Gatesms.eu

Mobile Solutions for Business

**TECHNICAL SPECIFICATIONS**

# XML Web API

**GATESMS.EU, version 1.1**

Prepared by: Gatesms.eu

# Contents

# Document version history

- 1.0 - Basic version 1.03.2011
- 1.1 - Description of SMS text length, and SPEEDsms options 20.12.2011

# Security

## *General requirements*

1. Communication in both directions must consider the security of the HTTP transmission described below.

## *HTTP transmission security mechanism*

Security for remote services via HTTP available is independent of the method and format of messages sent

**General Principles**

Remote methods return the http status **of less than 300** when an operation succeeds. Usually, this will be either status **200** or **204.** If the operation fails, the service returns an error status **greater than or equal to the 400.** For clarification sake, any status of 400 inclusive thru 500 exclusive only means an error in your request (for example, wrong parameters - 400, an authentication error - 403, bad address - 404). Statuses over 500 inclusive mean an error on the server side. It is possible that, after removal of the fault, the same request can be handled properly. This means that the error status of below
500 does not make sense to repeat the request, and for the status of 500 or less, you can retry the request. The

objectives of the security mechanism:

1. Client and server authentication
2. Verification of the accuracy and veracity of the request and response
3. Protection against "reversing the clock," and processing many of the same requests in a row with the same timestamp.

For each client, the server prepares a pair consisting of **KeyID** and **SecretKey** where **KeyID** is an identifier *(an open login specified in the registration process)* and **SecretKey** is a secret key *(a password 10 characters long)*. You can either read or generate the secret key after having logged in from the *USER DATA tab*. Both of these values are known to the client and the server.

Authentication is based on two HTTP headers added to the request and response. It is also acceptable to use an **X-BP header.**

1. **X-GT-Timestamp** - an operation imestamp (unix timestamp)

2. **X-GT-Auth** - an additional header for authentication.
   Example: X-GT-Auth: [KeyID]:[Hash] where Hash is a message digest prepared as described below.
   *X-GT-Auth: 1234567:098f6bcd4621d373cade4e832627b4f6.*

**For request**
Hash = MD5(HTTP_method + URI_Path_Query + MD5_Body + Accept + X-BP-Timestamp + SecretKey)

Where MD5 is a hashing function and '+' indicates a concatenation of strings, and:

1. **HTTP_method** is GET, POST, PUT, etc.
2. **URI_Path_Query** is the path and query string, which is the part of the URL starting with the first character of '/', e.g.: /sms_xmlapi.php
3. **MD5_Body** - is MD5 from the content of the request if existing
4. **Accept** - header specifying the response format (compatible with the standard), has a value like application/xml
5. **X-GT-Timestamp** - a timestamp for an operation (same as in the header) having been performed
6. **SecretKey** - the client's key

**Example:**
MD5(POST/sms_xmlapi.phpZXQW345YULAccept:application/xml12844675753LQBwxTiDnv)

The request receiving party is required to check out whether the **Hash** is correct. If not, the 403 status should be returned.

**For response**

Hash = MD5(MD5_Body + Content-Type + X-GT-Timestamp + SecretKey)

where **SecretKey** is the key of the client from whom we have received the message

**Example**

MD5(9843f33LQd8xTiDnvGAKoc8n7pQ5qi3CW9SG584ContentType:application/xml12844675753LQBwxTiDnv)

The client receiving a response from the server is required to determine whether the response is signed properly if the request was properly handled. Responses with status>=400 do not have to be signed.
In case of an error, the error messages can be included in the HTTP response as text.

# The definition of remote services

Data will be transmitted mainly in XML format. The both sides have an obligation to ensure that the documents sent comply with the XML standard. This concerns, in particular, the rules on the use of special characters in the messages. An XML document is sent by way of the *POST* **body** parameter. Prior to sending, the content of this parameter has to be URL encoded by replacing the non-alphanumeric characters with two hexadecimal digits preceded by a percent sign (%), and having the spaces encoded as plus (+). The content must be encoded only after calculating the md5 function. The most common characters resulting in transmission errors: '+' => '%2B', '&' => '%26', '#' => '%23'.

For downloading status messages, an additional **X-GT-Action** header is required: **Get Status,** for importing the **X-GT-Action** database of numbers**: Put HLR**.

The server returns an additional header, such as **X-GT-Status: 002** after successfully importing a message package or, in the case of global errors, such as the lack of adequate funds in your account, server errors, Internet or GSM providers, and similar status codes.

The Accept and Content-Type headers must be set correctly, for example:

Content-Type: application/x-www-form-urlencoded

Accept: application/xml

### *Sending messages to the Gatesms.eu application*

Address of the service: *http://www.gatesms.eu/sms_xmlapi.php*

Metoda : **POST**
The variable name that sends data using the POST method: **body**
Content-Type: application/x-www-form-urlencoded
Accept: application/xml

**Sample request - the *body* parameter:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<package test="true">
<msg pl="true" phone="48518778404" localid="123-34-AA-33" isflash="true" from="Your Name">Sample content 1</msg>
<msg wap="" pl="true" phone="48518778404" localid="123-34-AA-34">Tresc z dziwnymi znakami [ i ] i &quot; oraz /</msg>
<msg pl="true" phone="48518778404" localid="123-34-AA-35" from="Your Name">Content 3</msg>
<msg phone="48518778404" localid="123-34-AA-36">Content sent to wrong number</msg>
<msg pl="true" phone="48518778404" localid="123-34-AA-37">Message with Polish characters ąśęłó</msg>
</package>
```

***package*** element*:*

- Optional attributes
  - *test* - *true* indicates a simulation of sending, set by default as *false*
- Mandatory elements
  - *msg* - at least one message to send ***msg*** element*:* Contains a text message to be sent
  - Mandatory attributes
    - *phone* - an 11-digit phone number for sending
    - *llocalid* - the message identifier of the client system
  - Optional attributes
    - *wap* - correct address (http://www.gatesms.eu), it allows you to send an active link - wap push *(the sender is required)*
    - *pl*- is the ability to use Polish characters, set by default as *false*
    - *isflash* - determines whether to send messages as flash sms, set by default as *false*
    - *from* - sender's name up to 11 characters long; if such a name is given that the client has no right to use, this will result in a sending error, the names can be registered by writingto the address pomoc@gatesms.eu
    - *time* - timestamp indicating the date and time of sending a message in the future; by default, the message is sent immediately

- *speed* – the parameter placing a message at the beginning of the queue of text messages being sent. When combined with the attribute *from*, the message is sent as a SPEEDsms with the reach time of less than 7 seconds.

**Sample response - an XML document**

Client receives the same response asking for change of status. *See Sending delivery reports.*

Content-Type: application/xml; charset: utf-8
Accept: application/xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<report>

    <status timestamp="1295959673"  localid="123-34-AA-33" ok="true" globalid="452342323">002</status>
    <status timestamp="1295959673"  localid="123-34-AA-34" ok="true" globalid="452342324">002</status>
    <status timestamp="1295959673"  localid="123-34-AA-35" ok="true" globalid="452342325">002</status>
    <status timestamp="1295959673"  localid="123-34-AA-36" ok="false" globalid="452342326">005</status>
    <status timestamp="1295959673"  localid="123-34-AA-37" ok="false" globalid="452342327">005</status>

</report>
```

The ***report*** element contains a list of ***status*** elements**.** The number of the elements is exactly the same as the number of sent in messages.

The ***status*** element*:*

- Must contain the status code of a message.
- Mandatory attributes
  - *timestamp* - unix timestamp, the time of generating the status messages
  - *localid* - imessage identifier in the client system, as sent in
  - *globalid* - message identifier in the gatesms.eu system
  - *ok* - *true* unambiguously determines if the message is right or wrong *(false)*
- Additional attributes
  - *net* - is the network to which the message was sent
  - *sender* – the number from which a text message was sent

***Notes on the service***

Messages are identified by *localid*. Thus, in one package of messages, you can send multiple messages to one phone number. If, for some reason (e.g., network problems), there will be re-transmitted a message with the same *localid,* the service is required to return the true status of the message without sending it again.

If the service is not performed correctly due to an incorrect request (a status of between 400 and 500 was returned), we assume that, surely, no message has been sent. If the service has returned the status of more than 500, the fate of the message is not certain. Such messages will be sent again with regard to checking their *localid.*

The maximum size of one message packet is 300 items.

## *Sending delivery reports*

If a *report* address is defined in the *QUEUE/HISTORY* tab after logging into the system, system will will be sending you messages according to the attached schedule, every time you change the message status code.

Address of the service: *XML: customer service address*

A list of current changes in the status of messages is sent. The format is the same as for the response to sending message, wherein the localid attribute is not required here. Status codes in the Appendix. An XML document is sent using the *POST body* parameter. The status of the remote service corresponds to the HTTP 204 status if the report is properly received. The server will retry sending the status for another 60 minutes at intervals of 5 minutes unless you get the http 200, 202, 204 status or an OK message. The remote service will respond to the server the 202 status if it wants to continue to wait for the 004 confirmation code. In the case of the status higher than 004, the service should answer with the 204 header.

Address of the service: *http://www.gatesms.eu/sms_xmlapi.php*
Method: **POST**
The variable name that sends data using the POST method: **body**
Content-Type: application/x-www-form-urlencoded

Accept: application/xml

X-GT-Action: Get Status

**Sample request - *the body parameter***

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<report>
        <status timestamp="1295959673" ok="true" globalid="452342323"/>
        <status timestamp="1295959673" ok="true" globalid="452342324"/>
        <status timestamp="1295959673" ok="true" globalid="452342325"/>
</report>
```

**Sample response - an XML document**

## *Sending inbound messages*

If an *SMS response* address is defined in the *QUEUE/HISTORY* tab after logging into the system, the system will be sending you *SMS responses* according to the attached schedule, every time it receives an SMS message.

Address of the service: *XML: customer address*
Method: **POST**
The variable name that sends data using the POST method: **body**
Content-Type: application/x-www-form-urlencoded
Accept: application/xml

**Sample request**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<inpackage>
        <msg to="48500100200" time="1297413504" phone="48500123456" >]akas wiadomosc</msg>
        <msg to="48500100200" time="1297413504" phone="48500123888" >]akas wiadomosc2</msg>
</inpackage>
```

The **inpackage** element contains a **msg** list**.** The **msg** element contains the message.

- Mandatory attributes
  - *to* - Gatesms access number to which the message has arrived
  - *time* - time of message delivery (unix timestamp)
  - *phone* - client's phone number

Option temporarily unavailable!

**Sample response**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<report>
<status ok="true" localid="123-221-3-0" globalid="1234567"/>
<status ok="true" localid="123-221-3-1" globalid="1234568"/>
</report>
```

The **report** element contains a list of the **status** elements The **status** element

- Mandatory attributes
  - *ok* - determines whether a message has been received properly
  - *localid* - message identifier in the client's system
  - globalid - message identifier in the Gatesms system

If the service receives a duplicate message, it will respond as if it has received the correct message.

## *Importing contacts to the base Gatesms.eu*

Address of the service: *to* *http://www.gatesms.eu/sms_xmlapi.php*

Method: POST

Content-Type: application/x-www-form-urlencoded
Accept: text/plain
X-GT-Action: Put HLR

This method sends a list of numbers to be imported to the database in order to verify the customer's network before sending. To save traffic, this method uses plain text for transmission as an exception.

**Sample request**

```
48602300000
48602300001
48602300002
48602300003
48602300004
```

The request contains a list of numbers separated by a new line character. The numbers are sent using the *POST* ***body*** parameter. This method returns the count of submitted numbers as text. The service adds only those numbers that are not in the repository. Duplicates are ignored.

**Sample response**

```
5
```

### *Message status codes*

000 - No connection to SMSC.
001 - No authorization, incorrect username or password.
002 - This message has been queued to be sent (waiting for confirmation).
003 - This message has been sent to the recipient.
004 - This message has been received by the recipient (acknowledgment of receipt).
005 - Error message.
006 - Inactive number .
007 - Error in the delivery of the message.
008 - Message received by the SMS center.
009 - GSM network error.
010 - This message has expired due to the inability to deliver to the recipient.
011 - This message has been queued for later sending.
012 - Provider error, contact your system administrator mmediately.
103 - No text in the message field or incomplete text field.
104 - Sender's field Incorrectly filled or missing.
105 - The text field is too long.
106 - Incorrect or missing number field.
107 - Invalid parameter type.
110 - SMSC does not support this type of message.
113 - The text field is too long.
201 - System error, contact your system administrator immediately.
202 - Not enough credits in your account.
203 - Operation not permitted, contact your system administrator immediately.
204 - Account disabled. Most likely blocked.
205 - Destination network locked.
301 - Missing or incorrect message identifier.
500 - Incorrectly filled sender field or text field too long.
600 - No credits in the Premium account for the given customer.
700 - No confirmation of recording the record.
800 - Deduplicator: No Recorded Record! Message repeated.
888 - Restart blocked text messages.
999 - External Infrastructure - transitional status.

### *SMS text length*

Table division of SMS's without Polish characters:

| Number of characters | Length of each SMS section | Number of SMS messages that the recipient receives | Number of SMS messages used for customer billing |
|---|---|---|---|
| 1-160 | 1 x 160 characters | 1 | 1 |
| 161-304 | 2 x 152 characters | 1 | 2 |
| 305-456 | 3 x 152 characters | 1 | 3 |
| 457-608 | 4 x 152 characters | 1 | 4 |

* With messages of more than 160 characters, the message content is divided into single SMS's where each of them is 153 characters long. The remaining 7 characters are reserved for information allowing to merge messages into one long SMS.

| Number of characters | Length of each SMS section | Niumber of SMS's that the recipient receives | Number of SMS's for customer billing |
|---|---|---|---|
| 1-70 | 1 x 70 characters | 1 | 1 |
| 71-134 | 2 x 67 characters | 1 | 2 |
| 135-201 | 3 x 67 characters | 1 | 3 |
| 202-268 | 4 x 67 characters | 1 | 4 |

* With messages of more than 70 characters, the message content is divided into single SMS's where each of them is 67 characters long. The remaining 3 characters are reserved for information allowing to merge messages into one long SMS.

# Appendix A

### *Class supporting Web XML Api*

Example of handling of the SMS_SENDER class from the sms_xmlsender.php file.

```
<?
include_once("example/sms_xmlsender.php");

//Class constructor and the parameters

   $sms = new SMS_SENDER;
   $sms->login='mylogin@gatesms.eu';
   $sms->pass='0a3eeeee8dew45';
   $sms->test='false';

//Sending packets of messages with a dynamic ID localId:

   $sms->localIdRandom=1;
   for($z=0;$z<12;$z++) {
            $sms->to=48501123456;
            $err = $sms->AddMsg("TEST A".$z);
            $LocalmsgId = $sms->localId;
   };
   $report = $sms->sendsms();
   exit;

//Sending packets of messages with own localId identifier:

   $sms->localIdRandom=0;
   for($z=0;$z<12;$z++) {
            $sms->localId='540';
            $sms->to=48601098765;
            $err = $sms->AddMsg("TEST A".$z);
   };
   $report = $sms->sendsms();
   exit;

//Query about the status

   for($z=100000;$z<102

   225;$z++) {
            $sms->globalId=$z;
            $sms->AddStat();
   };
   $report = $sms->GetStat();
   exit;

//Adding numbers for verification

   $numery=array(500600300,602301605,723898549);
   echo $sms->AddHlr($numery);
   exit;

?>
```