SMS Traffic API

Technical description
of protocols for SMS messages terminating

Version 1.85

Phone number format 3
Sender name (originator) 4
Reliability and fault tolerance 5
HTTP(S) protocol 6
Parameters list for HTTP(S) protocol 6
Examples of responses 12
SMS message status checking 16
Automatically status receiving method 18
Account balance checking 19
SMS message statuses 21
Error codes for SMS Traffic HTTP(S) API 22
Subscribers lists managing 24
SMPP protocol 28
Encodings 32
Common errors 33
UDH-concatenation 35
SOAP protocol 37
SendBulkSms method 39
SendIndividualSms method 40
GetSmsStatus Method 41
GetBalance Method 42
SMTP protocol (version 1) 43
SMTP protocol (version 2) 45
Mobile Virtual Numbers (Incoming Numbers) 47
Forwarding MO messages to email 47
Forwarding MO messages to HTTP script 47
Forwarding MO messages via SMPP 47
Request a list of MO messages 48
Shortcode 50
Document history 51

Phone number format

The phone number of the subscriber has to be in an international format like *country_code operator_prefix phone_number* (without leading "+" symbol).

Country code	Operator prefix	Phone number	Country and operator
7	916	1112233	Russia, MTS
380	67	1234567	Ukraine, Kievstar
34	6400	12345	Spain, Jazz Telecom
371	2231	1234	Latvia, Tele2

Sender name (originator)

You can set any alpha-numeric sender while sending SMS messages, but there is a limitation of the length of the sender you can set in GSM standard. In case you want to set a phone number as a sender so maximum length of originator will be 15 digits, in case you want to set some words as a sender (any set of digits, latin letters or even some symbols) so maximum length of originator will be 11 symbols. In other standards could be other limitaions and restrictions of sender possible values.

Different operators can declare some additional restrictions of senders that could be used for SMS messages termination on the operators network. It can be restrictions of letters and symbols using or otherwise it can be restrictions of using phone numbers as a sender, or it can be white lists of approved senders and sometimes we meet blacklists of senders etc.

It is restricted to send SMS messages to Russian subscribers from senders like phone numbers, any numeric senders are restricted by Russian operators.

Every Ukrainian operators demand every sender to be approved. So before you start working with Ukraine subscribers you have to send a request to our email tech.support@smsmail.ru to activate senders you wish to use. In the request you have to notice a traffic type you will send (financial, news, marketing etc), traffic urgency (so 24/7 or day only), the company name that will generate this traffic and company's web site. Numeric senders are restricted and will not be approved.

Most of Russian operators have black lists of senders. SMS Traffic also has its own black list of senders that were used for SPAM, fraud etc. We don't provide these lists to any other side.

There can be different operator reaction in case you terminate a message with restricted sender. Usually there will be "Rejected" delivery report, but sometimes other after-effects can be like fake "Delivered" report or simple "Undelivered" report.

According to GSM 03.38 you can use any symbol from encoding "default GSM alphabet" in sender. But if we meet symbols from extended table of encoding "default GSM alphabet" we replace it with symbols from main table trying to pick up symbols looks like replaced ones.

We know from our great experience that some SMSCs and mobile phones do not support full table of GSM encoding so we recommend you to use the set "a-z", "A-Z", "0-9", "-", "_", ".".

Reliability and fault tolerance

We have two different geografically separated communication centers for better reliability and fault tolerance. So in case you have some problem while working with our main communication center you can re-connect to the failover one that completely duplicate all the functions of the main center.

API protocol	Main	Reserve
HTTP	https://api.smstraffic.ru/multi.php	https://api2.smstraffic.ru/multi.php
SMPP	server1.smstraffic.ru:4442	server2.smstraffic.ru:4442
SMPP (TLS)	server1.smstraffic.ru:4441	server2.smstraffic.ru:4441

There no need for re-connecting to any reserve server if you choose to work with SMTP protocol. All reserve servers are set in corresponding MX records so email letters routing should be done automatically.

An address of the reserve server is provided in WSDL for SOAP protocol.

HTTP(S) protocol

You can call web script api.smstraffic.ru/multi.php with GET or POST method with necessary parameters for sending SMS messages. You can choose between usual HTTP and secure HTTPS protocols, ports are standard: 80 for HTTP and 443 for HTTPS.

HTTP-requests have to comply with the requirements of HTTP cpecifications (RFC 2616). Every parameter should be encoded correctly (RFC 2396, section 2.4). HTTP 1.0 and 1.1. are supported. If you work in HTTP 1.1. then be ready to get response in chunked transfer encoding. Also you have to set header «Content-Type: application/x-www-form-urlencoded» in POST requests.

Parameters list for HTTP(S) protocol

Name	Possible values	Description
login	Text	Your login on SMS Traffic platform. Mandatory parameter.
password	Text	Your password on SMS Traffic platform. Mandatory parameter.
phones	A list of phone numbers comma separated.	A list of phone numbers in an international format comma separated (number format is described in "Phone number format" section). At least one phone number should be in a list. Mandatory parameter in case you don't use *group* parameter.
message	Text. Maximum length is 17085 in case of *rus*=1 or *rus*=5, maximum length is 39015 in case of *rus*=0	Text of SMS message to send. Length of one SMS could be 160 symbols maximum with latin symbols (*rus*=0) or 70 symbols otherwise (*rus*=1 or *rus*=5). If length is more than these upper limits then message will be separated into several parts with length of each 153 symbols for latin SMS and 67 for cyrillic and others. If you have some pre-ordered limit of parts for every message you can use parameter *max_parts*. Parameter *message* is mandatory in case you don't use *individual_messages*.

Name	Possible values	Description
rus	0	The *message* is in "Windows-1251" encoding. Every Russian letters will be translited to latin analogues. Maximum length of one SMS message is 160 symbols or 153 symbols for one part of long concatenated message. This value is default.
	1	The *message* is in "Windows-1251" encoding. Maximum length of one SMS message is 70 symbols or 67 symbols for one part of long concatenated message. No matter if only latin symbols are in message.
	5	The *message* is in "UTF-8" encoding. Maximum length of one SMS message is 70 symbols or 67 symbols for one part of long concatenated message. No matter if only latin symbols are in message.
originator	Alpha-numeric (maximum length is 11 symbols)	This is a text name of SMS sender that subscriber will see on the mobile phone screen. It can contain latin letters, digits and some special symbols like "-", "_", ".". Every cyrillic letters will be translated into latin analogue. The subscriber can't call back to this sender or send an SMS as an answer to such a sender. Examples: "MyCompany", "787-35-95", "SMS.Traffic", "SMS TRAFFIC".
	Digits only (maximum length is 15 digits)	The subscriber can call back or send an SMS message as an answer when receiving an SMS from such a sender. Examples: "74957873595", "88001000258", "3299".
flash	1	A message will be sent as a Flash-SMS. Usually Flash-SMS appear on the mobile phone screen immediately and can't be saved in mobile phone memory. Maximum length of Flash-SMS message is 160 symbols (*rus*=0) or 70 symbols (*rus*=1 or *rus*=5).
	0	Usual SMS message. 0 is default value.

Name	Possible values	Description
start_date	Date and time in format "YYYY-MM-DD HH:MM:SS"	Date and time of SMS message sending. If you leave this parameter empty the message will be sent immediately. Time have to be in MSK timezone and heve to be not more than 3 days in a future. In case the *start_date* is 5 minute or more in a future then *want_sms_ids* can't be used. Example: "2016-03-01 09:00:00".
max_parts	A number from 1 to 255	The maximum number of parts the *message* can be separated. If occur the situation when number of parts after separating of *message* is more than *max_parts* so only first *max_parts* parts will be sent and the rest will be skipped. 255 is a default value.
gap	Float number	An interval (in seconds) between SMS messages from current HTTP(S) request. Example: "1", "0.5", "0.05". You have to keep in mind that every part of concatenated SMS message is considered as a single SMS message. 1 is a default value. 0.05 is a minimum value.
group	Text	A group name, defined in private web-interface of SMS Traffic platform, that wanted to be used for SMS messaging. This parameter replaces a *phones* parameter and *message* will be terminated onto every phone number from this *group*.
timeout	Positive integer number	This parameter can be used to set a time of life of SMS message. The message will be expiered in *timeout* seconds after submition to the SMS Traffic platform. This parameter is supported by only short list of operators and the effect is not guaranteed. Minimum value is 10 minutes but this limit is not strict and only recommended because most of operators that support such feature will ignore this parameter if it is set to value less than 10 minutes. Maximum value is 86400 (equal to 24 hours)

Name	Possible values	Description

individual_messages	0 or 1	If you need to send individual message to every subscriber then you can set parameter *individual_messages*=1, leave *message* empty and set parameter *phones* in format:
		`phone_number1 message1`
		`phone_number2 message2`
		`phone_number3 message3`
		(phone number and corresponding message are space separated, different phones+messages are separated by line break (ASCII-symbol 0xA), message text can't contain line break symbol) 0 is a default value.

delimiter	text	This parameter can be used with *individual_messages* if you want to have line break symbol in SMS message text. For example you can set *delimiter*=ABC and then *phones* could be:
		`76161234567 message`
		`text`**`ABC`**`7903123456 text of message`
		`on different`
		`lines`
		For sure message text in this example shouldn't contain "ABC". Line break symbol (ASCII-symbol 0xA) is a default value.

Name	Possible values	Description
want_sms_ids	0 or 1	If you have to track the delivery status of the message later then you have to set *want_sms_ids*=1. In this case you will receive an information about sms identifiers that were assigned by SMS Traffic platform to every SMS in XML in HTTP(S) response. Due to HTTP(S) protocol is a protocol for online work so you can't use *want_sms_ids* with *start_date*. Also *want_sms_ids* can be used when every message from current HTTP(S) request could be sent not later than 5 minutes starting from time the request was received by SMS Traffic platform. For example, if *gap*=1 and *phones* contains 301 phone numbers so we know that last 301th message will be sent in 301 second that later than 5 minutes. In this case 418 error will be in response XML and no one message will be sent. 0 is a default value.

with_push_id	0 or 1	This parameter can be used with *want_sms_ids*=1 и *individual_messages*=1. You can use it if you have to send every message with your own unique identifier, then you will get SMS Traffic usual sms identifiers binded to your own internal ids. Usually used for long concatenated messages. Internal id should be set before every phone number separated by ":". Obviously internal id shouldn't contain ":". Example:
		`push_id1:phone_number1 long message1`
		`12345678:phone_number2 long message2`
		`one-more:phone_number3 long message3`
		0 is a default value.

Name	Possible values	Description
ignore_phone_format	0 or 1	If you are not sure that every phone number you set in the request is correct then you can set *ignore_phone_format*=1. In usual case if *ignore_phone_format*=0 and some number is incorrect an error 418 will be in the answer and no messages will be sent. For *ignore_phone_format*=1 every phone numbers will be counted as correct, but messages to incorrect numbers will be billed according to your tariff. 0 is a default value.
two_byte_concat	0 or 1	This parameter allow you to choose a method of UDH-concatenation. If you set *two_byte_concat*=1 then 2-bytes reference number will be used, 1-byte reference number will be used otherwise. 2-bytes reference number could reduce a probability of incorrect concatenation in subscriber's mobile phone but also reduce a maximum length of one part of long message by 1 symbol. So maximum length for parts of such messages will be 152 or 66 symbols. Also see UDH-concatenation section for detailed description of UDH-concatenation. 0 is a default value.

Examples of responses

An XML with the result of SMS message queueing will be in a response. The examples below are formatted for better view so the value of "Content-Length" header can be different from actual length of formatted XML in example. Actually all the response body will be in one string.

An example of one single message sending:

Request

```
POST /multi.php HTTP/1.0
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 78
Connection: close

login=mylogin&password=mypassword&phones=78001234567&message=test+%F2%E5%F1%F2
```

Response

```
HTTP/1.1 200 OK
Server: nginx
Date: Tue, 11 Dec 2012 11:05:15 GMT
Content-Type: text/xml
Connection: close
Content-Length: 130

<?xml version="1.0" ?>
<reply>
  <result>OK</result>
  <code></code>
  <description>queued 1 messages</description>
</reply>
```

In case the request was successful the field *result* should contain "OK", field *code*=0, field *description* says the number of messages that were queueed. In case there was some error the field *result*=ERROR, field *code* > 0 and field *description* describes an error in text mode (for example, "authentication failed"). An error code = 1000 means that the problem is internal and temporary so you can try to repeat the request some time later or (better) to forward this request to reserve server. An example of response in case of error:

Request

```
POST /multi.php HTTP/1.0
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 78
Connection: close

login=mylogin&password=mypassword&phones=78001234567&message=test+%F2%E5%F1%F2
```

Response

```
HTTP/1.1 200 OK
Server: nginx
Date: Tue, 11 Dec 2012 11:05:15 GMT
Content-Type: text/xml
Connection: close
Content-Length: 130

<?xml version="1.0" ?>
<reply>
  <result>ERROR</result>
  <code>401</code>
  <description>login param is missing</description>
</reply>
```

If you request sms_id by additional parameter *want_sms_ids*=1 then there will be one more tag in response XML with the list of identifiers. Just now an identifier is a unsigned 8-bytes number. Example:

Request

```
POST /multi.php HTTP/1.0
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 78
Connection: close

login=mylogin&password=mypassword&want_sms_ids=1&phones=
79051112233,79261112233&message=test+%F2%E5%F1%F2
```

Response

```
HTTP/1.1 200 OK
Server: nginx
Date: Tue, 11 Dec 2012 11:05:15 GMT
Content-Type: text/xml
Connection: close
Content-Length: 130

<?xml version="1.0" ?>
<reply>
  <result>OK</result>
  <code></code>
  <description>queued 2 messages</description>
  <message_infos>
    <message_info>
      <phone>79051112233</phone>
      <sms_id>1000472891</sms_id>
    </message_info>
    <message_info>
      <phone>79261112233</phone>
      <sms_id>1000472892</sms_id>
    </message_info>
  </message_infos>
</reply>
```

If you add *with_push_id*=1 so some more info will appear in a response XML:

Request

```
POST /multi.php HTTP/1.0
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 78
Connection: close

login=mylogin&password=mypassword&want_sms_ids=1&with_push_id=1&individual_messages=1&
delimiter=ABC&phones=a:79051112233+<длинное сообщение>ABCb:79261112233+hello+test
```

Response

```
HTTP/1.1 200 OK
Server: nginx
Date: Tue, 11 Dec 2012 11:05:15 GMT
Content-Type: text/xml
Connection: close
Content-Length: 130

<reply>
  <result>OK</result>
  <code></code>
  <description>queued 3 messages</description>
  <message_infos>
    <message_info>
      <phone>79051112233</phone>
      <sms_id>8287366071</sms_id>
      <push_id>a</push_id>
    </message_info>
    <message_info>
      <phone>79051112233</phone>
      <sms_id>8287366073</sms_id>
      <push_id>a</push_id>
    </message_info>
    <message_info>
      <phone>79261112233</phone>
      <sms_id>8287366075</sms_id>
      <push_id>b</push_id>
    </message_info>
  </message_infos>
</reply>
```

You can find basic examples of programming code in different languages (PHP, Java, C++ etc) in our private web-interface. We don't recommend this code for commercial using as it was performed only as an example.

SMS message status checking

You have to call the same script api.smstraffic.ru/multi.php for checking SMS message status. For status checking you have to use following parameters:

```
login=your login
password=your password
operation=status
sms_id=list of sms identifiers comma separated
```

An XML will be returned as a result of request. A field *submition_date* means the date and time the messages was received by SMS Traffic platform. A field *send_date* means the date and time when the message was sent to operator. A field *last_status_change_date* describe the date and time when the status was changed last time.

last_status_change_date for Russian and Ukrainian phone numbers will be in MSK timezone. *last_status_change_date* for other phone numbers usually will be in subscriber local timezone but it can't be guarateed.

We save an information about message statuses for two days, after two days come we move them to the archive. If you try to ask the status of old message so you get an error "no such message or this message does not belong to you".

An example for one single message status:

Request

```
POST /multi.php HTTP/1.0
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 78
Connection: close

login=mylogin&password=mypassword&operation=status&sms_id=8287713
301
```

Response

```
HTTP/1.1 200 OK
Server: nginx
Date: Tue, 11 Dec 2012 11:05:15 GMT
Content-Type: text/xml
Connection: close
Content-Length: 130

<reply>
  <submition_date>2012-12-11 14:11:39</submition_date>
  <send_date>2012-12-11 14:11:39</send_date>
  <last_status_change_date>2012-12-11
14:12:00</last_status_change_date>
  <status>Delivered</status>
  <error></error>
  <sms_id>8287713301</sms_id>
</reply>
```

An example for several messages statuses:

Request

```
POST /multi.php HTTP/1.0
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 78
Connection: close

login=mylogin&password=mypassword&operation=status&sms_id=8287713301,8287713303,82877133031
```

Response

```
HTTP/1.1 200 OK
Server: nginx
Date: Tue, 11 Dec 2012 11:05:15 GMT
Content-Type: text/xml
Connection: close
Content-Length: 130

<reply>
  <sms>
    <error></error>
    <submition_date>2012-12-11 14:11:39</submition_date>
    <send_date>2012-12-11 14:11:39</send_date>
    <last_status_change_date>2012-12-11 14:12:00</last_status_change_date>
    <sms_id>8287713301</sms_id>
    <status>Expired</status>
  </sms>
  <sms>
    <error></error>
    <submition_date>2012-12-11 14:11:39</submition_date>
    <send_date>2012-12-11 14:11:40</send_date>
    <last_status_change_date>2012-12-11 14:12:00</last_status_change_date>
    <sms_id>8287713303</sms_id>
    <status>Delivered</status>
  </sms>
  <sms>
    <error>
      no such message or this message does not belong to you
    </error>
    <sms_id>82877133031</sms_id>
  </sms>
</reply>
```

Automatically status receiving method

You can receive every event of status changing from SMS Traffic platfrom. You have to set up a script on your web-server that could process the next POST parameters:

sms_id – identifier of the message
status – delivery status
delivery_date – date and time of delivery or date and time of last status changing

Identifier is an unsighned 8-byte number. Delivery date is in format "YYYY-MM-DD HH:MM:SS". You have to send us an URL of your scrypt and then we will set it up for calling after every status changing event.

Request

```
POST /status_callback_url HTTP/1.1
Connection: Close
User-Agent: Java/1.7.0_17
Host: example.com
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Content-type: application/x-www-form-urlencoded
Content-Length: 69

sms_id=15166254108&status=Delivered&delivery_date=2013-11-07
17:04:00
```

Response

```
HTTP/1.1 200 OK
Server: nginx/1.4.1
Date: Thu, 07 Nov 2013 13:05:08 GMT
Content-Type: text/plain
Content-Length: 0
Connection: close
```

Account balance checking

You have to call the same script api.smstraffic.ru/multi.php for checking account balance. For balance checking you have to use following parameters:

```
login=your login
password=your password
operation=account
```

Request

```
POST /multi.php HTTP/1.0
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 78
Connection: close

login=mylogin&password=mypassword&operation=account
```

Response

```
HTTP/1.1 200 OK
Server: nginx
Date: Tue, 11 Dec 2012 11:05:15 GMT
Content-Type: text/xml
Connection: close
Content-Length: 130

<?xml version="1.0" ?>
<reply>
  <account>10025</account>
</reply>
```

SMS message statuses

Value	Type
No status (blank string)	Transitional
Buffered SMSC	Transitional
Delivered	Final
Non Delivered	Final
Rejected	Final
Expired	Final
Deleted	Final
Unknown status	Final

Every message will get final status in 24 hours after it was sent.

Error codes for SMS Traffic HTTP(S) API

Code	Description
401	*login* parameter is missing
402	*password* parameter is missing
403	*phones* parameter is missing
404	Incompatible parameters
405	*message* parameter is missing
406	Wap push message is too long
407	*phones* parameter is empty
408	Unsupported message type: "*message_type*"
409	Missing UDH
410	*max_parts* cannot be specified
411	Authentication failed
412	Wrong IP
413	No such group: "*group_name*"
414	Group is empty
415	Not enough funds to complete the operation
416	Invalid start_date format: "*start_date*"
417	start date "*start_date*" is in the past
418	SMS ids could not be provided for scheduled messages
419	You are not allowed to use this route
420	Message "*message_text*" is too long
421	*originator* parameter is too long
422	Phone is missing in line "*line number*"*: string*
423	Message for phone "*phone number*" is empty
424	Message "*message text*" for phone "*phone number*" is too long
425	Phone "*phone number*" is too short. No messages has been sent
426	Phone "*phone number*" is too long. No messages has been sent

Code	Description
427	"*phone number*": wrong phone length. No messages has been sent
428	"*phone number*": wrong phone format. No messages has been sent
429	"*phone number*": unsupported phone. No messages has been sent
430	"*phone number*": invalid phone. No messages has been sent
431	No open subscriptions for +"*phone number*". No messages has been sent
432	Blocked phone: "*phone number*". No messages has been sent
433	*sms_id* parameter is missing
434	No such message or this message does not belong to you
435	Can't cancel message "*sms_id*"
436	Originator "*originator*" is forbidden
437	The message is longer than 160 characters after transliteration "*message text*"
438	Template found in message, but group is not set
439	You can't send SMS by HTTP
440	"*phones*" parameter is not set or set incorrectly
441	File format with parametes is wrong
442	Wrong number of parameters
501	End date of message sending is in the past
502	Start date of message sending is later than end date of message sending
1000	Temporary error

Subscribers lists managing

There is a possibility to manage your lists of subscribers. You have to call the script api.smstraffic.ru/list.php with parameters:

Parameter name	Possible values	Description
login	Text	Your login on SMS Traffic platform. Mandatory parameter.
password	Text	Your password on SMS Traffic platform. Mandatory parameter.
operation	status_all	Set this operation value to get info about all lists. Default value for parameter *operation*.
	status	An info about specified list.
	add_member	Adding phone numbers to the list specified by *group_id*.
	remove_member	Deleting phone numbers to the list specified by *group_id*.
member	List of phone numbers comma separated	This parameter is used for *operation*=add_member or remove_member. Not more than 5000 numbers in one request recommended to set.
group_id	Integer positive 4-byte number	Unique identifier of the list of subscribers. You can get all group_ids from request with *operation*=status_all

Examples

Getting an info about all lists:

Request

```
POST /list.php HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 54
Connection: close

login=mylogin&password=mypassword
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 09 Jun 2014 13:21:30 GMT
Server: Apache
Content-Length: 365
Connection: close
Content-Type: text/xml

<?xml version="1.0" ?>
<reply>
  <result>OK</result>
  <code></code>
  <description>total groups: 4</description>
  <groups>
    <group>
      <id>59353</id>
      <name>hello world</name>
      <created>2012-12-26 11:46:06</created>
      <congratulate></congratulate>
    </group>
    <group>
      <id>59355</id>
      <name>клиенты</name>
      <created>2012-12-26 11:48:15</created>
      <congratulate></congratulate>
    </group>
    <group>
      <id>59357</id>
      <name>test</name>
      <created>2012-12-26 11:48:49</created>
      <congratulate></congratulate>
    </group>
  </groups>
</reply>
```

Description of result fields:

id — unique identifier of subscribers list.
name — text name of the list. It is displayed in SMS Traffic web-interface.
created — date and time of the list creation in MSK timezone.
congratulate — status of the service of auto "happy bithday" congratulations.

Adding numbers to the list:

Request

```
POST /list.php HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 54
Connection: close

login=mylogin&password=mypassword&group_id=59353&
operation=add_member&member=79012223344,79082223344
```

Response

```

HTTP/1.1 200 OK
Date: Mon, 09 Jun 2014 13:21:30 GMT
Server: Apache
Content-Length: 365
Connection: close
Content-Type: text/xml

<?xml version="1.0" ?>
<reply>
  <result>OK</result>
  <code></code>
  <description>added or updated members: 2</description>
</reply>
```

Deleting numbers from the list:

Request

```
POST /list.php HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded
Content-Length: 54
Connection: close

login=mylogin&password=mypassword&group_id=59353&
operation=remove_member&member=79012223344,79082223344
```

Response

```

HTTP/1.1 200 OK
Date: Mon, 09 Jun 2014 13:21:30 GMT
Server: Apache
Content-Length: 365
Connection: close
Content-Type: text/xml

<?xml version="1.0" ?>
<reply>
  <result>OK</result>
  <code></code>
  <description>added or updated members: 2</description>
</reply>
```

SMPP protocol

SMS message sending, delivery reports receiving, subscribers generated SMS messages receiving can be perfomed by SMPP protocol that is standard for SMS messaging service. We use SMPP v3.4. You can download a specification from our web-site.

It is possible to limit an access to our SMSC by IP address. There is no any restrictions regarding what IP is used to connect our SMSC by default. If you want us to set such a restriction for your account you have to write to our support team and your SMPP connection will be limited by addresses you want to whitelist. We can whitelist not only a list of IPs but masks also (subnets of A, B, C classes that means /8, /16 and /24 по CIDR classification).

Any type of SMPP connection is supported with following restriction: not more than one receiver'a/transceiver for one our host + system-id. If you need more than one connection to one host so it is possible to get it with a set of system-ids of sub-accounts.

20 sms/sec is a default rate value but every client rate can be changed for its own needs. In case you send more than rate maximum you will receive an error ESME_RTHROTTLED in submit_resp packets.

Enquire_links are necessary. If your SMPP client connect as a transmitter so we wait enquire_links from the client and in case SMPP server don't see any enquire_link in 2 minutes the it breaks the connection. If your SMPP client connect as a receiver so our SMPP server sends enquire_links every 30 seconds. If server don't receive enquire_link_resp in 2 minutes it breaks the connection. If your SMPP client connect as a transceiver so enquire_links have to be sent by both sides. We recommend 30 second interval for sending enquire_links to our SMPP server.

There is no any special PDU features for our SMPP server. Query_sm, data_sm, submit_multi_sm, replace_sm, sar, tlv (for SUBMIT_SM), message_payload does not supported.

Bind parameters for SMPP connection to our SMSC:

Host	server1.smstraffic.ru (main server) or server2.smstraffic.ru (reserve server)
Port	4442

system_id	Login and password can be got from SMS Traffic manager. You can also use for other protocols and even for web-interface. Maximum length of login is 15 symbols and maximum length of password is 8 symbols (SMPP restrictions).
password	
interface_version	0x34
system_type	SMSC ignore these parameters, leave it blank
addr_ton	
addr_npi	
address_range	

We recommend to connect to main server1.smstraffic.ru and re-connect to reserve server2.smstraffic.ru in case of any problems.

Submit_sm parameters:

source_addr_ton	1 (for cifer sender) or 5 (for alpha-numeric sender)
source_addr_npi	1 (for cifer sender) or 0 (for alpha-numeric sender)
source_addr	Alpha-numeric sender (11 symbols length maximum) or cifer sender (15 symbols length maximum). See details in "Sender name (originator)".
dest_addr_ton	1
dest_addr_npi	1
destination_addr	Phone number in international format. See details in "Phone number format".
esm_class	Defined by SMPP specification
protocol_id	
validity_period	
registered_delivery	
data_coding	
sm_length	
short_message	

service_type	Do not supported
priority_flag	
schedule_delivery_time	
replace_if_present_flag	
sm_default_msg_id	

short_message field of DELIVER_SM format.

Field	Size (octets)	Type
Id	10	C-Octet String (Decimal)
sub	3	C-Octet String Fixed Length (Decimal)
dlvrd	3	C-Octet String Fixed Length (Decimal)
submit date	10	C-Octet String Fixed Length (Decimal)
done date	10	C-Octet String Fixed Length (Decimal)
stat	7	C-Octet String Fixed Length
err	var. max. 3	C-Octet String (Decimal)
text	var. max. 65	C-Octet String (Decimal)

SMSC adds two tlv tags to delivery report: message_state (0x0427) that contains a delivery status and receipted_message_id (0x001e) that contains a message identifier assigned by SMS Traffic platform. If delivery report is negative so TLV network_error_code (0x0423) also added. "Network_type" byte is set to 0x08.

Error codes list from field *err* of delivery report:

Code	English description
1	The subscriber is absent or out of a coverage
2	Call barred service activated (Could be made in case of negative balance)
3	Unknown subscriber
4	Memory capacity exceeded
5	Equipment protocol error
6	Teleservice not provisioned

Code	English description
7	Facility not supported
8	Subscriber is busy
9	Roaming restrictions
10	Timeout
11	SS7 routing error
12	Internal system failure
13	SMSC failure
14	Illegal subscriber
15	Message queue full

Encodings

We treat a Default SMSC alphabet (data_coding=0) as a "default GSM alphabet" (GSM 03.38). If you send in GSM 03.38 or latin1 (data_coding 0 or 3, respectively) only next symbols are supported (hex codes are for GSM encoding):

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
0.	@	£	$	¥	è	é	ù	ì	ò	Ç	LF	Ø	ø	CR	Å	å
1.		_											Æ	æ	ß	É
2.	SP	!	"	#	¤	%	&	'	(	)	*	+	,	-	.	/
3.	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4.	¡	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5.	P	Q	R	S	T	U	V	W	X	Y	Z	Ä	Ö	Ñ	Ü	§
6.	¿	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7.	p	q	r	s	t	u	v	w	x	y	z	ä	ö	ñ	ü	à

We suport correct processing only for symbols from an upper table if you use *data_coding*=0 or 3. We don't modify messages texts for other *data_coding* values.

You don't need to convert message text to 7bit pack for encoding GSM 03.38.

Common errors

There could different errors occur while you use SMPP connection. There is a list of most common errors you can face to and descriptions of its possible elimination.

ESME Already in Bound State (0x00000005)

This error ESME_RALYBND can appear if you try to make new transceiver or receiver connection to the host while you already have the same connection with same system_id.

If you need to have different connections to one our host it can be reached by creating additional sub-accounts that could be made by SMS Traffic manager. Additional sub-accounts use common billing but could have different routing rules and message priority.

Throttling error (0x00000058)

This error ESME_RTHROTTLED can occur when you try to send faster than maximum rate for current account. 20 sms per second is a default value.

You have to set up your SMPP client to re-send messages that were rejected with such error with tiny timeout waiting.

Invalid Password (0x0000000E)

An ESME_RINVPASWD can appear in BIND request processing if you set incorrect system-id or password or you have insufficient funds to continue to work.

You have to check if system-id and password are correct. Sometimes it can be correct but too long. SMPP v3.4. have restrictions for the length of system-id (15 symbols) and password 8 symbols). Then you have to check your account balance.

Invalid Source Address (0x0000000A)

An error ESME_RINVSRCADR can occur if you send message with incorrect sender. Also there could be some restrictions on your account regarding this sender.

You have to check if all limitations that were descrobed in "Sender name (originator)" are respected. Check for spaces in the beginning and the end of sender name. If everything seems to be correct then contact our manager for fixing the situation.

Invalid Dest Address (0x0000000B)

An error ESME_RINVDSTADR can occur if you send message to incorrect phone number. If phone number seems to be correct then whole the route can be blocked for your account.

You have to check if the phone number is correct and it is in an international format like in section Phone number format is described. Then you have to check if the operator this subscriber belong to is approved by agreement between SMS Traffic and your company. You can check an operator by phone number in SMS Traffic web-interface on Check mobile operator page.

UDH-concatenation

UDH-concatenation should be used for sending long multi-part messages. This is the only possible method for long messages termination in GSM networks. UDH is a part of message body (a part of field *short_message* from mandatory parameters of SUBMIT_SM packet) There are two types of UDH-concatenation that differ by one information element (IE) of UDH and reference number inside this IE.

Mostly used method is a method with 1-byte reference number:

05 00 03 D4 03 02 04 3F 04 40 04 38 04 32 04 35 04 42 00 20 00 3A 00 29 ...

length of UDH	Information Element Identifier(IEI)	length of IE	reference number	total number of parts	current part (count from 1)	message text part
05	00	03	D4	03	02	04 3F 04 40 04 38 04 32 04 35 04 42 00 20 00 3A 00 29 ...

Second method is to use 2-bytes reference number:

06 08 04 A7 D4 03 02 04 3F 04 40 04 38 04 32 04 35 04 42 00 20 00 3A 00 29 ...

length of UDH	Information Element Identifier (IEI)	length of IE	reference number	total number of parts	current part (count from 1)	message text part
06	08	04	A7 D4	03	02	04 3F 04 40 04 38 04 32 04 35 04 42 00 20 00 3A 00 29 ...

If you use 2-byte concatenation then the possibility of incorrect concatenation of long messages on subscribers mobile phone become really low. We have an experience when mobile phone have concatenated one part from current SMS message and second part from an old one SMS message. This can occur because of small number of reference number values for 1-byte concatenation. If you use 2-byte concatenation so you have to keep in mind that maximum number of symbols in one part of message will be lower: 152 symbols for latin messages and 66 symbols for UCS-2 messages.

You have to force 7 bit UDHI Indicator in esm_class parameter to indicate UDH in short_message. For example, 0x40 is a typical value for conacatenated messages.

Example of SMS message concatenated with two parts:

05 00 03 AF 02 01 ... message body ... — first part
05 00 03 AF 02 02 ... message body ... — second part

Reference number UDH parameter allow mobile phone to define different parts of the same concatenated message. Two last octets allow mobile phone to concatenate parts correctly.
There are different indication rules that depend on the mobile phone model. Usually there are two algorithm of concatenated message indication:

1. Some mobile phones indicate whole message only when every part of the message have arrived.
2. Some mobile phones indicate every arrived part and absent part is usually indicated in some special manner.

UDH is a part of message text so maximum size of every part of concatenated message become less for the size of UDH. For example maximum size of part for latin messages is 153 symbols for 1-byte reference number (152 symbols for 2-byte reference number), maximum size of part for UCS-2 messages is 67 symbols for 1-byte reference number (66 symbols for 2-byte reference number)

SOAP protocol

You can use our web-service to send SMS messages by calling appropriate methods of the web-service.

you can get WSDL here: https://soap.smstraffic.ru/soap.wsdl
Server address is: https://soap.smstraffic.ru/soap.php

All types and methods supported are described in WSDL. All parameters described in WSDL have the same behaviour as in HTTP(S) protocol.

There are description of next methods in WSDL:

- SendBulkSms — send one message to one or several phone numbers by one request
- SendIndividualSms — send individual messages to every subscriber
- GetSmsStatus — send request for a status of delivery
- GetBalance — send request of an account balance

SendBulkSms and SendIndividualSms methods are designed for sending SMS messages. Options tag must be in every query. Login and password elements are mandatory for the tag Options and other elements are optional:

Element	**Description**
Login	Your login in SMS Traffic system. This element is manadatory.
Password	Password from your account. This element is manadatory.
Originator	The name of sender. If not specified, the name will be taken from the default originator option in the account settings on SMS Traffic platform. More details can be found in the description of parameter originator from HTTP protocol.
WantSmsIds	If set to true, then there will be sms ids attached to each message in the response. Default value is false. More details can be found in the description of parameter *want_sms_ids* from HTTP protocol.
IgnorePhoneFormat	Turns off the phone correctness checking. Default value is false. More details can be found in the description of parameter *ignore_phone_format* from HTTP protocol.

Element	Description
LatinEncoding	If set to *true* then SMS Traffic platform treats message text encoding as latin or transliteration of ciryllic. In this case maximum length of one SMS message is 160 symbols or 153 symbols for one part of long concatenated message. If set to *false* then SMS Traffic platform treats message text encoding as UCS-2. In this case maximum length of one SMS message is 70 symbols or 67 symbols for one part of long concatenated message.
Gap	Desired interval in milliseconds between messages. Default value is 50. More details can be found in the description of parameter *gap* from HTTP protocol.
Timeout	SMS validity period in seconds. Default value is 86400 (24 hours). More details can be found in the description of parameter *timeout* from HTTP protocol.
Flash	Send sms as flash-sms. Default value is false. More details can be found in the description of parameter *flash* from HTTP protocol.
StartDate	Date and time of sending SMS. By default messages will be sent immediately. More details can be found in the description of parameter *start_date* from HTTP protocol.
MaxParts	Maximum number of parts that a long text message can be splitted on. Default value is 255. More details can be found in the description of parameter *max_parts* from HTTP protocol.
TwoByteConcat	This parameter allows to set the method of UDH for message concatenating. Default value is false. More details can be found in the description of parameter *two_byte_concat* from HTTP protocol.

A response of the query contains a Status tag always. This tag contains a status of processing the whole query:

Element	Description
IsSuccess	*true* if the query was processed successfuly, *false otherwise*.
Code	An error code of the response. If it is equal to 0 then the query was processed without any errors. The list of possible error codes are similar to list of HTTP error codes, see the description in HTTP protocol.

Element	Description
Description	Текстовое описание ошибки. Значение этого элемента полезно записать в логи.

SendBulkSms method

SendBulkSms method allows to send the same text message to a list of phone numbers.

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns="http://soap.smstraffic.ru/soap.wsdl">
  <env:Body>
    <BulkSms>
      <Options>
        <Login>example</Login>
        <Password>p@ssw0rd!</Password>
      </Options>
      <Message>hello world</Message>
      <Phone>78003336655</Phone>
      <Phone>78002224477</Phone>
      <Phone>78001114444</Phone>
    </BulkSms>
  </env:Body>
</env:Envelope>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns="http://soap.smstraffic.ru/soap.wsdl">
  <env:Body>
    <SmsResponse>
      <Status>
        <IsSuccess>true</IsSuccess>
        <Code></Code>
        <Description>queued 3 messages</Description>
      </Status>
    </SmsResponse>
  </env:Body>
</env:Envelope>
```

SendIndividualSms method

SendIndividualSms method allows to send a message with different text to each subscriber.

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns="http://soap.smstraffic.ru/soap.wsdl">
  <env:Body>
    <IndividualSms>
      <Options>
        <Login>example</Login>
        <Password>p@ssw0rd!</Password>
      </Options>
      <Pairs>
        <Phone>78003336655</Phone>
        <Message>individual</Message>
      </Pairs>
      <Pairs>
         <Phone>78002224477</Phone>
         <Message>message</Message>
      </Pairs>
      <Pairs>
        <Phone>78001114444</Phone>
        <Message>by phone</Message>
      </Pairs>
    </IndividualSms>
  </env:Body>
</env:Envelope>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns="http://soap.smstraffic.ru/soap.wsdl">
  <env:Body>
    <SmsResponse>
      <Status>
        <IsSuccess>true</IsSuccess>
        <Code></Code>
        <Description>queued 3 messages</Description>
      </Status>
    </SmsResponse>
  </env:Body>
</env:Envelope>
```

GetSmsStatus Method

GetSmsStatus method returns the delivery status of a message by sms_id.

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns="http://soap.smstraffic.ru/soap.wsdl">
  <env:Body>
    <StatusRequest>
      <Login>example</Login>
      <Password>p@ssw0rd!</Password>
      <SmsId>27371658818</SmsId>
      <SmsId>1</SmsId>
    </StatusRequest>
  </env:Body>
</env:Envelope>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns="http://soap.smstraffic.ru/soap.wsdl">
  <env:Body>
    <StatusResponse>
      <Status>
        <IsSuccess>true</IsSuccess>
        <Code></Code>
        <Description></Description>
      </Status>
      <SmsInfo>
        <SmsId>27374567618</SmsId>
        <Error></Error>
        <SubmissionDate>2015-04-02T15:19:00</SubmissionDate>
        <SendDate>2015-04-02T15:19:00</SendDate>
<LastStatusChangeDate>2015-04-02T15:19:00</LastStatusChangeDate>
        <Status>Expired</Status>
      </SmsInfo>
      <SmsInfo>
        <SmsId>1</SmsId>
        <Error>no such message or this message does not belong to
you</Error>
      </SmsInfo>
    </StatusResponse>
  </env:Body>
</env:Envelope>
```

GetBalance Method

GetBalance method returns current account balance.

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns="http://soap.smstraffic.ru/soap.wsdl">
  <env:Body>
    <BalanceRequest>
      <Login>texample</Login>
      <Password>p@ssw0rd!</Password>
    </BalanceRequest>
  </env:Body>
</env:Envelope>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns="http://soap.smstraffic.ru/soap.wsdl">
  <env:Body>
    <BalanceResponse>
      <Status>
        <IsSuccess>true</IsSuccess>
        <Code></Code>
        <Description></Description>
      </Status>
      <Amount>52782.35</Amount>
    </BalanceResponse>
  </env:Body>
</env:Envelope>
```

SMTP protocol (version 1)

If you want to send SMS message via email you have to write (or generate) an email to *login*@corp.smsmail.ru, where login is your login to SMS Traffic platform. Phone number have to be set as a subject of an email and SMS message text - in a body of email. You can write message text in cyrillic and the text will be transliterated or sent in UCS-2 depending on your settings.

To see and change your settings please visit our web-interface on lk.smstraffic.ru/options.php.

Example 1:

```
From: yourname@yourdomain.ru
To: yourlogin@corp.smsmail.ru
Subject: 79161234567

Vam neobhodimo oplatit’ schet $17.65. Dlya spravok zvonite
1234567.
```

A phone number of the subscriber can be set in three ways:

- In email subject. In this case you have to send an email to *login*@corp.smsmail.ru as described earlier
- In recipient address like 79161112233.*login*@corp.smsmail.ru

Example 2:

```
From: yourname@yourdomain.ru
To: 79161234567.yourlogin@corp.smsmail.ru
Subject: ne imeyet znacheniya

Vam neobhodimo oplatit’ schet $17.65. Dlya spravok zvonite
1234567.
```

SMTP v.1 wasn't designed for intensive bulk messaging. For big volumes of SMS messages please use SMPP or HTTP(S) protocols or our web-interface. 50 email per second is a recommended rate maximum for this protocol.

We recommend you to secure your account from an unapproved usage by others. You can set SMTP-header protection (recommended) or input a limited set of phone numbers allowed to be passed as a recipient of an SMS. We can authorize you by

Received headers your SMTP server set. You can send an email to tech.support@smsmail.ru through your SMTP server with short description: your login and your preferred security choice - by headers. Then we will set your Received header as a trusted one.

If you prefer to secure by setting the list of recipients then you cen set it on lk.smstraffic.ru/options.php.

We recommend you to use at least one of protection type for SMTP v.1. Also you can switch off even a possibility of SMTP v.1. using, this also can be done on lk.smstraffic.ru/options.php. SMTP v.1. is switched off by default.

There is no possibility to set sender (originator) using SMTP v.1 so the default value for your account will be used as a sender. You can see and change value of default sender on our web-interface settings page.

SMTP protocol (version 2)

There is another way to send SMS messages via email. In second version you can use all possibilities that HTTP can grant.

You have to send an email to multi@smtp2.smsmail.ru. Subject of letter does not matter, in the body of email you should write pairs “parameter: value”, where parameters could be taken from HTTP description. Every pair should be on one individual string and separated by blank one.

Example:

```
From: yourname@yourdomain.ru
To: multi@smtp2.smsmail.ru
Subject: ne imeyet znacheniya

login: yourlogin

password: yourpass

phones: 79161234567

message: Hello mister Smith.
Thank you for the registration and welcome to our service.
Please call to 447712345678 for details.

originator: MyCompany
```

Note: you can’t use two “end of string” symbols one after another in field *message*.

You can also use an additional field *reply_to_email* with value equal to 1 as in an example below:

```
From: yourname@yourdomain.ru
To: multi@smtp2.smsmail.ru
Subject: ne imeyet znacheniya

login: yourlogin

password: yourpass
```

```
phones: 79161234567

message: Vam neobhodimo oplatit’ schet $17.65. Dlya spravok
zvonite 1234567.

originator: MyCompany
```

`reply_to_email: 1`

If you set this parameter then we send an answer e-mail back to sender address. The text of the answer will be exactly the same if an HTTP(S) protocol was used. Example:

```
<?xml version="1.0"?>
<reply>
  <result>OK</result>
  <code></code>
  <description>queued 1 messages</description>
  <message_infos>
    <message_info>
      <phone>79161234567</phone>
      <sms_id>1014190631</sms_id>
    </message_info>
  </message_infos>
</reply>
```

It is recommended to use *reply_to_email* parameter while testing SMTP v.2 protocol. If you don't set *reply_to_email* in a request email then no answer will be sent.

SMTP v.2 wasn't designed for intensive bulk messaging. For big volumes of SMS messages please use SMPP or HTTP(S) protocols or our web-interface. 50 email per second is a recommended rate maximum for this protocol and 10000 SMS messages is recommended maximum for one letter.

Mobile Virtual Numbers (Incoming Numbers)

If you have a MVN service activated then you can receive messages from your subscribers (MO messages, mobile originated messages). After service activating you will get a special incoming number that was set up on operator's SMSC to forward all SMS messages received to SMS Traffic platform. After MO messages being received by SMS Traffic platform they can be sent to your email address or via HTTP or SMPP protocols.

Forwarding MO messages to email

You have just to send us your email address you wish to receive MO messages and we will register it in your settings. You can also change an address by yourself on the page lk.smstraffic.ru/en/options.php.

Forwarding MO messages to HTTP script

You have to create a script on your web-server which have to process GET and POST requests from our SMSC with parameters:

```
phone – subscriber phone number
message – SMS message text (Windows-1251 encoding by default)
sms_id – unique id of MO message (for avoiding the duplicating on
the client side; if you receive an id you'd got already you have
to answer exactly the same when you got it first time).
```

Your script have to answer "OK" with response code 200.

You can edit scripts parameters by yourself on Settings section of our web-interface. In case you wish to use POST method for receiving data you have to set up just an URL of your script. In case you wish to use GET method for receiving data you have to set up a template of format:
URL_of_your_script?message={{message}}&phone={{phone}}&sms_id={{sms_id}}

Also you can simply send us an URL of the script and we will register it on SMS Traffic platform.

Forwarding MO messages via SMPP

You can also send MO messages via SMPP protocol. You have to switch on a corresponding option on Settings section of our web-interface You can receive MO messages only if you have set an SMPP connection as a Receiver or Transceiver.

Request a list of MO messages

You can get all MO messages by calling our script api.smstraffic.ru/multi.php with parameters:

```
login=your login
password=your password
operation=incoming

optional parameters:
from_date=starting date and time of period requested, MSK
timezone, format – «YYYY-MM-DD» или «YYYY-MM-DD HH:MM:SS». 2 days
is maximum period you can request.
from_phone=phone number
count=maximum number of MO messages in one request
want_sms_ids=1 – back XML will contain tag with SMS identifiers
```

As a response you get an XML with the data about MO messages.

```
<?xml version="1.0" ?>
<reply>
  <count>125</count>
  <messages>
    <message>
      <from_phone>79031234567</from_phone>
      <text>sms message text</text>
      <send_date>2007-02-12 01:12:59</send_date>
      <delivery_date>2007-02-12 01:13:05</delivery_date>
      <sms_id>5269156759</sms_id>
    </message>
    ...
    <message>
      <from_phone>79037654321</from_phone>
      <text>sms message</text>
      <send_date>2007-02-11 06:02:13</send_date>
      <delivery_date>2007-02-12 06:02:18</delivery_date>
      <sms_id>5269156775</sms_id>
    </message>
  </messages>
```

```
</reply>
```

We recommend you to use possibilities described in "Forwarding MO messages to HTTP script" section if you wish to get MO messages immediately.

Shortcode

You have to create a script on your web-server which have to process GET and POST requests from our SMSC with parameters:

```
phone – subscriber phone number
message – SMS message text (Windows-1251 encoding by default)
shortcode – shortcode the subscriber sent message to
password – password to secure the request
sms_id – unique id of MO message (for avoiding the duplicating on
the client side; if you receive an id you'd got already you have
to answer exactly the same when you got it first time).
```

When we receive an MO message on your shortcode we call your script at the moment. The request has to be processed in short time and your script has to answer with message that we will send back to subscriber. There answer have to be clear text, no formatting is applying here.

Also you can simply send us an URL of the script and we will register it on SMS Traffic platform for your shortcode.

You can edit scripts parameters by yourself on Settings section of our web-interface. In case you wish to use POST method for receiving data you have to set up just an URL of your script. In case you wish to use GET method for receiving data you have to set up a template of format:
URL_of_your_script?message={{message}}&phone={{phone}}&sms_id={{sms_id}}&shortcode={{shortcode}}

An example of PHP script can be found on https://lk.smstraffic.ru/doc/samples.zip in shortcode directory.

Document history

Date of change	Version	Description
2015-12-10	1.85	All document structure has been re-designed. New SOAP protocol description. New URLs for HTTP API.
2012-09-25	1.84	A section "Automatically status receiving method" is added.
2012-06-08	1.81	*want_sms_ids* parameter is added for operation=incoming in HTTP(S) API description.
2012-03-23	1.80	SMPP detailed description.
2012-01-22	1.77	server2.smstraffic.ru set as a main host for SMPP protocol.
2011-12-29	1.76	Several new error codes were added for HTTP(S)API.
2011-10-04	1.74	timeout parameter was described for HTTP(S) API.
2011-09-27	1.72	Restrictions for SMTP protocols were described. Error code 440 was added for HTTP(S) protocol.
2011-08-12	1.69	*want_sms_ids* parameter description was updated for HTTP(S) protocol.
2010-12-14	1.63	SMPP hosts were set instead of IP addresses.
2010-01-11	1.55	New error codes list were created for HTTP(S) protocol.
2009-07-06	1.50	SMPP protocol description was updated.
2009-05-14	1.48	SOAP protocol description
2009-03-31	1.47	IP addresses were changed for SMPP protocol.
2008-12-02	1.45	rus=5 value description was added.
2007-11-23	1.39	*flash* parameter description for HTTP(S) protocol was added.
2007-10-28	1.37	Different examples were added.
2007-07-18	1.35	Destination address was changed to multi@smtp2.smsmail.ru for SMTP v.2.
2007-06-8	1.31	*start_date* parameter was added for HTTP(S) API description.
2007-05-29	1.28	Reserve server description was added.

Date of change	Version	Description
2006-10-24	1.19	MO messages requesting via HTTP(S) possibility was added.
2006-08-15	1.18	Account balance requesting vai HTTP(S) was added.
2006-06-30	1.13	*originator* parameter description was added for HTTP(S) protocol. SMTP v.2 was added.
2006-06-28	1.9	*gap* parameter was added to HTTP(S) protocol.
2006-03-23	1.6	*max_parts* and *udh* parameters were added to HTTP(S) protocol.
2005-02-21	1.3	Additional parameters for HTTP(S) protocol were added.
2004-08-02	1.1	Date of document creation