



EJOIN HTTP API DEVELOPMENT SPEC

Version: 2.4.0



Project Name:	Internal
Project Code:	Confidential
Doc Name:	EJOIN HTTP API DEVELOPMENT SPEC
Version:	2.4.0
Release	0
Prepared by:	PengJian
Endorsed by:	
Accepted by:	
Doc. Effective Date:	
Date of Expiry:	Date: N/A

Related Document

Doc No.	Description

Distribution

Copy No.	Holder
1	
2	

Amendment History

Change Number	Revision Description	Chapter Affected	Revision Number	Changed By	Date
001	First draft	All	1.0	PengJian	2015.05.18
002	Modify some message interfaces	5.2.4 5.3.1 5.3.4 6.2.3	1.1	PengJian	2015.06.03
003	Implement and modify goip_send_cmd.htm Increase SMS task attribute coding, smsc	4.1.3 5.3.1.3	1.1	PengJian	2015.01.29
004	Modify and implement get and customize device status	3.1.1 3.1.2	1.2	PengJian	2016.04.17
005	Increase the save/reset command interface	4.1.3	1.2	PengJian	2016.06.13
006	Increase the get/set command interface	4.1.2	1.2	PengJian	2016.06.20
007	Increase SMS query interface	7	1.3	PengJian	2016.06.22
008	Add query SMS statistics and query call statistics interface Merge device-related HTTP documents and modify the document name		1.8	LiGuanB in	2019.01.03
009	Add example	5.4	1.9	Xiongjiao	2019.06.27
010	Add sending MMS parameters	5.3.1.3	2.0	chenkc	2019.11.28
011	Add MMS attachment parameters	5.3.1.3	2.0	Chenkc	2020.5.14
012	Add security interface	3		LiGuanB in	2021.8.05
013	Add “to_title_array” parameter	17	2.0	Xiongjiao o	2021.12.27
014	Add modify IMEI parameters	28	2.0	FZP	2023.04.12

015	Modify the “goip_get_status” interface 1. Add all_slots url parameters 2. Add the inserted field to the return result 3. Add slot_active in the command output fields	4.1.2 4.1.4	2.0	LGB	2023.9.26
016	Add SIM Card Numbers	6.4.4	2.0	FZP	2023.10.25
017	Add MMS receiving	11	2.1	lgb	2024.03.28
018	Add ESIM	12	2.2	lgb	2024.07.03
019	1.Add ledon and ledoff operations. 2.The port status includes an added led state.	4.1.4、5	2.3	lgb	2024.7.19
020	Modified port status code and details 1. Add status code 15: Access Mobile Network 2. Add status code 16: module response timeout	4.1.4	2.3.1	LMY	2024.9.10
021	Modify ESIM related interfaces	12	2.4	lgb	2024.11.22

Table of Content

1 Overview	1
2 Copyright Notice	1
3 Interface Security	1
3.1 Interactive process	1
3.1.1	2
3.2 Interactive description	2
3.2.1 Parameter	2
3.2.2 Response message	2
3.2.3 Subsequent URL request parameters	3
3.3 Statement of calculation	3
4 Status Notification	4
4.1.1 URL	4
4.1.2 Parameter	4
4.1.3 Dev-status message	5
4.1.4 Port -status message	6
5 Operate command sending	7
5.1.1 URL	7
host: Device IP address	7
5.1.2 Parameter	7
5.1.3 Data	8
6 SMS Sending	10
6.1 Topology	10
6.2 SMS sending flow chart	10
6.2.1 <i>SMS sending</i>	10
6.2.2 Sending status report	10
6.2.3 Sending status query	10
6.2.4 Pause the SMS task	10
6.2.5 Resume the paused SMS task	10
6.2.6 Delete the SMS task	10
6.2.7 Query the SMS task	10
6.3 Message description	11
6.3.1 SMS Sending	11
6.3.2 URL	11
6.3.3 Task send report	15
6.3.4 Pause the SMS task	16
6.3.5 Resume the paused SMS task	18
6.3.6 <i>Delete the SMS task</i>	19
6.3.7 Query the SMS task	21
6.4 Example	22
6.4.1 Status notification	22
6.4.2 Send command	23
6.4.3 Modify IMEI parameters	23
6.4.4 Add SIM Card Numbers	24
6.4.5 Send SMS	24
6.4.6 Pause the SMS task	25
6.4.7 Resume the paused SMS task	25
6.4.8 <i>Delete the SMS task</i>	26

6.4.9 Query the SMS task	26
7 Receive the SMS	27
7.1 Topology	27
7.2 Message description	27
7.2.1 URL	27
7.2.2 Parameter	27
7.2.3 Data	27
8 Query SMS	28
8.1 Query process	28
8.2 Message description	28
8.2.1 URL	28
8.2.2 Parameter	28
8.2.3 Data	29
9 Query SMS statistics	30
9.1 Message description	30
9.1.1 URL	30
9.1.2 Parameter	30
9.1.3 Response	31
10 Query call statistics	33
10.1 Message description	33
10.1.1 URL	33
10.1.2 Parameter	33
10.1.3 Response	34
11 MMS receiving	36
11.1 Configuration	36
11.2 Message Push Format	36
11.3 Push Message Example	37
12 ESIM	38
12.1 Query ESIM	38
12.1.1 URL	38
12.1.2 Parameter	38
12.1.3 Response	39
12.2 Write to ESIM	40
12.2.1 URL	40
12.2.2 Parameter	40
12.2.3 POST data	40
12.2.4 Response	41
12.3 Delete ESIM	41
12.3.1 URL	41
12.3.2 Parameter	42
12.3.3 POST data	42
12.3.4 Response	42

1 Overview

This document specifies HTTP SMS API of Ejoin device, include SMS receiving, single sending, group-sending and status of Sending task querying.

Based on 'Ejoin SMS sending API', this document extra support Group-Sending and multi task and status report support in one message.

This SMS API is still base on HTTP, and add POST request support, it provide multitask and status report by JSON array. About JSON array format, please refer its standardization state.

2 Copyright Notice

This document is just for Ejoin R&D team reference. If in need, it can be offered to a cooperation project developer.

3 Interface Security

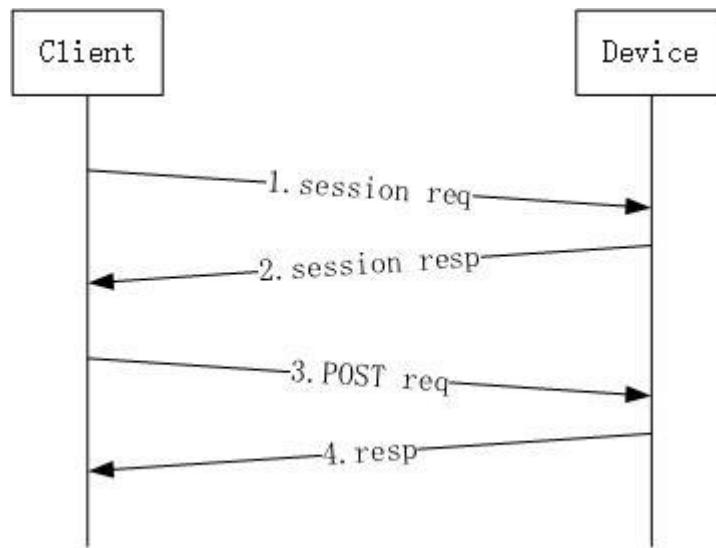
This section describes a way to ciphertext the body content submitted using HTTP POST, providing some security for the submitted data.

3.1 Interactive process

To enable encrypted transmission, a handshake is required to establish a secure session, as shown in Figure 3.1, as described below:

1. Setting up a secure session: The client uses `http://host:port/crypt_sess.json` to request the device to set up a session;
2. Secure session establishment Reply: After authenticating and setting the client, the device replies to the client's request;
3. Common interface request: After the session is successfully established, the client uses the ciphertext mode to submit data to the device through HTTP POST;
4. Reply on the common interface: After receiving the ciphertext POST, the device uses the negotiated encryption method decrypt POST, processes the data, and replies to the client in plaintext.

3.1.1



3.2 Interactive description

The establishment of a secure session requires that client interacts with the device through the special URL
http://host:port/crypt_sess.json

During the interaction, the two parties need to exchange some information. The information described as follows.

3.2.1 Parameter

Parameter	Description	Default	Required	Remark
username	Device account	None	Y	
nonce	Initial random string	None	Y	A random string provided by The client. See section 3.3
expires	Timeout value	180	N	Life cycle of a secure session
auth	Customer session ID	None	Y	The ID of the session (client-session-id), on the client. See section 3.3(part 1) for the calculation method.
crypt	Encryption method	None	N	The encryption method used in this secure session. The value is (see section 3.3 For the calculation method Of each encryption method KEY) .

3.2.2 Response message

The reply information is in JSON format. The values are as follows:

Parameter	Data type	Description	Default	Required	Remark
code	Int	result code	None	Y	Interface processing result,0 indicates Success, and non-0 Indicates failure
desc	String	Instructions of the result	None	Y	When code is not 0, The Failure cause is Described.
Session	String	Device session ID	None	Y	ID of the session on the Device.
expires	Int	Timeout value	180s	N	Life cycle of a secure session

3.2.3 Subsequent URL request parameters

After an encrypted session is established, the username and password are not required for the URL of the subsequent request, but the following parameters are required:

Parameter	Description	Default	Required	Remark
seq	sequence number	None	Y	The sequence number should Be increments in subsequent requests. If the seq of the previous request is 10, then the seq of the next request must be greater than 10, In principle, it should be 11.
auth	Customer session ID	None	Y	Note that this field is not the same as the auth field Representing client-session-id
session	Session ID	None	Y	When a session is established, The device returns the session id to the client. (server-session-id)
expires	Thmeout value	180	N	Reset the life cycle of a secure session

3.3 Statement of calculation

There are some parameters in a secure session that need to be computed. The follwing describes how these parameters are computed (the following '+' represents a connection string)

1.Auth(ID of an initial client session)=MD5(username + password + cnonce + url-resource),url-resource is the rest of the URL excluding http://host:port and parameters.such as

http://host:port/goip_get_status.html?url=xxx&period=0 for this request,url-resource refers to [/goip_get_status.html](#),the rest of the fields are as described in section 3.2;

2.Auth(ID of the subsequent URL calculated in real time)= MD5(username + password + session + seq + url-resource),url-resource is same as above, session is the server-session-ID returned when the session is established, the rest of the fields are as described in section 3.2;

3.Encrypt KEY=MD5(username + password + auth + session + seq),the auth field is the client-session-id calculated during the establishment of the session,that is,the auth calculated in step 1,session is the server-session-ID returned when the session is established,the rest of the fields are as described in section 3.2;

4 Status Notification

This device will send a HTTP POST request which include the device running status information to the SMS server when it reaches either of below conditions.

- 1.One of device port status changed.
- 2.Community period expired.

4.1.1 URL

Server can send a GET request to get the status on device by below URL:

http://host:port/goip_get_status.html?url=xxx&period=0

Device report url:

Server send a get request like above to specific the URL.

4.1.2 Parameter

Parameter	Description	Default	Required	Remark
Url	Specific the report URL	None	N	Tips: Special characters need do URL encode Just need specific once time.
Period	Report period	60	N	>0:Report status according to the specified report period,the minimum value is 60,the unit is second 0:cancel status report



all_sims	Get all the card status	0	N	0: disable 1: enable
all_slots	Get the status of all card slots	0	N	0: disable 1: enable

4.1.3 Dev-status message

Device send 'dev-status' to server periodically.

Dev-status message(information in HTTP message Body) is a JSON array string that comprised of ports status.

Parameter Content-Type in HTTP head should set to “[application/json; charset=utf-8](#)”.

Request samples:

```
{
  "type": "dev-status",
  "seq": 1,
  "expires": 180,
  "mac": "00-30-f1-01-02-03",
  "ip": "192.168.1.67",
  "max-ports": 32,
  "max-slots": 4,
  "status": [
    {
      "port": "1A",...
    },
    {
      "port": "2B",...
    },
    ...
    {
      "port": "32D",...
    }
  ]
}
```

Components state:

Parameter	Data type	Description	Default	Required
Type	String	Message type.	None	Y (dev-status)
seq	Int	Device status message sequence number, start from 1.	None	Y
expires	Int	Dev-status sending period.	180s	N
mac	String	Device MAC.	None	Y
ip	String	Device IP address.	None	Y
max-ports	Int	Total ports of device.	None	Y
max-slots	String	Total SIM slots of device.	1	N
status	array	Status of device port.	None	Y

For the description of the port status, see 3.1.4 [Port Status]

4.1.4 Port -status message

Any one port status changed, device will send ‘port-status’ message to server immediately.

Status message (information in HTTP message Body) is a JSON array string that comprised of ports status. Parameter Content-Type in HTTP head should set to “[application/json; charset=utf-8](#)”.

Request samples:

```
{
    "type": "port-status",
    "port": "2B",
    "seq": 1,
    "status": "3 OK",
    "bal": "100.00",
    "opr": "46000 China Mobile",
    "sn": "139xxxxxxxxx",
    "imei": "86xxxxxxxxxxxxxx",
    "imsi": "xxx",
    "iccid": "xxx"
}
```

Components state:

Parameter	Data type	Description	Default	Required
Type	String	Message type	None	Y (port-status)
port	String	Describe the current working port and SIM slot. Like 1.01,1.02.....32.04	None	Y
sim	String	Simpool's identification	None	N
seq	Int	The port is incremented from port 1		
st	string	Port status code + detail 0: No SIM card 1: Exist idel SIM card 2: Registering 3: Registered 4: call connected 5: no balance or alarm 6: Register failed 7: SIM card locked by device 8: SIM card locked by operator 9: Recognize SIM card error 11: Card Detected 12: User locked 13: Port inter-calling 14: Inter-calling holding 15: Access Mobile Network 16: module Response timeout	None	Y
bal	Floating	SIM card balance(yy.mm)	None	Y

	point			
opr	String	SIM card operator name and ID valid while parameter “st” equal to 3 or 4	None	N
sn	string	SIM number	None	N
imei	string	IMEI of module	None	N
active	int	Current card. 1:yes; 0:no	None	N
imsi	string	IMSI num of SIM card	None	N
iccid	string	ICCID num of SIM card	None	N
inserted	int	A card is inserted into a slot.1:yes; 0:no	None	N
Slot_active	int	Whether the card slot is enabled.1:enabled;0:disabled	None	N
sig	int	Signal value of the module	None	N
led	int	Led is enabled or not, 1 is yes, 0 is no	None	N

5 Operate command sending

Device offer API to server to change device status, like change sim card, lock/unlock port, reboot device.

About USSD command sending, please refer to “Ejoin HTTP-USSD API(V1.0).docx”;

Device will send message to the server by port status notification when lock port or change SIM card.

5.1.1 URL

http://host:port/goip_send_cmd.html

host: Device IP address

port: Device webpage management port, default value is 80.

5.1.2 Parameter

Parameter	Description	Default	Required	Ramark
Version	API version	1.1	Y	This document describes specification support only
username	Device account	None	Y	
Password	Device password	None	Y	
Op	Operation description	None	N	get:get device configuration set: set device configuration lock: lock port unlock:unlock port switch:switch SIM card reset: reboot module save: save configuration reboot:reboot device ledon: turn on led ledoff: turn off led
par_name(n)	Parameter name of the	The value	N	The name of the parameter to

	get/set operation	of the set operation		be actually operated. If the parameter is an array parameter, you can use parentheses with parameters, and the subscript starts from 0. Sms_url: destination URL for SMS push
--	-------------------	----------------------	--	--

5.1.3 Data

Device support a single command to operate multi-ports and multi-commands to operate different port.

Command sending message (information in HTTP message Body) is a JSON array string that composed by one or more command. Parameter “Content-Type” in HTTP head should set to [“application/json; charset=utf-8”](#).

Request samples:

```
{
  "type": "command",
  "op": "lock",
  "ports": "1.01,2.02,3.03,4-32",
  "ops": [
    {
      "op": "lock",
      "ports": "1.01"
    },
    {
      "op": "switch",
      "ports": "2B"
    }
  ]
}
```

Components state:

Parameter	Data Type	Description	Default	Required
type	string	Message type	None	Y (command)
op	string	operation type lock: lock port unlock: unlock port switch: switch SIM card reset: reboot module save: save configuration reboot: reboot device multiple: multiple command ledon: turn on led ledoff: turn off led	None	Y



ports	string	<p>the port NO. need to operate. all,*: all ports It is valid while op parameter equal to lock/unlock/switch/reset/ledon/ledoff When op=lock unlock, if you do not specify the SIM slot, the entire port will be locked/unlocked. When op=switch, it means to switch to this location. When op=reset, only the port number is valid.</p>	None	<p>Y (op=lock op=unlock op=switch op=reset op=ledon op=ledoff)</p>
ops	array	<p>Multi-command array Valid when op is multiple</p>	None	<p>Y (op= multiple)</p>

Attributes of tasks in json data

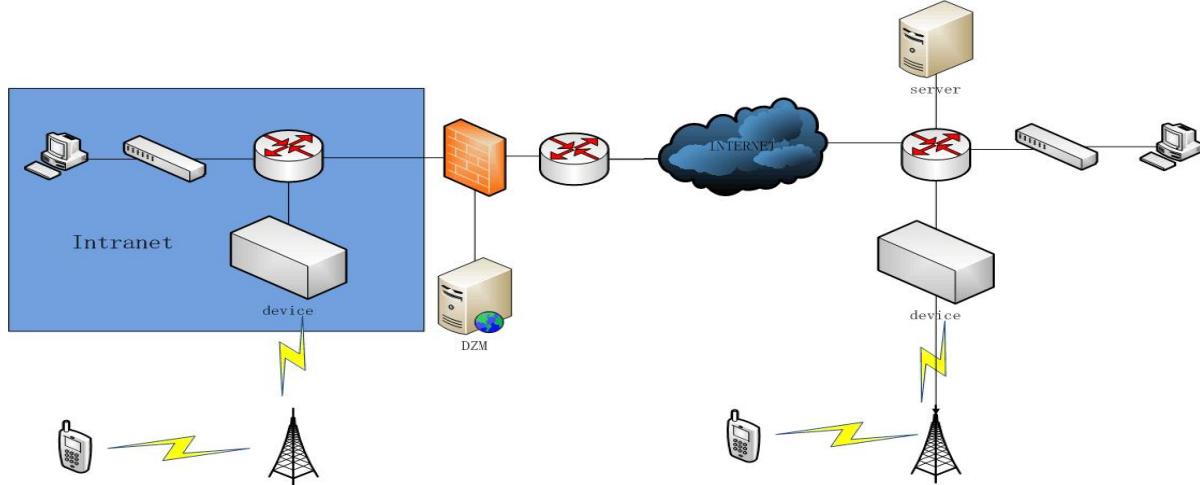
Parameter	Data type	Description	Default	Required
op	string	<p>Operation type lock: lock port unlock:unlock port switch:switch SIM card reset: reboot module ledon: turn on led ledoff: turn off led</p>	None	Y
ports	string	<p>The port that needs to perform the operation: When op=lock unlock, if you do not specify the SIM slot, the entire port will be Locked /unlocked. When op=switch, it means to switch to this location. When op=reset, only the port number is valid.</p>	None	Y

6 SMS Sending

Server send SMS sending request to device by Ethernet, and it keep sending task in a JSON array.

6.1 Topology

While device and server not in a same LAN, device HTTP port should been Nat mapping out on boundary router, so server can send message to device actively.



6.2 SMS sending flow chart

Server can send SMS by GET/POST request.

GET request use API specified in 'Ejoin http SMS sending API' document.

POST request can be used to send long SMS(more than 300 characters) or send more than one SMS in a request message.

6.2.1 SMS sending

After got the SMS sending request from server, device will design a task ID(related to the sending report) to create one or more SMS sending task, and after those task finished, device will send 200ok message to server.

6.2.2 Sending status report

While SMS send successfully/failed/timeout, device will put the result in cache, after meet condition(cache storage full or cache time reached),device will send POST request to report one or more task execute result. If it is group sending, device will report execute result periodically until all num in this group sending task get a result.

6.2.3 Sending status query

Server can send a GET request to query task sending status, like successfully send statics, failed information(num and reason),current sending num and so on.

6.2.4 Pause the SMS task

When a SMS task is still waiting to be sent in the send queue, one or more SMS can be paused.

6.2.5 Resume the paused SMS task

Resume the sending of one or more suspended SMS tasks.

6.2.6 Delete the SMS task

Delete one or more SMS tasks waiting to be sent in the send queue.

6.2.7 Query the SMS task

Query the SMS task waiting to be sent in the send queue.

6.3 Message description

6.3.1 SMS Sending

6.3.1.1 URL

http://host:port/goip_post_sms.html

host: Device IP address

port: Device webpage management port, default value is 80.

6.3.1.2 Parameter

Parameter	Description	Default	Required	Remark
version	API version	1.0	N	1.0: Compatible with previous API 1.1: Support for the description of this document
username	Device username	None	Y	
password	Device password	None	Y	

Note: All URL parameters only appear in the GET request. For POST, it will appear in the JSON attribute of the body segment. The following will not be explained.

6.3.1.3 Data

The task data (the body segment of the HTTP message) is a JSON format string consisting of one or more tasks, and the value of the HTTP header parameter "Content-Type" is "<application/json; charset=utf-8>".

Request samples:

```
{
  "type": "send-sms",
  "task_num":n,
  "tasks": [
    {
      "tid":tid_1, ...
    },
    ...,
    {
      "tid":tid_n, ...
    }
  ]
}
```

Components state:

Parameter	Data type	Description	Default	Required	Remark
type	string	Message type	None	N	



sr_url	string	Status report forward url	System configuration	N	
sr_cnt	string	Max num of SMS result can keep in cache.	100	N	1. The buffer number reaches this value, the buffered report is sent immediately, and the timer is reset.
sr_prd	Int	The max time SMS result can keep in cache	30	N	2. The time expired, even if not enough reports are received, send immediately, then reset the timer
sms_url	string	SMS forward URL	System configuration	N	
sms_cnt	Int	Max num of SMS can keep in cache	1	N	1. When the buffer number reaches this value, the buffered SMS is sent immediately and the timer is reset. 2. In order to be compatible with old customers, this parameter must be set to a value greater than 1, in order to buffer the received SMS and use the new sending mechanism.
sms_prd	Int	Max time SMS can keep in cache	30	N	1. The time expired, even if not enough SMS is received, it is sent immediately, then the timer is reset.
task_num	string	Total task	1	Y	
tasks	Array	Specific SMS send task	None	Y	

Attributes of tasks in json data

Parameter	Data type	Description	Default	Required
tid	Int	Task ID	None	Y
from	String	Use','-' to assign more than one port(from channel 1)	Choose of device	N
to	string	one or more(use ',' to distinguish) recipients	None	Y
to_all	String	1 represents all ports	1	N
sms	string	SMS content	None	Y
chs	string	Character coding set (utf8 base64)	utf8	N
coding	Int	pecific the SMS codec: 0:not assign 1:USC2 2:7bit	0	N
smstype	Int	SMS type 0: SMS, 1: MMS,	0	N

		2: MMS with multiple numbers and multiple subject		
smstitle	string	MMS subject: MMS subject format is UTF-8, this parameter could be empty;		N
to_title_array	Array	<p>Multiple MMS recipients and subject arrays. This field is valid only when smstype=2;</p> <p>Format:</p> <p>"to_title_array":[[{"number1","title1"}, {"number2","title2"}, ... {"number n","title n"}]]</p>	None	N
attachments	string	<p>1、MMS attachment, This field take effect only when smstype=1 or 2;</p> <p>2、Field content format:</p> <p>“File type1 base64 code file content 1; File type2 base64 code file content 2; File type n base64 code file content n”</p> <p>3、Remarks:</p> <ul style="list-style-type: none"> a. There can be a most of 5 attachment, all attachment total length cannot exceed 100K, at present only support jpg、gif、txt、mp3 formatted file. b. File type and file content use symbol separate c. Attachment is could be empty, send the content of the SMS filed as a TXT attachment <p>4、example:</p> <p>“txt zsSxvg==;jpg 4AAQSkZJRgABAgAAZABk”</p>		N
smsc	string	Store the SMSC number.		N
intvl	string	Interval of 2 SMS while device sending them(ms)	0	N
tmo	Int	Max time while waiting sending result(second).	30	N
sdr	Int	If enable the successfully send report. 1:enable,0:disable.	Disable	N
fdr	Int	If enable the failed report.1:enable,0:disable.	Enable	N
dr	Int	If enable SMS delivery report.1:enable,0:disable.	Disable	N
sr_prd	Int	Status report period (seconds), 0: not on, >0: on Control only the period in which a single task generates a report	60	N
sr_cnt	Int	Single-state report SMS number, less than 1 will use the default value Control only the number of processed messages that have been reported by a single task	10	N

Tips:

1. While not choose a port, device will pick up a random port to send SMS to every recipient.
2. While chose more than one ports, and just one SMS recipient , then device will use all chose port to send SMS to this recipient.
3. If more ports, and more recipient designed, then every port will send SMS to every recipient separately.

6.3.1.4 Response

Task sending is a JSON array composed by one or more task send status,

Parameter “Content-Type” in HTTP head should set to “[application/json; charset=utf-8](#)”.

Request samples:

```
{
  "code": 200,
  "reason": "OK",
  "type": "task-status",
  "status": [
    {
      "tid":tid_1,
      "status": "0 OK"
    },
    ...,
    {
      "tid":tid_n,
      "status":"2 Invalid Port"
    }
  ]
}
```

Status components state:

Parameter	Data type	Description	Default	Required
tid	Int	Task ID	None	Y
status	String	Task status code 0: OK 1: Invalid User 2: Invalid Port 3: USSD Expected 4: Pending USSD 5: SIM Unregistered 6: Timeout 7: Server Error 8: SMS expected 9: TO expected 10: Pending Transaction 11: TID Expected 12: FROM Expected 13: Duplicated TaskId 14: Unauthorized 15: Invalid CMD 16: Too Many Task 17: MMS Title expected 18: Too Many MMS Attachments 19: MMS Attachments expected	None	Y

		20: MMS Attachments Cache size Overlimit If task status ID not equal to 0, then means this task not received by device.		
--	--	--	--	--

6.3.2 Task send report

6.3.2.1 URL

Server or device configuration decision.

6.3.2.2 Parameter

Parameter	Description	Default	Required	Remark
Version	API version	1.1	Y	This document describes specification support only

6.3.2.3 Data

Status-report message is a JSON array composed by one or more task send report.

Parameter “Content-Type” in HTTP head should set to “[application/json; charset=utf-8](#)”.

Request samples:

```
{
  "type": "status-report",
  "rpt_num":n,
  "rpts": [
    {
      "tid":tid_1, ...
    },
    ...,
    {
      "tid":tid_n, ...
    }
  ]
}
```

Components state:

Parameter	Data type	Description	Default	Required
type	string	Message type	None	Y (status-report)
rpt_num	string	Report num	1	Y
rpts	Array	Detail status report	None	Y

Attributes reported in json data

Parameter	Data type	Description	Default	Required
tid	Int	related task ID .	None	Y
sending	Int	total num of SMS in sending.	None	Y



sent	Int	total num of SMS successfully send.	None	Y
failed	Int	total num of SMS failed send.	None	Y
unsent	Int	total SMS in cache waiting for send	None	Y
sdr	array	Successfully send details(one list for one num) Success report is a array, [0]: recipients num index(based on group sending),int [1]: num, string [2]: SMS sending port(1.01,2.02,...), string [3]: SMS send timestamp in UTC time, int	None	N
fdr	array	Fail send details(one num one list) [0]: recipients num index(based on group sending),int [1]: num, string [2]: SMS sending port(1.01,2.02,...), string [3] : SMS send timestamp in UTC time.int [4] : Progress reason, code+details[refer to 1.0 API] [5] : Carrier reason, code+ description. Valid when [4] is failed to send	None	N

Tips:

1. tid related to the task ID in the send-sms message.
2. sending, sent, failed are accumulated num in one task.
3. Sdr(successfully send record) list records between 2 status-report, server should keep before details.
4. fdr(failed detail record) list records between 2 status-report, server should keep before details.

6.3.3 Pause the SMS task

6.3.3.1 URL

http://host:port/goip_pause_sms.html

host: Device IP address

port: Device webpage management port, default value is 80.

6.3.3.2 Parameter

Parameter	Description	Default	Required	Remark
version	API version	1.1	Y	This document describes specification support only
username	Device username	None	Y	
password	Device password	None	Y	

6.3.3.3 Data

The data (the body segment of the HTTP message) is a JSON format string consisting of one or more tasks, and the value of the HTTP header parameter "Content-Type" is "<application/json; charset=utf-8>".

Request samples:

```
{
  "tids": [tid1,tid2,...,tidn]
}
```

Components state:

Parameter	Data type	Description	Default	Required
tids	String	Restore one or more tasks ID that are sent.	None	No. When there is no field, all send tasks are suspended.

The body of the query response is a JSON format string consisting of one or more tasks.

The value of the HTTP header parameter "Content-Type" is "[application/json; charset=utf-8](#)".

Request samples:

```
{
  "code": 200,
  "reason": "OK",
  "results": [
    {
      "tid":tid_1,
      "status": "0 OK"
    },
    ...,
    {
      "tid":tid_n,
      "status": "2 Invalid Port"
    }
  ]
}
```

Components state:

Parameter	Data type	Description	Default	Required
code	Int	Interface processing result code	None	Y
reason	String	Interface processing results	None	Y
results	Array	Task processing results. When there is no tids in the request, this field is not included in the result.	None	N
tid	Int	Task ID	None	N
status	String	Task status code 0: OK 1: Invalid User 2: Invalid Port 3: USSD Expected 4: Pending USSD 5: SIM Unregistered 6: Timeout 7: Server Error 8: SMS expected 9: TO expected 10: Pending Transaction	None	N



		11: TID Expected 12: FROM Expected If task status ID not equal to 0, then means this task not received by device.		
--	--	---	--	--

6.3.4 Resume the paused SMS task

6.3.4.1 URL

http://host:port/goip_resume_sms.html

host: device IP address

port: device web port, the default is 80, optional.

6.3.4.2 Parameter

Parameter	Description	Default	Required	Remark
version	API version	1.1	Y	This document describes specification support only
username	Device username	None	Y	
password	Device password	None	Y	

6.3.4.3 Data

Data (The body segment of the HTTP message) is a JSON array composed by one

or more task. Parameter Content-Type in HTTP head should set to

["application/json; charset=utf-8"](#).

```
{  
    "tids": [tid1,tid2,...,tidn  
]  
}
```

Attributes instructions:

Parameter	Data type	Description	Default	Required
tids	String	The task ID to be restored	None	NO. Restore all sending tasks when there is no field

The body of response is a JSON array composed by one or more task ID.

Parameter "Content-Type" in HTTP head should set to ["application/json; charset=utf-8"](#).

Request samples:

```
{  
    "code": 200,  
    "reason": "OK",  
    "results": [  
        {  
            "tid":tid_1,  
            "status": "0 OK"  
        },  
        ...  
    ]  
}
```

```

{
    "tid":tid_n,
    "status": "2 Invalid Port"
}
]
}

```

Attributes instructions:

Parameter	Data type	Description	Default	Required
code	Int	Interface processing result code	None	Y
reason	string	Interface processing results	None	Y
results	array	Task processing results	None	Y
tid	Int	Task ID	None	Y
status	string	The status code and reason description of the task 0: OK 1: Invalid User 2: Invalid Port 3: USSD Expected 4: Pending USSD 5: SIM Unregistered 6: Timeout 7: Server Error 8: SMS expected 9: TO expected 10: Pending Transaction 11: TID Expected 12: FROM Expected	None	Y
		If the task's response code is not 0, it indicates that the task is not accepted by the device.		

6.3.5 Delete the SMS task.

6.3.5.1 URL

http://host:port/goip_remove_sms.html

host: device IP address.

port: device web port, the default is 80, optional.

6.3.5.2 Parameter

Parameter	Description	Default	Required	Remark
version	API version	1.1	Y	This document describes specification support only
username	Device username	None	Y	
password	Device password	None	Y	

6.3.5.3 Data

Data(The body segment of the HTTP message)is a JSON array composed by one or more task. Parameter Content-Type in HTTP head should set to "[application/json; charset=utf-8](#)".

```
{
  "tids": [tid1,tid2,...,tidn
  ]
}
```

Attributes instructions:

Parameter	Data type	Description	Default	Required
tids	String	The task ID to be restored.	None	NO Restore all sending tasks when there is no field.

The body of response is a JSON array composed by one or more task ID. Parameter “Content-Type” in HTTP head should set to "[application/json; charset=utf-8](#)".

Request samples:

```
{
  "code": 200,
  "reason": "OK",
  "results": [
    {
      "tid":tid_1,
      "status": "0 OK"
    },
    ...,
    {
      "tid":tid_n,
      "status": "2 Invalid Port"
    }
  ]
}
```

Attributes instructions:

Parameter	Data type	Description	Default	Required
code	Int	Interface processing result code	None	Y
reason	string	Interface processing results	None	Y
results	array	Task processing results . If there is no TIDS in the request, this field is not included in the result	None	Y
tid	Int	Task ID	None	Y
status	string	The status code and reason description of the task 0: OK 1: Invalid User 2: Invalid Port 3: USSD Expected 4: Pending USSD 5: SIM Unregistered 6: Timeout 7: Server Error 8: SMS expected	None	Y

		9: TO expected 10: Pending Transaction 11: TID Expected 12: FROM Expected If the task's response code is not 0, it indicates that the task is not accepted by the device.		
--	--	--	--	--

6.3.6 Query the SMS task

6.3.6.1 URL

`http://host:port/goip_get_tasks.html?version=xxx&username=root&password=root&port=xxx&pos=xxx&num=xxx&has_content=xx`

host: device IP address.

port: device web port, the default is 80, optional.

6.3.6.2 Parameter

Parameter	Description	Default	Required	Remark
version	API version	1.1	Y	This document describes specification support only
username	Device username	None	Y	
password	Device password	None	Y	
port	Send port. Starting from 1	None	Y	
Has_content	Whether to bring a text message in the result.	0	N	0 –Do not return text message content. 1 –return text message content
Pos	The starting position of the request. 0 represents the beginning of the first task.	None	Y	The location of this field is in the order of the submission of the message task order
Num	The number of Request messages.	10	N	

The body of response is a JSON array composed by one or more task ID. Parameter Content-Type in HTTP head should set to "`application/json; charset=utf-8`".

```
{
  "tids": [tid1,tid2,...,tidn
  ]
}
```

Attributes instructions:

Parameter	Data type	Description	Default	Required
tids	String	The task ID to be query.	None	N

The body of response is a JSON array composed by one or more task ID.

Parameter “Content-Type” in HTTP head should set to “[application/json; charset=utf-8](#)”.

Request samples:

```
{
    "code": 200,
    "reason": "OK",
    "total_num": n,
    "task_num": n,
    "tasks": [
        {
            "tid": tid_1, ...
        },
        ...,
        {
            "tid": tid_n, ...
        }
    ]
}
```

Attributes instructions:

Parameter	Data type	Description	Default	Required
code	Int	Interface processing result code.	None	Y
reason	string	Interface processing results	None	Y
Total_num	Int	The number of tasks.	None	Y
Task_num	Int	Query the number of tasks returned.	None	Y
Tasks	Array	Query the returned array of tasks.	None	Y

Attributes instructions of task:

Tasks	array	Query the returned array of tasks.	None	Y
Tid	Int	Task id	None	Y
From	string	One or more (commas, short horizontal connections) send ports (from 1)	Device selection	N
To	string	One or more (comma connected) message receiver number.	None	Y
Sms	string	Message content	None	N
Chs	string	Code set (utf8 base64)	Utf-8	N
Coding	Int	Specifies the encoding of the message sent. 0: Do not specify 1: USC2 2: 7bit	0	N
State	Int	Task status: 0 - normal, 1 - suspended.	None	Y

6.4 Example

6.4.1 Status notification

Url:



http://192.168.1.67:80/goip_get_status.html?url=http://192.168.1.142&period=60&username=root&pass

word=root

Response:

```
1  {
2      "type": "dev-status",
3      "seq": 3,
4      "expires": 60,
5      "mac": "00-30-f1-00-6c-55",
6      "ip": "192.168.1.67",
7      "ver": "532-604-924-041-100-000",
8      "max-ports": 32,
9      "max-slot": 4,
10     "status": [
11         {
12             "port": "1.01",
13             "sim": "",
14             "seq": 8,
15             "st": 3,
16             "imei": "898610599441327",
17             "iccid": "89860116289910033431",
18             "imsi": "460014978212473",
19             "sn": "",
20             "opr": "46001 CHINA UNICOM GSM",
21             "bal": "0.00",
22             "sig": 15,
23             "tot_dur": "0/-1",
24             "mon_dur": "0/-1",
25             "day_dur": "0/-1"
26         }
27     ]
28 }
```

6.4.2 Send command

Url:

http://192.168.1.67:80/goip_send_cmd.html?username=root&password=root

Body: {"type":"command", "op":"switch", "ports":"2.02"}

Response:

```
1  {
2      "code": 0,
3      "reason": "OK"
4 }
```

6.4.3 Modify IMEI parameters

Url:

http://192.168.1.67:80/goip_send_cmd.html?username=xx&password=xx&op=set

Body: sim_imei(n)=xxx&sim_imei(n)=xxx

Method: POST

Response:

```
1  {
2      "code": 0,
3      "reason": "OK",
4      "par_set": 2
5 }
```

Reference:

information in HTTP message Body is a text, many imei parameters modify use & to connect.

sim_imei(n)=xxxx,

n = (port_number - 1) * slots_per_port + (slot_number - 1) xxx = imei parameters

Example:

sim_imei(21)=865847053403202&sim_imei(51)=865847053403213

21 means 6B, B is slot_number 2 ,so 6B= (6 - 1) * 4+(2-1)=21

51 means 13D, D is slot_number 4,so 13D =(13-1)*4+(4-1)=51

6.4.4 Add SIM Card Numbers

Url:

http://192.168.1.67:80/goip_send_cmd.html?username=root&password=root&op=set

Body: sim_number(n)=xxx&sim_number(n)=xxx

Method: POST

Response:

```

1  {
2      "code": 0,
3      "reason": "OK",
4      "par_set": 2
5  }

```

Reference:

information in HTTP message Body is a text, many imei parameters modify use & to connect.

sim_number(n)=xxxx,

n = (port_number - 1) * slots_per_port + (slot_number - 1), xxx = sim_numbers parameter

Example:

sim_number(192)=1358021178&sim_number(196)=123456762221

192 represents 25.01, where max-slot is 8, and the slot number for 01 is 1.

So, 25.01=(25 - 1)*8+(1-1)=192.

196 represents 25.05, max-slot is 8, and the slot number for 05 is 5.

So, 25.05=(25-1)*8+(5-1)=196.

6.4.5 Send SMS

Url:

http://192.168.1.67:80/goip_post_sms.html?username=root&password=root

Body:

```

{
  "type": "send-sms",
  "task_num": 1,
  "tasks": [
    {
      "tid": 1223,
      "to": "13686876620",
      "sms": "hello123"
    }
  ]
}

```

Response:

```
1 [
2   "code": 200,
3   "reason": "OK",
4   "type": "task-status",
5   "status": [
6     {
7       "tid": 1223,
8       "status": "0 OK"
9     }
10  ]
11]
```

6.4.6 Pause the SMS task

Url:

http://192.168.1.67:80/goip_pause_sms.html?username=root&password=root

Body:

```
{"tids": [2,3]}
```

Response:

```
1 [
2   "code": 200,
3   "reason": "OK",
4   "results": [
5     {
6       "tid": 2,
7       "status": "0 OK"
8     },
9     {
10       "tid": 3,
11       "status": "0 OK"
12     }
13   ]
14 ]
```

6.4.7 Resume the paused SMS task

Url:

http://192.168.1.67:80/goip_resume_sms.html?username=root&password=root

Body:

```
{"tids": [2,3]}
```

Response:

```
1 [
2   "code": 200,
3   "reason": "OK",
4   "results": [
5     {
6       "tid": 2,
7       "status": "0 OK"
8     },
9     {
10       "tid": 3,
11       "status": "0 OK"
12     }
13   ]
14 ]
```

6.4.8 Delete the SMS task

Url:

http://192.168.1.67:80/goip_remove_sms.html?username=root&password=root

Body:

```
{"tids": [2, 3]}
```

Response:

```
1 {
2     "code": 200,
3     "reason": "OK",
4     "results": [
5         {
6             "tid": 2,
7             "status": "0 OK"
8         },
9         {
10            "tid": 3,
11            "status": "0 OK"
12        }
13    ]
14 }
```

6.4.9 Query the SMS task

Url:

http://192.168.1.67:80/goip_get_tasks.html?version=1.1&username=root&password=root&port=1&pos=0&has_content=1

Response:

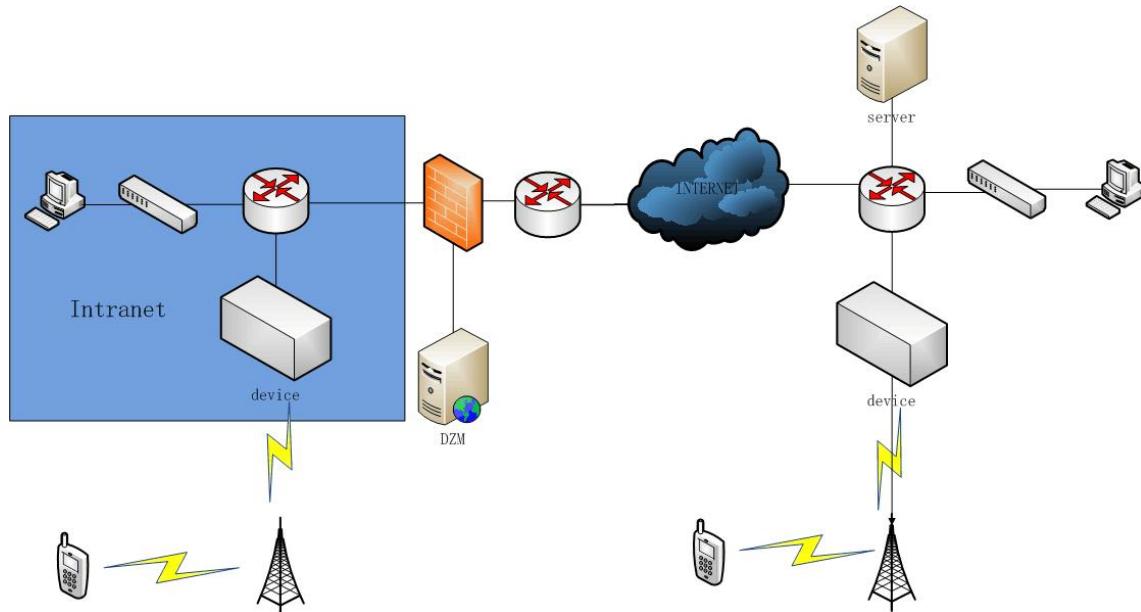
```
{
    "code": 200,
    "reason": "OK",
    "total_num": 5,
    "task_num": 5,
    "tasks": [
        {
            "tid": 2,
            "state": 0,
            "from": "",
            "to": "13686876820",
            "chs": "utf8",
            "coding": 0,
            "sms": "hello123"
        },
        {
            "tid": 3,
            "state": 0,
            "from": "",
            "to": "13686876820",
            "chs": "utf8",
            "coding": 0,
            "sms": "hello123"
        },
        {
            "tid": 4,
            "state": 0,
            "from": "",
            "to": "10010",
            "chs": "utf8",
            "coding": 0,
            "sms": "CYVF"
        }
    ]
}
```

7 Receive the SMS

Device will send SMS to server by POST request, while it get SMS from operator.

7.1 Topology

While forwarding SMS, device send a request to server first, so even device in LAN, and server in a public net, it also can work effectively



7.2 Message description

7.2.1 URL

Default set by device, server can revise it in SMS sending quest message.

7.2.2 Parameter

Parameter	Description	Default	Required	Remark
version	API version	1.1	Y	This document describes specification support only

7.2.3 Data

A JSON format string consisting of one or more short messages. The value of the HTTP header parameter "Content-Type" is "[application/json; charset=utf-8](#)".

Request samples:

```
{
  "type": "recv-sms",
  "sms_num": n,
  "sms": [
    [],
    ...,
    []
  ]
}
```

Attributes instructions:

Parameter	Data type	Description	Default	Required
type	String	Data type	None	Y (recv-sms)
sms_num	String	Total SMS num	None	Y
sms	Array	SMS array	None	Y

For saving the bandwidth, SMS content also storage in a array.

[0]: Delivery report flag,0:normal SMS, 1:This is a delivery report

[1]: Receive report('1.01','1.02')

[2]: timestamp while device got this SMS

[3]: Sender(If Delivery report flag set to 1, then sender should be SMCC)

[4]: Recipient(If Delivery report flag set to 1,then recipient should be the original recipient.)

[5]: SMS content:

Delivery report: "code scts", code is 0 for successful delivery, utf-8

Ordinary SMS: BASE64 encoding of utf-8

8 Query SMS

Customers can actively query the SMS received by the device through HTTP GET/POST request.

8.1 Query process

1. Query all messages received by the device without the query parameters (except for user-verified parameters).
2. Use the value of next_sms returned by the device as the value of the sms_id parameter to query subsequent SMS messages.
3. If the device restarts, return a different ssrc synchronization source ID and return to step 1.
4. Repeat step 2

8.2 Message description

8.2.1 URL

```
http://host:port/goip_get_sms.html?username=root&password=root&sms_id=xx
x&sms_num=xxx
```

host: Device IP address

port: Device webpage management port, default value is 80.

8.2.2 Parameter

Parameter	Description	Default	Required	Remark
sms_id	Start SMS ID	1	N	1: The first SMS ID received by the device

sms_num	Specify the number of SMS to be queried	0	N	0: Query all SMS
sms_del	Delete the SMS that has been returned by the query	0	N	0: Do not delete, 1: delete

8.2.3 Data

A JSON format string consisting of one or more short messages. The value of the HTTP header parameter "Content-Type" is "[application/json; charset=utf-8](#)".

Request samples:

```
{
  "code": 0,
  "reason": "OK",
  "ssrc": "0123456789abcdef",
  "sms_num": 2,
  "next_sms": 3,
  "data": [
    [
      0,
      "1B",
      1466506477,
      "10010",
      "13265825775",
      "5bCK5pWs55qE55So5oi377yM5oKo5aW977yB5o6o6I2Q5oKo5L2/55So44CQ5omL5py66JC15Lia5Y6F44CR5Li
A56uZ5byP5YWN5rWB6YeP5pyN5Yqh5bmz5Y+wIGh0dHA6Ly91LjEwMDewLmNuL2R0Y2Qg77yM6L275p2+5p+l6K+i6K+d6LS544C
B5L2Z6aKd5Y+K6K+m5Y2V77yb5oiW5Zue5aSN5Lu15LiL5pWw5a2X5Lu56CB6I635Y+W5oKo6ZyA6KaB55qE5pyN5Yqh77yaDQo
xMDEu5b2T5pyI6K+d6LS577ybDQoxMDIu5Y+v55So5L2Z6aKd77ybDQowLuWNh+e6pzRH77ybDQoxLuvnei0ueWPiuenr+WIhu+
8mw0KMi7otKbmiLfmn6Xor6LvvJsNCjMu5YWF5YC877ybDQo0LuWouaIt+acjeWKoe+8mw0KNS7kuJrliqHlip7nkIbvvJsNCjY
u5aKe5YC85Lia5Yqh77ybDQo3LuecgeS7veS4k+wMuu+8mw0KOS7ng63ngrnkv4PplIDjgIINCuW5v+S4n0iBl0mAmuOAggAAAAA
="],
    [
      0,
      "1B",
      1466506670,
      "10010",
      "13265825775",
      "5bCK5pWs55qE55So5oi377yM5oKo5aW977yB5o6o6I2Q5oKo5L2/55So44CQ5omL5py66JC15Lia5Y6F44CR5Li
A56uZ5byP5YWN5rWB6YeP5pyN5Yqh5bmz5Y+wIGh0dHA6Ly91LjEwMDewLmNuL2R0Y2Qg77yM6L275p2+5p+l6K+i6K+d6LS544C
B5L2Z6aKd5Y+K6K+m5Y2V77yb5oiW5Zue5aSN5Lu15LiL5pWw5a2X5Lu56CB6I635Y+W5oKo6ZyA6KaB55qE5pyN5Yqh77yaDQo
xMDEu5b2T5pyI6K+d6LS577ybDQoxMDIu5Y+v55So5L2Z6aKd77ybDQowLuWNh+e6pzRH77ybDQoxLuvnei0ueWPiuenr+WIhu+
8mw0KMi7otKbmiLfmn6Xor6LvvJsNCjMu5YWF5YC877ybDQo0LuWouaIt+acjeWKoe+8mw0KNS7kuJrliqHlip7nkIbvvJsNCjY
u5aKe5YC85Lia5Yqh77ybDQo3LuecgeS7veS4k+wMuu+8mw0KOS7ng63ngrnkv4PplIDjgIINCuW5v+S4n0iBl0mAmuOAggAAAAA
="]
  ]
}
```

Property description:

Parameter	Data type	Description	Default	Required
code	int	Operation code	None	Y
reason	string	Reason description	None	Y
ssrc	string	Synchronization source identifier The device generates a new ssrc each time it runs. So the value changes, re-query	None	Y
sms_num	int	Number of SMS queried	None	Y
next_sms	int	The next SMS ID	None	Y
data	int	Queryed SMS content	None	Y

For saving the bandwidth, SMS content also storage in a array. [0]:

Delivery report flag,0:normal SMS, 1:This is a delivery report [1]: Receive report('1.01','1.02')

[2] : timestamp while device got this SMS

[3] : Sender(If Delivery report flag set to 1, then sender should be SMCC)

[4] : Recipient(If Delivery report flag set to 1,then recipient should be the original recipient.)

[5] : SMS content:

Delivery report: "code scts", code is 0 for successful delivery, utf-8 Ordinary SMS:
BASE64 encoding of utf-8

9 Query SMS statistics

The client can actively query the short message statistics of the device through the HTTP GET request.

9.1 Message description

9.1.1 URL

`http://host:port/goip_get_sms_stat.html?version=xxx&username=xxx&password=xxx&ports=xxx&slots=xxx&type=xxx`

host: Device IP address

port: Device webpage management port, default value is 80.

9.1.2 Parameter



Parameter	Description	Default	Required	Remark
version	API version	1.0	N	1. 0 :Compatible with previous API 1. 1 :Support for the description of this document
username	Device Username	None	Y	
password	Device password	None	Y	
ports	The specified port number (valued from 1). The values are as follows: 1) all: all ports; 2) Specify a single port; 3) 1-2, 4: Port numbers separated by short numbers, specifying multiple ports, where "-" indicates a continuous port number;	all	N	
slots	The specified card slot number (value from 1). The values are as follows: 1) .Do not specify: current card 2) all: all card slots; 3) 2: Specify a single card slot; 4) 1-2, 4: A card slot number separated by a short number, specifying a plurality of card slots, where "-" indicates a continuous card slot number;	Current card	N	
type	Statistic type. The values are as follows: 1) 0: The last hour; 2) 1: The last two hours; 3) 2: today; 4) 3: cumulative;	0	N	

9.1.3 Response

The body of the response is a JSON array, and the value of the HTTP header parameter "Content-Type" is "[application/json; charset=utf-8](#)".

Request samples:

```
{  
  "code": 200,  
  "reason": "OK",  
  "count":n,  
  "stats": [  
    {  
      "port":xxx,...
```

},...
}]}

Property description:

Parameter	Data type	Description	Default	Required
code	int	Interface processing result code	None	Y
reason	string	Interface processing result reason	None	Y
count	int	Number of statistical results	None	Y
stats	stat result array	Array of statistical results (stat results are as follows)	None	Y

stat result description:

Parameter	Data type	Description	Default	Required
port	int	Port number (value from 1)	None	Y
slot	int	Slot number (value from 1)	None	Y
received	int	Number of SMS received	None	Y
sent	int	Number of SMS sent	None	Y
sent_ok	int	Number of successful SMS sent	None	Y
sent_failed	int	Number of failed SMS sent	None	Y
con_failed	int	Number of consecutive failed SMS	None	Y
unsent	int	Number of unsent SMS	None	Y
sending	int	Number of SMS being sent	None	Y

10 Query call statistics

The client can actively query the call statistics of the device through the HTTP GET request.

10.1 Message description

10.1.1 URL

```
http://host:port/goip_get_call_stat.html?version=xxx&username=xx
x&password=xxx&ports=xxx&slots=xxx&type=xxx
```

host: Device IP address

port: Device webpage management port, default value is 80.

10.1.2 Parameter

Parameter	Description	Default	Required	Remark
version	API version	1.0	N	1.0: Compatible with previous API 1.1: Support for the description of this document
username	Device Username	None	Y	
password	Device password	None	Y	
ports	The specified port number (valued from 1). The values are as follows: (1) all: all ports; (2) 2: Specify a single port; (3) 1-2, 4: Port numbers separated by short numbers, specifying multiple ports, where "-" indicates a continuous port number;	all	N	

slots	The specified card slot number (value from 1). The values areas follows: (1) Do not specify: current card (2) all: all card slots; (3) 2: Specify a single card slot; (4) 1-2, 4: A card slot number separated by a short number, specifying a plurality of card slots, where "-" indicates a continuous card slot number;	Current card	N	
type	statistic type. The values are as follows: (1) 0: The last hour; (2) 1: The last two hours; (3) 2: today; (4) 3: cumulative;	0	N	

10.1.3 Response

The body of the response is a JSON array, and the value of the HTTP header parameter "Content-Type" is "[application/json; charset=utf-8](#)".

Request samples:

```
{
  "code": 200,
  "reason": "OK",
  "count":n,
  "stats": [
    {
      "port":xxx,...
    }...
  ]
}
```

Property description:

Parameter	Data type	Description	Default	Required
code	int	Interface processing result code	None	Y
reason	string	Interface processing result reason	None	Y
count	int	Number of statistical results	None	Y
stats	stat result array	Array of statistical results (stat results are as follows)	None	Y

stat result description:

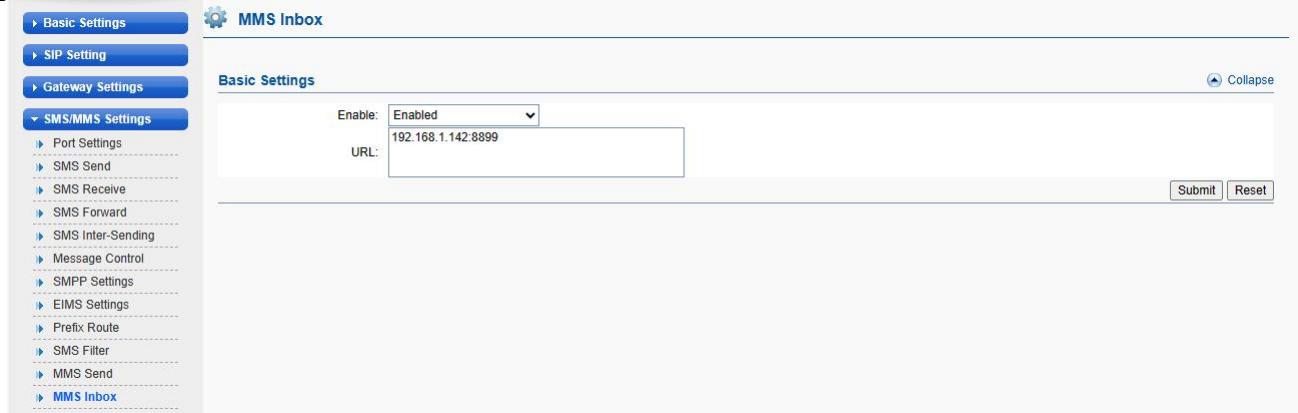
Parameter	Data type	Description	Default	Required
port	int	Port number (value from 1)	None	Y
slot	int	Slot number (value from 1)	None	Y
calls	int	Number of calling	None	Y
alerted	int	Number of alerting	None	Y
connected	int	Number of connected calls	None	Y
con_failed	int	Number of failed calls	None	Y
nc	string	No Carriers success ratio ("n/m"), the numerator n is the success number, and the denominator m is the total number	None	Y
pdd	int	PDD	None	Y
acd	int	ACD	None	Y
asr	int	ASR percentage value, for example, asr is 43, which is 43%	None	Y
tcd	int	Cumulative call duration (minutes)	None	Y
act_tcd	int	(actual) actual cumulative call duration (seconds)	None	Y

11 MMS receiving

At this stage, the main function is to push the received MMS content to the configured server address via HTTP.

11.1 Configuration

In the 'SMS/MMS Settings' menu of the device, the 'MMS Inbox' page allows you to configure the push receiving server address, as shown in the figure below:



- Enable: Whether to enable the MMS content push function.
- URL: The HTTP server address that receives MMS content push.

11.2 Message Push Format

MMS content is pushed via HTTP POST. Please refer to the HTTP standard for details. Only special HTTP header fields are explained below (Note: header fields may not exist). In addition, header fields starting with 'Emms-' are custom fields, while others are standard HTTP fields:

header field name	Code	parameter	Instructions
Emms-Subject	BASE64	charset: Character set, such as UTF-8	MMS subject
Emms-From	BASE64	charset: Character set, such as UTF-8 type: Sender address type, usually a phone number, known as PLMN	MMS sending address
Emms-To	BASE64	charset: Character set, such as UTF-8 type: Sender address type, usually a phone number, known as PLMN	MMS receiving address
Date	None	None	MMS sending time

Content-Type	None	Related to specific types	The specifics need to be decided according to the operator, but attention should be paid to the processing of multipart related types
--------------	------	---------------------------	---

11.3 Push Message Example

The following is a MMS message sent during a certain test. (Note: Ellipsis in the image represents omitted content which is irrelevant to the format.)

```

POST / HTTP/1.1
Host: 192.168.1.142:8899
Date: Thu, 28 Mar 2024 09:47:10 GMT
Emms-Subject: c3ViamVjdAw; charset=US-ASCII
Emms-From: Kzg2MTM3NTEwNjQxMjk=; type=PLMN; charset=UTF-8
Emms-To: ODYxMzU4MDI1Njg0Nw==; type=PLMN; charset=UTF-8
Keep-Alive: timeout=180
Connection: Keep-Alive
Content-Type: multipart/related; boundary=032817471092001110508; type="application/smil"; start=<0.smil>
Content-Length: 176603

--032817471092001110508
Content-Type: application/smil
Content-ID: <0.smil>

.....
--032817471092001110508
Content-Type: image/jpeg; name="IMG_3337.jpg"
Content-Location: IMG_3337.jpg

.....
--032817471092001110508
Content-Type: text/plain; charset=US-ASCII
Content-Location: text_1.txt

content00
--032817471092001110508--

```

12 ESIM

This API provides operations to query, write to, and delete ESIM on the device. The write and delete operations for ESIM cannot be completed immediately, so when a successful HTTP request is submitted, the status field returned is 1, indicating that the task has been submitted but the operation has not been performed yet. To know the result of the previously submitted operation, it is necessary to use the query ESIM interface regularly to obtain various ESIM information and judge the value of the status field.

12.1 Query ESIM

12.1.1 URL

Request the following URL via HTTP **GET** method:

`http://host:port/goip_get_esims.html?version=xxx&username=xxx&password=xxx&ports=xxx`

Method:**GET**

host: device IP address

port: device web port, the default is 80, optional.

12.1.2 Parameter

Parameter	Description	Default	Required	Remark
version	API version	1.0	N	1.0: Compatible with previous API 1.1: Support for the description of this document
username	Device Username	None	Y	
password	Device password	None	Y	
ports	The specified port number (value from 1). The values are as follows: (1) all: all ports; (2) 2: Specify a single port; (3) 1-2, 4: Port numbers separated by short numbers, specifying multiple ports, where "-" indicates a continuous port number;	all	N	

12.1.3 Response

The value of the HTTP header parameter "Content-Type" is
 "application/json; charset=utf-8".

Request samples:

```
{
  "code": 200,
  "reason": "OK",
  "count": n,
  "stats": [
    {
      "port": xxx, ...
    }, ...
  ]
}
```

Property description:

Parameter	Data type	Description	Default	Required
code	int	Interface processing result code	None	Y
reason	string	Interface processing result reason	None	Y
count	int	Number of statistical results	None	Y
ports	query result array	An array of query results. Each result item in the array is as follows	None	Y

Description of the query result item:

Parameter	Data type	Description	Default	Required
port	int	Port number (value from 1)	None	Y
eid	string	eUICC ID	None	Y
esim_state	int	If or not the eid is obtained correctly: 0 - no 1 - yes	None	Y
profiles	array	The array of ESIM information for this port, with each ESIM information as follows	None	Y

ESIM information array item description:

Parameter	Data type	Description	Default	Required
slot	int	Slot number (value from 1)	None	Y
enabled	int	If or not the current profile is enabled: 0 - disabled 1 - enabled	None	Y
iccid	string	ESIM corresponding iccid	None	Y
ac	string	Activation code	None	Y
cc	string	Confirmation code	None	Y
provider	string	Name of service provider	None	Y
exust	int	If or not the current slot has been written into the profile: 0 - no 1 - yes	None	Y

op	int	Profile operation of the current slot: 0 - no operation 1 - writing 2 - deleting	None	Y
status	int	The current esim profile operation status, with the following values The values are as follows: 0 - No operation 1 - Operation request submitted 2 - Profile operation is in progress on the current port 3 - Operation error, the specific reason is described in the reason field	None	Y
reason	string	Causes of operational errors	None	Y

12.2 Write to ESIM

12.2.1 URL

Request the following URL via HTTP **POST** method:

http://host:port/goip_write_esims.html?version=xxx&username=xx&password=xx

Method:**POST**

host: device IP address

port: device web port, the default is 80, optional.

12.2.2 Parameter

Parameter	Description	Default	Required	Remark
version	API version	1.0	N	1.0: Compatible with previous API 1.1: Support for the description of this document
username	Device Username	None	Y	
password	Device password	None	Y	

12.2.3 POST data

The data of an HTTP POST request is a JSON array, where each data item is a structure of the following fields:

Parameter	Data type	Description	Default	Required
port	int	Port number (value from 1)	None	Y
slot	int	Slot number (value from 1)	None	Y
ac	string	Activation code	None	Y
cc	string	Confirmation code	None	Y

12.2.4 Response

The value of the HTTP header parameter "Content-Type" is
 "application/json; charset=utf-8".

Request samples:

```
{
  "code": 200,
  "reason": "OK",
  "count":n,
  "stats": [
    {
      "port":xxx, ...
    },...
  ]
}
```

Property description:

Parameter	Data type	Description	Default	Required
code	int	Interface processing result code	None	Y
reason	string	Interface processing result reason	None	Y
ports	result array	An array of results. Each result item in the array is as follows	None	Y

Description of the result item:

Parameter	Data type	Description	Default	Required
port	int	port number (value from 1)	None	Y
slot	int	slot number (value from 1)	None	Y
status	int	The current esim profile submission write operation status, the values are as follows: 1 - Write request has been submitted 2 - The current port is performing profile operation, submission failed 3 - Other errors, submission failed	None	Y

12.3 Delete ESIM

12.3.1 URL

Request the following URL via HTTP **POST** method:

```
http://host:port/goip_delete_esims.html?version=xxx&username=xxx&password=xxx
```

Method:**POST**

host: device IP address

port: device web port, the default is 80, optional.

12.3.2 Parameter

Parameter	Description	Default	Required	Remark
version	API version	1.0	N	1.0: Compatible with previous API 1.1: Support for the description of this document
username	Device Username	None	Y	
password	Device password	None	Y	

12.3.3 POST data

The data of an HTTP POST request is a JSON array, where each data item is a structure of the following fields:

Parameter	Data type	Description	Default	Required
port	int	port number (value from 1)	None	Y
slot	int	slot number (value from 1)	None	Y

12.3.4 Response

The value of the HTTP header parameter "Content-Type" is
[“application/json; charset=utf-8”](https://www.w3.org/Protocols/rfc1341/4_Content-Type.html).

Request samples:

```
{
  "code": 200,
  "reason": "OK",
  "count":n,
  "stats": [
    {"port":xxx,...}...]}
}
```

Property description:

Parameter	Data type	Description	Default	Required
code	int	Interface processing result code	None	Y
reason	string	Interface processing result reason	None	Y
ports	result array	An array of results. Each result item in the array is as follows	None	Y

Description of the result item:

Parameter	Data type	Description	Default	Required
port	int	port number (value from 1)	None	Y
slot	int	slot number (value from 1)	None	Y
status	int	The current esim profile submission deletion operation status, with the following values The values are as follows: 1 - The deletion request has been submitted 2 - The current port is performing a profile operation, and the submission failed 3 - Other errors, submission failed	None	Y