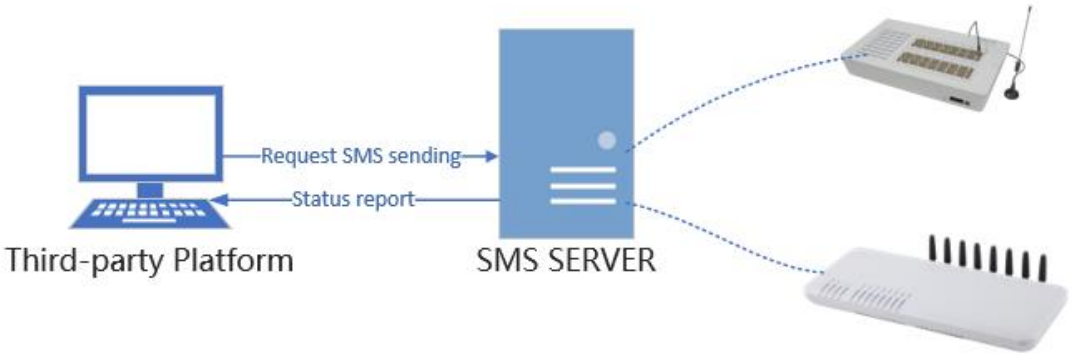
1 Overview

SMS Server v1.27 or above supports a highly efficient interface with a third party SMS platform. A typical network topology diagram is shown below.



This interface supports both http and json protocols.

Post syntax:

Post http://<IP>/goip/<command/

Json syntax for submitting data:

```
{
    "auth":{"username":"login name","password":"login password"},
    Other parameters …
}
```

Notes:

1. 【"auth":{"username":"login name","password":"login password"}】, any request must accompanied with the login name and password. If the login information is not provided, the SMS server will return an error message 401.
2. <command> refers to the port that is providing different functions.
   - ✧ sendsms, Sending SMS
   - ✧ querysms, Query SMS transmit status
   - ✧ querylines, Query the status of all lines in the GoIP
3. At the end of a URL request, "/" character cannot be omitted.
4. All characters MUST be encoded in UTF-8 format.

2 Sending SMS

2.1 Send a SMS Request

Command: sendsms

Json Parameters:

Name	Description
auth	Mandatory， Authentication information must be provided.
provider	Optional，Specify the Carrier name of a GoIP line.
goip_line	Optional，Specify which GoIP line is going to be used to send SMS.
number	Mandatory，Specify the recipient numbers (separated by ","and the max. is 9999)
content	Mandatory，The SMS content to be sent.

Example:

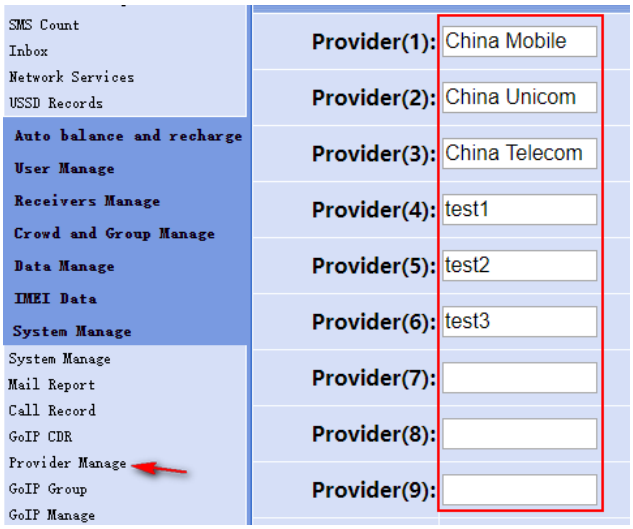
Using a China Mobile line to send out a SMS to the number 13600000000, 13611111111, 13622222222. The SMS content is "Testing".

Syntax:

```
POST http://<IP>/goip/sendsms/
{
    "auth":{"username":"login name","password":"login password"},
    "provider":"China Mobile",
    "number":"13600000000,13611111111,13622222222",
    "content":"Testing"
}
```

Notes:

1. Providers are defined in the in the SMS Server as shown below.



These names can be the actual provider names or names that are created for classification purpose.

2. goip_line and provider are optional.
   - ✧ If **provider** is specified, then a line that belongs to the provider will be used for sending SMS. A line will be reused based on the round robin method.
   - ✧ If **goip_line** is specified, then the specified line will be used.
   - ✧ If both **provider** and **goip line** are specified, then **goip line** has priority over **provider**. This means **provider** will be ignored.
   - ✧ If both provider and goip line are not specified, then all the line selection is based on the algorithm described in 4 below.
3. If the same number is repeated in a request, then they will be treated as one number only.
4. Line reusing algorithm: Round robin algorithm is employed. Initially, all lines within the group are put in a circular buffer. If **goip line** is specified, then the corresponding line is taken out from the buffer directly to send the SMS. Once the SMS is sent, it will then put to the end of the circular buffer. Otherwise, a line is taken out from the start of the buffer to send the SMS and put to the end afterward.

2.2 SMS Server's Response to a Send SMS Request

Once SMS Server receives a Send SMS Request, it immediately responds with a json message.

{"result":"ACCEPT","taskID":"<ID>"}
Or
{"result":"REJECT","reason":"<content>}

Name	Description
result	This reports back a message whether the send request is accepted or not. ✧ ACCEPT – Send request accepted. ✧ REJECT – Send request rejected.
taskID	Once the send request is accepted, a unique <ID> for the send request is generated.
reason	Reason for rejecting a send SMS request is specified in the <content> field ✧ none line – the line specified is not usable (either not able to register to GSM or SMS function is disabled). ✧ none_provider – provider not exist ✧ unknown – unknown

Examples:

{"result":"ACCEPT","taskID":"5689"}

{"result":"REJECT","reason":"none_line"}

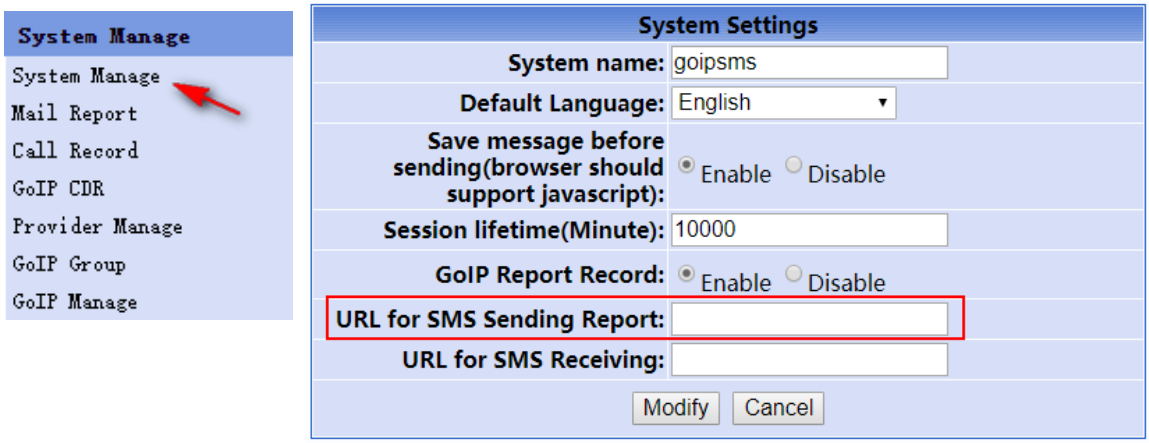
Note:

A send SMS request may contain up to 9999 recipient numbers. This means up to 999 SMS may have to be sent. For each send request, a "**taskID**" is generated and a sub ID is generated for each recipient number. Therefore, each individual SMS can then be queried with the reference number syntax as <task ID>.<number>.

For example, if the task ID generated is 5689 and the recipient number is 13600000000, then unique reference number for the SMS is 5689.13600000000. Please see section 3.1 for more information.

2.3 SMS Transmit Status Report

SMS Server can be configured to send a SMS transmit status report to a URL address as shown in the diagram below (red box).



When sending a SMS is completed, the SMS server immediately sends out a status report via http with the following Json data.

{"taskID":<ID>,""goip_line":"<line name>","send":"<status>","<otherparameter>":"<value>"}

Json Parameters:

Name	Description
taskID	This is the task ID including the recipient number. It identifies the SMS task is originated from which send request and the recipient number. Example: 3578.13612345678
goip_line	This identifies which GoIP line actually sent out the SMS.

send	SMS Status: ✧ succeeded – Sent completed ✧ failed – Sent failed ✧ unsend – SMS not sent yet ✧ sending – Sending SMS in progress
receipt	This indicate if there is a response from the recipient or not. When a response is received from the recipient, then send " receipt = 1"
err_code	This indicates the error code when sending the SMS failed. Please refer to 《Appendix A CMS ERROR》 for more information.

Examples:

1. Sending SMS failed:

Json data:

{"taskID":"5689.13600000000","goip_line":"G101","send":"failed","err_code":"350"}

2. Sending SMS succeed:

Json data:

{"taskID":"5689.13600000000","goip_line":"G101","send":"succeeded"}

3. Sending SMS succeed and a response is also received from the recipient.

Json data:

{"taskID":"5689.13600000000","goip_line":"G101","send":"succeeded","receipt":"1"}

Note:

SMS Server will not report status of those not able to send and those are sending in progress. They will be reported only upon receiving a "querysms" request.

3 Querying SMS Transmit Status

3.1 Send a SMS Query Request

In addition to sending certain status report automatically (see section 2), the SMS Server also supports a query command that allows a third party to make inquiry.

Command: querysms

Json Parameters:

Command	Description
auth	Mandatory, Authentication information must be provided
taskID	Mandatory, This specifies which SMS send request to be queried. taskID may also include the recipient number as described in section 2.

Examples:

1. Send a query request on a previous Send SMS request .

Syntax:

POST http://<IP>/goip/querysms/

```
{
    "auth":{"username":"login name ","password":"login passwod"},
    "taskID":"5689"
}
```

Note: taskID is the ID that is responded by the SMS Server after receiving the Send SMS Request.

2. Send a query request on a specific SMS transmission. The taskID must also include the recipient number.

Syntax:

POST http://<IP>/goip/querysms/

```
{
    "auth":{"username":"login name","password":"login password"},
    "taskID":"5689.13600000000"
}
```

3.2 SMS Server's Response to a SMS Status Query Request

The response to a SMS status query request is in an array of Json objects. Please refer to Section 2.3 for more details. Below is the syntax for response.

```
[
{"taskID":<ID1>,""goip_line":"<line1>","send":"<status>","<otherparameter>":"<value>"},
{"taskID":<ID2>","goip_line":"<line2>","send":"<status>","<otherparameter>":"<value>"},
.
.
.
]
```

Example:

Below is an example of a SMS status query response for the taskID 5689 with 4 recipients. Please note that all recipients are shown if the recipient number is not specified.

```
[
  {"taskID":"5689.13600000000","goip_line":"G101","send":"failed","err_code":"350"},
  {"taskID":"5689.13600000001","goip_line":"G102","send":" succeeded "},
  {"taskID":"5689.13600000002","goip_line":"null","send":"sending"},
  {"taskID":"5689.13600000003","goip_line":"null","send":"unsend"},
]
```

Notes:

1. If a taskID with a recipient number is specified, only the status json object for that number is reported in the array.
2. If the taskID specified does not exist, the returned response is an empty array.

4 Querying GoIP Line Status

4.1 Send a GoIP Line Status Query Request

Command: querylines

Json Parameter:

Name	Description
Auth	Mandatory，Authentication information must be provided.

Example:

POST http://<ip>/goip/querylines/

{"auth":{"username":"login name","password":"login password"}}

4.2 GoIP's Response to a GoIP Line Status Query Request

The response to a GoIP line status query request is in an array of Json objects with the format as show below.

[
{"goip_line":"<line1>","online":"<value>","reg":"<status>","remain_sms":"<value>","day_remain_sms":"<value>"},
{"goip_line":"<line2>","online":"<value>","reg":"<status>","remain_sms":"<value>","day_remain_sms":"<value>"},
…
]

Json Parameters:

Name	Description
goip_line	GOIP Line ID
online	SMS Server Connection status ✧ 1 – Connected ✧ 0 – Not Connected
reg	GOIP SIM Registration Status ✧ LOGIN – SIM registered to cellular ✧ LOGOUT – SIM not registered to cellular ✧ Null – Eqv. to LOGOUT
remain_sms	The remaining number of SMS transmissions permitted. "-1" means that there is no limit.
day_remain_sms	The daily remaining number of SMS transmissions permitted. "-1" means that there is no daily limited.

Examples:

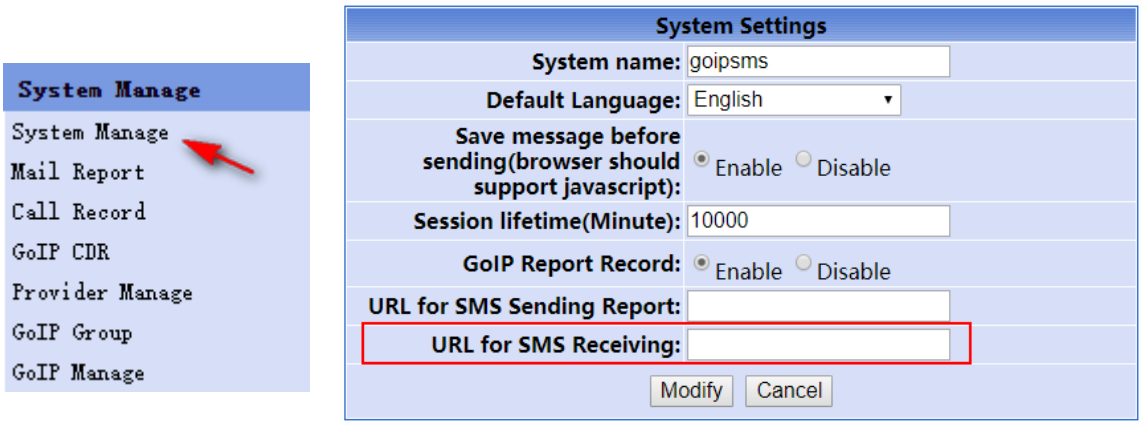

```
[
  {"goip_line":"G101","online":"1","reg":"LOGOUT","remain_sms":"-1","day_remain_sms":"-1"},
  {"goip_line":"G102","online":"1","reg":"LOGIN","remain_sms":"856","day_remain_sms":"56"},
  {"goip_line":"G103","online":"1","reg":"","remain_sms":"-1","day_remain_sms":"-1"},
  {"goip_line":"G104","online":"0","reg":"","remain_sms":"-1","day_remain_sms":"-1"}
]
```


Note:

The maximum and daily limits per GoIP line are configured in the SMS Server. The remaining number and the daily remaining number of SMS transmissions are stored in the parameters "remain_sms" and "day_remain_sms" respectively.

5 Forwarding SMS Received via GoIP Lines

SMS Server can be configured to forward the SMS received to the URL programmed in the System Settings below.



Once a SMS from a GoIP line is received, SMS Server forwards the SMS to the URL specified by using http POST command. The Json format is shown below.

Json Parameters:

Name	Description
goip_line	This shows the ID of the GoIP line received the SMS.
from_number	This shows the number of the SMS sender.
content	SMS content
recv_time	Time received and the format is: "YYYY-MM-DD hh:mm:ss"

Example:


```
{   "goip_line":"G101",
    "from_number":"10086",
    "content":"Dear Customer: This event has ended.  Thank you for your support!",
    "recv_time":"2019-09-26 15:26:05"
}
```

Appendix A CMS ERROR Codes

When SMS service failure occurs, the GSM network returns an error codes for the reason of failure. The error codes and their corresponding meanings are shown in the list below.

1, "Unassigned (unallocated) number"
8, "Operator determined barring"
10, "Call barred"
17, "Network failure"
21, "Short message transfer rejected"
22, "Memory capacity exceeded"
27, "Destination out of service"
28, "Unidentified subscriber"
29, "Facility rejected"
30, "Unknown Subscriber"
38, "Network out of order"
41, "Temporary failure"
42, "Congestion"
47, "Resources unvailable, unspecified"
50, "Requested facility not subscribed"
69, "Requested facility not implemented"
81, "Invalid short message reference value"
95, "Invalid message, unspecified"
96, "Invalid mandatory information"
97, "Message type non-existent or not implemented"
98, "Message not compatible with short message protocol state"
99, "Information element non-existent or not implemented"
111, "Protocol error, unspecified"
127, "Interworking unspecified"
128, "Telematic interworking not supported"
129, "Short message type 0 not supported"
130, "Cannot replace short message"
143, "Unspecified TP-PID error"
144, "Data coding scheme (alphabet) not supported"
145, "Message class not supported"
159, "Unspecified TP-DCS error"
160, "Command cannot be actioned"
161, "Command unsupported"
175, "Unspecified TP-Command error"
176, "TPDU not supported"

192, "SC busy"
193, "No SC subscription"
194, "SC system failure"
195, "Invalid SME address"
196, "Destination SME barred"
197, "SM Rejected-Duplicate SM"
198, "TP-VPF not supported"
199, "TP-VP not supported"
208, "SIM SMS storage full"
209, "No SMS storage capability in SIM"
210, "Error in MS"
211, "Memory Capacity Exceeded"
212, "SIM Application Toolkit Busy"
213, "SIM data download error"
224, "TP_FCS_APPL_ERR_START"
254, "TP_FCS_APPL_ERR_STOP"
255, "TP_FCS_UNSPECIFIED"
300, "ME failure"
301, "SMS service of ME reserved"
302, "operation not allowed"
303, "operation not supported"
305, "invalid text mode parameter"
312, "PH-SIM PIN necessary"
310, "SIM not inserted"
311, "SIM PIN required"
312, "SIM PUK required"
313, "SIM failure"
314, "SIM busy"
315, "SIM wrong"
317, "SIM PIN2 required"
318, "SIM PUK2 required"
319, "incorrect PUK1"
320, "memory failure"
321, "invalid memory index"
322, "memory full"
330, "SMSC address unknown"
331, "no network service"
332, "network timeout"
340, "no +CNMA acknowledgement expected"

512, "MN_SMS_RP_ACK"
513, "MN_SMS_TIMER_EXPIRED"
514, "MN_SMS_FORW_AVAIL_FAILED"
515, "MN_SMS_FORW_AVAIL_ABORTED"
516, "MS invalid TP-Message-Type-Indicator"
517, "MS no TP-Status-Report in Phase 1"
518, "MS no TP-Reject-Duplicate in Phase 1"
519, "MS no TP-Reply-Path in Phase 1"
520, "MS no TP-User-Data-Header in Phase 1"
521, "MS missing TP-Validity-Period"
522, "MS invalid TP-Service-Centre-Time-Stamp"
523, "MS missing TP-Destination-Address"
524, "MS invalid TP-Destination-Address"
525, "MS missing Service-Centre-Address"
526, "MS invalid Service-Centre-Address"
527, "MS invalid alphabet"
528, "MS invalid TP-User-Data-Length"
529, "MS missing TP-User-Data"
530, "MS TP-User-Data too long"
531, "MS no Command-Request in Phase 1"
532, "MS Cmd-Req invalid TP-Destination-Address"
533, "MS Cmd-Req invalid TP-User-Data-Length"
534, "MS Cmd-Req invalid TP-User-Data"
535, "MS Cmd-Req invalid TP-Command-Type"
536, "MN MNR creation failed"
537, "MS CMM creation failed"
538, "MS network connection lost"
539, "MS pending MO SM transfer"
540, "RP-Error OK"
541, "RP-Error OK no icon display"
542, "SMS-PP Unspecified"
543, "SMS rejected By SMS CONTROL"