Web Services Integration

SOAP Gateway

Version 2.0

Table of Contents

Introduction Page 3

Technical Platform

Request to send SMS Page 3

Example of PHP request Page 5

Sample SOAP requests and responses Page 6

Possible error codes Page 9

Appendix A: Receipt notifications Page 10

Appendix B: Set of GSM7 characters Page 11

Introduction

SOAP Gateway platform allows users to send messages through a standard protocol using HTTP or HTTPS. This way, many messages may be sent in a single request. To access your statistics and invoicing data, you can enter the website http://plusmms.net with your user log-in.

This documentation describes the required and optional parameters to use all possibilities for sending SMS messages following the SOAP specifications.

TECHNICAL PLATFORM

Request to send SMS

The available configuration options, the URL that should be called, and the parameters supported will be detailed below.

The service description can be obtained from the URL:

http://gateway.plusmms.net/soap/api.wsdl

Possible parameters:

- **username**: User provided by the company.
- **password:** Password provided by the company
- **text:** message of the text. At most you can have 160 characters if you do not specify that the message is multi-part (see 'parts' parameter). The text must be encoded in UTF-8
- **to:** mobile phone number of the message recipient. You must include the prefix (e.g.: In Spain 34666666666). This field allows you to specify multiple recipients, for all numbers in an array must be included.
- **from**: Text Sender, this label will consist of 15 numbers or 11 alphanumeric .
- **coding (optional):** Possible values are "gsm" and "utf-16". The value "gsm" for normal deliveries with GSM7 coding and 160 characters per message and the value "utf-16" for UCS2 coding (UTF16) and 70 characters per message. If not specified, the default value is "gsm"
- **fSend (optional):** Date message sent. If you need to send scheduled messages, the delivery date can be specified indicating the date in the

following format YYYYmmddHHiiss indicating the date format (e.g.: 20130215142000 would be February 15, 2013 at 14:20). If sending is immediate, this parameter does not have to be specified.

- **parts (optional):** Indicates the maximum number of parts in which the message to be sent will be divided. This variable is set to 1 by default, so if it is not specified and a message over 160 characters for encoding "gsm" is sent, the message will fail. Keep in mind that concatenated messages can only be 153 characters per party and each part is billed as one sending. The server will only use the minimum necessary parts for sending text even if the specified number of parts is greater than necessary. If the number of parts is less than that required for sending the text, the sending will fail with error 105.
- **trsec (optional)**: Boolean type. With the "false" value the server does not change any character in the message; this is the default value. With the value "true", server handles to modify the common invalid characters in GSM7 to valid characters with the following translation table: 'á' => 'a', 'í'=>'i', 'ó'=>'o', 'ú'=>'u', 'ç'=>'Ç', 'Á'=>'A', 'Í'=>'I', 'Ó'=>'O', 'Ú'=>'U', 'À'=>'A', 'È'=>'E', 'Ì'=>'I', 'Ò'=>'O', 'Ù'=>'U', 'º' => '', 'ª' => '', 'Õ' => 'O', 'õ' => 'o', 'â' => 'a', 'ê' => 'e', 'î'=>'i', 'ô'=>'o', 'û'=>'u', 'Â'=>'A', 'Ê'=>'E', 'Î'=>'I', 'Ô'=>'O', 'Û'=>'U', 'ã' => 'a', 'Ã' => 'A'

-

Example of PHP request:

```
<?php
$client = new SoapClient("http://gateway.plusmms.net/soap/api.wsdl");

$message['username'] = 'myuser';
$message['password'] = 'mypass';
$message['to'] = array('34666555444');
$message['text'] = 'Test con eñe €uro';
$message['from'] = 'test';

//Optional parameters, uncoment to use
//$message['fsend'] = '20150215142000';
//$message['coding'] = 'gsm';
//$message['dlrurl'] = 'http://server.com/not.php?rem=%p&tel=%P&stat=%d;
//$message['parts'] = 1;
//$message['trsec'] = true;

try {
   $response = $client->sendMsg($message);
   if ($response->errorCode===0) {
      echo "Remote server has accepted the message.";
   } else {
      echo "Message rejected: ".$response->errorDescription;
   }
} catch (SoapFault $exception) {
   echo $exception;
}
?>
```

The password (password) and the client code (username) will be provided by the company. It should be mentioned that in order to increase system security, the customer must specify the IP from where it will connect; only sending from the IP specified by the customer will be allowed.

Example of SOAP requests and responses

Examples of request to a single recipient:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ns1="http://gateway.plusmms.net/soap/">
    <SOAP-ENV:Body>
        <ns1:sendMsg>
            <username>myuser</username>
            <password>mypass</password>
            <to>34666555444</to>
            <text>Test message</text>
            <from>MSG</from>
        </ns1:sendMsg>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example of an answer to a single recipient

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ns1="http://gateway.plusmms.net/soap/">
    <SOAP-ENV:Body>
        <ns1:sendMsgResponse>
            <sendResult>
                <to>34927690739</to>
                <accepted>true</accepted>
                <id>104650056</id>
            </sendResult>
            <errorCode>0</errorCode>
            <errorDescription>Accepted for delivery</errorDescription>
        </ns1:sendMsgResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example of a request to multiple recipients:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
                   xmlns:ns1="http://gateway.plusmms.net/soap/">
                   <SOAP-ENV:Body>
                        <ns1:sendMsg>
                             <username>myuser</username>
                             <password>mypass</password>
                             <to>34666555444</to>
                             <to>34666555445</to>
                             <text>Test message</text>
                             <from>MSG</from>
                        </ns1:sendMsg>
                   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example of an answer to multiple recipients:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:ns1="http://gateway.plusmms.net/soap/">
   <SOAP-ENV:Body>
        <ns1:sendMsgResponse>
             <sendResult>
                  <to>34927690739</to>
                  <accepted>true</accepted>
                  <id>104650108</id>
             </sendResult>
             <sendResult>
                  <to>34927690734</to>
                  <accepted>true</accepted>
                  <id>104650109</id>
             </sendResult>
             <errorCode>0</errorCode>
             <errorDescription>Accepted for delivery</errorDescription>
        </ns1:sendMsgResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example of an answer to multiple recipients with errors in a recipient:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:ns1="http://gateway.plusmms.net/soap/">
 <SOAP-ENV:Body>
        <ns1:sendMsgResponse>
                <sendResult>
                        <to>34927690739</to>
                        <accepted>true</accepted>
                        <id>104652299</id>
                        <error />
                </sendResult>
                <sendResult>
                        <to>34927690734a</to>
                        <accepted>false</accepted>
                        <id />
                        <error>No valid recipients</error>
                </sendResult>
                <errorCode>0</errorCode>
                <errorDescription>Accepted for delivery</errorDescription>
        </ns1:sendMsgResponse>
 </SOAP-ENV:Body>
              </SOAP-ENV:Envelope>
```

Example of an answer to multiple recipients with errors in a recipient:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ns1="http://gateway.plusmms.net/soap/">
    <SOAP-ENV:Body>
        <ns1:sendMsgResponse>
                <sendResult />
                <errorCode>113</errorCode>
                <errorDescription>Invalid coding</errorDescription>
        </ns1:sendMsgResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Possible error codes:

0 => "Accepted for delivery",
101 => "Internal Database error",
102 => "No valid recipients",
103 => "Username or password unknown",
104 => "Text message missing",
105 => "Text message too long",
106 => "Sender missing",
107 => "Sender too long",
108 => "No valid Date time for send",
109 => "Notification URL incorrect",
110 => "Exceeded maximum parts allowed or incorrect number of parts",
111 => "Not enough credits",
112 => "IP address not allowed",
113 => "Invalid coding"

Appendix A: Receipt notifications

If you wish to receive receipt notifications in real time, you must specify the variable "dlr-url" with the URL of the client where you want to be notified of the sending status.

The operation consists in specifying the URL where you want to make a request to our server for each http request when a notification from the operator is received. To do this the customer must have an http server capable of receiving such notifications.

Our server will send the variables by the GET method as the customer wants. To do that in the URL that you send us, you have to put the variable name followed by an escape character that will contain the value, the escape characters have the form of the "%" character followed by a letter. This would be a URL example:

http://mi.server.com/notifica.php?remitente=%p&tel=%P&estado=%d

These are the defined escape characters:

NRS %i identifier that was delivered when the sending was done

%d value of the receipt notification

%p the sender of the SMS

%P the phone number of the SMS receiver

%t date of sending the message to "YYYY-MM-DD HH: MM" format, eg, "1999-09-21 14:18"

The %d value will return the final state of sending to us, the possible values are:

- 1: The message has been delivered to the recipient.
- 2: The message could not be delivered to the recipient.
- 4: The message was delivered to the SMSC, it is an intermediate notification, not an end result
- 16: It could not be delivered to the end operator

To better explain the process, an example of how the sending of an sms and the receipt notification would be is given below.

First, we send the sms with the dlr-url variable to indicate the URL where we want to receive the delivery notification. We will add to this URL our dispatch identifier to identify it unambiguously when we receive it. The final url for the notification would be: http://mi.server.com/notifica.php?idenvio=7584remitente=%p&tel=%P&estado=%d

Appendix B: Set of GSM7 characters

Basic set of characters

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70
0x00	@	Δ	SP	0	¡	P	¿	p
0x01	£	_	!	1	A	Q	a	q
0x02	$	Φ	"	2	B	R	b	r
0x03	¥	Γ	#	3	C	S	c	s
0x04	è	Λ	¤	4	D	T	d	t
0x05	é	Ω	%	5	E	U	e	u
0x06	ù	Π	&	6	F	V	f	v
0x07	ì	Ψ	'	7	G	W	g	w
0x08	ò	Σ	(	8	H	X	h	x
0x09	Ç	Θ	)	9	I	Y	i	y
0x0A	LF	Ξ	*	:	J	Z	j	z
0x0B	Ø	ESC	+	;	K	Ä	k	ä
0x0C	ø	Æ	,	<	L	Ö	l	ö
0x0D	CR	æ	-	=	M	Ñ	m	ñ
0x0E	Å	ß	.	>	N	Ü	n	ü
0x0F	å	É	/	?	O	§	o	à

Extension of the basic character set, these characters occupy two positions

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70
0x00					|			
0x01								
0x02								
0x03								
0x04		^						
0x05							€	
0x06								
0x07								
0x08			{					
0x09			}					
0x0A	FF							
0x0B		SS2						
0x0C				[				
0x0D	CR2			~				
0x0E				]				
0x0F			\					