



LANDING SMS SERVER HTTP GATEWAY

Version 1.6

This API provides access to all the features of 360nrs using Json.

LATEST CHANGES

- Version 1.2 18/09/2017 Added list, update and delete scheduled mailings functions
Correction of errors
- Version 1.3 07/11/2017 Added *campaignName* and *tags* parameters
- Version 1.4 16/04/2018 Python, Java and C # examples added
- Version 1.5 04/10/2018 Added parameter to specify substitution variables.
- Version 1.6 04/01/2019 Added *templateId* parameter.

CONTENTS

INTRODUCTION	Pág. 4
TECHNICAL PLATFORM	Pág. 4
Request to send SMS	Pág. 4
Parameters	Pág. 5
Example of basic request	Pág. 7
Example of CURL request	Pág. 7
Example of PHP request	Pág. 8
Example of PYTHON request	Pág. 8
Example of JAVA request	Pág. 9
Example of C# request	Pág. 11
Examples of responses	Pág. 12
Managing scheduled messages	Pág. 13
List	Pág. 13
Parameters	Pág. 13
Examples of requests	Pág. 13
Examples of responses	Pág. 13
Update	Pág. 14
Parameters	Pág. 14
Examples of requests	Pág. 14
Examples of responses	Pág. 14
Delete	Pág. 15
Parameters	Pág. 15
Examples of requests	Pág. 15
Examples of responses	Pág. 15
Examples of responses (errors)	Pág.16
Appendix A – Delivery notification	Pág. 17
Appendix B – Character set	Pág. 18

INTRODUCTION

The HTTP server platform allows users to send messages. To access your statistics and invoicing details, please go to <https://www.360nrs.com> with your user details.

Communication with the client will not be via any API provided by the company, as it will simply be HTTP communication, with some parameters to the indicated URL.

This process is detailed below.

PLATAFORMA TÉCNICA

Each request made must include client authorisation in the header of the HTTP request. This will use basic HTTP access authentication.

The authorisation header is constructed by combining the "username: password" string and encoding it in Base64. This chain is prefixed by the "Authorisation: Basic" chain.

For example, for the username "myuser" and password "mypass", the resulting header would be: Authorisation: Basic bWI1c2VyOm1pcGFzcw==

The available configuration options, the URL that should be called, and the parameters supported will be detailed below.

REQUEST TO SEND LANDING SMS

URL: <https://dashboard.360nrs.com/api/rest/landingsms>

METHOD: **POST**

PARAMETERS

Parameter	Type	Mandatory	Description
message	string	Yes	Text of message. At most, there can be 160 characters if it is not specified that the message is multi-part (see 'parts' parameter). The text must be encoded in UTF-8. The pattern "{LAND_URL}" must be included in the body of the message. This pattern will be replaced by the link to the landing page.
to	array	Yes	Mobile phone number to receive message. Must include the prefix (e.g. in Spain 34666666666). This field allows you to specify multiple recipients.
from	string	Yes	Sender text, this label will consist of 15 numbers or 11 alphanumeric characters.
templateBody	string	No	Content of the landing page in HTML format and UTF-8 encoding. Required without <i>templateId</i> .
templateId	integer	No	Template ID to send as the content of the landing page. Required without <i>templateBody</i> .
encoding	string	No	Possible values are "gsm" and "utf-16". The value "gsm" for normal mailings with GSM7 coding and 160 characters per message, and the value "utf-16" for UCS2 (UTF16) coding and 70 characters per message. If not specified, the default value is "gsm".
scheduleDate	string	No	Date of message sent in UTC format. If you need to send scheduled messages, the delivery date can be specified by

			<p>indicating the date in the following format YYYYMMDDHHIISS (e.g. 20130215142000 would be 15 February 2013 at 14:20:00). For immediate mailings, this parameter does not have to be specified.</p>
parts	integer	No	<p>Indicates the maximum number of parts into which the scheduled message will be divided. This variable is set to 1 by default, so if it is not specified and a message over 160 characters for gsm encoding is sent, the message will fail. Keep in mind that concatenated messages may only be 153 characters per part in gsm and 67 characters per part in utf-16 and each part is charged as one message. The server will only use the minimum necessary parts for sending the text even if the specified number of parts is higher than necessary. If the number of parts is less than that required for sending the text, the message will fail with error 105. The maximum number of parts allowed is 8.</p>
notificationUrl	string	No	<p>URL to receive delivery notifications.</p>
trans	integer	No	<p>Possible values are 1 and 0. With value 0, the server does not change any character in the message, as this is the default value. With value 1, the server modifies the common invalid characters in GSM7 to valid characters with the following translation table: 'á' => 'a', 'í'=>'i', 'ó'=>'o', 'ú'=>'u', 'ç'=>'Ç', 'Á'=>'A', 'Í'=>'I', 'Ó'=>'O', 'Ú'=>'U', 'À'=>'A', 'È'=>'E', 'Ì'=>'I', 'Ò'=>'O', 'Ù'=>'U', 'ø' => " , 'a' => " , 'Ö' => 'O', 'ö' => 'o', 'â' => 'a', 'ê' => 'e', 'ï'=>'i', 'ô'=>'o', 'û'=>'u', 'Â'=>'A', 'Ê'=>'E', 'Ï'=>'I', 'Ï'=>'O', 'Û'=>'U', 'ä' => 'a', 'Ä' => 'A'</p>

campaignName	String	No	Campaign name. If specified, a campaign will be created with the name indicated in the dashboard and which will contain the mailing statistics. If a campaign with this name already exists, the delivery statistics will be added to the existing campaign.
tags	array	No	<i>campaignName</i> is required if this parameter is specified. List of tags to add to the campaign. The tags can be used to filter the statistics in the dashboard.
sub	array	No	array with substitution variables that will be applied to the message.

EXAMPLE OF BASIC REQUEST

```
{
  "to":["34666555444"],
  "message": "TEST API LANDING {LAND_URL}",
  "from": "LANDINGAPI",
  "templateBody":"<html><head><title>LANDING TEST</title></head><body><h1>HELLO
LANDING</h1></body></html>"
}
```

Note: To generate an unsubscribe URL you must add: {UNSUB_URL} within the "message" parameter.

SUBSTITUTION VARIABLES

Custom variables can be indicated in the message body and in the HTML code of the template. These variables will be replaced by the contact's custom variables or by the variables indicated in the "sub" parameter.

When using the sub parameter, the array must contain as many items as the send recipients, using the following format:

```
{
  "from": "TEST",
  "to": ["34666555444", "34666555333"],
  "message": "Hello {name}",
  "templateBody":"<html><head><title>LANDING TEST</title></head><body><h1>HELLO
{name}</h1></body></html>",
```

```
"sub" : [  
  {"name": "first contact name"},// first recipient variables  
  {"name": "second contact name"}// second recipient variables  
]  
}
```

EXAMPLE OF CURL REQUEST

```
curl -X POST \  
https://dashboard.360nrs.com/api/rest/landingsms \  
-H 'authorization: Basic bWl1c2VyOm1pcGFzcmw==' \  
-H 'cache-control: no-cache' \  
-H 'content-type: application/json' \  
-d '{  
  "to": ["34666555444"],  
  "message": "TEST API LANDING {LAND_URL}",  
  "from": "LANDINGAPI",  
  "templateBody": "<html><head><title>LANDING TEST</title></head><body><h1>HELLO  
LANDING</h1></body></html>"  
}'
```


EXAMPLE OF PHP REQUEST

```
<?php
$post["to"] = array("34666555444");
$post["message"] = "TEST API LANDING {LAND_URL}";
$post["from"] = "LANDINGAPI";
$post["templateBody"] = "<html><head><title>LANDING
TEST</title></head><body><h1>HELLO LANDING</h1></body></html>";
$user = "miuser";
$password = "mipass";

try {
    $ch = curl_init();

    curl_setopt($ch, CURLOPT_URL, "https://dashboard.360nrs.com/api/rest/landingsms");
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
    curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($post));
    curl_setopt($ch, CURLOPT_HTTPHEADER, array(
        "Accept: application/json",
        "Content-type: application/json",
        "Authorization: Basic " . base64_encode($user . ":" .
$password));
    $result = curl_exec($ch);
    var_dump($result);
} catch (Exception $exc) {
    echo $exc->getTraceAsString();
}
```

EXAMPLE OF PYTHON REQUEST

```
import base64
import json
import pycurl

if __name__ == "__main__":
    url = "https://dashboard.360nrs.com/api/rest/landingsms"
    usrPass = "miuser:mipass"

    data = json.dumps({
        "to":["34666555444"],
        "message":"Test Api Landing {LAND_URL}",
        "from":"landingAPI",
        "templateBody":"<html><head><title>LANDING
TEST</title></head><body><h1>HELLO LANDING</h1></body></html>",
        "campaignName":"Name Campaign",
    })
```

```

b64Val = base64.b64encode(usrPass)
headers = ["Accept:Application/json", "Authorization:Basic %s" % b64Val]

c = pycurl.Curl()
c.setopt(pycurl.URL, url)
c.setopt(pycurl.HTTPHEADER, headers)
c.setopt(pycurl.POST, 1)
c.setopt(pycurl.POSTFIELDS, data)

c.setopt(pycurl.SSL_VERIFYHOST, 0)
c.setopt(pycurl.SSL_VERIFYPEER, 0)
c.perform()

http_code = c.getinfo(pycurl.HTTP_CODE)
print(http_code)

```

EXAMPLE OF JAVA REQUEST

```

import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.URL;
import javax.net.ssl.HttpsURLConnection;
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;

public class ApiLandingSms {

    public static void main(String args[]) throws IOException {

        String url =
"https://dashboard.360nrs.com/api/rest/landingsms";
        URL obj = new URL(url);
        HttpsURLConnection con = (HttpsURLConnection)
obj.openConnection();

        con.setRequestMethod("POST");
        String userpass = "miuser" + ":" + "mipass";
        String basicAuth = "Basic " +
javax.xml.bind.DatatypeConverter.printBase64Binary(userpass.getBytes("UTF-8"));
        con.setRequestProperty("Authorization", basicAuth);
        con.setRequestProperty("Accept", "application/json");

```

```

        JSONObject data = new JSONObject();
        JSONArray to = new JSONArray();

        to.add("34666555444");
        data.put("to", to);
        data.put("message", "Test Api Landing {LAND_URL}");
        data.put("from", "landingAPI");
        data.put("templateBody", "<html><head><title>LANDING
TEST</title></head><body><h1>HELLO LANDING</h1></body></html>");
        data.put("campaignName", "Name Campaign");
        String jsonText = data.toJSONString();

        con.setDoOutput(true);
        try (DataOutputStream wr = new
DataOutputStream(con.getOutputStream())) {
            wr.writeBytes(jsonText);
            wr.flush();
            wr.close();

            }

            BufferedReader in = new BufferedReader(
new InputStreamReader(con.getInputStream()));
            String inputLine;
            StringBuffer response = new StringBuffer();

            while ((inputLine = in.readLine()) != null) {
response.append(inputLine);
            }
            in.close();

            System.out.println(response.toString());
        }
}

```

EXAMPLE OF C# REQUEST

```
using Newtonsoft.Json;
using System;
using System.IO;
using System.Net;

namespace nrs_api
{
    class Program
    {
        static void Main(string[] args)
        {

            var httpWebRequest =
(HttpWebRequest)WebRequest.Create("https://dashboard.360nrs.com/api/rest/landingsm
s");

            httpWebRequest.Method = "POST";
            httpWebRequest.Accept = "application/json";
            String username = "miuser";
            String password = "mipass";
            String encoded =
System.Convert.ToBase64String(System.Text.Encoding.GetEncoding("ISO-8859-1").GetBy
tes(username + ":" + password));
            httpWebRequest.Headers.Add("Authorization", "Basic " + encoded);

            using (var streamWriter = new
StreamWriter(httpWebRequest.GetRequestStream()))
            {
                string[] to = new string[1];
                to[0] = "573203745060";
                var message = "Test Api Landing {LAND_URL}";
                var from = "landingAPI";
                var templateBody = "<html><head><title>LANDING
TEST</title></head><body><h1>HELLO LANDING</h1></body></html>";
                var campaignName = "Campaign Name";

                var data = new
                {
                    to = to,
                    message = message,
                    from = from,
                    templateBody = templateBody,
                    campaignName = campaignName
                };
                string json = JsonConvert.SerializeObject(data);
```


MANAGING SCHEDULED MESSAGES

LIST

Lists all scheduled mailings. Can be filtered by type (SMS or MAILING) and can specify one GUID, multiple GUIDs or no GUIDs (to list all).

The content of the message is not shown in the list, unless a single GUID is specified.

URL: <https://dashboard.360nrs.com/api/rest/scheduled>

METHOD: **GET**

PARAMETERS

Parameter	Type	Mandatory	Description
guid	integer, array	No	Identifier of message or messages. One identifier, an array of identifiers or no identifiers can be used to show all.
type	string	No	SMS or MAILING

EXAMPLES OF REQUESTS

<https://dashboard.360nrs.com/api/rest/scheduled>

<https://dashboard.360nrs.com/api/rest/scheduled?guid=100>

<https://dashboard.360nrs.com/api/rest/scheduled?guid=100,101,102>

<https://dashboard.360nrs.com/api/rest/scheduled?type=SMS>

<https://dashboard.360nrs.com/api/rest/scheduled?guid=100&type=SMS>

EXAMPLES OF RESPONSES

With several GUID or without GUID:

```
{ "result": [
  { "guid": "100", "type": "SMS", "created_at": "2017-09-18 10:49:12", "updated_at": "2017-09-18 10:49:12", "scheduled_at": "2017-11-11 10:10:10" },
  { "guid": "101", "type": "SMS", "created_at": "2017-09-18 10:49:12", "updated_at": "2017-09-18 10:49:12", "scheduled_at": "2017-11-11 10:10:10" },
  { "guid": "102", "type": "SMS", "created_at": "2017-09-18 10:49:12", "updated_at": "2017-09-18 10:49:12", "scheduled_at": "2017-11-11 10:10:10" }
], "total": 3 }
```

Specifying a single GUID:

```
{ "result": { "guid": "100", "type": "SMS", "content": "Contenido del SMS", "created_at": "2017-09-18 10:49:12", "updated_at": "2017-09-18 10:49:12", "scheduled_at": "2017-11-11 10:10:10" }, "total": 1 }
```

UPDATE

Updates the scheduled date of a mailing, multiple mailings or all scheduled mailings.

Can be filtered by GUID and/or type.

BEWARE: If no GUID is specified, all scheduled messages will be updated.

URL: <https://dashboard.360nrs.com/api/rest/scheduled>

METHOD: **PUT**

PARAMETERS

Parameter	Type	Mandatory	Description
guid	integer, array	No	Identifier of message or messages. One identifier, an array of identifiers or no identifiers can be used to show all.
type	string	No	SMS or MAILING
scheduleDate	string	Si	Date of sending message in UTC format. If you need to send scheduled messages, the delivery date can be specified by indicating the date in the following format YYYYMMDDHHIISS (e.g. 20130215142000 would be 15 February 2013 at 14:20:00). For immediate mailings, this parameter does not have to be specified.

EXAMPLES OF REQUESTS

```
{"guid": 100, "scheduleDate": "20171011093000"}  
{"guid": [100,101], "scheduleDate": "20171011093000"}  
{"type": "SMS", "scheduleDate": "20171011093000"}  
{"scheduleDate": "20171011093000"}
```

EXAMPLES OF RESPONSES

```
{"result":true,"updated":1}  
{"result":true,"updated":2}
```

DELETE

Deletes a mailing, multiple mailings or all scheduled mailings.

Can be filtered by GUID and/or type.

BEWARE: If no GUID is specified, all scheduled messages will be deleted.

URL: <https://dashboard.360nrs.com/api/rest/scheduled>

METHOD: **DELETE**

PARAMETERS

Parameter	Type	Mandatory	Description
guid	integer, array	No	Identifier of message or messages. One identifier, an array of identifiers or no identifiers can be used to show all.
type	string	No	SMS or MAILING

EXAMPLES OF REQUESTS

```
{"guid": 100}
```

```
{"guid": [100,101]}
```

```
{"type": "SMS"}
```

EXAMPLES OF RESPONSES

```
{"result":true,"deleted":1}
```

```
{"result":true,"deleted":2}
```


EXAMPLES OF RESPONSES

CÓDIGO ESTADO 400 (BAD REQUEST):

```
{"error":{"code":102,"description":"No valid recipients"}}
{"error":{"code":104,"description":"Text message missing"}}
{"error":{"code":105,"description":"Text message too long"}}
{"error":{"code":106,"description":"Sender missing"}}
{"error":{"code":107,"description":"Sender too long"}}
{"error":{"code":108,"description":"No valid Datetime for send"}}
{"error":{"code":109,"description":"Notification URL incorrect"}}
{"error":{"code":110,"description":"Exceeded maximum parts allowed or incorrect number of parts"}}
{"error":{"code":113,"description":"Invalid coding"}}
{"error":{"code":120,"description":"Invalid GUID"}}
{"error":{"code":121,"description":"Invalid scheduled date"}}
```

STATUS CODE 401 (UNAUTHORISED):

```
{"error":{"code":103,"description":"Username or password unknown"}}
{"error":{"code":111,"description":"Not enough credits"}}
```

STATUS CODE 402 (PAYMENT REQUIRED):

```
{"error":{"code":111,"description":"Not enough credits"}}
```

STATUS CODE 500 (INTERNAL SERVER ERROR):

```
{"error":{"code":122,"description":"Update error"}}
{"error":{"code":123,"description":"Delete error"}}
```

APPENDIX A: DELIVERY NOTIFICATIONS

If you wish to receive delivery notifications in real time, you must specify the variable “**notificationUrl**” with the URL of the client where you want mailing status to be notified.

The operation involves specifying the URL where you want to make a request to our server for each HTTP request when a notification from the operator is received. To do this, the client must have an HTTP server capable of receiving such notifications.

Our server will send the variables using the GET method as the client wants. To do that in the URL that you send us, you have to write the variable name followed by an escape character that will contain the value; escape characters have the form of the “%” character followed by a letter. This is an example URL:

<http://mi.server.com/notifica.php?remitente=%p&tel=%P&estado=%d>

These are the defined escape characters:

%i NRS identifier that was delivered when mailing was sent

%d value of delivery notification

%p sender of SMS

%P phone number of SMS recipient

%t date of message sent with "YYYY-MM-DD HH: MM" format, e.g. "1999-09-21

14:18"

The %d value will return the final mailing status to us, with the possible values being:

1: Message has been delivered to recipient

2: Message could not be delivered to recipient

4: Message has been delivered to SMSC, this is an intermediate notification, not an end result

16: Message could not be delivered to end operator

To better explain the process, below is an example of the SMS message and its delivery notification.

First, we send the SMS with the notificationUrl variable to indicate the URL where we want to receive the delivery notification. We will add our mailing identifier to this URL to identify it unambiguously when we receive it. The final URL for the notification would be:

<http://mi.server.com/notifica.php?idenvio=7584remitente=%p&tel=%P&estado=%d>

Therefore, the final call to send the SMS would be:

```
curl -X POST -H "Content-Type: application/json" -H "Accept: application/json" -H "Authorization: Basic bWl1c2VyOm1pcGFzcw==" -d '{"to":["34666555444"],"message":"mensaje de texto","from":"TEST","notificationUrl":"http://mi.server.com/notifica.php?idenvio=7584remitente=%p&tel=%P&estado=%d"}' https://dashboard.360nrs.com/api/rest/sms
```

Assuming that all messages can be delivered, we will receive three requests to the notifica.php script with status=1, sender=TEST, iddelivery=7584, and the corresponding phone number.

APPENDIX B: GSM7 CHARACTER SET

BASIC CHARACTER CODE

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70
0x00	@	Δ	SP	0	i	P	ı	p
0x01	£	_	!	1	A	Q	a	q
0x02	\$	Φ	"	2	B	R	b	r
0x03	¥	Γ	#	3	C	S	c	s
0x04	è	Λ	α	4	D	T	d	t
0x05	é	Ω	%	5	E	U	e	u
0x06	ù	Π	&	6	F	V	f	v
0x07	ì	Ψ	'	7	G	W	g	w
0x08	ò	Σ	(8	H	X	h	x
0x09	ç	Θ)	9	I	Y	i	y
0x0A	LF	Ξ	*	:	J	Z	j	z
0x0B	Ø	ESC	+	;	K	Ä	k	ä
0x0C	ø	Æ	,	<	L	Ö	l	ö
0x0D	CR	æ	-	=	M	Ñ	m	ñ
0x0E	Å	β	.	>	N	Ü	n	ü
0x0F	å	É	/	?	O	§	o	à

EXTENSION OF BASIC CHARACTER SET
THESE CHARACTERS OCCUPY TWO POSITIONS

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70
0x00								
0x01								
0x02								
0x03								
0x04		^						
0x05							€	
0x06								
0x07								
0x08			{					
0x09			}					
0x0A	FF							
0x0B		SS2						
0x0C				[
0x0D	CR2			~				
0x0E]				
0x0F			\					