

Methods of Using GoIP for SMS Applications

Application Note: How to use GoIP for SMS Applications

This application note describes how you can integrate GoIPs in a SMS platform for sending and receiving SMS.

Method 1: Using the SMS Server.

Requirement: Customers need to install SMS Server and configure GoIPs to register to the SMS Server. Please visit the <https://www.goip-pro.ru> to get more information on the SMS Server. Please contact technical support if you require help in installing and configuring the SMS Server.

Once GoIPs have registered to the SMS Server, you now have a pool of GSM channels for sending and receiving SMS. Next, you need to create your own http server to send and receive SMS via the SMS Server.

To send a SMS to the SMS server, your http server just need to use the HTTP API (both GET and POST methods are supported) in the SMS Server. The command syntax is

```
http://<sms server ip>/en/dosend.php?USERNAME=<your id>&PASSWORD=<your password>&smsprovider=<provider(1 to 10)>&goipname=<GoIP ID>&smsnum=<SMS recipient number(s)>&method=2&Memo=<message>
```

Notes:

1. USERNAME is your login name to the SMS server.
2. PASSWORD is your login password to the SMS server.
3. smsprovider specifies which provider in the SMS Server Provider List is going to be used for sending the SMS. There are a maximum 10 providers in the list and the reference number is used to refer to the provider. Please do not use the provider name defined. Please make sure that there are GoIP channel(s) assigned to the provider specified.
4. goipname is the GoIP ID defined for each GoIP channel. Please make sure that the status of the GoIP ID in the SMS Server is LOGIN. If this parameter is not specified, SMS Server will choose a GSM channel in the provider specified to send the SMS.
5. smsnum is the list of the phone number of the SMS recipients. Use command (,) to separate phone numbers.
6. method=2 is a fixed parameter when using the HTTP API to send SMS. Just specify it as is.
7. Memo is the SMS content (cannot exceed 1400 characters).

Examples:

1. <http://192.168.2.1/goip/en/dosend.php?USERNAME=root&PASSWORD=root&smsprovider=1&goipname=goip1&smsnum=10086,10087,10088&method=2&Memo=hello>
2. <http://192.168.2.2/goip/en/dosend.php?USERNAME=root&PASSWORD=root&smsprovider=1&smsnum=123457,123456&method=2&Memo=hello>

When the SMS Server receives a SMS, you can enable the SMS Server to POST the received message to your http server. This is done by assigning your http server address when you configure a GoIP channel as shown below. Three parameters are included in the HTTP message.

1. name is the GoIP ID representing the GoIP channel where the SMS is originated.
2. number is the phone number of the SMS sender.
3. content is the SMS received.

Modify a GoIP Settings	
ID:	108
Provider:	metfone
Group:	None(None)
New Password:	
Confirm Password:	
SMS Remain Count:	
SMS Max. Limit:	
SMS Remain Count This Day:	
SMS Daily Limit:	
Forward SMS to Email:	<input type="checkbox"/>
Forward SMS to HTTP:	<input checked="" type="checkbox"/>
POST HTTP Address:	http://192.168.1.2/receive
Will not change password with blank input box named "New Password"	
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

Method 2: Using SMMP Protocol

GoIP supports SMPP protocol version 3.3 and 3.4 and it functions as a SMPP SMSC. For more information, please download the SMPP Interface Manual from official website.

This method allows you to use a third party SMPP client to talk to the GoIP directly. We have tested the GoIP SMPP protocol with the [Ozeki NG – SMS Gateway version 4.2.60](#). This software is free to use for a limited time.

1. Enabling GoIP for SMPP SMSC service
 - a. Login to the GoIP built-in web and go to the Preference page under Configuration.
 - b. Enable SMPP SMSC
 - c. Fill in the ID, Password, and Port accordingly. ID is the name used to refer to the whole GoIP. If you want to specify which channel is used to send SMS, you will have to add its channel number (2-digit). 01 is referred as the channel 1. For example, if the ID is set to “goip1”, then the ID name for name for channel 1 is “goip101”. Please always take a note of the number of channels available when addressing an individual channel.

SMPP SMSC	<input checked="" type="radio"/> Enable <input type="radio"/> Disable
ID	<input type="text"/>
Password	<input type="text"/>
Port	7777

2. Configuring a SMPP External Short Message Entities (ESME)

- a. This is a non-mobile entity that submits message or receives messages from an SMSC. You will have to develop your own ESME or purchase it from a third party. There are open source SMPP client that you can use to develop your own ESME. As for our testing, we use the Ozeki NG – SMS Gateway and the configuration to the GoIP (SMSC) is shown below.

- Host is the IP address of your GoIP (SMSC). If it is installed behind NAT, the address here is the external address (public IP) of your NAT. You also need to configure the NAT to forward the port number specified to the IP address of the GoIP.
- Port must be the same as the one programmed in the GoIP.
- Username is the ID name programmed in the GoIP. In this example, the ID specified is “goip1” rather than a specific channel in the GoIP. When messages are received, the GoIP send out the messages via all channels in a round robin manner. Alternatively, if the Username is set to a specific channel, for example “goip101”, then GoIP will only send the messages via the channel specified.
- Password is the password programmed in the GoIP.

Please specify the SMPP server settings. (This information is provided by the SMS service provider when you subscribe to the service.)

Host:

Port: Use source

Username: (Optional, 0

Password:

Please specify the telephone number and assign a service provider name

- b. It is recommended that the Transceiver mode should be used in the Connection settings as shown below. This allows messages to be send and receive at the same time .

SMPP settings | Logging | **Connection** | Message su

SMPP bind / unbind

Protocol: v3.3 v3.4

System type:

Bind mode: Transmitter
 Receiver
 Transceiver

Address range:

Bind TON:

Bind NPI:

Send SMPP UNBIND at disconnect

Override Reconnect time to seconds

3. Sending SMS

Messages sent from an ESME can be encoded in UCS2 or ASCII. For UCS2 format, the maximum number of characters in a message is 70. For ASCII format, it is converted automatically to 7-bit ASCII for transmission and the maximum number of characters in a message is 160.

Message Type	SMS:TEXT
To: (Addressbook)	10086
Message text: 4 character(s), 1 SMS message(s)	test
	<input checked="" type="checkbox"/> Include newline characters in message.
	<input type="checkbox"/> Schedule for later sending
	<input type="button" value="OK"/>

4. Message Response from GoIP

Below capture shows the message responses from GoIP.



Method 3: Using GoIP native HTTP Interface

GoIP can accept messages via its HTTP interface and then sends them out via its GSM channel(s) accordingly. This is a simple way send messages directly via GoIP.

1. Send a message via a HTTP Send command

Example:

http://192.168.2.190/default/en_US/send.html?u=admin&p=admin1&l=1&n=10086&m=test

where

192.168.2.190 is IP address of the GoIP

u=admin means username=admin

p=admin1 means password=admin1

l=1 means using GSM Channel/Line 1 to send the message

n=10086 means the SMS recipient number is 10086

m=test means the message content is "test"

Here are the two possible responses from the HTTP Send command.

1. If an error occurs, it returns "ERROR, <reason>"
Possible reason could be user error, password error, Lx GSM logout, etc.
2. If the message is accepted, it returns "Sending, L1 Send SMS to 10086; ID:55c489da"

2. Send a HTTP inquiry to check on the message status.

Example:

http://192.168.2.190/default/en_US/send_status.xml?u=admin&p=admin1

Here is a typical response to the message sent above. Since the GoIP used has 8 channels, all channel status are shown here. The status for the message ID = 55c489da is "DONE", this means that the message is sent successfully.

```

▼ <send-sms-status>
  <id1>55c489da</id1>
  <status1>DONE</status1>
  <error1/>
  <id2/>
  <status2>DONE</status2>
  <error2>CSML_LOGOUT</error2>
  <id3/>
  <status3>DONE</status3>
  <error3/>
  <id4/>
  <status4>DONE</status4>
  <error4/>
  <id5/>
  <status5>DONE</status5>
  <error5/>
  <id6/>
  <status6>DONE</status6>
  <error6/>
  <id7/>
  <status7>DONE</status7>
  <error7/>
  <id8/>
  <status8>DONE</status8>
  <error8/>
</send-sms-status>

```

3. When the message is sent successfully, then the next message can be sent.

Method 4: Using GoIP's SMS API

The SMS Server mentioned in Method 1 is written using the GoIP's SMS API as described in the document [SMS Interface User Manual](#) which can be downloaded from official website.

Instead of communicating with the SMS Server, you can use this SMS API to develop your SMS application to communicate with the GoIP via UDP directly. This method demands more development capability and effort.