Instructions for Using New API

Change Logs

Version 1.1 2019-10-16

1. Add ‘get_status’ to API.
2. Add more examples.

Version 1.0 2016-12-8

1. Add ‘request_status_report’ parameter to SMS request;
2. Add pushing feature;
3. Add ‘slot, callstate, signal, gprs’ parameter to API of ‘Get Port Information’;
4. Add port API.

Version 0.5 2015-7-22

1. Add CDR API;
2. Add STK API.

Version 0.4 2015-4-24

1. Change the format of sending SMS request; allow each number to include a user id;
2. Add an API for querying the current number of SMS in the waiting queue;
3. Add the ‘user_id’ parameter to ‘Query SMS Result’.

Version 0.3 2015-3-20

1. Add ‘DELIVERED’ status to ‘Send SMS Result’.

Version 0.2 2015-1-22

1. Add ‘user_id’ to ‘Send SMS Request’.

Version 0.1 2014-12-25

First Version

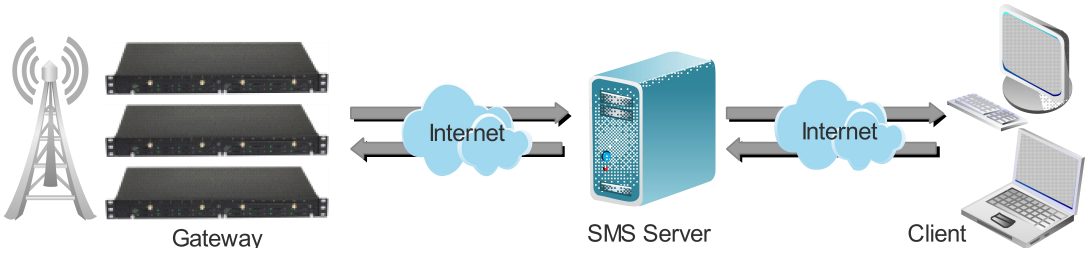
Contents

1 Introduction of API

1.1 Application Scenarios

API enables an SMS server to communicate with lots of gateways, and then users can send and receive SMS through this server.

1.2 Network Topology



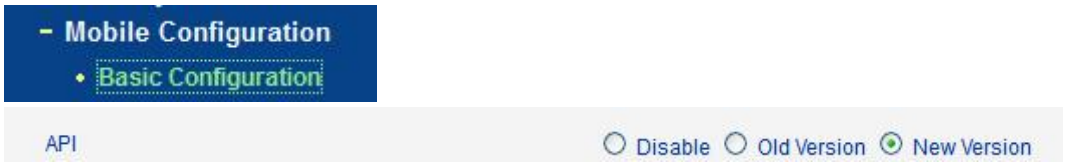
1.3 Key Features

- Support sending and receiving of SMS/USSD
- Support batch sending of SMS
- Support 'SMS Delivery Status'
- Support obtaining of basic port information

1.4 Before You Start

1) The API is based on HTTP and JSON. So please check how to send HTTP request and how to encode/decode JSON data before you write an application with this API. This document takes cURL as an example to demonstrate how to use this API, and please convert it to your own programming language if there is a need.

2) Please enable the API function before your test. On the web interface of the gateway, select Mobile Configuration → Basic Configuration in the menu bar, and then select new-version API. Only version 1102 or later version of the gateway supports the new-version API.



3) Something about SMS.

a) The gateway supports 2 types of encoding, namely GSM 7 bit and UCS2. GSM 7bit is suitable for sending messages that contain ASCII and some Latin alphabets, while UCS2 is suitable for sending anything, including Chinese, Korean, Japanese and even emoji .

b) For GSM 7bit, it is possible to send up to 160 characters (packed in up to 140 octets) in one SMS message. But for UCS2, one SMS message could only contain 70 characters at maximum.

c) The gateway supports Concatenated_SMS, which means the gateway will split your long messages into smaller SMS.

2 Send SMS

2.1 Request

```
POST https://gateway_ip/api/send_sms
```

2.2 Request Parameters

Parameter	Type	Description
Required Parameter		
text	String	The content of SMS, support grammar like #param#
param	Array of Object	Each element is an object, which can be any of the following : **number**: a digit string no more than 24 bytes **text_param**: an array of string. Each element is used to replace #param# in the **text** **user_id**: an integer greater than or equal to 0. It is recommended the user_id is an unique value for each SMS. user_id is the index of SMS, which is used to identify the SMS while user query SMS sending result. (refer to 3 “Query SMS sending Result”)
Optional Parameter		
port	Array of Integer	Number of port for sending SMS Each port number should be an integer, ranging from 0 to 31
encoding	String	It can be ‘unicode’or ‘gsm-7bit’ Default is ‘unicode’
request_status_report	Boolen	It is used to determine whether to use SMS status report. It can be true or false Default is true

2.3 Response Parameter

Parameter	Type	Description
error_code	Integer	Codes that may be returned: 202: Request has been accepted and will be processed later; 400: Request format is not valid 413: the count of telephone numbers is more than 128 or text content is over 1500 bytes 500: other errors 550: No available port for sending sms.
sn	String	SN of the gateway
sms_in_queue	Integer	Number of SMS messsages waiting for being processed
task_id	Integer	Used to stop corresponding sending task later

2.4 Example of Sending SMS

Simple Request:

```
curl -k --anyauth -u admin:admin -d
'{"text":"ye","param":[{"number":"10086"}]}' -H "Content-Type:
application/json"  https://gateway_ip/api/send_sms
```

Request with #param:

```
curl -k --anyauth -u admin:admin -d
'{"text":"#param#","port":[2,3],"param":[{"number":"10086","te
xt_param":["bj"],"user_id":1},{"number":"10086",
"text_param":["ye"],"user_id":2}]}' -H "Content-Type:
application/json"  https://gateway_ip/api/send_sms
```

Note:

1. The example above comes with the default username and password. Use the real username and password of your own gateway.

2. Use the real ip of your gateway to replace the gateway_ip.

Response:

{"error_code":202,
"sn":"xxxx-xxxx-xxxx-xxxx","sms_in_queue":2,"task_id":2}

3 Query SMS Sending Result

3.1 Request

```
POST https://gateway_ip/api/query_sms_result
```

3.2 Request Parameters

Parameter	Type	Description
Optional Parameter		
number	Array of Strings	The number of the string arrays should not exceed 32, while the length of each string should not be more than 24 bytes.
port	Array of Integers	The port(s) for sending SMS Each port should be an integer ranging from 0 to 31
time_after	String	String like "YYYY-MM-DD HH:MM:SS" Query the records of SMS messages sent after this time
time_before	String	String like "YYYY-MM-DD HH:MM:SS"; Query the records of SMS messages sent before this time
user_id	Array of Integers	Each user ID here is used to match the user ID carried in **Send Sms Request** This user_id is the unique value which is set during sending SMS. (refer to “send SMS”, “Request parameters”) **This parameter is recommended for practice.**

3.3 Response Parameter

Parameter	Type	Description
error_code	Integer	codes that may be returned: 200: Request has been accepted and processed 400: Request format is not valid 413: the count of telephone numbers is more than 32 500: other errors
sn	String	SN of the gateway
result	Array of Objects	Results of SMS sending; Each result includes the following information: **port**: port for sending SMS **number**: destination number of the sent SMS **user_id**: match with the user ID carried in SMS sending request **time**: the time of sending SMS status: sending status, which could be FAILED, SENDING, SENT_OK and DELIVERED **count**: count of SMS segments. A long SMS message is divided into several segments and then is sent. **succ_count**: count of SMS segments that are sent successfully **ref_id**: the first reference ID of this SMS, which is used to match SMS delivery status. The range of reference ID is from 0 to 255. Reference ID is generally used together with telephone number, port number and even sending time to match a delivery status imsi: IMSI of the SIM card.

3.4 Example of Querying SMS Sending Result

Request:

```
curl -k --anyauth -u admin:admin -d '{"user_id":[1,2]}' -H
"Content-Type: application/json"
https://gateway_ip/api/query_sms_result
```

Response:

{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx","result":[{"port":0,"user_id":1,"number":"12351","time":"2014-12-21 12:06:01","status":"SENT_OK","count":3, "succ_count":3, "ref_id":12,"imsi":"460004642148063"}]}

Sms result can also be pushed to your application, like

{"sn":"xxxx-xxxx-xxxx-xxxx","sms_result":[{"port":1,"number":"10086","time":"2016-07-12 01:46:02","status":"DELIVERED","count":1,"succ_count":1,"ref_id":215,"imsi":"460004642148063"}]}

4 Query SMS Delivery Status

4.1 Request

```
POST https://gateway_ip/api/query_sms_deliver_status
```

4.2 Request Parameter

Parameter	Type	Description
Optional Parameter		
number	Array of Strings	The count of telephone numbers should not be more than 32. The length of each number should not exceed 24 bytes
port	Array of Integers	The port(s) for sending SMS messages Each port should be an integer ranging from 0 to 31
time_after	String	String like "YYYY-MM-DD HH:MM:SS" Query the records of SMS messages sent after this time
time_before	String	String like "YYYY-MM-DD HH:MM:SS" Query the records of SMS messages sent before this time

4.3 Response Parameter

Parameter	Type	Description
error_code	Integer	Codes that may be returned: 200: Request has been accepted and processed

		400: Request format is not valid 413: the count of telephone numbers is more than 32 500: other errors
sn	String	SN of the gateway
result	Array	Results of delivery status; each result includes the following information: **port**: port for sending SMS messages **number**: destination number **time**: time of sending SMS messages **ref_id**: reference ID which is used to match SMS sending result **status_code**: delivery status which ranges from 0 to 255; 0: it means the SMS has been received by peer 32 ~ 63: temporary error 64~255: permanent error For more information about status code, please search "SMS Status Report. imsi: IMSI of the SIM card.

4.4 Example of Querying SMS Delivery Status

Request:

```
curl -k --anyauth -u admin:admin -d '{"number":["12341234"],
"port":[1,2,3], "time_after":"2014-12-12 19:29:19",
"time_before":"2014-12-12 19:29:19"}' -H "Content-Type:
application/json"
https://gateway_ip/api/query_sms_deliver_status
```

Response:

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx","result":[{"port":
0, "number":"12341234","time":"2014-12-21 12:06:01","ref_id":12,
"status_code":0,"imsi":"460004642148063"}]}
```

Deliver status can also be pushed to your application, like

```
{"sn":"xxxx-xxxx-xxxx-xxxx","sms_deliver_status":[{"port":1,"n
umber":"10086","time":"2016-07-12
```

15:46:53","ref_id":215,"status_code":0,"imsi":"460004642148063
"}]}

5 Query Number of SMS Messages to Be Sent

5.1 Request

GET https://gateway_ip/api/query_sms_in_queue

5.2 Request Parameter

None

5.3 Response Parameter

Parameter	Type	Description
error_code	Integer	Codes that may be returned: 200：Request has been accepted and processed 500: Other errors
sn	String	SN of the gateway
in_queue	Integer	Number of SMS messages waiting to be sent

5.4 Example of Querying Number of SMS Messages to Be Sent

Request:

http://gateway_ip/api/query_incoming_sms

Response:

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx","in_queue":0}
```

6 Receive SMS

6.1 Request

GET https://gateway_ip/api/query_incoming_sms

6.2 Request Parameter

Parameter	Type	Description
Optional Parameter		
incoming_sms_id	Integer	It is an integer greater than 0 When a SMS server/client request for SMS messages, the gateway will return the messages whose ID are greater than this value to SMS server/client Default value is 0
flag	String	Optional values of this parameter include: "unread": unread SMS messages (whose status will change to ‘read’ after they are read by this API) "read": SMS messages that have been read

		"all": SMS messages that are unread and read Default value is "unread"
port	Array of Integers	Port(s) for receiving SMS messages Each port should be an integer ranging from 0 to 31

6.3 Response Parameter

Parameter	Type	Description
error_code	Integer	Codes that may be returned: 200： it is legal request and gateway will return SMS messages according to the request 500: Other errors
sn	String	SN of the gateway
sms	Array of Object	SMS messages; each message include the following information: **incoming_sms_id**: message ID in gateway's database **port**: port of receiving SMS message **number**: telephone number of SMS sender **smsc**: telephone number of SMS center **timestamp**: time of receiving SMS message **text**: content of a SMS message
read	Integer	count of SMS messages that have been read
unread	Integer	count of SMS messages that are unread.

6.4 Example of Receiving SMS

Request:

```
https://gateway_ip/api/query_incoming_sms?flag=all
```

Response:

{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx","sms":[{"incoming_sms_id":1, "port":2, "number": "123456789","smsc":"+8613800123456","timestamp":"2014-12-09 17:11:18","text":"This is a test"},],"read":1, "unread":0}

Incoming sms can also be pushed to your application, like

{"sn":"da00-0030-1901-2817","sms":[{"incoming_sms_id":1,"port":1,"number":"6717","smsc":"+8613800757511","timestamp":"2016-07-12 15:46:18","text":"test"}]}

6.5 Suggestion

1. Use Push Service for getting SMS.

2. Incoming sms id should be increased every time. For example, First request, without incoming_sms_id is OK. But Seconde request, the incoming_sms_id should be the max id in the first response.

7 Send USSD

7.1 Request

POST https://gateway_ip/api/send_ussd

7.2 Request Parameter

Parameter	Type	Description
Required Parameter		
text	String	The content of SMS message to be sent; its length should not exceed 60 bytes. When command is "send", text should not be empty; when command is "cancel", text can be empty.

port	Array of Integers	Port(s) for sending USSD Each port should be an integer ranging from 0 to 31
Optional Parameter		
command	String	It can be "send" or "cancel"; default value is "send"

7.3 Response Parameter

Parameter	Type	Description
error_code	Integer	Codes that may be returned: 202: request has been accepted and processed. 400: it is an illegal request, for example, the command is “send” but text does not exist. 500: other errors
sn	String	SN of the gateway
result	Array	Results; each result include the following information: **Port**: port for sending USSD **Status**: its value may be any of the following 200: USSD is sent successfully 486: the port is busy currently, for example, SMS is being sending through the port 503: the port is not registered currently

7.4 Example of Sending USSD

Request:

```
curl -k --anyauth -u admin:admin -d
'{"port":[1,2,3],"command":"send","text":"*125#"}' -H
"Content-Type: application/json"
https://gateway_ip/api/send_ussd
```

Response:

{"error_code":202,"sn":"xxxx-xxxx-xxxx-xxxx","result":[{"port":0, "status":503},{"port":1, "status":503},{"port":2, "status":200}]}

8 Query USSD Reply

8.1 Request

GET https://gateway_ip/api/query_ussd_reply

8.2 Request Parameter

Parameter	Type	Description
Required Parameter		

port	Array	Port(s) for querying USSD reply Each port should be an integer ranging from 0 to 31

8.3 Response Parameter

Parameter	Type	Description
error_code	Integer	Codes that may be returned: 200: Request has been accepted and processed 400: Request format is not valid 500: Other errors
sn	String	SN of the gateway
reply	Array	Replys; each reply includes the following information: port: port for receiving USSD reply text: USSD reply

8.4 Example of Querying USSD Reply

Request:

```
https://gateway_ip/api/query_ussd_reply?port=1,2,3
```

Response:

```
{"error code":200,"sn":"xxxx-xxxx-xxxx-xxxx","reply":[{"port":
1, "text":" "},{"port":2, "text": "Test…"},{"port":3, "text":
""}]}
```

USSD can also be pushed to your application, like:

```
{"sn":"da00-0030-1901-2817","ussd":[{"port":1,"text":"Thank
you!"}]}
```

9 Stop SMS Sending Task

9.1 Request

```
GET https://gateway_ip/api/stop_sms
```

9.2 Request Parameter

Parameter	Type	Description
Required Parameter		
task_id	Integer	The ID of the task to be stopped, which corresponds with the task-id carried in the response of SMS sending request

9.3 Response Parameter

Parameter	Type	Description
error_code	Integer	Codes that may be returned: 200: SMS sending task has been stopped 404: SMS sending task cannot be found 500: Other errors
sn	String	SN of the gateway

9.4 Example of Stopping SMS Sending Task

Request:

```
https://gateway_ip/api/stop_sms?task_id=1
```

Response:

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx"}
```

10 Get Port Information of Gateway

10.1 Request

GET https://gateway_ip/api/get_port_info

10.2 Request Parameter

Parameter	Type	Description
Required Parameter		
info_type	Array of Strings	Strings; each string cloud be any of the following strings: type,imei,imsi,iccid,number,reg,slot,callstate, signal,gprs,remain_credit,remain_monthly_credit,remain_daily_credit,remain_daily_calltime,remain_hourly_call time,remain_daily_connect
Optional Parameter		
port	Array	Port(s) whose information is to be queried Each port should be an integer ranging from 0 to 31.

10.3 Response Parameter

Parameter	Type	Description
error_code	Integer	Codes that may be returned: 200: Request has been processed. 400: Request format is not valid. 500: Other errors
sn	String	SN of the gateway
info	Array	Each piece of port information is returned according to the query request; **port**: the port which is queried **type**: could be GSM, CDMA, WCDMA, LTE or UNKNOWN (when type is not recognized) **imei**: IMEI of this port **imsi**: IMSI of the SIM card inserted into this port **iccid**: ICCID of the SIM card inserted into this port **number**: the mobile number of this SIM card **reg**: register status of this port, which could be POWER_OFF, NO_SIM, PIN_REQUIRE, PUK_REQUIRE, UNREGISTER, SEARCHING_NETWORK, REGISTER_OK or UNKNOWN.

		slot: slot of this port (when the gateway is a multi-sim gateway); slot number ranges from 0 to 3; 255 will be returned when the gateway is not a multi-sim gateway. **callstate**: the current state of this port, which could be Idle, Processing, Ringing, Active, Alerting, Call Waiting, Call holding or Unknown. **signal**: signal strenth of this port, ranging from 0 to 31 **gprs**: gprs which is attached or detached to this port **remain_credit:** total remain credit of this port **remain_monthly_credit:** monthly remain credit of this port **remain_daily_credit:** daily remain credit of this port **remain_daily_calltime:** daily remain call time of this port **remain_hourly_calltime:** hourly remain call time of this port **remain_daily_connect:** daily remain connect times of this port

10.4 Example of Getting Port Information

Request:

```
https://gateway_ip/api/get_port_info?port=1,2,3&info_type=imei,
imsi,iccid,smsc,type,number,reg,slot,callstate,signal,gprs,rem
ain_credit,remain_monthly_credit,remain_daily_credit:,remain_d
aily_calltime,remain_hourly_calltime,remain_daily_connect
```

Response:

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx","info":[{"port":1,
"type":"WCDMA","imei":"863070017005173","imsi":"","iccid":"","
smsc":"","number":"","reg":"NO_SIM","callstate":"Idle","signal
":0,"gprs":"detached","remain_credit":"1000.00","remain_monthl
y_credit":"500.00","remain_daily_credit":"300.00","remain_dail
y_call_time":"100","remain_hourly_call_time":"10","remain_dail
y_connected":"100"},{"port":2,
"type":"GSM","imei":"358967042917201","imsi":"460016529802215",
"iccid":"89860114840400428150","smsc":"+8613010868500","number
":"","reg":"REGISTER_OK","callstate":"Idle","signal":0,"gprs":
"detached","remain_credit":"1000.00","remain_monthly_credit":"
500.00","remain_daily_credit":"300.00","remain_daily_call_time
":"100","remain_hourly_call_time":"10","remain_daily_connected
```

":"100"},{"port":3,
"type":"GSM","imei":"358967042917201","imsi":"","iccid":"","sm
sc":"","number":"","reg":"NO_SIM","callstate":"Idle","signal":
0,"gprs":"detached","remain_credit":"1000.00","remain_monthly_
credit":"500.00","remain_daily_credit":"300.00","remain_daily_
call_time":"100","remain_hourly_call_time":"10","remain_daily_
connected":"100"}]}

11 Set Port Information of Gateway

11.1 Request

GET https://gateway_ip/api/set_port_info

11.2 Request Parameter

Parameter	Type	Description
Required Parameter		

action	String	Action could be slot, reset or power Slot: choose a slot (when the gateway is a multi-sim gateway) Reset: reset the gsm/cdma/wcdma module Power: turn on or turn off the gsm/cdma/wcdma module imei: to modify IMEI, param should be IMEI number: to modify number in SIM, param should be phone number lock: to lock SIM card, param should be PIN of the SIM unlock: to unlock SIM card, param should be PIN of the SIM block: block SIM unblock: unblock SIM CallForward: to set call forward, see set call forward for detail CheckCallForward: to send checking call forward command
param	String	Slot: slot of this port(when the gateway is a multi-sim gateway), ranging from 0 to 3; Power: could be “on” or “off”.
port	Integer	Port number is an integer ranging from 0 to 31

11.3 Response Parameter

Parameter	Type	Description
error_code	Integer	Codes that may be returned: 200: Request has been processed 400: Request format is not valid. 500: other errors
sn	String	SN of the gateway

11.4 Example of Setting Port Information

Request:

```
https://gateway_ip/api/set_port_info?port=1&action=power&param
=off
```

Response:

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx”}
```

12 Get CDR

12.1 Request

POST https://gateway_ip/api/get_cdr

12.2 Request Parameter

Parameter	Type	Description
Required Parameter		
Optional Parameter		
port	Array	Port(s) whose CDR is to be queried; Each port number should be an integer ranging from 0 to 31.
time_after	String	String like "YYYY-MM-DD HH:MM:SS"; Query the CDRs which are sent after this time.
time_before	String	String like "YYYY-MM-DD HH:MM:SS"; Query CDRs are sent before this time.

12.3 Response Parameter

Parameter	Type	Description
error_code	Integer	Codes that may be returned: 200: Request has been processed 400: Request format is not valid 500: Other errors
sn	String	SN of the gateway
cdr	Array	Each CDR will be returned according to the query. **port**: port number; **start_date**: start time of the call; **answer_date**: answer time of the call; **duration**: call duration **source_number**: source number **destination_number**: destination number **direction**: call direction which could be gsm->ip, ip->gsm or callback;

		ip: the source ip of an ip->gsm call **codec**: it could be： G.711U, G.723.1, G.711A or G.729AB; **hangup**: the party to hang up the call ; it could be the called party, the calling party or the gateway; **gsm_code**: the code of the reason why the GSM side hangs up the call; **bcch**: BCCH used for this call

12.4 Example of Getting CDR

Request:

```
curl -k --anyauth -u admin:admin -d '{"port":[2,3]}' -H
"Content-Type: application/json"  http://gateway_ip/api/get_cdr
```

Response:

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx","cdr":[{"port":2,
"start_date":"2015-07-21 16:35:20","answer_date":"2015-07-21
16:35:31","duration":3,"source_number":"1010","destination_num
ber":"6717","direction":"ip->gsm","ip":"172.16.100.136","codec
":"G.711U","hangup":"called","gsm_code":16,"bcch":""}]}
```

13 Query STK Info

13.1 Request

```
GET https://gateway_ip/GetSTKView
```

13.2 Request Parameters

Parameter	Type	Description
Required Parameter		
port	Integer	

13.3 Response Parameter

Parameter	Type	Description
title	String	The title of STK
text	String	The content of prompt during the query of STK information
input_type	Integer	0: display only 1: desktop 2: select item 3. require "yes" or "no" response 4: require one digit input 5: require one character input 8: require input of only digit(s) 9: require input of digit(s) without echo 10: require input of character(s) 11: require input of character(s) without echo

		12: locked status
frame_id	Integer	The frame id of current STK view
item	Object	It is included only when input_type is 1 or 2
Item. item_id	Integer	The id of item
Item .item_string	String	The name of item

13.4 Example of Querying STK Info

Request:

```
http://gateway_ip/GetSTKView?port=0
```

Response:

```
{"title":"神州行天地
","item":[{"item_id":1,"item_string":"item1_text"},{"item_id":
2,"item_string":"itme2_text"},{"item_id":3,"item_string":"item
3_text"}],"input_type":"2,"frame_id":750}
```

14 STK Operation

14.1 Request

POST https://gateway_ip/STKGo

14.2 Request Parameter

Parameter	Type	Description
Required Parameter		
port	Integer	
Optional Parameter		
item	Integer	
param	String	
action	String	“ok”, ”cancle”or ”home”

14.3 Example of STK Operation

Request:

```
curl -k --anyauth -u admin:admin -d '{"port":7,"item":1}' -H
"Content-Type: application/json" -H "Content-Type:
application/json" http://gateway_ip/STKGo
```

```
curl -k --anyauth -u admin:admin -d '{"port":7,"action":"cancle"}'
-H "Content-Type: application/json" -H "Content-Type:
application/json" http://gateway_ip/STKGo
```

15 Query Frame ID of STK

15.1 Request

```
GET https://gateway_ip/GetSTKCurrFrameIndex
```

15.2 Request Parameter

Parameter	Type	Description
Required Parameter		
port	Integer	

15.3 Response Parameter

Parameter	Type	Description
frame_id	Integer	

15.4 Example of Querying Frame ID

Request:

```
https://gateway_ip/GetSTKCurrFrameIndex?port=0
```

Response:

```
{"frame_id":32}
```

16 Get Device Status

16.1 Request

POST https://gateway_ip/api/get_status

16.2 Request Parameter

Parameter	Type	Description
Required Parameter		
performance		

16.3 Response Parameter

Parameter	Type	Description
cpu_used	String	CPU usage

flash_total	String	flash size
flash_used	String	Used flash size
memory_total	String	Memory size
memory_cached	String	Cache size
memory_buffers	String	Buffer size
memory_free	String	Free memory size
memory_used	String	Used memory size

16.4 Example

Request:

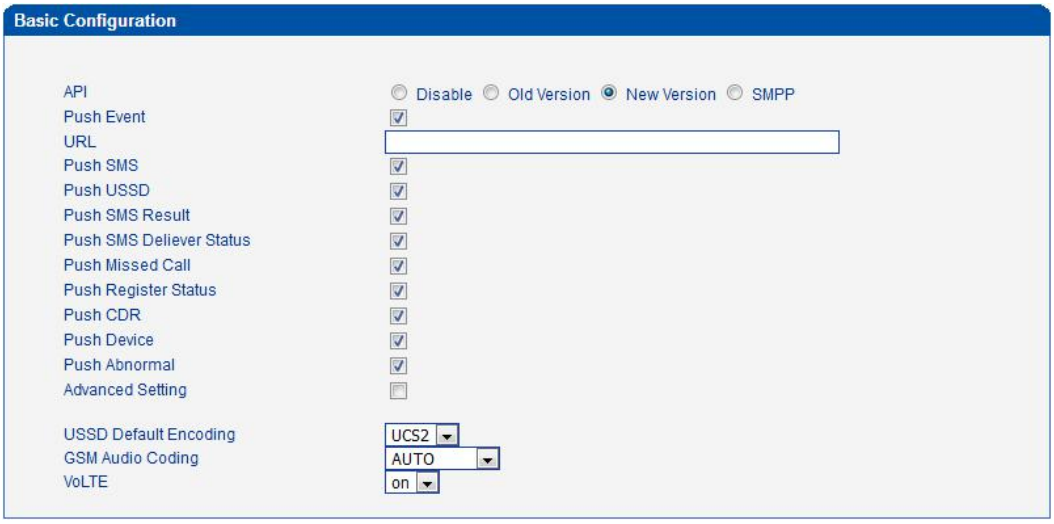
```
curl -k --anyauth -u admin:admin -d '["performance"]' -H "Content-Type: application/json"  https://gateway_ip/api/get_status
```

Response:


```
{"performance":{"cpu_used":"39","flash_total":"27648","flash_used":"17428","memory_total":"109448","memory_cached":"34192","memory_buffers":"0","memory_free":"58928","memory_used":"50520"}}
```

17 Push Event

In the latest version, the gateway can push some events such as SMS, SMS sending results and SMS delivery status to your web server.



Once you selected push SMS, all SMS messages received by the gateway will be pushed to your web server. The context of SMS pushing is very similar to the response of **querying incoming sms**.

```
{"sn":"xxxx-xxxx-xxxx-xxxx","sms":[{"incoming_sms_id":1,"port":
1,"number":"6717","smsc":"+8613800757511","timestamp":"2016-07
-12 15:46:18","text":"test"}]}
```

The result of SMS pushing is similar to the response of **Query Send SMS Result**.

{"sn":"xxxx-xxxx-xxxx-xxxx","sms_result":[{"port":1,"number":"10086","time":"2016-07-12 01:46:02","status":"DELIVERED","count":1,"succ_count":1,"ref_id":215,"imsi":"460004642148063"}]}

SMS delivery status

{"sn":"xxxx-xxxx-xxxx-xxxx","sms_deliver_status":[{"port":1,"number":"10086","time":"2016-07-12 15:46:53","ref_id":215,"status_code":0,"imsi":"460004642148063"}]}

USSD

{"sn":"xxxx-xxxx-xxxx-xxxx","ussd":[{"port":1,"text":"Thank you!"}]}

SIM Register Status

{"sn":"xxxx-xxxx-xxxx-xxxx","register":[{"port":8,"iccid":"89860040191844710023","imsi":"460004642148063","number":"13714637674","status":"up","sequence":2,"slot":2}]}

{"sn":"xxxx-xxxx-xxxx-xxxx","register":[{"port":8,"iccid":"<NULL>","imsi":"<NULL>","number":"13714637674","status":"down","sequence":1,"slot":2}]}

Note: Multi-SIM gateway will push SIM Register Status with "slot".

CDR:

{"sn":"xxxx-xxxx-xxxx-xxxx","cdr":[{"port":2,"start_date":"2015-07-21 16:35:20","answer_date":"2015-07-21 16:35:31","duration":3,"source_number":"1010","destination_number":"6717","direction":"ip->gsm","ip":"172.16.100.136","codec":"G.711U","hangup":"called","gsm_code":16,"bcch":""}]}

Note: Only answered call will be pushed. For All the CDR, please ues "get_cdr"

Device:

{"sn":"xxxx-xxxx-xxxx-xxxx","device":{"port_number":32,"IP":"172.18.55.142","MAC":"F8-A0-3D-48-E5-19","status":"power_off"}}

{"sn":"xxxx-xxxx-xxxx-xxxx","device":{"port_number":32,"IP":"172.18.55.142","MAC":"F8-A0-3D-48-E5-19","status":"power_on"}}

Abnormal:

{"sn":"xxxx-xxxx-xxxx-xxxx","exception_info":{"port":0,"type":"call_fail","action":"reset"}}

Note: Exception Event Handling should be enabled.

18 CALL Forward

18.1 Set Call Forward Request

GET https://gateway_ip/api/set_port_info

18.2 Request Parameter

Parameter	Type	Description
Required Parameter		
port	Integer	
action	String	Should be “CallForward”
param	String	Could be like Unconditional, NoReply, Busy, Not_Reachable, CancelAll
number	String	Phone Number

18.3 The First to Read Call Forward Request

```
GET https://gateway_ip/api/set_port_info
```

18.4 The First Request Parameter

Parameter	Type	Description
Required Parameter		
port	Integer	
action	String	Should be “CheckCallForward”

18.5 The Second To Read Call Forward Request

```
GET https://gateway_ip/api/get_port_info
```

18.6 The Second Request Parameter

Parameter	Type	Description
Required Parameter		
port	Integer	
info_type	String	Should be “CallForward”

18.7 Example

1. Set call forward

```
https://gateway_ip/api/set_port_info?port=8&action=CallForward&param=Unconditional&number=15013828917
```

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx"}
```

2. Send checking command

```
https://gateway_ip/api/set_port_info?port=8&action=CheckCallForward
```

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx"}
```

3. Get result

```
https://gateway_ip/api/get_port_info?port=8&info_type=CallForward
```

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx","info":[{"port":8,"CallForwarding":{"Unconditional":"15013828917"}}]}
```

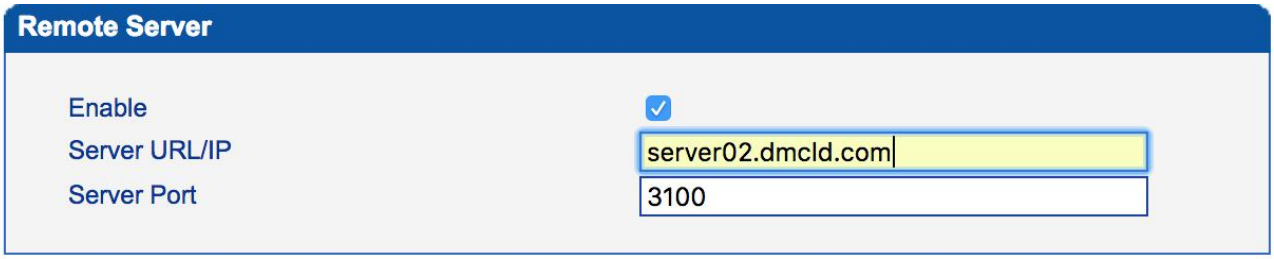
19 NAT

19.1 Application Scenarios

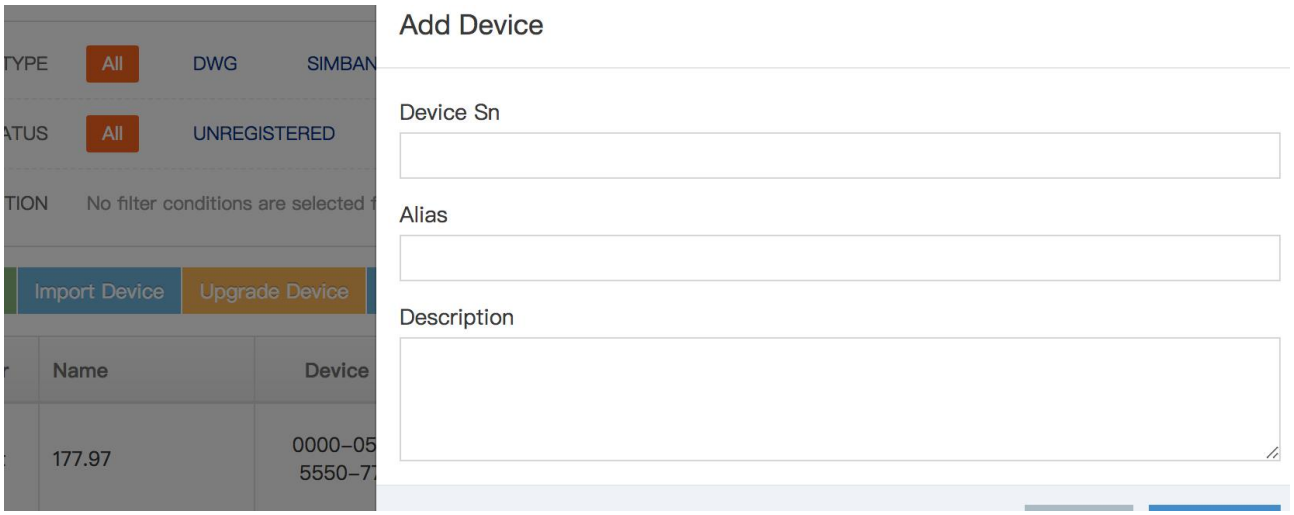
DRP server can be used to communicate with the gateway behind NAT.

19.2 Configuration section

(1) Enable "Tools -> Remote Connection Configuration" in the gateway.



(2) Add gateway in DRP server.



19.3 API Changes

Based on the IP and port (in the configuration file) that the DRP server is configured, the API client sends the request to that address;

(1) Connect to the DRP server and send the password of the account for authentication

Request: https://drpserver_ip:port/doLogin?Username=admin&password=admin

After the DRP server authentication passes, it returns 200 OK, indicating that the server accepts the subsequent requests from the API client;

(2) After passing the authentication, send the device serial number to the DRP server:

Request: https://drpserver_ip:port/remoteWeb?Product_sns= 0123-4567-890A-BCDE

After the DRP server matches the corresponding device, it replies 200 OK, indicating that subsequent requests of the API client are forwarded to the device specified by the serial number;

(3) Follow-up according to the specific function to send the corresponding request, see the API specific usage (the model api client requests are sent to the DRP server);

20 FAQ

20.1 How can you specify a port for sending SMS?

When you need to specify a port for sending SMS, you can set port parameter in the SMS sending request. If you don't need to send SMS through a specific port, port parameter is not necessary.

20.2 How to match an SMS sending result with an SMS sending request?

It is advised to use user_id in the SMS sending request. You can use different user id when SMS is sent to a different telephone number, and then SMS sending result will also carry this user id. Each user id needs to be unique, and generally it is changed in an increasing order.

20.3 How fast can a SMS be sent?

It takes about 5 to 8 seconds to send a SMS message on a GSM gateway, and about 2 to 4 seconds on a LTE gateway. The time depends on the GSM network quality. When the GSM network is in a bad condition, it may take about 1 minute.

20.4 How can you get SMS delivery status?

Generally, it is not necessary to query delivery status, as delivery status is included in SMS sending Result. But if you want to control the SMS delivery, please see the following:

SMS delivery status is received by gateway in unspecific time. It is connected with an SMS sending result by a ref_id. And it is better to query delivery status after SENT_OK is returned. For a long SMS message which will be divided into 3 smaller SMS segments, if the first ref_id in SMS sending result is 0 and those of other two SMS segments are 1 and 2 respectively, at least 3 delivery statuses will be received, carrying ref_id 0, 1 and 2 respectively.

20.5 How can you disable SMS delivery status?

Set the value of ‘request_status_report’parameter into ‘false’ in the SMS sending request.

20.6 How to send HTTP(s) request in the program?

C/C++ and PHP can use libcurl for sending HTTP(s) request. The parameter in libcurl is almost the same in curl utility.

curl	libcurl
-k	CURLOPT_SSL_VERIFYPEER, CURLOPT_SSL_VERIFYHOST,
-u	CURLOPT_USERPWD
--anyauth	CURLOPT_HTTPAUTH
-H	CURLOPT_HTTPHEADER
-d	CURLOPT_POSTFIELDSCURLOPT_POSTFIELDS

PHP example:

```
$curl = curl_init($url);

curl_setopt($curl, CURLOPT_HTTPAUTH, CURLAUTH_ANY ) ;

curl_setopt($curl, CURLOPT_HEADER, true);

curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);

curl_setopt($curl, CURLOPT_HTTPHEADER, array("Content-type:
application/json;"));

curl_setopt($curl, CURLOPT_USERPWD, "admin:admin");

curl_setopt($curl, CURLOPT_POST, true);

curl_setopt($curl, CURLOPT_POSTFIELDS, $content);

curl_setopt($curl, CURLOPT_FOLLOWLOCATION, 1);

curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 0);

curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, 0);
```

C#,JAVA,PYTHON and other programming language have their own lib for sending http request, please check the programming manual for help.