



Globitel HTTP API SPECIFICATION

INTRODUCTION

This document is intended for users who want to integrate our SMS services with their application or site.

To be able to use the HTTP API, please register to our site for a web account and contact us by email to enable the HTTP functionality for your account. Include your username in the email.

The HTTP API is a simple type of communication with our gateway. It can be used either in the form of a HTTP POST or GET.

You can check your available credits and see sms reports from HTTP API from our web interface:
<http://www.smsbts.com/clients/login.php>

SEND A SMS

The commands are made up of two segments. The first is for authentication and the second is the parameters needed for each command. Some parameters are mandatory and some others are optional.

In order to send an SMS, our system needs to authenticate you as a valid user. You must provide your username and password (of HTTP account) as parameters to the command. The other two mandatory parameters are the "text" which is the sms text that you want to send and the "to" which is the mobile number(s) that you want to send sms to.

GET METHOD

If you use **GET** method to connect with us, you can send a sms up to 20 recipients with one command. When you call this command you must wait for the server response and then you can call again the command.

The parameters for sending a sms are:

- **User:** your username for your HTTP account.
- **Password:** your password for your HTTP account.
- **Text:** the content of the sms. It must be URL encoded.
- **To:** this parameter specifies the recipient's phone number. You can send one sms with one command to max 20 recipients. If there is more than one mobile number they must be separated by commas. The mobile numbers must be in international format, with country code followed by number, without leading

zero and special characters like +. For example the following number in the UK 07901231234 will become 447901231234.

- **From:** this parameter specifies the sender name that will appear on the mobile. It can be either a valid international format number between 1 and 16 characters long or an eleven length alphanumeric string. Different lengths may be supported by different networks.
- **Coding:** the coding of the text. Valid values are 0, 1, and 2. 0 is the default value (if unset) for 7 bit encoding (for Greek 160 characters see APPENDIX C). 1 is for 8 bit encoding and 2 is for Unicode encoding (UCS-2 encoding). For example some Arabic chars will be in this format: %06%33%06%28%00%20%06%27%06%2A. If Unicode encoding is used you can send up to 69 chars (for further information see APPENDIX D).
- **deliv_ack:** set this parameter to 1 if you want to have delivery reports for this sms or 0 if you don't need delivery reports. Default value is 1 if you omit this parameter.

Example 1

Suppose we have an HTTP account with username: test and password test123.

```
http://www.smsbts.com/httpapi/sendmsg.php?user=test&password=test123&to=30694799234&text=This+is+a+simple+sms&from=John
```

This simple example shows how to send a sms from "test" user account to mobile number 30694799234. The text of the sms is "This is a simple sms" and the sender is "John".

Server Response:

ID: 41794000ce82c585a10677d9583a987ebefce7e0 To: 30694799234

That means that everything is ok and the sms was accepted by our gateway. The server returns an id for every accepted sms. This is useful later if we want to ask for the status of this sms, we will use this id.

If something goes wrong the server will return an error code and a short description like this:

ERR: 001, Authentication failed

That means that an error occurred and the sms will not be delivered. The 001 error code means that the system cannot authenticate the user with the given username and password (for further information see APPENDIX B).

Example 2

Suppose we have an HTTP account with username: test and password test123.

We want to send the same sms with Arabic chars to 5 mobile numbers.

```
http://www.smsbts.com/httpapi/sendmsg.php?user=test&password=test123&to=30694799234,30699123456,30693345789,30697345567,30694345876&text=%06%33%06%28%00%20%06%27%06%2A&from=TESTSMS&coding=2&deliv_ack=0
```

This simple example shows how to send a sms from "test" user account to multiple mobile numbers. To use Unicode encoding we must set coding=2. We have set deliv_ack=0 so we will not have delivery reports for these sms.

POST METHOD

You can use post method in order to connect with our service. The difference between **GET** and **POST** methods is in the way of sending data to the page, while the GET method sends data using URL, the POST method sends them through the standard entrance. Also notice that although get method can contain until **2 Kbytes** per call, if you use **POST** method, the maximum size of the post content must not exceeded **32 Kbytes**. The constraint is that in post method you have to pass the parameters with the exact following names:

- **User:** your username for your HTTP account.
- **Password:** your password for your HTTP account.
- **Text:** the content of the sms. It must be URL encoded.
- **To:** this parameter specifies the recipient's phone number. If there is more than one mobile number they must be separated by commas. The mobile numbers must be in international format, with country code followed by number, without leading zero and special characters like +. For example the following number in the UK 07901231234 will become 447901231234.
- **From:** this parameter specifies the sender name that will appear on the mobile. It can be either a valid international format number between 1 and 16 characters long or an eleven length alphanumeric string. Different lengths may be supported by different networks.
- **Coding:** the coding of the text. Valid values are 0, 1, and 2. 0 is the default value (if unset) for 7 bit encoding (for Greek 160 characters see APPENDIX C). 1 is for 8 bit encoding and 2 is for Unicode encoding (UCS-2 encoding). For example some Arabic chars will be in this format: %06%33%06%28%00%20%06%27%06%2A. If Unicode encoding is used you can send up to 69 chars (for further information see APPENDIX D).

deliv_ack: set this parameter to 1 if you want to have delivery reports for this sms or 0 if you don't need delivery reports. Default value is 1 if you omit this parameter.

Server Response:

ID: 41794000ce82c585a10677d9583a987ebefceie0 To: 30694799234

ID: 4sfkllafj4385ygh8543ojgfa98435439085435e7e4 To: 30699123456

ID: 347658yreg87fdg687687df6g8d7g68d7g687i6dd To: 30693345789

ID: 34o954fidgjoifdsfg98dsf7g985679543fd98gh987d To: 30697345567

ID: 56toitrjoigjdoi5965409joigj854u76eirut98truyu8try To: 30694345876

That means that everything is ok and the sms was accepted by our gateway to be delivered to all the recipients.

Query for the status of an SMS

With this command you can query the status of a sent sms. You need to specify your username and password and the id of the sms that you want to have the status. You shouldn't query many times for the same sms. You can store the status locally at your system.

So to query the status of the sms we sent at example 1, we use the following command:

```
http://www.smsbts.com/httpapi/querymsg.php?user=test&password=test123&apimsgid=41794000ce82c585a10677d9583a987ebefce7e0
```

Server Response:

apimsgid: 41794000ce82c585a10677d9583a987ebefce7e0 status 1

That means that this sms have status 1, which means delivered to the mobile.

SMS Status Callback

You can have the status of a sent sms as soon as it is updated on our server. This is more powerful because you don't have to ask for every sms its status and you get the status directly. In order to enable this function for your account you must give us a URL to where we will send the status and the id of the sms.

For example your URL can be like this:

```
www.yoursite.com/smsapi/getsmsstatus.php
```

When our server updates the status to "1" of a sms sent by your account, the following URL will be called:

```
www.yoursite.com/smsapi/getsmsstatus.php?id=41794000ce82c585a10677d9583a987ebefce7e0&status=1
```

So you can get the status of the sms and store it to your database (for further information see APPENDIX A).

Query for available credits

With this command you can query the available credits that you have at your HTTP account.

Example

```
http://www.smsbts.com/httpapi/getCredits.php?user=test&password=test123
```

Server Response:

1513

You have 1513 credits left to your HTTP account.

APPENDIX A: Status Codes

The following table lists the status codes that our gateway returns to query commands.

0	Message unknown. The apimsgid you provide is not valid
---	--

1	Delivery success. The sms is delivered to the mobile
2	Delivery failure. An error occurred delivering the sms to the mobile
4	SMS buffered. SMS has been queued at our gateway for delivery at a later time
8	Delivery success. The sms is delivered to the operator *
16	SMS rejected. There was an error routing the sms

Table 1

*Some operators don't send always status 1 when the sms is being delivered to the mobile. Thus you can have status 8 while the sms has been delivered to the mobile.

APPENDIX B: Error Codes

The following table lists the error messages are generated by our gateway. There is no charge if these errors are generated when sending a sms.

001	Authentication failed
002	Invalid Query
003	Empty Query
301	No credits left
303	To many receivers for one post
401	Empty receiver number
402-403	Invalid format of receiver number
404	Max length is 160 for non Unicode sms
405	Max length is 69 for Unicode sms
501	Unknown destination number
631-650	System error, please contact us

Table 2

APPENDIX C: Info about Greek Capital characters (160) message format

In order to send a Greek capital letters message you have to encode your text message following the steps below

1. Find out the **hex code** for the Greek capital characters that differ to Latin capital characters at **Table 3** below (red rows).

Hex	Dec	Character name	Character	ISO-8859-1 DEC
0x00	0	COMMERCIAL AT	@	64
0x01	1	POUND SIGN	£	163
0x02	2	DOLLAR SIGN	\$	36
0x03	3	YEN SIGN	¥	165
0x04	4	LATIN SMALL LETTER E WITH GRAVE	è	232
0x05	5	LATIN SMALL LETTER E WITH ACUTE	é	233
0x06	6	LATIN SMALL LETTER U WITH GRAVE	ù	249
0x07	7	LATIN SMALL LETTER I WITH GRAVE	ì	236
0x08	8	LATIN SMALL LETTER O WITH GRAVE	ò	242
0x09	9	LATIN CAPITAL LETTER C WITH CEDILLA	Ç	199
0x0A	10	LINE FEED		10
0x0B	11	LATIN CAPITAL LETTER O WITH STROKE	Ø	216
0x0C	12	LATIN SMALL LETTER O WITH STROKE	ø	248
0x0D	13	CARRIAGE RETURN		13
0x0E	14	LATIN CAPITAL LETTER A WITH RING ABOVE	Å	197
0x0F	15	LATIN SMALL LETTER A WITH RING ABOVE	å	229
0x10	16	GREEK CAPITAL LETTER DELTA	Δ	
0x11	17	LOW LINE	_	95
0x12	18	GREEK CAPITAL LETTER PHI	Φ	
0x13	19	GREEK CAPITAL LETTER GAMMA	Γ	
0x14	20	GREEK CAPITAL LETTER LAMBDA	Λ	
0x15	21	GREEK CAPITAL LETTER OMEGA	Ω	
0x16	22	GREEK CAPITAL LETTER PI	Π	
0x17	23	GREEK CAPITAL LETTER PSI	Ψ	
0x18	24	GREEK CAPITAL LETTER SIGMA	Σ	
0x19	25	GREEK CAPITAL LETTER THETA	Θ	
0x1A	26	GREEK CAPITAL LETTER XI	Ξ	
0x1B	27	ESCAPE TO EXTENSION TABLE		
0x1B0A	27 10	FORM FEED		12
0x1B14	27 20	CIRCUMFLEX ACCENT	^	94
0x1B28	27 40	LEFT CURLY BRACKET	{	123
0x1B29	27 41	RIGHT CURLY BRACKET	}	125

0x1B2F	27 47	REVERSE SOLIDUS (BACKSLASH)	\	92
0x1B3C	27 60	LEFT SQUARE BRACKET	[91
0x1B3D	27 61	TILDE	~	126
0x1B3E	27 62	RIGHT SQUARE BRACKET]	93
0x1B40	27 64	VERTICAL BAR		124
0x1B65	27 101	EURO SIGN	€	164 (ISO-8859-15)
0x1C	28	LATIN CAPITAL LETTER AE	Æ	198
0x1D	29	LATIN SMALL LETTER AE	æ	230
0x1E	30	LATIN SMALL LETTER SHARP S (German)	ß	223
0x1F	31	LATIN CAPITAL LETTER E WITH ACUTE	É	201
0x20	32	SPACE		32
0x21	33	EXCLAMATION MARK	!	33
0x22	34	QUOTATION MARK	"	34
0x23	35	NUMBER SIGN	#	35
0x24	36	CURRENCY SIGN	¤	164 (ISO-8859-1)
0x25	37	PERCENT SIGN	%	37
0x26	38	AMPERSAND	&	38
0x27	39	APOSTROPHE	'	39
0x28	40	LEFT PARENTHESIS	(40
0x29	41	RIGHT PARENTHESIS)	41
0x2A	42	ASTERISK	*	42
0x2B	43	PLUS SIGN	+	43
0x2C	44	COMMA	,	44
0x2D	45	HYPHEN-MINUS	-	45
0x2E	46	FULL STOP	.	46
0x2F	47	SOLIDUS (SLASH)	/	47
0x30	48	DIGIT ZERO	0	48
0x31	49	DIGIT ONE	1	49
0x32	50	DIGIT TWO	2	50
0x33	51	DIGIT THREE	3	51
0x34	52	DIGIT FOUR	4	52
0x35	53	DIGIT FIVE	5	53
0x36	54	DIGIT SIX	6	54
0x37	55	DIGIT SEVEN	7	55
0x38	56	DIGIT EIGHT	8	56
0x39	57	DIGIT NINE	9	57
0x3A	58	COLON	:	58
0x3B	59	SEMICOLON	;	59
0x3C	60	LESS-THAN SIGN	<	60
0x3D	61	EQUALS SIGN	=	61
0x3E	62	GREATER-THAN SIGN	>	62
0x3F	63	QUESTION MARK	?	63
0x40	64	INVERTED EXCLAMATION MARK	¡	161
0x41	65	LATIN CAPITAL LETTER A	A	65

0x42	66	LATIN CAPITAL LETTER B	B	66
0x43	67	LATIN CAPITAL LETTER C	C	67
0x44	68	LATIN CAPITAL LETTER D	D	68
0x45	69	LATIN CAPITAL LETTER E	E	69
0x46	70	LATIN CAPITAL LETTER F	F	70
0x47	71	LATIN CAPITAL LETTER G	G	71
0x48	72	LATIN CAPITAL LETTER H	H	72
0x49	73	LATIN CAPITAL LETTER I	I	73
0x4A	74	LATIN CAPITAL LETTER J	J	74
0x4B	75	LATIN CAPITAL LETTER K	K	75
0x4C	76	LATIN CAPITAL LETTER L	L	76
0x4D	77	LATIN CAPITAL LETTER M	M	77
0x4E	78	LATIN CAPITAL LETTER N	N	78
0x4F	79	LATIN CAPITAL LETTER O	O	79
0x50	80	LATIN CAPITAL LETTER P	P	80
0x51	81	LATIN CAPITAL LETTER Q	Q	81
0x52	82	LATIN CAPITAL LETTER R	R	82
0x53	83	LATIN CAPITAL LETTER S	S	83
0x54	84	LATIN CAPITAL LETTER T	T	84
0x55	85	LATIN CAPITAL LETTER U	U	85
0x56	86	LATIN CAPITAL LETTER V	V	86
0x57	87	LATIN CAPITAL LETTER W	W	87
0x58	88	LATIN CAPITAL LETTER X	X	88
0x59	89	LATIN CAPITAL LETTER Y	Y	89
0x5A	90	LATIN CAPITAL LETTER Z	Z	90
0x5B	91	LATIN CAPITAL LETTER A WITH DIAERESIS	Ä	196
0x5C	92	LATIN CAPITAL LETTER O WITH DIAERESIS	Ö	214
0x5D	93	LATIN CAPITAL LETTER N WITH TILDE	Ñ	209
0x5E	94	LATIN CAPITAL LETTER U WITH DIAERESIS	Ü	220
0x5F	95	SECTION SIGN	§	167
0x60	96	INVERTED QUESTION MARK	¿	191
0x61	97	LATIN SMALL LETTER A	a	97
0x62	98	LATIN SMALL LETTER B	b	98
0x63	99	LATIN SMALL LETTER C	c	99
0x64	100	LATIN SMALL LETTER D	d	100
0x65	101	LATIN SMALL LETTER E	e	101
0x66	102	LATIN SMALL LETTER F	f	102
0x67	103	LATIN SMALL LETTER G	g	103
0x68	104	LATIN SMALL LETTER H	h	104
0x69	105	LATIN SMALL LETTER I	i	105
0x6A	106	LATIN SMALL LETTER J	j	106
0x6B	107	LATIN SMALL LETTER K	k	107
0x6C	108	LATIN SMALL LETTER L	l	108
0x6D	109	LATIN SMALL LETTER M	m	109

0x6E	110	LATIN SMALL LETTER N	n	110
0x6F	111	LATIN SMALL LETTER O	o	111
0x70	112	LATIN SMALL LETTER P	p	112
0x71	113	LATIN SMALL LETTER Q	q	113
0x72	114	LATIN SMALL LETTER R	r	114
0x73	115	LATIN SMALL LETTER S	s	115
0x74	116	LATIN SMALL LETTER T	t	116
0x75	117	LATIN SMALL LETTER U	u	117
0x76	118	LATIN SMALL LETTER V	v	118
0x77	119	LATIN SMALL LETTER W	w	119
0x78	120	LATIN SMALL LETTER X	x	120
0x79	121	LATIN SMALL LETTER Y	y	121
0x7A	122	LATIN SMALL LETTER Z	z	122
0x7B	123	LATIN SMALL LETTER A WITH DIAERESIS	ä	228
0x7C	124	LATIN SMALL LETTER O WITH DIAERESIS	ö	246
0x7D	125	LATIN SMALL LETTER N WITH TILDE	ñ	241
0x7E	126	LATIN SMALL LETTER U WITH DIAERESIS	ü	252
0x7F	127	LATIN SMALL LETTER A WITH GRAVE	à	224

Table 3

2. In the table above, every character contains the hex code of it (hex column). For example the Greek capital character Λ (Greek capital letter LAMBDA) has the hex code: $0x14$
3. In general the hex code has the format $x_1x_2x_3x_4$. The appropriate encoding/format of a character that our http API accepts follows the pattern $\%x_3x_4$. For example the Greek character Λ (Greek capital letter LAMBDA) must be formatted as follows: $\%14$
4. Example for Capital characters Greek message:

For example suppose we have an HTTP account with username: *test* and password: *test123*. We want to send the Greek word “*ΕΛΛΗΝΙΚΟ*” (you can see the corresponding characters in *Table 3*) to 5 mobile numbers. The URL must be formatted as follows:

```
http://www.smsbts.com/httpapi/sendmsg.php?user=test&password=test123&to=30694799234,30699123456,30693345789,30697345567,30694345876&text=E%14%14HNIKO&from=TESTSMS
```

APPENDIX D: Info about Unicode message format

In order to send a Unicode sms using our http API you have to set the *coding parameter* to 2 and encode your text message following the steps below.

1. Find out the Unicode Alphabet script page you need for your language, in the Unicode.org web site (<http://www.unicode.org/charts/>). For example for the Greek language you follow the link [Greek](#) and you will come with this the script page <http://www.unicode.org/charts/PDF/U0370.pdf>.
2. In the Unicode chart page every table cell contains a Unicode character and below the hex code of it. For example the Greek character α (small alpha) has the hex code: $03B1$

- In general the hex code has the format $x_1x_2x_3x_4$. The appropriate encoding/format of a character that our http API accepts follows the pattern $\%x_1x_2\%x_3x_4$. For example the Greek character α (small alpha) must be formatted as follows: $\%03\%B1$
- Example for Greek Unicode message:

For example suppose we have an HTTP account with username: *test* and password: *test123*. We want to send the Greek word “ $\gamma\rho\alpha$ ” (you can see the corresponding characters in *Table 4*) to 5 mobile numbers. The URL must be formatted as follows:

```
http://www.smsbts.com/httpapi/sendmsg.php?user=test&password=test123&to=30694799234,30699123456,30693345789,30697345567,30694345876&text=%03%B3%03%B1%03%C4%03%B1&from=TESTSMS&coding=2&deliv_ack=0
```

α 03B1	ρ 03C1
β 03B2	ς 03C2
γ 03B3	σ 03C3
δ 03B4	τ 03C4

Table 4

Notes:

- Generally this is UCS2 that uses BigEndianUnicode Encoding.
- You may need to write some custom code in the programming language of your preference in order to achieve the correct encoding/formatting.