

*Wau***SMS**

Integration HTTP GET

Version 2.0

Index

Introduction	Pag 3
Technical platform	Pag 4
Request to send SMS	Pag 4
Responses to HTTP requests	Pag 5
Appendix A: Receipt notifications	Pag 6
Appendix B: Set of GSM7 characters	Pag 7

Introduction

The http server platform allows users to send messages through this platform. To access your statistics and invoicing data you Access to the website **<http://dashboard.wausms.local>** with your user log-in.

Communication with the client will not be through any API provided by the company, as it will simply be HTTP communication, with some parameters to the indicated URL. This process is detailed below.

TECHNICAL PLATFORM

Request to send SMS

To send SMS through the platform, the client should send a request to the website above, using certain parameters (text to send, phone number, recipient, sender, etc.), and we will send a response confirmation if everything worked, or an error response if it didn't

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

To create the URL, the client must make a GET call to the following address:
<http://dashboard.wausms.local/Api/get/send.php>

Going through a series of parameters:

- **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 (Eg: Spain 34666666666). This field allows you to specify multiple recipients; this requires separate phone numbers by space (Eg 34666666666 34666666667 34666666668)
- **from:** Text Sender, this label will consist of 15 numbers or 11 alphanumeric.
- **username:** User provided by the company.
- **password:** Password of client in **WauSMS**.
- **coding:** The possible values are "gsm", "gsm-pt" and "utf-16". The default value is "gsm". The maximum number of characters for normal messages is 160 for the GSM7 encoding, 155 for the GSM-PT encoding and 70 for the UCS2 encoding (UTF16). The maximum number of characters for concatenated messages is 155 for the GSM7 encoding, 149 for the GSM-PT encoding and 67 for the UCS2 encoding (UTF16).
- **dlr-mask(optional):** To receive delivery confirmations always leave at 8, if it is not needed this parameter does not have to be specified.
- **fSend (optional):** Date and time the message was sent. If you need to send scheduled messages, the date can be specified in the format YYYYmmddHHiiSS (eg: 20130215142000 would be February 15th, 2013 at 14:20 UTC). The date must be specified in UTC time (GMT + 0). Shipments can not be scheduled later than 30 days. In case of an immediate sending, 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 modified and a message over 160 characters for encoding 0 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 of parts for sending text even if the specified number of parts is bigger than necessary. If the number of parts is less than the required for sending the text, the sending will fail with error 105.
- **trsec (optional):** Possible values are 1 and 0. With the 0 value the server does not change any character in the message, this is the default value. With the value 1, server modifies the common invalid characters in GSM7 into valid characters according to 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", "Û"=>"U", "o"=>"", "a"=>"", "Ö"=>"O", "ö"=>"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".
- **reference (optional):** Sending reference. If not specified, a reference will be generated automatically each month with the following nomenclature: API_SMS_yyyy_mm where yyyy is the current year and mm the current month.
- **tags (optional):** Tags separated by space. Eg: tag1 tag2

Call example:

```
http://dashboard.wausms.local/Api/get/send.php?username=tu_user&password=tu_pass&to=3465467445+34627455334+34627545454&text= Prueba+de+envio&from=TEST&coding=0&dlr-mask=8
```

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 client must specify the IP from where it will connect, only sending from the IP specified by the client will be allowed.

As could be seen, the response returned by the platform against a SMS sending request has the following syntax:

0: Accepted for delivery. ID 53337

In case of error, a description of the reason for the error will be returned.

Responses to HTTP requests

The syntax of the response has the following format:

{error code}: {descriptive message}. ID {sending identifier}

The identifier is only delivered if the request has succeeded. Response examples would be: Response examples would be:

0: Accepted for delivery. ID 53337

105: Text message missing.

The possible answers are:

Code	Description
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 Datetime for send
109	Notification URL incorrect
110	Exceeded maximum parts allowed or incorrect number of parts
111	Not enough credits

Appendix A: Receipt notifications

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

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 client must have an http server able to receive such notifications.

Our server will send the variables by the GET method as the client 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:

- **%i** Identifier of **WauSMS** 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 message sending in "YYYY-MM-DD HH: MM" format, eg, "1999-09-21 14:18".
- **%s** status (REJECTD, DELIVRD, EXPIRED, DELETED, UNDELIV, ACCEPTD, UNKNOWN, RECEIVED).
- **%y** DLR date of the message with "YYYY-MM-DD HH:MM" format, ex. "2020-09-21 14:19".
- **%n** part number (concatenated messages).

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

- **1:** Message is delivered to destination
- **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 ending operator

To better explain the process, an example of how the sending of an sms and the receipt notification will happen is provided below.

Firstly, send the sms with the dlr-url variable to indicate the URL where you want to receive the sending notification. We will add this URL to our dispatch identifier to identify it clearly when we receive it. The final url for the notification would be: **http://mi.server.com/notifica.php?idenvio=123&remitente=%p&tel=%P&estado=%d**

Therefore the final call we should do to send the SMS would be:

```
http://dashboard.wausms.local/Api/get/send.php?username=tu_user&password=tu_login&to=34666555444+3466655533+34666555222&text=Prueba+de+envio&from=TEST&coding=0&dlr-mask=8&dlr-url=http%3A%2F%2Fmi.server.com%2Fnotifica.php%3Fidenvio%3D7584%26remitente%3D%25p%26tel%3D%25P%26estado%3D%25d
```

Assuming that all messages could be delivered, three requests with the state = 1, sender = TEST, ID sending = 7584, and the matching phone number will be delivered to the notification.php script

Appendix B: Set of GSM7 characters

Basic set of characters

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70
0x00	@	?	SP	0	i	P	ı	p
0x01	£	_	!	1	A	Q	a	q
0x02	\$?	"	2	B	B	b	r
0x03	¥	?	#	3	C	S	c	s
0x04	è	?	¤	4	D	T	d	t
0x05	é	?	%	5	E	U	e	u
0x06	ù	?	?	6	F	V	f	v
0x07	ì	?	'	7	G	W	g	w
0x08	ò	?	(8	H	X	h	x
0x09	Ç	?)	9	I	Y	i	y
0x0A	LF	?	*	:	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							€	

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70
0x06								
0x07								
0x08			}					
0x09			{					
0x0A	FF							
0x0B		SS2						
0x0C				[
0x0D	CR2			~				
0x0E]				
0x0F			\					