

## Documentation for the Greter sms API v2.1

### Table of Contents

1. <u>Introduction</u> .....	3
1.1. <u>How to start</u> .....	3
2. <u>Single SMS Message sending</u> .....	3
3. <u>Multiple SMS Sending</u> .....	4
4. <u>Deleting scheduled messages</u> .....	4
5. <u>Account balance – credits</u> .....	5
6. <u>Callback procedure – delivery confirmation status</u> .....	5
7. <u>Delivery confirmation status</u> .....	6
8. <u>MASS messages with separate content given separately using parameters and using custom variable IDX</u> .....	6
9. <u>Email2SMS</u> .....	7
10. <u>Sending WAP PUSH</u> .....	7
11. <u>Last sentence</u> .....	8
12. <u>Errors</u> .....	8
13. <u>Encoding</u> .....	9
14. <u>Examples</u> .....	9

## 1. Introduction

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

### 1.1 How to start

To start using Gretor SMS API you need to register and create an account on our website:

**URL :** <http://sms.gretor.eu/register.html>

Registration is completely free. Your account is ready to use just after you register, but we recommend you to verify at least one sender name or number. Every message sent before validation has the „Gretor SMS“ as a default name. Verification of one name takes 1 SMS.

## 2. Single SMS Message sending

URL links needed for connecting with our application:

- <http://sms.gretor.eu/api/send.do> - for unsecured connections
- <https://sms.gretor.eu/api/send.do> - for SSL secured connections (SSL 128bit v3.0)

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>	Pass word to your account hashed with MD5.
<i>to*</i>	Mobile phone number of the recipient ( i.e. 44123456789)
<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). The Maximal message length is set to 457 normal characters or 201 if specials chars are used. Message with special chars is send as one block of 3 messages joined together. Detailed information about special characters are given in chapter 11.
<i>from</i>	Name or number of the sender. As a default the message name is set to "GretorSMS". You can always change this name after verification of your number or purchasing particular sender name. (&from=active_name)
<i>encoding</i>	This parameter describes the encoding of a message text. Windows-1250 is set as default. If an UTF-8 encoding is needed please use „utf-8“ in this field. (&encoding=utf-8)
<i>flash</i>	Sending of message in flash mode can be activated by setting this parameter to "1". Flash SMS messages are automatically presented on the mobile screen and cannot be saved. (&flash=1)
<i>idx</i>	Optional custom value sent with SMS (&idx=123)
<i>test</i>	When <i>test</i> parameter is set to "1" message won't be sent but response will be displayed (&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)
<i>date</i>	Date in unix timestamp when message will be sent (&date=1287734110)
<i>datacoding</i>	Use this parameter for sending GSM 7-bit coding (for <b>Greek characters</b> &datacoding=gsm) (for <b>WAP PUSH</b> &datacoding=bin)
<i>single</i>	Returns ERROR:12 when message will have more than one part (single=1)
<i>nounicode</i>	Returns ERROR:11 when message have special characters

\* - necessary fields

**Request:** [http://sms.gretor.eu/api/send.do?  
username=username&password=password\\_md5&  
to=44123456789&message=test%20messages](http://sms.gretor.eu/api/send.do?username=username&password=password_md5&to=44123456789&message=test%20messages)

**Response:** **OK:<ID>:<CREDITS>**

or (when error occur)

**ERROR: < ERR >**

**<ID>** Message unique ID. You will need it for delivery confirmation.

**<CREDITS>** Number of used credits (i.e. text sent in 3 messages will return value 3xSMS amount).

**<ERR>** Error code (check the error code list in chapter 12).

**Example:** **OK:08090301123437935:0.025**  
or  
**ERROR:102**

### 3. Multiple SMS sending

Sending messages to the group of recipients is similar to single number submission (presented in chapter 2). All you have to do is to fill the field "to" with multiple set of recipient numbers. 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 this message is greater than the number of available credits on your account, the system will respond with **103** error code and all messages will be rejected.

If some recipients numbers will be invalid than all others messages will be send except of those with incorrect number or prefix. Delivery reports will not concern the invalid messages which were rejected during message submission. Messages to the same recipient number will be send as single SMS message cost of 1 credit.

**Request:** [http://sms.gretor.eu/api/send.do?  
username=username&password=password&  
to=44123456789,44123456780,44123456781&message=test%20messages](http://sms.gretor.eu/api/send.do?username=username&password=password&to=44123456789,44123456780,44123456781&message=test%20messages)

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

**<ID>** ID of a sent message. You will need it for delivery confirmation.

**<CREDITS>** Number of charged credits (ex. SMS divided into 3 parts charges 3 credits).

**<PHONE>** Recipients phone number.

**Example:** **OK:08090301123437935:0.025:44123456789;OK:08090301123437936:0.025:44123456780;OK:08090301123437937:0.025:44123456781;**

We do not recommend you sending more than 1000 SMS messages in one request.

### 4. Deleting scheduled messages

Parameter	Description
username*	Username used to identify a user in Gretor SMS system
password*	Pass word to your account hashed with MD5.
sch_del*	ID of message to delete (returned after sending)

Request: [http://sms.gretor.eu/api/send.do?  
username=username&password=password&  
sch\\_del=09040611493943359](http://sms.gretor.eu/api/send.do?username=username&password=password&sch_del=09040611493943359)

Response: **OK**  
or (when ID of message doesn't exist)

Example: **ERROR:301**

## 5. Account balance - credits

Parameter	Description
username*	Username used to identify a user in Gretor SMS system
password*	Pass word to your account hashed with MD5.
credits*	Value "1" required.

Request: [http://sms.gretor.eu/api/send.do?  
username=username&password=password&  
credits=1](http://sms.gretor.eu/api/send.do?username=username&password=password&credits=1)

Response: **Credits: <AMOUNT>**  
**<AMOUNT>** Amount of credits at your sms.gretor.eu account.

Example: **Credits: 0.025**

## 6. Callback procedure – delivery confirmation status

We offer you a possibility to run any available script in the web with callback delivery report. In order to use this option please log in on our page <http://sms.gretor.eu> and set the "Callback address" in "your account" panel.

Example.: [http://www.MyPage.com/status\\_update.php](http://www.MyPage.com/status_update.php)

We suggest you to check the correctness of your script address before its submission.

When this request is called, the system will respond and pass to your script all parameters needed for delivery confirmation (GET method is used).

Script have to return OK (**echo "OK";**)

<b>MsgId</b>	Message ID
<b>status</b>	Response confirmation code (NID) – check the status list in point 7
<b>idx</b>	Optional – idx value which was sent with SMS
<b>donedate</b>	Time (in timestamp) of delivery report
<b>username</b>	Username from which message was sent
<b>points<sup>1</sup></b>	Price per SMS (for example 0.031)

\*All characters are case sensitive.

<sup>1</sup> Field will be blank or 0 if message sent to Greece and price will be estimated (number not checked by HLR yet)

Callback can return upto 5 statuses at once separated by commas. For example:

**MsgId=17101000090360359, 17101000090360358, 17101000090360357,  
status=404,404,404  
denedate=1284635547,1284635547,1284635547**

## 7. Delivery confirmation status

Parameter	Description
username*	Username used to identify a sms.gretor.eu client.
password*	Pass word to your account hashed with MD5.
status*	ID of message (returned after sending)

**Request:** [http://sms.gretor.eu/api/send.do?  
username=username&password=password&  
status=17101000090360359](http://sms.gretor.eu/api/send.do?username=username&password=password&status=17101000090360359)

**Response:** **OK: < NID >**  
<NID> Message status code (check the status list).

**Example:** **OK:404**

**ATTENTION!** Use only when CALLBACK procedure failed.

### Status list:

Number NID	STATUS	Description
401	NOT_FOUND	Invalid message ID or report expired
402	EXPIRED	Message expired
<b>403</b>	<b>SENT</b>	Message has been sent
<b>404</b>	<b>DELIVERED</b>	Message delivered
<b>405</b>	<b>UNDELIVERED</b>	Message undelivered (roaming error)
406	FAILED	Message cannot be delivered
407	REJECTED	Message rejected (unavailable recipient)
408	UNKNOWN	Mobile operator Cannot be identified
<b>409</b>	<b>QUEUED</b>	Message is queued for delivery

## 8. MASS messages with separate content given separately using parameters and using custom variable IDX.

It is possible to send up to 100 messages with separate contents in each request. If you want to send more than 100 messages, you have to do more http request (you can do it without any delay).

Content parameters should be defined as **param1,param2,param3, param4** and they will be replaced for [%1%], [%2%], [%3%] and [%4%] in message content. Values of parameters have to be separated with pipe char „|” like in following example:

**param1=abc|def|ghi|jkl&param2=abc2|def2|ghi2|jkl2**

Number of parameters has to be equal to the number of recipient numbers given in request. If not – ERROR 18 will be returned and messages will not be send. Please remember that total length of message will be increased by the length of parameter.

If one of the numbers will be incorrect then our system will omit this number, and all others parameters connected to this number also will be omitted. Ex. number1 is wrong, so INDEX1 and PARAM1 will be omitted (if they exists in request).

There is possible to send MASS messages with **IDX** parameter different to each message. To do this add at the end of request:

**&idx=idx1|idx2|idx3|idx4**

Number of IDX have to be equal to the number of recipient numbers given in request.

**Parameters:**

You can use variable parameters in message content only when you introduce them:

<b>[%1%]</b>	Text of parameter 1
<b>[%2%]</b>	Text of parameter 2
<b>[%3%]</b>	Text of parameter 3
<b>[%4%]</b>	Text of parameter 4

**Example:**

```
http://sms.gretor.eu/api/send.do?
username=username&password=pass_md5&from=johndoe&to=44508360972,31502107135_
&message=test%20message,%20param1:%20[%1%]%20param2:%20[%2%]
%20test&param1=John,Tom&param2=10,20
```

In this example the following messages will be send:

**Message1: 'test message, param1: John param2: 10 test'**

**Message2: 'test message, param1: Tom param2: 20 test'**

**9. Email2sms**

To send an SMS you have to send an e-mail according to the following example:

ADDRESS: **send.do@sms.gretor.eu**  
SUBJECT: **login@Password\_32characters\_in\_md5**  
CONTENT: **from=sender&to=number&message=message\_content**

**ATTENTION!**

- Password have to be encrypted using MD5! <http://sms.gretor.eu/md5.php>
- Message content have to be specified at the end **as a last parameter!!!**

Example:

ADDRESS: **send.do@sms.gretor.eu**  
SUBJECT: **login@8456fky567gb3bg37b357b3457b3457**  
CONTENT: **from=44123456789&to=44123456789&message=Message contents**

In case when report is needed please use the following model:

ADDRESS: **send.do@sms.gretor.eu**  
SUBJECT: **login@Password\_32characters\_in\_md5**  
CONTENT: **from=44123456789&to=44123456789&report=1&message=Message contents**

**You can personify every message using parameters (see chapter 8).**

You can also send messages using different than UTF-8 (default) encoding by using parameter encoding (see chapter 13)

**10. Sending WAP PUSH**

Before sending WAP PUSH content and URL have to be converted to BIN using our tool: <http://sms.gretor.eu/wapconv.php>

Converted content and URL use as &message parameter and add **&datacoding=bin**.

**Example:**

http://sms.gretor.eu/api/send.do?  
username=username&password=md5\_pass&from=name&to=4550050053&datacoding=bin&message=  
**0605040b8423f0860601ae02056a0045c60c03687474703a2f2f777772e6871736d732e636f6**  
**d0007010348696768205175616c6974792053657276696365000101**

**11. Last sentence**

Specials characters are assumed to be all characters not included in the following set of characters: @£\$ ¥èùìòçøå\_{}[\~]|/ÆæßÉ!"#%&'()\*+,-./0-9; <=>?A-ZÄÖÑÛŞła-zäöñüà <enter>

**Exemplary implementation in PHP script (encoding UTF-8):**

```
// Function is_spec_char returns 1 or 0 depending if the message text contains any of specials characters.

function is_spec_char($message)
{
    mb_regex_encoding("UTF-8");
    if(mb_ereg("^[^\@£$¥èùìòçøå_{}[\~]|/ÆæßÉ!"#%&'()*+,-./0-9;<=>?A-ZÄÖÑÛŞła-zäöñüà \r\n]", $message)) return 1;
    else return 0;
}
```

**12. Errors****Error codes list:**

ERR	Description
11	Message length too long
12	Message length too long, over 160 characters (when single=1)
13	Invalid recipient mobile number
14	Invalid sender name
15	Unsupported sender number
16	Unsupported sender name
17	Flash messages with special characters are not allowed
18	Invalid number of parameters
19	To many messages to send in one request(only 100 allowed)
20	Invalid amount of indexes (idx parameter)
101	Invalid authorization info
102	Invalid account ID/password combination.
103	Insufficient SMS credits on your account
200	Unsuccessful message submission
300	Invalid value of field points (value 1 is required when this parameter is used)
301	ID of messages doesn't exist
400	Invalid message ID of a status response
999	Internal error (please contact Gretor SMS)

## 13. Encoding

You can set different encoding of messages ( by default is UTF-8 ) by additional parameter the **&encoding** parameter to your HTTP GET request. Available encodings types are:

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

Example: **&encoding=iso-8859-7**

If you want to send messages using GSM 7-bit default alphabet, please add „**datacoding**” parameter to the request : **&datacoding=gsm**. Adding this parameter should be used for messages that contain special greek characters.

Example: **&datacoding=gsm**

## 14. Examples

### PHP

How to send an SMS using fopen function.

```
<?php  
  
$username = "user";  
$password = md5("password");  
$to = "441234567890";  
$from = "441234567890";  
$message= urlencode("your message");  
  
if ($username && $password && $to && $message) {  
    $data = "?username=$username&password=$password&  
        to=$to&message=$message&from=$from";  
    $file = fopen('http://sms.gretor.eu/api/send.do'.$data,'r'); $result  
    = fread($file,1024);  
    fclose($file);  
  
    echo $result;  
}  
  
?>
```



**PHP  
CALLBACK report collecting.**

Callback report:

```
<?php
// Attention!
//Following example is just a basic implementation - it does not check the correctness of input data stream of
the //GET variables: MsgId and status. It is recommended to verify an IP address of a computer which sends a
//request.

if($_GET['MsgId'] && $_GET['status'] ) {
mysql_select_db('database_name,mysql_connect('localhost','login','password'));
mysql_query("UPDATE sms SET sms_status = '".$_GET['status']."' WHERE sms_id='".$_GET['MsgId']."'
LIMIT 1");
mysql_close();

echo "OK";
}

?>
```

Callback report with own index parameter.

```
<?php
///
// If SMS was sent with idx parameter, callback will return it in variable $_GET['idx'].
//

if($_GET['MsgId'] && $_GET['status'] ) {
mysql_select_db('database_name,mysql_connect('localhost','login','password'));
mysql_query("UPDATE sms SET sms_status = '".$_GET['status']."' , sms_index = '".$_GET['idx']."' WHERE
sms_id='".$_GET['MsgId']."' LIMIT 1");
mysql_close();

echo "OK";
}

?>
```

**Example SMS submission request using HTML form in PHP**

```
<?php
i f (strlen($_POST['sms_from'])>0 &&
strlen($_POST['sms_message'])>0)
{
$username = "login";
$password = md5("your_password");
$from = $_POST['sms_from'];
$to = $_POST['sms_to'];
$message= urlencode($_POST['sms_message']);
echo
file_get_contents("http://sms.gretor.eu/send.do?".
"username=$username&password=$password&to=$to&".
"message=$message&from=$from",FALSE,NULL,0,100);
}
?>
<html>
<body>
<form name="sms" method="POST" action="">
<table>
<tr>
<td>From:</td>
<td><input type="text" name="sms_from"
value=""></td> </tr>
<tr>
<td>To:</td>
<td><input type="text" name="sms_to" value=""></td>
</tr>
<tr>
<td>Message:</td>
<td><textarea name="sms_message"></textarea></td>
</tr>
<tr>
<td colspan="2"><input type="submit" value="send"></td>
</tr>
</table>
</form>
</body>
</html>
```