



Documentation for SMSAPI.com (REST API)

Ver. 1.2



TABLE OF CONTENTS

1. Introduction

1.1 How to start

1.2 IP filter for API interface

1.3 API password

I. SMS SENDING

2. Single SMS Message sending

2.1 Sending Fast messages (with highest priority)

2.2 Scheduled SMS sending

2.3 Deleting single scheduled messages

3. Bulk SMS sending

3.1 Sending SMS messages to numbers from contacts database group

3.2 Bulk personalized SMS sending using parameters

3.3 Bulk SMS sending using IDX parameter

3.4 Sending messages with cut.li

4. Sending messages using templates

5. mail2SMS – Sending SMS using e-mail

6. Account balance checking

IV. CALLBACK

7. SMS delivery confirmation receiving – CALLBACK procedure

8. Bulk CALLBACK

9. cut.li CALLBACK

VI. HLR – Number activity checking

10. HLR Lookup

VII. LAST SENTENCE

11. Last sentence

Appendix 1 – Delivery reports list

Appendix 2 – Error codes

Appendix 3 – Encoding

Appendix 4 – Example scripts

History of changes



1. Introduction

SMSAPI is a high quality SMS platform which enables you to integrate any of your applications with our SMS message sending and receiving system. The main advantage of our system is its simplicity of implementation. The SMS message may have your company name or any phone number that you owns as sender name. Every message sent from our system has its own unique id which allows you to receive its delivery confirmation.

1.1 How to start

To start using SMSAPI you need to create an account on our website: <http://www.smsapi.com/> Registration is completely free of charge. Your account is ready to use just after registration, but we recommend you to verify at least one sender name or number. Every message sent before validation has the „INFO“ as a default name.

1.2 IP filter for API interface

In order to improve interface API protection you may set list of IP addresses which will be whitelisted. You may do it in „SETTINGS“ → „API“ tab „IP addresses filter“ in field **IP addresses filter**. It will be possible to send messages only from these whitelisted IP addresses (attempts of sending from other IPs will result in responding: ERROR:105). IP addresses should be separated by semicolon.

1.3 API password

API password is the same as web panel password after registration. You may change your API password in Settings → API. Changing your web panel password **does not** change API password.

2. Single SMS Message sending

URL links needed for connecting with our application:

- <https://api.smsapi.com/sms.do> – for SSL secured connections
- <https://api2.smsapi.com/sms.do> – backup for SSL secured connections
- <http://api.smsapi.com/sms.do> – for standard connections (**Unrecommended!**)
- <http://api2.smsapi.com/sms.do> – backup for standard connections (**Unrecommended!**)

Messages should be sent as a HTTP GET or POST request to our system:

Parameter	Description
<i>username</i> *	Username used to identify a user in our system
<i>password</i> *	Password to Your account hashed with MD5
<i>to</i>	Recipient's mobile phone number (i.e. 44123456789).
<i>group</i>	Name of the group from the contacts database to which message should be sent.
<i>message</i> *	The message text. Content of one message is normally 160 characters per single SMS or 70 in case of using at least one special character (polish characters are considered to be special characters). The maximal message is set to 918 normal characters or 402 if special chars are used and it is being sent as one block of 6 messages joined together and charged as six messages. Detailed information about special characters are given in chapter 9.
<i>from</i>	Name of the sender. As a default the sender name is set to „INFO“. Only verified names are being accepted (&from=active_name). Sender name may be set after logging into web panel on https://ssl.smsapi.com/ in SETTINGS → SENDER NAMES.
<i>encoding</i>	This parameter describes the encoding of the message text. UTF-8 is set as default. If another encoding is needed parameter encoding should have following value: for iso-8859-2 (latin2) – it should be &encoding=iso-8859-2 for Windows-1250 – it should be &encoding=windows-1250
<i>flash</i>	Sending a message in flash mode can be activated by setting this parameter to „1“. Flash SMS are automatically presented on the mobile screen and have to be manually saved to be stored in inbox. (&flash=1)



<i>test</i>	When parameter <i>test</i> is set to „1” message won't be sent but response will be displayed, there is no charge for such test messages. (&test=1)
<i>details</i>	When <i>details</i> parameter is set to „1” more details in response will be displayed (message length and sms count). (&details=1)
<i>date</i>	Date in UNIX timestamp (&date=1287734110) or in ISO 8601 (&date=2012-05-10T08:40:27+00:00) when message will be sent (&date=1287734110). Setting a past date will result in sending message instantly.
<i>date_validate</i>	When parameter <i>date_validate</i> is set to „1” checks if date if given in proper format. Returns ERROR:54 if not.
<i>datacoding</i>	This parameter allows to send binary messages. (&datacoding=bin) Parameters udh and message should be given as HEX (for example message=616263 if you want to send abc).
<i>udh</i>	UDH header required when parameter &datacoding=bin is used, for WAP PUSH messages udh should have value &udh=0605040b8423f0
<i>allow_duplicates</i>	When parameter <i>allow_duplicates</i> is set to „1” allows to send message to duplicated numbers in one request (usefull i.e. for parametrized message contents)
<i>idx</i>	Optional custom value sent with SMS and sent back in CALLBACK (i.e. &idx=123).
<i>check_idx</i>	When parameter <i>check_idx</i> is set to „1” prevents from sending more than one message with the same idx in last 24h. When this parameter is set and message with the same idx was already sent error 53 is returned.
<i>max_parts</i>	Defines maximum message parts allowed, maximum value allowed is 6. ERROR: 12 will be returned when the message has more parts than defined. Default value can be set in customers panel.
<i>nounicode</i>	When parameter <i>nounicode</i> is set to „1” prevents from sending messages containing special characters. ERROR: 11 will be returned when the message contains special characters.
<i>normalize</i>	When parameter <i>normalize</i> is set to „1” special chars in message will be replaced with their equivalents (ê-e, ñ-n, ý-y ...).
<i>fast</i>	Setting this parameter to „1” will result in sending message with the highest priority which ensures the quickest possible time of delivery. Fast messages costs 50% more than normal message. Attention! Mass and marketing messages must not be sent with fast parameter.
<i>expiration_date</i>	Message expiration date (in UNIX timestamp or in ISO 8601) is a date after which message won't be delivered if it wasn't delivered yet. The difference between date sent and expiration date shouldn't be less than 15 minutes and more than 72 hours (we recommend using minimum 1 hour and maximum 12 hours difference). Time will be set with tolerance +/- 5 minutes.
<i>notify_url</i>	Parameter allows to set CALLBACK URL for message from request. This parameter may be used when there is no default CALLBACK URL for this user or when it should be different than default one (notify_url has higher priority than default callback).
<i>format</i>	Parameter &format=json causes, that response is sending in JSON format.

* - required field

ATTENTION! Parameters **group** and **to** are exchangeable, one of these two parameter has to appear in a request. Lack of any or appearing both of them will result in returning ERROR:13.

Request: **https://api.smsapi.com/sms.do?username=username&password=passwordMD5&from=sender_name&to=44123456789&message=message_content&format=json**



I. Examples of responses when **&format=json** parameter used:

a) in case of success:

```
{
  "count": 1,
  "list": [
    {
      "id": "1460969715572091219", //message id
      "points": 0.16, //price of delivery
      "number": "44123456789", //recipient number with country prefix
      "date_sent": 1460969712, //send date
      "submitted_number": "44123456789", //phone number in request
      "status": "QUEUE" //message status
    }
  ]
}
```

b) in case of failure:

```
{
  "invalid_numbers": [
    {
      "number": "44123456789", //recipient number with country prefix
      "submitted_number": "44123456789", //phone number in request
      "message": "Invalid phone number" //error description
    }
  ],
  "error": 13, //error code
  "message": "No correct phone numbers" //error description
}
```

II. Examples of responses without **&format** parameter:

a) in case of success:

Response: **OK:<ID>:<POINTS>**
 example: **OK:17101000090360359:0.165**

b) in case of failure:

Response: **ERROR:<ERR>**
 example: **ERROR:102h**

Where:

<ID> Message unique ID. You will need it for delivery confirmation.
 Length of this parameter in each request may vary, but it won't be more than 32 characters long.

<POINTS> Amount of used credits (i.e. Text sent in 3 messages will return 3xSMS amount)

<ERR> Error code (check the error code list in Appendix 2)

In case of any problems with sending requests to basic URL (as in example above) backup URL may be used:
<https://api2.smsapi.com/>

2.1 Sending Fast messages (with highest priority)

Setting parameter fast to „1” (&fast=1) will result in sending message with the highest priority which ensures the quickest possible time of delivery. Fast messages costs 50% more than normal message.



Request: **[https://api.smsapi.com/sms.do?
username=username&password=passwordMD5&from=sender_name&to=4412345678
9&fast=1&message=Fast message content&format=json](https://api.smsapi.com/sms.do?username=username&password=passwordMD5&from=sender_name&to=44123456789&fast=1&message=Fast message content&format=json)**

Response: **OK:<ID>:<POINTS>**

or (when error occur)

ERROR:<ERR>

<ID> Message unique ID. You will need it for delivery confirmation.
Length of this parameter in each request may vary, but it won't be more than 32 characters long.

<POINTS> Amount of used credits (i.e. Text sent in 3 messages will return 3xSMS amount)

<ERR> Error code (check the error code list in Appendix 2)

Example: **OK:17101000090360359:0.70**

or

ERROR:102

ATTENTION! Mass and marketing messages mustn't be sent with fast parameter. In case of sending message to more than one recipient in a single request messages will be sent as normal ones, without parameter &fast=1.

In case of any problems with sending requests to basic URL (as in example above) backup URL may be used:
<https://api2.smsapi.com/>

2.2 Scheduled SMS sending

To send message at specified date and hour parameter **date** has to be used. This parameter should be in UNIX timespamp format.

Request: **[https://api.smsapi.com/sms.do?
username=username&password=passwordMD5&to=44123456789&date=1577878200
&message=scheduled message content&format=json](https://api.smsapi.com/sms.do?username=username&password=passwordMD5&to=44123456789&date=1577878200&message=scheduled message content&format=json)**

Response: **OK:<ID>:<POINTS>**

or (when error occur)

ERROR:<ERR>

In case of any problems with sending requests to basic URL (as in example above) backup URL may be used:
<https://api2.smsapi.com/>



2.3 Deleting single scheduled messages

Parameter	Description
<i>username</i> *	Username used to identify a user in SMSAPI system.
<i>password</i> *	Password to Your account hashed in MD5
<i>sch_del</i> *	ID of message to delete (returned after sending one), for message to be deleted separate IDs with comma.

* - required field

Request: **[https://api.smsapi.com/sms.do?username=username&password=passwordMD5&sch_del=09040616088106874&format=json](https://api.smsapi.com/sms.do?username=<u>username</u>&password=<u>passwordMD5</u>&sch_del=<u>09040616088106874</u>&format=json)**

I. Examples of responses when **&format=json** parameter used:

a) in case of success:

```
{
  "count": 1,           //quantity of messages to delete
  "list": [
    {
      "id": "1462974172199599210" //message id to delete
    }
  ]
}
```

b) in case of failure:

```
{
  "error": 301,           //error code
  "message": "Not exists ID message" //error description
}
```

II. Examples of responses without **&format** parameter:

Response: **OK**

or (when ID od message doesn't exist)

ERROR:301

Example ID: 09040616088106874

In case of any problems with sending requests to basic URL (as in example above) backup URL may be used:
<https://api2.smsapi.com/>

3. Bulk SMS sending

Sending messages to a group of recipients is similar to single submissions (presented in chapter 2). The only difference is filling field **to** with set of many recipients' numbers (not only one number). In order to successfully send this type of message, we recommend to pass all parameters in a **HTTP POST** request just to assure that all recipients numbers will be submitted correctly.

If the total cost of sending these messages is higher than the number of available credits on user's account, the system will respond **103** error code and none of messages will be sent. If some of given recipients' numbers are invalid (unrecognized by SMSAPI due to wrong prefix or landline number) than these numbers will be skipped and not include in bulk, messages will be sent to the rest of numbers. Delivery reports will not concern the skipped numbers. If any number will appear more than once in one request message will be sent only once to this recipient.



Request: **https://api.smsapi.com/sms.do?
username=username&password=passwordMD5&from=sender_name
&to=48500500500,48501501501,48502502502&message=message&format=json**

I. Examples of responses when **&format=json** parameter used:

a) in case of success:

```
{
  "count": 3,
  "list": [
    {
      "id": "1460978572913968440",
      "points": 0.16,
      "number": "48500500500",
      "date_sent": 1460978579,
      "submitted_number": "48500500500",
      "status": "QUEUE"
    },
    {
      "id": "1460978572913968450",
      "points": 0.16,
      "number": "48501501501",
      "date_sent": 1460978579,
      "submitted_number": "48501501501",
      "status": "QUEUE"
    },
    {
      "id": "1460978572913968460",
      "points": 0.16,
      "number": "48502502502",
      "date_sent": 1460978579,
      "submitted_number": "48502502502",
      "status": "QUEUE"
    }
  ]
}
```

b) in case of failure:

```
{
  "invalid_numbers": [
    {
      "number": "456456456",
      "submitted_number": "456456456",
      "message": "Invalid phone number"
    },
    {
      "number": "321321321",
      "submitted_number": "321321321",
      "message": "Invalid phone number"
    }
  ],
  "error": 13,
  "message": "No correct phone numbers"
}
```

II. Examples of responses without **&format** parameter:

a) in case of success

Response: **OK:<ID>:<POINTS>:<PHONE>;...;...;...**

example response:

OK:17101000090567759:0.14:500500500;OK:171010000903455357:0.14:501500501;



b) in case of failure:

Response: ERROR:<err>

where:

<ID> Message unique ID. You will need it for delivery confirmation.
Length of this parameter in each request may vary, but it won't be more than 32 characters long.

<POINTS> Number of used credits (i.e. Text sent in 3 messages will return 3xSMS amount)

<PHONE> Recipient's phone number

Example:

OK:17101000090567759:0.14:500500500;OK:171010000903455357:0.14:501500501;
OK:17101000096577326:0.14:502502502;

Notice that in multiple SMS sending recipients' numbers are in response as well, and all message data are separate by a semicolon (after the last message there is semicolon as well).

Recommended maximum number of messages sent in one request for POST method is 10000, for GET method it is 200 messages.

In case of any problems with sending requests to basic URL (as in example above) backup URL may be used:
<https://api2.smsapi.com/>

3.1 Sending SMS messages to numbers from contacts database group

It is possible to send messages to group of numbers from contacts database. In order to do that such group should be first created in web panel menu contacts database.

You can insert own custom field into message contents. Custom fields can be defined in Contacts database → Custom fields tab. To insert custom field into message use [%contact.field name%]. This expression will be replaced with corresponding value assigned to contact.

An example of sending messages to group from contacts database is like follows:

Request: [https://api.smsapi.com/sms.do?username=username&password=passwordMD5&group=test_goup&message=Test message sent to contacts from contacts database - example custom field: \[%contact.field name%\]](https://api.smsapi.com/sms.do?username=username&password=passwordMD5&group=test_goup&message=Test message sent to contacts from contacts database - example custom field: [%contact.field name%])

Response: OK:<ID>:<POINTS>:<PHONE>;...;...;...

<ID> Message unique ID. You will need it for delivery confirmation.
Length of this parameter in each request may vary, but it won't be more than 32 characters long.

<POINTS> Number of used credits (i.e. Text sent in 3 messages will return 3xSMS amount)

<PHONE> Recipient's phone number

Example: OK:17101000090567759:0.14:500500500;OK:171010000903455357:0.14:501500501;
OK:17101000096577326:0.14:502502502;

In case of any problems with sending requests to basic URL (as in example above) backup URL may be used:
<https://api2.smsapi.com/>

3.2 Bulk personalized SMS sending using parameters

There is possibility to send up to 100 personalized messages in one request using personalization parameters. To send more personalized messages more than one request has to be used. Personalization parameters should be defined in request as **param1**, **param2**, **param3**, **param4**, which will replace tags [%1%], [%2%], [%3%] and [%4%] in message content. Values of these parameters have to be separated by pipe char „|” according to the template below:

param1=Ann|Michael|Andrew¶m2=Smith|Thomas|Davis



The number of parameters has to be exactly the same as number of recipients in a request otherwise ERROR: 18 will be returned and message won't be sent.

IMPORTANT! Length of message may be different depending on the length of parameter value.

If one of numbers will be invalid message to this number will be skipped and the rest will be sent.

Parameters

After defining parameters they may be used in message content:

Parameter	Description
[%1%]	Value of parameter 1 (param1)
[%2%]	Value of parameter 2 (param2)
[%3%]	Value of parameter 3 (param3)
[%4%]	Value of parameter 4 (param4)

Example:

```
https://api.smsapi.com/sms.do?
username=username&password=passwordMD5&from=sender_name&to=4860011122
2,4850011122&message=Message content, parametr1: [%1%] parametr2: [%2%]
&param1=John|Ann&param2=30|40
```

Message will have following contents:

Message 1 : Message content, parametr1: **John** parametr2: **30**
 Message 2 : Message content, parametr1: **Ann** parametr2: **40**

In case of any problems with sending requests to basic URL (as in example above) backup URL may be used:
<https://api2.smsapi.com/>

3.3 Bulk SMS sending using IDX parameter

There is possibility to send mass messages with custom parameter **IDX** different for each message. This parameter will be then returned in CALLBACK. With parameter **idx** additional parameter **check_idx** may be used (&check_idx=1). Using check_idx parameter prevents from two message with the same idx parameter value in last 24h. Example of using idx parameter:

```
idx=idx1|idx2|idx3|idx4
```

Number of **IDX** parameters has to be equal to the number of recipients' numbers given in request.

In case of any problems with sending requests to basic URL (as in example above) backup URL may be used:
<https://api2.smsapi.com/>.

3.4 Sending messages with cut.li

This function allows to use in SMS messages the shortened URL - <http://cut.li>, which will redirect the message recipient to URL which will be added as parameter. In the SMS message, the URL should be added as parameter [%goto:url_address%] will be visible eg. as <http://cut.li/ABCD> (each recipient will receive unique end of link "/ABCD").

Example:

```
https://api.smsapi.com/sms.do?
username=username&password=passwordMD5&from=sender_name
&to=49500500500,42501501501,47502502502&message=test message with short
link: [%goto:www.smsapi.com%] end.
```



In case of any problems with sending requests to basic URL (as in example above) backup URL may be used:
<https://api2.smsapi.com/>.

4. Sending messages using templates

Using templates simplifies changing standard notification messages (may be used in shops, internet services, medical clinics etc.) without changing php script that implements SMS sending.

To use templates You should:

- After logging on <https://ssl.smsapi.com/> add template in „SMS messages“ → „Templates“ menu
- Places that should be replaced by a parameter should be given [%N%] where N is number between 1 and 4 (parameter number)
- To use a template in API request there should appear &templates=template_name in the request
- Apart from all basic parameters while using templates following parameters are available:

Parameter	Description
<i>template</i>	Template name
<i>paramN</i>	The value of this parameter will replace [%N%] in the template where N is a number between 1 and 4
<i>single</i>	If the message will contain more than 160 chars (single message) it won't be sent and ERROR:12 will be replied (&single=1)

Example:

Template name: **Notify**
 Template content: *Hello [%1%], Your order has been sent. The shipment number is [%2%] You may follow it on our site.*

**[https://api.smsapi.com/sms.do?
 username=username&password=passwordMD5&from=sender_name
 &to=49500500500&template=Notify¶m1=Mark¶m2=BG12344423](https://api.smsapi.com/sms.do?username=username&password=passwordMD5&from=sender_name&to=49500500500&template=Notify¶m1=Mark¶m2=BG12344423)**

The content of sent message:
*Hello Mark, Your order has been sent. The shipment number is **BG12344423** You may follow it on our site.*

In case of any problems with sending requests to basic URL (as in example above) backup URL may be used:
<https://api2.smsapi.com/>



5. mail2SMS – Sending SMS using e-mail

To send sms using mail2sms e-mail should be created according to following scheme:

TO: **send.do@smsapi.com**
 SUBJECT: **username@password_hashed_in_md5**
 CONTENT: **from=sender name&to=number&message=message_content**

ATTENTION! Password should be given hashed in MD5

Example:

TO: **send.do@smsapi.com**
 SUBJECT: **username@8456fky567gb3bg37b357b3457b3457**
 CONTENT: **from=sender name&to=number&message=message_content**

Adding parameter **raport=1** will result in sending back e-mail containing request status (sent confirmation or error code – this is useful while testing the service):

TO: **send.do@smsapi.com**
 SUBJECT: **username@ password_hashed_in_md5**
 CONTENT: **from=sender name&to=number&raport=1&message=message_content**

Mail may be sent in plain / quotedprintable / base64 encoding. Sender name (parameter &from=) have to be active.

IMPORTANT! The recipient's number mustn't start with „+” sign. Parameter „message” must be at the end of the e-mail.

6. Account balance checking

Current account balance may be checked using one of the extended function. Find below example of using this function.

Parameter	Description
<i>username</i> *	Username used to identify a user in our system
<i>password</i> *	API password to Your account hashed with MD5
<i>credits</i> *	Enter a value „1”
<i>Format</i>	Parameter &format=json causes, that response is sending in JSON format

* - required field

Request: **https://api.smsapi.com/user.do?username=username&password=MD5password&credits=1**

I. Examples of responses when **&format=json** parameter used:

a) in case of success:

```
{
  "points": 100.00, // number of credits available in SMSAPI service
}
```

II. Examples of responses without **&format** parameter:

a) in case of success:

Response: **Credits: <CREDITS>**

or (when error occur)

b) in case of failure:



ERROR:<ERR>

<CREDITS> number of points available for this user

Example: Credits: 100.000

In case of any problems with sending requests to basic URL (as in example above) backup URL may be used:
<https://api2.smsapi.com/>.

7. SMS delivery confirmation receiving – CALLBACK procedure

We offer You possibility to run any available script in the web with callback delivery reports. In order to use this option please login on our site <https://api.smsapi.com> and set the „Callback address DLR_SMS” in „API Settings” → „Callback URLs”.

Example.: http://www.my_site.com/status_update.php

It is important that entered address is a valid address to existing, available script.

After updating message status in SMSAPI system the update will be sent to callback script (1 to 5 statuses in one request). Parameter will be sent using GET method separated by commas:

```
$_GET['Msgld']=09062414383994024,09062414383994025
$_GET['status']=403,404
```

Parameter are described in following table:

parameter	Description
Msgld*	Message ID. Length of this parameter in each request may vary, but it won't be more than 32 characters long.
status*	Status code, list of codes can be find in 'Appendix 1'
status_name	Status name, list of names can be find in 'Appendix 1'
idx*	Optional parameter send with SMS
donedate*	Date in UNIX timestamp of delivery report
username*	Username which sent SMS
points*	Number of used credits
to*	Mobile phone number of the recipient
mcc	Mobile country code
mnc	Mobile network code

*All characters are case sensitive!

Script have to return OK (*echo „OK”*), otherwise the system will be sending requests every 300 seconds.

8. Bulk CALLBACK

Bulk CALLBACK script receives request the moment that sending bulk started. This feature is generally usefull for scheduled bulks. Request contains parameters presented in table below. In order to enable Bulk CALLBACK please enter CALLBACK script URL in „API Settings” → „Callback URLs”.

Example: http://www.my_site.pl/bulk_callback.php

It is important that entered address is a valid address to existing, available script.

Parameter are described in following table:

Parameter	Description
type*	Messages' type SMS/MMS/VMS
all*	Total number of messages (phone numbers) in bulk



points*	Bulk's cost.
to*	Defines if the bulk was sent using „Numbers from file”, „Numbers and content from file” or „Numbers from contacts database” tab.
info*	contacts database group name for bulks send using „Numbers from contacts database”.
text*	Message content.

* - All characters are case sensitive!

Script have to return OK (**echo „OK”**), otherwise the system will be sending requests every 300 seconds.

9. cut.li CALLBACK

In order to check cut.li URL visits please log in to web panel and enter in „Settings” → „API” the address to a script to which information about URL visit should be sent.

Example: http://www.my_site.com/bulk_callback.php

It is important that entered address is a valid address to existing, available script.

Parameter will be sent using GET method separated by commas:

MsgId=09062414383994024
operating_system=Android

Parameter are described in following table:

Parameter	Description
MsgId*	Message ID. Length of this parameter in each request may vary, but it won't be more than 32 characters long.
to*	Recipient's phone number
click_date*	Time of first link click in unixtime format
device*	Device type
operating_system*	Operating system
browser*	Browser type
username*	Username from which message was sent

* - All characters are case sensitive!

Script has to return OK (**echo „OK”**), otherwise the system will be sending requests every 300 seconds.

10. HLR Lookup

HLR (Home location Register) is an extent base including different kinds of information about every working telephone number in GSM. In order to use this option you should request one of following URL :

- <http://api.smsapi.com/hlr.do> – for standard connections
- <https://api.smsapi.com/hlr.do> – for SSL secured connections

with proper parameters described below. All information about numbers will be sent to address given on our site <https://api.smsapi.com> at „Callback address HLR” in „API Settings” → „Callback URLs”.

It is important that entered address is a valid address to existing, available script.

After checking number in HLR information about number will be sent to given URL in POST table. There might be up to 20 numbers in one request.

Parameter are described in following table:

Parameter	Description
username *	Username used to identify a user in our system
password *	Password to Your account hashed with MD5
number*	Numbers to be checked in HLR.



<i>idx</i>	Optional custom value sent with SMS and sent back in CALLBACK. Parameetr idx may contain up to 36 chars, allowed are digits 0 – 9 and letters a – z (parameter <u>is not</u> case sensitive). (&idx=123)
------------	--

* - required field

Request: **https://api.smsapi.com/hlr.do?username=username&password=passwordMD5&number=44123123123,44234234234&idx=123,234&format=json**

I. Examples of responses when **&format=json** parameter used:

a) in case of success:

```
{
  "status": "OK",
  "number": "48100200300",
  "id": "46567088",
  "price": "0.0500"
}
```

b) in case of failure:

```
{
  "status": "ERROR",
  "number": "481002003001", // invalid phone number
  "error": 13 // error code
}
```

II. Examples of responses without **&format** parameter:

a) in case of success:

Response: **OK:<NUMBER>:<ID>:<POINTS>;OK:<NUMBER>:<ID>:<POINTS>;...;**
 e.g. response: **OK:44123123123:80625:0.006;OK:44234234234:80627:0.006;**

b) in case of failure:

Response: **ERROR:<NUMBER>:<ERR>;ERROR:<NUMBER>:<ERR>;...;**
 e.g. response: **OK:44123123123:80625:0.006;ERROR:4433412333:13;**

where:

- <NUMBER>** Checked number
 - <ID>** Checking unique ID.
 - <POINTS>** Number of used credits
 - <ERR>** Error code
- or
- OK:48600600600:80625:0.05;ERROR:48500600700:13**

Returned parameters to script are described in following table:

Parameter	Description
id*	ID returned as a response to the request
number*	Checked number
mcc*	Mobile country code
mnc*	Mobile network code
info*	Name of network or description of error
status*	OK when number is correct, FAIL when number is wrong
date*	UNIX timestamp when number was checked



ported*	0 number not ported, 1 number ported
ported_from*	null when number is not ported or name of network from which number is ported
idx*	Optional custom value sent with HLR request and sent back in CALLBACK (&idx=123)

* - All characters are case sensitive!

The data are sent in UTF8 encoding.

The list of possible error, which may appear in info field, with description is in Appendix 2 – Error codes.

Example of data sent to the script callback:

Array

```
(
  [0] => Array
  (
    [id] => 80625
    [number] => 48600600600
    [mcc] => 260
    [mnc] => 2
    [info] => T-Mobile
    [status] => OK
    [date] => 1302703609
    [ported] => 0
    [ported_from] => null
  )

  [1] => Array
  (
    [id] => 80627
    [number] => 48500600700
    [mcc] => 260
    [mnc] => 2
    [info] => ABSENT_SUBSCRIBER
    [status] => FAIL
    [date] => 1302703609
    [ported] => 0
    [ported_from] => null
  )
)
```

Script have to return OK (echo „OK“), otherwise the system will be sending requests every 300 seconds.



11. Last sentence

Special characters are these that don't fulfill regular expression: @£\$¥èéùìòçøå_{}~\|/ÆæßÉ!#"¤%&'()*+,-./0-9:;<=>?A-ZÄÖÑÜŞşz-a-zöönüà <enter>

ATTENTION! Chars: ^ { } [] ~ \ | € <enter> according to GSM specification are being counted double (when no special characters are in message).

List of special chars that may be changed to normal ones using parameter *&normalize*:

```
'normalize_chars' => array(
    'Š'=>'S', 'š'=>'s', 'Ś'=>'S', 'ś'=>'s', 'Đ'=>'Dj', 'đ'=>'dj', 'Ž'=>'z', 'ž'=>'z', 'Ž'=>'Z', 'ž'=>'Z', 'Ž'=>'Z',
    'ž'=>'z', 'Č'=>'C', 'č'=>'c', 'Ć'=>'C', 'ć'=>'c', 'À'=>'A', 'À'=>'A', 'Á'=>'A', 'Á'=>'A', 'Â'=>'A', 'Â'=>'A', 'Ã'=>'A', 'Ã'=>'A',
    'Ä'=>'A', 'Ä'=>'A', 'Æ'=>'A', 'Ç'=>'C', 'È'=>'E', 'É'=>'E', 'Ë'=>'E', 'ë'=>'e', 'Ê'=>'E', 'Ê'=>'E', 'Ï'=>'I', 'Í'=>'I',
    'Î'=>'I', 'Ï'=>'I', 'Ñ'=>'N', 'Ò'=>'O', 'Ó'=>'O', 'Ô'=>'O', 'Ë'=>'L', 'Ÿ'=>'I', 'Ń'=>'N', 'ń'=>'n', 'Ō'=>'O',
    'Ō'=>'O', 'Ø'=>'O', 'Ù'=>'U', 'Ú'=>'U', 'Û'=>'U', 'Ü'=>'U', 'Ý'=>'Y', 'Þ'=>'B', 'ß'=>'Ss', 'à'=>'a', 'á'=>'a',
    'â'=>'a', 'ã'=>'a', 'ä'=>'a', 'å'=>'a', 'æ'=>'a', 'ç'=>'c', 'è'=>'e', 'é'=>'e', 'ê'=>'e', 'ë'=>'e',
    'ì'=>'i', 'í'=>'i', 'î'=>'i', 'ï'=>'i', 'ð'=>'o', 'ñ'=>'n', 'ò'=>'o', 'ó'=>'o', 'ô'=>'o', 'õ'=>'o',
    'ø'=>'o', 'ù'=>'u', 'ú'=>'u', 'û'=>'u', 'ý'=>'y', 'þ'=>'b', 'ÿ'=>'y', 'Ŕ'=>'R', 'ŕ'=>'r',
),
```

ATTENTION! Characters outside this set will be treated as special, even if the parameter *&normalize* is set.

Points charges table:

Without special characters		With special or/and polish characters	
Characters amount	Number of parts	Characters amount	Number of parts
160	1 part	70	1 part
306	2 parts	134	2 parts
459	3 parts	201	3 parts
612	4 parts	268	4 parts
765	5 parts	335	5 parts
918	6 parts	402	6 parts

ATTENTION! The newest SMSAPI technical documentation is always in „HELP” on our website <https://www.smsapi.com/>.

CALLBACK requests may be sent from one of following IP addresses: **85.194.241.82, 89.174.81.103, 89.174.81.98, 89.174.81.102, 91.185.184.29, 91.185.185.2 or 185.36.169.252**



Appendix 1 – Delivery reports list

Status list:

Number	Status	Description
401	NOT_FOUND	Wrong ID or report has expired
402	EXPIRED	Messages expired.
403	SENT	Message is sent without final delivery report.
404	DELIVERED	Message is delivered to recipient
405	UNDELIVERED	Message is undelivered (invalid number, roaming error etc)
406	FAILED	Sending message failed – please report it to us
407	REJECTED	Message is undelivered (invalid number, roaming error etc)
408	UNKNOWN	No report (message may be either delivered or not)
409	QUEUE	Message is waiting to be sent
410	ACCEPTED	Message is delivered to operator
412	STOP	Bulk has been stopped by the user.

Appendix 2 – Error codes

Error codes list:

ERROR	Description
8	Error in request (Please report)
11	Message too long or there is no message or parameter nounicode is set and special characters (including Polish characters) are used.
12	Message has more parts than defined in <i>&max_parts</i> parameter.
13	Lack of valid phone numbers (invalid or blacklisted numbers)
14	Wrong sender name
17	FLASH message cannot contain special characters
18	Invalid number of parameters
19	Too many messages in one request
20	Invalid number of IDX parameters
25	Parameters <i>&normalize</i> and <i>&datacoding</i> musn't appear in the same request.
27	Too long IDX parameter. Maximum 255 chars.
28	Invalid time_restriction parameter value. Available values are: FOLLOW, IGNORE or NEAREST_AVAILABLE.
30	Wrong UDH parameter when <i>&datacoding=bin</i>
40	No group with given name in contacts database
41	Chosen group is empty
50	Messages may be scheduled up to 3 months in the future
52	Too many attempts of sending messages to one number (maximum 10 attempts whin 60s)
53	Not unique idx parameter, message with the same idx has been already sent and <i>&check_idx=1</i> .
54	Wrong date - (only unix timestamp and ISO 8601)
56	The difference between date sent and expiration date can't be less than 1 and more than 72 hours.
70	Invalid URL in <i>notify_url</i> parameter.



72	-
74	Sending date doesn't match date sent restrictions set for the account.
101	Invalid authorization info
102	Invalid username or password
103	Insufficient credits on Your account
104	No such template
105	Wrong IP address (for IP filter turned on)
110	Action not allowed for your account
200	Unsuccessful message submission
201	System internal error (please report)
202	Too many simultaneous request, message won't be sent
203	Too many requests. Please try again later.
301	ID of messages doesn't exist
400	Invalid message ID of a status response
999	System internal error (please report)

HLR error's list:

Error	Description
UNKNOWN_SUBSCRIBER	Invalid, not active number. Error is permanent.
ABSENT_SUBSCRIBER	Number turned off or out of range. Number is considered to be inactive but it may change back to active once it is in range. Error is temporary.
TELESERVICE_NOT_PROVISIONED	The recipient has no SMS subscription. Error is permanent.
SYSTEM_FAILURE	Temporary network or protocol failure
HLR_LOCAL_CANCEL / HLR_ABORT	Temporary problem or lost reach
CALL_BARRED	Barring of the recipients number. Error is permanent.

Appendix 3 – Encoding

Default encoding is utf-8. However you can set different encoding of messages by additional parameter **&encoding** in your HTTP request. Available encoding types are:

- 'iso-8859-1'
- 'iso-8859-2'
- 'iso-8859-3'
- 'iso-8859-4'
- 'iso-8859-5'
- 'iso-8859-7'
- 'windows-1250'
- 'windows-1251'
- 'utf-8'

Example: https://api.smsapi.com/sms.do?username=username&password=passwordMD5&to=4850000000&encoding=utf-8&message=message_content



Appendix 4 – Example scripts

Sending SMS message using GET method

```
<?php
$params = array(
    'username' => 'your_username', //username from SMSAPI
    'password' => 'md-password', //or 'password' => md5('open-text-password'),
    'to' => '4412334445566', //destination number
    'from' => 'SMSAPI', //sender name has to be active
    'message' => 'content of message', //message content
);
if ($params['username'] && $params['password'] && $params['to'] && $params['message']) {
    $data = '?'.http_build_query($params);
    $plik = fopen('https://api.smsapi.com/sms.do'.$data, 'r');
    $wynik = fread($plik,1024);
    fclose($plik);
    echo $wynik;
}
?>
```

Sending SMS message using POST method

```
<?php
$username = 'your_username'; //username from SMSAPI
$password = 'md-password'; //or 'password' => md5('open-text-password'),
$to = '4412334445566'; //destination number
$from = 'SMSAPI'; //sender name has to be active
$message = "content of message"; //message content
$url = 'https://api.smsapi.com/sms.do';
$c = curl_init();
curl_setopt($c, CURLOPT_URL, $url);
curl_setopt($c, CURLOPT_POST, true);
curl_setopt($c, CURLOPT_POSTFIELDS, 'username='.$username.'&password='.$password.'&from='
.$from.'&to='.$to.'&message='.$message);
curl_setopt($c, CURLOPT_RETURNTRANSFER, true);
$content = curl_exec($c);
curl_close($c);
echo $content;
?>
```

Receiving delivery report example (CALLBACK SMS DLR)

```
<?php
if($_GET['MsgId'] && $_GET['status'] ) {
    mysql_select_db('database_name',mysql_connect('localhost','login','password'));
    $arIds = explode(',',$_GET['MsgId']);
    $arStatus = explode(',',$_GET['status']);
    $arIdx = explode(',',$_GET['idx']);

    if($arIds){
        foreach($arIds as $k => $v){
            mysql_query("UPDATE sms SET sms_status =
            '".mysql_real_escape_string($arStatus[$k])."', sms_index =
            '".mysql_real_escape_string($arIdx[$k])."' WHERE sms_id = '".mysql_real_escape_string($v)."'
            LIMIT 1");
        }

        mysql_close();
        echo "OK";
    }
}
?>
```



History of changes

Version	Date	Changes
Version 1.2	19.08.2016	1. Introduction of format=json parameter
Version 1.1	26.08.2015	1. Introduction of <i>allow_duplicates</i> parameter
Version 1.0	08.07.2015	1. First version.

ATTENTION! You can find the newest version of SMSAPI technical documentation in tab „Help” on our site <http://www.smsapi.com/>.