SMG Wireless Gateway API

Synway Information Engineering Co., Ltd

www.synway.net

Content

Chapter 1 Overview

These API interfaces are implemented based on HTTP. They can be divided by function into two categories: one for receiving the request and the other for feeding back the request. Authentication is required at each submit. Use the account and the password specified for API on the gateway. Besides, the parameter header is case insensitive and all punctuations involved in must be English.

1.1 For Receiving Request

Post http://GateWayIP/API/TaskHandle

Parameter Format:

{“event”:“evnetinfo”,”task”:”task info”}

1.2 For Feeding Back Request

Post http://GateWayIP/API/QueryInfo

Parameter Format:

{“result”:“result info”,“content”:“content info”}

1.3 For Request Authentication

Authentication is required for all requests. Use the account and the password specified for API on the gateway.

1.3.1 Setting Request Authentication

Click ***Advanced Settings->System Parameters->API Parameters->API Enabled->Yes,*** and fill in ***Username for API Call*** and ***Password for API Call***.

1.3.2 Reference Case

PHP Case

```
1. /**
2.  *Need to install curl extension
3.  */
```

```
class CHttpClient
{
    static public function PostWithJson($strUrl, $strPost, &$strResponse)
    {
        $szUrNmPswd = $GLOBALS['g_username'].':'.$GLOBALS['g_password'];
        $headerArray =array("Content-type:application/json;charset='utf-8'","Accept:application/json");
        $curl = curl_init();
        curl_setopt($curl, CURLOPT_ENCODING, "");
        curl_setopt($curl, CURLOPT_URL, $strUrl);
        //Add user authentication information (username and password) here
        curl_setopt($curl, CURLOPT_USERPWD, $szUrNmPswd);
        curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, FALSE);
        curl_setopt($curl, CURLOPT_SSL_VERIFYHOST,FALSE);
        curl_setopt($curl, CURLOPT_POST, 1);
        //strPost saves POST request behavior, such as: {"event":"getportinfo"}
        curl_setopt($curl, CURLOPT_POSTFIELDS, $strPost);
        curl_setopt($curl,CURLOPT_HTTPHEADER,$headerArray);
        curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
        $response = curl_exec($curl);
        //Get the POST return data, find the ok keyword to determine the request verification success
        $strResponse = curl_multi_getcontent($curl) ;
        // POST return code, 200 ok success, view http for other error codes
        $code = curl_getinfo($curl, CURLINFO_HTTP_CODE);
        curl_close($curl);
        if($code == 200)
            return $response;
        else
            return array('code'=>$code, 'response'=>$response);
    }
}
```

C++ Case

```
/**
 *Using the libcurl network protocol library, you can download the source code from https://github/curl/curl
 *Use 7.59.0 branch here, static compilation with vs2013
 */
int CHttpClient::PostWithJson(const std::string & strUrl, const std::string & strJson, std::string & strResponse)
{
    CURLcode res;
    wchar_t szUrNmPswd[100] = { 0 };
```

```
CURL* curl = curl_easy_init();
std::string stUrNmPswd = "";
swprintf(szUrNmPswd, g_username + _T(":") + g_password);
stUrNmPswd = WStringToUTF8(szUrNmPswd);
if (NULL == curl)
    return CURLE_FAILED_INIT;
curl_easy_setopt(curl, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
//Add user authentication information (username and password) here
curl_easy_setopt(curl, CURLOPT_USERPWD, stUrNmPswd.c_str());
curl_easy_setopt(curl, CURLOPT_URL, strUrl.c_str());
curl_easy_setopt(curl, CURLOPT_POST, 1);
 curl_slist *plist = curl_slist_append(NULL, "Content-Type:application/json;charset=UTF-8");
curl_easy_setopt(curl, CURLOPT_HTTPHEADER, plist);
//Set the request action, such as: {"event":"getportinfo"}
curl_easy_setopt(curl, CURLOPT_POSTFIELDS, strJson.c_str());
curl_easy_setopt(curl, CURLOPT_READFUNCTION, NULL);
//Set the callback function to save the received data
curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, OnWriteData);
//The pointer source of the callback function lpVoid, get the saved data, find the ok keyword to determine the request verification success
curl_easy_setopt(curl, CURLOPT_WRITEDATA, (void *)&strResponse);
curl_easy_setopt(curl, CURLOPT_NOSIGNAL, 1);
curl_easy_setopt(curl, CURLOPT_CONNECTTIMEOUT, 3);
curl_easy_setopt(curl, CURLOPT_TIMEOUT, 50);
curl_easy_setopt(curl, CURLOPT_SSL_VERIFYPEER, 0L);
curl_easy_setopt(curl, CURLOPT_SSL_VERIFYHOST, 0L);
curl_easy_setopt(curl, CURLOPT_VERBOSE, 1L);
//Returns 0 for Ok, non-0 for error occurrence
res = curl_easy_perform(curl);
curl_easy_cleanup(curl);
return res;
}
```

Java Case

```
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import org.apache.commons.codec.binary.Base64;
public String sendMessageHttp(String mobiles, String smscontent) {
    int respCode=0;
    StringBuilder result = new StringBuilder();
    HttpURLConnection connection = null;
```

```
11.    try {
12.        URL postUrl = new URL(SEND_SMS_URL);
13.        //Open connection
14.        connection = (HttpURLConnection) postUrl.openConnection();
15.        connection.setDoOutput(true);
16.        connection.setDoInput(true);
17.        connection.setRequestMethod("POST");
18.        connection.setUseCaches(false);
19.        connection.setInstanceFollowRedirects(true);
20.         connection.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
21.        // Add an authentication field
22.        connection.setRequestProperty("Authorization", "Basic " +Base64.encodeBase64String(n
    ew String(SEND_SMS_USER+":"+SEND_SMS_PASSWD).getBytes("utf-8")));
23.        connection.connect();
24.        DataOutputStream out = new DataOutputStream(connection.getOutputStream());
25.        StringBuffer sendMessage = new StringBuffer();
26.        //POST request
27.        sendMessage.append("{\"event\":\"txsms\",\"num\":\""+mobiles+"\",\"port\":\"-1\",\"encoding
    \":\"8\",\"smsinfo\":\""+smscontent+"\"}");
28.        out.write(sendMessage.toString().getBytes("UTF-8"));
29.        out.flush();
30.        out.close();
31.        respCode=connection.getResponseCode();
32.        //Set the code, otherwise irrecognizable in Chinese
33.        BufferedReader reader = new BufferedReader(new InputStreamReader(connection.getIn
    putStream(), "UTF-8"));
34.        String lines;
35.        while ((lines = reader.readLine()) != null) {
36.          result.append(lines);
37.        }
38.    } catch (Exception e) {
39.        e.printStackTrace();
40.    } finally {
41.        if (connection != null) {
42.            connection.disconnect();
43.        }
44.    }
45.    return result.toString();
46. }
```

Chapter 2 API Description

2.1 Acquiring Device Information

2.1.1 Obtaining Port Status

1) Request

POST http://GateWayIP/API/QueryInfo, or

POST http://GateWayIP/API/QueryPortInfo (use this link if querying frequently)

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of 'getportinfo'.

Example:

{"event":"getportinfo"}

Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	string	Message Type: 'ok' means successful, 'error' means return failed.
content	string	Message Content: If 'OK', it gives the information of the port; if 'error', it tells the error reason. Port Status: 0—Idle, 1—Off-hook, 2—Ringing, 3—Talking, 4,5,6—Dialing, 7—Pending, 9—Ringback, 10—Interior, 11--Unavailable

Example:

The request returns successful:

{"result":"ok","content":"total:8;portstate:0,0,0,0,0,0,0,0"}

Description:

Total: Quantity of ports.

Portstate: Port status. Every two ports are separated by ','.

The request returns failed:

{"result":"error","content":"error reason"}

Description:

'error reason' describes the exact failure.

2.1.2 Obtaining Port SIM Status

1) Request

POST http://GateWayIP/API/QueryPortInfo

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of 'getportinfoex'.

Example:

{"event":"getportinfoex"}

Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	string	Message Type: 'ok' means successful, 'error' means return failed.
content	string	Message Content: Information about the port with the request returning successful. 'error' tells the error reason. Port Status: 0—In use, 16—Available, 17--Unavailable

Example:

The request returns successful:

{"result":"ok","content":"total:16;portstate:0,17,17,17:17,17,17,17;"}

Description:

Total: Quantity of ports.

Portstate: Status of the SIM card on the port. Every two ports are separated by ':'.

The request returns failed:

{"result":"error","content":"error reason"}

Description:

'error reason' describes the exact failure.

2.1.3 Obtaining Port BS Connection Status

1) Request

POST http://GateWayIP/API/QueryInfo, or

POST http://GateWayIP/API/QueryPortConnectState (use this link if querying frequently)

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of ‘getportconnectstate’.

Example:

{“event”:“getportconnectstate”}

Description: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	string	Message Type: ‘ok’ means successful, ‘error’ means return failed.
content	string	Message Content: If ‘OK’, it gives the information about the port BS connection status; if ‘error’, it tells the error reason. BS Connection Status: 0—Unconnected, 1—Connected, 2—Connecting, 3—Rejected, 4—Unknown, 5—Roaming

Example:

The request returns successful:

{"result":"ok","content":"total:8;ConnectState:1,1,1,1,0,0,0,0."}

Description:

Total: Quantity of ports.

Connectstate: BS connection status for each port.

The request returns failed:

{“result”:“error”,“content”:“error reason”}

Description:

‘error reason’ describes the exact failure.

2.1.4 Obtaining Wireless Parameters Information

1) Request

POST http://GateWayIP/API/QueryInfo, or

POST http://GateWayIP/API/QueryWireLessInfo (use this link if querying frequently)

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of ‘getwirelessinfo’.
type	string	The type of query, with the value of ”porttype”, ”ICCID”, ”IMEI”, ”IMSI” and ” PhoneNo”
port	string	The port to be queried, separated by ‘,’, optional. If this parameter isn’t filled in, it means all ports shall be queried.

Example 1:

{"event":"getwirelessinfo","type":"porttype"}
Description:
Query the type for all ports.

Example 2:
{"event":"getwirelessinfo","type":"IMEI",”port”:”1,2,3”}
Description:
Query the IMEI for Port 1, Port 2 and Port 3.
Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	string	Message Type: ‘ok’ means successful, ‘error’ means return failed.
content	string	Message Content: If ‘OK’, it gives the information about the wireless parameters; if ‘error’, it tells the error reason.

Example 1:
The request returns successful:
{"result":"ok","content":"PortType:WCDMA"}
Description:
PortType: The port type is WCDMA

Example 2:
The request returns successful:
{"result":"ok","content":"2:89860058111551226470.3:89860058111551226471."}
Description:
The ICCID of port 2 is 89860058111551226470
The ICCID of port 3 is 89860058111551226471

The request returns failed:
{“result”:“error”,“content”:“error reason”}
Description:
‘error reason’ describes the exact failure.

2.1.5 Obtaining Detailed Port Information

1) Request

POST http://GateWayIP/API/QueryInfo

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of 'getDetailPortInfo'.

Example 1:

{"event":"getDetailPortInfo"}

Description:

Query the detailed information of all ports.

Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	string	Message Type: 'ok' means successful, 'error' means return failed.
content	string	Message Content: If 'OK', it gives the information of the port; if 'error', it tells the error reason. Port Status: 0—Idle, 1—Off-hook, 2—Ringing, 3—Talking, 4,5,6—Dialing, 7—Pending, 9—Ringback, 10—Interior, 11--Unavailable

Example 1:

The request returns successful:

{"result":"ok","totalPort":"32","portState":"11,11,11,11,11,11,11,0,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11","portOp":"-1,-1,-1,-1,-1,-1,-1,CHN-CT,,,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1","portType":"LTE-C","portSig":"-1,-1,-1,-1,-1,-1,-1,100,0,0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1"}

Description:

totalPort: There are 32 ports in total on the gateway.

portState: The status of all ports on the gateway.

portOp: The operator information for all ports on the gateway.

portType: The port type is LTE-C.

portSig: The signal value of the port on the gateway.

The request returns failed:

{“result”:“error”,“content”:“error reason”}

Description:

'error reason' describes the exact failure.

2.1.6 Obtaining Port Call Time

1) Request

POST http://GateWayIP/API/QueryInfo

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of 'querycalltime'.
port	string	Ports to be queried, separated by ','. It is essential.

Example 1:

{"event":"querycalltime","port":"1,8"}

Description:

Query the call time of Port 1 and Port 8.

Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	string	Message Type: 'ok' means successful, 'error' means return failed.
content	string	Message Content: If 'OK', it gives the information about the wireless parameters; if 'error', it tells the error reason.

Example 1:

The request returns successful:

{"result":"ok","content":[{"port":"1","limitTime":"-1","channels":"-1:0;-1:0;-1:0;-1:0"},{"port":"8","limitTime":"100","channels":"-1:0;-1:0;-1:0;8618906814358:36"}]}

Description:

{"port":"1","limitTime":"-1","channels":"-1:0;-1:0;-1:0;-1:0"}

limitTime=-1 means the Maximum Call Time Limit is not enabled for Port 1, each call time is 0.

{"port":"8","limitTime":"100","channels":"-1:0;-1:0;-1:0;8618906814358:36"}

limitTime=100 means the maximum call time for Port 8 is 100 minutes; "-1:0;-1:0;-1:0;8618906814358:36" means all A, B, C slots on Port 8 don't have numbers, Slot D has a phone number of 8618906814358 and its call time is 36 minutes.

The request returns failed:

{"result":"error","content":"error reason"}

Description:

'error reason' describes the exact failure.

2.2 SMS

2.2.1 Sending SMS

1) Request

POST http://GateWayIP/API/TaskHandle, or

POST http://GateWayIP/API/SendSMS (use this link if sending SMS frequently)

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of 'txsms'.
userid	string	User ID, used for returning result, optional.
num	string	Destination number, up to 50 numbers allowed, separated by ','.
port	string	Port used to send SMS, with the default value of -1. If there are more than one port available, which are separated by ',', the gateway will automatically select idle ones.
encoding	string	SMS encoding format, 0: bit7, 8: UCS-2.
smsinfo	string	SMS content. The length of bit7 cannot exceed 600, and that of USC-2 cannot exceed 300.
smsc	string	SMS center number, optional

Example:

{"event":"txsms","userid":"0","num":"10086,10087,10088","port":"1,2,3","encoding":"0","smsinfo":"hello world!"}

Description:

User ID 0 wants to use Port 1, Port 2 and Port 3 to send 'hello world!' respectively to three numbers 10086, 10087 and 10088. If Port 1, Port 3 are idle and Port 2 is busy, the gateway will use Port 1 to send the message to 10086, use Port 3 to send it to 10087, and then wait until one of these three ports becomes idle and use it to send the message to 10087.Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	String	Message Type: 'ok' means successful, 'error' means return failed.
content	String	Message Content: If successful, what returned is the task ID; if failed, what returned is the failure reason.

Example:

The request returns successful:

{"result":"ok","content":"taskid:0"}

Description: taskid represents the unique flag for this task. Use this task flag to query the sending result.

The request returns failed:
{“result”:“error”,“content”:“error reason”}
Description: ‘error reason’ describes the exact failure.

Error Type	Description
sms encoding format does not support	SMS encoding format unsupported
txsms fifo is full	SMS sending queue is full
sms data too long	SMS content too long
no available port	No ports available
no phonenumber	No sending number
no quantity available	Available SMS on the port used up

2.2.2 Inquiring SMS Sending Result

1) Request
POST http://GateWayIP/API/QueryInfo, or
POST http://GateWayIP/API/QueryTxSMS (use this link if querying frequently)

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of ‘querytxsms’.
userid	string	User ID number, used to return results, optional.
taskid	string	The task ID to query. More than one task can be query at the same time, and they are separated by ‘,’.
portid	string	The port ID use to query the SMS sending result, separated by ‘,’. Either portid or taskid can exist at a time.
num	string	The number to query, separated by ‘,’, optional.
port	String	The port to query, separated by ‘,’, optional.

Example 1: {“event”:“querytxsms”,“taskid”:“1,2”}
Description: Query the results of Task 1 and Task 2.

Example 2: {“event”:“querytxsms”,“taskid”:“0”,“num”:“10086”,“port”:“1”}
Description: Query the result of Task 0 sent by Port 1 and with the destination number of 10086.

Example 3: {“event”:“querytxsms”,“portid”:“1,2”}
Description: Query the result of sending SMS by Port 12.
Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	string	Message Type: 'ok' means successful, 'error' means return failed.
content	string	Message Content: If successful, what returned is the query result. See below for details. If failed, what returned is the failure reason. Sending Status: 0-None, 1-Wait, 2-Sending, 3-Successful, 4-Failed 5-Timeout Failed: Frequently sending a short message to a number will cause failure. Timeout: Wrong port settings or always no available port being searched will cause timeout.

Example 1:

The request returns successful:

{"result":"ok","content":"[taskid:0:2:3;1:10086:2:0;2:10087:3:0]"}

Description: [taskid:0:2:3;1:10086:2:0;2:10087:3:0]

taskid:0:2:3 Task ID is 0, sent number in this task is 2, userid is 3

1:10086:2:0 Send from Port 1 to 10086, status value is 2 (sending), need receipt or not (0)

2:10087:3:0 Send from Port 2 to 10087, status value is 3 (successful), need receipt or not (0)

Example 2:

{"result":"ok","content":"[taskid:0:1:3;1:10086:3:1]"}

Description: [taskid:0:1:3;1:10086:3:1]

taskid:0:1:3 Task ID is 0, sent number in this task is 1, userid is 3

1:10086:3:1 Send from Port 1 to 10086, status value is 3 (successful), need receipt or not (1)

The request returns failed: {"result":"error","content":"error reason"}

Description: 'error reason' describes the exact failure.

Example 3:

{"result":"OK","content":"port:1:[taskid:0:1:10086:2:1;taskid:1:1:10087:3:1]"}

Description: port:1:[taskid:0:1:10086:2:1;taskid:1:1:10087:3:1]

There are two tasks for querying the SMS sending result of Port 1.

taskid:0:1:10086:2:1 Task ID is 0, send from Port 1 to 10086, status value is 2 (sending), need receipt

taskid:1:1:10087:3:1 Task ID is 1, send from Port 1 to 10087, status value is 3 (successful), need receipt

2.2.2.1 New API Request for Inquiring SMS Sending Result

1) Request

POST http://GateWayIP/API/QueryInfo

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of 'newquerytxsms'.
userid	string	User ID number, used to return results, optional.
taskid	string	The task ID to query, essential. More than one task can be queried at the same time, and they are separated by ','.
num	string	The number to query, separated by ',', optional. Only one is supported at present.
port	String	The port to query, separated by ',', optional. Only one is supported at present.

Example 1: {"event":"newquerytxsms","taskid":"1,2"}

Description: Query the results of Task 1 and Task 2.

Example 2: {"event":"newquerytxsms","taskid":"0","num":"10086","port":"1"}

Description: Query the result of a task whose ID is 0, sending port is 1 and destination number is 10086.

Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	string	Message Type: 'ok' means successful, 'error' means return failed.
content	string	Message Content: If successful, what returned is the query result. See below for details. If failed, what returned is the failure reason. Sending Status: 0-None, 1-Wait, 2-Sending, 3-Successful, 4-Failed 5-Timeout Failed: Frequently sending a short message to a number will cause failure. Timeout: Wrong port settings or always no available port being searched will cause timeout.

Example 1:

The request returns successful:

{"result":"ok","content":"[taskid:0:2:-1;1:10086:2;2:10087:3],
[taskid:1:1;-1;1:10088:2]"}

Description: [taskid:0:2:-1;1:10086:2;2:10087:3]

taskid:0:2:-1 Task ID is 0, sent number in this task is 2, userid is -1
1:10086:2 Send from Port 1 to 10086, status value is 2 (sending)
2:10087:3 Send from Port 2 to 10087, status value is 3 (successful)

Example 2:
{“result”:“oK”,“content”:“[taskid:0:1:-1;1:10086:3]”}
Description: [taskid:0:1:-1;1:10086:3]
taskid:0:1:-1 Task ID is 0, sent number in this task is 1, userid is -1
1:10086:3 Send from Port 1 to 10086, status value is 3 (successful)
The request returns failed: {“result”:“error”,“content”:“error reason”}
Description: ‘error reason’ describes the exact failure.

Error Type	Description
portid and taskid cannot exist at the same time	portid and taskid cannot exist at the same time
portid or taskid is NULL	portid and taskid are empty
error(port error)	The port number is greater than the maximum number of channels or less than 1
Task does not exist	No corresponding taskid found

2.2.3 Obtaining Inbox Information

1) Request
POST http://GateWayIP/API/QueryInfo, or
POST http://GateWayIP/API/QueryRxSMS (use this link if querying frequently)

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of ‘queryrxsms’.
begintime	string	Time stamp, with the smallest unit of second, optional. Get the SMS after the time stamp.
endtime	string	Time stamp, with the smallest unit of second, optional. Get the SMS before the time stamp.
num	string	SMS source number, separated by ‘,’, optional.
port	string	Obtain SMS from the designated port, separated by ‘,’, optional.
delete	string	Delete a short message from the inbox after querying it, optional

Example 1:
{“event”:“queryrxsms”,“begintime”:“20150901180000”,
“endtime”:“20150915180000”,“num”:“10086”,”port”:”1,2”}
Description:

Get the SMS sent from 10086 to Port 1 and 2 between 2015-09-01 18:00:00 and 2015-09-15 18:00:00

Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

Example 2:

{"event":"queryrxsms","begintime":"20191010162700","endtime":"20191010162745","delete":"1"}

Description:

Query the SMS between 2019-10-10 16:27:00 and 2019-10-10 16:27:45 and delete all the SMS found; if the value of delete is 0, only query but not delete the found SMS.

Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	string	Message Type: 'ok' means successful, 'error' means return failed.
content	string	Message Content: If successful, what returned is the query result; if failed, what returned is the failure reason.

Example:

The request returns successful:

{"result":"ok","content":"total:0"}

Description: No SMS accords with the condition.

{"result":"ok",
"content":"total:2;
20150911111111:1(A)(152555588888):10086: Hello world!|E;
20150911111112:2(B)(-1):10086:20150911111112|:This is a test message|E"}

Description: Find two SMS which accords with the condition.

Total: Total number of SMS.

SMS information structure: Time: Port(Slot number)(SIM card number): Source Number: Content |E

The request returns failed:

{"result":"error","content":"error reason"}

Description:

'error reason' describes the exact failure.

Note:

Slot numbers (A) (B) (C) (D) respectively correspond to four SIM card slots; (-1) means this port can accommodate only one SIM card.

SIM card number is the phone number of this SIM card (15255558888); (-1) means the number is null.

2.2.3.1 Obtaining Port Received Information

1) Request

POST http://GateWayIP/API/QueryInfo

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of 'newqueryrxsms'.
port	string	Obtain SMS from the designated port, separated by ',', essential.

Example:

{"event":"newqueryrxsms","port":"3,4"}

Description:

Acquire all SMS received by Port 3 and Port 4.

Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	string	Message Type: 'ok' means successful, 'error' means return failed.
content	string	Message Content: If successful, what returned is the query result. If failed, what returned is the failure reason.

Example:

The request returns successful:

{"result":"ok","content":"total:0"}

Description: No matched SMS.

{"result":"ok",

"content":"total:4;

2018-06-26 12:40:22:3(A) (13588280649):1065813919: SMS Content 1|E;2018-06-26 11:29:55:4(B) (13588280649):10086: SMS Content 2|E;2018-07-03 14:06:14:4(C) (+8618668137917):10016: SMS Content 3|E;2018-07-04 09:34:01:4(C) (+8618668137917):10010: SMS Content 4|E"}

Description: Query four pieces of SMS that meet the match.

Total: Total number of SMS

SMS information structure: Time: Port (Slot Number)(SIM Card Number): source number: SMS content |E

The request returns failed:

{"result":"error","content":"error reason"}

Description: 'error reason' describes the exact failure.

Note:

Slot numbers (A) (B) (C) (D) respectively correspond to four SIM card slots; (-1) means this port can accommodate only one SIM card.
SIM card number is the phone number of this SIM card (15255558888); (-1) means the number is null.

This function applies to frequent SMS services which require the check of SMS received by the corresponding port at any time. In most cases, we recommend you use the above 2.2.3 'queryrxsms'.

2.2.4 Obtaining Outbox Information

1) Request
POST http://GateWayIP/API/QueryInfo, or
POST http://GateWayIP/API/QuerySxSMS (use this link if querying frequently)

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of 'querysxsms'.
begintime	string	Time stamp, with the smallest unit of second, optional. Get the SMS after the time stamp.
endtime	string	Time stamp, with the smallest unit of second, optional. Get the SMS before the time stamp.
num	string	SMS destination number, separated by ',', optional.
port	string	Obtain the SMS from the designated port, separated by ',', optional.

Example:
{"event":"querysxsms","begintime":"20150901180000",
"endtime":"20150915180000","num":"10086","port":"1,2"}
Description:
What to get are the SMS from Port 1 and Port 2 to 10086 during the time 2015-09-01 18:00:00 to 2015-09-15 18:00:00.
Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	string	Message Type: 'ok' means successful, 'error' means return failed.
content	string	Message Content: If successful, what returned is the query result; if failed, what returned is the failure reason.

Example:
The request returns successful:

{“result”:“ok”,“content”:“total:0”}
Description: No SMS accords with the condition.
{“result”:“ok”,
“content”:“total:2;
20150911111111:1(-1)(15255558888):10086:2:Hello world!|E;
20150911111112:2(C)(15255558888):10086:0:This is a test message|E”}
Description: Find two SMS which accords with the condition.
Total: Total number of SMS.
SMS information structure:Time:Port(Slot number)(SIM card number):Source Number: Source:Content|E

The request returns failed:
{“result”:“error”,“content”:“error reason”}
Description:
‘error reason’ describes the exact failure.
Source:
0—-Web Page
1—-Email
2—-API
3—-System
4—-SIP
5—-SMPP

2.2.5 Deleting SMS from Inbox&Outbox

1) Request
POST http://GateWayIP/API/TaskHandle, or
POST http://GateWayIP/API/DeleteSMS (use this link if deleting SMS frequently)

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of ‘deletesms’.
port	string	Delete SMS from the designated port, separated by ‘,’, optional.
phonenum	string	Delete SMS from the designated phone number, separated by ‘,’, optional.
begintime	string	Time stamp, with the smallest unit of second, optional. Delete the SMS after the time stamp.
endtime	string	Time stamp, with the smallest unit of second, optional. Delete the SMS before the time stamp.
smsbox	string	r: delete Inbox, s: delete Outbox, r, s: delete all.
clear	string	1: delete the SMS accords with above condition, 2: clear all SMS.

Example 1:

{“event”:“deletesms”,”port”:”1,2”,”phonenum”:”10086”,“begintime”:“20150901180000”,“endtime”:“20150915180000”, “smsbox”:“r”,”clear”:”1” }

Description:

Delete the SMS in the Inbox from 10086 to Port 1 and Port 2 during the time 2015-09-01 18:00:00 to 2015-09-15 18:00:00.

Example 2:

{“event”:“deletesms”,”port”:”1.A,2.B”,”phonenum”:”10086”,“begintime”:“20150901180000”,“endtime”:“20150915180000”,“smsbox”:“r”,”clear”:”1” }

Description:

Delete the SMS received from 10086 to Slot A, Port 1 and Slot B, Port 2 during the time between 2015-09-01 18:00:00 and 2015-09-15 18:00:00.

Example 3:

{“event”:“deletesms”,”port”:”1,2”,”phonenum”:”10086”,“begintime”:“20150901180000”,“endtime”:“20150915180000”, “smsbox”:“s”,”clear”:”1” }

Description:

Delete the SMS in the Outbox from Port 1 and Port 2 to 10086 during the time 2015-09-01 18:00:00 to 2015-09-15 18:00:00.

Example 4:

{“event”:“deletesms”,”port”:”1,2”,”phonenum”:”10086”,“begintime”:“20150901180000”,“endtime”:“20150915180000”, “smsbox”:“r,s”,”clear”:”1” }

Description:

Delete the SMS in the Inbox and Outbox which are receive/send between Port 1, Port 2 and 10086 during the time 2015-09-01 18:00:00 to 2015-09-15 18:00:00.

Example 5:

{“event”:“deletesms”,”port”:”1,2”,”phonenum”:”10086”,“begintime”:“20150901180000”,“endtime”:“20150915180000”, “smsbox”:“r”,”clear”:”2” }

Description:

Delete all SMS in the Inbox.

Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	string	Message Type: ‘ok’ means successful, ‘error’ means return failed.
content	string	Message Content: If successful, what returned is the query result; if failed, what returned is the failure reason.

Example:

The request returns successful:
{“result”:“ok”,“content”:“total:0”}
Description: No SMS accords with the condition.
{“result”:“ok”,“content”:“total:2;
Description: Delete two pieces of SMS which accords with the condition.
Total: Total number of deleted SMS.

The request returns failed:
{“result”:“error”,“content”:“error reason”}
Description:
‘error reason’ describes the exact failure.
About port 1.A : Delete the SMS on Slot A , Port 1;
1.B: Delete the SMS on Slot B, Port 1;
1.C: Delete the SMS on Slot C, Port 1;
1.D: Delete the SMS on Slot D, Port 1;
1: Delete the SMS on all the slots in Port 1;

2.2.6 Querying SMS Receipt

1) Request
POST http://GateWayIP/API/QueryInfo, or
POST http://GateWayIP/API/QueryTxSMSRpt (use this link if querying frequently)

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of ‘querytxsmsrpt’.
port	string	Query the SMS receipt on the designated port, one port limited
num	string	Send the SMS to the designated number, one number limited.
rptnum	string	The number of short messages to be queried for receipts, optional.

Example 1:
{“event”:“querytxsmsrpt”,”port”:”1”,”num”:”10086”}
Description:
Query the receipt for the SMS sent from Port 1 to 10086.
Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	string	Message Type: ‘ok’ means successful, ‘error’ means return failed.
content	string	Message Content:

		If successful, what returned is the query result; if failed, what returned is the failure reason.

Example:

The request returns successful:

{“result”:“ok”,“content”:“OK”}

Description:

The SMS sent from Port 1 to 10086 has been received successfully by 10086.

The request returns failed:

{“result”:“error”,“content”:“error reason”}

Description:

‘error reason’ describes the exact failure.

2.2.7 Querying SMS Sending Results for All taskid

1) Request

POST http://GateWayIP/API/QueryInfo

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of ‘querytxallsms’.

Example 1:

{“event”:“querytxallsms”}

Description:

Query the SMS sending results for all taskid.

Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	string	Message Type: ‘ok’ means successful, ‘error’ means return failed.
smstotal	string	Total number of queried SMS
SMS0/1/2:..	string	Detailed information of the SMS

Example:

The request returns successful:

{“result”:“ok”,"smstotal":"3","SMS0":"0:20181229110042:8:10001:3","SMS1":"1:20181229110054:9:10001:3","SMS2":"2:20181229110107:10:10000:3"}

Description:

The gateway sends out 3 pieces of short messages in total.

"SMS0":"0:20181229110042:8:10001:3" indicates the first short message, whose taskid is

0, sent out at 11:0'42" on December 29th in 2018 from Port 8, with the target number of 10001. It is sent successfully.

The request returns failed:
{"result":"error","content":"error reason"}
Description:
'error reason' describes the exact failure.

Error Type	Description
clear error	Neither delete according to conditions nor empty the entire file
smsbox error	Not one of the operations to delete inbox, delete outbox and delete all

2.3 USSD

2.3.1 Sending USSD

1) Request
POST http://GateWayIP/API/TaskHandle, or
POST http://GateWayIP/API/SendUSSD (use this link if sending USSD frequently)

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of 'txussd'.
port	string	The port used to send USSD, separated by ','.
ussd	string	USSD content, no longer than 255.

Example:
{"event":"txussd","port":"1,2,3","ussd":"*121#"}
Description:
Use Ports 1, 2, 3 to send USSD content '*121#'.
Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	string	Message Type: 'ok' means successful, 'error' means return failed.
content	string	Message Content: If successful, what returned is the query result; if failed, what returned is the failure reason.

Example:
The request returns successful:

{“result”:“ok”,“content”:“1:0;2:1;3:1”}
Description:
Port 1 sends unsuccessfully while Ports 2, 3 send successfully.

The request returns failed:
{“result”:“error”,“content”:“error reason”}
Description:
‘error reason’ describes the exact failure.

2.3.2 Obtaining USSD Sending Result

1) Request
POST http://GateWayIP/API/QueryInfo, or
POST http://GateWayIP/API/QueryRxUSSD (use this link if querying frequently)

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of ‘rxussd’.
port	string	The port used to obtain USSD result, separated by ‘,’.
begintime	string	Time stamp, with the smallest unit of second, optional. Get the SMS after the time stamp, optional.
endtime	string	Time stamp, with the smallest unit of second, optional. Get the SMS before the time stamp, optional.

Example:
{“event”:“rxussd”,“port”:“1,2”,”begintime”:”20160727152000”,”endtime”:”20160727162000”}
Description:
Get the USSD of Port 1 and Port 2 during the time 2016-07-27 15:20:00 to 2016-07-27 16:20:00.
Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	string	Message Type: ‘ok’ means successful, ‘error’ means return failed.
content	string	Message Content: If successful, what returned is the USSD result; if failed, what returned is the failure reason.

Example:
The request returns successful:
{“result":"ok","content":”total:1;2|0|20160721132505|99|1| NULL|E”}
{“result":"ok","content":”total:1;2|0|20160721132505|99|1| MSISDN:13023569874|E”}

Description: Find a USSD which accords with the condition.
Total: Total number of USSD.
USSD structure: Port |Order number |Time |Content |Sending result |Receiving result |E
Sending result: 1 denotes successful, 0 denotes failed
Receiving result: NULL denotes no return

The request returns failed:
{“result”:“error”,“content”:“error reason”}
Description:
‘error reason’ describes the exact failure.

2.3.3 Deleting USSD

1) Request
POST http://GateWayIP/API/TaskHandle, or
POST http://GateWayIP/API/DeleteUSSD (use this link if deleting USSD frequently)

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of ‘deleteussd’.
port	string	The port used to delete USSD, separated by ‘,’. (-1) denotes all.
num	string	The sequence number used to delete USSD, separated by ‘,’. (-1) denotes all.

Example:
{“event”:“deleteussd”,“port”:“1,2,3”,”num”:”1,2,3”}
Description:
Delete the USSD of port 1, port 2 and port3 which sequence number is 1, 2 and 3.
{“event”:“deleteussd”,“port”:“1”,”num”:”-1”}
Description:
Delete the USSD content of port 1.
{“event”:“deleteussd”,“port”:“-1”}
Description:
Delete all USSD files.
Note:
1. The parameter “num” will be invalid and the USSD file will be deleted if the parameter “port” is set to -1.
2. Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description

result	string	Message Type: 'ok' means successful, 'error' means return failed.
content	string	Message Content: If successful, what returned is the query result; if failed, what returned is the failure reason.

Example:

The request returns successful:

{"result":"ok","content":"total:1"}

Description:

Deletion successful, delete an USSD file

The request returns failed:

{"result":"error","content":"error reason"}

Description:

'error reason' describes the exact failure.

2.4 Switching SIM Card

2.4.1 Switching SIM Card

1) Request

POST http://GateWayIP/API/SwitchCard

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of 'switchcard'.
port	string	The port where the SIM card needs switching.

Example:

{"event":"switchcard","port":"1"}

Description:

Switch the SIM card on Port 1 to the next available SIM card in another slot.

{"event":"switchcard","port":"1.B"}

Switch the SIM card on Port 1 to that in Slot B on Port 1.

Note: Authentication is required for the request. Use the account and the password specified for API on the gateway.

3) Return Value

Parameter	Category	Description
result	string	Message Type: 'ok' means successful, 'error' means return failed.
content	string	Message Content: If successful, what returned is the port information; if failed,

		what returned is the failure reason.

Example:

The request returns successful:

{“result”:“ok”,“content”:“1.1”}

Description:

Successful to switch the SIM card on Port 1.

{“result”:“ok”,“content”:“1.0”}

Description:

Fail to switch the SIM card on Port 1, probably due to no available SIM card.

The request returns failed:

{“result”:“error”,“content”:“error reason”}

Description:

‘error reason’ describes the exact failure.

2.5 Call Forwarding

1) Request

POST http://GateWayIP/API/TaskHandle, or

POST http://GateWayIP/API/SetCallForwarding (use this link if querying frequently)

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of ‘callforwarding’.
port	string	Inquired ports. Multiple ports are separated by ‘,’. If not filled, it will return ‘no port’; if the port is unavailable, it will prompt ‘exist error port’.
unconditional	string	Unconditional call forwarding. Fill in 0 if not to set.
busy	string	Call forward on busy, which belongs to conditional call forwarding. Fill in 0 if not to set.
noreply	string	Call forward on No Answer, which belongs to conditional call forwarding. Fill in 0 if not to set.
noreachable	string	Unreachable call forwarding, which belongs to conditional call forwarding. Fill in 0 if not to set.
cancleall	string	Cancel all, with the value set to 4. It is not required to set with conditional and unconditional call forwarding.

Notes:

1. In non-cancleall state, the four parameters unconditional, busy, noreply and noreachable all need to be set.
2. When setting conditional call forwarding, unconditional call forwarding must be set to 0 and vice versa. Conditional call forwarding cannot be set together with unconditional call forwarding

Example 1:

{"event":"callforwarding","port":"5,8","unconditional":"13544445555","busy":"0","noreply":"0","noreachable":"0"}

Note: Set unconditional call forwarding for Port 5 and 8, the number is 13544445555.

Example 2:

{"event":"callforwarding","port":"1,5,8","cancleall":"4"}

Note: Cancel all call forwarding settings for Port 1, 5 and 8.

3) Return Value

Parameter	Category	Description
result	string	Message Type: 'ok' means successful, 'error' means return failed.
content	string	Message Content: If successful, what returned is the wireless information; if failed, what returned is the failure reason.

Example 1:

The request returns successful:

{"result":"ok","content":"taskid:0"}

The request returns failed:

{"result":"error","content":"request error"}

2.5.1 Updating Call Forwarding Status

1) Request

POST http://GateWayIP/API/TaskHandle, or

POST http://GateWayIP/API/SetCallForwarding (use this link if querying frequently)

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of 'updatecallforward'.
port	string	Inquired ports. Multiple ports are separated by ','. If not filled, it will return 'no port'; if the port is unavailable, it will prompt 'exist error port'.

Example 1:

{"event":"updatecallforward","port":"5,8"}

Note: Update the call forwarding status for Port 5 and 8.

3) Return Value

Parameter	Category	Description
result	string	Message Type: 'ok' means successful, 'error' means return failed.

content	string	Message Content: If successful, what returned is the wireless information; if failed, what returned is the failure reason.

Example 1:

The request returns successful:

{"result":"ok","content":"taskid:0"}

2.5.2 Querying Call Forwarding

1) Request

POST http://GateWayIP/API/TaskHandle, or

POST http://GateWayIP/API/SetCallForwarding (use this link if querying frequently)

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of 'querycallforward'.
port	string	Inquired ports. Multiple ports are separated by ','. If not filled, it will return 'no port'; if the port is unavailable, it will prompt 'exist error port'.

Example 1:

{"event":"querycallforward","port":"5,8"}

Note: Query whether Port 5 and 8 are set successfully.

3) Return Value

Parameter	Category	Description
result	string	Message Type: 'ok' means successful, 'error' means return failed.
content	string	Message Content: Include the port and four call forwarding states: Unconditional, busy, noreply, noreachable.

Example 1:

The request returns successful:

{"result":"ok","content":"port:5,unconditional:+861315824514,busy:0,noreply:0,noreachable:0;port:8,unconditional:0,busy:0,noreply:0,noreachable:0;"}

Example 2:

The request returns failed:

{"result":"error","content":"request error"}

Port query error:

{"result":"error","content":"exist error port"}