

SMSify

HTTP Application programming interface

[www.SMSify.com](http://www.SMSify.com)  
<mailto:info@smsify.com>

# SMSify HTTP API

## Introduction

SMSify system offers various methods to send and receive SMS messages. This document contains specifications for following methods:

- Send messages using HTTP XML POST
- Send messages using HTTP GET
- Receive messages using HTTP GET

SMSify system also offers the Network Query solution. This is a service in regards to Mobile Number Portability (MNP), enabling the identification of network that a mobile phone number belongs to, and if the mobile number does exist. Please see chapter 4 for the specification details.

## 1. HTTP(S) XML POST

The URL used to post XML formatted data is:

<http://89.164.98.10/sms/AddOn/SMSService/XML/XMLInput.aspx>

Duplicate numbers in one post request will be filtered in the SMSify system (single recipient can't receive more than 1 message per request).

### 1.1. XML

The XML formatted string must have „XML=” at the beginning.

There are two ways of formatting XML string:

- Without registered delivery
- With registered delivery

XML formatted without registered delivery:

```
<SMS>
  <authentication>
    <username></username>
    <password></password>
  </authentication>
  <message>
    <sender></sender>
    <text></text>
    <flash></flash>
    <type></type>
    <bookmark></bookmark>
    <binary></binary>
    <datacoding></datacoding>
    <esmclass></esmclass>
    <srcton></srcton>
    <srcnpi></srcnpi>
    <destton></destton>
    <destnpi></destnpi>
  </message>
  <recipients>
    <gsm></gsm>
    <gsm></gsm>
    <gsm></gsm>
    <gsm></gsm>
  </recipients>
</SMS>
```

XML formatted with registered delivery:

```

<SMS>
  <authentication>
    <username></username>
    <password></password>
  </authentication>
  <message>
    <sender></sender>
    <text></text>
    <flash></flash>
    <type></type>
    <bookmark></bookmark>
    <binary></binary>
    <datacoding></datacoding>
    <esmclass></esmclass>
    <srcton></srcton>
    <srcnpi></srcnpi>
    <destton></destton>
    <destnpi></destnpi>
  </message>
  <recipients>
    <gsm messageId="clientmsgID1"></gsm>
    <gsm messageId="clientmsgID2"></gsm>
    <gsm messageId="clientmsgID3"></gsm>
    <gsm messageId="clientmsgID4"></gsm>
  </recipients>
</SMS>

```

Text in green is for optional parameters.

## 1.2. Parameters specifications

Authentication	username	Client's username for login into SMSify system
	password	Client's password for login into SMSify system
Message	sender	Dynamic sender ID of the message; allowed length for alphanumeric string up to 11 characters or for numeric string up to 14 characters
	text	The message body (at the moment 160 characters)
	flash	Can be 0 or 1 0 sends a normal SMS 1 sends Flash SMS

	type	optional parameter, to send wap bookmaks value has to be set to "bookmark", to send concatenated SMS, value has to be set to "longSMS" (for text messages only)
	wapurl	Wap push content: For example: www.smsify.com/something.jpg
	binary	Binary content, using hexadecimal format. Example: 410A0D4243. Cannot be used together with "text" parameter
	Data coding	Data coding parameter, default 0. Example: 8 (unicode data)
	Esm class	Esm_class parameter, default 0.
	SrcTon	Source - ton parameter (see below)
	Srcnpi	Source - npi parameter (see below)
	Destton	Destination - ton parameter (see below)
	Destnpi	Destination – npi parameter (see below)
Recipients	GSM	Destination address of the message, must be in international format without leading „0“ or „+“ For example: 41793026727
	GSM messageId="clientmsgID"	Registered delivery; client set the msgID, later will be described how to collect delivery reports

Parameters src-ton and dest-ton may have one of the following values:

Unknown	0
International	1
National	2
Network Specific	3
Subscriber Number	4
Alphanumeric	5
Abbreviated	6

Parameters src-npi and dest-npi may have one of the following values:

Unknown	0
ISDN (E163/E164)	1
Data (X.121)	3
Telex (F.69)	4
Land Mobile (E.212)	6
National	8
Private	9
ERMES	10
Internet (IP)	14
WAP Client Id (to be defined)	18

As an example, if you want to send message with originator (sender – name) "12345" (note, no leading "+"), you should indicate src-ton = 2 (national), src-npi = 1. If you want to add leading "+" in originator, you should use "src-ton" = 1 (international), src-npi = 1.

If you want to use alphanumeric originator, please set "src-ton"=5 (alphanumeric), src-npi = 1.

If you do not specify src-ton and src-npi parameters, your message will be sent with src-ton=1 for numeric sender, and src-ton=5 for alphanumeric sender.

## 1.3. Return XML

After the POST XML initiated by the client some status codes will be available.

The return XML string format will be:

```
<RESPONSE>
  <status>status_code</status>
  <credits><credits>
</RESPONSE>.
```

Status codes:

Status	Value	Description
AUTH_FAILED	-1	Invalid username and/or password
XML_ERROR	-2	Incorrect XML format
NOT_ENOUGH_CREDITS	-3	Not enough credits in user account
NO_RECIPIENTS	-4	No good recipients
SEND_OK	> 0	Number of messages that will be sent

## 2. HTTP(S) GET

Example for normal text message:

<http://89.164.98.10/sms/Addon/SMSService/SendSMS.aspx?user=xxx&password=xxxx&sender=Friend&SMSText=mesagetext&IsFlash=0&GSM=38598514674>

Example for binary parameter usage:

<http://89.164.98.10/sms/Addon/SMSService/SendSMS.aspx?user=xxx&password=xxxx&sender=Friend&binary=41424344&GSM=38598514674>

In order to use UDH, you have to use esmclass parameter:

<http://89.164.98.10/sms/Addon/SMSService/SendSMS.aspx?user=xxx&password=xxxx&sender=Friend&binary=06050400010241424344&GSM=38598514674&esmclass=64>

Query string parameters:

Parameter	Description
user	Username
password	Password
sender	Sender name of the message For alphanumeric sender it can be up to 11 characters For numeric sender it can be up to 14 characters
SMSText	Message text (160 characters)

IsFlash	0 – false 1- true flash message → displays directly on cell's screen optional parameter: default value = false
GSM	Recipient GSM number in international format (38598xxxx, 38591xxxx, ...)
Type	Optional parameter: For wap bookmarks set to type=bookmark, for concatenated (long) SMS set type=LongSMS
Bookmark	The wap URL link
DataCoding	Data-coding parameter Optional parameter, default value = 0
Esmclass	Esm_class parameter Optional parameter, default value = 0
Binary	Binary content, optional parameter Format same as in XML <binary> parameter
Srcton	Source-ton, please check XML parameter description
Srcnpi	Source-npi, please check XML parameter description
Destton	Destination-ton, please check XML parameter description
Destnpi	Destination-npi, please check XML parameter description

Return values:

Value	Description
-1	SEND_ERROR Not in use actually
-2	NOT_ENOUGHCREDITS
-3	NETWORK_NOTCOVERED
-4	SOCKET_EXCEPTION Not in use actually
-5	INVALID_USER_OR_PASS
-6	MISSING_DESTINATION_ADDRESS
-7	MISSING_SMSTEXT
-8	MISSING_SENDRNAME
-9	DESTADDR_INVALIDFORMAT
-10	MISSING_USERNAME
-11	MISSING_PASS
-13	INVALID_DESTINATION_ADDRESS
> 0	Successfully, return value is the ID of sent message

## 2.1. Additional HTTP GET commands

Additional HTTP GET commands use following syntax:

<http://89.164.98.10/sms/Addon/SMSService/command.aspx?username=X&password=X&cmd=X>

Currently available commands are:

CREDITS – returns available credits on your account

## 3. Collecting delivery reports

There are 2 methods of collecting delivery reports: PUSH and PULL.

### 3.1. PUSH method

To be able to collect delivery reports we will need to set the delivery report URL for you. Please contact us with this and we will gladly set this up for you.

If the client's delivery report URL is unavailable because of any reason, we will try to send delivery report again for a 60 seconds, then for five minutes, and then every hour in next 24 hours period. If your URL is not available for the whole period, delivery reports will be lost.

The format of the XML delivery report structure will be:

```
<deliveryreport>
  <message id="msgID" sentdate="xxxxx" donedate="xxxxx"
  status="xxxxxx" />
  .....
</deliveryreport>
```

Delivery report example:

```
<DeliveryReport><message id="1023012301" sentdate=""
donedate="2005/7/19 22:0:0" status="NOT_SENT" /></DeliveryReport>
```

XML attributes description:

Attribute	Description	
id	Client's Id of the message	
sentdate	date/time when message was submitted from the client to the SMSify system. (format: yyyy/mm/dd hh:mm:ss)	
donedate	date/time when SMSC notified delivery report to the SMSify system (format: yyyy/mm/dd hh:mm:ss)	
status	NOT_SENT	The message is enqueued in the SMSify system but was not cannot be submitted to SMSC (possible reason SMSC connection is down)
	SENT	The message is sent over a route that does not support delivery reports
	NOT_DELIVERED	The message could not be delivered
	DELIVERED	The message is successfully delivered to the recipient.
	NOT_ALLOWED	The client has no authorization to send to the specified network (the message will not be charged)
	INVALID_DESTINATION_ADDRESS	GSM recipient is invalid / wrong
	INVALID_SOURCE_ADDRESS	You have specified wrong / invalid / unallowed source address (sender name)
	ROUTE_NOT_AVAILABLE	You are trying to use routing that is not available for your account
	NOT_ENOUGH_CREDITS	There is no available credits on your account to send the message
	INVALID_MESSAGE_FORMAT	Your message has invalid format

### 3.2. PULL method

The URL to get DR's over HTTP GET method is:

<http://89.164.98.10/sms/Addon/SMSService/XML/GetXMLDr.aspx?user=X&password=x>

Parameters:

- User
- Password

Return values:

- 5 - invalid username and/or password
- 10 - missing username
- 11 - missing password

The XML delivery report structure is the same as defined in chapter 1.4.1.

## 4. Network query

After executing network query, you will receive response flag, indicating whether the mobile phone number (*MSISDN*) is active or not. This is asynchronous request, so you have to provide HTTP URL that can handle our response.

To execute a network query, you have to send HTTP GET request in the following format:

<http://89.164.98.10/sms/AddOn/NQ/query.aspx?username=xxx&password=xxx&number=xxx>

As a result, you will receive reference number of your request. Result of network query will be reported to you using this reference number. You have to prepare URL that can be used for reporting results. Network query result is reported in the following format:

<http://your-url?status=STATUS&number=NUMBER&ref=ID&NECNetworkType=NECNETWORK&NECValue=NECVALUE&NormNetCode=NORMNETCODE&GSMError=GSMERROR>

Description of parameters:

Parameter	Description	Mandatory?
STATUS	0 / 1 / 2 – Response flag indicating that network query has been executed successfully	Yes
REF	Reference number of your request	Yes
NUMBER	requested <i>MSISDN</i>	Yes
NECNETWORKTYPE	Network error value	No
NECVALUE	Network error code (0 = <i>MSISDN</i> exists, 1 = <i>MSISDN</i> absent, 2 = <i>MSISDN</i> does not exist)	No
NORMNETCODE	Normalized network code	No
GSMERROR	GSM Error code	No

If the STATUS parameter is 1, then you will receive the network code (NORMNETCODE) of the network the *MSISDN* belongs to.

If the request cannot be answered, then STATUS will be 0.

If the STATUS parameter is 2, then your request contained illegal data and could not be processed.

## Receiving SMS messages into your system

We provide few different ways for collecting SMS messages sent by GSM phone of your customers. As an example, we can host your GSM SIM cards at our GSM modem farm. When your customer sends SMS message to that SIM, it arrives in our system. For more detailed specifications and options, please do not hesitate to ask our sales department.

After message arrived to our system, we can forward it to your server using HTTP GET request. You have to provide us URL we should use. It means that you have to prepare such URL on your web server. We are able to forward following parameters:

Parameter	Description
Sender	Sender of SMS message (GSM phone number)
Receiver	Receiver number (if available)
Text	Text of received message
Bin	Binary content of received message
Datetime	Date and time when message arrived
Datacoding	Data - coding of the message
Esmclass	ESM-class parameter of the message

"Receiver parameter" will be set to value of your GSM SIM mobile number (if you are using SIM hosting to receive messages).

In case you provided URL with both "bin" and "text" parameters, take care at the following.

If datacoding parameter is "0", then we will forward to you only message text, "bin" parameter will be set to "" (empty string). If datacoding is not "0" (example "8" = unicode message), then we will send you binary content only, parameter "text" will be set to "" (empty string).

However, if you do not support both parameters ("bin" and "text") in URL (of course, you should use at least one of them, in order to receive message content) - in such case, we will provide everything, no matter what is in datacoding parameter. We use "send only binary or only text" logic to make HTTP GET requests as short as possible.

As an example, if you provide following URL :

[http://some.server.com/incoming\\_sms.php?who=%sender%&what=%text%](http://some.server.com/incoming_sms.php?who=%sender%&what=%text%)

then our system will make following HTTP request (after receiving message from +38598123123, that says "ABC"):

[http://some.server.com/incoming\\_sms.php?who=38598123123&what=ABC](http://some.server.com/incoming_sms.php?who=38598123123&what=ABC)

Note that there is no leading "+" in "sender" field. In case you want to use "binary" parameter instead of text, you should provide following URL:

[http://some.server.com/incoming\\_sms.php?who=%sender%&what=%bin%](http://some.server.com/incoming_sms.php?who=%sender%&what=%bin%)

and now we will make the following request:

[http://some.server.com/incoming\\_sms.php?who=38598123123&what=414243](http://some.server.com/incoming_sms.php?who=38598123123&what=414243)

Note that binary content is in hexadecimal format.