SMS API

ver. 2.0, Sep 1, 2018.

This document describes application interface (API) between SMS service provider (SP) and SMS gateway (SMSGW).

Bulk SMS

Bulk SMS

Your Bulk API key: YOUR_API_KEY

Text2Reach API has simple HTTP GET request interface and also JSON-RPC interface. To use JSON-RPC interface, please read the JSON-RPC Specification (http://www.jsonrpc.org/specification).

1. Send SMS

1.1 HTTP interface

To send a bulk message, SP must make a HTTP GET request to address: *https://api.text2reach.com/sms/send* query string values are specified below in the parameters table. **All parameters should be properly url-encoded.** The returned HTML body for the request will be a message ID or an error code.

Example:
https://api.text2reach.com/sms/send?api_key=YOUR_API_KEY&phone=37126378162&from=DEMO&
message=Hello+world

1.2 JSON-RPC interface

To send a bulk message, SP must make a JSON-RPC request to address:
https://api.text2reach.com/sms/?api_key=YOUR_API_KEY

Method:

```
send( from, phone, message, type, unicode, timestamp, report_url, expires, schedule, blacklist
```

Example:

```
<?php
    require_once 'jsonRPCClient.php';
    define("API_KEY", "YOUR_API_KEY");

    $j = new jsonRPCClient("https://api.text2reach.com/sms/?api_key=" . API_KEY);
    $msg_id = $j->send(
            'DEMO',
            '37126378162',
            'Hello world!',
            'txt',
            false,
            0,
            'http://www.yoursite.com/sms/report/handler/',
            0,
            0,
            false
    );
    // save the $msg_id in DB and use it for report requests
?>
```

1.3 Parameters

Name	Required	Default value	Description
api_key	yes	-	Your registered Bulk API key
from	yes	-	source address of the message
phone	yes	-	destination address in international form
message	yes	-	plain text message up to 160 (1071) characters, Unicode message up to 70 (469) characters
type	no	txt	message type ("txt" for text, "bin" for binary)
unicode	no	true	if set to false, characters not in GSM 7bit alphabet will be converted (ā => a) or removed
timestamp	no	0	Unix timestamp of a time when end user opted in to receive messages. If parameter is not set the current timestamp is used.
report_url	no	false	URL for delivery reports
expires	no	0	Delivery time in seconds
schedule	no	-	Unix timestamp of a time when the message is scheduled for delivery
blacklist	no	false	Checks if the number is in blacklist before sending. Returns error -503 if found
shortlink	no	0	Set 1 if you need to curtail web-links in message by Bitly service. Links must begins with "http://"

1.4 Description

This bulk API call will send the text message to the provided phone number. Your account will be checked for the required funds. If the message cannot be sent, the response will be a negative integer representing the error.

The message will be considered as plain if all the characters are present in the GSM 7bit alphabet (see the table below). It is possible to send up to 1071 characters which will be split into seven parts, 153 characters per part. You will be charged for each part accordingly. 7 characters out of 160 are reserved to identify and concatenate multiple parts and applies only to messages having more than one part. If the message contains unicode characters then you can send up to 469 characters, 63 characters per part, max 7 parts. 7 characters out of 63 are reserved to identify and concatenate multiple parts and applies only to messages having more than one part.

The mandatory parameter “from” can be a telephone number in international format, a short code or a textual sender name. Maximum length of an international number is 15 digits not including “+” or “00” prefix. Maximum length of a textual sender name is 11 alphanumeric characters.

The optional parameter “type” specifies the format of the parameter “message”. In case the type is set to “bin”, parameter “message” should be in hexadecimal format.

The optional parameter “unicode” specifies the encoding used for parameter “message”. If it is set to “false”, then the message will be checked against GSM 7bit alphabet (see the table below) and all the symbols not in the table will be converted to the respective latin characters or completely removed. If the parameter “unicode” is set to “true” and the message contains characters not in the GSM 7bit alphabet, then the message will be considered as utf-8 and reduce the length of the message to 70 (469) characters.

The optional parameter “timestamp” specifies the time when the user opted in to receive the message. If an end user

has signed up for your service and has provided a mobile phone number to receive advertisements or informational messages, then you should provide “timestamp” parameter with the sign-up time. If an owner of a mobile phone number is changed, number is closed or operator is changed, then the message will not be sent. If the new owner of a number opts in to receive messages, a new timestamp should be used.

The “report_url” parameter should be used to receive delivery reports for sent messages. Please see below for detailed request specification.

If the “expire” parameter is not specified, the operator default expire time is used. If specified and the message is not delivered in the specified amount of seconds, you will received "expired" status. Exact behavior depends on the end user mobile phone operator validity period of the message and is not guaranteed to match the expire time specified in the parameter.

The “schedule” parameter should be set if you want to delay the delivery of the message. The parameter must be in two weeks range from the current time.

2. Manual delivery reports

You can request a status of the sent message by an api call.

2.1 HTTP interface

To request a status, SP must make a HTTP GET request to address: *https://api.text2reach.com/sms/status* query string values are specified below in the parameters table. The returned HTTP body will contain a status of the message.

Example:
https://api.text2reach.com/sms/status?api_key=YOUR_API_KEY&msg_id=1234567

2.2 JSON-RPC interface

To request a status, SP must make a JSON-RPC request to address:
https://api.text2reach.com/sms/?api_key=YOUR_API_KEY

Method:

```
status( msg_id )
```

Example:

```
<?php
    require_once 'jsonRPCClient.php';
    define("API_KEY", "YOUR_API_KEY");

    $j = new jsonRPCClient("https://api.text2reach.com/sms/?api_key=" . API_KEY);
    $status = $j->status(1234567);

    // save the status in DB for the msg_id=1234567
?>
```

2.3 Parameters

Name	Required	Default value	Description
api_key	yes	-	your registered Bulk API key
msg_id	yes	-	message ID from “bulk send” api call

3. Automatic delivery reports

Alternatively reports will be delivered to the “report_url” if it is provided in the send bulk api call. Delivery interface will be the same as used in the sending, e.g., if you used json-rpc interface to send a message, then the delivery report will be sent through json-rpc interface, too.

3.1 HTTP interface

SMSGW makes a HTTP GET request to a “report_url” address provided by SP with the parameters found below. The SP handler script must respond with text “OK” or else the report will be retried for a total of 10 times with 2n minutes interval (n – retries counter) and then discarded.

3.2 JSON-RPC interface

To handle report requests through JSON-RPC interface, you must have the minimum implementation as provided below:

```
<?php
    require_once 'jsonRPCServer.php';

    class handler {
            public function report($msg_id, $status, $retries) {
                    // handle the report
                    return true;
            }
    }

    $server = new handler();
    jsonRPCServer::handle($server) or die('Invalid request');
?>
```

3.3 Parameters

Name	Description
msg_id	message id of the message this report belongs to, provided as a result of sending the message
status	status of the sent message, • **delivered** – the message was delivered successfully • **undelivered** – the message was not delivered • **expired** – the delivery reached time limit and was considered undelivered • **canceled** – the message was canceled before scheduled sendout • **rejected** – the message was rejected by SMSGW and was not charged • **pending** – the message is not jet delivered (this status is only reported using manual delivery checking) • **unknown** – message status is not known
retries	how many times SMSGW tried to deliver report to the “report_url”

4. Extended delivery report

You can request a details of the sent message by an api call.

4.1 HTTP interface

To request a details, SP must make a HTTP GET request to address: *https://api.text2reach.com/sms/details* query

string values are specified below in the parameters table. The returned HTTP body will contain a status of the message.

Example:
https://api.text2reach.com/sms/details?api_key=YOUR_API_KEY&msg_id=1234567

4.2 JSON-RPC interface

To request a detailed status, SP must make a JSON-RPC request to address:
https://api.text2reach.com/sms/?api_key=YOUR_API_KEY

Method:

```
details( msg_id )
```

Example:

```
<?php
    require_once 'jsonRPCClient.php';
    define("API_KEY", "YOUR_API_KEY");

    $j = new jsonRPCClient("https://api.text2reach.com/sms/?api_key=" . API_KEY);
    $status = $j->status(1234567);

    // save the status in DB for the msg_id=1234567
?>
```

4.3 Parameters

Name	Required	Default value	Description
api_key	yes	-	your registered Bulk API key
msg_id	yes	-	message ID from “bulk send” api call

4.4 Returns

Error:

```
{
    "success" : 0,
    "message" : "Unknown msg_id",
    "data" : []
}
```

Successfull:

```
{
    "success" : 1,
    "message" : "ok",
    "data" : {
        "msg_id" : "1234567",
        "source" : "info",  // from number/alpha
        "destination" : "37129123456", // to number
        "country" : {
            "name" : "Latvia",
            "iso" : "lv",
            "mcc" : "247"
        },
        "operator" : {
            "name":"Best Mobile Operator",
            "mnc":"10"
        },
        "created" : "2017-01-01 10:00:00",
        "delivered" : "2017-01-01 10:00:03",
        "schedule" : "N",
        "scheduled" : null,
        "message" : "Hello World!",
        "multipart" : "N",
        "parts" : 1,
        "price: : 0.0232,
        "sum" : 0.0232,
        "status" : "delivered" // statuses: delivered | undelivered | expired | canceled |
    }
}
```

5. Cancel SMS

You can cancel pending/scheduled message by an api call.

5.1 HTTP interface

To cancel message sending, SP must make a HTTP GET request to address: *https://api.text2reach.com/sms/cancel* query string values are specified below in the parameters table. The returned HTTP body will contain a status of the message.

Example:
https://api.text2reach.com/sms/cancel?api_key=YOUR_API_KEY&msg_id=1234567

5.2 JSON-RPC interface

To cancel message sending, SP must make a JSON-RPC request to address:
https://api.text2reach.com/sms/?api_key=YOUR_API_KEY

Method:

```
cancel( msg_id )
```

Example:

```
<?php
    require_once 'jsonRPCClient.php';
    define("API_KEY", "YOUR_API_KEY");

    $j = new jsonRPCClient("https://api.text2reach.com/sms/?api_key=" . API_KEY);
    $result = $j->cancel(1234567);

    // result = ok | forbidden | error | unknown
?>
```

5.3 Parameters

Name	Required	Default value	Description
api_key	yes	-	your registered Bulk API key
msg_id	yes	-	message ID from “bulk send” api call

6. Credits balance

Prepaid customers can request remains of credits on account.

6.1 HTTP interface

To request a balance, SP must make a HTTP GET request to address: *https://api.text2reach.com/sms/credit* The returned HTTP body will contain a amount of credits.

Example:
https://api.text2reach.com/sms/credit?api_key=YOUR_API_KEY

6.2 JSON-RPC interface

To request a status, SP must make a JSON-RPC request to address:
https://api.text2reach.com/sms/?api_key=YOUR_API_KEY

Method:

```
credit()
```

Example:

```
<?php
    require_once 'jsonRPCClient.php';
    define("API_KEY", "YOUR_API_KEY");

    $j = new jsonRPCClient("https://api.text2reach.com/sms/?api_key=" . API_KEY);
    $prices = $j->credit();
?>
```

6.3 Parameters

Name	Required	Default value	Description
api_key	yes	-	your registered Bulk API key

7. Prices

7.1 HTTP interface

To get all available prices by countries, SP must make a HTTP GET request to address: https://api.text2reach.com /sms/prices query string values are specified below in the parameters table. The returned HTTP body will contain an json structure.

Example:
https://api.text2reach.com/sms/prices?api_key=YOUR_API_KEY

7.2 JSON-RPC interface

To get all available prices by countries, SP must make a JSON-RPC request to address:
https://api.text2reach.com/sms/?api_key=YOUR_API_KEY

Method:

```
prices()
```

Example:

```
<?php
    require_once 'jsonRPCClient.php';
    define("API_KEY", "YOUR_API_KEY");

    $j = new jsonRPCClient("https://api.text2reach.com/sms/?api_key=" . API_KEY);
    $prices = $j->prices();
?>
```

7.3 Parameters

Name	Required	Default value	Description
api_key	yes	-	your registered Bulk API key

7.4 Returns

Example:

```
[
    ...
    {
        country : "Belgium",
        iso     : "be",
        code    : 32,
        operator: "Proximus",
        mcc     : 206,
        mnc     : 1,
        type    : "all",
        price   : 0.123456,
        note    : null
    },
    ...
]
```

Error codes

If the message cannot be sent, the return value for msg_id will contain a negative integer which represents an error from the table below:

Value	Description
-10	Geneal system error
-11	Wrong API KEY for the request
-12	Wrong Message ID
-14	Wrong source address
-15	Wrong destination address
-16	Wrong “type”, must be “txt” or “bin”
-17	Wrong message length (empty)
-18	Wrong message length (too long)
-19	Wrong "schedule" value
-20	Wrong "expires" value
-21	Phone in blacklist
-22	No route destination
-34	Message failed
-35	Client undefined

GSM 7bit characters

Character	Unicode	UTF-8	Character	Unicode	UTF-8	Character	Unicode	UTF-8
@	0040	40	+	002B	2B	V	0056	56
£	00A3	C2A3	,	002C	2C	W	0057	57
$	0024	24	-	002D	2D	X	0058	58
¥	00A5	C2A5	.	002E	2E	Y	0059	59
è	00E8	C3A8	/	002F	2F	Z	005A	5A
é	00E9	C3A9	0	0030	30	Ä	00C4	C384
ù	00F9	C3B9	1	0031	31	Ö	00D6	C396
ì	00EC	C3AC	2	0032	32	Ñ	00D1	C391
ò	00F2	C3B2	3	0033	33	Ü	00DC	C39C
Ç	00C7	C387	4	0034	34	§	00A7	C2A7
	000A	0A	5	0035	35	¿	00BF	C2BF

Character	Unicode	UTF-8	Character	Unicode	UTF-8	Character	Unicode	UTF-8
Ø	00D8	C398	6	0036	36	a	0061	61
ø	00F8	C3B8	7	0037	37	b	0062	62
	000D	0D	8	0038	38	c	0063	63
Å	00C5	C385	9	0039	39	d	0064	64
å	00E5	C3A5	:	003A	3A	e	0065	65
Δ	0394	CE94	;	003B	3B	f	0066	66
_	005F	5F	<	003C	3C	g	0067	67
Φ	03A6	CEA6	=	003D	3D	h	0068	68
Γ	0393	CE93	>	003E	3E	i	0069	69
Λ	039B	CE9B	?	003F	3F	j	006A	6A
Ω	03A9	CEA9	¡	00A1	C2A1	k	006B	6B
Π	03A0	CEA0	A	0041	41	l	006C	6C
Ψ	03A8	CEA8	B	0042	42	m	006D	6D
Σ	03A3	CEA3	C	0043	43	n	006E	6E
Θ	0398	CE98	D	0044	44	o	006F	6F
Ξ	039E	CE9E	E	0045	45	p	0070	70
	00A0	C2A0	F	0046	46	q	0071	71
Æ	00C6	C386	G	0047	47	r	0072	72
æ	00E6	C3A6	H	0048	48	s	0073	73
ß	00DF	C39F	I	0049	49	t	0074	74
É	00C9	C389	J	004A	4A	u	0075	75
	0020	20	K	004B	4B	v	0076	76
!	0021	21	L	004C	4C	w	0077	77
"	0022	22	M	004D	4D	x	0078	78
#	0023	23	N	004E	4E	y	0079	79
¤	00A4	C2A4	O	004F	4F	z	007A	7A
%	0025	25	P	0050	50	ä	00E4	C3A4
&	0026	26	Q	0051	51	ö	00F6	C3B6
'	0027	27	R	0052	52	ñ	00F1	C3B1

Character	Unicode	UTF-8	Character	Unicode	UTF-8	Character	Unicode	UTF-8
(	0028	28	S	0053	53	ü	00FC	C3BC
)	0029	29	T	0054	54	à	00E0	C3A0
*	002A	2A	U	0055	55			
[1]	000C	0C	\ [1]	005C	5C	| [1]	007C	7C
^ [1]	005E	5E	[[1]	005B	5B	€ [1]	20AC	E282AC
{ [1]	007B	7B	~ [1]	007E	7E			
} [1]	007D	7D	] [1]	005D	5D			

[1] - GSM 7bit single part message may contain up to 160 7bit septets. Marked characters use two septets in the message.