



CloudSMS API Guide

Version 5.2

Table of Content

Overview	3
Integrated Push Messaging	4
PushBox Mobile App	4
Your own mobile app via MACH Push Messaging Service	5
MT API Details	5
Send Message	5
Send Long Concatenated Messages	12
MO API Details	13
Receive Message	13
Message Delivery Notification	15
Delivery Notification API details	15
Appendix A: List of Return Codes	17

1 Overview

CommzGate CloudSMS provides a set of functions for mobile message sending and receiving exposed as HTTPS-based API. The API defined in this document are for applications or services to make use of the CommzGate Cloud Service over the Internet or private leased network.

To make use of the CloudSMS API, you require an account from CommzGate as well as the URL of the CommzGate CloudSMS Service on the Internet.

You can get a CloudSMS account at <https://portal.commzgate.com/app/signup.php>

Once you have logged in to your web account, click on the API tab to get your API ID. Use the link provided to create a password for your API ID as well.

The CommzGate CloudSMS URL that you call will be in the form of: <https://www.commzgate.net/gateway/SendMsg>

For Pro Plan customers, you have the option of using our reserved API domain at <https://vip.commzgate.net>

This reserved domain requires that your source IP address be whitelisted. Please contact your account manager regarding source IP whitelisting.

HTTP Methods:

The POST method may be used for the API unless explicitly stated otherwise.

There are two main categories in the API:

A) MT (Mobile Terminate) API involve sending messages from the **CommzGate CloudSMS to Mobile Device**.

B) MO (Mobile Originate) API involve receiving messages from the **Mobile Device to CommzGate CloudSMS**.

For MT API, HTTP POST request is initiated by your Application and directed to the CommzGate CloudSMS Service.

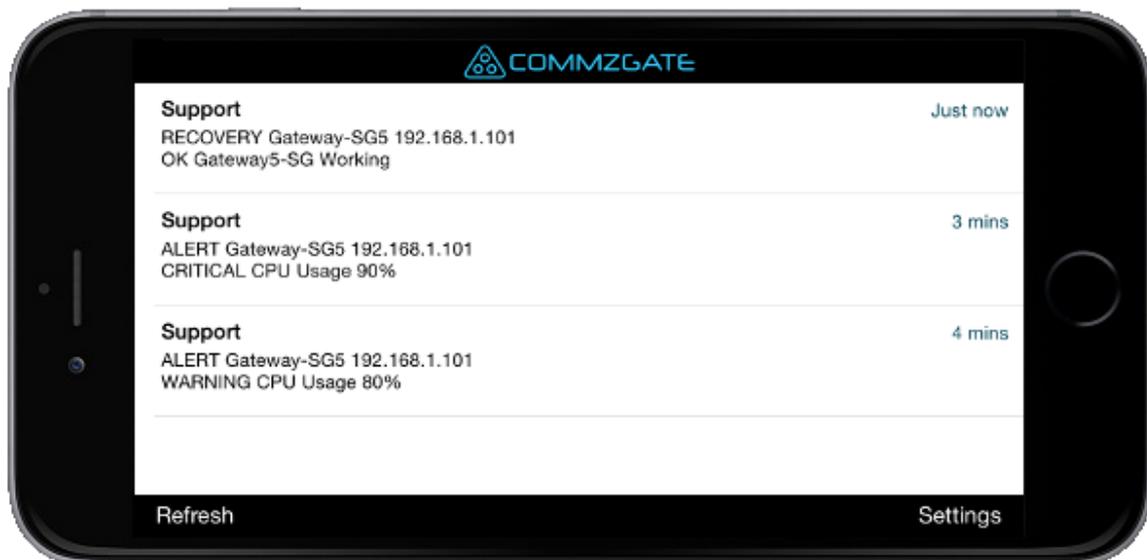
For MO API, HTTP POST request is initiated by the CommzGate CloudSMS Service and directed to your Application.

This document defines the API, their parameters, and return values. Note that parameter names are **case sensitive**. Also, none of the parameter values that are passed in should contain the pipe “|” character. The standard HTTP encoding rules must be used when passing text as parameters.

2 Integrated Push Messaging

2.1 PushBox Mobile App

The CommzGate CloudSMS API supports the sending of Push Notification messages to recipients who have the PushBox mobile application installed on their smartphones.



PushBox is perfectly suited for organizations with teams or a member base, it can help you improve communications drastically while keeping the costs low.

PushBox is supported for iPhone and Android, and can be downloaded from:
<http://www.commzgate.com/page/downloads/>

2.2 Your own mobile app via MACH Push Messaging Service

MACH is our push-messaging service that lets you send mobile messages using our CloudSMS Portal & CloudSMS API.

MACH Push Messaging is currently supported in the MT API. (See section 3) Using the MT API, recipient numbers with MACH-enabled mobile apps installed will automatically receive the message as a MACH Push Notification message. Otherwise the message will be delivered via SMS.

If you wish to integrate your own smartphone app to MACH, you can do so by registering for a free account at <http://app.themach.com>

3 MT API Details

3.1 Send Message

URL

<https://www.commzgate.net/gateway/SendMsg>

Description Sends a message to the specified mobile number.

HTTP Method POST

Field	Description
ID	<i>Type=String, Length=50</i> API ID, pre-assigned by the CommzGate CloudSMS Service. You can view this value by logging in to your CloudSMS Portal account.
Password	<i>Type=String, Length=50</i> API password. You can configure or change this by logging in to your CloudSMS Portal account.

Mobile	<p><i>Type=String, Length=20</i></p> <p>Message recipient mobile phone number (single number, include country code and exclude “+” sign) e.g. 6591502554</p> <p>See the ‘Batch’ parameter option if you wish to send the same message to more than 1 mobile number using a single MT API request.</p>
Type	<p><i>Type=Char, Length=1</i></p> <p>Message Content Type: “A” – ASCII content of less than 160 characters “LA” – ASCII content of less than 459 characters “H” – Hexadecimal “U” – Unicode “AUTO” – Automatically detect</p> <p>When using the ‘Type=LA’, the message ID returned will be suffixed by a ‘C’. eg. 01010,1_14931761715062610C Depending on the exact number of message parts the long message content is split into, the actual message ID of each message part will be in the form 1_14931761715062610_XXC where XX indicates the part number of the long message. eg. Part 1: 1_14931761715062610_01C Part 2: 1_14931761715062610_02C</p> <p>When using ‘Type=U’, send the message content as double-byte UTF-16 represented using hexadecimal characters. See example in next section.</p> <p>When using the ‘Type=AUTO’, the Type parameter value will be automatically selected based on message content. Long messages are also supported.</p> <p>If you send a message containing characters within the GSM character set, the Type parameter value will be set to A. If this message content is more than 160 characters, with Type set to LA.</p>

	<p>If you send a message containing UTF-8 characters outside of the GSM character set, and the entire message content will be automatically converted into 16-bit hexadecimal with the Type parameter value set to U. <i>e.g 使用 english</i></p>
Message	<p><i>Type=String, Length=Variable (see below)</i></p> <p>Message Content If Type = "A", not more than 160 characters If Type = "LA", not more than 459 characters</p> <p>If Type = "H", not more than 280 (less the length of UserHeader value) in hexadecimal characters If Type = "U", not more than 70 characters</p>
Sender Optional	<p><i>Type=String, Length=11 (max)</i></p> <p>Alphanumeric Originating Address Also know as Number Masking or TPOA</p> <p><i>* IMPORTANT: To prevent spoofing of Number Mask values, we require you register Number Mask values you wish to use with us, else using this option will result in your message not being delivered. Number Mask registration charges may apply.</i></p>
UserHeader Optional	<p><i>Type=String, Length=not more than 280 (less the length of Message value) in hexadecimal characters.</i></p> <p>This is the value of the User Data Header that is used for certain messages types, such as long concatenated messages.</p>
OperatorID Optional	<p><i>Type=String, Length=20 (Case-Sensitive)</i></p> <p>Specifying this field will override the default MT routing. The exact parameter value must be obtained from the CommzGate..</p> <p><i>* IMPORTANT: You must have an Operator value assigned to you by you account manager in order to use this option.</i></p>

OTP Optional	<p><i>Type=String, Length= variable (Case-Sensitive)</i></p> <p>Works only when the Type parameter is set to 'A'</p>
	<p>Set this to 'true' to enable basic OTP mode.</p> <p>This parameter allows you to send SMS containing One-Time Passwords (OTP). Specifying OTP='true' will notify the CommzGate CloudSMS API to look for the placeholder *OTP* in your message content, and replace this with a randomly generated 5-digit number (the OTP). This same OTP 5-digit number will be appended to the Return String that is returned to your requesting application, after the Message ID, separated by a comma. Use this OTP value returned to you to compare against the input value submitted by the mobile recipient, they should match.</p>
	<p>Set this to 'advanced' to enable advanced OTP mode.</p> <p>In advanced mode, the message workflow mimics that of a traditional RADIUS authentication system. Additional parameter options for sending OTP SMS are available to configure OTP length, expiry time etc. and OTP validation is provided by the CloudSMS API.</p> <p>The following describes how an advanced mode OTP is triggered and subsequently validated.</p> <ol style="list-style-type: none"> i. Customer System (eg. Web Portal) validates initial user login credentials (eg. via local LDAP or Active Directory node) ii. If credentials are validated, Customer System sends OTP request via HTTPS to "CloudSMS", which in turn generates and delivers OTP SMS to user. iii. User enters OTP value into Customer System, the Customer System in turn validates the OTP value entered against "CloudSMS" via a HTTPS request. See Section 3 for the 'OTP Validation' API.

	<p>iv. “CloudSMS” replies with the OTP validation result. (e.g. ACCEPT or REJECT)</p> <p>v. Customer System responses to User Login request, based on OTP validation result returned by “CloudSMS”.</p> <p>This feature is available for Pro Plan users only. Please contact your account manager to activate this feature and to view the complete API addendum.</p>
<p>MACH</p> <p>Optional</p>	<p><i>Type=String, Length= variable (Case-Sensitive)</i></p> <p>Set this to ‘false’ to route the message via SMS only.</p> <p>If a recipient mobile number is registered on the CommzGate PushBox mobile app or a custom MACH-enabled mobile app, the message will arrive on the user device as a Push Notification by default. Use the parameter ‘MACH=false’ to force-route the message to be delivered via SMS.</p>
<p>Batch</p> <p>Optional</p>	<p><i>Type=String, Length= variable (Case-Sensitive)</i></p> <p>Set this to ‘true’ enable batch MT mode and use the following API URL: https://www.commzgate.net/gateway/SendBatchMsg.php</p> <p>You are able to include up to 1000 mobile numbers in the ‘Mobile’ parameter, with each mobile number separated by a comma. (,) i.e. You will be able to send the same message content to up to 1000 mobile numbers, via a single POST request.</p> <p>For successful requests, a single batch message ID will be returned. To check on the status of each individual message in the batch request, use this batch message ID on the CloudSMS portal to retrieve the details of each message in the batch.</p>

Return String	Description
---------------	-------------

Return Code	The success/error acknowledgement (from the CommzGate CloudSMS Service) of the API request. Defined in Return Codes List.
Message ID	This is a unique message identifier that is generated by CloudSMS.
OTP	Optional. If you had enabled the OTP parameter in your MT request, the 5-digit number generated will be returned here.

A- Example for English Message

ID=1000002

Password= qrxy53tohh88

Mobile=6591122334

Type=A

Message=Happy+Birthday+to+you

The following return string is returned:

01010, 01_118334327757067122

The return string values are mapped as follows:

Result Code 01010 (Success)
 Message ID 01_118334327757067122

B- Example for Chinese Message

(To send the Character 人)

ID=1000002

Password= qrxy53tohh88

Mobile=6591122334

Type=U

Message=4EBA

The following return string is returned:

01010, API_01_118334327757067122

The return string values are mapped as follows:

Result Code 01010 (Success)

Message ID 01_118334327757067122

C- Example for Binary Message

(To send a Nokia Ringtone)

ID=1000002

Password= qrxxy53tohh88

Mobile=6591122334

Type=H

UserHeader=06050415810000

Message=024A3A51D195CDD004001B2055059061056055855054854082084990000

The following return string is returned:

01010

The return string values are mapped as follows:

Result Code *01010 (Success)*
Message ID 01_118334327757067122

D- Example for English Message with OTP

ID=1000002

Password= qrxxy53tohh88

Mobile=6591122334

Type=A

Message=Your One-Time Password is *OTP*

OTP=true

The following return string is returned:

01010, API_01_118334327757067122,34215

The return string values are mapped as follows:

Result Code *01010 (Success)*
Message ID 01_118334327757067122
OTP generated 34215

3.2 Send Long Concatenated Messages

If you are looking to send long concatenated messages easily, consider using the 'Type=LA' parameter as described in the MT API, instead of using the 'UserHeader' parameter.

The parameter "UserHeader" is used when sending special binary formatted message (melodies, pictures etc) or when concatenating multipart messages into 1 long message.

The data sent in the UserHeader corresponds to the User-Data-Header segment within Transport Protocol Data Unit (PDU) as defined in the GSM specifications (3GPP TS 23.040).

An example UserHeader for sending a long concatenated message is

```
0500030F0201
```

In this example UserHeader the first 3 Octets *050003* represent the standard identifier for a concatenated message:

```
05--User Data Header Length (5 octets)
```

```
00--Concatenated message (8-bit reference)
```

```
03--Information Element Data Length
```

The next 3 Octets are described as follows:

```
0F--Concatenated Message Reference. This is a modulo 256 number which remains the same for all segments composing a concatenated message.
```

```
02--Total Message Parts (value 0-255) . This number represents the number of segments composing the concatenated message.
```

```
01--Part Number of this Segment (value 0-255). The first segment of a concatenated message has a sequence number of 1. Value 0 is reserved.
```

Further examples of User Header settings can be obtained from each respective handset manufacturers' SMS specification documents. (eg. Nokia Smart Message FAQ)

As long as the required binary User Data Header info is submitted in the *UserHeader* parameter and the rest of the binary message body in the *Message* parameter, the SMS will be delivered as specified.

4 MO API Details

4.1 Receive Message

Whenever the CommzGate Cloud Service receives an SMS, it will look up its own Application Routing table to determine which application the SMS is to be forwarded to. Once this has been resolved, the CommzGate Cloud Service will post a HTTP request to the application and pass in argument values.

The application must have a web-component that is able to receive a **HTTP POST method call**. The URL path of the application's receiving component must be pre-registered with the CommzGate Cloud Service.

Field	Description
Mobile	The originating mobile number of the received message
Type	Message Content Type: "A" – ASCII "H" – Hexadecimal "U" – Unicode This indicates the type of Message Content being received. For Type=U, the Message Content is in Unicode UTF-16 represented by Hex characters, 2 Hex characters per character. For a mixture of English and non-English content, the Type is set to U.
SMTtype	This parameter applies only if Type="H". The value of this field is taken as <destination port><originating port> in the User Data Header, according to the Nokia Smart Messaging specifications; some are shown below- VCARD 23F40000 LOGO 15820000

	PICTURE 158A0000 RINGTONE 15810000
Message	<p>Message Content</p> <p>If Type = "A", not more than 160 characters</p> <p>If Type = "H", not more than 280 HEX characters</p> <p>If Type = "U", not more than 280 HEX characters in UTF-16</p> <p>(URL-Encoded content)</p> <p>If the message is a Smart Message (e.g. ring-tone, logo etc), Type will be set to H and the Message field will contain only the content (no header information) of the Smart Message. This will contain the ENTIRE Smart Message, since the CommzGate Cloud Service will perform the concatenation of the message parts.</p>
Timestamp	The date-time that the message was received by the CommzGate Cloud Service. The format is YYYY-MM-DD hh:mm:ss
ServiceNum	This value specifies the Service Number / short-code / mobile number to which the SMS MO was sent.
ConnID	This value specifies the unique ID of the Connector through which the CommzGate Cloud Service received the incoming message. The possible values of the ConnID must be obtained from the deployment architect of the system.
OperatorID	This value specifies the mobile operator / service provider that the Connector receives the incoming message from. Note that this is different from the ConnID in that there may be multiple Connectors for a single mobile operator / service provider. If you only need to know which mobile operator / service provider the message is from (and not the <i>specific connector</i>), then use this field.
MessageID	This is the unique message ID assigned to the message by CommzGate. Use this for cross-checking with CommzGate when troubleshooting issues.

Example

Assuming the receive path of the application is:

202.22.123.5:8081/messagegateway/receive.jsp

The CommzGate Cloud Service receives an incoming SMS and determines that it is to be routed to this application. The CommzGate Cloud Service then calls a HTTP POST with the following details:

202.22.123. 5:8081/messagegateway/receive.jsp

Mobile=6598765432

Type=A

Message=Testing

Timestamp=2003-02-03 20:15:05

ServiceNum=1944367

Note: The CommzGate Cloud Service will expect an acknowledgement code 'OK' to be returned by the application, otherwise it will treat the message as undelivered and will either retry, or log the message as 'Undelivered'.

5 Message Delivery Notification

NOTE: This is an optional add-on feature available only with a Pro account

Whenever you successfully submit a message to the CommzGate Cloud Service, and the success code returned by the CommzGate Cloud Service only indicates the successful acceptance of your message attempt for subsequent delivery to the appropriate Mobile Operator the message is intended for. The Mobile Operator might subsequently accept this message or it might be rejected for one reason or another.

To find out the eventual status of the message submitted, you can either manually review the message status reports via the Customer Access Portal, or you can setup your application to receive the Delivery Notification automatically sent by the CommzGate Cloud Service via a HTTP POST.

5.1 Delivery Notification API details

The application must have a web-component that is able to receive a ***HTTP POST method call***. The URL path of the application's receiving component must be pre-registered with the CommzGate Cloud Service. If you have already setup a web-component to receive MO messages from CommzGate Cloud Service, this same web-component must be setup to receive Delivery Notifications as well.

*Delivery Notification feature is optional and must be enabled for your account.

If you have already setup a Web Component to receive MO messages from CommzGate Cloud Service (see Section 3), the following are additional parameters fields that CommzGate Cloud Service will send as HTTP POST to you.

Field	Description
MsgID	The unique message ID for the message that was successfully submitted to the CommzGate Cloud Service. This message ID was returned by the CommzGate Cloud Service to your application as part of the success code. (See Section 2) The MsgID is used to identify the message for which the Status value is for.
Status	This is the Status of the messages that was submitted by you application earlier. Possible values are: SUCCESS STILL IN QUEUE ERROR- PREPAID BLOCK ERROR- NUMBER OUTPORTED ERROR- NUMBER INVALID ERROR- TELCO-SIDE ERROR

Example

Assuming the receive path of the application is:

202.22.123.5:8081/messagegateway/receive.jsp

The CommzGate Cloud Service then calls a HTTP POST with the following details:

202.22.123. 5:8081/messagegateway/receive.jsp

MsgID= API_01_118334327757067122

Status=SUCCESS

Note: On receiving the Delivery Notification, your application has to return a HTTP header 200. No other return acknowledgement from your application is necessary.

Appendix A: List of Return Codes

Result Code	Description
01010	SMS MT request successfully submitted.
01011	Invalid request format, please check parameters and parameter values.
01012	Unauthorized access. Either the API ID does not exist or API password is invalid, the account has been suspended, or the request is being originated from a non-authorized IP address.
01013	Transient System error, please retry after 60 seconds
01014	Unable to route to mobile operator for the attempted mobile number.
01015	The credit balance for the API account is not sufficient. (for prepaid accounts only)
01018	The mobile number attempted is blacklisted.

PACIFIC SYNERGY

Pacific Synergy Pte Ltd
1 Commonwealth Lane #03-07
ONE COMMONWEALTH
Singapore 149544
Phone +65 64750 450

www.commzgate.com

CommzGate is a registered trademark of Pacific Synergy Pte Ltd
Various product and service names referenced herein may be trademarks of Pacific Synergy Pte Ltd.
All other product and service names mentioned may be trademarks of their respective owners.

Copyright © 2001-2019 Pacific Synergy Pte Ltd
All rights reserved.