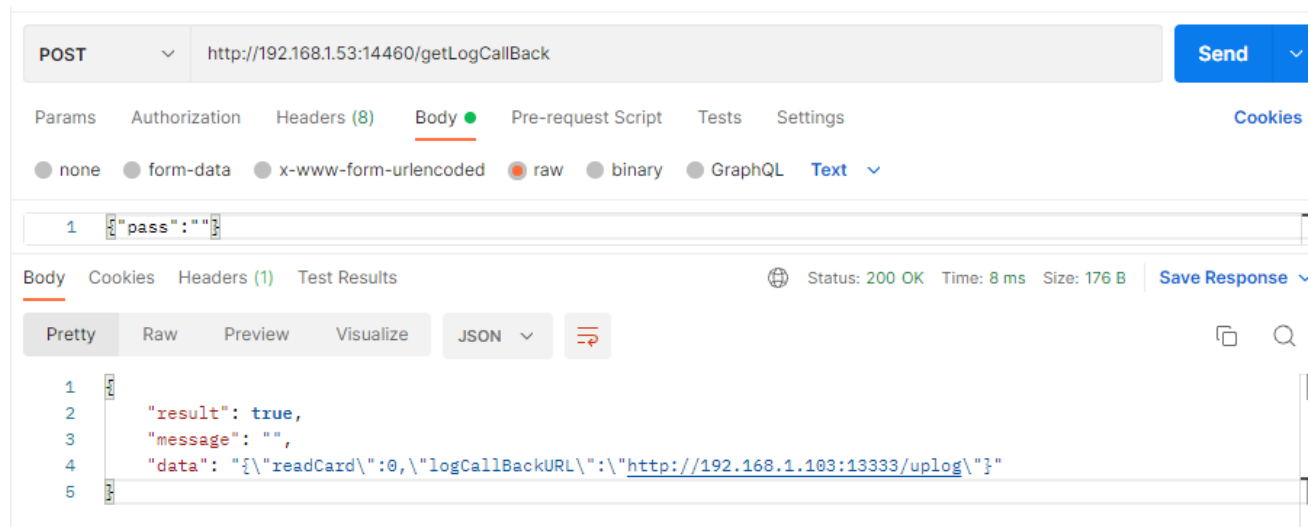
```
{"deviceKey": "FFFFAAB69B83FC3D", "havLog": 0, "lockStatus": [{"doorId": 1, "status": 1}]}
```

3.25. Get the active upload log server address (getLogCallBack)

3.25.1. Request data

Method	URL			
POST	http://Device IP:14460/getLogCallBack			
Parameter name	Describe	Type	Must send	Additional notes
pass	Password	String	Y	

3.25.2. Postman example



3.25.3. Example

● Request example

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

● Return to example

```
{
  "result": true,
  "message": "",
  "data": "{\"readCard\":0,\"logCallBackURL\": \"http://192.168.1.103:13333/uplog\"}"
}
```

Note: The parameters are the same as the settings

3. 26. Issuing department (setDepartment)

Note: The department list collection can upload up to 50 departments at once, with ID as the unique identifier

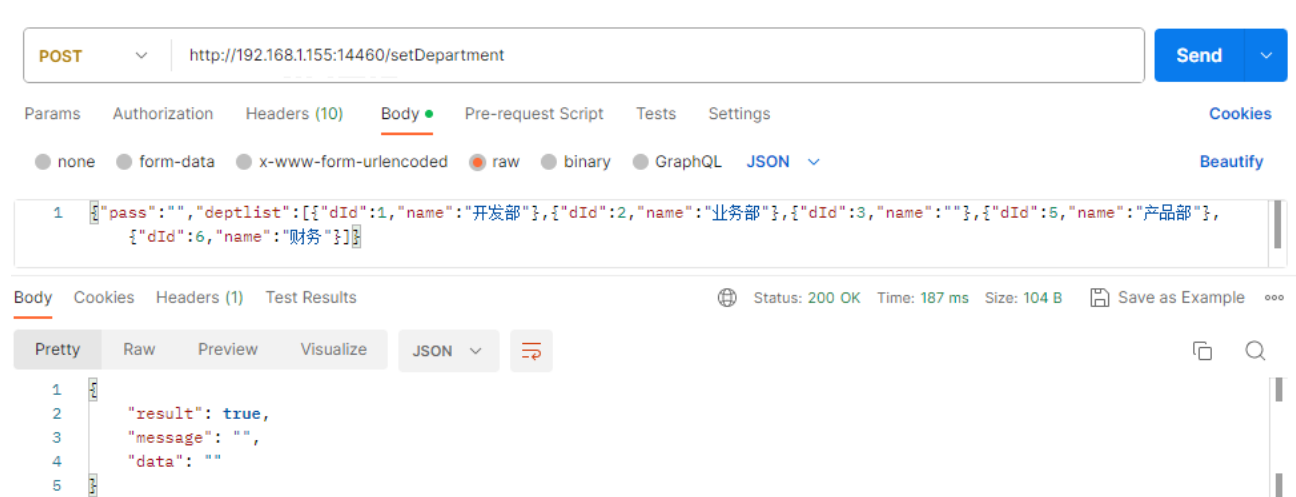
Issuance processing instructions: The machine only modifies the corresponding ID issued this time, and does not process other existing ID names

3. 26. 1. Request data

Method	URL
POST	http:// IP:14460/setDepartment

Parameter name		Describe	Type	Must send	Additional notes
pass		password	String	Y	
deptlist		Department List	Json	Y	Up to 50 departments
	dId	Department ID	Int	Y	1~50
	name	Department Name	String	Y	Supports up to 32 bytes

3.26.2. Postman example



3.26.3. Example

● Request example

```
{ "pass": "", "deptlist": [ { "dId": 1, "name": "Development" }, { "dId": 2, "name": "Development" }, { "dId": 3, "name": "Development" }, { "dId": 4, "name": "Development" }, { "dId": 5, "name": "Development" }, { "dId": 6, "name": "Development" }, { "dId": 7, "name": "Development" }, { "dId": 8, "name": "Development" }, { "dId": 9, "name": "Development" }, { "dId": 10, "name": "Development" }, { "dId": 11, "name": "Development" }, { "dId": 12, "name": "Development" }, { "dId": 13, "name": "Development" }, { "dId": 14, "name": "Development" }, { "dId": 15, "name": "Development" }, { "dId": 16, "name": "Development" }, { "dId": 17, "name": "Development" }, { "dId": 18, "name": "Development" }, { "dId": 19, "name": "Development" }, { "dId": 20, "name": "Development" }, { "dId": 21, "name": "Development" }, { "dId": 22, "name": "Development" }, { "dId": 23, "name": "Development" }, { "dId": 24, "name": "Development" }, { "dId": 25, "name": "Development" }, { "dId": 26, "name": "Development" }, { "dId": 27, "name": "Development" }, { "dId": 28, "name": "Development" }, { "dId": 29, "name": "Development" }, { "dId": 30, "name": "Development" }, { "dId": 31, "name": "Development" }, { "dId": 32, "name": "Development" }, { "dId": 33, "name": "Development" }, { "dId": 34, "name": "Development" }, { "dId": 35, "name": "Development" }, { "dId": 36, "name": "Development" }, { "dId": 37, "name": "Development" }, { "dId": 38, "name": "Development" }, { "dId": 39, "name": "Development" }, { "dId": 40, "name": "Development" } ] }
```

```
ame": "Development"}, {"dId": 41, "name": "Development"}, {"dId": 42, "name": "Development"}, {"dId": 43, "name": "Development"}, {"dId": 44, "name": "Development"}, {"dId": 45, "name": "Development"}, {"dId": 46, "name": "Development"}, {"dId": 47, "name": "Development"}, {"dId": 48, "name": "Development"}, {"dId": 49, "name": "Development"}, {"dId": 50, "name": "Development"}]]}
```

● Return to example

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

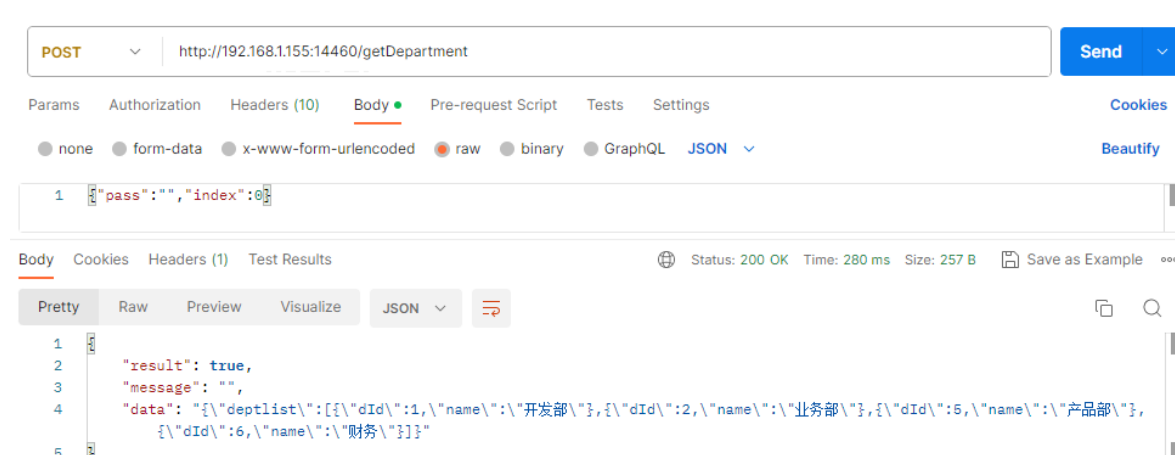
3. 27. Obtain department (getDepartment)

Note: The index starts from 0, and every time it is successfully obtained, the device will return a new index. This index will then be returned to the device until it appears, complete, Representative acquisition completed

3. 27. 1. Request data

Method	URL			
POST	http:// IP:14460/getDepartment			
Parameter name	Describe	Type	Must send	Additional notes
pass	password	String	Y	
index	index	Int	Y	Index starts from 0

3. 27. 2. Postman example



3.27.3. Example

● Request example

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

● Return to example

```
{
  "result": true,
  "message": "",
  "data": "{\"deptlist\": [{\"dId\": 1, \"name\": \"Development\"}, {\"dId\": 2, \"name\": \"business\"}, {\"dId\": 5, \"name\": \"production department\"}, {\"dId\": 6, \"name\": \"finance\"}]}"
}
```

4. UDP protocol

4.1. General return description of Interface

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

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

4.2. Search device (UDP broadcast) (1001)

4.2.1. Parameter description

Instruction: 1001	Description: UDP sends broadcast data (255.255.255.25514440)
Request parameters	

<pre> { Pass: "",//Udp communication password, 8 characters or less, ex-factory password is "" AppCode: 1001//Command ID } </pre>
Response parameters
<pre> { "result": true, "message": "", "data": { DeviceKey: "5024668358A3BF1C",//Serial number DeviceType: 70,//Device type, reference FirmWareVer: "2021082501",//version number DoorType: 2,//door type, 1: single door plank 2: double door plank 4: four door plank TermId: 1,//device number OnlineWay: 0,//Networking method EnableDHCP: 0,//Whether to enable dynamic IP acquisition 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: 14440//Server port LableSN: 5000004,//tag serial number DeviceModel: "JH51",//model Oem: "FFFEFD9CFBFAF9F8F7F6F5F4F3F2F1F0"//oem, complete settings by configuring OEM } } </pre>

4.3. Configure IP (UDP broadcast) (1002)

4.3.1. Parameter description

Instruction: 1002 Description: UDP sends broadcast data (255.255.255.25514440)
Request parameters
<pre>{ Pass: "",//UDP communication password and this parameter is required for udp protocol "deviceKey":"FFFFFFFFFFFFFFFF", "appCode": 1002, NewPass: "123456",//Udp communication password. The password is 6 digits or less, and the old password is not modified, or only the latest password is transmitted EnabledDHCP: 0,//Automatically obtain IP. 0 disabled, 1 enabled "localIp": "192.168.1.200", "subAdd": "255.255.255.0", "defaGate": "192.168.1.1" "termId": 1 //device number }</pre>
Response parameters
<pre>{"result":true,"message":"","data":""}</pre>

4.4. Configure OEM (UDP broadcast) (1010)

4.4.1. Parameter description

Instruction: 1010 Description: UDP sends broadcast data (255.255.255.25514440)
Request parameters
<pre>{ "pass": "", DeviceKey: "5024668358A3BF1C",//Serial number AppCode: 1010,//Command ID OemPassword: "FFFEFD9CFBFAF9F8F7F6F5F4F3F2F1F0"//OEM password, hexadecimal digits, 32 characters }</pre>
Response parameters
<pre>{"result":true,"message":"","data":""}</pre>

4.5. Configure device model (UDP broadcast) (1014)

4.5.1. Parameter description

Instruction: 1014 Description: UDP sends broadcast data (255.255.255.255:14440)	
Request parameters	
{ "pass": "", DeviceKey: "5024668358A3BF1C",//Serial number AppCode: 1014,//Command ID DeviceModel: ""//Set the model, such as JH50, with a maximum of 8 characters }	
Response parameters	
{"result":true,"message":"","data":""}	

5. Type definition

5.1. Verification type - time part verification type

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

5.2. log alarm code

Value (decimal)	meaning
-----------------	---------

0	Normal
80	Illegal card
81	Door not closed after timeout
83	Fire Alarm
86	Forced entry
90	Expired validity period
91	Stranger
95	Alarm release
110	Verification passed without opening the door
113	Password error
115	Anti duplication card

5.3. log the way 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)

5.4. Device type

Value (decimal)	meaning
32	Face all-in-one LAN

5.5. Door control type

Value (decimal)	meaning
1	Single door panel (single door two-way, 2 readers)
2	Double-door panel (double-door two-way, 4 readers)
4	Four door panels (four doors two-way, 8 readers)

5.6. Door control mode

Value (decimal)	meaning
0	On-line
1	Normally open
2	Normally closed