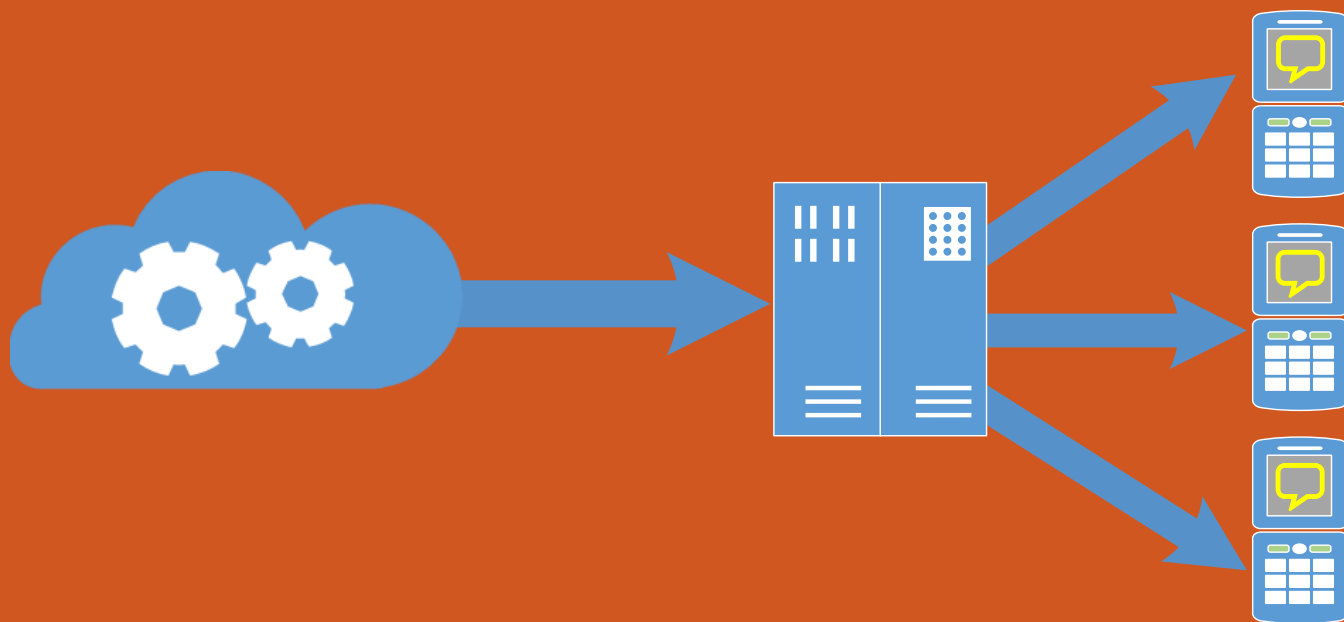


# SMS ILLICO HTTP API



## TECHNICAL DOCUMENTATION

Version 2.0 – Updated on July 25, 2016

## TABLE OF CONTENTS

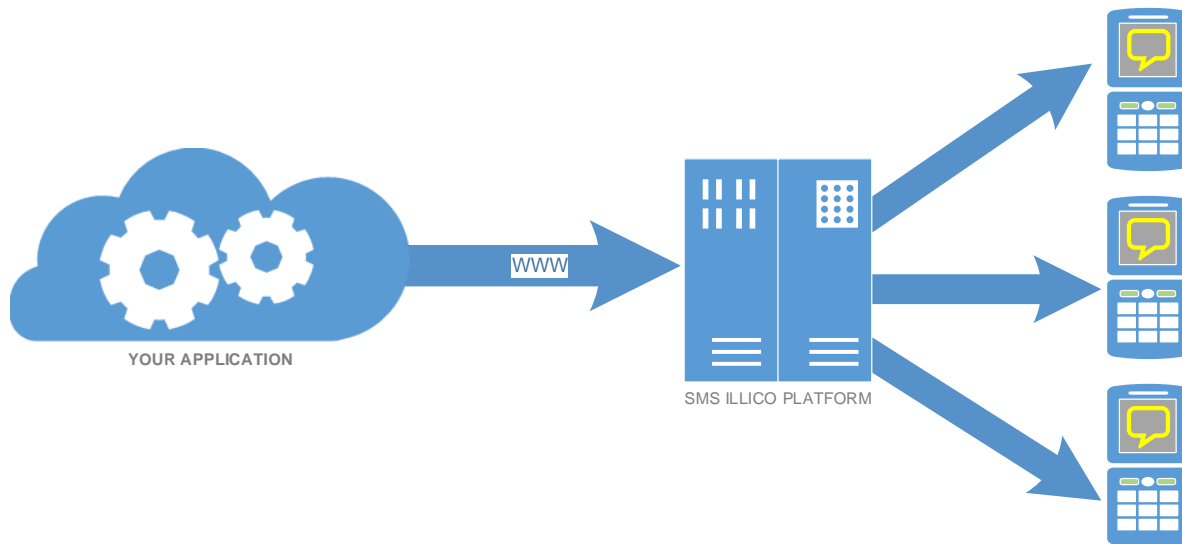
<b>INTRODUCTION</b>	<b>3</b>
<b>I. XML HTTP POST</b>	<b>4</b>
I.1. XML Format	4
I.2. Specification of the sending parameters	5
I.3. XML Response	5
I.4. Example	6
<b>II. JSON HTTP POST</b>	<b>9</b>
II.1. JSON Format	9
II.2. Specification of the sending parameters	10
II.3. JSON Response	10
II.4. Example	10
<b>III. HTTP GET</b>	<b>14</b>
III.1. Specification of the sending parameters	14
III.2. Response to the requests	14
<b>CONTACT US</b>	<b>15</b>

## INTRODUCTION

This document provides developers with the commands enabling them to integrate sending SMS services by using the SMS illico HTTP API. The SMS illico HTTP API can be used to send SMS or to receive delivery reports (acknowledgement of receipt/non-reception, NPAI...etc.).

The SMS illico system provides three sending methods:

1. XML through the HTTP (POST) protocol
2. JSON through the HTTP (POST) protocol
3. Parameters sent to URL via HTTP (GET)



In what follows we will see how to proceed to the sending of SMS by using one and the other of these two methods.

Finally, we will see how to retrieve the delivery reports (acknowledgement of receipt /non-reception, NPAI...etc.).

## I. XML HTTP POST

The XML (HTTP POST) must be sent to the following URL:

<http://smsillico-ao.com/api/sendsms/xml>

### I.1. XML Format

The XML format string must start with: « XML= ». You have then the choice between two structures of data:

Without identification of the sending	Without identification of the sending
<pre> &lt;SMS&gt;   &lt;authentication&gt;     &lt;username&gt;&lt;/username&gt;     &lt;password&gt;&lt;/password&gt;   &lt;/authentication&gt;   &lt;message&gt;     &lt;sender&gt;&lt;/sender&gt;     &lt;text&gt;&lt;/text&gt;   &lt;/message&gt;   &lt;recipients&gt;     &lt;gsm&gt;&lt;/gsm&gt;     &lt;gsm&gt;&lt;/gsm&gt;     &lt;gsm&gt;&lt;/gsm&gt;     &lt;gsm&gt;&lt;/gsm&gt;   &lt;/recipients&gt; &lt;/SMS&gt;                     </pre>	<pre> &lt;SMS&gt;   &lt;authentication&gt;     &lt;username&gt;&lt;/username&gt;     &lt;password&gt;&lt;/password&gt;   &lt;/authentication&gt;   &lt;message&gt;     &lt;sender&gt;&lt;/sender&gt;     &lt;text&gt;&lt;/text&gt;   &lt;/message&gt;   &lt;recipients&gt;     &lt;gsm messageId="msgID1"&gt;&lt;/gsm&gt;     &lt;gsm messageId="msgID2"&gt;&lt;/gsm&gt;     &lt;gsm messageId="msgID3"&gt;&lt;/gsm&gt;     &lt;gsm messageId="msgID4"&gt;&lt;/gsm&gt;   &lt;/recipients&gt; &lt;/SMS&gt;                     </pre>

As described above, you can define a specific identifier for each SMS sent, and this, thanks to the attribute **messageID**. This attribute makes it possible in particular to trace a group of SMS thanks to the delivery reports (acknowledgement of receipt of non-reception, NPAI...etc.). in fact, among the elements collected in these reports, we find the **messageID**, which will enable you to have statistical elements by bulk sending (by operation). **messageID** is an optional attribute.

## I.2. Specification of the sending parameters

AUTHENTICATION	<b>username</b>	Client login (communicated after signing up on <a href="http://www.smsillico-ao.com">www.smsillico-ao.com</a> )
	<b>password</b>	Client password (communicated after signing up on <a href="http://www.smsillico-ao.com">www.smsillico-ao.com</a> )
MESSAGE	<b>sender</b>	Either short code (« 3333 ») or customized sender (ex : « smsillicoID », 11 alphanumeric characters maximum, must be validated beforehand by our services)
	<b>text</b>	Message text (160 alphanumeric characters maximum for one SMS)
RECIPIENTS	<b>gsm</b>	Angola mobile phone number (ex : 123 456 789 / 999 888 777 /...)
	<b>messageid</b>	(Optional) SMS sending identifier, defined by the customer (retrievable in the delivery report)

The creation of new « Sender » must be done via our online service: <http://www.smsillico-ao.com>. The validation of the sender by our service can take up to 24hrs and 48hrs

## I.3. XML Response

After the POST XML and if all the checking is correct, the client receives in return an XML in the following format:

Without identification of the sending	Without identification of the sending
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;results&gt;   &lt;result&gt;     &lt;status&gt;0&lt;/status&gt;      &lt;messageid&gt;Smsillico_Messageid&lt;     /messageid&gt;     &lt;destination&gt;&lt;/destination&gt;   &lt;/result&gt;   &lt;result&gt;     &lt;status&gt;1&lt;/status&gt;      &lt;messageid&gt;Smsillico_Messageid&lt;     /messageid&gt;     &lt;destination&gt;&lt;/destination&gt;   &lt;/result&gt; &lt;/results&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;results&gt;   &lt;result&gt;     &lt;status&gt;0&lt;/status&gt;     &lt;messageid&gt;msgID1&lt;/messageid&gt;     &lt;destination&gt;&lt;/destination&gt;   &lt;/result&gt;   &lt;result&gt;     &lt;status&gt;0&lt;/status&gt;     &lt;messageid&gt;msgID2&lt;/messageid&gt;     &lt;destination&gt;&lt;/destination&gt;   &lt;/result&gt; &lt;/results&gt;</pre>

## I.4. Example

POST HTTP request using an XML structure without identification of the sending

< ?php

```
// smsillico POST URL
```

```
$postUrl = "http://api.smsillico-ao.com//sendsms/xml";
```

```
// XML structure of Data
```

```
$xmlString = "<SMS>
```

```
    <authentication>
```

```
        <username>xxxx</username>
```

```
        <password>xxxxxx</password>
```

```
    </authentication>
```

```
    <message>
```

```
        <sender>Le Président</sender>
```

```
        <text>Salut à tous !</text>
```

```
    </message>
```

```
    <recipients>
```

```
        <gsm>07060504</gsm>
```

```
        <gsm>05040302</gsm>
```

```
        <gsm>01020304</gsm>
```

```
        <gsm>66676869</gsm>
```

```
    </recipients>
```

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

```
// Insertion of the POST "XML" variable name before the data in XML format
```

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

```
// In this example, the POST request is completed thanks to the Curl library
```

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

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

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

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

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

```
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: application/xml));
```

```
// Response of the POST request
```

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

```

curl_close($ch);

// Writing of the response

echo $response;

?>
    
```

POST HTTP request using an XML structure with identification of the sending:

```

< ?php

// smsillico POST URL

$postUrl = "http://api.smsillico-ao.com/sendsms/xml";

// XML structure of Data

$xmlString = "<SMS>
                <authentication>
                    <username>xxxx</username>
                    <password>xxxxxx</password>
                </authentication>
                <message>
                    <sender>Le Président</sender>
                    <text>Salut à tous !</text>
                </message>
                <recipients>
                    <gsm messageId="1000">07060504</gsm>
                    <gsm messageId="1001">05040302</gsm>
                    <gsm messageId="1002">01020304</gsm>
                    <gsm messageId="1000">66676869</gsm>
                </recipients>
            </SMS>";
    
```

// Insertion of the POST "XML" variable name before the data in XML format

```

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

// In this example, the POST request is completed thanks to the Curl library

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $postUrl);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $jsonString);
    
```

```
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type:
application/xml));

// Response of the POST request

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

// Writing of the response

echo $response;

?>
```



## II. JSON HTTP POST

### II.1. JSON Format

The string in JSON format is as follows. You have then the choice between two structures of data:

Without identification of the sending	Without identification of the sending
<pre>{   "authentication": {     "username": "test",     "password": "test"   },   "messages": [     {       "sender": "Sender",       "text": "Hello",       "recipients": [         {           "gsm": "385951111111"         },         {           "gsm": "385952222222"         },         {           "gsm": "385953333333"         }       ]     }   ] }</pre>	<pre>{   "authentication": {     "username": "test",     "password": "test"   },   "messages": [     {       "sender": "Sender",       "text": "Hello",       "recipients": [         {           "gsm": "385951111111",           "messageID": "clientsmsgID"         },         {           "gsm": "385952222222",           "messageID": "clientsmsgID"         }       ],       {         "gsm": "385953333333",         "messageID": "clientsmsgID"       }     }   ] }</pre>

As described above, you can define a specific identifier for each SMS sent, and this, thanks to the attribute **messageID**. This attribute makes it possible in particular to trace a group of SMS thanks to the delivery reports (acknowledgement of receipt of non-reception, NPAI...etc.). in fact, among the elements collected in these reports, we find the **messageID**, which will enable you to have statistical elements by bulk sending (by operation). **messageID** is an optional attribute.

## II.2. Specification of the sending parameters

AUTHENTICATION	username	Client login (communicated after signing up on <a href="http://www.smsillico-ao.com">www.smsillico-ao.com</a> )
	password	Client password (communicated after signing up on <a href="http://www.smsillico-ao.com">www.smsillico-ao.com</a> )
MESSAGE	sender	Either short code (« 3333 ») or customized sender (ex : « smsillicoID », 11 alphanumeric characters maximum, must be validated beforehand by our services)
	text	Message text (160 alphanumeric characters maximum for one SMS)
RECIPIENTS	gsm	Angola mobile phone number (ex : 123 456 789 / 999 888 777 /...)
	messageid	(Optional) SMS sending identifier, defined by the customer (retrievable in the delivery report)

The creation of new « Sender » must be done via our online service: <http://www.smsillico-ao.com>. The validation of the sender by our service can take up to 24hrs and 48hrs

## II.3. JSON Response

After the POST JSON and if all the checkings are correct, the client receives in return an JSON in the following format:

Without identification of the sending	Without identification of the sending
<pre>{   "results": [     {       "status": "0",       "destination": "385951111111"     }   ] }</pre>	<pre>{   "results": [     {       "status": "0",       "messageid": "10210011344550330860",       "destination": "385951111111"     }   ] }</pre>

## II.4. Example

POST HTTP request using a JSON structure without identification of the sending

< ?php

// smsillico POST URL

```
$postUrl = "http://api.smsillico-ao.com/sendsms/json";
```

```
//JSON structure of Data
```

```
$jsonString = '{
    "authentication": {
        "username": "xxxxxxx",
        "password": "xxxxx"
    },
    "message": [
        {
            "sender": "SMSILLICO",
            "text": "Test SMS",
            "recipients": [
                {
                    "gsm": "07060504"
                },
                {
                    "gsm": "05040302"
                },
                {
                    "gsm": "66676869"
                }
            ]
        }
    ]
}';
```

```
// Insertion of the POST "JSON" variable name before the data in JSON format
```

```
$fields = urlencode($jsonString);
```

```
// In this example, the POST request is completed thanks to the Curl library
```

```
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $postUrl);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $jsonString);
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type:
application/json'));
```

```
// Response of the POST request
```

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

```

curl_close($ch);

// Writing of the response

echo $response;

?>
    
```

POST HTTP request using a JSON structure with identification of the sending:

```

< ?php

// smsillico POST URL

$postUrl = "http://api.smsillico-ao.com/sendsms/json";

//JSON structure of Data

$jsonString = '{
    "authentication": {
        "username": "xxxxxxx",
        "password": "xxxxx"
    },
    "message": [
        {
            "sender": "SMSILLICO",
            "text": "Test SMS",
            "recipients": [
                {
                    "gsm": "07060504",
                    "messageld": "1000"
                },
                {
                    "gsm": "05040302",
                    "messageld": "1001"
                },
                {
                    "gsm": "66676869",
                    "messageld": "1002"
                }
            ]
        }
    ]
}';

// Insertion of the POST "JSON" variable name before the data in JSON format

$fields = urlencode($xmlString);
    
```

// In this example, the POST request is completed thanks to the Curl library

```
$ch = curl_init();  
curl_setopt($ch, CURLOPT_URL, $postUrl);  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);  
curl_setopt($ch, CURLOPT_POST, true);  
curl_setopt($ch, CURLOPT_POSTFIELDS, $jsonString);  
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type:  
application/json'));
```

// Response of the POST request

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

// Writing of the response

```
echo $response;
```

?>

### III. HTTP GET

HTTP GET requests will have to be sent to the URL : <http://api.smsillico-ao.com/sendsms/plain>

An example of SMS sent via an HTTP GET request: <http://api.smsillico-ao.com/sendsms/plain?username=XXX&password=XXXXX&sender=SMSILLICO&text=Hello&recipients=07060504;66676869>

#### III.1. Specification of the sending parameters

<b>username</b>	Client login (communicated after signing up on <a href="http://www.smsillico-ao.com">www.smsillico-ao.com</a> )
<b>password</b>	Client password (communicated after signing up on <a href="http://www.smsillico-ao.com">www.smsillico-ao.com</a> )
<b>sender</b>	Either short code (« 3333 ») or customized sender (ex : « smsillicoID », 11 alphanumeric characters maximum, must be validated beforehand by our services)
<b>text</b>	Message text (160 alphanumeric characters maximum for one SMS)
<b>recipients</b>	Angola mobile phone number (ex : 123 456 789 / 999 888 777 / ...)

#### III.2. Response to the requests

After sending of the HTTP GET request, the customer receives instantaneously a return message relating to the sending or a flow (XML, JSON):

STATUS	DESCRIPTION
<b>ACCOUNT_ERROR</b>	Authentication error (login/password)
<b>ACCOUNT_DISABLED</b>	Deactivated account (contact support)
<b>EMPTY_RECIPIENTS</b>	No recipient in the request
<b>NOT_ENOUGH_CREDITS</b>	Insufficient credit
<b>SENDER_NOT_ALLOWED</b>	Sender name not authorized
<b>INVALID_DESTINATION_ADDRESS</b>	Error in recipients numbers
<b>GENERAL_ERROR</b>	Sending error , multiple causes

## CONTACT US

### SMS ILLICO

E-mail : [support@smsillico-ao.com](mailto:support@smsillico-ao.com)

Telephone: +244 914 858 593 / +244 926 400 467

[www.smsillico-ao.com](http://www.smsillico-ao.com)

### SMS ILLICO is powered by LOGICMIND

Rua do Club Hípicos, Condomínio Classicos do Sul

Luanda - Angola

[info@logicmind.net](mailto:info@logicmind.net)

[www.logicmind.net](http://www.logicmind.net)