



HTTP/API SPECIFICATION

TECHNICAL DOCUMENTATION

HTTP Application Programming Interface

Prepared by Olutola M Obembe

PRIMAX TECHNOLOGIES LIMITED

<http://www.smsbag.com>

<http://www.primaxng.com>

©2011 Mobile Messaging System



Contents

1. Introduction
2. HTTP Application Programming Interface
 - 2.1 Introduction
 - 2.2 Message Submission
 - 2.2.1 HTTP GET
 - 2.2.1.1 Query String Parameters
 - 2.2.1.2 Return Values
 - 2.2.2 HTTP XML POST
 - 2.2.2.1 XML
 - 2.2.2.2 Parameter Specifications
 - 2.2.2.3 Return XML
 - 2.2.2.4 Examples
 - 2.3 Delivery Reports
 - 2.3.1 PUSH METHOD
 - 2.3.2 PULL METHOD



1. Introduction

This document provides developers with instructions for integrating SMS messaging services into various solutions using smsBAG HTTP application programming interface (HTTP API). smsBAG HTTP API can be used for sending SMS messages, collecting delivery reports, receiving SMS messages, and LONG code solutions.

It describes smsBAG HTTP API methods, URLs and parameters needed as well as providing samples. The following methods are available

- Send messages using HTTP XML POST
- Send messages using HTTP GET
- Collection of Delivery Reports – via XML format



2. HTTP Application Programming Interface

2.1 Introduction

The smsBAG system offers various methods to send and receive SMS messages. This chapter contains specifications for the following methods.

- Send messages using HTTP XML POST: This method allows one to send SMS messages to a number of recipients using the XML-formatted data sent to a corresponding URL.
- Send messages using HTTP GET: Similar to the HTTP XML POST, this method allows sending messages passing parameters directly as query string variables via HTTP GET.
- Collection of Delivery Reports – This gives you the ability to collect XML formatted delivery reports from sent messages using either the push (HTTP POST) to a predefined call-back URL or the pull method by making HTTP GET request to a corresponding URL.

2.2 Message Submission

2.2.1 HTTP GET

The URL used to send messages via HTTP GET is

Data Centre 1: <http://api.smsbag.com/api/sendsms/plain>

Data Centre 2: <http://api2.smsbag.com/api/sendsms/plain>

Examples of normal SMS message

http://api.smsbag.com/api/sendsms/plain?user=user&password=pass&sender=sender&SMSText=New_Message&GSM=2348062741555



2.2.1.1 Query String Parameters

user	Client Username for smsBAG system
password	Client Password for smsBAG system
sender	Dynamic message Sender ID Alphanumeric : max 11 characters Numeric: max 14 xcharacters
SMSText	Message body (160 characters for 1 message)
GSM	Recipient GSM number in International format
IsFlash	Can be 0 or 1 0 sends a normal SMS 1 sends a Flash SMS
Type	Optional Parameter To send WAP bookmarks: value "bookmark" To send concatenated SMS: value "longSMS"
bookmark	WAP PUSH content Example: www.smsbag.com/pix.jpg
Binary	Binary content using hexadecimal format Example 410A0D4245 – Cannot be used together wit text parameter

2.2.1.2 Return Values

Value	Description
-2	NOT_ENOUGH_CREDITS
-3	NETWORK_NOT_COVERED
-5	INVALID_USER_OR_PASS
-6	MISSING_DESTINATION_ADDRESS
-7	MISSING_SMSTEXT
-8	MISSING_SENDERNAME
-9	DESTADDR_INVALIDFORMAT
-10	MISSING_USERNAME
-11	MISSING_PASS
-13	INVALID_DESTINATION_ADDRESS
>0	SUCCESSFUL



2.2.2 HTTP(S) XML POST

The URL used to post XML formatted data is

Data Centre 1: <http://api.smsbag.com/api/sendsms/xml>

Data Centre 2: <http://api2.smsbag.com/api/sendsms/xml>

2.2.2.1.1 XML

The XML formatted string must have XML=" at the beginning. There are two ways of formatting the XML String.

Without Registered Delivery	With Registered Delivery
<pre><SMS> <authentication> <username></username> <password></password> </authentication> <message> <sender></sender> <text></text> <flash></flash> <type></type> <wapurl></wapurl> <binary></binary> </message> <recipients> <gsm></gsm> <gsm></gsm> </recipients> </SMS></pre>	<pre><SMS> <authentication> <username></username> <password></password> </authentication> <message> <sender></sender> <text></text> <flash></flash> <type></type> <wapurl></wapurl> <binary></binary> </message> <recipients> <gsm messageId="msgiD1"></gsm> <gsm messageId="msgiD2"></gsm> </recipients> </SMS></pre>

As shown in the XML formats described above. XML formatted with delivery contains a different `<gsm>` tag which includes the `messageId` attribute. That is the main difference between these two formats and it means that when using XML formatted without registered delivery, it is possible to collect delivery reports from sent SMS messages but those reports will have `messageId` generated by the smsBAG system. Therefore connecting delivery reports with its SMS message will not be possible.



One the other hand, when using XML formatted with registered delivery, each delivery report will contain the **messageId** attribute with a value equal to the value of the **messageId** attribute defined by the client in **<gsm>** tags of every recipient in XML formatted with registered delivery. This is useful if the client wants to collect delivery reports for specific SMS messages – and it can be done by using the **messageId** of those messages.

2.2.2.2 Parameter Specifications

Authentication	username	Client Username for smsBAG system
	password	Client Password for smsBAG system
Message	sender	Dynamic message Sender ID Alphanumeric : max 11 characters Numeric: max 14 xcharacters
	text	Message body (160 characters for 1 message)
	flash	Can be 0 or 1 0 sends a normal SMS 1 sends a Flash SMS
	type	Optional Parameter To send WAP bookmarks: value "bookmark" To send concatenated SMS: value "longSMS"
	wapurl	WAP PUSH content Example: www.smsbag.com/pix.jpg
	binary	Binary content using hexadecimal format Example 410A0D4245 – Cannot be used together wit text parameter
Recipients	GSM	Message destination address, must be in international format without leading 0s or + E.g 2348062741555
	GSM messageId="msld"	Registered delivery – messageId set by client



2.2.2.3 Return XML

After the POST XML is initiated by the client, some status codes will be available. The return XML string format will be:

```
<RESPONSE>
  <status>status_code</status>
  <credits>credit_amount</credits>
</RESPONSE>
```

Status Codes

STATUS	VALUE	DESCRIPTION
AUTH_FAILED	-1	Invalid username and/or password
XML_ERROR	-2	Incorrect XML format
NOT_ENOUGH_CREDITS	-3	Not enough credits in user account
NO_RECIPIENTS	-4	No good recipients
GENERAL_ERROR	-5	Error in processing your request
SEND_OK	>0	Number of messages that will be sent

2.2.2.4 Example [PHP using CURL]

[<?php](#)

```
// smsBAG's POST URL
$postUrl = "http://api2.smsbag.com/api/sendsms/xml";
// XML-formatted data
$xmlString =
"<SMS>
<authentication>
  <username> </username>
  <password></password>
</authentication>
<message>
  <sender>Friend</sender>
  <text>Message from your friend!</text>
</message>
<recipients>
```




With registered Delivery

```
<gsm messageId="76090909092222">2348062741555</gsm>
```

```
<gsm messageId="176080808091111">2348065246516</gsm>
```

Or

Without Registered Delivery

```
<gsm >2348062741555</gsm>
```

```
<gsm >2348065246516</gsm>
```

```
</recipients>
```

```
</SMS>";
```

```
// previously formatted XML data becomes value of "XML" POST variable
```

```
$fields = "XML=" . urlencode($xmlString);
```

```
// in this example, POST request was made using PHP's CURL
```

```
$ch = curl_init();
```

```
curl_setopt($ch, CURLOPT_URL, $postUrl);
```

```
curl_setopt($ch, CURLOPT_POST, 1);
```

```
curl_setopt($ch, CURLOPT_POSTFIELDS, $fields);
```

```
// response of the POST request
```

```
$response = curl_exec($ch);
```

```
curl_close($ch);
```

```
// write out the response
```

```
echo $response;
```

```
?>
```

2.3 Delivery Reports

With this API method you can collect sent SMS delivery reports (by using one of two sending SMS messages methods described above). As soon as delivery reports for sent messages are received in the smsBAG system, they will be forwarded to you as an XML formatted string.

If you used a POST sending method with XML data formatted with registered delivery, each delivery report will have the same messageId attribute as the message for which it is being sent. If you used the POST method with XML data formatted without registered delivery or the GET method, the messageId attribute of collected delivery reports will be generated by the smsBAG system.



There are two methods of collecting delivery reports: PUSH and PULL.

2.3.1 PUSH METHOD

To be able to receive delivery report via the PUSH method, you will need to specify a delivery URL to us to configure.

If your delivery report URL is unavailable for any reason, another attempt to forward the delivery report will be made in 60 seconds, another in five minutes and subsequently every hour for the next 24 hours. If your URL is not available for the entire time, the delivery report will be lost.

The format of the XML delivery report structure will be:

```
<DeliveryReport>
<message id="msgId" sentdate="xxxx" donedate="xxxx" status="xxxx" />
<message id="msgId" sentdate="xxxx" donedate="xxxx" status="xxxx" />
</DeliveryReport>
```

XML Attributes Description

Attribute	Description	
id	Clients message ID	
sentdate	Date/Time when message was submitted from the client to the smsBAG system Format (yyyy/mm/dd hh:mm:ss)	
donedate	Date/Time when SMSC notified the smsBAG system of the delivery report Format (yyyy/mm/dd hh:mm:ss)	
status	NOT_SENT	The message could not be submitted to the Recipient Operator (possible reasons – Operator SMSC connection is down)
	SENT	The message was sent over a route that does not support delivery reports or delivery report not yet available
	NOT_DELIVERED	The message could not be delivered
	DELIVERED	The message was successfully delivered to the recipient



	NOT_ALLOWED	The client has no authorization to send to the specified network (the message will not be charged)
	INVALID_DESTINATION_ADDRESS	Invalid/Incorrect GSM recipient
	INVALID_SOURCE_ADDRESS	You have specified incorrect/invalid/not allowed source address (sender name)
	ROUTE_NOT_AVAILABLE	You are trying to use routing that is not available for your account
	NOT_ENOUGH_CREDITS	There are no available credits on your account to send the message
	INVALID_MESSAGE_FORMAT	Your message has invalid format

Example script for reading raw POST data sent to delivery report URL by PUSH method – for example, delivery URL may be <http://yourserver.com/dlreport.php> (PHP scripting language)

```
<?php
## read raw data
$postData = file_get_contents("php://input");
##extract XML structure from server
$dom = new DOMDocument();
$dom->loadXML ($postData);
##create new Xpath
$xpath = new domxpath($dom);
##query message XML element
$reports = $xpath->query("/DeliveryReport/message");
##write out attributes of each message element
foreach ($reports as $node) {
    echo "br>id: " . $node->getAttribute('id');
    echo "br>sentdate: " . $node->getAttribute('sentdate');
    echo "br>donedate: " . $node->getAttribute('donedate');
    echo "br>status: " . $node->getAttribute('status');
}
?>
```

©2011 Mobile Messaging System

<http://www.smsbag.com>
<http://www.primaxng.com>



2.3.2 PULL METHOD

The URL to get delivery reports over HTTP GET method is:

<http://api.smsbag.com/api/dlrpull?user=xxx&password=xxx>
<http://api2.smsbag.com/api/dlrpull?user=xxx&password=xxx>

Parameters

User

Password

MessageId – Optional

Return Values

5 – Invalid username and /or Password

10 – Missing Username

11 – Missing Password

The XML delivery report structure is the same as defined in PUSH method.

Examples of delivery report

```
<DeliveryReport>
```

```
<message id="162731295071200" sentdate="2011/1/15 18:40:3" donedate="2011/1/15 18:40:9"  
status="DELIVERED" />
```

```
<message id="553831295071200" sentdate="2011/1/15 18:40:3" donedate="2011/1/15 18:40:12"  
status="DELIVERED" />
```

```
</DeliveryReport>
```

If you have any issues using this document or the API, kindly contact us via

E: info@smsbag.com

T: 234-8062741555

Contact – Olutola Michael

©2011 Mobile Messaging System

<http://www.smsbag.com>

<http://www.primaxng.com>