

SMG Wireless Gateway API

Copyright © 1995-2017 Synway Information Engineering Co., Ltd
website: www.synway.net

Content

Content.....	i
Revision History	ii
Chapter 1 Overview	1
1.1 For Receiving Request	1
1.2 For Feeding Back Request.....	1
1.3 For Request Authentication	1
Chapter 2 API Description	2
2.1 Acquiring Device Information.....	2
2.1.1 Obtaining Port Status.....	2
2.1.2 Obtaining Port BS Connection Status	2
2.1.3 Obtaining Wireless Parameters Information	3
2.2 SMS	5
2.2.1 Sending SMS.....	5
2.2.2 Inquiring SMS Sending Result	6
2.2.3 Obtaining Inbox Information	7
2.2.4 Obtain Outbox Information	8
2.2.5 Delete Inbox.....	9
2.2.6 Query SMS Receipt.....	11
2.3 USSD.....	12
2.3.1 Sending USSD.....	12
2.3.2 Obtaining USSD Sending Result.....	13
2.3.3 Delete USSD	14

Revision History

Date	Version	Comments	Writer
2015-09-15	1.1	Initial publication	
2015-12-30	1.2	New Revision	
2016-08-18	1.3	New Revision	
2016-12-09	1.5	New Revision	
2017-04-16	1.7	New Revision	
2017-10-16	1.8	New Revision	

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.

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.

Chapter 2 API Description

2.1 Acquiring Device Information

2.1.1 Obtaining Port Status

1) Request

POST <http://GateWayIP/API/QueryInfo>

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: Information about the port with the request returning successful. 'error' 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 BS Connection Status

1) Request

POST <http://GateWayIP/API/QueryInfo>

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.

4) 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.

2.1.3 Obtaining Wireless Parameters Information

1) Request

POST <http://GateWayIP/API/QueryInfo>

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": "0,1,2"}
```

Description:

Query the IMEI for Port 0, Port 1 and Port 2.

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

5) 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.

2.2 SMS

2.2.1 Sending SMS

1) Request

POST <http://GateWayIP/API/TaskHandle>

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.

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.

2.2.2 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 ‘querytxsms’.
taskid	string	The task ID to query. More than one task can be query at the same time, and they are separated by ‘,’.
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.

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;2:10087:3],  
[taskid:1:1;3:10088:2]”}
```

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

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

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:0:1:3;1:10086:3]"}`

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

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

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.

2.2.3 Obtaining Inbox Information

1) Request

POST <http://GateWayIP/API/QueryInfo>

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:

`{"event": "queryrxsms", "begintime": "20150901180000",
"endtime": "20150915180000", "num": "10086"}`

Description:

What to get are the SMS from 10086 to Port 1 and Port 2 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(A)(15255558888):10086: Hello world!|E;
```

```
20150911111112:2(B)(-1):10086:2015091111112|: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.

2.2.4 Obtain Outbox Information

1) Request

POST <http://GateWayIP/API/QueryInfo>

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 destination number, separated by ',', optional.
port	string	Obtain the SMS from the designated port, separated by ',', optional.

Example:

```
{"event": "queryxsms", "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:

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

2.2.5 Delete Inbox

1) Request

[POST http://GateWayIP/API/TaskHandle](#)

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.
phonenumber	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": "1", "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 Query SMS Receipt

1) Request

POST <http://GateWayIP/API/QueryInfo>

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:

2.3 USSD

2.3.1 Sending USSD

1) Request

[POST http://GateWayIP/API/TaskHandle](#)

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>

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 Delete USSD

1) Request

[POST http://GateWayIP/API/TaskHandle](#)

2) Request Parameter

Parameter	Category	Description
event	string	The action of request, with the value of 'deleteusssd'.
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": "deleteusssd", "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": "deleteusssd", "port": "1", "num": "-1"}
```

Description:

Delete the USSD content of port 1.

```
{"event": "deleteusssd", "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.