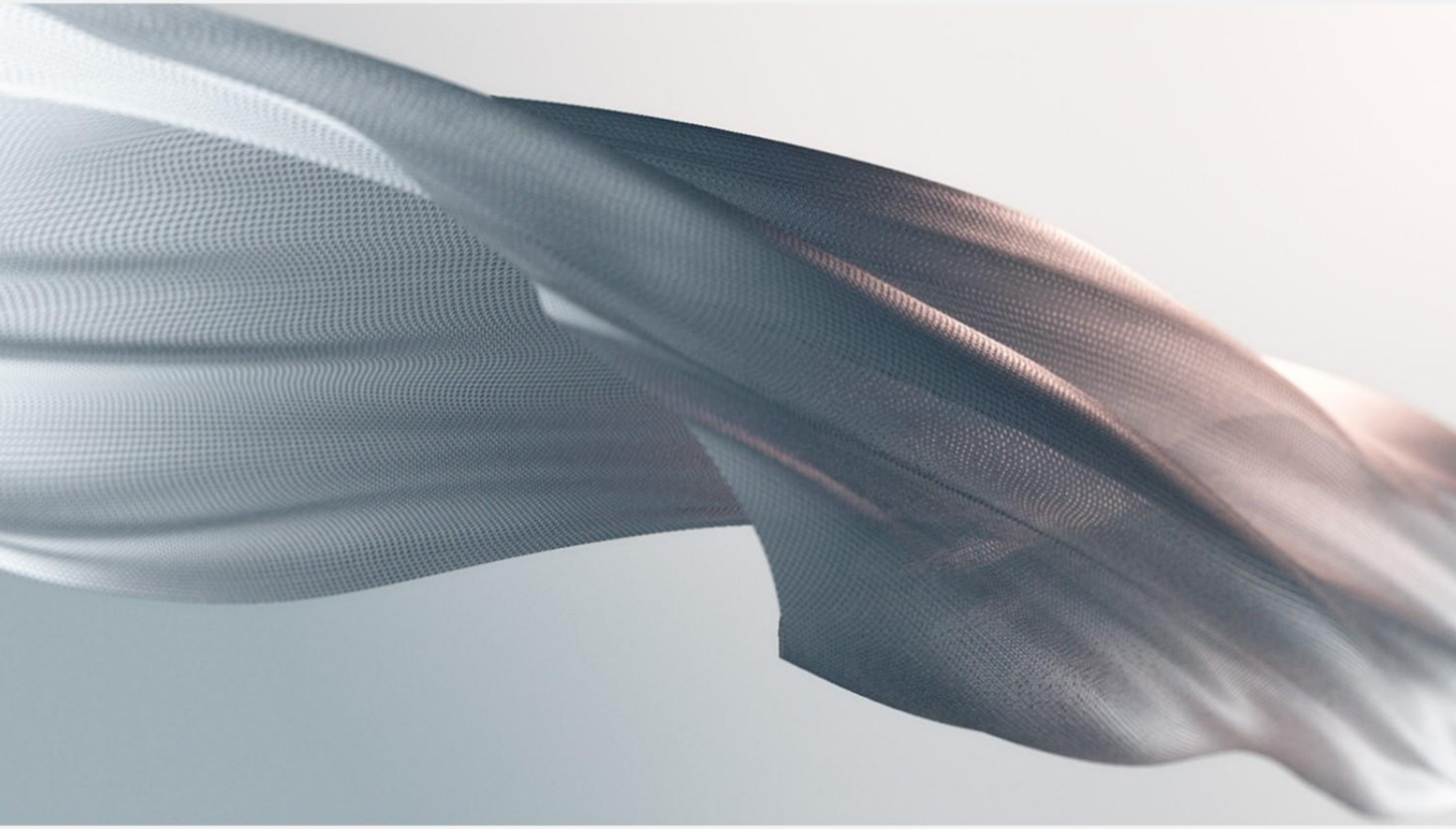
Ejoin
SMS HTTP API Simple Doc

Welcome to Ejointech API documentation. In order to use Ejointech for sending or receiving messages from any external application or system you have, we have provided the SMS API functionalities. It allows you to send and receive SMS via API.

If need development version please click “Ejoin Gateway HTTP API V2.1.0” to download.



Contents

Ejointech SMS API Reference (v2) **1**

1. SMS Sending **3**

1.1 Topology 3
1.2 SMS Sending Process 3
1.3 Message state 4

1.3.2.2 parameter **10**

2. SMS Forwarding **12**

2.1.1 SMS Forward URL 12
2.1.2 Parameter 13
2.1.3 Data 13

3.Device Configuration **14**

Notes: the minimum interval is 30s 14

4.Status Notification **14**

4.1 Port Status 15
4.2 Device Status 18
4.3 SMS Receive 20
4.4 SMS Send 21
4.5 Sim Cards Status 23
Have a question or an idea? We can help you! 24

1. SMS Sending

The server submits an SMS sending request to the device via HTTP over the network, carrying task information in JSON format data.

1.1 Topology

Since using HTTP Protocol, when the device and server are not within the same network, it's necessary to ensure network connectivity by using switches, port mapping on routers, or setting up a DMZ (Demilitarized Zone) to allow the server to establish a TCP connection to the device.

1.2 SMS Sending Process

The server submits SMS using GET/POST requests. The GET request is used for backward compatibility with the API defined in the document "Ejoin http SMS sending API.PDF," while the POST request can be used for sending long SMS messages (300 characters) or submitting multiple SMS sending tasks.

1.2.1 SMS sending

After receiving an SMS sending request from the server, the device creates one or more SMS sending tasks using the specified Task ID (used for correlating with subsequent status reports) and then responds to the server with a 200 OK response message.

1.2.2 Sending status report

When an SMS is successfully sent, fails, or times out, the device caches the result for that task. When certain conditions are met (reaching the cache capacity limit or the cache time expiration), it reports the results of one or more SMS tasks to the server via a POST request. If it's a bulk SMS, the device periodically sends the results of the bulk SMS at the specified time, continuing until all the bulk SMS numbers within the task have received their sending results.

1.2.3 Sending status query

During the task generation cycle (including the phase where task results are cached), the server can query the status of tasks using a GET request. This query can include information such as the number of successful sends, failure details (numbers and reasons), currently active numbers being sent, and more.

1.3 Message state

1.3.1 SMS Sending Task

1.3.1.1 SMS Sending Example URL

> http://host:port/goip_post_sms.html

> host: Device IP address
>
> port: Device's web backend port, defaulting to 80. Optional

1.3.1.2 HTTP parameter

Parameter	Description	Default	Required
version	API version	1.0	N
username	Device Username	None	Y
password	Device password	None	Y

Note: All URL parameters only appear in GET requests. For POST requests, they will appear as JSON properties in the body section, as no further explanation is provided below.

1.3.1.3 send-sms message

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

> **{“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	Y	send-sms
sr_url	string	Status report forward url		N	
sr_cnt	int	Maximum Cache Count for Status Reports	100	N	When the Cache count reaches this value, immediately send the Cache reports and reset the timer to 0
sr_prd	int	Maximum Cache Time for Status Reports	30s	N	If this time limit is exceeded, send the reports immediately, even if a sufficient number of reports haven't been received, and then reset the timer.
sms_url	string	SMS forward URL		N	

sms_cnt	int	Max number of SMS can keep in cache	1	N	1. When the Cache count reaches this value, immediately send the Cache SMS messages and reset the timer. 2. In order to maintain compatibility with older machines, this parameter must be set to a value greater than 1 to enable the Cache of received SMS messages and utilize the new sending mechanism.
sms_prd	int	Max time SMS can keep in cache	30	N	If this time limit is exceeded, send the SMS messages immediately, even if an insufficient number of messages have been received, and then reset the timer.
task_num	string	Total task	1	Y	
tasks	array	Specific SMS send task	None	Y	

Attributes in task array state:

Parameter	**Data type**	**Description**	**Default**	**Required**
tid	int	Task ID	None	Y

from	string	One or more sending ports (starting from Channel 1), connected by commas or hyphens	Choose by device	N
to	string	recipient telephone number (or numbers separated with comma)	None	Y
flash_sms	int	Specific this SMS if flash sms or not. 0 = normal SMS , 1 = SMS will be sent as flash message	None	N
sms	string	your SMS message	None	Y
chs	string	Character coding set （utf8|base64）	utf8	N
coding	int	Specific the SMS codec: 0:not assign 1:USC2 2:7bit	0	N
smsc	string	Store the SMSC number.		
intvl	string	Interval of 2 SMS while device sending SMS (in milliseconds)	“0”	N
tmo	int	Maximum time for waiting for the SMS sending results	30	30
sdr	int	Whether to enable SMS delivery status reports:	0	N

		1 for enable, 0 for disable.		
fdr	int	Whether to enable failure details reports: 1 for enable, 0 for disable.	1	N
dr	int	Whether to enable SMS delivery reports: 1 for enable, 0 for disable.	0	0
sr_prd	int	Sending status report period.(in seconds): 0 for not enabled, >0 for enabled. This parameter controls the interval at which status reports are generated for individual tasks. Just can control the current task sending status report period.	60	N
sr_cnt	int	Total sending status in one report message, it will use Default while stetted less than 1	10	N

Notes:

1. When no port is specified, the system automatically selects a port to send a message to each recipient.
2. When a port is specified and there is only one recipient, the message will be

sent from each specified port.

3. When a port is specified and multiple recipients are specified, the system selects the specified port to send a message to each recipient.

1.3.1.4 Task received reply

The response data (the body section of the HTTP response) is a JSON-formatted string consisting of one or more task statuses. The value of the HTTP header parameter "Content-Type" is "application/json;charset=utf-8."

> **{ “ code ” :200, “ reason ” ： “ OK ”, “ type ” : “ task-status ” , “status”:[{“tid”:tid_1, “status”：“0 OK”},…, {“tid”:tid_n, “status”：“2 Invalid Port”}]}**

Attributes 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	None	Y

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

1.3.2 Task send report

1.3.2.1 URL

Webpage configure or set in *send-sms message*.

1.3.2.2 parameter

Parameter	Description	Description	Required	Remark
version	API version	1.1	Y	

1.3.2.3 status-report message

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’

> **{“type”:“status-report”,“rpt_num”:n,“rpts”:[{“tid”:tid_1, …},…, {“tid”:tid_n, …}]}**

Attributes state：

Parameter	Data type	Description	Default	Required

type	string	message type	None	Y（status-report）
rpt_num	string	report number	1	Y
rpts	array	Detail status report	None	Y

Attributes of parameter ‘rpts’ state:

Parameter	**Data type**	**Description**	**Default**	**Required**
tid	int	related task ID	None	Y
sending	int	total number of sms currently being sent.	None	Y
sent	int	total number of SMS successfully sent.	None	Y
failed	int	The total number of SMS failed to send.	None	Y
unsent	int	total SMS in cache waiting to send	None	Y
sdr	array	Successfully send details(one list for one num) Success report is an array, [0]: recipients num index(based on group sending),int [1]: num, string [2]: SMS sending port(1A,2B,...), string [3]: SMS send timestamp in UTC time, int	None	N
fdr	array	Fail send details(one num one list)	None	N

		[0]: recipients num index(based on group sending),int [1]: num, string [2]: SMS sending port(1A,2B,...), string [3]: SMS send timestamp in UTC time.int [4]: Progress reason, code+ details[refer to 1.0 API] [5]:operator reason, code+ details. valid while [4] progress reason have detail.		

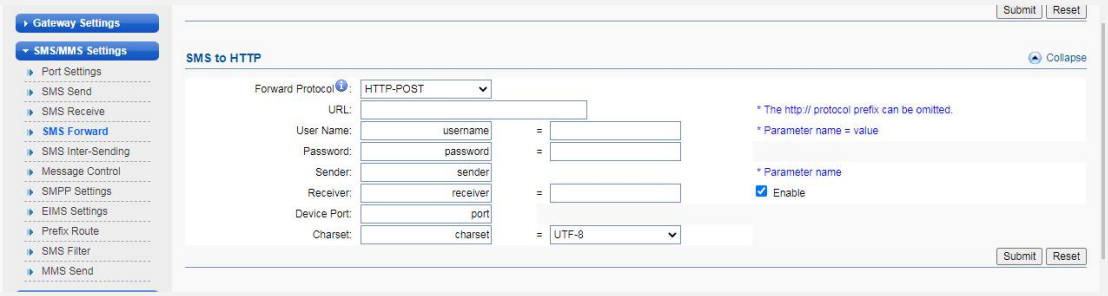
Notes:
1. **'tid'** corresponds to the Task ID in the task request.
2. '**sending'**, '**sent'**, and **'failed'** are cumulative statistics for that task.
3.'**sdr'** (Success Details Record) is a record between two reports, and the server should retain previous data.
4. '**fdr'** (Failure Details Record) is a record between two reports, and the server should retain previous data.

2. SMS Forwarding

2.1.1 SMS Forward URL

Configured and specified on the device SMS forwarding page.

The forwarding URL can be set on the SMS send page.(ejoin gateway >> URL)



2.1.2 Parameter

Parameter	Description	Default	Required
Username	Server login account	No	No
Password	Server login password	No	No
Sender	Telephone number of SMS sender (for inbox)	No	No
Receiver	telephone number of SMS receiver ; if not specified, it will carry the SIM card number that received the SMS	No	No
Port	The original port number that receives SMS.	No	No
Charset	Charset	UTF-8	Yes

2.1.3 Data

If the Charset parameter is not configured, the content defaults to UTF-8 text.

```
POST /myweb/goip?username=root&password=root&port=4&sender=13510956503 HTTP/1.1
Accept: */*
Accept-Language: en
User-Agent: Ejoin SMSC/1.0
Host: 192.168.1.31:8080
Connection: Close
Content-Type: application/octet-stream; charset=utf-8
Content-Length:   78

Sender: 13510956503
SMSC: 8613010888500
SCTS: 14072322113832

Hello,world!
```

Sender: The original sender.

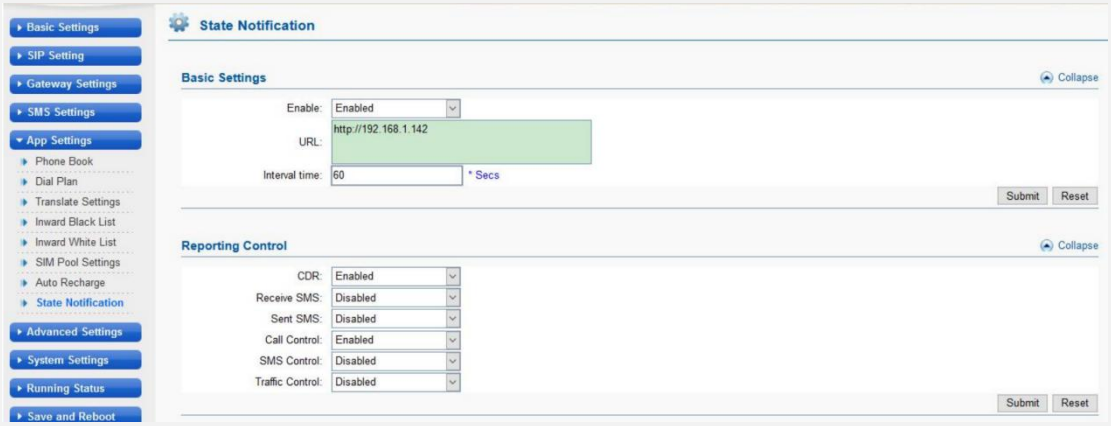
SMSC: Short Message Service Center number.

SCTS: Timestamp when the device received this SMS.

The string following an empty line is the original SMS content.

3.Device Configuration

> **Notes: the minimum interval is 30s**



4.Status Notification

Status notification content is controlled by reporting control switch and related function switch.

1. CDR：need enable CDR report (default enable);
2. SMS receive: need enable received sms report;
3. SMS sent: need enable send sms report;
4. Call control data: need enable call control data report(default enable),also need enable related function, as follows:

a) Call duration control need configured as FLASH;

b) Call numbers control need configured as FLASH;

c) Talk number control needs configured as FLASH;

5. SMS control data: need enable SMS control data report, and enable SMS control function;

6. Traffic control data: need enable traffic control data report;

After setting up the report control, the device sends the device status to the configured URL by HTTP POST request.

7. If the status of a port on the device has changed, reports port status
8. The configured interval was reached,report device status
9. Device receives SMS, report SMS content
10. Device sends SMS, reports SMS content
11. When the device has a call, report call billed

4.1 Port Status

When the status of a port on the device has changed, instantly send the data of type port-status to the server.

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

> **{"type":"port-status", "port":"1.02", "sim":"", "seq":1, "st":11, "imei":"353689530707903", "iccid":"", "imsi":"", "sn":"", "opr":"0 CHINA MOBILE", "bal":"0.00", "sig":0, "tot_dur":"0/-1", "mon_dur":"0/-1", "day_dur":"0/-1"}**

Components state:

Parameter	Type	Description	Default	Required
type	Strin	Message type	N	Y（port-status）
Port	String	Describe the current working port and SIM slot,like	N	Y

		1.01,1.02.....32.04		
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: Sim Card Be Detected 12：User locked 13：Port inter-calling 14：Inter-calling holding	N	Y
active	String	0：Not currently card 1：Currently card	N	Y
Bal	Floating point	SIM card balance (yy.mm)	N	Y
opr	String	SIM card operator name and ID valid while parameter “st”equal 3 or 4	None	N
sn	String	SIM Cards number	None	N
imei	String	IMEI of module	None	N

imsi	String	IMSI of SIM card	None	N
iccid	String	ICCID of SIM card	None	N
sig	String	Signal of module	None	N
tot_dur	String	Accumulated call duration Statistics "tot_dur":"n/m",n Represents the number of minutes used, m represents the total number of minutes, and -1 represents unlimited	None	N
mon_dur	string	Monthly Call duration statistics	None	N
day_dur	string	Daily call duration statistics	None	N
mcn_tot	string	Accumulative Total call number "mcn_tot":"n/m",n represents the number of numbers used,m Represents the total number of configured, -1 represents unlimited	None	N
mcn_mon	string	Monthly call number	None	N
mcn_day	string	Daily call number	None	N
mtn_tot	string	Accumulative total call talk number "mtn_tot":"n/m",n represents talk number of used,m represents total number of configured,-1 represents unlimited	None	N
mtn_mon	string	Monthly talk number	None	N

mtn_day	string	Daily talk number	None	N
smsctl_tot	string	Accumulative total send SMS "smsctl_tot":"n/m",n represents number of SMS sent，m represent The number SMS can be sent, -1 represents unlimited	None	N
smsctl_mon	string	Monthly SMS Send	None	N
smsctl_day	string	Daily SMS Send	None	N
dataflow_tot	string	Accumulative total traffic(MB)	None	N
dataflow_day	string	Daily total traffic(MB)	None	N
dataflow_24hours	string	Total traffic within the last 24 hours(MB)	None	N
dataflow_1hour	string	Total traffic within the Last 1 hours(MB)	None	N
dataflow_last	string	The traffic consumed by the Finally access(MB)	None	N
dataflow_url	string	The last URL accessed(MB)	None	N

4.2 Device Status

The device periodically sends the device status of type dev-status to the server.

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”.

> **{"type":"dev-status","seq":3,"expires":60000,"mac":"00-30-f1-00-6d-b1",**
>
> **"ip":"192.168.1.44","ver":"532-600-999-041-100-000","max-ports":32,"max-slot":4,"status":[{"port":"1.02","sim":"","seq":11,"st":3,"imei":"353689530707903", "iccid":"89860042191782899541", "imsi":"460000990123698", "sn":"",**
>
> **"opr":"46000 CHINA MOBILE", "bal":"0.00", "sig":31, "tot_dur":"1/-1", "mon_dur":"1/-1", "day_dur":"1/-1"},{"port":"2.02",…},…, {"port": "32.02",…}]}**

Components state:

Parameter	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 Address	None	Y
ip	String	Device IP address	None	Y
ver	String	Device software version	None	N
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

4.3 SMS Receive

When the device receives SMS, it will send a report with type reve-sms to the server.

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”.

> **{"type":"recv-sms","sms_num":1,"sms":[[0,"1.02",1548407412,"8613715266978","","U2VuZGVyOiA4NNTEwOAzDQpTTVNDOiA4NjEzODAwNzU1NTAwDQpTQ1RTOiAxOTAxM**
>
> **jUDQp5YXE="]]}**

Components state:

Parameter	**Type**	**Description**	**Default**	**required**
type	String	Message type	None	Y（recv-sms）
sim_src	int	The source of SIM card： • 0 – local card; • 1 – remote card;	0	Y
sms_num	String	SMS number	None	N
sms	Array	SMS array	None	N

The sms itself is also an array（to save network traffic），in turn：

[0]: the symbol whether is a deliver report,0：SMS，1：deliver report

[1]: receive port and slot：values are related to the sin_src field ,when the sim_src is local card, value is “port”.”local slot”, For example 1.01，2.02，…,32.04，among

1.01 .02 .03 .04 each represent A B C D;

When sim_src is remote card, value is “port.simpool.slot”,example “1.sp128.016”

[2] : the timestamp when the device received the SMS/report

[3] : sender（if it’s deliver reports, is the message service center number）

[4]: receiver（if it’s deliver reports, is the receiver of the original SMS）

[5]: SMS content：the contents are related to the value [0],if it’s an Ordinary SMS,the default is BASE64 encoding, and it need to decoded;if it’s an deliver report,the content is“n xxxx”, among n is deliver report status code,0 represent succeed,other value are dependent on the communication module used，xxxx is the operator timestamp.

4.4 SMS Send

When the device send SMS, it will send a report with type reve-sms to the server. 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”.

> **{"type":"sent-sms", "port":"2.01", "seq":18, "sms_tid":2, "src_type":2, "src_ip":"", "src_port":0, "status":"0 ", "time":1562235746, "dur":330, "recver":"12122222222", "smsc":"", "charset":"UTF-8", "data":"test sms"}**

Components state:

Parameter	**Type**	**Description**	**Default**	**Required**	Note
type	string	Message type， Values is sent-sms	None	Y	
sim_src	Int	The source of SIM card： • 0 – local card; • 1 – remote card	0	Y	

port	String	Port and slot，values are Related to the sim_src field When the sim_src is local card,value is“port”.” local slot”, For example 1.01, 2.02，…, 32.04,among 1.01 .02 .03 .04 each represent A B C D; When sim_src is remote Card, value is “port.simpool.slot” example“1.sp128.016”	None	Y	
seq	Int	Device status message Sequence number start from 1. When reported device status, not sequence number		Y	
sms_tid	Int			Y	
src_type	Int	SMS submitter type 0：ETMS 1: EIMS 2: WEB 3: HTTP 4: SMPP 5: SIP	None	Y	
src_ip	String	SMS submitter IP	“”	N	
src_port	Int	SMS submitter port		N	
status	Stri	The status code and describe to		Y	

	ng	send SMS “code desc” code=0：succeed			
time	Int	The time stamp to send SMS（UTC）		Y	
dur	Int	The time it consume to send SMS(ms)	0	Y	
receiver	String	Receiver		Y	
smsc	String	SMS service center number	“”	N	
charset	String	SMS unicode UTF-8|BASE64	UTF-8	N	BASE64: Base64 encodes the Unicode source and removes the newline
data	String	SMS content		Y	

4.5 Sim Cards Status

Get All Sim Cards Status:

> http://ip/goip_get_status.html?username=root&password=root&all_sims=1

Get specific sim card status

> http://ip/goip_get_status.html?username=root&password=root&port=1

Disable/enable ports

> http://ip/goip_send_cmd.html?username=root&password=root&op=unlock&port=2.01

Switch sim cards

> http://ip:port/goip_send_cmd.html?username=root&password=root&port=1.01&op=switch

Switch sim cards of all ports (for example 64 ports)

> http://ip/goip_send_cmd.html?port=1.01-64.01&op=switch

Delete sms:

> http://ip:port/goip_get_sms.html?username=root&password=root&sms_id=xxx&sms_num=xxx

Have a question or an idea? We can help you!

If you have any technical questions regarding API feel free to contact our Support Team.