HighCONNEXION
PART OF HighCo GROUP

High
WAY

The API that allows you to send SMS

Document version	Date	Comments
2.0	16/12/2022	Start of documentation
2.0.1	13/02/2023	Finalisation of the initial version of the V60 documentation
2.2.0	10/03/2023	Add hmac authentication
2.3.0	07/06/2023	Installation of SFTP + sms_operator in webhooks
2.3.1	04/12/2023	Add maximum values authorized on the <to>element
2.9.0	03/04/2024	Enable long SMS management functions (nb_sms and truncature elements) Added the following functions: • Retrieving the status of an SMS with a ret_id (in 4.2) • Global query on a push (highpush_stat.php) (in 4.3) • Retrieving responses in CSV format (in 5.2) • Campaign cancellation (in 6.1) • Credit check via API (in 6.2) Added the following webservices • /campaign/sms/status [GET] (in 7.3) • /campaign/status [GET] (in 7.4) • /campaign/credit [GET] (in 7.5)
2.10.0	14/05/2024	Added the following webservices • retrieveFilterStatus (en 4.1.2) • clicks] (en 6.3)
2.13.0	06/09/2024	Update of part 2.1 relating to API return, which can be retrieved identically to the V50 version of our API (TXT), or in a new JSON format

CONTENTS

1 Presentation

1.1 General presentation

HighSMS is an API from HighConnexion that lets you send SMS messages.
This API communicates with mobile operators via our router, to submit SMS messages and retrieve delivery receipts.

This document describes this API and the various uses and settings associated with it.

1.2 Lexicon

The following terms are used throughout the documentation:

push :
A push is when several messages are sent to the API within a single call.
The push is used as a reference for each message thanks to the push_id.

push_id :
The push_id is the number returned when the /campaign API is called.
Don't hesitate to register it. If you need help, the push_id will help you find the information you need more quickly.

OneShot :
OneShot is a single message push, the special feature of which is that it is sent directly and does not take the start_date and start_time parameters into account. If you don't want this behaviour, you can add the noone parameter. The message will then be treated as a push.

one_id :
Same as push_id but for the OneShot.

ret_id :
The ret_id is a parameter which will allow you to identify your message. Remember to set a unique id so that you can find the message you want. Please note that if you set it at push level, then at message level, then at to level, it will be taken into account as follows:
First the ret_id of the to, if there is none, the ret_id of the message, otherwise the ret_id of the push.

1.3 Creating your account

To use HighSMS, 2 accounts are required, a billing account and an application account. They are created and credited by HighConnexion.

Technical account or billing account
This account allows you to manage a monthly credit, set the routes that will be open for your SMS traffic (France / International) and choose the type of traffic concerned by the content of your messages (marketing or notification).
Billing is based on the traffic recorded on this account.

Application account (HighSMS)
When this account is created, HighConnexion gives you the following parameters:

- A login (also called an accountid)
- A password
- The url of the API to call

URL delivered later by your High Connexion account manager

The login/password pair is used as a parameter in the API.
A HighSMS account can be linked to several billing accounts, and vice versa.

1.4 Notion of route_type

These types of road are named according to their quality. The following equivalence is used for French traffic:
Bronze => "Direct Marketing" type mailing, use of Marketing France type short code
Silver => Send "Notification" type, using short code typed Notification France
Gold => International route

A technical account will be created for each route required to manage your traffic.
An application account will therefore be functional for 3 different uses, depending on the choice of route_type.

1.5 Securing your account by IP restriction :

If you have fixed IP addresses, they must be communicated to High Connexion in order to whitelist them and optimise the security of your traffic.

2 SMS submission via the API

The API makes it possible to submit SMS messages without using a web browser. This API can be requested in different ways.
To re-use the examples in our documentation, please replace these values with those of your implementation.

2.1 Simple API call

2.1.1 *API call*

You can run a quick POST test using the following parameters:

Name	Mandatory	Example	Comment
push	O		
accountid	O	fred	Value given by HighConnexion
password	O	password	Value given by HighConnexion
message			
text	O	Hello%20World	
to	O	0619896895	Telephone number: (+33619896895) or in local format for France (0619896895, and even without the leading zero (619896895)

2.1.2 *Back to API*

When the API is requested, it creates a container (push) to which the SMS messages to be sent are uniquely attached.
The number returned is the internal Id of this container, called push_id.

When the container is not a push but a OneShot, the Id of this container is returned (one_id).

In very marginal cases, it could happen that the API is unable to finalise the creation of the container and returns something other than an Id.

In all cases, you must not make another call to the API in a loop if it returns something other than a numeric value*.

*The API return can be configured either in JSON or in TEXT. This configuration must be defined when creating your account.

JSON Response:

- If an error occurred :

```
{
        "status": "error"
        "data": {
                "error":"ERROR_TYPE"
        }
}
```

- Otherwise :

```
{
	"status": "success",
	"data": {
		"id": 1871016995
	}
}
```

TEXT Response :

- If an error occurred :

```
ERROR_TYPE
```

- Otherwise :

```
1871016995
```

2.2 Structure details (XML/JSON)

2.2.1 XML/JSON structure

The XML / JSON contains a <push> item followed by one or more <message> items.
Some attributes can be positioned in the <push> part, or in the <message> part.

If a parameter is defined in both, the parameter in the <message> part takes precedence over that in the <push> part.

2.2.2 Element<push>

The <push> element contains one or more <message> message elements.

<push> attributes

Name	**Mandatory**	**Example value**	**Comment**
accountid	O	Fred	Value given by HighConnexion
password	O	Xpasswordx	Value given by HighConnexion
class_type **Not available on High SMS V60 Unplanned development**	~~N~~	~~1~~	~~In special cases you can be required to set this variable to 0 indicates "Flash Mode". In the other standard cases, 1 means "In the phone memory~~
route_type	N	G *Default = "your default account route*	If your account authorises several routes, you can choose a specific route. In general, you only have one route for a given account and this parameter can be left empty. Otherwise the possible value could be: 'P', 'G', or 'S' depending on your configuration.
start_date	N	20014-12-12 Default = *today*	If positioned, the SMS will not be sent before this date.

			To send the SMS immediately, do not set a date.
start_time	N	13 :00 Default = '00 :00	In special cases, if the start_date is set but in the past, the start_time is used with the current date. **Warning:** if start_time is NOT set and start date is set to a value > the current date, this is interpreted as start_time at MINUIT (00:00:00). Dates and times are local time (Paris)
stop_time	N		If stop_time is set, no SMS messages are sent to the sending Gateway after this time and any remaining SMS messages are sent the following day, starting from start_time. Note that SMS messages sent to the Gateway before this time can be sent to the operator after this time (although there is little time lag).
userdata	N	User Data Multiple Sent	This is the reference that is returned in the status files.
name	N	Push Multiple	Name used for statistics and exports.
sender	N	API_HIGHSMS	If set, replaces the 'Sender' field in the SMS. The value MUST NOT be 100% numeric and must be a maximum of 11 characters long. If empty, it will be replaced by the short code. If you set this value, and if the SMS is of the "marketing" type, you must include a statement such as 'STOP at 36105' in the text of your message, to allow the user to unsubscribe.
ret_id	N	Push_1	If set, it is stored in push.psh_ext_id. <push>.ret_id is associated with the campaign (push). This is a different field to <to>.ret_id which is an id associated with a unit SMS and defined by the customer.
ret_url [DEPRECATED] (Cf webhook URL)	N	https://highsms-hcnx.hcnx.eu/receivedDlr	If it is positioned, when the status of the SMS changes, a call is made to this url. The specifications for this call are detailed in section 4.1 of this document.
ret_mo_url [DEPRECATED] (Cf webhook URL)	N	https://highsms-hcnx.hcnx.eu/receivedMo	Url that will be called in the event of a response from the user (MO) This is a url from your platform.
webhook_url	N	https://highsms-hcnx.hcnx.eu/receivedWebhook	URL for receiving SMS tracking notifications. If set, it takes precedence over the ret_url and ret_mo_url fields. We therefore advise you not to use these ret_url and ret_mo_url fields if webhook_url is used.
priority **Not available on High SMS V60 Unplanned development**	~~N~~	~~2~~	~~Increases push priority.~~ ~~Must not be increased as part of marketing-type mass mailings.~~

noone	N	0 or 1 (default 0)	If it is set to 1, it forces the push to a single recipient to be considered as a push and not as a OneShot.
nb_sms	N	1 à 10	Maximum number of SMS messages. If set to a value greater than 0, the message cannot be transmitted in addition to nb_sms SMS.
truncature	N	0 or 1	Automatic truncation. If set to 1, a message that is too long compared to nb_sms will be truncated (and sent). Otherwise the message will not be sent and a TOO_LONG (22) error will be set.

2.2.3 ***Element <message>***

<message> contains a <text> element, one or more <to> elements and possibly <param> elements.

Attributes of the <message> element

Name	Mandatory	Example value	Comment
route_type	N	G *Default = "your default account route*	If your account authorises several routes, you can choose a specific route. In general, you only have one route for a given account and this parameter can be left empty. Otherwise the possible value could be: 'P', 'G', or 'S' depending on your configuration.
class_type **Not available on High SMS V60 Unplanned development**	~~N~~	~~1~~	~~In special cases you can be required to set this variable to 0 indicates "Flash Mode". In the other standard cases, 1 means "In the phone memory~~
start_date	N	20014-12-12 *default=today*	If positioned, the SMS will not be sent before this date.
start_time	N	13:00 default='00 :00'	In special cases, if the start_date is set but in the past, the start_time is used with the current date. **Warning**: if start_time is NOT set and start date is set to a value > the current date, this is interpreted as start_time at MINUIT (00:00:00). Dates and times are local time (Paris)

stop_time	N		If stop_time is set, no SMS messages are sent to the sending Gateway after this time and any remaining SMS messages are sent the following day, starting from start_time. Note that SMS messages sent to the Gateway before this time may be sent to the operator after this time (although there is generally little delay).
sender	N	API_HIGHSMS	If set, replaces the 'Sender' field in the SMS. The value MUST NOT begin with a number and must be a maximum of 11 characters long (1FOE is forbidden, FOE12345678 is permitted). If you set this value, and if the SMS is of the "marketing" type, you must include a statement such as 'STOP at 36105' in the text of your message, to allow the user to unsubscribe.
ret_url [DEPRECATED] (Cf webhook URL)	N	https://highsms-hcnx.hcnx.eu/receivedDlr	If it is positioned, when the status of the SMS changes, a call is made to this url. The specifications for this call are detailed in section 4.1 of this document.
ret_mo_url [DEPRECATED] (Cf webhook URL)	N	https://highsms-hcnx.hcnx.eu/receivedMo	Url that will be called in the event of a response from the user (MO) This is a url from your platform.
webhook_url	N	https://highsms-hcnx.hcnx.eu/receivedWebhook	URL for receiving SMS tracking notifications.
Ref_model **Not available on High SMS V60 Unplanned development**	~~N~~	~~Y default='N'~~	~~If it is set to 'Y', this indicates that <text> is not the text of a message, but the name of a model that has been created via the web interface. See examples specific.~~
Priority **Not available on High SMS V60 Unplanned development**	~~N~~	~~2~~	~~Increases push priority.~~ ~~Must not be increased as part of marketing-type mass mailings.~~
userdata	N	myClientName	Enables you to set additional information to be retrieved by SMSPUSHER or other reports
target_url	N	http://www.mysite.com/confirmation_parisien?nom=%NOM%&amp;city_name=%CITY_NAME%&amp;access=phone	Set a %URL% variable in the message text, which will be replaced by a tiny url (short url) that will redirect to this url. The tiny url that will replace this url is 17 characters long.
target_url_salt **Not available on High SMS V60**	~~N~~	~~4~~	~~If positioned, add a number to the tiny_url, to prevent a sequential scan of potential tiny_url regularly gives a valid result.~~

Unplanned development			

2.2.4 *<text> element*

The <text> element is used for the message text.

<text> has no attribute, just a value.
In general, this is simply the text of the SMS.

Examples of <text> values:
- The text of the message
<text>Hello world</text>
- the message text with a variable
<text>Hello world, %FIRST_NAME%</text>

As in the example above, <text> can contain variables. You can use any string as a variable name. However, we advise you to use %VAR_NAME% for the variable named 'var_name' (see 2.2.6 *Element<param>)*.

Note: The %URL% variable is reserved for tiny urls (see 2.2.7 Tiny urls).

2.2.5 *Element <to>*

<to> is used for recipient information. It is an element of <message>.
<to> can contain <param> elements, used to personalise the message text.
For the value of the telephone number (mobile), you can use either the international format (+33619896895) or the local format for French numbers (0619896895, possibly without the leading 0, such as 619896895).
For international shipments, check with your sales contact that the target country is correctly configured for your account.

Attributes of the <to> element

Name	Mandatory	Example value	Comment
value	O	+33619896895	Recipient number
ret_id	N	231467	You can use this field to associate a unique Id for each SMS. The ret_id is associated with an SMS (different from <push>.ret_id) , is stored in the database and returned in the status information.

There is a maximum limit on the number of values integrated into the <to> element. **The maximum number of values (i.e., recipient numbers) is 10,000.**
This helps optimize the handling of calls and, consequently, your overall traffic in the long run.

2.2.6 *Element <param>*

<param> is used to give the parameters of the message. It is an element of <to> or <message>.

<param> must have 2 attributes, "var" for the variable name , "value" for its value

Attributes of the <param> element

Name	Mandatory	Example value	Comment

var	O	%NAME%	We advise you to use % to enclose the variable name and to use upper case letters.
value	O	Peps	

Note: The %URL% variable is reserved for tiny urls (see 2.2.7 Tiny urls).

2.2.7 *The tinyurl*

If the message text contains %URL% and the message contains a "target_url" attribute, the tiny url generation mechanism is activated:

- a tiny url, unique for each message recipient, is generated
- %URL% is replaced by this tiny_url
- the target_url is manufactured (if it contains variables, the contents are replaced)
- the redirect between the tiny_url (the one in the SMS received by the recipient, which they will click on) and the target_url with the personalised variables is prepared.

The tiny url that will replace this url is 17 characters long. (Example: zv.hcnx.fr/uroLhp)

For example:

```
{
  "push : {
    "accountid": "fred",
    "password": "xpasswordx",
    "noone: 1,
    "message": [{
      "target_url":
"https://www.mysite.com/confirmation?origine=%VAR_MESSAGE%&amp;nom=%NOM%&amp;nom_ville=
%NOM_VILLE%&access=phone",
      "text" : "Hello, %NAME% from %VAR_MESSAGE% (%NAME_TOWN%). Click on %URL%",
      "to": [{
        "value": "+33619896895",
        "param" : [
         {
           "var": "%LASTNAME%",
           "value : "Fred
         },
         {
           "var": "%CITY_NAME%",
           "value: "Lyon
          }
        ]
      }]
    }]
  }
}
```

The text of the SMS will be :
"Hello Fred, from Province (LYON). Click on b.hcnx.fr/3Fs5lpD "
Clicking on the tiny url (b.hcnx.fr/3Fs5lpD) will redirect to :
http://www.mysite.com/confirmation?origine=Province&nom=Fred&nom_ville=LYON&access=phone

Note that it is essential that the '&' character in the value part (the target url) is escaped as '&amp ;'.
If this is not the case, you will get an ERROR_PARSING_XML with an "; expected".

2.2.8 Element <binary> (Unicode management)

Unicode management allows users to send messages with characters not available in UTF8. These include Russian, Japanese and Arabic characters, as well as smileys.

The <binary> element is used for the message content in **place of the <text> element** in cases where the message is to be sent in Unicode.

The bytes in the <binary> field can only be encoded in 2 forms (in Unicode a character is encoded in 2 bytes):

- The form **hh** :
  The Q character, for example, is coded 0051
- The form **%hh** :
  The Q character, for example, is encoded %00%51

Examples of <binary> values:
<binary>%00%42%00%6f%00%6e%00%6a%00%6f%00%75%00%72%00%20%00%65%00%6e%00%20%00%6a%00%61%00%70%00%6f%00%6e%00%61%00%69%00%73%00%20%00%73%00%27%00%e9%00%63%00%72%00%69%00%74%00%20%00%3a%00%20%3053%3093%306b%3061%306f</binary>

<binary>0042006f006e006a006f0075007200200065006e0020006a00610070006f006e0061006900730020 0073002700e900630072006900740020003a002030533093306b3061306f</binary>

To display on the phone :
Hello in Japanese is written: こんにちは

More details:
Unicode management takes into account the <param> element provided that var and value are **encoded in the same form** as the content in <binary>.
The parameters to be changed in the URL must be encoded in UTF8 so that they can be replaced correctly.
See appendix 9.1.1 for an example of a complex message in unicode.

2.3 Calling methods

2 possible call methods: XML and JSON
What is sent in XML can be submitted with the same parameter values in JSON.
This structure can be transmitted in an HTPP(s) request, or transmitted in a file and sent by (S)FTP.

2.3.1 Example Simple XML

```
< ?xml version= "1.0 " encoding= "iso-8859-1 " ?>
<push
accountid= "fred "
password= "xpasswordx "
>
```

```
<message>
<text>Helllo world!</text>
<to>0619896895</to>

</message>
</push>
```

2.3.2 Simple JSON example

```
{
  "push: {
    "accountid": "fred ",
    "password": "xpasswordx",
    "message": {
      "text": "Hello world !",
      "to": [
        {
          "value": "0619896895"
        }
      ]
    }
  }
}
```

2.3.3 Sending an XML file via http s (POST or GET)

You can send the various parameters, structured in XML via https by setting it to POST or GET. [Send as GET deprecated]
The content of the XML must be put into a variable called **xml**.

Below is an example using GET

```
https://highsms-hcnx.hcnx.eu/api?xml=<push accountid="fred"
password="xpasswordx"><message><text>Hello world
xml!</text><to>0619896895</to></message></push>
```

See appendix 9.1.3 for a complete POST example.

~~2.3.4 Sending XML in https via FastXML~~

Not available on High Push V60
Unplanned development

2.3.5 Sending XML/JSON via (S)FTP

To use this feature, you need to create a dedicated SFTP repository.
Please contact our teams if you are interested in this feature.

The creation of a campaign via SFTP repository makes it possible to process large campaigns.
Once the campaign has been sent, a statistics file will be available that will evolve over time.

It is possible to create a campaign using two different formats: XML and JSON.

There is a maximum limit on the size of the file that the SFTP repository can process for each format.

For XML, the file size must not exceed 100 mega (approximately 1,000,000 recipients).
For JSON, the file size must not exceed 40 mega (approximately 400,000 recipients).

Beyond this limit, the campaign will not be processed.

Note: If there is only one phone number in the XML/JSON file, and no start_date and start_time are set, the file will be treated as a "oneshot". The SMS will be sent instantly.

In order to be processed, the extension of the source file must be the type of data format.
An XML file must have the extension **.xml**, a JSON file must have the extension **.json**.

Once the file has been deposited (and not before) and in order to start processing, an **.ack** file with the **same name** (extension not included) as the source file must be deposited.

The presence of this file will indicate that transmission of the source file is complete and that processing can begin.
<u>**The .ack file must therefore be uploaded after the source file.**</u>

<u>Example:</u>
```
32016 dec. 17:26 PushCampaign.xml
0 16 dec. 17:26 PushCampaign.ack
```
Or
```
32016 Dec 17:26 PushCampaign.json
0 16 Dec 17:26 PushCampaign.ack
```

When processing of the file begins, an empty .run file is created.
This file indicates that the campaign is being processed.

At this point you can see in your ftp directory :
```
32016 dec. 17:26 PushCampaign.json
0 16 dec. 17:26 PushCampaign.ack
0 16 Dec 17:26 PushCampaign-230427092806.run
```

When the campaign is processed, several different types of file will be created: .run, .ret, .ok and .nok.

In order to guarantee the uniqueness of the names of the files created, the naming of these files will be based on the name of the source file and the date on which the file was processed.

Thus, the name of the files created will be: the name of the source file, a hyphen, the processing date (year month day hour minute second) then the extension.

<u>Example:</u>
If a PushCampaign.xml file is submitted on 27 April 2023 at 15:30 and processing has begun, the name of the file created will be : **PushCampaign-230427153000.run**.

Once processing is complete, an .ok or .nok file is created indicating that campaign processing is complete.
An .ok file indicates that the campaign was processed successfully, whereas a .nok file indicates that there was a problem during processing.
The .nok file will contain the error encountered when processing the file.

Once processing is complete, the source files (.xml/json and .ack) will be moved to a non-accessible archive directory.

At this point you can see in your ftp directory :

```
0 16 Dec 17:27 PushCampaign-230427092806.ok
```

Once the SMS messages have been sent (submitted to the operator), a .ret
file is created.
It contains the information from the statistical management of communication acknowledgements.
This file is updated every hour for 72 hours (the time after which undelivered SMS messages are considered to have expired).
The first SMS messages appear with the status 'SENT'.

```
PushCampaign-230427092806.ret :
"2014-12-16 17:27:02"; "2";"";""; "1";"";""; "1"
"2014-12-16 17:27:02"; "+33619896895"; "SENT"; "2014-12-16 17:31:04"; "1866810063";""; "Hello world
Multiple";""; "User
Data Multiple Sent"
"2014-12-16 17:27:02"; "+33699999999"; "SENT"; "2014-12-16 17:31:04"; "1866810063";""; "Hello world
Multiple"; "INVALID"; "User Data Multiple Sent"
```

In this example 1866810063 is what we call the "push_id

The .ret is updated and our PushCampaign.ret will look like this after a while :

```
"2014-12-16 17:27:02"; "2";"";""; "1";"";""; "1"
"2014-12-16 17:27:02"; "+33619896895"; "RECEIVED"; "2014-12-16 17:31:04"; "1866810063";""; "Hello
world
Multiple";""; "User Data Multiple Sent"
"2014-12-16 17:27:02"; "+33699999999"; "ERROR"; "2014-12-16 17:31:04"; "1866810063";""; "Hello world
Multiple"; "INVALID"; "User Data Multiple Sent"
```

2.4 Specific features of international mailings

If your SMSPusher account settings allow it, you can send SMS messages to any country.
In this case :

- You must enter the numbers in international format (**to** field) with the + followed by the country code (+33619896895 for example for a French number).
- We strongly advise you to set a sender (**sender** field). Depending on the country you want to send SMS to, please contact your CSM High Connexion for advice, as some countries require you to declare the sender. Some countries have their own specificities.

3 OneShot treatment

In the case of a push, a record is created in the database and each SMS sent is linked to this push.
You can then access consolidated statistics for this campaign.

When the push is created in the database, a separate process scans the database until the start_date and start_time trigger sending. If this start date is not set, the push will start with the first scan (scan every minute).
The push is then sent in full.

In the case of OneShot sending, a single record is created and the SMS is sent directly.
OneShot mode is automatically used if in the API submission there is only one SMS to be sent immediately (whether in API GET, XML via GET, POST or file).
A specific parameter (noone) can be used to deactivate this automatic operation.

The best way to send OneShot is to use the following web service: **FastApi**, another solution offered by High Connexion and more suitable.
Ask your High Connexion contact for specific FastApi documentation.
In the case of this FastApi call, the call is made directly to the Gateway (SMSPusher), without any record being created in the back office (HighSMS). This mode of communication is therefore reserved for certain jobs with high call frequencies.

4 Acknowledgement management

The API covers two very different functional requirements:

- Mass dispatch and real-time tracking, information transmitted in real time by our servers to your servers
- One-off dispatch with statistical tracking, initiated by your server or manually

4.1 Real-time acknowledgement management

4.1.1 Standard use

In this case, our server uses the http(s) protocol to call your server each time the state of the SMS changes, and the ret_url is called.
[ret_url DEPRECATED (we recommend using the webhook_url function)]

The ret_url is therefore called in http GET with the following parameters:

- push_id (transmitted in response to the call)
- ret_id (set by you in the initial call)
- to (recipient's telephone number in international format)
- status (a status or an error code in the case of ERROR)
- text (the text of the message sent)

The date on which the SR was received (**in UTC**) can also be sent if __DATE__ is set in the ret_url.
Similarly, the operator can be retrieved if __OPERATOR__ appears in the ret_url.
The operator is returned in the form MCC MNC (Example 20801 for Orange)

If ret_url is not set, no call is made.
If the webhook_url is set, the ret_url and ret_mo_url will be ignored. See section 4.4 Receiving webhooks for more details.

In all cases, the push_id, corresponding to the unique identifier of your push, is returned.
The text of the SMS is returned in the <text> field.
All the status values that can be found in ret_url are specified in appendix 9.6 of this document.

4.1.2 Use of an intermediate filtering script

It may happen that you do not want to receive all the statuses, but only those that require processing in the CRM.
For example, only receive "hard bounces" of the NPAI or INVOP type, corresponding to the numbers to be deleted from the mailing list.

A script that meets this need is: RetrieveFilterStatus.php

By replacing :
ret_url="http://{your_webhook_url}"
or
webhook_url="http://{your_webhook_url}"
by
ret_url=https://{your_endpoint}.hcnx.eu/RetrieveFilterStatus.php?target_ret_url=http://{your_ret_url}&status=11,13
or

webhook_url=https://{your_endpoint}.hcnx.eu/RetrieveFilterStatus.php?target_ret_url=http://{your_webhook_url}& status=11,13

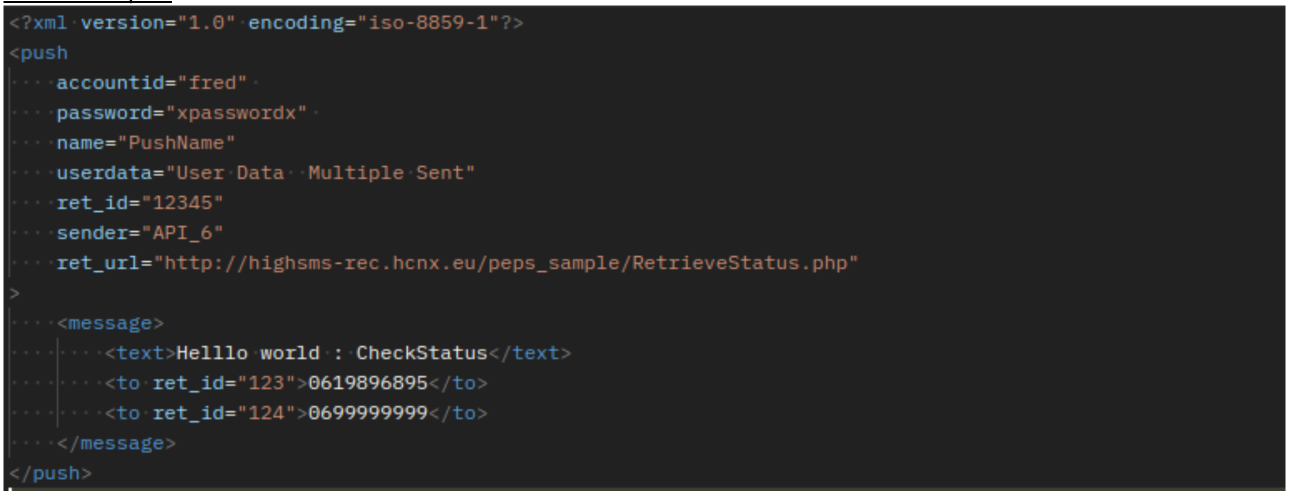
Only ERROR_NPAI and ERROR_INVOP statuses will be submitted to http://{your_ret_url}

This avoids saturating the SR processing server when sending with RECEIVED statuses that may be of no interest.

4.2 Retrieving the status of an SMS with a ret_id

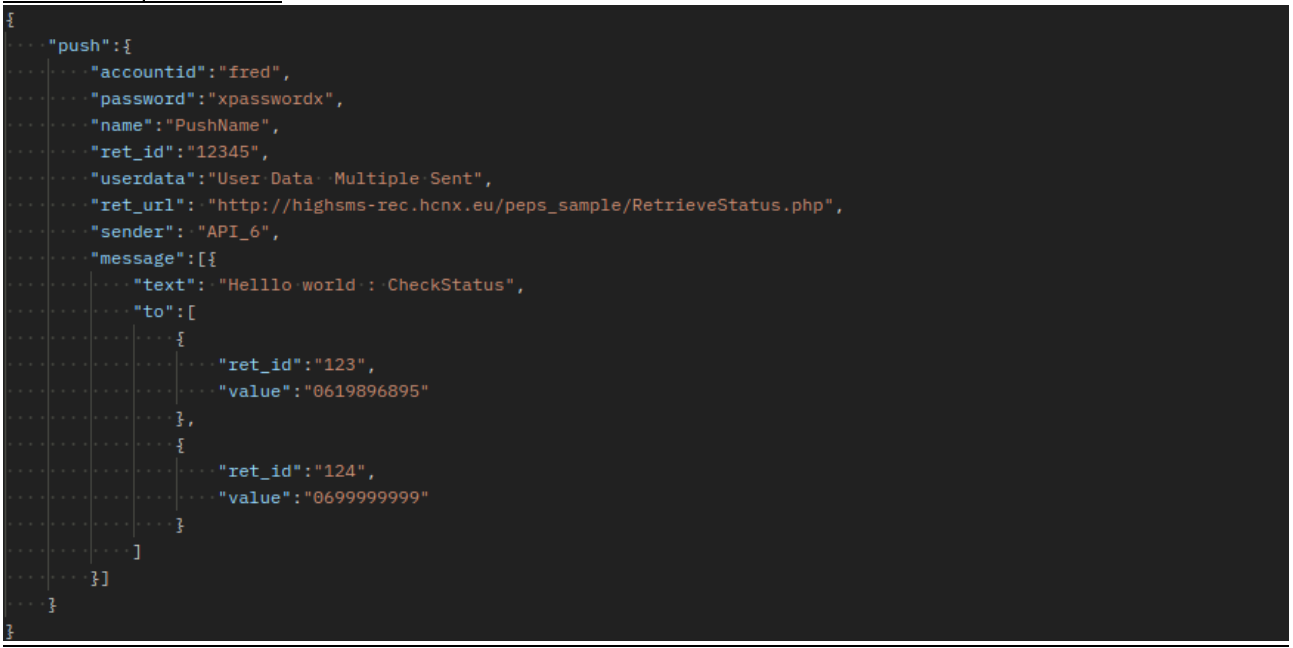
To retrieve the status of an SMS, you must have sent an XML or JSON that has a ret_id for each <to> attribute.

XML Example:



```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<push
    accountid="fred"
    password="xpasswordx"
    name="PushName"
    userdata="User Data  Multiple Sent"
    ret_id="12345"
    sender="API_6"
    ret_url="http://highsms-rec.hcnx.eu/peps_sample/RetrieveStatus.php"
>
    <message>
        <text>Helllo world : CheckStatus</text>
        <to ret_id="123">0619896895</to>
        <to ret_id="124">0699999999</to>
    </message>
</push>
```

Same example in JSON:



```json
{
    "push":{
        "accountid":"fred",
        "password":"xpasswordx",
        "name":"PushName",
        "ret_id":"12345",
        "userdata":"User Data  Multiple Sent",
        "ret_url": "http://highsms-rec.hcnx.eu/peps_sample/RetrieveStatus.php",
        "sender": "API_6",
        "message":[{
            "text": "Helllo world : CheckStatus",
            "to":[
                {
                    "ret_id":"123",
                    "value":"0619896895"
                },
                {
                    "ret_id":"124",
                    "value":"0699999999"
                }
            ]
        }]
    }
}
```

You can then call **ws_status_for_ret_id.**
The parameters to provide are the accountid, password and ret_id of the <to>.

In your example:

https://{your_endpoint}.hcnx.eu/ws_status_for_ret_id?accountid=fred&password=xpasswordx&ret_id=123

The response is CSV with the following headers : Vers, Statut, Status description, Opérateur, MVNO, Date.
Example:
"Vers";"Statut";"Status description";"Opérateur";"MVNO";"Date"
"+33619896895";"2";"Erreur envoi de message";"";"";"2015-12-04 10:04:52"

Note : You can also make POST request.

4.3 Global query on a push (highpush_stat.php)

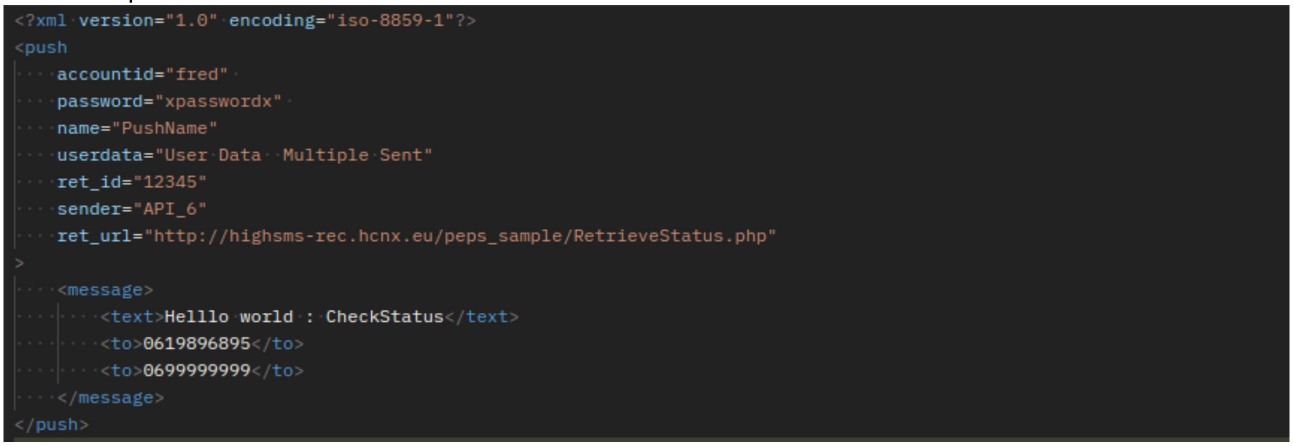
If you don't want to have your own real-time database but want to know the status of your campaign SMS from time to time, you can call our server in http.
The endpoint to call is **highpush_stat.php**

To avoid loading the servers, DO NOT CALL this script too frequently, and do not set up an automatic and recurring call system.

Name	Mandatory	Example value	Comment
accountid	O	fred	Value given by HighConnexion.
password	O	xpasswordx	Value given by HighConnexion.
push_id	O si ret_id non défini	1871011352	Internal value of the push_id, returned when XML/JSON is provided by an HTTP request.
ret_id	O si push_id non défini	12345	Value provided in the XML/JSON, **associated with the push, not the individual message**

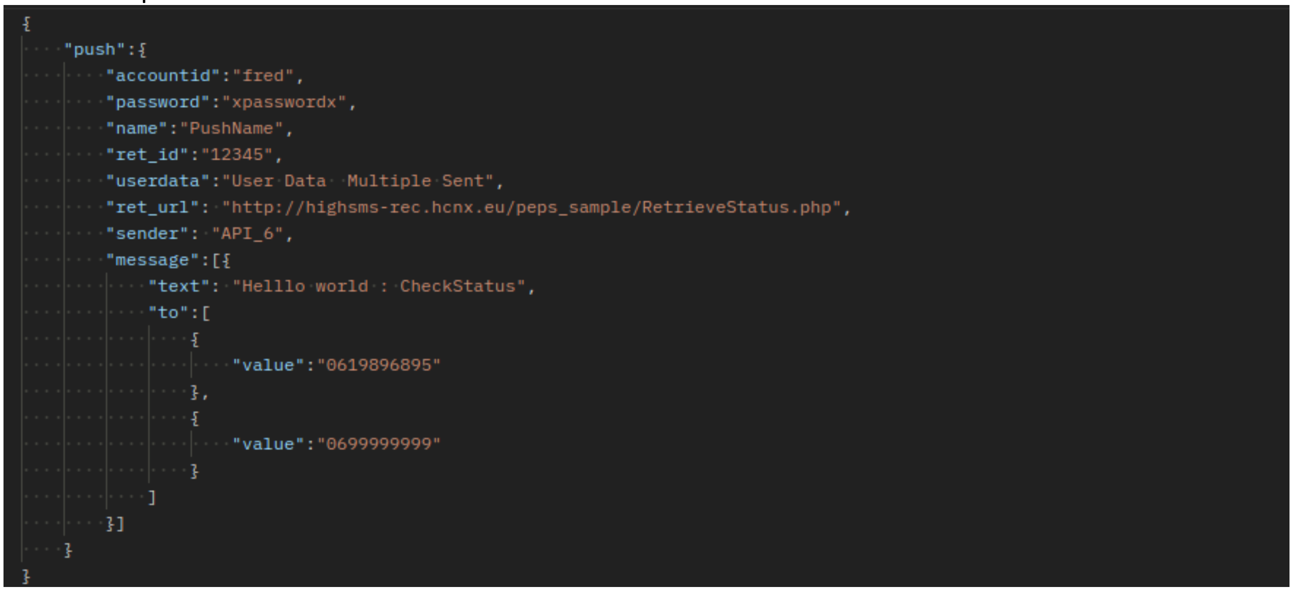
To use this script you need to provide your accountid, password and either the push_id or the ret_id

XML Example :



```
<?xml version="1.0" encoding="iso-8859-1"?>
<push
    accountid="fred"
    password="xpasswordx"
    name="PushName"
    userdata="User Data  Multiple Sent"
    ret_id="12345"
    sender="API_6"
    ret_url="http://highsms-rec.hcnx.eu/peps_sample/RetrieveStatus.php"
>
    <message>
        <text>Helllo world : CheckStatus</text>
        <to>0619896895</to>
        <to>0699999999</to>
    </message>
</push>
```

Same example in JSON :



```
{
    "push":{
        "accountid":"fred",
        "password":"xpasswordx",
        "name":"PushName",
        "ret_id":"12345",
        "userdata":"User Data  Multiple Sent",
        "ret_url": "http://highsms-rec.hcnx.eu/peps_sample/RetrieveStatus.php",
        "sender": "API_6",
        "message":[{
            "text": "Helllo world : CheckStatus",
            "to":[
                {
                    "value":"0619896895"
                },
                {
                    "value":"0699999999"
                }
            ]
        }]
    }
}
```

We can query either using the push_id:
https://{your_endpoint}.hcnx.eu/highpush_stat.php?accountid=fred&password=xpasswordx&push_id=1871011352
Either by using ret_id
https://{your_endpoint}.hcnx.eu/highpush_stat.php?accountid=fred&password=xpasswordx&ret_id=12345

Note :
In case of multiple pushes with the same ret_id, only the data associated with the last ret_id is returned.
The response is a CSV file.

Lines in bold italics are not sent

Send date	***Total***	***Net_error***	***Waiting***	***NPAI***	***expired***	***Other error***	***received***		
08/01/15 11:38	2			1			1		
Send date	***Phone Number***	***Status***	***Status Update***	***push_id***		***text***	***Error Status***	***userdata***	***Parent_id***
08/01/15 11:38	33619896895	RECEIVED	08/01/15 12:22	1871011352		Helllo world : CheckStatus		User Data Multiple Sent	User Data Multiple Sent
08/01/15 11:38	33699999999	ERROR	08/01/15 12:22	1871011352		Helllo world : CheckStatus	INVOP	User Data Multiple Sent	User Data Multiple Sent

First line is a summary :
- PushDate
- Total number of messages (which may be different from the number of SMS)
- Total of net_error (ERROR_NETWORK => 14)
- Total waiting (pas de réponse Error ou Received)
- Total npai (ERROR_NPAI=>11)
- Total expired (ERROR_EXPIRED=>12)

- Total other errors
- Total Received

Then there is a line per SMS (number which may be different from the number of messages)
- PushDate
- Phone number (to)
- Status
  - SENT
  - RECEIVED
  - ERROR
- Last status update (RECEIVED status update may arrive long after SMS is received)
- *Empty*
- Message text
- Error status (if status = Error)
  - NPAI
  - EXPIRED
  - INVOP
  - NETWORK
  - BLOCKED
  - TOO_LONG
  - UNKNOWN
- Userdata (field userdata of the xml)
- Parent SMS ID in the case of the second part of a long SMS

4.4 Receipt of webhooks

Webhooks are api calls that notify you with a uniform json structure for the following events:
 - One click
- Sending the text message (SENT)
 - The sms return status (RECEIVED or ERROR)
- The user's response

Each type of webhook is optional, so if you only want to receive the sms you receive and not the rest, you can set up a configuration to select the type of webhook you want to receive.
It is not possible to request that the webhook of a sent sms be sent to one url and that the webhook of a received sms be sent to a different url. All webhooks can only be sent to a single address. The aim of standardising these is to simplify processing for you.

If the ret_url and/or ret_mo_url are set, the webhook_url will be ignored and therefore it will not be possible to benefit from any webhook.

JSON specification

field in the Webhook JSON	Type	Field explanation
event	Object	Event object
sms	Object	SMS subject
url	Object	Object URL

Event" object

field in the Webhook JSON	Type	Field explanation
event_status	String	Event
event_http_id	Integer	HTTP call ID (HCNX ID)
event_date	Datetime	Date of event

List of events :
Event_status = [MT, DLR, MO, CLICK]

Subject "sms

field in the Webhook JSON	**Type**	**Field explanation**
sms_hcnx_id	String	SMS Id (HCNX Id (mt_id))
sms_client_id (ref doc : ret_id du <to>)	String	SMS ID (Customer ID)
sms_client_campaign_id (ref doc: push ret_id)	String	Id of the Customer's campaign
sms_msisdn	String	MSISDN
sms_status	String	SMS status
sms_message	String	Message compiled
sms_sender	String	Sender Id compiled
sms_response	String	Message in reply
sms_user_data (ref doc: userdata)	String	Optional data supplied by the customer (example: customer's customer)
sms_counter	int	Number of text messages in the message
sms_operator	String	MccMcn to identify the operator who sent the message.

Status list (MT)
sms_status = [
CREATED, // MT taken into account at HCNX
SENT, // MT sent to the operator
ERROR_INVOP, // MT: Invalid operator for this Msisdn
ERROR_NOCREDIT, // Credit limit exceeded for the application account
ERROR_BLACKLIST // MT: the number is blacklisted on the HCNX platform
ERROR_BADPASSWORD // incorrect accountid and password
ERROR_UNKNOWN // Other MT errors
]

Status list (DLR)
sms_status = [
RECEIVED, // MT received by the terminal
ERROR_NPAI, // MT: Msisdn does not live at the address given
ERROR_EXPIRED // MT: expired, usually after 24 hours
ERROR_INVOP, // MT: Invalid operator for this Msisdn
ERROR_NETWORK, // MT in error on the operator network
ERROR_CREDIT // The account used no longer has credit
ERROR_NPAIPORTED // MT: Msisdn does not live at the address given and the number has been ported
ERROR_UNKNOWN // Other MT errors
]

Status list for event MO
sms_status = [
INCOMING, // MO received from the terminal
]
Status list for event CLICK

```
sms_status = [
RECEIVED, // MT received by the terminal
]
```

url" object

field in the Webhook JSON	**Type**	**Field explanation**
short_url	String	short url
target_url	String	long url

An example of each type of event and possible notification is given in appendix 9.1.5.

5 Response management

5.1 Transmission of responses to ret_mo_url

[ret_mo_url DEPRECATED (use of the webhook_url function is recommended)]

If the user responds to the SMS (sends an MO in response to an MT), the content of the SMS is sent back to the MO management url, either the one explicitly positioned in the call, or the one configured by default in the user's technical account.
In our example it is http://myaddress.com/TreatMO.php

This url is called in http in GET format with all the parameters set by the platform.

The main parameters used are listed below:

Name	**Example value**	**Comment**
FROM	+33619896895	Recipient's telephone number, in international format (+336...) equivalent to the <to> field in the initial request.
TO	36105	Short code used by the recipient to reply
MESSAGE	STOP%20Thanks %20but%20stop .	Message text, escaped url.
RECEPTION_DATE	2014-12-12T13:00:05	Date received, ISO format.
TO_OP_ID	20810	Operator ID. *Main references French operators :* 20801: Orange 20810 : SFR 20820: Bouygues 20815 : Free 20823: Virgin Mobile 20826: NRJ MOBILE 20827 :Coriolis
ORIG_ID	99134567	push_id (the one returned in the initial HTTP request)
RET_ID	123456	ret_id as set in the initial http request
ORIG_MESSAGE	Hello%20World	Text of the initial MT SMS

You can see a detailed example in appendix 9.1.4 SR and MO recovery. Apart from the parameters listed above, the rest of the data is of little interest.

5.2 Retrieving responses in CSV format

To retrieve the responses (including STOP) corresponding to a sending via the API, it is necessary to make a call to the "answer" webservice with the ret_id of the push.
In our example:
https://{your_endpoint}.hcnx.eu/answer.php?accountid=fred&password=xpasswordx&ret_id=12345&start_date=2024-03-10

This WebService is intended to be used after a campaign to retrieve the STOPS.
It can be penalizing if you search on a date range that is too large.
Also in all cases, the search is only carried out on the last 40 days.

Parameters

Name	Mandatory	Exampl value	Comment
accountid	O	fred	Value given by HighConnexion (HighSMS account).
password	O	xpasswordx	Value given by HighConnexion (HighSMS account).
ret_id	O	12345	**ret_id** you provided in the XML/JSON**, associated with the push, not the individual message**
start_date	O	2024-03-10	Filter start date
end_date		2024-03-10	Filter end date

The generated file is a CSV which contains the following columns :

Nom	Valeur exemple	Commentaire
Date and time	10/03/2024 12:16:41	Response date and time (JJ/MM/AAAA HH :MM :SS)
From	33619896895	Transmitter number
To	36105	Recipient number
Stop	1	1 if STOP otherwise 0
Message	STOP	Response text

It should be noted that this WebService can also be used to retrieve the responses corresponding to a campaign created with the interface. In this case, the ret_id of the request must be given the name of the campaign.

6 Other API calls

6.1 Campaign cancellation

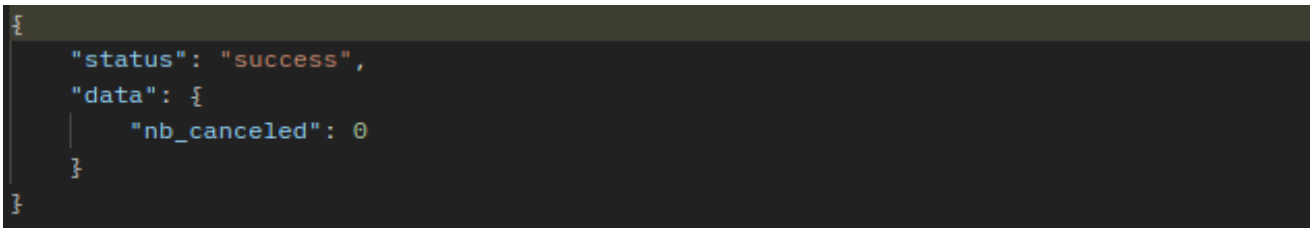
Canceling a campaign is only possible if it has been scheduled with an ext_id and has not yet started.
From then on, it is possible to cancel a campaign by calling the following URL whose parameters are detailed below:
https://{your_endpoint}.hcnx.eu/api?accountid=fred&password=xpasswordx&ext_id=12345&action=cancel_campaign

Parameters

Name	Mandatory	Example value	Comment
accountid	O	fred	Value given by HighConnexion (HighSMS account).
password	O	xpasswordx	Value given by HighConnexion (HighSMS account).
action	O	cancel_campaign	

ext_id	O	12345	**ret_id** you provided in the XML/JSON**, associated with the push, not the individual message**

In response, you will receive the number of campaigns that were canceled* or an error message.



```
{
    "status": "success",
    "data": {
        "nb_canceled": 0
    }
}
```

In this example, since the campaign has already been sent, no campaign is canceled.

*There may be multiple canceled campaigns if you split the same campaign into smaller ones for performance reasons.

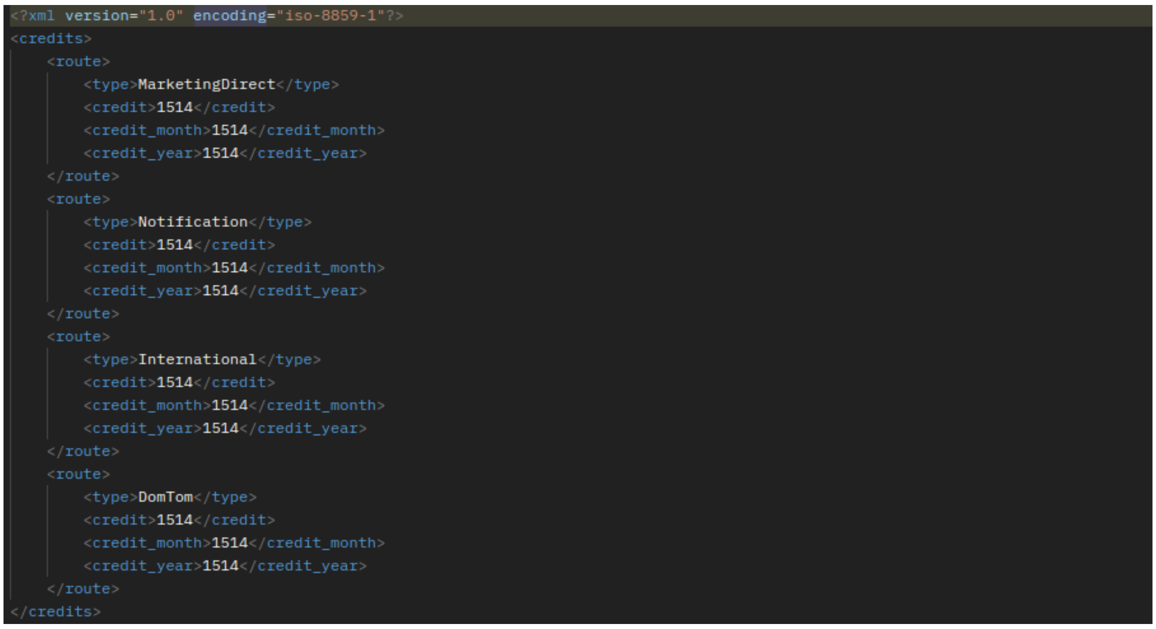
6.2 Credit check via API

You can check your credit before submitting a push. To do this, use the **credits** query.
The only parameters to provide are the accountid and password.

For example :
https://{your_endpoint}.hcnx.eu/credits?accountid=fred&password=xpasswordx

The response is an XML that indicates, for each account route, the authorized credit.
The credit can be global, monthly or annually.
- -1 indicates that there is no credit limitation.
- 0 indicates "no more credit"

Below is the response to the call for Fred's account which has a "Gold", "Silver", "Bronze" and "Premium" route with 1514 credits.



```
<?xml version="1.0" encoding="iso-8859-1"?>
<credits>
    <route>
        <type>MarketingDirect</type>
        <credit>1514</credit>
        <credit_month>1514</credit_month>
        <credit_year>1514</credit_year>
    </route>
    <route>
        <type>Notification</type>
        <credit>1514</credit>
        <credit_month>1514</credit_month>
        <credit_year>1514</credit_year>
    </route>
    <route>
        <type>International</type>
        <credit>1514</credit>
        <credit_month>1514</credit_month>
        <credit_year>1514</credit_year>
    </route>
    <route>
        <type>DomTom</type>
        <credit>1514</credit>
        <credit_month>1514</credit_month>
        <credit_year>1514</credit_year>
    </route>
</credits>
```

6.3 Retrieving clicked links

It is possible for a push whose MTs include a short URL generated by us, to obtain the details of the clicks: recipient who clicked and number of clicks per recipient. To do this, you must call the “clicks” WebService with the ret_id of the push. In our example with https://{your_endpoint}.hcnx.eu as the root url:
https://{your_endpoint}.hcnx.eu/clicks.php?accountid=fred&password=xpasswordx&ret_id=12345&start_date=2024-05-01

It can be penalizing if you search over too large a date range.
Also in all cases, the search is only carried out over the last 40 days.

Parameters

Name	Mandatory	Example value	Comment
accountid	O	fred	Value given by HighConnexion (HighSMS account).
password	O	xpasswordx	Value given by HighConnexion (HighSMS account).
ret_id		12345	ret_id you provided in the XML, associated with the push, not the individual message
start_date	O	2024-05-01	Filter start date
end_date		2024-05-01	Filter end date

The generated file is in CSV format and contains the following columns:

Name	Example value	Comment
Date et heure	2024-05-01 12:16:41	MT date and time (AAAA-MM-JJ HH :MM :SS)
From	33619896895	Transmitter number
Message	Recevez votre coupon Mes Vacances en cliquant sur http://b.HCNX.eu/5f.	MT text
Tiny_url	http://b.HCNX.eu/5f	Tiny URL present in MT message
Target_url	http://mesvacances.fr/Reduction.html	URL that the Tinyurl refers to
Date du dernier clic	2024-05-01 14 :15 :31	Date and time of the last click made by the transmitter
Nombre de clics	3	Number of times the sender clicked on this link
Userdata	Compte soleil	Identifier used to position additional information that will be returned by SMSPUSHER or other reports

Please note that this Web Service can also be used to know the clicks made in response to a OneShot. It will then be necessary to ensure to provide the ret_id positioned in the <to> element when sending the MT or, if it has been omitted, the ret_id positioned in the <push> element.

Likewise, this Web Service can be used to know the clicks made in response to a campaign created with the Highway web interface (formerly HighPush or HighSMS). The ret_id provided in the parameters must then correspond to the name of the campaign.

7 API Rest / JSON

HighSMS implements all the functionalities in a Rest / JSON API.
This is where you'll find all the Web Services that make up the Rest API.

7.1 /campaign [POST]

Description :

Enables SMS campaigns to be sent.

Request :

- Method : POST
- POST data in JSON format :

```
{
        "push: {
                "accountid": "fred",
                "password": "xpasswordx",
                "start_date": "2017-03-08",
                "start_time": "17:25",
                "userdata": "MONDOR_Cardio",
                "sender": "MONDOR",
                "ret_id": "Push_Mondor_324",
                "ret_url": "",
                "priority: 2,
                "ret_mo_url": "",
                "noone": false,
                "nb_sms": 1,
                "truncature": 1,
                "message": [
                        {
                                "text": "Hello %NAME%, your appointment for %DATE% is confirmed",
                                "to": [
                                        {
                                                "value": "+33623456789",
                                                "ret_id": "Mess4563",
                                                "param": [
                                                        {
                                                                "var": "%NAME%",
                                                                "value": "DUPONT
                                                        },
                                                        {
                                                                "var": "%DATE%",
                                                                "value": "Thursday 9 March 2017, 8:30 a.m."
                                                        }

                                                ]
                                        },
                                        {
                                                "value": "+33623475352"
                                                "ret_id": "Mess4564",
                                                "param": [
                                                        {
                                                                "var": "%NAME%",
                                                                "value": "DURAND
                                                        },
                                                        {
                                                                "var": "%DATE%",
                                                                "value": "Thursday 9 March 2017, 8:30 a.m."
                                                        }
                                        ]
                                }
                        ],

                }
        ]
    }
}
```

Response:

- If an error has occurred :

```
{
        "status": "error
        "data": {
                "error": "ERROR_TYPE"
        }
}
```

- Otherwise :

```
{
        "status": "success",
        "data": {
                "id: 1871016995
        }
}
```

Note on special characters
Special characters must be set in HTML syntax :
"text": "Notification "&lt;\".&amp; " for Notification "<".&

7.2 /campaign/cancel [POST]

Description :

Used to cancel a campaign that has not yet started.

Request :

- Method : POST
- POST data in JSON format :

```
{
        "accountid": "fred",
        "password": "xpasswordx",
        "action": "cancel_campaign",
        "ext_id: 12345
}
```

The **ext_id** (client-side push identifier) is the **ret_id** associated with the push provided in the call to /campaign

Response:
Returns the number of campaigns that have been cancelled* or an error message.

```
{
        "status": "success",
        "data": {
                "nb_canceled": 0
        }
}
```

*There may be several cancelled campaigns if you have split a single campaign into several smaller ones for performance reasons.

The campaign can be cancelled up to 10 minutes before it is launched.

Please note:
The campaign is not really cancelled, but its status is forced to _CAMPAIGN_STATUS_FINISHED, which means that no more SMS messages associated with this campaign are submitted to the operator.

7.3 /campaign/sms/status [GET]

Description :

Allows you to obtain information regarding the sending of an SMS.

Request :

- Method : GET
- GET parameters to pass :

```
"accountid": "fred"
"password": "xpasswordx"
"ret_id": 12345
```

Response :

```
{
          "status": "success",
          "data": {
                    "sms":
          [
                              {
                              "number": "+33619896895",
                              "status": 6,
                              "status_desc": "RECEIVED",
                              "operator": "Orange France",
                              "MVNO": "20801",
                              "date": "2015-12-04 10:04:52",
                              },
          ],
                    }
          }
}
```

7.4 /campaign/status [GET]

Description :

Allows you to obtain campaign status and its SMS.

Request :

- Method : GET
- GET parameters to pass :

```
"accountid": "fred"
"password": "xpasswordx"
/* Renseigner l'un des deux obligatoirement */
"push_id": 1871011352
"ret_id": 12345
```

Response :

```
{
		"status": "success",
		"data": {
				"push_id": 1871016995,
				"ret_id": "Push_Mondor_324",
				"sender": "MONDOR",
				"userdata": "MONDOR_Cardio",
				"summary": {
						"send_date": "2017-03-08 17:25",
						"total_message": 2,
						"total_sms": 2,
						"net_error": 0,
						"waiting": 0,
						"npai": 1,
						"expired": 0,
						"other": 0,
						"received": 1,
				},
				"message": [
						{
						"send_date": "2017-03-08 17:25",
						"number": "33623456789",
						"status": "RECEIVED",
						"status_update": "2017-03-08 17:27",
						"ret_id": "Mess4563",
						"text": " Hello Dupont, your appointment for Thursday March 9, 2017, 8:30 a.m. is confirmed. ",
						"nb_sms": 1,
						"error_status": null,
						},
						{
						"send_date": "2017-03-08 17:25",
						"number": "33699999999",
						"status": "ERROR",
						"status_update": "2017-03-085 17:36",
						"ret_id": "Mess4564",
						"text": " Hello Durand, your appointment for Thursday March 9, 2017, 8:30 a.m. is confirmed. ",
						"nb_sms": 1,
						"error_status": "NPAI",
						}
				]
		}
}
```

If we pass the push_id (HighConnexion internal identifier) it is necessarily unique.
If we pass the ret_id (set by the client) and there are several pushes associated with this ret_id we only provide the information of the last one.

Values of `"status"`

- RECEIVED
- ERROR
- SENT

Values of `"error_status" (`if `"status" =` ERROR)

- NPAI
- EXPIRED
- INVOP
- NETWORK
- BLOCKED
- TOO_LONG
- UNKNOWN

7.5 /campaign/credit [GET]

Description :

Allow you to obtain informations about your credit.

Request :

- Method : GET
- GET parameters to pass :

```
"accountid": "fred"
"password": "xpasswordx"
```

Réponse :

```
{
	"status": "success",
	"data": {
		"route": [
			{
				"type": "Notification",
				"credits": -1,
				"credits_month": -1,
				“credits_year”:-1
			}
		]
	}
}
```

Note : -1 indicates that there is no limit.

All the following APIs are not and will not be developed on High SMS V60

~~7.6 /contact_list/create_list[POST]~~

~~7.7 /contact_list/add_contacts[POST]~~

~~7.8 /contact_list/get_lists[GET]~~

~~7.9 /contact_list/delete_contacts[POST]~~

~~7.10 /contact_list/delete_list[GET]~~

~~7.11 /contact_list/prepare_push[POST]~~

~~7.12 /contact_list/send_push[POST]~~

~~7.13 /contact_list/stat_push[GET]~~

~~7.14 /contact_list/cancel_push[POST]~~

~~7.15 /contact_list/concat_lists[POST]~~

8 Token-based security for rotating IPs

8.1 Recommended Method

8.1.1 Authentication for API calls :

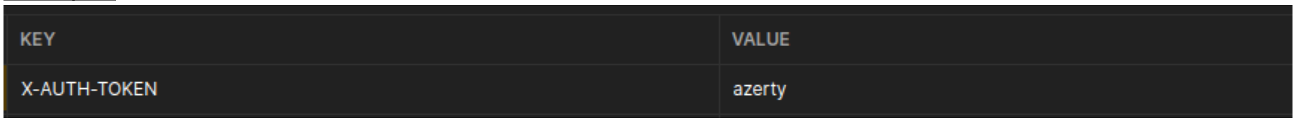
Description :

In order to verify your identity, you will be given an authentication token if you wish to use this means of authentication. Please attach this to all subsequent API calls.

Request:

Include the **X-AUTH-TOKEN** key and its value in the header of your call.

Example:



KEY	VALUE
X-AUTH-TOKEN	azerty

See appendix 9.2 Example of a HMAC cURL call for an example of a complete call.

8.1.2 HMAC generation :

Description :

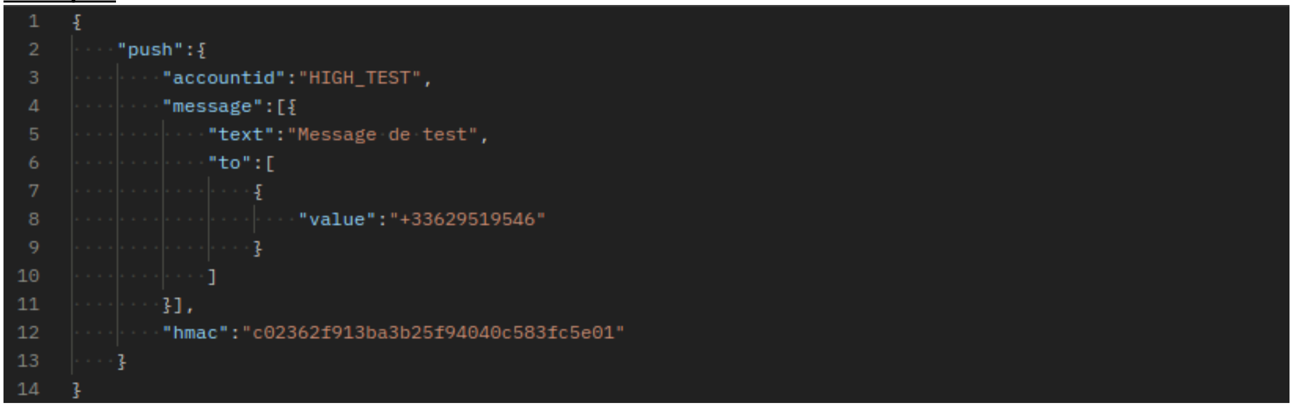
An HMAC is an authentication code calculated using a cryptographic hash function in combination with a secret key (hmac_key). It can be used to simultaneously verify the integrity of the data and the authenticity of a message.

To enable this, an HMAC will be required at the end of each call. This is calculated on the basis of the content of the call + the HMAC key given to you when you create an account.

Query:

Take all the fields from the call, calculate the HMAC with the HMAC key and integrate the result in the hmac field of the body.

Example:



```
{
    "push":{
        "accountid":"HIGH_TEST",
        "message":[{
            "text":"Message de test",
            "to":[
                {
                    "value":"+33629519546"
                }
            ]
        }],
        "hmac":"c02362f913ba3b25f94040c583fc5e01"
    }
}
```

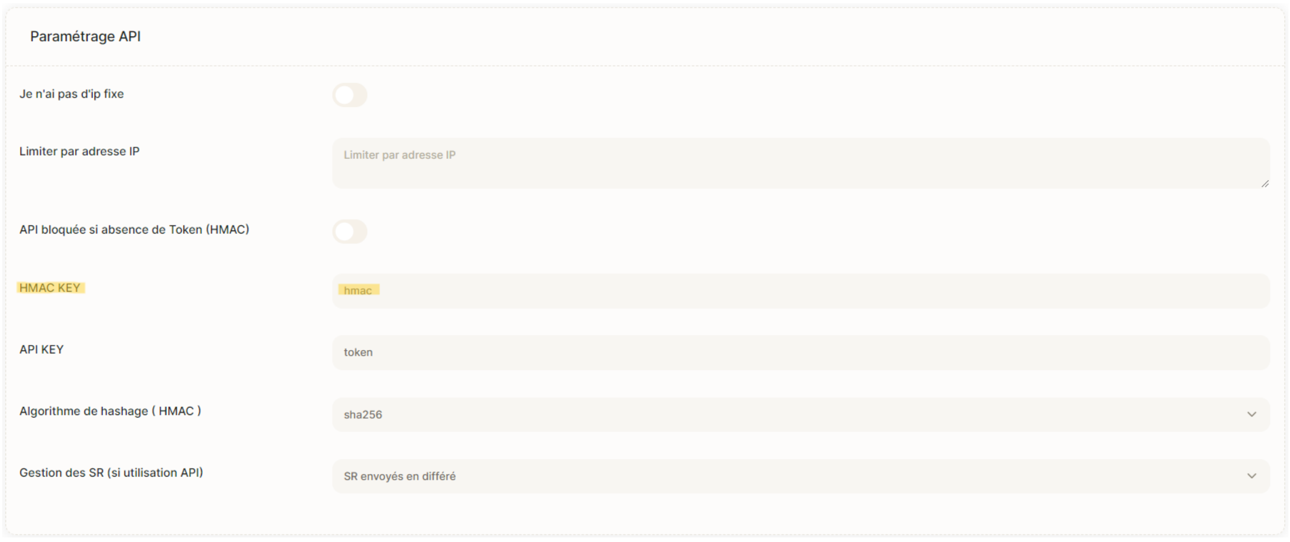
See appendix 9.3 Example of an Hmac calculation script in php for the calculation script.

8.2 Deprecated Method

<u>8.2.1 Description :</u>

An IP security system on the API has been implemented.

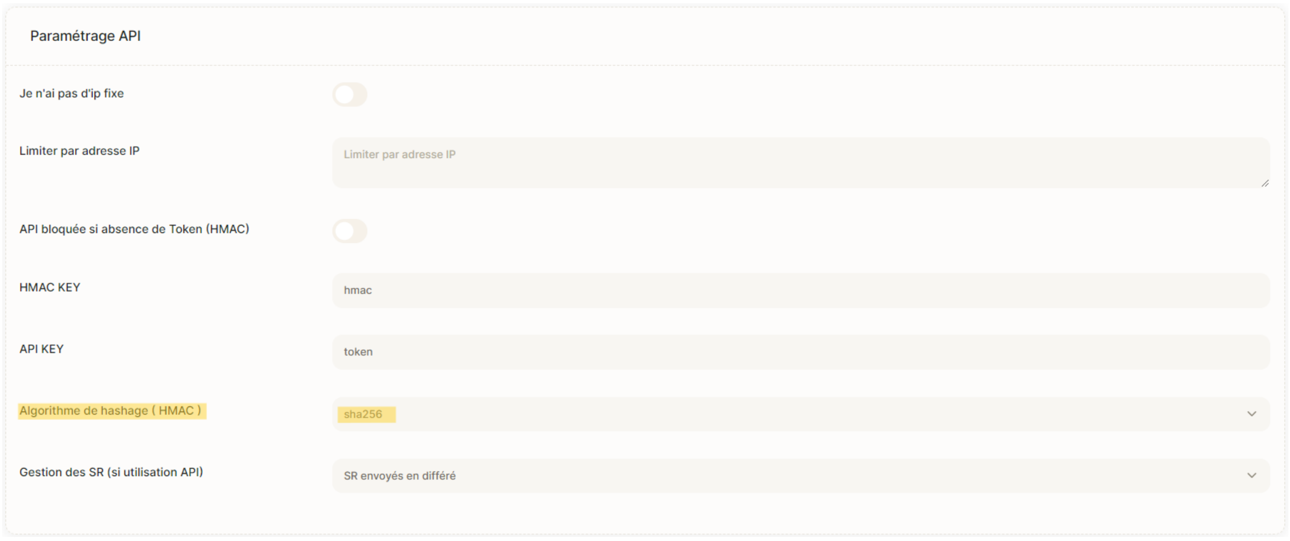
If it cannot be applied (use of rotating IPs), a token security system can be substituted for it (or intervene in addition) for API calls (GET|POST method with XML and JSON format).
This token will be populated with the “hmac” parameter.



Paramétrage API

Je n'ai pas d'ip fixe	
Limiter par adresse IP	Limiter par adresse IP
API bloquée si absence de Token (HMAC)	
HMAC KEY	hmac
API KEY	token
Algorithme de hashage (HMAC)	sha256
Gestion des SR (si utilisation API)	SR envoyés en différé

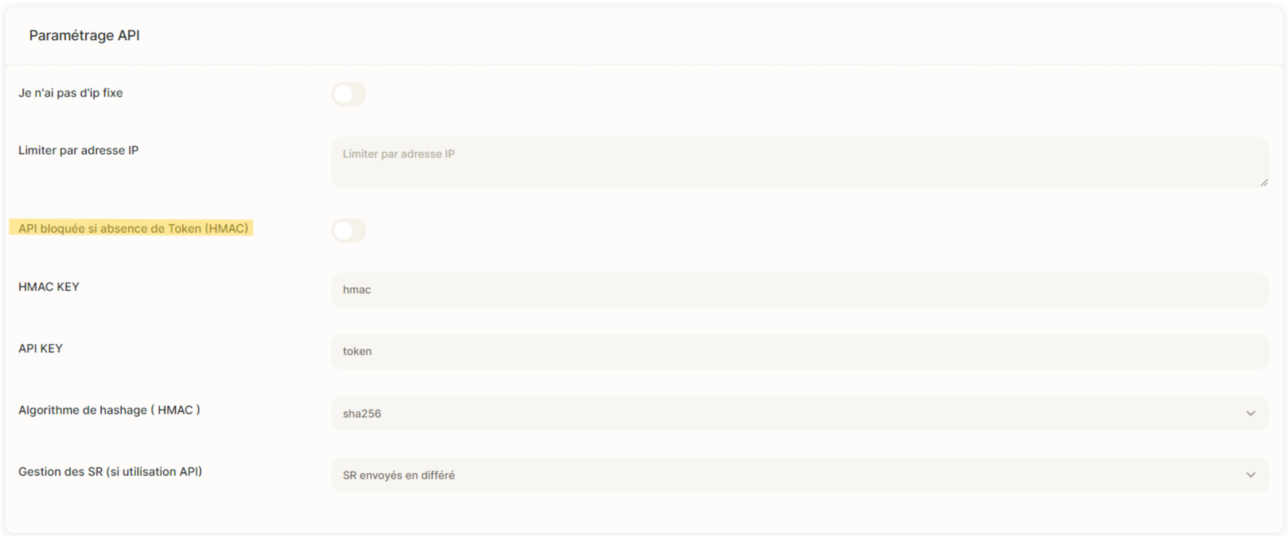
You can choose a hashing algorithm for this HMAC (currently sha256 or md5). By default, the algorithm is sha256.



Paramétrage API

Je n'ai pas d'ip fixe	
Limiter par adresse IP	Limiter par adresse IP
API bloquée si absence de Token (HMAC)	
HMAC KEY	hmac
API KEY	token
Algorithme de hashage (HMAC)	sha256
Gestion des SR (si utilisation API)	SR envoyés en différé

The presence of this “hmac” parameter during an API call can be made mandatory or left optional for testing purposes. This option can also be enabled in the application account configuration.


Paramétrage API

Je n'ai pas d'ip fixe	
Limiter par adresse IP	Limiter par adresse IP
API bloquée si absence de Token (HMAC)	
HMAC KEY	hmac
API KEY	token
Algorithme de hashage (HMAC)	sha256
Gestion des SR (si utilisation API)	SR envoyés en différé

8.2.2 ***Building the HMAC parameter***

To build this « hmac » token, you must :

- Prepare the string to be encoded by concatenating some values of the parameters used when calling the API and the API key (APIKEY) that was provided to you with a separator '&' between each value.
- Use a hash system with the "SHA256" algorithm and the APIKEY as the encoding key to calculate the hmac.
- Add a parameter named « hmac » when calling the API by giving it the calculated value.

An example to generate the « hmac » parameter is given below.

Parameters taken into account to prepare the string to be encoded :

- accountid
- password
- ret_id
- push_id
- ret_url
- ret_mo_url

Example of hmac construction for a call to highpush_stat.php

```
<?php
$DEBUG=0 ;

// Call parameters
$url = https://{your_endpoint}.hcnx.eu/highpush_stat.php';
$ACCOUNTID="HIGH_PUSH";
$PASSWORD= "xpasswordx";
$PUSH_ID="1871011352";

// What is specific to the hmac
$API_KEY="azertyuiop";  // Key given by HighConnexion
$WithHmac=1; // We use the hmac

// Preparing the paramaters
 $body_req = array(
 "accountid" => $ACCOUNTID,
  "password" =>  $PASSWORD,
  "push_id" => $PUSH_ID,
  );
  if ($WithHmac)
     {
     $hmac=CalculHmac($API_KEY,$body_req);
     $body_req['hmac']=$hmac;
     }
// Building the final url (if Get)
if ($method == 'get') {
  $url .= '?'.http_build_query($body_req);
}

// The hmac calculation function
function CalculHmac($apiKey,$params)
   {
   global $DEBUG;
   $keyHmac = array('accountid', 'password','ret_id'  ,'push_id', 'ret_url','ret_mo_url');

   $encode = '';
   foreach($params as $key => $val){
   $trouve=in_array($key, $keyHmac);
              if (in_array($key, $keyHmac)){
                           $encode .= $val."&";
                 }
   }
   $encode .= $apiKey;
   $hmacEncoderSHA256 = hash_hmac("sha256",$encode,$apiKey);
   $DEBUG && print "hash_hmac(sha256,$encode,$apiKey)=$hmacEncoderSHA256\n";
   return $hmacEncoderSHA256;
   }
```

Which gives the url :
https://{your_endpoint}.hcnx.eu/highpush_stat.php?accountid=HIGH_PUSH&password=xpasswordx&push_id=1871011352&hmac=d181b7b30b7fd44724e132318e462a6f094472287de1dc88290ae4d4db59d362

Parameters details :

Name	Mandatory	Example value	Comment
accountid	O	HIGH_PUSH	Value given by HighConnexion.
password	O	xpasswordx	Value given by HighConnexion.
push_id	O si ret_id non défini	1871011352	Internal value of push_id, returned when XML/JSON is provided by an HTTP request.
Hmac		d181b7b30b7fd44724e132318e462a6f094472287de1dc88290ae4d4db59d362	Hash generated in 'sha256' on the concatenation of HIGH_PUSH&xpasswordx&1871011352&azertyuiop with the key azertyuiop for encoding. (Assume that azertyuiop is the value of APIKEY)

POST call example (same principle of calculating hmac)

Description :

Allows sending SMS campaigns.

Request :

- Method : POST
- POST data in JSON :

- HMAC : Hash generated in ‘sha256’ on the concatenation of HIGH_TEST& xpasswordx&Push_Mondor_324&https://{your_endpoint}.hcnx.eu RetrieveStatus.php&azertyuiop

```
{
        "push": {
                "accountid": "HIGH_TEST",
                "password": "xpasswordx",
                "start_date": "2017-03-08",
                "start_time": "17:25",
                "userdata": "MONDOR_Cardio",
                "sender": "MONDOR",
                "ret_id": "Push_Mondor_324",
                "ret_url": "https://highpush-v50.hcnx.eu/RetrieveStatus.php",
                “hmac”: “4a2840cead3c1b701c6a97be8d8c38276719954d8b45aac6282e102a87498”
                "priority": 2,
                "ret_mo_url": "",
                "noone": false,
                "nb_sms": 1,
                "truncature": 1,
                "message": [
                        {
                                "text": "Bonjour %NAME%, votre rendez-vous du %DATE% est confirmé",
                                "to": [
                                        {
                                                "value": "+33623456789",
                                                "ret_id": "Mess4563",
                                                "param": [
                                                        {
                                                                "var": "%NAME%",
                                                                "value": "DUPONT"
                                                        },
                                                        {
                                                                "var": "%DATE%",
                                                                "value": "Jeudi 9 mars 2017, 8h30"
                                                        }

                                                ]
                                        },
                                        {
                                                "value": "+33623475352"
                                                "ret_id": "Mess4564",
                                                "param": [
                                                        {
                                                                "var": "%NAME%",
                                                                "value": "DURAND"
                                                        },
                                                        {
                                                                "var": "%DATE%",
                                                                "value": "Jeudi 9 mars 2017, 8h30"
                                                        }

                                                ]
                                        }
                                ],

                        }
                ]
        }
}
```

Response :

- If an error occurred :

```
{
        "status": "error"
        "detail": "ERROR_TYPE"
}
```

- If a token error occurred :

```
{
        "status": "error"
        "detail": “Error has occurred : Hmac non valide”
}
```

- Otherwise :

```
{
        "status": "success",
        "data": {
                "id": 1871016995
        }
}
```

9 Appendix

9.1 Examples

9.1.1 Complex message in Unicode :

We would like to send this personalised message:
こんにちは %FIRSTNAME% 私たちのサイトを見に来てください: %URL%
Which means in French: Bonjour %PRENOM% viens voir notre site: %URL%.

```
{
  "push:{
    "accountid": "account",
    "password": "password",
    "message":[{
      "target_url": "www.highconnexion.com?ville=%VILLE%",

"binary":"30533093306b3061306f002000250050005200045004e004f004d0025002079c1305f3061306e30b
530a430c83092898b306b67653066304f306030553044003a00200025005500520004c0025",
      "to":[{
          "ret_id": "ret_id1",
          "value":"+33666666666",
          "param": [{
             "var": "002500500052000450004e004f004d0025", // %PRENOM% in unicode
             "value": "0050006100730063006100 6c" // Pascal in unicode
            },
            {
              "var": "%CITY%",
              "value": "Lyon
            }]
        }]
    }]
  }
}
```

This will produce the following message:
こんにちは Pascal 私たちのサイトを見に来てください: zi.hcnx.fr/TiWmRI

And the url to which the tiny url redirects :
https://www.highconnexion.com/?ville=Lyon

9.1.2 How to build an XML

While the example shown above is very simple, the XML format has been designed to drive large SMS campaigns with many parameters.

Below is a more complete example:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<push
accountid="fred"
password="xpasswordx"
userdata="User Data Multiple Sent"
ret_id="Push_1"
sender="API_HIGHSMS"
>
<message>
<text>Helllo world Multiple</text>
<to>0619896895</to>
<to>0699999999</to>
</message>
</push>
```

9.1.3 POST call with XML variable

```
<?php
 #My SMS settings
  $accountid='fred';
  $password='xpasswordx';
  $urlapi='https://highsms-hcnx.hcnx.eu /api';
  $ret_url='http://myaddress.com/RetrieveStatus.php';
  $ret_mo_url='http://myaddress.com/sample/TreatMO.php';

  $to='0619896895';

 # If the Sender is not set or is empty, the ShortCode (36105 in our example) is used.
 # $sender='MYTEST';

 #If I position this identifier, I get it back from SR and MO
  $suryident='MYTEST_123';
  $text='Send your answer to 36105';


 #The making of XML

 $xml= <<<EOXML
 <?xml version="1.0" encoding="utf-8"?>
 <push
 accountid="$accountid"
 password="$password"
 ret_url="$ret_url"
 ret_mo_url="$ret_mo_url"
 sender="$sender"
 ret_id="$suryident"
```

```
>
<message>
<text>$text</text>
<to>$to
</to>
</message>
</push>
```

9.1.4 Recovery of SR and MO

[DEPRECATED]

Example of RetrieveStatus.php source code

```
# cat RetrieveStatus.php
<?php
$req_dump = print_r($_REQUEST, TRUE);
$fp = fopen('/tmp/request.log', 'a');
fwrite($fp, "RetrieveStatus:\n");
fwrite($fp, $req_dump);
fclose($fp);
```

Example of TreatMO.php source code

```
# cat TreatMO.php
<?php
$req_dump = print_r($_REQUEST, TRUE);
$fp = fopen('/tmp/request.log', 'a');
fwrite($fp, "TreatMO:\n");
fwrite($fp, $req_dump);
fclose($fp);
```

Example after execution

After answering "Thanks but summer sucks" on 36105

```
# cat /tmp/request.log
RetrieveStatus:
Array
(
   [push_id] => 568300865
[status] => 6
   [to] => +33619896895
[ret_id] =>
   [text] => The weather was fine this summer
)

TreatMO:
Array
(
   [FROM] => +33619896895
   [ID] => 568442694
   [TO] => 36105
[MESSAGE] =>Thanks but a rotten summer
   [VALIDITY_DATE] => 2014-12-19T17:48:16
[GET_STATUS] => 0
   [CLIENT] => HIGHPUSH
   [CLASS_TYPE] => 0
   [RECEPTION_DATE] => 2014-12-18T17:48:16
   [TO_OP_ID] => 20810
   [INITIAL_OP_ID] => 20810
   [STATUS] => POSTING_18327_4807
   [EMAIL] =>
   [BINARY] => 0
   [PARAM] => 33609003663
   [USER_DATA] => HIGHPUSH
[USER_DATA_2] =>Thanks but a rotten summer
[BULK_ID] => 0
   [MO_ID] => 0
   [APPLICATION_ID] => 0
   [ACCOUNT_ID] => 39
   [GW_MESSAGE_ID] => 0
   [READ_STATUS] => 0
   [TARIFF] => 0
   [REQUEST_ID] => 33609003663
   [TAC] => (null)
   [REASON] => 2014-12-18 18:48:15
   [FORMAT] =>
   [MVNO] =>
   [DATE] => 2014-12-18T17:48:16
   [ORIG_ID] => 568300865
   [ORIG_MESSAGE] => The weather was fine this summer.
   [ORIG_DATE] => 2014-12-18 17:48:16
   [RET_ID] =>
)
```

9.1.5 Webhook content

MT taken into account by HCNX following submission via the API

```
{
        "event": {
                "event_status": "MT",
                "event_http_id": 56,
                "event_date" : "2022-04-10 10:30:22"
        },
"sms    ": {
                "sms_hcnx_id" : "1234567890", // MT_ID
                "sms_client_id": "123e4567-e89b-12d3-a456-426655440000",
                "sms_client_campaign_id": "ABC123",
                "sms_msisdn": "+33635393605",
                "sms_status": "CREATED",
                "sms_message": "Hello Thibault, register here: http://a.hcnx.eu/ABC123",
                "sms_sender": "HCNX",
                "sms_response": "",
                "sms_user_data": "client_customer_id",
                "sms_counter": 1,
                "sms_operator": ""
        }
}
```

MT submitted to the operator by HCNX

```
{
        "event": {
                "event_status": "MT",
                "event_http_id": 56,
                "event_date" : "2022-04-10 10:30:23"
        },
"sms    ": {
                "sms_hcnx_id" : "123456789",  // MT_ID
                "sms_client_id": "123e4567-e89b-12d3-a456-426655440000",
                "sms_client_campaign_id": "ABC123",
                "sms_msisdn": "+33635393605",
                "sms_status": "SENT",    // SENT or ERROR : {reason_error}
                "sms_message": "Hello Thibault, register here: http://a.hcnx.eu/ABC123",
                "sms_sender": "HCNX",
                "sms_response": "",
                "sms_user_data": "client_customer_id",
                "sms_counter": 1,
                "sms_operator": "20801
        }
}
```

DLR received from the operator

```
{
        "event": {
                "event_status": "DLR",
                "event_http_id": 56,
                "event_date" : "2022-04-10 10:30:25"
        },
"sms    ": {
                "sms_hcnx_id": "123456789",
                "sms_client_id": "123e4567-e89b-12d3-a456-426655440000",
                "sms_client_campaign_id": "ABC123",
                "sms_msisdn": "+33635393605",
                "sms_status" : "RECEIVED", // RECEIVED or ERROR : {reason_error}
                "sms_message": "Hello Thibault, register here: http://a.hcnx.eu/ABC123",
                "sms_sender": "HCNX",
                "sms_response": "",
                "sms_user_data": "client_customer_id",
                "sms_counter": 1,
                "sms_operator": "20801
        }
}
```

MO received

```
{
        "event": {
                "event_status": "MO",
                "event_http_id": 56,
                "event_date" : "2022-04-10 10:30:30"
        },
"sms     ": {
                "sms_hcnx_id": "123456789",
                "sms_client_id": "123e4567-e89b-12d3-a456-426655440000",
                "sms_client_campaign_id": "ABC123",
                "sms_msisdn": "+33635393605",
                "sms_status": "RECEIVED",
                "sms_message": "Hello Thibault, register here: http://a.hcnx.eu/ABC123",
                "sms_sender": "HCNX",
                "sms_response": "STOP",
                                                            // MO message
                "sms_user_data": "client_customer_id",
                "sms_counter": 1,
                "sms_operator": "20801
        }
}
```

Click

```
{
        "event": {
                "event_status": "CLICK",
                "event_http_id": 56,
                "event_date" : "2022-04-10 10:30:35"
        },
"sms     ": {
                "sms_hcnx_id": "123456789",
                "sms_client_id": "123e4567-e89b-12d3-a456-426655440000",
                "sms_client_campaign_id": "ABC123",
                "sms_msisdn": "+33635393605",
                "sms_status": "RECEIVD",
                "sms_message": "Hello Thibault, register here: http://a.hcnx.eu/ABC123",
                "sms_sender": "HCNX",
                "sms_response": "",
                "sms_user_data": "client_customer_id",
                "sms_counter": 1,
                "sms_operator": ""
        },
        "url" : {
                "short_url" : "http://a.hcnx.eu/ABC123",
                                                            // Short Url compiled
                "target_url": "http://google.com?firstname=Thibault&id=123e4567-e89b-12d3-a456-
426655440000"        // Compiled     target url
        }
}
```

9.2 Example of a cURL HMAC call :

```
curl --location --request POST 'http://highsms-hcnx.hcnx.eu/campaign
--header 'X-AUTH-TOKEN: azerty \
--header 'Content-Type: application/json' \
--data-raw '{
  "push:{
    "accountid": "HIGH_TEST",
    "message":[{
      "text": "Message de test",
      "to":[
        {
          "value":"+33629519546"
        }
      ]
    }],
    "hmac":"c02362f913ba3b25f94040c583fc5e01"
  }
}
```

9.3 Example of an Hmac calculation script in php :

An hmac_key is associated with the customer account and then transmitted to the customer in order to perform the same calculation on both sides and check the integrity of the data transmitted.

Example:

```
<?php

function hmac_calculate($array, $hmac_key = ''){
    // Prevent the parameters for calculating the HmacService from including an HmacService...
    if (isset($array['hmac'])) {
      unset($array['hmac']);
    }

    // Depending on the api called
    if (isset($parameters['push']['hmac'])) {
      unset($parameters['push']['hmac']);
    }

    // Algo
    //ksort($array);

    return hash_hmac('md5', http_build_query($array), $hmac_key);
  }
```

9.4 Character table

Authorised characters in message content :
With the exception of the LINE FEED and CARRIAGE RETURN characters, all the characters in the ISO-8859-1 table that are also part of the ETSI GSM 03.38 table can be used directly.
Link to the ETSI GMS 03.38 table: https://en.wikipedia.org/wiki/GSM_03.38

Details

- Carriage returns must be represented by a 127 (0x7F) character;
- The EURO SIGN must be represented by character 128 (0x80) or 164 (0xA4) ;
- The ETSI GSM 03.38 characters that do not exist in the ISO-8859-1 table have no smstext representation and therefore cannot be used;
- ISO-8859-1 characters that do not exist in ETSI GSM 03.38 will be converted to the nearest character (e.g. ç will be replaced by c). Details are given in the table below.
- **The characters in the GSM 03.38 extension table: ^ { } \ [~] | € count as two characters in the 160 characters allowed in a message.**

Specificity of long SMS messages :
For a long text message, i.e. one that exceeds 160 characters, the basis on which you should count the number of text messages that make up your message is 153 rather than 160. This is because the equivalent of 7 characters are used to make the junction between each SMS.
Examples:
A 160-character message counts as 1 text message
A 306-character message counts as 2 SMS messages (153+153)
A 320-character message counts as 3 SMS messages (153+153+14)

The following table indicates for each character in the ISO-8859-1 table the character in the GSM 03.38 table used to represent it:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	N/A	<SH>	<SX>	<EX>	<ET>	<EQ>	<AK>	<BL>	<BS>	<HT>	<LF>	<VT>	<FF>	<CR>	<SO>	<SI>
	N/A	?	?	?	?	?	?	?	?		<LF>	?	<FF>	<CR>	?	?
1x	<DL>	<D1>	<D2>	<D3>	<D4>	<NK>	<SY>	<EB>	<CN>	<EM>	<SB>	<EC>	<FS>	<GS>	<RS>	<US>
	?	?	?	?	?	?	?	?	?	?	?	<EC>	?	?	?	?
2x		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	?	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	<DT>
	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	<LF>
8x	<PA>	<HO>	<BH>	<NH>	<IN>	<NL>	<SA>	<ES>	<HS>	<HJ>	<VS>	<PD>	<PU>	<RI>	<S2>	<S3>
	<PA>	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
9x	<DC>	<P1>	<P2>	<TS>	<CC>	<MW>	<SG>	<EG>	<SS>	<GC>	<SC>	<CI>	<ST>	<OC>	<PM>	<AC>

	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Ax	<NS>	¡	¢	£	¤	¥	¦	§	¨	©	ª	"	¬		®	¯
		¡	c	£	¤	¥	¡	§	"	C	a	<	!	-	R	-
Bx	°	±	²	³	'	µ	¶	-	¸	¹	º	"	¼	½	¾	¿
	o	?	2	3	'	u	?	.	,	i	o	>	?	?	?	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
	A	A	A	A	Ä	Å	Æ	Ç	E	É	E	E	I	I	I	I
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
	?	Ñ	O	O	O	O	Ö	x	Ø	U	U	U	Ü	Y	?	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
	à	a	a	a	ä	å	æ	c	è	é	e	e	ì	i	i	i
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ
	?	ñ	ò	o	o	o	ö	?	ø	ù	u	u	ü	y	?	y

And here is the complete GSM 03.38 table:

Hexa	Decimal	Name	Character	Decimal in smstext
0x00	0	COMMERCIAL AT	@	64
0x01	1	POUND SIGN	£	163
0x02	2	DOLLAR SIGN	$	36
0x03	3	YEN SIGN	¥	165
0x04	4	LATIN SMALL LETTER E WITH GRAVE	è	232
0x05	5	LATIN SMALL LETTER E WITH ACUTE	é	233
0x06	6	LATIN SMALL LETTER U WITH GRAVE	ú	250
0x07	7	LATIN SMALL LETTER I WITH GRAVE	ì	236
0x08	8	LATIN SMALL LETTER O WITH GRAVE	ò	242
0x09	9	LATIN CAPITAL LETTER C WITH CEDILLA	Ç	199
0x0A	10	LINE FEED		127
0x0B	11	LATIN CAPITAL LETTER O WITH STROKE	Ø	216
0x0C	12	LATIN SMALL LETTER O WITH STROKE	ø	248
0x0D	13	CARRIAGE RETURN		127
0x0E	14	LATIN CAPITAL LETTER A WITH RING ABOVE	Å	197
0x0F	15	LATIN SMALL LETTER A WITH RING ABOVE	å	229
0x10	16	GREEK CAPITAL LETTER DELTA	Δ	
0x11	17	LOW LINE	_	95
0x12	18	GREEK CAPITAL LETTER PHI	Φ	
0x13	19	GREEK CAPITAL LETTER GAMMA	Γ	
0x14	20	GREEK CAPITAL LETTER LAMBDA	Λ	
0x15	21	GREEK CAPITAL LETTER OMEGA	Ω	
0x16	22	GREEK CAPITAL LETTER PI	Π	
0x17	23	GREEK CAPITAL LETTER PSI	Ψ	
0x18	24	GREEK CAPITAL LETTER SIGMA	Σ	

0x19	25	GREEK CAPITAL LETTER THETA	Θ	
0x1A	26	GREEK CAPITAL LETTER XI	Ξ	
0x1B	27	ESCAPE TO EXTENSION TABLE		
0x1B0A	27 10	FORM FEED		12
0x1B14	27 20	CIRCUMFLEX ACCENT	^	94
0x1B28	27 40	LEFT CURLY BRACKET	{	123
0x1B29	27 41	RIGHT CURLY BRACKET	}	125
0x1B2F	27 47	REVERSE SOLIDUS (BACKSLASH)	\	92
0x1B3C	27 60	LEFT SQUARE BRACKET	[	91
0x1B3D	27 61	TILDE	~	126
0x1B3E	27 62	RIGHT SQUARE BRACKET	]	93
0x1B40	27 64	VERTICAL BAR	|	124
0x1B65	27 101	EURO SIGN	€	27 101
0x1C	28	LATIN CAPITAL LETTER AE	Æ	198
0x1D	29	LATIN SMALL LETTER AE	æ	230
0x1E	30	LATIN SMALL LETTER SHARP S (German)	ß	223
0x1F	31	LATIN CAPITAL LETTER E WITH ACUTE		202
0x20	32	SPACE		32
0x21	33	EXCLAMATION MARK	!	33
0x22	34	MARK QUOTATION	"	34
0x23	35	NUMBER SIGN	#	35
0x24	36	CURRENCY SIGN	¤	164
0x25	37	PERCENT SIGN	%	37
0x26	38	AMPERSAND	&	38
0x27	39	APOSTROPHE	'	39
0x28	40	LEFT PARENTHESIS	(	40
0x29	41	RIGHT PARENTHESIS	)	41
0x2A	42	ASTERISK	*	42
0x2B	43	PLUS SIGN	+	43
0x2C	44	COMMA	,	44
0x2D	45	HYPHEN-MINUS	-	45
0x2E	46	FULL STOP	.	46
0x2F	47	SOLIDUS (SLASH)	/	47
0x30	48	DIGIT ZERO	0	48
0x31	49	DIGIT ONE	1	49
0x32	50	DIGIT TWO	2	50
0x33	51	DIGIT THREE	3	51
0x34	52	DIGIT FOUR	4	52
0x35	53	DIGIT FIVE	5	53
0x36	54	DIGIT SIX	6	54
0x37	55	DIGIT SEVEN	7	55
0x38	56	DIGIT EIGHT	8	56
0x39	57	DIGIT NINE	9	57
0x3A	58	COLON	:	58
0x3B	59	SEMICOLON	;	59
0x3C	60	LESS-THAN SIGN	<	60

0x3D	61	EQUALS SIGN	=	61
0x3E	62	GREATER-THAN SIGN	>	62
0x3F	63	QUESTION MARK	?	63
0x40	64	INVERTED EXCLAMATION MARK	¡	161
0x41	65	LATIN CAPITAL LETTER A	A	65
0x42	66	LATIN CAPITAL LETTER B	B	66
0x43	67	LATIN CAPITAL LETTER C	C	67
0x44	68	LATIN CAPITAL LETTER D	D	68
0x45	69	LATIN CAPITAL LETTER E	E	69
0x46	70	LATIN CAPITAL LETTER F	F	70
0x47	71	LATIN CAPITAL LETTER G	G	71
0x48	72	LATIN CAPITAL LETTER H	H	72
0x49	73	LATIN CAPITAL LETTER I	I	73
0x4A	74	LATIN CAPITAL LETTER J	J	74
0x4B	75	LATIN CAPITAL LETTER K	K	75
0x4C	76	LATIN CAPITAL LETTER L	L	76
0x4D	77	LATIN CAPITAL LETTER M	M	77
0x4E	78	LATIN CAPITAL LETTER N	N	78
0x4F	79	LATIN CAPITAL LETTER O	O	79
0x50	80	LATIN CAPITAL LETTER P	P	80
0x51	81	LATIN CAPITAL LETTER Q	Q	81
0x52	82	LATIN CAPITAL LETTER R	R	82
0x53	83	LATIN CAPITAL LETTER S	S	83
0x54	84	LATIN CAPITAL LETTER T	T	84
0x55	85	LATIN CAPITAL LETTER U	U	85
0x56	86	LATIN CAPITAL LETTER V	V	86
0x57	87	LATIN CAPITAL LETTER W	W	87
0x58	88	LATIN CAPITAL LETTER X	X	88
0x59	89	LATIN CAPITAL LETTER Y	Y	89
0x5A	90	LATIN CAPITAL LETTER Z	Z	90
0x5B	91	LATIN CAPITAL LETTER A WITH DIAERESIS	Ä	196
0x5C	92	LATIN CAPITAL LETTER O WITH DIAERESIS	Ö	214
0x5D	93	LATIN CAPITAL LETTER N WITH TILDE	Ñ	209
0x5E	94	LATIN CAPITAL LETTER U WITH DIAERESIS	Ü	220
0x5F	95	SIGN SECTION	§	167
0x60	96	INVERTED QUESTION MARK	¿	191
0x61	97	LATIN SMALL LETTER A	a	97
0x62	98	LATIN SMALL LETTER B	b	98
0x63	99	LATIN SMALL LETTER C	c	99
0x64	100	LATIN SMALL LETTER D	d	100
0x65	101	LATIN SMALL LETTER E	e	101
0x66	102	LATIN SMALL LETTER F	f	102
0x67	103	LATIN SMALL LETTER G	g	103
0x68	104	LATIN SMALL LETTER H	h	104
0x69	105	LATIN SMALL LETTER I	i	105
0x6A	106	LATIN SMALL LETTER J	j	106

0x6B	107	LATIN SMALL LETTER K	k	107
0x6C	108	LATIN SMALL LETTER L	l	108
0x6D	109	LATIN SMALL LETTER M	m	109
0x6E	110	LATIN SMALL LETTER N	n	110
0x6F	111	LATIN SMALL LETTER O	o	111
0x70	112	LATIN SMALL LETTER P	p	112
0x71	113	LATIN SMALL LETTER Q	q	113
0x72	114	LATIN SMALL LETTER R	r	114
0x73	115	LATIN SMALL LETTER S	s	115
0x74	116	LATIN SMALL LETTER T	t	116
0x75	117	LATIN SMALL LETTER U	u	117
0x76	118	LATIN SMALL LETTER V	v	118
0x77	119	LATIN SMALL LETTER W	w	119
0x78	120	LATIN SMALL LETTER X	x	120
0x79	121	LATIN SMALL LETTER Y	y	121
0x7A	122	LATIN SMALL LETTER Z	z	122
0x7B	123	LATIN SMALL LETTER A WITH DIAERESIS	ä	228
0x7C	124	LATIN SMALL LETTER O WITH DIAERESIS	ö	246
0x7D	125	LATIN SMALL LETTER N WITH TILDE	ñ	241
0x7E	126	LATIN SMALL LETTER U WITH DIAERESIS	ü	252
0x7F	127	LATIN SMALL LETTER A WITH GRAVE	à	224
0x80	128	EURO SIGN	€	
0xA4	164	EURO SIGN	€	

9.5 XML errors

Here is a list of XML errors that may be encountered:

XML_ERROR_NONE
XML_ERROR_NO_MEMORY
XML_ERROR_SYNTAX
XML_ERROR_NO_ELEMENTS
XML_ERROR_INVALID_TOKEN
XML_ERROR_UNCLOSED_TOKEN
XML_ERROR_PARTIAL_CHAR
XML_ERROR_TAG_MISMATCH
XML_ERROR_DUPLICATE_ATTRIBUTE
XML_ERROR_JUNK_AFTER_DOC_ELEMENT
XML_ERROR_PARAM_ENTITY_REF
XML_ERROR_UNDEFINED_ENTITY
XML_ERROR_RECURSIVE_ENTITY_REF
XML_ERROR_ASYNC_ENTITY
XML_ERROR_BAD_CHAR_REF
XML_ERROR_BINARY_ENTITY_REF
XML_ERROR_ATTRIBUTE_EXTERNAL_ENTITY_REF
XML_ERROR_MISPLACED_XML_PI
XML_ERROR_UNKNOWN_ENCODING
XML_ERROR_INCORRECT_ENCODING
XML_ERROR_UNCLOSED_CDATA_SECTION
XML_ERROR_EXTERNAL_ENTITY_HANDLING

9.6 List of status codes

Here is the list of statuses that can be returned to you:

NOT_SENT=>1 (sent to the router, not yet sent to the operator)
ERROR=>2 (unable to contact the router)
QUEUE=>3 (taken into account by the router)
SENT=>4 (sent to the operator)
RECEIVED=>6 (received by recipient)
ERROR_NPAI=>11 ("N'habites Pas à l'Adresse Indiquée": the operator contacted does not know or no longer knows this number)
ERROR_EXPIRED=>12 (No reception of RECEIVED after SENT after a certain time, usually 24 hours)
ERROR_INVOP=>13 (Invalid operator, number not attached to an operator slot)
ERROR_NETWORK=>14 (Network problem during routing)
ERROR_CREDIT=>15 (Insufficient credit)
ERROR_UNKNOWN=>16 (Other errors, ...)
ERROR_BLOCKED=>23 (Recipient blocked by operator)
ERROR_NPAIPORTED=>25