



# THE SMS FACTORY

SMS - HTTP API

## 1. Introduction

Creation of a new SMS campaign involves 3 steps:

1. Creating the Campaign Header.
2. Adding messages to the Message Queue associated to the campaign header.
3. Wrapping Up the campaign.

## 2. Creating the Campaign Header [CREATECAMPAIGN]

The first call that needs to be made to the API is a simple request to create the new campaign. This is done by invoking the following URL:

```
/api.ashx?command=CREATECAMPAIGN&userEmail=####&password=####&campaignName=####&senderName=####
```

Where the parameters are:

command=CREATECAMPAIGN – Indicates that this is a request to create a new Campaign. userEmail=#### – Client's User Name. password=#### – Client's Password. campaignName=#### – Name of the campaign (used to identify it).

senderName=#### – Name that will appear as sender to all the campaign recipients. *Note that Sender Name format restrictions apply.*

The response to this call will be a status code (0 for errors, 1 if successful) followed by either a unique campaign identifier or the error message:

1,00000000-0000-0000-0000-000000000000 – *Successfully created a new campaign*

0,CAMPAIGN\_CREATION\_ERROR\_\_INVALID\_SENDER\_NAME

0,CAMPAIGN\_CREATION\_ERROR\_\_INVALID\_CAMPAIGN\_NAME

0,CAMPAIGN\_CREATION\_ERROR\_\_INVALID\_CREDENTIALS 0,CAMPAIGN\_CREATION\_ERROR

## 3. Adding Messages to the Message Queue [ADDMESSAGE]

Once we have successfully created a campaign header and have its CampaignId, we can start queuing messages to the campaign. We do this using the following URL:

```
/api.ashx?command=ADDMESSAGE&userEmail=####&password=####&campaignId=0000000 0-0000-0000-0000000000000000&messageText=####&phoneNumber=####
```

Where the parameters are:



command= ADDMESSAGE – Indicates that this is a request to queue a message to a campaign. userEmail=##### – Client's User Name. password=##### – Client's Password. campaignId=##### – The campaign identifier obtained when successfully creating a campaign. messageText=##### – The text of the SMS. Note that SMS restrictions apply, including character limit, character count and character restrictions.

phoneNumber=##### – The phone number to where the SMS will be sent. *Currently it must be a 9 digit valid Spanish mobile phone.*

The response to this call will be a status code (0 for errors, 1 if successful) followed by either a unique message identifier or the error message:

1,99999999-9999-9999-9999-999999999999 – *Successfully queued a message to the campaign*

0,ADD\_MESSAGE\_ERROR\_\_INVALID\_CREDENTIALS

0,ADD\_MESSAGE\_ERROR\_\_INVALID\_CAMPAIGN\_ID

0,ADD\_MESSAGE\_ERROR\_\_INVALID\_MESSAGE\_TEXT

0,ADD\_MESSAGE\_ERROR\_\_INVALID\_PHONE\_NUMBER 0,ADD\_MESSAGE\_ERROR

## 4. Wrapping Up the campaign [WRAPUP]

After queuing all messages to a campaign, we call a last URL that will mark the campaign as ready to send. Note that the system will perform a check to verify if the user has enough credit to create the campaign:

/api.ashx?command=WRAPUP&useremail=#####&password=#####&campaignId=00000000000-0000-0000000000000000

Where the parameters are:

command= WRAPUP – Indicates that this is a request complete a campaign. userEmail=##### – Client's User Name. password=##### – Client's Password. campaignId=##### – The campaign identifier obtained when successfully creating a campaign.

The response to this call will be a status code (0 for errors, 1 if successful) followed by either a success message or the error message:

1,CAMPAIGN\_WRAP\_UP\_SUCCESSFUL – *Successfully wrapped up the campaign*

0,CAMPAIGN\_WRAP\_UP\_ERROR\_\_INVALID\_CREDENTIALS

0,CAMPAIGN\_WRAP\_UP\_ERROR\_\_INVALID\_CAMPAIGN\_ID

0,CAMPAIGN\_WRAP\_UP\_ERROR\_\_INSUFICIENT\_CREDIT 0,CAMPAIGN\_WRAP\_UP\_ERROR

## 5. Summary

After successfully creating a campaign, adding messages to it and wrapping it up, the campaign will start sending the messages immediately. The status of the campaign can be checked using the Web Platform at any moment or can be queried using the CHECKCAMPAIGN and CHECKMESSAGE Api calls.

## 6. Checking Individual Messages

### [CHECKMESSAGE]

It is possible to check the status of each individual message by calling a check message command and providing the message ID. This ID is obtained as part of the response from the ADDMESSAGE command. Each message has its own individual ID that should not be confused with the campaign ID:

```
/api.ashx?command=CHECKMESSAGE&useremail=####&password=####&messageId=000000 00-0000-0000-0000000000000000
```

Where the parameters are:

command= CHECKMESSAGE – Indicates that this is a request to check a message’s status. userEmail=#### – Client’s User Name. password=#### – Client’s Password.

messageId=#### – The message identifier obtained when successfully adding a message to a campaign.

The response to this call will be a status code (0 for errors, 1 if successful) followed by either a success message or the error message:

```
1,CHECK_MESSAGE__MESSAGE_NOT_SENT_YET – Message is waiting in queue
1,CHECK_MESSAGE__MESSAGE_COULD_NOT_BE_SENT – Message rejected
1,CHECK_MESSAGE__CANNOT_BE_DELIVERED – Message accepted but cannot be delivered
1,CHECK_MESSAGE__PENDING_DELIVERY – Message sent and waiting to be delivered
1,CHECK_MESSAGE__DELIVERED – Message delivered
0,CHECK_MESSAGE_ERROR__INVALID_CREDENTIALS
0,CHECK_MESSAGE_ERROR__INVALID_MESSAGE_ID
0,CHECK_MESSAGE_ERROR__MESSAGE_ID_NOT_FOUND
0,CHECK_MESSAGE_ERROR
```

*NOTE: “0” responses mean that something was wrong in the API call. “1” responses mean the API call is properly formed (but does not necessary mean the message was successfully sent, check the exact response).*

## 7. Code Sample

```
protected String DoHttpRequest(String request)
{
    HttpWebRequest req = (HttpWebRequest)WebRequest.Create(request);
    HttpWebResponse res = (HttpWebResponse)req.GetResponse();

    String returnString = "";        if
(res.StatusCode == HttpStatusCode.OK)
    {
        Stream receiveStream = res.GetResponseStream();
        StreamReader readStream = null;        if
(res.CharacterSet == null)          readStream = new StreamReader(receiveStream);        else
readStream = new StreamReader(receiveStream, Encoding.GetEncoding(res.CharacterSet));        returnString =
readStream.ReadToEnd();            res.Close();            readStream.Close();
    }        return
returnString;
}        protected void
RunApiTest()
{
    //init variables
    String ApiTestAccount_Email = "test@api.com";
    String ApiTestAccount_Pass = "1234";
```



```
String Api_Url =
"http://www.thesmsfactory.com/envio/api.ashx?command={0}&useremail={1}&password={2}";
String Api_CreateCampaignUrl = String.Format(Api_Url, "CREATECAMPAIGN", ApiTestAccount_Email,
ApiTestAccount_Pass) + "&campaignName={0}&senderName={1}";
String Api_AddMessageUrl = String.Format(Api_Url, "ADDMESSAGE", ApiTestAccount_Email,
ApiTestAccount_Pass) + "&campaignId={0}&messageText={1}&phoneNumber={2}";
String Api_WrapUpUrl = String.Format(Api_Url, "WRAPUP", ApiTestAccount_Email, ApiTestAccount_Pass) + "&campaignId={0}";
String Api_CheckMessageUrl = String.Format(Api_Url, "CHECKMESSAGE", ApiTestAccount_Email, ApiTestAccount_Pass) +
"&messageId={0}";

String[][] messages = new String[3][];
messages[0] = new String[4] { "600000000", "Message to first contact", "", "" };
messages[1] = new
String[4] { "600000001", "Message to second contact", "", "" };
messages[2] = new String[4] { "600000002", "Message
to third contact", "", "" };
String campaignName = "Test Campaign";
String campaignSender = "ApiTest";

//CREATECAMPAIGN
String apiCreateCampaignResponse = DoHttpRequest(String.Format(Api_CreateCampaignUrl, campaignName, campaignSender));
if (apiCreateCampaignResponse.Split(',') [0] == "1")
{
String campaignId = apiCreateCampaignResponse.Split(',') [1];
for (int i = 0; i
< messages.Length; i++)
{
//ADDMESSAGE (Looping)
String apiAddMessageResponse = DoHttpRequest(String.Format(Api_AddMessageUrl, campaignId, messages[i][1],
messages[i][0]));
if (apiAddMessageResponse.Split(',') [0] == "1")
{
messages[i][2] = apiAddMessageResponse.Split(',') [1];
}
}
//WRAPUP
String apiWrapUpResponse = DoHttpRequest(String.Format(Api_WrapUpUrl, campaignId));
if
(apiWrapUpResponse.Split(',') [0] == "1")
{
for (int i = 0; i < messages.Length; i++)
{
//CHECKMESSAGE (Looping) - Should be done later.
String apiCheckMessageResponse = DoHttpRequest(String.Format(Api_CheckMessageUrl, messages[i][2]));
messages[i][3] = apiCheckMessageResponse;
}
}
}

//print results
foreach (String[] message in messages)
{
foreach (String part
in message)
{
Response.Write(part + " - ");
}
Response.Write("<br />");
}
}
```