dormakaba
Face Recognition Terminal 91 60-K5

Terminel Interface V 2.5

EN

Contents

HTTP(S)+MQTT Mode

1.Http(s) Interface Encryption and Data Structures

1.1 Modalities of the request

The HTTP(S) communication between the face recognition terminal and the server, if no special instructions, the default use of HTTP(S) POST request, the request format is Content-Type: application/json. when the device is registered to the platform (equivalent to logging in), the backstage will return a token, and the subsequent Http(S) request needs to carry this token in the request header.。

1.2 Encryption methods

For Http(s) POST request need to unify the request body (Http(s) request body) for AES encryption, the key needs to be negotiated with the system backend developers, for the get request do not need to do encryption, but the response data or Content-Type: application/json will be response data AES encryption.
In order to facilitate the development and testing, the test environment allows plaintext transmission, when the client requests the interface, in the Http(s) header, add deivces=app, then it means that the transmission is in plaintext, and the request data and response data will not be encrypted. (Note: the formal environment, even if there is added to the request header, will be forced to encrypt and decrypt).

When the server interface encrypts the response data, it will add encryption=true in the response header to indicate that the response data has been encrypted and the client needs to decrypt it before processing.
Example of encryption: For example, before encrypting the fields of the request for device registration to the platform: (the meaning of the fields is explained later) Before encryption:

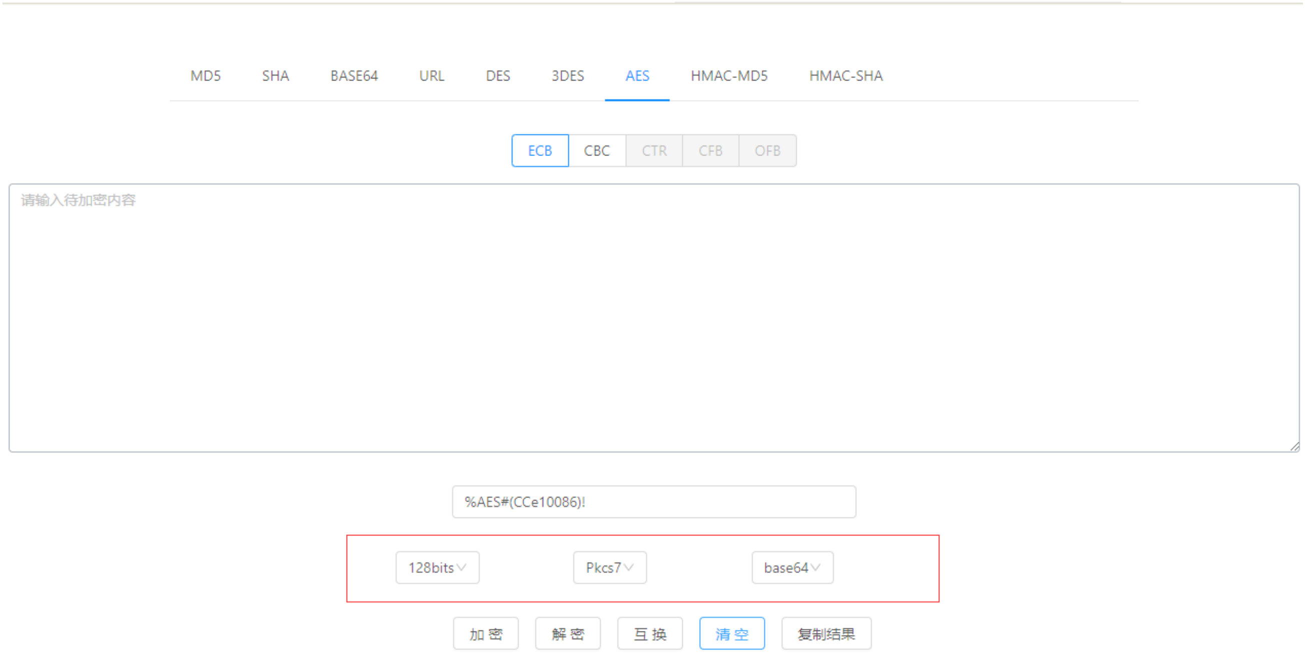
r

```
{"appVersionCode":1,"appVersionName":"1.0","deptId":101,"devLanguage":"zh_CN","devName":"TR08A-A","devSn":"YGKJ20204200040","model":"TR08A-A","networkIp":"192.168.199.176","networkType":0,"onlineStatus":0,"romVersion":"rk3399_all-YT19.2-L02-C01-U00-eng.ygkj.20200903.085939-release"}
```

AES key is：1234567890123456
AES key after encryption：

VqpMjnqVhVHZRQ8BCYCEBjZ/OCbOubOSgfcXesoqru1/99e3VgJFtRVzRBZOSD3jay8nHqto4PZ5Wvn
gThqzGbC61JLQkmYyLJddFWjudTJY5Uo2Yb/J0juB0a/nCuC3XM6VfiOQTaDeWCOv4J5XzRC87yJb8
ZApgrA0XE95o2/IKmBwGjbje+EfU6RUbkL7L0v9YjI+r6ZIZXvd+NPPOQZKqpv4

AES Encrypted reference or online checking：Http(s)s://www.keylala.cn/aes



%AES#(CCe10086)!

128bits

Pkcs7

base64

1.3 Http(s) response format

The uniform return format of the server-side Http(s) interface is：

```
{
    "success":true，
    "code":0,
    "desc":"成功",
    "data":{}
}
```

Return parameter description：

Field Name	Type	Description
success	Boolean	true Indicates successful request，false Indicates a failed request
code	Int	Error code, check the description of the error code
desc	String	Error code, check the description of the error code

data	Object	may be an object, may be an array, when success is false, may be null

1.4 Http(s) error code description

The server-side Http(s) interface needs to refer to the following table to return the error code

code	desc	remark
0	successful	
301	warning	
400	failed	400 some errors are written directly in the desc
403	parameter error	normally caused by encrypted request parameters or inconsistent request field formats (int to String, etc.)
4174	token failure	Token failure requires a re-call to the registration interface to get the
500	error	Normally caused by backend interface error
501	parameters can't be null	required fields are empty resulting in
502	body parameter can’t be null	
503	header parameter can’t be null	

2. Http(s) Interface

After the face recognition terminal starts, it first calls the request registration interface to send the registration request, and after the server side verifies it, it returns the necessary server side parameters (such as the Http(s) token and the mqtt broker parameters) to the face recognition terminal, and after the face recognition terminal connects to the mqtt broker and subscribes to the relevant topics, the server side can issue commands to the face recognition terminal.

The server side can realize "request registration", "add user", "add pass record", "modify user", "delete user" a total of five Http(s) interface.

Description of request address: When the face recognition terminal is initialized, the server address is required to be input, and when the terminal executes the Http(s) request, the server address is spliced with the Http(s) request address in the following section, e.g., the server address Http(s)://192.168.2.10:8088 is set at initialization, and the complete address when the request is executed for registration will be: Http(s)://192.168.2.10:8088/api/devices/login .

2.1 Request for registration

Brief description： After the facial recognition terminal APP starts, it sends a registration request to the server, and after the server verifies it, it returns the necessary server-side parameters (such as the Http(s) token and the mqtt broker parameters) to the face recognition terminal, and the face recognition terminal gets the server-side parameters, so that it can make other Http(s) requests and connect to the mqtt broker.

Request address： /api/devices/login

Request method： POST

Request Sample：

```
{"appVersionCode":1,"appVersionName":"1.0","deptId":101,"devLanguage":"zh_CN","devName":"TR08A-A","devSn":"YGKJ20204200040","model":"TR08A-A","networkIp":"192.168.199.176","networkType":0,"onlineStatus":0,"romVersion":"rk3399_all-YT19.2-L02-C01-U00-eng.ygkj.20200903.085939-release"}
```

Description of request parameters：

Field Name	**Type**	**Remark**
appVersionCode	Int	APP version

appVersionName	String	APP version name，like 1.1.2
deptId	Long	Device department ID (company)
devLanguage	String	Device Language
devName	String	Device name
devSn	String	Device serial no(only ID)
model	String	Device model
networkIp	String	Device IP address
networkType	Int	Device network type(0=Wifi, 1= Ethernet)
onlineStatus	Int	Online status（0=offline，1=online）when registering, will transfer the value=0 after MQTT connected will automatically switch to 1
romVersion	String	ROM version

Server-side Example for data return：

```
{
  "code":0,
  "data":{
       "mqttUserName":"android",
       "mqttPassowrd":"android",
       "mqttUrl":"tcp://192.168.20.168:1883",
       "token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJZR0tKMjAyMDQyM
       DAwNDAiLCJleHAiOjE2MDI4NzM5NjV9.6d5Jk_wSVwOChL9mh5CO3hfJ7udQp32N
       doiFEOcOXdU"
     },
  "desc":"成功",
  "success":true
}
```

Return field description：

Field Name	**Type**	**Remark**
mqttUserName	String	MQTT client account
mqttPassowrd	String	MQTT client key
mqttUrl	String	MQTT server address

token	String	The backend interface needs to carry this field in the request header

2.2 Add user

Brief description：Operator add new user through the UI at the terminal, and when the operator confirms the addition, the device uploads the personnel information to the server through this interface, and the server needs to refer to the example to return the data in order to synchronize the local database of the face recognition terminal.

Request address：/api/devices/addUser

Request Method：POST

Terminal request form sample：

```
{
   "userName": "张三",
   "userCode": "YGKJ123456",
   "userPhone": "15568781345",
   "faceUrl": "",/**Base64 Coded photos*/
   "sex": 1,
   "status": 1,
   "devSn": "YGKJ20204202773",
   "cardNum": "ABCD",
   "wiegandNum":"100",
   "company":"",
   "department":"",
   "group":"",
   "remark":"",
   "expiry":""
}
```

Request Parameter Description：

Field	Type	Description
userName	String	User name
userCode	String	User code
userPhone	String	User phone no
faceUrl	String	Facial photo:Base64 format
sex	String	Sex 0=male，1=female
status	Int	0:active，1:deactivated
devSn	String	Device number
cardNum	String	RFID card number
wiegandNum	String	Wiegand number，after successful user authentication, send the Weigand number through the Weigand interface
company	String	Company
department	String	department
group	String	group
remark	String	remark
expiry	String	Validition format： yyyy-MM-dd hh:mm:ss,yyyy-MM-dd hh:mm:ss Separated by a comma, the first is the start time of the expiration date and the second is the end time. If an expiration date is to be used, it must be set up in the complete format

Server-side return sample：

```
{
    "code":0,
    "desc":"成功",
    "success":true,
    “data”:{
        "id": 0, // Database ID, assigned in the server and returned to the client for saving。
        "deptId": 0,
        "userName": "",
        "userCode": "",
        "userPhone": "",
        "faceUrl": "", // Path to the full face photo (not the same as the base64 photo data in the
        request)
        "sex": 0,
        "status": 0,
        "devSn": "",
        "cardNum": "ABCD",
        "wiegandNum":"100",
        "company":"",
        "department":"",
        "group":"",
        "remark":"",
        "expiry":"",
        "searchValue": "",
        "createBy": "",
        "createTime": "2020-10-24 08:18:51",
        "updateBy": "",
        "updateTime": "2020-10-24 08:18:51",
        "remark": "",
    }
}
```

Note: After adding a new user, the modified user information will be returned, and the client needs to save the user's data ID and the complete photo path.

2.3 User modify

Brief description: The operator can modifies the user information through the UI of the f terminal, and when the operator confirms the modification, the device uploads the modified user information to the server, and the server returns the data with reference to the return example in order to synchronize the local database of the face recognition terminal.

Request address: /api/devices/editUser

Request method: POST

Terminal request form sample:

```
{
    "id": 1, // transfer the ID of the new return
    "userName": “李四”,
    "userCode": "YGKJ354”,
    "userPhone": "15568781345",
    "faceUrl": "",/**Base64 coded photo*/
    "sex": 1,
    "status": 1,
    "devSn": "YGKJ20204202773",
    "cardNum": "ABCD",
    "wiegandNum":"100",
    "company":"",
    "department":"",
    "group":"",
    "remark":"",
    "expiry":""
}
```

Request Parameter Description：

Field	Type	Description
id	int	User ID, the ID returned when adding a new user
userName	String	User name
userCode	String	User code
userPhone	String	User phone no
faceUrl	String	Facial photo:Base64 format
sex	String	Sex 0=male，1=female
status	Int	0:active，1:deactivated
devSn	String	Device number
cardNum	String	RFID card number
wiegandNum	String	Wiegand number，after successful user authentication, send the Weigand number through the Weigand interface
company	String	Company
department	String	department
group	String	group
remark	String	remark
expiry	String	Validition format： yyyy-MM-dd hh:mm:ss,yyyy-MM-dd hh:mm:ss Separated by a comma, the first is the start time of the expiration date and the second is the end time. If an expiration date is to be used, it

		must be set up in the complete format

Server-side return sample：

```
{
    "code":0,
    "desc":"成功",
    "success":true,
    “data”:{
        "id": 0, //数据ID
        "deptId": 0,
        "userName": "",
        "userCode": "",
        "userPhone": "",
        "faceUrl": "", // Path to the full face photo (not the same as the base64 photo data in the
        request)

        "sex": 0,
        "status": 0,
        "devSn": "",
        "cardNum": "ABCD",
        "wiegandNum":"100",
        "company":"",
        "department":"",
        "group":"",
        "remark":"",
        "expiry":"",
        "searchValue": "",
        "createBy": "",
        "createTime": "2020-10-24 08:18:51",
        "updateBy": "",
        "updateTime": "2020-10-24 08:18:51",
        "remark": "",
    }
}
```

Note: After modifying the user, the modified user information will be returned, and the client needs to save the path of the updated photo.

2.4 Delete User

Brief description：The operator deletes the user in the terminal through the UI, and after the operator confirms the deletion, the device uploads the id of the user that needs to be deleted to the server, and the server needs to delete the user information or the associated information in the database of the server and return the result to the face recognition terminal.

Request address: /api/devices/deleteUser

Request method：POST

Terminal request form sample:

```
{
   "id": 14,
   "devSn": "SZ20051309B104SK9P6"
}
```

Request Parameter Description:

Filed Name	**Type**	**Description**
id	int	User ID
devSn	String	Device No

Server-side return sample:

```
{
        "code":0,
        "desc":"成功",
        "success":true
}
```

2.5 Access log upload

Brief description：Terminal upload the access log

Request address：api/devices/passRecord/addRecord

Request method：POST

Important Tips：In order to try to ensure that the server side does not miss the pass records due to network or downtime and other problems, the device has a synchronization flag for each record internally, the request parameter when uploading the record contains the local data table id, after the server side receives the record, it must return the data table id to the device, and the device modifies the synchronization flag after it receives the pass record. If the upload fails or no return data is received from the server side, the device will try to upload the record again. When the network condition is not good, it is recommended that the server side determines whether the record is duplicated or not.

Terminal request form sample：

```
{
    "id": 1,
    "devUserId": 1,
    "devUserDeptId": 1,
    "userName": "访客",
    "faceUrl": "", /**photo format is Base64*/upload
    "head": "", /**photo format is Base64*/upload
    "devSn": "YGKJ20204202773",
    "devName": "TR08A-A",
    "passageTime": "2020-10-23 11:16:20",
    "temperature": 3660,
    " facemask ":1，
    "atType":0，
    "remark":""
}
/**note：3660= 36.6*/ boday temperature
```

Request Parameter Description：

Filed name	Type	Description
id	Int	Device data table ID value of access log
devUserId	Int	User ID
devUserDeptId	Int	User department ID
userName	String	User name
faceUrl	String	User facial photo
devSn	String	Device no
devName	String	Device name
passageTime	Date	Access time
temperature	Int	Body temperature
facemask	Int	With or without mask
atType	Int	Attendance type, 0: no attendance function, 1: automatic, 2: manually punched in to work, 3: manually punched out of work, 4: manually punched out (the remark field is the reason for going out)
remark	String	Business trip or other
passStatus	Int	Passage Log state, save the state by bit, when it is 0, it means normal passage, other values have errors. Bit0:Face recognition did not match successfully; Bit1:Card number error, not matched successfully; Bit2:QR code error, not matched successfully; Bit3:Abnormal body temperature; Bit4:not wearing mask (when mask detection is turned on); Bit5:Failure of live body detection; Bit6:Personnel information is not within the validity period; Bit7:timeout for opening the door; Bit8: force open the door;

Server-side return sample：

```
{
    "code":0,
    "desc":"成功",
    "success":true,
    "data":{
        "params":{},
        "id":47 // Corresponds to the id in the request parameter, the id values must be the same
    }
}
```

3. MQTT subject descriptions

After the terminal registers to the server through Http(s), the server returns the connection method of the mqtt server, and then the face recognition terminal immediately connects to the Mqtt Broker and sets up the relevant reconnection mechanism.

Facial Terminal Subscription Subject (Publisher: Server, Subscriber: Facial Terminal)

Subject	Description
/_dispatch/_get_state/{device_sn}	Get subject status: the server side commands the face recognition terminal to send the device status.
/_dispatch/command/{device_sn}	Command Subject: The server sends commands to the face recognition terminal.

Note: {devices_sn} indicates the serial number of the current device.

Facial Terminal Publishing Subject(Publisher: Facial Terminal, Subscriber: Server)

Subject	Description
/_report/state	Subject status：terminal reports the status
/_report/received	Command execution status topic: the execution status of commands reported to the server by the terminal

/will	Will reports，example：
	{
	"sn": "", //device SN 号
	"version": "1.0", //app version
	"state": "OFFLINE", //status OFFLINE
	}

4. MQTT message description

4.1 Status subject (facial terminal equipment status reporting))

Terminal publishes the status of the device via the status topic /_report/state in the following three cases:

- The server publishes an update state message instruction to the /_dispatch/_get_state/{device_sn} topic, and the face recognition terminal receives the instruction and sends a state message to /_report/state, and the content of the update state message instruction is in the following format::

```
{
    "reply": "state"
}
```

- Terminal when starting application：
- Terminal starts the face recognition application every 30 seconds sends a status message to /_report/state with the following message content：

```
{
    "sn":"YGKJ2021DM0800003",//device serial no
    "version":"1.1.2",//APP version
    "state":"ONLINE",
    "userSpace", 6000,//user space，单位：MB
    "availableSpace", 5000//user remain space，单位：MB
}
```

4.2 Reporting of command subjects and results of command execution

The server sends commands to the terminal by posting a message to subject /_dispatch/command/{device_sn} with the following message content format：

```
{
    "id": 1,
    "feedbackUrl": null,
    "type":1,
    "operations":{},
    "devSn":"YGKJ2021DM0800003",
}
```

Command Message Field Descriptions

Field name	**Type**	**Decription**
id	Long	Command ID
feedbackUrl	String	The command executes the callback interface, and the reserved fields are temporarily empty

type	Int	command type, refer to the contents in the specific command.
operations	Object	Data carried by the command
devSn	String	Device serial number

After the terminal executes the command, it reports the command execution status to the server through the topic /_report/received or the Http(s) address of the feedbackUrl in the command, and the execution status message format is as follows：

```
{
    "operations": {
        "id":11,              //command ID
        "executeStatus": 1, //  execution status，1=success，2=failure
        "remark":""           //failure reason
    },
    "devSn": "YGKJ2021DM0800003"
}
```

Status Message Field Description：

Field Name	**Type**	**Description**
id	Long	ID of the command issued by the server
operations	Object	State-carrying data
devSn	String	Serial number

4.2.1 Add user

Brief description：Server side download user to the terminal

type = 3

Important tips：When adding users, because you need to download photos, extract feature values and operate the database, it consumes more time, so please wait for the equipment to return the operation result when calling this interface, and do not call the interface of adding, modifying, deleting and querying during this process. If you need to add a large number of people, it is recommended to add them in batches, with each batch not exceeding 100 people.

Server-to-subject /_dispatch/command/{device_sn} published message sample：

```
{
"devSn" : "YGKJ2021DM0800003",
"id" : 10,
"operations" : [{
        "createBy": "", /** creator */
        "createTime": 1602843134000, /**create time*/
        "deptId": 104, /**department Id*/
        "faceUrl":
        "Http(s)://192.168.20.168:8088/api/download/L3Byb2ZpbGUvdXBsb2FkLzIwMjAvMTAvMTYvMW
        Y0OTRmNDAtYWQ0Yi00YT
        MxLTg0MWUtZDRiN2I4MmMwYWFmLmpwZw%3D%3D", /**facial photo path*/
        "id": 10, /**user ID*/
        "sex": 0, /**sex 0 as：mail、1 as：female 2：unkown
        "status": 0, /**account status 0 as active、1 as deactivate
        "updateBy": "", /** modifier */
        "userCode": "003", /**user code*/
        "userName": "Test", /**user name*/
        "userPhone": "15575681394",/**user phone*/
        "cardNum": "ABCD",
        "wiegandNum":"100",
        "company":"",
        "department":"",
        "group":"",
        "remark":"",
        "expiry":""
     }
  ]
  "type" : 3
}
```

Command Parameter Description：

Filed	Type	Decription
userName	String	User name
userCode	String	User code
userPhone	String	User phone no
faceUrl	String	Facial photo:Base64 format
sex	String	Sex 0=male，1=female
status	Int	0:active，1:deactivated
devSn	String	Device number
cardNum	String	RFID card number
wiegandNum	String	Wiegand number， after successful user authentication, send the Weigand number through the Weigand interface
company	String	Company
department	String	department
group	String	group
remark	String	remark
expiry	String	Validition format： yyyy-MM-dd hh:mm:ss,yyyy-MM-dd hh:mm:ss Separated by a comma, the first is the start time of the expiration date and the second is the end time. If an expiration date is to be used, it must be set up in the complete format

After the terminal executes the completion of the command to increase the number of users through subject /_report/received returns the execution status, as shown in the following example：

```
{
    "devSn" : "YGKJ2021DM0800003",
    "operations" : {
    "executeStatus" : 1,
    "id" : 547,
    "remark" : "add user 1 person,success 1 person",
    "result" : [ {
        "code" : 0,     //0 added successful，less than 0 added failed
        "feature" : "", //facial feature，BASE64 format
        "userCode" : "",//user code
```

```
            "id" : 10031//user database ID
        } ]
    }
}
```

Return parameter description：

Field Name	Type	Decription
code	Int	Add person result, 0 add success, less than 0 add failure, -2: open photo failure, -3: already in face library, -4: insert database failure, -6: extract feature value failure.
feature	String	Base64 format face feature values, the server side can store the returned face feature values in the server-side database, and when sending down personnel to other devices, only the face feature values can be sent without sending the photos, so that the operation can improve the speed of sending down and reduce the amount of data, but it will lead to a lack of photos of the personnel on other devices.
userCode	String	User code
id	Int	User database ID

4.2.2 Modify user

Brief description： Service side modified user infomation

Type = 4

Server to subject /_dispatch/command/{device_sn} published message sample：

```
{
  "devSn" : "YGKJ2021DM0800003",
  "id" : 10,
  "operations" :{
          "deptId": 104, /**department Id*/
          "faceUrl":
          "Http(s)://192.168.20.168:8088/api/download/L3Byb2ZpbGUvdXBsb2F%3D%3D", /**人脸图片
```

```
        路径*/
        "id": 10, /**user ID*/
        "sex": 0, /**Sex 0 as：male、1 as：female2：unknown
        "status": 0, /**account status 0 as active、1 as deactivate
        "updateTime": 1602845520414, /**modified time*/
        "userCode": "003", /**user no
        "userName": "test", /**user name*/
        "userPhone": "15575681399",/**user phone number*/
        "cardNum": "ABCD",
        "wiegandNum":"100",
        "company":"",
        "department":"",
        "group":"",
        "remark":"",
        "expiry":""
    }
    "type" : 4
}
```

Command Parameter Description：same as added user

After the terminal executes and completes the modification of the user instruction，through subject /_report/received returns the execution status, as shown in the following example：

```
{
    "operations": {
        "id":11,                //command ID
        "executeStatus": 1, //  execution status，1=success，2=failed
        "remark":""             //failure reason
    },
    "devSn": "YGKJ2021DM0800003"
}
```

4.2.3 Delete user

Brief description：The server deletes the user information in the terminal。

Type = 5

Server to subject /_dispatch/command/{device_sn} published message sample：

```
{
      "devSn" : "YGKJ2021DM0800003",
      "id" : 546,
      "operations" : [ {
            "id" : 10031,
            "params" : { }
       }, {
            "id" : 10032,
            "params" : { }
      } ],
      "type" : 5
}
```

Command Parameter Description：

Filed	Type	Description
id	Int	User database ID

After the terminal finishes executing the command to delete the user, through subject /_report/received returns the execution status, as shown in the following example：

```
{
      "devSn" : "YGKJ2021DM0800003",
      "operations" : {
      "executeStatus" : 1,
      "id" : 546,
      "result" : [ {
            "code" : 0,//0 deletion，less than 0 deletion failed
            "id" : 10031//user ID
      }, {
            "code" : 0,
            "id" : 10032
      } ]
   }
}
```

4.2.4 Query users

Brief description：The server queries the information of the users inside the terminal, and the terminal returns the information of the users that meet the query conditions to the server after execution.

Type=1000

Server to subject /_dispatch/command/{device_sn} published message sample：

```
{
    "type":1000,
    "id":123,
    "devSn":"YGKJ2021DM0800003",
    "feedbackUrl":"",
    "operations":{
        "emp_id": "tyy",
        "keyword": "",
        "need_feature": false,
        "need_photo": false,
        "page_num": 1,
        "page_idx": 0
    }
}
```

Command Parameter Description：

Filed	**Type**	**mandatory field**	**Description**
feedbackUrl	String	N	If feedbackUrl is not empty, the query interface will upload the query results to this path via the The query interface will upload the query results to this path via the POST method, while the The query interface will upload the query results to this path via the POST method, and at the same time post the status of the command

			execution to the /_report/received subject successful or not. When feedbackUrl is empty, the query results will be uploaded to this path via the POST method. /_report/received subject
emp_id	String	N	User id, precise query
keyword	String	N	keyword fuzzy query for name and user number. When emp_id is not empty, the keyword is ignored and only the Execute precise query
need_feature	Bool	Y	Whether to return user characteristics
need_photo	Bool	Y	if or not it returns a photo, when the device does not have a photo, it will not return a photo even if the photo is true, it will not return
page_num	Int	N	Number of users per page received by the Center
page_idx	Int	N	Page start index received by the center, fuzzy query may have a large amount of data is available, the keyword can be left unchanged, but the page_idx increases each time page_num is increased until the device The center returns a failure to realize the paging query.For precision searches, page_num and page_idx have no significance

After the terminal executes the command query for users， through subject batt/_report/received returns the execution status, as shown in the following example：

：

```
{
    "devSn": "YGKJ2021DM0800003",
    "operations": {
        "executeStatus": 1,
        "id": 123,
        "users": {
            "id": 1,
            "cardNum": "",
            "company": "",
            "department": "",
            "group": "",
            "index": 0,
            "pin": "",
            "remark": "",
            "sex": 0,
            "status": 0,
            "total": 1,
            "type": 0,
            "userCode": "tyy",
            "userName": "luo",
            "userPhone": "",
            "wiegandNum": "",
            "feature": "",
            "photo": "",
            "syncStatus"： 0
        }
    }
}
```

Return parameter description： The users parameter (with the exception of feature and photo) is the same as the parameter in Add user.

Filed	Type	Description
feature	String	User facial features, BASE64 format
photo	String	User photo, BASE64 format
syncStatus	Int	User and server synchronization status flag, 0: synchronized, -1: deleted but not synchronized on the device side, 1: added but not synchronized on the device side, 2: modified but not synchronized on the device side.

Notes:

Whether to include the feature and photo fields is determined by the need_feature and need_photo in the command arguments;When using a fuzzy query, users is a json array and the precise query is a single json object.

4.2.5 Delete access log

Brief description： The server side deletes the access records of the user in the terminal, which can delete all the access logs or delete the access logs before a certain point in time.。

Type=1001

Server to subject /_dispatch/command/{device_sn} published message sample：

```
{
    "type":1001,
    "id":123,
    "devSn":"YGKJ2021DM0800003",
    "feedbackUrl":"",
    "operations":{
        "deleteAll":true
        "timeline":1621407072
    }
}
```

Request Parameters：

Filed Name	**Type**	**Description**
deleteAll	Bool	Whether to delete all access logs
timeline	Int	Timestamp, seconds since 1970, the device deletes passes prior to this timestamp.

Delete All and timeline only one parameter is required。

After the terminal executes the command delete users， through subject /_report/received returns the execution status, as shown in the following example：

```
:
{
    "operations": {
        "id":11,                  //command ID
        "executeStatus": 1, //  execution status，1=success，2=failed
        "remark":""               //reason of failed
    },
    "devSn": "YGKJ2021DM0800003"
}
```

4.2.6 Update configuration (parameterization/reading)

Brief description：The server sets or reads the system parameters of the terminal, and it can set all or part of the parameters, but the terminal returns all the current parameters (except the login password) to the server through the /_report/received subject.

Type=8

Server to subject /_dispatch/command/{device_sn} published message sample：：

```
{
    "devSn": "YGKJ2021DM0800003",
    "id": 1074,
    "type": 8,
    "operations": {
        "deviceName":"",
        "company":"",
        "devicePos":"",
        "showName":true,
        "miniWnd":0,
        "wiegandEnable":true,
        "rs485Enable":true,
        "rs485Baud":9600,
        "gateTimeout":0,
        "relayCtrlDuration":0,
        "relayDelayDuration":0,
        "doorSensorEnable":false,
        "doorBellEnable":false,
        "preventionEnable":false,
        "tempThreshold":37.4,
        "tempDistance":100,
        "thermalSensor":true,
        "saveRegPhoto":true,
        "attendance":false,
        "attendanceBtn":false,
        "businessTrip":[
            "reason 1",
            "reason 2"
        ]
        "adminPassword": "123456",
        "alarmDuration": 0,
        "alarmEnabled": true,
        "brightness": 100,
        "cardNumDecimal": true,
        "cardNumReverse": false,
        "deviceVolume": 100,
        "featureThreshold": 95,
        "idleTime": 10,
        "living": 0,
        "lowPower": true,
        "lowPowerMode": 1,
        "maskDetection": 0,
        "minSize": 140,
        "openDuration": 0,
        "passMethod": 1,
        "playVoice": false,
        "recogizeInterval": 12,
        "temperature": 1,
        "wiegandFmt": 26,
        "recgSuccessText": "welcome",
        "recgFailedText": "fail",
        "passScene": true,
        "passHeadPhoto": true,
        "recordStranger": true,
        "language": "zh_cn",
        "duplicatePhoto": false,
        "fillLight": 0,
        "recordLimitTime": 35,
        "recordLimitNumber": 10000,
        "redLightsBrightness": 12,
        "greenLightsBrightness": 2,
            "whiteLightsBrightness": 100
      },
}
```

Filed Name	Type	Description
deviceName	String	Device Name
company	String	Company Name
devicePos	String	Device position
showName	Bool	During the verification whether display the name
miniWnd	Int	Main interface small window display, 0: no display, 1: display registration photo and temperature measurement, 3: temperature measurement
wiegandEnable	Bool	Wiegand enable
rs485Enable	Bool	RS485 enable
rs485Baud	Int	RS485 [2400,4800,9600,19200,38400,57600,115200]
gateTimeout	Int	Door opening time
relayCtrlDuration	Int	Relay control period
relayDelayDuration	Int	Relay delay teim
doorSensorEnable	Bool	Door contact enable
doorBellEnable	Bool	Door bell enable
preventionEnable	Bool	Tamper proof enable
tempThreshold	Float	Body temperature alarm threshold
tempDistance	Int	Temperature measurement distance
saveRegPhoto	Bool	Whether or not to save the registration photo
attendance	Bool	T&A enable
businessTrip		T&A absence reason
attendanceBtn	Bool	T&A button
adminPassword	String	Device UI login password
brightness	Int	Screen Backlight Brightness [1,100]
deviceVolume	Int	Device Volume [0,100]
featureThreshold	Int	Facial Recognition Threshold
living	Int	Live body detection switch (0=off, 1=on)
recogizeInterval	Int	Repeat Recognition Interval, after a successful recognition, you need to wait for the recogizeInterval before you can recognize it again.
minSize	Int	Recognize minimum face size (pixels)
temperature	Int	Temperature detection switch (0=off, 1=on)
playVoice	Bool	Voice Announcement Prompt Switch
lowPower	Bool	Low power enable switch, when turned on, when the proximity sensor does not detect an object for a continuous idleTime time, the device turns off the fill light, IR light, NFC module, and stops the face detection function to reduce power consumption.

lowPowerMode	Int	Low power mode, bit0: screen saver mode, bit1: screen off mode;
idleTime	Int	Idle time before entering low power.
passMethod	Int	Passage, 0: face/card/QR code, 1: face + card
openDuration	Int	Relay release period
alarmEabled	Bool	Alarm function when the door is opened but not closed (requires access control hardware support so that the device can sense the open/closed state of the door)
alarmDuration	Int	Alarm duration
cardNumDecimal	Bool	decimal card number
cardNumReverse	Bool	inverted sequence card number
wiegandFmt	Int	Weigand format, 26/32/34. After verification, the device sends the user's Weigand number via Weigand intreface
recgSuccessText	String	The screen displays details when the verification is successful
recgFailedText	String	The screen displays details when the verification is failed
passScene	Bool	Save the verification photo
passHeadPhoto	Bool	Save the verification small facial photo
recordStranger	Bool	Save strangers photo (recognition failure)
language	String	System language, support zh_cn and english
duplicatePhoto	Bool	Whether or not the presence of identical or similar photographs is allowed when registering people, TRUE is allowed.
fillLight	Int	Fill light mode, 0: auto (determined by low power function), 1 always on, 2 always off
recordLimitTime	Int	Maximum retention time of the passage record in days. The equipment is checked once a day at 00:00 hours or when the equipment is started.
recordLimitNumber	Int	The maximum number of pass records to be saved, when the number of records reaches this value, each new record will delete the oldest record.
redLightsBrightness	Int	Brightness of the red fill light, value range 1~255.
greenLightsBrightness	Int	Brightness of the green fill light, value range 1~255.
whiteLightsBrightness	Int	Brightness of the white fill light, value range 1~255.

After the terminal executes the command update configuration， through /_report/received subject to server side returns all the current parameters (except the login password), and the returned parameter fields are the same as the command parameter fields.

4.2.7 Device Control

Brief description :The server can set the time of the terminal, restart the device, restart the face recognition application, open the door remotely, and upgrade the face recognition application through this interface.

Type = 9

Server to subject /_dispatch/command/{device_sn} published message

sample：

```
{
  "devSn" : "YGKJ2021DM0800003",
  "id" : 24,
  "operations" : {
          "devAction" : 1,
          "timeServer": "1.ntp.org",
          "timeZone": "",
          "ntp": true,
          "time": "2021-01-31 12:00:00",
          "apkUrl": "Http(s)://......",


     },
  "type" : 9
}
```

Time Set Example：

```
{
  "devSn" : "YGKJ2021DM0800003",
  "id" : 24,
  "operations" : {
          "devAction" : 1,
          "timeServer": "1.ntp.org",
          "timeZone": "",
          "ntp": true,
            "time": "2021-01-31 12:00:00"
        },
  "type" : 9
}
```

Reboot System Example：

```
{
  "devSn" : "YGKJ2021DM0800003",
  "id" : 24,
  "operations" : {
            "devAction" : 2
        },
  "type" : 9
}
```

Reboot System Example：

```
{
  "devSn" : "YGKJ2021DM0800003",
  "id" : 24,
  "operations" : {
            "devAction" : 3
        },
  "type" : 9
}
```

Remote Door Opening Example：

```
{
  "devSn" : "YGKJ2021DM0800003",
  "id" : 24,
  "operations" : {
            "devAction" : 4
        },
  "type" : 9
}
```

Update Application Example：

```
{
  "devSn" : "YGKJ2021DM0800003",
  "id" : 24,
  "operations" : {
            "devAction" : 5,
            "apkUrl": "Http(s)://......"
        },
  "type" : 9
}
```

Upload log example：

```
{
  "devSn" : "YGKJ2021DM0800003",
  "id" : 24,
    "feedbackUrl" : "Http(s)://......",//upload log file package（tar.gz）interface，method use PUT
  "operations" : {
          "devAction" : 10
      },
  "type" : 9
}
```

Command Parameter Description：

Field Name	Type	Description	Remark
devAction	Int	1 for time setting, 2 for: restarting the system, 3 for: restarting the software, 4 for: opening the door remotely, 5 for: updating the software, 10 for: uploading logs	
timeServer	String	NTP Server address	Only devAction=1vaild
timeZone	String	Timezone	Only devAction=1 vaild ， GMT+0:00~GMT+14:00 、 GMT-0:00~GMT-12:00
ntp	Bool	Enable ntp function	Only devAction=1valid
time	String	Date and time, in the format of "2021-01-31 12:00:00", with a space between the date and time.	Ony devAction=1and ntp=true valid
apkUrl	String	Terminal APP Upgrade Package Download	Only devAction=5 valid

After the terminal executes the command device control， through subject /_report/received the returned parameter fields are the same as the command parameter fields.

：

```
{
    "operations": {
          "id":11,                    //command ID
          "executeStatus": 1, //  execution status，1=success，2=failed
          "remark":""                //failed reason
    },
    "devSn": "YGKJ2021DM0800003"
}
```

4.2.8 download voice package

Brief description： The server modifies the prompting voice of the terminal, and operator can create their own prompting voice and send it down to the device to replace the voice in the device.

Type=1002

Server to subject /_dispatch/command/{device_sn} published message sample：

```
{
    "type":1002,
    "id":123,
    "devSn":"TR05BL-RV1109",
    "feedbackUrl":"",
    "operations":{
        "voiceType":0,
        "voiceData":"",
        "language":"english"
    }
}
```

Command Parameter Description：

Filed Name	**Type**	**Descrition**
voiceType	Int	0 Welcome voice; 1 Pass prompt; 2 Verification failure prompt; 3 Wear a mask prompt; 4 Swipe card prompt; 5 Scan QR code prompt; 6 Abnormal body temperature; 7 Initialization failure prompt
voiceData	String	BASE64 encoding of audio files (wav format)
language	String	The language to which the audio file belongs, set to the current language of the system for voice prompts if default.

Terminal saves the voice and then through the /_report/received subject to server side returns the execution status：

```
{
      "devSn" : " TR05BL-RV1109",
      "operations" : {
            "executeStatus" : 1,
            "id" : 123
      }
}
```

4.2.9 Obtaining photo feature values

Brief description： The server side can obtain the face feature value of the photo through this interface. The server side can use this function to obtain the feature value and store it in the server-side database, so that when sending a person to the device, it can send the feature value without sending the photo or the photo link, which improves the sending speed or reduces the data traffic.Type=1003

Server to subject /_dispatch/command/{device_sn} published message sample：

```
{
      "type":1003,
      "id":123,
      "devSn":" TR05BL-RV1109",
      "feedbackUrl":"",
      "operations":{
            "photo":""// BASE64 encoding of photo files (jpg format)
      }
}
```

Command Parameter Description：

Field Name	Type	Description
photo	String	BASE64 encoding of photo files (jpg format)

After the terminal extracts the face feature values，through/_report/received subject to server side return feature value or execution status：

```
{
        "devSn" : "YGKJ2021DM0800003",
        "operations" : {
                "executeStatus" : 1,
                "id" : 123,
                "result" : {
                        "feature" : ""//facial feature
                }
        }
}
```

Return parameter description：

Filed Name	Type	Description
feature	String	BASE64 encoding of facuak feature values

4.2.10 Detect whether the photo is already in the device's face base bank

Brief description：The server sends photos to the terminal, which extracts feature values and searches the local face base database to determine whether they have been added to the terminal's local face database, and then returns them to the server.

Type=1004

Server to subject /_dispatch/command/{device_sn} published message

sample：

：

```
{
    "type":1004,
    "id":123,
    "devSn":" TR05BL-RV1109",
    "feedbackUrl":"",
    "operations":{
        "photo":""// BASE64 encoding of photo files (jpg format)
    }
}
```

Command Parameter Description：

Field Name	Type	Description
photo	String	BASE64 encoding of photo files (jpg format)

Return data

```
{
    "devSn" : "YGKJ2021DM0800003",
    "operations" : {
        "executeStatus" : 2,
        "id" : 123
    }
}
```

Return parameter description：

Field Name	Type	Description
executeStatus	Int	1：exist，2：non-existent

4.2.11 Get un-upload access logs

Request data format：

```
{
    "type":1005,
    "id":123,
    "devSn":" TR05BL-RV1109",
    "feedbackUrl":"",
    "operations":{
    }
}
```

Return data{

```
    "devSn" : "YGKJ2021DM0800003",
    "operations" : {
        "executeStatus" : 1,
        "id" : 123,
        "result" : {
            "unsyncedCnt" : 0
        }
    }
}
```

4.2.12 Query for unsynchronized users

Brief description： In the case of network disconnection or network failure, the user operates the personnel pool of the terminal (such as adding personnel, modifying personnel, and deleting personnel) through the UI of the face recognition terminal, which will create the problem of unsynchronization with the server side. When the face recognition terminal comes back online and connects to the server, the server can execute this command once to check whether there are unsynchronized personnel. When there are unsynchronized personnel, the server must call "add passer", "modify passer" or "delete passer" in the MQTT interface as appropriate to make the server and the face recognition terminal synchronized. server and the face recognition terminal to synchronize.

Request data format：

```
{
    "type":1006,
    "id":123,
    "devSn":"TR05BL-RV1109",
    "feedbackUrl":"",
    "operations":{
    }
}
```

Return data：Same as query users(with photos and facial feature).

4.2.13 Delete all users

Brief description：The server can use this interface to quickly delete the user information in terminal, and the execution speed of this interface is faster than that of the aforementioned "Delete Passers" interface.

Note: This interface only deletes the users' information, and does not delete the access log.

Type=1007

Server to subject /_dispatch/command/{device_sn} published message sample：

```
{
    "type":1007,
    "id":123,
    "devSn":"TR05BL-RV1109",
    "feedbackUrl":"",
    "operations":{
    }
}
```

After the terminal execute the delete command through subject/_report/received returns the execution status，as shown in the following example：

```
{
    "operations": {
        "id":11,                //command ID
        "executeStatus": 1, //  execution status，1=success，2=failed
        "remark":""              //failed reason
    },
    "devSn": "YGKJ2021DM0800003"
}
```

4.2.14 Query Access Logs

Brief description：The server side queries the access logs in the database of the face recognition terminal through this interface. In general, the pass records have been uploaded to the server side through the new pass records in the HTTP(S) interface, but the server side can still verify the pass records in the server side through this interface. Note: Since mqtt broker may have limitations on the size of the message packet, the server side needs to set the query parameters accurately to prevent the message packet size from exceeding the limitation resulting in the server side not being able to receive the data.

Type=1008

Server to subject/_dispatch/command/{device_sn} published message

sample：

```
{
    "type":1008,
    "id":123,
    "devSn":"TR05BL-RV1109",
    "feedbackUrl":"",
    "operations":{
        "keyword":"",
        "startStamp":123,
        "endStamp":123
    }
}
```

Command Parameter Description：

Field Name	Type	Description
keyword	String	Search user name or number
startStamp	Int	Timestamp, seconds since 1970, search record start time
endStamp	Int	Timestamp, seconds since 1970, search record start time

4.2.15 Setting the screen saver picture

Brief description：The server side sets the customized screensaver picture of the terminal through this interface, and can set multiple pictures to realize the dynamic effect of cyclic switching. When the picture array is empty, the picture previously issued by the server side will be cleared and the default picture of the device will be used again.

Type=1100

Server to subject /_dispatch/command/{device_sn} published message sample：

```
{
    "type":1100,
    "id":123,
    "devSn":"TR05BL-RV1109",
    "feedbackUrl":"",
    "operations":[
        {
            "index":0,
            "pic":""
        },
        {
            "index":1,
            "pic":""
        }
    ]
}
```

Command Parameter Description:

Filed Name	Type	Descriptoin
index	Int	Picture number
pic	String	Base64 encoding of image data

After the terminal execute the command through subject /_report/received returns the execution status，as shown in the following example：

：

```
{
    "operations": {
        "id":11,            //command ID
        "executeStatus": 1, //  execution status，1=success，2=failed
        "remark":""         //failed reason
    },
    "devSn": "YGKJ2021DM0800003"
}
```

4.2.16 SSL certificates download

Brief description: The platform issues ssl certificates for Http(s)s and mqtt through this interface, restart the application after issuance, and the application will connect to the platform through ssl. ；

Type=1101

```
{
"devSn" : "YGKJ2021DM0800003",
"id" : 12,
"operations" : {
```

"stamp":"2022-08-12T09:06:22",
"certs":"I7aL5Gh43ggr0+9u9MYGVfhghmJ5AzE00kB5jFQ2uPjygSfppvXbZlaTZBfcNPfbRgab54xy9VID0Iafdxc9dju
Zlq+SG6bZ/RuAHnLti3SSzN91WAtkCM5dOuIs/izsGyxKZMqpyXgIqPXi9ekKqqzMwkU5zxZsVPFSNcZxLtEmoQ9+Sjva8U
IrRrgEH/rQ/cTjYgqFhJ2I0JpoINtGCSndQTQaj/B9NNveRVukWY5QpAkwt6Nf4xcktx9rgVwF11dJHiiFuzbZkdUhp3rIkZiG1qj
GBMUCoxlfhuklldkZts+NN96FZ7pZC1Ak0X9517JmSTHxcBzJZ+FIelx7FoExBj/xYu1g1nhZA/oIdAlRqfl/L3/YgzVFYb5Aavd
7pk6JeYphMUNQ+sR8EQrnx6fW7FwN8PfNyoneOyromgcFlb7xuVGHH7sK3CArtZywAJFW8ip5i+RtihxD8u+bzmFcPBR1
zpZR9NJGd+qLIj7edj1E4dPoHCH2LYfKULfF0WxiOdvo2Ai96PwXIONKTQyyC6vBaZUkAIbau7ugJzPvpPFx+jloym6+R60
nMbX451Oq8HBDen0LIOBviKoY7XIO+sBlJJtUsz82QzMisirLRHjvDxMHCm76ykNkjoqebc4P4YPYScuXyYfcIKm3SMWHf
Ypgg/+TKNCJJ45ASjYjKJ6iLv2qkXtsdOwT/Rai0PMhZRABIIHDZqo56jPGsLrKeA7wrkZhKig7Y7gQjkRppV1ZkRtHMkIH1S
izSxZN8ZP8ttivH6Qv668CR0U0Gx0INKwnoYmkzDSFn7wchKYMkoJh71z3KiyEYsLthPAMYz16DTRLXwIrQHnREkDBBR
g9ythkHdA/G1MREvDT9oH7QaXcHXfDkFqqbF93H+vUHI8AKT2uwUYLPitSa+WzoA7J1k4MlPtgHdKRcRHQqARHEqxM
E68aAwwPE28XNqlSOiHmhMCOV24F61gGvcCec9t0/RIvjsPKJeTcLGkbyMuk5HpsIlsQEhnVwH1dVLsgQV8VXyqrmyscB
AVBKNyJ0DpJOYFTv24ieGSJtKqQRzN/LUmOqyoMhSA+0uoFbC4DbbypJFZ28DlsBwM224TSXe5hZOSPuP8EEXXrnY
RddBKO98Okq+s+9XYPlMCvikUwrfdDRT6/yKnQ08xiV/qPicIHlowJdfya/wmbrvp1l/ulWoTQA4T91E1G6BQjzZEaKCEVwP
E/8pI/qhZieP7zXhrI3beG6ceeobABKG7eNIsrNX/6+pySLINMVBCUQQH7QJxryzuQSucaSd/Mim5XkeQQW4VCENMaRP
RJFBYGTEBQhW4pxbp/hcbS8U3vRK6wVd8od2bCFCLJr9xkhLHPqlJtNGa3uD+RRQjbXsVafe/r4nYqK0IjSf6NawyKm2fz
YswRefy1Ok+yjaJ7hagtoO4SFwnaMVaXTzjw42MAOwiCWN378Spbwe3o0LsfNc+IO80HTt22Jyg9cU//o3rP4iNTDjWeh4/q
+bpUuEQJ8FdzBwYUobvowbWamY1NO4QU1pe56V/rGbCW9DKfFKkHH7f+o8i5yfv1uQ81e1DOQ12sqTY5K6gqLfSbLC
CPjOJOt8CdanLrxxTQSyZEPkY66JiyqxEUHf7qX6Rpzmof5d9Wf2W6eY5skr7CCA6uk+nzQGhnSQP4N/9YBxdkQ1wJIRr
hyV1wKD0NwtdrAnu6btr+TE+Xnv0+QGxBHfxqGb1i1GIDyGWYCRmIswQrxKfLje+qNsmBdYsVLnpSD09mS9g0wrl2z6uR
S5KuFzAM74st0OiohyhHwG89Sv4UA5lQkyMBUff5gb6rC6VWviuodiXEyVh9BiyVJ6gqtuCnYj3NTRVv03qBXG++IOcpow3
2taC6V+eo6GqhjrZes5LDzALSHvyBxWatjyRxWJMxg04Ks/6V0gIeohM3nixHtzwsMkXAghJRT8c39TgdG0qkk9Gblapjhrk6
QvvEjBNb40scLT/pH0fk0ja2C1t9XmXwSTXgOSfbgz43Z7yGouZwW5HNrFMKtAGr3ZUqaHq0sVNHmKww4qc+GOSWM0
VV2C/pTd65B9r6MUZYwUGLaBvquk+UCAdmcNnj5dSIZQ6GGaukcMkTLzZlW3GW1gKtRKUIK14Wdp/H+TAXqDb74Gu
P463FJBSUAxtS4gYHBdfmcIXNaXkV2CiCTbGRX7q1RS1IiVJU4peR3xv4Wy0zSrOEsqwPBiaFOllCreV20VavYxMHCPe1
P1yyx9ZNRU14kVr5aKcSHs5xaAxjgzNoGKKFpkn/9uCktE3e/o4zBXmWJcJmCI+rKVpK/VBj2qBIVfKJHwADYR+XQ6dlVP
Tg5JCOmvupzVbTGLVKbJp1tqSZp+BhOXC1qx6zmaSBdvrZ9DupxptPoR/+YeKn473yuCq7ho3HIHvf4KEcyuR3YG0vl4lg
x8TCmzRCAMjL5q+u7lwuCtB9X7wBejOQXp4VEIJRo43TJFhIGPVRUIDr+ewZQoM/62JqKqfvS6UO1Hdxp29md/XRJLTM
x+wtu1oCwrUar98lFw/QQ2IpDdI03QCTtmKRcgywfsMhSmTL/DfjtshuaY2PG/elxzgll1dGZugqKhb1sU1PX16z4vdkd5rjFst
mdMPZM2MexaQp7KFnWNF3pr4zuGkrsKwXCa4wQTwCfsyXRCav/vd973Aq31Im4DOcbiax6dxMUdkQhAT3/ZezsI/V1V4
4BJ2QRg6xGh7Xk9IUS7QXSKkW59GnGuCvAInBBEB/MN0VIF4N81ILYkvdEc7NadxF+KCksn4+3Uh6ZQQ2YbVhu+9ev
RVGBvhCoM5snjwlb5wJRNcpqrn3V/DUeUPvPzce7mYvf98jvtiUypU6rSC+y57JyGLIMZdbjIHDw490lr9ly5uXJSuCEG+TbJ
dFHo9MMMofQcrQtxrPbStrlDxe/i5Ec44PtppgVv4mTOIdves4SsOpC/H+LDEVJnjnnJXsvIq83325pe9qF+UcS3o0oxLcjT0x
CLRB8SPhcB6+/DIfZstcQabVLQyJBSxA/XznAzYJbSLIE5bHy7SgWE7R6vr8AYvTGtreQ2S5z5BF2UMZ7TYOdJaivxDUR
Vd+zVTBBxEtECHjkpihvvo+QYzmV5tOdDAsruFnm/DVwInNXUuGGfgnqLKVywjLYHt2noqHetbPqnlktqbPrAI5nkr9Fgqsdh
UI12/mHOsHXL2ycnqAB1/6wSB0Uh1/F6fgKWxKZp02gbVV6c85PGjxVxcu2uT+WN+aX9q5dcPe1UXIC7lAGpj9Otx4FPVv
TtWTroUYU7Wn5WhQJed7ENstB6KUa7V/TgXvrUpZJEVMBSmBjUpRurKyIxzNr+keJPewassLXvXR5/DDQndmJ8KcECn
gjiLzYpUjrCrXH+msjd7rdRmRhTCcl9i1B+kjeQls3GZgb4Ihk8A61N9yiXUdklcyznoCnOyJgI6pJTrjDUosqiXp+TbBCl9z2/joS
bdk2do+ID18kGZKyAp6UxeVwbl+sfKE+1uUPMztguIRQrkij8q15UKFOS0K1j4WoEUSRjr37ihRLUVGQCaPNLTueUhddkQj
4TNcw0AnC1gBbUQyhAdYcapixE+AmTR5EFkNdRoFymeQcqhfSegg2MOmA/oOQoFv8J895aMD9ygGBFzEiqWWLjizkd
0esMIqPc3MB58ilAbtYp+5Mvoc79Jva3xsGal7nrKoYa5Ylg2IO5D1WfYz5BDK8hjaeLLmtD2P+166ymK0f1iYXhoRClHcG6v
xun+8BpNn2yUYLRpN6bkpPmzwSNLBF3ZC0qZ+ZJ+J1SYdvnPZPu0J2MzEC7Ppa8UyjDGuq8yXEGK+FAdjWbblEfdisrF
6hJs1MHKDFAIM6D9VRPKIrXiq8aRuCDUiJXBFAXlFgj+Uo7KLrthkwjkbAN6xPVxAatnNir/lPta3zehEkTIykhE848JEDWBvi
saU6KZmjhUMo8Cwtf3UFLPaFNc1xyt5/O9tUz5AbrzM9FadK0WDimWZER9rns2n8y7DdGRBj61iDr+QSScjKib/8VYBa0

PxLV2SoFGpoea+e+IOzttKi83QrYjGT1luPFmAJAebH6gzI0HS89oGplxU5O2zOF1hudZoQy9TTTRLb7eSgYAKmvtJSe8S
HO5Gcj5CDHT0BAzD6XZEOJ+evdmGrwyvMo7z65Dt4Y3KCtlrLk6t5iAuoWtxq97VASODs8HVq0cLskisSPlt6S+CUKPhS
Miz/wwK+u44fw6WF3xyDSKpc65NNDY7+70K/vs406ymFytnK56cJkEyYdkkL/LJuzwz2vUKYcaRLKXbtT9Bsq8Hva/vhE/E
nXZJjtmR9QrtIRC7ikmhLb2mmf8Nh24FrcLVDK2TU83LedLFC9j3GbDqQcffs798fW7RSQWTcBefpDanR5DI0Qg/ni0J+wK
euBg3NgOUNC5IXdfCPiG9DYR0GjLQVrUmJ0KCvLhCEiRFoSaliLeI08sydJyb1GK4YredcjxiRjpQB12x6YaPeO4nlMNFGA
tPqBGEfyWTjOR8a5MIS7sqJBZm3cLkgZlwTi88ZytMe5NuBw+sSqq1DXj4qQia1hRUleSItN/zvZRKRjlHLFWFFHyuT9v8/V
tz9+AiJb/GYOZj0EUR5/MgW+ovatPwhtclIxmfca3CvPc6RHxB5qy4PM2cPaYH5Cwk7JfnVRXEdSRrzYI2ThzNXhjpnFHJTt
fWuZ5IPCq5l0ec5V0ZaAi49oq3YOy+MxVFI1FSp/I13jOWc0+bWq5k2URt8dW7naX+jM3++5bxInDKrHY7Q/1Ou5pSRo98
YxMaSLWA0vKiNGLj/70N9g3a136Y/qh/naVAVvDKIzQ6VMX+rBSVWMCvi7uN3+VXqrJycQXiRigpfcZ2AKlYwtBpuiH2BfgL
mG8zwIxzly46Ynr8lWG5bKR31Ah9qT+5w0zTxzym2Y17KU2A5FsH5jynCUstJQJ4/rtGcUARIR9TmRL7zJIBZax6xR6RSE
aPRwoWRw3qIGfpJYzKVcESZqq2mcbOlUSTxigAgeiruemD4/1gLgf0H/2I00CNjv49GKmI6Yv7cN7NG1zUkRn4zz6caVbzn
eL7/6FF4P7Wtc50rKAMDK/31kEVwc3wlj/uB1yGP83t4N3aDGSiYUnnJyN1SRZHjXVgKoznBYmQorPLZLER+P2uhfcDkU
sbMgg8hltYNq26kyyGL+lCcb3UGIvo3j9vOBpk1tXUOJ/DsVYvZlhP8hQHThzUGN/gJj/kB9ZMLl1xPntz5VP9KRx0LI7vUZo
TxTXdTsgXEUqVEjVZXyxvJZbMYD9juKN6Zckm5+sZWf+NJ5qosKFS2Wx1wwA2fOz7n7OGuu79+0+bDg8US/wAnt4hkyf
vTmK/ossTcJsgcyMPmEsiMaFkmd6FAv4A0AOe5jnidis15DM5hTv4OJUZwi5qP33r3vihpngKiGi8qyq+qIHc1Zn2fKHnV4lJs
qK106v96ggBnRpSMOpTZ/kUv1ILn5Nh5AonpnZW3fVERRCELnnycbp32JL7na3wXbwq8nvqPu5FUdpSMhY3qibEdjlXs+j
QyoQlNyv3FJGT2VjzvPSIiSR5OjioKZIOlyBgkHiZVkO5aHr8FsJK1a7/C9j59zPdvJEe/H9faZQ7POG5tiW54U5bZ8Hi1Zuw
Qdbdu1uqYCRN6FSAQxIXPsC2OM2bBXz66oNsNkMXM7Q18A6C7THuFZxxukn7zBaqGj/nHKnNOPcTxa0Ldy9SkMjDB
WRBCgeyDSY182UmtOgQeinCospSgOH5Fcmyb8hrcOOxEhvXatlVPWS5YJ1+DytJhm03fhj+dLeqcXfNvXj63aL4XljGsxlB
SFybaZkXVbQGeTtd1b8FhlK8Y7Fj13nj5gB8LZ7bXz1yu62fv1tC/94hNJnqk53AbFMKw/xbDoeGAwcxb2s6vQth5b5up6OB
KuwTVwP/N/NIZ0LiXSfjKLMEs3zm9mhrR0BLyO7DhcztFMaf3Lm+9x9/YkOK1KBRg0MrPbt/+w3RtlNz+EXmhBFbY2irjDZ
KFz59L4FqvVZnCOeq1+j+FKyhBp496/w7vEetKdJ2b98GxUgfbVCcrOiAuOiNV+oeciv9/Xk04harPFHtCadzzu7bEY1U6XL/
4YJ19jirP5MPnVn1QlabPl3P7hqrdjm1p492ocBSEyU1RwjKQVcMfW6+xJkJmecZPu4slkz/KwnMJVHfXXq7M0cNopYgbfn/
XtHqJWJFLygF9vU10BQfCJ95umnKHpiMjhOn+Za14EF3WsAkJos8MPfuyQHABry9O8lhGX/w2x5owej0UOrmMoexYhM0
+h2c8j9i1hATyUjhxs5yRkpiso15KP9xTZJV2nMbUb3hVmgGyejJc0NwoxLid6k8L3epcsmAXqI2yq49l9B1rIfiHv6/E+dYyJ+T
T31eJHSUlhwOYNzx5wA61+xITMwyLEhag6jw1ApFnxSFHY21L5yPh1VxZnDeuva0AJHCOI+q5xoOjZdbnqAEg5+sNltlPt
T6f+ljyC8wVfKeJaCG1batMT92d0fCq++MitM7tdmO5b2YkgOLPgQ9UDAHstC0M/uu1wpZsXjDYxYgYOUcOKhMUJK4kzz
WTALRMG+e7hzVw1funWIEOIA3s48AJEkfJrMnVep7KHsBDvyoZ59KLmB1bk+O09uQDUQOS8cxep09C5q9f23t32LYojU
Jr6h1Q68+UlsZgmcRM2niUKFznQ3yRjif/Azb1IbWloN4pqah67l9S9nGvOLA5MPfjsuJuQBDNnekQMPTQSM/f2Uu87URM
KoWiHu8+0s6uBTRR1KsnbnuizJNDT7IVu7S2qSHBNNRFEbLN+muqVMPSOkjv3S1PgMfJTzg/h/XfddlGwjXmAbr1Gewf5
zFgg/O1adCXoGr1HvJMnUYfpXzOK0OAvpUoF5lSw2YTpL6CZXsMAVY3IsNnZ8g0GNB4BhRRtGbRr/yaqJiwzIkbu+vZhL
qIzghdlLPaXBdwKNSa4UpZSYEpaucu/VfplVo6182akWfMJVoSMZgK+FC5XPANmGdG1VO38FMGDU5Zrs/dyqkobbKlr9i
UYDKskZCyPMS/pDGt0dANGV6SQDYOQmGuWaiBqoCfvpqI2emcNxJaahvYxPK4i7chHyrNiDY/EsM9nOdbPrgQnVTz1
O6DOKQKxbvmv6QTU1iSXdQ4EX2di/yG3lrt1gxQuYHZwh7/6jyuWJ9oVWFc/tDnXgCuy5ouF3GHGy0USdtS0NSBPuGA
w7LPfvxQ2lDG4DbwhF/St4kTyybzzqpC/r7ggUTVXxoijvMLRPsq6cCbpQR5Hdg38h9EVctTEwWLq92JKkUiepJEJ5dUsWp
fPaAE6U0kwAHTtufoej7k/6fy37ID1Xao2X5z7efGesKV96zA/m0f0FsfhWkIZDzXJfIEKMMbqUb+qGXGfBssTt6/nTP6Bvqsh
9RWKHYBRYxPrNYAkg+Lj8CcbnLJSU+GOZWGNfkMS7rqpPWtwVrKqWL9LsSg2rWT+zbxpzroAIczVY+DNAdGIpdvwX0
FXZ9JMSg/tAPRGp4gUPyQDUwD2By292EUnGQgtJQAIeZHY/LCBc4QUr9A75Ep7qaci36VGiLzt9Owj3UurmcjmWPoW7j
zARoy153vgzX++s4YNbkuTxIUYCAuJitU8ckBCsEJSvZKjc2Y8y5YYyT8bFaY3t0Hm0yeyGZIpvDRY6uAAbwCCXA+pCK7
dbA7H/4Dy+9odX7O66TUBrR4HzPUF1wecp3hNV+8VTBsUQoAvibHS5qGL/gS4Ama3HRF53KU+2lpWxSbtla4hFE29fu
WA0xWg6K1ck3gkAetvo4LCknMDaZfz41TXqc9b93cAiS2yHlrvS0ESjoEgDqRYOxb+dkDg1JWRDEp2Y2jyY5vfRn4GzxJL
5Tu4r+onPUwpUgYOfbUCfjWf1+d86r+Iq/8o9/5LMWhqLmFqeKGudagzI4talUrp6nNE5X4TlmIB0KPay8wE40smcQlcXqHE
wct1wzlyxu9hikjJx8XmiYNvvxT8bvzquqUdTae2DeK09/Zx8xro1Y/Gg5RyuIS7g0yMKBmmLYmysFQ0HK6CdDcsYzdRNDx
zw8BiVyN2sOA9xazXVuCaijx2nP7EfYrOh1PKT/jmx/tnk6qZR6Xz5kTFLAXLZiTNSAWNN6SgRy4NspS1GWjpT1FWEvpqo

```
H9Ml8fOwYHCHrtjvus19oNXE8ogtLsEx5mMXHw8eRYWgyX1VmamDumtD/6zcTGu22sAaGgrTGc2B0AUcctiXOgzTykDy
qzTvKACfOrZeqpTZ5djG0QPcvA6U3FCr8l+HzIGKDWRknEs8vGFFPEoXApo11j4mF85Fv3Gzq0zT23WRmnmOOR2cGx
8kM8Hu04jffVeeUUBbmSOa6PdXumGkUipIB2z3810Mf6Q82W8Q5VKK4iErfq3lY0zN4A2j/vrcnLuIJkOLA28I8tdc9Zk7bsc
Uj1C7b0fJYVg4jtLhwdsIwTJ22oF//egrXSRRKK9VgefLwE+Mw3/TZEhpMXkU0hL68yU7JPXgCI9+drStA20Hf3rb+YznaqU
Cavrggcc7aOk1FTw++p7+rT9XDnZorqAi3twldUSyrWZJTl+/l64x2Uu+fYvicptA5YUnhgVOoTCPHmRjXy0mULHjHZQUFm
QoYWRoQ7JeBqPWalOmZK2V4rfAuUmp5NCM1UU6eblICbA9D9Rhf9lslkvsUywDPs/Ox0JCWimwLDarIBrGtfoh4faDp2g
1gPhcYTbb/KRYVZoxE80crbExlPzJLfTUSvC+4L/t02lXiX0pN8OQGHhcCee2HKL7JEV1UKAmkmapKpE2upoqlw21arIxyc
5INTthZ4urnwPYZBK+2q/V0kHojFTU/+VSK1bzcfLMVEgZkAKN6V0h/T5KE9P5+0VVgXnfNyFsP9TWHu5jddVmfCKRW2
QCp6tfrboBUEt7JbGGn5+XkM4wsoUoXRWuE0E9rTdIU0auXip4rBF3hjtcQAKotu7UxQQFGKyI9Z5pUeH3/+4waX+DL8S
XJWaJk2RYmj8ykFfixUmnJWG/XlTPe6E+cjbrSjXpl6B5FRXSoTBiz/0ULllcsFCG6v1NBc22v6uGRlw8J99tTxgqPhvlTAF9K
aKdzxKyktCGR7mwlejDAFJM9Dg+4U3DnAODbrU/3d6Ri4orZaAyF7/TM+NLIwXEXmSRaOjBVhSXkKao9s1n4L+NVEGD
dfMukbkdBsF2J6Xi/XE3PX/PxnoPvXrWyC3UzZR8h1IO1xk0rjhZmhJswd/JNnc0LV1TuN8wbJ/xhDWkmOiHZ4ZNtxTTgqko
sPymnBBlwEMruj4ZZGxfdY4aJHmz3N+VdseeH5TTDYJo770ViLsotCeTIQDLQXGmcEZwSKFqdc1v9XG7ydRohP7k0jj5d
cDug8MzUqUoGdqSUbdTQwRFsUpqNSsm+wUh2/XmbwRUWsnqPoIO965YNKjS5DOEjHvvnvo35/O2oBSXR8m1vVSxA
H66IGPm8xVztb/pcXiFfUnL3yDmQaKeouZw/QoeIbdVbkumwEfg0/Q7pycCtk8s1sEJ5Zf4ngpOp/dZHOvtlocQVBX4rQgfZ7
W3bECNNeqV+BN2qRXHKK3knsIz0WdXOgEQ4ToIqb7GCUq84ejQqNIxIE5qENUNdW8ZdeReEFyiLV/pk1LDVI3EOXbIR
XAsbYw5DTApVzBdRMjjekw3AdD5ktRTwsNNQ3l4F6E3RZb9TZNtTMWmtPhe9WyNtOICrR73oP7lNdtg9rO9l6OpDchTB
sND39nvAeSuO2MqWBJ5fZO6L2tCARxpnHMGt9g2cXD2wPjOSlej360j7b0ryyS2XJIoQgZSMi6cxbrz44Sa90ZUGNgB05h
58onXTqo6d+VAsuOGznILdKjFIvIT5N2rDX2GZJJlDIP3tlQd"
   } ,
"type" : 1101
}
```

Command Parameter Description：

Filed Name	Type	Description
stamp	String	Certs the timestamp of the time when the encryption was performed, and abandons the execution of this instruction when the difference between the timestamp and the system time of the device exceeds a certain period of time；

Certs	String	Https and mqtt certificates encrypted base64 encoded strings, the original unencrypted string is as follows:
		{
		"Https_host": "",
		"Https_cacert": "",
		"Https_client-cert": "",
		"Https_client-key": "",
		"mqtt_cacert": "",
		"mqtt_client-cert": "",
		"mqtt_client-key": ""
		}
		The certificate is PEM encoded and converted to a single line string by deleting the start line -----BEGIN xxxxxx----- and the end line -----END xxxxx----- and deleting the carriage return line feed.

		Encryption: AES, mode Cbc, padding pkcs7, bits 256, offset iv (16 bytes): I*!7Pr*WAeH5JB7g AES key (32 bytes) consists of: 1, fixed part (8 bytes): yecontek 2, the last 8 bytes of the device serial number, such as the serial number is less than 8 bytes, the left side of the complement '0', such as the length of the serial number is more than 8 bytes, only take the right side of the 8 bytes. 3、Time stamp part (8 bytes), take the right 8 bytes in the stamp field. 4, the administrator password of the device (8 bytes): if the password is less than 8 bytes, supplement '0' on the left side, if the length of the password is more than 8 bytes, only take the right side 8 bytes. AES key example: serial number YGKJ2021DM0800003, timestamp 2022-08-12T09:06:22, administrator password 123456, then the AES key is yecontekM080000309:06:2200123456 AES encrypted authentication site: Http(s)s://www.mklab.cn/utils/aes

4.2.17 RGB camera parameter setting

Brief description： The server sets the parameters of the RGB camera through this interface, and you need to restart the application to take effect after modifying the parameters.

Type=1102

Server to subject /_dispatch/command/{device_sn} published message

sample：

```
{
    ″type″:1102,
    ″id″:123,
    ″devSn″:″YGKJ2021DM0800003″,
    ″feedbackUrl″:″″,
    ″operations″:{
        ″antiFlicker″ : 50,
        ″fps″ : 25
```

Parameter description：

Filed Name	Type	Description
antiFlicker	int	Anti-flicker, values: 0, 50, 60.(optional) 0: Turn off anti-flicker; 50: external frequency 50Hz;60：外部频率 60Hz；
fps	Int	Camera frame rate, take the value 10, 15, 20, 25, 30, the specific value is related to the camera module, please confirm before setting. (Optional)
hdr	Boolean	Whether to enable HDR function; (optional)

The device returns the following data：

```
{
"devSn" : "YGKJ2021DM0800003",
"operations" : {
"executeStatus" : 1,
"id" : 123,
"result" : {
"antiFlicker" : 50,
"fps" : 25,
"hdr" : true
      },
"type" : 1102
   }
}
```

4.2.18 Get RFID module information

Brief description：The server gets the type of RFID module in the device and the module firmware version through this interface.。

Type=1103

Server to subject/_dispatch/command/{device_sn} published message

sample：

```
{
   "devSn" : "YGKJ2021DM0800003",
   "id" : 1074,
   "type" : 1103,
   "operations" : {

   }
}
```

Device to /_report/received example of returned data is as follows：

```
{
   "devSn" : "YGKJ2021DM0800003",
   "operations" : {
      "executeStatus" : 1,
      "id" : 1074,
      "remark" : "success",
      "result" : {
         "fwVer" : 39,
         "moduleType" : 0
      },
      "type" : 1103
   }
}
```

Return parameter description：

Filed Name	**Type**	**Description**
moduleType	Int	RFID Module type，0：Mifare Classic，1：Legic MRD；
fwVer	Int	Firmware version

4.2.19 Setting Mifare Classic Parameters

Brief description：The server gets the type of Mifare classic reader module in the device and the module firmware version through this interface.

Type=1104

Server to subject /_dispatch/command/{device_sn} published message sample：：

```
{
        "devSn": "YGKJ2021DM0800003",
        "id": 1074,
        "type": 1104,
        "operations": {
                "dekeMifareKeyA":"656667686970",
                "dekeMifareKeyB":"ffffffffffff",
                "dekeMifareKeyType":0,
                "dekeMifareMethod":0,
                "dekeMifareBlockAddr":4,
                "dekeMifareBlockReadFrom":0,
                "dekeMifareBlockReadLen":4
        }
}
```

Parameter description：

Filed Name	**Type**	**Description**
dekeMifareMethod	Int	Read mode, 0: read UID only, 1: read content from M1's block.
dekeMifareKeyA	String	A key to be verified when reading the block, a string in HEX format.
dekeMifareKeyB	String	The B-key to be verified when reading the block, a string in HEX format.
dekeMifareKeyType	Int	Choose to verify key A or key B when reading blocks, 0: verify key A, 1: verify key B
dekeMifareBlockAddr	Int	The address/index of the block being read when a read block is selected;

dekeMifareBlockReadFrom	Int	When reading a block, data with a length of dekeMifareBlockReadLen from dekeMifareBlockReadFrom is extracted from the 16-byte block data as the valid read block data.
dekeMifareBlockReadLen	Int	

Device to /_report/received return data sample：

```
{
   "devSn" : "YGKJ2021DM0800003",
   "operations" : {
      "executeStatus" : 1,
      "id" : 1074,
      "remark" : "success",
      "result" : {
         "dekeMifareKeyA" : 0,
         "dekeMifareKeyB" : 0,
         "dekeMifareKeyType" : 0
      },
      "type" : 1104
   }
}
```

4.2.20 Setting LEGIC RFID module Parameter

Brief description: The server sets the card reading parameters of LEGIC module through this interface, the parameters of this interface should be read in conjunction with the IIC communication protocol of LEGIC module; the fields sent down do not need to contain all the fields, but only need to contain the fields to be modified;

Type=1107

Serve to subject /_dispatch/command/{device_sn} published message sample：:

```
{
        "devSn": "YGKJ2021DM0800003",
        "id": 25,
        "operations": {
                "14443a_4_read_at": 0,
                "14443a_7_read_at": 3,
                "14443a_uid_len": 4,
                "14443a_uid_seq": 1,
                "card_type": 7,
                "cpu_comm_mode": 0,
                "cpu_crypto_alg": 0,
                "cpu_data_fmt": 0,
                "cpu_efid" : "abcd",
                "cpu_fid" : "fedc",
                "cpu_key": "00000000000000000000000000000000",
                "cpu_key_num": 0,
                "cpu_read_at": 0,
                "cpu_read_len": 4,
                "desfire_aid": "000000",
                "desfire_comm_mode": 0,
```

```
                "desfire_crypto_alg": 3,
                "desfire_data_fmt": 0,
                "desfire_fid": "0000",
                "desfire_key":
"000000000000000000000000000000000000000000000000",
                "desfire_key_num": 0,
                "desfire_read_at": 0,
                "desfire_read_len": 4,
                "hid_iclass_read_at": 4,
                "hid_iclass_read_len": 4,
                "hid_iclass_uid_seq": 1,
                "legic_14443a_data_fmt": 0,
                "legic_14443a_read_at": 0,
                "legic_14443a_read_len": 4,
                "legic_15693_data_fmt": 0,
                "legic_15693_read_at": 0,
                "legic_15693_read_len": 4,
                "legic_15693_uid_seq": 1,
                "legic_prime_data_fmt": 0,
                "legic_prime_read_at": 0,
                "legic_prime_read_len": 4,
                "legic_prime_uid_seq": 1,
```

Parameter description：

Filed Name	Type	Description
card_type	Int	Byte[0], Supported Card Types
read_mode	Int	Byte[1], Read UID or segment
legic_sam_len	Int	Byte[2], LEGIC (SAM) Length
legic_sam	String	Byte[3~14], LEGIC Token Value, 12 bytes, required for LEGIC card read segment
legic_sec	Int	Byte[15], LEGIC section number
legic_prime_uid_seq	Int	Byte[16], Legic prime uid sequences
legic_prime_read_at	Int	Byte[17], Legic prime read start byte
legic_prime_read_len	Int	Byte[18], Legic prime card read byte length
legic_prime_data_fmt	Int	Byte[19], Legic prime card data format
legic_14443a_read_at	Int	Byte[20], Legic 14443A read start byte from [0].
legic_14443a_read_len	Int	Byte[21], Legic 14443A card read byte length
legic_14443a_data_fmt	Int	Byte[22], Legic 14443A card data format
legic_15693_uid_seq	Int	Byte[23], Legic 15693 uid sequences
legic_15693_read_at	Int	Byte[24], Legic 15693 read start byte
legic_15693_read_len	Int	Byte[25], Legic 15693 card read byte length
legic_15693_data_fmt	Int	Byte[26], Legic 15693 card data format
hid_iclass_uid_seq	Int	Byte[27], HID iCLASS uid sequences
hid_iclass_read_at	Int	Byte[28], HID iCLASS read start byte
hid_iclass_read_len	Int	Byte[29], HID iCLASS card read byte length
sony_felica_uid_seq	Int	Byte[30], sonyFelica uid sequences
sony_felica_read_at	Int	Byte[31], sonyFelica read start byte
sony_felica_read_len	Int	Byte[32], sonyFelica card read byte length
14443a_uid_seq	Int	Byte[33], 14443a uid sequences
14443a_4_read_at	Int	Byte[34], 14443_4 byte card read start byte
14443a_7_read_at	Int	Byte[35], 14443_7 byte card read start byte
14443a_uid_len	Int	Byte[36], 14443A card read UID byte length
mifare_read_at	Int	Byte[37], mifare byte card read start byte
mifare_read_len	Int	Byte[38], mifare card read byte length
mifare_data_fmt	Int	Byte[39], mifare data format
mifare_block_num	Int	Byte[40], mifare block number read by the card
mifare_key_type	Int	Byte[41], MIFAER Card Authentication Key Type
mifare_key	String	Byte[42~47], MIFARE Card Authentication Key
desfire_crypto_alg	Int	Byte[48], DESFire Card Encryption Algorithm
desfire_aid	String	Byte[49~51], DESFire Card AID
desfire_key_num	Int	Byte[52], DESFire Card Read Verification Key Number
desfire_key	String	Byte[53~76], DESFire card reading key

desfire_fid	Int	Byte[77], DESFire Card File FID
desfire_comm_mode	Int	Byte[78], DESFire card communication method
desfire_read_at	Int	Byte[79], DESFire Card read start byte, starting at [0]
desfire_read_len	Int	Byte[80], DESFire Card Read Byte Length
desfire_data_fmt	Int	Byte[81], DESFire Card Data Format
cpu_crypto_alg	Int	Byte[82], CPU Card Encryption Algorithm
cpu_fid	String	Byte[83~84], CPU card FID
cpu_key_num	Int	Byte[85], CPU Card Read Verification Key Number
cpu_key	String	Byte[86~101], CPU card reading key
cpu_efid	String	Byte[102-103], CPU card fileEFID
cpu_comm_mode	Int	Byte[104], CPU card communication method
cpu_read_at	Int	Byte[105], CPU Card read start byte, starting at [0]
cpu_read_len	Int	Byte[106], CPU Card Read Byte Length
cpu_data_fmt	Int	Byte[107], CPU Card Data Format

example of data returned by a device is as follows：

```
{
  "devSn" : "YGKJ2021DM0800003",
  "operations" : {
    "executeStatus" : 2,
    "id" : 25,
    "remark" : "failed",
    "result" : {
      "code" : -1
    },
    "type" : 1107
  }
}
```

Return parameter description：

Filed Name	Type	Description
code	Int	Set the parameter return code: 0: Success; -1: module type error; -2: Buffer length error; -3: I2C transmit failure; -4: module ack error; -5: module setup parameter failure;

4.2.21 Read parameters of LEGIC module

Brief description: The server gets the parameters of the LEGIC reader module through this interface.,

Type=1108

The server sends data as follows：

```
{
   "devSn" : "YGKJ2021DM0800003",
   "id" : 1074,
   "type" : 1108,
   "operations" : { }
}
```

The data returned from the device side is as follows, the contents of the fields refer to "Setting the parameters of the LEGIC card reader module".

```
{
   "devSn" : "YGKJ2021DM0800003",
   "operations" : {
      "executeStatus" : 1,
      "id" : 1074,
      "remark" : "success",
      "result" : {
         "14443a_4_read_at" : 0,
         ......
      },
      "type" : 1108
   }
}
```