



---

# SendQuick MIM API Guide

## Version 2.1

---

TalariaX Pte Ltd  
No 76, Playfair Road,  
#08-01, LHK2  
Singapore 369583

E-mail: [info@talariax.com](mailto:info@talariax.com)  
Web: [www.talariax.com](http://www.talariax.com)

## Table of Contents

1. Introduction .....	3
1.1 Overall Transaction Flow .....	3
2. Send and Receive MIM API .....	3
2.1 Sending Messages .....	3
2.1.1 HTTP format.....	5
2.1.2 JSON format.....	5
2.1.3 XML format.....	7
2.2 Receiving Messages .....	9
2.3 Receive Outgoing Message Status .....	10
2.4 Attachment Support.....	11
2.5 Whatsapp Template Messages.....	11
2.6 Status Code List .....	12

## 1. Introduction

This document gives an overview on the different methods of connecting between sendQuick SMS gateway (sQ) and an application server (AppServ), for sending and receiving mobile instant messenger (MIM) messages.

### 1.1 Overall Transaction Flow

Refer to the diagram below:

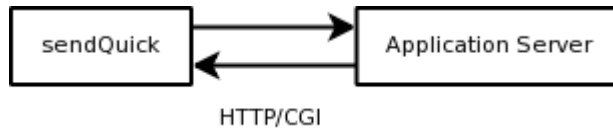


Figure 1.1: Overall Transaction Flow

Description:

1. Both sQ and AppServ will be communicating via standard HTTP POST access.
2. AppServ need to enable its web service to accept HTTP POST submit from sQ.
3. sQ will need to be configured (per MIM routing) with the URL path submission to AppServ.
4. For sending messages, AppServ will need to initiate a HTTP POST to the API of the sQ system.

## 2. Send and Receive MIM API

### 2.1 Sending Messages

API URL path: /api/message.php

This single API supports three (3) message request format. An equivalent HTTP header must be supplied according to the format used:

<b>Format</b>	<b>HTTP Header</b>
HTTP	Content-Type: application/x-www-form-urlencoded
JSON	Content-Type: application/json
XML	Content-Type: application/xml

The input parameters are as follows:

<b>Parameter</b>	<b>Data Type</b>	<b>Description</b>	<b>Possible Value</b>
id (Mandatory)	UUID	<ul style="list-style-type: none"> <li>Recipient's UUID (GUID)</li> <li>Mobile number (Whatsapp)</li> </ul>	123e4567-e89b-12d3-a456-426655440000
text (Mandatory)	Alphanumeric	<ul style="list-style-type: none"> <li>Message</li> <li>Attachment URL<sup>@</sup></li> <li>Template messages (Whatsapp)<sup>w</sup></li> </ul>	Example 1: This is a test message for ASCII. Example 2: 안녕하세요 수 있습니다 for UTF8.
type (Optional)	Alphanumeric	Content type: <ul style="list-style-type: none"> <li>text – ASCII text messages</li> <li>utf8 – UTF8 messages (non-ASCII characters)</li> <li>image, file, audio or video</li> </ul>	text (default)
priority (Optional)	Integer	Value should be an integer between 1 (highest) to 9 (lowest).	5 (default)
clientid (Optional)	Alphanumeric	Identifier tag for your application.	marketing1
trackid* (Optional)	Integer	Tracking ID for each request.	123456789
status_url* (Optional)	Alphanumeric	Post back URL when the message has been processed and completed.	http://192.168.1.10/webroot/mymessage-status.php
template (Optional)	Alphanumeric	For Whatsapp only. Template Name as configured in the sendquick.io portal <sup>w</sup>	issue_resolution

\* The usage of trackid and status\_url will be discussed in the section 2.3

@ Sending attachments will be discussed in the section 2.4

# Standard HTTP POST treat '+' as a blank space.

<sup>w</sup> Sending Whatsapp template messages will be discussed in the section 2.5

## 2.1.1 HTTP format

### Example for Single Recipient

[http://192.168.1.101/api/message.php?action=send&id=123e4567-e89b-12d3-a456-426655440000&text=Hello&type=text&priority=5&clientid=testApp&status\\_url=http://abc.com&trackid=123456789](http://192.168.1.101/api/message.php?action=send&id=123e4567-e89b-12d3-a456-426655440000&text=Hello&type=text&priority=5&clientid=testApp&status_url=http://abc.com&trackid=123456789)

### Example for Multiple Recipients

[http://192.168.1.101/api/message.php?action=send&id=123e4567-e89b-12d3-a456-426655440000,765e4321-e89b-65d4-a321-426655440000&text=Hello&type=text&priority=5&clientid=testApp&status\\_url=http://abc.com&trackid=123456789,987654321](http://192.168.1.101/api/message.php?action=send&id=123e4567-e89b-12d3-a456-426655440000,765e4321-e89b-65d4-a321-426655440000&text=Hello&type=text&priority=5&clientid=testApp&status_url=http://abc.com&trackid=123456789,987654321)

\* This example uses GET for ease of understanding, actual request must be made using POST.

### Example Response

Type	Response
For single recipient	status:200,description:Success,id:<trackid>
For multiple recipients <sub>n</sub>	status:200,description:Success,id:<trackid>, ... ,<trackid <sub>n</sub> >
For error	status:400,description:Missing request body

\* Refer to section 2.6 for the full status code list

## 2.1.2 JSON format

### Example for Single Recipient

```
{
  "action": "send",
  "payload": {
    "message": {
      "type": "text",
      "text": "Hello",
      "priority": "5",
      "clientid": "marketing1",
      "status_url": "http://abc.com",
      "template": "sample_template",
    },
    "recipients": [
      {"id": "123e4567-e89b-12d3-a456-426655440000", "trackid": "123456789"}
    ]
  }
}
```

### **Example for Multiple Recipients**

```
{
  "action": "send",
  "payload": {
    "message": {
      "type": "text",
      "text": "Hello",
      "priority": "5",
      "status_url": "http://abc.com",
    },
    "recipients": [
      {"id": "123e4567-e89b-12d3-a456-426655440000", "trackid": "123456789"},
      {"id": "765e4321-e89b-65d4-a321-426655440000", "trackid": "987654321"}
    ]
  }
}
```

### **Example Response**

Type	Response
For single recipient	{ "data": {

	<pre>                 "status":200,                 "description":"Success",                 "messages":[{"id":"&lt;trackid&gt;"}]             }         }     </pre>
For multiple recipients n	<pre>         {             "data":             {                 "status":200,                 "description":"Success",                 "messages":[                     {"id":"&lt;trackid&gt;"},                     ...                     {"id":"&lt;trackid_n&gt;"}                 ]             }         }     </pre>

## 2.1.3 XML format

### Example for Single Recipient

```

<?xml version="1.0"?>
<data>
    <action>send</action>
    <payload>
        <message>
            <type>text</type>
            <text>Hello</text>
            <priority>5</priority>
            <clientid>marketing1</clientid>
            <status_url>http://abc.com</status_url>
            <template>sample_template</token>
        </message>
        <recipients>
            <id>123e4567-e89b-12d3-a456-426655440000</id>
            <trackid>123456789</trackid>
        </recipients>
    </payload>
</data>
    
```

### **Example for Multiple Recipients**

```

<?xml version="1.0"?>
<data>
  <action>send</action>
  <payload>
    <message>
      <type>text</type>
      <text>Hello</text>
      <priority>5</priority>
      <status_url>http://abc.com<status_url>
    </message>
    <recipients>
      <id>123e4567-e89b-12d3-a456-426655440000</id>
      <trackid>123456789</trackid>
    </recipients>
    <recipients>
      <id>765e4321-e89b-65d4-a321-426655440000</id>
      <trackid>987654321</trackid>
    </recipients>
  </payload>
</data>

```

### **Example Response**

Type	Response
For single recipient	<pre> &lt;?xml version="1.0"?&gt; &lt;data&gt;   &lt;status&gt;200&lt;/status&gt;   &lt;description&gt;Success&lt;/description&gt;   &lt;messages&gt;     &lt;id&gt;&lt;trackid&gt;&lt;/id&gt;   &lt;/messages&gt; &lt;/data&gt; </pre>
For multiple recipients <sub>n</sub>	<pre> &lt;?xml version="1.0"?&gt; &lt;data&gt;   &lt;status&gt;200&lt;/status&gt;   &lt;description&gt;Success&lt;/description&gt;   &lt;messages&gt;     &lt;id&gt;&lt;trackid&gt;&lt;/id&gt;     ... </pre>



	<pre>&lt;id&gt;&lt;trackid n&gt;&lt;/id&gt; &lt;/messages&gt; &lt;/data&gt;</pre>
--	---

## 2.2 Receiving Messages

This is for incoming messages. sendQuick will POST to the Client API registered in sendQuick.

**Outgoing  
Webhook**

```
http://yourclientserverip/apipath/api.cgi
```

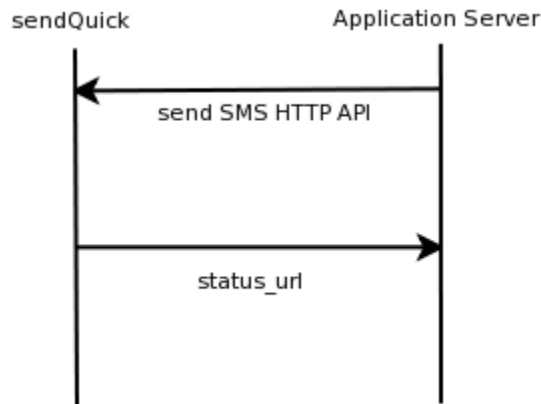
The receiving script should accept the following input parameters:

<i>Parameters</i>	<i>Description</i>	<i>Possible Value</i>
event	The user action.	'optin' or 'message'
id	The UUID of the user or group	123e4567-e89b-12d3-a456-

<i>Parameters</i>	<i>Description</i>	<i>Possible Value</i>
		426655440000
userid	The actual MIM user ID	CCZK2QX7T
name	The user profile or group name	My Profile
type	The user type	'Individual' or 'Group'
origin	The MIM type	FACEBOOK
route	The MIM routing name registered	Facebook_Bot
message	The content of the message.	Hello

## **2.3 Receive Outgoing Message Status**

This is for sendQuick server to call back client's status\_url to update the message status, after sending out SMS.



The status\_url should accept the following input parameters:

<i>Parameters</i>	<i>Data Type</i>	<i>Description</i>
trackid	string	This will be the trackid submitted by the AppServ for sQ to track the status of the outgoing SMS
status	string	This will be the status of the outgoing SMS. <ul style="list-style-type: none"> <li>• Y - Sent successful</li> <li>• F - Failed to send.</li> <li>• D - Message Delete from server</li> <li>• R - Received delivery report from recipient mobile phone.</li> </ul>
totalsms	integer	This will be the integer value of the total SMS used for sending the complete message. NOTE: Not applicable for status D and R
mno	string	The mobile number of the recipient

## 2.4 Attachment Support

- Attachment URL must be **publicly accessible**.

- Attachment must have the correct MIME type (e.g. audio/mpeg for audio).
- You may try other file type not specified here, but delivery will not be guaranteed

<b>Platform</b>	<b>Image</b>	<b>File</b>	<b>Audio</b>	<b>Video</b>
Facebook <small>F</small>	jpg,gif,png	All (< 25 MB)	wav (< 25 MB)	mp4 (< 25 MB)
Telegram	jpg,gif,png	pdf, zip (< 20 MB)	mp3 (< 20 MB)	mp4 (< 20 MB)
Line	jpg (< 1 MB)	Not supported	m4a (< 10 MB, length < 60s)	mp4 (< 10 MB, length < 60s)
Wechat <sup>w</sup>	jpg,gif,png (< 2 MB)	Not supported	mp3, amr (< 2 MB, length < 60s)	mp4 (< 10 MB)
Slack	jpg,gif,png	All (< 1 MB)	Send as File	Send as File
Webex Teams	jpg,gif,png,bmp	All (< 100 MB)	Send as File	Send as File
MS Teams	jpg,png	pdf	wav	mp4
Whatsapp DC	All <= 100MB jpg,png	All <= 100MB text/plain, text/csv, application/pdf, application/msword, application/x-tar, application/rtf.0, application/vnd.ms-powerpoint, application/vnd.openxmlformats-officedocument.presentationml.presentation, application/vnd.openxmlformats-officedocument.wordprocessingml.document, application/vnd.openxmlformats-officedocument.spreadsheetml.sheet, application/vnd.oasis.opendocument.presentation, application/vnd.oasis.opendocument.spreadsheet, application/vnd.ms-excel,	All <= 16MB audio/aac, audio/mp4, audio/amr, audio/mpeg, audio/ogg, audio/opus	All <= 16MB video/mp4 and audio/aac, video/3gp p

Platform	Image	File	Audio	Video
		application/vnd.oasis.opendocument.text		
Viber	jpeg, png or non-animated gif and file extension as the last path segment. Max image size: 1MB on iOS, 3MB on Android	Max size 50 MB. Refer “2.7 Viber Forbidden File Formats” for unsupported file types	Follow File section	mp4 or H264. Max size 26 MB and file extension as the last path segment

<sup>F</sup> Maximum resolution for images is 85 Megapixel

<sup>W</sup> Files will expire in 3 days

**Example:**

<http://192.168.1.101/api/message.php?action=send&id=6590000000&type=image&text=https://myapp.com/image.jpg>

## 2.5 Whatsapp Template Messages

The following parameters must be included in the API request:

1. template=<template name>
2. text=<{{1}},{{2}}, ... ,{{n}}>

The ‘text’ parameter accepts CSV format. Replace {{n}} with your desired text. Contact your account admin for the list of available template.

**Example:**

[http://192.168.1.101/api/message.php?action=send&id=6590000000&type=text&template=new\\_issue\\_alerts&text=Sendquick,01/06/2020,12:00,\"Issue Alert\"](http://192.168.1.101/api/message.php?action=send&id=6590000000&type=text&template=new_issue_alerts&text=Sendquick,01/06/2020,12:00,\)

## 2.6 Status Code List

<i>Status</i>	<i>Description</i>
200	Success
400	Missing request body / recipient id / text
403	Service disabled / Access deny / Authentication failed
404	Route / Template not found (Whatsapp DC)
405	Method (GET) not allowed
406	Request is not readable (XML and JSON)

## 2.7 Viber Forbidden File Formats

<b>Extension</b>	<b>Format</b>	<b>Operating system(s)</b>
ACTION	Automator Action	Mac OS
APK	Application	Android
APP	Executable	Mac OS
BAT	Batch File	Windows
BIN	Binary Executable	Windows, Mac OS, Linux
CMD	Command Script	Windows
COM	Command File	Windows
COMMAND	Terminal Command	Mac OS
CPL	Control Panel Extension	Windows
CSH	C Shell Script	Mac OS, Linux
EXE	Executable	Windows
GADGET	Windows Gadget	Windows
INF1	Setup Information File	Windows
INS	Internet Communication Settings	Windows
INX	InstallShield Compiled Script	Windows
IPA	Application	iOS
ISU	InstallShield Uninstaller Script	Windows
JOB	Windows Task Scheduler Job File	Windows
JSE	JScript Encoded File	Windows
KSH	Unix Korn Shell Script	Linux

<b>Extension</b>	<b>Format</b>	<b>Operating system(s)</b>
LNK	File Shortcut	Windows
MSC	Microsoft Common Console Document	Windows
MSI	Windows Installer Package	Windows
MSP	Windows Installer Patch	Windows
MST	Windows Installer Setup Transform File	Windows
OSX	Executable	Mac OS
OUT	Executable	Linux
PAF	Portable Application Installer File	Windows
PIF	Program Information File	Windows
PRG	Executable	GEM
PS1	Windows PowerShell Cmdlet	Windows
REG	Registry Data File	Windows
RGS	Registry Script	Windows
RUN	Executable	Linux
SCT	Windows Scriptlet	Windows
SHB	Windows Document Shortcut	Windows
SHS	Shell Scrap Object	Windows
U3P	U3 Smart Application	Windows
VB	VBScript File	Windows
VBE	VBScript Encoded Script	Windows
VBS	VBScript File	Windows
VBSCRIPT	Visual Basic Script	Windows
WORKFLOW	Automator Workflow	Mac OS
WS	Windows Script	Windows
WSF	Windows Script	Windows