



SMSBOX.GR

HTTP API v 1.4

31.10.12

Send SMS

Using our HTTP you can connect to our gateway and send SMS from your web page or application very simple. You can use the HTTP api in any programming language or dynamic web pages.

To send an SMS you have to call an url like the following:

<http://www.smsbox.gr/httpapi/sendsms.php?username=x&password=x&text=x&from=x&to=x>

You can also send the parameters using the post method at the same url:

<http://www.smsbox.gr/httpapi/sendsms.php>

The 'username' parameter is the username of your smsbox account, the 'password' is your password, 'text' is the text of the SMS, 'from' is the sender id and 'to' is the recipient's mobile phone number.

For example if we call the following url:

[http://www.smsbox.gr/httpapi/sendsms.php?
username=user&password=passwd&text=my+first+SMS
%21&from=mysite&to=306941231231](http://www.smsbox.gr/httpapi/sendsms.php?username=user&password=passwd&text=my+first+SMS%21&from=mysite&to=306941231231)

we will send an SMS to the mobile phone number 306941231231 with the text "my first SMS!" and sender id "mysite". The username is "user" and the password is "passwd".

The parameters must be url encoded. So the black space is converted to + and the ! to %21. The programming languages have functions for convert a string to an url encoded string. For more info you can visit:

<http://en.wikipedia.org/wiki/Percent-encoding>

The sender id can be up to 11 latin characters or 13 digits. Only the latin characters (a-z, A-Z), the digits 0-9 and the following characters ! () * # % . , ? - = + ; & are permitted.

The text can be up to 612 characters. The following characters:

[,] , { , } , ~ , ^ , | , \ and € need 2 bytes so they are count as 2 characters. If the text is more than 612 characters, server returns a message to you and the SMS are not delivered. If the text is up to 160 chars you are charged 1 SMS, from 161 to 306 chars you are charged 2 SMS (long SMS), from 307 to 459 3 SMS and from 460 up to 612 you are charged 4 SMS.

The GSM 7-bit format is used for text. For more info you can visit:

http://en.wikipedia.org/wiki/Short_message_service

Using this format we can send an SMS up to 160 chars (1 SMS cost). If we use unicode format then we can write any character in any language, but each SMS can be up to 70 characters.

So if you want to use gsm 7-bit format you must convert every character of your text to latin characters. For example the character ä must convert to a , the á to a and the ñ to n. The euro character (€) should be converted into %80.

The Greek letters are supported using the following conversion (see gsm 7bit table):

Γ	%13
Δ	%10
Θ	%19
Λ	%14
Ξ	%1A
Π	%16
Σ	%18
Φ	%12
Ψ	%17
Ω	%15
€	%80

So if we want to send an SMS with the text "it costs 10€" we have to call the following url:

<http://www.smsbox.gr/httpapi/sendsms.php?>

`username=user&password=passwd&text=it+costs+10%80&from=mysite&to=306941231231`

The new line character is the %0D%0A and it is counted like 2 characters.

We can send up to 100 SMS with one HTTP call. The mobile phone numbers should be separated by commas:

<http://www.smsbox.gr/httpapi/sendsms.php?>

`username=user&password=passwd&text=my+second+SMS`

`%21&from=mysite&to=306941231231,306971231231,306931231231,306941231231`

When you make a HTTP call like above, the server processes the data and returns to you a reply. This reply is important, you should read first, and **then** you should make another call to send more SMS. That means that you should not call the url more than once at the same time, but wait the server response and then make another HTTP call.

The server response for the first SMS is like the following:

20 9534538 306941231231

The first number (20) is the status of SMS, the second (9534538) is the SMS id (unique for each SMS) and the third is the number you sent.

If you send more than 1 SMS (like the second example) the response will be like the following:

20 9534538 306941231231

20 9534539 306971231231

20 9534540 306931231231

41 9534541 0941231231

Each line concerns one SMS (one mobile phone number).

The status codes are (initial status) :

20 : the SMS is accepted from our gateway

40 : no SMS available at your smsbox account

41 : the number is wrong (wrong digits number of unknown prefix)

60 : SMS rejected

Your account is charged only if the SMS status is 20.

Delivery Reports

You can be informed when an SMS status changes, using our call back url functionality. In order to use it with your account, you should give us a url, where we will send you the SMS id and its new status, when an SMS status changes. The url might be like the following :

<http://yourserver.gr/getsmsstatus.php>

So, when the first SMS will be delivered to the mobile phone, we will call a url :

<http://yourserver.gr/getsmsstatus.php?id=9534538&status=1>

The number 9534538 is the SMS id that the server returns when you send the SMS and the status 1 means that the SMS was delivered to the mobile phone.

The possible different SMS status that you can get with the delivery reports are the following:

1 : the SMS has been delivered to the mobile

2 : the SMS has not been delivered

Except the previous way to get the delivery reports, you can also request from the server the status of specific SMS that you have sent by the http api.

You call a url like the following:

www.smsbox.gr/httpapi/getreports.php?

`username=username&password=password&ids=42841532,42837170`

The ids parameter is the sms id that the server returns when we send the sms. They can be up to 100 sms id, separated by commas.

The server response is like the following:

`42837170 1 306947576421`

`42841532 1 306947992731`

In each line there is the sms id, the sms status and the mobile phone number of the recipient. The sms status will be stored on our server for at least one year from the date of send.

SMS SCHEDULING

You can schedule the sms not to be sent at the time you call the HTTP api, but in the future. To schedule the sms, you call the api as before and you just add one more parameter, the "datetosend". The parameter "datetosend" will have as value the date/time that you want the sms to be sent. The date/time must be in the future and in the following format: `datetosend=2018-10-29-11:51` . That means year-month-day-hour-minutes. If the date is not in the future, the sms are not sent and the server returns : "past date".

For example

<http://www.smsbox.gr/httpapi/sendsms.php?>

`username=user&password=passwd&text=my+second+SMS%21&from=mysite&to`

`=306971231231,306991231231,306931231231,306941231231&datetosend=2013-11-26-10:26`

The server response will be like the following:

```
Sessionid 231055
10 49858279 306971231231
10 49858280 306991231231
10 49858281 306931231231
10 49858282 306941231231
```

At the first line there is the "Sessionid". This is a unique number, and it concerns all the SMS of the same url call (1-100). It is used if the user wants to cancel this scheduled sending. The next lines have the number 10, followed by the SMS id and the mobile number. The status "10" means that the SMS have been scheduled correctly.

To cancel any scheduled messages, without of course be charged for them, you must have saved the sessionid mentioned above.

To cancel these messages we have to call the following url:

```
http://www.smsbox.gr/httpapi/cancelsms.php?
username=user&password=passwd&sessionid=231055
```

The server returns 1 if cancellation is done and 0 if not (eg wrong sessionid or already canceled or SMS)

You can also see the scheduled SMS if you login at your account from the web (www.smsbox.gr) and click on "scheduled SMS". From there you can also cancel the scheduled SMS by pressing the "X" button.

Account balance

You can check the balance at your account by calling the following url:

```
http://www.smsbox.gr/httpapi/getbalance.php?username=user&password=pass
```

where 'user' is your user name and 'pass' your password.
The server will return your account balance.

Examples using the HTTP Api

Below is an example in php, shows how to send SMS using the HTTP Api.

```
$from = urlencode($from);
$text = urlencode($text);
$fp = fopen('http://www.smsbox.gr/httpapi/sendsms.php?username=xxx&
password=xxx&from='.$from.'&to='.$to.'&text='.$text, 'r');
    $content = "";
while ($line = fread($fp, 1024)) {
    $content .= $line;
}
fclose($fp);
```

After the url call, the \$content variable will have the server response value.